User's Manual

16

RL78/G23

User's Manual: Hardware

16-Bit Single-Chip Microcontrollers

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

Renesas Electronics
www.renesas.com

Rev.2.00 Jun 2026

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

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/

Trademarks

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

© 2026 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/G23 and design and develop application systems and programs for these devices.
The target products are as follows.

<R7F100G>

- 30-pin: R7F100GAx (x = F, G, H, J)
- 32-pin: R7F100GBx (x = F, G, H, J)
- 36-pin: R7F100GCx (x = F, G, H, J)
- 40-pin: R7F100GEx (x = F, G, H, J)
- 44-pin: R7F100GFx (x = F, G, H, J, K, L, N)
- 48-pin: R7F100GGx (x = F, G, H, J, K, L, N)
- 52-pin: R7F100GJx (x = F, G, H, J, K, L, N)
- 64-pin: R7F100GLx (x = F, G, H, J, K, L, N)
- 80-pin: R7F100GMx (x = G, H, J, K, L, N)
- 100-pin: R7F100GPx (x = G, H, J, K, L, N)
- 128-pin: R7F100GSx (x = J, K, L, N)

<R7F104G>

- 32-pin: R7F104GBG
- 44-pin: R7F104GFJ
- 48-pin: R7F104GGJ

Purpose

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

Organization

The RL78/G23 manual is separated into two parts: this manual and User's Manual: Software (common to the RL78 family).

RL78/G23 User's Manual: Hardware (This Manual)	**RL78 Family User's Manual: Software**
• Pin functions • Internal block functions • Interrupts • Other on-chip peripheral functions • Electrical characteristics	• 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**. The mark “<R>” shows major revised points. The revised points can be easily searched by copying an “<R>” in the PDF file and specifying it in the “Find what:” field.
- 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 instructions:
  - → Refer to the separate document **RL78 Family User’s Manual: Software (R01US0015E).**

Conventions

Data significance: Higher digits on the left and lower digits on the right

Active low representations: $\overline{\times\times\times}$ (overscore over pin and signal name)

Note: Footnote for item marked with Note in the text

Caution: Information requiring particular attention

Remark: Supplementary information

Numerical representations: Binary numbers are represented as xxxx or xxxxB, where each x is 0 or 1.

Decimal numbers are represented as xxxx, where each x is a numeral from 0 to 9.

Hexadecimal numbers are represented as xxxxH, where each x is a number from 0 to 9 or a letter from A to F.

Related Documents

Documents Related to Devices

Document Name	Document No.
RL78/G23 User's Manual: Hardware	This manual
RL78 Family User's Manual: Software	R01US0015E

Documents Related to Flash Memory Programming and On-chip Debugging

Document Name	Document No.
PG-FP6 Flash Memory Programmer User's Manual	**Note 1**
E2 Emulator User's Manual	R20UT3538E
E2 Emulator Lite RTE0T0002LKCE00000R User's Manual	R20UT3240E
Renesas Flash Programmer Flash Memory Programming Software User's Manual	**Note 2**

Note 1. For a list of the documents relevant to the PG-FP6, visit the Web page below.
https://www.renesas.com/us/en/software-tool/pg-fp6

Note 2. For a list of the documents relevant to the Renesas Flash Programmer, visit the Web page below.
https://www.renesas.com/us/en/software-tool/renesas-flash-programmer-programming-gui

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

Other Documents

Document Name	Document No.
Renesas Microcontrollers RL78 Family	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

1. Outline 1
1.1 Differences between R7F100G and R7F104G Products 1
1.2 Features 3
1.3 List of Part Numbers 7
1.4 Pin Configuration (Top View) 11
1.4.1 30-pin R7F100G products 11
1.4.2 32-pin R7F100G products 14
1.4.3 36-pin R7F100G products 17
1.4.4 40-pin R7F100G products 20
1.4.5 44-pin R7F100G products 23
1.4.6 48-pin R7F100G products 26
1.4.7 52-pin R7F100G products 29
1.4.8 64-pin R7F100G products 32
1.4.9 80-pin R7F100G products 41
1.4.10 100-pin R7F100G products 45
1.4.11 128-pin R7F100G products 55
1.4.12 32-pin R7F104G product 61
1.4.13 44-pin R7F104G product 64
1.4.14 48-pin R7F104G product 67
1.5 Pin Identification 70
1.6 Block Diagram 71
1.7 Outline of Functions 72

2. Pin Functions 81
2.1 Functions of Port Pins 81
2.1.1 R7F100G products with 30 pins 82
2.1.2 R7F100G products with 32 pins 84
2.1.3 R7F100G products with 36 pins 86
2.1.4 R7F100G products with 40 pins 88
2.1.5 R7F100G products with 44 pins 90
2.1.6 R7F100G products with 48 pins 92
2.1.7 R7F100G products with 52 pins 94
2.1.8 R7F100G products with 64 pins 97
2.1.9 R7F100G products with 80 pins 100
2.1.10 R7F100G products with 100 pins 103
2.1.11 R7F100G products with 128 pins 107
2.1.12 R7F104G product with 32 pins 111
2.1.13 R7F104G product with 44 pins 113
2.1.14 R7F104G product with 48 pins 115
2.2 Non-GPIO Pin Functions 117
2.2.1 Non-GPIO pin functions of the R7F100G products per number of pins 117
2.2.2 Non-GPIO pin functions of the R7F104G products per number of pins 124
2.2.3 Description of non-GPIO pin functions 128
2.2.4 VBAT pin 130

2.3 Connection of Unused Pins 132
2.4 Block Diagrams of Pins 133

3. CPU Architecture 170
3.1 Memory Space 171
3.1.1 Internal program memory space 181
3.1.2 Mirror area 185
3.1.3 Internal data memory space 187
3.1.4 Special function register (SFR) area 188
3.1.5 Extended special function register (2nd SFR: 2nd special function register) area 188
3.1.6 Data memory addressing 189
3.2 Processor Registers 190
3.2.1 Control registers 190
3.2.2 General-purpose registers 192
3.2.3 ES and CS registers 193
3.2.4 Special function registers (SFRs) 194
3.2.5 Extended special function registers (2nd SFRs) 199
3.3 Instruction Address Addressing 215
3.3.1 Relative addressing 215
3.3.2 Immediate addressing 215
3.3.3 Table indirect addressing 216
3.3.4 Register indirect addressing 217
3.4 Types of Addressing for Processing Data Addresses 218
3.4.1 Implied addressing 218
3.4.2 Register addressing 219
3.4.3 Direct addressing 220
3.4.4 Short direct addressing 221
3.4.5 SFR addressing 222
3.4.6 Register indirect addressing 223
3.4.7 Based addressing 224
3.4.8 Based indexed addressing 227
3.4.9 Stack addressing 228

4. Port Functions 231
4.1 Port Functions 231
4.2 Port Configuration 231
4.2.1 Port 0 233
4.2.2 Port 1 233
4.2.3 Port 2 234
4.2.4 Port 3 234
4.2.5 Port 4 234
4.2.6 Port 5 235
4.2.7 Port 6 235
4.2.8 Port 7 235
4.2.9 Port 8 236
4.2.10 Port 9 236
4.2.11 Port 10 236
4.2.12 Port 11 236
4.2.13 Port 12 237

4.2.14 Port 13 237
4.2.15 Port 14 238
4.2.16 Port 15 238
4.3 Registers for Controlling the Port Function 239
4.3.1 Port mode registers (PMxx) 251
4.3.2 Port registers (Pxx) 254
4.3.3 Pull-up resistor option registers (PUxx) 255
4.3.4 Port input mode registers (PIMxx) 256
4.3.5 Port output mode registers (POMxx) 257
4.3.6 Port digital input disable registers (PDIDISxx) 258
4.3.7 Port mode control A registers (PMCAxx) 259
4.3.8 Port mode control T registers (PMCTxx) 262
4.3.9 Port mode control E registers (PMCExx) 263
4.3.10 Peripheral I/O redirection register (PIOR) 264
4.3.11 Global digital input disable register (GDIDIS) 268
4.3.12 Output current control enable register (CCDE) 269
4.3.13 Output current select registers (CCSx) 271
4.3.14 40-mA port output control register (PTDC) 272
4.3.15 Port function output enable registers (PFOEx) 274
4.3.16 Port mode select register (PMS) 275
4.4 Port Function Operations 276
4.4.1 Writing to I/O port 276
4.4.2 Reading from I/O port 276
4.4.3 Operations on I/O port 276
4.4.4 Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching the EV_{DD} voltage 276
4.4.5 Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching I/O buffers 277
4.5 Register Settings When Using Multiplexed Function 281
4.5.1 Basic concept of using multiplexed functions 281
4.5.2 Register settings when the output of a multiplexed function is not used 283
4.5.3 Register settings and port pin state 284
4.5.4 Examples of register settings for port and multiplexed functions 284
4.6 Points for Caution When Using Port Function 331
4.6.1 Point for Caution on 1-bit memory manipulation instruction for port register (Pxx) 331
4.6.2 Notes on specifying the pin settings 332

5. Operation State Control 333
5.1 Configuration of Operation State Control 333
5.2 Registers for Controlling the Operation State Control 335
5.2.1 Flash operating mode select register (FLMODE) 335
5.2.2 Flash operating mode protect register (FLMWRP) 337
5.3 Initial Setting of Operating Modes of the Flash Memory 338
5.4 Transitions between Operating Modes of the Flash Memory 339
5.5 Details of Operating Modes of the Flash Memory 340
5.5.1 Details of HS (high-speed main) mode 340
5.5.2 Details of LS (low-speed main) mode 341
5.5.3 Details of LP (low-power main) mode 342
5.5.4 Details on SUB mode 343

6. Clock Generator 344
6.1 Functions of Clock Generator 344
6.2 Configuration of Clock Generator 346
6.3 Registers for Controlling the Clock Generator 348
6.3.1 Clock operation mode control register (CMC) 349
6.3.2 System clock control register (CKC) 352
6.3.3 Clock operation status control register (CSC) 354
6.3.4 Oscillation stabilization time counter status register (OSTC) 355
6.3.5 Oscillation stabilization time select register (OSTS) 357
6.3.6 Peripheral enable registers 0, 1 (PER0, PER1) 359
6.3.7 Subsystem clock supply mode control register (OSMC) 364
6.3.8 Subsystem clock select register (CKSEL) 366
6.3.9 High-speed on-chip oscillator frequency select register (HOCODIV) 367
6.3.10 Middle-speed on-chip oscillator frequency select register (MOCODIV) 368
6.3.11 High-speed system clock division register (MOSCDIV) 369
6.3.12 High-speed on-chip oscillator trimming register (HIOTRM) 370
6.3.13 Middle-speed on-chip oscillator trimming register (MIOTRM) 371
6.3.14 Low-speed on-chip oscillator trimming register (LIOTRM) 372
6.3.15 Standby mode release setting register (WKUPMD) 373
6.4 System Clock Oscillator 374
6.4.1 X1 oscillator 374
6.4.2 XT1 oscillator 374
6.4.3 High-speed on-chip oscillator 378
6.4.4 Middle-speed on-chip oscillator 378
6.4.5 Low-speed on-chip oscillator 378
6.5 Operations of the Clock Generator 379
6.6 Controlling Clocks 381
6.6.1 Example of setting the high-speed on-chip oscillator 381
6.6.2 Example of setting the X1 oscillator clock 382
6.6.3 Example of setting the XT1 oscillator clock 383
6.6.4 State transitions of the CPU clock 384
6.6.5 Conditions before changing the CPU clock and processing after changing the CPU clock 388
6.6.6 Time required for switchover of the CPU clock and main system clock 392
6.6.7 Conditions before clock oscillation is stopped 394
6.7 Resonator and Oscillator Constants 395

7. Timer Array Unit (TAU) 396
7.1 Functions of Timer Array Unit 398
7.1.1 Independent channel operation function 398
7.1.2 Simultaneous channel operation function 400
7.1.3 8-bit timer function (only for channels 1 and 3) 401
7.1.4 LIN-bus supporting function (only for channel 7 of unit 0) 401
7.2 Configuration of Timer Array Unit 402
7.2.1 Timer counter register mn (TCRmn) 408
7.2.2 Timer data register mn (TDRmn) 410
7.3 Registers for Controlling the Timer Array Unit 411
7.3.1 Peripheral enable register 0 (PER0) 412
7.3.2 Peripheral reset control register 0 (PRR0) 414

7.3.3 Timer clock select register m (TPSm) ... 415
7.3.4 Timer mode register mn (TMRmn) ... 419
7.3.5 Timer status register mn (TSRmn) ... 423
7.3.6 Timer channel enable status register m (TEm) ... 424
7.3.7 Timer channel start register m (TSm) ... 425
7.3.8 Timer channel stop register m (TTm) ... 427
7.3.9 Timer input select register 0 (TIS0) ... 428
7.3.10 Timer input select register 1 (TIS1) ... 429
7.3.11 Timer output enable register m (TOEm) ... 430
7.3.12 Timer output register m (TOm) ... 431
7.3.13 Timer output level register m (TOLm) ... 432
7.3.14 Timer output mode register m (TOMm) ... 433
7.3.15 Input switch control register (ISC) ... 434
7.3.16 Noise filter enable registers 1, 2 (NFEN1, NFEN2) ... 435
7.3.17 Registers for controlling the port functions multiplexed with the inputs and outputs of timers ... 438
7.4 Basic Specifications of the Timer Array Unit ... 439
7.4.1 Basic specifications of the simultaneous channel operation function ... 439
7.4.2 Basic specifications of the 8-bit timer function (only for channels 1 and 3) ... 441
7.5 Operations of Counters ... 442
7.5.1 Counter clock (f_{TCLK}) ... 442
7.5.2 Timing of the start of counting ... 444
7.5.3 Operations of counters ... 445
7.6 Channel Output (TOmn Pin) Control ... 450
7.6.1 TOmn pin output circuit configuration ... 450
7.6.2 TOmn pin output setting ... 451
7.6.3 Cautions on channel output operation ... 452
7.6.4 Collective manipulation of TOmn bit ... 458
7.6.5 Timer interrupts and TOmn outputs when counting is started ... 459
7.7 Timer Input (TImn) Control ... 460
7.7.1 TImn input circuit configuration ... 460
7.7.2 Noise filter ... 460
7.7.3 Cautions on channel input operation ... 461
7.8 Independent Channel Operation Function of Timer Array Unit ... 462
7.8.1 Operation as an interval timer or for square wave output ... 462
7.8.2 Operation as an external event counter ... 466
7.8.3 Operation as a frequency divider (only for channel 0 of unit 0) ... 470
7.8.4 Operation for input pulse interval measurement ... 475
7.8.5 Operation for input signal low- and high-level width measurement ... 480
7.8.6 Operation as a delay counter ... 484
7.9 Simultaneous Channel Operation Function of Timer Array Unit ... 488
7.9.1 Operation for the one-shot pulse output function ... 488
7.9.2 Operation for the PWM function ... 495
7.9.3 Operation for the multiple PWM output function ... 502
7.10 Cautions When Using Timer Array Unit ... 510
7.10.1 Cautions when using timer output ... 510
7.10.2 Point for caution when a timer output is to be used as an event input for the ELCL ... 510

8. Realtime Clock (RTC) 511
8.1 Functions of Realtime Clock 511
8.2 Configuration of the Realtime Clock 512
8.3 Registers for Controlling the Realtime Clock 514
8.3.1 Peripheral enable register 0 (PER0) 515
8.3.2 Subsystem clock supply mode control register (OSMC) 516
8.3.3 Realtime clock control register 0 (RTCC0) 517
8.3.4 Realtime clock control register 1 (RTCC1) 519
8.3.5 Second count register (SEC) 521
8.3.6 Minute count register (MIN) 521
8.3.7 Hour count register (HOUR) 522
8.3.8 Day count register (DAY) 524
8.3.9 Day-of-week count register (WEEK) 525
8.3.10 Month count register (MONTH) 526
8.3.11 Year count register (YEAR) 526
8.3.12 Time error correction register (SUBCUD) 527
8.3.13 Alarm minute register (ALARMWM) 528
8.3.14 Alarm hour register (ALARMWH) 528
8.3.15 Alarm day-of-week register (ALARMWW) 529
8.3.16 Registers for controlling the port functions multiplexed with the realtime clock output 529
8.4 Operations of the Realtime Clock 530
8.4.1 Starting the realtime clock operation 530
8.4.2 Shifting to HALT or STOP mode after starting operation 531
8.4.3 Reading from and writing to the counters of the realtime clock 532
8.4.4 Setting alarm by the realtime clock 536
8.4.5 1 Hz output by the realtime clock 537
8.4.6 Example of time error correction by the realtime clock 538

9. 32-bit Interval Timer (TML32) 543
9.1 Overview 543
9.2 Registers for Controlling the 32-bit Interval Timer 546
9.2.1 Peripheral enable register 1 (PER1) 547
9.2.2 Peripheral reset control register 1 (PRR1) 548
9.2.3 Interval timer compare registers 0mn (ITLCMP0mn) 549
9.2.4 Interval timer compare registers 0n (ITLCMP0n) 549
9.2.5 Interval timer capture register 00 (ITLCAP00) 550
9.2.6 Interval timer control register (ITLCTL0) 551
9.2.7 Interval timer clock select register 0 (ITLCSEL0) 553
9.2.8 Interval timer frequency division register 0 (ITLFDIV00) 554
9.2.9 Interval timer frequency division register 1 (ITLFDIV01) 555
9.2.10 Interval timer capture control register 0 (ITLCC0) 556
9.2.11 Interval timer status register (ITLS0) 558
9.2.12 Interval timer match detection mask register (ITLMKF0) 560
9.3 Operation 561
9.3.1 Counter mode settings 561
9.3.2 Capture mode settings 564
9.3.3 Timer operation 565
9.3.4 Capture operation 566
9.3.5 Interrupt 568

9.3.6 Interval timer setting procedures 570

10. Clock Output/Buzzer Output Controller (PCLBUZ) 574
10.1 Functions of Clock Output/Buzzer Output Controller 574
10.2 Configuration of Clock Output/Buzzer Output Controller 576
10.3 Registers for Controlling the Clock Output/Buzzer Output Controller 576
10.3.1 Clock output select registers n (CKSn) 576
10.3.2 Registers for controlling the port functions multiplexed with the clock or buzzer outputs 578
10.4 Operations of the Clock Output/Buzzer Output Controller 579
10.4.1 Operation of output pins 579
10.5 Point for Caution When the Clock Output/Buzzer Output Controller is to be Used 579

11. Watchdog Timer (WDT) 580
11.1 Functions of Watchdog Timer 580
11.2 Configuration of Watchdog Timer 581
11.3 Register for Controlling the Watchdog Timer 582
11.3.1 Watchdog timer enable register (WDTE) 582
11.4 Operation of Watchdog Timer 583
11.4.1 Controlling operation of watchdog timer 583
11.4.2 Setting overflow time of watchdog timer 584
11.4.3 Setting window open period of watchdog timer 585
11.4.4 Setting watchdog timer interval interrupt 587

12. A/D Converter (ADC) 588
12.1 Function of A/D Converter 588
12.2 Configuration of A/D Converter 591
12.3 Registers for Controlling the A/D Converter 593
12.3.1 Peripheral enable register 0 (PER0) 594
12.3.2 Peripheral reset control register 0 (PRR0) 595
12.3.3 A/D converter mode register 0 (ADM0) 596
12.3.4 A/D converter mode register 1 (ADM1) 615
12.3.5 A/D converter mode register 2 (ADM2) 616
12.3.6 12-bit/10-bit A/D conversion result register (ADCRn) 619
12.3.7 8-bit A/D conversion result register (ADCRnH) 620
12.3.8 Analog input channel specification register (ADS) 621
12.3.9 Conversion result comparison upper limit setting register (ADUL) 624
12.3.10 Conversion result comparison lower limit setting register (ADLL) 624
12.3.11 A/D test register (ADTES) 625
12.3.12 Registers for controlling the port functions multiplexed with the analog inputs of the A/D converter 626
12.4 A/D Converter Operations 627
12.5 Input Voltage and Conversion Results 629
12.6 Operating Modes of the A/D Converter 630
12.6.1 Software trigger no-wait mode: Select mode and sequential conversion mode 630
12.6.2 Software trigger no-wait mode: Select mode and one-shot conversion mode 631
12.6.3 Software trigger no-wait mode: Scan mode and sequential conversion mode 632
12.6.4 Software trigger no-wait mode: Scan mode and one-shot conversion mode 633
12.6.5 Software trigger wait mode: Select mode and sequential conversion mode 634

12.6.6 Software trigger wait mode: Select mode and one-shot conversion mode 635
12.6.7 Software trigger wait mode: Scan mode and sequential conversion mode 636
12.6.8 Software trigger wait mode: Scan mode and one-shot conversion mode 637
12.6.9 Hardware trigger no-wait mode: Select mode and sequential conversion mode 638
12.6.10 Hardware trigger no-wait mode: Select mode and one-shot conversion mode 639
12.6.11 Hardware trigger no-wait mode: Scan mode and sequential conversion mode 640
12.6.12 Hardware trigger no-wait mode: Scan mode and one-shot conversion mode 641
12.6.13 Hardware trigger wait mode: Select mode and sequential conversion mode 643
12.6.14 Hardware trigger wait mode: Select mode and one-shot conversion mode 644
12.6.15 Hardware trigger wait mode: Scan mode and sequential conversion mode 645
12.6.16 Hardware trigger wait mode: Scan mode and one-shot conversion mode 646
12.7 Flows of Setting the A/D Converter Modes 647
12.7.1 Settings in software trigger no-wait mode 647
12.7.2 Settings in software trigger wait mode 648
12.7.3 Settings in hardware trigger no-wait mode 649
12.7.4 Settings in hardware trigger wait mode 650
12.7.5 Example of settings when temperature sensor output voltage, internal reference voltage, or TSCAP voltage of the CTSU is selected in the software trigger no-wait mode and one-shot conversion mode 651
12.7.6 Settings in test mode 652
12.8 SNOOZE Mode 653
12.8.1 A/D conversion by inputting a software trigger 653
12.8.2 A/D conversion by inputting a hardware trigger 655
12.9 How to Read A/D Converter Characteristics Table 659
12.10 Points for Caution When the A/D Converter is to be Used 663

13. D/A Converter (DAC) 667
13.1 Functions of D/A Converter 667
13.2 Configuration of D/A Converter 668
13.3 Registers for Controlling the D/A Converter 669
13.3.1 Peripheral enable register 1 (PER1) 669
13.3.2 Peripheral reset control register 1 (PRR1) 670
13.3.3 D/A converter mode register (DAM) 671
13.3.4 D/A conversion value setting register i (DACSi) 671
13.3.5 Registers for controlling the event output from the logic and event link controller 672
13.3.6 Registers for controlling the port functions multiplexed with the analog outputs of the D/A converter 672
13.4 Operations of D/A Converter 673
13.4.1 Operation in normal operation mode 673
13.4.2 Operation in realtime output mode 674
13.4.3 Timing for outputting D/A conversion value 675
13.5 Points for Caution when the D/A Converter is to be Used 676

14. Comparator (CMP) 677
14.1 Functions of Comparator 677
14.2 Configuration of Comparator 678
14.3 Registers for Controlling the Comparator 679
14.3.1 Peripheral enable register 1 (PER1) 680
14.3.2 Peripheral reset control register 1 (PRR1) 681

14.3.3 Comparator mode setting register (COMPMDR) 682
14.3.4 Comparator filter control register (COMPFIR) 684
14.3.5 Comparator output control register (COMPOCR) 686
14.3.6 Registers for controlling the port functions multiplexed with the analog inputs and outputs of the comparator 687
14.4 Operation 688
14.4.1 Comparator i digital filter 690
14.4.2 Comparator i interrupts 690
14.4.3 Event signal output for the logic and event link controller (ELCL) 691
14.4.4 Comparator i output 692
14.4.5 Stopping or supplying the clock to the comparator 692

15. Serial Array Unit (SAU) 693
15.1 Functions of Serial Array Unit 695
15.1.1 Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31) 695
15.1.2 UART (UART0 to UART3) 696
15.1.3 Simplified I^2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31) 697
15.2 Configuration of Serial Array Unit 698
15.2.1 Shift register 701
15.2.2 Lower 8 or 9 bits of the serial data register mn (SDRmn) 701
15.3 Registers for Controlling the Serial Array Unit 703
15.3.1 Peripheral enable register 0 (PER0) 704
15.3.2 Peripheral reset control register 0 (PRR0) 706
15.3.3 Serial clock select register m (SPSm) 707
15.3.4 Serial mode register mn (SMRmn) 709
15.3.5 Serial communication operation setting register mn (SCRmn) 711
15.3.6 Serial data register mn (SDRmn) 713
15.3.7 Serial flag clear trigger register mn (SIRmn) 715
15.3.8 Serial status register mn (SSRmn) 716
15.3.9 Serial channel start register m (SSm) 718
15.3.10 Serial channel stop register m (STm) 719
15.3.11 Serial channel enable status register m (SEm) 720
15.3.12 Serial output enable register m (SOEm) 721
15.3.13 Serial output register m (SOm) 722
15.3.14 Serial output level register m (SOLm) 724
15.3.15 Serial standby control register m (SSCm) 726
15.3.16 Input switch control register (ISC) 728
15.3.17 Noise filter enable register 0 (NFEN0) 730
15.3.18 Registers for controlling the port functions multiplexed with the inputs and outputs of the serial array unit 731
15.3.19 UART loopback select register (ULBS) 732
15.4 Operation Stop Mode 733
15.4.1 Stopping the operation by units 733
15.4.2 Stopping the operation by channels 734
15.5 Operation of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31) Communication 735
15.5.1 Master transmission 738
15.5.2 Master reception 747
15.5.3 Master transmission/reception 757

15.5.4 Slave transmission 767
15.5.5 Slave reception 776
15.5.6 Slave transmission/reception 784
15.5.7 SNOOZE mode 794
15.5.8 Calculating transfer clock frequency 799
15.5.9 Procedure for processing errors that occurred during simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31) communication 801
15.6 Operation of UART (UART0 to UART3) Communication 802
15.6.1 UART transmission 805
15.6.2 UART reception 815
15.6.3 SNOOZE mode 822
15.6.4 Calculating baud rate 829
15.6.5 Procedure for processing errors that occurred during UART (UART0 to UART3) communication 833
15.7 LIN Communication Operation 834
15.7.1 LIN transmission 834
15.7.2 LIN reception 837
15.8 Operation of Simplified I2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31) Communication 843
15.8.1 Address field transmission 846
15.8.2 Data transmission 852
15.8.3 Data reception 855
15.8.4 Stop condition generation 859
15.8.5 Calculating transfer rate 861
15.8.6 Procedure for processing errors that occurred during simplified I2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31) communication 864

16. Serial Interface IICA (IICA) 865
16.1 Functions of Serial Interface IICA 865
16.2 Configuration of Serial Interface IICA 868
16.3 Registers for Controlling Serial Interface IICA 871
16.3.1 Peripheral enable register 0 (PER0) 872
16.3.2 Peripheral reset control register 0 (PRR0) 873
16.3.3 IICA control register n0 (IICCTLn0) 874
16.3.4 IICA status register n (IICSn) 878
16.3.5 IICA flag register n (IICFn) 882
16.3.6 IICA control register n1 (IICCTLn1) 884
16.3.7 IICA low-level width setting register n (IICWLn) 887
16.3.8 IICA high-level width setting register n (IICWHn) 887
16.3.9 Registers for controlling the port functions multiplexed with the IICA serial interface inputs and outputs 888
16.4 I2C Bus Mode Functions 889
16.4.1 Pin configuration 889
16.4.2 Setting transfer clock by using IICWLn and IICWHn registers 890
16.5 I2C Bus Definitions and Control Methods 892
16.5.1 Start conditions 892
16.5.2 Address 893
16.5.3 Transfer direction specification 893
16.5.4 Acknowledge (ACK) 894
16.5.5 Stop condition 895

16.5.6 Clock stretching 896
16.5.7 Release from clock stretching 898
16.5.8 Timing of generation of the interrupt request signal (INTIICAn) and control of clock stretching 899
16.5.9 Address match detection method 900
16.5.10 Error detection 900
16.5.11 Extension code 901
16.5.12 Arbitration 902
16.5.13 Wakeup function 904
16.5.14 Communication reservation 907
16.5.15 Cautions 911
16.5.16 Communication operations 912
16.5.17 Timing of I^2C interrupt request signal (INTIICAn) occurrence 920
16.6 Timing Charts 940

17. Serial Interface UARTA (UARTA) 955
17.1 Overview 955
17.2 Registers for Controlling the Serial Interface UARTA 957
17.2.1 Peripheral enable register 1 (PER1) 958
17.2.2 Transmit buffer register n (TXBAn) 959
17.2.3 Receive buffer register n (RXBAn) 960
17.2.4 Operation mode setting register n0 (ASIMAn0) 961
17.2.5 Operation mode setting register n1 (ASIMAn1) 963
17.2.6 Baud rate generator control register n (BRGCAn) 964
17.2.7 Status register n (ASISAn) 965
17.2.8 Status clear trigger register n (ASCTAn) 967
17.2.9 UARTA clock select register 0 (UTA0CK) 968
17.2.10 UARTA clock select register 1 (UTA1CK) 970
17.2.11 UART loopback select register (ULBS) 971
17.2.12 Registers for controlling the port functions multiplexed with the inputs and outputs of the UARTA serial interface 972
17.3 Operation 973
17.3.1 Operation stop mode 973
17.3.2 UART mode 973
17.3.3 Receive data noise filter 987
17.3.4 Baud rate generator 988
17.4 Points for Caution when the Serial Interface UARTA is to be Used 994
17.4.1 Port setting for the RxDAn pin 994
17.4.2 Serial interfaces 994
17.4.3 Point for caution when selecting the serial interface UARTAn operation clock (f_{UTAn}) 994

18. Remote Control Signal Receiver (REMC) 995
18.1 Remote Control Signal Receiver 995
18.2 Registers for Controlling the Remote Control Signal Receiver 997
18.2.1 Peripheral enable register 1 (PER1) 998
18.2.2 Peripheral reset control register 1 (PRR1) 999
18.2.3 Function select register 0 (REMCON0) 1000
18.2.4 Function select register 1 (REMCON1) 1002
18.2.5 Status register (REMSTS) 1003

18.2.6 Interrupt control register (REMINT) 1007
18.2.7 Compare control register (REMCPC) 1008
18.2.8 Compare value setting register (REMCPD) 1009
18.2.9 Header pattern minimum width setting register (HDPMIN) 1009
18.2.10 Header pattern maximum width setting register (HDPMAX) 1010
18.2.11 Data 0 pattern minimum width setting register (D0PMIN) 1010
18.2.12 Data 0 pattern maximum width setting register (D0PMAX) 1011
18.2.13 Data 1 pattern minimum width setting register (D1PMIN) 1011
18.2.14 Data 1 pattern maximum width setting register (D1PMAX) 1012
18.2.15 Special data pattern minimum width setting register (SDPMIN) 1012
18.2.16 Special data pattern maximum width setting register (SDPMAX) 1013
18.2.17 Pattern end setting register (REMPE) 1013
18.2.18 Receiver standby control register (REMSTC) 1014
18.2.19 Receive bit count register (REMRBIT) 1015
18.2.20 Receive data 0 register (REMDAT0) 1016
18.2.21 Receive data j register (REMDATj) 1017
18.2.22 Measurement result register (REMTIM) 1018
18.2.23 Registers for controlling the port functions multiplexed with the remote control signal input 1018
18.3 Operation 1019
18.3.1 Overview of REMC operation 1019
18.3.2 Initial setting 1019
18.3.3 Pattern setting 1021
18.3.4 Operating clock 1024
18.3.5 RIN0 input 1025
18.3.6 Pattern detection 1027
18.3.7 Pattern end 1032
18.3.8 Receive data buffer 1033
18.3.9 Compare function 1037
18.3.10 Error pattern reception 1038
18.3.11 Storing base timer value when pattern is detected 1040
18.3.12 Interrupts 1041
18.3.13 SNOOZE mode function 1042
18.4 Points for Caution when the Remote Control Signal Receiver is to be Used 1046
18.4.1 Access to registers when operation of the remote control signal receiver is enabled 1046
18.4.2 Timing of changing the register values 1046
18.4.3 RIN0 input control 1046
18.4.4 Changing the operating clock 1046
18.4.5 Reading registers 1047

19. Data Transfer Controller (DTC) 1048
19.1 Functions of DTC 1048
19.2 Configuration of DTC 1050
19.3 Registers for Controlling the DTC 1051
19.3.1 Allocation of DTC control data area and DTC vector table area 1052
19.3.2 Control data allocation 1053
19.3.3 Vector table 1055
19.3.4 Peripheral enable register 1 (PER1) 1057
19.3.5 DTC control register j (DTCCRj) 1058

19.3.6 DTC block size register j (DTBLSj) 1060
19.3.7 DTC transfer count register j (DTCCTj) 1061
19.3.8 DTC transfer count reload register j (DTRLDj) 1061
19.3.9 DTC source address register j (DTSARj) 1062
19.3.10 DTC destination address register j (DTDARj) 1062
19.3.11 DTC activation enable register i (DTCENi) 1063
19.3.12 DTC base address register (DTCBAR) 1066
19.4 DTC Operation 1067
19.4.1 Activation sources 1068
19.4.2 Normal mode 1069
19.4.3 Repeat mode 1072
19.4.4 Chain transfers 1076
19.5 Points for Caution When the DTC is to be Used 1078
19.5.1 Setting DTC control data and vector table 1078
19.5.2 Allocation of DTC control data area and DTC vector table area 1078
19.5.3 DTC pending instruction 1079
19.5.4 Operation when accessing data flash memory area 1079
19.5.5 Number of DTC execution clock cycles 1080
19.5.6 DTC response time 1081
19.5.7 DTC activation sources 1081
19.5.8 Operation in standby mode 1082

20. Logic and Event Link Controller (ELCL) 1083
20.1 Functions of the ELCL 1083
20.2 Configuration of ELCL 1083
20.3 Registers for Controlling the ELCL 1093
20.3.1 Input signal select registers n (ELISELn) 1094
20.3.2 Event link L1 signal select registers 0 to 3 (ELL1SEL0 to ELL1SEL3) 1099
20.3.3 Event link L1 signal select registers 4, 5 (ELL1SEL4, ELL1SEL5) 1101
20.3.4 Event link L1 signal select register 6 (ELL1SEL6) 1102
20.3.5 Logic cell block L1 control register (ELL1CTL) 1103
20.3.6 Event link L1 output select registers 0 to 3 (ELL1LNK0 to ELL1LNK3) 1104
20.3.7 Event link L1 output select registers 4, 5 (ELL1LNK4, ELL1LNK5) 1105
20.3.8 Event link L1 output select register 6 (ELL1LNK6) 1106
20.3.9 Event link L2 signal select registers 0 to 3 (ELL2SEL0 to ELL2SEL3) 1107
20.3.10 Event link L2 signal select registers 4, 5 (ELL2SEL4, ELL2SEL5) 1109
20.3.11 Event link L2 signal select register 6 (ELL2SEL6) 1110
20.3.12 Logic cell block L2 control register (ELL2CTL) 1111
20.3.13 Event link L2 output select registers 0 to 3 (ELL2LNK0 to ELL2LNK3) 1112
20.3.14 Event link L2 output select registers 4, 5 (ELL2LNK4, ELL2LNK5) 1113
20.3.15 Event link L2 output select register 6 (ELL2LNK6) 1114
20.3.16 Event link L3 signal select registers 0 to 3 (ELL3SEL0 to ELL3SEL3) 1115
20.3.17 Event link L3 signal select registers 4, 5 (ELL3SEL4, ELL3SEL5) 1117
20.3.18 Event link L3 signal select register 6 (ELL3SEL6) 1118
20.3.19 Logic cell block L3 control register (ELL3CTL) 1119
20.3.20 Event link L3 output select registers 0 to 3 (ELL3LNK0 to ELL3LNK3) 1120
20.3.21 Event link L3 output select registers 4, 5 (ELL3LNK4, ELL3LNK5) 1121
20.3.22 Event link L3 output select register 6 (ELL3LNK6) 1122
20.3.23 Output signal select registers n (ELOSELn) 1123

20.3.24 Output signal enable register (ELOENCTL) 1125
20.3.25 Output signal monitor register (ELOMONI) 1127
20.3.26 Registers for controlling the port functions multiplexed with the I/O pins of the logic and event link controller 1128
20.4 ELCL Operation 1129
20.5 Procedure of Setting the ELCL 1130
20.6 Points for Caution when the ELCL is to be Used 1133

21. Interrupt Functions 1134
21.1 Interrupt Function Types 1134
21.2 Interrupt Sources and Configuration 1134
21.3 Registers for Controlling the Interrupt Functions 1145
21.3.1 Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H) 1156
21.3.2 Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H) 1158
21.3.3 Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H) 1160
21.3.4 External interrupt rising edge enable registers (EGP0, EGP1), External interrupt falling edge enable registers (EGN0, EGN1) 1164
21.3.5 Program status word (PSW) 1166
21.3.6 Registers for controlling the port functions multiplexed with the interrupt inputs 1167
21.4 Interrupt Servicing Operations 1168
21.4.1 Maskable interrupt request acknowledgment 1168
21.4.2 Software interrupt request acknowledgment 1171
21.4.3 Multiple interrupt servicing 1171
21.4.4 Interrupt request held pending 1175

22. Key Interrupt Function 1176
22.1 Functions of the Key Interrupt 1176
22.2 Configuration of the Key Interrupt 1176
22.3 Registers for Controlling the Key Interrupt 1178
22.3.1 Key return control register (KRCTL) 1178
22.3.2 Key return mode register 0 (KRM0) 1179
22.3.3 Key return flag register (KRF) 1180
22.3.4 Registers for controlling the port functions multiplexed with the key interrupt inputs 1181

23. Standby Function 1182
23.1 Standby Function 1182
23.2 Registers for Controlling the Standby Function 1183
23.2.1 Standby mode release setting register (WKUPMD) 1184
23.2.2 Memory power reduction control register (PSMCR) 1185
23.3 Standby Operation 1187
23.3.1 HALT mode 1187
23.3.2 STOP mode 1194
23.3.3 SNOOZE mode 1199

24. Reset Function 1203
24.1 Timing of Reset Operation 1205

24.2 Registers for Controlling the Reset Function 1209
24.2.1 Reset control flag register (RESF) 1209
24.2.2 Power-on-reset status register (PORSR) 1212
24.2.3 Peripheral reset control register 0 (PRR0) 1213
24.2.4 Peripheral reset control register 1 (PRR1) 1214

25. Power-on-reset Circuit (POR) 1215
25.1 Functions of Power-on-reset Circuit 1215
25.2 Configuration of Power-on-reset Circuit 1216
25.3 Operation of Power-on-reset Circuit 1217

26. Voltage Detector (LVD) 1219
26.1 Functions of Voltage Detector 1219
26.2 Configuration of Voltage Detector 1220
26.3 Registers for Controlling the Voltage Detector 1221
26.3.1 Voltage detection register (LVIM) 1221
26.3.2 LVD detection flag clearing register (LVDFCLR) 1222
26.3.3 Voltage detection level register (LVIS) 1223
26.4 Operation of Voltage Detector 1225
26.4.1 Setting for use in the reset mode 1225
26.4.2 Setting for use in the interrupt mode 1228
26.5 Points for Caution when the Voltage Detector Is to be Used 1231

27. Safety Functions 1234
27.1 Overview of Safety Functions 1234
27.2 Registers for Controlling the Safety Functions 1235
27.3 Operation of Safety Functions 1235
27.3.1 Flash memory CRC operation (high-speed CRC) 1235
27.3.1.1 Flash memory CRC control register (CRC0CTL) 1236
27.3.1.2 Flash memory CRC operation result register (PGCRCL) 1238
27.3.2 CRC operation (general-purpose CRC) 1240
27.3.2.1 CRC input register (CRCIN) 1240
27.3.2.2 CRC data register (CRCD) 1241
27.3.3 Flash memory guard function 1242
27.3.3.1 Code flash memory guard register (GFLASH0) 1242
27.3.3.2 Data flash memory guard register (GFLASH1) 1243
27.3.3.3 Flash security area guard register (GFLASH2) 1244
27.3.4 RAM parity error detection 1245
27.3.4.1 RAM parity error control register (RPECTL) 1245
27.3.5 RAM guard function 1247
27.3.5.1 Invalid memory access detection control register (IAWCTL) 1247
27.3.6 SFR guard function 1248
27.3.6.1 Invalid memory access detection control register (IAWCTL) 1248
27.3.7 Illicit memory access detection 1249
27.3.7.1 Invalid memory access detection control register (IAWCTL) 1250
27.3.8 Guard function of invalid memory access detection control register 1251
27.3.8.1 Guard register of IAWCTL register (GIAWCTL) 1251
27.3.9 Frequency detection 1252
27.3.9.1 Timer input select register 0 (TIS0) 1253

27.3.10 Testing of the A/D converter 1254
27.3.10.1 A/D test register (ADTES) 1256
27.3.10.2 Analog input channel specification register (ADS) 1257
27.3.11 Detection of the digital output signal level of the I/O pins 1259
27.3.11.1 Port mode select register (PMS) 1259
27.3.12 UART loopback 1260
27.3.12.1 UART loopback select register (ULBS) 1261

28. Security Functions 1262
28.1 True Random Number Generator (TRNG) 1262
28.1.1 Function of the true random number generator 1262
28.1.2 Registers for controlling the true random number generator 1262
28.1.2.1 Random number seed command register 0 (TRNGSCR0) 1263
28.1.2.2 Random number seed data register (TRNGSDR) 1264
28.1.3 Operations of the true random number generator 1265
28.2 Flash Read Protection 1266
28.2.1 Function of flash read protection 1266
28.2.2 Setting of flash read protection 1266
28.2.3 Operation 1268
28.3 Unique ID 1269
28.3.1 Function of a unique ID 1269
28.3.2 ASCII codes representing the product name 1269

29. SNOOZE Mode Sequencer (SMS) 1270
29.1 Functions of the SNOOZE Mode Sequencer 1270
29.2 Configuration of the SNOOZE Mode Sequencer 1270
29.3 Registers for Controlling the SNOOZE Mode Sequencer 1271
29.3.1 Peripheral enable register 1 (PER1) 1271
29.3.2 Peripheral reset control register 1 (PRR1) 1272
29.3.3 Sequencer instruction registers p (SMSIp) 1273
29.3.4 Sequencer general-purpose registers n (SMSGn) 1276
29.3.5 Sequencer control register (SMSC) 1278
29.3.6 Sequencer status register (SMSS) 1280
29.4 Operations of the SNOOZE Mode Sequencer 1281
29.4.1 Internal operations of the SNOOZE mode sequencer 1281
29.4.2 Memory space allocated to the sequencer 1282
29.4.3 Sequencer flags 1284
29.4.4 Interrupt from the SNOOZE Mode Sequencer 1284
29.4.5 Procedures for running the SNOOZE mode sequencer 1285
29.4.6 States of the SNOOZE mode sequencer 1287
29.5 Commands for Use in Processing by the Sequencer 1288
29.5.1 8-bit data transfer 1 1289
29.5.2 8-bit data transfer 2 1290
29.5.3 16-bit data transfer 1 1291
29.5.4 16-bit data transfer 2 1292
29.5.5 1-bit data setting 1293
29.5.6 1-bit data clearing 1294
29.5.7 1-bit data transfer 1295
29.5.8 Word addition 1296

29.5.9 Word subtraction 1297
29.5.10 Word comparison 1298
29.5.11 Logical shift right 1299
29.5.12 Branch 1 (SCY = 1) 1300
29.5.13 Branch 2 (SCY = 0) 1301
29.5.14 Branch 3 (SZ = 1) 1302
29.5.15 Branch 4 (SZ = 0) 1303
29.5.16 Wait 1304
29.5.17 Conditional wait 1 (bit = 1) 1306
29.5.18 Conditional wait 2 (bit = 0) 1307
29.5.19 Termination 1308
29.5.20 DTC activation 1309
29.6 Operation in Standby Modes 1310

30. Capacitive Sensing Unit (CTSU2L) 1311
30.1 Overview 1313
30.1.1 Functional overview of the CTSU 1313
30.1.2 Measurement states 1315
30.2 Registers for Controlling the Capacitive Sensing Unit 1316
30.2.1 Peripheral enable register 1 (PER1) 1317
30.2.2 Peripheral reset control register 1 (PRR1) 1318
30.2.3 CTSU control registers AL and AH (CTSUCRAL, CTSUCRAH) 1319
30.2.4 CTSU control registers BL and BH (CTSUCRBL, CTSUCRBH) 1327
30.2.5 CTSU measurement channel registers L and H (CTSUMCHL, CTSUMCHH) 1330
30.2.6 CTSU channel enable control registers AL, AH, BL, and BH (CTSUCHACAL, CTSUCHACAH, CTSUCHACBL, CTSUCHACBH) 1332
30.2.7 CTSU channel transmit/receive control registers AL, AH, BL, and BH (CTSUCHTRCAL, CTSUCHTRCAH, CTSUCHTRCBL, CTSUCHTRCBH) 1334
30.2.8 CTSU status register L (CTSUSRL) 1336
30.2.9 CTSU sensor offset registers 0 and 1 (CTSUSO0, CTSUSO1) 1339
30.2.10 CTSU sensor counter registers (CTSUSC, CTSUUC) 1342
30.2.11 CTSU calibration registers L and H (CTSUDBGR0, CTSUDBGR1) 1343
30.2.12 CTSU sensor unit clock control registers AL, AH, BL, and BH (CTSUSUCLK0, CTSUSUCLK1, CTSUSUCLK2, CTSUSUCLK3) 1346
30.2.13 CTSU trimming registers AL and AH (CTSUTRIM0, CTSUTRIM1) 1348
30.2.14 CTSU trimming registers BL and BH (CTSUTRIM2, CTSUTRIM3) 1349
30.2.15 Registers for controlling the port functions multiplexed with the inputs and outputs of the capacitive sensing unit 1350
30.3 Operation 1351
30.3.1 Principles of measurement 1351
30.3.2 Initial setting procedure 1354
30.3.3 Measurement methods 1355
30.3.4 Scan modes 1358
30.3.5 Multi-clock measurement 1358
30.3.6 Interrupts 1359
30.4 Usage Notes on the Capacitive Sensing Unit 1361
30.4.1 Evaluation of detection by the capacitive sensing unit 1361
30.4.2 TSCAP pin 1361
30.4.3 Changing the clock and port settings during measurement 1361

30.4.4 Software trigger 1362
30.4.5 External trigger 1362
30.4.6 Reading the CTSU sensor counter registers (CTSUSC and CTSUUC) 1362
30.4.7 Forced stop 1362
30.4.8 Sampling cycle when the spread spectrum is in use 1362
30.4.9 Settings of the transmission pin when the self-capacitance method is to be used 1362
30.4.10 Points to note on noise countermeasures 1363
30.4.11 Memory occupancy 1364

31. Regulator 1365
31.1 Overview 1365

32. Option Bytes 1366
32.1 Functions of Option Bytes 1366
32.1.1 User option bytes (000C0H to 000C2H or 040C0H to 040C2H) 1366
32.1.2 On-chip debug option byte (000C3H or 040C3H) 1367
32.2 Format of User Option Bytes 1368
32.3 Format of On-chip Debug Option Byte 1372
32.4 Setting of Option Bytes 1373

33. Flash Memory 1374
33.1 Serial Programming Using Flash Memory Programmer 1375
33.1.1 Programming environment 1378
33.1.2 Communications mode 1378
33.2 Serial Programming Using External Device (that Incorporates UART) 1379
33.2.1 Programming environment 1379
33.2.2 Communications mode 1380
33.3 Handling of Pins on the Board 1381
33.3.1 P40/TOOL0 pin 1381
33.3.2 $\overline{\text{RESET}}$ pin 1382
33.3.3 Port pins 1383
33.3.4 REGC pin 1383
33.3.5 Power supply 1383
33.4 Programming Method 1384
33.4.1 Serial programming procedure 1384
33.4.2 Flash memory programming mode 1385
33.4.3 Selecting communications mode 1386
33.4.4 Communications commands 1387
33.5 Processing Times for Commands When the Dedicated Flash Memory Programmer Is in Use (Reference Values) 1389
33.6 Self-programming 1390
33.6.1 Self-programming procedure 1391
33.6.2 Registers for controlling the flash memory 1392
33.6.2.1 Flash address pointer registers H and L (FLAPH, FLAPL) 1393
33.6.2.2 Flash end address pointer registers H and L (FLSEDH, FLSEDL) 1394
33.6.2.3 Flash write buffer registers H and L (FLWH, FLWL) 1396
33.6.2.4 Flash protect command register (PFCMD) 1397
33.6.2.5 Flash status register (PFS) 1398
33.6.2.6 Flash programming mode control register (FLPMC) 1399

33.6.2.7 Flash area selection register (FLARS) 1400
33.6.2.8 Flash memory sequencer initial setting register (FSSET) 1401
33.6.2.9 Flash memory sequencer control register (FSSQ) 1403
33.6.2.10 Flash extra area sequencer control register (FSSE) 1405
33.6.2.11 Flash registers initialization register (FLRST) 1407
33.6.2.12 Flash memory sequencer status registers H and L (FSASTH, FSASTL) 1408
33.6.2.13 Flash security flag monitoring register (FLSEC) 1410
33.6.2.14 Flash FSW monitoring register E (FLFSWE) 1411
33.6.2.15 Flash FSW monitoring register S (FLFSWS) 1412
33.6.2.16 Data flash control register (DFLCTL) 1413
33.6.2.17 Interrupt vector jump enable register (VECTCTRL) 1414
33.6.2.18 Interrupt vector change registers 0 and 1 (FLSIVC0, FLSIVC1) 1415
33.6.3 Setting the flash memory control mode 1416
33.6.4 Initializing the registers for use with the flash memory sequencer 1418
33.6.5 Setting the operating frequency of the flash memory sequencer 1418
33.6.6 Rewriting the flash memory 1419
33.6.7 Interrupts in code flash memory programming mode 1430
33.6.8 Example of executing the commands to rewrite the flash memory areas 1432
33.6.9 Notes on self-programming 1435
33.7 Boot Swap Function 1436
33.8 Flash Shield Window Function 1438
33.9 Security Settings 1439
33.10 Data Flash Memory 1442
33.10.1 Overview of the data flash memory 1442
33.10.2 Procedure for accessing the data flash memory 1443

34. On-chip Debugging 1444
34.1 Connection between the E2 or E2 Lite On-chip Debugging Emulator and This MCU 1444
34.2 Connection between the External Device that Incorporates UART and This MCU 1445
34.3 Security Settings for On-chip Debugging 1445
34.4 Allocation of Memory Spaces to User Resources 1446

35. BCD Correction Circuit 1448
35.1 BCD Correction Circuit Function 1448
35.2 Register for Controlling the BCD Correction Circuit 1448
35.2.1 BCD correction result register (BCDADJ) 1448
35.3 BCD Correction Circuit Operation 1449

36. Instruction Set 1451
36.1 Conventions Used in Operation List 1452
36.1.1 Operand identifiers and description methods 1452
36.1.2 Description of operation column 1453
36.1.3 Description of flag operation column 1454
36.1.4 PREFIX instruction 1454
36.2 Operation List 1455

37. Electrical Characteristics 1473
37.1 Absolute Maximum Ratings 1474

37.2 Characteristics of the Oscillators 1476
37.2.1 Characteristics of the X1 oscillator 1476
37.2.2 Characteristics of the XT1 oscillator 1476
37.2.3 Characteristics of the on-chip oscillators 1477
37.3 DC Characteristics 1478
37.3.1 Characteristics of pins 1478
37.3.2 Characteristics of the supply current 1487
37.4 AC Characteristics 1511
37.5 Characteristics of the Peripheral Functions 1517
37.5.1 Serial array unit 1517
37.5.2 Serial interface UARTA 1546
37.5.3 Serial interface IICA 1547
37.6 Characteristics of the Analog Circuits 1550
37.6.1 Characteristics of the A/D converter 1550
37.6.2 Characteristics of the temperature sensor and internal reference voltage 1555
37.6.3 Characteristics of the D/A converter 1555
37.6.4 Characteristics of the comparator 1555
37.6.5 Characteristics of the POR circuit 1556
37.6.6 Characteristics of the LVD circuit 1557
37.6.7 Characteristics of the rising slope of the power supply voltage 1559
37.7 Characteristics of Retention of RAM Data 1560
37.8 Characteristics of Flash Memory Programming 1560
37.9 Dedicated Flash Memory Programmer Communication (UART) 1562
37.10 Timing of Entry to Flash Memory Programming Modes 1562

38. Package Drawings 1563
38.1 30-pin Products 1563
38.2 32-pin Products 1564
38.3 36-pin Products 1567
38.4 40-pin Products 1568
38.5 44-pin Products 1569
38.6 48-pin Products 1572
38.7 52-pin Products 1575
38.8 64-pin Products 1576
38.9 80-pin Products 1580
38.10 100-pin Products 1583
38.11 128-pin Products 1585

APPENDIX A Revision History 1586
A.1 Major Revisions in This Edition 1586
A.2 Revision History of Preceding Editions 1589

Section 1 Outline

<R> 1.1 Differences between R7F100G and R7F104G Products

The RL78/G23 Group consists of two types of products: R7F100G and R7F104G. The R7F100G products are pin-compatible with those in the RL78/G13 Group and are available in 30- to 128-pin variants. The R7F104G products are based on the R7F100G products but feature an expanded number of CTSU channels and enhanced touch-sensing functionality. They are available in 32-, 44-, and 48-pin variants. **Table 1 - 1** lists the differences between the R7F100G and R7F104G products with 32, 44, and 48 pins.

Table 1 - 1 Peripheral Functions (1/2)

RL78/G23		R7F100G Products			R7F104G Products		
		32-pin	44-pin	48-pin	32-pin	44-pin	48-pin
Subsystem clock	Subsystem clock X (f_{SX})	32.768 kHz (V_{DD} = 1.6 to 5.5 V or V_{DD} = 2.4 to 5.5 V when f_{XT} is selected)	32.768 kHz (V_{DD} = 1.6 to 5.5 V)		—	32.768 kHz (V_{DD} = 1.6 to 5.5 V)	
I/O port	CMOS I/O	24	33	37	27	37	40
	N-ch open-drain I/O [withstand voltage: 6 V]	3	4	4	0	0	0
	Controlled current drive port	7	7	8	3	3	3
Timers	16-bit timer	8 channels			8 channels	12 channels	
	Timer output	4 channels (PWM outputs: 3 Note 1)	5 channels (PWM outputs: 4 Note 1)		5 channels (PWM outputs: 5 Note 1)	8 channels (PWM outputs: 6 Note 1)	10 channels (PWM outputs: 8 Note 1)
		8 channels (PWM outputs: 7 Note 1 Note 2)			4 channels (PWM outputs: 4 Note 1 Note 2	8 channels (PWM outputs: 6 Note 1 Note 2	10 channels (PWM outputs: 8 Note 1 Note 2
	RTC output	1 channel			—	1 channel	
Clock output/buzzer output		2 channels			1 channel		
8-/10-/12-bit resolution A/D converter		8 channels	10 channels	10 channels	7 channels	8 channels	10 channels
Comparator		2 channels			1 channel		
Serial interfaces	Simplified SPI (CSI)	3 channels	4 channels	5 channels	2 channels		
	Simplified I^2C	3 channels	4 channels	5 channels	3 channels		
	UARTA	—	2 channels		1 channel		
	I^2C bus	1 channel	2 channels		—		1 channel
Data transfer controller (DTC)		30 sources	35 sources	36 sources	31 sources		
Capacitive sensing unit	Code flash memory size: 96 to 128 KB	3 channels	6 channels	8 channels	10 channels	—	—
	Code flash memory size: 192 to 768 KB	7 channels	14 channels	16 channels	—	24 channels	25 channels

Table 1 - 1 Peripheral Functions (2/2)

RL78/G23		R7F100G Products			R7F104G Products		
		32-pin	44-pin	48-pin	32-pin	44-pin	48-pin
Vectored interrupt sources	Internal	32	39		39	41	42
	External	6	7	10	8		
Key interrupt		—	4	6	6		

Note 1. The number of PWM outputs varies depending on the setting of channels in use (the number of masters and slaves). For details, see **7.9.3 Operation for the multiple PWM output function**.

Note 2. This applies when the corresponding bit in the peripheral I/O redirection register (PIOR) is set to 1.

The R7F104G products require changing specific bits in the port mode registers (PMxx) and port mode control A registers (PMCAxx) from their initial values to 0. For details, see **4.3 Registers for Controlling the Port Function**.
The R7F104G product with 32 pins requires specific settings for the clock operating mode. For details, see **6.3.1 Clock operation mode control register (CMC)** and **6.3.3 Clock operation status control register (CSC)**.

1.2 Features

Ultra-low power consumption technology

- Operation from a single power supply (V_{DD}) within the 1.6-V to 5.5-V range
- HALT mode
- STOP mode
  High-speed wakeup from the STOP mode is possible.
- SNOOZE mode

RL78 CPU core

- CISC architecture with 3-stage pipeline
- The minimum instruction execution time can be changed from high to ultra-low speed.
  - High speed: 0.03125 µs at 32-MHz operation with the high-speed on-chip oscillator clock
  - Ultra-low speed: 30.5 µs at 32.768-kHz operation with the subsystem clock
- Multiply/divide/multiply & accumulate instructions are supported.
- Address space: 1 MB
- General-purpose registers: (8-bit register × 8) × 4 banks
- On-chip RAM: 12 to 48 KB

Code flash memory

- Code flash memory: 96 to 768 KB
- Block size: 2 KB
- Security function: Prohibition of block erase and rewriting
- On-chip debugging
- Self-programming with boot swapping and flash shield window

Data flash memory

- Data flash memory: 8 KB
- Background operation (BGO): Instructions can be executed from the program memory while rewriting the data flash memory.
- Number of rewrites: 1,000,000 times (typ.)

High-speed on-chip oscillator

- Selectable from among 32 MHz, 24 MHz, 16 MHz, 12 MHz, 8 MHz, 6 MHz, 4 MHz, 3 MHz, 2 MHz, and 1 MHz
- High accuracy: ±1.0% (V_{DD} = 1.8 to 5.5 V, T_A = -20 to +85°C)

Middle-speed on-chip oscillator

- Selectable from among 4 MHz, 2 MHz, and 1 MHz with adjustability

Low-speed on-chip oscillator

- 32.768 kHz (typ.) with adjustability

Operating ambient temperature

- T_A = -40 to +85°C (2D: Consumer applications)
- T_A = -40 to +105°C (3C: Industrial applications)

Power management and reset function

- Power-on-reset circuit (POR)
- Voltage detectors (LVD0 and LVD1)

Data transfer controller (DTC)

- Transfer modes: Normal transfer mode, repeat transfer mode, block transfer mode
- Activation sources: Activated by interrupt sources.
- Chain transfer function

SNOOZE mode sequencer (SMS)

- Calculations and comparison of values by the commands for use in processing by the sequencer can realize intermittent operations where the RL78/G23 does not have to return to normal operation.
- <R> Sequentially handling a total of 32 processes with the use of desired commands from among 20 different ones
- The SNOOZE mode sequencer offers operation with low power consumption without using the CPU, flash memory, and RAM.

Logic and event link controller (ELCL)

- Event signals can be set up between specified peripheral functions.
- The signals can be generated by the input of multiple event signals to the logic circuit.
- Flip-flop circuits are incorporated to handle setting and resetting functions.

Serial interfaces

- Simplified SPI (CSI[Note 1]): 2 to 8 channels
- UART/UART (LIN-bus supported)/UARTA: 3 to 6 channels
- I^2C/Simplified I^2C: 3 to 10 channels

Remote control signal receiver

- 1 channel
- Matching of 4 waveform patterns (header, data 0, data 1, and special data)

Timers

- 16-bit timer: 8 to 16 channels
- 32-bit interval timer: 1 channel in 32-bit counter mode
  2 channels in 16-bit counter mode
  4 channels in 8-bit counter mode
- Realtime clock: 1 channel (counting of one second to 99 years, alarm interrupt, and clock correction)
- Watchdog timer: 1 channel (operates with the low-speed on-chip oscillator clock)

A/D converter

- 8-/10-/12-bit resolution
- Analog input: 7 to 26 channels
- Internal reference voltage (1.48 V (typ.)) and temperature sensor

D/A converter

- 8-bit resolution
- Analog output: 2 channels
- Output voltage: 0 V to V_{DD}
- Realtime output function

Comparator

- 1 or 2 channels
- Operating modes: Comparator high-speed mode and comparator low-speed mode
- The external reference voltage and the internal reference voltage or D/A converter output are selectable as the reference voltage.

Capacitive sensing unit

- Operating voltage condition: V_{DD} = 1.8 to 5.5 V
- Self-capacitance method: A single pin configures a single key, supporting up to 32 keys
- Mutual capacitance method: Matrix configuration with 8 × 8 pins, supporting up to 64 keys

Input/output port pins

- Number of port pins: 26 to 120
  N-ch open-drain I/O (withstand voltage of 6 V): 0 to 4
  N-ch open-drain I/O (V_{DD} withstand voltage**Note 2**/EV_{DD} withstand voltage**Note 3**): 10 to 33
  Controlled current drive port pins: 2 to 8
- Can be set to N-ch open drain or TTL input buffer, and use of an on-chip pull-up resistor can be specified.
- Connectable to a device with different voltage (1.8, 2.5, or 3 V)

Others

- Key interrupt input
- Clock output/buzzer output controller
- Binary-coded decimal (BCD) correction circuit

Note 1. Although the CSI function is generally called SPI, it is also called CSI in this product, so it is referred to as such in this manual.

Note 2. This applies to the 30- to 52-pin products.

Note 3. This applies to the 64- to 128-pin products.

Remark The functions mounted depend on the product. For details, see **1.7 Outline of Functions**.

O ROM, RAM capacities

Code flash memory	Data flash memory	RAM	R7F100G Products of the RL78/G23 Group					
			30 pins	32 pins	36 pins	40 pins	44 pins	48 pins
768 KB	8 KB	48 KB	—	—	—	—	R7F100GFN	R7F100GGN
512 KB	8 KB	48 KB	—	—	—	—	R7F100GFL	R7F100GGL
384 KB	8 KB	32 KB	—	—	—	—	R7F100GFK	R7F100GGK
256 KB	8 KB	24 KB	R7F100GAJ	R7F100GBJ	R7F100GCJ	R7F100GEJ	R7F100GFJ	R7F100GGJ
192 KB	8 KB	20 KB	R7F100GAH	R7F100GBH	R7F100GCH	R7F100GEH	R7F100GFH	R7F100GGH
128 KB	8 KB	16 KB	R7F100GAG	R7F100GBG	R7F100GCG	R7F100GEG	R7F100GFG	R7F100GGG
96 KB	8 KB	12 KB	R7F100GAF	R7F100GBF	R7F100GCF	R7F100GEF	R7F100GFF	R7F100GGF

Code flash memory	Data flash memory	RAM	R7F100G Products of the RL78/G23 Group				
			52 pins	64 pins	80 pins	100 pins	128 pins
768 KB	8 KB	48 KB	R7F100GJN	R7F100GLN	R7F100GMN	R7F100GPN	R7F100GSN
512 KB	8 KB	48 KB	R7F100GJL	R7F100GLL	R7F100GML	R7F100GPL	R7F100GSL
384 KB	8 KB	32 KB	R7F100GJK	R7F100GLK	R7F100GMK	R7F100GPK	R7F100GSK
256 KB	8 KB	24 KB	R7F100GJJ	R7F100GLJ	R7F100GMJ	R7F100GPJ	R7F100GSJ
192 KB	8 KB	20 KB	R7F100GJH	R7F100GLH	R7F100GMH	R7F100GPH	—
128 KB	8 KB	16 KB	R7F100GJG	R7F100GLG	R7F100GMG	R7F100GPG	—
96 KB	8 KB	12 KB	R7F100GJF	R7F100GLF	—	—	—

Code flash memory	Data flash memory	RAM	R7F104G Products of the RL78/G23 Group		
			32 pins	44 pins	48 pins
768 KB	8 KB	48 KB	—	—	—
512 KB	8 KB	48 KB	—	—	—
384 KB	8 KB	32 KB	—	—	—
256 KB	8 KB	24 KB	—	R7F104GFJ	R7F104GGJ
192 KB	8 KB	20 KB	—	—	—
128 KB	8 KB	16 KB	R7F104GBG	—	—
96 KB	8 KB	12 KB	—	—	—

1.3 List of Part Numbers

Figure 1 - 1 Part Number, Memory Size, and Package of RL78/G23

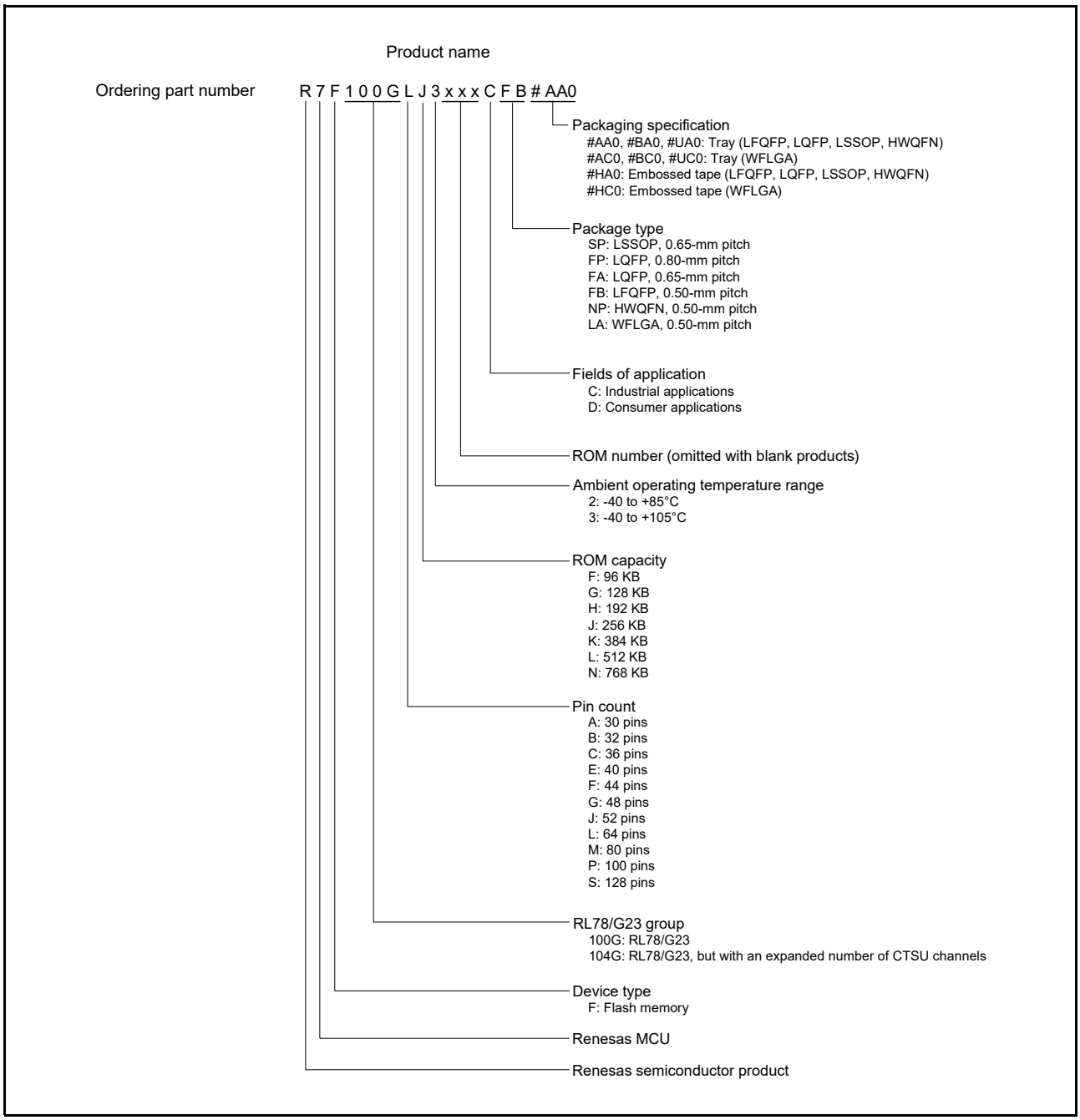


Table 1 - 2 List of Ordering Part Numbers (1/3)

For the R7F100G products

Pin Count	Package	Fields of Application Note	Ordering Part Number		Renesas Code
			Product Name	Packaging Specification	
30	30-pin plastic LSSOP (9.85 mm (300), 0.65-mm pitch)	C	R7F100GAF3CSP, R7F100GAG3CSP, R7F100GAH3CSP, R7F100GAJ3CSP	#AA0, #BA0, #UA0, #HA0	PLSP0030JB-B
		D	R7F100GAF2DSP, R7F100GAG2DSP, R7F100GAH2DSP, R7F100GAJ2DSP		
32	32-pin plastic HWQFN (5 × 5 mm, 0.5-mm pitch)	C	R7F100GBF3CNP, R7F100GBG3CNP, R7F100GBH3CNP, R7F100GBJ3CNP	#AA0, #BA0, #UA0, #HA0	PWQN0032KE-A
		D	R7F100GBF2DNP, R7F100GBG2DNP, R7F100GBH2DNP, R7F100GBJ2DNP		
	32-pin plastic LQFP (7 × 7 mm, 0.80-mm pitch)	C	R7F100GBF3CFP, R7F100GBG3CFP, R7F100GBH3CFP, R7F100GBJ3CFP	#AA0, #BA0, #UA0, #HA0	PLQP0032GB-A, PLQP0032GE-A
		D	R7F100GBF2DFP, R7F100GBG2DFP, R7F100GBH2DFP, R7F100GBJ2DFP		
36	36-pin plastic WFLGA (4 × 4 mm, 0.50-mm pitch)	C	R7F100GCF3CLA, R7F100GCG3CLA, R7F100GCH3CLA, R7F100GCJ3CLA	#AC0, #BC0, #UC0, #HC0	PWLG0036KB-A
		D	R7F100GCF2DLA, R7F100GCG2DLA, R7F100GCH2DLA, R7F100GCJ2DLA		
40	40-pin plastic HWQFN (6 × 6 mm, 0.50-mm pitch)	C	R7F100GEF3CNP, R7F100GEG3CNP, R7F100GEH3CNP, R7F100GEJ3CNP	#AA0, #BA0, #UA0, #HA0	PWQN0040KD-A
		D	R7F100GEF2DNP, R7F100GEG2DNP, R7F100GEH2DNP, R7F100GEJ2DNP		
44	44-pin plastic LQFP (10 × 10 mm, 0.80-mm pitch)	C	R7F100GFF3CFP, R7F100GFG3CFP, R7F100GFH3CFP, R7F100GFJ3CFP, R7F100GFK3CFP, R7F100GFL3CFP, R7F100GFN3CFP	#AA0, #BA0, #UA0, #HA0	PLQP0044GC-A, PLQP0044GE-A
		D	R7F100GFF2DFP, R7F100GFG2DFP, R7F100GFH2DFP, R7F100GFJ2DFP, R7F100GFK2DFP, R7F100GFL2DFP, R7F100GFN2DFP		
48	48-pin plastic LFQFP (7 × 7 mm, 0.50-mm pitch)	C	R7F100GGF3CFB, R7F100GGG3CFB, R7F100GGH3CFB, R7F100GGJ3CFB, R7F100GGK3CFB, R7F100GGL3CFB, R7F100GGN3CFB	#AA0, #BA0, #UA0, #HA0	PLQP0048KB-B, PLQP0048KL-A
		D	R7F100GGF2DFB, R7F100GGG2DFB, R7F100GGH2DFB, R7F100GGJ2DFB, R7F100GGK2DFB, R7F100GGL2DFB, R7F100GGN2DFB		
	48-pin plastic HWQFN (7 × 7 mm, 0.50-mm pitch)	C	R7F100GGF3CNP, R7F100GGG3CNP, R7F100GGH3CNP, R7F100GGJ3CNP, R7F100GGK3CNP, R7F100GGL3CNP, R7F100GGN3CNP	#AA0, #BA0, #UA0, #HA0	PWQN0048KC-A
		D	R7F100GGF2DNP, R7F100GGG2DNP, R7F100GGH2DNP, R7F100GGJ2DNP, R7F100GGK2DNP, R7F100GGL2DNP, R7F100GGN2DNP		

Table 1 - 2 List of Ordering Part Numbers (2/3)

For the R7F100G products

Pin Count	Package	Fields of Application Note	Ordering Part Number		Renesas Code
			Product Name	Packaging Specification	
52	52-pin plastic LQFP (10 × 10 mm, 0.65-mm pitch)	C	R7F100GJF3CFA, R7F100GJG3CFA, R7F100GJH3CFA, R7F100GJJ3CFA, R7F100GJK3CFA, R7F100GJL3CFA, R7F100GJN3CFA	#AA0, #BA0, #UA0, #HA0	PLQP0052JA-A
		D	R7F100GJF2DFA, R7F100GJG2DFA, R7F100GJH2DFA, R7F100GJJ2DFA, R7F100GJK2DFA, R7F100GJL2DFA, R7F100GJN2DFA		
64	64-pin plastic LQFP (12 × 12 mm, 0.65-mm pitch)	C	R7F100GLF3CFA, R7F100GLG3CFA, R7F100GLH3CFA, R7F100GLJ3CFA, R7F100GLK3CFA, R7F100GLL3CFA, R7F100GLN3CFA	#AA0, #BA0, #UA0, #HA0	PLQP0064JA-A
		D	R7F100GLF2DFA, R7F100GLG2DFA, R7F100GLH2DFA, R7F100GLJ2DFA, R7F100GLK2DFA, R7F100GLL2DFA, R7F100GLN2DFA		
	64-pin plastic LFQFP (10 × 10 mm, 0.50-mm pitch)	C	R7F100GLF3CFB, R7F100GLG3CFB, R7F100GLH3CFB, R7F100GLJ3CFB, R7F100GLK3CFB, R7F100GLL3CFB, R7F100GLN3CFB	#AA0, #BA0, #UA0, #HA0	PLQP0064KB-C, PLQP0064KL-A
		D	R7F100GLF2DFB, R7F100GLG2DFB, R7F100GLH2DFB, R7F100GLJ2DFB, R7F100GLK2DFB, R7F100GLL2DFB, R7F100GLN2DFB		
	64-pin plastic WFLGA (5 × 5 mm, 0.50-mm pitch)	C	R7F100GLF3CLA, R7F100GLG3CLA, R7F100GLH3CLA, R7F100GLJ3CLA, R7F100GLK3CLA, R7F100GLL3CLA, R7F100GLN3CLA	#AC0, #BC0, #UC0, #HC0	PWLG0064KB-A
		D	R7F100GLF2DLA, R7F100GLG2DLA, R7F100GLH2DLA, R7F100GLJ2DLA, R7F100GLK2DLA, R7F100GLL2DLA, R7F100GLN2DLA		
80	80-pin plastic LQFP (14 × 14 mm, 0.65-mm pitch)	C	R7F100GMG3CFA, R7F100GMH3CFA, R7F100GMJ3CFA, R7F100GMK3CFA, R7F100GML3CFA, R7F100GMN3CFA	#AA0, #BA0, #UA0, #HA0	PLQP0080JA-B
		D	R7F100GMG2DFA, R7F100GMH2DFA, R7F100GMJ2DFA, R7F100GMK2DFA, R7F100GML2DFA, R7F100GMN2DFA		
	80-pin plastic LFQFP (12 × 12 mm, 0.50-mm pitch)	C	R7F100GMG3CFB, R7F100GMH3CFB, R7F100GMJ3CFB, R7F100GMK3CFB, R7F100GML3CFB, R7F100GMN3CFB	#AA0, #BA0, #UA0, #HA0	PLQP0080KB-B, PLQP0080KJ-A
		D	R7F100GMG2DFB, R7F100GMH2DFB, R7F100GMJ2DFB, R7F100GMK2DFB, R7F100GML2DFB, R7F100GMN2DFB		

Table 1 - 2 List of Ordering Part Numbers (3/3)

For the R7F100G products

Pin Count	Package	Fields of Application Note	Ordering Part Number		Renesas Code
			Product Name	Packaging Specification	
100	100-pin plastic LFQFP (14 × 14 mm, 0.50-mm pitch)	C	R7F100GPG3CFB, R7F100GPH3CFB, R7F100GPJ3CFB, R7F100GPK3CFB, R7F100GPL3CFB, R7F100GPN3CFB	#AA0, #BA0, #UA0, #HA0	PLQP0100KB-B
		D	R7F100GPG2DFB, R7F100GPH2DFB, R7F100GPJ2DFB, R7F100GPK2DFB, R7F100GPL2DFB, R7F100GPN2DFB		
	100-pin plastic LQFP (14 × 20 mm, 0.65-mm pitch)	C	R7F100GPG3CFA, R7F100GPH3CFA, R7F100GPJ3CFA, R7F100GPK3CFA, R7F100GPL3CFA, R7F100GPN3CFA	#AA0, #BA0, #UA0, #HA0	PLQP0100JC-A
		D	R7F100GPG2DFA, R7F100GPH2DFA, R7F100GPJ2DFA, R7F100GPK2DFA, R7F100GPL2DFA, R7F100GPN2DFA		
128	128-pin plastic LFQFP (14 × 20 mm, 0.50-mm pitch)	C	R7F100GSJ3CFB, R7F100GSK3CFB, R7F100GSL3CFB, R7F100GSN3CFB	#AA0, #BA0, #UA0, #HA0	PLQP0128KD-A
		D	R7F100GSJ2DFB, R7F100GSK2DFB, R7F100GSL2DFB, R7F100GSN2DFB		

For the R7F104G products

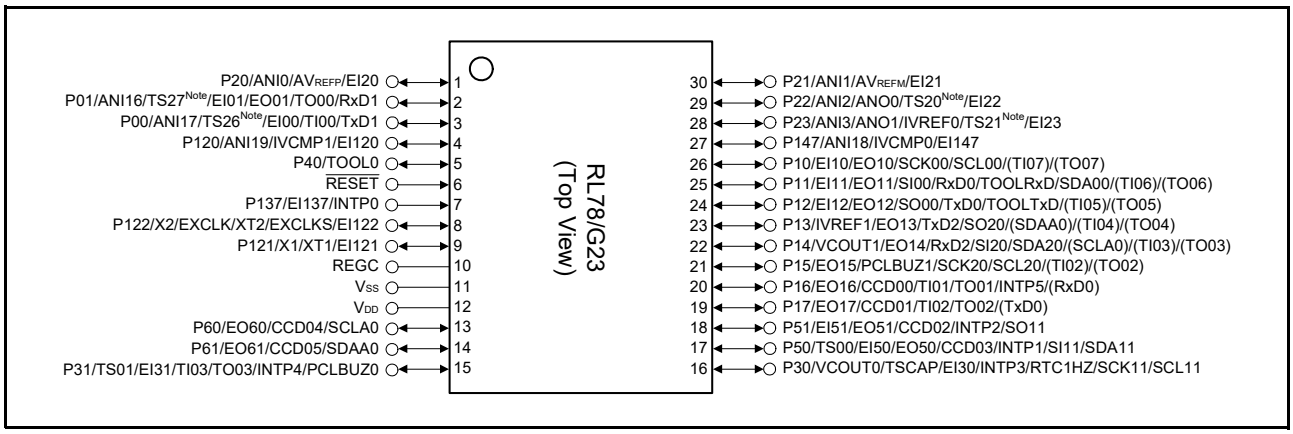
Pin Count	Package	Fields of Application Note	Ordering Part Number		Renesas Code
			Product Name	Packaging Specification	
32	32-pin plastic LQFP (7 × 7 mm, 0.80-mm pitch)	C	R7F104GBG3CFP	#BA0, #UA0, #HA0	PLQP0032GB-A
44	44-pin plastic LQFP (10 × 10 mm, 0.80-mm pitch)	C	R7F104GFJ3CFP	#BA0, #UA0, #HA0	PLQP0044GF-A
48	48-pin plastic LFQFP (7 × 7 mm, 0.50-mm pitch)	C	R7F104GGJ3CFB	#BA0, #UA0, #HA0	PLQP0048KB-B

Note For the fields of application, see **Figure 1 - 1 Part Number, Memory Size, and Package of RL78/G23**.

1.4 Pin Configuration (Top View)

1.4.1 30-pin R7F100G products

- 30-pin plastic LSSOP (9.85 mm (300), 0.65-mm pitch)



Note Not present in products with 128 or fewer Kbytes of code flash memory.

Caution **Connect the REGC pin to Vss via a capacitor (0.47 to 1 µF).**

Remark 1. For pin identification, see **1.5 Pin Identification**.

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

Table 1 - 3 Multiplexed Pin Functions of the 30-pin R7F100G Products (1/2)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
30LSSOP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
1	P20	—	EI20	—	ANI0/ AVREFP	—	—	—	—	—	—	—	—	—	—	—
2	P01	—	EI01/ EO01	—	ANI16	—	—	—	—	TS27 **Note**	TO00	—	RxD1	—	—	—
3	P00	—	EI00	—	ANI17	—	—	—	—	TS26 **Note**	TI00	—	TxD1	—	—	—
4	P120	—	EI120	—	ANI19	—	IVCMP1	—	—	—	—	—	—	—	—	—
5	P40	—	—	TOOL0	—	—	—	—	—	—	—	—	—	—	—	—
6	—	—	—	$\overline{\text{RESET}}$	—	—	—	—	—	—	—	—	—	—	—	—
7	P137	—	EI137		—	—	—	INTP0	—	—	—	—	—	—	—	—
8	P122	—	EI122	X2/XT2/ EXCLK/ EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—
9	P121	—	EI121	X1/XT1	—	—	—	—	—	—	—	—	—	—	—	—
10	—	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—
11	—	—	—	VSS	—	—	—	—	—	—	—	—	—	—	—	—
12	—	—	—	VDD	—	—	—	—	—	—	—	—	—	—	—	—
13	P60	CCD04	EO60	—	—	—	—	—	—	—	—	—	—	SCLA0	—	—
14	P61	CCD05	EO61	—	—	—	—	—	—	—	—	—	—	SDAA0	—	—
15	P31	—	EI31	PCLBUZ0	—	—	—	INTP4	—	TS01	TI03/ TO03	—	—	—	—	—
16	P30	—	EI30	—	—	—	VCOUT0	INTP3	—	TSCAP	—	RTC1HZ	SCK11/ SCL11	—	—	—
17	P50	CCD03	EI50/ EO50	—	—	—	—	INTP1	—	TS00	—	—	SI11/ SDA11	—	—	—
18	P51	CCD02	EI51/ EO51	—	—	—	—	INTP2	—	—	—	—	SO11	—	—	—
19	P17	CCD01	EO17	—	—	—	—		—	—	TI02/ TO02	—	(TxD0)	—	—	—
20	P16	CCD00	EO16	—	—	—	—	INTP5	—	—	TI01/ TO01	—	(RxD0)	—	—	—
21	P15	—	EO15	PCLBUZ1	—	—	—	—	—	—	(TI02)/ (TO02)	—	SCK20/ SCL20	—	—	—
22	P14	—	EO14	—	—	—	VCOUT1	—	—	—	(TI03)/ (TO03)	—	SI20/ RxD2/ SDA20	(SCLA0)	—	—
23	P13	—	EO13	—	—	—	IVREF1	—	—	—	(TI04)/ (TO04)	—	SO20/ TxD2	(SDAA0)	—	—
24	P12	—	EI12/ EO12	TOOLTxD	—	—	—	—	—	—	(TI05)/ (TO05)	—	SO00/ TxD0	—	—	—
25	P11	—	EI11/ EO11	TOOLRxD	—	—	—	—	—	—	(TI06)/(TO06)	—	SI00/ RxD0/ SDA00	—	—	—
26	P10	—	EI10/ EO10	—	—	—	—	—	—	—	(TI07)/ (TO07)	—	SCK00/ SCL00	—	—	—
27	P147	—	EI147	—	ANI18	—	IVCMP0	—	—	—	—	—	—	—	—	—

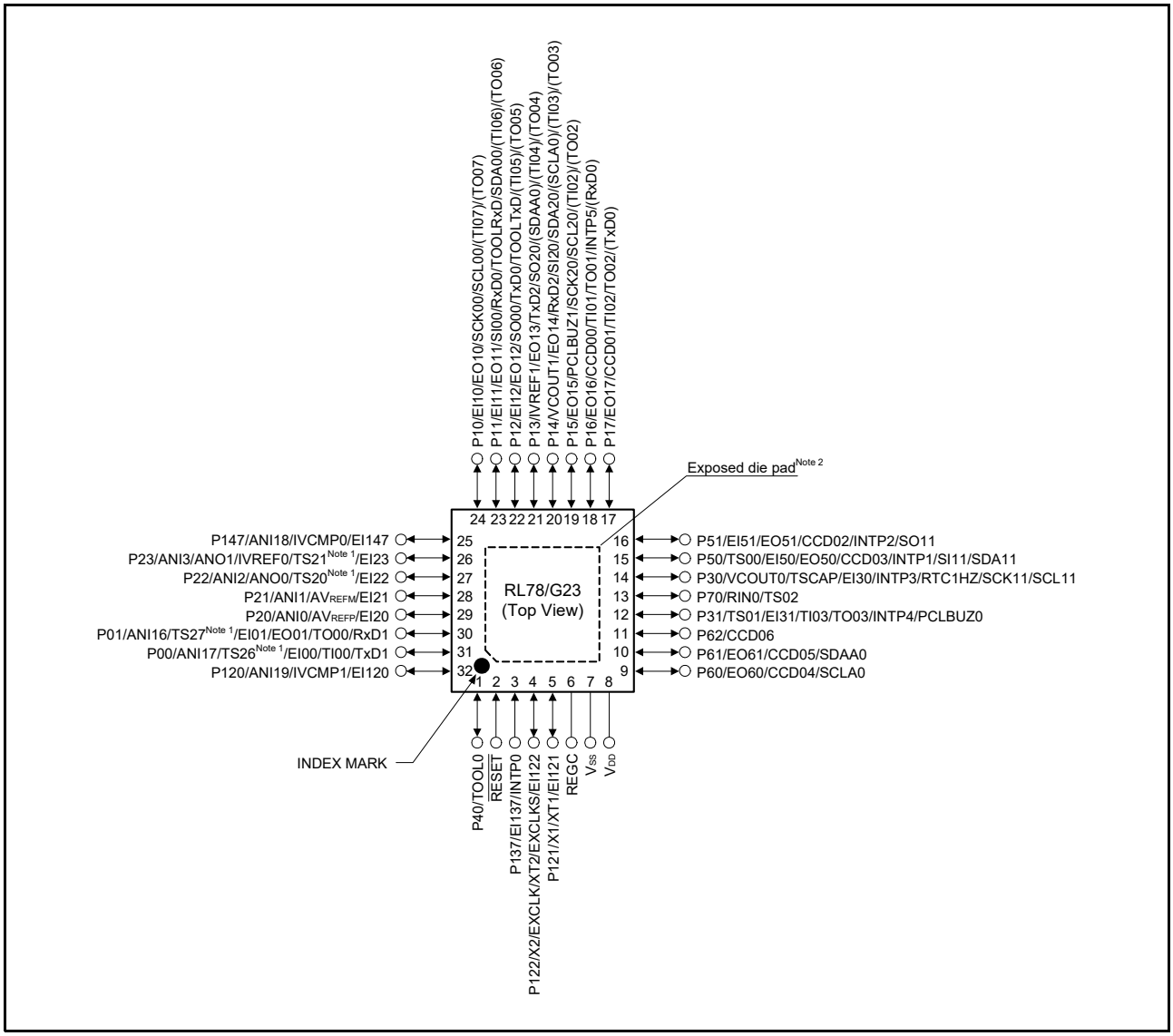
Table 1 - 3 Multiplexed Pin Functions of the 30-pin R7F100G Products (2/2)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
30LSSOP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
28	P23	—	EI23	—	ANI3	ANO1	IVREF0	—	—	TS21 **Note**	—	—	—	—	—	—
29	P22	—	EI22	—	ANI2	ANO0	—	—	—	TS20 **Note**	—	—	—	—	—	—
30	P21	—	EI21	—	ANI1/ AVREFM	—	—	—	—	—	—	—	—	—	—	—

Note Not present in products with 128 or fewer Kbytes of code flash memory.

1.4.2 32-pin R7F100G products

- 32-pin plastic HWQFN (5 × 5 mm, 0.50-mm pitch)
- 32-pin plastic LQFP (7 × 7 mm, 0.80-mm pitch)



Note 1. Not present in products with 128 or fewer Kbytes of code flash memory.

Note 2. The 32-pin plastic LQFP (7 × 7 mm, 0.80-mm pitch) products do not have an exposed die pad.

Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 µF).

Remark 1. For pin identification, see **1.5 Pin Identification**.

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

Remark 3. It is recommended to connect an exposed die pad to Vss.

Table 1 - 4 Multiplexed Pin Functions of the 32-pin R7F100G Products (1/2)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
32HWQFN, 32LQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
1	P40	—	—	TOOL0	—	—	—	—	—	—	—	—	—	—	—	—
2	—	—	—	$\overline{\text{RESET}}$	—	—	—	—	—	—	—	—	—	—	—	—
3	P137	—	EI137	—	—	—	—	INTP0	—	—	—	—	—	—	—	—
4	P122	—	EI122	X2/XT2/ EXCLK/ EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—
5	P121	—	EI121	X1/XT1	—	—	—	—	—	—	—	—	—	—	—	—
6	—	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—
7	—	—	—	V_{SS}	—	—	—	—	—	—	—	—	—	—	—	—
8	—	—	—	V_{DD}	—	—	—	—	—	—	—	—	—	—	—	—
9	P60	CCD04	EO60	—	—	—	—	—	—	—	—	—	—	SCLA0	—	—
10	P61	CCD05	EO61	—	—	—	—	—	—	—	—	—	—	SDAA0	—	—
11	P62	CCD06	—	—	—	—	—	—	—	—	—	—	—	—	—	—
12	P31	—	EI31	PCLBUZ0	—	—	—	INTP4	—	TS01	TI03/ TO03	—	—	—	—	—
13	P70	—	—	—	—	—	—	—	—	TS02	—	—	—	—	—	RIN0
14	P30	—	EI30	—	—	—	VCOUT0	INTP3	—	TSCAP	—	RTC1HZ	SCK11/ SCL11	—	—	—
15	P50	CCD03	EI50/ EO50	—	—	—	—	INTP1	—	TS00	—	—	SI11/ SDA11	—	—	—
16	P51	CCD02	EI51/ EO51	—	—	—	—	INTP2	—	—	—	—	SO11	—	—	—
17	P17	CCD01	EO17	—	—	—	—	—	—	—	TI02/ TO02	—	(TxD0)	—	—	—
18	P16	CCD00	EO16	—	—	—	—	INTP5	—	—	TI01/ TO01	—	(RxD0)	—	—	—
19	P15	—	EO15	PCLBUZ1	—	—	—	—	—	—	(TI02)/ (TO02)	—	SCK20/ SCL20	—	—	—
20	P14	—	EO14	—	—	—	VCOUT1	—	—	—	(TI03)/ (TO03)	—	SI20/ RxD2/ SDA20	(SCLA0)	—	—
21	P13	—	EO13	—	—	—	IVREF1	—	—	—	(TI04)/ (TO04)	—	SO20/ TxD2	(SDAA0)	—	—
22	P12	—	EI12/ EO12	TOOLTxD	—	—	—	—	—	—	(TI05)/ (TO05)	—	SO00/ TxD0	—	—	—
23	P11	—	EI11/ EO11	TOOLRxD	—	—	—	—	—	—	(TI06)/ (TO06)	—	SI00/ RxD0/ SDA00	—	—	—
24	P10	—	EI10/ EO10	—	—	—	—	—	—	—	(TI07)/ (TO07)	—	SCK00/ SCL00	—	—	—
25	P147	—	—	—	ANI18	—	IVCMP0	—	—	—	—	—	—	—	—	—
26	P23	—	EI23	—	ANI3	ANO1	IVREF0	—	—	TS21 **Note**	—	—	—	—	—	—
27	P22	—	EI22	—	ANI2	ANO0	—	—	—	TS20 **Note**	—	—	—	—	—	—

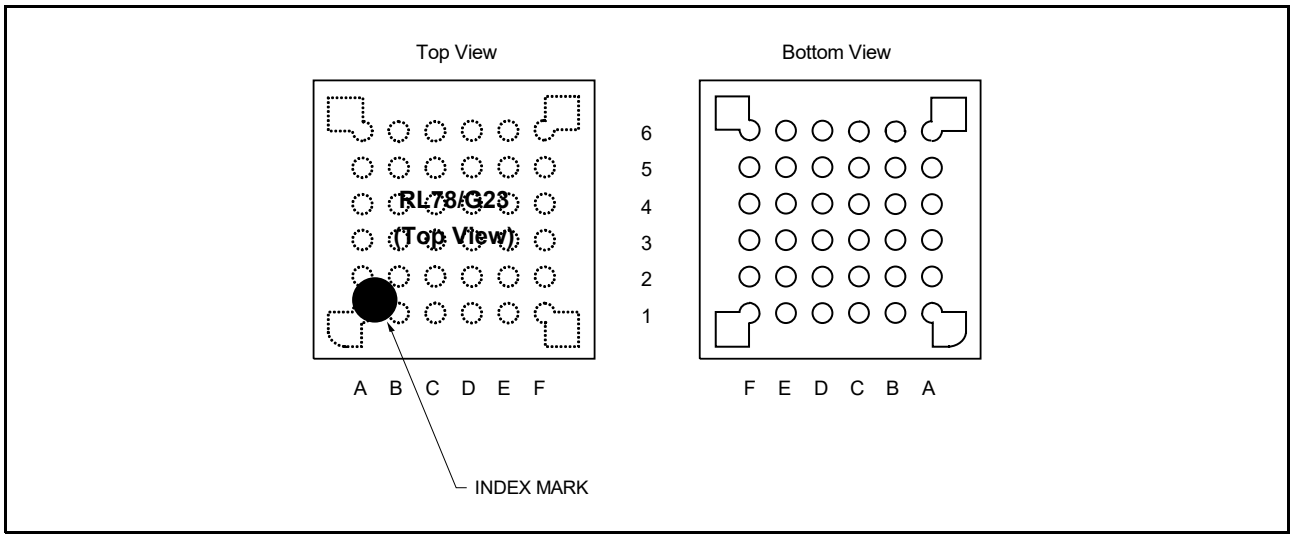
Table 1 - 4 Multiplexed Pin Functions of the 32-pin R7F100G Products (2/2)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
32HWQFN, 32LQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
28	P21	—	EI21	—	ANI1/ AVREFM	—	—	—	—	—	—	—	—	—	—	—
29	P20	—	EI20	—	ANI0/ AVREFP	—	—	—	—	—	—	—	—	—	—	—
30	P01	—	EI01/ EO01	—	ANI16	—	—	—	—	TS27 **Note**	TO00	—	RxD1	—	—	—
31	P00	—	EI00	—	ANI17	—	—	—	—	TS26 **Note**	TI00	—	TxD1	—	—	—
32	P120	—	EI120	—	ANI19	—	IVCMP1	—	—	—	—	—	—	—	—	—

Note Not present in products with 128 or fewer Kbytes of code flash memory.

1.4.3 36-pin R7F100G products

- 36-pin plastic WFLGA (4 × 4 mm, 0.50-mm pitch)



	A	B	C	D	E	F
6	P60/EO60/CCD04/ SCLA0	VDD	P121/X1/XT1/EI121	P122/X2/EXCLK/XT2/ EXCLKS/EI122	P137/EI137/INTP0	P40/TOOL0
5	P62/CCD06	P61/EO61/CCD05/ SDAA0	VSS	REGC	$\overline{\text{RESET}}$	P120/ANI19/IVCMP1/ EI120
4	P72/TS04/SO21/ TxDA0	P71/TS03/SI21/ SDA21/RxDA0	P14/VCOUT1/EO14/ RxD2/SI20/SDA20/ (SCLA0)/(TI03)/ (TO03)	P31/TS01/EI31/TI03/ TO03/INTP4/ PCLBUZ0	P00/TS26Note/EI00/ TI00/TxD1	P01/TS27Note/EI01/ EO01/TO00/RxD1
3	P50/TS00/EI50/EO50/ CCD03/INTP1/SI11/ SDA11	P70/TS02/RIN0/ SCK21/SCL21	P15/EO15/PCLBUZ1/ SCK20/SCL20/ (TI02)/(TO02)	P22/ANI2/ANO0/ TS20Note/EI22	P20/ANI0/AVREFP/ EI20	P21/ANI1/AVREFM/ EI21
2	P30/VCOUT0/TSCAP/ EI30/INTP3/RTC1HZ/ SCK11/SCL11	P16/EO16/CCD00/ TI01/TO01/INTP5/ (RxD0)	P12/EI12/EO12/SO00/ TxD0/TOOLTxD/ (TI05)/(TO05)	P11/EI11/EO11/SI00/ RxD0/TOOLRxD/ SDA00/(TI06)/(TO06)	P24/ANI4/TS22Note	P23/ANI3/ANO1/ IVREF0/TS21Note/ EI23
1	P51/EI51/EO51/ CCD02/INTP2/ SO11	P17/EO17/CCD01/ TI02/TO02/(TxD0)	P13/IVREF1/EO13/ TxD2/SO20/(SDAA0)/ (TI04)/(TO04)	P10/EI10/EO10/ SCK00/SCL00/ (TI07)/(TO07)	P147/ANI18/IVCMP0/ EI147	P25/ANI5/TS23Note

Note Not present in products with 128 or fewer Kbytes of code flash memory.

Caution **Connect the REGC pin to Vss via a capacitor (0.47 to 1 µF).**

Remark 1. For pin identification, see **1.5 Pin Identification**.

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

Table 1 - 5 Multiplexed Pin Functions of the 36-pin R7F100G Products (1/2)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
36WFLGA	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
A1	P51	CCD02	EI51/ EO51	—	—	—	—	INTP2	—	—	—	—	SO11	—	—	—
A2	P30	—	EI30	—	—	—	VCOUT0	INTP3	—	TSCAP	—	RTC1HZ	SCK11/ SCL11	—	—	—
A3	P50	CCD03	EI50/ EO50	—	—	—	—	INTP1	—	TS00	—	—	SI11/ SDA11	—	—	—
A4	P72	—	—	—	—	—	—	—	—	TS04	—	—	SO21	—	TxDA0	—
A5	P62	CCD06	—	—	—	—	—	—	—	—	—	—	—	—	—	—
A6	P60	CCD04	EO60	—	—	—	—	—	—	—	—	—	—	SCLA0	—	—
B1	P17	CCD01	EO17	—	—	—	—	—	—	—	TI02/ TO02	—	(TxD0)	—	—	—
B2	P16	CCD00	EO16	—	—	—	—	INTP5	—	—	TI01/ TO01	—	(RxD0)	—	—	—
B3	P70	—	—	—	—	—	—	—	—	TS02	—	—	SCK21/ SCL21	—	—	RIN0
B4	P71	—	—	—	—	—	—	—	—	TS03	—	—	SI21/ SDA21	—	RxDA0	—
B5	P61	CCD05	EO61	—	—	—	—	—	—	—	—	—	—	SDAA0	—	—
B6	—	—	—	VDD	—	—	—	—	—	—	—	—	—	—	—	—
C1	P13	—	EO13	—	—	—	IVREF1	—	—	—	(TI04)/ (TO04)	—	SO20/ TxD2	(SDAA0)	—	—
C2	P12	—	EI12/ EO12	TOOLTxD	—	—	—	—	—	—	(TI05)/ (TO05)	—	SO00/ TxD0	—	—	—
C3	P15	—	EO15	PCLBUZ1	—	—	—	—	—	—	(TI02)/ (TO02)	—	SCK20/ SCL20	—	—	—
C4	P14	—	EO14	—	—	—	VCOUT1	—	—	—	(TI03)/ (TO03)	—	SI20/ RxD2/ SDA20	(SCLA0)	—	—
C5	—	—	—	Vss	—	—	—	—	—	—	—	—	—	—	—	—
C6	P121	—	EI121	X1/XT1	—	—	—	—	—	—	—	—	—	—	—	—
D1	P10	—	EI10/ EO10	—	—	—	—	—	—	—	(TI07)/ (TO07)	—	SCK00/ SCL00	—	—	—
D2	P11	—	EI11/ EO11	TOOLRxD	—	—	—	—	—	—	(TI06)/ (TO06)	—	SI00/ RxD0/ SDA00	—	—	—
D3	P22	—	EI22	—	ANI2	ANO0	—	—	—	TS20 **Note**	—	—	—	—	—	—
D4	P31	—	EI31	PCLBUZ0	—	—	—	INTP4	—	TS01	TI03/ TO03	—	—	—	—	—
D5	—	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—
D6	P122	—	EI122	X2/XT2/ EXCLK/ EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—
E1	P147	—	EI147	—	ANI18	—	IVCMP0	—	—	—	—	—	—	—	—	—
E2	P24	—	—	—	ANI4	—	—	—	—	TS22 **Note**	—	—	—	—	—	—

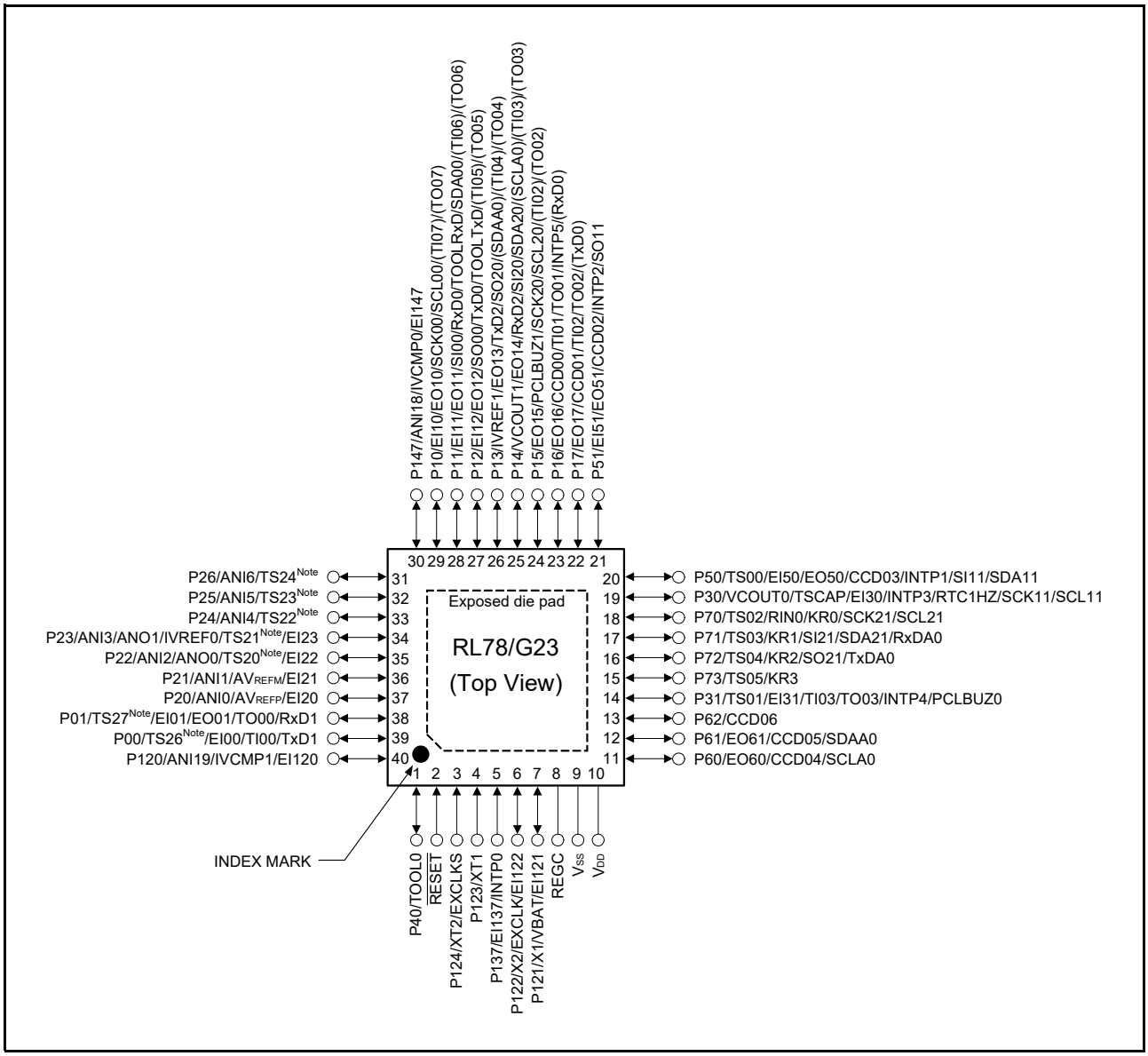
Table 1 - 5 Multiplexed Pin Functions of the 36-pin R7F100G Products (2/2)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
36WFLGA	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
E3	P20	—	EI20	—	ANI0/ AVREFP	—	—	—	—	—	—	—	—	—	—	—
E4	P00	—	EI00	—	—	—	—	—	—	TS26 **Note**	TI00	—	TxD1	—	—	—
E5	—	—	—	$\overline{\text{RESET}}$	—	—	—	—	—	—	—	—	—	—	—	—
E6	P137	—	EI137	—	—	—	—	INTP0	—	—	—	—	—	—	—	—
F1	P25	—	—	—	ANI5	—	—	—	—	TS23 **Note**	—	—	—	—	—	—
F2	P23	—	EI23	—	ANI3	ANO1	IVREF0	—	—	TS21 **Note**	—	—	—	—	—	—
F3	P21	—	EI21	—	ANI1/ AVREFM	—	—	—	—	—	—	—	—	—	—	—
F4	P01	—	EI01/ EO01	—	—	—	—	—	—	TS27 **Note**	TO00	—	RxD1	—	—	—
F5	P120	—	EI120	—	ANI19	—	IVCMP1	—	—	—	—	—	—	—	—	—
F6	P40	—	—	TOOL0	—	—	—	—	—	—	—	—	—	—	—	—

Note Not present in products with 128 or fewer Kbytes of code flash memory.

1.4.4 40-pin R7F100G products

- 40-pin plastic HWQFN (6 × 6 mm, 0.50-mm pitch)



Note Not present in products with 128 or fewer Kbytes of code flash memory.

Caution **Connect the REGC pin to Vss via a capacitor (0.47 to 1 µF).**

Remark 1. For pin identification, see **1.5 Pin Identification**.

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

Remark 3. It is recommended to connect an exposed die pad to Vss.

Table 1 - 6 Multiplexed Pin Functions of the 40-pin R7F100G Products (1/2)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
40HWQFN	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
1	P40	—	—	TOOL0	—	—	—	—	—	—	—	—	—	—	—	—
2	—	—	—	$\overline{\text{RESET}}$	—	—	—	—	—	—	—	—	—	—	—	—
3	P124	—	—	XT2/ EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—
4	P123	—	—	XT1	—	—	—	—	—	—	—	—	—	—	—	—
5	P137	—	EI137	—	—	—	—	INTP0	—	—	—	—	—	—	—	—
6	P122	—	EI122	X2/EXCLK	—	—	—	—	—	—	—	—	—	—	—	—
7	P121	—	EI121	X1/VBAT	—	—	—	—	—	—	—	—	—	—	—	—
8	—	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—
9	—	—	—	V_{SS}	—	—	—	—	—	—	—	—	—	—	—	—
10	—	—	—	V_{DD}	—	—	—	—	—	—	—	—	—	—	—	—
11	P60	CCD04	EO60	—	—	—	—	—	—	—	—	—	—	SCLA0	—	—
12	P61	CCD05	EO61	—	—	—	—	—	—	—	—	—	—	SDAA0	—	—
13	P62	CCD06	—	—	—	—	—	—	—	—	—	—	—	—	—	—
14	P31	—	EI31	PCLBUZ0	—	—	—	INTP4	—	TS01	TI03/ TO03	—	—	—	—	—
15	P73	—	—	—	—	—	—	—	KR3	TS05	—	—	—	—	—	—
16	P72	—	—	—	—	—	—	—	KR2	TS04	—	—	SO21	—	TxDA0	—
17	P71	—	—	—	—	—	—	—	KR1	TS03	—	—	SI21/ SDA21	—	RxDA0	—
18	P70	—	—	—	—	—	—	—	KR0	TS02	—	—	SCK21/ SCL21	—	—	RIN0
19	P30	—	EI30	—	—	—	VCOUT0	INTP3	—	TSCAP	—	RTC1HZ	SCK11/ SCL11	—	—	—
20	P50	CCD03	EI50/ EO50	—	—	—	—	INTP1	—	TS00	—	—	SI11/ SDA11	—	—	—
21	P51	CCD02	EI51/ EO51	—	—	—	—	INTP2	—	—	—	—	SO11	—	—	—
22	P17	CCD01	EO17	—	—	—	—	—	—	—	TI02/ TO02	—	(TxD0)	—	—	—
23	P16	CCD00	EO16	—	—	—	—	INTP5	—	—	TI01/ TO01	—	(RxD0)	—	—	—
24	P15	—	EO15	PCLBUZ1	—	—	—	—	—	—	(TI02)/ (TO02)	—	SCK20/ SCL20	—	—	—
25	P14	—	EO14	—	—	—	VCOUT1	—	—	—	(TI03)/ (TO03)	—	SI20/ RxD2/ SDA20	(SCLA0)	—	—
26	P13	—	EO13	—	—	—	IVREF1	—	—	—	(TI04)/ (TO04)	—	SO20/ TxD2	(SDAA0)	—	—
27	P12	—	EI12/ EO12	TOOLTxD	—	—	—	—	—	—	(TI05)/ (TO05)	—	SO00/ TxD0	—	—	—
28	P11	—	EI11/ EO11	TOOLRxD	—	—	—	—	—	—	(TI06)/ (TO06)	—	SI00/ RxD0/ SDA00	—	—	—

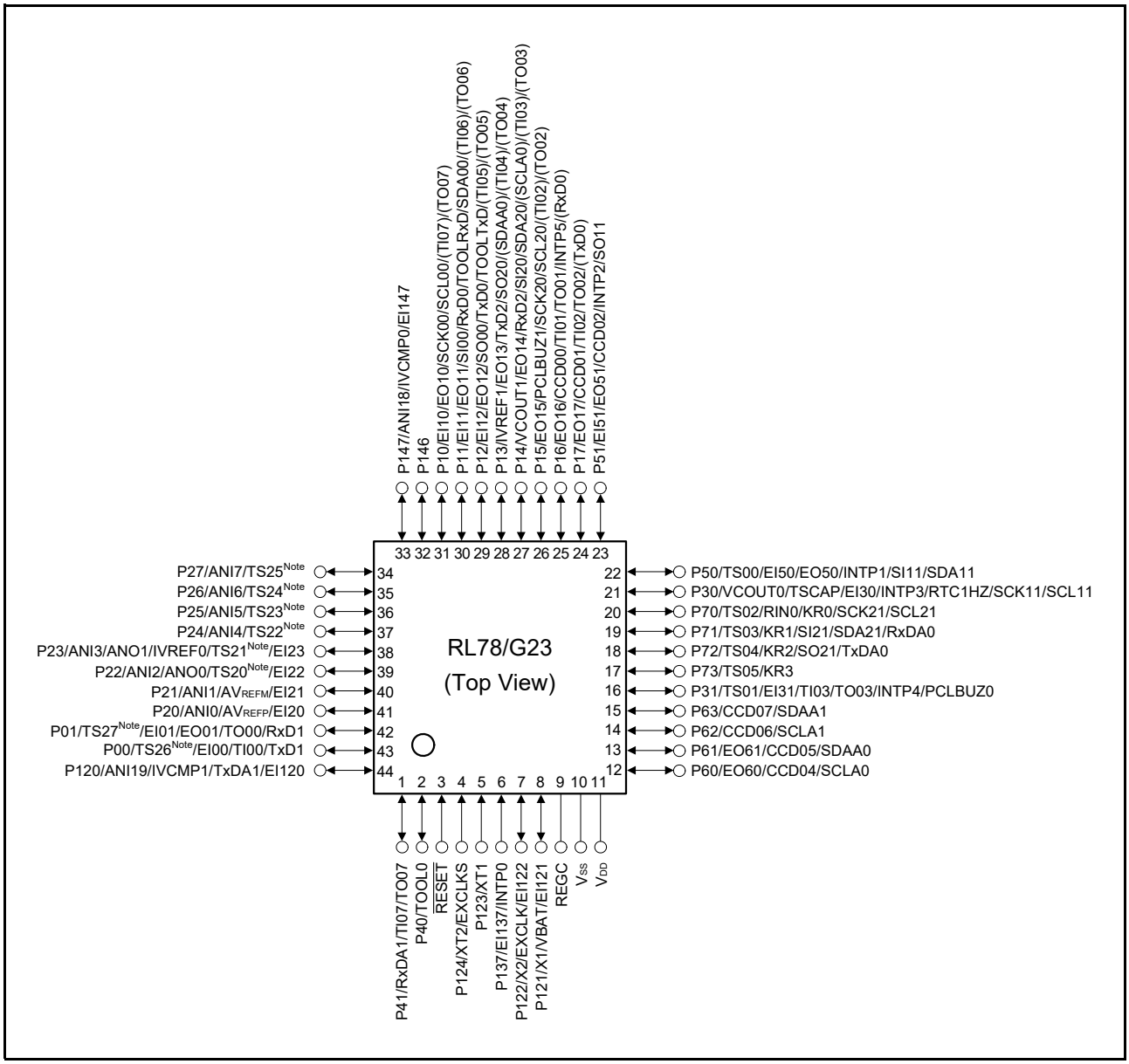
Table 1 - 6 Multiplexed Pin Functions of the 40-pin R7F100G Products (2/2)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
40HWQFN	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
29	P10	—	EI10/ EO10	—	—	—	—	—	—	—	(TI07)/ (TO07)	—	SCK00/ SCL00	—	—	—
30	P147	—	EI147	—	ANI18	—	IVCMP0	—	—	—	—	—	—	—	—	—
31	P26	—	—	—	ANI6	—	—	—	—	TS24 **Note**	—	—	—	—	—	—
32	P25	—	—	—	ANI5	—	—	—	—	TS23 **Note**	—	—	—	—	—	—
33	P24	—	—	—	ANI4	—	—	—	—	TS22 **Note**	—	—	—	—	—	—
34	P23	—	EI23	—	ANI3	ANO1	IVREF0	—	—	TS21 **Note**	—	—	—	—	—	—
35	P22	—	EI22	—	ANI2	ANO0	—	—	—	TS20 **Note**	—	—	—	—	—	—
36	P21	—	EI21	—	ANI1/ AVREFM	—	—	—	—	—	—	—	—	—	—	—
37	P20	—	EI20	—	ANI0/ AVREFP	—	—	—	—	—	—	—	—	—	—	—
38	P01	—	EI01/ EO01	—	—	—	—	—	—	TS27 **Note**	TO00	—	RxD1	—	—	—
39	P00	—	EI00	—	—	—	—	—	—	TS26 **Note**	TI00	—	TxD1	—	—	—
40	P120	—	EI120	—	ANI19	—	IVCMP1	—	—	—	—	—	—	—	—	—

Note Not present in products with 128 or fewer Kbytes of code flash memory.

1.4.5 44-pin R7F100G products

- 44-pin plastic LQFP (10 × 10 mm, 0.80-mm pitch)



Note Not present in products with 128 or fewer Kbytes of code flash memory.

Caution **Connect the REGC pin to Vss via a capacitor (0.47 to 1 µF).**

Remark 1. For pin identification, see **1.5 Pin Identification**.

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

Table 1 - 7 Multiplexed Pin Functions of the 44-pin R7F100G Products (1/2)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
44LQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
1	P41	—	—	—	—	—	—	—	—	—	TI07/ TO07	—	—	—	RxDA1	—
2	P40	—	—	TOOL0	—	—	—	—	—	—	—	—	—	—	—	—
3	—	—		$\overline{\text{RESET}}$	—	—	—	—	—	—	—	—	—	—	—	—
4	P124	—	—	XT2/ EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—
5	P123	—	—	XT1	—	—	—	—	—	—	—	—	—	—	—	—
6	P137	—	EI137	—	—	—	—	INTP0	—	—	—	—	—	—	—	—
7	P122	—	EI122	X2/EXCLK	—	—	—	—	—	—	—	—	—	—	—	—
8	P121	—	EI121	X1/VBAT	—	—	—	—	—	—	—	—	—	—	—	—
9	—	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—
10	—	—	—	V_{SS}	—	—	—	—	—	—	—	—	—	—	—	—
11	—	—	—	V_{DD}	—	—	—	—	—	—	—	—	—	—	—	—
12	P60	CCD04	EO60	—	—	—	—	—	—	—	—	—	—	SCLA0	—	—
13	P61	CCD05	EO61	—	—	—	—	—	—	—	—	—	—	SDAA0	—	—
14	P62	CCD06	—	—	—	—	—	—	—	—	—	—	—	SCLA1	—	—
15	P63	CCD07	—	—	—	—	—	—	—	—	—	—	—	SDAA1	—	—
16	P31	—	EI31	PCLBUZ0	—	—	—	INTP4	—	TS01	TI03/ TO03	—	—	—	—	—
17	P73	—	—	—	—	—	—	—	KR3	TS05	—	—	—	—	—	—
18	P72	—	—	—	—	—	—	—	KR2	TS04	—	—	SO21	—	TxDA0	—
19	P71	—	—	—	—	—	—	—	KR1	TS03	—	—	SI21/ SDA21	—	RxDA0	—
20	P70	—	—	—	—	—	—	—	KR0	TS02	—	—	SCK21/ SCL21	—	—	RIN0
21	P30	—	EI30	—	—	—	VCOUT0	INTP3	—	TSCAP	—	RTC1HZ	SCK11/ SCL11	—	—	—
22	P50	—	EI50/ EO50	—	—	—	—	INTP1	—	TS00	—	—	SI11/ SDA11	—	—	—
23	P51	CCD02	EI51/ EO51	—	—	—	—	INTP2	—	—	—	—	SO11	—	—	—
24	P17	CCD01	EO17	—	—	—	—		—	—	TI02/ TO02	—	(TxD0)	—	—	—
25	P16	CCD00	EO16	—	—	—	—	INTP5	—	—	TI01/ TO01	—	(RxD0)	—	—	—
26	P15	—	EO15	PCLBUZ1	—	—	—	—	—	—	(TI02)/ (TO02)	—	SCK20/ SCL20	—	—	—
27	P14	—	EO14	—	—	—	VCOUT1	—	—	—	(TI03)/ (TO03)	—	SI20/ RxD2/ SDA20	(SCLA0)	—	—
28	P13	—	EO13	—	—	—	IVREF1	—	—	—	(TI04)/ (TO04)	—	SO20/ TxD2	(SDAA0)	—	—
29	P12	—	EI12/ EO12	TOOLTxD	—	—	—	—	—	—	(TI05)/ (TO05)	—	SO00/ TxD0	—	—	—

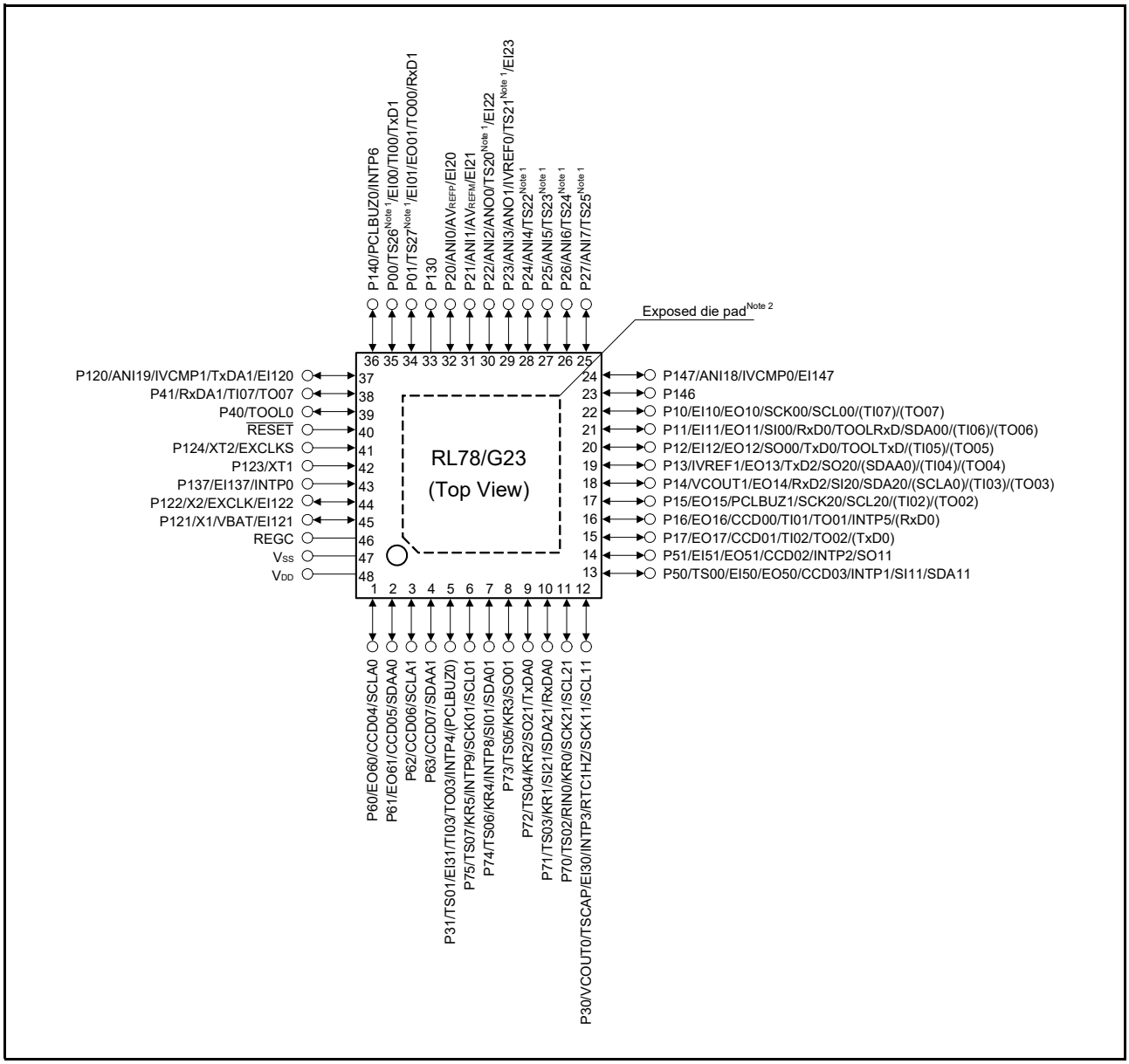
Table 1 - 7 Multiplexed Pin Functions of the 44-pin R7F100G Products (2/2)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
44LQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
30	P11	—	EI11/ EO11	TOOLRxD	—	—	—	—	—	—	(TI06)/ (TO06)	—	SI00/ RxD0/ SDA00	—	—	—
31	P10	—	EI10/ EO10	—	—	—	—	—	—	—	(TI07)/ (TO07)	—	SCK00/ SCL00	—	—	—
32	P146	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
33	P147	—	EI147	—	ANI18	—	IVCMP0	—	—	—	—	—	—	—	—	—
34	P27	—	—	—	ANI7	—	—	—	—	TS25 **Note**	—	—	—	—	—	—
35	P26	—	—	—	ANI6	—	—	—	—	TS24 **Note**	—	—	—	—	—	—
36	P25	—	—	—	ANI5	—	—	—	—	TS23 **Note**	—	—	—	—	—	—
37	P24	—	—	—	ANI4	—	—	—	—	TS22 **Note**	—	—	—	—	—	—
38	P23	—	EI23	—	ANI3	ANO1	IVREF0	—	—	TS21 **Note**	—	—	—	—	—	—
39	P22	—	EI22	—	ANI2	ANO0	—	—	—	TS20 **Note**	—	—	—	—	—	—
40	P21	—	EI21	—	ANI1/ AVREFM	—	—	—	—	—	—	—	—	—	—	—
41	P20	—	EI20	—	ANI0/ AVREFP	—	—	—	—	—	—	—	—	—	—	—
42	P01	—	EI01/ EO01	—	—	—	—	—	—	TS27 **Note**	TO00	—	RxD1	—	—	—
43	P00	—	EI00	—	—	—	—	—	—	TS26 **Note**	TI00	—	TxD1	—	—	—
44	P120	—	EI120	—	ANI19	—	IVCMP1	—	—	—	—	—	—	—	TxDA1	—

Note Not present in products with 128 or fewer Kbytes of code flash memory.

1.4.6 48-pin R7F100G products

- 48-pin plastic LFQFP (7 × 7 mm, 0.50-mm pitch)
- 48-pin plastic HWQFN (7 × 7 mm, 0.50-mm pitch)



Note 1. Not present in products with 128 or fewer Kbytes of code flash memory.

Note 2. The 48-pin plastic LFQFP (7 × 7 mm, 0.50-mm pitch) products do not have an exposed die pad.

Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 µF).

Remark 1. For pin identification, see **1.5 Pin Identification**.

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

Remark 3. It is recommended to connect an exposed die pad to Vss.

Table 1 - 8 Multiplexed Pin Functions of the 48-pin R7F100G Products (1/2)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
48LFQFP, 48HWQFN	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
1	P60	CCD04	EO60	—	—	—	—	—	—	—	—	—	—	SCLA0	—	—
2	P61	CCD05	EO61	—	—	—	—	—	—	—	—	—	—	SDAA0	—	—
3	P62	CCD06	—	—	—	—	—	—	—	—	—	—	—	SCLA1	—	—
4	P63	CCD07	—	—	—	—	—	—	—	—	—	—	—	SDAA1	—	—
5	P31	—	EI31	(PCLBUZ0)	—	—	—	INTP4	—	TS01	TI03/ TO03	—	—	—	—	—
6	P75	—	—	—	—	—	—	INTP9	KR5	TS07	—	—	SCK01/ SCL01	—	—	—
7	P74	—	—	—	—	—	—	INTP8	KR4	TS06	—	—	SI01/ SDA01	—	—	—
8	P73	—	—	—	—	—	—	—	KR3	TS05	—	—	SO01	—	—	—
9	P72	—	—	—	—	—	—	—	KR2	TS04	—	—	SO21	—	TxDA0	—
10	P71	—	—	—	—	—	—	—	KR1	TS03	—	—	SI21/ SDA21	—	RxDA0	—
11	P70	—	—	—	—	—	—	—	KR0	TS02	—	—	SCK21/ SCL21	—	—	RIN0
12	P30	—	EI30	—	—	—	VCOUT0	INTP3	—	TSCAP	—	RTC1HZ	SCK11/ SCL11	—	—	—
13	P50	CCD03	EI50/ EO50	—	—	—	—	INTP1	—	TS00	—	—	SI11/ SDA11	—	—	—
14	P51	CCD02	EI51/ EO51	—	—	—	—	INTP2	—	—	—	—	SO11	—	—	—
15	P17	CCD01	EO17	—	—	—	—	—	—	—	TI02/ TO02	—	(TxD0)	—	—	—
16	P16	CCD00	EO16	—	—	—	—	INTP5	—	—	TI01/ TO01	—	(RxD0)	—	—	—
17	P15	—	EO15	PCLBUZ1	—	—	—	—	—	—	(TI02)/ (TO02)	—	SCK20/ SCL20	—	—	—
18	P14	—	EO14	—	—	—	VCOUT1	—	—	—	(TI03)/ (TO03)	—	SI20/ RxD2/ SDA20	(SCLA0)	—	—
19	P13	—	EO13	—	—	—	IVREF1	—	—	—	(TI04)/ (TO04)	—	SO20/ TxD2	(SDAA0)	—	—
20	P12	—	EI12/ EO12	TOOLTxD	—	—	—	—	—	—	(TI05)/ (TO05)	—	SO00/ TxD0	—	—	—
21	P11	—	EI11/ EO11	TOOLRxD	—	—	—	—	—	—	(TI06)/ (TO06)	—	SI00/ RxD0/ SDA00	—	—	—
22	P10	—	EI10/ EO10	—	—	—	—	—	—	—	(TI07)/ (TO07)	—	SCK00/ SCL00	—	—	—
23	P146	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
24	P147	—	EI147	—	ANI18	—	IVCMP0	—	—	—	—	—	—	—	—	—
25	P27	—	—	—	ANI7	—	—	—	—	TS25 **Note**	—	—	—	—	—	—
26	P26	—	—	—	ANI6	—	—	—	—	TS24 **Note**	—	—	—	—	—	—

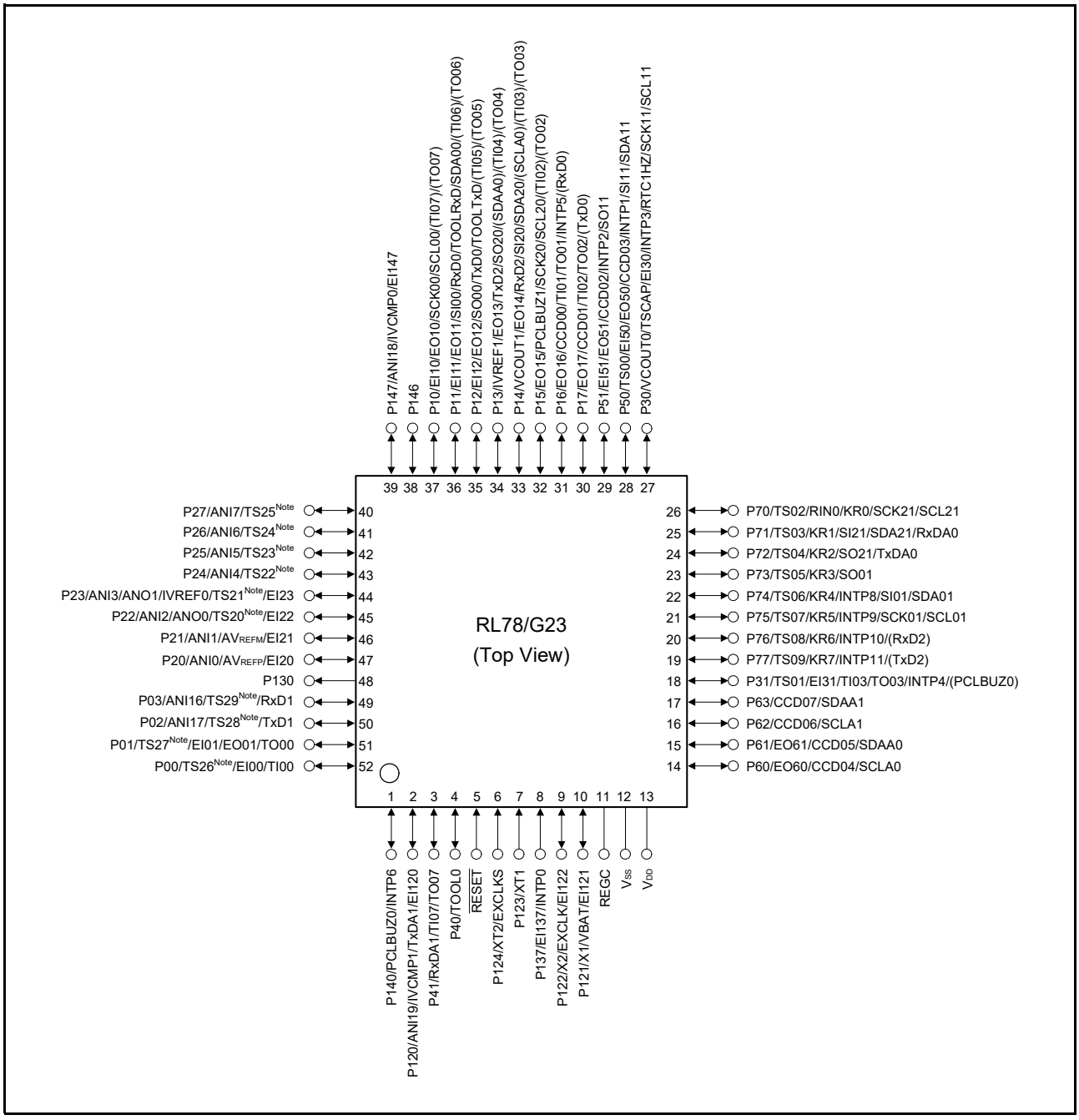
Table 1 - 8 Multiplexed Pin Functions of the 48-pin R7F100G Products (2/2)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
48LFQFP, 48HWQFN	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
27	P25	—	—	—	ANI5	—	—	—	—	TS23 **Note**	—	—	—	—	—	—
28	P24	—	—	—	ANI4	—	—	—	—	TS22 **Note**	—	—	—	—	—	—
29	P23	—	EI23	—	ANI3	ANO1	IVREF0	—	—	TS21 **Note**	—	—	—	—	—	—
30	P22	—	EI22	—	ANI2	ANO0	—	—	—	TS20 **Note**	—	—	—	—	—	—
31	P21	—	EI21	—	ANI1/ AVREFM	—	—	—	—	—	—	—	—	—	—	—
32	P20	—	EI20	—	ANI0/ AVREFP	—	—	—	—	—	—	—	—	—	—	—
33	P130	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
34	P01	—	EI01/ EO01	—	—	—	—	—	—	TS27 **Note**	TO00	—	RxD1	—	—	—
35	P00	—	EI00	—	—	—	—	—	—	TS26 **Note**	TI00	—	TxD1	—	—	—
36	P140	—	—	PCLBUZ0	—	—	—	INTP6	—	—	—	—	—	—	—	—
37	P120	—	EI120	—	ANI19	—	IVCMP1	—	—	—	—	—	—	—	TxDA1	—
38	P41	—	—	—	—	—	—	—	—	—	TI07/ TO07	—	—	—	RxDA1	—
39	P40	—	—	TOOL0	—	—	—	—	—	—	—	—	—	—	—	—
40	—	—	—	$\overline{\text{RESET}}$	—	—	—	—	—	—	—	—	—	—	—	—
41	P124	—	—	XT2/ EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—
42	P123	—	—	XT1	—	—	—	—	—	—	—	—	—	—	—	—
43	P137	—	EI137	—	—	—	—	INTP0	—	—	—	—	—	—	—	—
44	P122	—	EI122	X2/EXCLK	—	—	—	—	—	—	—	—	—	—	—	—
45	P121	—	EI121	X1/VBAT	—	—	—	—	—	—	—	—	—	—	—	—
46	—	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—
47	—	—	—	VSS	—	—	—	—	—	—	—	—	—	—	—	—
48	—	—	—	VDD	—	—	—	—	—	—	—	—	—	—	—	—

Note Not present in products with 128 or fewer Kbytes of code flash memory.

1.4.7 52-pin R7F100G products

- 52-pin plastic LQFP (10 × 10 mm, 0.65-mm pitch)



Note Not present in products with 128 or fewer Kbytes of code flash memory.

Caution **Connect the REGC pin to Vss via a capacitor (0.47 to 1 µF).**

Remark 1. For pin identification, see **1.5 Pin Identification**.

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

Table 1 - 9 Multiplexed Pin Functions of the 52-pin R7F100G Products (1/2)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
52LFQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
1	P140	—	—	PCLBUZ0	—	—	—	INTP6	—	—	—	—	—	—	—	—
2	P120	—	EI120	—	ANI19	—	IVCMP1	—	—	—	—	—	—	—	TxDA1	—
3	P41	—	—	—	—	—	—	—	—	—	TI07/ TO07	—	—	—	RxDA1	—
4	P40	—	—	TOOL0	—	—	—	—	—	—	—	—	—	—	—	—
5	—	—	—	$\overline{\text{RESET}}$	—	—	—	—	—	—	—	—	—	—	—	—
6	P124	—	—	XT2/ EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—
7	P123	—	—	XT1	—	—	—	—	—	—	—	—	—	—	—	—
8	P137	—	EI137	—	—	—	—	INTP0	—	—	—	—	—	—	—	—
9	P122	—	EI122	X2/EXCLK	—	—	—	—	—	—	—	—	—	—	—	—
10	P121	—	EI121	X1/VBAT	—	—	—	—	—	—	—	—	—	—	—	—
11	—	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—
12	—	—	—	V_{SS}	—	—	—	—	—	—	—	—	—	—	—	—
13	—	—	—	V_{DD}	—	—	—	—	—	—	—	—	—	—	—	—
14	P60	CCD04	EO60	—	—	—	—	—	—	—	—	—	—	SCLA0	—	—
15	P61	CCD05	EO61	—	—	—	—	—	—	—	—	—	—	SDAA0	—	—
16	P62	CCD06	—	—	—	—	—	—	—	—	—	—	—	SCLA1	—	—
17	P63	CCD07	—	—	—	—	—	—	—	—	—	—	—	SDAA1	—	—
18	P31	—	EI31	(PCLBUZ0)	—	—	—	INTP4	—	TS01	TI03/ TO03	—	—	—	—	—
19	P77	—	—	—	—	—	—	INTP11	KR7	TS09	—	—	(TxD2)	—	—	—
20	P76	—	—	—	—	—	—	INTP10	KR6	TS08	—	—	(RxD2)	—	—	—
21	P75	—	—	—	—	—	—	INTP9	KR5	TS07	—	—	SCK01/ SCL01	—	—	—
22	P74	—	—	—	—	—	—	INTP8	KR4	TS06	—	—	SI01/ SDA01	—	—	—
23	P73	—	—	—	—	—	—	—	KR3	TS05	—	—	SO01	—	—	—
24	P72	—	—	—	—	—	—	—	KR2	TS04	—	—	SO21	—	TxDA0	—
25	P71	—	—	—	—	—	—	—	KR1	TS03	—	—	SI21/ SDA21	—	RxDA0	—
26	P70	—	—	—	—	—	—	—	KR0	TS02	—	—	SCK21/ SCL21	—	—	RIN0
27	P30	—	EI30	—	—	—	VCOUT0	INTP3	—	TSCAP	—	RTC1HZ	SCK11/ SCL11	—	—	—
28	P50	CCD03	EI50/ EO50	—	—	—	—	INTP1	—	TS00	—	—	SI11/ SDA11	—	—	—
29	P51	CCD02	EI51/ EO51	—	—	—	—	INTP2	—	—	—	—	SO11	—	—	—
30	P17	CCD01	EO17	—	—	—	—	—	—	—	TI02/ TO02	—	(TxD0)	—	—	—
31	P16	CCD00	EO16	—	—	—	—	INTP5	—	—	TI01/ TO01	—	(RxD0)	—	—	—

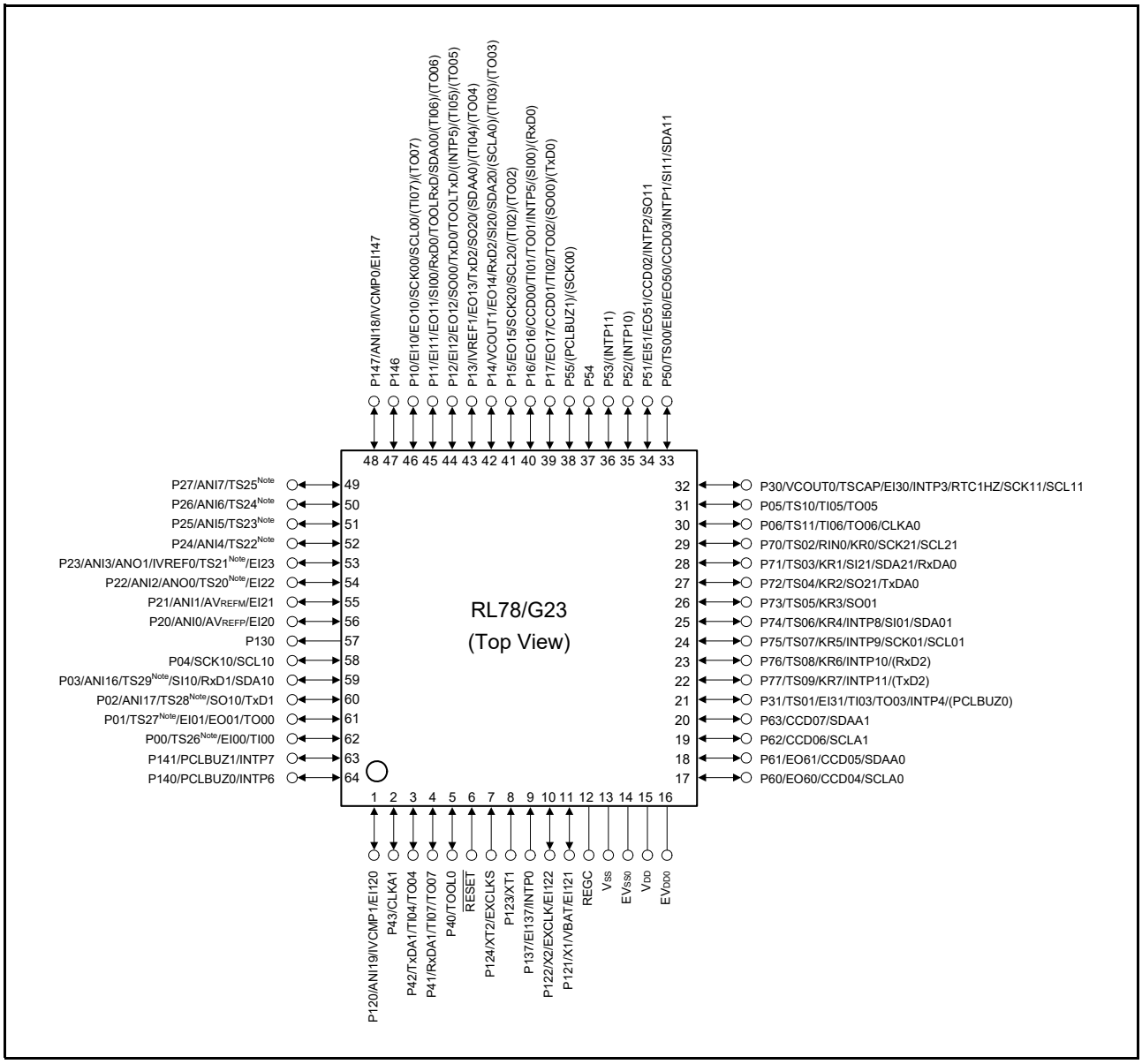
Table 1 - 9 Multiplexed Pin Functions of the 52-pin R7F100G Products (2/2)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
52LFQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
32	P15	—	EO15	PCLBUZ1	—	—	—	—	—	—	(TI02)/ (TO02)	—	SCK20/ SCL20	—	—	—
33	P14	—	EO14	—	—	—	VCOUT1	—	—	—	(TI03)/ (TO03)	—	SI20/ RxD2/ SDA20	(SCLA0)	—	—
34	P13	—	EO13	—	—	—	IVREF1	—	—	—	(TI04)/ (TO04)	—	SO20/ TxD2	(SDAA0)	—	—
35	P12	—	EI12/ EO12	TOOLTxD	—	—	—	—	—	—	(TI05)/ (TO05)	—	SO00/ TxD0	—	—	—
36	P11	—	EI11/ EO11	TOOLRxD	—	—	—	—	—	—	(TI06)/ (TO06)	—	SI00/ RxD0/ SDA00	—	—	—
37	P10	—	EI10/ EO10	—	—	—	—	—	—	—	(TI07)/ (TO07)	—	SCK00/ SCL00	—	—	—
38	P146	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
39	P147	—	EI147	—	ANI18	—	IVCMP0	—	—	—	—	—	—	—	—	—
40	P27	—	—	—	ANI7	—	—	—	—	TS25 **Note**	—	—	—	—	—	—
41	P26	—	—	—	ANI6	—	—	—	—	TS24 **Note**	—	—	—	—	—	—
42	P25	—	—	—	ANI5	—	—	—	—	TS23 **Note**	—	—	—	—	—	—
43	P24	—	—	—	ANI4	—	—	—	—	TS22 **Note**	—	—	—	—	—	—
44	P23	—	EI23	—	ANI3	ANO1	IVREF0	—	—	TS21 **Note**	—	—	—	—	—	—
45	P22	—	EI22	—	ANI2	ANO0	—	—	—	TS20 **Note**	—	—	—	—	—	—
46	P21	—	EI21	—	ANI1/ AVREFM	—	—	—	—	—	—	—	—	—	—	—
47	P20	—	EI20	—	ANI0/ AVREFP	—	—	—	—	—	—	—	—	—	—	—
48	P130	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
49	P03	—	—	—	ANI16	—	—	—	—	TS29 **Note**	—	—	RxD1	—	—	—
50	P02	—	—	—	ANI17	—	—	—	—	TS28 **Note**	—	—	TxD1	—	—	—
51	P01	—	EI01/ EO01	—	—	—	—	—	—	TS27 **Note**	TO00	—	—	—	—	—
52	P00	—	EI00	—	—	—	—	—	—	TS26 **Note**	TI00	—	—	—	—	—

Note Not present in products with 128 or fewer Kbytes of code flash memory.

1.4.8 64-pin R7F100G products

- 64-pin plastic LQFP (12 × 12 mm, 0.65-mm pitch)
- 64-pin plastic LFQFP (10 × 10 mm, 0.50-mm pitch)



Note Not present in products with 128 or fewer Kbytes of code flash memory.

Caution 1. Connect the EVSS0 pin to the same ground as the VSS pin.
Caution 2. Make sure that the voltage on the VDD pin is no less than that on the EVDD0 pin.
Caution 3. Connect the REGC pin to VSS via a capacitor (0.47 to 1 μF).

Remark 1. For pin identification, see **1.5 Pin Identification**.
Remark 2. When using the microcontroller for an application where the noise generated inside the microcontroller must be reduced, it is recommended to supply separate powers to the VDD and EVDD0 pins and connect the VSS and EVSS0 pins to separate ground lines.
Remark 3. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

Table 1 - 10 Multiplexed Pin Functions of the 64-pin R7F100G Products (1/3)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
64LFQFP, 64LQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
1	P120	—	EI120	—	ANI19	—	IVCMP1	—	—	—	—	—	—	—	—	—
2	P43	—	—	—	—	—	—	—	—	—	—	—	—	—	CLKA1	—
3	P42	—	—	—	—	—	—	—	—	—	TI04/ TO04	—	—	—	TxDA1	—
4	P41	—	—	—	—	—	—	—	—	—	TI07/ TO07	—	—	—	RxDA1	—
5	P40	—	—	TOOL0	—	—	—	—	—	—	—	—	—	—	—	—
6	—	—	—	$\overline{\text{RESET}}$	—	—	—	—	—	—	—	—	—	—	—	—
7	P124	—	—	XT2/ EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—
8	P123	—	—	XT1	—	—	—	—	—	—	—	—	—	—	—	—
9	P137	—	EI137	—	—	—	—	INTP0	—	—	—	—	—	—	—	—
10	P122	—	EI122	X2/EXCLK	—	—	—	—	—	—	—	—	—	—	—	—
11	P121	—	EI121	X1/VBAT	—	—	—	—	—	—	—	—	—	—	—	—
12	—	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—
13	—	—	—	V_{SS}	—	—	—	—	—	—	—	—	—	—	—	—
14	—	—	—	EV_{SS0}	—	—	—	—	—	—	—	—	—	—	—	—
15	—	—	—	V_{DD}	—	—	—	—	—	—	—	—	—	—	—	—
16	—	—	—	EV_{DD0}	—	—	—	—	—	—	—	—	—	—	—	—
17	P60	CCD04	EO60	—	—	—	—	—	—	—	—	—	—	SCLA0	—	—
18	P61	CCD05	EO61	—	—	—	—	—	—	—	—	—	—	SDAA0	—	—
19	P62	CCD06	—	—	—	—	—	—	—	—	—	—	—	SCLA1	—	—
20	P63	CCD07	—	—	—	—	—	—	—	—	—	—	—	SDAA1	—	—
21	P31	—	EI31	(PCLBUZ0)	—	—	—	INTP4	—	TS01	TI03/ TO03	—	—	—	—	—
22	P77	—	—	—	—	—	—	INTP11	KR7	TS09	—	—	(TxD2)	—	—	—
23	P76	—	—	—	—	—	—	INTP10	KR6	TS08	—	—	(RxD2)	—	—	—
24	P75	—	—	—	—	—	—	INTP9	KR5	TS07	—	—	SCK01/ SCL01	—	—	—
25	P74	—	—	—	—	—	—	INTP8	KR4	TS06	—	—	SI01/ SDA01	—	—	—
26	P73	—	—	—	—	—	—	—	KR3	TS05	—	—	SO01	—	—	—
27	P72	—	—	—	—	—	—	—	KR2	TS04	—	—	SO21	—	TxDA0	—
28	P71	—	—	—	—	—	—	—	KR1	TS03	—	—	SI21/ SDA21	—	RxDA0	—
29	P70	—	—	—	—	—	—	—	KR0	TS02	—	—	SCK21/ SCL21	—	—	RIN0
30	P06	—	—	—	—	—	—	—	—	TS11	TI06/ TO06	—	—	—	CLKA0	—
31	P05	—	—	—	—	—	—	—	—	TS10	TI05/ TO05	—	—	—	—	—

Table 1 - 10 Multiplexed Pin Functions of the 64-pin R7F100G Products (2/3)

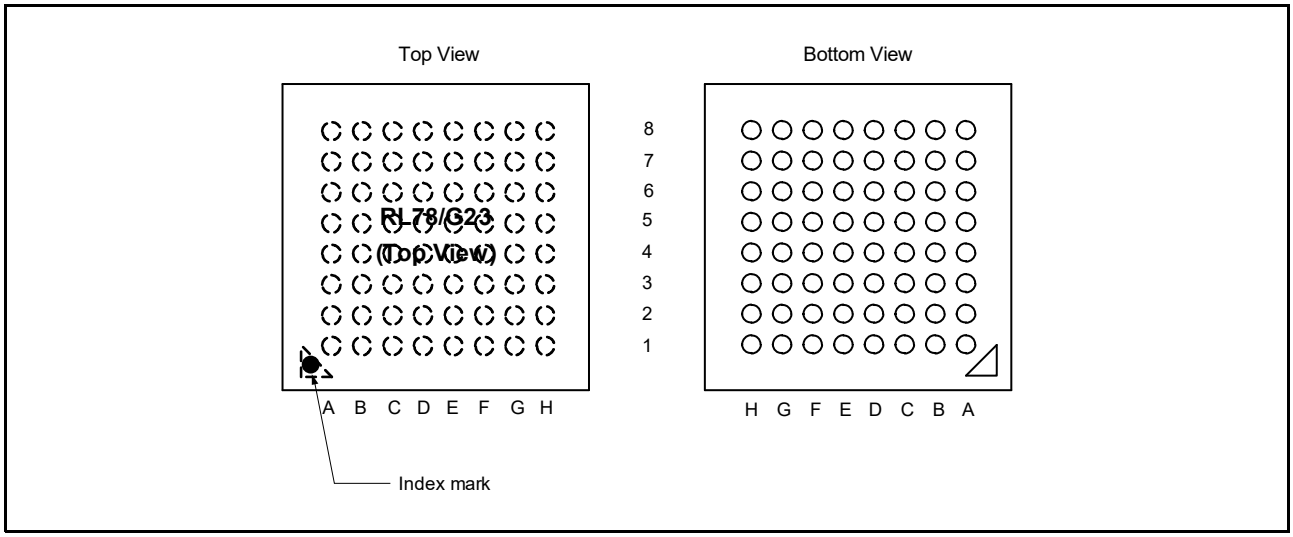
Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
64LFQFP, 64LQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
32	P30	—	EI30	—	—	—	VCOUT0	INTP3	—	TSCAP	—	RTC1HZ	SCK11/ SCL11	—	—	—
33	P50	CCD03	EI50/ EO50	—	—	—	—	INTP1	—	TS00	—	—	SI11/ SDA11	—	—	—
34	P51	CCD02	EI51/ EO51	—	—	—	—	INTP2	—	—	—	—	SO11	—	—	—
35	P52	—	—	—	—	—	—	(INTP10)	—	—	—	—	—	—	—	—
36	P53	—	—	—	—	—	—	(INTP11)	—	—	—	—	—	—	—	—
37	P54	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
38	P55	—	—	(PCLBUZ1)	—	—	—	—	—	—	—	—	(SCK00)	—	—	—
39	P17	CCD01	EO17	—	—	—	—	—	—	—	TI02/ TO02	—	(SO00)/ (TxD0)	—	—	—
40	P16	CCD00	EO16	—	—	—	—	INTP5	—	—	TI01/ TO01	—	(SI00)/ (RxD0)	—	—	—
41	P15	—	EO15	—	—	—	—	—	—	—	(TI02)/ (TO02)	—	SCK20/ SCL20	—	—	—
42	P14	—	EO14	—	—	—	VCOUT1	—	—	—	(TI03)/ (TO03)	—	SI20/ RxD2/ SDA20	(SCLA0)	—	—
43	P13	—	EO13	—	—	—	IVREF1	—	—	—	(TI04)/ (TO04)	—	SO20/ TxD2	(SDAA0)	—	—
44	P12	—	EI12/ EO12	TOOLTxD	—	—	—	(INTP5)	—	—	(TI05)/ (TO05)	—	SO00/ TxD0	—	—	—
45	P11	—	EI11/ EO11	TOOLRxD	—	—	—	—	—	—	(TI06)/ (TO06)	—	SI00/ RxD0/ SDA00	—	—	—
46	P10	—	EI10/ EO10	—	—	—	—	—	—	—	(TI07)/ (TO07)	—	SCK00/ SCL00	—	—	—
47	P146	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
48	P147	—	EI147	—	ANI18	—	IVCMP0	—	—	—	—	—	—	—	—	—
49	P27	—	—	—	ANI7	—	—	—	—	TS25 **Note**	—	—	—	—	—	—
50	P26	—	—	—	ANI6	—	—	—	—	TS24 **Note**	—	—	—	—	—	—
51	P25	—	—	—	ANI5	—	—	—	—	TS23 **Note**	—	—	—	—	—	—
52	P24	—	—	—	ANI4	—	—	—	—	TS22 **Note**	—	—	—	—	—	—
53	P23	—	EI23	—	ANI3	ANO1	IVREF0	—	—	TS21 **Note**	—	—	—	—	—	—
54	P22	—	EI22	—	ANI2	ANO0	—	—	—	TS20 **Note**	—	—	—	—	—	—
55	P21	—	EI21	—	ANI1/ AVREFM	—	—	—	—	—	—	—	—	—	—	—
56	P20	—	EI20	—	ANI0/ AVREFP	—	—	—	—	—	—	—	—	—	—	—
57	P130	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 1 - 10 Multiplexed Pin Functions of the 64-pin R7F100G Products (3/3)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
64LFQFP, 64LQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
58	P04	—	—	—	—	—	—	—	—	—	—	—	SCK10/ SCL10	—	—	—
59	P03	—	—	—	ANI16	—	—	—	—	TS29 **Note**	—	—	SI10/ RxD1/ SDA10	—	—	—
60	P02	—	—	—	ANI17	—	—	—	—	TS28 **Note**	—	—	SO10/ TxD1	—	—	—
61	P01	—	EI01/ EO01	—	—	—	—	—	—	TS27 **Note**	TO00	—	—	—	—	—
62	P00	—	EI00	—	—	—	—	—	—	TS26 **Note**	TI00	—	—	—	—	—
63	P141	—	—	PCLBUZ1	—	—	—	INTP7	—	—	—	—	—	—	—	—
64	P140	—	—	PCLBUZ0	—	—	—	INTP6	—	—	—	—	—	—	—	—

Note Not present in products with 128 or fewer Kbytes of code flash memory.

- 64-pin plastic WFLGA (5 × 5 mm, 0.50-mm pitch)



	A	B	C	D	E	F	G	H
8	EVDD0	EVSS0	P121/X1/EI121/ VBAT	P122/X2/EXCLK/ EI122	P137/INTP0/ EI137	P123/XT1	P124/XT2/ EXCLKS	P120/ANI19/ IVCMP1/EI120
7	P60/CCD04/ SCLA0/EO60	VDD	VSS	REGC	$\overline{\text{RESET}}$	P01/TS27Note/ EI01/EO01/ TO00	P00/TS26Note/ EI00/TI00	P140/PCLBUZ0/ INTP6
6	P61/CCD05/ SDAA0/EO61	P62/CCD06/ SCLA1	P63/CCD07/ SDAA1	P40/TOOL0	P41/TI07/TO07/ RxDA1	P43/CLKA1	P02/ANI17/ TS28Note/ SO10/TxD1	P141/PCLBUZ1/ INTP7
5	P77/KR7/TS09/ INTP11/(TxD2)	P31/TI03/TO03/ INTP4/TS01/ EI31/(PCLBUZ0)	P53/(INTP11)	P42/TI04/TO04/ TxDA1	P03/ANI16/ TS29Note/ SI10/RxD1/ SDA10	P04/SCK10/ SCL10	P130	P20/ANI0/ AVREFP/EI20
4	P75/KR5/TS07/ INTP9/SCK01/ SCL01	P76/KR6/TS08/ INTP10/(RxD2)	P52/(INTP10)	P54	P16/CCD00/ TI01/TO01/ INTP5/EO16/ (SI00)/(RxD0)	P21/ANI1/ AVREFM/EI21	P22/ANI2/ANO0/ EI22/TS20Note	P23/ANI3/ANO1/ IVREF0/EI23/ TS21Note
3	P70/KR0/TS02/ RIN0/SCK21/ SCL21	P73/KR3/TS05/ SO01	P74/KR4/TS06/ INTP8/SI01/ SDA01	P17/CCD01/ TI02/TO02/ EO17/(SO00)/ (TxD0)	P15/SCK20/ SCL20/EO15/ (TI02)/(TO02)	P12/SO00/TxD0/ TOOLTxD/EI12/ EO12/(INTP5)/ (TI05)/(TO05)	P24/ANI4/ TS22Note	P26/ANI6/ TS24Note
2	P30/INTP3/ TSCAP/ RTC1HZ/EI30/ VCOUT0/ SCK11/SCL11	P72/KR2/TS04/ SO21/TxDA0	P71/KR1/TS03/ SI21/SDA21/ RxDA0	P06/TS11/TI06/ TO06/CLKA0	P14/RxD2/ SI20/SDA20/ VCOUT1/EO14/ (SCLA0)/(TI03)/ (TO03)	P11/SI00/RxD0/ TOOLRxD/ SDA00/EI11/ EO11/(TI06)/ (TO06)	P25/ANI5/ TS23Note	P27/ANI7/ TS25Note
1	P05/TS10/TI05/ TO05	P50/CCD03/ TS00/EI50/ EO50/INTP1/ SI11/SDA11	P51/CCD02/ EI51/EO51/ INTP2/SO11	P55/(PCLBUZ1)/ (SCK00)	P13/TxD2/SO20/ IVREF1/EO13/ (SDAA0)/(TI04)/ (TO04)	P10/SCK00/ SCL00/EI10/ EO10/(TI07)/ (TO07)	P146	P147/ANI18/ EI147/IVCMP0

Note Not present in products with 128 or fewer Kbytes of code flash memory.

(Cautions and Remarks are listed on the next page.)

Caution 1. Connect the EV_{SS0} pin to the same ground as the V_{SS} pin.
Caution 2. Make sure that the voltage on the V_{DD} pin is no less than that on the EV_{DD0} pin.
Caution 3. Connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF).

Remark 1. For pin identification, see **1.5 Pin Identification**.
Remark 2. When using the microcontroller for an application where the noise generated inside the microcontroller must be reduced, it is recommended to supply separate powers to the V_{DD} and EV_{DD0} pins and connect the V_{SS} and EV_{SS0} pins to separate ground lines.
Remark 3. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

Table 1 - 11 Multiplexed Pin Functions 2 of the 64-pin R7F100G Products (1/3)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
64WFLGA	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
A1	P05	—	—	—	—	—	—	—	—	TS10	TI05/ TO05	—	—	—	—	—
A2	P30	—	EI30	—	—	—	VCOUT0	INTP3	—	TSCAP	—	RTC1HZ	SCK11/ SCL11	—	—	—
A3	P70	—	—	—	—	—	—	—	KR0	TS02	—	—	SCK21/ SCL21	—	—	RIN0
A4	P75	—	—	—	—	—	—	INTP9	KR5	TS07	—	—	SCK01/ SCL01	—	—	—
A5	P77	—	—	—	—	—	—	INTP11	KR7	TS09	—	—	(TxD2)	—	—	—
A6	P61	CCD05	EO61	—	—	—	—	—	—	—	—	—	—	SDAA0	—	—
A7	P60	CCD04	EO60	—	—	—	—	—	—	—	—	—	—	SCLA0	—	—
A8	—	—	—	EV_{DD0}	—	—	—	—	—	—	—	—	—	—	—	—
B1	P50	CCD03	EI50/ EO50	—	—	—	—	INTP1	—	TS00	—	—	SI11/ SDA11	—	—	—
B2	P72	—	—	—	—	—	—	—	KR2	TS04	—	—	SO21	—	TxDA0	—
B3	P73	—	—	—	—	—	—	—	KR3	TS05	—	—	SO01	—	—	—
B4	P76	—	—	—	—	—	—	INTP10	KR6	TS08	—	—	(RxD2)	—	—	—
B5	P31	—	EI31	(PCLBUZ0)	—	—	—	INTP4	—	TS01	TI03/ TO03	—	—	—	—	—
B6	P62	CCD06	—	—	—	—	—	—	—	—	—	—	—	SCLA1	—	—
B7	—	—	—	V_{DD}	—	—	—	—	—	—	—	—	—	—	—	—
B8	—	—	—	EV_{SS0}	—	—	—	—	—	—	—	—	—	—	—	—
C1	P51	CCD02	EI51/ EO51	—	—	—	—	INTP2	—	—	—	—	SO11	—	—	—
C2	P71	—	—	—	—	—	—	—	KR1	TS03	—	—	SI21/ SDA21	—	RxDA0	—
C3	P74	—	—	—	—	—	—	INTP8	KR4	TS06	—	—	SI01/ SDA01	—	—	—
C4	P52	—	—	—	—	—	—	(INTP10)	—	—	—	—	—	—	—	—
C5	P53	—	—	—	—	—	—	(INTP11)	—	—	—	—	—	—	—	—
C6	P63	CCD07	—	—	—	—	—	—	—	—	—	—	—	SDAA1	—	—
C7	—	—	—	V_{SS}	—	—	—	—	—	—	—	—	—	—	—	—
C8	P121	—	EI121	X1/VBAT	—	—	—	—	—	—	—	—	—	—	—	—
D1	P55	—	—	(PCLBUZ1)	—	—	—	—	—	—	—	—	(SCK00)	—	—	—
D2	P06	—	—	—	—	—	—	—	—	TS11	TI06/ TO06	—	—	—	CLKA0	—
D3	P17	CCD01	EO17	—	—	—	—	—	—	—	TI02/ TO02	—	(SO00)/ (TxD0)	—	—	—
D4	P54	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
D5	P42	—	—	—	—	—	—	—	—	—	TI04/ TO04	—	—	—	TxDA1	—
D6	P40	—	—	TOOL0	—	—	—	—	—	—	—	—	—	—	—	—
D7	—	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—

Table 1 - 11 Multiplexed Pin Functions 2 of the 64-pin R7F100G Products (2/3)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
64WFLGA	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
D8	P122	—	EI122	X2/EXCLK	—	—	—	—	—	—	—	—	—	—	—	—
E1	P13	—	EO13	—	—	—	IVREF1	—	—	—	(TI04)/ (TO04)	—	SO20/ TxD2	(SDAA0)	—	—
E2	P14	—	EO14	—	—	—	VCOUT1	—	—	—	(TI03)/ (TO03)	—	SI20/ RxD2/ SDA20	(SCLA0)	—	—
E3	P15	—	EO15	—	—	—	—	—	—	—	(TI02)/ (TO02)	—	SCK20/ SCL20	—	—	—
E4	P16	CCD00	EO16	—	—	—	—	INTP5	—	—	TI01/ TO01	—	(SI00)/ (RxD0)	—	—	—
E5	P03	—	—	—	ANI16	—	—	—	—	TS29 **Note**	—	—	SI10/ RxD1/ SDA10	—	—	—
E6	P41	—	—	—	—	—	—	—	—	—	TI07/ TO07	—	—	—	RxDA1	—
E7	—	—	—	$\overline{\text{RESET}}$	—	—	—	—	—	—	—	—	—	—	—	—
E8	P137	—	EI137	—	—	—	—	INTP0	—	—	—	—	—	—	—	—
F1	P10	—	EI10/ EO10	—	—	—	—	—	—	—	(TI07)/ (TO07)	—	SCK00/ SCL00	—	—	—
F2	P11	—	EI11/ EO11	TOOLRxD	—	—	—	—	—	—	(TI06)/ (TO06)	—	SI00/ RxD0/ SDA00	—	—	—
F3	P12	—	EI12/ EO12	TOOLTxD	—	—	—	(INTP5)	—	—	(TI05)/ (TO05)	—	SO00/ TxD0	—	—	—
F4	P21	—	EI21	—	ANI1/ AV_{REFM}	—	—	—	—	—	—	—	—	—	—	—
F5	P04	—	—	—	—	—	—	—	—	—	—	—	SCK10/ SCL10	—	—	—
F6	P43	—	—	—	—	—	—	—	—	—	—	—	—	—	CLKA1	—
F7	P01	—	EI01/ EO001	—	—	—	—	—	—	TS27 **Note**	TO00	—	—	—	—	—
F8	P123	—	—	XT1	—	—	—	—	—	—	—	—	—	—	—	—
G1	P146	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
G2	P25	—	—	—	ANI5	—	—	—	—	TS23 **Note**	—	—	—	—	—	—
G3	P24	—	—	—	ANI4	—	—	—	—	TS22 **Note**	—	—	—	—	—	—
G4	P22	—	EI22	—	ANI2	ANO0	—	—	—	TS20 **Note**	—	—	—	—	—	—
G5	P130	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
G6	P02	—	—	—	ANI17	—	—	—	—	TS28 **Note**	—	—	SO10/ TxD1	—	—	—
G7	P00	—	EI00	—	—	—	—	—	—	TS26 **Note**	TI00	—	—	—	—	—
G8	P124	—	—	XT2/ EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—
H1	P147	—	EI147	—	ANI18	—	IVCMP0	—	—	—	—	—	—	—	—	—

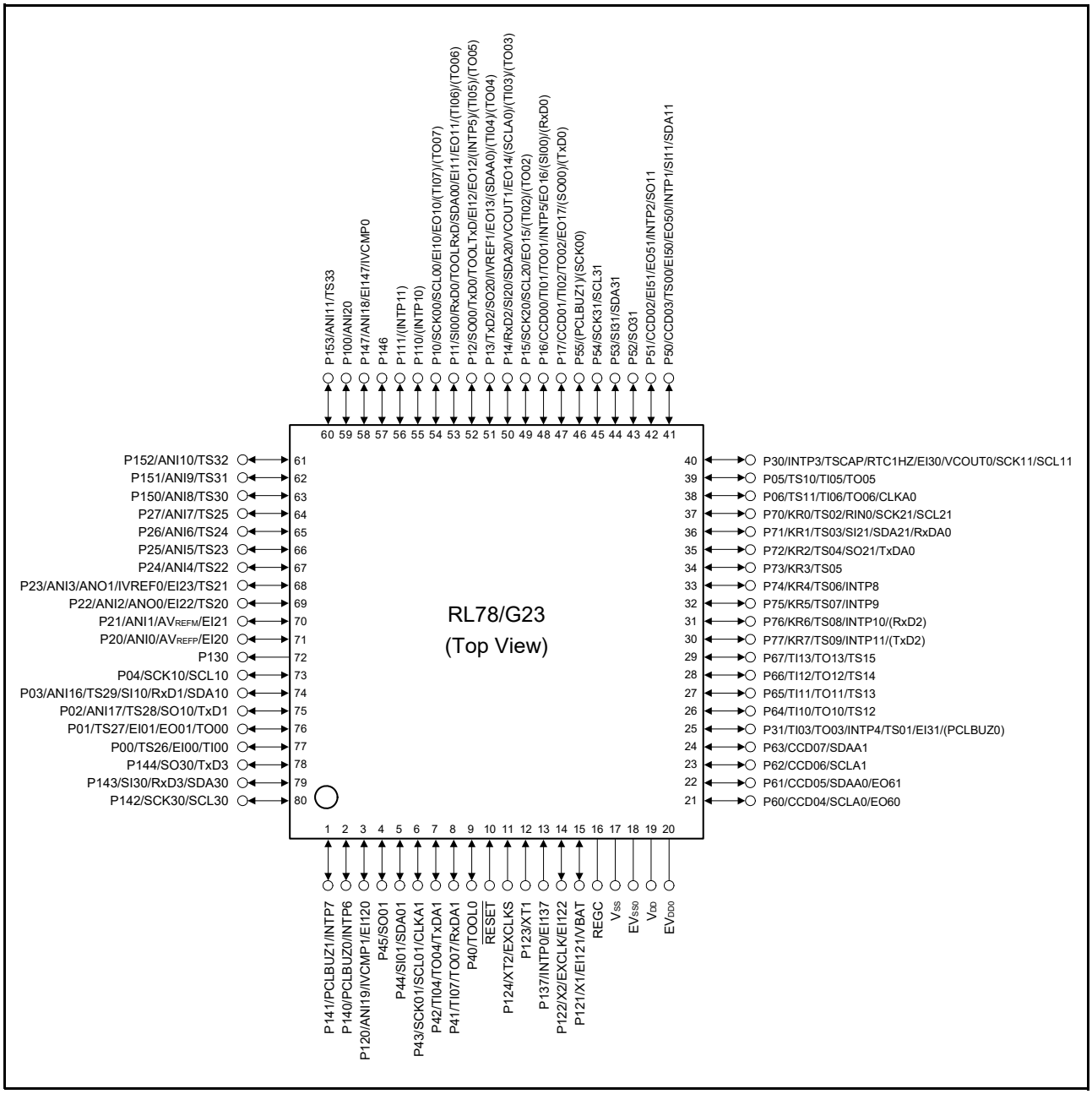
Table 1 - 11 Multiplexed Pin Functions 2 of the 64-pin R7F100G Products (3/3)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
64WFLGA	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
H2	P27	—	—	—	ANI7	—	—	—	—	TS25 **Note**	—	—	—	—	—	—
H3	P26	—	—	—	ANI6	—	—	—	—	TS24 **Note**	—	—	—	—	—	—
H4	P23	—	EI23	—	ANI3	ANO1	IVREF0	—	—	TS21 **Note**	—	—	—	—	—	—
H5	P20	—	EI20	—	ANI0/ AVREFP	—	—	—	—	—	—	—	—	—	—	—
H6	P141	—	—	PCLBUZ1	—	—	—	INTP7	—	—	—	—	—	—	—	—
H7	P140	—	—	PCLBUZ0	—	—	—	INTP6	—	—	—	—	—	—	—	—
H8	P120	—	EI120	—	ANI19	—	IVCMP1	—	—	—	—	—	—	—	—	—

Note Not present in products with 128 or fewer Kbytes of code flash memory.

1.4.9 80-pin R7F100G products

- 80-pin plastic LQFP (14 × 14 mm, 0.65-mm pitch)
- 80-pin plastic LFQFP (12 × 12 mm, 0.50-mm pitch)



Caution 1. Connect the EVSS0 pin to the same ground as the VSS pin.

Caution 2. Make sure that the voltage on the VDD pin is no less than that on the EVDD0 pin.

Caution 3. Connect the REGC pin to VSS via a capacitor (0.47 to 1 μF).

Remark 1. For pin identification, see **1.5 Pin Identification**.

Remark 2. When using the microcontroller for an application where the noise generated inside the microcontroller must be reduced, it is recommended to supply separate powers to the VDD and EVDD0 pins and connect the VSS and EVSS0 pins to separate ground lines.

Remark 3. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

Table 1 - 12 Multiplexed Pin Functions of the 80-pin R7F100G Products (1/3)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
80LFQFP, 80LQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
1	P141	—	—	PCLBUZ1	—	—	—	INTP7	—	—	—	—	—	—	—	—
2	P140	—	—	PCLBUZ0	—	—	—	INTP6	—	—	—	—	—	—	—	—
3	P120	—	EI120	—	ANI19	—	IVCMP1	—	—	—	—	—	—	—	—	—
4	P45	—	—	—	—	—	—	—	—	—	—	—	SO01	—	—	—
5	P44	—	—	—	—	—	—	—	—	—	—	—	SI01/ SDA01	—	—	—
6	P43	—	—	—	—	—	—	—	—	—	—	—	SCK01/ SCL01	—	CLKA1	—
7	P42	—	—	—	—	—	—	—	—	—	TI04/ TO04	—	—	—	TxDA1	—
8	P41	—	—	—	—	—	—	—	—	—	TI07/ TO07	—	—	—	RxDA1	—
9	P40	—	—	TOOL0	—	—	—	—	—	—	—	—	—	—	—	—
10	—	—	—	$\overline{\text{RESET}}$	—	—	—	—	—	—	—	—	—	—	—	—
11	P124	—	—	XT2/ EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—
12	P123	—	—	XT1	—	—	—	—	—	—	—	—	—	—	—	—
13	P137	—	EI137	—	—	—	—	INTP0	—	—	—	—	—	—	—	—
14	P122	—	EI122	X2/EXCLK	—	—	—	—	—	—	—	—	—	—	—	—
15	P121	—	EI121	X1/VBAT	—	—	—	—	—	—	—	—	—	—	—	—
16	—	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—
17	—	—	—	V_{SS}	—	—	—	—	—	—	—	—	—	—	—	—
18	—	—	—	EV_{SS0}	—	—	—	—	—	—	—	—	—	—	—	—
19	—	—	—	V_{DD}	—	—	—	—	—	—	—	—	—	—	—	—
20	—	—	—	EV_{DD0}	—	—	—	—	—	—	—	—	—	—	—	—
21	P60	CCD04	EO60	—	—	—	—	—	—	—	—	—	—	SCLA0	—	—
22	P61	CCD05	EO61	—	—	—	—	—	—	—	—	—	—	SDAA0	—	—
23	P62	CCD06	—	—	—	—	—	—	—	—	—	—	—	SCLA1	—	—
24	P63	CCD07	—	—	—	—	—	—	—	—	—	—	—	SDAA1	—	—
25	P31	—	EI31	(PCLBUZ0)	—	—	—	INTP4	—	TS01	TI03/ TO03	—	—	—	—	—
26	P64	—	—	—	—	—	—	—	—	TS12	TI10/ TO10	—	—	—	—	—
27	P65	—	—	—	—	—	—	—	—	TS13	TI11/ TO11	—	—	—	—	—
28	P66	—	—	—	—	—	—	—	—	TS14	TI12/ TO12	—	—	—	—	—
29	P67	—	—	—	—	—	—	—	—	TS15	TI13/ TO13	—	—	—	—	—
30	P77	—	—	—	—	—	—	INTP11	KR7	TS09	—	—	(TxD2)	—	—	—
31	P76	—	—	—	—	—	—	INTP10	KR6	TS08	—	—	(RxD2)	—	—	—
32	P75	—	—	—	—	—	—	INTP9	KR5	TS07	—	—	—	—	—	—

Table 1 - 12 Multiplexed Pin Functions of the 80-pin R7F100G Products (2/3)

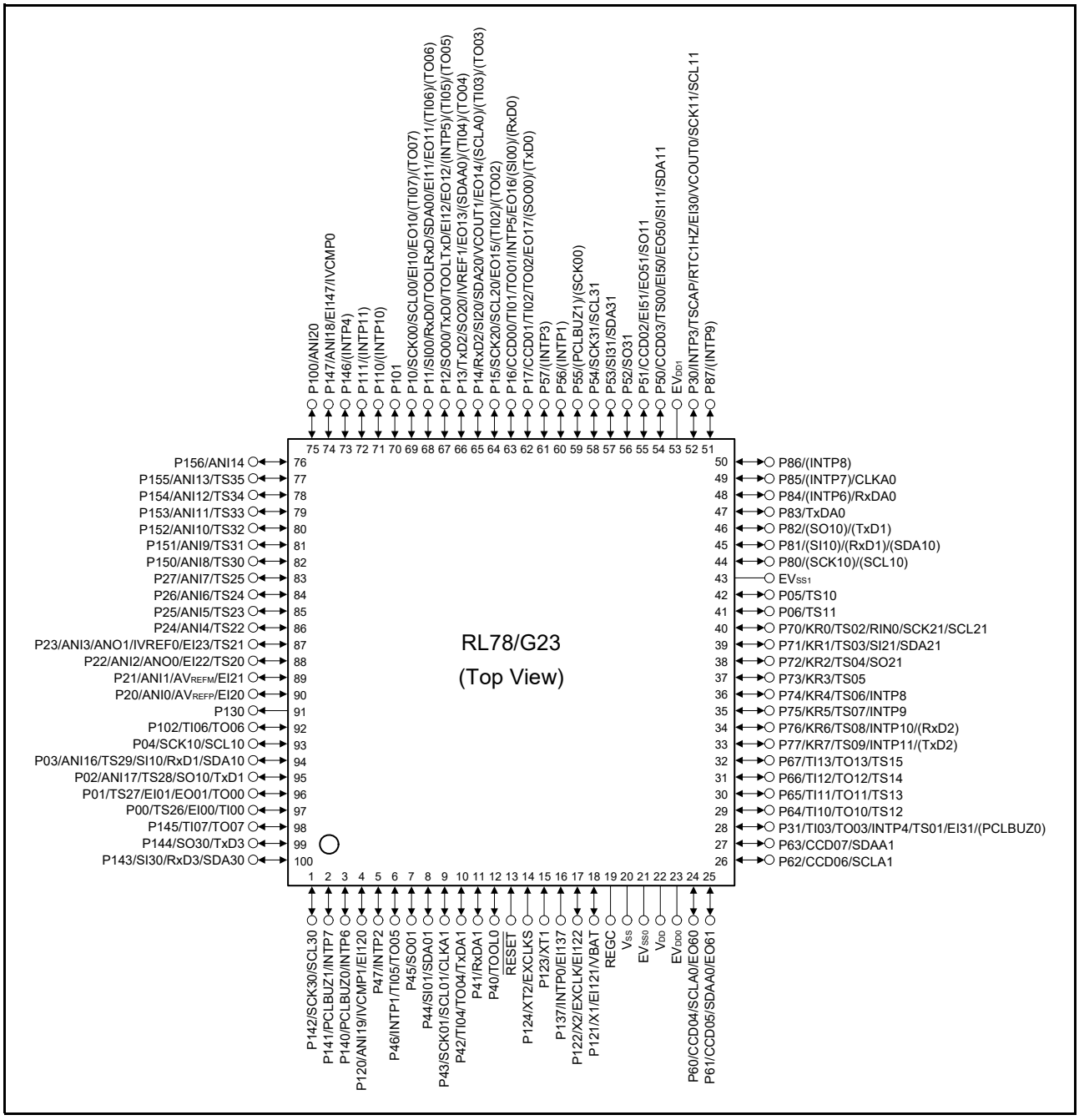
Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
80LFQFP, 80LQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
33	P74	—	—	—	—	—	—	INTP8	KR4	TS06	—	—	—	—	—	—
34	P73	—	—	—	—	—	—	—	KR3	TS05	—	—	—	—	—	—
35	P72	—	—	—	—	—	—	—	KR2	TS04	—	—	SO21	—	TxDA0	—
36	P71	—	—	—	—	—	—	—	KR1	TS03	—	—	SI21/ SDA21	—	RxDA0	—
37	P70	—	—	—	—	—	—	—	KR0	TS02	—	—	SCK21/ SCL21	—	—	RIN0
38	P06	—	—	—	—	—	—	—	—	TS11	TI06/ TO06	—	—	—	CLKA0	—
39	P05	—	—	—	—	—	—	—	—	TS10	TI05/ TO05	—	—	—	—	—
40	P30	—	EI30	—	—	—	VCOUT0	INTP3	—	TSCAP	—	RTC1HZ	SCK11/ SCL11	—	—	—
41	P50	CCD03	EI50/ EO50	—	—	—	—	INTP1	—	TS00	—	—	SI11/ SDA11	—	—	—
42	P51	CCD02	EI51/ EO51	—	—	—	—	INTP2	—	—	—	—	SO11	—	—	—
43	P52	—	—	—	—	—	—	—	—	—	—	—	SO31	—	—	—
44	P53	—	—	—	—	—	—	—	—	—	—	—	SI31/ SDA31	—	—	—
45	P54	—	—	—	—	—	—	—	—	—	—	—	SCK31/ SCL31	—	—	—
46	P55	—	—	(PCLBUZ1)	—	—	—	—	—	—	—	—	(SCK00)	—	—	—
47	P17	CCD01	EO17	—	—	—	—	—	—	—	TI02/ TO02	—	(SO00)/ (TxD0)	—	—	—
48	P16	CCD00	EO16	—	—	—	—	INTP5	—	—	TI01/ TO01	—	(SI00)/ (RxD0)	—	—	—
49	P15	—	EO15	—	—	—	—	—	—	—	(TI02)/ (TO02)	—	SCK20/ SCL20	—	—	—
50	P14	—	EO14	—	—	—	VCOUT1	—	—	—	(TI03)/ (TO03)	—	SI20/ RxD2/ SDA20	(SCLA0)	—	—
51	P13	—	EO13	—	—	—	IVREF1	—	—	—	(TI04)/ (TO04)	—	SO20/ TxD2	(SDAA0)	—	—
52	P12	—	EI12/ EO12	TOOLTxD	—	—	—	(INTP5)	—	—	(TI05)/ (TO05)	—	SO00/ TxD0	—	—	—
53	P11	—	EI11/ EO11	TOOLRxD	—	—	—	—	—	—	(TI06)/ (TO06)	—	SI00/ RxD0/ SDA00	—	—	—
54	P10	—	EI10/ EO10	—	—	—	—	—	—	—	(TI07)/ (TO07)	—	SCK00/ SCL00	—	—	—
55	P110	—	—	—	—	—	—	(INTP10)	—	—	—	—	—	—	—	—
56	P111	—	—	—	—	—	—	(INTP11)	—	—	—	—	—	—	—	—
57	P146	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
58	P147	—	EI147	—	ANI18	—	IVCMP0	—	—	—	—	—	—	—	—	—
59	P100	—	—	—	ANI20	—	—	—	—	—	—	—	—	—	—	—
60	P153	—	—	—	ANI11	—	—	—	—	TS33	—	—	—	—	—	—

Table 1 - 12 Multiplexed Pin Functions of the 80-pin R7F100G Products (3/3)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
80LFQFP, 80LQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
61	P152	—	—	—	ANI10	—	—	—	—	TS32	—	—	—	—	—	—
62	P151	—	—	—	ANI9	—	—	—	—	TS31	—	—	—	—	—	—
63	P150	—	—	—	ANI8	—	—	—	—	TS30	—	—	—	—	—	—
64	P27	—	—	—	ANI7	—	—	—	—	TS25	—	—	—	—	—	—
65	P26	—	—	—	ANI6	—	—	—	—	TS24	—	—	—	—	—	—
66	P25	—	—	—	ANI5	—	—	—	—	TS23	—	—	—	—	—	—
67	P24	—	—	—	ANI4	—	—	—	—	TS22	—	—	—	—	—	—
68	P23	—	EI23	—	ANI3	ANO1	IVREF0	—	—	TS21	—	—	—	—	—	—
69	P22	—	EI22	—	ANI2	ANO0	—	—	—	TS20	—	—	—	—	—	—
70	P21	—	EI21	—	ANI1/ AVREFM	—	—	—	—	—	—	—	—	—	—	—
71	P20	—	EI20	—	ANI0/ AVREFP	—	—	—	—	—	—	—	—	—	—	—
72	P130	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
73	P04	—	—	—	—	—	—	—	—	—	—	—	SCK10/ SCL10	—	—	—
74	P03	—	—	—	ANI16	—	—	—	—	TS29	—	—	SI10/ RxD1/ SDA10	—	—	—
75	P02	—	—	—	ANI17	—	—	—	—	TS28	—	—	SO10/ TxD1	—	—	—
76	P01	—	EI01/ EO01	—	—	—	—	—	—	TS27	TO00	—	—	—	—	—
77	P00	—	EI00	—	—	—	—	—	—	TS26	TI00	—	—	—	—	—
78	P144	—	—	—	—	—	—	—	—	—	—	—	SO30/ TxD3	—	—	—
79	P143	—	—	—	—	—	—	—	—	—	—	—	SI30/ RxD3/ SDA30	—	—	—
80	P142	—	—	—	—	—	—	—	—	—	—	—	SCK30/ SCL30	—	—	—

1.4.10 100-pin R7F100G products

- 100-pin plastic LFQFP (14 × 14 mm, 0.50-mm pitch)



Caution 1. Connect the EVSS0 and EVSS1 pins to the same ground as the VSS pin.

Caution 2. Make sure that the voltage on the VDD pin is no less than that on the EVDD0 and EVDD1 pins. Also make sure that the voltage on the EVDD0 is the same as that on the EVDD1 pin.

Caution 3. Connect the REGC pin to VSS via a capacitor (0.47 to 1 μF).

Remark 1. For pin identification, see **1.5 Pin Identification**.

Remark 2. When using the microcontroller for an application where the noise generated inside the microcontroller must be reduced, it is recommended to supply separate powers to the VDD, EVDD0, and EVDD1 pins and connect the VSS, EVSS0, and EVSS1 pins to separate ground lines.

Remark 3. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

Table 1 - 13 Multiplexed Pin Functions of the 100-pin R7F100G Products (1/4)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
100LFQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
1	P142	—	—	—	—	—	—	—	—	—	—	—	SCK30/ SCL30	—	—	—
2	P141	—	—	PCLBUZ1	—	—	—	INTP7	—	—	—	—	—	—	—	—
3	P140	—	—	PCLBUZ0	—	—	—	INTP6	—	—	—	—	—	—	—	—
4	P120	—	EI120	—	ANI19	—	IVCMP1	—	—	—	—	—	—	—	—	—
5	P47	—	—	—	—	—	—	INTP2	—	—	—	—	—	—	—	—
6	P46	—	—	—	—	—	—	INTP1	—	—	TI05/ TO05	—	—	—	—	—
7	P45	—	—	—	—	—	—	—	—	—	—	—	SO01	—	—	—
8	P44	—	—	—	—	—	—	—	—	—	—	—	SI01/ SDA01	—	—	—
9	P43	—	—	—	—	—	—	—	—	—	—	—	SCK01/ SCL01	—	CLKA1	—
10	P42	—	—	—	—	—	—	—	—	—	TI04/ TO04	—	—	—	TxDA1	—
11	P41	—	—	—	—	—	—	—	—	—	—	—	—	—	RxDA1	—
12	P40	—	—	TOOL0	—	—	—	—	—	—	—	—	—	—	—	—
13	—	—	—	$\overline{\text{RESET}}$	—	—	—	—	—	—	—	—	—	—	—	—
14	P124	—	—	XT2/ EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—
15	P123	—	—	XT1	—	—	—	—	—	—	—	—	—	—	—	—
16	P137	—	EI137	—	—	—	—	INTP0	—	—	—	—	—	—	—	—
17	P122	—	EI122	X2/EXCLK	—	—	—	—	—	—	—	—	—	—	—	—
18	P121	—	EI121	X1/VBAT	—	—	—	—	—	—	—	—	—	—	—	—
19	—	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—
20	—	—	—	V_{SS}	—	—	—	—	—	—	—	—	—	—	—	—
21	—	—	—	EV_{SS0}	—	—	—	—	—	—	—	—	—	—	—	—
22	—	—	—	V_{DD}	—	—	—	—	—	—	—	—	—	—	—	—
23	—	—	—	EV_{DD0}	—	—	—	—	—	—	—	—	—	—	—	—
24	P60	CCD04	EO60	—	—	—	—	—	—	—	—	—	—	SCLA0	—	—
25	P61	CCD05	EO61	—	—	—	—	—	—	—	—	—	—	SDAA0	—	—
26	P62	CCD06	—	—	—	—	—	—	—	—	—	—	—	SCLA1	—	—
27	P63	CCD07	—	—	—	—	—	—	—	—	—	—	—	SDAA1	—	—
28	P31	—	EI31	(PCLBUZ0)	—	—	—	INTP4	—	TS01	TI03/ TO03	—	—	—	—	—
29	P64	—	—	—	—	—	—	—	—	TS12	TI10/ TO10	—	—	—	—	—
30	P65	—	—	—	—	—	—	—	—	TS13	TI11/ TO11	—	—	—	—	—
31	P66	—	—	—	—	—	—	—	—	TS14	TI12/ TO12	—	—	—	—	—

Table 1 - 13 Multiplexed Pin Functions of the 100-pin R7F100G Products (2/4)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
100LFQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
32	P67	—	—	—	—	—	—	—	—	TS15	TI13/ TO13	—	—	—	—	—
33	P77	—	—	—	—	—	—	INTP11	KR7	TS09	—	—	(TxD2)	—	—	—
34	P76	—	—	—	—	—	—	INTP10	KR6	TS08	—	—	(RxD2)	—	—	—
35	P75	—	—	—	—	—	—	INTP9	KR5	TS07	—	—	—	—	—	—
36	P74	—	—	—	—	—	—	INTP8	KR4	TS06	—	—	—	—	—	—
37	P73	—	—	—	—	—	—	—	KR3	TS05	—	—	—	—	—	—
38	P72	—	—	—	—	—	—	—	KR2	TS04	—	—	SO21	—	—	—
39	P71	—	—	—	—	—	—	—	KR1	TS03	—	—	SI21/ SDA21	—	—	—
40	P70	—	—	—	—	—	—	—	KR0	TS02	—	—	SCK21/ SCL21	—	—	RIN0
41	P06	—	—	—	—	—	—	—	—	TS11	—	—	—	—	—	—
42	P05	—	—	—	—	—	—	—	—	TS10	—	—	—	—	—	—
43	—	—	—	EV_{SS1}	—	—	—	—	—	—	—	—	—	—	—	—
44	P80	—	—	—	—	—	—	—	—	—	—	—	(SCK10)/ (SCL10)	—	—	—
45	P81	—	—	—	—	—	—	—	—	—	—	—	(SI10)/ (RxD1)/ (SDA10)	—	—	—
46	P82	—	—	—	—	—	—	—	—	—	—	—	(SO10)/ (TxD1)	—	—	—
47	P83	—	—	—	—	—	—	—	—	—	—	—	—	—	TxDA0	—
48	P84	—	—	—	—	—	—	(INTP6)	—	—	—	—	—	—	RxDA0	—
49	P85	—	—	—	—	—	—	(INTP7)	—	—	—	—	—	—	CLKA0	—
50	P86	—	—	—	—	—	—	(INTP8)	—	—	—	—	—	—	—	—
51	P87	—	—	—	—	—	—	(INTP9)	—	—	—	—	—	—	—	—
52	P30	—	EI30	—	—	—	VCOUT0	INTP3	—	TSCAP	—	RTC1HZ	SCK11/ SCL11	—	—	—
53	—	—	—	EV_{DD1}	—	—	—	—	—	—	—	—	—	—	—	—
54	P50	CCD03	EI50/ EO50	—	—	—	—	—	—	TS00	—	—	SI11/ SDA11	—	—	—
55	P51	CCD02	EI51/ EO51	—	—	—	—	—	—	—	—	—	SO11	—	—	—
56	P52	—	—	—	—	—	—	—	—	—	—	—	SO31	—	—	—
57	P53	—	—	—	—	—	—	—	—	—	—	—	SI31/ SDA31	—	—	—
58	P54	—	—	—	—	—	—	—	—	—	—	—	SCK31/ SCL31	—	—	—
59	P55	—	—	(PCLBUZ1)	—	—	—	—	—	—	—	—	(SCK00)	—	—	—
60	P56	—	—	—	—	—	—	(INTP1)	—	—	—	—	—	—	—	—
61	P57	—	—	—	—	—	—	(INTP3)	—	—	—	—	—	—	—	—
62	P17	CCD01	EO17	—	—	—	—	—	—	—	TI02/ TO02	—	(SO00)/ (TxD0)	—	—	—

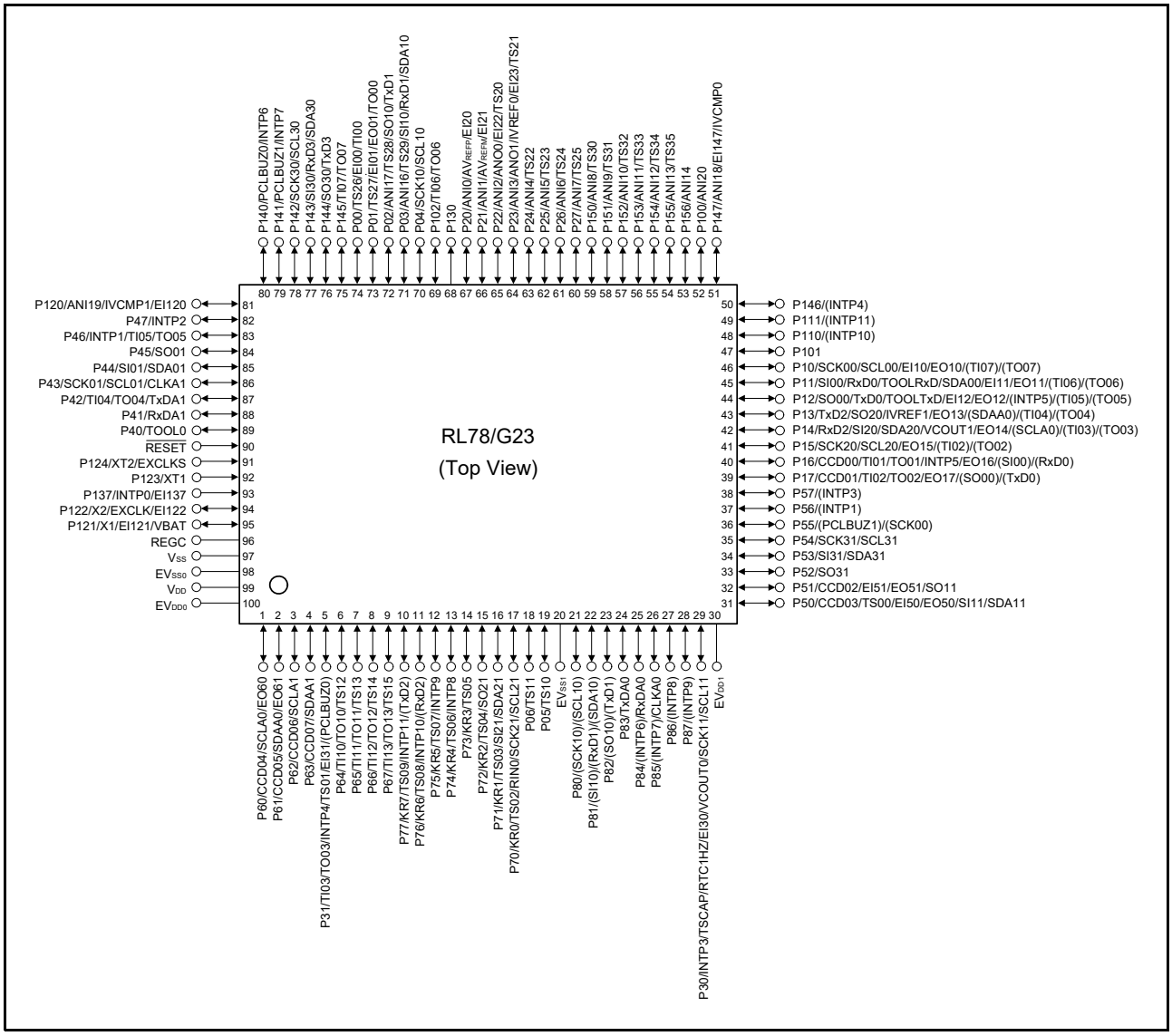
Table 1 - 13 Multiplexed Pin Functions of the 100-pin R7F100G Products (3/4)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
100LFQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
63	P16	CCD00	EO16	—	—	—	—	INTP5	—	—	TI01/ TO01	—	(SI00)/ (RxD0)	—	—	—
64	P15	—	EO15	—	—	—	—	—	—	—	(TI02)/ (TO02)	—	SCK20/ SCL20	—	—	—
65	P14	—	EO14	—	—	—	VCOUT1	—	—	—	(TI03)/ (TO03)	—	SI20/ RxD2/ SDA20	(SCLA0)	—	—
66	P13	—	EO13	—	—	—	IVREF1	—	—	—	(TI04)/ (TO04)	—	SO20/ TxD2	(SDAA0)	—	—
67	P12	—	EI12/ EO12	TOOLTxD	—	—	—	(INTP5)	—	—	(TI05)/ (TO05)	—	SO00/ TxD0	—	—	—
68	P11	—	EI11/ EO11	TOOLRxD	—	—	—	—	—	—	(TI06)/ (TO06)	—	SI00/ RxD0/ SDA00	—	—	—
69	P10	—	EI10/ EO10	—	—	—	—	—	—	—	(TI07)/ (TO07)	—	SCK00/ SCL00	—	—	—
70	P101	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
71	P110	—	—	—	—	—	—	(INTP10)	—	—	—	—	—	—	—	—
72	P111	—	—	—	—	—	—	(INTP11)	—	—	—	—	—	—	—	—
73	P146	—	—	—	—	—	—	(INTP4)	—	—	—	—	—	—	—	—
74	P147	—	EI147	—	ANI18	—	IVCMP0	—	—	—	—	—	—	—	—	—
75	P100	—	—	—	ANI20	—	—	—	—	—	—	—	—	—	—	—
76	P156	—	—	—	ANI14	—	—	—	—	—	—	—	—	—	—	—
77	P155	—	—	—	ANI13	—	—	—	—	TS35	—	—	—	—	—	—
78	P154	—	—	—	ANI12	—	—	—	—	TS34	—	—	—	—	—	—
79	P153	—	—	—	ANI11	—	—	—	—	TS33	—	—	—	—	—	—
80	P152	—	—	—	ANI10	—	—	—	—	TS32	—	—	—	—	—	—
81	P151	—	—	—	ANI9	—	—	—	—	TS31	—	—	—	—	—	—
82	P150	—	—	—	ANI8	—	—	—	—	TS30	—	—	—	—	—	—
83	P27	—	—	—	ANI7	—	—	—	—	TS25	—	—	—	—	—	—
84	P26	—	—	—	ANI6	—	—	—	—	TS24	—	—	—	—	—	—
85	P25	—	—	—	ANI5	—	—	—	—	TS23	—	—	—	—	—	—
86	P24	—	—	—	ANI4	—	—	—	—	TS22	—	—	—	—	—	—
87	P23	—	EI23	—	ANI3	ANO1	IVREF0	—	—	TS21	—	—	—	—	—	—
88	P22	—	EI22	—	ANI2	ANO0	—	—	—	TS20	—	—	—	—	—	—
89	P21	—	EI21	—	ANI1/ AVREFM	—	—	—	—	—	—	—	—	—	—	—
90	P20	—	EI20	—	ANI0/ AVREFP	—	—	—	—	—	—	—	—	—	—	—
91	P130	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
92	P102	—	—	—	—	—	—	—	—	—	TI06/ TO06	—	—	—	—	—
93	P04	—	—	—	—	—	—	—	—	—	—	—	SCK10/ SCL10	—	—	—

Table 1 - 13 Multiplexed Pin Functions of the 100-pin R7F100G Products (4/4)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
100LFQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
94	P03	—	—	—	ANI16	—	—	—	—	TS29	—	—	SI10/ RxD1/ SDA10	—	—	—
95	P02	—	—	—	ANI17	—	—	—	—	TS28	—	—	SO10/ TxD1	—	—	—
96	P01	—	EI01/ EO01	—	—	—	—	—	—	TS27	TO00	—	—	—	—	—
97	P00	—	EI00	—	—	—	—	—	—	TS26	TI00	—	—	—	—	—
98	P145	—	—	—	—	—	—	—	—	—	TI07/ TO07	—	—	—	—	—
99	P144	—	—	—	—	—	—	—	—	—	—	—	SO30/ TxD3	—	—	—
100	P143	—	—	—	—	—	—	—	—	—	—	—	SI30/ RxD3/ SDA30	—	—	—

- 100-pin plastic LQFP (14 × 20 mm, 0.65-mm pitch)



Caution 1. Connect the EVSS0 and EVSS1 pins to the same ground as the VSS pin.

Caution 2. Make sure that the voltage on the VDD pin is no less than that on the EVDD0 and EVDD1 pins. Also make sure that the voltage on the EVDD0 is the same as that on the EVDD1 pin.

Caution 3. Connect the REGC pin to VSS via a capacitor (0.47 to 1 µF).

Remark 1. For pin identification, see **1.5 Pin Identification**.

Remark 2. When using the microcontroller for an application where the noise generated inside the microcontroller must be reduced, it is recommended to supply separate powers to the VDD, EVDD0, and EVDD1 pins and connect the VSS, EVSS0, and EVSS1 pins to separate ground lines.

Remark 3. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

Table 1 - 14 Multiplexed Pin Functions 2 of the 100-pin R7F100G Products (1/4)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
100LQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
1	P60	CCD04	EO60	—	—	—	—	—	—	—	—	—	—	SCLA0	—	—
2	P61	CCD05	EO61	—	—	—	—	—	—	—	—	—	—	SDAA0	—	—
3	P62	CCD06	—	—	—	—	—	—	—	—	—	—	—	SCLA1	—	—
4	P63	CCD07	—	—	—	—	—	—	—	—	—	—	—	SDAA1	—	—
5	P31	—	EI31	(PCLBUZ0)	—	—	—	INTP4	—	TS01	TI03/ TO03	—	—	—	—	—
6	P64	—	—	—	—	—	—	—	—	TS12	TI10/ TO10	—	—	—	—	—
7	P65	—	—	—	—	—	—	—	—	TS13	TI11/ TO11	—	—	—	—	—
8	P66	—	—	—	—	—	—	—	—	TS14	TI12/ TO12	—	—	—	—	—
9	P67	—	—	—	—	—	—	—	—	TS15	TI13/ TO13	—	—	—	—	—
10	P77	—	—	—	—	—	—	INTP11	KR7	TS09	—	—	(TxD2)	—	—	—
11	P76	—	—	—	—	—	—	INTP10	KR6	TS08	—	—	(RxD2)	—	—	—
12	P75	—	—	—	—	—	—	INTP9	KR5	TS07	—	—	—	—	—	—
13	P74	—	—	—	—	—	—	INTP8	KR4	TS06	—	—	—	—	—	—
14	P73	—	—	—	—	—	—	—	KR3	TS05	—	—	—	—	—	—
15	P72	—	—	—	—	—	—	—	KR2	TS04	—	—	SO21	—	—	—
16	P71	—	—	—	—	—	—	—	KR1	TS03	—	—	SI21/ SDA21	—	—	—
17	P70	—	—	—	—	—	—	—	KR0	TS02	—	—	SCK21/ SCL21	—	—	RIN0
18	P06	—	—	—	—	—	—	—	—	TS11	—	—	—	—	—	—
19	P05	—	—	—	—	—	—	—	—	TS10	—	—	—	—	—	—
20	—	—	—	EV_{SS1}	—	—	—	—	—	—	—	—	—	—	—	—
21	P80	—	—	—	—	—	—	—	—	—	—	—	(SCK10)/ (SCL10)	—	—	—
22	P81	—	—	—	—	—	—	—	—	—	—	—	(SI10)/ (RxD1)/ (SDA10)	—	—	—
23	P82	—	—	—	—	—	—	—	—	—	—	—	(SO10)/ (TxD1)	—	—	—
24	P83	—	—	—	—	—	—	—	—	—	—	—	—	—	TxDA0	—
25	P84	—	—	—	—	—	—	(INTP6)	—	—	—	—	—	—	RxDA0	—
26	P85	—	—	—	—	—	—	(INTP7)	—	—	—	—	—	—	CLKA0	—
27	P86	—	—	—	—	—	—	(INTP8)	—	—	—	—	—	—	—	—
28	P87	—	—	—	—	—	—	(INTP9)	—	—	—	—	—	—	—	—
29	P30	—	EI30	—	—	—	VCOUT0	INTP3	—	TSCAP	—	RTC1HZ	SCK11/ SCL11	—	—	—
30	—	—	—	EV_{DD1}	—	—	—	—	—	—	—	—	—	—	—	—

Table 1 - 14 Multiplexed Pin Functions 2 of the 100-pin R7F100G Products (2/4)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
100LQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
31	P50	CCD03	EI50/ EO50	—	—	—	—	—	—	TS00	—	—	SI11/ SDA11	—	—	—
32	P51	CCD02	EI51/ EO51	—	—	—	—	—	—	—	—	—	SO11	—	—	—
33	P52	—	—	—	—	—	—	—	—	—	—	—	SO31	—	—	—
34	P53	—	—	—	—	—	—	—	—	—	—	—	SI31/ SDA31	—	—	—
35	P54	—	—	—	—	—	—	—	—	—	—	—	SCK31/ SCL31	—	—	—
36	P55	—	—	(PCLBUZ1)	—	—	—	—	—	—	—	—	(SCK00)	—	—	—
37	P56	—	—	—	—	—	—	(INTP1)	—	—	—	—	—	—	—	—
38	P57	—	—	—	—	—	—	(INTP3)	—	—	—	—	—	—	—	—
39	P17	CCD01	EO17	—	—	—	—	—	—	—	TI02/ TO02	—	(SO00)/ (TxD0)	—	—	—
40	P16	CCD00	EO16	—	—	—	—	INTP5	—	—	TI01/ TO01	—	(SI00)/ (RxD0)	—	—	—
41	P15	—	EO15	—	—	—	—	—	—	—	(TI02)/ (TO02)	—	SCK20/ SCL20	—	—	—
42	P14	—	EO14	—	—	—	VCOUT1	—	—	—	(TI03)/ (TO03)	—	SI20/ RxD2/ SDA20	(SCLA0)	—	—
43	P13	—	EO13	—	—	—	IVREF1	—	—	—	(TI04)/ (TO04)	—	SO20/ TxD2	(SDAA0)	—	—
44	P12	—	EI12/ EO12	TOOLTxD	—	—	—	(INTP5)	—	—	(TI05)/ (TO05)	—	SO00/ TxD0	—	—	—
45	P11	—	EI11/ EO11	TOOLRxD	—	—	—	—	—	—	(TI06)/ (TO06)	—	SI00/ RxD0/ SDA00	—	—	—
46	P10	—	EI10/ EO10	—	—	—	—	—	—	—	(TI07)/ (TO07)	—	SCK00/ SCL00	—	—	—
47	P101	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
48	P110	—	—	—	—	—	—	(INTP10)	—	—	—	—	—	—	—	—
49	P111	—	—	—	—	—	—	(INTP11)	—	—	—	—	—	—	—	—
50	P146	—	—	—	—	—	—	(INTP4)	—	—	—	—	—	—	—	—
51	P147	—	EI147	—	ANI18	—	IVCMP0	—	—	—	—	—	—	—	—	—
52	P100	—	—	—	ANI20	—	—	—	—	—	—	—	—	—	—	—
53	P156	—	—	—	ANI14	—	—	—	—	—	—	—	—	—	—	—
54	P155	—	—	—	ANI13	—	—	—	—	TS35	—	—	—	—	—	—
55	P154	—	—	—	ANI12	—	—	—	—	TS34	—	—	—	—	—	—
56	P153	—	—	—	ANI11	—	—	—	—	TS33	—	—	—	—	—	—
57	P152	—	—	—	ANI10	—	—	—	—	TS32	—	—	—	—	—	—
58	P151	—	—	—	ANI9	—	—	—	—	TS31	—	—	—	—	—	—
59	P150	—	—	—	ANI8	—	—	—	—	TS30	—	—	—	—	—	—
60	P27	—	—	—	ANI7	—	—	—	—	TS25	—	—	—	—	—	—

Table 1 - 14 Multiplexed Pin Functions 2 of the 100-pin R7F100G Products (3/4)

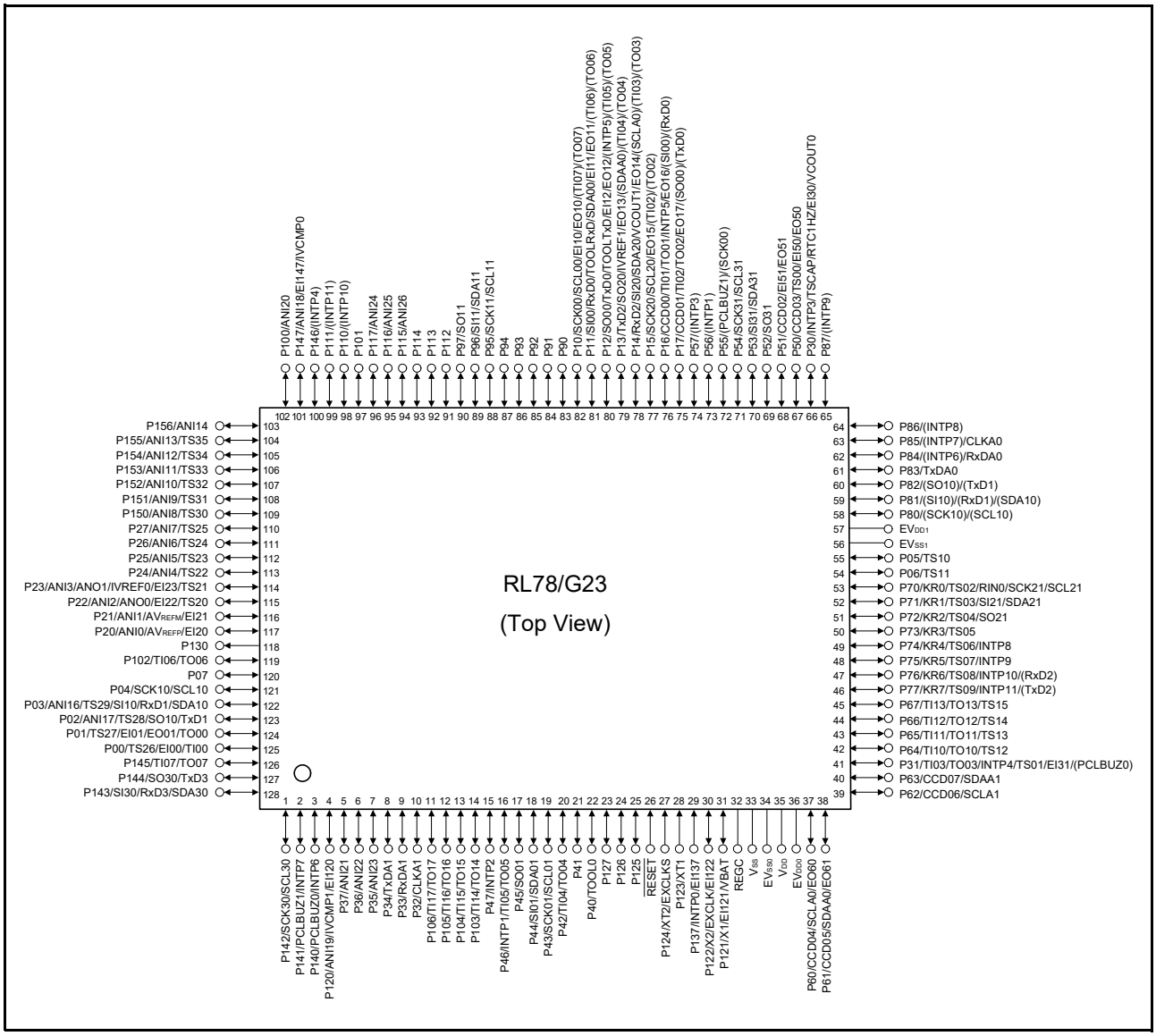
Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
100LQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
61	P26	—	—	—	ANI6	—	—	—	—	TS24	—	—	—	—	—	—
62	P25	—	—	—	ANI5	—	—	—	—	TS23	—	—	—	—	—	—
63	P24	—	—	—	ANI4	—	—	—	—	TS22	—	—	—	—	—	—
64	P23	—	EI23	—	ANI3	ANO1	IVREF0	—	—	TS21	—	—	—	—	—	—
65	P22	—	EI22	—	ANI2	ANO0	—	—	—	TS20	—	—	—	—	—	—
66	P21	—	EI21	—	ANI1/ AVREFM	—	—	—	—	—	—	—	—	—	—	—
67	P20	—	EI20	—	ANI0/ AVREFP	—	—	—	—	—	—	—	—	—	—	—
68	P130	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
69	P102	—	—	—	—	—	—	—	—	—	TI06/ TO06	—	—	—	—	—
70	P04	—	—	—	—	—	—	—	—	—	—	—	SCK10/ SCL10	—	—	—
71	P03	—	—	—	ANI16	—	—	—	—	TS29	—	—	SI10/ RxD1/ SDA10	—	—	—
72	P02	—	—	—	ANI17	—	—	—	—	TS28	—	—	SO10/ TxD1	—	—	—
73	P01	—	EI01/ EO01	—	—	—	—	—	—	TS27	TO00	—	—	—	—	—
74	P00	—	EI00	—	—	—	—	—	—	TS26	TI00	—	—	—	—	—
75	P145	—	—	—	—	—	—	—	—	—	TI07/ TO07	—	—	—	—	—
76	P144	—	—	—	—	—	—	—	—	—	—	—	SO30/ TxD3	—	—	—
77	P143	—	—	—	—	—	—	—	—	—	—	—	SI30/ RxD3/ SDA30	—	—	—
78	P142	—	—	—	—	—	—	—	—	—	—	—	SCK30/ SCL30	—	—	—
79	P141	—	—	PCLBUZ1	—	—	—	INTP7	—	—	—	—	—	—	—	—
80	P140	—	—	PCLBUZ0	—	—	—	INTP6	—	—	—	—	—	—	—	—
81	P120	—	EI120	—	ANI19	—	IVCMP1	—	—	—	—	—	—	—	—	—
82	P47	—	—	—	—	—	—	INTP2	—	—	—	—	—	—	—	—
83	P46	—	—	—	—	—	—	INTP1	—	—	TI05/ TO05	—	—	—	—	—
84	P45	—	—	—	—	—	—	—	—	—	—	—	SO01	—	—	—
85	P44	—	—	—	—	—	—	—	—	—	—	—	SI01/ SDA01	—	—	—
86	P43	—	—	—	—	—	—	—	—	—	—	—	SCK01/ SCL01	—	CLKA1	—
87	P42	—	—	—	—	—	—	—	—	—	TI04/ TO04	—	—	—	TxDA1	—
88	P41	—	—	—	—	—	—	—	—	—	—	—	—	—	RxDA1	—
89	P40	—	—	TOOL0	—	—	—	—	—	—	—	—	—	—	—	—

Table 1 - 14 Multiplexed Pin Functions 2 of the 100-pin R7F100G Products (4/4)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
100LQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
90	—	—	—	$\overline{\text{RESET}}$	—	—	—	—	—	—	—	—	—	—	—	—
91	P124	—	—	XT2/ EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—
92	P123	—	—	XT1	—	—	—	—	—	—	—	—	—	—	—	—
93	P137	—	EI137	—	—	—	—	INTP0	—	—	—	—	—	—	—	—
94	P122	—	EI122	X2/EXCLK	—	—	—	—	—	—	—	—	—	—	—	—
95	P121	—	EI121	X1/VBAT	—	—	—	—	—	—	—	—	—	—	—	—
96	—	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—
97	—	—	—	V_{SS}	—	—	—	—	—	—	—	—	—	—	—	—
98	—	—	—	EV_{SS0}	—	—	—	—	—	—	—	—	—	—	—	—
99	—	—	—	V_{DD}	—	—	—	—	—	—	—	—	—	—	—	—
100	—	—	—	EV_{DD0}	—	—	—	—	—	—	—	—	—	—	—	—

1.4.11 128-pin R7F100G products

- 128-pin plastic LFQFP (14 × 20 mm, 0.50-mm pitch)



Caution 1. Connect the EVSS0 and EVSS1 pins to the same ground as the VSS pin.

Caution 2. Make sure that the voltage on the VDD pin is no less than that on the EVDD0 and EVDD1 pins. Also make sure that the voltage on the EVDD0 is the same as that on the EVDD1 pin.

Caution 3. Connect the REGC pin to VSS via a capacitor (0.47 to 1 μF).

Remark 1. For pin identification, see **1.5 Pin Identification**.

Remark 2. When using the microcontroller for an application where the noise generated inside the microcontroller must be reduced, it is recommended to supply separate powers to the VDD, EVDD0, and EVDD1 pins and connect the VSS, EVSS0, and EVSS1 pins to separate ground lines.

Remark 3. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

Table 1 - 15 Multiplexed Pin Functions of the 128-pin R7F100G Products (1/5)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
128LFQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
1	P142	—	—	—	—	—	—	—	—	—	—	—	SCK30/ SCL30	—	—	—
2	P141	—	—	PCLBUZ1	—	—	—	INTP7	—	—	—	—	—	—	—	—
3	P140	—	—	PCLBUZ0	—	—	—	INTP6	—	—	—	—	—	—	—	—
4	P120	—	EI120	—	ANI19	—	IVCMP1	—	—	—	—	—	—	—	—	—
5	P37	—	—	—	ANI21	—	—	—	—	—	—	—	—	—	—	—
6	P36	—	—	—	ANI22	—	—	—	—	—	—	—	—	—	—	—
7	P35	—	—	—	ANI23	—	—	—	—	—	—	—	—	—	—	—
8	P34	—	—	—	—	—	—	—	—	—	—	—	—	—	TxDA1	—
9	P33	—	—	—	—	—	—	—	—	—	—	—	—	—	RxDA1	—
10	P32	—	—	—	—	—	—	—	—	—	—	—	—	—	CLKA1	—
11	P106	—	—	—	—	—	—	—	—	—	TI17/ TO17	—	—	—	—	—
12	P105	—	—	—	—	—	—	—	—	—	TI16/ TO16	—	—	—	—	—
13	P104	—	—	—	—	—	—	—	—	—	TI15/ TO15	—	—	—	—	—
14	P103	—	—	—	—	—	—	—	—	—	TI14/ TO14	—	—	—	—	—
15	P47	—	—	—	—	—	—	INTP2	—	—	—	—	—	—	—	—
16	P46	—	—	—	—	—	—	INTP1	—	—	TI05/ TO05	—	—	—	—	—
17	P45	—	—	—	—	—	—	—	—	—	—	—	SO01	—	—	—
18	P44	—	—	—	—	—	—	—	—	—	—	—	SI01/ SDA01	—	—	—
19	P43	—	—	—	—	—	—	—	—	—	—	—	SCK01/ SCL01	—	—	—
20	P42	—	—	—	—	—	—	—	—	—	TI04/ TO04	—	—	—	—	—
21	P41	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
22	P40	—	—	TOOL0	—	—	—	—	—	—	—	—	—	—	—	—
23	P127	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
24	P126	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
25	P125	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
26	—	—	—	$\overline{\text{RESET}}$	—	—	—	—	—	—	—	—	—	—	—	—
27	P124	—	—	XT2/ EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—
28	P123	—	—	XT1	—	—	—	—	—	—	—	—	—	—	—	—
29	P137	—	EI137	—	—	—	—	INTP0	—	—	—	—	—	—	—	—
30	P122	—	EI122	X2/EXCLK	—	—	—	—	—	—	—	—	—	—	—	—
31	P121	—	EI121	X1/VBAT	—	—	—	—	—	—	—	—	—	—	—	—
32	—	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—

Table 1 - 15 Multiplexed Pin Functions of the 128-pin R7F100G Products (2/5)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
128LFQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
33	—	—	—	V_{SS}	—	—	—	—	—	—	—	—	—	—	—	—
34	—	—	—	EV_{SS0}	—	—	—	—	—	—	—	—	—	—	—	—
35	—	—	—	V_{DD}	—	—	—	—	—	—	—	—	—	—	—	—
36	—	—	—	EV_{DD0}	—	—	—	—	—	—	—	—	—	—	—	—
37	P60	CCD04	EO60	—	—	—	—	—	—	—	—	—	—	SCLA0	—	—
38	P61	CCD05	EO61	—	—	—	—	—	—	—	—	—	—	SDAA0	—	—
39	P62	CCD06	—	—	—	—	—	—	—	—	—	—	—	SCLA1	—	—
40	P63	CCD07	—	—	—	—	—	—	—	—	—	—	—	SDAA1	—	—
41	P31	—	EI31	(PCLBUZ0)	—	—	—	INTP4	—	TS01	TI03/ TO03	—	—	—	—	—
42	P64	—	—	—	—	—	—	—	—	TS12	TI10/ TO10	—	—	—	—	—
43	P65	—	—	—	—	—	—	—	—	TS13	TI11/ TO11	—	—	—	—	—
44	P66	—	—	—	—	—	—	—	—	TS14	TI12/ TO12	—	—	—	—	—
45	P67	—	—	—	—	—	—	—	—	TS15	TI13/ TO13	—	—	—	—	—
46	P77	—	—	—	—	—	—	INTP11	KR7	TS09	—	—	(TxD2)	—	—	—
47	P76	—	—	—	—	—	—	INTP10	KR6	TS08	—	—	(RxD2)	—	—	—
48	P75	—	—	—	—	—	—	INTP9	KR5	TS07	—	—	—	—	—	—
49	P74	—	—	—	—	—	—	INTP8	KR4	TS06	—	—	—	—	—	—
50	P73	—	—	—	—	—	—	—	KR3	TS05	—	—	—	—	—	—
51	P72	—	—	—	—	—	—	—	KR2	TS04	—	—	SO21	—	—	—
52	P71	—	—	—	—	—	—	—	KR1	TS03	—	—	SI21/ SDA21	—	—	—
53	P70	—	—	—	—	—	—	—	KR0	TS02	—	—	SCK21/ SCL21	—	—	RIN0
54	P06	—	—	—	—	—	—	—	—	TS11	—	—	—	—	—	—
55	P05	—	—	—	—	—	—	—	—	TS10	—	—	—	—	—	—
56	—	—	—	EV_{SS1}	—	—	—	—	—	—	—	—	—	—	—	—
57	—	—	—	EV_{DD1}	—	—	—	—	—	—	—	—	—	—	—	—
58	P80	—	—	—	—	—	—	—	—	—	—	—	(SCK10)/ (SCL10)	—	—	—
59	P81	—	—	—	—	—	—	—	—	—	—	—	(SI10)/ (RxD1)/ (SDA10)	—	—	—
60	P82	—	—	—	—	—	—	—	—	—	—	—	(SO10)/ (TxD1)	—	—	—
61	P83	—	—	—	—	—	—	—	—	—	—	—	—	—	TxDA0	—
62	P84	—	—	—	—	—	—	(INTP6)	—	—	—	—	—	—	RxDA0	—
63	P85	—	—	—	—	—	—	(INTP7)	—	—	—	—	—	—	CLKA0	—
64	P86	—	—	—	—	—	—	(INTP8)	—	—	—	—	—	—	—	—

Table 1 - 15 Multiplexed Pin Functions of the 128-pin R7F100G Products (3/5)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
128LFQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
65	P87	—	—	—	—	—	—	(INTP9)	—	—	—	—	—	—	—	—
66	P30	—	EI30	—	—	—	VCOUT0	INTP3	—	TSCAP	—	RTC1HZ	—	—	—	—
67	P50	CCD03	EI50/ EO50	—	—	—	—	—	—	TS00	—	—	—	—	—	—
68	P51	CCD02	EI51/ EO51	—	—	—	—	—	—	—	—	—	—	—	—	—
69	P52	—	—	—	—	—	—	—	—	—	—	—	SO31	—	—	—
70	P53	—	—	—	—	—	—	—	—	—	—	—	SI31/ SDA31	—	—	—
71	P54	—	—	—	—	—	—	—	—	—	—	—	SCK31/ SCL31	—	—	—
72	P55	—	—	(PCLBUZ1)	—	—	—	—	—	—	—	—	(SCK00)	—	—	—
73	P56	—	—	—	—	—	—	(INTP1)	—	—	—	—	—	—	—	—
74	P57	—	—	—	—	—	—	(INTP3)	—	—	—	—	—	—	—	—
75	P17	CCD01	EO17	—	—	—	—	—	—	—	TI02/ TO02	—	(SO00)/ (TxD0)	—	—	—
76	P16	CCD00	EO16	—	—	—	—	INTP5	—	—	TI01/ TO01	—	(SI00)/ (RxD0)	—	—	—
77	P15	—	EO15	—	—	—	—	—	—	—	(TI02)/ (TO02)	—	SCK20/ SCL20	—	—	—
78	P14	—	EO14	—	—	—	VCOUT1	—	—	—	(TI03)/ (TO03)	—	SI20/ RxD2/ SDA20	(SCLA0)	—	—
79	P13	—	EO13	—	—	—	IVREF1	—	—	—	(TI04)/ (TO04)	—	SO20/ TxD2	(SDAA0)	—	—
80	P12	—	EI12/ EO12	TOOLTxD	—	—	—	(INTP5)	—	—	(TI05)/ (TO05)	—	SO00/ TxD0	—	—	—
81	P11	—	EI11/ EO11	TOOLRxD	—	—	—	—	—	—	(TI06)/ (TO06)	—	SI00/ RxD0/ SDA00	—	—	—
82	P10	—	EI10/ EO10	—	—	—	—	—	—	—	(TI07)/ (TO07)	—	SCK00/ SCL00	—	—	—
83	P90	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
84	P91	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
85	P92	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
86	P93	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
87	P94	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
88	P95	—	—	—	—	—	—	—	—	—	—	—	SCK11/ SCL11	—	—	—
89	P96	—	—	—	—	—	—	—	—	—	—	—	SI11/ SDA11	—	—	—
90	P97	—	—	—	—	—	—	—	—	—	—	—	SO11	—	—	—
91	P112	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
92	P113	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
93	P114	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 1 - 15 Multiplexed Pin Functions of the 128-pin R7F100G Products (4/5)

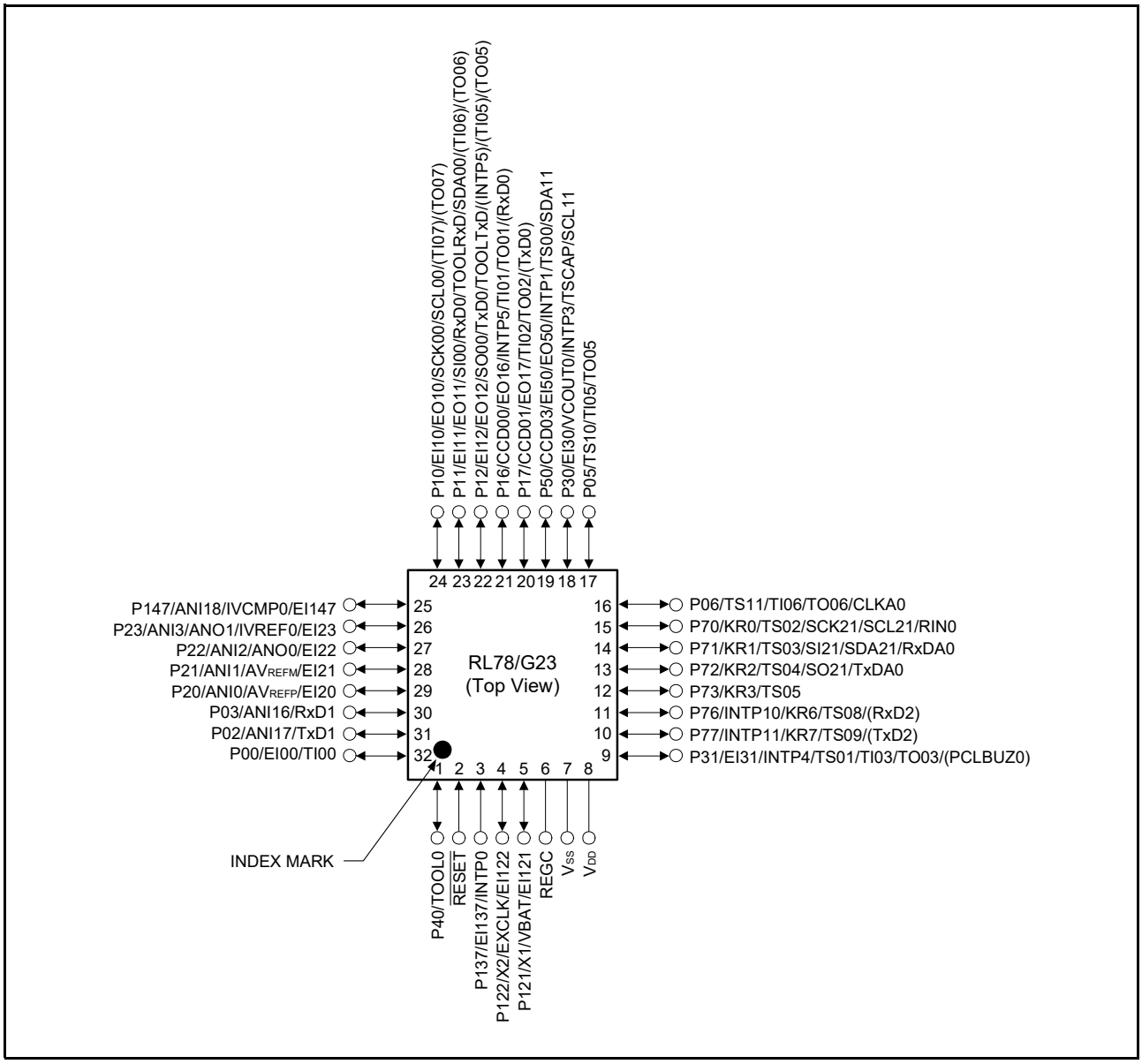
Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
128LFQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
94	P115	—	—	—	ANI26	—	—	—	—	—	—	—	—	—	—	—
95	P116	—	—	—	ANI25	—	—	—	—	—	—	—	—	—	—	—
96	P117	—	—	—	ANI24	—	—	—	—	—	—	—	—	—	—	—
97	P101	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
98	P110	—	—	—	—	—	—	(INTP10)	—	—	—	—	—	—	—	—
99	P111	—	—	—	—	—	—	(INTP11)	—	—	—	—	—	—	—	—
100	P146	—	—	—	—	—	—	(INTP4)	—	—	—	—	—	—	—	—
101	P147	—	EI147	—	ANI18	—	IVCMP0	—	—	—	—	—	—	—	—	—
102	P100	—	—	—	ANI20	—	—	—	—	—	—	—	—	—	—	—
103	P156	—	—	—	ANI14	—	—	—	—	—	—	—	—	—	—	—
104	P155	—	—	—	ANI13	—	—	—	—	TS35	—	—	—	—	—	—
105	P154	—	—	—	ANI12	—	—	—	—	TS34	—	—	—	—	—	—
106	P153	—	—	—	ANI11	—	—	—	—	TS33	—	—	—	—	—	—
107	P152	—	—	—	ANI10	—	—	—	—	TS32	—	—	—	—	—	—
108	P151	—	—	—	ANI9	—	—	—	—	TS31	—	—	—	—	—	—
109	P150	—	—	—	ANI8	—	—	—	—	TS30	—	—	—	—	—	—
110	P27	—	—	—	ANI7	—	—	—	—	TS25	—	—	—	—	—	—
111	P26	—	—	—	ANI6	—	—	—	—	TS24	—	—	—	—	—	—
112	P25	—	—	—	ANI5	—	—	—	—	TS23	—	—	—	—	—	—
113	P24	—	—	—	ANI4	—	—	—	—	TS22	—	—	—	—	—	—
114	P23	—	EI23	—	ANI3	ANO1	IVREF0	—	—	TS21	—	—	—	—	—	—
115	P22	—	EI22	—	ANI2	ANO0	—	—	—	TS20	—	—	—	—	—	—
116	P21	—	EI21	—	ANI1/ AVREFM	—	—	—	—	—	—	—	—	—	—	—
117	P20	—	EI20	—	ANI0/ AVREFP	—	—	—	—	—	—	—	—	—	—	—
118	P130	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
119	P102	—	—	—	—	—	—	—	—	—	TI06/ TO06	—	—	—	—	—
120	P07	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
121	P04	—	—	—	—	—	—	—	—	—	—	—	SCK10/ SCL10	—	—	—
122	P03	—	—	—	ANI16	—	—	—	—	TS29	—	—	SI10/ RxD1/ SDA10	—	—	—
123	P02	—	—	—	ANI17	—	—	—	—	TS28	—	—	SO10/ TxD1	—	—	—
124	P01	—	EI01/ EO01	—	—	—	—	—	—	TS27	TO00	—	—	—	—	—
125	P00	—	EI00	—	—	—	—	—	—	TS26	TI00	—	—	—	—	—
126	P145	—	—	—	—	—	—	—	—	—	TI07/ TO07	—	—	—	—	—

Table 1 - 15 Multiplexed Pin Functions of the 128-pin R7F100G Products (5/5)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
128LFQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
127	P144	—	—	—	—	—	—	—	—	—	—	—	SO30/ TxD3	—	—	—
128	P143	—	—	—	—	—	—	—	—	—	—	—	SI30/ RxD3/ SDA30	—	—	—

1.4.12 32-pin R7F104G product

- 32-pin plastic LQFP (7 × 7 mm, 0.80-mm pitch)



Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF).

Remark 1. For pin identification, see **1.5 Pin Identification**.

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

Table 1 - 16 Multiplexed Pin Functions of the 32-pin R7F104G Product (1/2)

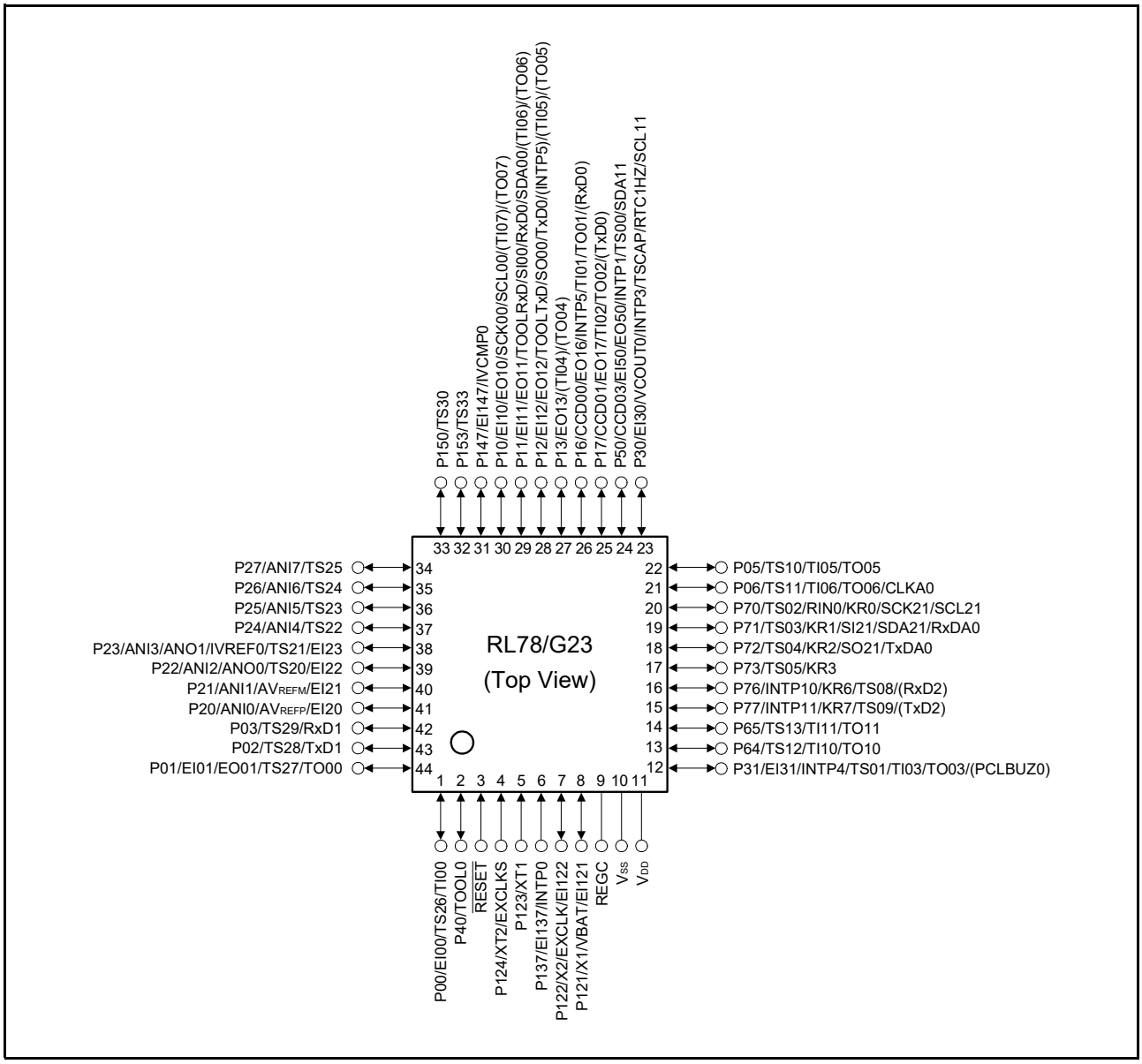
Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
32LQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
1	P40	—	—	TOOL0	—	—	—	—	—	—	—	—	—	—	—	—
2	—	—	—	$\overline{\text{RESET}}$	—	—	—	—	—	—	—	—	—	—	—	—
3	P137	—	EI137	—	—	—	—	INTP0	—	—	—	—	—	—	—	—
4	P122	—	EI122	X2/EXCLK	—	—	—	—	—	—	—	—	—	—	—	—
5	P121	—	EI121	X1/VBAT	—	—	—	—	—	—	—	—	—	—	—	—
6	—	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—
7	—	—	—	V_{SS}	—	—	—	—	—	—	—	—	—	—	—	—
8	—	—	—	V_{DD}	—	—	—	—	—	—	—	—	—	—	—	—
9	P31	—	EI31	(PCLBUZ0)	—	—	—	INTP4	—	TS01	TI03/ TO03	—	—	—	—	—
10	P77	—	—	—	—	—	—	INTP11	KR7	TS09	—	—	(TxD2)	—	—	—
11	P76	—	—	—	—	—	—	INTP10	KR6	TS08	—	—	(RxD2)	—	—	—
12	P73	—	—	—	—	—	—	—	KR3	TS05	—	—	—	—	—	—
13	P72	—	—	—	—	—	—	—	KR2	TS04	—	—	SO21	—	TxDA0	—
14	P71	—	—	—	—	—	—	—	KR1	TS03	—	—	SI21/ SDA21	—	RxDA0	—
15	P70	—	—	—	—	—	—	—	KR0	TS02	—	—	SCK21/ SCL21	—	—	RIN0
16	P06	—	—	—	—	—	—	—	—	TS11	TI06/ TO06	—	—	—	CLKA0	—
17	P05	—	—	—	—	—	—	—	—	TS10	TI05/ TO05	—	—	—	—	—
18	P30	—	EI30	—	—	—	VCOUT0	INTP3	—	TSCAP	—	—	SCL11	—	—	—
19	P50	CCD03	EI50/ EO50	—	—	—	—	INTP1	—	TS00	—	—	SDA11	—	—	—
20	P17	CCD01	EO17	—	—	—	—	—	—	—	TI02/ TO02	—	(TxD0)	—	—	—
21	P16	CCD00	EO16	—	—	—	—	INTP5	—	—	TI01/ TO01	—	(RxD0)	—	—	—
22	P12	—	EI12/ EO12	TOOLTxD	—	—	—	(INTP5)	—	—	(TI05)/ (TO05)	—	SO00/ TxD0	—	—	—
23	P11	—	EI11/ EO11	TOOLRxD	—	—	—	—	—	—	(TI06)/ (TO06)	—	SI00/ RxD0/ SDA00	—	—	—
24	P10	—	EI10/ EO10	—	—	—	—	—	—	—	(TI07)/ (TO07)	—	SCK00/ SCL00	—	—	—
25	P147	—	EI147	—	ANI18	—	IVCMP0	—	—	—	—	—	—	—	—	—
26	P23	—	EI23	—	ANI3	ANO1	IVREF0	—	—	—	—	—	—	—	—	—
27	P22	—	EI22	—	ANI2	ANO0	—	—	—	—	—	—	—	—	—	—
28	P21	—	EI21	—	ANI1/ AV_{REFM}	—	—	—	—	—	—	—	—	—	—	—
29	P20	—	EI20	—	ANI0/ AV_{REFP}	—	—	—	—	—	—	—	—	—	—	—
30	P03	—	—	—	ANI16	—	—	—	—	—	—	—	RxD1	—	—	—

Table 1 - 16 Multiplexed Pin Functions of the 32-pin R7F104G Product (2/2)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
32LQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
31	P02	—	—	—	ANI17	—	—	—	—	—	—	—	TxD1	—	—	—
32	P00	—	EI00	—	—	—	—	—	—	—	TI00	—	—	—	—	—

1.4.13 44-pin R7F104G product

- 44-pin plastic LQFP (10 × 10 mm, 0.80-mm pitch)



Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 µF).

Remark 1. For pin identification, see **1.5 Pin Identification**.

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

Table 1 - 17 Multiplexed Pin Functions of the 44-pin R7F104G Product (1/2)

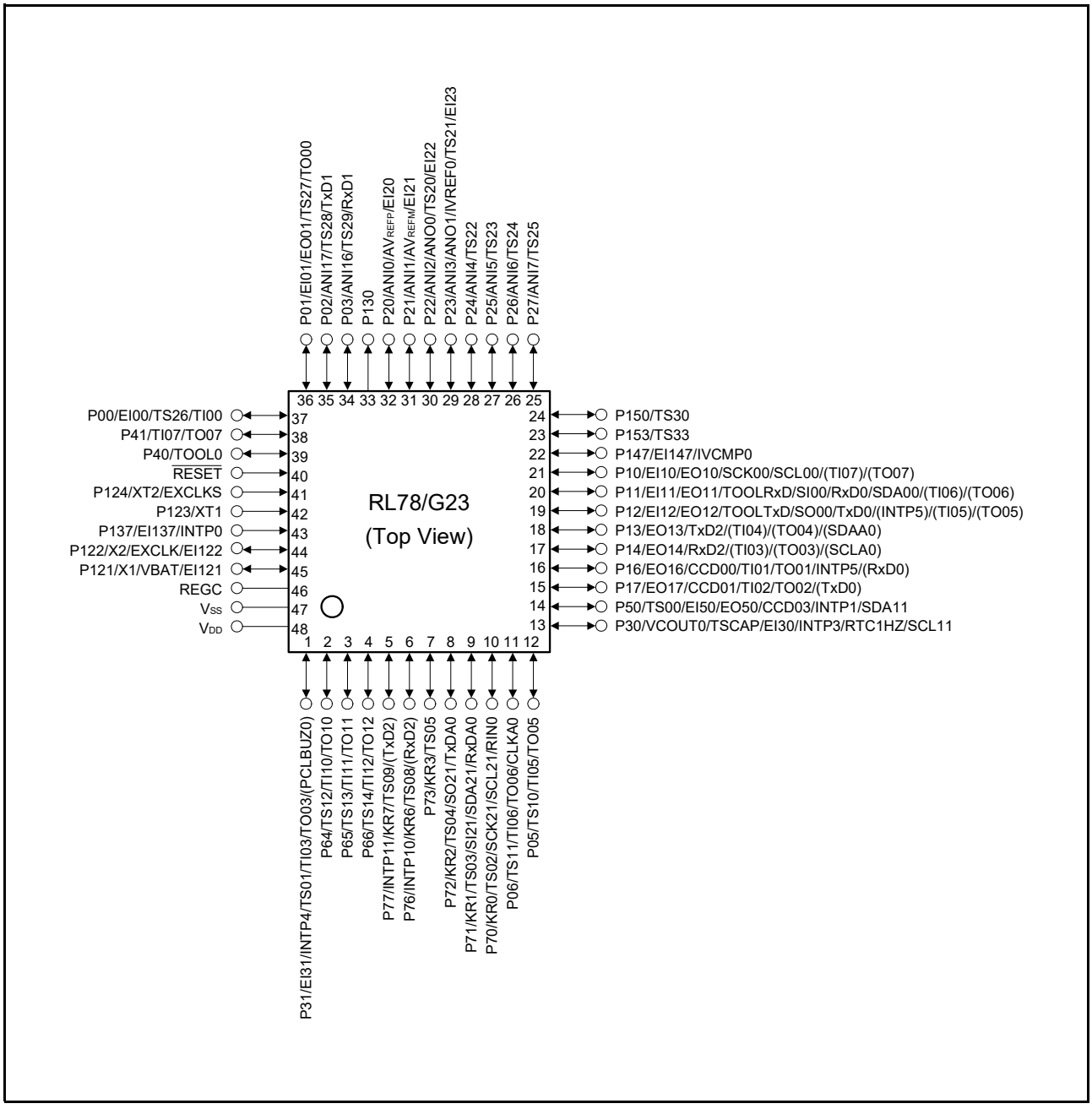
Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
44LQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
1	P00	—	EI00	—	—	—	—	—	—	TS26	TI00	—	—	—	—	—
2	P40	—	—	TOOL0	—	—	—	—	—	—	—	—	—	—	—	—
3	—	—		$\overline{\text{RESET}}$	—	—	—	—	—	—	—	—	—	—	—	—
4	P124	—	—	XT2/ EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—
5	P123	—	—	XT1	—	—	—	—	—	—	—	—	—	—	—	—
6	P137	—	EI137	—	—	—	—	INTP0	—	—	—	—	—	—	—	—
7	P122	—	EI122	X2/EXCLK	—	—	—	—	—	—	—	—	—	—	—	—
8	P121	—	EI121	X1/VBAT	—	—	—	—	—	—	—	—	—	—	—	—
9	—	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—
10	—	—	—	V_{SS}	—	—	—	—	—	—	—	—	—	—	—	—
11	—	—	—	V_{DD}	—	—	—	—	—	—	—	—	—	—	—	—
12	P31	—	EI31	(PCLBUZ0)	—	—	—	INTP4	—	TS01	TI03/ TO03	—	—	—	—	—
13	P64	—	—	—	—	—	—	—	—	TS12	TI10/ TO10	—	—	—	—	—
14	P65	—	—	—	—	—	—	—	—	TS13	TI11/ TO11	—	—	—	—	—
15	P77	—	—	—	—	—	—	INTP11	KR7	TS09	—	—	(TxD2)	—		—
16	P76	—	—	—	—	—	—	INTP10	KR6	TS08	—	—	(RxD2)	—	—	—
17	P73	—	—	—	—	—	—	—	KR3	TS05	—	—	—	—	—	—
18	P72	—	—	—	—	—	—	—	KR2	TS04	—	—	SO21	—	TxDA0	—
19	P71	—	—	—	—	—	—	—	KR1	TS03	—	—	SI21/ SDA21	—	RxDA0	—
20	P70	—	—	—	—	—	—	—	KR0	TS02	—	—	SCK21/ SCL21	—	—	RIN0
21	P06	—	—	—	—	—	—	—	—	TS11	TI06/ TO06	—	—	—	CLKA0	—
22	P05	—	—	—	—	—	—	—	—	TS10	TI05/ TO05	—	—	—	—	—
23	P30	—	EI30	—	—	—	VCOUT0	INTP3	—	TSCAP	—	RTC1HZ	SCL11	—	—	—
24	P50	CCD03	EI50/ EO50	—	—	—	—	INTP1	—	TS00	—	—	SDA11	—	—	—
25	P17	CCD01	EO17	—	—	—	—	—	—	—	TI02/ TO02	—	(TxD0)	—	—	—
26	P16	CCD00	EO16	—	—	—	—	INTP5	—	—	TI01/ TO01	—	(RxD0)	—	—	—
27	P13	—	EO13	—	—	—	—	—	—	—	(TI04)/ (TO04)	—	—	—	—	—
28	P12	—	EI12/ EO12	TOOLTxD	—	—	—	(INTP5)	—	—	(TI05)/ (TO05)	—	SO00/ TxD0	—	—	—
29	P11	—	EI11/ EO11	TOOLRxD	—	—	—	—	—	—	(TI06)/ (TO06)	—	SI00/ RxD0/ SDA00	—	—	—

Table 1 - 17 Multiplexed Pin Functions of the 44-pin R7F104G Product (2/2)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
44LQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
30	P10	—	EI10/ EO10	—	—	—	—	—	—	—	(TI07)/ (TO07)	—	SCK00/ SCL00	—	—	—
31	P147	—	EI147	—	—	—	IVCMP0	—	—	—	—	—	—	—	—	—
32	P153	—	—	—	—	—	—	—	—	TS33	—	—	—	—	—	—
33	P150	—	—	—	—	—	—	—	—	TS30	—	—	—	—	—	—
34	P27	—	—	—	ANI7	—	—	—	—	TS25	—	—	—	—	—	—
35	P26	—	—	—	ANI6	—	—	—	—	TS24	—	—	—	—	—	—
36	P25	—	—	—	ANI5	—	—	—	—	TS23	—	—	—	—	—	—
37	P24	—	—	—	ANI4	—	—	—	—	TS22	—	—	—	—	—	—
38	P23	—	EI23	—	ANI3	ANO1	IVREF0	—	—	TS21	—	—	—	—	—	—
39	P22	—	EI22	—	ANI2	ANO0	—	—	—	TS20	—	—	—	—	—	—
40	P21	—	EI21	—	ANI1/ AVREFM	—	—	—	—	—	—	—	—	—	—	—
41	P20	—	EI20	—	ANI0/ AVREFP	—	—	—	—	—	—	—	—	—	—	—
42	P03	—	—	—	—	—	—	—	—	TS29	—	—	RxD1	—	—	—
43	P02	—	—	—	—	—	—	—	—	TS28	—	—	TxD1	—	—	—
44	P01	—	EI01/ EO01	—	—	—	—	—	—	TS27	TO00	—	—	—	—	—

1.4.14 48-pin R7F104G product

- 48-pin plastic LFQFP (7 × 7 mm, 0.50-mm pitch)



Caution Connect the REGC pin to Vss via a capacitor (0.47 to 1 µF).

Remark 1. For pin identification, see **1.5 Pin Identification**.

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

Table 1 - 18 Multiplexed Pin Functions of the 48-pin R7F104G Product (1/2)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
48LFQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
1	P31	—	EI31	(PCLBUZ0)	—	—	—	INTP4	—	TS01	TI03/ TO03	—	—	—	—	—
2	P64	—	—	—	—	—	—	—	—	TS12	TI10/ TO10	—	—	—	—	—
3	P65	—	—	—	—	—	—	—	—	TS13	TI11/ TO11	—	—	—	—	—
4	P66	—	—	—	—	—	—	—	—	TS14	TI12/ TO12	—	—	—	—	—
5	P77	—	—	—	—	—	—	INTP11	KR7	TS09	—	—	(TxD2)	—	—	—
6	P76	—	—	—	—	—	—	INTP10	KR6	TS08	—	—	(RxD2)	—	—	—
7	P73	—	—	—	—	—	—	—	KR3	TS05	—	—	—	—	—	—
8	P72	—	—	—	—	—	—	—	KR2	TS04	—	—	SO21	—	TxDA0	—
9	P71	—	—	—	—	—	—	—	KR1	TS03	—	—	SI21/ SDA21	—	RxDA0	—
10	P70	—	—	—	—	—	—	—	KR0	TS02	—	—	SCK21/ SCL21	—	—	RIN0
11	P06	—	—	—	—	—	—	—	—	TS11	TI06/ TO06	—	—	—	CLKA0	—
12	P05	—	—	—	—	—	—	—	—	TS10	TI05/ TO05	—	—	—	—	—
13	P30	—	EI30	—	—	—	VCOUT0	INTP3	—	TSCAP	—	RTC1HZ	SCL11	—	—	—
14	P50	CCD03	EI50/ EO50	—	—	—	—	INTP1	—	TS00	—	—	SDA11	—	—	—
15	P17	CCD01	EO17	—	—	—	—	—	—	—	TI02/ TO02	—	(TxD0)	—	—	—
16	P16	CCD00	EO16	—	—	—	—	INTP5	—	—	TI01/ TO01	—	(RxD0)	—	—	—
17	P14	—	EO14	—	—	—	—	—	—	—	(TI03)/ (TO03)	—	RxD2	(SCLA0)	—	—
18	P13	—	EO13	—	—	—	—	—	—	—	(TI04)/ (TO04)	—	TxD2	(SDAA0)	—	—
19	P12	—	EI12/ EO12	TOOLTxD	—	—	—	(INTP5)	—	—	(TI05)/ (TO05)	—	SO00/ TxD0	—	—	—
20	P11	—	EI11/ EO11	TOOLRxD	—	—	—	—	—	—	(TI06)/ (TO06)	—	SI00/ RxD0/ SDA00	—	—	—
21	P10	—	EI10/ EO10	—	—	—	—	—	—	—	(TI07)/ (TO07)	—	SCK00/ SCL00	—	—	—
22	P147	—	EI147	—	—	—	IVCMP0	—	—	—	—	—	—	—	—	—
23	P153	—	—	—	—	—	—	—	—	TS33	—	—	—	—	—	—
24	P150	—	—	—	—	—	—	—	—	TS30	—	—	—	—	—	—
25	P27	—	—	—	ANI7	—	—	—	—	TS25	—	—	—	—	—	—
26	P26	—	—	—	ANI6	—	—	—	—	TS24	—	—	—	—	—	—
27	P25	—	—	—	ANI5	—	—	—	—	TS23	—	—	—	—	—	—
28	P24	—	—	—	ANI4	—	—	—	—	TS22	—	—	—	—	—	—

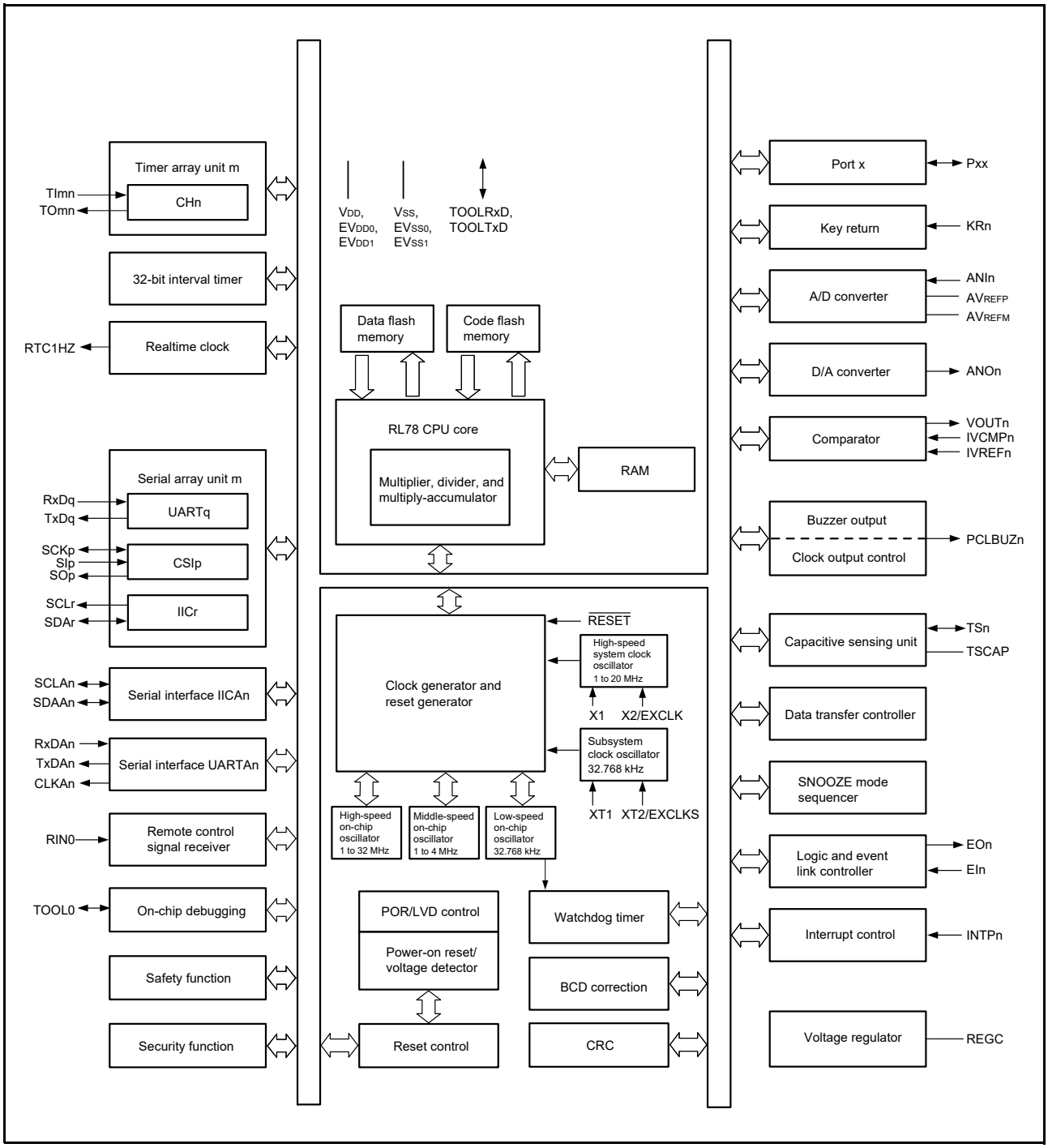
Table 1 - 18 Multiplexed Pin Functions of the 48-pin R7F104G Product (2/2)

Pin Number	I/O			Power supply, system clock, and debugging	Analog Circuits			HMIs			Timers		Communications Interfaces			
48LFQFP	Digital port	Controlled current drive port	ELCL input/output port		A/D converter (ADC)	D/A converter (DAC)	Comparator (CMP)	Interrupt	Key interrupt	Capacitive sensing unit (CTSU2L)	Timer array unit (TAU)	Realtime Clock (RTC)	Serial array unit (SAU)	Serial interface IICA (IICA)	Serial interface UARTA (UARTA)	Remote control signal receiver (REMC)
29	P23	—	EI23	—	ANI3	ANO1	IVREF0	—	—	TS21	—	—	—	—	—	—
30	P22	—	EI22	—	ANI2	ANO0	—	—	—	TS20	—	—	—	—	—	—
31	P21	—	EI21	—	ANI1/ AVREFM	—	—	—	—	—	—	—	—	—	—	—
32	P20	—	EI20	—	ANI0/ AVREFP	—	—	—	—	—	—	—	—	—	—	—
33	P130	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
34	P03	—	—	—	ANI16	—	—	—	—	TS29	—	—	RxD1	—	—	—
35	P02	—	—	—	ANI17	—	—	—	—	TS28	—	—	TxD1	—	—	—
36	P01	—	EI01/ EO01	—	—	—	—	—	—	TS27	TO00	—	—	—	—	—
37	P00	—	EI00	—	—	—	—	—	—	TS26	TI00	—	—	—	—	—
38	P41	—	—	—	—	—	—	—	—	—	TI07/ TO07	—	—	—	—	—
39	P40	—	—	TOOL0	—	—	—	—	—	—	—	—	—	—	—	—
40	—	—	—	$\overline{\text{RESET}}$	—	—	—	—	—	—	—	—	—	—	—	—
41	P124	—	—	XT2/ EXCLKS	—	—	—	—	—	—	—	—	—	—	—	—
42	P123	—	—	XT1	—	—	—	—	—	—	—	—	—	—	—	—
43	P137	—	EI137	—	—	—	—	INTP0	—	—	—	—	—	—	—	—
44	P122	—	EI122	X2/EXCLK	—	—	—	—	—	—	—	—	—	—	—	—
45	P121	—	EI121	X1/VBAT	—	—	—	—	—	—	—	—	—	—	—	—
46	—	—	—	REGC	—	—	—	—	—	—	—	—	—	—	—	—
47	—	—	—	VSS	—	—	—	—	—	—	—	—	—	—	—	—
48	—	—	—	VDD	—	—	—	—	—	—	—	—	—	—	—	—

1.5 Pin Identification

ANI0 to ANI14, ANI16 to ANI26:	Analog input
ANO0, ANO1:	Analog output
AV_{REFM}:	Analog reference voltage minus
AV_{REFP}:	Analog reference voltage plus
CCD00 to CCD07:	Controlled current drive output
CLKA0, CLKA1:	Asynchronous serial clock output
EI00, EI01, EI10 to EI12, EI20 to EI23, EI30, EI31, EI50, EI51, EI120 to EI122, EI137, EI147:	Logic & event link controller input
EO01, EO10 to EO17, EO50, EO51, EO60, EO61:	Logic & event link controller output
EV_{DD0}, EV_{DD1}:	Power supply for port
EV_{SS0}, EV_{SS1}:	Ground for port
EXCLK:	External clock input (main system clock)
EXCLKS:	External clock input (subsystem clock)
INTP0 to INTP11:	Interrupt request from peripheral modules
IVCMP0, IVCMP1:	Comparator input
IVREF0, IVREF1:	Comparator reference input
KR0 to KR7:	Key return input
P00 to P07:	Port 0
P10 to P17:	Port 1
P20 to P27:	Port 2
P30 to P37:	Port 3
P40 to P47:	Port 4
P50 to P57:	Port 5
P60 to P67:	Port 6
P70 to P77:	Port 7
P80 to P87:	Port 8
P90 to P97:	Port 9
P100 to P106:	Port 10
P110 to P117:	Port 11
P120 to P127:	Port 12
P130, P137:	Port 13
P140 to P147:	Port 14
P150 to P156:	Port 15
PCLBUZ0, PCLBUZ1:	Programmable clock output/buzzer output
REGC:	Regulator capacitance
$\overline{\text{RESET}}$:	Reset
RIN0:	IR remote controller input
RTC1HZ:	Realtime clock correction clock (1 Hz) output
RxD0 to RxD3, RxDA0, RxDA1:	Receive data
SCLA0, SCLA1, SCK00, SCK01, SCK10, SCK11, SCK20, SCK21, SCK30, SCK31:	Serial clock input/output
SCL00, SCL01, SCL10, SCL11, SCL20, SCL21, SCL30, SCL31:	Serial clock output
SDAA0, SDAA1, SDA00, SDA01, SDA10, SDA11, SDA20, SDA21, SDA30, SDA31:	Serial data input/output
SI00, SI01, SI10, SI11, SI20, SI21, SI30, SI31:	Serial data input
SO00, SO01, SO10, SO11, SO20, SO21, SO30, SO31:	Serial data output
TSCAP:	Touch sensor capacitance
TI00 to TI07, TI10 to TI17:	Timer input
TO00 to TO07, TO10 to TO17:	Timer output
TOOL0:	Data input/output for tool
TOOLRxD, TOOLTxD:	Data input/output for external device
TS00 to TS15, TS20 to TS35:	Capacitive touch sensor
TxD0 to TxD3, TxDA0, TxDA1:	Transmit data
VBAT:	Battery backup power supply
VCOUT0, VCOUT1:	Comparator output
V_{DD}:	Power supply
V_{SS}:	Ground
X1, X2:	Crystal oscillator (main system clock)
XT1, XT2:	Crystal oscillator (subsystem clock)

1.6 Block Diagram



Caution 1. Only the 32- to 128-pin R7F100G and 32- to 48-pin R7F104G products incorporate the remote control signal receiver.

Caution 2. Only the 36- to 128-pin R7F100G and 32- to 48-pin R7F104G products incorporate the serial interface UARTAn.

Caution 3. Only the 40- to 128-pin R7F100G and 32- to 48-pin R7F104G products incorporate the key return function.

Caution 4. Only the 30- to 128-pin R7F100G and 48-pin R7F104G products incorporate the serial interface IICAn.

Remark m: Unit number, n: Channel number, p: Simplified SPI (CSI) number, q: UART number, r: Simplified I^2C number, xx: Port number

1.7 Outline of Functions

[30- to 36-pin R7F100G and 32-pin R7F104G products]

Caution This outline describes the functions at the time when the peripheral I/O redirection register (PIOR) is set to 00H.

(1/3)

Item		30-pin	32-pin		36-pin
		R7F100GAx	R7F100GBx	R7F104GBG	R7F100GCx
Code flash memory		96 to 256 KB	96 to 256 KB	128 KB	96 to 256 KB
Data flash memory		8 KB	8 KB	8 KB	8 KB
RAM		12 to 24 KB	12 to 24 KB	16 KB	12 to 24 KB
Address space		1 MB			
CPU/ peripheral hardware clock frequency (f_{CLK})	Main system clock	HS (high-speed main) mode: 1 to 32 MHz (V_{DD} = 1.8 to 5.5 V) HS (high-speed main) mode: 1 to 4 MHz[Note 1] (V_{DD} = 1.6 to 5.5 V) LS (low-speed main) mode: 1 to 24 MHz (V_{DD} = 1.8 to 5.5 V) LS (low-speed main) mode: 1 to 4 MHz[Note 1] (V_{DD} = 1.6 to 5.5 V) LP (low-power main) mode: 1 to 2 MHz[Note 2] (V_{DD} = 1.6 to 5.5 V)			
	Subsystem clock	SUB mode: 32.768 kHz (V_{DD} = 1.6 to 5.5 V)			
Main system clock	High-speed system clock (f_{MX})	1 to 20 MHz			
	High-speed on-chip oscillator clock (f_{IH})	1 MHz, 2 MHz, 3 MHz, 4 MHz, 6 MHz, 8 MHz, 12 MHz, 16 MHz, 24 MHz, 32 MHz			
	Middle-speed on-chip oscillator clock (f_{IM})	1 MHz, 2 MHz, 4 MHz			
Subsystem clock	Subsystem clock X (f_{SX})	32.768 kHz (V_{DD} = 1.6 to 5.5 V or V_{DD} = 2.4 to 5.5 V when f_{XT} is selected)		—	32.768 kHz (V_{DD} = 1.6 to 5.5 V or V_{DD} = 2.4 to 5.5 V when f_{XT} is selected)
	Low-speed on-chip oscillator clock (f_{IL})	32.768 kHz (typ.)			
General-purpose registers		8 bits × 32 registers (8 bits × 8 registers × 4 banks)			
Minimum instruction execution time		0.03125 μs (at the 32-MHz operation with the high-speed on-chip oscillator clock (f_{IH}))			
Instruction set		• Data transfer (8/16 bits) • Adder and subtractor/logical operation (8/16 bits) • Multiplication (8 bits × 8 bits, 16 bits × 16 bits), division (16 bits ÷ 16 bits, 32 bits ÷ 32 bits) • Multiplication and accumulation (16 bits × 16 bits + 32 bits) • Rotate, barrel shift, and bit manipulation (set, reset, test, and Boolean operation), etc.			
I/O port	Total number of pins	26	28	28	32
	CMOS I/O	23 (N-ch open-drain I/O [V_{DD} withstand voltage]: 10)	24 (N-ch open-drain I/O [V_{DD} withstand voltage]: 10)	27 (N-ch open-drain I/O [V_{DD} withstand voltage]: 10)	28 (N-ch open-drain I/O [V_{DD} withstand voltage]: 12)
	CMOS input	1	1	1	1
	CMOS output	—	—	—	—
	N-ch open-drain I/O (withstand voltage: 6 V)	2	3	—	3
	Controlled current drive port	6	7	3	7

(2/3)

Item		30-pin	32-pin		36-pin
		R7F100GAx	R7F100GBx	R7F104GBG	R7F100GCx
Timers	16-bit timer	8 channels			
	Watchdog timer	1 channel			
	Realtime clock (RTC)	1 channel			
	32-bit interval timer (TML32)	1 channel in 32-bit counter mode, 2 channels in 16-bit counter mode, 4 channels in 8-bit counter mode			
	Timer output	4 channels (PWM outputs: 3Note 3), 8 channels (PWM outputs: 7Note 3)Note 4		5 channels (PWM outputs: 5Note 3), 4 channels (PWM outputs: 4Note 3)Note 4	4 channels (PWM outputs: 3Note 3), 8 channels (PWM outputs: 7Note 3)Note 4
	RTC output	1 channel		—	1 channel
Clock output/buzzer output		2 channels		1 channel	2 channels
		• 3.91 kHz, 7.81 kHz, 15.63 kHz, 2 MHz, 4 MHz, 8 MHz, 16 MHz (at the 32-MHz operation with the main system clock (f_{MAIN})) • 256 Hz, 512 Hz, 1.024 kHz, 2.048 kHz, 4.096 kHz, 8.192 kHz, 16.384 kHz, 32.768 kHz (at the 32.768-kHz operation with the low-speed peripheral clock (f_{SXP}))			
8-/10-/12-bit resolution A/D converter		8 channels		7 channels	8 channels
D/A converter		2 channels			
Comparator		2 channels		1 channel	2 channels
Serial interfaces		[30- and 32-pin R7F100G products] • Single simplified SPI (CSI) or simplified I^2C channel, or single UART • Single simplified SPI (CSI) or simplified I^2C channel, or single UART • Single simplified SPI (CSI) or simplified I^2C channel, or single UART also supporting the LIN bus [32-pin R7F104G product] • Single simplified SPI (CSI) or simplified I^2C channel, or single UART • Single simplified I^2C or UART channel, and not usable for simplified SPI (CSI) • Single simplified SPI (CSI) or simplified I^2C channel, or single UART also supporting the LIN bus [36-pin R7F100G products] • Single simplified SPI (CSI) or simplified I^2C channel, or single UART • Single simplified SPI (CSI) or simplified I^2C channel, or single UART • Two simplified SPI (CSI) or simplified I^2C channels, or single UART also supporting the LIN bus			
	UARTA	—		1 channel	
	I^2C bus	1 channel		—	1 channel
Remote control signal receiver		—	1 channel		
Data transfer controller (DTC)		30 sources	30 sources	31 sources	32 sources
Logic and event link controller (ELCL)		1			
SNOOZE mode sequencer (SMS)		1			
Capacitive sensing unit	Code flash memory size: 96 to 128 KB	2	3	10	5
	Code flash memory size: 192 to 768 KB	6	7	—	11
Vectored interrupt sources	Internal	31	32	39	35
	External	6	6	8	6
Key interrupt		—		6	—

(3/3)

Item		30-pin	32-pin		36-pin
		R7F100GAx	R7F100GBx	R7F104GBG	R7F100GCx
Reset		• Reset by $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on-reset • Internal reset by voltage detectors (LVD0 and LVD1) • Internal reset by illegal instruction execution**Note 5** • Internal reset by RAM parity error • Internal reset by illegal-memory access			
Power-on-reset circuit		Detection voltage • 1.50 V (typ.)			
Voltage detector	LVD0	Detection voltage • Rising edge: 1.69 V to 3.96 V (6 stages) • Falling edge: 1.65 V to 3.88 V (6 stages)			
	LVD1	Detection voltage • Rising edge: 1.67 V to 4.16 V (18 stages) • Falling edge: 1.63 V to 4.08 V (18 stages)			
On-chip debugging		Available (tracing supported)			
Power supply voltage		V_{DD} = 1.6 to 5.5 V			
Operating ambient temperature		T_A = -40 to +85°C (2D: Consumer applications), T_A = -40 to +105°C (3C: Industrial applications)			

Note 1. Overwrite the flash memory during operation at 2 MHz or a lower frequency.

Note 2. When the flash memory is to be overwritten, switch to high-speed main (HS) mode or low-speed main (LS) mode.

Note 3. The number of PWM outputs varies depending on the setting of channels in use (the number of masters and slaves). For details, see **7.9.3 Operation for the multiple PWM output function**.

Note 4. This applies when the corresponding bit in the peripheral I/O redirection register (PIOR) is set to 1.

Note 5. In normal operation, executing the instruction code FFH triggers an internal reset, but this is not the case during emulation by the on-chip debugging emulator.

[40- to 48-pin R7F100G and 44- and 48-pin R7F104G products]

Caution This outline describes the functions at the time when the peripheral I/O redirection register (PIOR) is set to 00H.

(1/3)

Item		40-pin	44-pin		48-pin	
		R7F100GEx	R7F100GFx	R7F104GFJ	R7F100GGx	R7F104GGJ
Code flash memory		96 to 256 KB	96 to 768 KB	256 KB	96 to 768 KB	256 KB
Data flash memory		8 KB	8 KB	8 KB	8 KB	8 KB
RAM		12 to 24 KB	12 to 48 KB	24 KB	12 to 48 KB	24 KB
Address space		1 MB				
CPU/ peripheral hardware clock frequency (f_{CLK})	Main system clock	HS (high-speed main) mode: 1 to 32 MHz (V_{DD} = 1.8 to 5.5 V) HS (high-speed main) mode: 1 to 4 MHz[Note 1] (V_{DD} = 1.6 to 5.5 V) LS (low-speed main) mode: 1 to 24 MHz (V_{DD} = 1.8 to 5.5 V) LS (low-speed main) mode: 1 to 4 MHz[Note 1] (V_{DD} = 1.6 to 5.5 V) LP (low-power main) mode: 1 to 2 MHz[Note 2] (V_{DD} = 1.6 to 5.5 V)				
	Subsystem clock	SUB mode: 32.768 kHz (V_{DD} = 1.6 to 5.5 V)				
Main system clock	High-speed system clock (f_{MX})	1 to 20 MHz				
	High-speed on-chip oscillator clock (f_{IH})	1 MHz, 2 MHz, 3 MHz, 4 MHz, 6 MHz, 8 MHz, 12 MHz, 16 MHz, 24 MHz, 32 MHz				
	Middle-speed on-chip oscillator clock (f_{IM})	1 MHz, 2 MHz, 4 MHz				
Subsystem clock	Subsystem clock X (f_{SX})	32.768 kHz (V_{DD} = 1.6 to 5.5 V)				
	Low-speed on-chip oscillator clock (f_{IL})	32.768 kHz (typ.)				
General-purpose registers		8 bits × 32 registers (8 bits × 8 registers × 4 banks)				
Minimum instruction execution time		0.03125 μs (at the 32-MHz operation with the high-speed on-chip oscillator clock (f_{IH}))				
Instruction set		• Data transfer (8/16 bits) • Adder and subtractor/logical operation (8/16 bits) • Multiplication (8 bits × 8 bits, 16 bits × 16 bits), division (16 bits ÷ 16 bits, 32 bits ÷ 32 bits) • Multiplication and accumulation (16 bits × 16 bits + 32 bits) • Rotate, barrel shift, and bit manipulation (set, reset, test, and Boolean operation), etc.				
I/O port	Total number of pins	36	40	40	44	44
	CMOS I/O	30 (N-ch open-drain I/O [V_{DD} withstand voltage]: 12)	33 (N-ch open-drain I/O [V_{DD} withstand voltage]: 12)	37 (N-ch open-drain I/O [V_{DD} withstand voltage]: 11)	36 (N-ch open-drain I/O [V_{DD} withstand voltage]: 13)	40 (N-ch open-drain I/O [V_{DD} withstand voltage]: 12)
	CMOS input	3	3	3	3	3
	CMOS output	—	—	—	1	1
	N-ch open-drain I/O (withstand voltage: 6 V)	3	4	—	4	—
	Controlled current drive port	7	7	3	8	3

(2/3)

Item		40-pin	44-pin		48-pin	
		R7F100GEx	R7F100GFx	R7F104GFJ	R7F100GGx	R7F104GGJ
Timers	16-bit timer	8 channels		12 channels	8 channels	12 channels
	Watchdog timer	1 channel				
	Realtime clock (RTC)	1 channel				
	32-bit interval timer (TML32)	1 channel in 32-bit counter mode, 2 channels in 16-bit counter mode, 4 channels in 8-bit counter mode				
	Timer output	4 channels (PWM outputs: 3Note 3), 8 channels (PWM outputs: 7Note 3)Note 4	5 channels (PWM outputs: 4Note 3), 8 channels (PWM outputs: 7Note 3)Note 4	8 channels (PWM outputs: 6Note 3), 8 channels (PWM outputs: 6Note 3)Note 4	5 channels (PWM outputs: 4Note 3), 8 channels (PWM outputs: 7Note 3)Note 4	10 channels (PWM outputs: 8Note 3), 10 channels (PWM outputs: 8Note 3)Note 4
	RTC output	1 channel				
Clock output/buzzer output		2 channels		1 channel	2 channels	1 channel
		• 3.91 kHz, 7.81 kHz, 15.63 kHz, 2 MHz, 4 MHz, 8 MHz, 16 MHz (at the 32-MHz operation with the main system clock (f_{MAIN})) • 256 Hz, 512 Hz, 1.024 kHz, 2.048 kHz, 4.096 kHz, 8.192 kHz, 16.384 kHz, 32.768 kHz (at the 32.768-kHz operation with the low-speed peripheral clock (f_{SXP}))				
8-/10-/12-bit resolution A/D converter		9 channels	10 channels	8 channels	10 channels	
D/A converter		2 channels				
Comparator		2 channels		1 channel	2 channels	1 channel
Serial interfaces		[40- and 44-pin R7F100G products] • Single simplified SPI (CSI) or simplified I^2C channel, or single UART • Single simplified SPI (CSI) or simplified I^2C channel, or single UART • Two simplified SPI (CSI) or simplified I^2C channels, or single UART also supporting the LIN bus [44- and 48-pin R7F104G products] • Single simplified SPI (CSI) or simplified I^2C channel, or single UART • Single simplified I^2C or UART channel, and not usable for simplified SPI (CSI) • Single simplified SPI (CSI) or simplified I^2C channel, or single UART also supporting the LIN bus [48-pin R7F100G products] • Two simplified SPI (CSI) or simplified I^2C channels, or single UART • Single simplified SPI (CSI) or simplified I^2C channel, or single UART • Two simplified SPI (CSI) or simplified I^2C channels, or single UART also supporting the LIN bus				
	UARTA	1 channel	2 channels	1 channel	2 channels	1 channel
	I^2C bus	1 channel	2 channels	—	2 channels	1 channel
Remote control signal receiver		1 channel				
Data transfer controller (DTC)		33 sources	35 sources	31 sources	36 sources	31 sources
Logic and event link controller (ELCL)		1				
SNOOZE mode sequencer (SMS)		1				
Capacitive sensing unit	Code flash memory size: 96 to 128 KB	6	6	—	8	—
	Code flash memory size: 192 to 768 KB	13	14	24	16	25
Vectored interrupt sources	Internal	35	39	41	39	42
	External	7	7	8	10	8
Key interrupt		4		6		

(3/3)

Item		40-pin	44-pin		48-pin	
		R7F100GEx	R7F100GFx	R7F104GFJ	R7F100GGx	R7F104GGJ
Reset		• Reset by $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on-reset • Internal reset by voltage detectors (LVD0 and LVD1) • Internal reset by illegal instruction execution**Note 5** • Internal reset by RAM parity error • Internal reset by illegal-memory access				
Power-on-reset circuit		Detection voltage • 1.50 V (typ.)				
Voltage detector	LVD0	Detection voltage • Rising edge: 1.69 V to 3.96 V (6 stages) • Falling edge: 1.65 V to 3.88 V (6 stages)				
	LVD1	Detection voltage • Rising edge: 1.67 V to 4.16 V (18 stages) • Falling edge: 1.63 V to 4.08 V (18 stages)				
On-chip debugging		Available (tracing supported)				
Power supply voltage		V_{DD} = 1.6 to 5.5 V				
Operating ambient temperature		T_A = -40 to +85°C (2D: Consumer applications), T_A = -40 to +105°C (3C: Industrial applications)				

Note 1. Overwrite the flash memory during operation at 2 MHz or a lower frequency.

Note 2. When the flash memory is to be overwritten, switch to high-speed main (HS) mode or low-speed main (LS) mode.

Note 3. The number of PWM outputs varies depending on the setting of channels in use (the number of masters and slaves). For details, see **7.9.3 Operation for the multiple PWM output function**.

Note 4. This applies when the corresponding bit in the peripheral I/O redirection register (PIOR) is set to 1.

Note 5. In normal operation, executing the instruction code FFH triggers an internal reset, but this is not the case during emulation by the on-chip debugging emulator.

[52- to 128-pin R7F100G products]

Caution This outline describes the functions at the time when the peripheral I/O redirection register (PIOR) is set to 00H.

(1/3)

Item		52-pin	64-pin	80-pin	100-pin	128-pin
		R7F100GJx	R7F100GLx	R7F100GMx	R7F100GPx	R7F100GSx
Code flash memory		96 to 768 KB	96 to 768 KB	128 to 768 KB	128 to 768 KB	256 to 768 KB
Data flash memory		8 KB	8 KB	8 KB	8 KB	8 KB
RAM		12 to 48 KB	12 to 48 KB	16 to 48 KB	16 to 48 KB	24 to 48 KB
Address space		1 MB				
CPU/ peripheral hardware clock frequency (f_{CLK})	Main system clock	HS (high-speed main) mode: 1 to 32 MHz (V_{DD} = 1.8 to 5.5 V) HS (high-speed main) mode: 1 to 4 MHz[Note 1] (V_{DD} = 1.6 to 5.5 V) LS (low-speed main) mode: 1 to 24 MHz (V_{DD} = 1.8 to 5.5 V) LS (low-speed main) mode: 1 to 4 MHz[Note 1] (V_{DD} = 1.6 to 5.5 V) LP (low-power main) mode: 1 to 2 MHz[Note 2] (V_{DD} = 1.6 to 5.5 V)				
	Subsystem clock	SUB mode: 32.768 kHz (V_{DD} = 1.6 to 5.5 V)				
Main system clock	High-speed system clock (f_{MX})	1 to 20 MHz				
	High-speed on-chip oscillator clock (f_{IH})	1 MHz, 2 MHz, 3 MHz, 4 MHz, 6 MHz, 8 MHz, 12 MHz, 16 MHz, 24 MHz, 32 MHz				
	Middle-speed on-chip oscillator clock (f_{IM})	1 MHz, 2 MHz, 4 MHz				
Subsystem clock	Subsystem clock X (f_{SX})	32.768 kHz (V_{DD} = 1.6 to 5.5 V)				
	Low-speed on-chip oscillator clock (f_{IL})	32.768 kHz (typ.)				
General-purpose registers		8 bits × 32 registers (8 bits × 8 registers × 4 banks)				
Minimum instruction execution time		0.03125 μs (at the 32-MHz operation with the high-speed on-chip oscillator clock (f_{IH}))				
Instruction set		• Data transfer (8/16 bits) • Adder and subtractor/logical operation (8/16 bits) • Multiplication (8 bits × 8 bits, 16 bits × 16 bits), division (16 bits ÷ 16 bits, 32 bits ÷ 32 bits) • Multiplication and accumulation (16 bits × 16 bits + 32 bits) • Rotate, barrel shift, and bit manipulation (set, reset, test, and Boolean operation), etc.				
I/O port	Total number of pins	48	58	74	92	120
	CMOS I/O	40 (N-ch open-drain I/O [V_{DD} withstand voltage]: 15)	50 (N-ch open-drain I/O [EV_{DD} withstand voltage]: 22[Note 3]/ 18[Note 4])	66 (N-ch open-drain I/O [EV_{DD} withstand voltage]: 27)	84 (N-ch open-drain I/O [EV_{DD} withstand voltage]: 31)	112 (N-ch open-drain I/O [EV_{DD} withstand voltage]: 33)
	CMOS input	3	3	3	3	3
	CMOS output	1	1	1	1	1
	N-ch open-drain I/O (withstand voltage: 6 V)	4	4	4	4	4
	Controlled current drive port	8	8	8	8	8

(2/3)

Item		52-pin	64-pin	80-pin	100-pin	128-pin
		R7F100GJx	R7F100GLx	R7F100GMx	R7F100GPx	R7F100GSx
Timers	16-bit timer	8 channels		12 channels		16 channels
	Watchdog timer	1 channel				
	Realtime clock (RTC)	1 channel				
	32-bit interval timer (TML32)	1 channel in 32-bit counter mode, 2 channels in 16-bit counter mode, 4 channels in 8-bit counter mode				
	Timer output	5 channels (PWM outputs: 4Note 5), 8 channels (PWM outputs: 7Note 5)Note 6	8 channels (PWM outputs: 7Note 5)	12 channels (PWM outputs: 10Note 5)		16 channels (PWM outputs: 14Note 5)
	RTC output	1 channel				
Clock output/buzzer output		2 channels				
		• 3.91 kHz, 7.81 kHz, 15.63 kHz, 2 MHz, 4 MHz, 8 MHz, 16 MHz (at the 32-MHz operation with the main system clock (fMAIN)) • 256 Hz, 512 Hz, 1.024 kHz, 2.048 kHz, 4.096 kHz, 8.192 kHz, 16.384 kHz, 32.768 kHz (at the 32.768-kHz operation with the low-speed peripheral clock (fSXP))				
8-/10-/12-bit resolution A/D converter		12 channels	12 channels	17 channels	20 channels	26 channels
D/A converter		2 channels				
Comparator		2 channels				
Serial interfaces		[52-pin products] • Two simplified SPI (CSI) or simplified I2C channels, or single UART • Single simplified SPI (CSI) or simplified I2C channel, or single UART • Two simplified SPI (CSI) or simplified I2C channels, or single UART also supporting the LIN bus [64-pin products] • Two simplified SPI (CSI) or simplified I2C channels, or single UART • Two simplified SPI (CSI) or simplified I2C channels, or single UART • Two simplified SPI (CSI) or simplified I2C channels, or single UART also supporting the LIN bus [80- to 128-pin products] • Two simplified SPI (CSI) or simplified I2C channels, or single UART • Two simplified SPI (CSI) or simplified I2C channels, or single UART • Two simplified SPI (CSI) or simplified I2C channels, or single UART also supporting the LIN bus • Two simplified SPI (CSI) or simplified I2C channels, or single UART				
	UARTA	2 channels				
	I2C bus	2 channels				
Remote control signal receiver		1 channel				
Data transfer controller (DTC)		36 sources	37 sources	39 sources		
Logic and event link controller (ELCL)		1				
SNOOZE mode sequencer (SMS)		1				
Capacitive sensing unit	Code flash memory size: 96 to 128 KB	10	12	30	32	32
	Code flash memory size: 192 to 768 KB	20	22	30	32	32
Vectored interrupt sources	Internal	39	39	44	44	48
	External	12	13	13	13	13

(3/3)

Item		52-pin	64-pin	80-pin	100-pin	128-pin
		R7F100GJx	R7F100GLx	R7F100GMx	R7F100GPx	R7F100GSx
Key interrupt		8				
Reset		• Reset by $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on-reset • Internal reset by voltage detectors (LVD0 and LVD1) • Internal reset by illegal instruction execution**Note 7** • Internal reset by RAM parity error • Internal reset by illegal-memory access				
Power-on-reset circuit		Detection voltage • 1.50 V (typ.)				
Voltage detector	LVD0	Detection voltage • Rising edge: 1.69 V to 3.96 V (6 stages) • Falling edge: 1.65 V to 3.88 V (6 stages)				
	LVD1	Detection voltage • Rising edge: 1.67 V to 4.16 V (18 stages) • Falling edge: 1.63 V to 4.08 V (18 stages)				
On-chip debugging		Available (tracing supported)				
Power supply voltage		V_{DD} = 1.6 to 5.5 V				
Operating ambient temperature		T_A = -40 to +85°C (2D: Consumer applications), T_A = -40 to +105°C (3C: Industrial applications)				

Note 1. Overwrite the flash memory during operation at 2 MHz or a lower frequency.

Note 2. When the flash memory is to be overwritten, switch to high-speed main (HS) mode or low-speed main (LS) mode.

Note 3. This only applies to the products with 96- or 128-Kbyte code flash memory.

Note 4. This only applies to the products with 192- to 768-Kbyte code flash memory.

Note 5. The number of PWM outputs varies depending on the setting of channels in use (the number of masters and slaves). For details, see **7.9.3 Operation for the multiple PWM output function**.

Note 6. This applies when the corresponding bit in the peripheral I/O redirection register (PIOR) is set to 1.

Note 7. In normal operation, executing the instruction code FFH triggers an internal reset, but this is not the case during emulation by the on-chip debugging emulator.

Section 2 Pin Functions

2.1 Functions of Port Pins

Pin I/O buffer power supplies depend on the product. The relationship between these power supplies and the pins is shown below.

Table 2 - 1 Pin I/O Buffer Power Supplies

1. 30- to 52-pin R7F100G and 32- to 48-pin R7F104G products

Power Supply	Corresponding Pins
V_{DD}	All pins

2. 64-pin R7F100G products

Power Supply	Corresponding Pins
EV_{DD0}	• Port pins other than P20 to P27, P121 to P124, and P137
V_{DD}	• P20 to P27, P121 to P124, and P137 • $\overline{RESET}$, REGC

3. 80-pin R7F100G products

Power Supply	Corresponding Pins
EV_{DD0}	• Port pins other than P20 to P27, P121 to P124, P137, and P150 to P153
V_{DD}	• P20 to P27, P121 to P124, P137, and P150 to P153 • $\overline{RESET}$, REGC

4. 100- and 128-pin R7F100G products

Power Supply	Corresponding Pins
EV_{DD0}, EV_{DD1}	• Port pins other than P20 to P27, P121 to P124, P137, and P150 to P156
V_{DD}	• P20 to P27, P121 to P124, P137, and P150 to P156 • $\overline{RESET}$, REGC

Caution EV_{DD0} and EV_{DD1} should have the same voltage.

2.1.1 R7F100G products with 30 pins

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P00	7-33-4	I/O	Analog function	ANI17/TS26Note 1/EI00/ TI00/TxD1	Port 0. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P01 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P00 as an output. P00 and P01 can be set for the analog pin functionsNote 2.
P01	8-33-2			ANI16/TS27Note 1/EI01/ EO01/TO00/RxD1	
P10	8-1-11	I/O	Input port	EI10/EO10/SCK00/SCL00/ (TI07)/(TO07)	Port 1. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P13 to P17 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P10 to P15 and P17 as outputs. P16 and P17 can be set as controlled current drive port pins. P13 can be set for the analog pin functionNote 2.
P11				EI11/EO11/SI00/RxD0/ TOOLRxD/SDA00/(TI06)/ (TO06)	
P12	7-1-12			EI12/EO12/SO00/TxD0/ TOOLTxD/(TI05)/(TO05)	
P13	8-6-9		Analog function	IVREF1/EO13/TxD2/SO20/ (SDAA0)/(TI04)/(TO04)	
P14	8-1-11		Input port	VCOUT1/EO14/RxD2/SI20/ SDA20/(SCLA0)/(TI03)/ (TO03)	
P15				EO15/PCLBUZ1/SCK20/ SCL20/(TI02)/(TO02)	
P16	8-38-1			EO16/CCD00/TI01/TO01/ INTP5/(RxD0)	
P17	8-38-2			EO17/CCD01/TI02/TO02/ (TxD0)	
P20	4-3-5	I/O	Analog function	ANI0/AVREFP/EI20	Port 2. 4-bit I/O port. Input or output can be specified in 1-bit units. P20 to P23 can be set for the analog pin functions Note 2.
P21				ANI1/AVREFM/EI21	
P22	4-35-1			ANI2/ANO0/TS20Note 1/ EI22	
P23	4-37-1			ANI3/ANO1/IVREF0/ TS21Note 1/EI23	
P30	7-31-2	I/O	Input port	VCOUT0/TSCAP/EI30/ INTP3/RTC1HZ/SCK11/ SCL11	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P31				TS01/EI31/TI03/TO03/ INTP4/PCLBUZ0	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P50	7-39-1	I/O	Input port	TS00/EI50/EO50/CCD03/ INTP1/SI11/SDA11	Port 5. 2-bit I/O port. N-ch open-drain output (withstand voltage of VDD) can be set for P50 as an output. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P50 and P51 can be set as controlled current drive port pins.
P51	7-38-1			EI51/EO51/CCD02/INTP2/ SO11	
P60	12-38-3	I/O	Input port	EO60/CCD04/SCLA0	Port 6. 2-bit I/O port. Input or output can be specified in 1-bit units. The outputs of P60 and P61 are N-ch open-drain (withstand voltage of 6 V). P60 and P61 can be set as controlled current drive port pins.
P61				EO61/CCD05/SDAA0	
P120	7-9-6	I/O	Analog function	ANI19/IVCMP1/EI120	Port 12. 3-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. N-ch open-drain output (withstand voltage of VDD) can be set for P120 as an output. P120 can be set for the analog pin function**Note 2**.
P121	7-2-1		Input port	X1/XT1/EI121	
P122				X2/EXCLK/XT2/EXCLKS/ EI122	
P137	2-1-3	Input	Input port	EI137/INTP0	Port 13. 1-bit input-only port.
P147	7-9-5	I/O	Analog function	ANI18/IVCMP0/EI147	Port 14. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P147 can be set for the analog pin function**Note 2**.
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to VDD, either directly or via a resistor.

Note 1. These functions are not present in products with 128 or fewer Kbytes of code flash memory.

Note 2. Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register (PMCAxx).

Remark Pin functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

2.1.2 R7F100G products with 32 pins

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P00	7-33-4	I/O	Analog function	ANI17/TS26Note 1/EI00/ TI00/TxD1	Port 0. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P01 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P00 as an output. P00 and P01 can be set for the analog pin functionsNote 2.
P01	8-33-2			ANI16/TS27Note 1/EI01/ EO01/TO00/RxD1	
P10	8-1-11	I/O	Input port	EI10/EO10/SCK00/SCL00/ (TI07)/(TO07)	Port 1. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P13 to P17 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P10 to P15 and P17 as outputs. P16 and P17 can be set as controlled current drive port pins. P13 can be set for the analog pin functionNote 2.
P11				EI11/EO11/SI00/RxD0/ TOOLRxD/SDA00/(TI06)/ (TO06)	
P12	7-1-12			EI12/EO12/SO00/TxD0/ TOOLTxD/(TI05)/(TO05)	
P13	8-6-9		Analog function	IVREF1/EO13/TxD2/SO20/ (SDAA0)/(TI04)/(TO04)	
P14	8-1-11		Input port	VCOUT1/EO14/RxD2/ SI20/SDA20/(SCLA0)/ (TI03)/(TO03)	
P15				EO15/PCLBUZ1/SCK20/ SCL20/(TI02)/(TO02)	
P16	8-38-1			EO16/CCD00/TI01/TO01/ INTP5/(RxD0)	
P17	8-38-2			EO17/CCD01/TI02/TO02/ (TxD0)	
P20	4-3-5	I/O	Analog function	ANI0/AVREFP/EI20	Port 2. 4-bit I/O port. Input or output can be specified in 1-bit units. P20 to P23 can be set for the analog pin functionsNote 2.
P21				ANI1/AVREFM/EI21	
P22	4-35-1			ANI2/ANO0/TS20Note 1/ EI22	
P23	4-37-1			ANI3/ANO1/IVREF0/ TS21Note 1/EI23	
P30	7-31-2	I/O	Input port	VCOUT0/TSCAP/EI30/ INTP3/RTC1HZ/SCK11/ SCL11	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P31				TS01/EI31/TI03/TO03/ INTP4/PCLBUZ0	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P50	7-39-1	I/O	Input port	TS00/EI50/EO50/CCD03/INTP1/SI11/SDA11	Port 5. 2-bit I/O port. N-ch open-drain output (withstand voltage of VDD) can be set for P50 as an output. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P50 and P51 can be set as controlled current drive port pins.
P51	7-38-1			EI51/EO51/CCD02/INTP2/SO11	
P60	12-38-3	I/O	Input port	EO60/CCD04/SCLA0	Port 6. 3-bit I/O port. Input or output can be specified in 1-bit units. The outputs of P60 to P62 are N-ch open-drain (withstand voltage of 6 V). P60 to P62 can be set as controlled current drive port pins.
P61				EO61/CCD05/SDAA0	
P62	12-38-1			CCD06	
P70	7-31-2	I/O	Input port	RIN0/TS02	Port 7. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P120	7-9-6	I/O	Analog function	ANI19/IVCMP1/EI120	Port 12. 3-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. N-ch open-drain output (withstand voltage of VDD) can be set for P120 as an output. P120 can be set for the analog pin function[Note 2].
P121	7-2-1		Input port	X1/XT1/EI121	
P122				X2/EXCLK/XT2/EXCLKS/EI122	
P137	2-1-3	Input	Input port	EI137/INTP0	Port 13. 1-bit input-only port.
P147	7-9-5	I/O	Analog function	ANI18/IVCMP0/EI147	Port 14. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P147 can be set for the analog pin function[Note 2].
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to VDD, either directly or via a resistor.

Note 1. These functions are not present in products with 128 or fewer Kbytes of code flash memory.

Note 2. Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register (PMCAxx).

Remark Pin functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

2.1.3 R7F100G products with 36 pins

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P00	7-31-3	I/O	Input port	TS26Note 1/EI00/TI00/ TxD1	Port 0. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P01 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P00 as an output.
P01	8-31-1			TS27Note 1/EI01/EO01/ TO00/RxD1	
P10	8-1-11	I/O	Input port	EI10/EO10/SCK00/SCL00/ (TI07)/(TO07)	Port 1. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P13 to P17 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P10 to P15 and P17 as outputs. P16 and P17 can be set as controlled current drive port pins. P13 can be set for the analog pin functionNote 2.
P11				EI11/EO11/SI00/RxD0/ TOOLRxD/SDA00/(TI06)/ (TO06)	
P12	7-1-12			EI12/EO12/SO00/TxD0/ TOOLTxD/(TI05)/(TO05)	
P13	8-6-9		Analog function	IVREF1/EO13/TxD2/SO20/ (SDAA0)/(TI04)/(TO04)	
P14	8-1-11		Input port	VCOUT1/EO14/RxD2/ SI20/SDA20/(SCLA0)/ (TI03)/(TO03)	
P15				EO15/PCLBUZ1/SCK20/ SCL20/(TI02)/(TO02)	
P16	8-38-1			EO16/CCD00/TI01/TO01/ INTP5/(RxD0)	
P17	8-38-2			EO17/CCD01/TI02/TO02/ (TxD0)	
P20	4-3-5	I/O	Analog function	ANI0/AVREFP/EI20	Port 2. 6-bit I/O port. Input or output can be specified in 1-bit units. P20 to P25 can be set for the analog pin functionsNote 2.
P21				ANI1/AVREFM/EI21	
P22	4-35-1			ANI2/ANO0/TS20Note 1/ EI22	
P23	4-37-1			ANI3/ANO1/IVREF0/ TS21Note 1/EI23	
P24	4-33-1			ANI4/TS22Note 1	
P25				ANI5/TS23Note 1	
P30	7-31-2	I/O	Input port	VCOUT0/TSCAP/EI30/ INTP3/RTC1HZ/SCK11/ SCL11	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P31				TS01/EI31/TI03/TO03/ INTP4/PCLBUZ0	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P50	7-39-1	I/O	Input port	TS00/EI50/EO50/CCD03/INTP1/SI11/SDA11	Port 5. 2-bit I/O port. N-ch open-drain output (withstand voltage of VDD) can be set for P50 as an output. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P50 and P51 can be set as controlled current drive port pins.
P51	7-38-1			EI51/EO51/CCD02/INTP2/SO11	
P60	12-38-3	I/O	Input port	EO60/CCD04/SCLA0	Port 6. 3-bit I/O port. Input or output can be specified in 1-bit units. The outputs of P60 to P62 are N-ch open-drain (withstand voltage of 6 V). P60 to P62 can be set as controlled current drive port pins.
P61				EO61/CCD05/SDAA0	
P62	12-38-1			CCD06	
P70	7-31-2	I/O	Input port	TS02/RIN0/SCK21/SCL21	Port 7. 3-bit I/O port. A TTL input buffer can be set for P71 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P71 and P72 as outputs. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P71	8-31-2			TS03/SI21/SDA21/RxDA0	
P72	7-31-3			TS04/SO21/TxDA0	
P120	7-9-6	I/O	Analog function	ANI19/IVCMP1/EI120	Port 12. 3-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. N-ch open-drain output (withstand voltage of VDD) can be set for P120 as an output. P120 can be set for the analog pin function**Note 2**.
P121	7-2-1		Input port	X1/XT1/EI121	
P122				X2/EXCLK/XT2/EXCLKS/EI122	
P137	2-1-3	Input	Input port	EI137/INTP0	Port 13. 1-bit input-only port.
P147	7-9-5	I/O	Analog function	ANI18/IVCMP0/EI147	Port 14. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P147 can be set for the analog pin function**Note 2**.
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to VDD, either directly or via a resistor.

Note 1. These functions are not present in products with 128 or fewer Kbytes of code flash memory.

Note 2. Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register (PMCAxx).

Remark Pin functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

2.1.4 R7F100G products with 40 pins

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P00	7-31-3	I/O	Input port	TS26Note 1/EI00/TI00/ TxD1	Port 0. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P01 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P00 as an output.
P01	8-31-1			TS27Note 1/EI01/EO01/ TO00/RxD1	
P10	8-1-11	I/O	Input port	EI10/EO10/SCK00/SCL00/ (TI07)/(TO07)	Port 1. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P13 to P17 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P10 to P15 and P17 as outputs. P16 and P17 can be set as controlled current drive port pins. P13 can be set for the analog pin functionNote 2.
P11				EI11/EO11/SI00/RxD0/ TOOLRxD/SDA00/(TI06)/ (TO06)	
P12	7-1-12			EI12/EO12/SO00/TxD0/ TOOLTxD/(TI05)/(TO05)	
P13	8-6-9		Analog function	IVREF1/EO13/TxD2/SO20/ (SDAA0)/(TI04)/(TO04)	
P14	8-1-11		Input port	VCOUT1/EO14/RxD2/ SI20/SDA20/(SCLA0)/ (TI03)/(TO03)	
P15				EO15/PCLBUZ1/SCK20/ SCL20/(TI02)/(TO02)	
P16	8-38-1			EO16/CCD00/TI01/TO01/ INTP5/(RxD0)	
P17	8-38-2			EO17/CCD01/TI02/TO02/ (TxD0)	
P20	4-3-5	I/O	Analog function	ANI0/AVREFP/EI20	Port 2. 7-bit I/O port. Input or output can be specified in 1-bit units. P20 to P26 can be set for the analog pin functionsNote 2.
P21				ANI1/AVREFM/EI21	
P22	4-35-1			ANI2/ANO0/TS20Note 1/ EI22	
P23	4-37-1			ANI3/ANO1/IVREF0/ TS21Note 1/EI23	
P24	4-33-1			ANI4/TS22Note 1	
P25				ANI5/TS23Note 1	
P26				ANI6/TS24Note 1	
P30	7-31-2	I/O	Input port	VCOUT0/TSCAP/EI30/ INTP3/RTC1HZ/SCK11/ SCL11	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P31				TS01/EI31/TI03/TO03/ INTP4/PCLBUZ0	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P50	7-39-1	I/O	Input port	TS00/EI50/EO50/CCD03/ INTP1/SI11/SDA11	Port 5. 2-bit I/O port. N-ch open-drain output (withstand voltage of VDD) can be set for P50 as an output. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P50 and P51 can be set as controlled current drive port pins.
P51	7-38-1			EI51/EO51/CCD02/INTP2/ SO11	
P60	12-38-3	I/O	Input port	EO60/CCD04/SCLA0	Port 6. 3-bit I/O port. Input or output can be specified in 1-bit units. The outputs of P60 to P62 are N-ch open-drain (withstand voltage of 6 V). P60 to P62 can be set as controlled current drive port pins.
P61				EO61/CCD05/SDAA0	
P62	12-38-1			CCD06	
P70	7-31-2	I/O	Input port	TS02/RIN0/KR0/SCK21/ SCL21	Port 7. 4-bit I/O port. A TTL input buffer can be set for P71 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P71 and P72 as outputs. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P71	8-31-2			TS03/KR1/SI21/SDA21/ RxDA0	
P72	7-31-3			TS04/KR2/SO21/TxDA0	
P73	7-31-2			TS05/KR3	
P120	7-9-6	I/O	Analog function	ANI19/IVCMP1/EI120	Port 12. 3-bit I/O port and 2-bit input-only port. For P120 to P122, input or output can be specified in 1-bit units. For P120 to P122, use of an on-chip pull-up resistor can be specified by a software setting at input port. N-ch open-drain output (withstand voltage of VDD) can be set for P120 as an output. P120 can be set for the analog pin function**Note 2**.
P121	7-2-1		Input port	X1/VBAT/EI121	
P122				X2/EXCLK/EI122	
P123	2-2-1	Input		XT1	
P124				XT2/EXCLKS	
P137	2-1-3	Input	Input port	EI137/INTP0	Port 13. 1-bit input-only port.
P147	7-9-5	I/O	Analog function	ANI18/IVCMP0/EI147	Port 14. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P147 can be set for the analog pin function**Note 2**.
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to VDD, either directly or via a resistor.

Note 1. These functions are not present in products with 128 or fewer Kbytes of code flash memory.

Note 2. Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register (PMCAxx).

Remark Pin functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

2.1.5 R7F100G products with 44 pins

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P00	7-31-3	I/O	Input port	TS26Note 1/EI00/TI00/ TxD1	Port 0. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P01 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P00 as an output.
P01	8-31-1			TS27Note 1/EI01/EO01/ TO00/RxD1	
P10	8-1-11	I/O	Input port	EI10/EO10/SCK00/SCL00/ (TI07)/(TO07)	Port 1. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P13 to P17 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P10 to P15 and P17 as outputs. P16 and P17 can be set as controlled current drive port pins. P13 can be set for the analog pin functionNote 2.
P11				EI11/EO11/SI00/RxD0/ TOOLRxD/SDA00/(TI06)/ (TO06)	
P12	7-1-12			EI12/EO12/SO00/TxD0/ TOOLTxD/(TI05)/(TO05)	
P13	8-6-9		Analog function	IVREF1/EO13/TxD2/SO20/ (SDAA0)/(TI04)/(TO04)	
P14	8-1-11		Input port	VCOUT1/EO14/RxD2/ SI20/SDA20/(SCLA0)/ (TI03)/(TO03)	
P15				EO15/PCLBUZ1/SCK20/ SCL20/(TI02)/(TO02)	
P16	8-38-1			EO16/CCD00/TI01/TO01/ INTP5/(RxD0)	
P17	8-38-2			EO17/CCD01/TI02/TO02/ (TxD0)	
P20	4-3-5	I/O	Analog function	ANI0/AVREFP/EI20	Port 2. 8-bit I/O port. Input or output can be specified in 1-bit units. P20 to P27 can be set for the analog pin functionsNote 2.
P21				ANI1/AVREFM/EI21	
P22	4-35-1			ANI2/ANO0/TS20Note 1/ EI22	
P23	4-37-1			ANI3/ANO1/IVREF0/ TS21Note 1/EI23	
P24	4-33-1			ANI4/TS22Note 1	
P25				ANI5/TS23Note 1	
P26				ANI6/TS24Note 1	
P27				ANI7/TS25Note 1	
P30	7-31-2	I/O	Input port	VCOUT0/TSCAP/EI30/ INTP3/RTC1HZ/SCK11/ SCL11	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P31				TS01/EI31/TI03/TO03/ INTP4/PCLBUZ0	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P41 as an input.
P41	8-1-3			RxDA1/TI07/TO07	

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P50	7-39-1	I/O	Input port	TS00/EI50/EO50/INTP1/ SI11/SDA11	Port 5. 2-bit I/O port. N-ch open-drain output (withstand voltage of VDD) can be set for P50 as an output. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P51 can be set as a controlled current drive port pin.
P51	7-38-1			EI51/EO51/CCD02/INTP2/ SO11	
P60	12-38-3	I/O	Input port	EO60/CCD04/SCLA0	Port 6. 4-bit I/O port. Input or output can be specified in 1-bit units. The outputs of P60 to P63 are N-ch open-drain (withstand voltage of 6 V). P60 to P63 can be set as controlled current drive port pins.
P61				EO61/CCD05/SDAA0	
P62	12-38-2			CCD06/SCLA1	
P63				CCD07/SDAA1	
P70	7-31-2	I/O	Input port	TS02/RIN0/KR0/SCK21/ SCL21	Port 7. 4-bit I/O port. A TTL input buffer can be set for P71 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P71 and P72 as outputs. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P71	8-31-2			TS03/KR1/SI21/SDA21/ RxDA0	
P72	7-31-3			TS04/KR2/SO21/TxDA0	
P73	7-31-2			TS05/KR3	
P120	7-9-6	I/O	Analog function	ANI19/IVCMP1/TxDA1/ EI120	Port 12. 3-bit I/O port and 2-bit input-only port. For P120 to P122, input or output can be specified in 1-bit units. For P120 to P122, use of an on-chip pull-up resistor can be specified by a software setting at input port. N-ch open-drain output (withstand voltage of VDD) can be set for P120 as an output. P120 can be set for the analog pin function**Note 2**.
P121	7-2-1		Input port	X1/VBAT/EI121	
P122				X2/EXCLK/EI122	
P123	2-2-1	Input		XT1	
P124				XT2/EXCLKS	
P137	2-1-3	Input	Input port	EI137/INTP0	Port 13. 1-bit input-only port.
P146	7-1-3	I/O	Input port	—	Port 14. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P147 can be set for the analog pin function**Note 2**.
P147	7-9-5		Analog function	ANI18/IVCMP0/EI147	
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to VDD, either directly or via a resistor.

Note 1. These functions are not present in products with 128 or fewer Kbytes of code flash memory.

Note 2. Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register (PMCAxx).

Remark Pin functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

2.1.6 R7F100G products with 48 pins

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P00	7-31-3	I/O	Input port	TS26Note 1/EI00/TI00/ TxD1	Port 0. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P01 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P00 as an output.
P01	8-31-1			TS27Note 1/EI01/EO01/ TO00/RxD1	
P10	8-1-11	I/O	Input port	EI10/EO10/SCK00/SCL00/ (TI07)/(TO07)	Port 1. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P13 to P17 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P10 to P15 and P17 as outputs. P16 and P17 can be set as controlled current drive port pins. P13 can be set for the analog pin functionNote 2.
P11				EI11/EO11/SI00/RxD0/ TOOLRxD/SDA00/(TI06)/ (TO06)	
P12	7-1-12			EI12/EO12/SO00/TxD0/ TOOLTxD/(TI05)/(TO05)	
P13	8-6-9		Analog function	IVREF1/EO13/TxD2/SO20/ (SDAA0)/(TI04)/(TO04)	
P14	8-1-11		Input port	VCOUT1/EO14/RxD2/ SI20/SDA20/(SCLA0)/ (TI03)/(TO03)	
P15				EO15/PCLBUZ1/SCK20/ SCL20/(TI02)/(TO02)	
P16	8-38-1			EO16/CCD00/TI01/TO01/ INTP5/(RxD0)	
P17	8-38-2			EO17/CCD01/TI02/TO02/ (TxD0)	
P20	4-3-5	I/O	Analog function	ANI0/AVREFP/EI20	Port 2. 8-bit I/O port. Input or output can be specified in 1-bit units. P20 to P27 can be set for the analog pin functionsNote 2.
P21				ANI1/AVREFM/EI21	
P22	4-35-1			ANI2/ANO0/TS20Note 1/ EI22	
P23	4-37-1			ANI3/ANO1/IVREF0/ TS21Note 1/EI23	
P24	4-33-1			ANI4/TS22Note 1	
P25				ANI5/TS23Note 1	
P26				ANI6/TS24Note 1	
P27				ANI7/TS25Note 1	
P30	7-31-2	I/O	Input port	VCOUT0/TSCAP/EI30/ INTP3/RTC1HZ/SCK11/ SCL11	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P31				TS01/EI31/TI03/TO03/ INTP4/(PCLBUZ0)	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P41 as an input.
P41	8-1-3			RxDA1/TI07/TO07	

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P50	7-39-1	I/O	Input port	TS00/EI50/EO50/CCD03/ INTP1/SI11/SDA11	Port 5. 2-bit I/O port. N-ch open-drain output (withstand voltage of VDD) can be set for P50 as an output. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P50 and P51 can be set as controlled current drive port pins.
P51	7-38-1			EI51/EO51/CCD02/INTP2/ SO11	
P60	12-38-3	I/O	Input port	EO60/CCD04/SCLA0	Port 6. 4-bit I/O port. Input or output can be specified in 1-bit units. The outputs of P60 to P63 are N-ch open-drain (withstand voltage of 6 V). P60 to P63 can be set as controlled current drive port pins.
P61				EO61/CCD05/SDAA0	
P62	12-38-2			CCD06/SCLA1	
P63				CCD07/SDAA1	
P70	7-31-2	I/O	Input port	TS02/RIN0/KR0/SCK21/ SCL21	Port 7. 6-bit I/O port. A TTL input buffer can be set for P71 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P71, P72, and P74 as outputs. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P71	8-31-2			TS03/KR1/SI21/SDA21/ RxDA0	
P72	7-31-3			TS04/KR2/SO21/TxDA0	
P73	7-31-2			TS05/KR3/SO01	
P74	7-31-3			TS06/KR4/INTP8/SI01/ SDA01	
P75	7-31-2			TS07/KR5/INTP9/SCK01/ SCL01	
P120	7-9-6	I/O	Analog function	ANI19/IVCMP1/TxDA1/ EI120	Port 12. 3-bit I/O port and 2-bit input-only port. For P120 to P122, input or output can be specified in 1-bit units. For P120 to P122, use of an on-chip pull-up resistor can be specified by a software setting at input port. N-ch open-drain output (withstand voltage of VDD) can be set for P120 as an output. P120 can be set for the analog pin function**Note 2**.
P121	7-2-1		Input port	X1/VBAT/EI121	
P122				X2/EXCLK/EI122	
P123	2-2-1	Input		XT1	
P124				XT2/EXCLKS	
P130	1-1-1	Output	Output port	—	Port 13. 1-bit output-only port and 1-bit input-only port.
P137	2-1-3	Input	Input port	EI137/INTP0	
P140	7-1-3	I/O	Input port	PCLBUZ0/INTP6	Port 14. 3-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P147 can be set for the analog pin function**Note 2**.
P146				—	
P147	7-9-5		Analog function	ANI18/IVCMP0/EI147	
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to VDD, either directly or via a resistor.

Note 1. These functions are not present in products with 128 or fewer Kbytes of code flash memory.

Note 2. Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register (PMCAxx).

Remark Pin functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

2.1.7 R7F100G products with 52 pins

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P00	7-31-3	I/O	Input port	TS26Note 1/EI00/TI00	Port 0. 4-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P01 and P03 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P00, P02, and P03 as outputs. P02 and P03 can be set for the analog pin functions Note 2.
P01	8-31-1			TS27Note 1/EI01/EO01/TO00	
P02	7-33-4		Analog function	ANI17/TS28Note 1/TxD1	
P03	8-33-3			ANI16/TS29Note 1/RxD1	
P10	8-1-11	I/O	Input port	EI10/EO10/SCK00/SCL00/(TI07)/(TO07)	Port 1. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P13 to P17 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P10 to P15 and P17 as outputs. P16 and P17 can be set as controlled current drive port pins. P13 can be set for the analog pin functionNote 2.
P11				EI11/EO11/SI00/RxD0/TOOLRxD/SDA00/(TI06)/(TO06)	
P12	7-1-12			EI12/EO12/SO00/TxD0/TOOLTxD/(TI05)/(TO05)	
P13	8-6-9		Analog function	IVREF1/EO13/TxD2/SO20/(SDAA0)/(TI04)/(TO04)	
P14	8-1-11		Input port	VCOUT1/EO14/RxD2/SI20/SDA20/(SCLA0)/(TI03)/(TO03)	
P15				EO15/PCLBUZ1/SCK20/SCL20/(TI02)/(TO02)	
P16	8-38-1			EO16/CCD00/TI01/TO01/INTP5/(RxD0)	
P17	8-38-2			EO17/CCD01/TI02/TO02/(TxD0)	
P20	4-3-5	I/O	Analog function	ANI0/AVREFP/EI20	Port 2. 8-bit I/O port. Input or output can be specified in 1-bit units. P20 to P27 can be set for the analog pin functionsNote 2.
P21				ANI1/AVREFM/EI21	
P22	4-35-1			ANI2/ANO0/TS20Note 1/EI22	
P23	4-37-1			ANI3/ANO1/IVREF0/TS21Note 1/EI23	
P24	4-33-1			ANI4/TS22Note 1	
P25				ANI5/TS23Note 1	
P26				ANI6/TS24Note 1	
P27				ANI7/TS25Note 1	
P30	7-31-2	I/O	Input port	VCOUT0/TSCAP/EI30/INTP3/RTC1HZ/SCK11/SCL11	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P31				TS01/EI31/TI03/TO03/INTP4/(PCLBUZ0)	

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P41 as an input.
P41	8-1-3			RxDA1/TI07/TO07	
P50	7-39-1	I/O	Input port	TS00/EI50/EO50/CCD03/INTP1/SI11/SDA11	Port 5. 2-bit I/O port. N-ch open-drain output (withstand voltage of VDD) can be set for P50 as an output. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P50 and P51 can be set as controlled current drive port pins.
P51	7-38-1			EI51/EO51/CCD02/INTP2/SO11	
P60	12-38-3	I/O	Input port	EO60/CCD04/SCLA0	Port 6. 4-bit I/O port. Input or output can be specified in 1-bit units. The outputs of P60 to P63 are N-ch open-drain (withstand voltage of 6 V). P60 to P63 can be set as controlled current drive port pins.
P61				EO61/CCD05/SDAA0	
P62	12-38-2			CCD06/SCLA1	
P63				CCD07/SDAA1	
P70	7-31-2	I/O	Input port	TS02/RIN0/KR0/SCK21/SCL21	Port 7. 8-bit I/O port. A TTL input buffer can be set for P71 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P71, P72, and P74 as outputs. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P71	8-31-2			TS03/KR1/SI21/SDA21/RxDA0	
P72	7-31-3			TS04/KR2/SO21/TxDA0	
P73	7-31-2			TS05/KR3/SO01	
P74	7-31-3			TS06/KR4/INTP8/SI01/SDA01	
P75	7-31-2			TS07/KR5/INTP9/SCK01/SCL01	
P76				TS08/KR6/INTP10/(RxD2)	
P77				TS09/KR7/INTP11/(TxD2)	
P120	7-9-6	I/O	Analog function	ANI19/IVCMP1/TxDA1/EI120	Port 12. 3-bit I/O port and 2-bit input-only port. For P120 to P122, input or output can be specified in 1-bit units. For P120 to P122, use of an on-chip pull-up resistor can be specified by a software setting at input port. N-ch open-drain output (withstand voltage of VDD) can be set for P120 as an output. P120 can be set for the analog pin function[Note 2].
P121	7-2-1		Input port	X1/VBAT/EI121	
P122				X2/EXCLK/EI122	
P123	2-2-1	Input		XT1	
P124				XT2/EXCLKS	
P130	1-1-1	Output	Output port	—	Port 13. 1-bit output-only port and 1-bit input-only port.
P137	2-1-3	Input	Input port	EI137/INTP0	
P140	7-1-3	I/O	Input port	PCLBUZ0/INTP6	Port 14. 3-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P147 can be set for the analog pin function[Note 2].
P146				—	
P147	7-9-5		Analog function	ANI18/IVCMP0/EI147	
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to VDD, either directly or via a resistor.

(Notes and Remark are listed on the next page.)

Note 1. These functions are not present in products with 128 or fewer Kbytes of code flash memory.

Note 2. Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register (PMCAxx).

Remark Pin functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

2.1.8 R7F100G products with 64 pins

(1/3)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P00	7-31-3	I/O	Input port	TS26Note 1/EI00/TI00	Port 0. 7-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P01, P03, and P04 as inputs. N-ch open-drain output (withstand voltage of EVDD) can be set for P00 and P02 to P04 as outputs. P02 and P03 can be set for the analog pin functions Note 2.
P01	8-31-1			TS27Note 1/EI01/EO01/ TO00	
P02	7-33-4		Analog function	ANI17/TS28Note 1/SO10/ TxD1	
P03	8-33-3			ANI16/TS29Note 1/SI10/ RxD1/SDA10	
P04	8-1-10		Input port	SCK10/SCL10	
P05	7-31-2			TS10/TI05/TO05	
P06				TS11/TI06/TO06/CLKA0	
P10	8-1-11	I/O	Input port	EI10/EO10/SCK00/SCL00/ (TI07)/(TO07)	Port 1. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P13 to P17 as inputs. N-ch open-drain output (withstand voltage of EVDD) can be set for P10 to P15 and P17 as outputs. P16 and P17 can be set as controlled current drive port pins. P13 can be set for the analog pin functionNote 2.
P11				EI11/EO11/SI00/RxD0/ TOOLRxD/SDA00/(TI06)/ (TO06)	
P12	7-1-12			EI12/EO12/SO00/TxD0/ TOOLTxD/ (INTP5)/ (TI05)/(TO05)	
P13	8-6-9		Analog function	IVREF1/EO13/TxD2/SO20/ (SDAA0)/(TI04)/(TO04)	
P14	8-1-11		Input port	VCOUT1/EO14/RxD2/ SI20/SDA20/(SCLA0)/ (TI03)/(TO03)	
P15				EO15/SCK20/SCL20/ (TI02)/(TO02)	
P16	8-38-1			EO16/CCD00/TI01/TO01/ INTP5/(SI00)/(RxD0)	
P17	8-38-2			EO17/CCD01/TI02/TO02/ (SO00)/(TxD0)	
P20	4-3-5	I/O	Analog function	ANI0/AVREFP/EI20	Port 2. 8-bit I/O port. Input or output can be specified in 1-bit units. P20 to P27 can be set for the analog pin functionsNote 2.
P21				ANI1/AVREFM/EI21	
P22	4-35-1			ANI2/ANO0/TS20Note 1/ EI22	
P23	4-37-1			ANI3/ANO1/IVREF0/ TS21Note 1/EI23	
P24	4-33-1			ANI4/TS22Note 1	
P25				ANI5/TS23Note 1	
P26				ANI6/TS24Note 1	
P27				ANI7/TS25Note 1	
P30	7-31-2	I/O	Input port	VCOUT0/TSCAP/EI30/ INTP3/RTC1HZ/SCK11/ SCL11	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P31				TS01/EI31/TI03/TO03/ INTP4/(PCLBUZ0)	

(2/3)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 4-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P41 and P43Note 1 as inputs. N-ch open-drain output (withstand voltage of EVDD) can be set for P42 and P43Note 1 as outputs.
P41	8-1-3			RxDA1/TI07/TO07	
P42	7-1-11			TxDA1/TI04/TO04	
P43	8-1-10			CLKA1	
P50	7-39-1	I/O	Input port	TS00/EI50/EO50/CCD03/INTP1/SI11/SDA11	Port 5. 6-bit I/O port. TTL input buffers can be set for P53Note 1, P54Note 1, and P55 as inputs. N-ch open-drain output (withstand voltage of EVDD) can be set for P50, P52 to P54Note 1, and P55 as outputs. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P50 and P51 can be set as controlled current drive port pins.
P51	7-38-1			EI51/EO51/CCD02/INTP2/SO11	
P52	7-1-11			(INTP10)	
P53	8-1-10			(INTP11)	
P54				—	
P55				(PCLBUZ1)/(SCK00)	
P60	12-38-3	I/O	Input port	EO60/CCD04/SCLA0	Port 6. 4-bit I/O port. Input or output can be specified in 1-bit units. The outputs of P60 to P63 are N-ch open-drain (withstand voltage of 6 V). P60 to P63 can be set as controlled current drive port pins.
P61				EO61/CCD05/SDAA0	
P62	12-38-2			CCD06/SCLA1	
P63				CCD07/SDAA1	
P70	7-31-2	I/O	Input port	TS02/RIN0/KR0/SCK21/SCL21	Port 7. 8-bit I/O port. A TTL input buffer can be set for P71 as an input. N-ch open-drain output (withstand voltage of EVDD) can be set for P71, P72, and P74 as outputs. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P71	8-31-2			TS03/KR1/SI21/SDA21/RxDA0	
P72	7-31-3			TS04/KR2/SO21/TxDA0	
P73	7-31-2			TS05/KR3/SO01	
P74	7-31-3			TS06/KR4/INTP8/SI01/SDA01	
P75	7-31-2			TS07/KR5/INTP9/SCK01/SCL01	
P76				TS08/KR6/INTP10/(RxD2)	
P77				TS09/KR7/INTP11/(TxD2)	
P120	7-9-6	I/O	Analog function	ANI19/IVCMP1/EI120	Port 12. 3-bit I/O port and 2-bit input-only port. For P120 to P122, input or output can be specified in 1-bit units. For P120 to P122, use of an on-chip pull-up resistor can be specified by a software setting at input port. N-ch open-drain output (withstand voltage of EVDD) can be set for P120 as an output. P120 can be set for the analog pin functionNote 2.
P121	7-2-1		Input port	X1/VBAT/EI121	
P122				X2/EXCLK/EI122	
P123	2-2-1	Input		XT1	
P124				XT2/EXCLKS	
P130	1-1-1	Output	Output port	—	Port 13. 1-bit output-only port and 1-bit input-only port.
P137	2-1-3	Input	Input port	EI137/INTP0	

(3/3)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P140	7-1-3	I/O	Input port	PCLBUZ0/INTP6	Port 14. 4-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P147 can be set for the analog pin function**Note 2**.
P141				PCLBUZ1/INTP7	
P146				—	
P147	7-9-5		Analog function	ANI18/IVCMP0/EI147	
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to V_{DD}, either directly or via a resistor.

Note 1. These functions are not present in products with 128 or fewer Kbytes of code flash memory.

Note 2. Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register (PMCAxx).

Remark Pin functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

2.1.9 R7F100G products with 80 pins

(1/3)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P00	7-31-3	I/O	Input port	TS26/EI00/TI00	Port 0. 7-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P01, P03, and P04 as inputs. N-ch open-drain output (withstand voltage of EV_{DD}) can be set for P00 and P02 to P04 as outputs. P02 and P03 can be set for the analog pin functions**Note**.
P01	8-31-1			TS27/EI01/EO01/TO00	
P02	7-33-4		Analog function	ANI17/TS28/SO10/TxD1	
P03	8-33-3			ANI16/TS29/SI10/RxD1/SDA10	
P04	8-1-10		Input port	SCK10/SCL10	
P05	7-31-2			TS10/TI05/TO05	
P06				TS11/TI06/TO06/CLKA0	
P10	8-1-11	I/O	Input port	EI10/EO10/SCK00/SCL00/(TI07)/(TO07)	Port 1. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P13 to P17 as inputs. N-ch open-drain output (withstand voltage of EV_{DD}) can be set for P10 to P15 and P17 as outputs. P16 and P17 can be set as controlled current drive port pins. P13 can be set for the analog pin function**Note**.
P11				EI11/EO11/SI00/RxD0/TOOLRxD/SDA00/(TI06)/(TO06)	
P12	7-1-12			EI12/EO12/SO00/TxD0/TOOLTxD/(INTP5)/(TI05)/(TO05)	
P13	8-6-9		Analog function	IVREF1/EO13/TxD2/SO20/(SDAA0)/(TI04)/(TO04)	
P14	8-1-11		Input port	VCOUT1/EO14/RxD2/SI20/SDA20/(SCLA0)/(TI03)/(TO03)	
P15				EO15/SCK20/SCL20/(TI02)/(TO02)	
P16	8-38-1			EO16/CCD00/TI01/TO01/INTP5/(SI00)/(RxD0)	
P17	8-38-2			EO17/CCD01/TI02/TO02/(SO00)/(TxD0)	
P20	4-3-5	I/O	Analog function	ANI0/AV_{REFP}/EI20	Port 2. 8-bit I/O port. Input or output can be specified in 1-bit units. P20 to P27 can be set for the analog pin functions**Note**.
P21				ANI1/AV_{REFM}/EI21	
P22	4-35-1			ANI2/ANO0/TS20/EI22	
P23	4-37-1			ANI3/ANO1/IVREF0/TS21/EI23	
P24	4-33-1			ANI4/TS22	
P25				ANI5/TS23	
P26				ANI6/TS24	
P27				ANI7/TS25	
P30	7-31-2	I/O	Input port	VCOUT0/TSCAP/EI30/INTP3/RTC1HZ/SCK11/SCL11	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P31				TS01/EI31/TI03/TO03/INTP4/(PCLBUZ0)	

(2/3)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 6-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P41, P43, and P44 as inputs. N-ch open-drain output (withstand voltage of EV_{DD}) can be set for P42 to P45 as outputs.
P41	8-1-3			RxDA1/TI07/TO07	
P42	7-1-11			TxDA1/TI04/TO04	
P43	8-1-10			SCK01/SCL01/CLKA1	
P44				SI01/SDA01	
P45	7-1-11			SO01	
P50	7-39-1	I/O	Input port	TS00/EI50/EO50/CCD03/INTP1/SI11/SDA11	Port 5. 6-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P53 to P55 as inputs. N-ch open-drain output (withstand voltage of EV_{DD}) can be set for P50 and P52 to P55 as outputs. P50 and P51 can be set as controlled current drive port pins.
P51	7-38-1			EI51/EO51/CCD02/INTP2/SO11	
P52	7-1-11			SO31	
P53	8-1-10			SI31/SDA31	
P54				SCK31/SCL31	
P55				(PCLBUZ1)/(SCK00)	
P60	12-38-3	I/O	Input port	EO60/CCD04/SCLA0	Port 6. 8-bit I/O port. Input or output can be specified in 1-bit units. The outputs of P60 to P63 are N-ch open-drain (withstand voltage of 6 V). For P64 to P67, use of an on-chip pull-up resistor can be specified by a software setting at input port. P60 to P63 can be set as controlled current drive port pins.
P61				EO61/CCD05/SDAA0	
P62	12-38-2			CCD06/SCLA1	
P63				CCD07/SDAA1	
P64	7-31-2			TS12/TI10/TO10	
P65				TS13/TI11/TO11	
P66				TS14/TI12/TO12	
P67				TS15/TI13/TO13	
P70	7-31-2	I/O	Input port	TS02/RIN0/KR0/SCK21/SCL21	Port 7. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P71 as an input. N-ch open-drain output (withstand voltage of EV_{DD}) can be set for P71, P72, and P74 as outputs.
P71	8-31-2			TS03/KR1/SI21/SDA21/RxDA0	
P72	7-31-3			TS04/KR2/SO21/TxDA0	
P73	7-31-2			TS05/KR3	
P74	7-31-3			TS06/KR4/INTP8	
P75	7-31-2			TS07/KR5/INTP9	
P76				TS08/KR6/INTP10/(RxD2)	
P77				TS09/KR7/INTP11/(TxD2)	
P100	7-3-3	I/O	Analog function	ANI20	Port 10. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P100 can be set for the analog pin function**Note**.
P110	7-1-3	I/O	Input port	(INTP10)	Port 11. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P111				(INTP11)	

(3/3)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P120	7-9-6	I/O	Analog function	ANI19/IVCMP1/EI120	Port 12. 3-bit I/O port and 2-bit input-only port. For P120 to P122, input or output can be specified in 1-bit units. For P120 to P122, use of an on-chip pull-up resistor can be specified by a software setting at input port. N-ch open-drain output (withstand voltage of EVDD) can be set for P120 as an output. P120 can be set for the analog pin function**Note**.
P121	7-2-1		Input port	X1/VBAT/EI121	
P122				X2/EXCLK/EI122	
P123	2-2-1	Input		XT1	
P124				XT2/EXCLKS	
P130	1-1-1	Output	Output port	—	Port 13. 1-bit output-only port and 1-bit input-only port.
P137	2-1-3	Input	Input port	EI137/INTP0	
P140	7-1-3	I/O	Input port	PCLBUZ0/INTP6	Port 14. 7-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P142 and P143 as inputs. N-ch open-drain output (withstand voltage of EVDD) can be set for P142 to P144 as outputs. P147 can be set for the analog pin function**Note**.
P141				PCLBUZ1/INTP7	
P142	8-1-10			SCK30/SCL30	
P143				SI30/RxD3/SDA30	
P144	7-1-11			SO30/TxD3	
P146	7-1-3			—	
P147	7-9-5		Analog function	ANI18/IVCMP0/EI147	
P150	4-33-1	I/O	Analog function	ANI8/TS30	Port 15. 4-bit I/O port. Input or output can be specified in 1-bit units. P150 to P153 can be set for the analog pin functions**Note**.
P151				ANI9/TS31	
P152				ANI10/TS32	
P153				ANI11/TS33	
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to VDD, either directly or via a resistor.

Note Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register (PMCAxx).

Remark Pin functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR).**

2.1.10 R7F100G products with 100 pins

(1/4)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P00	7-31-3	I/O	Input port	TS26/EI00/TI00	Port 0. 7-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P01, P03, and P04 as inputs. N-ch open-drain output (withstand voltage of EV_{DD}) can be set for P00 and P02 to P04 as outputs. P02 and P03 can be set for the analog pin functions**Note**.
P01	8-31-1			TS27/EI01/EO01/TO00	
P02	7-33-4		Analog function	ANI17/TS28/SO10/TxD1	
P03	8-33-3			ANI16/TS29/SI10/RxD1/SDA10	
P04	8-1-10		Input port	SCK10/SCL10	
P05	7-31-2			TS10	
P06				TS11	
P10	8-1-11	I/O	Input port	EI10/EO10/SCK00/SCL00/(TI07)/(TO07)	Port 1. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P13 to P17 as inputs. N-ch open-drain output (withstand voltage of EV_{DD}) can be set for P10 to P15 and P17 as outputs. P16 and P17 can be set as controlled current drive port pins. P13 can be set for the analog pin function**Note**.
P11				EI11/EO11/SI00/RxD0/TOOLRxD/SDA00/(TI06)/(TO06)	
P12	7-1-12			EI12/EO12/SO00/TxD0/TOOLTxD/(INTP5)/(TI05)/(TO05)	
P13	8-6-9		Analog function	IVREF1/EO13/TxD2/SO20/(SDAA0)/(TI04)/(TO04)	
P14	8-1-11		Input port	VCOUT1/EO14/RxD2/SI20/SDA20/(SCLA0)/(TI03)/(TO03)	
P15				EO15/SCK20/SCL20/(TI02)/(TO02)	
P16	8-38-1			EO16/CCD00/TI01/TO01/INTP5/(SI00)/(RxD0)	
P17	8-38-2			EO17/CCD01/TI02/TO02/(SO00)/(TxD0)	
P20	4-3-5	I/O	Analog function	ANI0/AV_{REFP}/EI20	Port 2. 8-bit I/O port. Input or output can be specified in 1-bit units. P20 to P27 can be set for the analog pin functions**Note**.
P21				ANI1/AV_{REFM}/EI21	
P22	4-35-1			ANI2/ANO0/TS20/EI22	
P23	4-37-1			ANI3/ANO1/IVREF0/TS21/EI23	
P24	4-33-1			ANI4/TS22	
P25				ANI5/TS23	
P26				ANI6/TS24	
P27				ANI7/TS25	
P30	7-31-2	I/O	Input port	VCOUT0/TSCAP/EI30/INTP3/RTC1HZ/SCK11/SCL11	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P31				TS01/EI31/TI03/TO03/INTP4/(PCLBUZ0)	

(2/4)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P41, P43, and P44 as inputs. N-ch open-drain output (withstand voltage of EV_{DD}) can be set for P42 to P45 as outputs.
P41	8-1-3			RxDA1	
P42	7-1-11			TxDA1/TI04/TO04	
P43	8-1-10			SCK01/SCL01/CLKA1	
P44				SI01/SDA01	
P45	7-1-11			SO01	
P46	7-1-3			INTP1/TI05/TO05	
P47				INTP2	
P50	7-39-1	I/O	Input port	TS00/EI50/EO50/CCD03/SI11/SDA11	Port 5. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P53 to P55 as inputs. N-ch open-drain output (withstand voltage of EV_{DD}) can be set for P50 and P52 to P55 as outputs. P50 and P51 can be set as controlled current drive port pins.
P51	7-38-1			EI51/EO51/CCD02/SO11	
P52	7-1-11			SO31	
P53	8-1-10			SI31/SDA31	
P54				SCK31/SCL31	
P55				(PCLBUZ1)/(SCK00)	
P56	7-1-3			(INTP1)	
P57				(INTP3)	
P60	12-38-3	I/O	Input port	EO60/CCD04/SCLA0	Port 6. 8-bit I/O port. Input or output can be specified in 1-bit units. The outputs of P60 to P63 are N-ch open-drain (withstand voltage of 6 V). For P64 to P67, use of an on-chip pull-up resistor can be specified by a software setting at input port. P60 to P63 can be set as controlled current drive port pins.
P61				EO61/CCD05/SDAA0	
P62	12-38-2			CCD06/SCLA1	
P63				CCD07/SDAA1	
P64	7-31-2			TS12/TI10/TO10	
P65				TS13/TI11/TO11	
P66				TS14/TI12/TO12	
P67				TS15/TI13/TO13	
P70	7-31-2	I/O	Input port	TS02/RIN0/KR0/SCK21/SCL21	Port 7. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P71 as an input. N-ch open-drain output (withstand voltage of EV_{DD}) can be set for P71, P72, and P74 as outputs.
P71	8-31-2			TS03/KR1/SI21/SDA21	
P72	7-31-3			TS04/KR2/SO21	
P73	7-31-2			TS05/KR3	
P74	7-31-3			TS06/KR4/INTP8	
P75	7-31-2			TS07/KR5/INTP9	
P76				TS08/KR6/INTP10/(RxD2)	
P77				TS09/KR7/INTP11/(TxD2)	

(3/4)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P80	8-1-10	I/O	Input port	(SCK10)/(SCL10)	Port 8. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P80, P81, and P84 as inputs. N-ch open-drain output (withstand voltage of EVDD) can be set for P80 to P83 as outputs.
P81				(SI10)/(RxD1)/(SDA10)	
P82	7-1-11			(SO10)/(TxD1)	
P83				TxDA0	
P84	8-1-3			RxDA0/(INTP6)	
P85	7-1-3			CLKA0/(INTP7)	
P86				(INTP8)	
P87				(INTP9)	
P100	7-3-3	I/O	Analog function	ANI20	Port 10. 3-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P100 can be set for the analog pin function[Note].
P101	7-1-3		Input port	—	
P102				TI06/TO06	
P110	7-1-3	I/O	Input port	(INTP10)	Port 11. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P111				(INTP11)	
P120	7-9-6	I/O	Analog function	ANI19/IVCMP1/EI120	Port 12. 3-bit I/O port and 2-bit input-only port. For P120 to P122, input or output can be specified in 1-bit units. For P120 to P122, use of an on-chip pull-up resistor can be specified by a software setting at input port. N-ch open-drain output (withstand voltage of EVDD) can be set for P120 as an output. P120 can be set for the analog pin function[Note].
P121	7-2-1		Input port	X1/VBAT/EI121	
P122				X2/EXCLK/EI122	
P123	2-2-1	Input		XT1	
P124				XT2/EXCLKS	
P130	1-1-1	Output	Output port	—	Port 13. 1-bit output-only port and 1-bit input-only port.
P137	2-1-3	Input	Input port	EI137/INTP0	
P140	7-1-3	I/O	Input port	PCLBUZ0/INTP6	Port 14. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P142 and P143 as inputs. N-ch open-drain output (withstand voltage of EVDD) can be set for P142 to P144 as outputs. P147 can be set for the analog pin function[Note].
P141				PCLBUZ1/INTP7	
P142	8-1-10			SCK30/SCL30	
P143				SI30/RxD3/SDA30	
P144	7-1-11			SO30/TxD3	
P145	7-1-3			TI07/TO07	
P146				(INTP4)	
P147	7-9-5		Analog function	ANI18/IVCMP0/EI147	
P150	4-33-1	I/O	Analog function	ANI8/TS30	Port 15. 7-bit I/O port. Input or output can be specified in 1-bit units. P150 to P156 can be set for the analog pin functions[Note].
P151				ANI9/TS31	
P152				ANI10/TS32	
P153				ANI11/TS33	
P154				ANI12/TS34	
P155				ANI13/TS35	
P156	4-3-5			ANI14	

(4/4)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to VDD, either directly or via a resistor.

Note Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register (PMCAxx).

Remark Pin functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

2.1.11 R7F100G products with 128 pins

(1/4)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P00	7-31-3	I/O	Input port	TS26/EI00/TI00	Port 0. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P01, P03, and P04 as inputs. N-ch open-drain output (withstand voltage of EV_{DD}) can be set for P00 and P02 to P04 as outputs. P02 and P03 can be set for the analog pin functions**Note**.
P01	8-31-1			TS27/EI01/EO01/TO00	
P02	7-33-4		Analog function	ANI17/TS28/SO10/TxD1	
P03	8-33-3			ANI16/TS29/SI10/RxD1/ SDA10	
P04	8-1-10		Input port	SCK10/SCL10	
P05	7-31-2			TS10	
P06				TS11	
P07	7-1-3			—	
P10	8-1-11	I/O	Input port	EI10/EO10/SCK00/SCL00/ (TI07)/(TO07)	Port 1. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, and P13 to P17 as inputs. N-ch open-drain output (withstand voltage of EV_{DD}) can be set for P10 to P15 and P17 as outputs. P16 and P17 can be set as controlled current drive port pins. P13 can be set for the analog pin function**Note**.
P11				EI11/EO11/SI00/RxD0/ TOOLRxD/ SDA00/(TI06)/(TO06)	
P12	7-1-12			EI12/EO12/SO00/TxD0/ TOOLTxD/ (INTP5)/(TI05)/(TO05)	
P13	8-6-9		Analog function	IVREF1/EO13/TxD2/SO20/ (SDAA0)/(TI04)/(TO04)	
P14	8-1-11		Input port	VCOUT1/EO14/RxD2/ SI20/SDA20/(SCLA0)/ (TI03)/(TO03)	
P15				EO15/SCK20/SCL20/ (TI02)/(TO02)	
P16	8-38-1			EO16/CCD00/TI01/TO01/ INTP5/(SI00)/(RxD0)	
P17	8-38-2			EO17/CCD01/TI02/TO02/ (SO00)/(TxD0)	
P20	4-3-5	I/O	Analog function	ANI0/AV_{REFP}/EI20	Port 2. 8-bit I/O port. Input or output can be specified in 1-bit units. P20 to P27 can be set for the analog pin functions**Note**.
P21				ANI1/AV_{REFM}/EI21	
P22	4-35-1			ANI2/ANO0/TS20/EI22	
P23	4-37-1			ANI3/ANO1/IVREF0/TS21/ EI23	
P24	4-33-1			ANI4/TS22	
P25				ANI5/TS23	
P26				ANI6/TS24	
P27				ANI7/TS25	

(2/4)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P30	7-31-2	I/O	Input port	VCOUT0/TSCAP/EI30/ INTP3/RTC1HZ	Port 3. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P33 as an input. N-ch open-drain output (withstand voltage of EVDD) can be set for P34 as an output. P35 to P37 can be set for the analog pin functions**Note**.
P31				TS01/EI31/TI03/TO03/ INTP4/(PCLBUZ0)	
P32	7-1-3			CLKA1	
P33	8-1-3			RxDA1	
P34	7-1-11			TxDA1	
P35	7-3-3		Analog function	ANI23	
P36				ANI22	
P37				ANI21	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P41, P43, and P44 as inputs. N-ch open-drain output (withstand voltage of EVDD) can be set for P42 to P45 as outputs.
P41	8-1-3			—	
P42	7-1-11			TI04/TO04	
P43	8-1-10			SCK01/SCL01	
P44				SI01/SDA01	
P45	7-1-11			SO01	
P46	7-1-3			INTP1/TI05/TO05	
P47				INTP2	
P50	7-39-1	I/O	Input port	TS00/EI50/EO50/CCD03	Port 5. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P53 to P55 as inputs. N-ch open-drain output (withstand voltage of EVDD) can be set for P50 and P52 to P55 as outputs. P50 and P51 can be set as controlled current drive port pins.
P51	7-38-1			EI51/EO51/CCD02	
P52	7-1-11			SO31	
P53	8-1-10			SI31/SDA31	
P54				SCK31/SCL31	
P55				(PCLBUZ1)/(SCK00)	
P56	7-1-3			(INTP1)	
P57				(INTP3)	
P60	12-38-3	I/O	Input port	EO60/CCD04/SCLA0	Port 6. 8-bit I/O port. Input or output can be specified in 1-bit units. The outputs of P60 to P63 are N-ch open-drain (withstand voltage of 6 V). For P64 to P67, use of an on-chip pull-up resistor can be specified by a software setting at input port. P60 to P63 can be set as controlled current drive port pins.
P61				EO61/CCD05/SDAA0	
P62	12-38-2			CCD06/SCLA1	
P63				CCD07/SDAA1	
P64	7-31-2			TS12/TI10/TO10	
P65				TS13/TI11/TO11	
P66				TS14/TI12/TO12	
P67				TS15/TI13/TO13	

(3/4)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P70	7-31-2	I/O	Input port	TS02/RIN0/KR0/SCK21/SCL21	Port 7. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P71 as an input. N-ch open-drain output (withstand voltage of EVDD) can be set for P71, P72, and P74 as outputs.
P71	8-31-2			TS03/KR1/SI21/SDA21	
P72	7-31-3			TS04/KR2/SO21	
P73	7-31-2			TS05/KR3	
P74	7-31-3			TS06/KR4/INTP8	
P75	7-31-2			TS07/KR5/INTP9	
P76				TS08/KR6/INTP10/(RxD2)	
P77				TS09/KR7/INTP11/(TxD2)	
P80	8-1-10	I/O	Input port	(SCK10)/(SCL10)	Port 8. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P80, P81, and P84 as inputs. N-ch open-drain output (withstand voltage of EVDD) can be set for P80 to P83 as outputs.
P81				(SI10)/(RxD1)/(SDA10)	
P82	7-1-11			(SO10)/(TxD1)	
P83				TxDA0	
P84	8-1-3			RxDA0/(INTP6)	
P85	7-1-3			CLKA0/(INTP7)	
P86				(INTP8)	
P87				(INTP9)	
P90	7-1-3	I/O	Input port	—	Port 9. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. N-ch open-drain output (withstand voltage of EVDD) can be set for P96 as an output.
P91				—	
P92				—	
P93				—	
P94				—	
P95				SCK11/SCL11	
P96	7-1-11			SI11/SDA11	
P97	7-1-3			SO11	
P100	7-3-3	I/O	Analog function	ANI20	Port 10. 7-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P100 can be set for the analog pin function[Note].
P101	7-1-3		Input port	—	
P102				TI06/TO06	
P103				TI14/TO14	
P104				TI15/TO15	
P105				TI16/TO16	
P106				TI17/TO17	
P110	7-1-3	I/O	Input port	(INTP10)	Port 11. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P115 to P117 can be set for the analog pin functions[Note].
P111				(INTP11)	
P112				—	
P113				—	
P114				—	
P115	7-3-3		Analog function	ANI26	
P116				ANI25	
P117				ANI24	

(4/4)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P120	7-9-6	I/O	Analog function	ANI19/IVCMP1/EI120	Port 12. 6-bit I/O port and 2-bit input-only port. For P120 to P122 and P125 to P127, input or output can be specified in 1-bit units. For P120 to P122 and P125 to P127, use of an on-chip pull-up resistor can be specified by a software setting at input port. N-ch open-drain output (withstand voltage of EVDD) can be set for P120 as an output. P120 can be set for the analog pin function**Note**.
P121	7-2-1		Input port	X1/VBAT/EI121	
P122				X2/EXCLK/EI122	
P123	2-2-1	Input		XT1	
P124				XT2/EXCLKS	
P125	7-1-3	I/O		—	
P126				—	
P127				—	
P130	1-1-1	Output	Output port	—	Port 13. 1-bit output-only port and 1-bit input-only port.
P137	2-1-3	Input	Input port	EI137/INTP0	
P140	7-1-3	I/O	Input port	PCLBUZ0/INTP6	Port 14. 8-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P142 and P143 as inputs. N-ch open-drain output (withstand voltage of EVDD) can be set for P142 to P144 as outputs. P147 can be set for the analog pin function**Note**.
P141				PCLBUZ1/INTP7	
P142	8-1-10			SCK30/SCL30	
P143				SI30/RxD3/SDA30	
P144	7-1-11			SO30/TxD3	
P145	7-1-3			TI07/TO07	
P146				(INTP4)	
P147	7-9-5		Analog function	ANI18/IVCMP0/EI147	
P150	4-33-1	I/O	Analog function	ANI8/TS30	Port 15. 7-bit I/O port. Input or output can be specified in 1-bit units. P150 to P156 can be set for the analog pin functions**Note**.
P151				ANI9/TS31	
P152				ANI10/TS32	
P153				ANI11/TS33	
P154				ANI12/TS34	
P155				ANI13/TS35	
P156	4-3-5			ANI14	
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to VDD, either directly or via a resistor.

Note Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register (PMCAxx).

Remark Pin functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

2.1.12 R7F104G product with 32 pins

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P00	7-31-3	I/O	Input port	EI00/TI00	Port 0. 5-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P03 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P00, P02, and P03 as outputs. P02 and P03 can be set for the analog pin functions **Note**.
P02	7-33-4		Analog function	ANI17/TxD1	
P03	8-33-3			ANI16/RxD1	
P05	7-31-2		Input port	TS10/TI05/TO05	
P06				TS11/TI06/TO06/CLKA0	
P10	8-1-11	I/O	Input port	EI10/EO10/SCK00/SCL00/(TI07)/(TO07)	Port 1. 5-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, P16, and P17 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P10 to P12 and P17 as outputs. P16 and P17 can be set as controlled current drive port pins.
P11				EI11/EO11/SI00/RxD0/TOOLRxD/SDA00/(TI06)/(TO06)	
P12	7-1-12			EI12/EO12/SO00/TxD0/TOOLTxD/(INTP5)/(TI05)/(TO05)	
P16	8-38-1			EO16/CCD00/TI01/TO01/INTP5/(RxD0)	
P17	8-38-2			EO17/CCD01/TI02/TO02/(TxD0)	
P20	4-3-5	I/O	Analog function	ANI0/AVREFP/EI20	Port 2. 4-bit I/O port. Input or output can be specified in 1-bit units. P20 to P23 can be set for the analog pin functions**Note**.
P21				ANI1/AVREFM/EI21	
P22	4-35-1			ANI2/ANO0/EI22	
P23	4-37-1			ANI3/ANO1/IVREF0/EI23	
P30	7-31-2	I/O	Input port	VCOUT0/TSCAP/EI30/INTP3/SCL11	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P31				TS01/EI31/TI03/TO03/INTP4/(PCLBUZ0)	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P50	7-39-1	I/O	Input port	TS00/EI50/EO50/CCD03/INTP1/SDA11	Port 5. 1-bit I/O port. N-ch open-drain output (withstand voltage of VDD) can be set for P50 as an output. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P50 can be set as a controlled current drive port pin.

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P70	7-31-2	I/O	Input port	TS02/RIN0/KR0/SCK21/SCL21	Port 7. 6-bit I/O port. A TTL input buffer can be set for P71 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P71 and P72 as outputs. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P71	8-31-2			TS03/KR1/SI21/SDA21/RxDA0	
P72	7-31-3			TS04/KR2/SO21/TxDA0	
P73	7-31-2			TS05/KR3	
P76				TS08/KR6/INTP10/(RxD2)	
P77				TS09/KR7/INTP11/(TxD2)	
P121	7-2-1	I/O	Input port	X1/VBAT/EI121	Port 12. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P122				X2/EXCLK/EI122	
P137	2-1-3	Input	Input port	EI137/INTP0	Port 13. 1-bit input-only port.
P147	7-9-5	I/O	Analog function	ANI18/IVCMP0/EI147	Port 14. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P147 can be set for the analog pin function[Note].
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to VDD, either directly or via a resistor.

Note Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register (PMCAxx).

Remark Pin functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

2.1.13 R7F104G product with 44 pins

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P00	7-31-3	I/O	Input port	TS26/EI00/TI00	Port 0. 6-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P01 and P03 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P00, P02, and P03 as outputs.
P01	8-31-1			TS27/EI01/EO01/TO00	
P02	7-33-4		Analog function Note 1	TS28/TxD1	
P03	8-33-3			TS29/RxD1	
P05	7-31-2		Input port	TS10/TI05/TO05	
P06				TS11/TI06/TO06/CLKA0	
P10	8-1-11	I/O	Input port	EI10/EO10/SCK00/SCL00/(TI07)/(TO07)	Port 1. 6-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, P13, P16, and P17 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P10 to P13 and P17 as outputs. P16 and P17 can be set as controlled current drive port pins.
P11				EI11/EO11/SI00/RxD0/TOOLRxD/SDA00/(TI06)/(TO06)	
P12	7-1-12			EI12/EO12/SO00/TxD0/TOOLTxD/(INTP5)/(TI05)/(TO05)	
P13	8-6-9		Analog function Note 1	EO13/(TI04)/(TO04)	
P16	8-38-1		Input port	EO16/CCD00/TI01/TO01/INTP5/(RxD0)	
P17	8-38-2			EO17/CCD01/TI02/TO02/(TxD0)	
P20	4-3-5	I/O	Analog function	ANI0/AVREFP/EI20	Port 2. 8-bit I/O port. Input or output can be specified in 1-bit units. P20 to P27 can be set for the analog pin functions Note 2.
P21				ANI1/AVREFM/EI21	
P22	4-35-1			ANI2/ANO0/TS20/EI22	
P23	4-37-1			ANI3/ANO1/IVREF0/TS21/EI23	
P24	4-33-1			ANI4/TS22	
P25				ANI5/TS23	
P26				ANI6/TS24	
P27				ANI7/TS25	
P30	7-31-2	I/O	Input port	VCOUT0/TSCAP/EI30/INTP3/RTC1HZ/SCL11	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P31				TS01/EI31/TI03/TO03/INTP4/(PCLBUZ0)	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P50	7-39-1	I/O	Input port	TS00/EI50/EO50/CCD03/ INTP1/SDA11	Port 5. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. N-ch open-drain output (withstand voltage of VDD) can be set for P50 as an output. P50 can be set as a controlled current drive port pin.
P64	7-31-2	I/O	Input port	TS12/TI10/TO10	Port 6. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P65				TS13/TI11/TO11	
P70	7-31-2	I/O	Input port	TS02/RIN0/KR0/SCK21/ SCL21	Port 7. 6-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P71 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P71 and P72 as outputs.
P71	8-31-2			TS03/KR1/SI21/SDA21/ RxDA0	
P72	7-31-3			TS04/KR2/SO21/TxDA0	
P73	7-31-2			TS05/KR3	
P76				TS08/KR6/INTP10/(RxD2)	
P77				TS09/KR7/INTP11/(TxD2)	
P121	7-2-1	I/O	Input port	X1/VBAT/EI121	Port 12. 2-bit I/O port and 2-bit input-only port. For P121 and P122, input or output can be specified in 1-bit units. For P121 and P122, use of an on-chip pull-up resistor can be specified by a software setting at input port.
P122				X2/EXCLK/EI122	
P123	2-2-1	Input		XT1	
P124				XT2/EXCLKS	
P137	2-1-3	Input	Input port	EI137/INTP0	Port 13. 1-bit input-only port.
P147	7-9-5	I/O	Analog function	IVCMP0/EI147	Port 14. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P147 can be set for the analog pin function**Note 2**.
P150	4-33-1	I/O	Analog function**Note 1**	TS30	Port 15. 2-bit I/O port. Input or output can be specified in 1-bit units.
P153				TS33	
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to VDD, either directly or via a resistor.

Note 1. After release from the reset state, set the corresponding bit or bits in port mode control A register (PMCAxx) to 0.

Note 2. Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register (PMCAxx).

Remark Pin functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

2.1.14 R7F104G product with 48 pins

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P00	7-31-3	I/O	Input port	TS26/EI00/TI00	Port 0. 6-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P01 and P03 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P00, P02, and P03 as outputs. P02 and P03 can be set for the analog pin functions **Note 1**.
P01	8-31-1			TS27/EI01/EO01/TO00	
P02	7-33-4		Analog function	ANI17/TS28/TxD1	
P03	8-33-3			ANI16/TS29/RxD1	
P05	7-31-2		Input port	TS10/TI05/TO05	
P06				TS11/TI06/TO06/CLKA0	
P10	8-1-11	I/O	Input port	EI10/EO10/SCK00/SCL00/(TI07)/(TO07)	Port 1. 7-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. TTL input buffers can be set for P10, P11, P13, P14, P16, and P17 as inputs. N-ch open-drain output (withstand voltage of VDD) can be set for P10 to P14 and P17 as outputs. P16 and P17 can be set as controlled current drive port pins.
P11				EI11/EO11/SI00/RxD0/TOOLRxD/SDA00/(TI06)/(TO06)	
P12	7-1-12			EI12/EO12/SO00/TxD0/TOOLTxD/(INTP5)/(TI05)/(TO05)	
P13	8-6-9		Analog function **Note 2**	EO13/TxD2/(SDAA0)/(TI04)/(TO04)	
P14	8-1-11		Input port	EO14/RxD2/(SCLA0)/(TI03)/(TO03)	
P16	8-38-1			EO16/CCD00/TI01/TO01/INTP5/(RxD0)	
P17	8-38-2			EO17/CCD01/TI02/TO02/(TxD0)	
P20	4-3-5	I/O	Analog function	ANI0/AVREFP/EI20	Port 2. 8-bit I/O port. Input or output can be specified in 1-bit units. P20 to P27 can be set for the analog pin functions **Note 1**.
P21				ANI1/AVREFM/EI21	
P22	4-35-1			ANI2/ANO0/TS20/EI22	
P23	4-37-1			ANI3/ANO1/IVREF0/TS21/EI23	
P24	4-33-1			ANI4/TS22	
P25				ANI5/TS23	
P26				ANI6/TS24	
P27				ANI7/TS25	
P30	7-31-2	I/O	Input port	VCOUT0/TSCAP/EI30/INTP3/RTC1HZ/SCL11	Port 3. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P31				TS01/EI31/TI03/TO03/INTP4/(PCLBUZ0)	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 2-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P41 as an input.
P41	8-1-3			TI07/TO07	

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Multiplexed Function	Function
P50	7-39-1	I/O	Input port	TS00/EI50/EO50/CCD03/ INTP1/SDA11	Port 5. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. N-ch open-drain output (withstand voltage of VDD) can be set for P50 as an output. P50 can be set as a controlled current drive port pin.
P64	7-31-2	I/O	Input port	TS12/TI10/TO10	Port 6. 3-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P65				TS13/TI11/TO11	
P66				TS14/TI12/TO12	
P70	7-31-2	I/O	Input port	TS02/RIN0/KR0/SCK21/ SCL21	Port 7. 6-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. A TTL input buffer can be set for P71 as an input. N-ch open-drain output (withstand voltage of VDD) can be set for P71 and P72 as outputs.
P71	8-31-2			TS03/KR1/SI21/SDA21/ RxDA0	
P72	7-31-3			TS04/KR2/SO21/TxDA0	
P73	7-31-2			TS05/KR3	
P76				TS08/KR6/INTP10/(RxD2)	
P77				TS09/KR7/INTP11/(TxD2)	
P121	7-2-1	I/O	Input port	X1/VBAT/EI121	Port 12. 2-bit I/O port and 2-bit input-only port. For P121 and P122, input or output can be specified in 1-bit units. For P121 and P122, use of an on-chip pull-up resistor can be specified by a software setting at input port.
P122				X2/EXCLK/EI122	
P123	2-2-1	Input		XT1	
P124				XT2/EXCLKS	
P130	1-1-1	Output	Output port	—	Port 13. 1-bit output-only port and 1-bit input-only port.
P137	2-1-3	Input	Input port	EI137/INTP0	
P147	7-9-5	I/O	Analog function	IVCMP0/EI147	Port 14. 1-bit I/O port. Input or output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. P147 can be set for the analog pin function[Note 1].
P150	4-33-1	I/O	Analog function[Note 2]	TS30	Port 15. 2-bit I/O port. Input or output can be specified in 1-bit units.
P153				TS33	
$\overline{\text{RESET}}$	2-1-1	Input	—	—	Input-only pin for the external reset signal. When an external reset signal is not in use, connect this pin to VDD, either directly or via a resistor.

Note 1. Digital or analog can be selected per pin (in 1-bit units) with the port mode control A register (PMCAxx).

Note 2. After release from the reset state, set the corresponding bit or bits in port mode control A register (PMCAxx) to 0.

Remark Pin functions in parentheses in the above table can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

2.2 Non-GPIO Pin Functions

2.2.1 Non-GPIO pin functions of the R7F100G products per number of pins

(1/7)

Function Name	128-pin	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin	40-pin	36-pin	32-pin	30-pin
ANI0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ANI1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ANI2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ANI3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ANI4	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
ANI5	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
ANI6	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—
ANI7	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
ANI8	✓	✓	✓	—	—	—	—	—	—	—	—
ANI9	✓	✓	✓	—	—	—	—	—	—	—	—
ANI10	✓	✓	✓	—	—	—	—	—	—	—	—
ANI11	✓	✓	✓	—	—	—	—	—	—	—	—
ANI12	✓	✓	—	—	—	—	—	—	—	—	—
ANI13	✓	✓	—	—	—	—	—	—	—	—	—
ANI14	✓	✓	—	—	—	—	—	—	—	—	—
ANI16	✓	✓	✓	✓	✓	—	—	—	—	✓	✓
ANI17	✓	✓	✓	✓	✓	—	—	—	—	✓	✓
ANI18	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ANI19	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ANI20	✓	✓	✓	—	—	—	—	—	—	—	—
ANI21	✓	—	—	—	—	—	—	—	—	—	—
ANI22	✓	—	—	—	—	—	—	—	—	—	—
ANI23	✓	—	—	—	—	—	—	—	—	—	—
ANI24	✓	—	—	—	—	—	—	—	—	—	—
ANI25	✓	—	—	—	—	—	—	—	—	—	—
ANI26	✓	—	—	—	—	—	—	—	—	—	—
ANO0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
ANO1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
IVCMP0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
IVCMP1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
IVREF0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
IVREF1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
VCOUT0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
VCOUT1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TS00	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TS01	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

(2/7)

Function Name	128-pin	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin	40-pin	36-pin	32-pin	30-pin
TS02	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
TS03	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
TS04	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
TS05	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—
TS06	✓	✓	✓	✓	✓	✓	—	—	—	—	—
TS07	✓	✓	✓	✓	✓	✓	—	—	—	—	—
TS08	✓	✓	✓	✓	✓	—	—	—	—	—	—
TS09	✓	✓	✓	✓	✓	—	—	—	—	—	—
TS10	✓	✓	✓	✓	—	—	—	—	—	—	—
TS11	✓	✓	✓	✓	—	—	—	—	—	—	—
TS12	✓	✓	✓	—	—	—	—	—	—	—	—
TS13	✓	✓	✓	—	—	—	—	—	—	—	—
TS14	✓	✓	✓	—	—	—	—	—	—	—	—
TS15	✓	✓	✓	—	—	—	—	—	—	—	—
TS20	✓	✓	✓	✓Note	✓Note	✓Note	✓Note	✓Note	✓Note	✓Note	✓Note
TS21	✓	✓	✓	✓Note	✓Note	✓Note	✓Note	✓Note	✓Note	✓Note	✓Note
TS22	✓	✓	✓	✓Note	✓Note	✓Note	✓Note	✓Note	✓Note	—	—
TS23	✓	✓	✓	✓Note	✓Note	✓Note	✓Note	✓Note	✓Note	—	—
TS24	✓	✓	✓	✓Note	✓Note	✓Note	✓Note	✓Note	—	—	—
TS25	✓	✓	✓	✓Note	✓Note	✓Note	✓Note	—	—	—	—
TS26	✓	✓	✓	✓Note	✓Note	✓Note	✓Note	✓Note	✓Note	✓Note	✓Note
TS27	✓	✓	✓	✓Note	✓Note	✓Note	✓Note	✓Note	✓Note	✓Note	✓Note
TS28	✓	✓	✓	✓Note	✓Note	—	—	—	—	—	—
TS29	✓	✓	✓	✓Note	✓Note	—	—	—	—	—	—
TS30	✓	✓	✓	—	—	—	—	—	—	—	—
TS31	✓	✓	✓	—	—	—	—	—	—	—	—
TS32	✓	✓	✓	—	—	—	—	—	—	—	—
TS33	✓	✓	✓	—	—	—	—	—	—	—	—
TS34	✓	✓	—	—	—	—	—	—	—	—	—
TS35	✓	✓	—	—	—	—	—	—	—	—	—
TSCAP	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CCD00	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CCD01	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CCD02	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CCD03	✓	✓	✓	✓	✓	✓	—	✓	✓	✓	✓
CCD04	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CCD05	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CCD06	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
CCD07	✓	✓	✓	✓	✓	✓	✓	—	—	—	—

(3/7)

Function Name	128-pin	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin	40-pin	36-pin	32-pin	30-pin
INTP0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTP1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTP2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTP3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTP4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTP5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTP6	✓	✓	✓	✓	✓	✓	—	—	—	—	—
INTP7	✓	✓	✓	✓	—	—	—	—	—	—	—
INTP8	✓	✓	✓	✓	✓	✓	—	—	—	—	—
INTP9	✓	✓	✓	✓	✓	✓	—	—	—	—	—
INTP10	✓	✓	✓	✓	✓	—	—	—	—	—	—
INTP11	✓	✓	✓	✓	✓	—	—	—	—	—	—
KR0	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—
KR1	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—
KR2	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—
KR3	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—
KR4	✓	✓	✓	✓	✓	✓	—	—	—	—	—
KR5	✓	✓	✓	✓	✓	✓	—	—	—	—	—
KR6	✓	✓	✓	✓	✓	—	—	—	—	—	—
KR7	✓	✓	✓	✓	✓	—	—	—	—	—	—
PCLBUZ0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PCLBUZ1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
REGC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
RTC1HZ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
$\overline{\text{RESET}}$	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
RIN0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
RxD0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
RxD1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
RxD2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
RxD3	✓	✓	✓	—	—	—	—	—	—	—	—
TxD0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TxD1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TxD2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TxD3	✓	✓	✓	—	—	—	—	—	—	—	—
RxDA0	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
RxDA1	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
TxDA0	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
TxDA1	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
CLKA0	✓	✓	✓	✓	—	—	—	—	—	—	—
CLKA1	✓	✓	✓	✓	—	—	—	—	—	—	—

(4/7)

Function Name	128-pin	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin	40-pin	36-pin	32-pin	30-pin
SCK00	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SCK01	✓	✓	✓	✓	✓	✓	—	—	—	—	—
SCK10	✓	✓	✓	✓	—	—	—	—	—	—	—
SCK11	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SCK20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SCK21	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
SCK30	✓	✓	✓	—	—	—	—	—	—	—	—
SCK31	✓	✓	✓	—	—	—	—	—	—	—	—
SCL00	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SCL01	✓	✓	✓	✓	✓	✓	—	—	—	—	—
SCL10	✓	✓	✓	✓	—	—	—	—	—	—	—
SCL11	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SCL20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SCL21	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
SCL30	✓	✓	✓	—	—	—	—	—	—	—	—
SCL31	✓	✓	✓	—	—	—	—	—	—	—	—
SDA00	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SDA01	✓	✓	✓	✓	✓	✓	—	—	—	—	—
SDA10	✓	✓	✓	✓	—	—	—	—	—	—	—
SDA11	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SDA20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SDA21	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
SDA30	✓	✓	✓	—	—	—	—	—	—	—	—
SDA31	✓	✓	✓	—	—	—	—	—	—	—	—
SI00	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SI01	✓	✓	✓	✓	✓	✓	—	—	—	—	—
SI10	✓	✓	✓	✓	—	—	—	—	—	—	—
SI11	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SI20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SI21	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
SI30	✓	✓	✓	—	—	—	—	—	—	—	—
SI31	✓	✓	✓	—	—	—	—	—	—	—	—
SO00	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SO01	✓	✓	✓	✓	✓	✓	—	—	—	—	—
SO10	✓	✓	✓	✓	—	—	—	—	—	—	—
SO11	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SO20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SO21	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
SO30	✓	✓	✓	—	—	—	—	—	—	—	—
SO31	✓	✓	✓	—	—	—	—	—	—	—	—

(5/7)

Function Name	128-pin	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin	40-pin	36-pin	32-pin	30-pin
SCLA0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SCLA1	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
SDAA0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
SDAA1	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
TI00	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TI01	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TI02	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TI03	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TI04	✓	✓	✓	✓	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)
TI05	✓	✓	✓	✓	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)
TI06	✓	✓	✓	✓	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)
TI07	✓	✓	✓	✓	✓	✓	✓	(✓)	(✓)	(✓)	(✓)
TI10	✓	✓	✓	—	—	—	—	—	—	—	—
TI11	✓	✓	✓	—	—	—	—	—	—	—	—
TI12	✓	✓	✓	—	—	—	—	—	—	—	—
TI13	✓	✓	✓	—	—	—	—	—	—	—	—
TI14	✓	—	—	—	—	—	—	—	—	—	—
TI15	✓	—	—	—	—	—	—	—	—	—	—
TI16	✓	—	—	—	—	—	—	—	—	—	—
TI17	✓	—	—	—	—	—	—	—	—	—	—
TO00	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TO01	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TO02	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TO03	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TO04	✓	✓	✓	✓	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)
TO05	✓	✓	✓	✓	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)
TO06	✓	✓	✓	✓	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)	(✓)
TO07	✓	✓	✓	✓	✓	✓	✓	(✓)	(✓)	(✓)	(✓)
TO10	✓	✓	✓	—	—	—	—	—	—	—	—
TO11	✓	✓	✓	—	—	—	—	—	—	—	—
TO12	✓	✓	✓	—	—	—	—	—	—	—	—
TO13	✓	✓	✓	—	—	—	—	—	—	—	—
TO14	✓	—	—	—	—	—	—	—	—	—	—
TO15	✓	—	—	—	—	—	—	—	—	—	—
TO16	✓	—	—	—	—	—	—	—	—	—	—
TO17	✓	—	—	—	—	—	—	—	—	—	—
EI00	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EI01	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EI10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EI11	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

(6/7)

Function Name	128-pin	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin	40-pin	36-pin	32-pin	30-pin
EI12	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EI20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EI21	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EI22	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EI23	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EI30	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EI31	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EI50	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EI51	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EI120	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EI121	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EI122	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EI137	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EI147	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EO01	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EO10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EO11	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EO12	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EO13	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EO14	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EO15	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EO16	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EO17	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EO50	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EO51	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EO60	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EO61	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
X1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
X2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EXCLK	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
XT1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
XT2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EXCLKS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
V_{DD}	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EV_{DD0}	✓	✓	✓	✓	—	—	—	—	—	—	—
EV_{DD1}	✓	✓	—	—	—	—	—	—	—	—	—
VBAT	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—
AV_{REFP}	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
AV_{REFM}	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
V_{SS}	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

(7/7)

Function Name	128-pin	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin	40-pin	36-pin	32-pin	30-pin
EV_{SS0}	✓	✓	✓	✓	—	—	—	—	—	—	—
EV_{SS1}	✓	✓	—	—	—	—	—	—	—	—	—
TOOLRxD	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TOOLTxD	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
TOOL0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Note Not present in products with 128 or fewer Kbytes of code flash memory.

Remark Parentheses indicate that the pin can be assigned via settings in the peripheral I/O redirection register (PIOR).

2.2.2 Non-GPIO pin functions of the R7F104G products per number of pins

(1/4)

Function Name	48-pin	44-pin	32-pin
ANI0	✓	✓	✓
ANI1	✓	✓	✓
ANI2	✓	✓	✓
ANI3	✓	✓	✓
ANI4	✓	✓	—
ANI5	✓	✓	—
ANI6	✓	✓	—
ANI7	✓	✓	—
ANI8	—	—	—
ANI9	—	—	—
ANI10	—	—	—
ANI11	—	—	—
ANI12	—	—	—
ANI13	—	—	—
ANI14	—	—	—
ANI16	✓	—	✓
ANI17	✓	—	✓
ANI18	—	—	✓
ANI19	—	—	—
ANI20	—	—	—
ANI21	—	—	—
ANI22	—	—	—
ANI23	—	—	—
ANI24	—	—	—
ANI25	—	—	—
ANI26	—	—	—
ANO0	✓	✓	✓
ANO1	✓	✓	✓
IVCMP0	✓	✓	✓
IVCMP1	—	—	—
IVREF0	✓	✓	✓
IVREF1	—	—	—
VCOUT0	✓	✓	✓
VCOUT1	—	—	—
TS00	✓	✓	✓
TS01	✓	✓	✓
TS02	✓	✓	✓
TS03	✓	✓	✓

Function Name	48-pin	44-pin	32-pin
TS04	✓	✓	✓
TS05	✓	✓	✓
TS06	—	—	—
TS07	—	—	—
TS08	✓	✓	✓
TS09	✓	✓	✓
TS10	✓	✓	✓
TS11	✓	✓	✓
TS12	✓	✓	—
TS13	✓	✓	—
TS14	✓	—	—
TS15	—	—	—
TS20	✓	✓	—
TS21	✓	✓	—
TS22	✓	✓	—
TS23	✓	✓	—
TS24	✓	✓	—
TS25	✓	✓	—
TS26	✓	✓	—
TS27	✓	✓	—
TS28	✓	✓	—
TS29	✓	✓	—
TS30	✓	✓	—
TS31	—	—	—
TS32	—	—	—
TS33	✓	✓	—
TS34	—	—	—
TS35	—	—	—
TSCAP	✓	✓	✓
CCD00	✓	✓	✓
CCD01	✓	✓	✓
CCD02	—	—	—
CCD03	✓	✓	✓
CCD04	—	—	—
CCD05	—	—	—
CCD06	—	—	—
CCD07	—	—	—
INTP0	✓	✓	✓

(2/4)

Function Name	48-pin	44-pin	32-pin
INTP1	✓	✓	✓
INTP2	—	—	—
INTP3	✓	✓	✓
INTP4	✓	✓	✓
INTP5	✓	✓	✓
INTP6	—	—	—
INTP7	—	—	—
INTP8	—	—	—
INTP9	—	—	—
INTP10	✓	✓	✓
INTP11	✓	✓	✓
KR0	✓	✓	✓
KR1	✓	✓	✓
KR2	✓	✓	✓
KR3	✓	✓	✓
KR4	—	—	—
KR5	—	—	—
KR6	✓	✓	✓
KR7	✓	✓	✓
PCLBUZ0	(✓)	(✓)	(✓)
PCLBUZ1	—	—	—
REGC	✓	✓	✓
RTC1HZ	✓	✓	—
$\overline{\text{RESET}}$	✓	✓	✓
RIN0	✓	✓	✓
RxD0	✓	✓	✓
RxD1	✓	✓	✓
RxD2	✓	(✓)	(✓)
RxD3	—	—	—
TxD0	✓	✓	✓
TxD1	✓	✓	✓
TxD2	✓	(✓)	(✓)
TxD3	—	—	—
RxDA0	✓	✓	✓
RxDA1	—	—	—
TxDA0	✓	✓	✓
TxDA1	—	—	—
CLKA0	✓	✓	✓
CLKA1	—	—	—
SCK00	✓	✓	✓

Function Name	48-pin	44-pin	32-pin
SCK01	—	—	—
SCK10	—	—	—
SCK11	—	—	—
SCK20	—	—	—
SCK21	✓	✓	✓
SCK30	—	—	—
SCK31	—	—	—
SCL00	✓	✓	✓
SCL01	—	—	—
SCL10	—	—	—
SCL11	✓	✓	✓
SCL20	—	—	—
SCL21	✓	✓	✓
SCL30	—	—	—
SCL31	—	—	—
SDA00	✓	✓	✓
SDA01	—	—	—
SDA10	—	—	—
SDA11	✓	✓	✓
SDA20	—	—	—
SDA21	✓	✓	✓
SDA30	—	—	—
SDA31	—	—	—
SI00	✓	✓	✓
SI01	—	—	—
SI10	—	—	—
SI11	—	—	—
SI20	—	—	—
SI21	✓	✓	✓
SI30	—	—	—
SI31	—	—	—
SO00	✓	✓	✓
SO01	—	—	—
SO10	—	—	—
SO11	—	—	—
SO20	—	—	—
SO21	✓	✓	✓
SO30	—	—	—
SO31	—	—	—
SCLA0	(✓)	—	—

(3/4)

Function Name	48-pin	44-pin	32-pin
SCLA1	—	—	—
SDAA0	(✓)	—	—
SDAA1	—	—	—
TI00	✓	✓	✓
TI01	✓	✓	✓
TI02	✓	✓	✓
TI03	✓	✓	✓
TI04	(✓)	(✓)	—
TI05	✓	✓	✓
TI06	✓	✓	✓
TI07	✓	(✓)	(✓)
TI10	✓	✓	—
TI11	✓	✓	—
TI12	✓	—	—
TI13	—	—	—
TI14	—	—	—
TI15	—	—	—
TI16	—	—	—
TI17	—	—	—
TO00	✓	✓	—
TO01	✓	✓	✓
TO02	✓	✓	✓
TO03	✓	✓	✓
TO04	(✓)	(✓)	—
TO05	✓	✓	✓
TO06	✓	✓	✓
TO07	✓	(✓)	(✓)
TO10	✓	✓	—
TO11	✓	✓	—
TO12	✓	—	—
TO13	—	—	—
TO14	—	—	—
TO15	—	—	—
TO16	—	—	—
TO17	—	—	—
EI00	✓	✓	✓
EI01	✓	✓	—
EI10	✓	✓	✓
EI11	✓	✓	✓
EI12	✓	✓	✓

Function Name	48-pin	44-pin	32-pin
EI20	✓	✓	✓
EI21	✓	✓	✓
EI22	✓	✓	✓
EI23	✓	✓	✓
EI30	✓	✓	✓
EI31	✓	✓	✓
EI50	✓	✓	✓
EI51	—	—	—
EI120	—	—	—
EI121	✓	✓	✓
EI122	✓	✓	✓
EI137	✓	✓	✓
EI147	✓	✓	✓
EO01	✓	✓	—
EO10	✓	✓	✓
EO11	✓	✓	✓
EO12	✓	✓	✓
EO13	✓	✓	—
EO14	✓	—	—
EO15	—	—	—
EO16	✓	✓	✓
EO17	✓	✓	✓
EO50	✓	✓	✓
EO51	—	—	—
EO60	—	—	—
EO61	—	—	—
X1	✓	✓	✓
X2	✓	✓	✓
EXCLK	✓	✓	✓
XT1	✓	✓	—
XT2	✓	✓	—
EXCLKS	✓	✓	—
V_{DD}	✓	✓	✓
EV_{DD0}	—	—	—
EV_{DD1}	—	—	—
VBAT	✓	✓	✓
AV_{REFP}	✓	✓	✓
AV_{REFM}	✓	✓	✓
V_{SS}	✓	✓	✓
EV_{SS0}	—	—	—

(4/4)

Function Name	48-pin	44-pin	32-pin
EV_{SS1}	—	—	—
TOOLRxD	✓	✓	✓

Function Name	48-pin	44-pin	32-pin
TOOLTxD	✓	✓	✓
TOOL0	✓	✓	✓

Remark Parentheses indicate that the pin can be assigned via settings in the peripheral I/O redirection register (PIOR).

2.2.3 Description of non-GPIO pin functions

(1/2)

Function Name	I/O	Function
ANI0 to ANI14, ANI16 to ANI26	Input	Analog voltage inputs for the A/D converter (see **Figure 12 - 52 Connections of VDD, AVREFP, and Analog Input Pins**)
ANO0, ANO1	Output	D/A converter outputs
IVCMP0, IVCMP1	Input	Analog voltage inputs for the comparator
IVREF0, IVREF1	Input	Reference voltage inputs for the comparator
VCOUT0, VCOUT1	Output	Comparator outputs
TS00 to TS15, TS20 to TS35	I/O	Electrostatic capacitance measurement pins (touch sensor)
TSCAP	—	Pin for connecting a power supply capacitor for use in measuring electrostatic capacitance. Connect this pin to Vss via a capacitor (10 nF). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.
CCD00 to CCD07	Output	Controlled current drive port pins
INTP0 to INTP11	Input	External interrupt request input pins for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.
KR0 to KR7	Input	Key interrupt inputs
PCLBUZ0, PCLBUZ1	Output	Clock outputs/buzzer outputs
REGC	—	Pin for connecting a regulator output stabilization capacitor for internal operation. Connect this 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.
RTC1HZ	Output	Realtime clock correction clock (1 Hz) output
$\overline{\text{RESET}}$	Input	This is the active-low system reset input pin. When an external reset signal is not used, connect this pin directly or via a resistor to VDD.
RIN0	Input	External pulse signal input pin for the remote control signal reception circuit
RxD0 to RxD3	Input	Serial data input pins for serial interfaces UART0 to UART3
TxD0 to TxD3	Output	Serial data output pins for serial interfaces UART0 to UART3
RxDA0, RxDA1	Input	Serial data input pins for serial interfaces UARTA0 and UARTA1
TxDA0, TxDA1	Output	Serial data output pins for serial interfaces UARTA0 and UARTA1
CLKA0, CLKA1	Output	Clock output pins for serial interfaces UARTA0 and UARTA1
SCK00, SCK01, SCK10, SCK11, SCK20, SCK21, SCK30, SCK31	I/O	Serial clock I/O pins for serial interfaces CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, and CSI31
SCL00, SCL01, SCL10, SCL11, SCL20, SCL21, SCL30, SCL31	Output	Serial clock output pins for serial interfaces IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, and IIC31
SDA00, SDA01, SDA10, SDA11, SDA20, SDA21, SDA30, SDA31	I/O	Serial data I/O pins for serial interfaces IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, and IIC31
SI00, SI01, SI10, SI11, SI20, SI21, SI30, SI31	Input	Serial data input pins for serial interfaces CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, and CSI31
SO00, SO01, SO10, SO11, SO20, SO21, SO30, SO31	Output	Serial data output pins for serial interfaces CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, and CSI31
SCLA0, SCLA1	I/O	Clock I/O pins for serial interfaces IICA0 and IICA1
SDAA0, SDAA1	I/O	Serial data I/O pins for serial interfaces IICA0 and IICA1
TI00 to TI07, TI10 to TI17	Input	Pins for input of an external counting clock or capture trigger to channels 0 to 7 of timer array units 0 and 1
TO00 to TO07, TO10 to TO17	Output	Timer output pins for channels 0 to 7 of timer array units 0 and 1

(2/2)

Function Name	I/O	Function
EI00, EI01, EI10 to EI12, EI20 to EI23, EI30, EI31, EI50, EI51, EI120 to EI122, EI137, EI147	Input	ELCL inputs
EO01, EO10 to EO17, EO50, EO51, EO60, EO61	Output	ELCL outputs
X1, X2	—	Resonator connection for the main system clock
EXCLK	Input	External clock input for the main system clock
XT1, XT2	—	Resonator connection for the subsystem clock
EXCLKS	Input	External clock input for the subsystem clock
V_{DD}	—	<30- to 52-pin R7F100G and 32- to 48-pin R7F104G products> Positive power supply for all pins <64- to 128-pin R7F100G products> Positive power supply for P20 to P27, P121 to P124, P137, P150 to P156, and pins in use for functions other than port functions
EV_{DD0}, EV_{DD1}	—	Positive power supply for port pins (other than P20 to P27, P121 to P124, P137, and P150 to P156)
VBAT	—	Power supply pin for battery backup
AV_{REFP}	Input	Positive reference voltage input for the A/D converter
AV_{REFM}	Input	Negative reference voltage input for the A/D converter
V_{SS}	—	<30- to 52-pin R7F100G and 32- to 48-pin R7F104G products> Ground voltage for all pins <64- to 128-pin R7F100G products> Ground voltage for P20 to P27, P121 to P124, P137, P150 to P156, and pins in use for functions other than port functions
EV_{SS0}, EV_{SS1}	—	Ground voltage for port pins (other than P20 to P27, P121 to P124, P137, and P150 to P156)
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 I/O for flash memory programmer or debugger

Caution The relationship between the voltage on P40/TOOL0 and the operating mode after release from the reset state is as follows.

P40/TOOL0	Operating Mode
EV_{DD}	**Normal operating mode**
0 V	**Flash memory programming mode**

For details, see 33.4 Programming Method. Pins EV_{DD0} and EV_{DD1} are not present in some products. In such cases, use the V_{DD} pins as the alternative.

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

2.2.4 VBAT pin

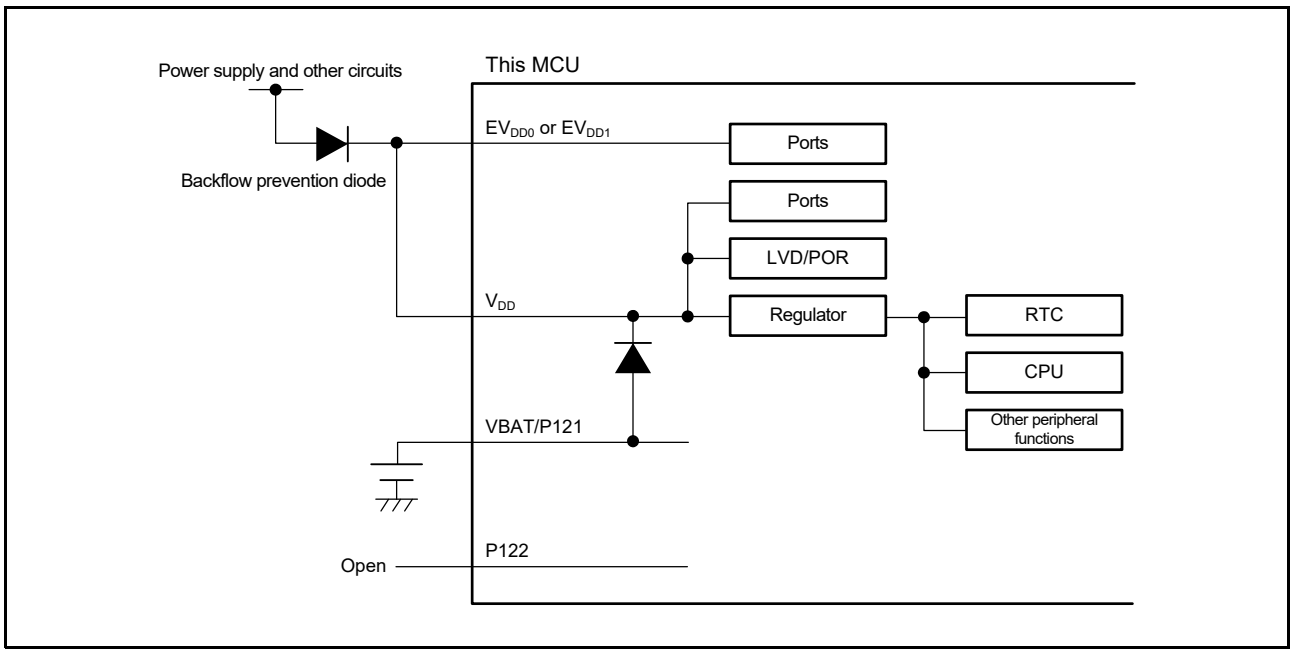
2.2.4.1 Function of the VBAT pin

The VBAT pin is used to connect the battery for use in backing up. Connecting the VBAT pin to a battery for use in backing up enables the supply of power from the VBAT pin when the power supply to the V_{DD} pin is shut off. The main purpose of the VBAT pin is to continue operation of the realtime clock (RTC).

2.2.4.2 Connecting the VBAT pin to the battery for use in backing up

Figure 2 - 1 shows an example of the connection of the VBAT pin. The VBAT pin supplies power to the V_{DD} pin through an internal diode. The internal diode for use with the VBAT pin is always connected to the V_{DD} pin. If preventing the backward flow of current, that is, current flowing from the VBAT pin to the power supply or other circuits that are connected to the V_{DD} pin through the diode, is required, externally connect a backflow prevention diode to the V_{DD}, EV_{DD0}, and EV_{DD1} pins. The allowed range of input voltage on the VBAT pin is 2.7 V to 5.5 V. The input voltage on the VBAT pin falling below 2.7 V while power is being supplied from the VBAT pin may lead to the generation of a POR reset due to the fall in the voltage across the diode. The maximum current that can be supplied through the VBAT pin is 150 μA.

Figure 2 - 1 Example of the Connection of the VBAT Pin

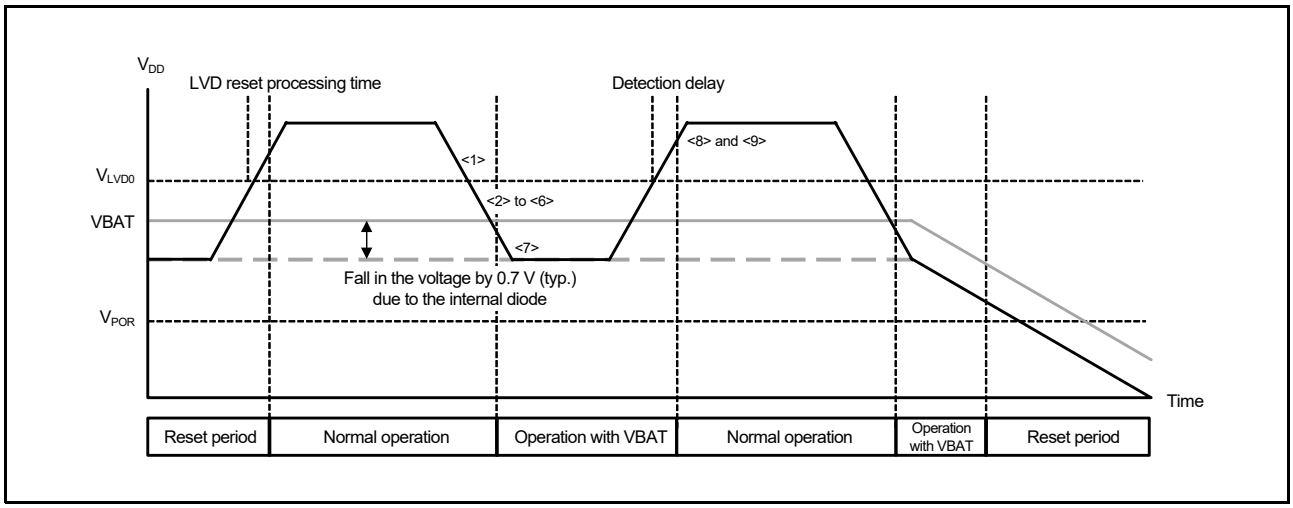


2.2.4.3 Using the VBAT pin

How to make the initial settings for the VBAT pin and an example of the procedure for switching the power supply pin to the VBAT pin are described below. This processing is to be completed before the voltage on the VDD pin falls below that supplied from the VBAT pin. In addition, **Figure 2 - 2** shows the state transitions in switching the power supply pin between the VDD and VBAT pins.

1. Making the initial settings for the VBAT pin
   Set the P121 pin to X1 oscillation mode (by setting the EXCLK and OSCSEL bits of the CMC register to 0 and 1, respectively, and the MSTOP bit of the CSC register to 1) in the initial settings.
2. Example of the procedure for switching the power supply pin to the VBAT pin
   - <1> Use LVD0 in the interrupt mode to generate an interrupt request when the power supply voltage (VDD) falls. The switching process starts in response to this interrupt request signal.
   - <2> Disable interrupts other than LVD0.
   - <3> Stop the operation of all peripheral functions other than the realtime clock (RTC).
   - <4> Change the output settings for port pins so that no current flows through them.
   - <5> Clear the interrupt request flag of LVD0.
   - <6> After having confirmed that the value of the LVD0F bit is 1 (VDD < detection voltage), if the CPU is operating with the main system clock, place this MCU in the STOP mode. If the CPU is operating with the subsystem clock, place the chip in the HALT mode.
   - <7> Keep the chip in the above state until LVD0 generates an interrupt request.
   - <8> Supplying the power supply voltage (VDD) back to the chip makes LVD0 generate an interrupt request, thereby releasing the chip from the STOP or HALT mode.
   - <9> After having confirmed that the value of the LVD0F bit is 0 (VDD ≥ detection voltage), make the settings to return the peripheral functions to normal operation while the power supply voltage (VDD) is being supplied.

Figure 2 - 2 State Transitions in Switching the Power Supply Pin between the VDD and VBAT Pins



When switching between the VDD and VBAT pins is too frequent, waiting for the fluctuations in voltage to settle between steps 4 and 5 in the procedure can prevent excessive switching between the VDD and VBAT pins.

Caution 1. Operation with the main system clock is prohibited while the battery is supplying power.

Caution 2. Make the setting to stop counting by the WDT when the battery is to supply power.
Moreover, input of the low level on the reset pin is prohibited. This is because attempting to do so leads to starting operation with the main system clock after release from the reset state, which requires current exceeding 150 µA, the maximum allowable current through the VBAT pin.

Caution 3. When the battery is to be used for the supply of power, set the P121 pin to the X1 oscillation mode; that is, do not set the P121 pin to the input or output mode.

2.3 Connection of Unused Pins

Table 2 - 2 shows the connections of unused pins.

Remark The pins mounted depend on the product. Refer to **1.4 Pin Configuration (Top View)** and **2.1 Functions of Port Pins**.

Table 2 - 2 Connections of Unused Pins

Pin Name	I/O	Recommended Connection of Unused Pins
P00 to P07	I/O	Input: Independently connect the pins to EV_{DD0}, EV_{DD1} or EV_{SS0}, EV_{SS1} via resistors. Output: Leave the pins open-circuit.
P10 to P17		
P20 to P27		Input: Independently connect the pins to V_{DD} or V_{SS} via resistors. Output: Leave the pins open-circuit.
P30 to P37		Input: Independently connect the pins to EV_{DD0}, EV_{DD1} or EV_{SS0}, EV_{SS1} via resistors. Output: Leave the pins open-circuit.
P40/TOOL0		Input: Independently connect the pin to EV_{DD0}, EV_{DD1} via a resistor or leave the pin open-circuit. Output: Leave the pin open-circuit.
P41 to P47		Input: Independently connect the pins to EV_{DD0}, EV_{DD1} or EV_{SS0}, EV_{SS1} via resistors. Output: Leave the pins open-circuit.
P50 to P57		
P60 to P63		Input: Independently connect the pins to EV_{DD0}, EV_{DD1} or EV_{SS0}, EV_{SS1} via resistors. Output: Set the output latch for the port to 0 and leave the pins open-circuit, or set it to 1 and independently connect the pins to EV_{DD0}, EV_{DD1} or EV_{SS0}, EV_{SS1} via resistors.
P64 to P67		Input: Independently connect the pins to EV_{DD0}, EV_{DD1} or EV_{SS0}, EV_{SS1} via resistors. Output: Leave the pins open-circuit.
P70 to P77		
P80 to P87		
P90 to P97		
P100 to P106		
P110 to P117		
P120		
P121, P122		Input: Independently connect the pins to V_{DD} or V_{SS} via resistors. Output: Leave the pins open-circuit.
P123, P124	Input	Set the EXCLKS and OSCSELS bits in the clock operation mode control register (CMC) to 0 and 1, respectively. Set the XTSTOP bit in the clock operation status control register (CSC) to 1, and leave the pins open-circuit[Note]. Alternatively, provide the pins with independent connections to V_{DD} or V_{SS} via resistors.
P125 to P127	I/O	Input: Independently connect the pins to EV_{DD0}, EV_{DD1} or EV_{SS0}, EV_{SS1} via resistors. Output: Leave the pins open-circuit.
P130	Output	Leave the pin open-circuit.
P137	Input	Set the PDIDIS137 bit in the port digital input disable register (PDIDIS13) to 1, and leave the pin open-circuit. Alternatively, provide the pin with an independent connection to V_{DD} or V_{SS} via a resistor.
P140 to P147	I/O	Input: Independently connect the pins to EV_{DD0}, EV_{DD1} or EV_{SS0}, EV_{SS1} via resistors. Output: Leave the pins open-circuit.
P150 to P156	I/O	Input: Independently connect the pins to V_{DD} or V_{SS} via resistors. Output: Leave the pins open-circuit.
$\overline{\text{RESET}}$	Input	Connect the pin directly or via a resistor to V_{DD}.
REGC	—	Connect the pin to V_{SS} via a capacitor (0.47 to 1 μF).

Note When the low-speed on-chip oscillator clock (f_{IL}) is selected for the CPU/peripheral hardware clock frequency (f_{CLK}), the current may increase approximately by 1 μA.

Remark For products that do not have an EV_{DD0}, EV_{DD1}, EV_{SS0}, or EV_{SS1} pin, read EV_{DD0} and EV_{DD1} as V_{DD}, and EV_{SS0} and EV_{SS1} as V_{SS}.

2.4 Block Diagrams of Pins

Figures 2 - 3 to **2 - 35** show the block diagrams of the pins described in **2.1.1 R7F100G products with 30 pins** to **2.1.14 R7F104G product with 48 pins**.

Figure 2 - 3 Pin Block Diagram for Pin Type 1-1-1

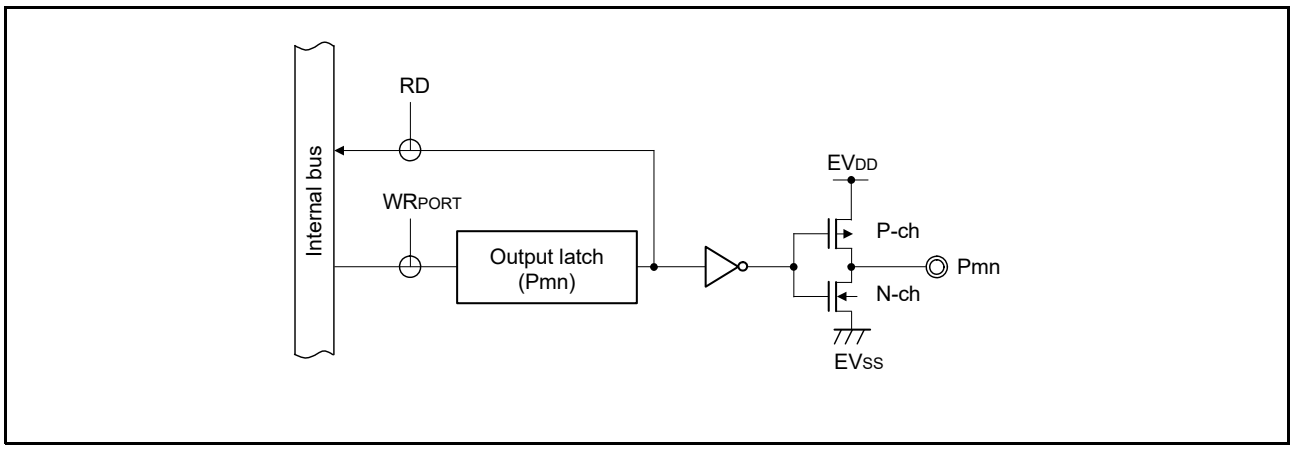


Figure 2 - 4 Pin Block Diagram for Pin Type 2-1-1

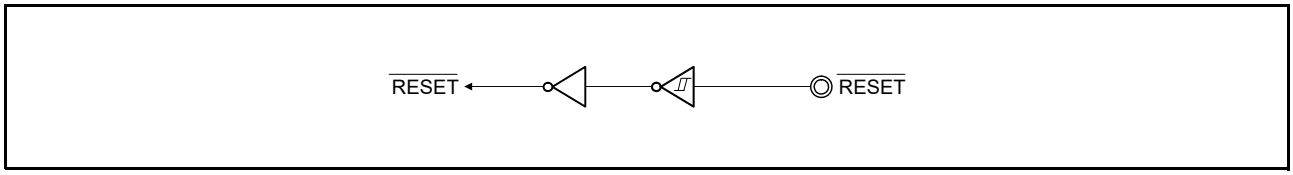
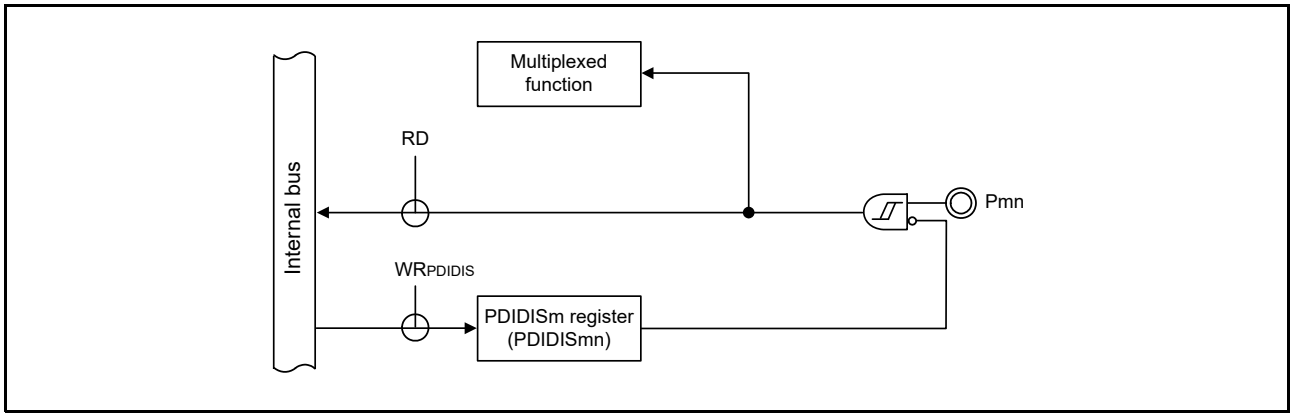
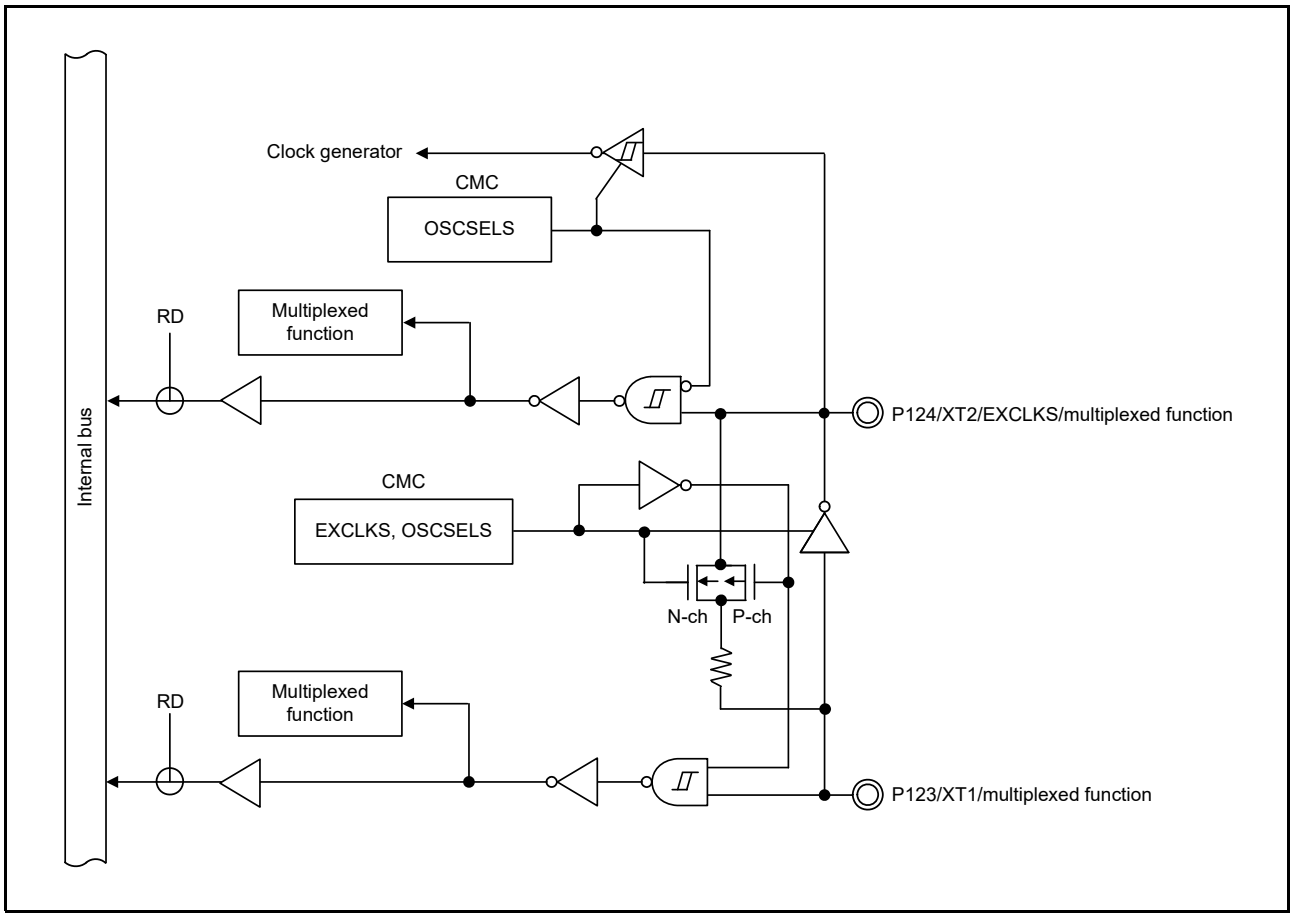


Figure 2 - 5 Pin Block Diagram for Pin Type 2-1-3



Remark For multiplexed functions, see **2.1 Functions of Port Pins**.

Figure 2 - 6 Pin Block Diagram for Pin Type 2-2-1



Remark For multiplexed functions, see **2.1 Functions of Port Pins**.

Figure 2 - 7 Pin Block Diagram for Pin Type 4-3-5

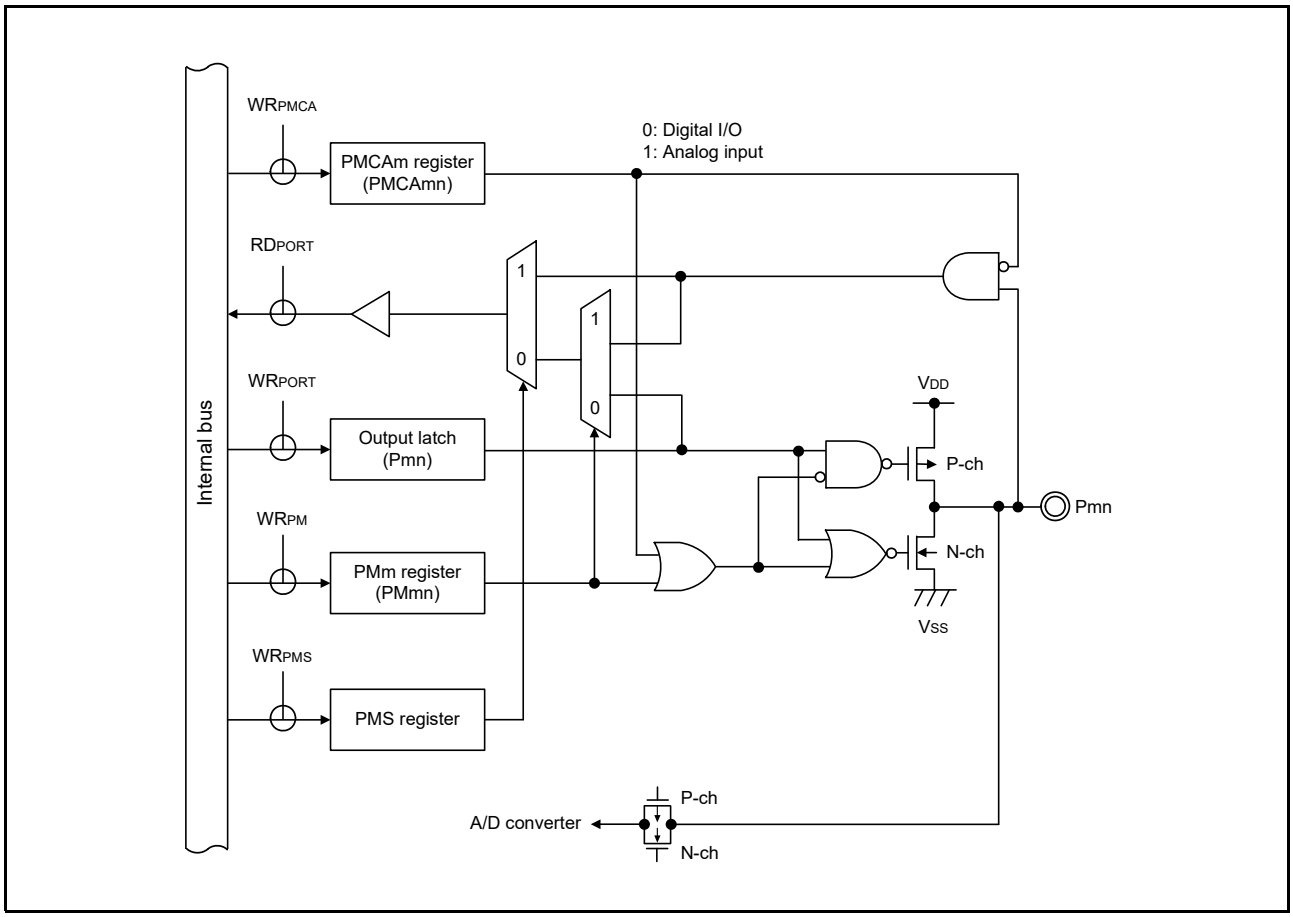


Figure 2 - 8 Pin Block Diagram for Pin Type 4-33-1

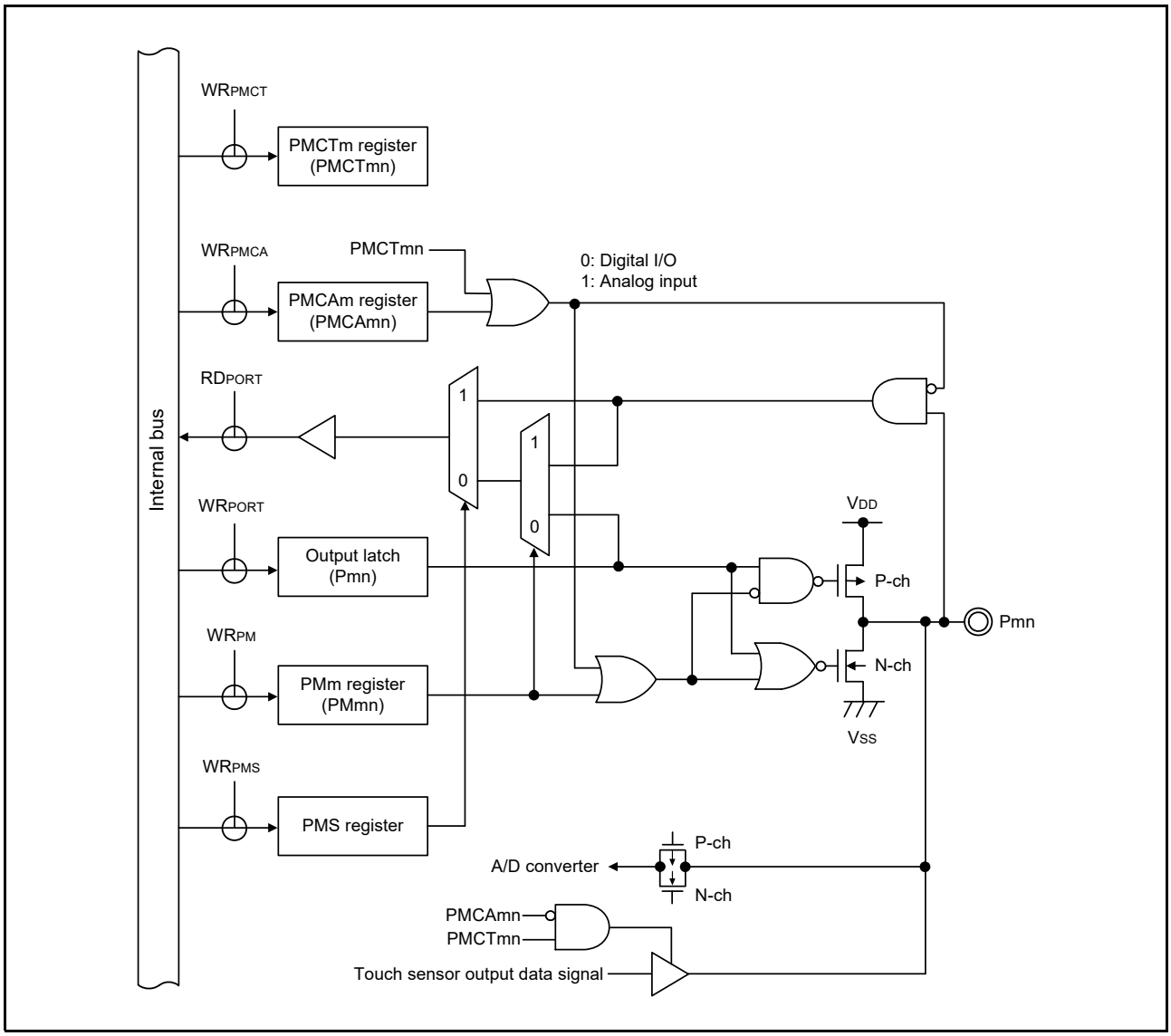


Figure 2 - 9 Pin Block Diagram for Pin Type 4-35-1

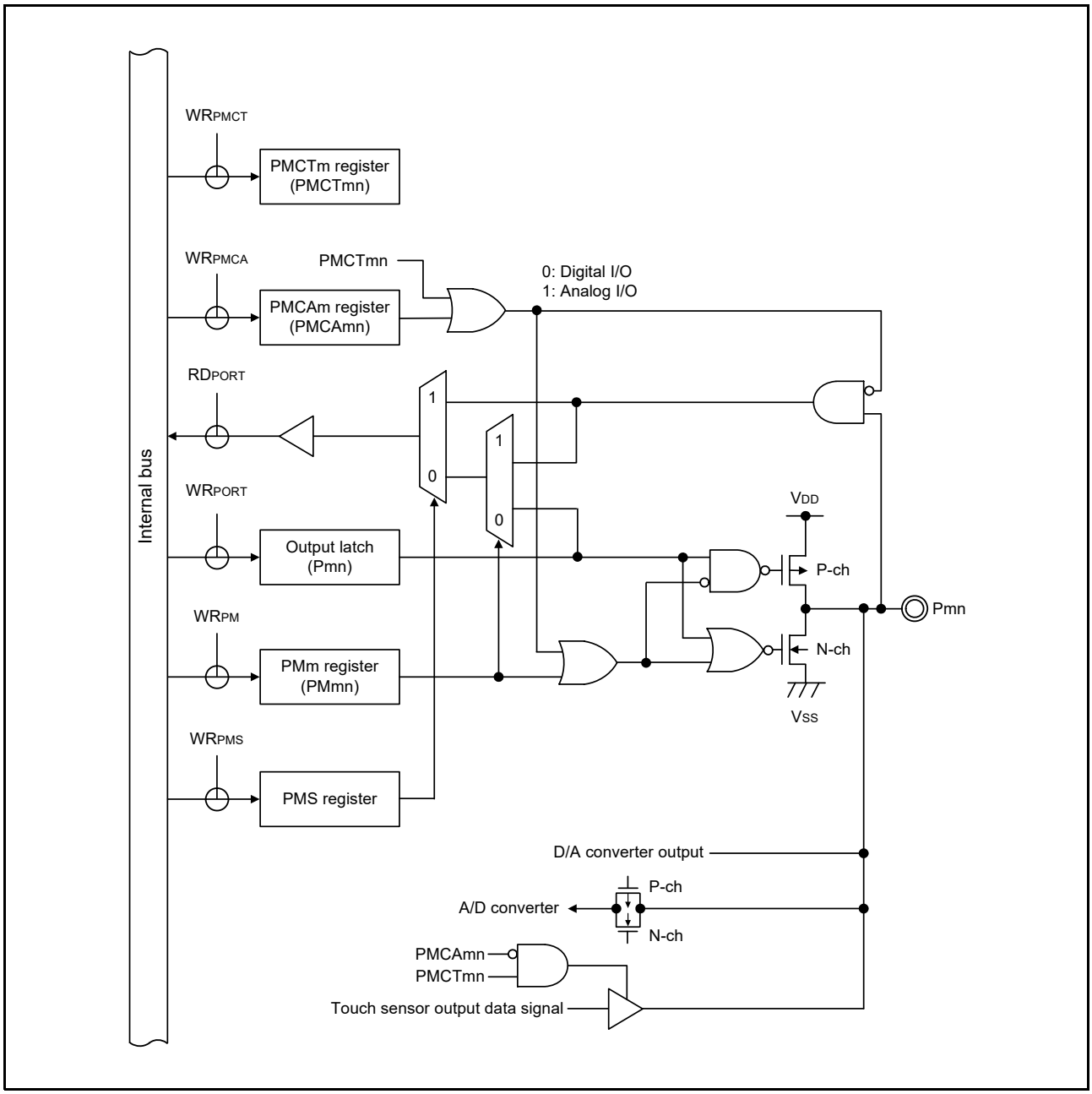


Figure 2 - 10 Pin Block Diagram for Pin Type 4-37-1

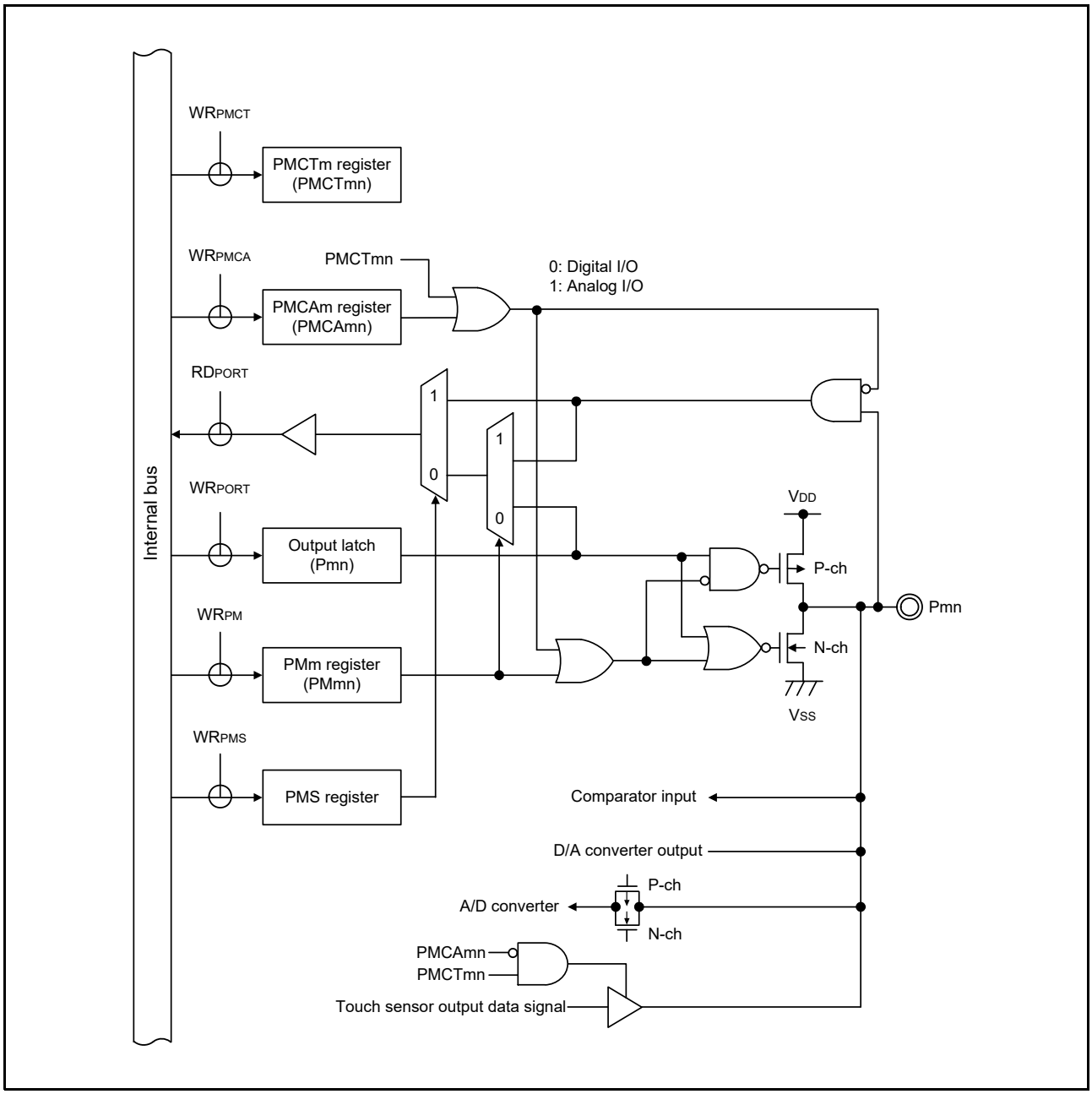
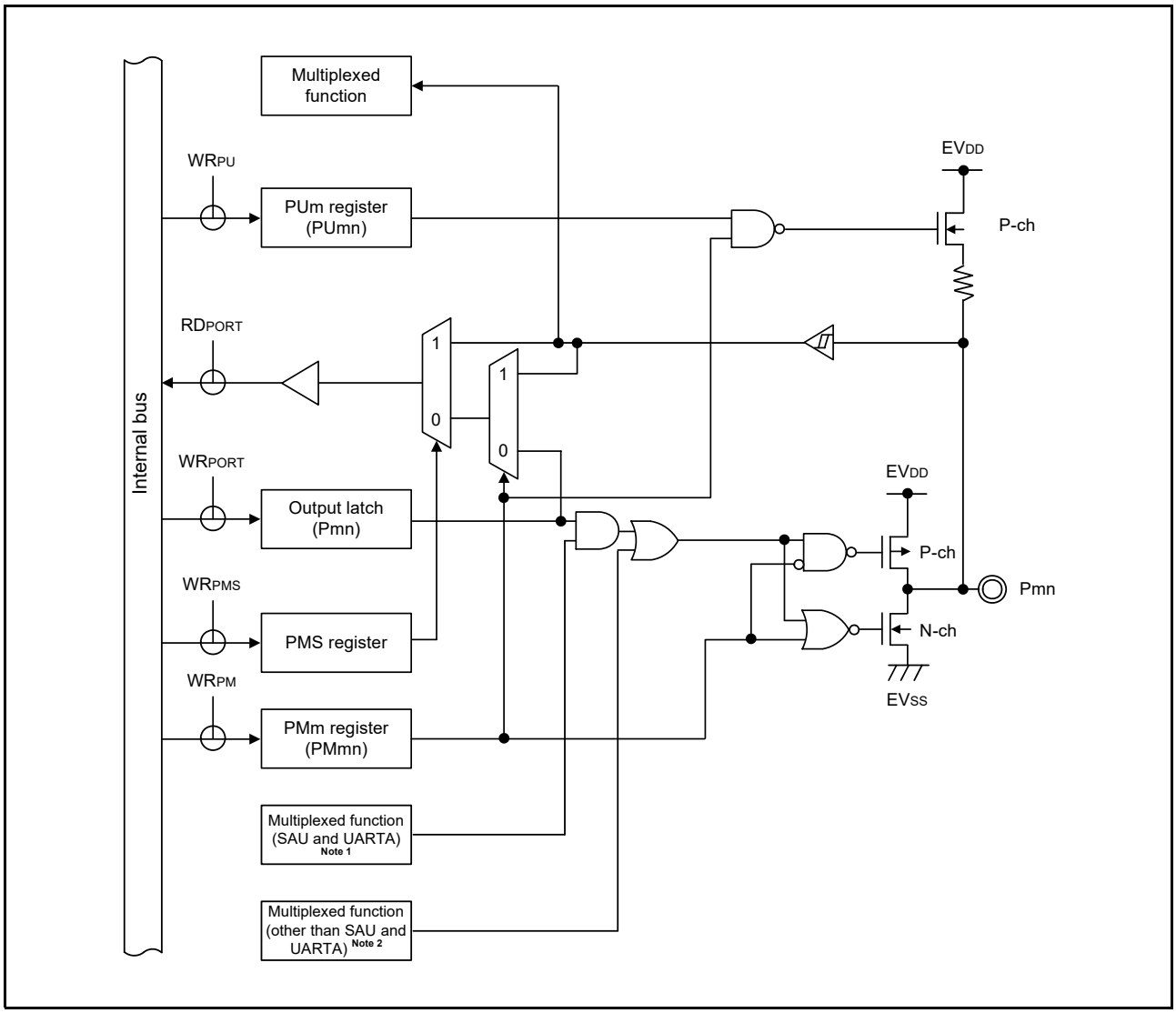


Figure 2 - 11 Pin Block Diagram for Pin Type 7-1-3



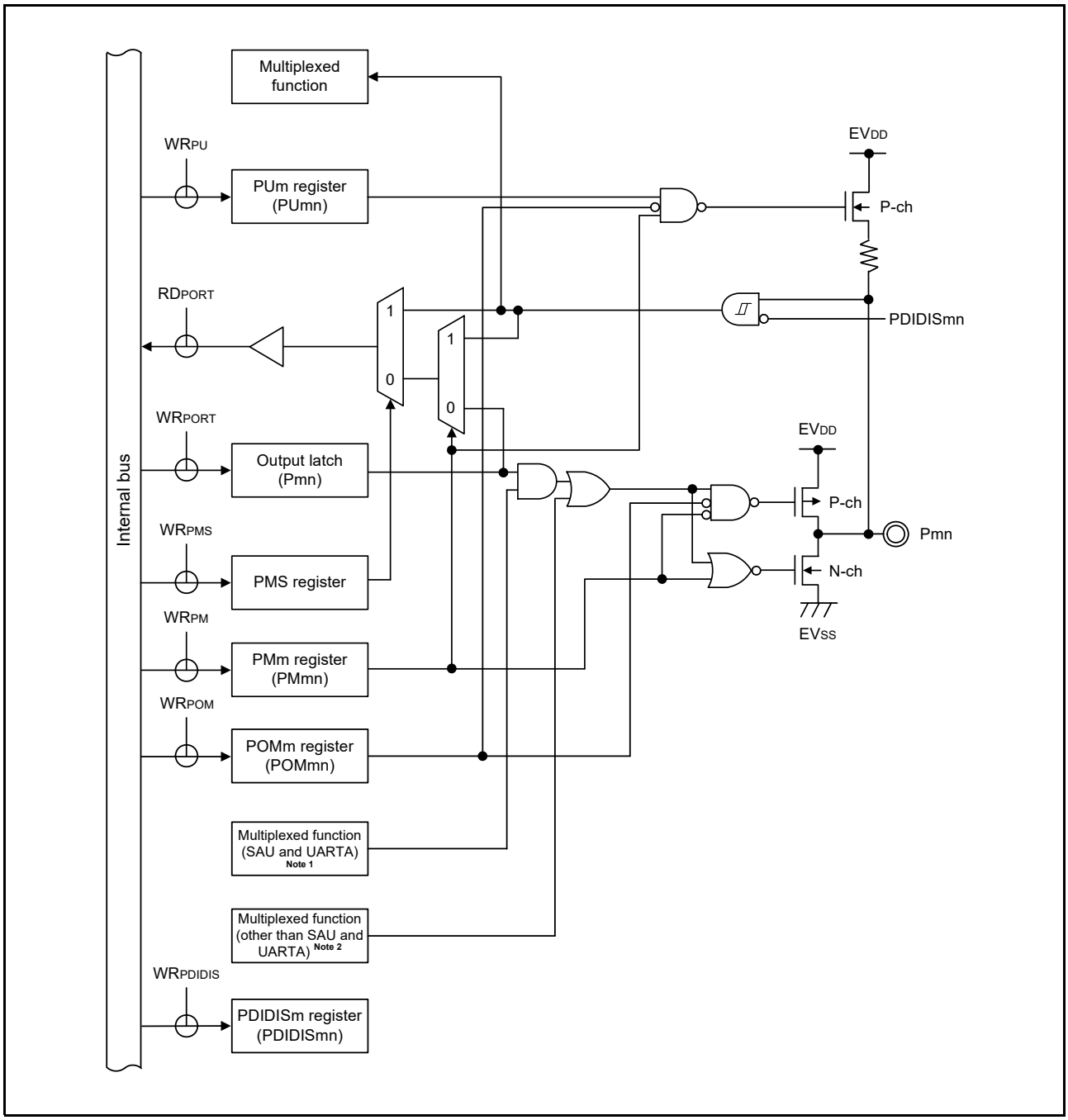
Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

Remark 1. For multiplexed functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

Figure 2 - 12 Pin Block Diagram for Pin Type 7-1-11



Note 1. This excludes the clock output from the UARTA.

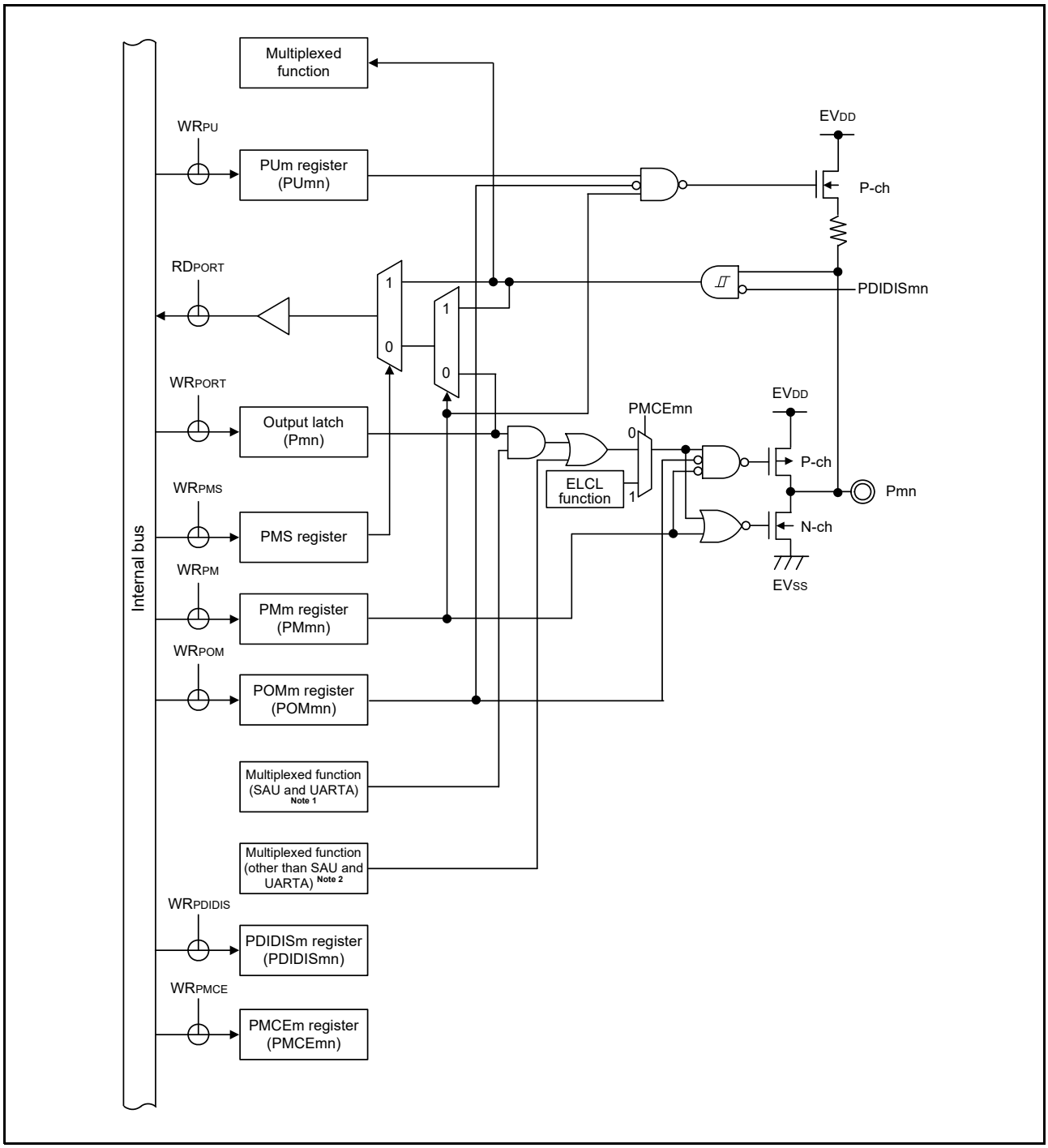
Note 2. This includes the clock output from the UARTA.

Caution **The input buffer is enabled even if the type 7-1-11 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 type 7-1-11 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.**

Remark 1. For multiplexed functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

Figure 2 - 13 Pin Block Diagram for Pin Type 7-1-12



Note 1. This excludes the clock output from the UARTA.

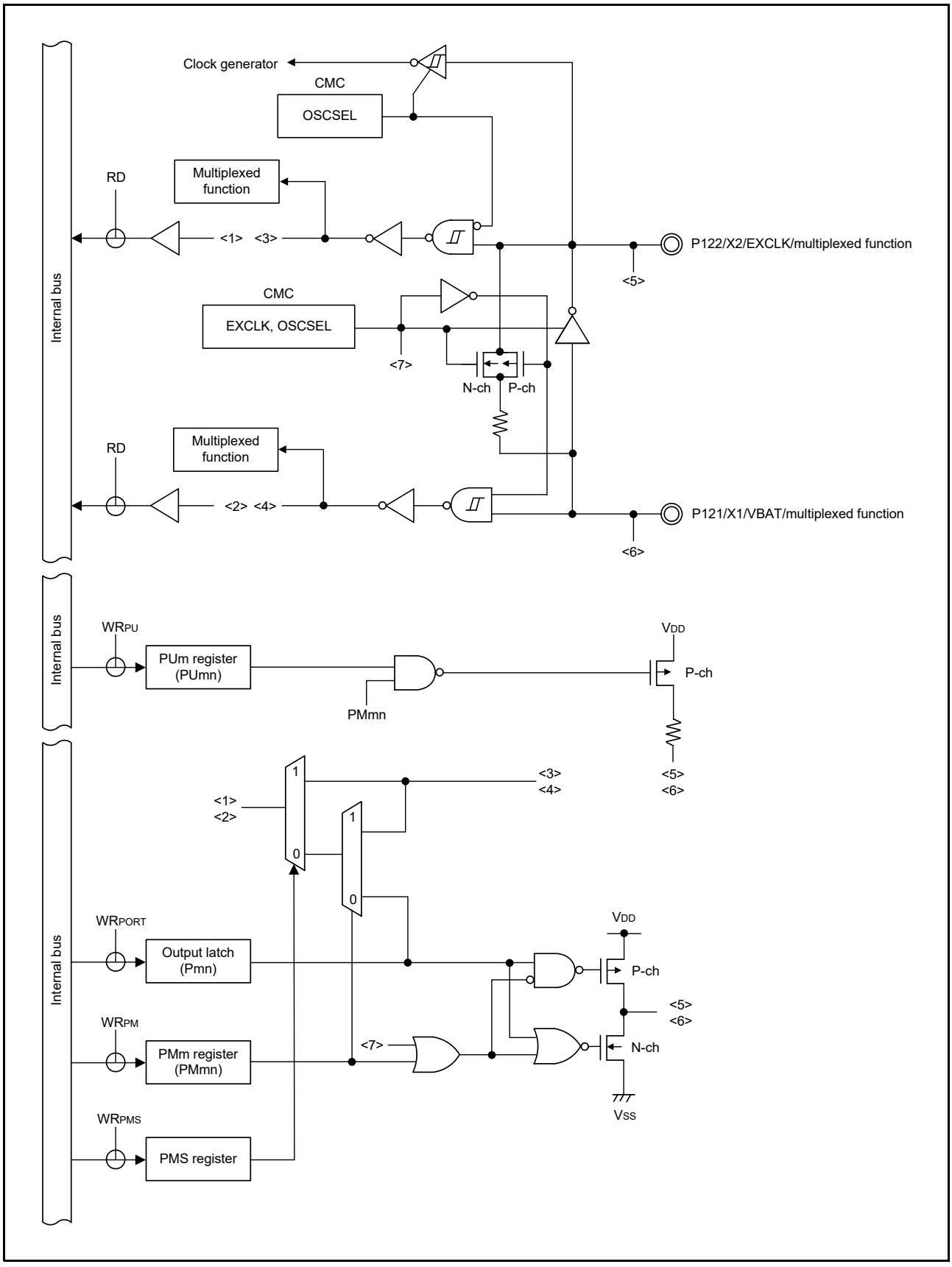
Note 2. This includes the clock output from the UARTA.

Caution **The input buffer is enabled even if the type 7-1-12 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 type 7-1-12 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.**

Remark 1. For multiplexed functions, see **2.1 Functions of Port Pins**.

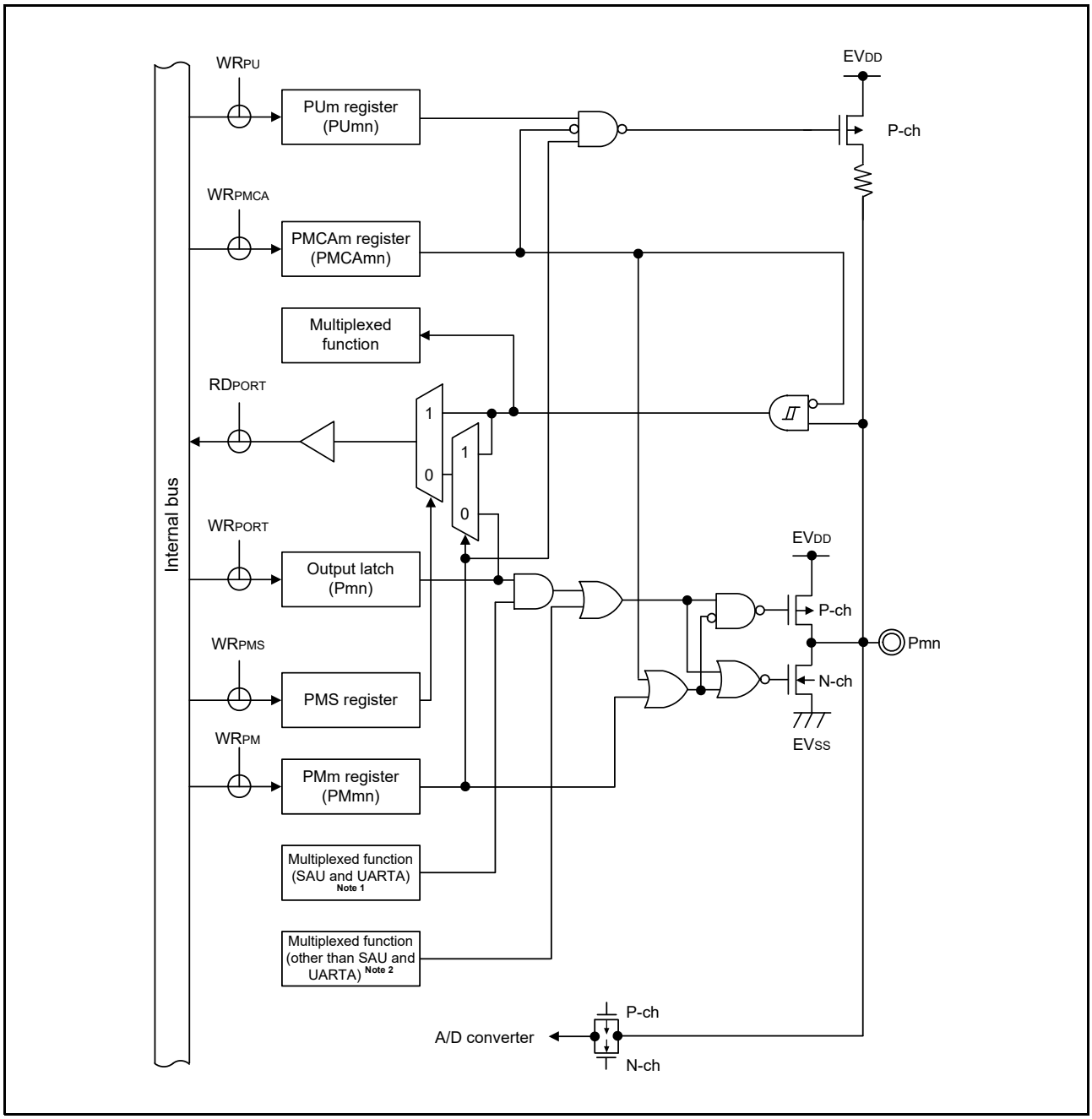
Remark 2. SAU: Serial array unit

Figure 2 - 14 Pin Block Diagram for Pin Type 7-2-1



Remark For multiplexed functions, see **2.1 Functions of Port Pins**.

Figure 2 - 15 Pin Block Diagram for Pin Type 7-3-3



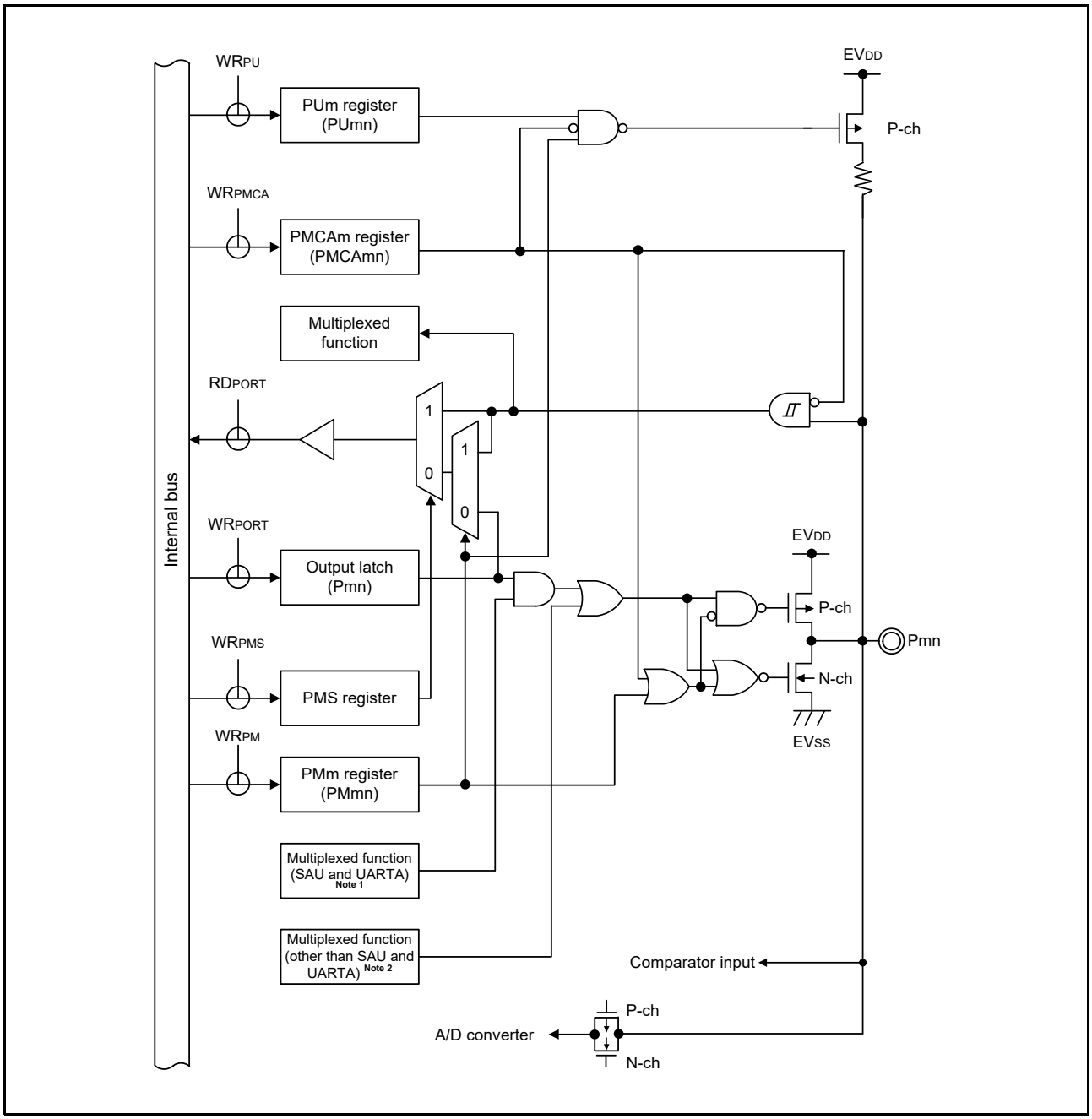
Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

Remark 1. For multiplexed functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

Figure 2 - 16 Pin Block Diagram for Pin Type 7-9-5



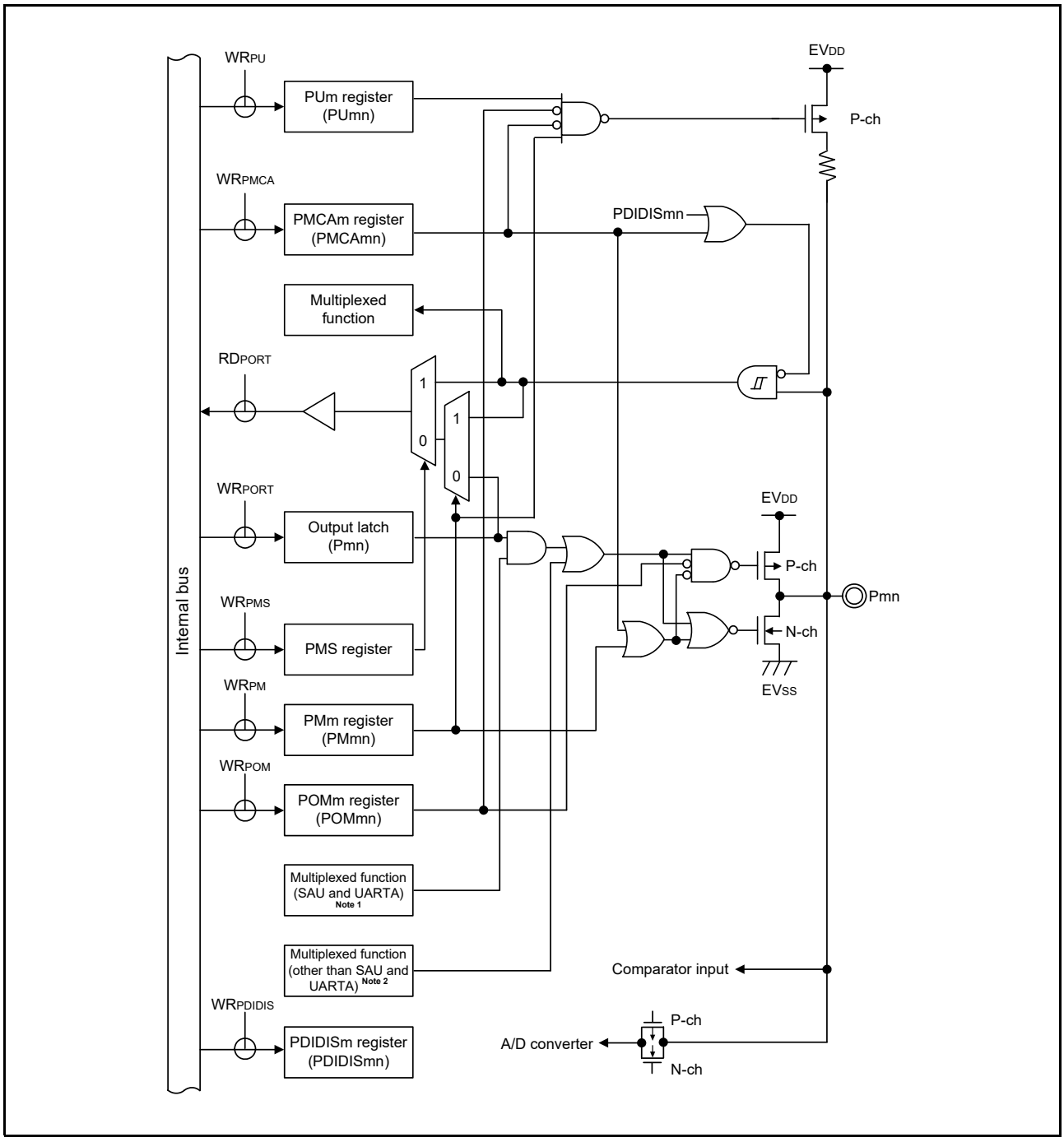
Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

Remark 1. For multiplexed functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

Figure 2 - 17 Pin Block Diagram for Pin Type 7-9-6



Note 1. This excludes the clock output from the UARTA.

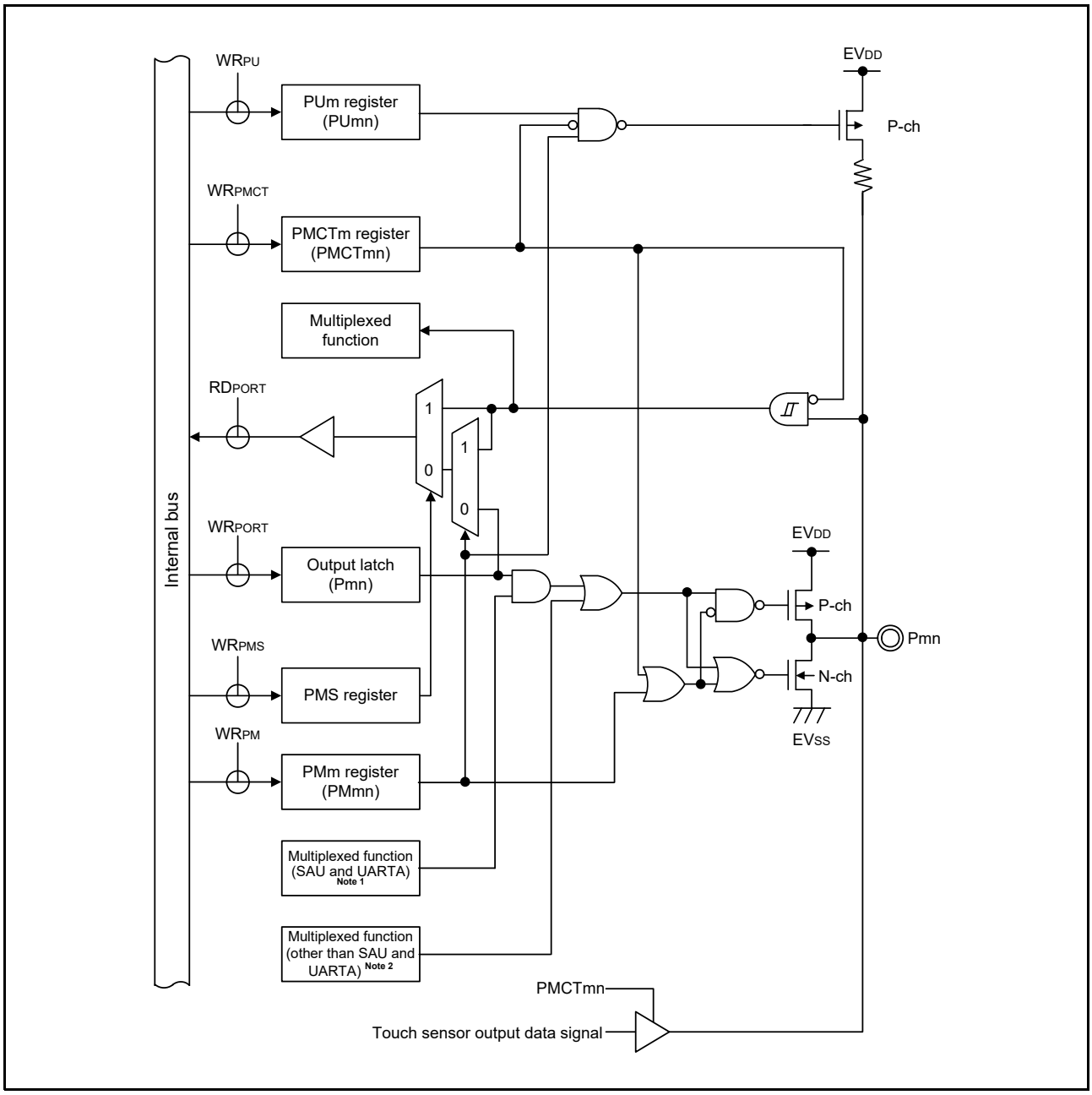
Note 2. This includes the clock output from the UARTA.

Caution **The input buffer is enabled even if the type 7-9-6 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 type 7-9-6 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.**

Remark 1. For multiplexed functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

Figure 2 - 18 Pin Block Diagram for Pin Type 7-31-2



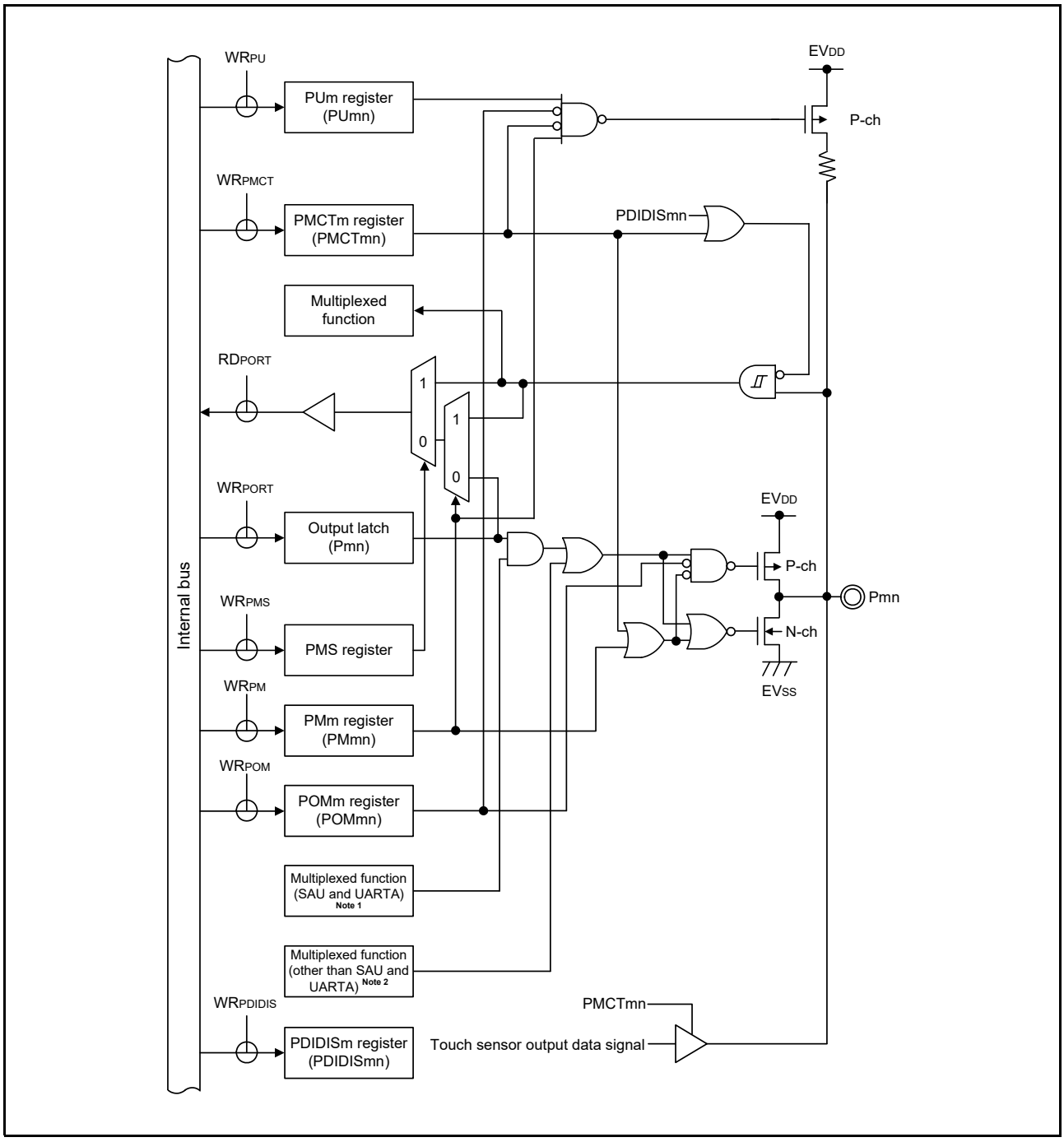
Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

Remark 1. For multiplexed functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

Figure 2 - 19 Pin Block Diagram for Pin Type 7-31-3



Note 1. This excludes the clock output from the UARTA.

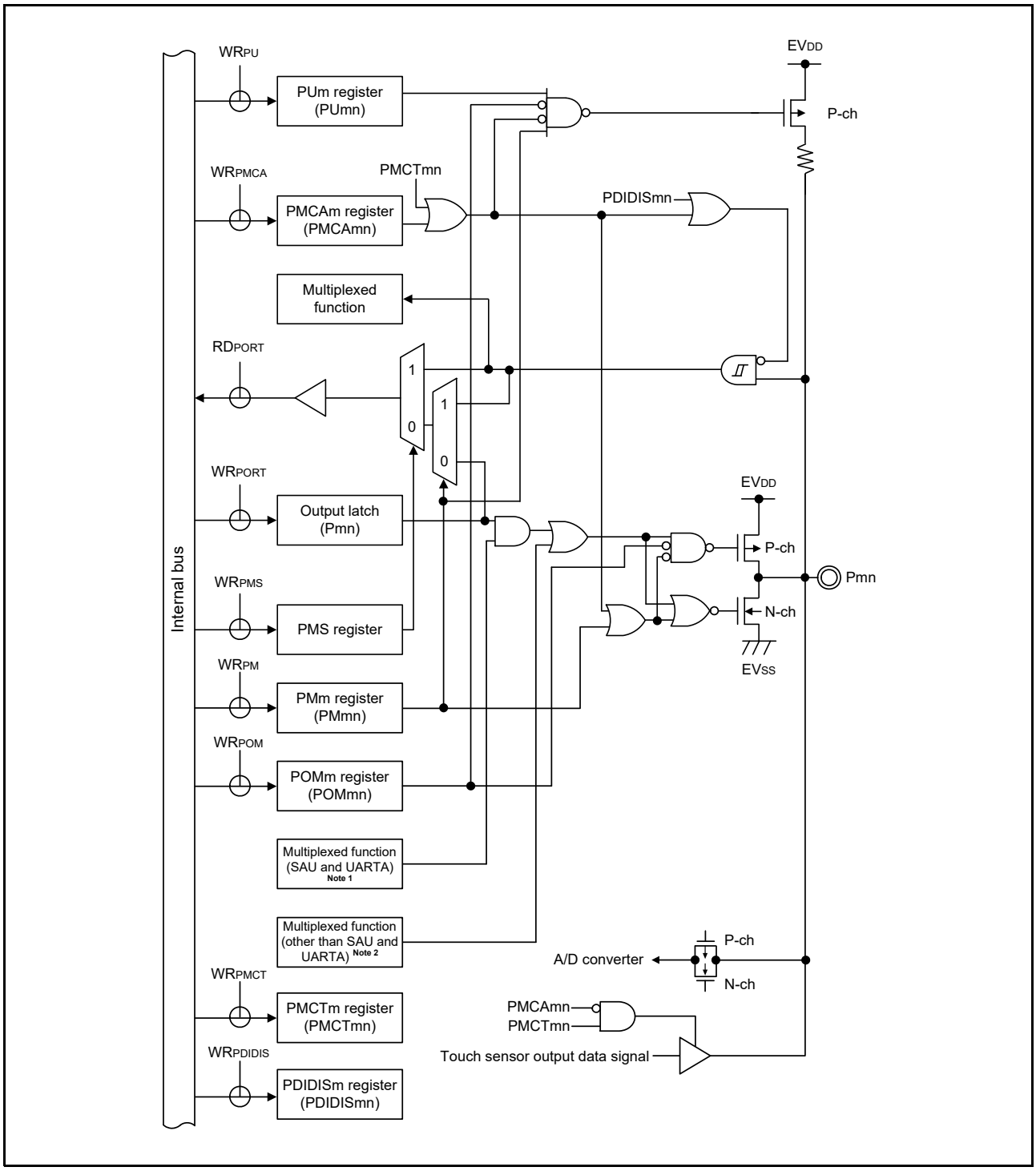
Note 2. This includes the clock output from the UARTA.

Caution **The input buffer is enabled even if the type 7-31-3 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 type 7-31-3 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.**

Remark 1. For multiplexed functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

Figure 2 - 20 Pin Block Diagram for Pin Type 7-33-4



Note 1. This excludes the clock output from the UARTA.

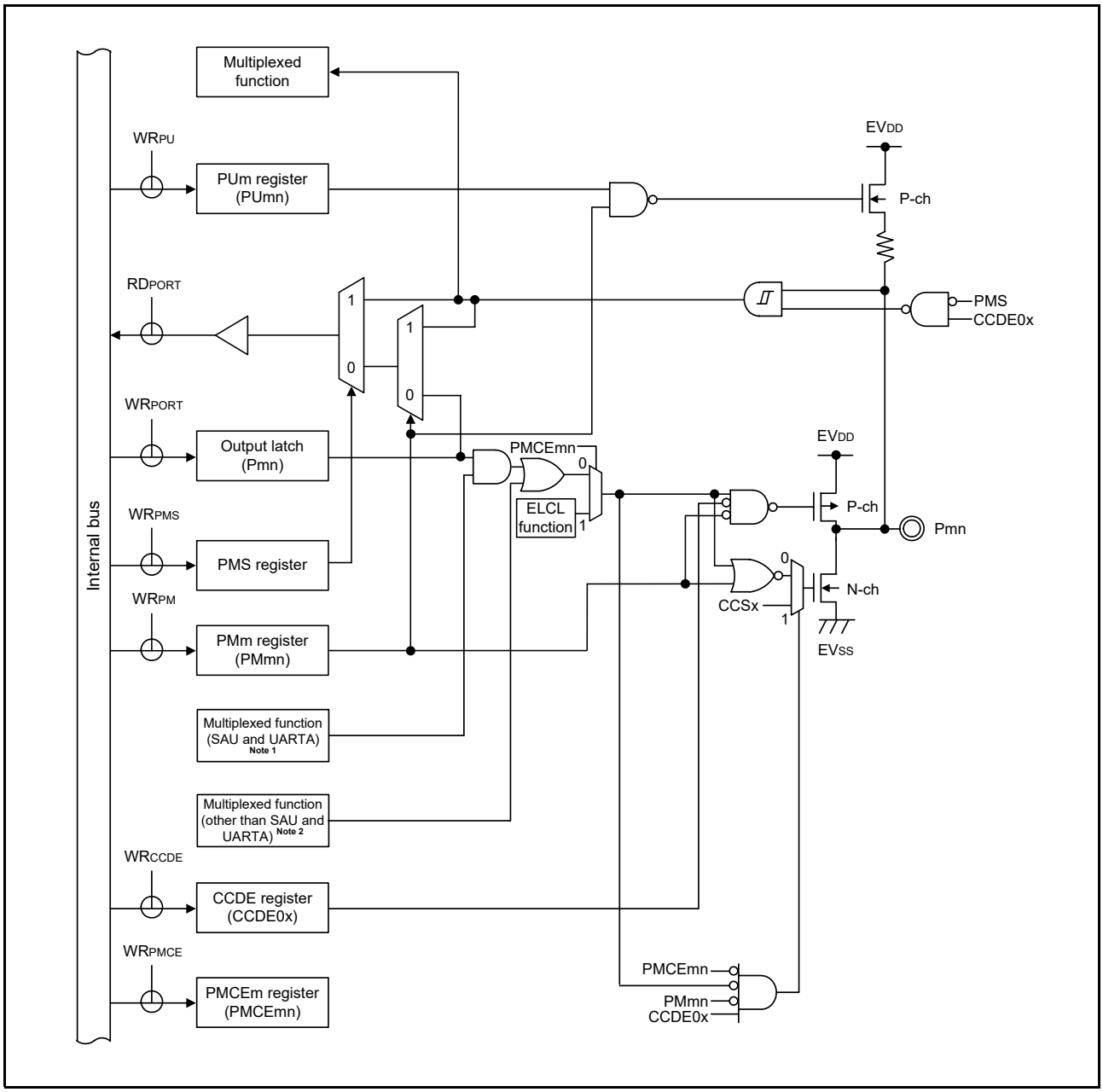
Note 2. This includes the clock output from the UARTA.

Caution **The input buffer is enabled even if the type 7-33-4 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 type 7-33-4 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.**

Remark 1. For multiplexed functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

Figure 2 - 21 Pin Block Diagram for Pin Type 7-38-1



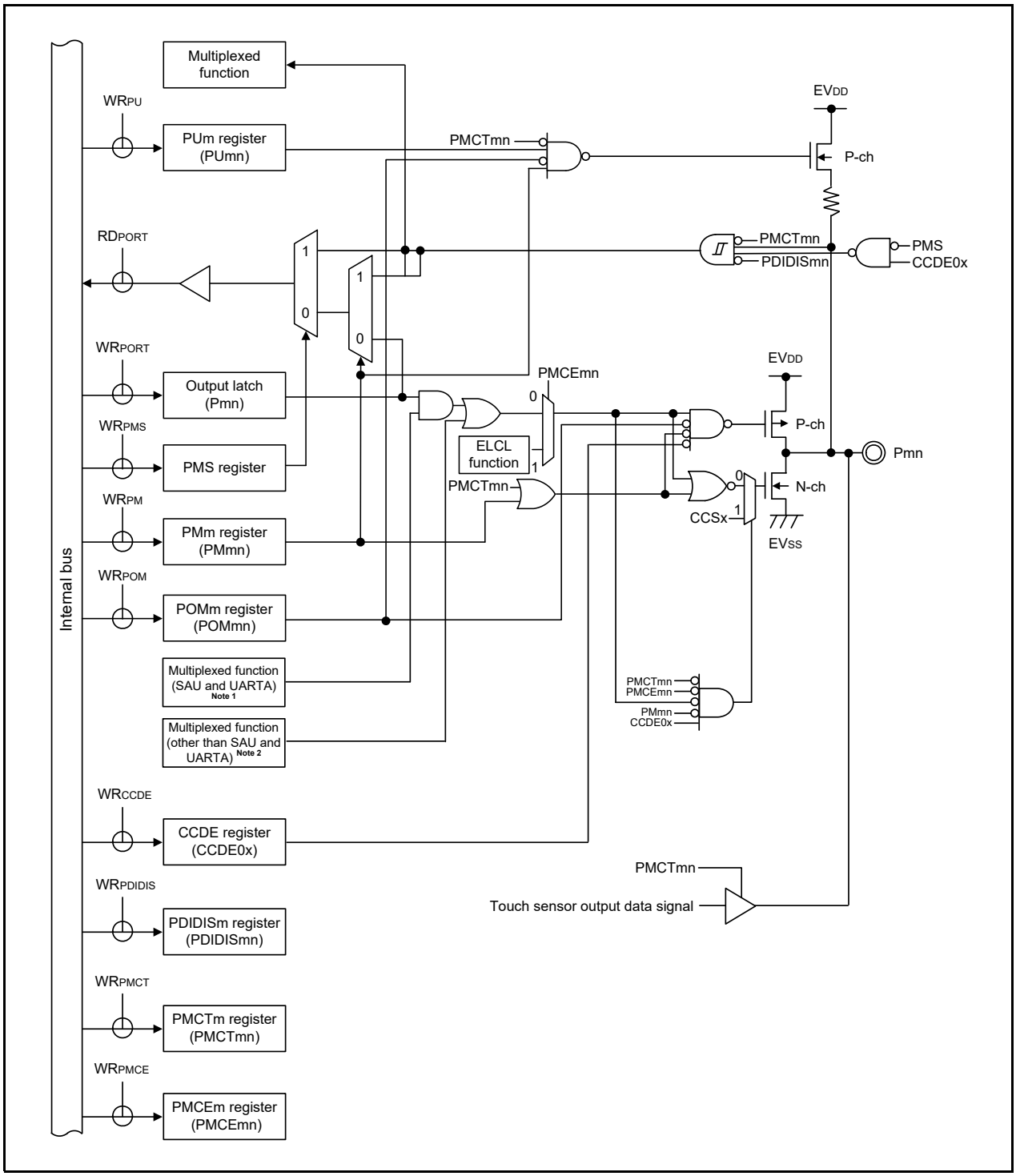
Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

Remark 1. For multiplexed functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

Figure 2 - 22 Pin Block Diagram for Pin Type 7-39-1



Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

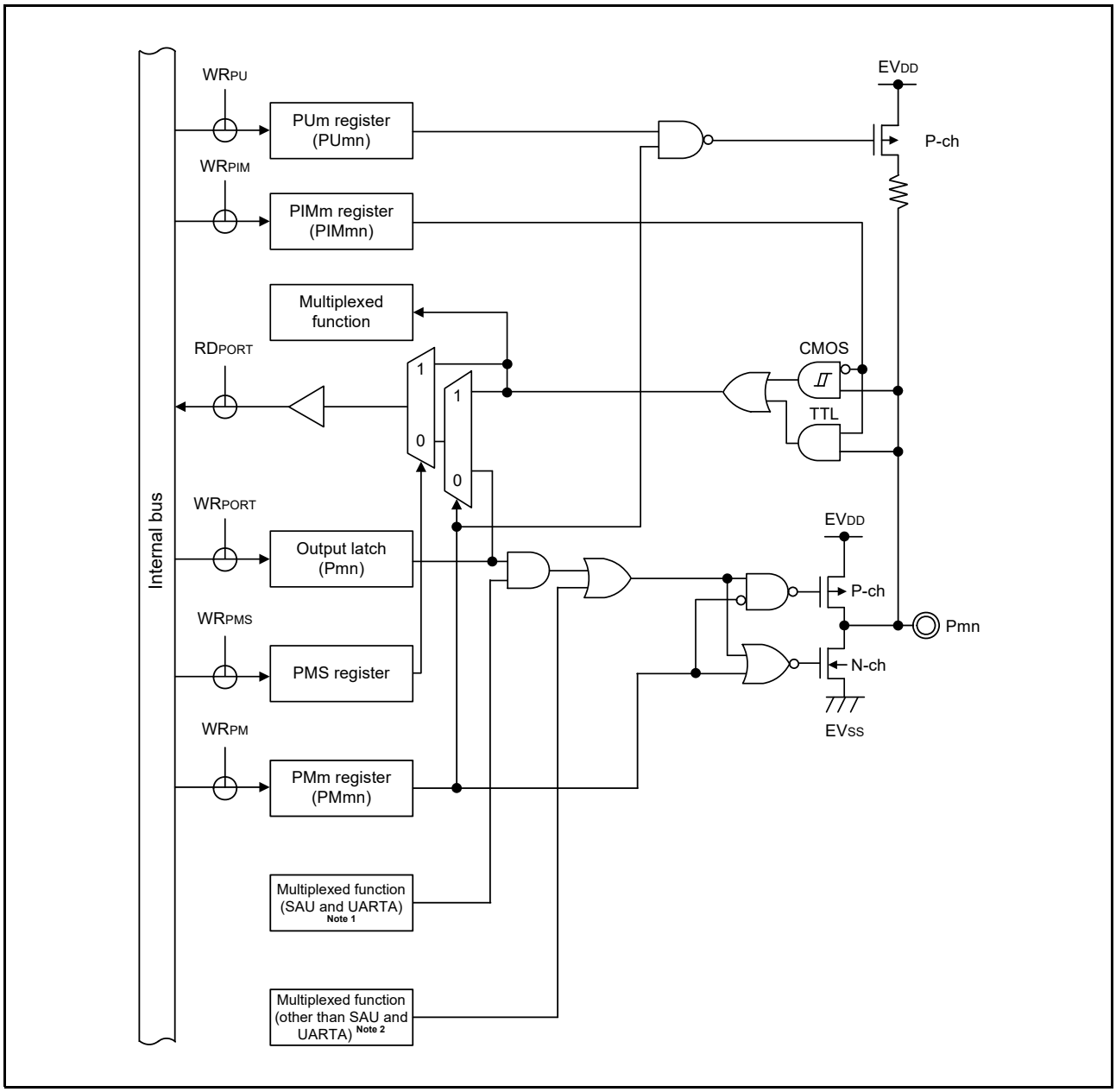
Caution 1. The input buffer is enabled even if the type 7-39-1 pin is operating as an output when the N-ch open-drain output mode is selected by the corresponding bit in the port output mode register (POMm). This may lead to a through current flowing through the type 7-39-1 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.

Caution 2. The 44-pin R7F100G products do not include the controlled current drive port pin in this pin type.

Remark 1. For multiplexed functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

Figure 2 - 23 Pin Block Diagram for Pin Type 8-1-3



Note 1. This excludes the clock output from the UARTA.

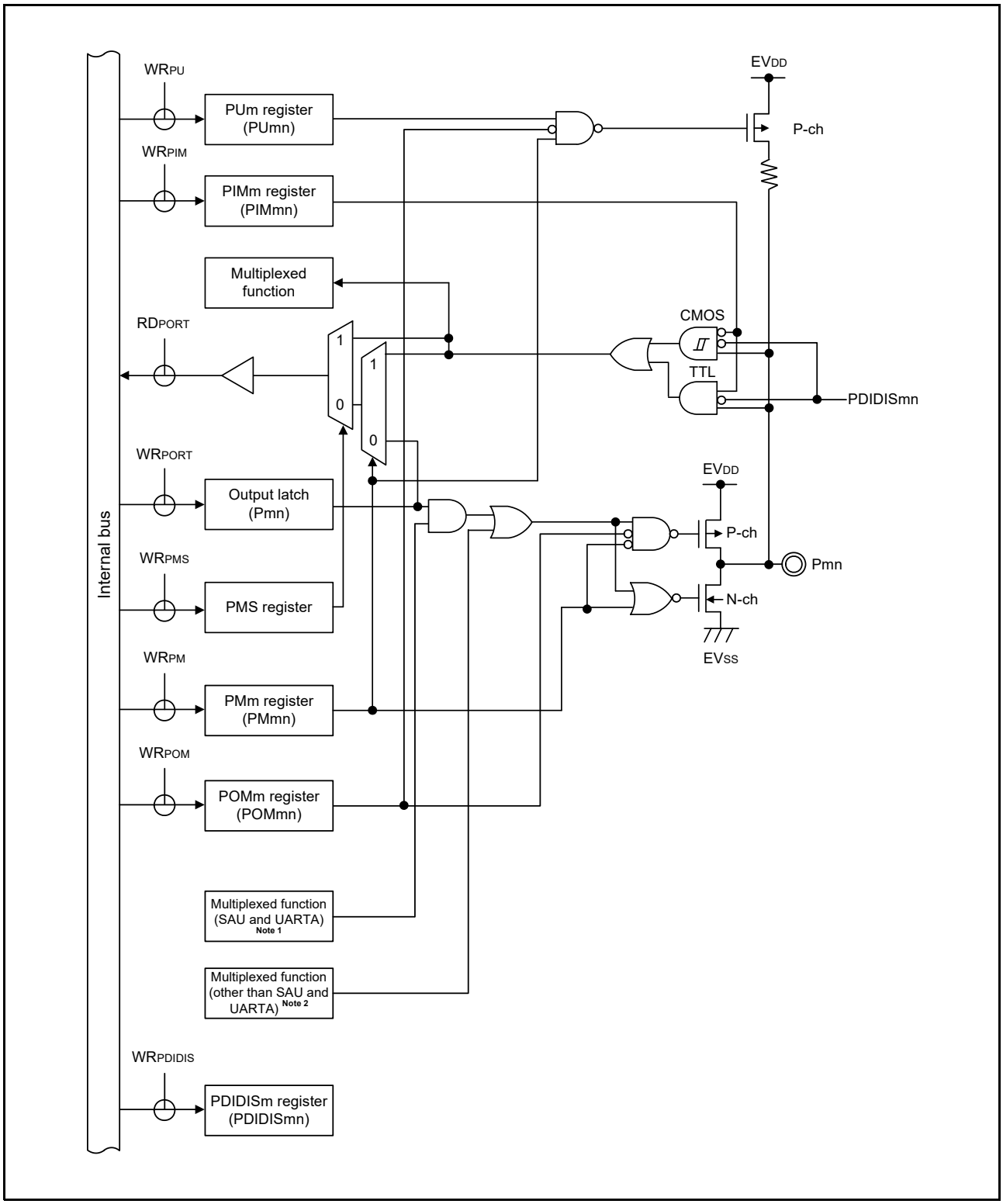
Note 2. This includes the clock output from the UARTA.

Caution **When the type 8-1-3 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 type 8-1-3 pin due to the configuration of the TTL input buffer. Drive the type 8-1-3 pin low to prevent the through current.**

Remark 1. For multiplexed functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

Figure 2 - 24 Pin Block Diagram for Pin Type 8-1-10



Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

(Cautions and Remarks are listed on the next page.)

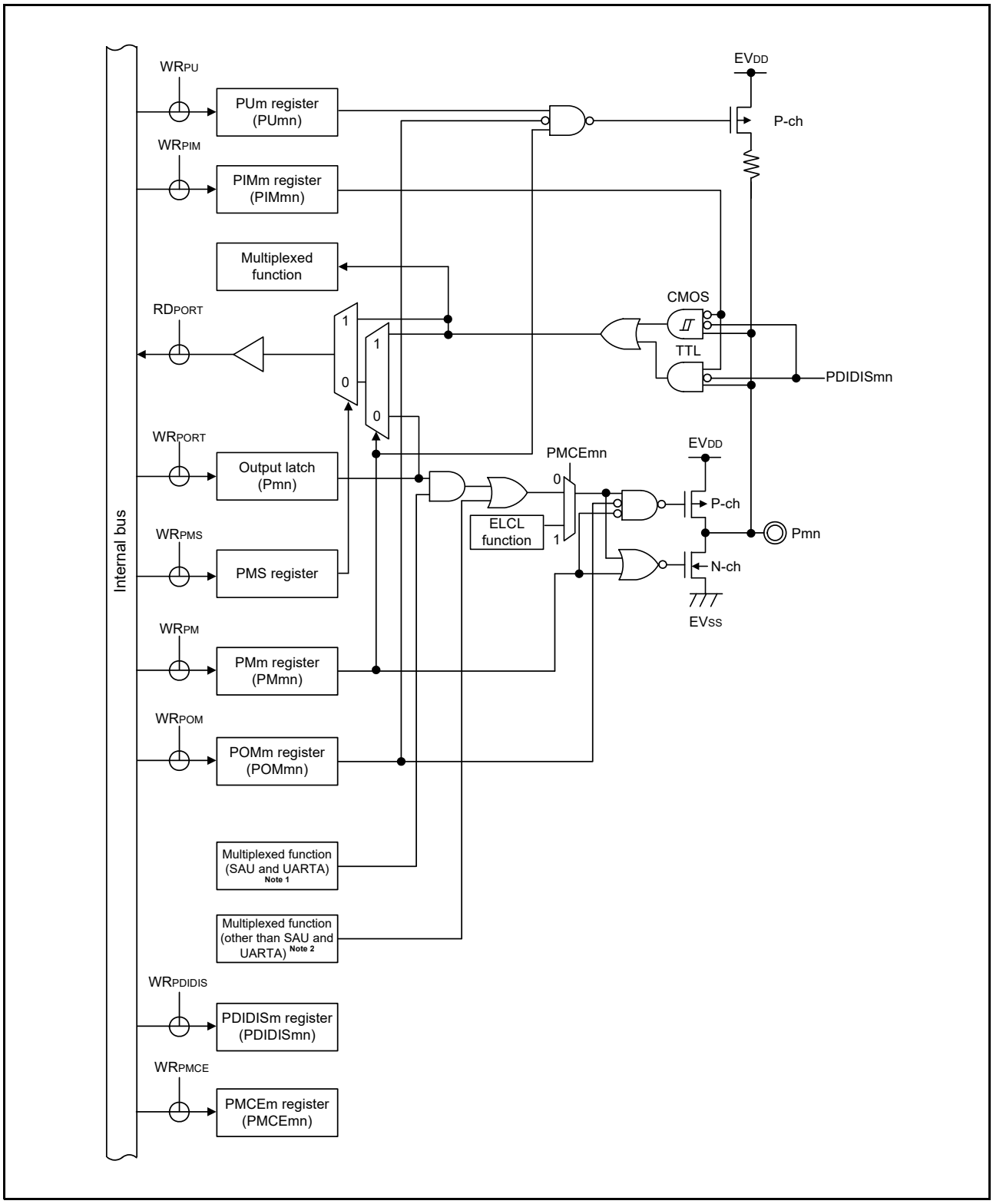
Caution 1. **The input buffer is enabled even if the type 8-1-10 pin is operating as an output when the N-ch open-drain output mode is selected by the corresponding bit in the port output mode register (POMm). This may lead to a through current flowing through the type 8-1-10 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.**

Caution 2. **When the type 8-1-10 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 type 8-1-10 pin due to the configuration of the TTL input buffer. Drive the type 8-1-10 pin low to prevent the through current.**

Remark 1. For multiplexed functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

Figure 2 - 25 Pin Block Diagram for Pin Type 8-1-11



Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

(Cautions and Remarks are listed on the next page.)

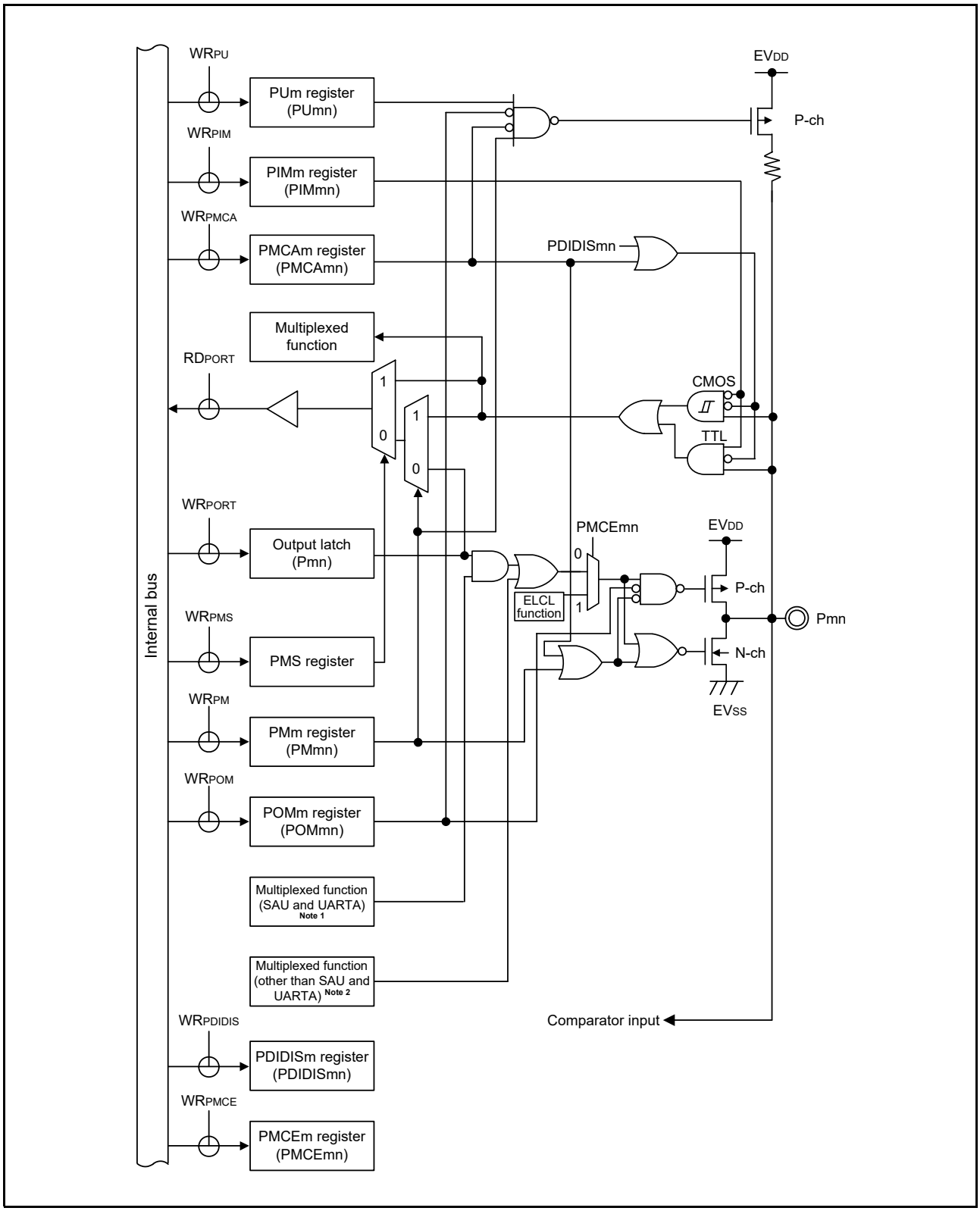
Caution 1. **The input buffer is enabled even if the type 8-1-11 pin is operating as an output when the N-ch open-drain output mode is selected by the corresponding bit in the port output mode register (POMm). This may lead to a through current flowing through the type 8-1-11 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.**

Caution 2. **When the type 8-1-11 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 type 8-1-11 pin due to the configuration of the TTL input buffer. Drive the type 8-1-11 pin low to prevent the through current.**

Remark 1. For multiplexed functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

Figure 2 - 26 Pin Block Diagram for Pin Type 8-6-9



Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

(Cautions and Remarks are listed on the next page.)

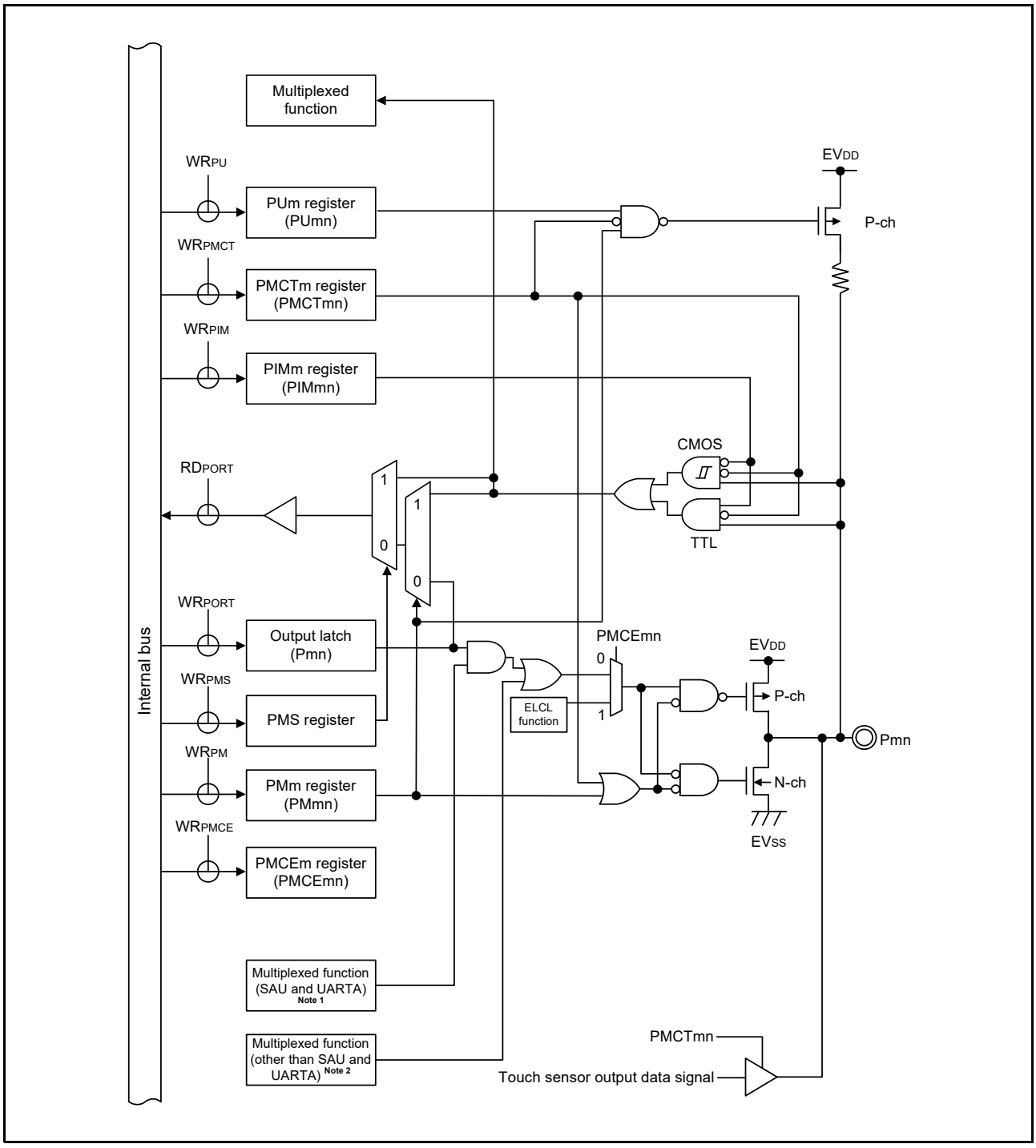
Caution 1. **The input buffer is enabled even if the type 8-6-9 pin is operating as an output when the N-ch open-drain output mode is selected by the corresponding bit in the port output mode register (POMm). This may lead to a through current flowing through the type 8-6-9 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.**

Caution 2. **When the type 8-6-9 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 type 8-6-9 pin due to the configuration of the TTL input buffer. Drive the type 8-6-9 pin low to prevent the through current.**

Remark 1. For multiplexed functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

Figure 2 - 27 Pin Block Diagram for Pin Type 8-31-1



Note 1. This excludes the clock output from the UARTA.

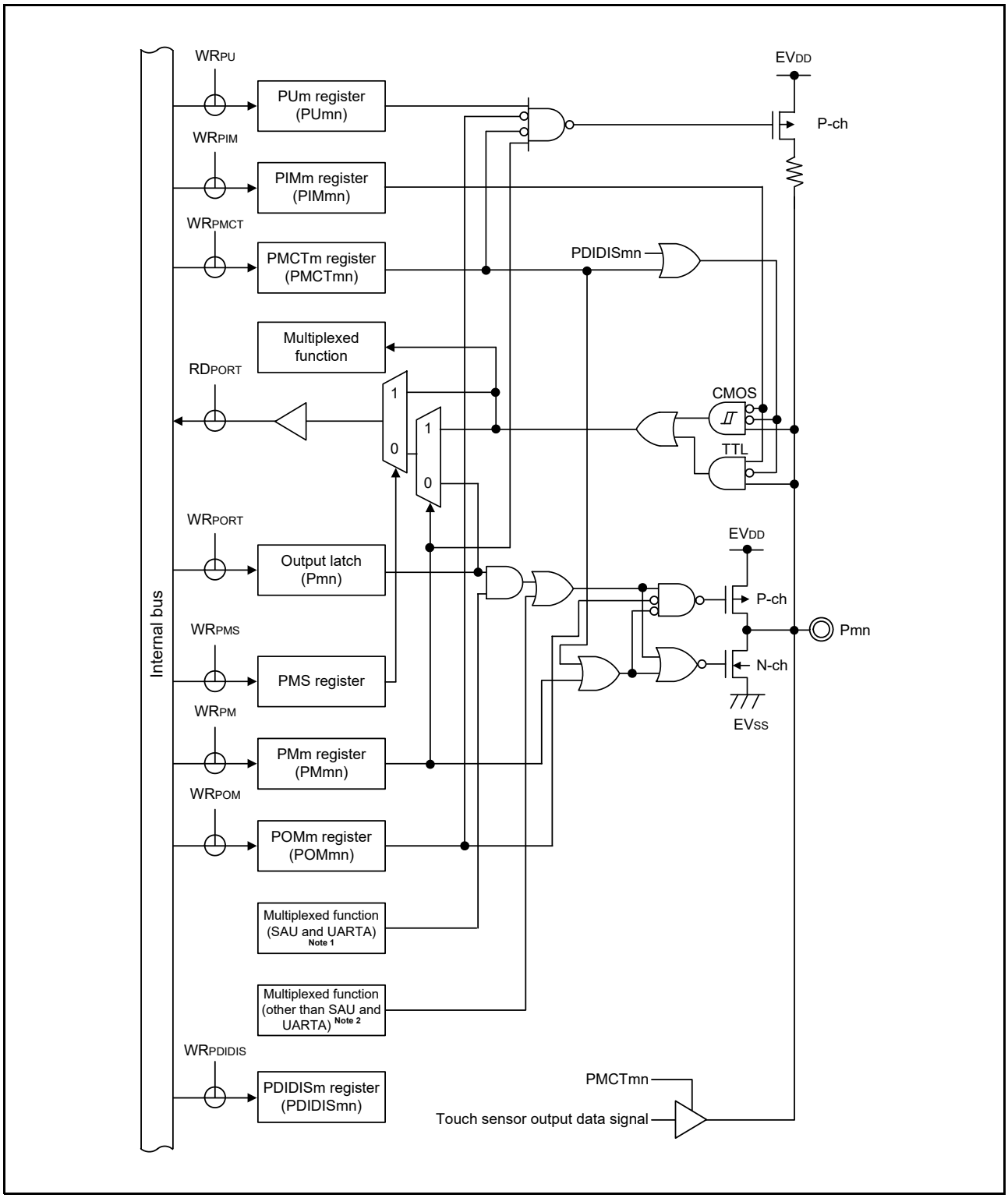
Note 2. This includes the clock output from the UARTA.

Caution **When the type 8-31-1 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 type 8-31-1 pin due to the configuration of the TTL input buffer. Drive the type 8-31-1 pin low to prevent the through current.**

Remark 1. For multiplexed functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

Figure 2 - 28 Pin Block Diagram for Pin Type 8-31-2



Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

(Cautions and Remarks are listed on the next page.)

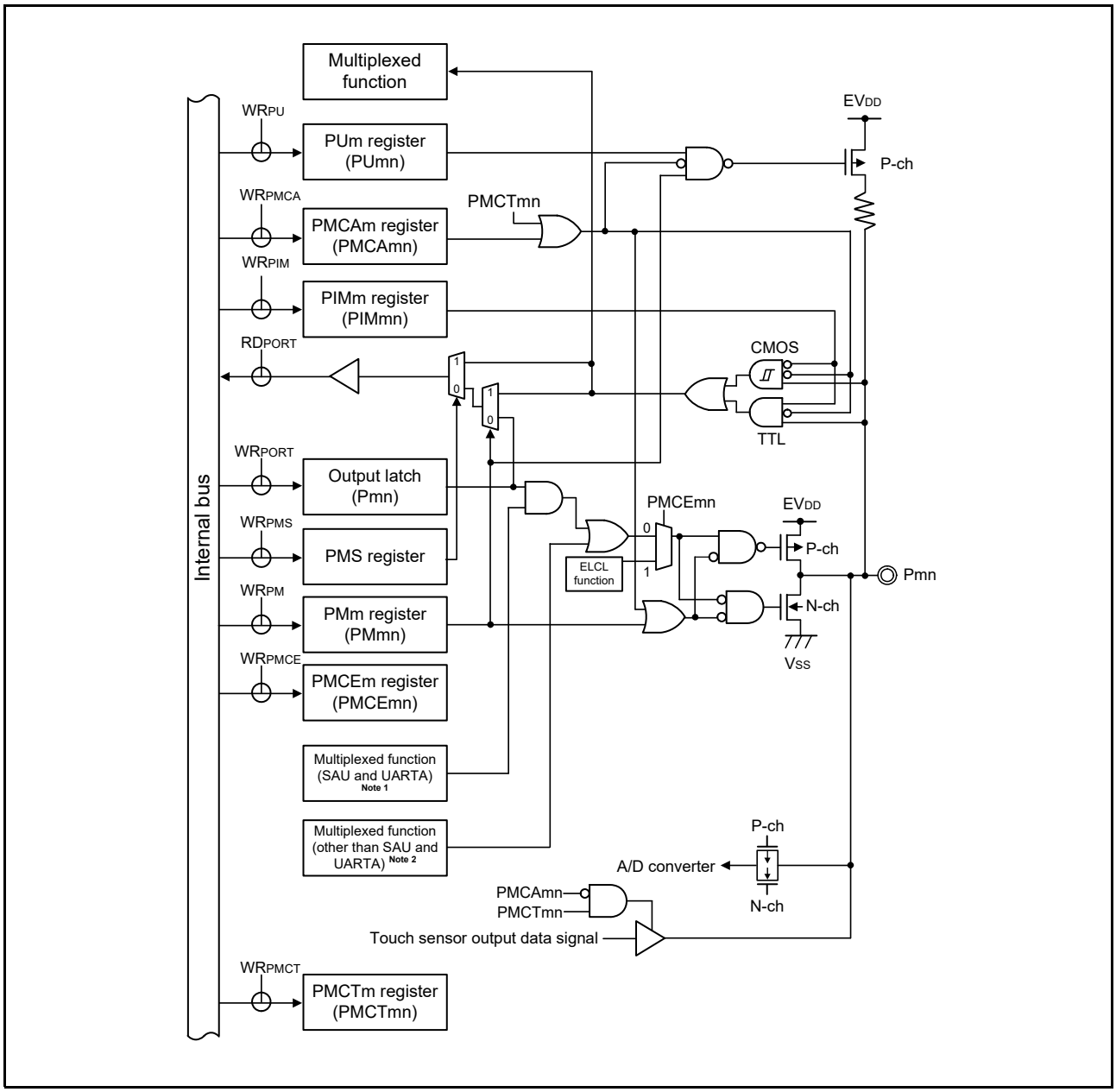
Caution 1. **The input buffer is enabled even if the type 8-31-2 pin is operating as an output when the N-ch open-drain output mode is selected by the corresponding bit in the port output mode register (POMm). This may lead to a through current flowing through the type 8-31-2 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.**

Caution 2. **When the type 8-31-2 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 type 8-31-2 pin due to the configuration of the TTL input buffer. Drive the type 8-31-2 pin low to prevent the through current.**

Remark 1. For multiplexed functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

Figure 2 - 29 Pin Block Diagram for Pin Type 8-33-2



Note 1. This excludes the clock output from the UARTA.

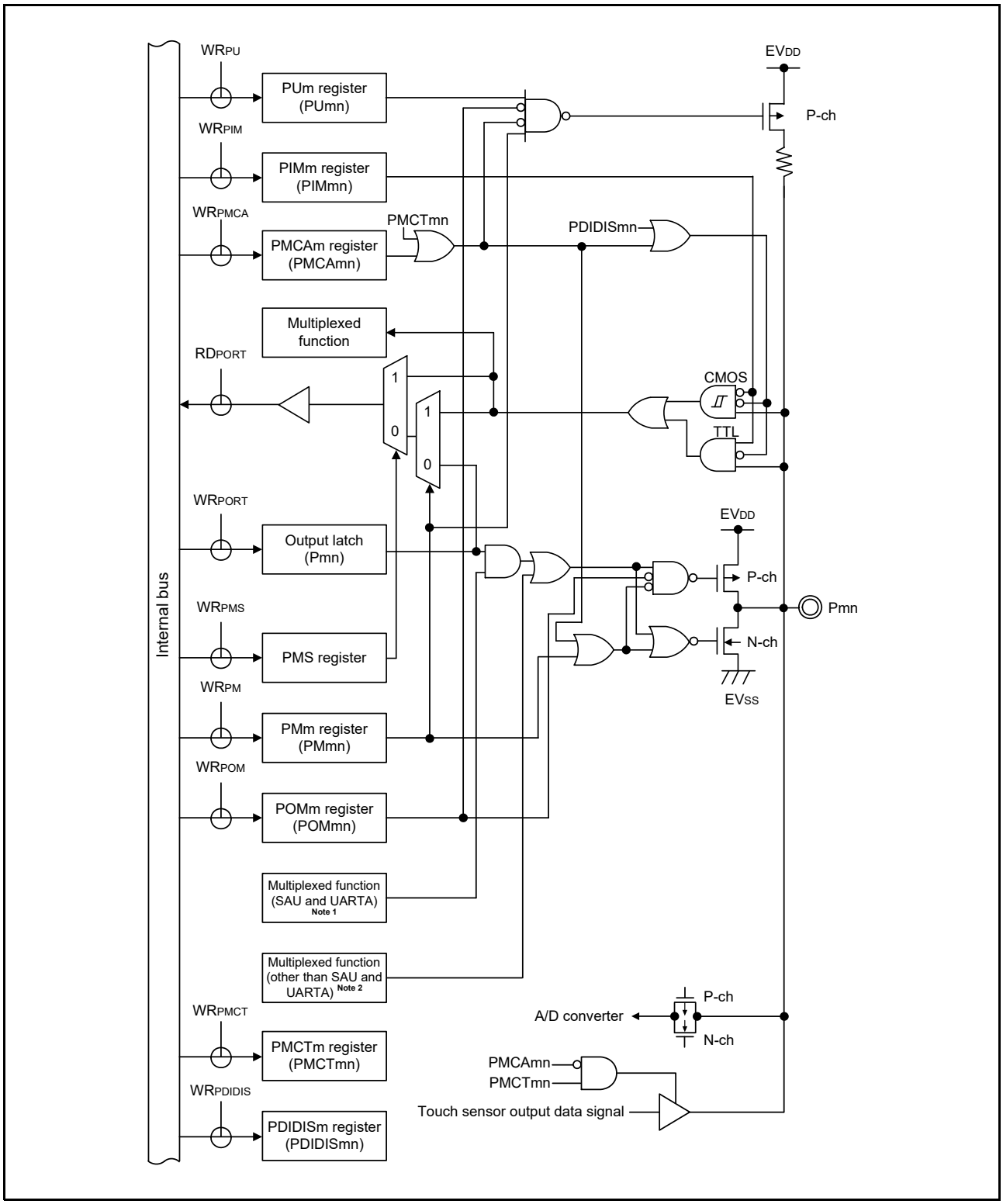
Note 2. This includes the clock output from the UARTA.

Caution **When the type 8-33-2 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 type 8-33-2 pin due to the configuration of the TTL input buffer. Drive the type 8-33-2 pin low to prevent the through current.**

Remark 1. For multiplexed functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

Figure 2 - 30 Pin Block Diagram for Pin Type 8-33-3



Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

(Cautions and Remarks are listed on the next page.)

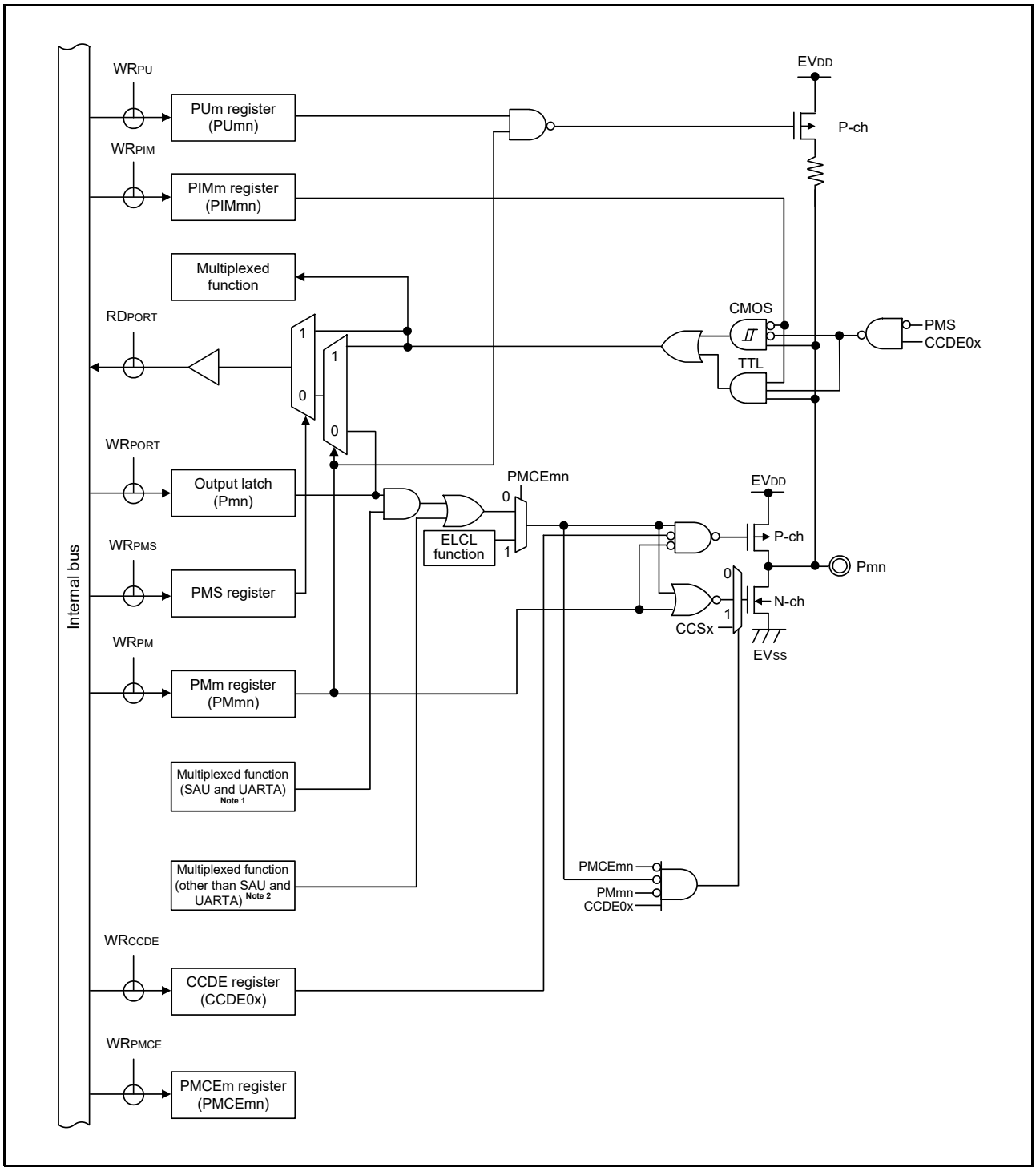
Caution 1. **The input buffer is enabled even if the type 8-33-3 pin is operating as an output when the N-ch open-drain output mode is selected by the corresponding bit in the port output mode register (POMm). This may lead to a through current flowing through the type 8-33-3 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.**

Caution 2. **When the type 8-33-3 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 type 8-33-3 pin due to the configuration of the TTL input buffer. Drive the type 8-33-3 pin low to prevent the through current.**

Remark 1. For multiplexed functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

Figure 2 - 31 Pin Block Diagram for Pin Type 8-38-1



Note 1. This excludes the clock output from the UARTA.

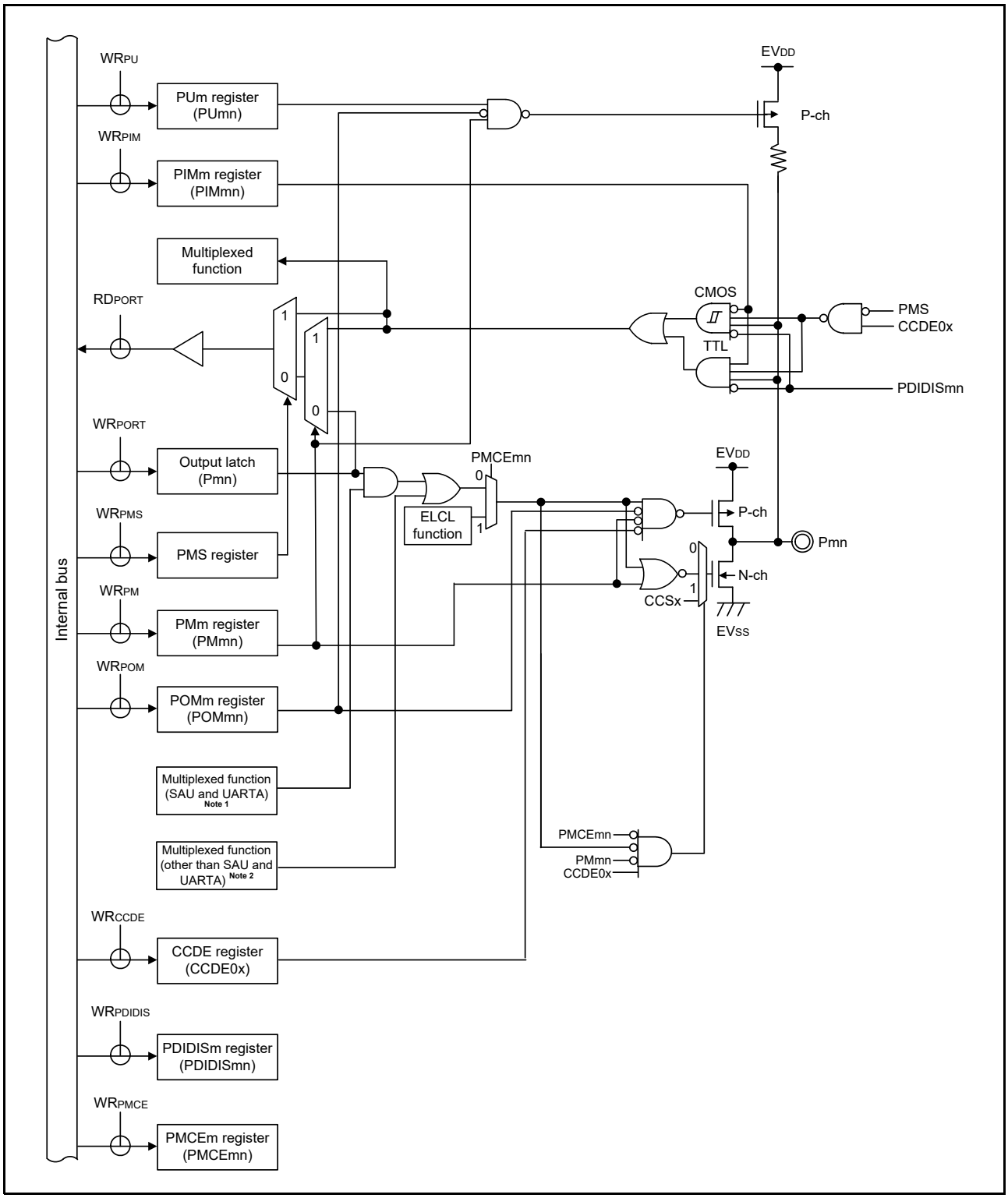
Note 2. This includes the clock output from the UARTA.

Caution **When the type 8-38-1 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 type 8-38-1 pin due to the configuration of the TTL input buffer. Drive the type 8-38-1 pin low to prevent the through current.**

Remark 1. For multiplexed functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

Figure 2 - 32 Pin Block Diagram for Pin Type 8-38-2



Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

(Cautions and Remarks are listed on the next page.)

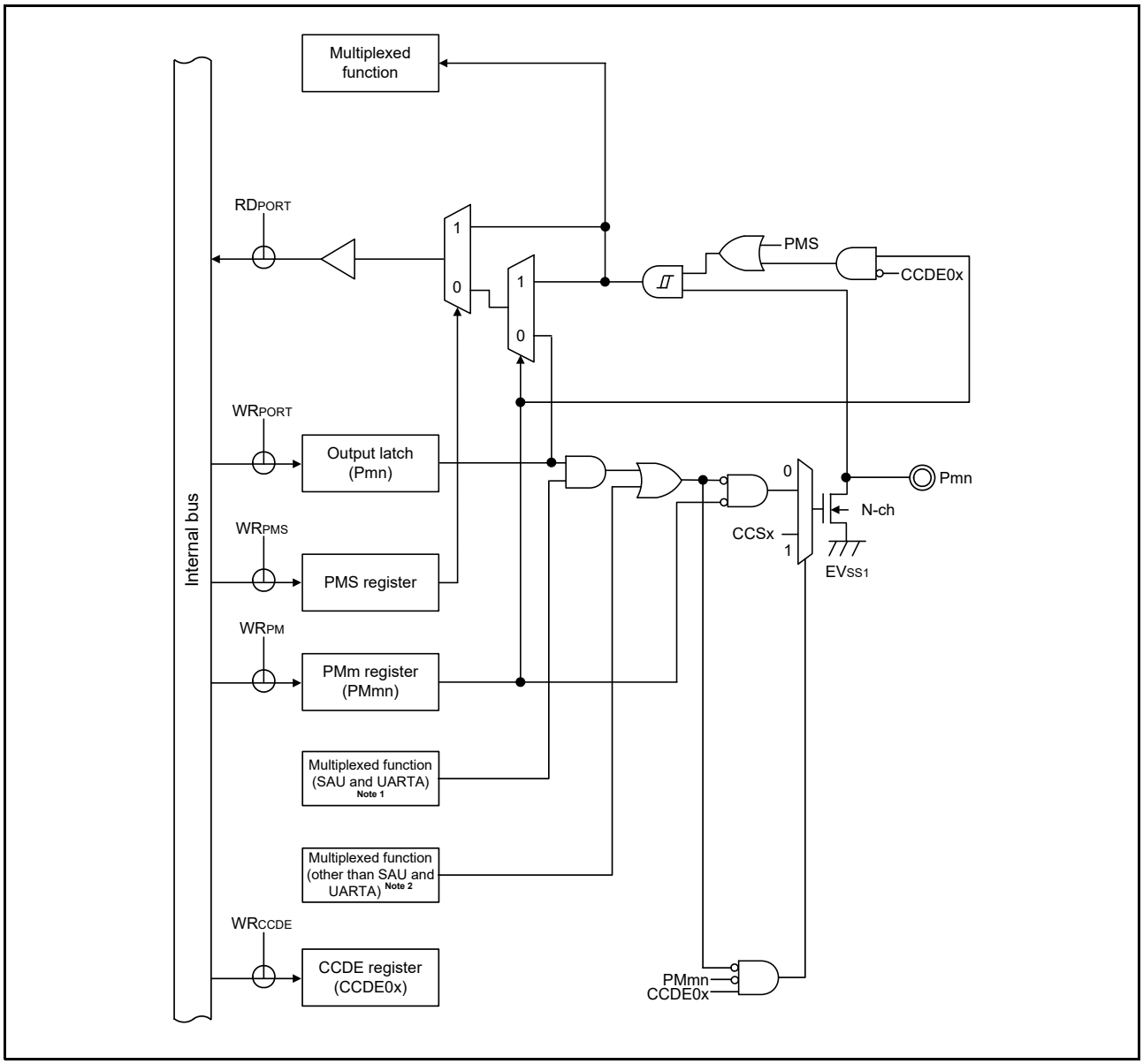
Caution 1. **The input buffer is enabled even if the type 8-38-2 pin is operating as an output when the N-ch open-drain output mode is selected by the corresponding bit in the port output mode register (POMm). This may lead to a through current flowing through the type 8-38-2 pin when the voltage level on this pin is intermediate. However, setting the corresponding bit of the given PDIDISm register to 1 prevents the flow of a through current.**

Caution 2. **When the type 8-38-2 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 type 8-38-2 pin due to the configuration of the TTL input buffer. Drive the type 8-38-2 pin low to prevent the through current.**

Remark 1. For multiplexed functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

Figure 2 - 33 Pin Block Diagram for Pin Type 12-38-1



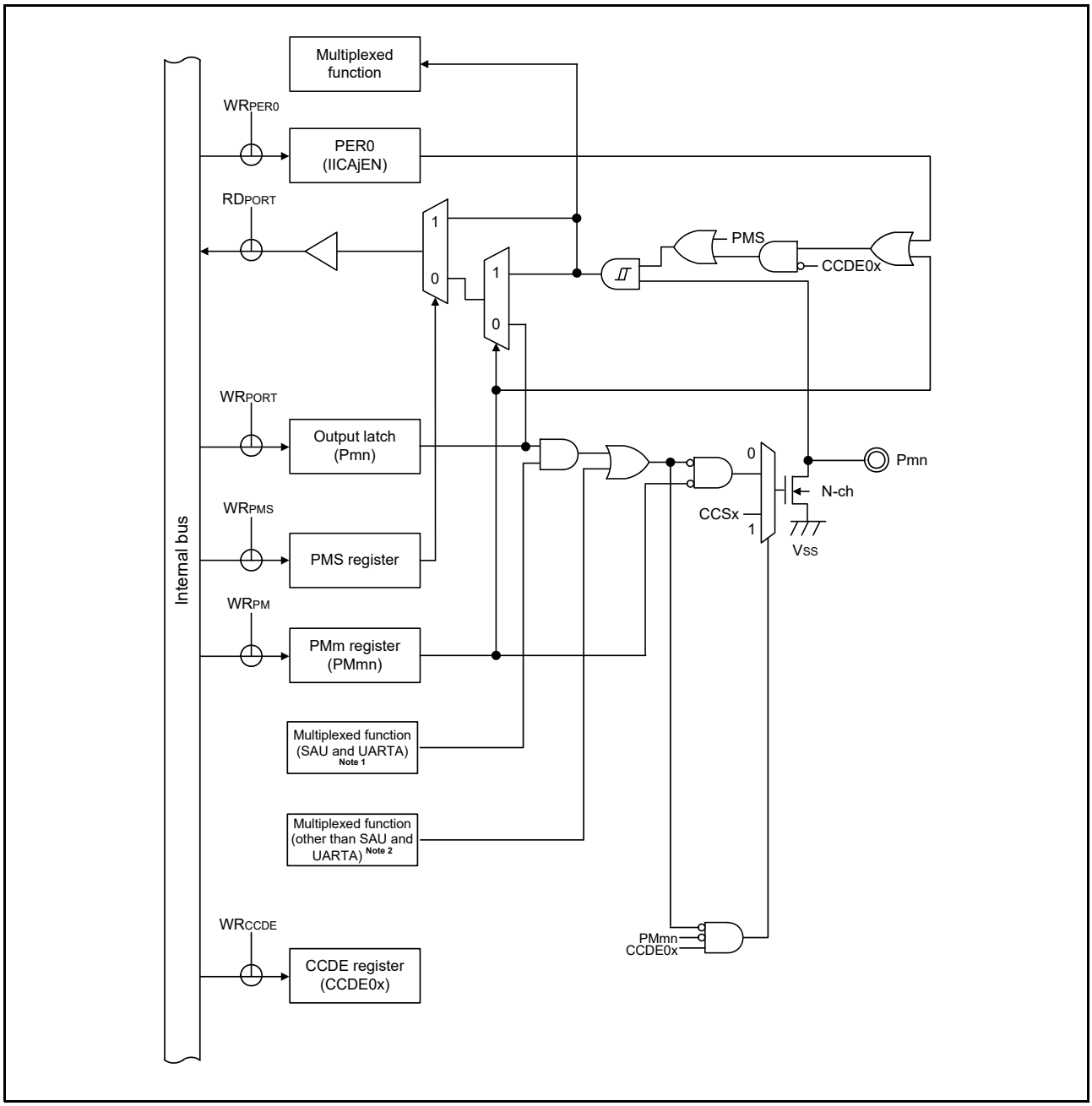
Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

Remark 1. For multiplexed functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

Figure 2 - 34 Pin Block Diagram for Pin Type 12-38-2



Note 1. This excludes the clock output from the UARTA.

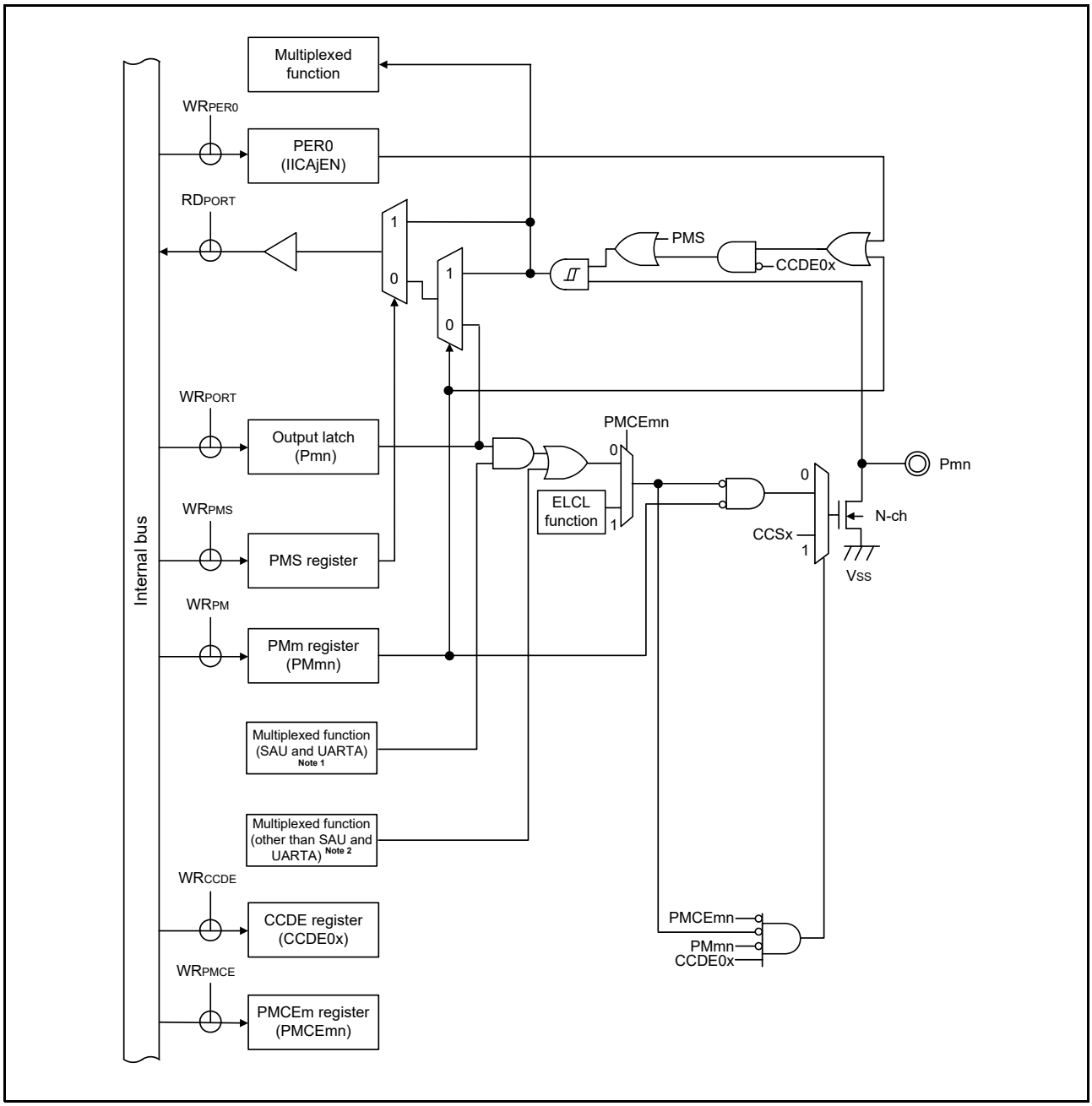
Note 2. This includes the clock output from the UARTA.

Caution **The input buffer is enabled when using the IICA with the IICAjEN bit in the PER0 register being set to 1. This may lead to a through current flowing through the type 12-38-2 pin when the voltage level on this pin is intermediate.**

Remark 1. For multiplexed functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

Figure 2 - 35 Pin Block Diagram for Pin Type 12-38-3



Note 1. This excludes the clock output from the UARTA.

Note 2. This includes the clock output from the UARTA.

Caution **The input buffer is enabled when using the IICA with the IICAjEN bit in the PER0 register being set to 1. This may lead to a through current flowing through the type 12-38-3 pin when the voltage level on this pin is intermediate.**

Remark 1. For multiplexed functions, see **2.1 Functions of Port Pins**.

Remark 2. SAU: Serial array unit

Section 3 CPU Architecture

This MCU features the RL78-S3 CPU core. The CPU core in the RL78-S3 employs a Harvard architecture which has an independent instruction-fetching bus, address bus and data bus. In addition, through the adoption of three-stage pipeline control over fetching, decoding, and memory access, the operation efficiency is markedly improved over that of the previous CPU core. The CPU core features high performance and highly functional instruction processing, and is suitable for use in various applications that require high-speed and high-functionality 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

The following multiply/divide instructions are available only in the RL78-S3 CPU core.
MULHU (unsigned 16-bit multiplication)
MULH (signed 16-bit multiplication)
DIVHU (unsigned 16-bit division)
DIVWU (unsigned 32-bit division)
MACHU (unsigned multiplication/accumulation (16 bits × 16 bits) + 32 bits)
MACH (signed multiplication/accumulation (16 bits × 16 bits) + 32 bits)

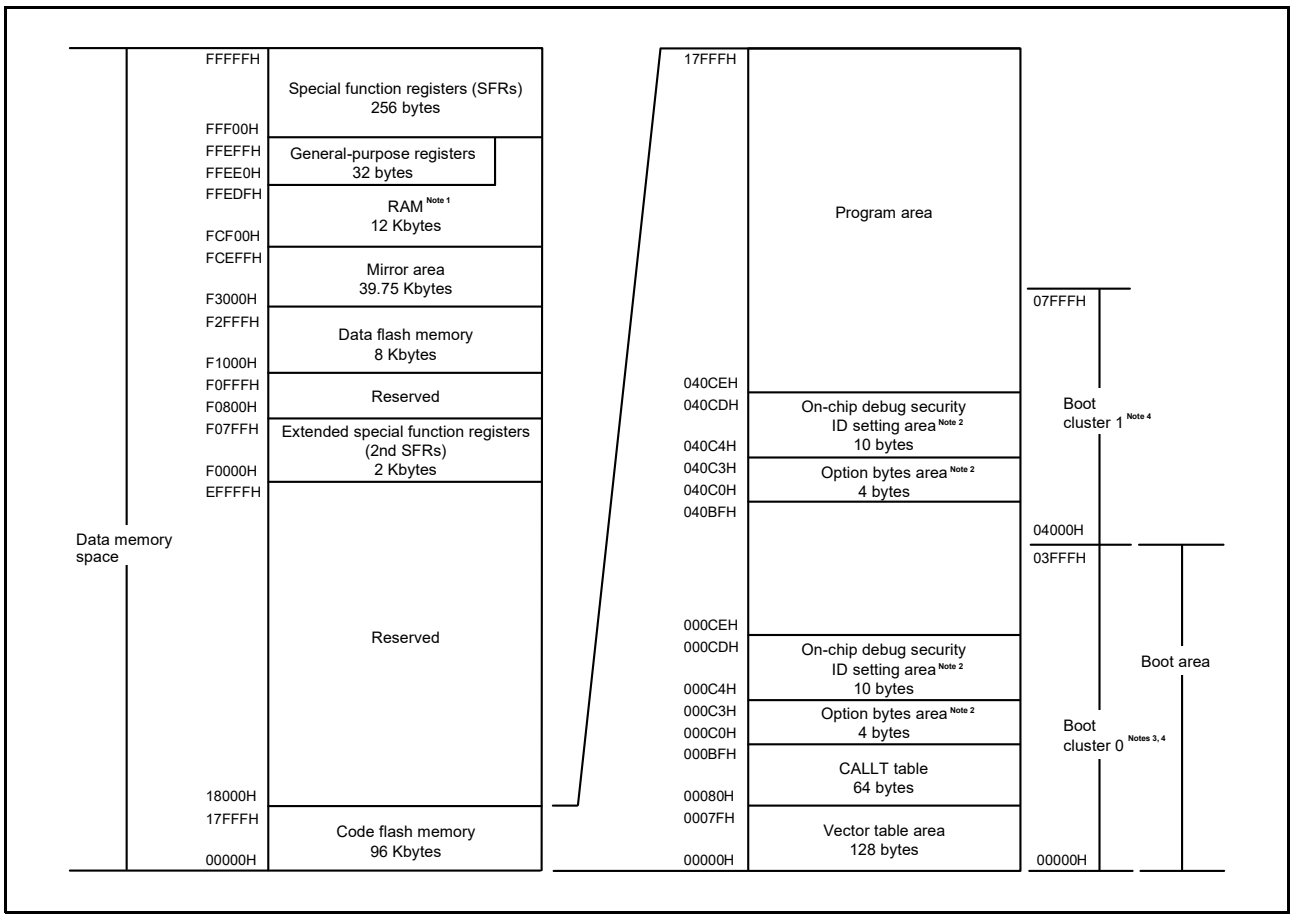
- Data allocation: Little endian

This MCU supports an OCD trace function.

3.1 Memory Space

A 1-Mbyte address space is accessible by this MCU. **Figure 3 - 1** to **Figure 3 - 7** show the memory maps.

Figure 3 - 1 Memory Map of the R7F100GxF (x = A, B, C, E, F, G, J, L) Products



Note 1. Instructions can be executed from the RAM area excluding the general-purpose register area.

Note 2. When boot swapping is not to be used, that is, when the value of the BTFLG bit in the FLSEC register is 1, set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.
When boot swapping is to be used or the value of the BTFLG bit in the FLSEC register is 0, set the option bytes to 000C0H to 000C3H and 040C0H to 040C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 040C4H to 040CDH.

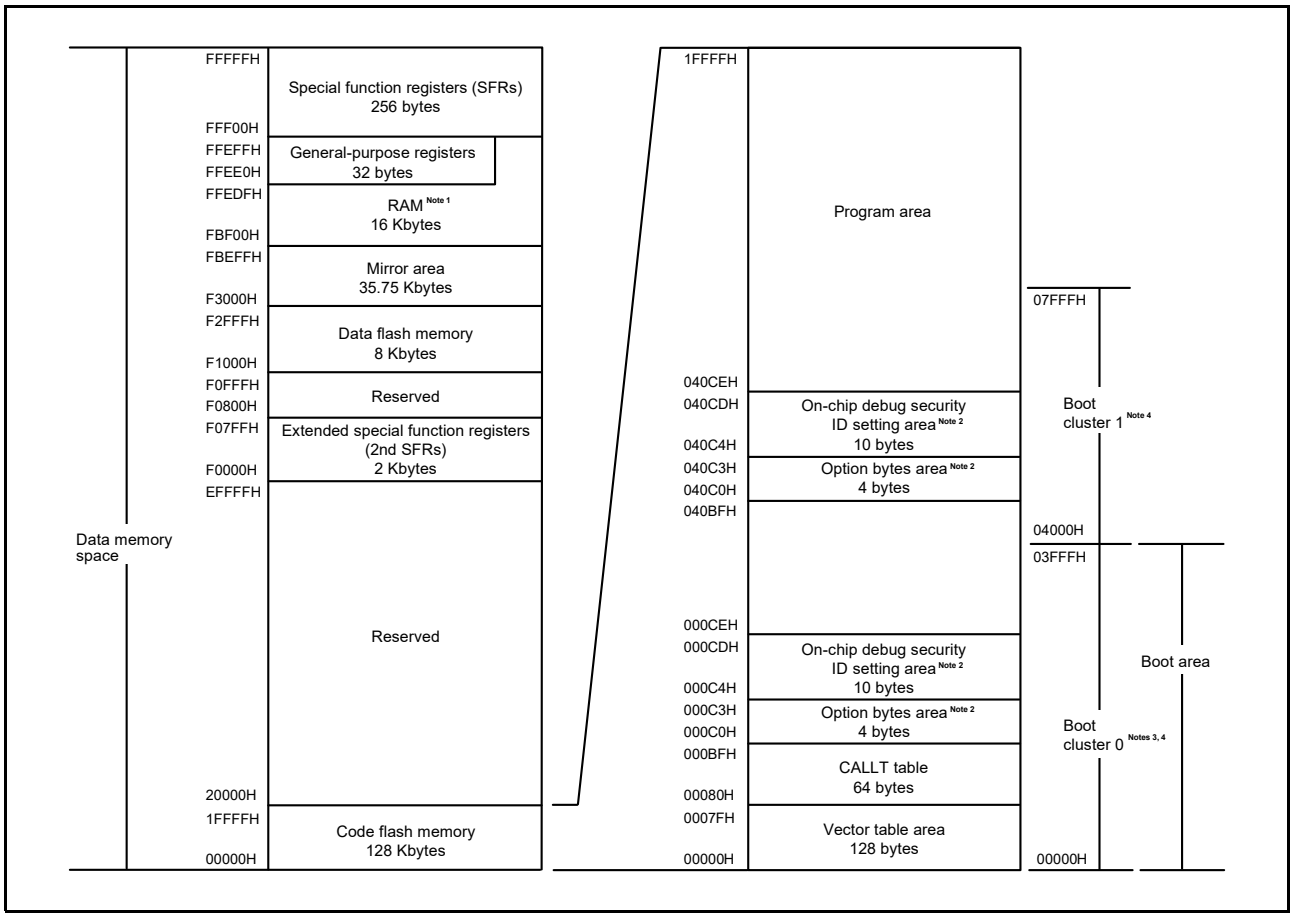
Note 3. Rewriting of the boot area can be prohibited by a security setting. See **33.9 Security Settings**.

Note 4. Boot cluster 0 is selected as the boot area at the time of shipment. When boot swapping is applied, the boot area is swapped between boot cluster 0 and boot cluster 1. See **33.7 Boot Swap Function**.

Caution **While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize a RAM area where data access is to proceed, and a RAM area and an additional 10 bytes beyond it when instructions are to be fetched from the RAM area, respectively. RAM parity error resets become enabled (RPERDIS = 0) following a reset. For details, see 27.3.4 RAM parity error detection.**

Remark The code flash memory area is divided into blocks, with each block being 2 Kbytes. For the correspondence between addresses and block numbers, see **Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory**.

Figure 3 - 2 Memory Map of the R7F100GxG (x = A, B, C, E, F, G, J, L, M, P) and R7F104GBG Products



Note 1. Instructions can be executed from the RAM area excluding the general-purpose register area.

Note 2. When boot swapping is not to be used, that is, when the value of the BTFLG bit in the FLSEC register is 1, set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.
When boot swapping is to be used or the value of the BTFLG bit in the FLSEC register is 0, set the option bytes to 000C0H to 000C3H and 040C0H to 040C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 040C4H to 040CDH.

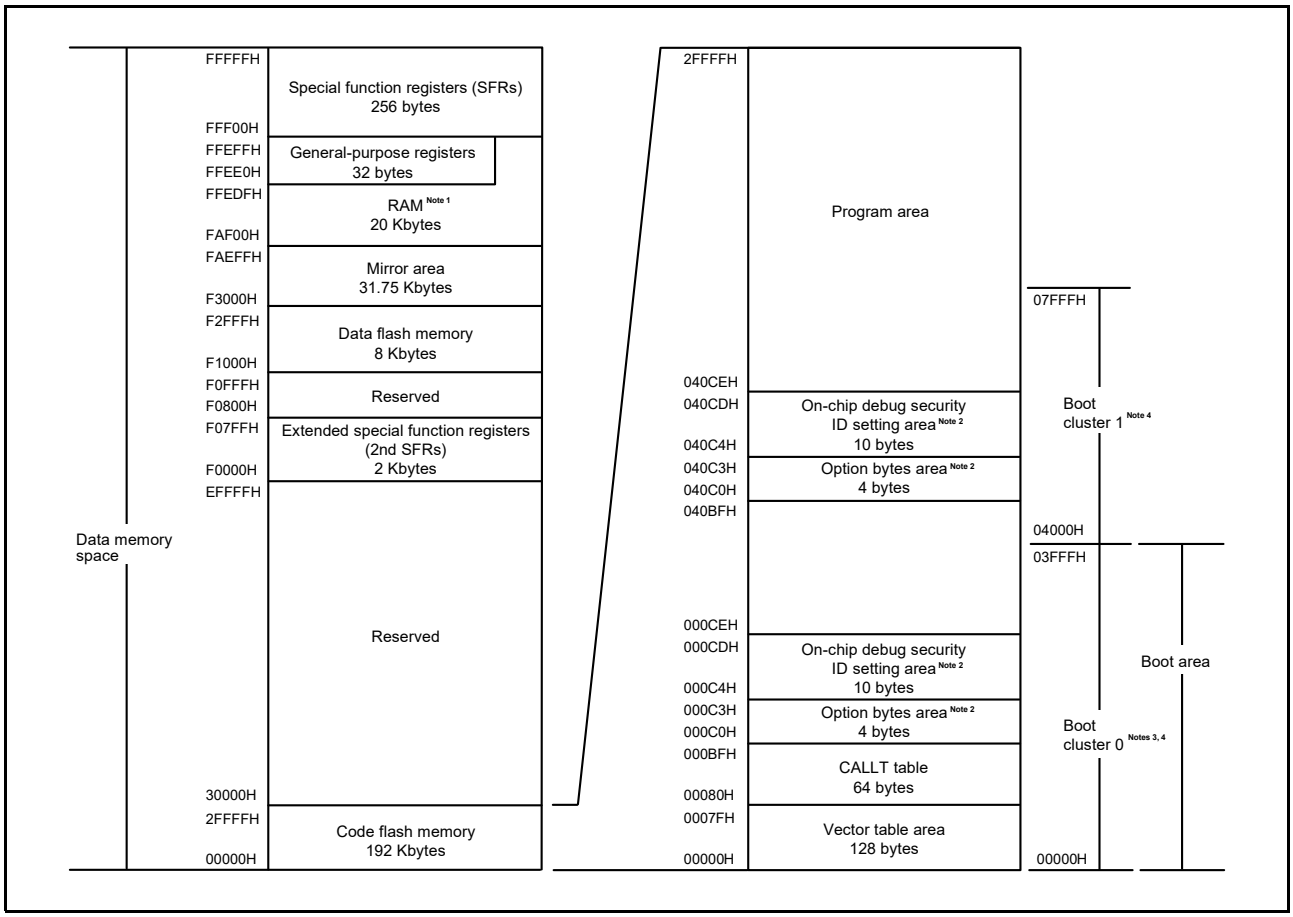
Note 3. Rewriting of the boot area can be prohibited by a security setting. See **33.9 Security Settings**.

Note 4. Boot cluster 0 is selected as the boot area at the time of shipment. When boot swapping is applied, the boot area is swapped between boot cluster 0 and boot cluster 1. See **33.7 Boot Swap Function**.

Caution **While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize a RAM area where data access is to proceed, and a RAM area and an additional 10 bytes beyond it when instructions are to be fetched from the RAM area, respectively. RAM parity error resets become enabled (RPERDIS = 0) following a reset. For details, see 27.3.4 RAM parity error detection.**

Remark The code flash memory area is divided into blocks, with each block being 2 Kbytes. For the correspondence between addresses and block numbers, see **Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory**.

Figure 3 - 3 Memory Map of the R7F100GxH (x = A, B, C, E, F, G, J, L, M, P) Products



Note 1. Instructions can be executed from the RAM area excluding the general-purpose register area.

Note 2. When boot swapping is not to be used, that is, when the value of the BTFLG bit in the FLSEC register is 1, set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.
When boot swapping is to be used or the value of the BTFLG bit in the FLSEC register is 0, set the option bytes to 000C0H to 000C3H and 040C0H to 040C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 040C4H to 040CDH.

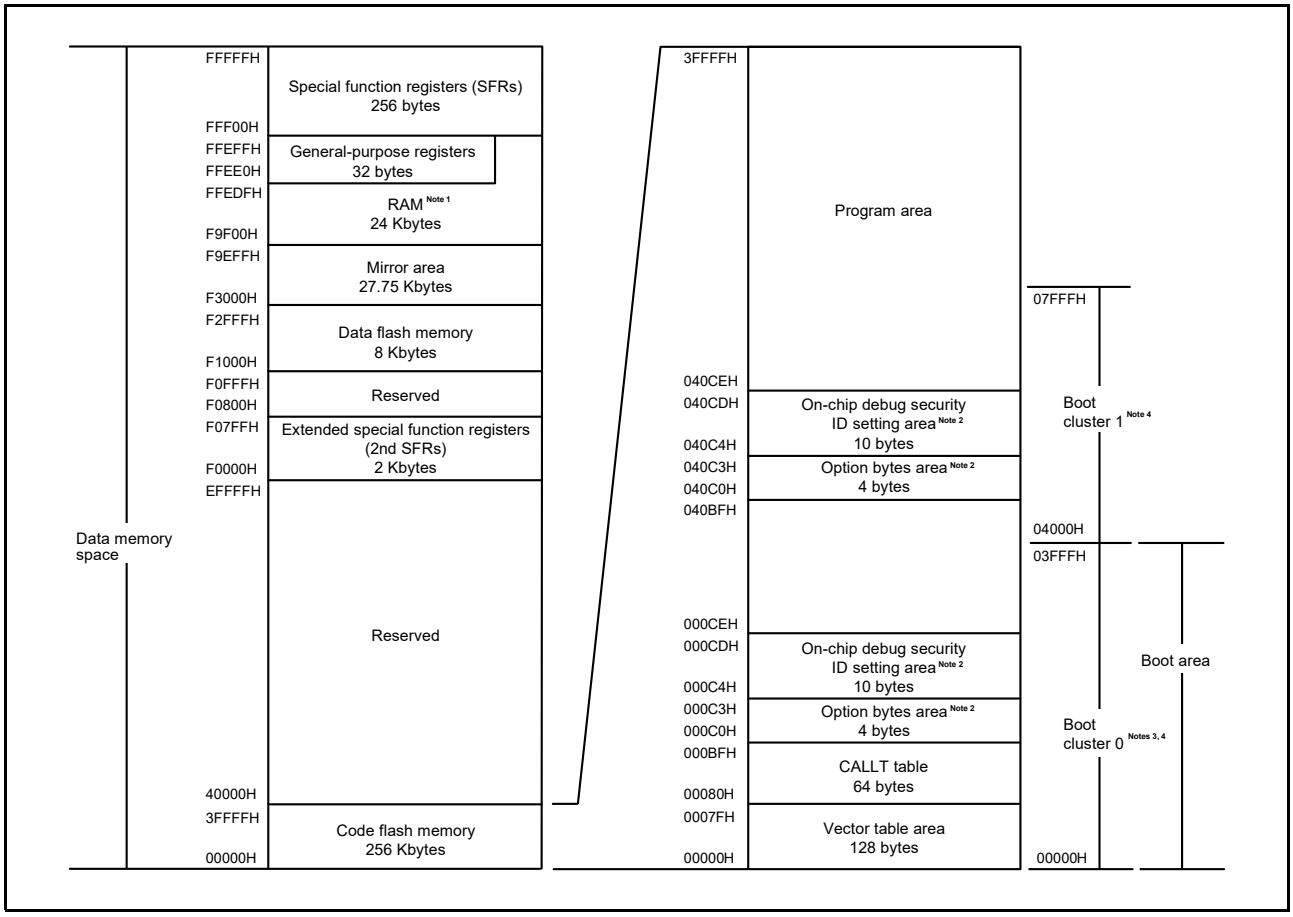
Note 3. Rewriting of the boot area can be prohibited by a security setting. See **33.9 Security Settings**.

Note 4. Boot cluster 0 is selected as the boot area at the time of shipment. When boot swapping is applied, the boot area is swapped between boot cluster 0 and boot cluster 1. See **33.7 Boot Swap Function**.

Caution **While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize a RAM area where data access is to proceed, and a RAM area and an additional 10 bytes beyond it when instructions are to be fetched from the RAM area, respectively. RAM parity error resets become enabled (RPERDIS = 0) following a reset. For details, see 27.3.4 RAM parity error detection.**

Remark The code flash memory area is divided into blocks, with each block being 1 Kbyte. For the correspondence between addresses and block numbers, see **Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory**.

Figure 3 - 4 Memory Map of the R7F100GxJ (x = A, B, C, E, F, G, J, L, M, P, S), R7F104GFJ, and R7F104GGJ Products



Note 1. Instructions can be executed from the RAM area excluding the general-purpose register area.

Note 2. When boot swapping is not to be used, that is, when the value of the BTFLG bit in the FLSEC register is 1, set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.
When boot swapping is to be used or the value of the BTFLG bit in the FLSEC register is 0, set the option bytes to 000C0H to 000C3H and 040C0H to 040C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 040C4H to 040CDH.

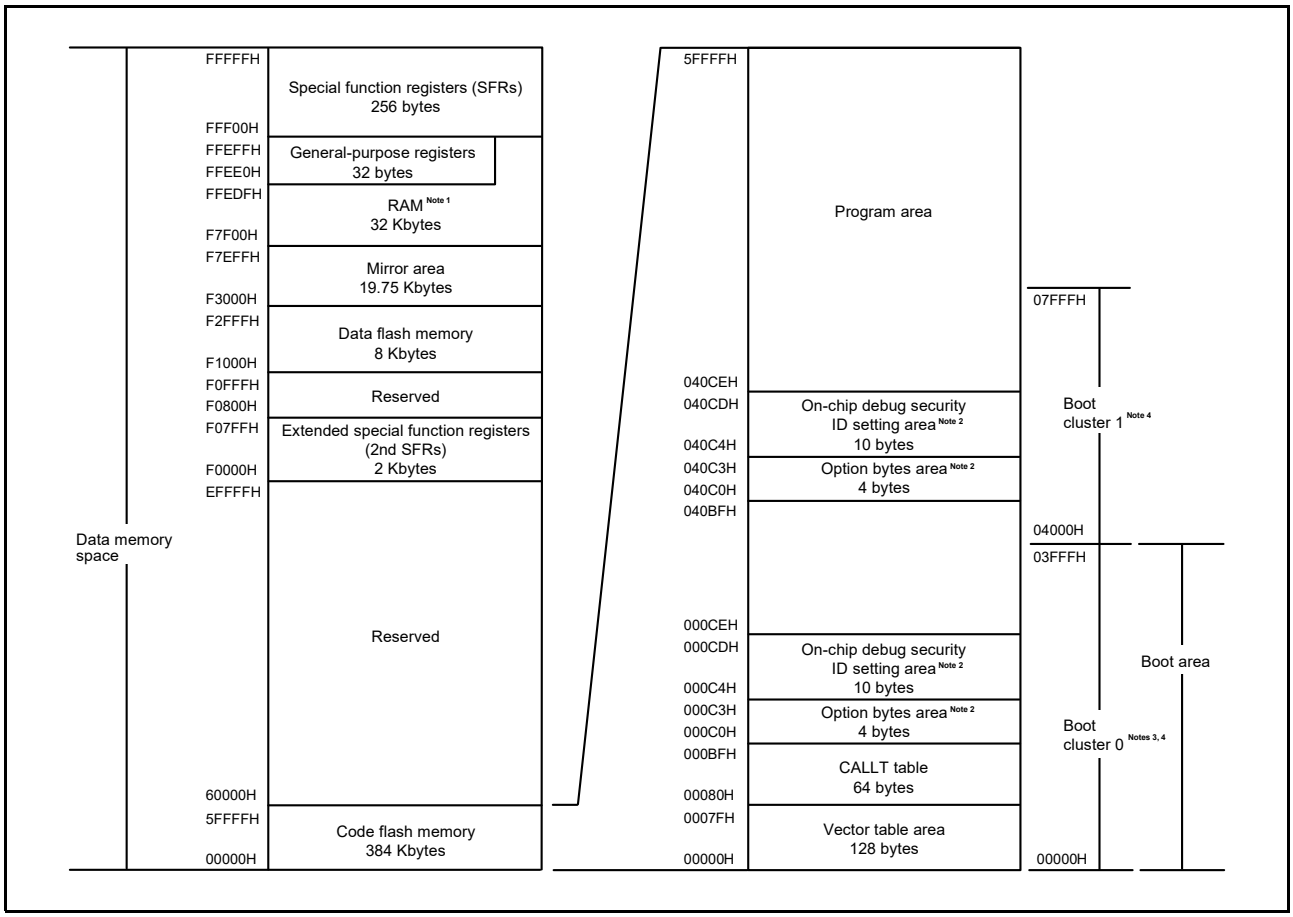
Note 3. Rewriting of the boot area can be prohibited by a security setting. See **33.9 Security Settings**.

Note 4. Boot cluster 0 is selected as the boot area at the time of shipment. When boot swapping is applied, the boot area is swapped between boot cluster 0 and boot cluster 1. See **33.7 Boot Swap Function**.

Caution **While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize a RAM area where data access is to proceed, and a RAM area and an additional 10 bytes beyond it when instructions are to be fetched from the RAM area, respectively. RAM parity error resets become enabled (RPERDIS = 0) following a reset. For details, see 27.3.4 RAM parity error detection.**

Remark The code flash memory area is divided into blocks, with each block being 2 Kbytes. For the correspondence between addresses and block numbers, see **Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory**.

Figure 3 - 5 Memory Map of the R7F100GxK (x = F, G, J, L, M, P, S) Products



Note 1. Instructions can be executed from the RAM area excluding the general-purpose register area.

Note 2. When boot swapping is not to be used, that is, when the value of the BTFLG bit in the FLSEC register is 1, set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.
When boot swapping is to be used or the value of the BTFLG bit in the FLSEC register is 0, set the option bytes to 000C0H to 000C3H and 040C0H to 040C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 040C4H to 040CDH.

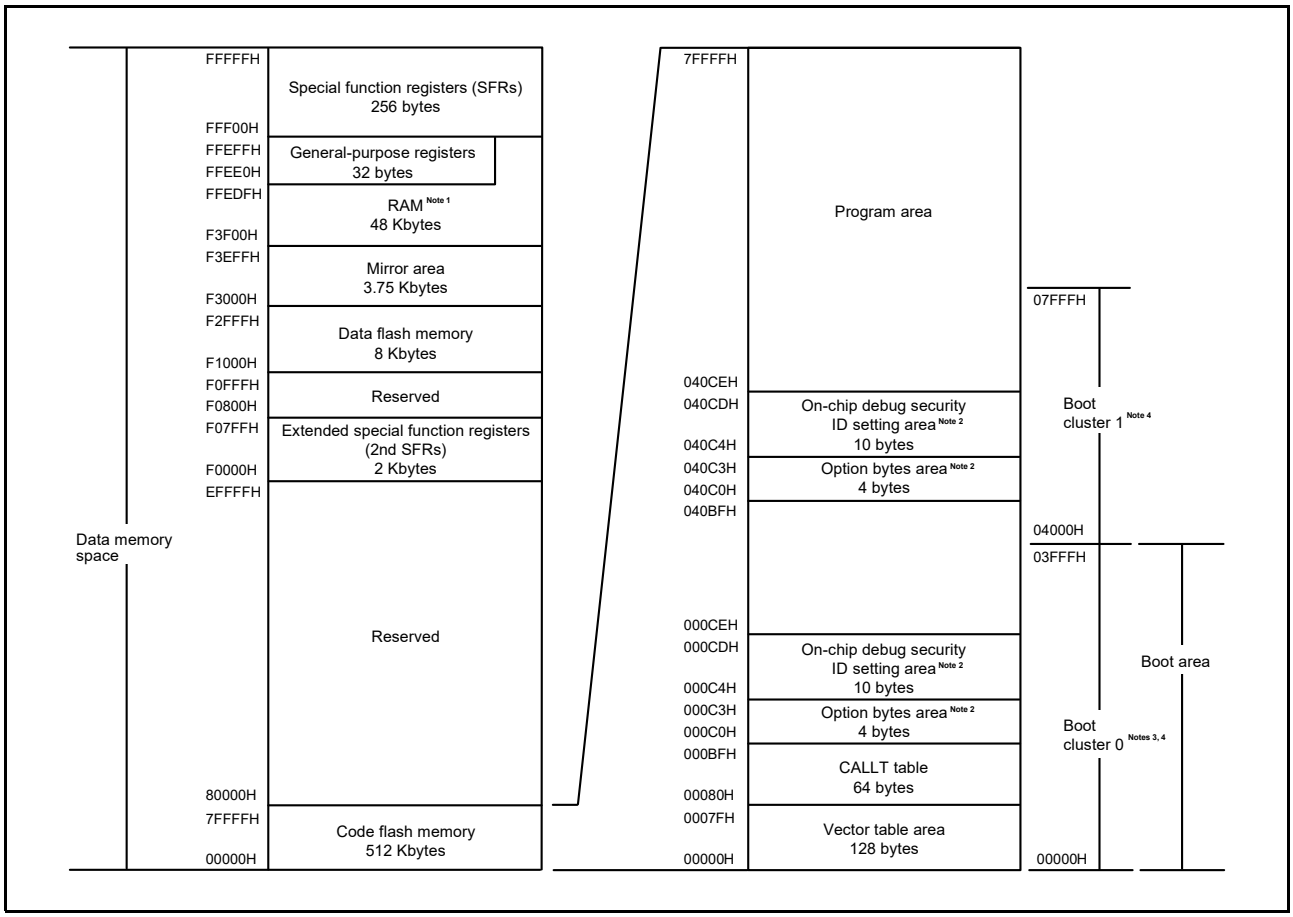
Note 3. Rewriting of the boot area can be prohibited by a security setting. See **33.9 Security Settings**.

Note 4. Boot cluster 0 is selected as the boot area at the time of shipment. When boot swapping is applied, the boot area is swapped between boot cluster 0 and boot cluster 1. See **33.7 Boot Swap Function**.

Caution **While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize a RAM area where data access is to proceed, and a RAM area and an additional 10 bytes beyond it when instructions are to be fetched from the RAM area, respectively. RAM parity error resets become enabled (RPERDIS = 0) following a reset. For details, see 27.3.4 RAM parity error detection.**

Remark The code flash memory area is divided into blocks, with each block being 2 Kbytes. For the correspondence between addresses and block numbers, see **Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory**.

Figure 3 - 6 Memory Map of the R7F100GxL (x = F, G, J, L, M, P, S) Products



Note 1. Instructions can be executed from the RAM area excluding the general-purpose register area.

Note 2. When boot swapping is not to be used, that is, when the value of the BTFLG bit in the FLSEC register is 1, set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.
When boot swapping is to be used or the value of the BTFLG bit in the FLSEC register is 0, set the option bytes to 000C0H to 000C3H and 040C0H to 040C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 040C4H to 040CDH.

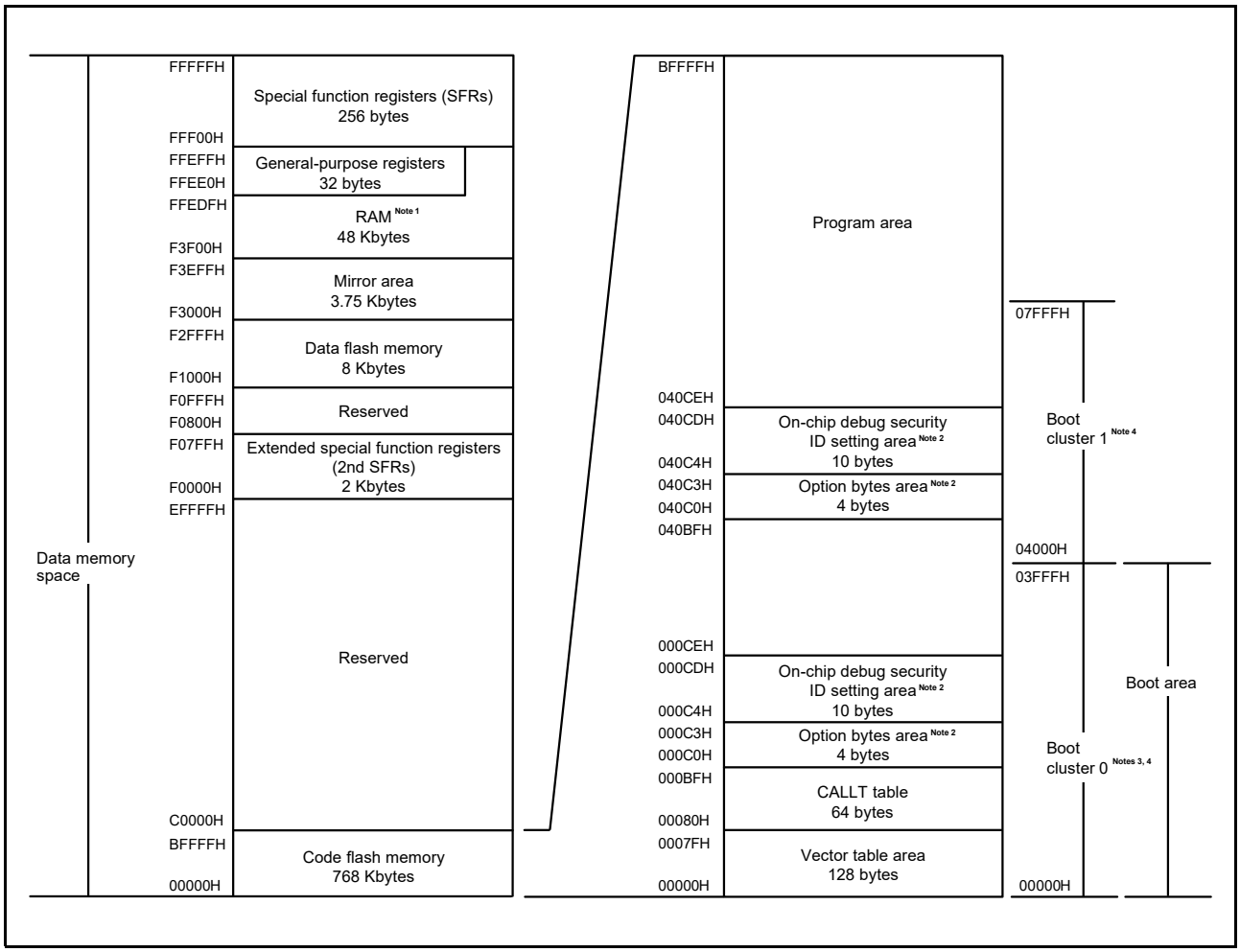
Note 3. Rewriting of the boot area can be prohibited by a security setting. See **33.9 Security Settings**.

Note 4. Boot cluster 0 is selected as the boot area at the time of shipment. When boot swapping is applied, the boot area is swapped between boot cluster 0 and boot cluster 1. See **33.7 Boot Swap Function**.

Caution **While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize a RAM area where data access is to proceed, and a RAM area and an additional 10 bytes beyond it when instructions are to be fetched from the RAM area, respectively. RAM parity error resets become enabled (RPERDIS = 0) following a reset. For details, see 27.3.4 RAM parity error detection.**

Remark The code flash memory area is divided into blocks, with each block being 2 Kbytes. For the correspondence between addresses and block numbers, see **Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory**.

Figure 3 - 7 Memory Map of the R7F100GxN (x = F, G, J, L, M, P, S) Products



Note 1. Instructions can be executed from the RAM area excluding the general-purpose register area.

Note 2. When boot swapping is not to be used, that is, when the value of the BTFLG bit in the FLSEC register is 1, set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.
When boot swapping is to be used or the value of the BTFLG bit in the FLSEC register is 0, set the option bytes to 000C0H to 000C3H and 040C0H to 040C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 040C4H to 040CDH.

Note 3. Rewriting of the boot area can be prohibited by a security setting. See **33.9 Security Settings**.

Note 4. Boot cluster 0 is selected as the boot area at the time of shipment. When boot swapping is applied, the boot area is swapped between boot cluster 0 and boot cluster 1. See **33.7 Boot Swap Function**.

Caution **While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize a RAM area where data access is to proceed, and a RAM area and an additional 10 bytes beyond it when instructions are to be fetched from the RAM area, respectively. RAM parity error resets become enabled (RPERDIS = 0) following a reset. For details, see 27.3.4 RAM parity error detection.**

Remark The code flash memory area is divided into blocks, with each block being 2 Kbytes. For the correspondence between addresses and block numbers, see **Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory**.

Table 3 - 1 shows the correspondence between the addresses and block numbers in the flash memory.

Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory (1/3)

Address	Block Number	Address	Block Number	Address	Block Number	Address	Block Number
00000H to 007FFH	000H	10000H to 107FFH	020H	20000H to 207FFH	040H	30000H to 307FFH	060H
00800H to 00FFFH	001H	10800H to 10FFFH	021H	20800H to 20FFFH	041H	30800H to 30FFFH	061H
01000H to 017FFH	002H	11000H to 117FFH	022H	21000H to 217FFH	042H	31000H to 317FFH	062H
01800H to 01FFFH	003H	11800H to 11FFFH	023H	21800H to 21FFFH	043H	31800H to 31FFFH	063H
02000H to 027FFH	004H	12000H to 127FFH	024H	22000H to 227FFH	044H	32000H to 327FFH	064H
02800H to 02FFFH	005H	12800H to 12FFFH	025H	22800H to 22FFFH	045H	32800H to 32FFFH	065H
03000H to 037FFH	006H	13000H to 137FFH	026H	23000H to 237FFH	046H	33000H to 337FFH	066H
03800H to 03FFFH	007H	13800H to 13FFFH	027H	23800H to 23FFFH	047H	33800H to 33FFFH	067H
04000H to 047FFH	008H	14000H to 147FFH	028H	24000H to 247FFH	048H	34000H to 347FFH	068H
04800H to 04FFFH	009H	14800H to 14FFFH	029H	24800H to 24FFFH	049H	34800H to 34FFFH	069H
05000H to 057FFH	00AH	15000H to 157FFH	02AH	25000H to 257FFH	04AH	35000H to 357FFH	06AH
05800H to 05FFFH	00BH	15800H to 15FFFH	02BH	25800H to 25FFFH	04BH	35800H to 35FFFH	06BH
06000H to 067FFH	00CH	16000H to 167FFH	02CH	26000H to 267FFH	04CH	36000H to 367FFH	06CH
06800H to 06FFFH	00DH	16800H to 16FFFH	02DH	26800H to 26FFFH	04DH	36800H to 36FFFH	06DH
07000H to 077FFH	00EH	17000H to 177FFH	02EH	27000H to 277FFH	04EH	37000H to 377FFH	06EH
07800H to 07FFFH	00FH	17800H to 17FFFH	02FH	27800H to 27FFFH	04FH	37800H to 37FFFH	06FH
08000H to 087FFH	010H	18000H to 187FFH	030H	28000H to 287FFH	050H	38000H to 387FFH	070H
08800H to 08FFFH	011H	18800H to 18FFFH	031H	28800H to 28FFFH	051H	38800H to 38FFFH	071H
09000H to 097FFH	012H	19000H to 197FFH	032H	29000H to 297FFH	052H	39000H to 397FFH	072H
09800H to 09FFFH	013H	19800H to 19FFFH	033H	29800H to 29FFFH	053H	39800H to 39FFFH	073H
0A000H to 0A7FFH	014H	1A000H to 1A7FFH	034H	2A000H to 2A7FFH	054H	3A000H to 3A7FFH	074H
0A800H to 0AFFFH	015H	1A800H to 1AFFFH	035H	2A800H to 2AFFFH	055H	3A800H to 3AFFFH	075H
0B000H to 0B7FFH	016H	1B000H to 1B7FFH	036H	2B000H to 2B7FFH	056H	3B000H to 3B7FFH	076H
0B800H to 0BFFFH	017H	1B800H to 1BFFFH	037H	2B800H to 2BFFFH	057H	3B800H to 3BFFFH	077H
0C000H to 0C7FFH	018H	1C000H to 1C7FFH	038H	2C000H to 2C7FFH	058H	3C000H to 3C7FFH	078H
0C800H to 0CFFFH	019H	1C800H to 1CFFFH	039H	2C800H to 2CFFFH	059H	3C800H to 3CFFFH	079H
0D000H to 0D7FFH	01AH	1D000H to 1D7FFH	03AH	2D000H to 2D7FFH	05AH	3D000H to 3D7FFH	07AH
0D800H to 0DFFFH	01BH	1D800H to 1DFFFH	03BH	2D800H to 2DFFFH	05BH	3D800H to 3DFFFH	07BH
0E000H to 0E7FFH	01CH	1E000H to 1E7FFH	03CH	2E000H to 2E7FFH	05CH	3E000H to 3E7FFH	07CH
0E800H to 0EFFFH	01DH	1E800H to 1EFFFH	03DH	2E800H to 2EFFFH	05DH	3E800H to 3EFFFH	07DH
0F000H to 0F7FFH	01EH	1F000H to 1F7FFH	03EH	2F000H to 2F7FFH	05EH	3F000H to 3F7FFH	07EH
0F800H to 0FFFFH	01FH	1F800H to 1FFFFH	03FH	2F800H to 2FFFFH	05FH	3F800H to 3FFFFH	07FH

Remark R7F100GxF (x = A, B, C, E, F, G, J, L): Block numbers 000H to 02FH
R7F100GxG (x = A, B, C, E, F, G, J, L, M, P): Block numbers 000H to 03FH
R7F100GxH (x = A, B, C, E, F, G, J, L, M, P): Block numbers 000H to 05FH
R7F100GxJ (x = A, B, C, E, F, G, J, L, M, P, S): Block numbers 000H to 07FH
R7F104GxG (x = B): Block numbers 000H to 03FH
R7F104GxJ (x = F, G): Block numbers 000H to 07FH

Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory (2/3)

Address	Block Number	Address	Block Number	Address	Block Number	Address	Block Number
40000H to 407FFH	080H	50000H to 507FFH	0A0H	60000H to 607FFH	0C0H	70000H to 707FFH	0E0H
40800H to 40FFFH	081H	50800H to 50FFFH	0A1H	60800H to 60FFFH	0C1H	70800H to 70FFFH	0E1H
41000H to 417FFH	082H	51000H to 517FFH	0A2H	61000H to 617FFH	0C2H	71000H to 717FFH	0E2H
41800H to 41FFFH	083H	51800H to 51FFFH	0A3H	61800H to 61FFFH	0C3H	71800H to 71FFFH	0E3H
42000H to 427FFH	084H	52000H to 527FFH	0A4H	62000H to 627FFH	0C4H	72000H to 727FFH	0E4H
42800H to 42FFFH	085H	52800H to 52FFFH	0A5H	62800H to 62FFFH	0C5H	72800H to 72FFFH	0E5H
43000H to 437FFH	086H	53000H to 537FFH	0A6H	63000H to 637FFH	0C6H	73000H to 737FFH	0E6H
43800H to 43FFFH	087H	53800H to 53FFFH	0A7H	63800H to 63FFFH	0C7H	73800H to 73FFFH	0E7H
44000H to 447FFH	088H	54000H to 547FFH	0A8H	64000H to 647FFH	0C8H	74000H to 747FFH	0E8H
44800H to 44FFFH	089H	54800H to 54FFFH	0A9H	64800H to 64FFFH	0C9H	74800H to 74FFFH	0E9H
45000H to 457FFH	08AH	55000H to 557FFH	0AAH	65000H to 657FFH	0CAH	75000H to 757FFH	0EAH
45800H to 45FFFH	08BH	55800H to 55FFFH	0ABH	65800H to 65FFFH	0CBH	75800H to 75FFFH	0EBH
46000H to 467FFH	08CH	56000H to 567FFH	0ACH	66000H to 667FFH	0CCH	76000H to 767FFH	0ECH
46800H to 46FFFH	08DH	56800H to 56FFFH	0ADH	66800H to 66FFFH	0CDH	76800H to 76FFFH	0EDH
47000H to 477FFH	08EH	57000H to 577FFH	0AEH	67000H to 677FFH	0CEH	77000H to 777FFH	0EEH
47800H to 47FFFH	08FH	57800H to 57FFFH	0AFH	67800H to 67FFFH	0CFH	77800H to 77FFFH	0EFH
48000H to 487FFH	090H	58000H to 587FFH	0B0H	68000H to 687FFH	0D0H	78000H to 787FFH	0F0H
48800H to 48FFFH	091H	58800H to 58FFFH	0B1H	68800H to 68FFFH	0D1H	78800H to 78FFFH	0F1H
49000H to 497FFH	092H	59000H to 597FFH	0B2H	69000H to 697FFH	0D2H	79000H to 797FFH	0F2H
49800H to 49FFFH	093H	59800H to 59FFFH	0B3H	69800H to 69FFFH	0D3H	79800H to 79FFFH	0F3H
4A000H to 4A7FFH	094H	5A000H to 5A7FFH	0B4H	6A000H to 6A7FFH	0D4H	7A000H to 7A7FFH	0F4H
4A800H to 4AFFFH	095H	5A800H to 5AFFFH	0B5H	6A800H to 6AFFFH	0D5H	7A800H to 7AFFFH	0F5H
4B000H to 4B7FFH	096H	5B000H to 5B7FFH	0B6H	6B000H to 6B7FFH	0D6H	7B000H to 7B7FFH	0F6H
4B800H to 4BFFFH	097H	5B800H to 5BFFFH	0B7H	6B800H to 6BFFFH	0D7H	7B800H to 7BFFFH	0F7H
4C000H to 4C7FFH	098H	5C000H to 5C7FFH	0B8H	6C000H to 6C7FFH	0D8H	7C000H to 7C7FFH	0F8H
4C800H to 4CFFFH	099H	5C800H to 5CFFFH	0B9H	6C800H to 6CFFFH	0D9H	7C800H to 7CFFFH	0F9H
4D000H to 4D7FFH	09AH	5D000H to 5D7FFH	0BAH	6D000H to 6D7FFH	0DAH	7D000H to 7D7FFH	0FAH
4D800H to 4DFFFH	09BH	5D800H to 5DFFFH	0BBH	6D800H to 6DFFFH	0DBH	7D800H to 7DFFFH	0FBH
4E000H to 4E7FFH	09CH	5E000H to 5E7FFH	0BCH	6E000H to 6E7FFH	0DCH	7E000H to 7E7FFH	0FCH
4E800H to 4EFFFH	09DH	5E800H to 5EFFFH	0BDH	6E800H to 6EFFFH	0DDH	7E800H to 7EFFFH	0FDH
4F000H to 4F7FFH	09EH	5F000H to 5F7FFH	0BEH	6F000H to 6F7FFH	0DEH	7F000H to 7F7FFH	0FEH
4F800H to 4FFFFH	09FH	5F800H to 5FFFFH	0BFH	6F800H to 6FFFFH	0DFH	7F800H to 7FFFFH	0FFH

Remark R7F100GxK (x = F, G, J, L, M, P, S): Block numbers 000H to 0BFH
R7F100GxL (x = F, G, J, L, M, P, S): Block numbers 000H to 0FFH

Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory (3/3)

Address	Block Number	Address	Block Number	Address	Block Number	Address	Block Number
80000H to 807FFH	100H	90000H to 907FFH	120H	A0000H to A07FFH	140H	B0000H to B07FFH	160H
80800H to 80FFFH	101H	90800H to 90FFFH	121H	A0800H to A0FFFH	141H	B0800H to B0FFFH	161H
81000H to 817FFH	102H	91000H to 917FFH	122H	A1000H to A17FFH	142H	B1000H to B17FFH	162H
81800H to 81FFFH	103H	91800H to 91FFFH	123H	A1800H to A1FFFH	143H	B1800H to B1FFFH	163H
82000H to 827FFH	104H	92000H to 927FFH	124H	A2000H to A27FFH	144H	B2000H to B27FFH	164H
82800H to 82FFFH	105H	92800H to 92FFFH	125H	A2800H to A2FFFH	145H	B2800H to B2FFFH	165H
83000H to 837FFH	106H	93000H to 937FFH	126H	A3000H to A37FFH	146H	B3000H to B37FFH	166H
83800H to 83FFFH	107H	93800H to 93FFFH	127H	A3800H to A3FFFH	147H	B3800H to B3FFFH	167H
84000H to 847FFH	108H	94000H to 947FFH	128H	A4000H to A47FFH	148H	B4000H to B47FFH	168H
84800H to 84FFFH	109H	94800H to 94FFFH	129H	A4800H to A4FFFH	149H	B4800H to B4FFFH	169H
85000H to 857FFH	10AH	95000H to 957FFH	12AH	A5000H to A57FFH	14AH	B5000H to B57FFH	16AH
85800H to 85FFFH	10BH	95800H to 95FFFH	12BH	A5800H to A5FFFH	14BH	B5800H to B5FFFH	16BH
86000H to 867FFH	10CH	96000H to 967FFH	12CH	A6000H to A67FFH	14CH	B6000H to B67FFH	16CH
86800H to 86FFFH	10DH	96800H to 96FFFH	12DH	A6800H to A6FFFH	14DH	B6800H to B6FFFH	16DH
87000H to 877FFH	10EH	97000H to 977FFH	12EH	A7000H to A77FFH	14EH	B7000H to B77FFH	16EH
87800H to 87FFFH	10FH	97800H to 97FFFH	12FH	A7800H to A7FFFH	14FH	B7800H to B7FFFH	16FH
88000H to 887FFH	110H	98000H to 987FFH	130H	A8000H to A87FFH	150H	B8000H to B87FFH	170H
88800H to 88FFFH	111H	98800H to 98FFFH	131H	A8800H to A8FFFH	151H	B8800H to B8FFFH	171H
89000H to 897FFH	112H	99000H to 997FFH	132H	A9000H to A97FFH	152H	B9000H to B97FFH	172H
89800H to 89FFFH	113H	99800H to 99FFFH	133H	A9800H to A9FFFH	153H	B9800H to B9FFFH	173H
8A000H to 8A7FFH	114H	9A000H to 9A7FFH	134H	AA000H to AA7FFH	154H	BA000H to BA7FFH	174H
8A800H to 8AFFFH	115H	9A800H to 9AFFFH	135H	AA800H to AAFFFH	155H	BA800H to BAFFFH	175H
8B000H to 8B7FFH	116H	9B000H to 9B7FFH	136H	AB000H to AB7FFH	156H	BB000H to BB7FFH	176H
8B800H to 8BFFFH	117H	9B800H to 9BFFFH	137H	AB800H to ABFFFH	157H	BB800H to BBFFFH	177H
8C000H to 8C7FFH	118H	9C000H to 9C7FFH	138H	AC000H to AC7FFH	158H	BC000H to BC7FFH	178H
8C800H to 8CFFFH	119H	9C800H to 9CFFFH	139H	AC800H to ACFFFH	159H	BC800H to BCFFFH	179H
8D000H to 8D7FFH	11AH	9D000H to 9D7FFH	13AH	AD000H to AD7FFH	15AH	BD000H to BD7FFH	17AH
8D800H to 8DFFFH	11BH	9D800H to 9DFFFH	13BH	AD800H to ADFFFH	15BH	BD800H to BDFFFH	17BH
8E000H to 8E7FFH	11CH	9E000H to 9E7FFH	13CH	AE000H to AE7FFH	15CH	BE000H to BE7FFH	17CH
8E800H to 8EFFFH	11DH	9E800H to 9EFFFH	13DH	AE800H to AEFFFH	15DH	BE800H to BEFFFH	17DH
8F000H to 8F7FFH	11EH	9F000H to 9F7FFH	13EH	AF000H to AF7FFH	15EH	BF000H to BF7FFH	17EH
8F800H to 8FFFFH	11FH	9F800H to 9FFFFH	13FH	AF800H to AFFFFH	15FH	BF800H to BFFFFH	17FH

Remark R7F100GxN (x = F, G, J, L, M, P, S): Block numbers 000H to 17FH

3.1.1 Internal program memory space

The internal program memory holds the program and table data. This MCU incorporates the internal ROM (flash memory) capacities listed in **Table 3 - 2**.

Table 3 - 2 Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
R7F100GxF (x = A, B, C, E, F, G, J, L)	Flash memory	98304 × 8 bits (00000H to 17FFFH)
R7F100GxG (x = A, B, C, E, F, G, J, L, M, P)		131072 × 8 bits (00000H to 1FFFFH)
R7F104GxG (x = B)		
R7F100GxH (x = A, B, C, E, F, G, J, L, M, P)		196608 × 8 bits (00000H to 2FFFFH)
R7F100GxJ (x = A, B, C, E, F, G, J, L, M, P, S)		262144 × 8 bits (00000H to 3FFFFH)
R7F104GxJ (x = F, G)		
R7F100GxK (x = F, G, J, L, M, P, S)		393216 × 8 bits (00000H to 5FFFFH)
R7F100GxL (x = F, G, J, L, M, P, S)		524288 × 8 bits (00000H to 7FFFFH)
R7F100GxN (x = F, G, J, L, M, P, S)		786432 × 8 bits (00000H to BFFFFH)

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. The vector table address can be changed to an address in RAM when self-programming is to proceed. For details, see **33.6.2.18 Interrupt vector change registers 0 and 1 (FLSIVC0, FLSIVC1)**.

Table 3 - 3 Vector Table (1/2)

Vector Table Address	Interrupt Source	R7F100G											R7F104G		
		128-pin	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin	40-pin	36-pin	32-pin	30-pin	48-pin	44-pin	32-pin
00000H	RESET, POR, LVD, WDT, TRAP, IAW, RPE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00004H	INTWDTI	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00006H	INTLVI	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00008H	INTP0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0000AH	INTP1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0000CH	INTP2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—
0000EH	INTP3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00010H	INTP4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00012H	INTP5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00014H	INTST2/INTCSI20/ INTIIC20	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	**Note 4**	**Note 4**	**Note 4**
00016H	INTSR2/INTCSI21/ INTIIC21	✓	✓	✓	✓	✓	✓	✓	✓	✓	**Note 1**	**Note 1**	✓	✓	✓
00018H	INTSRE2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	INTTM11H	✓	✓	✓	—	—	—	—	—	—	—	—	✓	✓	—
0001AH	INTELCL	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0001CH	INTSMSE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0001EH	INTST0/INTCSI00/ INTIIC00	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00020H	INTTM00	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00022H	INTSRE0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	INTTM01H	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00024H	INTST1/INTCSI10/ INTIIC10	✓	✓	✓	✓	**Note 2**	**Note 2**	**Note 2**	**Note 2**	**Note 2**	**Note 2**	**Note 2**	**Note 2**	**Note 2**	**Note 2**
00026H	INTSR1/INTCSI11/ INTIIC11	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	**Note 5**	**Note 5**	**Note 5**
00028H	INTSRE1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	INTTM03H	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0002AH	INTIICA0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
0002CH	INTSR0/INTCSI01/ INTIIC01	✓	✓	✓	✓	✓	✓	**Note 3**	**Note 3**	**Note 3**	**Note 3**	**Note 3**	**Note 3**	**Note 3**	**Note 3**
0002EH	INTTM01	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00030H	INTTM02	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00032H	INTTM03	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00034H	INTAD	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00036H	INTRTC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00038H	INTITL	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0003AH	INTKR	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—	✓	✓	✓
0003CH	INTST3/INTCSI30/ INTIIC30	✓	✓	✓	—	—	—	—	—	—	—	—	—	—	—
0003EH	INTSR3/INTCSI31/ INTIIC31	✓	✓	✓	—	—	—	—	—	—	—	—	—	—	—
00040H	INTTM13	✓	✓	✓	—	—	—	—	—	—	—	—	✓	✓	—
00042H	INTTM04	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 3 - 3 Vector Table (2/2)

Vector Table Address	Interrupt Source	R7F100G											R7F104G		
		128-pin	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin	40-pin	36-pin	32-pin	30-pin	48-pin	44-pin	32-pin
00044H	INTTM05	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00046H	INTTM06	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00048H	INTTM07	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0004AH	INTP6	✓	✓	✓	✓	✓	✓	—	—	—	—	—	—	—	—
0004CH	INTP7	✓	✓	✓	✓	—	—	—	—	—	—	—	—	—	—
0004EH	INTP8	✓	✓	✓	✓	✓	✓	—	—	—	—	—	—	—	—
00050H	INTP9	✓	✓	✓	✓	✓	✓	—	—	—	—	—	—	—	—
00052H	INTFL	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00054H	INTP10	✓	✓	✓	✓	✓	—	—	—	—	—	—	✓	✓	✓
	INTCMP0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00056H	INTP11	✓	✓	✓	✓	✓	—	—	—	—	—	—	✓	✓	✓
	INTCMP1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—
00058H	INTURE0	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—	✓	✓	✓
	INTTM10	✓	✓	✓	—	—	—	—	—	—	—	—	✓	✓	—
0005AH	INTURE1	✓	✓	✓	✓	✓	✓	✓	—	—	—	—	—	—	—
	INTTM11	✓	✓	✓	—	—	—	—	—	—	—	—	✓	✓	—
0005CH	INTTM12	✓	✓	✓	—	—	—	—	—	—	—	—	✓	✓	—
0005EH	INTSRE3	✓	✓	✓	—	—	—	—	—	—	—	—	—	—	—
	INTTM13H	✓	✓	✓	—	—	—	—	—	—	—	—	✓	✓	—
00060H	INTCTSUWR	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00062H	INTIICA1	✓	✓	✓	✓	✓	✓	✓	—	—	—	—	—	—	—
00064H	INTCTSURD	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00066H	INTCTSUFN	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
00068H	INTREMC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	✓	✓	✓
0006AH	INTUT0	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—	✓	✓	✓
0006CH	INTUR0	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—	✓	✓	✓
0006EH	INTUT1	✓	✓	✓	✓	✓	✓	✓	—	—	—	—	—	—	—
00070H	INTUR1	✓	✓	✓	✓	✓	✓	✓	—	—	—	—	—	—	—
00072H	INTTM14	✓	—	—	—	—	—	—	—	—	—	—	—	—	—
00074H	INTTM15	✓	—	—	—	—	—	—	—	—	—	—	—	—	—
00076H	INTTM16	✓	—	—	—	—	—	—	—	—	—	—	—	—	—
00078H	INTTM17	✓	—	—	—	—	—	—	—	—	—	—	—	—	—
0007AH	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
0007CH	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
0007EH	BRK	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Note 1. INTSR2 is only present in this product.

Note 2. INTST1 is only present in this product.

Note 3. INTSR0 is only present in this product.

Note 4. INTST2 is only present in this product.

Note 5. INTSR1 and INTIIC11 are only present in this product.

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.

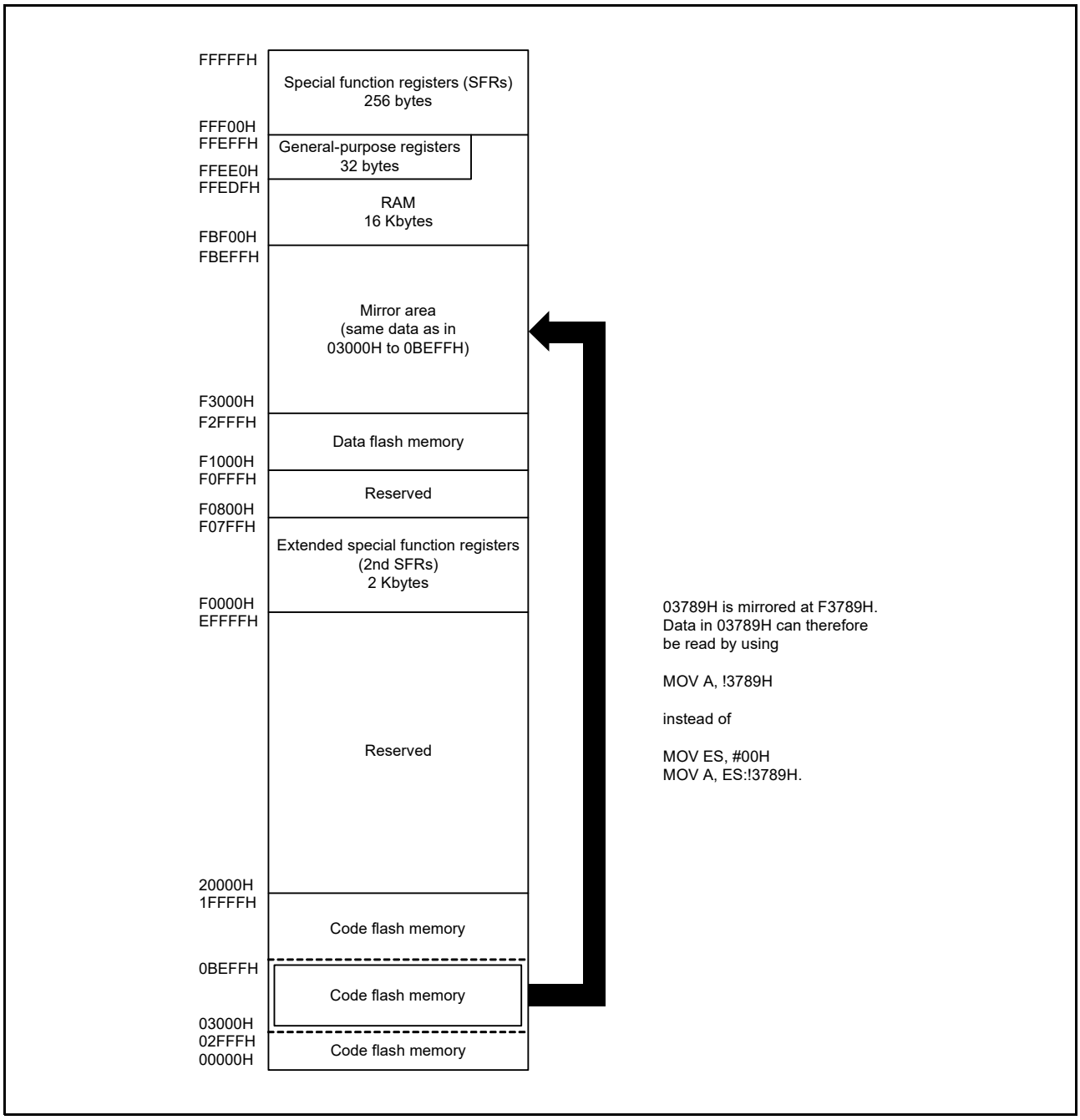
3. Option bytes area
   A 4-byte area of 000C0H to 000C3H can be used as an option bytes area. Set the option byte at 040C0H to 040C3H when boot swapping is to be used or the value of the BTFLG bit in the FLSEC register is 0. For details, see **Section 32 Option Bytes**.

4. On-chip debug security ID setting area
   A 10-byte area of 000C4H to 000CDH and 040C4H to 040CDH can be used as an on-chip debug security ID setting area. Set the 10-byte security ID for on-chip debugging at 000C4H to 000CDH when boot swapping is not to be used, that is, the value of the BTFLG bit in the FLSEC register is 1, and at both 000C4H to 000CDH and 040C4H to 040CDH when boot swapping is to be used or the value of the BTFLG bit in the FLSEC register is 0. For details, see **Section 34 On-chip Debugging**.

3.1.2 Mirror area

In this MCU, the code flash memory area from 00000H to 0FFFFH or 10000H to 1FFFFH is mirrored at F0000H to FFFFFH. The setting of the effective bit of the processor mode control register (PMC) determines which range of the code flash memory area is to be mirrored. 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 memory can be read with the shorter code. However, the code flash memory area is not mirrored to the special function registers (SFRs), extended special function registers (2nd SFRs), 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: R7F100GxG (x = A, B, C, E, F, G, J, L, M, P) products with 128-Kbyte flash memory and 16-Kbyte RAM



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. The value of this register following a reset is 00H.

Figure 3 - 8 Format of Processor Mode Control Register (PMC)

Address: FFFFEH
After reset: 00H
R/W: 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

Caution 1. After setting the PMC register, wait for at least one instruction and access the mirror area.
Caution 2. When boot swapping is executed while the MAA bit is 0, data at 03000H to 07FFFH are mirrored at F3000H to F7FFFH after boot swapping.

3.1.3 Internal data memory space

This MCU incorporates RAM with the capacities listed in **Table 3 - 4**.

Table 3 - 4 Internal RAM Capacity

Part Number	Internal RAM
R7F100GxF (x = A, B, C, E, F, G, J, L)	12288 × 8 bits (FCF00H to FFEFFH)
R7F100GxG (x = A, B, C, E, F, G, J, L, M, P)	16384 × 8 bits (FBF00H to FFEFFH)
R7F104GxG (x = B)	
R7F100GxH (x = A, B, C, E, F, G, J, L, M, P)	20480 × 8 bits (FAF00H to FFEFFH)
R7F100GxJ (x = A, B, C, E, F, G, J, L, M, P, S)	24576 × 8 bits (F9F00H to FFEFFH)
R7F104GxJ (x = F, G)	
R7F100GxK (x = F, G, J, L, M, P, S)	32768 × 8 bits (F7F00H to FFEFFH)
R7F100GxL (x = F, G, J, L, M, P, S)	49152 × 8 bits (F3F00H to FFEFFH)
R7F100GxN (x = F, G, J, L, M, P, S)	49152 × 8 bits (F3F00H to FFEFFH)

The internal RAM can be used as a data area and a program area where instructions are fetched (the space to which the general-purpose registers are allocated cannot be used for instruction fetching). 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. The internal RAM is used as stack memory. Some areas in the internal memory can be placed in shutdown mode for low-power operation. Data cannot be retained in the areas placed in shutdown mode. For details, see **23.2.2 Memory power reduction control register (PSMCR)**.

Caution 1. The space to which the general-purpose registers are allocated (FFEE0H to FFEFFH) cannot be used for instruction fetching or the stack.

Caution 2. The use of the RAM area in the following products is prohibited when the trace function of on-chip debugging is in use.
R7F100GxG (x = A, B, C, E, F, G, J, L, M, P): FC300H to FC6FFH
R7F100GxJ (x = A, B, C, E, F, G, J, L, M, P, S): FA300H to FA6FFH
R7F100GxL, R7F100GxN (x = F, G, J, L, M, P, S): F4300H to F46FFH
R7F104GxG (x = B): FC300H to FC6FFH
R7F104GxJ (x = F, G): FA300H to FA6FFH

3.1.4 Special function register (SFR) area

The special function registers (SFRs) in the on-chip peripheral modules are allocated in the area FFF00H to FFFFFH (see **Table 3 - 5 List of Special Function Registers (SFRs)** in **3.2.4 Special function registers (SFRs)**).

Caution Only access the addresses to which SFRs are assigned.

3.1.5 Extended special function register (2nd SFR: 2nd special function register) area

The extended special function registers (2nd SFRs) in the on-chip peripheral modules are allocated in the area F0000H to F07FFH (see **Table 3 - 6 List of Extended Special Function Registers (2nd SFRs)** in **3.2.5 Extended special function registers (2nd SFRs)**).

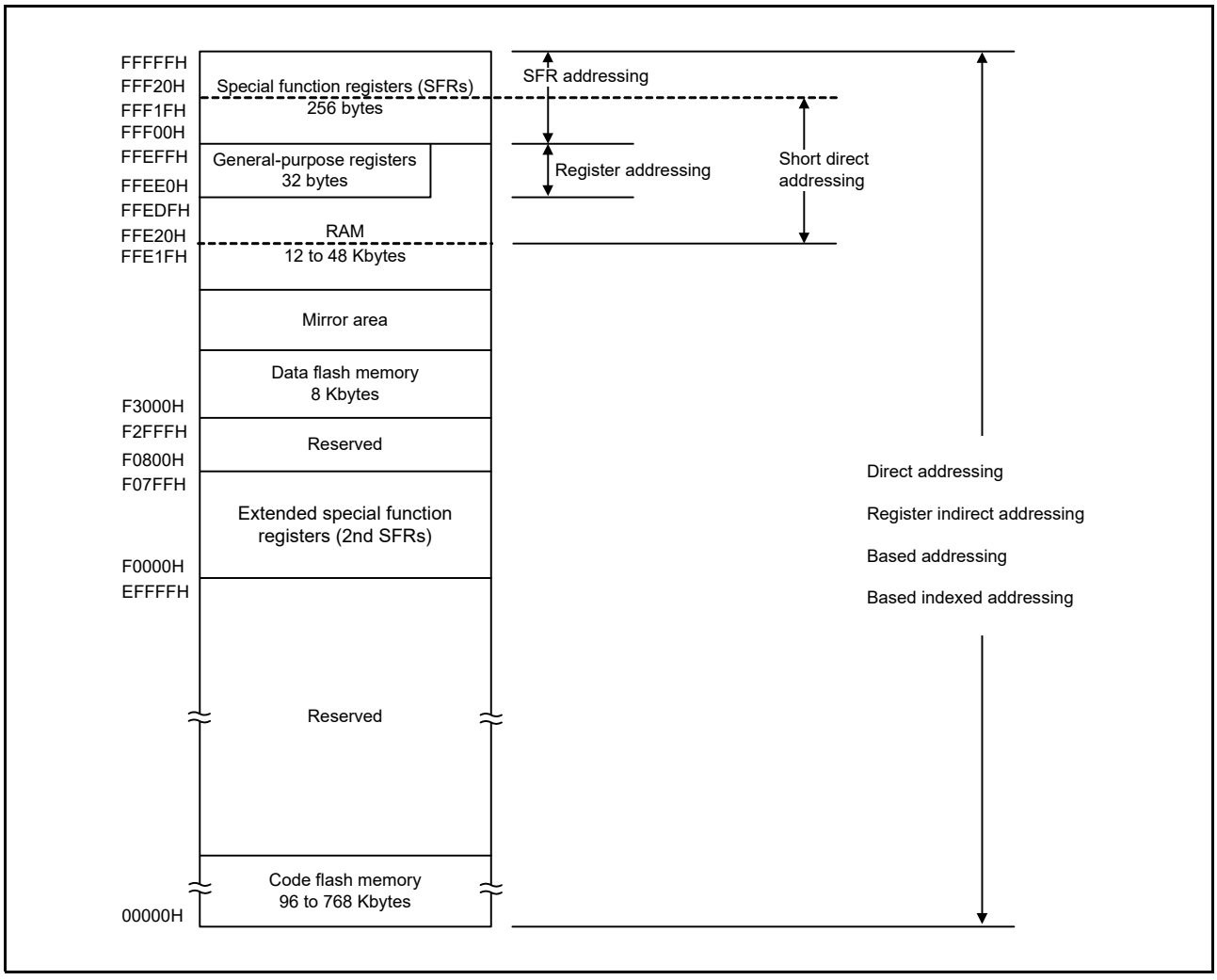
Caution 1. Only access the addresses to which 2nd SFRs are assigned.

Caution 2. In the area for the 2nd SFRs, the registers of the capacitive sensing unit (CTSU2L) are allocated to the address range from F0500H to F0535H and the registers of the true random number generator (TRNG) are allocated to F0540H and F0542H. The CPU is placed in the wait state and does not proceed to the next instruction during access to registers for either module. Accordingly, the CPU entering this state lengthens the number of clock cycles to execute an instruction by the number of cycles of waiting. Specifically, the CPU waits for one clock cycle during access (whether reading or writing) to the registers of the capacitive sensing unit (CTSU2L) and true random number generator (TRNG).

3.1.6 Data memory addressing

<R> Addressing refers to the method of specifying the addresses of next instructions to be executed or the addresses of the registers or memory locations relevant to the execution of the instructions. This MCU provides a wide range of addressing modes to ease programming. For areas of memory containing data in particular, special addressing methods designed to apply the functions of special function registers (SFRs) and general-purpose registers are available for use. **Figure 3 - 9** shows the correspondence between data memory and types of addressing. For details of each type of addressing, see **3.4 Types of Addressing for Processing Data Addresses**.

Figure 3 - 9 Correspondence Between Data Memory and Types of Addressing



3.2 Processor Registers

This MCU incorporates the following processor registers.

3.2.1 Control registers

The control registers control the program sequence, state, 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. The values of the reset vector table at addresses 00000H and 00001H are set in the program counter following a reset.

Figure 3 - 10 Format of Program Counter

Symbol	19																			0
PC																				

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. The value of the PSW following a reset is 06H.

Figure 3 - 11 Format of Program Status Word

Symbol	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 to 0 upon DI instruction execution or interrupt acknowledgment and is set to 1 upon EI instruction execution.

b) Zero flag (Z)
   When the operation result is zero, this flag is set to 1. It is reset to 0 in all other cases.

c) Register bank select flags (RBS1, RBS0)
   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 to 1. It is reset to 0 in all other cases.

e) In-service priority flags (ISP1, ISP0)
These flags manage the priority of acknowledgeable maskable vectored interrupts. Vectored interrupt requests specified lower than the value of ISP1 and ISP0 flags by the priority specification flag registers (PRn0L, PRn0H, PRn1L, PRn1H, PRn2L, PRn2H, PRn3L, and PRn3H; see **21.3.3 Priority specification flag registers**) cannot 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 - 12 Format of Stack Pointer

Symbol	15	14	13	12	11	10	9	8
SP	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8

7	6	5	4	3	2	1	0
SP7	SP6	SP5	SP4	SP3	SP2	SP1	0

In stack addressing through a stack pointer, the SP is decremented ahead of write (save) to the stack memory and is incremented after read (restore) from the stack memory.

Caution 1. Since the contents of the SP become undefined following a reset, be sure to initialize the SP before using the stack.

Caution 2. The space to which the general-purpose registers are allocated (FFEE0H to FFEFFH) cannot be used for instruction fetching or the stack.

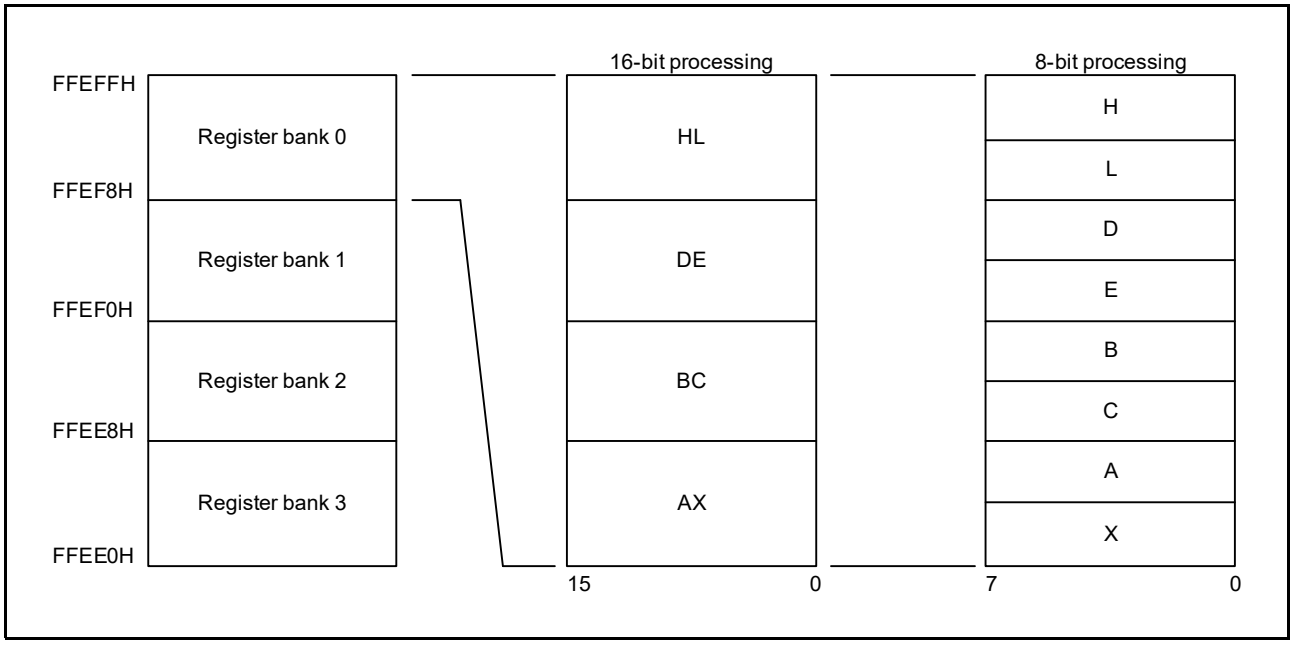
Caution 3. The internal RAM area in the following products cannot be used as stack memory when using the on-chip debugging trace function.
R7F100GxG (x = A, B, C, E, F, G, J, L, M, P): FC300H to FC6FFH
R7F100GxJ (x = A, B, C, E, F, G, J, L, M, P, S): FA300H to FA6FFH
R7F100GxL, R7F100GxN (x = F, G, J, L, M, P, S): F4300H to F46FFH
R7F104GxG (x = B): FC300H to FC6FFH
R7F104GxJ (x = F, G): FA300H to FA6FFH

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). 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 interrupt processing for each bank.

Caution The space to which the general-purpose registers are allocated (FFEE0H to FFEFFH) cannot be used for instruction fetching or the stack.

Figure 3 - 13 Configuration of General-purpose Registers (Function Name)



3.2.3 ES and CS registers

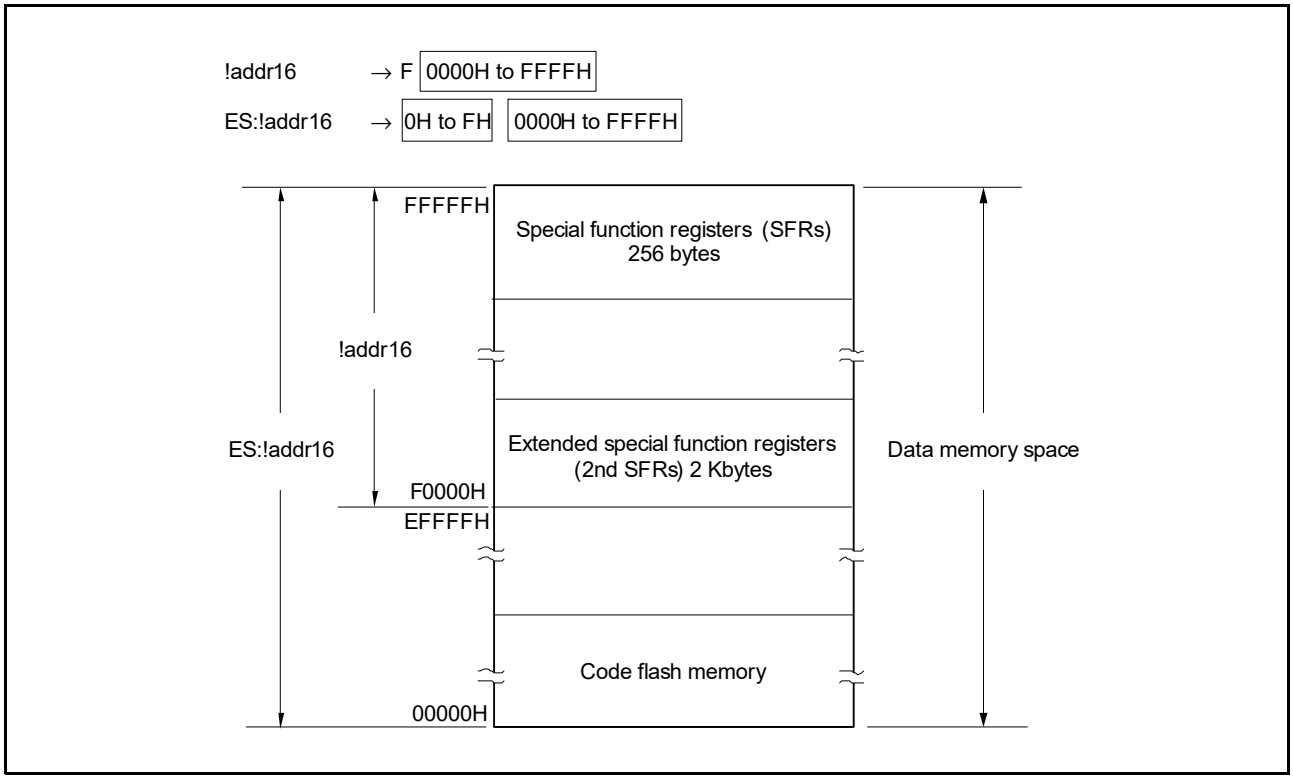
The ES register and CS register are used to specify the higher address for data access and when a branch instruction is executed (register indirect addressing), respectively. The default value of the ES register after reset is 0FH, and that of the CS register is 00H.

Figure 3 - 14 Configuration of ES and CS Registers

Symbol	7	6	5	4	3	2	1	0
ES	0	0	0	0	ES3	ES2	ES1	ES0
Symbol	7	6	5	4	3	2	1	0
CS	0	0	0	0	CS3	CS2	CS1	CS0

Though the data area which can be accessed with 16-bit addresses is the 64 Kbytes from F0000H to FFFFFH, using the ES register as well extends this to the 1 Mbyte from 00000H to FFFFFH.

Figure 3 - 15 Extension of Data Area Which Can Be Accessed



3.2.4 Special function registers (SFRs)

Unlike a general-purpose register, each special function register (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 memory 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 as follows for the 1-bit memory manipulation instruction operand (sfr.bit).
  When the bit name is defined: <Bit name>
  When the bit name is not defined: <Register name>, <Bit number> or <Address>, <Bit number>
- 8-bit manipulation
  Describe the symbol defined by the assembler for the 8-bit memory manipulation instruction operand (sfr). This manipulation can also be specified with an address.
- 16-bit manipulation
  Describe the symbol defined by the assembler for the 16-bit memory manipulation instruction operand (sfrp). When specifying an address, describe an even address.

Table 3 - 5 gives lists of the SFRs. The meanings of items in the table are as follows.

- Symbol
  This item indicates the address of a special function register. It is a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler. When using the assembler, debugger, and simulator, symbols can be written as an instruction operand.
- R/W
  This item indicates whether the corresponding SFR can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulable bit units
  "✓" indicates the manipulable bit unit (1, 8, or 16). "—" indicates a bit unit for which manipulation is not possible.
- After reset
  Items in this column indicate the states (values) of each of the registers after generation of a reset signal.

Caution **Only access the addresses to which SFRs are assigned.**

Remark For 2nd SFRs, see **3.2.5 Extended special function registers (2nd SFRs)**.

Table 3 - 5 List of Special Function Registers (SFRs) (1/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1 bit	8 bits	16 bits	
FFF00H	Port register 0	P0		R/W	✓	✓	—	00H
FFF01H	Port register 1	P1		R/W	✓	✓	—	00H
FFF02H	Port register 2	P2		R/W	✓	✓	—	00H
FFF03H	Port register 3	P3		R/W	✓	✓	—	00H
FFF04H	Port register 4	P4		R/W	✓	✓	—	00H
FFF05H	Port register 5	P5		R/W	✓	✓	—	00H
FFF06H	Port register 6	P6		R/W	✓	✓	—	00H
FFF07H	Port register 7	P7		R/W	✓	✓	—	00H
FFF08H	Port register 8	P8		R/W	✓	✓	—	00H
FFF09H	Port register 9	P9		R/W	✓	✓	—	00H
FFF0AH	Port register 10	P10		R/W	✓	✓	—	00H
FFF0BH	Port register 11	P11		R/W	✓	✓	—	00H
FFF0CH	Port register 12	P12		R/W	✓	✓	—	Undefined
FFF0DH	Port register 13	P13		R/W	✓	✓	—	Undefined
FFF0EH	Port register 14	P14		R/W	✓	✓	—	00H
FFF0FH	Port register 15	P15		R/W	✓	✓	—	00H
FFF10H	Serial data register 00	TXD0/ SIO00	SDR00	R/W	—	✓	✓	0000H
FFF11H		—			—	—		
FFF12H	Serial data register 01	RXD0/ SIO01	SDR01	R/W	—	✓	✓	0000H
FFF13H		—			—	—		
FFF14H	Serial data register 12	TXD3/ SIO30	SDR12	R/W	—	✓	✓	0000H
FFF15H		—			—	—		
FFF16H	Serial data register 13	RXD3/ SIO31	SDR13	R/W	—	✓	✓	0000H
FFF17H		—			—	—		
FFF18H	Timer data register 00	TDR00		R/W	—	—	✓	0000H
FFF19H								
FFF1AH	Timer data register 01	TDR01L	TDR01	R/W	—	✓	✓	00H
FFF1BH		TDR01H			—	✓		00H
FFF1EH	12-bit/10-bit A/D conversion result register	ADCR		R	—	—	✓	0000H
FFF1FH	8-bit A/D conversion result register	ADCRH		R	—	✓	—	00H
FFF20H	Port mode register 0	PM0		R/W	✓	✓	—	FFH
FFF21H	Port mode register 1	PM1		R/W	✓	✓	—	FFH
FFF22H	Port mode register 2	PM2		R/W	✓	✓	—	FFH
FFF23H	Port mode register 3	PM3		R/W	✓	✓	—	FFH
FFF24H	Port mode register 4	PM4		R/W	✓	✓	—	FFH
FFF25H	Port mode register 5	PM5		R/W	✓	✓	—	FFH
FFF26H	Port mode register 6	PM6		R/W	✓	✓	—	FFH
FFF27H	Port mode register 7	PM7		R/W	✓	✓	—	FFH
FFF28H	Port mode register 8	PM8		R/W	✓	✓	—	FFH
FFF29H	Port mode register 9	PM9		R/W	✓	✓	—	FFH
FFF2AH	Port mode register 10	PM10		R/W	✓	✓	—	FFH
FFF2BH	Port mode register 11	PM11		R/W	✓	✓	—	FFH
FFF2CH	Port mode register 12	PM12		R/W	✓	✓	—	FFH
FFF2EH	Port mode register 14	PM14		R/W	✓	✓	—	FFH
FFF2FH	Port mode register 15	PM15		R/W	✓	✓	—	FFH

Table 3 - 5 List of Special Function Registers (SFRs) (2/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1 bit	8 bits	16 bits	
FFF30H	A/D converter mode register 0	ADM0		R/W	✓	✓	—	00H
FFF31H	Analog input channel specification register	ADS		R/W	✓	✓	—	00H
FFF32H	A/D converter mode register 1	ADM1		R/W	✓	✓	—	00H
FFF34H	Key return control register	KRCTL		R/W	✓	✓	—	00H
FFF35H	Key return flag register	KRF		R/W	—	✓	—	00H
FFF37H	Key return mode register 0	KRM0		R/W	✓	✓	—	00H
FFF38H	External interrupt rising edge enable register 0	EGP0		R/W	✓	✓	—	00H
FFF39H	External interrupt falling edge enable register 0	EGN0		R/W	✓	✓	—	00H
FFF3AH	External interrupt rising edge enable register 1	EGP1		R/W	✓	✓	—	00H
FFF3BH	External interrupt falling edge enable register 1	EGN1		R/W	✓	✓	—	00H
FFF44H	Serial data register 02	TXD1/ SIO10	SDR02	R/W	—	✓	✓	0000H
FFF45H		—			—	—		
FFF46H	Serial data register 03	RXD1/ SIO11	SDR03	R/W	—	✓	✓	0000H
FFF47H		—			—	—		
FFF48H	Serial data register 10	TXD2/ SIO20	SDR10	R/W	—	✓	✓	0000H
FFF49H		—			—	—		
FFF4AH	Serial data register 11	RXD2/ SIO21	SDR11	R/W	—	✓	✓	0000H
FFF4BH		—			—	—		
FFF50H	IICA shift register 0	IICA0		R/W	—	✓	—	00H
FFF51H	IICA status register 0	IICS0		R	✓	✓	—	00H
FFF52H	IICA flag register 0	IICF0		R/W	✓	✓	—	00H
FFF54H	IICA shift register 1	IICA1		R/W	—	✓	—	00H
FFF55H	IICA status register 1	IICS1		R	✓	✓	—	00H
FFF56H	IICA flag register 1	IICF1		R/W	✓	✓	—	00H
FFF64H	Timer data register 02	TDR02		R/W	—	—	✓	0000H
FFF65H								
FFF66H	Timer data register 03	TDR03L	TDR03	R/W	—	✓	✓	00H
FFF67H		TDR03H			—	✓		00H
FFF68H	Timer data register 04	TDR04		R/W	—	—	✓	0000H
FFF69H								
FFF6AH	Timer data register 05	TDR05		R/W	—	—	✓	0000H
FFF6BH								
FFF6CH	Timer data register 06	TDR06		R/W	—	—	✓	0000H
FFF6DH								
FFF6EH	Timer data register 07	TDR07		R/W	—	—	✓	0000H
FFF6FH								
FFF70H	Timer data register 10	TDR10		R/W	—	—	✓	0000H
FFF71H								
FFF72H	Timer data register 11	TDR11L	TDR11	R/W	—	✓	✓	00H
FFF73H		TDR11H			—	✓		00H
FFF74H	Timer data register 12	TDR12		R/W	—	—	✓	0000H
FFF75H								
FFF76H	Timer data register 13	TDR13L	TDR13	R/W	—	✓	✓	00H
FFF77H		TDR13H			—	✓		00H

Table 3 - 5 List of Special Function Registers (SFRs) (3/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1 bit	8 bits	16 bits	
FFF78H	Timer data register 14	TDR14		R/W	—	—	✓	0000H
FFF79H								
FFF7AH	Timer data register 15	TDR15		R/W	—	—	✓	0000H
FFF7BH								
FFF7CH	Timer data register 16	TDR16		R/W	—	—	✓	0000H
FFF7DH								
FFF7EH	Timer data register 17	TDR17		R/W	—	—	✓	0000H
FFF7FH								
FFFA0H	Clock operation mode control register	CMC		R/W	—	✓	—	00H
FFFA1H	Clock operation status control register	CSC		R/W	✓	✓	—	C0H
FFFA2H	Oscillation stabilization time counter status register	OSTC		R	✓	✓	—	00H
FFFA3H	Oscillation stabilization time select register	OSTS		R/W	—	✓	—	07H
FFFA4H	System clock control register	CKC		R/W	✓	✓	—	00H
FFFA5H	Clock output select register 0	CKS0		R/W	✓	✓	—	00H
FFFA6H	Clock output select register 1	CKS1		R/W	✓	✓	—	00H
FFFA7H	Subsystem clock select register	CKSEL		R/W	✓	✓	—	00H
FFFA8H	Reset control flag register	RESF		R	—	✓	—	Undefined **Note 1**
FFFA9H	Voltage detection register	LVIM		R/W	✓	✓	—	00H**Note 2**
FFFAAH	Voltage detection level register	LVIS		R/W	✓	✓	—	19H
FFFABH	Watchdog timer enable register	WDTE		R/W	—	✓	—	9AH/ 1AH**Note 3**
FFFACH	CRC input register	CRCIN		R/W	—	✓	—	00H
FFFD0H	Interrupt request flag register 2	IF2L	IF2	R/W	✓	✓	✓	00H
FFFD1H		IF2H		R/W	✓	✓		00H
FFFD2H	Interrupt request flag register 3	IF3L	IF3	R/W	✓	✓	✓	00H
FFFD3H		IF3H		R/W	✓	✓		00H
FFFD4H	Interrupt mask flag register 2	MK2L	MK2	R/W	✓	✓	✓	FFH
FFFD5H		MK2H		R/W	✓	✓		FFH
FFFD6H	Interrupt mask flag register 3	MK3L	MK3	R/W	✓	✓	✓	FFH
FFFD7H		MK3H		R/W	✓	✓		FFH
FFFD8H	Priority specification flag register 02	PR02L	PR02	R/W	✓	✓	✓	FFH
FFFD9H		PR02H		R/W	✓	✓		FFH
FFFDAH	Priority specification flag register 03	PR03L	PR03	R/W	✓	✓	✓	FFH
FFFDBH		PR03H		R/W	✓	✓		FFH
FFFDCH	Priority specification flag register 12	PR12L	PR12	R/W	✓	✓	✓	FFH
FFFDDH		PR12H		R/W	✓	✓		FFH
FFFDEH	Priority specification flag register 13	PR13L	PR13	R/W	✓	✓	✓	FFH
FFFDFH		PR13H		R/W	✓	✓		FFH
FFFE0H	Interrupt request flag register 0	IF0L	IF0	R/W	✓	✓	✓	00H
FFFE1H		IF0H		R/W	✓	✓		00H
FFFE2H	Interrupt request flag register 1	IF1L	IF1	R/W	✓	✓	✓	00H
FFFE3H		IF1H		R/W	✓	✓		00H
FFFE4H	Interrupt mask flag register 0	MK0L	MK0	R/W	✓	✓	✓	FFH
FFFE5H		MK0H		R/W	✓	✓		FFH
FFFE6H	Interrupt mask flag register 1	MK1L	MK1	R/W	✓	✓	✓	FFH
FFFE7H		MK1H		R/W	✓	✓		FFH
FFFE8H	Priority specification flag register 00	PR00L	PR00	R/W	✓	✓	✓	FFH
FFFE9H		PR00H		R/W	✓	✓		FFH

Table 3 - 5 List of Special Function Registers (SFRs) (4/4)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1 bit	8 bits	16 bits	
FFFEAH	Priority specification flag register 01	PR01L	PR01	R/W	✓	✓	✓	FFH
FFFEBH		PR01H		R/W	✓	✓		FFH
FFFECH	Priority specification flag register 10	PR10L	PR10	R/W	✓	✓	✓	FFH
FFFEDH		PR10H		R/W	✓	✓		FFH
FFFEEH	Priority specification flag register 11	PR11L	PR11	R/W	✓	✓	✓	FFH
FFFEFH		PR11H		R/W	✓	✓		FFH
FFFF0H	Multiplication and accumulation register (L)	MACRL		R/W	—	—	✓	0000H
FFFF1H								
FFFF2H	Multiplication and accumulation register (H)	MACRH		R/W	—	—	✓	0000H
FFFF3H								
FFFFEH	Processor mode control register	PMC		R/W	✓	✓	—	00H

Note 1. For the reset resources, see **Section 24 Reset Function**.

Note 2. The initial value depends on the source of the reset. For details, see **26.3.1 Voltage detection register (LVIM)**.

Note 3. The reset value of the WDTE register is determined by the setting of the option byte.

Remark For 2nd SFRs, see **Table 3 - 6 List of Extended Special Function Registers (2nd SFRs)**.

3.2.5 Extended special function registers (2nd SFRs)

Unlike a general-purpose register, each extended special function register (2nd SFR) has a special function. 2nd 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 2nd SFR area, however, is 1 byte longer than an instruction that accesses the SFR area. 2nd SFRs can be manipulated like general-purpose registers, using operation, transfer, and bit memory manipulation instructions. The manipulable bit units, 1, 8, and 16, depend on the 2nd SFR type. Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describe as follows for the 1-bit memory manipulation instruction operand (!addr16.bit).
  When the bit name is defined: <Bit name>
  When the bit name is not defined: <Register name>, <Bit number> or <Address>, <Bit number>
- 8-bit manipulation
  Describe the symbol defined by the assembler for the 8-bit memory manipulation instruction operand (!addr16). This manipulation can also be specified with an address.
- 16-bit manipulation
  Describe the symbol defined by the assembler for the 16-bit memory manipulation instruction operand (!addr16). When specifying an address, describe an even address.

Table 3 - 6 gives lists of the 2nd SFRs. The meanings of items in the table are as follows.

- Symbol
  This item indicates the address of a 2nd SFR. It is a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler. When using the assembler, debugger, and simulator, symbols can be written as an instruction operand.
- R/W
  This item indicates whether the corresponding 2nd SFR can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulable bit units
  "✓" indicates the manipulable bit unit (1, 8, or 16). "—" indicates a bit unit for which manipulation is not possible.
- After reset
  Items in this column indicate the states (values) of each of the registers after generation of a reset signal.

Caution **Only access the addresses to which 2nd SFRs are assigned.**

Remark For SFRs in the SFR area, see **3.1.4 Special function register (SFR) area**.

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (1/15)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0010H	A/D converter mode register 2	ADM2		R/W	✓	✓	—	00H
F0011H	Conversion result comparison upper limit setting register	ADUL		R/W	—	✓	—	FFH
F0012H	Conversion result comparison lower limit setting register	ADLL		R/W	—	✓	—	00H
F0013H	A/D test register	ADTES		R/W	—	✓	—	00H
F0020H	12-bit/10-bit A/D conversion result register 0	—	ADCR0	R	—	—	✓	0000H
F0021H	8-bit A/D conversion result register 0	ADCR0H		R	—	✓	—	00H
F0022H	12-bit/10-bit A/D conversion result register 1	—	ADCR1	R	—	—	✓	0000H
F0023H	8-bit A/D conversion result register 1	ADCR1H		R	—	✓	—	00H
F0024H	12-bit/10-bit A/D conversion result register 2	—	ADCR2	R	—	—	✓	0000H
F0025H	8-bit A/D conversion result register 2	ADCR2H		R	—	✓	—	00H
F0026H	12-bit/10-bit A/D conversion result register 3	—	ADCR3	R	—	—	✓	0000H
F0027H	8-bit A/D conversion result register 3	ADCR3H		R	—	✓	—	00H
F0030H	Pull-up resistor option register 0	PU0		R/W	✓	✓	—	00H
F0031H	Pull-up resistor option register 1	PU1		R/W	✓	✓	—	00H
F0033H	Pull-up resistor option register 3	PU3		R/W	✓	✓	—	00H
F0034H	Pull-up resistor option register 4	PU4		R/W	✓	✓	—	01H
F0035H	Pull-up resistor option register 5	PU5		R/W	✓	✓	—	00H
F0036H	Pull-up resistor option register 6	PU6		R/W	✓	✓	—	00H
F0037H	Pull-up resistor option register 7	PU7		R/W	✓	✓	—	00H
F0038H	Pull-up resistor option register 8	PU8		R/W	✓	✓	—	00H
F0039H	Pull-up resistor option register 9	PU9		R/W	✓	✓	—	00H
F003AH	Pull-up resistor option register 10	PU10		R/W	✓	✓	—	00H
F003BH	Pull-up resistor option register 11	PU11		R/W	✓	✓	—	00H
F003CH	Pull-up resistor option register 12	PU12		R/W	✓	✓	—	00H
F003EH	Pull-up resistor option register 14	PU14		R/W	✓	✓	—	00H
F0040H	Port input mode register 0	PIM0		R/W	✓	✓	—	00H
F0041H	Port input mode register 1	PIM1		R/W	✓	✓	—	00H
F0043H	Port input mode register 3	PIM3		R/W	✓	✓	—	00H
F0044H	Port input mode register 4	PIM4		R/W	✓	✓	—	00H
F0045H	Port input mode register 5	PIM5		R/W	✓	✓	—	00H
F0047H	Port input mode register 7	PIM7		R/W	✓	✓	—	00H
F0048H	Port input mode register 8	PIM8		R/W	✓	✓	—	00H
F004EH	Port input mode register 14	PIM14		R/W	✓	✓	—	00H
F0050H	Port output mode register 0	POM0		R/W	✓	✓	—	00H
F0051H	Port output mode register 1	POM1		R/W	✓	✓	—	00H
F0053H	Port output mode register 3	POM3		R/W	✓	✓	—	00H
F0054H	Port output mode register 4	POM4		R/W	✓	✓	—	00H
F0055H	Port output mode register 5	POM5		R/W	✓	✓	—	00H
F0057H	Port output mode register 7	POM7		R/W	✓	✓	—	00H
F0058H	Port output mode register 8	POM8		R/W	✓	✓	—	00H
F0059H	Port output mode register 9	POM9		R/W	✓	✓	—	00H
F005CH	Port output mode register 12	POM12		R/W	✓	✓	—	00H
F005EH	Port output mode register 14	POM14		R/W	✓	✓	—	00H
F0060H	Port mode control A register 0	PMCA0		R/W	✓	✓	—	FFH
F0061H	Port mode control A register 1	PMCA1		R/W	✓	✓	—	FFH
F0062H	Port mode control A register 2	PMCA2		R/W	✓	✓	—	FFH

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (2/15)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0063H	Port mode control A register 3	PMCA3		R/W	✓	✓	—	FFH
F006AH	Port mode control A register 10	PMCA10		R/W	✓	✓	—	FFH
F006BH	Port mode control A register 11	PMCA11		R/W	✓	✓	—	FFH
F006CH	Port mode control A register 12	PMCA12		R/W	✓	✓	—	FFH
F006EH	Port mode control A register 14	PMCA14		R/W	✓	✓	—	FFH
F006FH	Port mode control A register 15	PMCA15		R/W	✓	✓	—	FFH
F0070H	Noise filter enable register 0	NFEN0		R/W	✓	✓	—	00H
F0071H	Noise filter enable register 1	NFEN1		R/W	✓	✓	—	00H
F0072H	Noise filter enable register 2	NFEN2		R/W	✓	✓	—	00H
F0073H	Input switch control register	ISC		R/W	✓	✓	—	00H
F0074H	Timer input select register 0	TIS0		R/W	—	✓	—	00H
F0075H	Timer input select register 1	TIS1		R/W	—	✓	—	00H
F0077H	Peripheral I/O redirection register	PIOR		R/W	—	✓	—	00H
F0078H	Invalid memory access detection control register	IAWCTL		R/W	—	✓	—	00H
F0079H	UART loop back select register	ULBS		R/W	✓	✓	—	00H
F007BH	Port mode select register	PMS		R/W	✓	✓	—	00H
F007DH	Global digital input disable register	GDIDIS		R/W	✓	✓	—	00H
F0090H	Data flash control register	DFLCTL		R/W	✓	✓	—	00H
F00A0H	High-speed on-chip oscillator trimming register	HIOTRM		R/W	—	✓	—	**Note 1**
F00A8H	High-speed on-chip oscillator frequency select register	HOCODIV		R/W	—	✓	—	**Note 2**
F00AAH	Flash operating mode select register	FLMODE		R/W	✓	✓	—	40H/80H/C0H **Note 3**
F00ABH	Flash operating mode protect register	FLMWRP		R/W	✓	✓	—	00H
F00B0H	Flash security flag monitoring register	FLSEC		R	—	—	✓	Undefined
F00B2H	Flash FSW monitoring register S	FLFSWS		R	—	—	✓	Undefined
F00B4H	Flash FSW monitoring register E	FLFSWE		R	—	—	✓	Undefined
F00B6H	Flash memory sequencer initial setting register	FSSET		R/W	—	✓	—	00H
F00B7H	Flash extra area sequencer control register	FSSE		R/W	✓	✓	—	00H
F00C0H	Flash protect command register	PFCMD		W	—	✓	—	Undefined
F00C1H	Flash status register	PFS		R	✓	✓	—	00H
F00F0H	Peripheral enable register 0	PER0		R/W	✓	✓	—	00H
F00F1H	Peripheral reset control register 0	PRR0		R/W	✓	✓	—	00H
F00F2H	Middle-speed on-chip oscillator frequency select register	MOCODIV		R/W	—	✓	—	00H
F00F3H	Subsystem clock supply mode control register	OSMC		R/W	✓	✓	—	Undefined **Note 4**
F00F5H	RAM parity error control register	RPECTL		R/W	✓	✓	—	00H
F00F9H	Power-on-reset status register	PORSR		R/W	✓	✓	—	00H
F00FAH	Peripheral enable register 1	PER1		R/W	✓	✓	—	00H
F00FBH	Peripheral reset control register 1	PRR1		R/W	✓	✓	—	00H
F00FEH	BCD correction result register	BCDADJ		R	—	✓	—	Undefined
F00FFH	Interrupt vector jump enable register	VECTCTRL		R/W	—	✓	—	00H
F0100H	Serial status register 00	SSR00L	SSR00	R	—	✓	✓	0000H
F0101H		—			—	—		
F0102H	Serial status register 01	SSR01L	SSR01	R	—	✓	✓	0000H
F0103H		—			—	—		
F0104H	Serial status register 02	SSR02L	SSR02	R	—	✓	✓	0000H
F0105H		—			—	—		

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (3/15)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0106H	Serial status register 03	SSR03L	SSR03	R	—	✓	✓	0000H
F0107H		—			—	—		
F0108H	Serial flag clear trigger register 00	SIR00L	SIR00	R/W	—	✓	✓	0000H
F0109H		—			—	—		
F010AH	Serial flag clear trigger register 01	SIR01L	SIR01	R/W	—	✓	✓	0000H
F010BH		—			—	—		
F010CH	Serial flag clear trigger register 02	SIR02L	SIR02	R/W	—	✓	✓	0000H
F010DH		—			—	—		
F010EH	Serial flag clear trigger register 03	SIR03L	SIR03	R/W	—	✓	✓	0000H
F010FH		—			—	—		
F0110H	Serial mode register 00	SMR00		R/W	—	—	✓	0020H
F0111H								
F0112H	Serial mode register 01	SMR01		R/W	—	—	✓	0020H
F0113H								
F0114H	Serial mode register 02	SMR02		R/W	—	—	✓	0020H
F0115H								
F0116H	Serial mode register 03	SMR03		R/W	—	—	✓	0020H
F0117H								
F0118H	Serial communication operation setting register 00	SCR00		R/W	—	—	✓	0087H
F0119H								
F011AH	Serial communication operation setting register 01	SCR01		R/W	—	—	✓	0087H
F011BH								
F011CH	Serial communication operation setting register 02	SCR02		R/W	—	—	✓	0087H
F011DH								
F011EH	Serial communication operation setting register 03	SCR03		R/W	—	—	✓	0087H
F011FH								
F0120H	Serial channel enable status register 0	SE0L	SE0	R	✓	✓	✓	0000H
F0121H		—			—	—		
F0122H	Serial channel start register 0	SS0L	SS0	R/W	✓	✓	✓	0000H
F0123H		—			—	—		
F0124H	Serial channel stop register 0	ST0L	ST0	R/W	✓	✓	✓	0000H
F0125H		—			—	—		
F0126H	Serial clock select register 0	SPS0L	SPS0	R/W	—	✓	✓	0000H
F0127H		—			—	—		
F0128H	Serial output register 0	SO0		R/W	—	—	✓	0F0FH
F0129H								
F012AH	Serial output enable register 0	SOE0L	SOE0	R/W	✓	✓	✓	0000H
F012BH		—			—	—		
F0134H	Serial output level register 0	SOL0L	SOL0	R/W	—	✓	✓	0000H
F0135H		—			—	—		
F0138H	Serial standby control register 0	SSC0L	SSC0	R/W	—	✓	✓	0000H
F0139H		—			—	—		
F0140H	Serial status register 10	SSR10L	SSR10	R	—	✓	✓	0000H
F0141H		—			—	—		
F0142H	Serial status register 11	SSR11L	SSR11	R	—	✓	✓	0000H
F0143H		—			—	—		
F0144H	Serial status register 12	SSR12L	SSR12	R	—	✓	✓	0000H
F0145H		—			—	—		

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (4/15)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0146H	Serial status register 13	SSR13L	SSR13	R	—	✓	✓	0000H
F0147H		—			—	—		
F0148H	Serial flag clear trigger register 10	SIR10L	SIR10	R/W	—	✓	✓	0000H
F0149H		—			—	—		
F014AH	Serial flag clear trigger register 11	SIR11L	SIR11	R/W	—	✓	✓	0000H
F014BH		—			—	—		
F014CH	Serial flag clear trigger register 12	SIR12L	SIR12	R/W	—	✓	✓	0000H
F014DH		—			—	—		
F014EH	Serial flag clear trigger register 13	SIR13L	SIR13	R/W	—	✓	✓	0000H
F014FH		—			—	—		
F0150H	Serial mode register 10	SMR10		R/W	—	—	✓	0020H
F0151H								
F0152H	Serial mode register 11	SMR11		R/W	—	—	✓	0020H
F0153H								
F0154H	Serial mode register 12	SMR12		R/W	—	—	✓	0020H
F0155H								
F0156H	Serial mode register 13	SMR13		R/W	—	—	✓	0020H
F0157H								
F0158H	Serial communication operation setting register 10	SCR10		R/W	—	—	✓	0087H
F0159H								
F015AH	Serial communication operation setting register 11	SCR11		R/W	—	—	✓	0087H
F015BH								
F015CH	Serial communication operation setting register 12	SCR12		R/W	—	—	✓	0087H
F015DH								
F015EH	Serial communication operation setting register 13	SCR13		R/W	—	—	✓	0087H
F015FH								
F0160H	Serial channel enable status register 1	SE1L	SE1	R	✓	✓	✓	0000H
F0161H		—			—	—		
F0162H	Serial channel start register 1	SS1L	SS1	R/W	✓	✓	✓	0000H
F0163H		—			—	—		
F0164H	Serial channel stop register 1	ST1L	ST1	R/W	✓	✓	✓	0000H
F0165H		—			—	—		
F0166H	Serial clock select register 1	SPS1L	SPS1	R/W	—	✓	✓	0000H
F0167H		—			—	—		
F0168H	Serial output register 1	SO1		R/W	—	—	✓	0F0FH
F0169H								
F016AH	Serial output enable register 1	SOE1L	SOE1	R/W	✓	✓	✓	0000H
F016BH		—			—	—		
F0174H	Serial output level register 1	SOL1L	SOL1	R/W	—	✓	✓	0000H
F0175H		—			—	—		
F0178H	Serial standby control register 1	SSC1L	SSC1	R/W	—	✓	✓	0000H
F0179H		—			—	—		
F0180H	Timer counter register 00	TCR00		R	—	—	✓	FFFFH
F0181H								
F0182H	Timer counter register 01	TCR01		R	—	—	✓	FFFFH
F0183H								
F0184H	Timer counter register 02	TCR02		R	—	—	✓	FFFFH
F0185H								

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (5/15)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0186H	Timer counter register 03	TCR03		R	—	—	✓	FFFFH
F0187H								
F0188H	Timer counter register 04	TCR04		R	—	—	✓	FFFFH
F0189H								
F018AH	Timer counter register 05	TCR05		R	—	—	✓	FFFFH
F018BH								
F018CH	Timer counter register 06	TCR06		R	—	—	✓	FFFFH
F018DH								
F018EH	Timer counter register 07	TCR07		R	—	—	✓	FFFFH
F018FH								
F0190H	Timer mode register 00	TMR00		R/W	—	—	✓	0000H
F0191H								
F0192H	Timer mode register 01	TMR01		R/W	—	—	✓	0000H
F0193H								
F0194H	Timer mode register 02	TMR02		R/W	—	—	✓	0000H
F0195H								
F0196H	Timer mode register 03	TMR03		R/W	—	—	✓	0000H
F0197H								
F0198H	Timer mode register 04	TMR04		R/W	—	—	✓	0000H
F0199H								
F019AH	Timer mode register 05	TMR05		R/W	—	—	✓	0000H
F019BH								
F019CH	Timer mode register 06	TMR06		R/W	—	—	✓	0000H
F019DH								
F019EH	Timer mode register 07	TMR07		R/W	—	—	✓	0000H
F019FH								
F01A0H	Timer status register 00	TSR00L	TSR00	R	—	✓	✓	0000H
F01A1H		—			—	—		
F01A2H	Timer status register 01	TSR01L	TSR01	R	—	✓	✓	0000H
F01A3H		—			—	—		
F01A4H	Timer status register 02	TSR02L	TSR02	R	—	✓	✓	0000H
F01A5H		—			—	—		
F01A6H	Timer status register 03	TSR03L	TSR03	R	—	✓	✓	0000H
F01A7H		—			—	—		
F01A8H	Timer status register 04	TSR04L	TSR04	R	—	✓	✓	0000H
F01A9H		—			—	—		
F01AAH	Timer status register 05	TSR05L	TSR05	R	—	✓	✓	0000H
F01ABH		—			—	—		
F01ACH	Timer status register 06	TSR06L	TSR06	R	—	✓	✓	0000H
F01ADH		—			—	—		
F01AEH	Timer status register 07	TSR07L	TSR07	R	—	✓	✓	0000H
F01AFH		—			—	—		
F01B0H	Timer channel enable status register 0	TE0L	TE0	R	✓	✓	✓	0000H
F01B1H		—			—	—		
F01B2H	Timer channel start register 0	TS0L	TS0	R/W	✓	✓	✓	0000H
F01B3H		—			—	—		
F01B4H	Timer channel stop register 0	TT0L	TT0	R/W	✓	✓	✓	0000H
F01B5H		—			—	—		

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (6/15)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F01B6H	Timer clock select register 0	TPS0		R/W	—	—	✓	0000H
F01B7H								
F01B8H	Timer output register 0	TO0L	TO0	R/W	—	✓	✓	0000H
F01B9H		—			—	—		
F01BAH	Timer output enable register 0	TOE0L	TOE0	R/W	✓	✓	✓	0000H
F01BBH		—			—	—		
F01BCH	Timer output level register 0	TOL0L	TOL0	R/W	—	✓	✓	0000H
F01BDH		—			—	—		
F01BEH	Timer output mode register 0	TOM0L	TOM0	R/W	—	✓	✓	0000H
F01BFH		—			—	—		
F01C0H	Timer counter register 10	TCR10		R	—	—	✓	FFFFH
F01C1H								
F01C2H	Timer counter register 11	TCR11		R	—	—	✓	FFFFH
F01C3H								
F01C4H	Timer counter register 12	TCR12		R	—	—	✓	FFFFH
F01C5H								
F01C6H	Timer counter register 13	TCR13		R	—	—	✓	FFFFH
F01C7H								
F01C8H	Timer counter register 14	TCR14		R	—	—	✓	FFFFH
F01C9H								
F01CAH	Timer counter register 15	TCR15		R	—	—	✓	FFFFH
F01CBH								
F01CCH	Timer counter register 16	TCR16		R	—	—	✓	FFFFH
F01CDH								
F01CEH	Timer counter register 17	TCR17		R	—	—	✓	FFFFH
F01CFH								
F01D0H	Timer mode register 10	TMR10		R/W	—	—	✓	0000H
F01D1H								
F01D2H	Timer mode register 11	TMR11		R/W	—	—	✓	0000H
F01D3H								
F01D4H	Timer mode register 12	TMR12		R/W	—	—	✓	0000H
F01D5H								
F01D6H	Timer mode register 13	TMR13		R/W	—	—	✓	0000H
F01D7H								
F01D8H	Timer mode register 14	TMR14		R/W	—	—	✓	0000H
F01D9H								
F01DAH	Timer mode register 15	TMR15		R/W	—	—	✓	0000H
F01DBH								
F01DCH	Timer mode register 16	TMR16		R/W	—	—	✓	0000H
F01DDH								
F01DEH	Timer mode register 17	TMR17		R/W	—	—	✓	0000H
F01DFH								
F01E0H	Timer status register 10	TSR10L	TSR10	R	—	✓	✓	0000H
F01E1H		—			—	—		
F01E2H	Timer status register 11	TSR11L	TSR11	R	—	✓	✓	0000H
F01E3H		—			—	—		
F01E4H	Timer status register 12	TSR12L	TSR12	R	—	✓	✓	0000H
F01E5H		—			—	—		

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (7/15)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F01E6H	Timer status register 13	TSR13L	TSR13	R	—	✓	✓	0000H
F01E7H		—			—	—		
F01E8H	Timer status register 14	TSR14L	TSR14	R	—	✓	✓	0000H
F01E9H		—			—	—		
F01EAH	Timer status register 15	TSR15L	TSR15	R	—	✓	✓	0000H
F01EBH		—			—	—		
F01ECH	Timer status register 16	TSR16L	TSR16	R	—	✓	✓	0000H
F01EDH		—			—	—		
F01EEH	Timer status register 17	TSR17L	TSR17	R	—	✓	✓	0000H
F01EFH		—			—	—		
F01F0H	Timer channel enable status register 1	TE1L	TE1	R	✓	✓	✓	0000H
F01F1H		—			—	—		
F01F2H	Timer channel start register 1	TS1L	TS1	R/W	✓	✓	✓	0000H
F01F3H		—			—	—		
F01F4H	Timer channel stop register 1	TT1L	TT1	R/W	✓	✓	✓	0000H
F01F5H		—			—	—		
F01F6H	Timer clock select register 1	TPS1		R/W	—	—	✓	0000H
F01F7H					—	—		
F01F8H	Timer output register 1	TO1L	TO1	R/W	—	✓	✓	0000H
F01F9H		—			—	—		
F01FAH	Timer output enable register 1	TOE1L	TOE1	R/W	✓	✓	✓	0000H
F01FBH		—			—	—		
F01FCH	Timer output level register 1	TOL1L	TOL1	R/W	—	✓	✓	0000H
F01FDH		—			—	—		
F01FEH	Timer output mode register 1	TOM1L	TOM1	R/W	—	✓	✓	0000H
F01FFH		—			—	—		
F0212H	Middle-speed on-chip oscillator trimming register	MIOTRM		R/W	—	✓	—	90H
F0213H	Low-speed on-chip oscillator trimming register	LIOTRM		R/W	—	✓	—	80H
F0214H	High-speed system clock division register	MOSCDIV		R/W	—	✓	—	00H
F0215H	Standby mode release setting register	WKUPMD		R/W	✓	✓	—	00H
F0216H	Memory power reduction control register	PSMCR		R/W	✓	✓	—	00H
F0218H	LVD detection flag clearing register	LVDFCLR		R/W	✓	✓	—	00H
F0220H	Second count register	SEC		R/W	—	✓	—	Undefined
F0221H	Minute count register	MIN		R/W	—	✓	—	Undefined
F0222H	Hour count register	HOUR		R/W	—	✓	—	Undefined
F0223H	Day-of-week count register	WEEK		R/W	—	✓	—	Undefined
F0224H	Day count register	DAY		R/W	—	✓	—	Undefined
F0225H	Month count register	MONTH		R/W	—	✓	—	Undefined
F0226H	Year count register	YEAR		R/W	—	✓	—	Undefined
F0227H	Time error correction register	SUBCUD		R/W	—	✓	—	00H
F0228H	Alarm minute register	ALARMWM		R/W	—	✓	—	Undefined
F0229H	Alarm hour register	ALARMWH		R/W	—	✓	—	Undefined
F022AH	Alarm day-of-week register	ALARMWW		R/W	—	✓	—	Undefined
F022BH	Realtime clock control register 0	RTCC0		R/W	✓	✓	—	00H
F022CH	Realtime clock control register 1	RTCC1		R/W	✓	✓	—	00H
F0230H	IICA control register 00	IICCTL00		R/W	✓	✓	—	00H
F0231H	IICA control register 01	IICCTL01		R/W	✓	✓	—	00H
F0232H	IICA low-level width setting register 0	IICWL0		R/W	—	✓	—	FFH

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (8/15)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0233H	IICA high-level width setting register 0	IICWH0	R/W	—	✓	—	FFH
F0234H	Slave address register 0	SVA0	R/W	—	✓	—	00H
F0238H	IICA control register 10	IICCTL10	R/W	✓	✓	—	00H
F0239H	IICA control register 11	IICCTL11	R/W	✓	✓	—	00H
F023AH	IICA low-level width setting register 1	IICWL1	R/W	—	✓	—	FFH
F023BH	IICA high-level width setting register 1	IICWH1	R/W	—	✓	—	FFH
F023CH	Slave address register 1	SVA1	R/W	—	✓	—	00H
F0260H	Port mode control T register 0	PMCT0	R/W	✓	✓	—	00H
F0262H	Port mode control T register 2	PMCT2	R/W	✓	✓	—	00H
F0263H	Port mode control T register 3	PMCT3	R/W	✓	✓	—	00H
F0265H	Port mode control T register 5	PMCT5	R/W	✓	✓	—	00H
F0266H	Port mode control T register 6	PMCT6	R/W	✓	✓	—	00H
F0267H	Port mode control T register 7	PMCT7	R/W	✓	✓	—	00H
F026FH	Port mode control T register 15	PMCT15	R/W	✓	✓	—	00H
F0280H	Port mode control E register 0	PMCE0	R/W	✓	✓	—	00H
F0281H	Port mode control E register 1	PMCE1	R/W	✓	✓	—	00H
F0285H	Port mode control E register 5	PMCE5	R/W	✓	✓	—	00H
F0286H	Port mode control E register 6	PMCE6	R/W	✓	✓	—	00H
F02A0H	Output current select register 0	CCS0	R/W	—	✓	—	00H
F02A4H	Output current select register 4	CCS4	R/W	—	✓	—	00H
F02A5H	Output current select register 5	CCS5	R/W	—	✓	—	00H
F02A6H	Output current select register 6	CCS6	R/W	—	✓	—	00H
F02A7H	Output current select register 7	CCS7	R/W	—	✓	—	00H
F02A8H	Output current control enable register	CCDE	R/W	✓	✓	—	00H
F02A9H	40-mA port output control register	PTDC	R/W	✓	✓	—	00H
F02AAH	Port function output enable register 0	PFOE0	R/W	✓	✓	—	FFH
F02ABH	Port function output enable register 1	PFOE1	R/W	✓	✓	—	FFH
F02B0H	Port digital input disable register 0	PDIDIS0	R/W	✓	✓	—	00H
F02B1H	Port digital input disable register 1	PDIDIS1	R/W	✓	✓	—	00H
F02B3H	Port digital input disable register 3	PDIDIS3	R/W	✓	✓	—	00H
F02B4H	Port digital input disable register 4	PDIDIS4	R/W	✓	✓	—	00H
F02B5H	Port digital input disable register 5	PDIDIS5	R/W	✓	✓	—	00H
F02B7H	Port digital input disable register 7	PDIDIS7	R/W	✓	✓	—	00H
F02B8H	Port digital input disable register 8	PDIDIS8	R/W	✓	✓	—	00H
F02B9H	Port digital input disable register 9	PDIDIS9	R/W	✓	✓	—	00H
F02BCH	Port digital input disable register 12	PDIDIS12	R/W	✓	✓	—	00H
F02BDH	Port digital input disable register 13	PDIDIS13	R/W	✓	✓	—	00H
F02BEH	Port digital input disable register 14	PDIDIS14	R/W	✓	✓	—	00H
F02C0H	Flash programming mode control register	FLPMC	R/W	—	✓	—	08H
F02C1H	Flash area selection register	FLARS	R/W	✓	✓	—	00H
F02C2H	Flash address pointer register L	FLAPL	R/W	—	—	✓	0000H
F02C4H	Flash address pointer register H	FLAPH	R/W	—	✓	—	00H
F02C5H	Flash memory sequencer control register	FSSQ	R/W	✓	✓	—	00H
F02C6H	Flash end address pointer register L	FLSEDL	R/W	—	—	✓	0000H
F02C8H	Flash end address pointer register H	FLSEDH	R/W	—	✓	—	00H
F02C9H	Flash registers initialization register	FLRST	R/W	✓	✓	—	00H
F02CAH	Flash memory sequencer status register L	FSASTL	R	✓	✓	—	00H/80H
F02CBH	Flash memory sequencer status register H	FSASTH	R	✓	✓	—	00H/04H

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (9/15)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F02CCH	Flash write buffer register L	FLWL		R/W	—	—	✓	0000H
F02CEH	Flash write buffer register H	FLWH		R/W	—	—	✓	0000H
F02E0H	DTC base address register	DTCBAR		R/W	—	✓	—	FDH
F02E8H	DTC activation enable register 0	DTCEN0		R/W	✓	✓	—	00H
F02E9H	DTC activation enable register 1	DTCEN1		R/W	✓	✓	—	00H
F02EAH	DTC activation enable register 2	DTCEN2		R/W	✓	✓	—	00H
F02EBH	DTC activation enable register 3	DTCEN3		R/W	✓	✓	—	00H
F02ECH	DTC activation enable register 4	DTCEN4		R/W	✓	✓	—	00H
F02F0H	Flash memory CRC control register	CRC0CTL		R/W	✓	✓	—	00H
F02F2H	Flash memory CRC operation result register	PGCRCL		R/W	—	—	✓	0000H
F02FAH	CRC data register	CRCD		R/W	—	—	✓	0000H
F0300H	Transmit buffer register 0	TXBA0		R/W	—	✓	—	FFH
F0301H	Receive buffer register 0	RXBA0		R	—	✓	—	FFH
F0302H	Operation mode setting register 00	ASIMA00		R/W	✓	✓	—	01H
F0303H	Operation mode setting register 01	ASIMA01		R/W	✓	✓	—	1AH
F0304H	Baud rate generator control register 0	BRGCA0		R/W	—	✓	—	FFH
F0305H	Status register 0	ASISA0		R	—	✓	—	00H
F0306H	Status clear trigger register 0	ASCTA0		R/W	✓	✓	—	00H
F0308H	Transmit buffer register 1	TXBA1		R/W	—	✓	—	FFH
F0309H	Receive buffer register 1	RXBA1		R	—	✓	—	FFH
F030AH	Operation mode setting register 10	ASIMA10		R/W	✓	✓	—	01H
F030BH	Operation mode setting register 11	ASIMA11		R/W	✓	✓	—	1AH
F030CH	Baud rate generator control register 1	BRGCA1		R/W	—	✓	—	FFH
F030DH	Status register 1	ASISA1		R	—	✓	—	00H
F030EH	Status clear trigger register 1	ASCTA1		R/W	✓	✓	—	00H
F0310H	UARTA clock select register 0	UTA0CK		R/W	✓	✓	—	00H
F0311H	UARTA clock select register 1	UTA1CK		R/W	✓	✓	—	00H
F0330H	D/A conversion value setting register 0	DACS0		R/W	—	✓	—	00H
F0331H	D/A conversion value setting register 1	DACS1		R/W	—	✓	—	00H
F0332H	D/A converter mode register	DAM		R/W	✓	✓	—	00H
F0340H	Comparator mode setting register	COMPMDR		R/W	✓	✓	—	00H
F0341H	Comparator filter control register	COMPFIR		R/W	✓	✓	—	00H
F0342H	Comparator output control register	COMPOCR		R/W	✓	✓	—	00H
F0360H	Interval timer compare register 00	ITLCMP000	ITLCMP00	R/W	—	✓	✓	FFFFH
F0361H		ITLCMP001			—	✓		
F0362H	Interval timer compare register 01	ITLCMP012	ITLCMP01	R/W	—	✓	✓	FFFFH
F0363H		ITLCMP013			—	✓		
F0364H	Interval timer capture register 00	ITLCAP00		R	—	—	✓	0000H
F0365H								
F0366H	Interval timer control register	ITLCTL0		R/W	✓	✓	—	00H
F0367H	Interval timer clock select register 0	ITLCSEL0		R/W	—	✓	—	00H
F0368H	Interval timer frequency division register 0	ITLFDIV00		R/W	—	✓	—	00H
F0369H	Interval timer frequency division register 1	ITLFDIV01		R/W	—	✓	—	00H
F036AH	Interval timer capture control register 0	ITLCC0		R/W	✓	✓	—	00H
F036BH	Interval timer status register	ITLS0		R/W	—	✓	—	00H
F036CH	Interval timer match detection mask register	ITLMKF0		R/W	—	✓	—	00H
F0380H	Sequencer instruction register 0	SMSI0		R/W	—	—	✓	0000H
F0381H								

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (10/15)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0382H	Sequencer instruction register 1	SMSI1	R/W	—	—	✓	0000H
F0383H							
F0384H	Sequencer instruction register 2	SMSI2	R/W	—	—	✓	0000H
F0385H							
F0386H	Sequencer instruction register 3	SMSI3	R/W	—	—	✓	0000H
F0387H							
F0388H	Sequencer instruction register 4	SMSI4	R/W	—	—	✓	0000H
F0389H							
F038AH	Sequencer instruction register 5	SMSI5	R/W	—	—	✓	0000H
F038BH							
F038CH	Sequencer instruction register 6	SMSI6	R/W	—	—	✓	0000H
F038DH							
F038EH	Sequencer instruction register 7	SMSI7	R/W	—	—	✓	0000H
F038FH							
F0390H	Sequencer instruction register 8	SMSI8	R/W	—	—	✓	0000H
F0391H							
F0392H	Sequencer instruction register 9	SMSI9	R/W	—	—	✓	0000H
F0393H							
F0394H	Sequencer instruction register 10	SMSI10	R/W	—	—	✓	0000H
F0395H							
F0396H	Sequencer instruction register 11	SMSI11	R/W	—	—	✓	0000H
F0397H							
F0398H	Sequencer instruction register 12	SMSI12	R/W	—	—	✓	0000H
F0399H							
F039AH	Sequencer instruction register 13	SMSI13	R/W	—	—	✓	0000H
F039BH							
F039CH	Sequencer instruction register 14	SMSI14	R/W	—	—	✓	0000H
F039DH							
F039EH	Sequencer instruction register 15	SMSI15	R/W	—	—	✓	0000H
F039FH							
F03A0H	Sequencer instruction register 16	SMSI16	R/W	—	—	✓	0000H
F03A1H							
F03A2H	Sequencer instruction register 17	SMSI17	R/W	—	—	✓	0000H
F03A3H							
F03A4H	Sequencer instruction register 18	SMSI18	R/W	—	—	✓	0000H
F03A5H							
F03A6H	Sequencer instruction register 19	SMSI19	R/W	—	—	✓	0000H
F03A7H							
F03A8H	Sequencer instruction register 20	SMSI20	R/W	—	—	✓	0000H
F03A9H							
F03AAH	Sequencer instruction register 21	SMSI21	R/W	—	—	✓	0000H
F03ABH							
F03ACH	Sequencer instruction register 22	SMSI22	R/W	—	—	✓	0000H
F03ADH							
F03AEH	Sequencer instruction register 23	SMSI23	R/W	—	—	✓	0000H
F03AFH							
F03B0H	Sequencer instruction register 24	SMSI24	R/W	—	—	✓	0000H
F03B1H							

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (11/15)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F03B2H	Sequencer instruction register 25	SMSI25	R/W	—	—	✓	0000H
F03B3H							
F03B4H	Sequencer instruction register 26	SMSI26	R/W	—	—	✓	0000H
F03B5H							
F03B6H	Sequencer instruction register 27	SMSI27	R/W	—	—	✓	0000H
F03B7H							
F03B8H	Sequencer instruction register 28	SMSI28	R/W	—	—	✓	0000H
F03B9H							
F03BAH	Sequencer instruction register 29	SMSI29	R/W	—	—	✓	0000H
F03BBH							
F03BCH	Sequencer instruction register 30	SMSI30	R/W	—	—	✓	0000H
F03BDH							
F03BEH	Sequencer instruction register 31	SMSI31	R/W	—	—	✓	0000H
F03BFH							
F03C0H	Sequencer general-purpose register 0	SMSG0	R	—	—	✓	0000H
F03C1H							
F03C2H	Sequencer general-purpose register 1	SMSG1	R/W	—	—	✓	0000H
F03C3H							
F03C4H	Sequencer general-purpose register 2	SMSG2	R/W	—	—	✓	0000H
F03C5H							
F03C6H	Sequencer general-purpose register 3	SMSG3	R/W	—	—	✓	0000H
F03C7H							
F03C8H	Sequencer general-purpose register 4	SMSG4	R/W	—	—	✓	0000H
F03C9H							
F03CAH	Sequencer general-purpose register 5	SMSG5	R/W	—	—	✓	0000H
F03CBH							
F03CCH	Sequencer general-purpose register 6	SMSG6	R/W	—	—	✓	0000H
F03CDH							
F03CEH	Sequencer general-purpose register 7	SMSG7	R/W	—	—	✓	0000H
F03CFH							
F03D0H	Sequencer general-purpose register 8	SMSG8	R/W	—	—	✓	0000H
F03D1H							
F03D2H	Sequencer general-purpose register 9	SMSG9	R/W	—	—	✓	0000H
F03D3H							
F03D4H	Sequencer general-purpose register 10	SMSG10	R/W	—	—	✓	0000H
F03D5H							
F03D6H	Sequencer general-purpose register 11	SMSG11	R/W	—	—	✓	0000H
F03D7H							
F03D8H	Sequencer general-purpose register 12	SMSG12	R/W	—	—	✓	0000H
F03D9H							
F03DAH	Sequencer general-purpose register 13	SMSG13	R/W	—	—	✓	0000H
F03DBH							
F03DCH	Sequencer general-purpose register 14	SMSG14	R/W	—	—	✓	0000H
F03DDH							
F03DEH	Sequencer general-purpose register 15	SMSG15	R	—	—	✓	FFFFH
F03DFH							
F03E0H	Sequencer control register	SMSC	R/W	✓	✓	—	00H
F03E1H	Sequencer status register	SMSS	R	✓	✓	—	00H

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (12/15)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0480H	Interrupt vector change register 0	FLSIVC0		R/W	—	—	✓	0000H
F0481H								
F0482H	Interrupt vector change register 1	FLSIVC1		R/W	—	—	✓	000FH
F0483H								
F0488H	Code flash memory guard register	GFLASH0		R/W	—	—	✓	0000H
F0489H								
F048AH	Data flash memory guard register	GFLASH1		R/W	—	—	✓	0000H
F048BH								
F048CH	Flash security area guard register	GFLASH2		R/W	—	—	✓	0000H
F048DH								
F048EH	Guard register of IAWCTL register	GIAWCTL		R/W	—	—	✓	0000H
F048FH								
F0500H	CTSU control register AL	CTSUCR0	CTSUCRAL	R/W	✓	✓	✓	0000H
F0501H		CTSUCR1			✓	✓		
F0502H	CTSU control register AH	CTSUCR2	CTSUCRAH	R/W	✓	✓	✓	0000H
F0503H		CTSUCR3			✓	✓		
F0504H	CTSU control register BL	CTSUSDPRS	CTSUCRBL	R/W	✓	✓	✓	0000H
F0505H		CTSUSST			—	✓		
F0506H	CTSU control register BH	—	CTSUCRBH	R/W	—	—	✓	0000H
F0507H		CTSUDCLKC			—	✓		
F0508H	CTSU measurement channel register L	CTSUMCH0	CTSUMCHL	R/W	—	✓	✓	3F3FH
F0509H		CTSUMCH1			—	✓		
F050AH	CTSU measurement channel register H	CTSUMFAF	CTSUMCHH	R/W	✓	✓	✓	0000H
F050BH		—			—	—		
F050CH	CTSU channel enable control register AL	CTSUCHAC0	CTSUCHACAL	R/W	✓	✓	✓	0000H
F050DH		CTSUCHAC1			✓	✓		
F050EH	CTSU channel enable control register AH	CTSUCHAC2	CTSUCHACAH	R/W	✓	✓	✓	0000H
F050FH		CTSUCHAC3			✓	✓		
F0510H	CTSU channel enable control register BL	CTSUCHAC4	CTSUCHACBL	R/W	✓	✓	✓	0000H
F0511H		CTSUCHAC5			✓	✓		
F0512H	CTSU channel enable control register BH	CTSUCHAC6	CTSUCHACBH	R/W	✓	✓	✓	0000H
F0513H		CTSUCHAC7			✓	✓		
F0514H	CTSU channel transmit/receive control register AL	CTSUCHTRC0	CTSUCHTRCAL	R/W	✓	✓	✓	0000H
F0515H		CTSUCHTRC1			✓	✓		
F0516H	CTSU channel transmit/receive control register AH	CTSUCHTRC2	CTSUCHTRCAH	R/W	✓	✓	✓	0000H
F0517H		CTSUCHTRC3			✓	✓		
F0518H	CTSU channel transmit/receive control register BL	CTSUCHTRC4	CTSUCHTRCBL	R/W	✓	✓	✓	0000H
F0519H		CTSUCHTRC5			✓	✓		
F051AH	CTSU channel transmit/receive control register BH	CTSUCHTRC6	CTSUCHTRCBH	R/W	✓	✓	✓	0000H
F051BH		CTSUCHTRC7			✓	✓		
F051CH	CTSU status register L	CTSUST1	CTSUSRL	R/W	✓	✓	✓	0000H
F051DH		CTSUST			✓	✓		

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (13/15)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0520H	CTSU sensor offset register 0	CTSUSO0		R/W	—	—	✓	0000H
F0521H								
F0522H	CTSU sensor offset register 1	CTSUSO1		R/W	—	—	✓	0000H
F0523H								
F0524H	CTSU sensor counter register L	CTSUSC		R	—	—	✓	0000H
F0525H								
F0526H	CTSU sensor counter register H	CTSUUC		R	—	—	✓	0000H
F0527H								
F0528H	CTSU calibration register L	CTSUDBGR0		R/W	—	—	✓	0000H
F0529H								
F052AH	CTSU calibration register H	CTSUDBGR1		R/W	—	—	✓	0000H
F052BH								
F052CH	CTSU sensor unit clock control register AL	CTSUSUCLK0		R/W	—	—	✓	0000H
F052DH								
F052EH	CTSU sensor unit clock control register AH	CTSUSUCLK1		R/W	—	—	✓	0000H
F052FH								
F0530H	CTSU sensor unit clock control register BL	CTSUSUCLK2		R/W	—	—	✓	0000H
F0531H								
F0532H	CTSU sensor unit clock control register BH	CTSUSUCLK3		R/W	—	—	✓	0000H
F0533H								
F0540H	Random number seed data register	TRNGSDR		R	—	✓	—	00H
F0542H	Random number seed command register 0	TRNGSCR0		R/W	✓	✓	—	00H
F0600H	CTSU trimming register AL	RTRIM	CTSUTRIM0	R/W	—	✓	✓	**Note 1**
F0601H		DACTRIM			—	✓		
F0602H	CTSU trimming register AH	SUADJD	CTSUTRIM1	R/W	—	✓	✓	**Note 1**
F0603H		TRESULT4			—	✓		
F0604H	CTSU trimming register BL	TRESULT0	CTSUTRIM2	R/W	—	✓	✓	**Note 1**
F0605H		TRESULT1			—	✓		
F0606H	CTSU trimming register BH	TRESULT2	CTSUTRIM3	R/W	—	✓	✓	**Note 1**
F0607H		TRESULT3			—	✓		
F0640H	Function select register 0	REMCON0		R/W	✓	✓	—	00H
F0641H	Function select register 1	REMCON1		R/W	✓	✓	—	00H
F0642H	Status register	REMSTS		R/W	✓	✓	—	00H
F0643H	Interrupt control register	REMINT		R/W	✓	✓	—	00H
F0645H	Compare control register	REMCPC		R/W	—	✓	—	00H
F0646H	Compare value setting register	REMCPD		R/W	—	—	✓	0000H
F0647H								
F0648H	Header pattern minimum width setting register	HDPMIN		R/W	—	—	✓	0000H
F0649H								
F064AH	Header pattern maximum width setting register	HDPMAX		R/W	—	—	✓	0000H
F064BH								
F064CH	Data 0 pattern minimum width setting register	D0PMIN		R/W	—	✓	—	00H
F064DH	Data 0 pattern maximum width setting register	D0PMAX		R/W	—	✓	—	00H
F064EH	Data 1 pattern minimum width setting register	D1PMIN		R/W	—	✓	—	00H
F064FH	Data 1 pattern maximum width setting register	D1PMAX		R/W	—	✓	—	00H
F0650H	Special data pattern minimum width setting register	SDPMIN		R/W	—	—	✓	0000H
F0651H								

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (14/15)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0652H	Special data pattern maximum width setting register	SDPMAX	R/W	—	—	✓	0000H
F0653H							
F0654H	Pattern end setting register	REMPE	R/W	—	—	✓	0000H
F0655H							
F0656H	Receiver standby control register	REMSTC	R/W	✓	✓	—	00H
F0657H	Receive bit count register	REMRBIT	R/W	✓	✓	—	00H
F0658H	Receive data 0 register	REMDAT0	R/W	✓	✓	—	00H
F0659H	Receive data 1 register	REMDAT1	R	—	✓	—	00H
F065AH	Receive data 2 register	REMDAT2	R	—	✓	—	00H
F065BH	Receive data 3 register	REMDAT3	R	—	✓	—	00H
F065CH	Receive data 4 register	REMDAT4	R	—	✓	—	00H
F065DH	Receive data 5 register	REMDAT5	R	—	✓	—	00H
F065EH	Receive data 6 register	REMDAT6	R	—	✓	—	00H
F065FH	Receive data 7 register	REMDAT7	R	—	✓	—	00H
F0660H	Measurement result register	REMTIM	R	—	—	✓	0000H
F0661H							
F0680H	Input signal select register 0	ELISEL0	R/W	—	✓	—	00H
F0681H	Input signal select register 1	ELISEL1	R/W	—	✓	—	00H
F0682H	Input signal select register 2	ELISEL2	R/W	—	✓	—	00H
F0683H	Input signal select register 3	ELISEL3	R/W	—	✓	—	00H
F0684H	Input signal select register 4	ELISEL4	R/W	—	✓	—	00H
F0685H	Input signal select register 5	ELISEL5	R/W	—	✓	—	00H
F0686H	Input signal select register 6	ELISEL6	R/W	—	✓	—	00H
F0687H	Input signal select register 7	ELISEL7	R/W	—	✓	—	00H
F0688H	Input signal select register 8	ELISEL8	R/W	—	✓	—	00H
F0689H	Input signal select register 9	ELISEL9	R/W	—	✓	—	00H
F068AH	Input signal select register 10	ELISEL10	R/W	—	✓	—	00H
F068BH	Input signal select register 11	ELISEL11	R/W	—	✓	—	00H
F0690H	Event link L1 signal select register 0	ELL1SEL0	R/W	—	✓	—	00H
F0691H	Event link L1 signal select register 1	ELL1SEL1	R/W	—	✓	—	00H
F0692H	Event link L1 signal select register 2	ELL1SEL2	R/W	—	✓	—	00H
F0693H	Event link L1 signal select register 3	ELL1SEL3	R/W	—	✓	—	00H
F0694H	Event link L1 signal select register 4	ELL1SEL4	R/W	—	✓	—	00H
F0695H	Event link L1 signal select register 5	ELL1SEL5	R/W	—	✓	—	00H
F0696H	Event link L1 signal select register 6	ELL1SEL6	R/W	—	✓	—	00H
F0697H	Logic cell block L1 control register	ELL1CTL	R/W	—	✓	—	00H
F0698H	Event link L1 output select register 0	ELL1LNK0	R/W	—	✓	—	00H
F0699H	Event link L1 output select register 1	ELL1LNK1	R/W	—	✓	—	00H
F069AH	Event link L1 output select register 2	ELL1LNK2	R/W	—	✓	—	00H
F069BH	Event link L1 output select register 3	ELL1LNK3	R/W	—	✓	—	00H
F069CH	Event link L1 output select register 4	ELL1LNK4	R/W	—	✓	—	00H
F069DH	Event link L1 output select register 5	ELL1LNK5	R/W	—	✓	—	00H
F069EH	Event link L1 output select register 6	ELL1LNK6	R/W	—	✓	—	00H
F06A0H	Event link L2 signal select register 0	ELL2SEL0	R/W	—	✓	—	00H
F06A1H	Event link L2 signal select register 1	ELL2SEL1	R/W	—	✓	—	00H
F06A2H	Event link L2 signal select register 2	ELL2SEL2	R/W	—	✓	—	00H
F06A3H	Event link L2 signal select register 3	ELL2SEL3	R/W	—	✓	—	00H
F06A4H	Event link L2 signal select register 4	ELL2SEL4	R/W	—	✓	—	00H

Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (15/15)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F06A5H	Event link L2 signal select register 5	ELL2SEL5	R/W	—	✓	—	00H
F06A6H	Event link L2 signal select register 6	ELL2SEL6	R/W	—	✓	—	00H
F06A7H	Logic cell block L2 control register	ELL2CTL	R/W	—	✓	—	00H
F06A8H	Event link L2 output select register 0	ELL2LNK0	R/W	—	✓	—	00H
F06A9H	Event link L2 output select register 1	ELL2LNK1	R/W	—	✓	—	00H
F06AAH	Event link L2 output select register 2	ELL2LNK2	R/W	—	✓	—	00H
F06ABH	Event link L2 output select register 3	ELL2LNK3	R/W	—	✓	—	00H
F06ACH	Event link L2 output select register 4	ELL2LNK4	R/W	—	✓	—	00H
F06ADH	Event link L2 output select register 5	ELL2LNK5	R/W	—	✓	—	00H
F06AEH	Event link L2 output select register 6	ELL2LNK6	R/W	—	✓	—	00H
F06B0H	Event link L3 signal select register 0	ELL3SEL0	R/W	—	✓	—	00H
F06B1H	Event link L3 signal select register 1	ELL3SEL1	R/W	—	✓	—	00H
F06B2H	Event link L3 signal select register 2	ELL3SEL2	R/W	—	✓	—	00H
F06B3H	Event link L3 signal select register 3	ELL3SEL3	R/W	—	✓	—	00H
F06B4H	Event link L3 signal select register 4	ELL3SEL4	R/W	—	✓	—	00H
F06B5H	Event link L3 signal select register 5	ELL3SEL5	R/W	—	✓	—	00H
F06B6H	Event link L3 signal select register 6	ELL3SEL6	R/W	—	✓	—	00H
F06B7H	Logic cell block L3 control register	ELL3CTL	R/W	—	✓	—	00H
F06B8H	Event link L3 output select register 0	ELL3LNK0	R/W	—	✓	—	00H
F06B9H	Event link L3 output select register 1	ELL3LNK1	R/W	—	✓	—	00H
F06BAH	Event link L3 output select register 2	ELL3LNK2	R/W	—	✓	—	00H
F06BBH	Event link L3 output select register 3	ELL3LNK3	R/W	—	✓	—	00H
F06BCH	Event link L3 output select register 4	ELL3LNK4	R/W	—	✓	—	00H
F06BDH	Event link L3 output select register 5	ELL3LNK5	R/W	—	✓	—	00H
F06BEH	Event link L3 output select register 6	ELL3LNK6	R/W	—	✓	—	00H
F06C0H	Output signal select register 0	ELOSEL0	R/W	—	✓	—	00H
F06C1H	Output signal select register 1	ELOSEL1	R/W	—	✓	—	00H
F06C2H	Output signal select register 2	ELOSEL2	R/W	—	✓	—	00H
F06C3H	Output signal select register 3	ELOSEL3	R/W	—	✓	—	00H
F06C4H	Output signal select register 4	ELOSEL4	R/W	—	✓	—	00H
F06C5H	Output signal select register 5	ELOSEL5	R/W	—	✓	—	00H
F06C6H	Output signal select register 6	ELOSEL6	R/W	—	✓	—	00H
F06C7H	Output signal select register 7	ELOSEL7	R/W	—	✓	—	00H
F06C8H	Output signal enable register	ELOENCTL	R/W	—	✓	—	00H
F06C9H	Output signal monitor register	ELOMONI	R	—	✓	—	00H

Note 1. The value after a reset is adjusted at the time of shipment.

Note 2. The value after a reset is a value set in the FRQSEL2 to FRQSEL0 bits of the user option byte (000C2H).

Note 3. The initial value of the FLMODE register is set to the value of the MODE1 and MODE0 bits updated with the set value of the CMODE1 and CMODE0 bits in the option byte at address 000C2H.

Note 4. In the initial value of the OSMC register, the value of the RTCLPC and WUTMMCK0 bits is 0 and that of the HIPREC bit is 1.

Remark For SFRs in the SFR area, see **Table 3 - 5 List of Special Function Registers (SFRs)**.

3.3 Instruction Address Addressing

3.3.1 Relative addressing

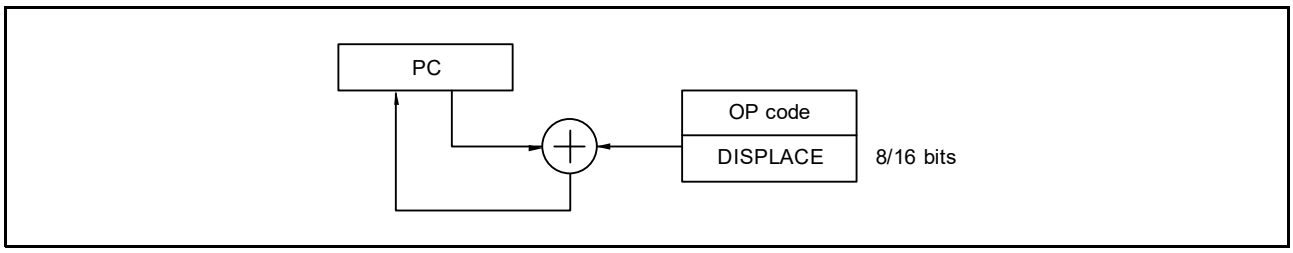
[Function]

In relative addressing, a displacement value[Note 1] included as the operand for the instruction is added to the program counter (PC) value[Note 2]. The resulting value is stored in the program counter and used as the branch destination address. Relative addressing is only applied to branch instructions.

Note 1. Signed two's complement data: –128 to +127 or –32768 to +32767

Note 2. The program counter value used in the calculation is the address where the next instruction starts.

Figure 3 - 16 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 - 17 Example of CALL !!addr20/BR !!addr20

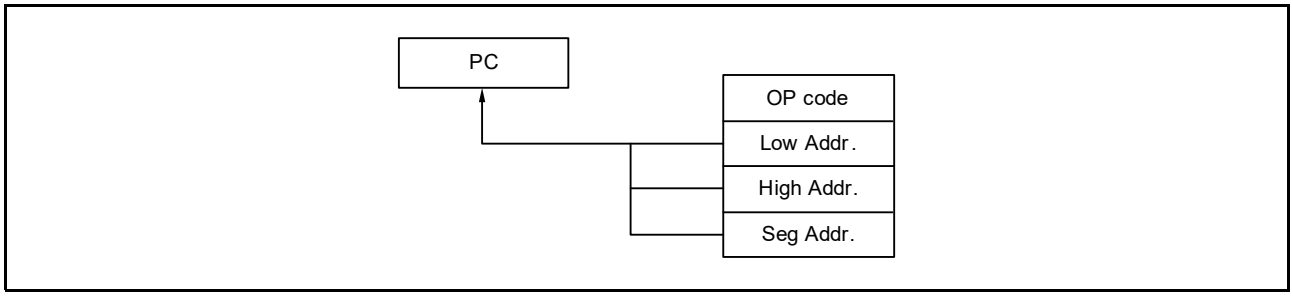
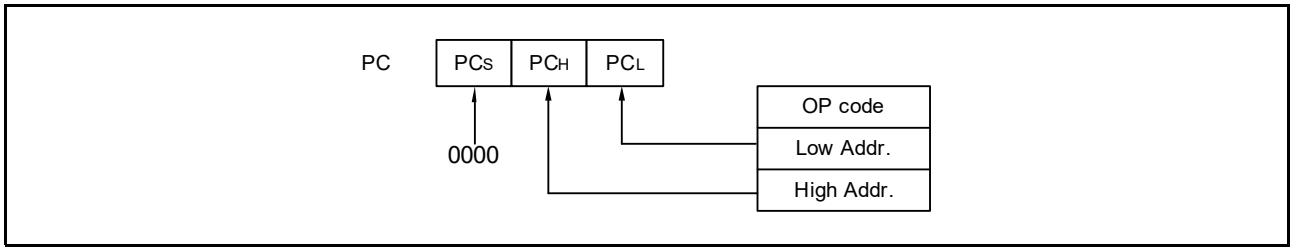


Figure 3 - 18 Example of CALL !addr16/BR !addr16

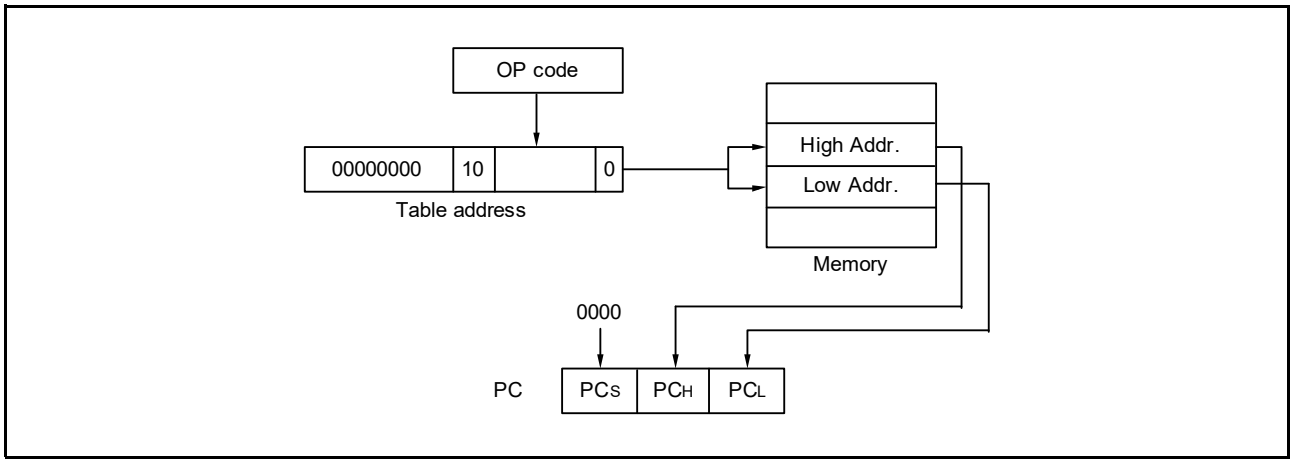


3.3.3 Table indirect addressing

[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-Kbyte space from 00000H to 0FFFFH.

Figure 3 - 19 Outline of Table Indirect Addressing

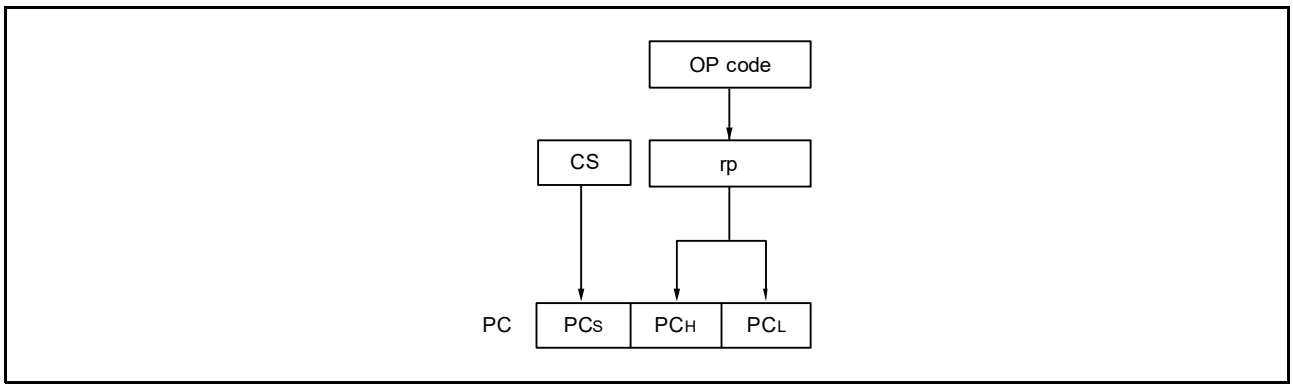


3.3.4 Register indirect addressing

[Function]

Register indirect 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 - 20 Outline of Register Indirect Addressing



3.4 Types of Addressing for Processing Data Addresses

3.4.1 Implied addressing

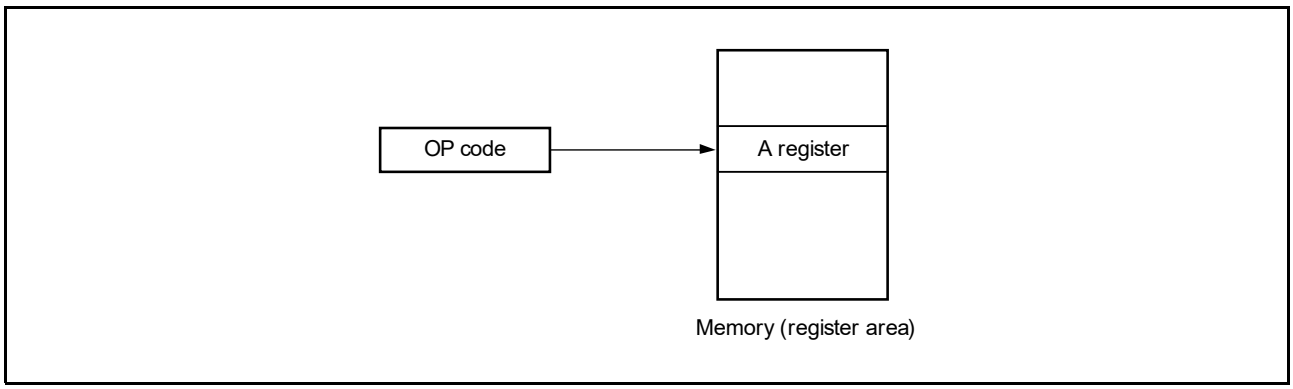
[Function]

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 - 21 Outline of Implied Addressing



3.4.2 Register addressing

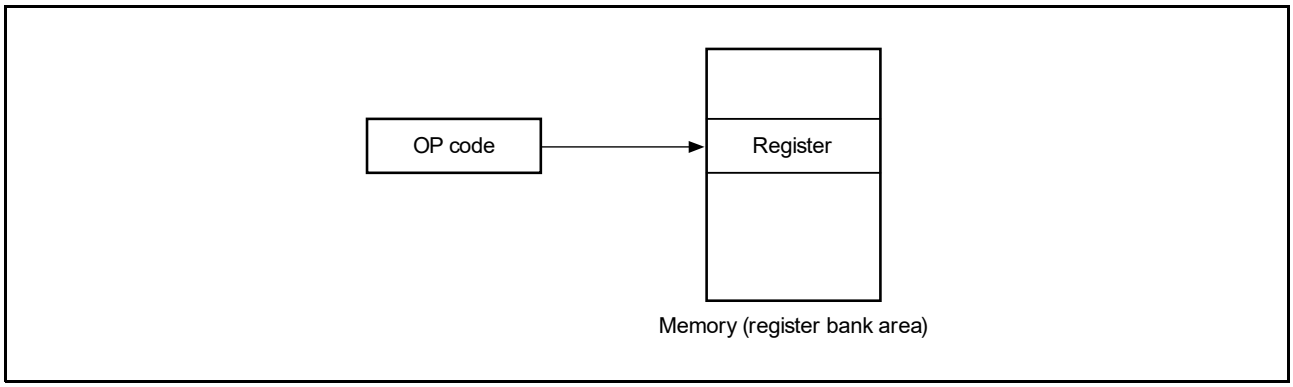
[Function]

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 - 22 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 - 23 Example of !addr16

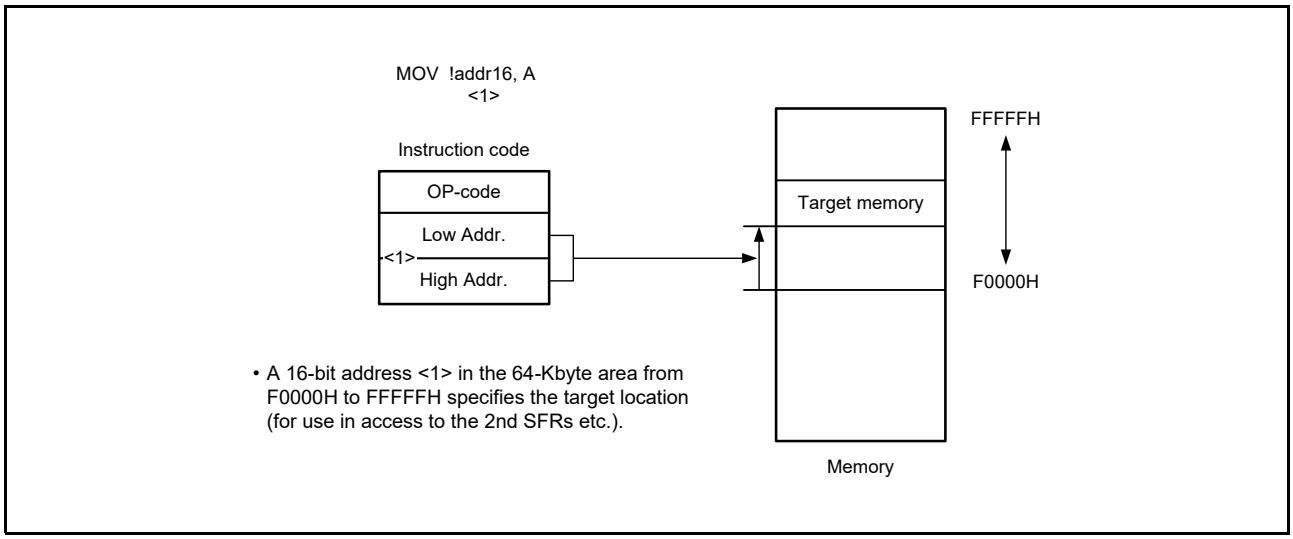
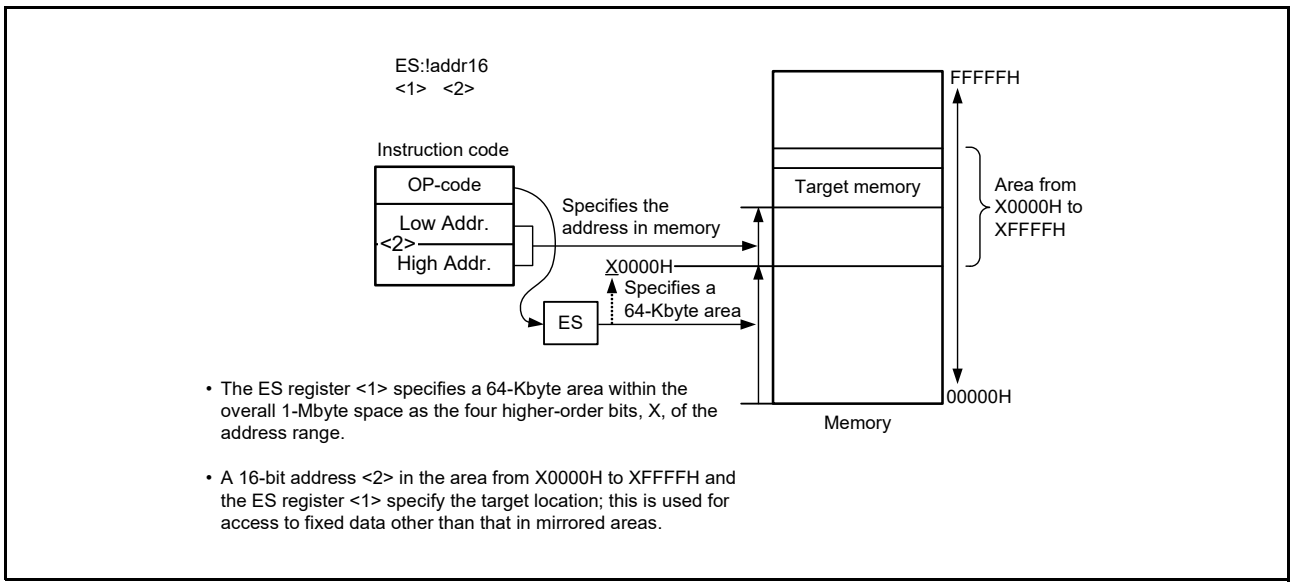


Figure 3 - 24 Example of ES:!addr16



3.4.4 Short direct addressing

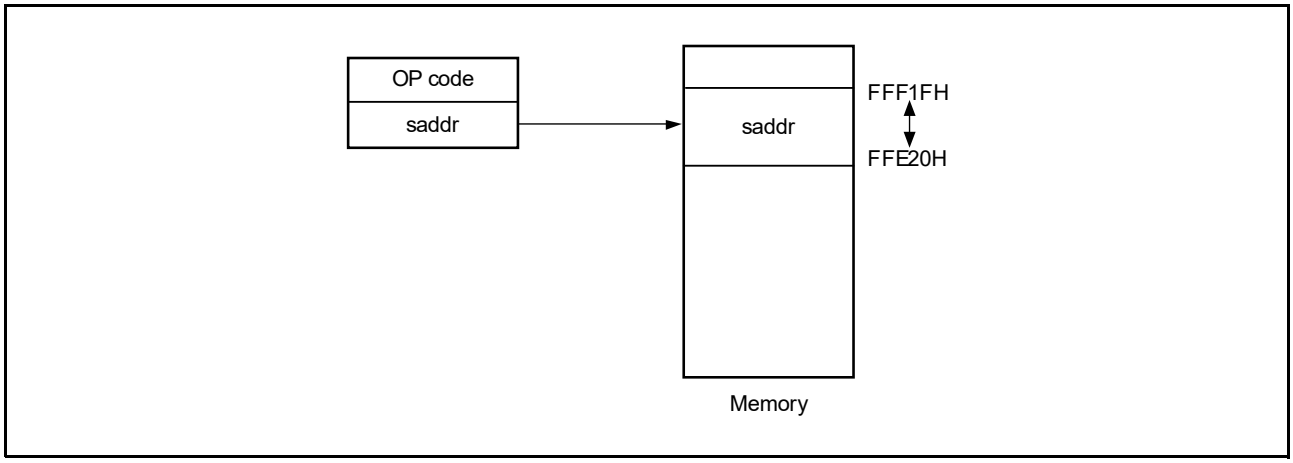
[Function]

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 - 25 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

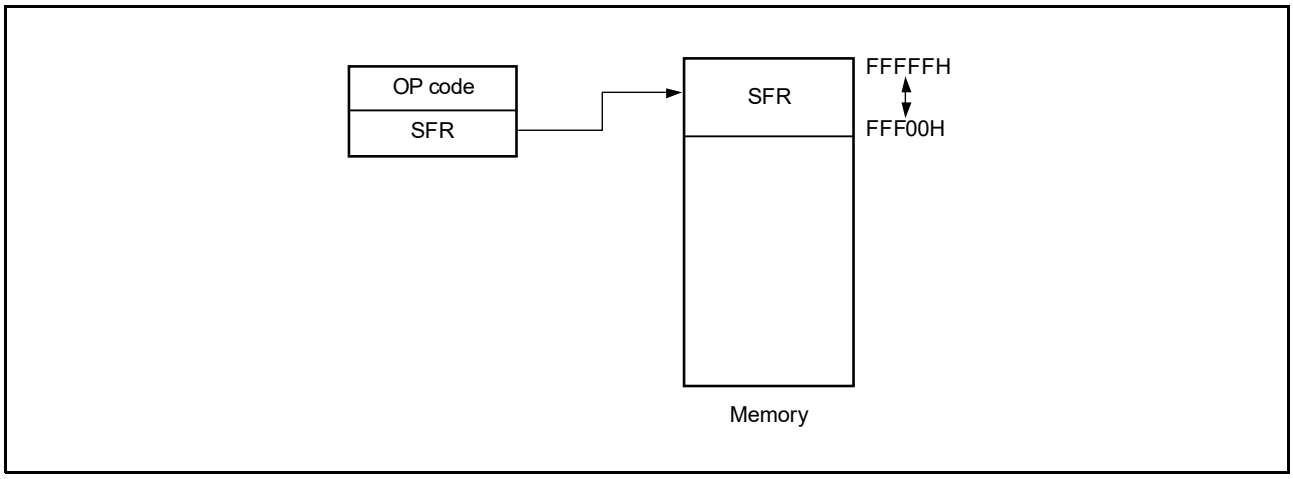
[Function]

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 - 26 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 - 27 Example of [DE], [HL]

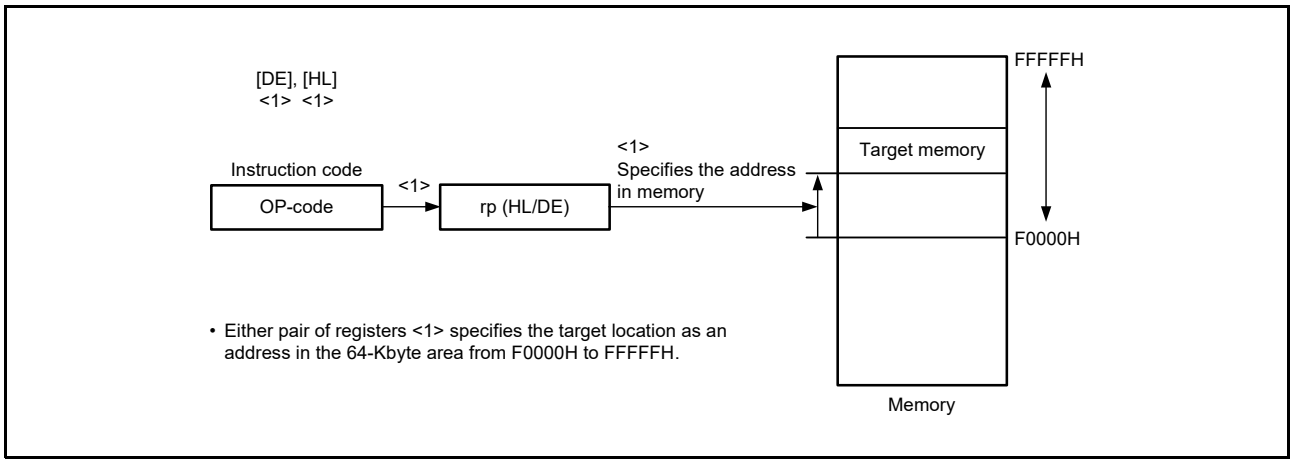
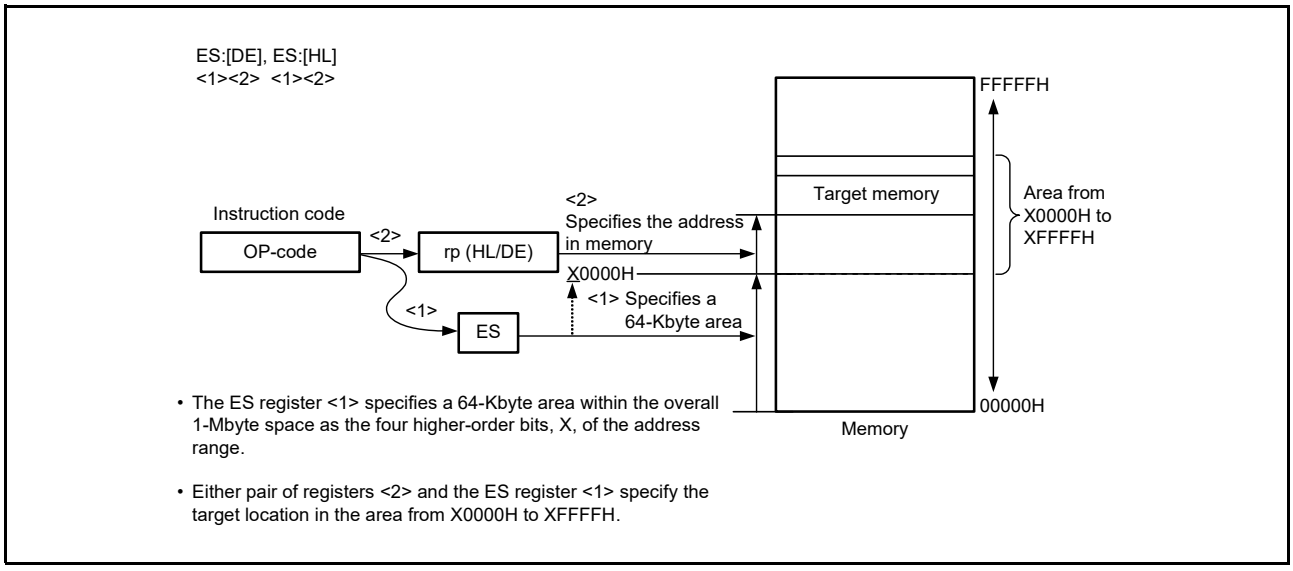


Figure 3 - 28 Example of ES:[DE], ES:[HL]



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 - 29 Example of [SP+byte]

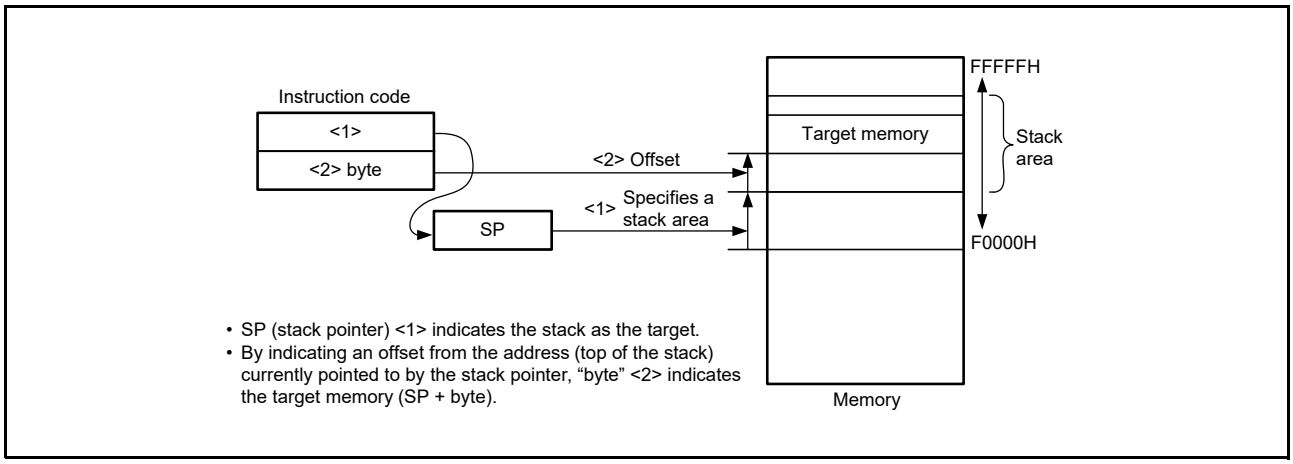


Figure 3 - 30 Example of [HL+byte], [DE+byte]

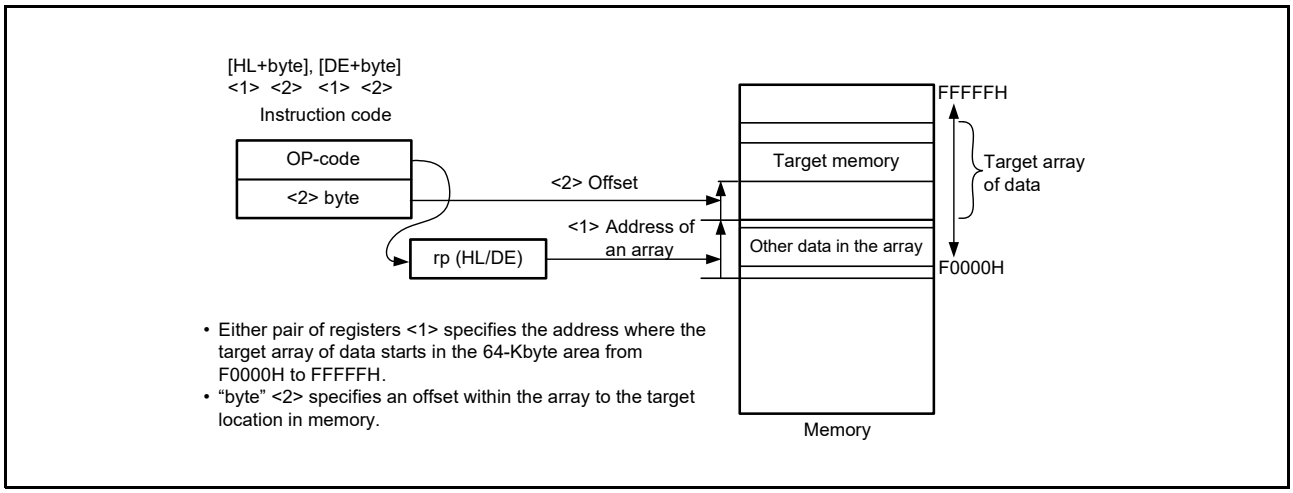


Figure 3 - 31 Example of word[B], word[C]

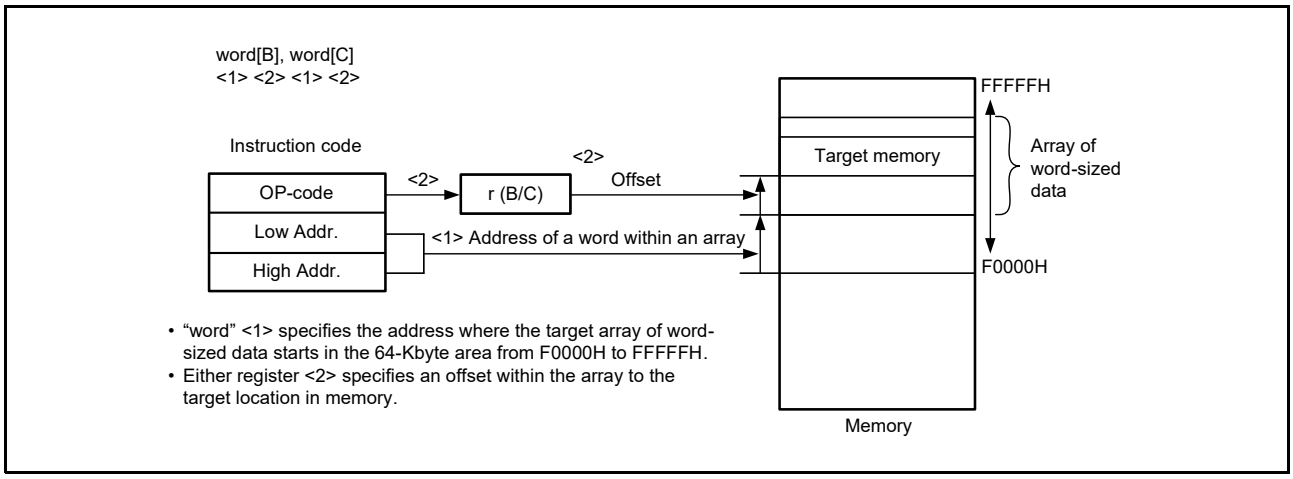


Figure 3 - 32 Example of word[BC]

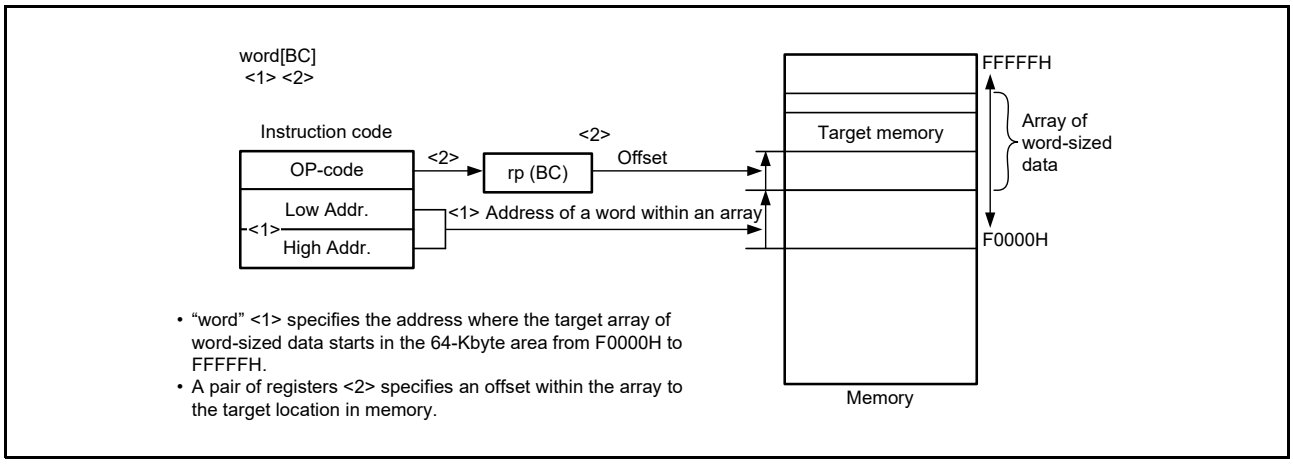


Figure 3 - 33 Example of ES:[HL+byte], ES:[DE+byte]

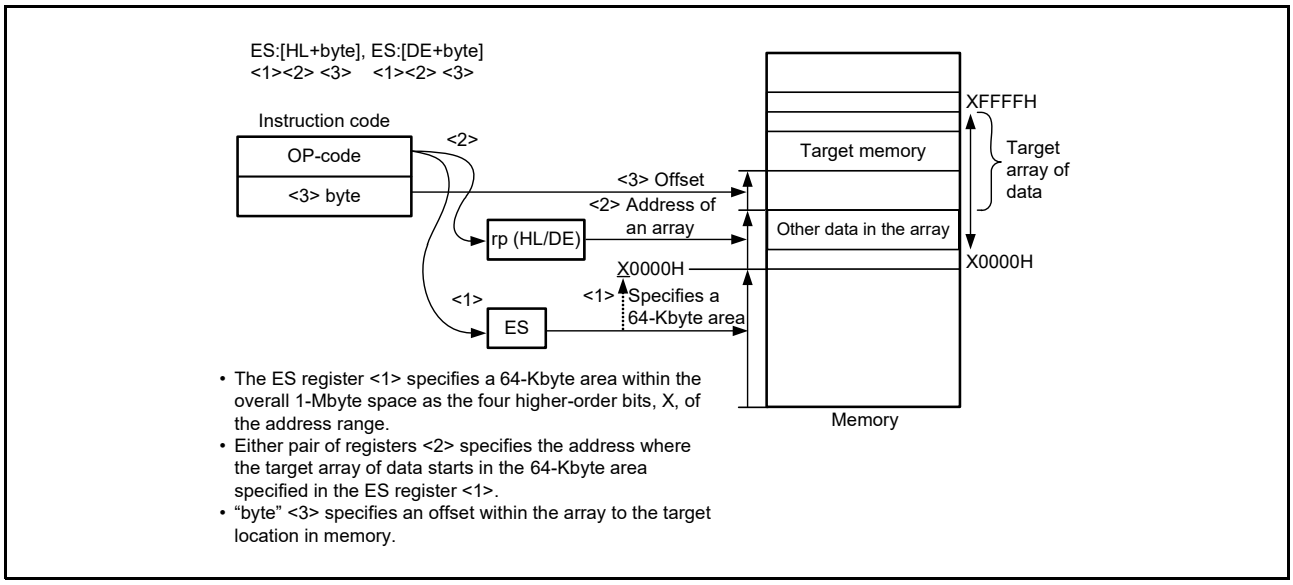
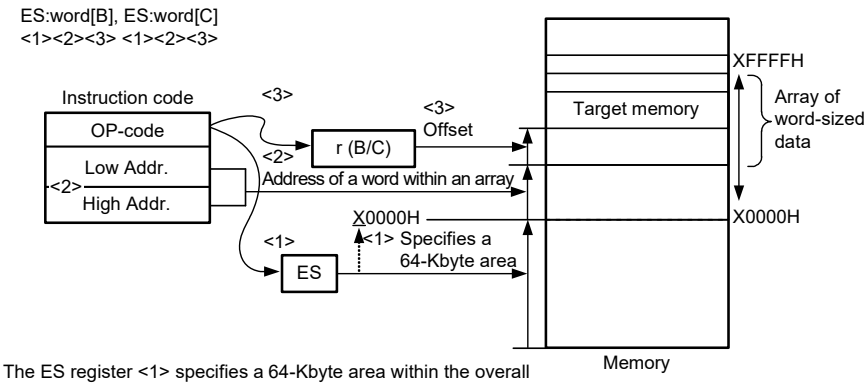
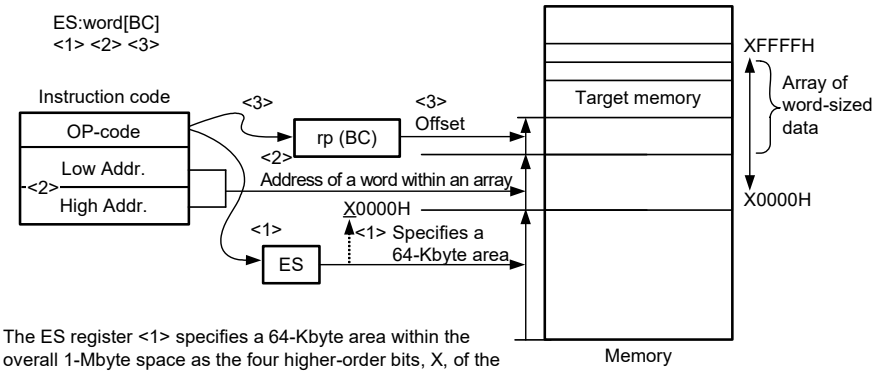


Figure 3 - 34 Example of ES:word[B], ES:word[C]



- 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 - 35 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 - 36 Example of [HL+B], [HL+C]

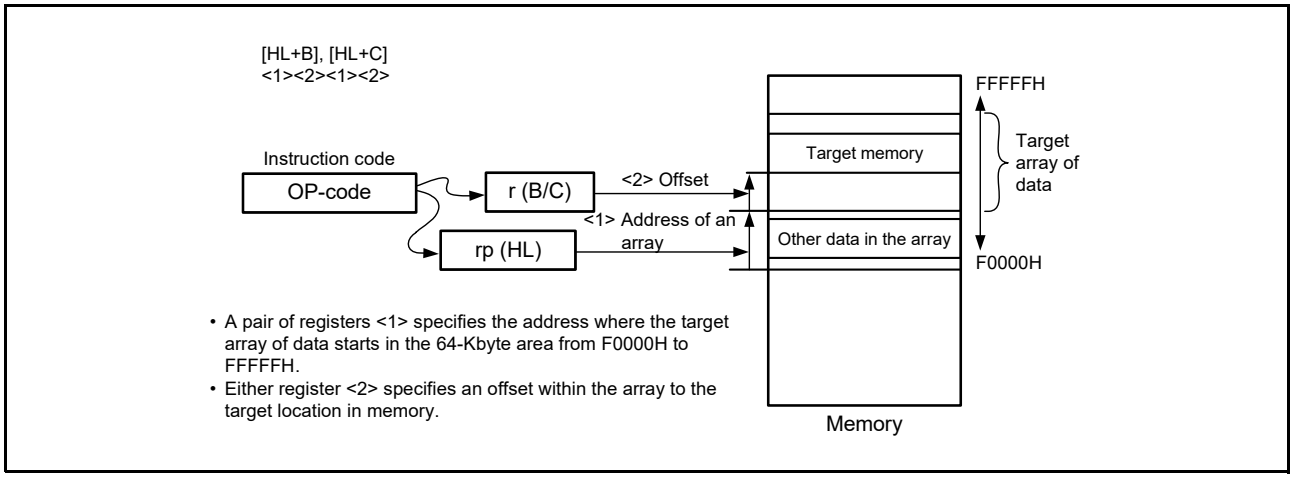
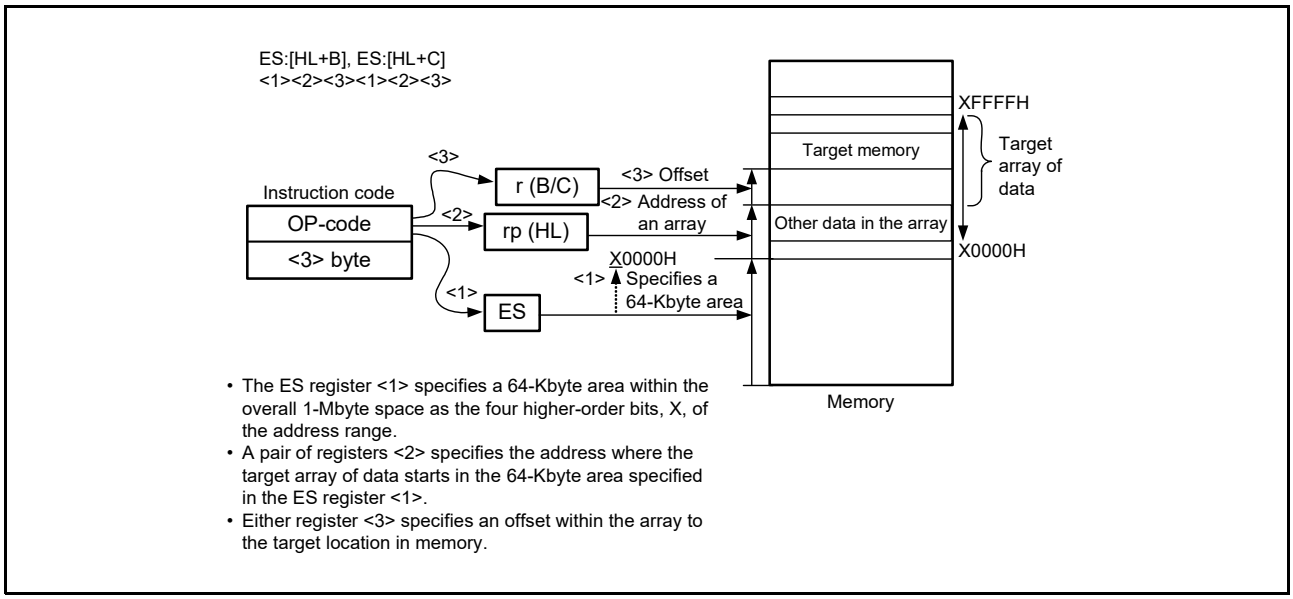


Figure 3 - 37 Example of ES:[HL+B], ES:[HL+C]



3.4.9 Stack addressing

[Function]

The stack area is indirectly addressed with the stack pointer (SP) values. 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

The data to be saved/restored by each stack operation is shown in **Figure 3 - 38** to **Figure 3 - 43**.

Figure 3 - 38 Example of PUSH rp

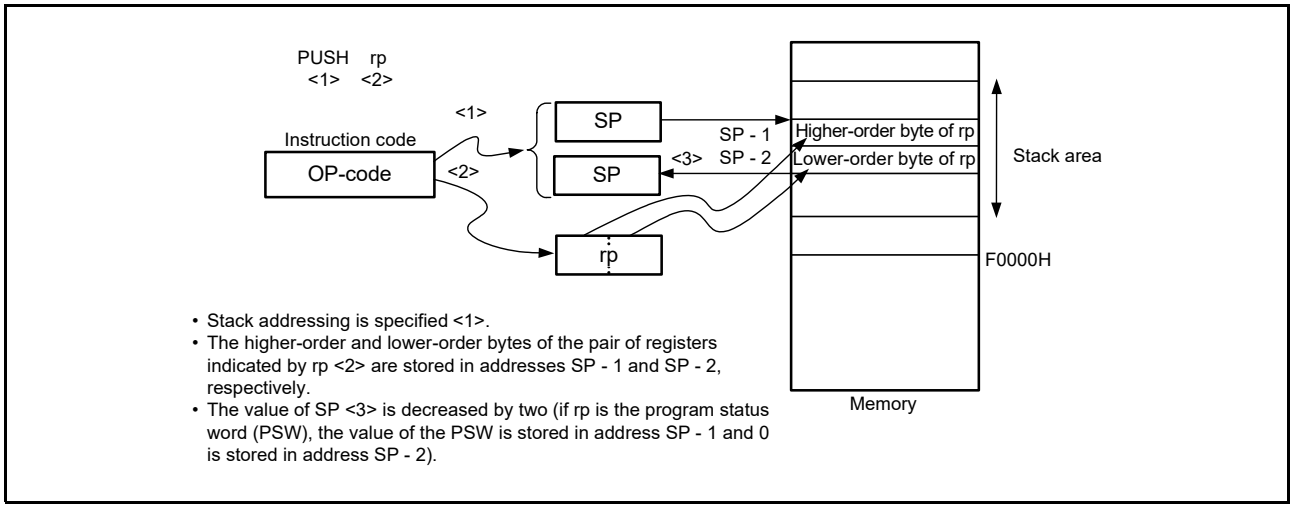
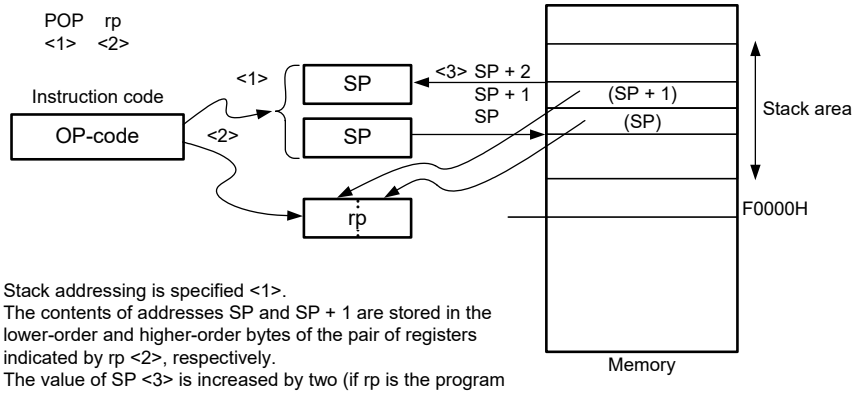
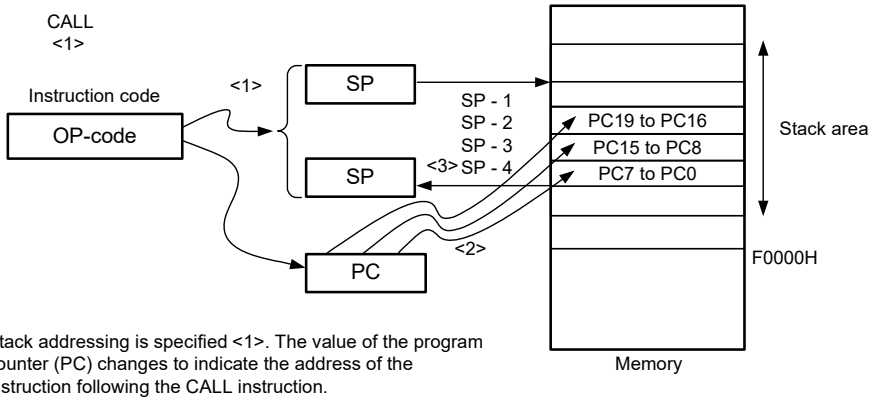


Figure 3 - 39 Example of POP



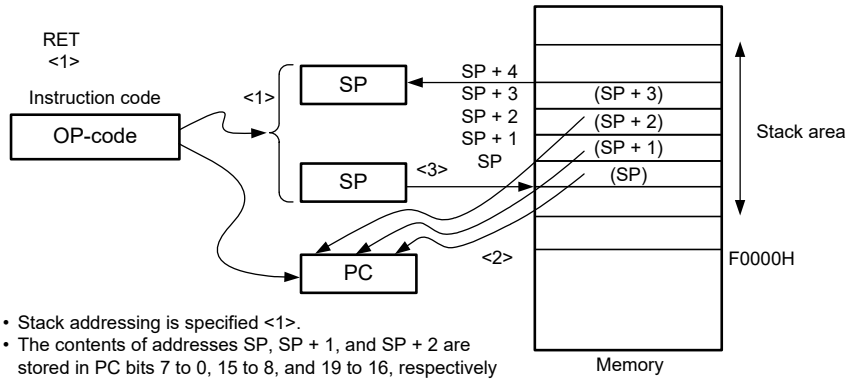
- Stack addressing is specified <1>.
- The contents of addresses SP and SP + 1 are stored in the lower-order and higher-order bytes of the pair of registers indicated by rp <2>, respectively.
- The value of SP <3> is increased by two (if rp is the program status word (PSW), the content of address SP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

Figure 3 - 40 Example of CALL, CALLT



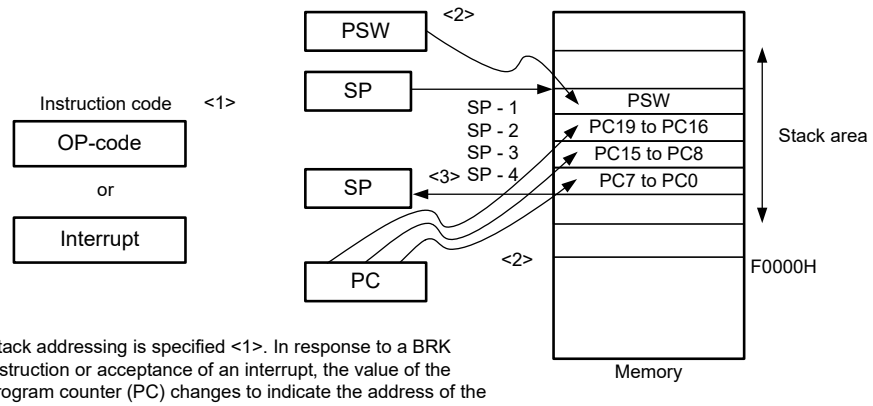
- Stack addressing is specified <1>. The value of the program counter (PC) changes to indicate the address of the instruction following the CALL instruction.
- The values of PC bits 19 to 16, 15 to 8, and 7 to 0 are stored in addresses SP - 2, SP - 3, and SP - 4, respectively <2>.
- The value of the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

Figure 3 - 41 Example of RET



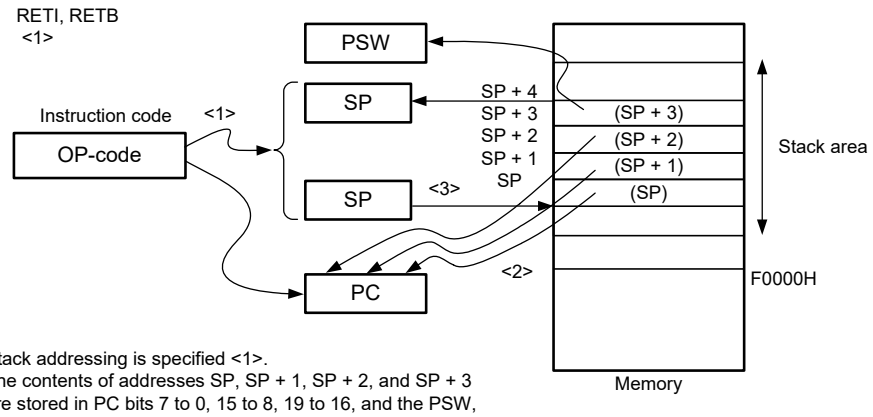
- Stack addressing is specified <1>.
- The contents of addresses SP, SP + 1, and SP + 2 are stored in PC bits 7 to 0, 15 to 8, and 19 to 16, respectively <2>.
- The value of SP <3> is increased by four.

Figure 3 - 42 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 - 43 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.

Section 4 Port Functions

4.1 Port Functions

This MCU has digital I/O ports, which enable variety of control operations. In addition to the function as digital I/O ports, these ports have several multiplexed functions. For details of the multiplexed functions, see **Section 2 Pin Functions**.

4.2 Port Configuration

Ports include the following hardware.

Table 4 - 1 Port Configuration (1/2)

Item	Configuration
Control registers	• Port mode registers (PMxx) (xx = 0 to 12, 14, 15) • Port registers (Pxx) (xx = 0 to 15) • Pull-up resistor option registers (PUxx) (xx = 0, 1, 3 to 12, 14) • Port input mode registers (PIMxx) (xx = 0, 1, 3 to 5, 7, 8, 14) • Port output mode registers (POMxx) (xx = 0, 1, 3 to 5, 7 to 9, 12, 14) • Port digital input disable registers (PDIDISxx) (xx = 0, 1, 3 to 5, 7 to 9, 12 to 14) • Port mode control A registers (PMCAxx) (xx = 0 to 3, 10 to 12, 14, 15) • Port mode control T registers (PMCTxx) (xx = 0, 2, 3, 5 to 7, 15) • Port mode control E registers (PMCExx) (xx = 0, 1, 5, 6) • Peripheral I/O redirection register (PIOR) • Global digital input disable register (GDIDIS) • Output current control enable register (CCDE) • Output current select registers (CCSx) (x = 0, 4 to 7) • 40-mA port output control register (PTDC) • Port function output enable registers (PFOE0, PFOE1) • Port mode select register (PMS)
Port	R7F100G • 30-pin products Total: 26 (CMOS I/O: 23 (N-ch open-drain I/O [withstand voltage of V_{DD}]: 10, controlled current drive port: 6), CMOS input: 1, N-ch open-drain I/O [withstand voltage of 6 V]: 2) • 32-pin products Total: 28 (CMOS I/O: 24 (N-ch open-drain I/O [withstand voltage of V_{DD}]: 10, controlled current drive port: 7), CMOS input: 1, N-ch open-drain I/O [withstand voltage of 6 V]: 3) • 36-pin products Total: 32 (CMOS I/O: 28 (N-ch open-drain I/O [withstand voltage of V_{DD}]: 12, controlled current drive port: 7), CMOS input: 1, N-ch open-drain I/O [withstand voltage of 6 V]: 3) • 40-pin products Total: 36 (CMOS I/O: 30 (N-ch open-drain I/O [withstand voltage of V_{DD}]: 12, controlled current drive port: 7), CMOS input: 3, N-ch open-drain I/O [withstand voltage of 6 V]: 3) • 44-pin products Total: 40 (CMOS I/O: 33 (N-ch open-drain I/O [withstand voltage of V_{DD}]: 12, controlled current drive port: 7), CMOS input: 3, N-ch open-drain I/O [withstand voltage of 6 V]: 4) • 48-pin products Total: 44 (CMOS I/O: 36 (N-ch open-drain I/O [withstand voltage of V_{DD}]: 13, controlled current drive port: 8), CMOS input: 3, CMOS output: 1, N-ch open-drain I/O [withstand voltage of 6 V]: 4) • 52-pin products Total: 48 (CMOS I/O: 40 (N-ch open-drain I/O [withstand voltage of V_{DD}]: 15, controlled current drive port: 8), CMOS input: 3, CMOS output: 1, N-ch open-drain I/O [withstand voltage of 6 V]: 4) • 64-pin products Total: 58 (CMOS I/O: 50 (N-ch open-drain I/O [withstand voltage of EV_{DD}]: 22**Note 1**/18**Note 2**, controlled current drive port: 8), CMOS input: 3, CMOS output: 1, N-ch open-drain I/O [withstand voltage of 6 V]: 4)

Table 4 - 1 Port Configuration (2/2)

Item	Configuration
Port	• 80-pin products Total: 74 (CMOS I/O: 66 (N-ch open-drain I/O [withstand voltage of EV_{DD}]: 27, controlled current drive port: 8), CMOS input: 3, CMOS output: 1, N-ch open-drain I/O [withstand voltage of 6 V]: 4) • 100-pin products Total: 92 (CMOS I/O: 84 (N-ch open-drain I/O [withstand voltage of EV_{DD}]: 31, controlled current drive port: 8), CMOS input: 3, CMOS output: 1, N-ch open-drain I/O [withstand voltage of 6 V]: 4) • 128-pin products Total: 120 (CMOS I/O: 112 (N-ch open-drain I/O [withstand voltage of EV_{DD}]: 33, controlled current drive port: 8), CMOS input: 3, CMOS output: 1, N-ch open-drain I/O [withstand voltage of 6 V]: 4) R7F104G • 32-pin product Total: 28 (CMOS I/O: 27 (N-ch open-drain I/O [withstand voltage of V_{DD}]: 10, controlled current drive port: 3), CMOS input: 1) • 44-pin product Total: 40 (CMOS I/O: 37 (N-ch open-drain I/O [withstand voltage of V_{DD}]: 11, controlled current drive port: 3), CMOS input: 3) • 48-pin product Total: 44 (CMOS I/O: 40 (N-ch open-drain I/O [withstand voltage of V_{DD}]: 12, controlled current drive port: 3), CMOS input: 3, CMOS output: 1)
Pull-up resistor	R7F100G • 30-pin products Total: 19 • 32-pin products Total: 20 • 36-pin products Total: 22 • 40-pin products Total: 23 • 44-pin products Total: 25 • 48-pin products Total: 28 • 52-pin products Total: 32 • 64-pin products Total: 42 • 80-pin products Total: 54 • 100-pin products Total: 69 • 128-pin products Total: 97 R7F104G • 32-pin product Total: 23 • 44-pin product Total: 27 • 48-pin product Total: 30

Note 1. For the R7F100G products with 192- to 768-Kbyte flash memory

Note 2. For the R7F100G products with 96- or 128-Kbyte flash memory

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 by the port mode register (PM0). When the P00 to P07 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by the pull-up resistor option register (PU0). Input to the P01, P03 and P04 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units by the port input mode register (PIM0). Output from the P00 and P02 to P04 pins can be set to the N-ch open-drain output (withstand voltage of V_{DD}**Note 1**/withstand voltage of EV_{DD}**Note 2**) mode in 1-bit units by the port output mode register (POM0). This port can also be used for timer I/O, A/D converter analog input, serial interface data I/O and clock I/O, capacitance measurement, and logic and event link controller I/O. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 4 - 7 Correspondence between Register Settings and Port Pin State**. P00 to P07 are set to the modes listed below following a reset.

- P00 and P01 pins of the 30- and 32-pin R7F100G products: Analog input mode
- P00, P01 and P04 to P07 pins of the 36- to 128-pin R7F100G products: Input mode
- P02 and P03 pins of the 36- to 128-pin R7F100G products: Analog input mode
- P00, P01, P05, and P06 pins of the 32- to 48-pin R7F104G products: Input mode
- P02 and P03 pins of the 32- to 48-pin R7F104G products: Analog input mode

Note 1. For the 30- to 52-pin R7F100G and 32- to 48-pin R7F104G products
Note 2. For the 64- to 128-pin R7F100G products

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 by the port mode register (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 the pull-up resistor option register (PU1). Input to the P10, P11, and P13 to P17 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units by the port input mode register (PIM1). Output from the P10 to P15 and P17 pins can be set to the N-ch open-drain output (withstand voltage of V_{DD}**Note 1**/withstand voltage of EV_{DD}**Note 2**) mode in 1-bit units by the port output mode register (POM1). Output from the P16 and P17 pins can be specified as controlled current drive port pins in 1-bit units by the output current control enable register (CCDE). This port can also be used for serial interface data I/O and clock I/O, UART data transmission and reception for external device connection when programming flash memory, clock/buzzer output, timer I/O, external interrupt request input, comparator reference voltage input, comparator output, and logic and event link controller I/O. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 4 - 7 Correspondence between Register Settings and Port Pin State.** P10 to P12 and P14 to P17 are set to input mode and P13 is set to analog input mode following a reset.

Note 1. For the 30- to 52-pin R7F100G and 32- to 48-pin R7F104G products
Note 2. For the 64- to 128-pin R7F100G products

4.2.3 Port 2

Port 2 is an I/O port with an output latch. Port 2 can be set to the input mode or output mode in 1-bit units by the port mode register (PM2). This port can also be used for A/D converter analog input, positive and negative reference voltage inputs of the A/D converter, D/A converter output, comparator reference voltage input, capacitance measurement, and logic and event link controller input. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 4 - 7 Correspondence between Register Settings and Port Pin State**. P20 to P27 are set to analog input mode following a reset.

4.2.4 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 by the port mode register (PM3). When the P30 to P37 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by the pull-up resistor option register (PU3). Input to the P33 pin can be specified through a normal input buffer or a TTL input buffer by the port input mode register (PIM3). Output from the P34 pin can be set to the N-ch open-drain output (withstand voltage of V_{DD}**Note 1**/withstand voltage of EV_{DD}**Note 2**) mode by the port output mode register (POM3). This port can also be used for external interrupt request input, realtime clock correction clock output, clock/buzzer output, timer I/O, serial interface data I/O and clock I/O, A/D converter analog input, comparator output, capacitance measurement, and logic and event link controller input. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 4 - 7 Correspondence between Register Settings and Port Pin State**. P30 to P34 are set to input mode and P35 to P37 are set to analog input mode following a reset.

Note 1. For the 30- to 52-pin R7F100G products
Note 2. For the 64- to 128-pin R7F100G products

4.2.5 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 by the port mode register (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 the pull-up resistor option register (PU4). Input to the P41, P43, and P44 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units by the port input mode register (PIM4). Output from the P42 to P45 pins can be set to the N-ch open-drain output (withstand voltage of V_{DD}**Note 1**/withstand voltage of EV_{DD}**Note 2**) mode in 1-bit units by the port output mode register (POM4). This port can also be used for data I/O for a flash memory programmer/debugger, timer I/O, serial interface data I/O and clock I/O, and external interrupt request input. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 4 - 7 Correspondence between Register Settings and Port Pin State**. P40 to P47 are set to input mode following a reset.

Note 1. For the 30- to 52-pin R7F100G products
Note 2. For the 64- to 128-pin R7F100G products

4.2.6 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 by the port mode register (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 the pull-up resistor option register (PU5). Input to the P53 to P55 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units by the port input mode register (PIM5). Output from the P50 and P52 to P55 pins can be set to the N-ch open-drain output (withstand voltage of V_{DD}**Note 1**/withstand voltage of EV_{DD}**Note 2**) mode in 1-bit units by the port output mode register (POM5). Output from the P50**Note 3** and P51 pins can be specified as controlled current drive port pins in 1-bit units by the output current control enable register (CCDE). This port can also be used for external interrupt request input, serial interface data I/O and clock I/O, capacitance measurement, and logic and event link controller I/O. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 4 - 7 Correspondence between Register Settings and Port Pin State**. P50 to P57 are set to input mode following a reset.

Note 1. For the 30- to 52-pin R7F100G and 32- to 48-pin R7F104G products
Note 2. For the 64- to 128-pin R7F100G products
Note 3. This applies to products other than the R7F100G with 44 pins.

4.2.7 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 by the port mode register (PM6). When the P64 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 the pull-up resistor option register (PU6). Output from the P60 to P63 pins is N-ch open-drain output (withstand voltage of 6 V). Output from the P60 to P63 pins can be specified as controlled current drive port pins in 1-bit units by the output current control enable register (CCDE). This port can also be used for serial interface data I/O and clock I/O, timer I/O, capacitance measurement, and logic and event link controller I/O. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 4 - 7 Correspondence between Register Settings and Port Pin State**. P60 to P67 are set to input mode following a reset.

4.2.8 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 by the port mode register (PM7). When this port is used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by the pull-up resistor option register (PU7). Input to the P71 pin can be specified through a normal input buffer or a TTL input buffer by the port input mode register (PIM7). Output from the P71, P72, and P74 pins can be set to the N-ch open-drain output (withstand voltage of V_{DD}**Note 1**/withstand voltage of EV_{DD}**Note 2**) mode in 1-bit units by the port output mode register (POM7). This port can also be used for key interrupt input, serial interface data I/O and clock I/O, external interrupt request input, capacitance measurement, and external pulse signal input for the remote controller signal reception function. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 4 - 7 Correspondence between Register Settings and Port Pin State**. P70 to P77 are set to input mode following a reset.

Note 1. For the 32- to 52-pin R7F100G and 32- to 48-pin R7F104G products
Note 2. For the 64- to 128-pin R7F100G products

4.2.9 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 by the port mode register (PM8). When this port is used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by the pull-up resistor option register (PU8). Input to the P80, P81, and P84 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units by the port input mode register (PIM8). Output from the P80 to P83 pins can be set to the N-ch open-drain output (withstand voltage of EV_{DD}) mode in 1-bit units by the port output mode register (POM8). This port can also be used for serial interface data I/O. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 4 - 7 Correspondence between Register Settings and Port Pin State**. P80 to P87 are set to input mode following a reset.

4.2.10 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 by the port mode register (PM9). When this port is used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by the pull-up resistor option register (PU9). Output from the P96 pin can be set to the N-ch open-drain output (withstand voltage of EV_{DD}) mode by the port output mode register (POM9). This port can also be used for serial interface data I/O and clock I/O. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 4 - 7 Correspondence between Register Settings and Port Pin State**. P90 to P97 are set to input mode following a reset.

4.2.11 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 by the port mode register (PM10). When this port is used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by the pull-up resistor option register (PU10). This port can also be used for timer I/O and A/D converter analog input. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 4 - 7 Correspondence between Register Settings and Port Pin State**. P100 is set to analog input mode and P101 to P106 are set to input mode following a reset.

4.2.12 Port 11

Port 11 is an I/O port with an output latch. Port 11 can be set to the input mode or output mode in 1-bit units by the port mode register (PM11). When this port is used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by the pull-up resistor option register (PU11). This port can also be used for A/D converter analog input. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 4 - 7 Correspondence between Register Settings and Port Pin State**. P110 to P114 are set to input mode and P115 to P117 are set to analog input mode following a reset.

4.2.13 Port 12

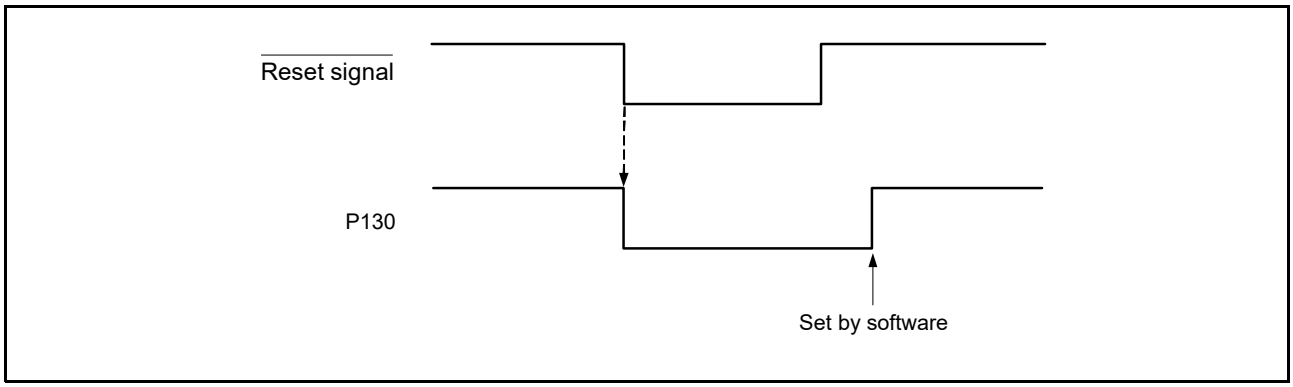
P120 to P122 and P125 to 127 are a 6-bit I/O port with an output latch. Port 12 can be set to the input mode or output mode in 1-bit units by the port mode register (PM12). When this port is used as an input port, use of an on-chip pull-up resistor can be specified by the pull-up resistor option register (PU12). Output from the P120 pin can be set to the N-ch open-drain output (withstand voltage of V_{DD}**Note 1**/withstand voltage of EV_{DD}**Note 2**) mode by the port output mode register (POM12). P123 and P124 are 2-bit input-only ports. This port can also be used for A/D converter analog input, comparator analog input, connection of a resonator for the main system clock, connection of a resonator for the subsystem clock, external clock input for the main system clock, external clock input for the subsystem clock, logic and event link controller input, and power supply for battery backup. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 4 - 7 Correspondence between Register Settings and Port Pin State**. P120 is set to analog input mode and P121 to P127 are set to input mode following a reset.

Note 1. For the 30- to 52-pin R7F100G products
Note 2. For the 64- to 128-pin R7F100G products

4.2.14 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 logic and event link controller input. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 4 - 7 Correspondence between Register Settings and Port Pin State**.

Remark When a reset signal is applied, P130 outputs a low-level signal. If P130 is set to a mode for outputting a high-level signal before a reset signal is applied, the P130 signal can be used to indicate a CPU reset.



4.2.15 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 by the port mode register (PM14). When the P140 to P147 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by the pull-up resistor option register (PU14). Input to the P142 and P143 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units by the port input mode register (PIM14). Output from the P142 to P144 pins can be set to the N-ch open-drain output (withstand voltage of EV_{DD}) mode in 1-bit units by the port output mode register (POM14). This port can also be used for clock/buzzer output, external interrupt request input, A/D converter analog input, serial interface data I/O and clock I/O, timer I/O, comparator analog input, and logic and event link controller input. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 4 - 7 Correspondence between Register Settings and Port Pin State**. P140 to P146 are set to input mode and P147 is set to analog input mode following a reset.

4.2.16 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 by the port mode register (PM15). This port can also be used for A/D converter analog input and capacitance measurement. Use the registers shown in **4.3 Registers for Controlling the Port Function** to specify the states of each of the pins. For the correspondence between register settings and pin state, see **Table 4 - 7 Correspondence between Register Settings and Port Pin State**. P150 to P156 are set to analog input mode following a reset.

4.3 Registers for Controlling the Port Function

The following registers are used to control the port functions.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Pull-up resistor option registers (PUxx)
- Port input mode registers (PIMxx)
- Port output mode registers (POMxx)
- Port digital input disable registers (PDIDISxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- Peripheral I/O redirection register (PIOR)
- Global digital input disable register (GDIDIS)
- Output current control enable register (CCDE)
- Output current select registers (CCSx)
- 40-mA port output control register (PTDC)
- Port function output enable registers (PFOEx)
- Port mode select register (PMS)

Caution The inclusion of registers and bit functions depends on the product. For details on registers and bit functions included in each product, see Table 4 - 2 to Table 4 - 4. Be sure to set bits to which no functions are allocated to their initial values.

Table 4 - 2 PMxx, Pxx, PUxx, PIMxx, POMxx, PDIDISxx, PMCAxx, PMCTxx, and PMCExx Registers and the Bits Implemented in 30- to 64-pin R7F100G Products with 96- or 128-Kbyte Flash Memory (1/3)

Port		Bit name									64 pins	52 pins	48 pins	44 pins	40 pins	36 pins	32 pins	30 pins
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PDIDISxx register	PMCAxx register	PMCTxx register	PMCExx register								
Port 0	0	PM00	P00	PU00	—	POM00	PDIDIS00	PMCA00 Note	—	—	✓	✓	✓	✓	✓	✓	✓	✓
	1	PM01	P01	PU01	PIM01	—	—	PMCA01 Note	—	PMCE01	✓	✓	✓	✓	✓	✓	✓	✓
	2	PM02	P02	PU02	—	POM02	PDIDIS02	PMCA02	—	—	✓	✓	—	—	—	—	—	—
	3	PM03	P03	PU03	PIM03	POM03	PDIDIS03	PMCA03	—	—	✓	✓	—	—	—	—	—	—
	4	PM04	P04	PU04	PIM04	POM04	PDIDIS04	—	—	—	✓	—	—	—	—	—	—	—
	5	PM05	P05	PU05	—	—	—	—	PMCT05	—	✓	—	—	—	—	—	—	—
	6	PM06	P06	PU06	—	—	—	—	PMCT06	—	✓	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Port 1	0	PM10	P10	PU10	PIM10	POM10	PDIDIS10	—	—	PMCE10	✓	✓	✓	✓	✓	✓	✓	✓
	1	PM11	P11	PU11	PIM11	POM11	PDIDIS11	—	—	PMCE11	✓	✓	✓	✓	✓	✓	✓	✓
	2	PM12	P12	PU12	—	POM12	PDIDIS12	—	—	PMCE12	✓	✓	✓	✓	✓	✓	✓	✓
	3	PM13	P13	PU13	PIM13	POM13	PDIDIS13	PMCA13	—	PMCE13	✓	✓	✓	✓	✓	✓	✓	✓
	4	PM14	P14	PU14	PIM14	POM14	PDIDIS14	—	—	PMCE14	✓	✓	✓	✓	✓	✓	✓	✓
	5	PM15	P15	PU15	PIM15	POM15	PDIDIS15	—	—	PMCE15	✓	✓	✓	✓	✓	✓	✓	✓
	6	PM16	P16	PU16	PIM16	—	—	—	—	PMCE16	✓	✓	✓	✓	✓	✓	✓	✓
	7	PM17	P17	PU17	PIM17	POM17	PDIDIS17	—	—	PMCE17	✓	✓	✓	✓	✓	✓	✓	✓
Port 2	0	PM20	P20	—	—	—	—	PMCA20	—	—	✓	✓	✓	✓	✓	✓	✓	✓
	1	PM21	P21	—	—	—	—	PMCA21	—	—	✓	✓	✓	✓	✓	✓	✓	✓
	2	PM22	P22	—	—	—	—	PMCA22	—	—	✓	✓	✓	✓	✓	✓	✓	✓
	3	PM23	P23	—	—	—	—	PMCA23	—	—	✓	✓	✓	✓	✓	✓	✓	✓
	4	PM24	P24	—	—	—	—	PMCA24	—	—	✓	✓	✓	✓	✓	✓	—	—
	5	PM25	P25	—	—	—	—	PMCA25	—	—	✓	✓	✓	✓	✓	✓	—	—
	6	PM26	P26	—	—	—	—	PMCA26	—	—	✓	✓	✓	✓	✓	—	—	—
	7	PM27	P27	—	—	—	—	PMCA27	—	—	✓	✓	✓	✓	—	—	—	—
Port 3	0	PM30	P30	PU30	—	—	—	—	PMCT30	—	✓	✓	✓	✓	✓	✓	✓	✓
	1	PM31	P31	PU31	—	—	—	—	PMCT31	—	✓	✓	✓	✓	✓	✓	✓	✓
	2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Port 4	0	PM40	P40	PU40	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓
	1	PM41	P41	PU41	PIM41	—	—	—	—	—	✓	✓	✓	✓	—	—	—	—
	2	PM42	P42	PU42	—	POM42	PDIDIS42	—	—	—	✓	—	—	—	—	—	—	—
	3	PM43	P43	PU43	—	—	—	—	—	—	✓	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 4 - 2 PMxx, Pxx, PUxx, PIMxx, POMxx, PDIDISxx, PMCAxx, PMCTxx, and PMCExx Registers and the Bits Implemented in 30- to 64-pin R7F100G Products with 96- or 128-Kbyte Flash Memory (2/3)

Port		Bit name									64 pins	52 pins	48 pins	44 pins	40 pins	36 pins	32 pins	30 pins
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PDIDISxx register	PMCAxx register	PMCTxx register	PMCExx register								
Port 5	0	PM50	P50	PU50	—	POM50	PDIDIS50	—	PMCT50	PMCE50	✓	✓	✓	✓	✓	✓	✓	✓
	1	PM51	P51	PU51	—	—	—	—	—	PMCE51	✓	✓	✓	✓	✓	✓	✓	✓
	2	PM52	P52	PU52	—	—	—	—	—	—	✓	—	—	—	—	—	—	—
	3	PM53	P53	PU53	—	—	—	—	—	—	✓	—	—	—	—	—	—	—
	4	PM54	P54	PU54	—	—	—	—	—	—	✓	—	—	—	—	—	—	—
	5	PM55	P55	PU55	PIM55	POM55	PDIDIS55	—	—	—	✓	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Port 6	0	PM60	P60	—	—	—	—	—	—	PMCE60	✓	✓	✓	✓	✓	✓	✓	✓
	1	PM61	P61	—	—	—	—	—	—	PMCE61	✓	✓	✓	✓	✓	✓	✓	✓
	2	PM62	P62	—	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	—
	3	PM63	P63	—	—	—	—	—	—	—	✓	✓	✓	✓	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Port 7	0	PM70	P70	PU70	—	—	—	—	PMCT70	—	✓	✓	✓	✓	✓	✓	✓	—
	1	PM71	P71	PU71	PIM71	POM71	PDIDIS71	—	PMCT71	—	✓	✓	✓	✓	✓	✓	—	—
	2	PM72	P72	PU72	—	POM72	PDIDIS72	—	PMCT72	—	✓	✓	✓	✓	✓	✓	—	—
	3	PM73	P73	PU73	—	—	—	—	PMCT73	—	✓	✓	✓	✓	✓	—	—	—
	4	PM74	P74	PU74	—	POM74	PDIDIS74	—	PMCT74	—	✓	✓	✓	—	—	—	—	—
	5	PM75	P75	PU75	—	—	—	—	PMCT75	—	✓	✓	✓	—	—	—	—	—
	6	PM76	P76	PU76	—	—	—	—	PMCT76	—	✓	✓	—	—	—	—	—	—
	7	PM77	P77	PU77	—	—	—	—	PMCT77	—	✓	✓	—	—	—	—	—	—
Port 8	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Port 9	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Port 10	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Port 11	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Port 12	0	PM120	P120	PU120	—	POM120	PDIDIS 120	PMCA 120	—	—	✓	✓	✓	✓	✓	✓	✓	✓
	1	PM121	P121	PU121	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓
	2	PM122	P122	PU122	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓
	3	—	P123	—	—	—	—	—	—	—	✓	✓	✓	✓	✓	—	—	—
	4	—	P124	—	—	—	—	—	—	—	✓	✓	✓	✓	✓	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 4 - 2 PMxx, Pxx, PUxx, PIMxx, POMxx, PDIDISxx, PMCAxx, PMCTxx, and PMCExx Registers and the Bits Implemented in 30- to 64-pin R7F100G Products with 96- or 128-Kbyte Flash Memory (3/3)

Port		Bit name									64 pins	52 pins	48 pins	44 pins	40 pins	36 pins	32 pins	30 pins
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PDIDISxx register	PMCAxx register	PMCTxx register	PMCExx register								
Port 13	0	—	P130	—	—	—	—	—	—	—	✓	✓	✓	—	—	—	—	—
	1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	P137	—	—	—	PDIDIS 137	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓
Port 14	0	PM140	P140	PU140	—	—	—	—	—	—	✓	✓	✓	—	—	—	—	—
	1	PM141	P141	PU141	—	—	—	—	—	—	✓	—	—	—	—	—	—	—
	2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	6	PM146	P146	PU146	—	—	—	—	—	—	✓	✓	✓	✓	—	—	—	—
	7	PM147	P147	PU147	—	—	—	PMCA 147	—	—	✓	✓	✓	✓	✓	✓	✓	✓
Port 15	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Note This bit is only present in the R7F100G products with 30 and 32 pins.

Table 4 - 3 PMxx, Pxx, PUxx, PIMxx, POMxx, PDIDISxx, PMCAxx, PMCTxx, and PMCExx Registers and the Bits Implemented in R7F100G Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin R7F100G Products with 128-Kbyte Flash Memory (1/4)

Port		Bit name									128 pins	100 pins	80 pins	64 pins	52 pins	48 pins	44 pins	40 pins	36 pins	32 pins	30 pins
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PDIDISxx register	PMCAxx register	PMCTxx register	PMCExx register											
Port 0	0	PM00	P00	PU00	—	POM00	PDIDIS00	PMCA00 Note	PMCT00	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	1	PM01	P01	PU01	PIM01	—	—	PMCA01 Note	PMCT01	PMCE01	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	2	PM02	P02	PU02	—	POM02	PDIDIS02	PMCA02	PMCT02	—	✓	✓	✓	✓	✓	—	—	—	—	—	—
	3	PM03	P03	PU03	PIM03	POM03	PDIDIS03	PMCA03	PMCT03	—	✓	✓	✓	✓	✓	—	—	—	—	—	—
	4	PM04	P04	PU04	PIM04	POM04	PDIDIS04	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—	—
	5	PM05	P05	PU05	—	—	—	—	PMCT05	—	✓	✓	✓	✓	—	—	—	—	—	—	—
	6	PM06	P06	PU06	—	—	—	—	PMCT06	—	✓	✓	✓	✓	—	—	—	—	—	—	—
	7	PM07	P07	PU07	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
Port 1	0	PM10	P10	PU10	PIM10	POM10	PDIDIS10	—	—	PMCE10	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	1	PM11	P11	PU11	PIM11	POM11	PDIDIS11	—	—	PMCE11	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	2	PM12	P12	PU12	—	POM12	PDIDIS12	—	—	PMCE12	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	3	PM13	P13	PU13	PIM13	POM13	PDIDIS13	PMCA13	—	PMCE13	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	4	PM14	P14	PU14	PIM14	POM14	PDIDIS14	—	—	PMCE14	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	5	PM15	P15	PU15	PIM15	POM15	PDIDIS15	—	—	PMCE15	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	6	PM16	P16	PU16	PIM16	—	—	—	—	PMCE16	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	7	PM17	P17	PU17	PIM17	POM17	PDIDIS17	—	—	PMCE17	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Port 2	0	PM20	P20	—	—	—	—	PMCA20	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	1	PM21	P21	—	—	—	—	PMCA21	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	2	PM22	P22	—	—	—	—	PMCA22	PMCT22	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	3	PM23	P23	—	—	—	—	PMCA23	PMCT23	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	4	PM24	P24	—	—	—	—	PMCA24	PMCT24	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
	5	PM25	P25	—	—	—	—	PMCA25	PMCT25	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
	6	PM26	P26	—	—	—	—	PMCA26	PMCT26	—	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—
	7	PM27	P27	—	—	—	—	PMCA27	PMCT27	—	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
Port 3	0	PM30	P30	PU30	—	—	—	—	PMCT30	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	1	PM31	P31	PU31	—	—	—	—	PMCT31	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	2	PM32	P32	PU32	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
	3	PM33	P33	PU33	PIM33	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
	4	PM34	P34	PU34	—	POM34	PDIDIS34	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
	5	PM35	P35	PU35	—	—	—	PMCA35	—	—	✓	—	—	—	—	—	—	—	—	—	—
	6	PM36	P36	PU36	—	—	—	PMCA36	—	—	✓	—	—	—	—	—	—	—	—	—	—
	7	PM37	P37	PU37	—	—	—	PMCA37	—	—	✓	—	—	—	—	—	—	—	—	—	—
Port 4	0	PM40	P40	PU40	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	1	PM41	P41	PU41	PIM41	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
	2	PM42	P42	PU42	—	POM42	PDIDIS42	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—	—
	3	PM43	P43	PU43	PIM43	POM43	PDIDIS43	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—	—
	4	PM44	P44	PU44	PIM44	POM44	PDIDIS44	—	—	—	✓	✓	✓	—	—	—	—	—	—	—	—
	5	PM45	P45	PU45	—	POM45	PDIDIS45	—	—	—	✓	✓	✓	—	—	—	—	—	—	—	—
	6	PM46	P46	PU46	—	—	—	—	—	—	✓	✓	—	—	—	—	—	—	—	—	—
	7	PM47	P47	PU47	—	—	—	—	—	—	✓	✓	—	—	—	—	—	—	—	—	—

Table 4 - 3 PMxx, Pxx, PUxx, PIMxx, POMxx, PDIDISxx, PMCAxx, PMCTxx, and PMCExx Registers and the Bits Implemented in R7F100G Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin R7F100G Products with 128-Kbyte Flash Memory (2/4)

Port		Bit name									128 pins	100 pins	80 pins	64 pins	52 pins	48 pins	44 pins	40 pins	36 pins	32 pins	30 pins
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PDIDISxx register	PMCAxx register	PMCTxx register	PMCExx register											
Port 5	0	PM50	P50	PU50	—	POM50	PDIDIS50	—	PMCT50	PMCE50	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	1	PM51	P51	PU51	—	—	—	—	—	PMCE51	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	2	PM52	P52	PU52	—	POM52	PDIDIS52	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—	—
	3	PM53	P53	PU53	PIM53	POM53	PDIDIS53	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—	—
	4	PM54	P54	PU54	PIM54	POM54	PDIDIS54	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—	—
	5	PM55	P55	PU55	PIM55	POM55	PDIDIS55	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—	—
	6	PM56	P56	PU56	—	—	—	—	—	—	✓	✓	—	—	—	—	—	—	—	—	—
	7	PM57	P57	PU57	—	—	—	—	—	—	✓	✓	—	—	—	—	—	—	—	—	—
Port 6	0	PM60	P60	—	—	—	—	—	—	PMCE60	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	1	PM61	P61	—	—	—	—	—	—	PMCE61	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	2	PM62	P62	—	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
	3	PM63	P63	—	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
	4	PM64	P64	PU64	—	—	—	—	PMCT64	—	✓	✓	✓	—	—	—	—	—	—	—	—
	5	PM65	P65	PU65	—	—	—	—	PMCT65	—	✓	✓	✓	—	—	—	—	—	—	—	—
	6	PM66	P66	PU66	—	—	—	—	PMCT66	—	✓	✓	✓	—	—	—	—	—	—	—	—
	7	PM67	P67	PU67	—	—	—	—	PMCT67	—	✓	✓	✓	—	—	—	—	—	—	—	—
Port 7	0	PM70	P70	PU70	—	—	—	—	PMCT70	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
	1	PM71	P71	PU71	PIM71	POM71	PDIDIS71	—	PMCT71	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
	2	PM72	P72	PU72	—	POM72	PDIDIS72	—	PMCT72	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
	3	PM73	P73	PU73	—	—	—	—	PMCT73	—	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—
	4	PM74	P74	PU74	—	POM74	PDIDIS74	—	PMCT74	—	✓	✓	✓	✓	✓	✓	—	—	—	—	—
	5	PM75	P75	PU75	—	—	—	—	PMCT75	—	✓	✓	✓	✓	✓	✓	—	—	—	—	—
	6	PM76	P76	PU76	—	—	—	—	PMCT76	—	✓	✓	✓	✓	✓	—	—	—	—	—	—
	7	PM77	P77	PU77	—	—	—	—	PMCT77	—	✓	✓	✓	✓	✓	—	—	—	—	—	—
Port 8	0	PM80	P80	PU80	PIM80	POM80	PDIDIS80	—	—	—	✓	✓	—	—	—	—	—	—	—	—	—
	1	PM81	P81	PU81	PIM81	POM81	PDIDIS81	—	—	—	✓	✓	—	—	—	—	—	—	—	—	—
	2	PM82	P82	PU82	—	POM82	PDIDIS82	—	—	—	✓	✓	—	—	—	—	—	—	—	—	—
	3	PM83	P83	PU83	—	POM83	PDIDIS83	—	—	—	✓	✓	—	—	—	—	—	—	—	—	—
	4	PM84	P84	PU84	PIM84	—	—	—	—	—	✓	✓	—	—	—	—	—	—	—	—	—
	5	PM85	P85	PU85	—	—	—	—	—	—	✓	✓	—	—	—	—	—	—	—	—	—
	6	PM86	P86	PU86	—	—	—	—	—	—	✓	✓	—	—	—	—	—	—	—	—	—
	7	PM87	P87	PU87	—	—	—	—	—	—	✓	✓	—	—	—	—	—	—	—	—	—
Port 9	0	PM90	P90	PU90	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
	1	PM91	P91	PU91	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
	2	PM92	P92	PU92	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
	3	PM93	P93	PU93	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
	4	PM94	P94	PU94	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
	5	PM95	P95	PU95	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
	6	PM96	P96	PU96	—	POM96	PDIDIS96	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
	7	PM97	P97	PU97	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—

Table 4 - 3 PMxx, Pxx, PUxx, PIMxx, POMxx, PDIDISxx, PMCAxx, PMCTxx, and PMCExx Registers and the Bits Implemented in R7F100G Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin R7F100G Products with 128-Kbyte Flash Memory (3/4)

Port		Bit name									128 pins	100 pins	80 pins	64 pins	52 pins	48 pins	44 pins	40 pins	36 pins	32 pins	30 pins
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PDIDISxx register	PMCAxx register	PMCTxx register	PMCExx register											
Port 10	0	PM100	P100	PU100	—	—	—	PMCA 100	—	—	✓	✓	✓	—	—	—	—	—	—	—	—
	1	PM101	P101	PU101	—	—	—	—	—	—	✓	✓	—	—	—	—	—	—	—	—	—
	2	PM102	P102	PU102	—	—	—	—	—	—	✓	✓	—	—	—	—	—	—	—	—	—
	3	PM103	P103	PU103	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
	4	PM104	P104	PU104	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
	5	PM105	P105	PU105	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
	6	PM106	P106	PU106	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Port 11	0	PM110	P110	PU110	—	—	—	—	—	—	✓	✓	✓	—	—	—	—	—	—	—	—
	1	PM111	P111	PU111	—	—	—	—	—	—	✓	✓	✓	—	—	—	—	—	—	—	—
	2	PM112	P112	PU112	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
	3	PM113	P113	PU113	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
	4	PM114	P114	PU114	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
	5	PM115	P115	PU115	—	—	—	PMCA 115	—	—	✓	—	—	—	—	—	—	—	—	—	—
	6	PM116	P116	PU116	—	—	—	PMCA 116	—	—	✓	—	—	—	—	—	—	—	—	—	—
	7	PM117	P117	PU117	—	—	—	PMCA 117	—	—	✓	—	—	—	—	—	—	—	—	—	—
Port 12	0	PM120	P120	PU120	—	POM120	PDIDIS 120	PMCA 120	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	1	PM121	P121	PU121	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	2	PM122	P122	PU122	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	3	—	P123	—	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—
	4	—	P124	—	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—
	5	PM125	P125	PU125	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
	6	PM126	P126	PU126	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
	7	PM127	P127	PU127	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
Port 13	0	—	P130	—	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓	—	—	—	—	—
	1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	P137	—	—	—	PDIDIS 137	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 4 - 3 PMxx, Pxx, PUxx, PIMxx, POMxx, PDIDISxx, PMCAxx, PMCTxx, and PMCExx Registers and the Bits Implemented in R7F100G Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin R7F100G Products with 128-Kbyte Flash Memory (4/4)

Port		Bit name									128 pins	100 pins	80 pins	64 pins	52 pins	48 pins	44 pins	40 pins	36 pins	32 pins	30 pins
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PDIDISxx register	PMCAxx register	PMCTxx register	PMCExx register											
Port 14	0	PM140	P140	PU140	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓	—	—	—	—	—
	1	PM141	P141	PU141	—	—	—	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—	—
	2	PM142	P142	PU142	PIM142	POM142	PDIDIS 142	—	—	—	✓	✓	✓	—	—	—	—	—	—	—	—
	3	PM143	P143	PU143	PIM143	POM143	PDIDIS 143	—	—	—	✓	✓	✓	—	—	—	—	—	—	—	—
	4	PM144	P144	PU144	—	POM144	PDIDIS 144	—	—	—	✓	✓	✓	—	—	—	—	—	—	—	—
	5	PM145	P145	PU145	—	—	—	—	—	—	✓	✓	—	—	—	—	—	—	—	—	—
	6	PM146	P146	PU146	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
	7	PM147	P147	PU147	—	—	—	PMCA 147	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Port 15	0	PM150	P150	—	—	—	—	PMCA 150	PMCT 150	—	✓	✓	✓	—	—	—	—	—	—	—	—
	1	PM151	P151	—	—	—	—	PMCA 151	PMCT 151	—	✓	✓	✓	—	—	—	—	—	—	—	—
	2	PM152	P152	—	—	—	—	PMCA 152	PMCT 152	—	✓	✓	✓	—	—	—	—	—	—	—	—
	3	PM153	P153	—	—	—	—	PMCA 153	PMCT 153	—	✓	✓	✓	—	—	—	—	—	—	—	—
	4	PM154	P154	—	—	—	—	PMCA 154	PMCT 154	—	✓	✓	—	—	—	—	—	—	—	—	—
	5	PM155	P155	—	—	—	—	PMCA 155	PMCT 155	—	✓	✓	—	—	—	—	—	—	—	—	—
	6	PM156	P156	—	—	—	—	PMCA 156	—	—	✓	✓	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Note This bit is only present in the R7F100G products with 30 and 32 pins.

Table 4 - 4 PMxx, Pxx, PUxx, PIMxx, POMxx, PDIDISxx, PMCAxx, PMCTxx, and PMCExx Registers and the Bits Implemented in R7F104G Products (1/4)

Port		Bit name									48 pins	44 pins	32 pins
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PDIDISxx register	PMCAxx register	PMCTxx register	PMCExx register			
Port 0	0	PM00	P00	PU00	—	POM00	PDIDIS00	—	PMCT00 **Note 1**	—	✓	✓	✓
	1	PM01	P01	PU01	PIM01	—	—	—	PMCT01 **Note 1**	PMCE01	✓	✓	✓ **Note 2**
	2	PM02	P02	PU02	—	POM02	PDIDIS02	PMCA02	PMCT02 **Note 1**	—	✓	✓ **Note 4**	✓
	3	PM03	P03	PU03	PIM03	POM03	PDIDIS03	PMCA03	PMCT03 **Note 1**	—	✓	✓ **Note 4**	✓
	4	PM04	P04	PU04	PIM04	POM04	PDIDIS04	—	—	—	✓ **Note 2**	✓ **Note 2**	✓ **Note 2**
	5	PM05	P05	PU05	—	—	—	—	PMCT05	—	✓	✓	✓
	6	PM06	P06	PU06	—	—	—	—	PMCT06	—	✓	✓	✓
	7	—	—	—	—	—	—	—	—	—	—	—	—
Port 1	0	PM10	P10	PU10	PIM10	POM10	PDIDIS10	—	—	PMCE10	✓	✓	✓
	1	PM11	P11	PU11	PIM11	POM11	PDIDIS11	—	—	PMCE11	✓	✓	✓
	2	PM12	P12	PU12	—	POM12	PDIDIS12	—	—	PMCE12	✓	✓	✓
	3	PM13	P13	PU13	PIM13	POM13	PDIDIS13	PMCA13	—	PMCE13	✓ **Note 4**	✓ **Note 4**	✓ **Note 2**
	4	PM14	P14	PU14	PIM14	POM14	PDIDIS14	—	—	PMCE14	✓	✓ **Note 2**	✓ **Note 2**
	5	PM15	P15	PU15	PIM15	POM15	PDIDIS15	—	—	PMCE15	✓ **Note 2**	✓ **Note 2**	✓ **Note 2**
	6	PM16	P16	PU16	PIM16	—	—	—	—	PMCE16	✓	✓	✓
	7	PM17	P17	PU17	PIM17	POM17	PDIDIS17	—	—	PMCE17	✓	✓	✓
Port 2	0	PM20	P20	—	—	—	—	PMCA20	—	—	✓	✓	✓
	1	PM21	P21	—	—	—	—	PMCA21	—	—	✓	✓	✓
	2	PM22	P22	—	—	—	—	PMCA22	PMCT22 **Note 1**	—	✓	✓	✓
	3	PM23	P23	—	—	—	—	PMCA23	PMCT23 **Note 1**	—	✓	✓	✓
	4	PM24	P24	—	—	—	—	PMCA24	PMCT24 **Note 1**	—	✓	✓	✓ **Note 2**
	5	PM25	P25	—	—	—	—	PMCA25	PMCT25 **Note 1**	—	✓	✓	✓ **Note 2**
	6	PM26	P26	—	—	—	—	PMCA26	PMCT26 **Note 1**	—	✓	✓	✓ **Note 2**
	7	PM27	P27	—	—	—	—	PMCA27	PMCT27 **Note 1**	—	✓	✓	✓ **Note 2**
Port 3	0	PM30	P30	PU30	—	—	—	—	PMCT30	—	✓	✓	✓
	1	PM31	P31	PU31	—	—	—	—	PMCT31	—	✓	✓	✓
	2	—	—	—	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—

Table 4 - 4 PMxx, Pxx, PUxx, PIMxx, POMxx, PDIDISxx, PMCAxx, PMCTxx, and PMCExx Registers and the Bits Implemented in R7F104G Products (2/4)

Port		Bit name									48 pins	44 pins	32 pins
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PDIDISxx register	PMCAxx register	PMCTxx register	PMCExx register			
Port 4	0	PM40	P40	PU40	—	—	—	—	—	—	✓	✓	✓
	1	PM41	P41	PU41	PIM41	—	—	—	—	—	✓	✓ Note 2	✓ Note 2
	2	PM42	P42	PU42	—	POM42	PDIDIS42	—	—	—	✓ Note 2	✓ Note 2	✓ Note 2
	3	PM43	P43	PU43	PIM43 Note 1	POM43 Note 1	PDIDIS43 Note 1	—	—	—	✓ Note 2	✓ Note 2	✓ Note 2
	4	PM44	P44	PU44	PIM44	POM44	PDIDIS44	—	—	—	✓ Note 2	✓ Note 2	—
	5	PM45	P45	PU45	—	POM45	PDIDIS45	—	—	—	✓ Note 2	✓ Note 2	—
	6	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—
Port 5	0	PM50	P50	PU50	—	POM50	PDIDIS50	—	PMCT50	PMCE50	✓	✓	✓
	1	PM51	P51	PU51	—	—	—	—	—	PMCE51	✓ Note 2	✓ Note 2	✓ Note 2
	2	PM52	P52	PU52	—	POM52 Note 1	PDIDIS52 Note 1	—	—	—	✓ Note 2	✓ Note 2	✓ Note 2
	3	PM53	P53	PU53	PIM53 Note 1	POM53 Note 1	PDIDIS53 Note 1	—	—	—	✓ Note 2	✓ Note 2	✓ Note 2
	4	PM54	P54	PU54	PIM54 Note 1	POM54 Note 1	PDIDIS54 Note 1	—	—	—	✓ Note 2	✓ Note 2	✓ Note 2
	5	PM55	P55	PU55	PIM55	POM55	PDIDIS55	—	—	—	✓ Note 2	✓ Note 2	✓ Note 2
	6	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—
Port 6	0	PM60	P60	—	—	—	—	—	—	PMCE60	✓ Note 2	✓ Note 2	✓ Note 2
	1	PM61	P61	—	—	—	—	—	—	PMCE61	✓ Note 2	✓ Note 2	✓ Note 2
	2	PM62	P62	—	—	—	—	—	—	—	✓ Note 2	✓ Note 2	✓ Note 2
	3	PM63	P63	—	—	—	—	—	—	—	✓ Note 2	✓ Note 2	✓ Note 2
	4	PM64	P64	PU64	—	—	—	—	PMCT64 Note 1	—	✓	✓	—
	5	PM65	P65	PU65	—	—	—	—	PMCT65 Note 1	—	✓	✓	—
	6	PM66	P66	PU66	—	—	—	—	PMCT66 Note 1	—	✓	✓ Note 2	—
	7	PM67	P67	PU67	—	—	—	—	PMCT67 Note 1	—	✓ Note 2	✓ Note 2	—
Port 7	0	PM70	P70	PU70	—	—	—	—	PMCT70	—	✓	✓	✓
	1	PM71	P71	PU71	PIM71	POM71	PDIDIS71	—	PMCT71	—	✓	✓	✓
	2	PM72	P72	PU72	—	POM72	PDIDIS72	—	PMCT72	—	✓	✓	✓
	3	PM73	P73	PU73	—	—	—	—	PMCT73	—	✓	✓	✓
	4	PM74	P74	PU74	—	POM74	PDIDIS74	—	PMCT74	—	✓ Note 2	✓ Note 2	✓ Note 2
	5	PM75	P75	PU75	—	—	—	—	PMCT75	—	✓ Note 2	✓ Note 2	✓ Note 2
	6	PM76	P76	PU76	—	—	—	—	PMCT76	—	✓	✓	✓
	7	PM77	P77	PU77	—	—	—	—	PMCT77	—	✓	✓	✓

Table 4 - 4 PMxx, Pxx, PUxx, PIMxx, POMxx, PDIDISxx, PMCAxx, PMCTxx, and PMCExx Registers and the Bits Implemented in R7F104G Products (3/4)

Port		Bit name									48 pins	44 pins	32 pins
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PDIDISxx register	PMCAxx register	PMCTxx register	PMCExx register			
Port 10	0	PM100	P100	PU100	—	—	—	PMCA100 Note 1	—	—	✓ Note 2	✓ Note 2	—
	1	—	—	—	—	—	—	—	—	—	—	—	—
	2	—	—	—	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—
Port 11	0	PM110	P110	PU110	—	—	—	—	—	—	✓ Note 2	✓ Note 2	—
	1	PM111	P111	PU111	—	—	—	—	—	—	✓ Note 2	✓ Note 2	—
	2	—	—	—	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—
Port 12	0	PM120	P120	PU120	—	POM120	PDIDIS120	PMCA120	—	—	✓ Note 2	✓ Note 2	✓ Note 2
	1	PM121	P121	PU121	—	—	—	—	—	—	✓	✓	✓
	2	PM122	P122	PU122	—	—	—	—	—	—	✓	✓	✓
	3	—	P123	—	—	—	—	—	—	—	✓	✓	✓ Note 3
	4	—	P124	—	—	—	—	—	—	—	✓	✓	✓ Note 3
	5	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—
Port 13	0	—	P130	—	—	—	—	—	—	—	✓	✓ Note 2	✓ Note 2
	1	—	—	—	—	—	—	—	—	—	—	—	—
	2	—	—	—	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	P137	—	—	—	PDIDIS137	—	—	—	✓	✓	✓
Port 14	0	PM140	P140	PU140	—	—	—	—	—	—	✓ Note 2	✓ Note 2	✓ Note 2
	1	PM141	P141	PU141	—	—	—	—	—	—	✓ Note 2	✓ Note 2	✓ Note 2
	2	PM142	P142	PU142	PIM142	POM142	PDIDIS142	—	—	—	✓ Note 2	✓ Note 2	—
	3	PM143	P143	PU143	PIM143	POM143	PDIDIS143	—	—	—	✓ Note 2	✓ Note 2	—
	4	PM144	P144	PU144	—	POM144	PDIDIS144	—	—	—	✓ Note 2	✓ Note 2	—
	5	—	—	—	—	—	—	—	—	—	—	—	—
	6	PM146	P146	PU146	—	—	—	—	—	—	✓ Note 2	✓ Note 2	✓ Note 2
	7	PM147	P147	PU147	—	—	—	PMCA147	—	—	✓	✓	✓

Table 4 - 4 PMxx, Pxx, PUxx, PIMxx, POMxx, PDIDISxx, PMCAxx, PMCTxx, and PMCExx Registers and the Bits Implemented in R7F104G Products (4/4)

Port		Bit name								48 pins	44 pins	32 pins
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PDIDISxx register	PMCAxx register	PMCTxx register	PMCExx register			
Port 15	0	PM150	P150	—	—	—	—	PMCA150 **Note 1**	PMCT150 **Note 1**	—	✓ **Note 4**	✓ **Note 4**	—
	1	PM151	P151	—	—	—	—	PMCA151 **Note 1**	PMCT151 **Note 1**	—	✓ **Note 2**	✓ **Note 2**	—
	2	PM152	P152	—	—	—	—	PMCA152 **Note 1**	PMCT152 **Note 1**	—	✓ **Note 2**	✓ **Note 2**	—
	3	PM153	P153	—	—	—	—	PMCA153 **Note 1**	PMCT153 **Note 1**	—	✓ **Note 4**	✓ **Note 4**	—
	4	—	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—	—

Note 1. This bit is only present in the R7F104G products with 44 and 48 pins.

Note 2. Be sure to leave the corresponding bits of the PMxx and PMCAxx registers set to 0. Also, be sure to leave the corresponding bits of the Pxx, PUxx, PIMxx, POMxx, PDIDISxx, PMCTxx, and PMCExx registers set to their initial values.

Note 3. Set the EXCLKS and OSCSELS bits in the clock operation mode control register (CMC) to 0 and 1, respectively. Set the XTSTOP bit in the clock operation status control register (CSC) to 1. For details on these registers, see **Section 6 Clock Generator**.

Note 4. Be sure to set the corresponding bits of the PMCAxx registers to 0.

4.3.1 Port mode registers (PMxx)

The PMxx registers specify input or output mode for the ports in 1-bit units. These registers can be set by a 1-bit or 8-bit memory manipulation instruction. The value of each PMxx register following a reset is FFH. To use a multiplexed function of a port pin, set the port mode register by referencing **4.5 Register Settings When Using Multiplexed Function**.

Figure 4 - 1 Format of Port Mode Registers (PMxx)

For the R7F100G products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FFF23H	FFH	R/W
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40	FFF24H	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FFF25H	FFH	R/W
PM6	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60	FFF26H	FFH	R/W
PM7	PM77	PM76	PM75	PM74	PM73	PM72	PM71	PM70	FFF27H	FFH	R/W
PM8	PM87	PM86	PM85	PM84	PM83	PM82	PM81	PM80	FFF28H	FFH	R/W
PM9	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90	FFF29H	FFH	R/W
PM10	1	PM106	PM105	PM104	PM103	PM102	PM101	PM100	FFF2AH	FFH	R/W
PM11	PM117	PM116	PM115	PM114	PM113	PM112	PM111	PM110	FFF2BH	FFH	R/W
PM12	PM127	PM126	PM125	1	1	PM122	PM121	PM120	FFF2CH	FFH	R/W
PM12	PM127	PM126	PM125	1	1	PM122	PM121	PM120	FFF2EH	FFH	R/W
PM15	1	PM156	PM155	PM154	PM153	PM152	PM151	PM150	FFF2FH	FFH	R/W

For the 32-pin R7F104G product

Symbol	7	6	5	4	3	2	1	0	Address	After reset
PM0	1	PM06	PM05	PM04 Note	PM03	PM02	PM01 Note	PM00	FFF20H	FFH
PM1	PM17	PM16	PM15 Note	PM14 Note	PM13 Note	PM12	PM11	PM10	FFF21H	FFH
PM2	PM27 Note	PM26 Note	PM25 Note	PM24 Note	PM23	PM22	PM21	PM20	FFF22H	FFH
PM3	1	1	1	1	1	1	PM31	PM30	FFF23H	FFH
PM4	1	1	1	1	PM43 Note	PM42 Note	PM41 Note	PM40	FFF24H	FFH
PM5	1	1	PM55 Note	PM54 Note	PM53 Note	PM52 Note	PM51 Note	PM50	FFF25H	FFH
PM6	1	1	1	1	PM63 Note	PM62 Note	PM61 Note	PM60 Note	FFF26H	FFH
PM7	PM77	PM76	PM75 Note	PM74 Note	PM73	PM72	PM71	PM70	FFF27H	FFH
PM12	1	1	1	1	1	PM122	PM121	PM120 Note	FFF2CH	FFH
PM14	PM147	PM146 Note	1	1	1	1	PM141 Note	PM140 Note	FFF2EH	FFH

For the 44-pin R7F104G product

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	PM06	PM05	PM04 Note	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W
PM1	PM17	PM16	PM15 Note	PM14 Note	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W
PM3	1	1	1	1	1	1	PM31	PM30	FFF23H	FFH	R/W
PM4	1	1	PM45 Note	PM44 Note	PM43 Note	PM42 Note	PM41 Note	PM40	FFF24H	FFH	R/W
PM5	1	1	PM55 Note	PM54 Note	PM53 Note	PM52 Note	PM51 Note	PM50	FFF25H	FFH	R/W
PM6	PM67 Note	PM66 Note	PM65	PM64	PM63 Note	PM62 Note	PM61 Note	PM60 Note	FFF26H	FFH	R/W
PM7	PM77	PM76	PM75 Note	PM74 Note	PM73	PM72	PM71	PM70	FFF27H	FFH	R/W
PM10	1	1	1	1	1	1	1	PM100 Note	FFF2AH	FFH	R/W
PM11	1	1	1	1	1	1	PM111 Note	PM110 Note	FFF2BH	FFH	R/W
PM12	1	1	1	1	1	PM122	PM121	PM120 Note	FFF2CH	FFH	R/W
PM14	PM147	PM146 Note	1	PM144 Note	PM143 Note	PM142 Note	PM141 Note	PM140 Note	FFF2EH	FFH	R/W
PM15	1	1	1	1	PM153	PM152 Note	PM151 Note	PM150	FFF2FH	FFH	R/W

For the 48-pin R7F104G product

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	PM06	PM05	PM04 **Note**	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W
PM1	PM17	PM16	PM15 **Note**	PM14	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W
PM3	1	1	1	1	1	1	PM31	PM30	FFF23H	FFH	R/W
PM4	1	1	PM45 **Note**	PM44 **Note**	PM43 **Note**	PM42 **Note**	PM41	PM40	FFF24H	FFH	R/W
PM5	1	1	PM55 **Note**	PM54 **Note**	PM53 **Note**	PM52 **Note**	PM51 **Note**	PM50	FFF25H	FFH	R/W
PM6	PM67 **Note**	PM66	PM65	PM64	PM63 **Note**	PM62 **Note**	PM61 **Note**	PM60 **Note**	FFF26H	FFH	R/W
PM7	PM77	PM76	PM75 **Note**	PM74 **Note**	PM73	PM72	PM71	PM70	FFF27H	FFH	R/W
PM10	1	1	1	1	1	1	1	PM100 **Note**	FFF2AH	FFH	R/W
PM11	1	1	1	1	1	1	PM111 **Note**	PM110 **Note**	FFF2BH	FFH	R/W
PM12	1	1	1	1	1	PM122	PM121	PM120 **Note**	FFF2CH	FFH	R/W
PM14	PM147	PM146 **Note**	1	PM144 **Note**	PM143 **Note**	PM142 **Note**	PM141 **Note**	PM140 **Note**	FFF2EH	FFH	R/W
PM15	1	1	1	1	PM153	PM152 **Note**	PM151 **Note**	PM150	FFF2FH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 12, 14, 15; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Note Be sure to leave this bit set to 0.

Caution Be sure to set bits that are not implemented to their initial values.

4.3.2 Port registers (Pxx)

The Pxx registers set the output latch values of ports. If data is read in input mode, the pin level is read. If data is read in output mode, the output latch value of the port is read**Note**. The Pxx registers can be set by a 1-bit or 8-bit memory manipulation instruction. The value of each Pxx register following a reset is 00H.

Note If P00 to P03, P13, P20 to P27, P35 to P37, P100, P115 to P117, P120, P147, or P150 to P156 is set up as an analog function port, when the port is read in the input mode, 0 is always returned instead of the pin level.
If P00 to P03, P05, P06, P22 to P27, P30, P31, P50, P64 to P67, P70 to P77, or P150 to P155 is set up as a capacitance measurement port, when the port is read in the input mode, 0 is always returned instead of the pin level.

Figure 4 - 2 Format of Port Registers (Pxx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	P07	P06	P05	P04	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
P2	P27	P26	P25	P24	P23	P22	P21	P20	FFF02H	00H (output latch)	R/W
P3	P37	P36	P35	P34	P33	P32	P31	P30	FFF03H	00H (output latch)	R/W
P4	P47	P46	P45	P44	P43	P42	P41	P40	FFF04H	00H (output latch)	R/W
P5	P57	P56	P55	P54	P53	P52	P51	P50	FFF05H	00H (output latch)	R/W
P6	P67	P66	P65	P64	P63	P62	P61	P60	FFF06H	00H (output latch)	R/W
P7	P77	P76	P75	P74	P73	P72	P71	P70	FFF07H	00H (output latch)	R/W
P8	P87	P86	P85	P84	P83	P82	P81	P80	FFF08H	00H (output latch)	R/W
P9	P97	P96	P95	P94	P93	P92	P91	P90	FFF09H	00H (output latch)	R/W
P10	0	P106	P105	P104	P103	P102	P101	P100	FFF0AH	00H (output latch)	R/W
P11	P117	P116	P115	P114	P113	P112	P111	P110	FFF0BH	00H (output latch)	R/W
P12	P127	P126	P125	P124	P123	P122	P121	P120	FFF0CH	Undefined	R/W**Note 1**
P13	P137	0	0	0	0	0	0	P130	FFF0DH	**Note 2**	R/W**Note 1**
P14	P147	P146	P145	P144	P143	P142	P141	P140	FFF0EH	00H (output latch)	R/W
P15	0	P156	P155	P154	P153	P152	P151	P150	FFF0FH	00H (output latch)	R/W

Pmn	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

Note 1. P123, P124, and P137 are only readable.
Note 2. P137: Undefined
P130: 0 (output latch)

Caution **Be sure to set bits that are not implemented to their initial values.**

Remark m = 0 to 15; n = 0 to 7

4.3.3 Pull-up resistor option registers (PUxx)

The PUxx registers specify whether to use the on-chip pull-up resistors. On-chip pull-up resistors can be used in 1-bit units only for the bits set to both normal output mode (POMmn = 0) and input mode (PMmn = 1) 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 are not connected to bits set to output mode and bits used as multiplexed-function output pins, regardless of the settings of these registers. The PUxx registers can be set by a 1-bit or 8-bit memory manipulation instruction. The value of the PU4 register following a reset is 01H, and that of the other PUxx is 00H.

Caution When data is input from a device operating at a different voltage to the TTL buffer for a port with the PIMxx register, set PUmn = 0 and pull up to the power supply of the device operating at a different voltage via an external resistor.

Figure 4 - 3 Format of Pull-up Resistor Option Registers (PUxx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	PU07	PU06	PU05	PU04	PU03	PU02	PU01	PU00	F0030H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	F0031H	00H	R/W
PU3	PU37	PU36	PU35	PU34	PU33	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	0	0	0	0	F0036H	00H	R/W
PU7	PU77	PU76	PU75	PU74	PU73	PU72	PU71	PU70	F0037H	00H	R/W
PU8	PU87	PU86	PU85	PU84	PU83	PU82	PU81	PU80	F0038H	00H	R/W
PU9	PU97	PU96	PU95	PU94	PU93	PU92	PU91	PU90	F0039H	00H	R/W
PU10	0	PU106	PU105	PU104	PU103	PU102	PU101	PU100	F003AH	00H	R/W
PU11	PU117	PU116	PU115	PU114	PU113	PU112	PU111	PU110	F003BH	00H	R/W
PU12	PU127	PU126	PU125	0	0	PU122	PU121	PU120	F003CH	00H	R/W
PU14	PU147	PU146	PU145	PU144	PU143	PU142	PU141	PU140	F003EH	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0, 1, 3 to 12, 14; 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 that are not implemented to their initial values.

4.3.4 Port input mode registers (PIMxx)

The PIMxx registers set the input buffer in 1-bit units. TTL input buffer can be selected during serial communication with an external device operating at a different voltage. The PIMxx registers can be set by a 1-bit or 8-bit memory manipulation instruction. The value of each PIMxx register following a reset is 00H.

Figure 4 - 4 Format of Port Input Mode Registers (PIMxx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM0	0	0	0	PIM04	PIM03	0	PIM01	0	F0040H	00H	R/W
PIM1	PIM17	PIM16	PIM15	PIM14	PIM13	0	PIM11	PIM10	F0041H	00H	R/W
PIM3	0	0	0	0	PIM33	0	0	0	F0043H	00H	R/W
PIM4	0	0	0	PIM44	PIM43	0	PIM41	0	F0044H	00H	R/W
PIM5	0	0	PIM55	PIM54	PIM53	0	0	0	F0045H	00H	R/W
PIM7	0	0	0	0	0	0	PIM71	0	F0047H	00H	R/W
PIM8	0	0	0	PIM84	0	0	PIM81	PIM80	F0048H	00H	R/W
PIM14	0	0	0	0	PIM143	PIM142	0	0	F004EH	00H	R/W

PIMmn	Pmn pin input buffer selection (m = 0, 1, 3, 4, 5, 7, 8, 14; n = 0 to 7)
0	Normal input buffer
1	TTL input buffer

Caution Be sure to set bits that are not implemented to their initial values.

4.3.5 Port output mode registers (POMxx)

The POMxx registers set the output mode in 1-bit units. N-ch open-drain output (withstand voltage of V_{DD}[Note 1]/ withstand voltage of EV_{DD}[Note 2]) mode can be selected during serial communication with an external device operating at a different voltage, and for the SDA00, SDA01, SDA10, SDA11, SDA20, SDA21, SDA30, and SDA31 pins during simplified I^2C communication with an external device operating at the same voltage. In addition, POMxx registers are used in combination with PUxx registers to specify whether to use on-chip pull-up resistors. The POMxx registers can be set by a 1-bit or 8-bit memory manipulation instruction. The value of each POMxx register following a reset is 00H.

Caution An on-chip pull-up resistor is not connected to a bit for which N-ch open-drain output (withstand voltage of V_{DD}[Note 1]/withstand voltage of EV_{DD}[Note 2]) mode (POMmn = 1) is set.

Note 1. This applies to the 30- to 52-pin R7F100G and 32- to 48-pin R7F104G products.
Note 2. This applies to the R7F100G products with 64 to 128 pins.

Figure 4 - 5 Format of Port Output Mode Registers (POMxx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM0	0	0	0	POM04	POM03	POM02	0	POM00	F0050H	00H	R/W
POM1	POM17	0	POM15	POM14	POM13	POM12	POM11	POM10	F0051H	00H	R/W
POM3	0	0	0	POM34	0	0	0	0	F0053H	00H	R/W
POM4	0	0	POM45	POM44	POM43	POM42	0	0	F0054H	00H	R/W
POM5	0	0	POM55	POM54	POM53	POM52	0	POM50	F0055H	00H	R/W
POM7	0	0	0	POM74	0	POM72	POM71	0	F0057H	00H	R/W
POM8	0	0	0	0	POM83	POM82	POM81	POM80	F0058H	00H	R/W
POM9	0	POM96	0	0	0	0	0	0	F0059H	00H	R/W
POM12	0	0	0	0	0	0	0	POM 120	F005CH	00H	R/W
POM14	0	0	0	POM 144	POM 143	POM 142	0	0	F005EH	00H	R/W

POMmn	Pmn pin output mode selection (m = 0, 1, 3 to 5, 7 to 9, 12, 14; n = 0 to 7)
0	Normal output mode
1	N-ch open-drain output (withstand voltage of V_{DD}[Note 1]/withstand voltage of EV_{DD}[Note 2]) mode

Note 1. This applies to the 30- to 52-pin R7F100G and 32- to 48-pin R7F104G products.
Note 2. This applies to the R7F100G products with 64 to 128 pins.

Caution Be sure to set bits that are not implemented to their initial values.

4.3.6 Port digital input disable registers (PDIDISxx)

The PDIDISxx registers are used to prevent flows of through-current into input buffers. When N-ch open-drain output is selected for serial communications with an external device operating at a different voltage or an input port is not to be used, power consumption can be lowered by setting the corresponding bit in the given PDIDISxx register to 1. The PDIDISxx registers can be set by a 1-bit or 8-bit memory manipulation instruction. The value of each PDIDISxx register following a reset is 00H.

Figure 4 - 6 Format of Port Digital Input Disable Registers (PDIDISxx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PDIDIS0	0	0	0	PDIDIS04	PDIDIS03	PDIDIS02	0	PDIDIS00	F02B0H	00H	R/W
PDIDIS1	PDIDIS17	0	PDIDIS15	PDIDIS14	PDIDIS13	PDIDIS12	PDIDIS11	PDIDIS10	F02B1H	00H	R/W
PDIDIS3	0	0	0	PDIDIS34	0	0	0	0	F02B3H	00H	R/W
PDIDIS4	0	0	PDIDIS45	PDIDIS44	PDIDIS43	PDIDIS42	0	0	F02B4H	00H	R/W
PDIDIS5	0	0	PDIDIS55	PDIDIS54	PDIDIS53	PDIDIS52	0	PDIDIS50	F02B5H	00H	R/W
PDIDIS7	0	0	0	PDIDIS74	0	PDIDIS72	PDIDIS71	0	F02B7H	00H	R/W
PDIDIS8	0	0	0	0	PDIDIS83	PDIDIS82	PDIDIS81	PDIDIS80	F02B8H	00H	R/W
PDIDIS9	0	PDIDIS96	0	0	0	0	0	0	F02B9H	00H	R/W
PDIDIS12	0	0	0	0	0	0	0	PDIDIS120	F02BCH	00H	R/W
PDIDIS13	PDIDIS137	0	0	0	0	0	0	0	F02BDH	00H	R/W
PDIDIS14	0	0	0	PDIDIS144	PDIDIS143	PDIDIS142	0	0	F02BEH	00H	R/W

PDIDISmn	Setting of input buffers (m = 0, 1, 3 to 5, 7 to 9, 12 to 14; n = 0 to 7)
0	Input to the input buffers is enabled.
1	Input to the input buffers is disabled. This prevents flows of through-current into the input buffers.

Caution **Be sure to set bits that are not implemented to their initial values.**

Remark For P123 and P124, low power consumption can be achieved by setting the EXCLKS bit to 0 and the OSCSELS bit to 1 in the clock operation mode control register (CMC) and setting the XTSTOP bit to 1 in the clock operation status control register (CSC).

4.3.7 Port mode control A registers (PMCAxx)

The PMCAxx registers specify the digital I/O or analog function in 1-bit units. These registers can be set by a 1-bit or 8-bit memory manipulation instruction. The value of each PMCAxx register following a reset is FFH.

Figure 4 - 7 Format of Port Mode Control A Registers (PMCAxx)

For the R7F100G products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMCA0	1	1	1	1	PMCA03	PMCA02	PMCA01	PMCA00	F0060H	FFH	R/W
PMCA1	1	1	1	1	PMCA13	1	1	1	F0061H	FFH	R/W
PMCA2	PMCA27	PMCA26	PMCA25	PMCA24	PMCA23	PMCA22	PMCA21	PMCA20	F0062H	FFH	R/W
PMCA3	PMCA37	PMCA36	PMCA35	1	1	1	1	1	F0063H	FFH	R/W
PMCA10	1	1	1	1	1	1	1	PMCA100	F006AH	FFH	R/W
PMCA11	PMCA117	PMCA116	PMCA115	1	1	1	1	1	F006BH	FFH	R/W
PMCA12	1	1	1	1	1	1	1	PMCA120	F006CH	FFH	R/W
PMCA14	PMCA147	1	1	1	1	1	1	1	F006EH	FFH	R/W
PMCA15	1	PMCA156	PMCA155	PMCA154	PMCA153	PMCA152	PMCA151	PMCA150	F006FH	FFH	R/W

For the 32-pin R7F104G product

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMCA0	1	1	1	1	PMCA03	PMCA02	1	1	F0060H	FFH	R/W
PMCA1	1	1	1	1	PMCA13 Note	1	1	1	F0061H	FFH	R/W
PMCA2	PMCA27 Note	PMCA26 Note	PMCA25 Note	PMCA24 Note	PMCA23	PMCA22	PMCA21	PMCA20	F0062H	FFH	R/W
PMCA12	1	1	1	1	1	1	1	PMCA120 Note	F006CH	FFH	R/W
PMCA14	PMCA147	1	1	1	1	1	1	1	F006EH	FFH	R/W

For the 44-pin R7F104G product

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMCA0	1	1	1	1	PMCA03 Note	PMCA02 Note	1	1	F0060H	FFH	R/W
PMCA1	1	1	1	1	PMCA13 Note	1	1	1	F0061H	FFH	R/W
PMCA2	PMCA27	PMCA26	PMCA25	PMCA24	PMCA23	PMCA22	PMCA21	PMCA20	F0062H	FFH	R/W
PMCA10	1	1	1	1	1	1	1	PMCA100 Note	F006AH	FFH	R/W
PMCA12	1	1	1	1	1	1	1	PMCA120 Note	F006CH	FFH	R/W
PMCA14	PMCA147	1	1	1	1	1	1	1	F006EH	FFH	R/W
PMCA15	1	1	1	1	PMCA153 Note	PMCA152 Note	PMCA151 Note	PMCA150 Note	F006FH	FFH	R/W

For the 48-pin R7F104G product

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMCA0	1	1	1	1	PMCA03	PMCA02	1	1	F0060H	FFH	R/W
PMCA1	1	1	1	1	PMCA13Note	1	1	1	F0061H	FFH	R/W
PMCA2	PMCA27	PMCA26	PMCA25	PMCA24	PMCA23	PMCA22	PMCA21	PMCA20	F0062H	FFH	R/W
PMCA10	1	1	1	1	1	1	1	PMCA100Note	F006AH	FFH	R/W
PMCA12	1	1	1	1	1	1	1	PMCA120Note	F006CH	FFH	R/W
PMCA14	PMCA147	1	1	1	1	1	1	1	F006EH	FFH	R/W
PMCA15	1	1	1	1	PMCA153Note	PMCA152Note	PMCA151Note	PMCA150Note	F006FH	FFH	R/W

PMCAmn	Selection of digital I/O or analog function for Pmn pin (m = 0 to 3, 10 to 12, 14, 15; n = 0 to 7)
0	Digital I/O
1	Analog function

Note Be sure to leave this bit set to 0.

Caution 1. Be sure to leave bits that are not implemented set to their initial values.

Caution 2. Select the input mode by using the port mode register (PM0 to PM3, PM10 to PM12, PM14, or PM15) for a port pin which is set as an analog function by the corresponding PMCAxx register.

Caution 3. Do not set the pin that is specified as digital I/O by the PMCAxx register to the analog function by the analog input channel specification register (ADS).

Caution 4. The value of the PMCA3, PMCA10, PMCA11, and PMCA15 registers are read as 00H in the 30- to 64-pin R7F100G products with 96- or 128-Kbyte flash memory and 32-pin R7F104G product.

4.3.8 Port mode control T registers (PMCTxx)

The PMCTxx registers specify the 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. The value of each PMCTxx register following a reset is 00H.

Figure 4 - 8 Format of Port Mode Control T Registers (PMCTxx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMCT0	0	PMCT06	PMCT05	0	PMCT03	PMCT02	PMCT01	PMCT00	F0260H	00H	R/W
PMCT2	PMCT27	PMCT26	PMCT25	PMCT24	PMCT23	PMCT22	0	0	F0262H	00H	R/W
PMCT3	0	0	0	0	0	0	PMCT31	PMCT30	F0263H	00H	R/W
PMCT5	0	0	0	0	0	0	0	PMCT50	F0265H	00H	R/W
PMCT6	PMCT67	PMCT66	PMCT65	PMCT64	0	0	0	0	F0266H	00H	R/W
PMCT7	PMCT77	PMCT76	PMCT75	PMCT74	PMCT73	PMCT72	PMCT71	PMCT70	F0267H	00H	R/W
PMCT15	0	0	PMCT155	PMCT154	PMCT153	PMCT152	PMCT151	PMCT150	F026FH	00H	R/W

PMCTmn	Selection of digital I/O or capacitance measurement function for Pmn pin (m = 0, 2, 3, 5 to 7, 15; n = 0 to 7)
0	Digital I/O
1	Capacitance measurement function

Caution Be sure to set bits that are not implemented to their initial values.

4.3.9 Port mode control E registers (PMCExx)

The PMCExx registers specify the digital I/O or ELCL output function in 1-bit units. These registers can be set by a 1-bit or 8-bit memory manipulation instruction. The value of each PMCExx register following a reset is 00H.

Figure 4 - 9 Format of Port Mode Control E Registers (PMCExx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMCE0	0	0	0	0	0	0	PMCE01	0	F0280H	00H	R/W
PMCE1	PMCE17	PMCE16	PMCE15	PMCE14	PMCE13	PMCE12	PMCE11	PMCE10	F0281H	00H	R/W
PMCE5	0	0	0	0	0	0	PMCE51	PMCE50	F0285H	00H	R/W
PMCE6	0	0	0	0	0	0	PMCE61	PMCE60	F0286H	00H	R/W

PMCEmn	Selection of digital I/O or ELCL output function for Pmn pin (m = 0, 1, 5, 6; n = 0 to 7)
0	Digital I/O
1	ELCL output function

Caution Be sure to set bits that are not implemented to their initial values.

4.3.10 Peripheral I/O redirection register (PIOR)

The PIOR register is used to specify whether to enable or disable the peripheral I/O redirection function. This function is used to switch multiplexed functions that are multiplexed with port pins to replace the port functions of the pins. Use the PIOR register to assign the target multiplexed function for redirection to a selected port pin before enabling the function. Note that the redirection setting cannot be changed once the operation of a function has been enabled. The PIOR register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 4 - 10 Format of Peripheral I/O Redirection Register (PIOR) (1/4)

Address: F0077H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
PIOR	0	0	PIOR5	PIOR4	PIOR3	PIOR2	PIOR1	PIOR0

For the R7F100G products

Bit	Alternative function	128- and 100-pin		80-pin		64-pin		52-pin		48-pin		44-pin		40-, 36-, 32-, and 30-pin	
		Setting value		Setting value		Setting value		Setting value		Setting value		Setting value		Setting value	
		0	1	0	1	0	1	0	1	0	1	0	1	0	1
PIOR5	INTP1	P46	P56	These functions are not available for use. Set this bit to 0 (default value).											
	INTP3	P30	P57												
	INTP4	P31	P146												
	INTP6	P140	P84												
	INTP7	P141	P85												
	INTP8	P74	P86												
	INTP9	P75	P87												
	TxD1	P02	P82												
	RxD1	P03	P81												
	SCL10	P04	P80												
	SDA10	P03	P81												
	SI10	P03	P81												
	SO10	P02	P82												
	SCK10	P04	P80												
PIOR4	PCLBUZ1	P141	P55	P141	P55	P141	P55								
	INTP5	P16	P12	P16	P12	P16	P12								
PIOR3	PCLBUZ0	P140	P31	P140	P31	P140	P31	P140	P31	P140	P31				
PIOR2	SCLA0	P60	P14	P60	P14	P60	P14	P60	P14	P60	P14	P60	P14	P60	P14
	SDAA0	P61	P13	P61	P13	P61	P13	P61	P13	P61	P13	P61	P13	P61	P13

Figure 4 - 10 Format of Peripheral I/O Redirection Register (PIOR) (2/4)

For the R7F100G products

Bit	Alternative function	128- and 100-pin		80-pin		64-pin		52-pin		48-pin		44-pin		40-, 36-, 32-, and 30-pin	
		Setting value		Setting value		Setting value		Setting value		Setting value		Setting value		Setting value	
		0	1	0	1	0	1	0	1	0	1	0	1	0	1
PIOR1	INTP10	P76	P110	P76	P110	P76	P52	P76	—	—	—	—	—	—	—
	INTP11	P77	P111	P77	P111	P77	P53	P77	—	—	—	—	—	—	—
	TxD2	P13	P77	P13	P77	P13	P77	P13	P77	P13	—	P13	—	P13	—
	RxD2	P14	P76	P14	P76	P14	P76	P14	P76	P14	—	P14	—	P14	—
	SCL20	P15	—	P15	—	P15	—	P15	—	P15	—	P15	—	P15	—
	SDA20	P14	—	P14	—	P14	—	P14	—	P14	—	P14	—	P14	—
	SI20	P14	—	P14	—	P14	—	P14	—	P14	—	P14	—	P14	—
	SO20	P13	—	P13	—	P13	—	P13	—	P13	—	P13	—	P13	—
	SCK20	P15	—	P15	—	P15	—	P15	—	P15	—	P15	—	P15	—
	TxD0	P12	P17	P12	P17	P12	P17	P12	P17	P12	P17	P12	P17	P12	P17
	RxD0	P11	P16	P11	P16	P11	P16	P11	P16	P11	P16	P11	P16	P11	P16
	SCL00	P10	—	P10	—	P10	—	P10	—	P10	—	P10	—	P10	—
	SDA00	P11	—	P11	—	P11	—	P11	—	P11	—	P11	—	P11	—
	SI00	P11	P16	P11	P16	P11	P16	P11	—	P11	—	P11	—	P11	—
	SO00	P12	P17	P12	P17	P12	P17	P12	—	P12	—	P12	—	P12	—
	SCK00	P10	P55	P10	P55	P10	P55	P10	—	P10	—	P10	—	P10	—
PIOR0	TI02/TO02	P17	P15	P17	P15	P17	P15	P17	P15	P17	P15	P17	P15	P17	P15
	TI03/TO03	P31	P14	P31	P14	P31	P14	P31	P14	P31	P14	P31	P14	P31	P14
	TI04/TO04	P42	P13	P42	P13	P42	P13	—	P13	—	P13	—	P13	—	P13
	TI05/TO05	P46	P12	P05	P12	P05	P12	—	P12	—	P12	—	P12	—	P12
	TI06/TO06	P102	P11	P06	P11	P06	P11	—	P11	—	P11	—	P11	—	P11
	TI07/TO07	P145	P10	P41	P10	P41	P10	P41	P10	P41	P10	P41	P10	—	P10

Figure 4 - 10 Format of Peripheral I/O Redirection Register (PIOR) (3/4)

For the R7F104G products

Bit	Alternative function	48-pin		44-pin		32-pin	
		Setting value		Setting value		Setting value	
		0	1	0	1	0	1
PIOR5	INTP1	These functions are not available for use. Set this bit to 0 (default value).					
	INTP3						
	INTP4						
	INTP6						
	INTP7						
	INTP8						
	INTP9						
	TxD1						
	RxD1						
	SCL10						
	SDA10						
	SI10						
	SO10						
	SCK10						
PIOR4	PCLBUZ1	—	—	—	—	—	—
	INTP5	P16	P12	P16	P12	P16	P12
PIOR3	PCLBUZ0	—	P31	—	P31	—	P31
PIOR2	SCLA0	—	P14	—	—	—	—
	SDAA0	—	P13	—	—	—	—
PIOR1	INTP10	P76	—	P76	—	P76	—
	INTP11	P77	—	P77	—	P77	—
	TxD2	P13	P77	—	P77	—	P77
	RxD2	P14	P76	—	P76	—	P76
	SCL20	—	—	—	—	—	—
	SDA20	—	—	—	—	—	—
	SI20	—	—	—	—	—	—
	SO20	—	—	—	—	—	—
	SCK20	—	—	—	—	—	—
	TxD0	P12	P17	P12	P17	P12	P17
	RxD0	P11	P16	P11	P16	P11	P16
	SCL00	P10	—	P10	—	P10	—
	SDA00	P11	—	P11	—	P11	—
	SI00	P11	—	P11	—	P11	—
	SO00	P12	—	P12	—	P12	—
	SCK00	P10	—	P10	—	P10	—

Figure 4 - 10 Format of Peripheral I/O Redirection Register (PIOR) (4/4)

For the R7F104G products

Bit	Alternative function	48-pin		44-pin		32-pin	
		Setting value		Setting value		Setting value	
		0	1	0	1	0	1
PIOR0	TI02/TO02	P17	—	P17	—	P17	—
	TI03/TO03	P31	P14	P31	—	P31	—
	TI04/TO04	—	P13	—	P13	—	—
	TI05/TO05	P05	P12	P05	P12	P05	P12
	TI06/TO06	P06	P11	P06	P11	P06	P11
	TI07/TO07	P41	P10	—	P10	—	P10

Remark —: These functions are not available for use.

4.3.11 Global digital input disable register (GDIDIS)

The GDIDIS register is used to prevent flows of through-current into the input buffers of input ports which use EVDD as the power supply when the EVDD power supply is turned off. When none of the I/O ports using EVDD as the power supply are in use, power consumption can be lowered by setting the GDIDIS0 bit to 1 and then turning the EVDD power supply off. Setting the GDIDIS0 bit to 1 prohibits input to all input buffers using EVDD as the power supply, preventing flows of through-current while the EVDD power supply is turned off. The GDIDIS register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Remark The GDIDIS register is equipped with the R7F100G products with 64 to 128 pins.

Figure 4 - 11 Format of Global Digital Input Disable Register (GDIDIS)

Address: F007DH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
GDIDIS	0	0	0	0	0	0	0	GDIDIS0

GDIDIS0	Setting of input buffers using EVDD power supply
0	Input to the input buffers is enabled (default).
1	Input to the input buffers is disabled. This prevents flows of through-current into the input buffers.

Remark 1. The GDIDIS register is equipped with the R7F100G products with 64 to 128 pins.
Remark 2. Even when input to input buffers is prohibited (GDIDIS0 = 1), peripheral functions which do not use port functions having EVDD as the power supply can be used.

Turn off the EVDD power supply with the following procedure.

<1> Disable input to input buffers (set GDIDIS0 = 1).
<2> Turn off the EVDD power supply.

Turn on again the EVDD power supply with the following procedure.

<1> Turn on the EVDD power supply.
<2> Enable input to input buffers (set GDIDIS0 = 0).

Caution 1. Do not input an input voltage equal to or greater than EVDD to an input port that uses EVDD as the power supply.

Caution 2. When input to input buffers is prohibited (GDIDIS0 = 1), the value read from the port register (Pxx) of a port that uses EVDD as the power supply is 1. When 1 is set in a port output mode register (POMxx) (N-ch open-drain output (withstand voltage of EVDD) mode), the value read from the corresponding port register (Pxx) is 0.

4.3.12 Output current control enable register (CCDE)

The CCDE register is used to specify the use of P16, P17, P50, P51, and P60 to P63 as controlled current drive port pins in 1-bit units. The setting of the corresponding output current select register (CCSx) controls a pin for which this function is selected to be at the low level and produce the selected output current (low-level output current) or to be in the high-impedance state. The CCDE register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 4 - 12 Format of Output Current Control Enable Register (CCDE) (1/2)

Address: F02A8H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
CCDE	CCDE07	CCDE06	CCDE05	CCDE04	CCDE03	CCDE02	CCDE01	CCDE00

CCDE07	Selection of digital I/O or output current control function for CCD07 (P63) pin
0	Digital I/O (multiplexed function other than current control function)
1	Current control function

CCDE06	Selection of digital I/O or output current control function for CCD06 (P62) pin
0	Digital I/O (multiplexed function other than current control function)
1	Current control function

CCDE05	Selection of digital I/O or output current control function for CCD05 (P61) pin
0	Digital I/O (multiplexed function other than current control function)
1	Current control function

CCDE04	Selection of digital I/O or output current control function for CCD04 (P60) pin
0	Digital I/O (multiplexed function other than current control function)
1	Current control function

CCDE03	Selection of digital I/O or output current control function for CCD03 (P50**Note**) pin
0	Digital I/O (multiplexed function other than current control function)
1	Current control function

CCDE02	Selection of digital I/O or output current control function for CCD02 (P51) pin
0	Digital I/O (multiplexed function other than current control function)
1	Current control function

CCDE01	Selection of digital I/O or output current control function for CCD01 (P17) pin
0	Digital I/O (multiplexed function other than current control function)
1	Current control function

Figure 4 - 12 Format of Output Current Control Enable Register (CCDE) (2/2)

CCDE00	Selection of digital I/O or output current control function for CCD00 (P16) pin
0	Digital I/O (multiplexed function other than current control function)
1	Current control function

Note This applies to products other than the R7F100G with 44 pins.

Caution 1. When a port pin is to be used with output current control, make the setting for the output current control function and then set the corresponding bit in the PMxx register for output mode.

Caution 2. The state of a pin takes 10 µs to become stable after 1 having been written to the corresponding bit of the CCDE register.

4.3.13 Output current select registers (CCSx)

The CCSx registers are used to set the current control for the port pins selected for output current control in the output current control enable register (CCDE) or to place them in the high-impedance state. For current control, the pins are at the low level and produce output currents controlled to be 2 mA, 5 mA, 10 mA, or 15 mA. The CCSx registers can be set by an 8-bit memory manipulation instruction. The value of each CCSx register following a reset is 00H.

Figure 4 - 13 Format of Output Current Select Registers (CCSx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
CCS0	0	0	0	0	0	CCS02	CCS01	CCS00	F02A0H	00H	R/W
CCS4	0	0	0	0	0	CCS42	CCS41	CCS40	F02A4H	00H	R/W
CCS5	0	0	0	0	0	CCS52	CCS51	CCS50	F02A5H	00H	R/W
CCS6	0	0	0	0	0	CCS62	CCS61	CCS60	F02A6H	00H	R/W
CCS7	0	0	0	0	0	CCS72	CCS71	CCS70	F02A7H	00H	R/W

CCSx2	CCSx1	CCSx0	Setting for the Low-level Output Current				
			x = 0 CCD00 to CCD03	x = 4 CCD04	x = 5 CCD05	x = 6 CCD06	x = 7 CCD07
0	0	0	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
0	0	1	2 mA	2 mA	2 mA	2 mA	2 mA
0	1	0	5 mA	5 mA	5 mA	5 mA	5 mA
0	1	1	10 mA	10 mA	10 mA	10 mA	10 mA
1	0	0	Setting prohibited	15 mA	15 mA	15 mA	15 mA
Others			Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited

Remark x = 0, 4 to 7

4.3.14 40-mA port output control register (PTDC)

The PTDC register is used to set the allowable output current to 40 mA when P70, P51, P17, P10, P101, P04, or P120 is an output at the low level. The pins among those stated above which can be set for 40-mA output depend on the product. **Table 4 - 5** is a list by product of port pins which can be set for 40-mA output.

Table 4 - 5 Products Having Port Pins that Support 40-mA Output (R7F100G Only)

Number of Pins	Product Names	Port Pins Supporting 40-mA Output
100	R7F100GPN, R7F100GPL, R7F100GPK (products with 768, 512, or 384 Kbytes of flash memory)	P04, P10, P101, P120
80	R7F100GMN, R7F100GML, R7F100GMK (products with 768, 512, or 384 Kbytes of flash memory)	P04, P10, P120
64	R7F100GLN, R7F100GLL, R7F100GLK (products with 768, 512, or 384 Kbytes of flash memory)	P04, P10, P120
52	R7F100GJN, R7F100GJL, R7F100GJK, R7F100GJJ, R7F100GJH, R7F100GJG, R7F100GJF	P17, P51, P70
48	R7F100GGN, R7F100GGL, R7F100GJK, R7F100GGJ, R7F100GGH, R7F100GGG, R7F100GGF	P17, P51, P70
44	R7F100GFN, R7F100GFL, R7F100GFK, R7F100GFJ, R7F100GFH, R7F100GFG, R7F100GFF	P17, P51, P70
40	R7F100GEJ, R7F100GEH, R7F100GEG, R7F100GEF	P17, P51, P70
36	R7F100GCJ, R7F100GCH, R7F100GCG, R7F100GCF	P17, P51, P70
32	R7F100GBJ, R7F100GBH, R7F100GBG, R7F100GBF	P17, P51, P70
30	R7F100GAJ, R7F100GAH, R7F100GAG, R7F100GAF	P17, P51

The PTDC register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 4 - 14 Format of 40-mA Port Output Control Register (PTDC)

Address: F02A9H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
PTDC	0	PTDC6	PTDC5	PTDC4	PTDC3	PTDC2	PTDC1	PTDC0

PTDC6	Allowable output current when a low level is output from P120 pin**Note 1**
0	IOL1 = 20.0 mA
1	IOL1 = 40.0 mA

PTDC5	Allowable output current when a low level is output from P04 pin**Note 1**
0	IOL1 = 20.0 mA
1	IOL1 = 40.0 mA

PTDC4	Allowable output current when a low level is output from P101 pin**Note 2**
0	IOL1 = 20.0 mA
1	IOL1 = 40.0 mA

PTDC3	Allowable output current when a low level is output from P10 pin**Note 1**
0	IOL1 = 20.0 mA
1	IOL1 = 40.0 mA

PTDC2	Allowable output current when a low level is output from P17 pin**Note 3**
0	IOL1 = 20.0 mA
1	IOL1 = 40.0 mA

PTDC1	Allowable output current when a low level is output from P51 pin**Note 3**
0	IOL1 = 20.0 mA
1	IOL1 = 40.0 mA

PTDC0	Allowable output current when a low level is output from P70 pin**Note 4**
0	IOL1 = 20.0 mA
1	IOL1 = 40.0 mA

Note 1. This is only usable in the 64- to 100-pin R7F100G products with 384 to 768 Kbytes of code flash memory.
Note 2. This is only usable in the 100-pin R7F100G products with 384 to 768 Kbytes of code flash memory.
Note 3. This is only usable in the R7F100G products with 30 to 52 pins.
Note 4. This is only usable in the R7F100G products with 32 to 52 pins.

Caution The PTDC register is not implemented in the R7F100G products with 128 pins.

4.3.15 Port function output enable registers (PFOEx)

The PFOEx registers are used to enable timer output, serial clock output, serial data output, and clock output for pins on which these functions are multiplexed. Disable the output when timer output, serial data output, and serial clock output are only to be input to the ELCL. The PFOEx registers can be set by a 1-bit or 8-bit memory manipulation instruction.
The value of each PFOEx register following a reset is FFH.

Figure 4 - 15 Format of Port Function Output Enable Register 0 (PFOE0)

Address: F02AAH
After reset: FFH
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
PFOE0	PFOE07	PFOE06	PFOE05	PFOE04	PFOE03	PFOE02	PFOE01	PFOE00

PFOE0n	TO0n pin output enable (n = 0 to 7)
0	Timer output to the TO0n pin is disabled. The timer output is only to be used for input to the ELCL and not for the TO0n pin. The TO0n pin is available for use for a multiplexed function or the port pin.
1	Timer output to the TO0n pin is enabled. If channel n of TAU0 is not in use, the TO0n pin is available for use for the port pin or a multiplexed function.

Figure 4 - 16 Format of Port Function Output Enable Register 1 (PFOE1) (1/2)

Address: F02ABH
After reset: FFH
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
PFOE1	1	1	PFOE15	PFOE14	PFOE13	PFOE12	PFOE11	PFOE10

PFOE15	CLKA0 pin output enable
0	Serial clock output to the CLKA0 pin is disabled. The serial clock output is only to be used for input to the ELCL and not for the CLKA0 pin. The CLKA0 pin is available for use for the port pin or a multiplexed function.
1	Serial clock output to the CLKA0 pin is enabled. If the serial clock output is not in use, the CLKA0 pin is available for use for the port pin or a multiplexed function.

PFOE14	TxDA0 pin output enable
0	Serial data output to the TxDA0 pin is disabled. The serial data output is only to be used for input to the ELCL and not for the TxDA0 pin. The TxDA0 pin is available for use for the port pin or a multiplexed function.
1	Serial data output to the TxDA0 pin is enabled. If UARTA0 is not in use, the TxDA0 pin is available for use for the port pin or a multiplexed function.

Figure 4 - 16 Format of Port Function Output Enable Register 1 (PFOE1) (2/2)

PFOE13	SCK01 and SCL01 pins output enable
0	Serial clock output to the SCK01 and SCL01 pins is disabled. The serial clock output is only to be used for input to the ELCL and not for the SCK01 or SCL01 pin. The SCK01 and SCL01 pins are available for use for the port pins or multiplexed functions.
1	Serial clock output to the SCK01 and SCL01 pins is enabled. If channel 1 of SAU0 is not in use, the SCK01 and SCL01 pins are available for use for the port pins or multiplexed functions.

PFOE12	SCK00 and SCL00 pins output enable
0	Serial clock output to the SCK00 and SCL00 pins is disabled. The serial clock output is only to be used for input to the ELCL and not for the SCK00 or SCL00 pin. The SCK00 and SCL00 pins are available for use for the port pins or multiplexed functions.
1	Serial clock output to the SCK00 and SCL00 pins is enabled. If channel 0 of SAU0 is not in use, the SCK00 and SCL00 pins are available for use for the port pins or multiplexed functions.

PFOE11	SO01 pin output enable
0	Serial data output to the SO01 pin is disabled. The serial data output is only to be used for input to the ELCL and not for the SO01 pin. The SO01 pin is available for use for the port pin or a multiplexed function.
1	Serial data output to the SO01 pin is enabled. If channel 1 of SAU0 is not in use, the SO01 pin is available for use for the port pin or a multiplexed function.

PFOE10	SO00 and TxD0 pins output enable
0	Serial data output to the SO00 and TxD0 pins is disabled. The serial data output is only to be used for input to the ELCL and not for the SO00 or TxD0 pin. The SO00 and TxD0 pins are available for use for the port pins or multiplexed functions.
1	Serial data output to the SO00 and TxD0 pins is enabled. If channel 0 of SAU0 is not in use, the SO00 and TxD0 pins are available for use for the port pins or multiplexed functions.

4.3.16 Port mode select register (PMS)

The PMS register is used to specify whether the value in the output latch for a port is read or the output level on a port pin is read when the pin is in output mode (the PMmn bit of the corresponding port mode register (PMxx) is 0). For details, see **27.3.11.1 Port mode select register (PMS)**.

4.4 Port Function Operations

Port operations differ depending on whether the input or output mode is selected, 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 state does not change. Therefore, byte data can be written to the ports used for both input and output. 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.
2. Input mode
   The pin state 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 the read value. The result of the operation is written to the output latch, but since the output buffer is off, the pin state does not change. Therefore, byte data can be written to the ports used for both input and output. The data of the output latch is cleared when a reset signal is generated.

4.4.4 Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching the EV_{DD} voltage

Communications with external devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) to this device is possible by switching EV_{DD}**Note** through the general-purpose I/O pins, in accord with the power supply of the device to be connected.

Note $EV_{DD} \leq V_{DD}$ must be satisfied.

4.4.5 Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching I/O buffers

The port input mode registers (PIMxx) and port output mode registers (POMxx) can be used to switch the I/O buffers to enable communications with external devices that have different operating voltages (1.8 V, 2.5 V, or 3 V) to this device. To receive input from external devices operating at different voltages, set the relevant bits of the port input mode registers (PIM0, PIM1, PIM3, PIM4, PIM5, PIM7, PIM8, and PIM14) in 1-bit units to switch from normal input (CMOS) to the TTL input buffers. To output data to external devices operating at different voltages, set the relevant bits of the port output mode registers (POM0, POM1, POM3, POM4, POM5, POM7, POM8, POM9, POM12, and POM14) in 1-bit units to switch from normal output (CMOS) to N-ch open-drain output (withstand voltage of V_{DD}**Note 1**/withstand voltage of EV_{DD}**Note 2**) mode. Setting the port digital input disable registers (PDIDIS0, PDIDIS1, PDIDIS3, PDIDIS4, PDIDIS5, PDIDIS7, PDIDIS8, PDIDIS9, PDIDIS12, PDIDIS13, and PDIDIS14) bit-by-bit can prevent the flow of through currents to the corresponding input buffers. The following describes connection through a serial interface to an external device operating at a different voltage.

Note 1. This applies to the 30- to 52-pin R7F100G and 32- to 48-pin R7F104G products.
Note 2. This applies to the R7F100G products with 64 to 128 pins.

1. Procedure for setting input pins of UART0 to UART3, UARTA0, UARTA1, CSI00, CSI01, CSI10, CSI20, CSI30, and CSI31 for use with the TTL input buffers

Function	Port	Product
UART0	P11	30- to 128-pin R7F100G and 32- to 48-pin R7F104G
	(P16)	30- to 128-pin R7F100G and 32- to 48-pin R7F104G
UART1	P03	52- to 128-pin R7F100G and 32- to 48-pin R7F104G
	(P81)	100- to 128-pin R7F100G
	P01	30- to 48-pin R7F100G
UART2	P14	30- to 128-pin R7F100G and 48-pin R7F104G
UART3	P143	80- to 128-pin R7F100G
UARTA0	P84	100- and 128-pin R7F100G
	P71	36- to 80-pin R7F100G and 32- to 48-pin R7F104G
UARTA1	P33	128-pin R7F100G
	P41	44- to 100-pin R7F100G
CSI00	P10, P11	30- to 128-pin R7F100G and 32- to 48-pin R7F104G
	(P55, P16)	64- to 128-pin R7F100G
CSI01	P43, P44	80- to 128-pin R7F100G
CSI10	P04, P03	64- to 128-pin R7F100G
	(P80, P81)	100- and 128-pin R7F100G
CSI20	P15, P14	30- to 128-pin R7F100G
CSI30	P142, P143	80- to 128-pin R7F100G
CSI31	P54, P53	80- to 128-pin R7F100G

Remark Pin functions in parentheses can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

<1> Pull up the input pin to be used to the voltage of the target device via an external resistor. The on-chip pull-up resistor cannot be used for this purpose.

<2> Set the corresponding bit of the PIM0, PIM1, PIM3 to PIM5, PIM8, and PIM14 registers to 1 to switch to the TTL input buffer. For V_{IH} and V_{IL}, refer to the DC characteristics when the TTL input buffer is selected.

<3> Enable the operation of the serial array unit and set the mode to the UART/simplified SPI (CSI**Note**) mode.

Note Although the CSI function is generally called SPI, it is also called CSI in this product, so it is referred to as such in this manual.

2. Procedure for setting output pins of UART0 to UART3, UARTA0, UARTA1, CSI00, CSI01, CSI10, CSI20, CSI30, and CSI31 for use with the N-ch open-drain output mode

Function	Port	Product
UART0	P12	30- to 128-pin R7F100G and 32- to 48-pin R7F104G
	(P17)	30- to 128-pin R7F100G and 32- to 48-pin R7F104G
UART1	P02	52- to 128-pin R7F100G and 32- to 48-pin R7F104G
	(P82)	100- to 128-pin R7F100G
	P00	30- to 48-pin R7F100G
UART2	P13	30- to 128-pin R7F100G and 48-pin R7F104G
UART3	P144	80- to 128-pin R7F100G
UARTA0	P83	100- and 128-pin R7F100G
	P72	36- to 80-pin R7F100G and 32- to 48-pin R7F104G
UARTA1	P34	128-pin R7F100G
	P42	64- to 100-pin R7F100G
	P120	44- to 52-pin R7F100G
CSI00	P10, P12	30- to 128-pin R7F100G and 32- to 48-pin R7F104G
	(P55, P17)	64- to 128-pin R7F100G
CSI01	P43, P45	80- to 128-pin R7F100G
CSI10	P04, P02	64- to 128-pin R7F100G
	(P80, P82)	100- and 128-pin R7F100G
CSI20	P15, P13	30- to 128-pin R7F100G
CSI30	P142, P144	80- to 128-pin R7F100G
CSI31	P54, P52	80- to 128-pin R7F100G

Remark Pin functions in parentheses can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

<1> Pull up the input pin to be used to the voltage of the target device via an external resistor. The on-chip pull-up resistor cannot be used for this purpose.
<2> The port pins are set for input (Hi-Z) after the reset state is released.
<3> Set the corresponding bit of the PDIDIS0, PDIDIS1, PDIDIS3, PDIDIS4, PDIDIS5, PDIDIS8, and PDIDIS14 registers to 1 to disable input to the input buffer.
<4> Set the output latch of the corresponding port to 1.
<5> Set the corresponding bit of the POM0, POM1, POM3 to POM5, POM8, and POM14 registers to 1 to set the N-ch open-drain output (withstand voltage of V_{DD}**Note 1**/withstand voltage of EV_{DD}**Note 2**) mode.
<6> Enable the operation of the serial array unit and set the mode to the UART/simplified SPI (CSI**Note 3**) mode.
<7> Set the corresponding bit of the PM0, PM1, PM3 to PM5, PM8, and PM14 registers to the output mode. At this time, the output data is high level, so the pin is in the Hi-Z state.

Note 1. This applies to the 30- to 52-pin R7F100G and 32- to 48-pin R7F104G products.
Note 2. This applies to the R7F100G products with 64 to 128 pins.
Note 3. Although the CSI function is generally called SPI, it is also called CSI in this product, so it is referred to as such in this manual.

3. Procedure for setting I/O pins of IIC00, IIC01, IIC10, IIC20, IIC30, and IIC31 for use in connection with a device operating at a different voltage (1.8 V, 2.5 V, or 3 V)
   - P10 and P11 for IIC00
   - P43 and P44 for IIC01
   - P03 and P04 (P81 and P80) for IIC10
   - P14 and P15 for IIC20
   - P142 and P143 for IIC30
   - P53 and P54 for IIC31

   Remark Pin functions in parentheses can be assigned via settings in the peripheral I/O redirection register (PIOR). For details, see **4.3.10 Peripheral I/O redirection register (PIOR)**.

   <1> Pull up the input pin to be used to the voltage of the target device via an external resistor. The on-chip pull-up resistor cannot be used for this purpose.
   <2> The port pins are set for input (Hi-Z) after the reset state is released.
   <3> Set the output latch of the corresponding port to 1.
   <4> Set the corresponding bit of the POM0, POM1, POM4, POM5, POM8, and POM14 registers to 1 to set the N-ch open-drain output (withstand voltage of V_{DD}**Note 1**/withstand voltage of EV_{DD}**Note 2**) mode.
   <5> Set the corresponding bit of the PIM0, PIM1, PIM4, PIM5, PIM8, and PIM14 registers to 1 to switch to the TTL input buffer. For V_{IH} and V_{IL}, refer to the DC characteristics when the TTL input buffer is selected.
   <6> Enable the operation of the serial array unit and set the mode to the simplified I^2C mode.
   <7> Set the corresponding bit of the PM0, PM1, PM4, PM5, PM8, and PM14 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.

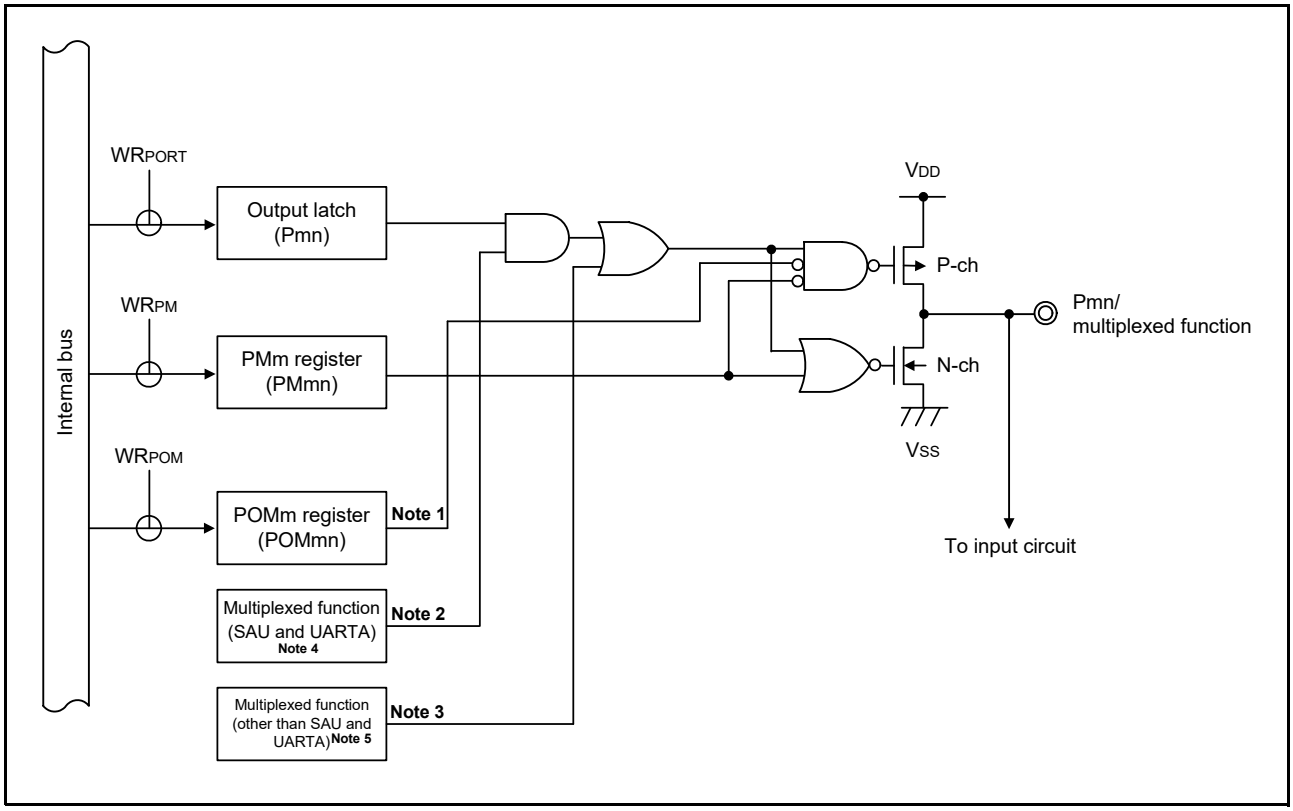
   Note 1. This applies to the 30- to 52-pin R7F100G and 32- to 48-pin R7F104G products.
   Note 2. This applies to the R7F100G products with 64 to 128 pins.

4.5 Register Settings When Using Multiplexed Function

4.5.1 Basic concept of using multiplexed functions

For a pin to which an analog function is also assigned, start by using the corresponding port mode control A register (PMCAxx) to specify whether to use the pin for the analog function or for digital input/output. Similarly, for a pin to which a capacitance measurement function is also assigned, use the corresponding port mode control T register (PMCTxx) to specify whether to use the pin for the capacitance measurement function or for digital input/output. For a pin to which a logic and event link controller output is also assigned, use the corresponding port mode control E register (PMCExx) to specify whether to use the pin for the logic and event link controller function. **Figure 4 - 17** shows the basic configuration of the output circuit for a pin that is used for digital input/output. The output from the output latch for the port-pin function and the output from a SAU or UARTA function multiplexed on the pin (excluding the clock output from a UARTA module) are input to an AND gate. The output of the AND gate is input to an OR gate. Outputs from functions other than the SAU and UARTA (timer array unit, realtime clock, clock/buzzer output, IICA, etc.) that are multiplexed on the same pin are also connected as inputs to the OR gate. Using a pin for the port function or a multiplexed function in such cases requires the appropriate settings to ensure that any unused multiplexed functions do not interfere with the output of the function intended for use. **Table 4 - 6** is a list indicating the concept of the basic settings for such cases.

Figure 4 - 17 Basic Configuration of Output Circuit for a Pin



Note 1. For pins that do not have corresponding bits in the POMm registers, consider this signal to be 0.
Note 2. For pins that do not have multiplexed functions of the indicated types, consider this signal to be 1.
Note 3. For pins that do not have multiplexed functions of the indicated types, consider this signal to be 0.
Note 4. The UARTA clock output is not included.
Note 5. The UARTA clock output is included.

Remark m: Port number (m = 0 to 15); n: Bit number (n = 0 to 7)

Table 4 - 6 Concept of Basic Settings

Output Function of Used Pin	Output Settings of Unused Multiplexed Function		
	Port-pins Output Function	Output Functions of the SAU and UARTA[Note 1]	Output Functions other than those of the SAU and UARTA[Note 2]
Port-pins output function	—	Output at the high level	Output at the low level
Output functions of the SAU and UARTA[Note 1]	High level	Output at the high level	Output at the low level
Output functions other than those of the SAU and UARTA[Note 2]	Low level	Don't care	Output at the low level[Note 3]

Note 1. The UARTA clock output is not included.

Note 2. The UARTA clock output is included.

Note 3. The output of the multiplexed functions which are not in use must be set to the low level because two or more output functions other than those of the SAU and UARTA may be multiplexed on the same pin. The output of the multiplexed functions which are not in use must be set to the high level because two or more output functions of the SAU and UARTA may be multiplexed on the same pin. For details on the setting method, see **4.5.2 Register settings when the output of a multiplexed function is not used**.

4.5.2 Register settings when the output of a multiplexed function is not used

When the output of a multiplexed function of the pin is not used, the following settings should be made. Note that when the peripheral I/O redirection function is applicable to a multiplexed function, the output can be switched to another pin by setting the peripheral I/O redirection register (PIOR). This allows usage of the port function or other multiplexed function assigned to the target pin.

1. SOp = 1, TxDq = 1 (settings when the serial output (SOp/TxDq) of SAU is not used)
   When the serial output (SOp/TxDq) is not used, such as a case in which only the serial input of SAU is used, set the bit in serial output enable register m (SOEm) which corresponds to the unused output to 0 (output disabled) and set the SOmn bit in serial output register m (SOm) to 1 (high-level output). When SOp and TxDq pins are to be used for the port pin functions, set the PFOE1x bits corresponding to the SOp, TxDq, and SCKp pins to 1. If a pin is to be used for a multiplexed function other than the port pin function, the corresponding PFOE1x bit can be set to 0. These are the same settings as the initial state.

2. SCKp = 1, SDAr = 1, SCLr = 1 (settings when channel n in SAU is not used)
   When SAU is not used, set bit n (SEmn) in serial channel enable status register m (SEm) to 0 (operation stopped state), set the bit in serial output enable register m (SOEm) which corresponds to the unused output to 0 (output disabled), and set the SOmn and CKOmn bits in serial output register m (SOm) to 1 (high-level output). When SCKp, SDAr, and SCLr pins are to be used for the port pin functions, set the PFOE1x bits corresponding to the SOp, TxDq, and SCKp pins to 1. If a pin is to be used for a multiplexed function other than the port pin function, the corresponding PFOE1x bit can be set to 0. These are the same settings as the initial state.

3. TOmn = 0 (settings when the output of channel n in TAU is not used)
   When the TOmn output of TAU is not used, set the bit in timer output enable register 0 (TOE0) which corresponds to the unused output to 0 (output disabled) and set the corresponding bit in timer output register 0 (TO0) to 0 (low-level output). These are the same settings as the initial state. Regardless of the selection for timer output operation, a TOmn output can also be stopped by setting the PFOE0x bit corresponding to the TOmn pin to 0.

4. SDAAn = 0, SCLAn = 0 (setting when IICA is not used)
   When IICA is not used, set the IICEn bit in IICA control register n0 (IICCTLn0) to 0 (operation stopped). This is the same setting as the initial state.

5. PCLBUZn = 0 (setting when clock/buzzer output is not used)
   When the clock/buzzer output is not used, set the PCLOEn bit in clock output select register n (CKSn) to 0 (output disabled). This is the same setting as the initial state.

6. TxDAn = 1, CLKAn = 0 (settings when the UARTA is not used)
   When the UARTA is not used, set the UARTAENn, TXEAn, and RXEAn bits in operation mode setting register n0 (ASIMAn0) to 0 (operation disabled). In addition, set the PFOE1x bit corresponding to the TxDAn pin to 1. These are the same settings as the initial state. Output of the CLKA0 signal from the CLKA0 pin can also be stopped by setting the PFOE1x bit corresponding to the CLKA0 pin to 0 regardless of the selection for UARTA operation.

Remark p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), q: UART number (q = 0 to 3),
r: IIC number (r = 00, 01, 10, 11, 20, 21, 30, 31), n: UARTA number (n = 0, 1)

4.5.3 Register settings and port pin state

Table 4 - 7 shows the correspondence between register settings and port pin state.

Table 4 - 7 Correspondence between Register Settings and Port Pin State

PMCAxx	PMCTxx	PMCExx	PMxx	Pxx	PUxx	CCDE0x	CCSx	Pin State
1	×	×	×	×	×	×	×	Analog input/output
0	1	×	×	×	×	×	×	Capacitance measurement
0	0	1	0	×	×	×	×	ELCL output
0	0	×	1	×	1	×	×	Pulled up
0	0	×	1	×	0	×	×	Hi-Z
0	0	0	0	1	×	×	×	Port output (high level)
0	0	0	0	0	×	1	000	Hi-Z
0	0	0	0	0	×	1	001 010 011 100	Controlled current drive port
0	0	0	0	0	×	0	×	Port output (low level)

Remark ×: Don’t care

4.5.4 Examples of register settings for port and multiplexed functions

Table 4 - 8 and **Table 4 - 9** show examples of register settings for port and multiplexed functions. The registers used to control the port functions should be set as shown in **Table 4 - 8** and **Table 4 - 9**. See the following remark for legends.

Remark —: Not supported
×: Don’t care
PIOR: Peripheral I/O redirection register
POMxx: Port output mode registers
PMCAxx: Port mode control A registers
PMCTxx: Port mode control T registers
PMCExx: Port mode control E registers
CCDE: Output current control enable register
CCSx: Output current select registers
PMxx: Port mode registers
Pxx: Port output latch
Functions in parentheses can be assigned via settings in the peripheral I/O redirection register (PIOR).

Caution In examples of register settings, the port digital input disable register (PDIDISxx) is set for input to the input buffer being enabled.

Table 4 - 8 Examples of Register and Output Latch Settings for Multiplexed Functions of 30- to 64-pin R7F100G Products with 96- or 128-Kbyte Flash Memory (1/18)

Pin Name	Function Used		PIOR	POMxx	PMCAxx	PMCTxx	PMCExx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O								SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock								
P00	P00	Input	—	×	0 **Note 1**	—	—	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0 **Note 1**	—	—	0	0/1	TxD1 = 1 **Note 2**	—								
		N-ch open-drain output	—	1	0 **Note 1**	—	—	0	0/1										
	ANI17	Analog input	—	×	1	—	—	1	×	×	—	✓	✓	—	—	—	—	—	—
	EI00	Input	—	×	0 **Note 1**	—	—	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓
	TI00	Input	—	×	0 **Note 1**	—	—	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓
	TxD1	Output	—	0/1	0 **Note 1**	—	—	0	1	×	—	✓	✓	✓	✓	✓	✓	—	—
P01	P01	Input	—	—	0 **Note 1**	—	0	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	—	0 **Note 1**	—	0	0	0/1	—	TO00 = 0								
	ANI16	Analog input	—	—	1	—	0	1	×	—	×	✓	✓	—	—	—	—	—	—
	EI01	Input	—	—	0 **Note 1**	—	0	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓
	EO01	Output	—	—	0 **Note 1**	—	1	0	×	—	—	✓	✓	✓	✓	✓	✓	✓	✓
	TO00	Output	—	—	0 **Note 1**	—	0	0	0	—	×	✓	✓	✓	✓	✓	✓	✓	✓
	RxD1	Input	—	—	0 **Note 1**	—	0	1	×	—	×	✓	✓	✓	✓	✓	✓	—	—
P02	P02	Input	—	×	0	—	—	1	×	×	—	—	—	—	—	—	—	✓	✓
		Output	—	0	0	—	—	0	0/1	TxD1/ SO10 = 1 **Note 3**	—								
		N-ch open-drain output	—	1	0	—	—	0	0/1										
	ANI17	Analog input	—	×	1	—	—	1	×	×	—	—	—	—	—	—	—	✓	✓
	TxD1	Output	—	0/1	0	—	—	0	1	×	—	—	—	—	—	—	—	✓	✓
	SO10	Output	—	0/1	0	—	—	0	1	×	—	—	—	—	—	—	—	—	✓
P03	P03	Input	—	×	0	—	—	1	×	×	—	—	—	—	—	—	—	✓	✓
		Output	—	0	0	—	—	0	0/1	SDA10 = 1 **Note 3**	—								
		N-ch open-drain output	—	1	0	—	—	0	0/1										
	ANI16	Analog input	—	×	1	—	—	1	×	×	—	—	—	—	—	—	—	✓	✓
	SI10	Input	—	×	0	—	—	1	×	×	—	—	—	—	—	—	—	—	✓
	RxD1	Input	—	×	0	—	—	1	×	×	—	—	—	—	—	—	—	✓	✓
	SDA10	I/O	—	1	0	—	—	0	1	×	—	—	—	—	—	—	—	—	✓

Note 1. This setting is only applicable in the R7F100G products with 30 and 32 pins.

Note 2. This setting is only applicable in the R7F100G products with 30 to 48 pins.

Note 3. This setting is only applicable in the R7F100G products with 64 pins.

Table 4 - 8 Examples of Register and Output Latch Settings for Multiplexed Functions of 30- to 64-pin R7F100G Products with 96- or 128-Kbyte Flash Memory (2/18)

Pin Name	Function Used		PIOR	POMxx	PMCAxx	PMCTxx	PMCExx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O								SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock								
P04	P04	Input	—	×	—	—	—	1	×	×	—	—	—	—	—	—	—	—	✓
		Output	—	0	—	—	—	0	0/1	SCK10/ SCL10 = 1	—								
		N-ch open-drain output	—	1	—	—	—	0	0/1										
	SCK10	Input	—	×	—	—	—	1	×	×	—	—	—	—	—	—	—	—	✓
		Output	—	0/1	—	—	—	0	1	×	—	—	—	—	—	—	—	—	✓
	SCL10	Output	—	0/1	—	—	—	0	1	×	—	—	—	—	—	—	—	—	✓
P05	P05	Input	—	—	—	0	—	1	×	—	×	—	—	—	—	—	—	—	✓
		Output	—	—	—	0	—	0	0/1	—	TO05 = 0								
	TS10	I/O	—	—	—	1	—	1	×	—	×	—	—	—	—	—	—	—	✓
	TI05	Input	PIOR0 = 0	—	—	0	—	1	×	—	×	—	—	—	—	—	—	—	✓
	TO05	Output	PIOR0 = 0	—	—	0	—	0	0	—	×	—	—	—	—	—	—	—	✓
P06	P06	Input	—	—	—	0	—	1	×	—	×	—	—	—	—	—	—	—	✓
		Output	—	—	—	0	—	0	0/1	—	TO06 = 0 CLKA0 = 0								
	TS11	I/O	—	—	—	1	—	1	×	—	×	—	—	—	—	—	—	—	✓
	TI06	Input	PIOR0 = 0	—	—	0	—	1	×	—	×	—	—	—	—	—	—	—	✓
	TO06	Output	PIOR0 = 0	—	—	0	—	0	0	—	CLKA0 = 0	—	—	—	—	—	—	—	✓
	CLKA0	Output	—	—	—	0	—	0	0	—	TO06 = 0	—	—	—	—	—	—	—	✓

Table 4 - 8 Examples of Register and Output Latch Settings for Multiplexed Functions of 30- to 64-pin R7F100G Products with 96- or 128-Kbyte Flash Memory (3/18)

Pin Name	Function Used		PIOR	POMxx	PMCAxx	PMCExx	CCDE	CCSx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O									SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock								
P10	P10	Input	—	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	—	0	—	—	0	0/1	SCK00/ SCL00 = 1	(TO07) = 0								
		N-ch open-drain output	—	1	—	0	—	—	0	0/1										
	EI10	Input	—	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	EO10	Output	—	0/1	—	1	—	—	0	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	SCK00	Input	PIOR1 = 0	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
		Output	PIOR1 = 0	0/1	—	0	—	—	0	1	×	(TO07) = 0	✓	✓	✓	✓	✓	✓	✓	✓
	SCL00	Output	PIOR1 = 0	0/1	—	0	—	—	0	1	×	(TO07) = 0	✓	✓	✓	✓	✓	✓	✓	✓
	(TI07)	Input	PIOR0 = 1	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	(TO07)	Output	PIOR0 = 1	0	—	0	—	—	0	0	×	×	✓	✓	✓	✓	✓	✓	✓	✓
P11	P11	Input	—	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	—	0	—	—	0	0/1	SDA00 = 1	(TO06) = 0	✓	✓	✓	✓	✓	✓	✓	✓
		N-ch open-drain output	—	1	—	0	—	—	0	0/1										
	EI11	Input	—	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	EO11	Output	—	0/1	—	1	—	—	0	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	SI00	Input	PIOR1 = 0	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	RxD0	Input	PIOR1 = 0	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	SDA00	I/O	PIOR1 = 0	1	—	0	—	—	0	1	×	(TO06) = 0	✓	✓	✓	✓	✓	✓	✓	✓
	(TI06)	Input	PIOR0 = 1	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	(TO06)	Output	PIOR0 = 1	0	—	0	—	—	0	0	×	×	✓	✓	✓	✓	✓	✓	✓	✓
P12	P12	Input	—	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	—	0	—	—	0	0/1	SO00/ TxD0 = 1	(TO05) = 0								
		N-ch open-drain output	—	1	—	0	—	—	0	0/1										
	EI12	Input	—	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	EO12	Output	—	0/1	—	1	—	—	0	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	SO00	Output	PIOR1 = 0	0/1	—	0	—	—	0	1	×	(TO05) = 0	✓	✓	✓	✓	✓	✓	✓	✓
	TxD0	Output	PIOR1 = 0	0/1	—	0	—	—	0	1	×	(TO05) = 0	✓	✓	✓	✓	✓	✓	✓	✓
	(INTP5)	Input	PIOR4 = 1	×	—	0	—	—	1	×	×	×	—	—	—	—	—	—	—	✓
	(TI05)	Input	PIOR0 = 1	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	(TO05)	Output	PIOR0 = 1	0	—	0	—	—	0	0	×	×	✓	✓	✓	✓	✓	✓	✓	✓

Table 4 - 8 Examples of Register and Output Latch Settings for Multiplexed Functions of 30- to 64-pin R7F100G Products with 96- or 128-Kbyte Flash Memory (4/18)

Pin Name	Function Used		PIOR	POMxx	PMCAxx	PMCExx	CCDE	CCSx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O									SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock								
P13	P13	Input	—	×	0	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0	—	—	0	0/1	TxD2/ SO20 = 1	(TO04) = 0 (SDAA0) = 0								
		N-ch open-drain output	—	1	0	0	—	—	0	0/1										
	EO13	Output	—	0/1	0	1	—	—	0	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	TxD2	Output	PIOR1 = 0	0/1	0	0	—	—	0	1	×	(TO04) = 0 (SDAA0) = 0	✓	✓	✓	✓	✓	✓	✓	✓
	SO20	Output	PIOR1 = 0	0/1	0	0	—	—	0	1	×	(TO04)= 0 (SDAA0)= 0	✓	✓	✓	✓	✓	✓	✓	✓
	(SDAA0)	I/O	PIOR2 = 1	1	0	0	—	—	0	0	×	(TO04) = 0	✓	✓	✓	✓	✓	✓	✓	✓
	(TI04)	Input	PIOR0 = 1	×	0	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	(TO04)	Output	PIOR0 = 1	0	0	0	—	—	0	0	×	(SDAA0) = 0	✓	✓	✓	✓	✓	✓	✓	✓
	IVREF1	Analog input	—	×	1	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
P14	P14	Input	—	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	—	0	—	—	0	0/1	SDA20 = 1	VCOUT1 = 0 (TO03) = 0 (SCLA0) = 0								
		N-ch open-drain output	—	1	—	0	—	—	0	0/1										
	EO14	Output	—	0/1	—	1	—	—	0	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	VCOUT1	Output	—	0	—	0	—	—	0	0	×	(TO03) = 0 (SCLA0) = 0	✓	✓	✓	✓	✓	✓	✓	✓
	RxD2	Input	PIOR1 = 0	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	SI20	Input	PIOR1 = 0	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	SDA20	I/O	PIOR1 = 0	1	—	0	—	—	0	1	×	VCOUT1 = 0 (TO03) = 0 (SCLA0) = 0	✓	✓	✓	✓	✓	✓	✓	✓
	(SCLA0)	I/O	PIOR2 = 1	1	—	0	—	—	0	0	×	VCOUT1 = 0 (TO03) = 0	✓	✓	✓	✓	✓	✓	✓	✓
	(TI03)	Input	PIOR0 = 1	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	(TO03)	Output	PIOR0 = 1	0	—	0	—	—	0	0	×	VCOUT1 = 0 (SCLA0) = 0	✓	✓	✓	✓	✓	✓	✓	✓
P15	P15	Input	—	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	—	0	—	—	0	0/1	SCK20/ SCL20 = 1	PCLBUZ1 = 0Note (TO02) = 0								
		N-ch open-drain output	—	1	—	0	—	—	0	0/1										
	EO15	Output	—	0/1	—	1	—	—	0	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	PCLBUZ1	Output	—	0	—	0	—	—	0	0	×	(TO02) = 0	✓	✓	✓	✓	✓	✓	✓	—
	SCK20	Input	PIOR1 = 0	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
		Output	PIOR1 = 0	0/1	—	0	—	—	0	1	×	PCLBUZ1 = 0Note (TO02) = 0	✓	✓	✓	✓	✓	✓	✓	✓
	SCL20	Output	PIOR1 = 0	0/1	—	0	—	—	0	1	×	PCLBUZ1 = 0Note (TO02) = 0	✓	✓	✓	✓	✓	✓	✓	✓
	(TI02)	Input	PIOR0 = 1	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	(TO02)	Output	PIOR0 = 1	0	—	0	—	—	0	0	×	PCLBUZ1 = 0Note	✓	✓	✓	✓	✓	✓	✓	✓

Note This setting is only applicable in the R7F100G products with 30 to 52 pins.

Table 4 - 8 Examples of Register and Output Latch Settings for Multiplexed Functions of 30- to 64-pin R7F100G Products with 96- or 128-Kbyte Flash Memory (5/18)

Pin Name	Function Used		PIOR	POMxx	PMCAxx	PMCExx	CCDE	CCSx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O									SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock								
P16	P16	Input	—	—	—	0	CCDE00 = 0	CCS02 to CCS00 = xxxB	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	—	—	0	CCDE00 = 0	CCS02 to CCS00 = xxxB	0	0/1	×	TO01 = 0								
	CCD00	Output	—	—	—	0	CCDE00 = 1	CCS02 to CCS00 = 001B to 011B	0	0	×	TO01 = 0	✓	✓	✓	✓	✓	✓	✓	✓
	EO16	Output	—	—	—	1	CCDE00 = 0	CCS02 to CCS00 = xxxB	0	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	TI01	Input	—	—	—	0	CCDE00 = 0	CCS02 to CCS00 = xxxB	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	TO01	Output	—	—	—	0	CCDE00 = 0	CCS02 to CCS00 = xxxB	0	0	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	INTP5	Input	PIOR4 = 0 **Note**	—	—	0	CCDE00 = 0	CCS02 to CCS00 = xxxB	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	(SI00)	Input	PIOR1 = 1	—	—	0	CCDE00 = 0	CCS02 to CCS00 = xxxB	1	×	×	×	—	—	—	—	—	—	—	✓
	(RxD0)	Input	PIOR1 = 1	—	—	0	CCDE00 = 0	CCS02 to CCS00 = xxxB	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
P17	P17	Input	—	×	—	0	CCDE01 = 0	CCS02 to CCS00 = xxxB	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	—	0	CCDE01 = 0	CCS02 to CCS00 = xxxB	0	0/1	(TxD0) = 1 (SO00) = 1 **Note**	TO02 = 0								
		N-ch open-drain output	—	1	—	0	CCDE01 = 0	CCS02 to CCS00 = xxxB	0	0/1										
	CCD01	Output	—	0/1	—	0	CCDE01 = 1	CCS02 to CCS00 = 001B to 011B	0	0	×	TO02 = 0	✓	✓	✓	✓	✓	✓	✓	✓
	EO17	Output	—	0/1	—	1	CCDE01 = 0	CCS02 to CCS00 = xxxB	0	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	TI02	Input	PIOR0 = 0	×	—	0	CCDE01 = 0	CCS02 to CCS00 = xxxB	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	TO02	Output	PIOR0 = 0	0	—	0	CCDE01 = 0	CCS02 to CCS00 = xxxB	0	0	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	(TxD0)	Output	PIOR1 = 1	0/1	—	0	CCDE01 = 0	CCS02 to CCS00 = xxxB	0	1	×	TO02 = 0	✓	✓	✓	✓	✓	✓	✓	✓
	(SO00)	Output	PIOR1 = 1	0/1	—	0	CCDE01 = 0	CCS02 to CCS00 = xxxB	0	1	×	TO02 = 0	—	—	—	—	—	—	—	✓

Note This setting is only applicable in the R7F100G products with 64 pins.

Table 4 - 8 Examples of Register and Output Latch Settings for Multiplexed Functions of 30- to 64-pin R7F100G Products with 96- or 128-Kbyte Flash Memory (6/18)

Pin Name	Function Used		PMCAxx	PMxx	Pxx	30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O											
P20	P20	Input	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓
		Output	0	0	0/1								
	ANI0	Analog input	1	1	×	✓	✓	✓	✓	✓	✓	✓	✓
	AVREFP	Reference voltage	1	1	×	✓	✓	✓	✓	✓	✓	✓	✓
	EI20	Input	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓
P21	P21	Input	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓
		Output	0	0	0/1								
	ANI1	Analog input	1	1	×	✓	✓	✓	✓	✓	✓	✓	✓
	AVREFM	Reference voltage	1	1	×	✓	✓	✓	✓	✓	✓	✓	✓
	EI21	Input	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓
P22	P22	Input	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓
		Output	0	0	0/1								
	ANI2	Analog input	1	1	×	✓	✓	✓	✓	✓	✓	✓	✓
	ANO0	Analog output	1	1	×	✓	✓	✓	✓	✓	✓	✓	✓
	EI22	Input	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓
P23	P23	Input	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓
		Output	0	0	0/1								
	ANI3	Analog input	1	1	×	✓	✓	✓	✓	✓	✓	✓	✓
	ANO1	Analog output	1	1	×	✓	✓	✓	✓	✓	✓	✓	✓
	IVREF0	Analog input	1	1	×	✓	✓	✓	✓	✓	✓	✓	✓
	EI23	Input	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓
P24	P24	Input	0	1	×	—	—	✓	✓	✓	✓	✓	✓
		Output	0	0	0/1								
	ANI4	Analog input	1	1	×	—	—	✓	✓	✓	✓	✓	✓
P25	P25	Input	0	1	×	—	—	✓	✓	✓	✓	✓	✓
		Output	0	0	0/1								
	ANI5	Analog input	1	1	×	—	—	✓	✓	✓	✓	✓	✓
P26	P26	Input	0	1	×	—	—	—	✓	✓	✓	✓	✓
		Output	0	0	0/1								
	ANI6	Analog input	1	1	×	—	—	—	✓	✓	✓	✓	✓
P27	P27	Input	0	1	×	—	—	—	—	✓	✓	✓	✓
		Output	0	0	0/1								
	ANI7	Analog input	1	1	×	—	—	—	—	✓	✓	✓	✓

Table 4 - 8 Examples of Register and Output Latch Settings for Multiplexed Functions of 30- to 64-pin R7F100G Products with 96- or 128-Kbyte Flash Memory (7/18)

Pin Name	Function Used		PIOR	PMCTxx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O					SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock								
P30	P30	Input	—	0	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0/1	SCK11/SCL11 = 1	VCOUT0 = 0 RTC1HZ = 0								
	TSCAP	—	—	1	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	EI30	Input	—	0	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	VCOUT0	Output	—	0	0	0	×	RTC1HZ = 0	✓	✓	✓	✓	✓	✓	✓	✓
	INTP3	Input	—	0	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	RTC1HZ	Output	—	0	0	0	×	VCOUT0 = 0	✓	✓	✓	✓	✓	✓	✓	✓
	SCK11	Input	—	0	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	1	×	VCOUT0 = 0 RTC1HZ = 0	✓	✓	✓	✓	✓	✓	✓	✓
	SCL11	Output	—	0	0	1	×	VCOUT0 = 0 RTC1HZ = 0	✓	✓	✓	✓	✓	✓	✓	✓
P31	P31	Input	—	0	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0/1	—	TO03 = 0 PCLBUZ0 = 0Note 1 (PCLBUZ0) = 0Note 2								
	TS01	I/O	—	1	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
	EI31	Input	—	0	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓
	TI03	Input	PIOR0 = 0	0	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓
	TO03	Output	PIOR0 = 0	0	0	0	—	PCLBUZ0 = 0Note 1 (PCLBUZ0) = 0Note 2	✓	✓	✓	✓	✓	✓	✓	✓
	INTP4	Input	—	0	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓
	PCLBUZ0	Output	—	0	0	0	—	TO03 = 0	✓	✓	✓	✓	✓	—	—	—
	(PCLBUZ0)	Output	PIOR3 = 1	0	0	0	—	TO03 = 0	—	—	—	—	—	✓	✓	✓

Note 1. This setting is only applicable in the R7F100G products with 30 to 44 pins.

Note 2. This setting is only applicable in the R7F100G products with 48 to 64 pins.

Table 4 - 8 Examples of Register and Output Latch Settings for Multiplexed Functions of 30- to 64-pin R7F100G Products with 96- or 128-Kbyte Flash Memory (8/18)

Pin Name	Function Used		PIOR	POMxx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O					SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock								
P40	P40	Input	—	—	1	×	—	—	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	—	0	0/1	—	—								
P41	P41	Input	—	—	1	×	—	×	—	—	—	—	✓	✓	✓	✓
		Output	—	—	0	0/1	—	TO07 = 0								
	RxDA1	Input	—	—	1	×	—	×	—	—	—	—	✓	✓	✓	✓
	TI07	Input	PIOR0 = 0	—	1	×	—	×	—	—	—	—	✓	✓	✓	✓
	TO07	Output	PIOR0 = 0	—	0	0	—	×	—	—	—	—	✓	✓	✓	✓
P42	P42	Input	—	×	1	×	×	×	—	—	—	—	—	—	—	✓
		Output	—	0	0	0/1	TxDA1 = 1	TO04 = 0	—	—	—	—	—	—	—	✓
		N-ch open-drain output	—	1	0	0/1			—	—	—	—	—	—	—	✓
	TxDA1	Output	—	0/1	0	1	×	TO04 = 0	—	—	—	—	—	—	—	✓
	TI04	Input	PIOR0 = 0	×	1	×	×	×	—	—	—	—	—	—	—	✓
	TO04	Output	PIOR0 = 0	0	0	0	×	×	—	—	—	—	—	—	—	✓
P43	P43	Input	—	—	1	×	—	×	—	—	—	—	—	—	—	✓
		Output	—	—	0	0/1	—	CLKA1 = 0	—	—	—	—	—	—	—	✓
	CLKA1	Output	—	—	0	0	—	×	—	—	—	—	—	—	—	✓

Table 4 - 8 Examples of Register and Output Latch Settings for Multiplexed Functions of 30- to 64-pin R7F100G Products with 96- or 128-Kbyte Flash Memory (9/18)

Pin Name	Function Used		PIOR	POMxx	PMCTxx	PMCExx	CCDE	CCSx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O									SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock								
P50	P50	Input	—	×	0	0	CCDE03 = 0	CCS02 to CCS00 = xxxB	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0	CCDE03 = 0	CCS02 to CCS00 = xxxB	0	0/1	SDA11 = 1	—								
		N-ch open-drain output	—	1	0	0	CCDE03 = 0	CCS02 to CCS00 = xxxB	0	0/1										
	TS00	I/O	—	×	1	0	CCDE03 = 0	CCS02 to CCS00 = xxxB	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓
	EI50	Input	—	×	0	0	CCDE03 = 0	CCS02 to CCS00 = xxxB	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓
	EO50	Output	—	0/1	0	1	CCDE03 = 0	CCS02 to CCS00 = xxxB	0	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓
	CCD03	Output	—	0/1	0	0	CCDE03 = 1	CCS02 to CCS00 = 001B to 011B	0	0	×	—	✓	✓	✓	✓	—	✓	✓	✓
	INTP1	Input	—	×	0	0	CCDE03 = 0	CCS02 to CCS00 = xxxB	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓
	SI11	Input	—	×	0	0	CCDE03 = 0	CCS02 to CCS00 = xxxB	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓
	SDA11	I/O	—	1	0	0	CCDE03 = 0	CCS02 to CCS00 = xxxB	0	1	×	—	✓	✓	✓	✓	✓	✓	✓	✓
P51	P51	Input	—	—	—	0	CCDE02 = 0	CCS02 to CCS00 = xxxB	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	—	—	0	CCDE02 = 0	CCS02 to CCS00 = xxxB	0	0/1	SO11 = 1	—								
	EI51	Input	—	—	—	0	CCDE02 = 0	CCS02 to CCS00 = xxxB	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓
	EO51	Output	—	—	—	1	CCDE02 = 0	CCS02 to CCS00 = xxxB	0	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓
	CCD02	Output	—	—	—	0	CCDE02 = 1	CCS02 to CCS00 = 001B to 011B	0	0	×	—	✓	✓	✓	✓	✓	✓	✓	✓
	INTP2	Input	—	—	—	0	CCDE02 = 0	CCS02 to CCS00 = xxxB	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓
	SO11	Output	—	—	—	0	CCDE02 = 0	CCS02 to CCS00 = xxxB	0	1	×	—	✓	✓	✓	✓	✓	✓	✓	✓

Table 4 - 8 Examples of Register and Output Latch Settings for Multiplexed Functions of 30- to 64-pin R7F100G Products with 96- or 128-Kbyte Flash Memory (10/18)

Pin Name	Function Used		PIOR	POMxx	PMCTxx	PMCExx	CCDE	CCSx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O									SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock								
P52	P52	Input	—	—	—	—	—	—	1	×	—	—	—	—	—	—	—	—	—	✓
		Output	—	—	—	—	—	—	0	0/1	—	—								
	(INTP10)	Input	PIOR1 = 1	—	—	—	—	—	1	×	—	—	—	—	—	—	—	—	—	✓
P53	P53	Input	—	—	—	—	—	—	1	×	—	—	—	—	—	—	—	—	—	✓
		Output	—	—	—	—	—	—	0	0/1	—	—								
	(INTP11)	Input	PIOR1 = 1	—	—	—	—	—	1	×	—	—	—	—	—	—	—	—	—	✓
P54	P54	Input	—	—	—	—	—	—	1	×	—	—	—	—	—	—	—	—	—	✓
		Output	—	—	—	—	—	—	0	0/1	—	—								
P55	P55	Input	—	×	—	—	—	—	1	×	×	×	—	—	—	—	—	—	—	✓
		Output	—	0	—	—	—	—	0	0/1	(SCK00) = 1	(PCLBUZ1) = 0								
		N-ch open-drain output	—	1	—	—	—	—	0	0/1										
	(PCLBUZ1)	Output	PIOR4 = 1	0	—	—	—	—	0	0	×	×	—	—	—	—	—	—	—	✓
	(SCK00)	Input	PIOR1 = 1	×	—	—	—	—	1	×	×	×	—	—	—	—	—	—	—	✓
		Output	PIOR1 = 1	0/1	—	—	—	—	0	1	×	(PCLBUZ1) = 0	—	—	—	—	—	—	—	✓

Table 4 - 8 Examples of Register and Output Latch Settings for Multiplexed Functions of 30- to 64-pin R7F100G Products with 96- or 128-Kbyte Flash Memory (11/18)

Pin Name	Function Used		PIOR	PMCExx	CCDE	CCSx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O							SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock								
P60	P60	Input	—	0	CCDE04 = 0	CCS42 to CCS40 = 000B	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓
		N-ch open-drain output (withstand voltage of 6 V)	—	0	CCDE04 = 0	CCS42 to CCS40 = 000B	0	0/1	—	SCLA0 = 0								
	EO60	Output	—	1	CCDE04 = 0	CCS42 to CCS40 = 000B	0	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓
	CCD04	Output	—	0	CCDE04 = 1	CCS42 to CCS40 = 001B to 100B	0	0	—	SCLA0 = 0	✓	✓	✓	✓	✓	✓	✓	✓
	SCLA0	I/O	PIOR2 = 0	0	CCDE04 = 0	CCS42 to CCS40 = 000B	0	0	—	×	✓	✓	✓	✓	✓	✓	✓	✓
P61	P61	Input	—	0	CCDE05 = 0	CCS52 to CCS50 = 000B	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓
		N-ch open-drain output (withstand voltage of 6 V)	—	0	CCDE05 = 0	CCS52 to CCS50 = 000B	0	0/1	—	SDAA0 = 0								
	EO61	Output	—	1	CCDE05 = 0	CCS52 to CCS50 = 000B	0	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓
	CCD05	Output	—	0	CCDE05 = 1	CCS52 to CCS50 = 001B to 100B	0	0	—	SDAA0 = 0	✓	✓	✓	✓	✓	✓	✓	✓
	SDAA0	I/O	PIOR2 = 0	0	CCDE05 = 0	CCS52 to CCS50 = 000B	0	0	—	×	✓	✓	✓	✓	✓	✓	✓	✓
P62	P62	Input	—	—	CCDE06 = 0	CCS62 to CCS60 = 000B	1	×	—	×	—	✓	✓	✓	✓	✓	✓	✓
		N-ch open-drain output (withstand voltage of 6 V)	—	—	CCDE06 = 0	CCS62 to CCS60 = 000B	0	0/1	—	SCLA1 = 0**Note**								
	CCD06	Output	—	—	CCDE06 = 1	CCS62 to CCS60 = 001B to 100B	0	0	—	SCLA1 = 0**Note**	—	✓	✓	✓	✓	✓	✓	✓
	SCLA1	I/O	—	—	CCDE06 = 0	CCS62 to CCS60 = 000B	0	0	—	×	—	—	—	—	✓	✓	✓	✓

Note This setting is only applicable in the R7F100G products with 44 to 64 pins.

Table 4 - 8 Examples of Register and Output Latch Settings for Multiplexed Functions of 30- to 64-pin R7F100G Products with 96- or 128-Kbyte Flash Memory (12/18)

Pin Name	Function Used		PIOR	PMCExx	CCDE	CCSx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O							SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock								
P63	P63	Input	—	—	CCDE07 = 0	CCS72 to CCS70 = 000B	1	×	—	×	—	—	—	—	✓	✓	✓	✓
		N-ch open-drain output (withstand voltage of 6 V)	—	—	CCDE07 = 0	CCS72 to CCS70 = 000B	0	0/1	—	SDAA1 = 0**Note**								
	CCD07	Output	—	—	CCDE07 = 1	CCS72 to CCS70 = 001B to 100B	0	0	—	SDAA1 = 0**Note**	—	—	—	—	✓	✓	✓	✓
	SDAA1	I/O	—	—	CCDE07 = 0	CCS72 to CCS70 = 000B	0	0	—	×	—	—	—	—	✓	✓	✓	✓

Note This setting is only applicable in the R7F100G products with 44 to 64 pins.

Table 4 - 8 Examples of Register and Output Latch Settings for Multiplexed Functions of 30- to 64-pin R7F100G Products with 96- or 128-Kbyte Flash Memory (13/18)

Pin Name	Function Used		PIOR	POMxx	PMCTxx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O						SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock								
P70	P70	Input	—	—	0	1	×	×	—	—	✓	✓	✓	✓	✓	✓	✓
		Output	—	—	0	0	0/1	SCK21/SCL21 = 1 Note 1	—								
	TS02	I/O	—	—	1	1	×	×	—	—	✓	✓	✓	✓	✓	✓	✓
	RIN0	Input	—	—	0	1	×	×	—	—	✓	✓	✓	✓	✓	✓	✓
	KR0	Input	—	—	0	1	×	×	—	—	—	—	✓	✓	✓	✓	✓
	SCK21	Input	—	—	0	1	×	×	—	—	—	✓	✓	✓	✓	✓	✓
		Output	—	—	0	0	1	×	—	—	—	✓	✓	✓	✓	✓	✓
	SCL21	Output	—	—	0	0	1	×	—	—	—	✓	✓	✓	✓	✓	✓
P71	P71	Input	—	×	0	1	×	×	—	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0	0/1	SDA21 = 1	—								
		N-ch open-drain output	—	1	0	0	0/1										
	TS03	I/O	—	×	1	1	×	×	—	—	—	✓	✓	✓	✓	✓	✓
	KR1	Input	—	×	0	1	×	×	—	—	—	—	✓	✓	✓	✓	✓
	SI21	Input	—	×	0	1	×	×	—	—	—	✓	✓	✓	✓	✓	✓
	SDA21	I/O	—	1	0	0	1	×	—	—	—	✓	✓	✓	✓	✓	✓
	RxDA0	Input	—	×	0	1	×	×	—	—	—	✓	✓	✓	✓	✓	✓
P72	P72	Input	—	×	0	1	×	×	—	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0	0/1	SO21 = 1 TxDA0 = 1	—								
		N-ch open-drain output	—	1	0	0	0/1										
	TS04	I/O	—	×	1	1	×	×	—	—	—	✓	✓	✓	✓	✓	✓
	KR2	Input	—	×	0	1	×	×	—	—	—	—	✓	✓	✓	✓	✓
	SO21	Output	—	0/1	0	0	1	TxDA0 = 1	—	—	—	✓	✓	✓	✓	✓	✓
	TxDA0	Output	—	0/1	0	0	1	SO21 = 1	—	—	—	✓	✓	✓	✓	✓	✓
P73	P73	Input	—	—	0	1	×	×	—	—	—	—	✓	✓	✓	✓	✓
		Output	—	—	0	0	0/1	SO01 = 1 Note 2	—								
	TS05	I/O	—	—	1	1	×	×	—	—	—	—	✓	✓	✓	✓	✓
	KR3	Input	—	—	0	1	×	×	—	—	—	—	✓	✓	✓	✓	✓
	SO01	Output	—	—	0	0	1	×	—	—	—	—	—	—	✓	✓	✓
P74	P74	Input	—	×	0	1	×	×	—	—	—	—	—	—	✓	✓	✓
		Output	—	0	0	0	0/1	SDA01 = 1	—								
		N-ch open-drain output	—	1	0	0	0/1										
	TS06	I/O	—	×	1	1	×	×	—	—	—	—	—	—	✓	✓	✓
	KR4	Input	—	×	0	1	×	×	—	—	—	—	—	—	✓	✓	✓
	INTP8	Input	—	×	0	1	×	×	—	—	—	—	—	—	✓	✓	✓
	SI01	Input	—	×	0	1	×	×	—	—	—	—	—	—	✓	✓	✓
	SDA01	I/O	—	1	0	0	1	×	—	—	—	—	—	—	✓	✓	✓

Note 1. This setting is only applicable in the R7F100G products with 36 to 64 pins.

Note 2. This setting is only applicable in the R7F100G products with 48 to 64 pins.

Table 4 - 8 Examples of Register and Output Latch Settings for Multiplexed Functions of 30- to 64-pin R7F100G Products with 96- or 128-Kbyte Flash Memory (14/18)

Pin Name	Function Used		PIOR	POMxx	PMCTxx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O						SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock								
P75	P75	Input	—	—	0	1	×	×	—	—	—	—	—	—	✓	✓	✓
		Output	—	—	0	0	0/1	SCK01/SCL01 = 1	—								
	TS07	I/O	—	—	1	1	×	×	—	—	—	—	—	—	✓	✓	✓
	KR5	Input	—	—	0	1	×	×	—	—	—	—	—	—	✓	✓	✓
	INTP9	Input	—	—	0	1	×	×	—	—	—	—	—	—	✓	✓	✓
	SCK01	Input	—	—	0	1	×	×	—	—	—	—	—	—	✓	✓	✓
		Output	—	—	0	0	1	×	—	—	—	—	—	—	✓	✓	✓
	SCL01	Output	—	—	0	0	1	×	—	—	—	—	—	—	✓	✓	✓
P76	P76	Input	—	—	0	1	×	—	—	—	—	—	—	—	—	✓	✓
		Output	—	—	0	0	0/1	—	—								
	TS08	I/O	—	—	1	1	×	—	—	—	—	—	—	—	—	✓	✓
	KR6	Input	—	—	0	1	×	—	—	—	—	—	—	—	—	✓	✓
	INTP10	Input	PIOR1 = 0	—	0	1	×	—	—	—	—	—	—	—	—	✓	✓
	(RxD2)	Input	PIOR1 = 1	—	0	1	×	—	—	—	—	—	—	—	—	✓	✓
P77	P77	Input	—	—	0	1	×	×	—	—	—	—	—	—	—	✓	✓
		Output	—	—	0	0	0/1	(TxD2) = 1	—								
	TS09	I/O	—	—	1	1	×	×	—	—	—	—	—	—	—	✓	✓
	KR7	Input	—	—	0	1	×	×	—	—	—	—	—	—	—	✓	✓
	INTP11	Input	PIOR1 = 0	—	0	1	×	×	—	—	—	—	—	—	—	✓	✓
	(TxD2)	Output	PIOR1 = 1	—	0	0	1	×	—	—	—	—	—	—	—	✓	✓

Table 4 - 8 Examples of Register and Output Latch Settings for Multiplexed Functions of 30- to 64-pin R7F100G Products with 96- or 128-Kbyte Flash Memory (15/18)

Pin Name	Function Used		POMxx	PMCAxx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O					SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock								
P120	P120	Input	×	0	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓
		Output	0	0	0	0/1	TxDA1 = 1	—								
		N-ch open-drain output	1	0	0	0/1										
	ANI19	Analog input	×	1	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓
	IVCMP1	Analog input	×	1	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓
	EI120	Input	×	0	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓
	TxDA1	Output	0/1	0	0	1	×	—	—	—	—	—	✓	✓	✓	—

Table 4 - 8 Examples of Register and Output Latch Settings for Multiplexed Functions of 30- to 64-pin R7F100G Products with 96- or 128-Kbyte Flash Memory (16/18)

Pin Name	Function Used		CMC		PMxx	Pxx	30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O	EXCLK, OSCSEL, EXCLKS, OSCSELS	XTSEL										
P121	P121	Input	00xxB/10xxB/11xxB	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓
			xx00B/xx10B/xx11B	1Note										
		Output	00xxB/10xxB/11xxB	0	0	0/1								
			xx00B/xx10B/xx11B	1Note										
	EI121	Input	00xxB/10xxB/11xxB	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓
			xx00B/xx10B/xx11B	1Note										
	VBAT	—	01xxB	0	1	×	—	—	—	✓	✓	✓	✓	✓
	X1	—	01xxB	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓
	XT1	—	xx01B	1	1	×	✓	✓	✓	—	—	—	—	—
P122	P122	Input	00xxB/10xxB	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓
			xx00B/xx10B	1Note										
		Output	00xxB/10xxB	0	0	0/1								
			xx00B/xx10B	1Note										
	EI122	Input	00xxB/10xxB	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓
			xx00B/xx10B	1Note										
	X2	—	01xxB	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓
	XT2	—	xx01B	1	1	×	✓	✓	✓	—	—	—	—	—
	EXCLK	Input	11xxB	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓
	EXCLKS	Input	xx11B	1	1	×	✓	✓	✓	—	—	—	—	—
P123	P123	Input	xx00B/xx10B/xx11B	0	—	×	—	—	—	✓	✓	✓	✓	✓
	XT1	—	xx01B	0	—	×	—	—	—	✓	✓	✓	✓	✓
P124	P124	Input	xx00B/xx10B	0	—	×	—	—	—	✓	✓	✓	✓	✓
	XT2	—	xx01B	0	—	×	—	—	—	✓	✓	✓	✓	✓
	EXCLKS	Input	xx11B	0	—	×	—	—	—	✓	✓	✓	✓	✓

Note This setting is only applicable in the R7F100G products with 30 to 36 pins.

Table 4 - 8 Examples of Register and Output Latch Settings for Multiplexed Functions of 30- to 64-pin R7F100G Products with 96- or 128-Kbyte Flash Memory (17/18)

Pin Name	Function Used		PMxx	Pxx	30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O										
P130	P130	Output	—	0/1	—	—	—	—	—	✓	✓	✓
P137	P137	Input	—	×	✓	✓	✓	✓	✓	✓	✓	✓
	EI137	Input	—	×	✓	✓	✓	✓	✓	✓	✓	✓
	INTP0	Input	—	×	✓	✓	✓	✓	✓	✓	✓	✓

Table 4 - 8 Examples of Register and Output Latch Settings for Multiplexed Functions of 30- to 64-pin R7F100G Products with 96- or 128-Kbyte Flash Memory (18/18)

Pin Name	Function Used		PIOR	POMxx	PMCAxx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin
	Function Name	I/O						SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock								
P140	P140	Input	—	—	—	1	×	—	×	—	—	—	—	—	✓	✓	✓
		Output	—	—	—	0	0/1	—	PCLBUZ0 = 0								
	PCLBUZ0	Output	PIOR3 = 0	—	—	0	0	—	×	—	—	—	—	—	✓	✓	✓
	INTP6	Input	—	—	—	1	×	—	×	—	—	—	—	—	✓	✓	✓
P141	P141	Input	—	—	—	1	×	—	×	—	—	—	—	—	—	—	✓
		Output	—	—	—	0	0/1	—	PCLBUZ1 = 0								
	PCLBUZ1	Output	PIOR4 = 0	—	—	0	0	—	×	—	—	—	—	—	—	—	✓
	INTP7	Input	—	—	—	1	×	—	×	—	—	—	—	—	—	—	✓
P146	P146	Input	—	—	—	1	×	—	—	—	—	—	—	✓	✓	✓	✓
		Output	—	—	—	0	0/1	—	—								
P147	P147	Input	—	—	0	1	×	—	—	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	—	0	0	0/1	—	—								
	ANI18	Analog input	—	—	1	1	×	—	—	✓	✓	✓	✓	✓	✓	✓	✓
	IVCMP0	Analog input	—	—	1	1	×	—	—	✓	✓	✓	✓	✓	✓	✓	✓
	EI147	Input	—	—	0	1	×	—	—	✓	✓	✓	✓	✓	✓	✓	✓

Table 4 - 9 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F100G Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin R7F100G Products with 128-Kbyte Flash Memory (1/22)

Pin Name	Function Used		PIOR	POMxx	PMCAxx	PMCTxx	PMCExx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin	80-pin	100-pin	128-pin
	Function Name	I/O								SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock											
P00	P00	Input	—	×	0Note 1	0	—	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0Note 1	0	—	0	0/1	TxD1 = 1 Note 2	—											
		N-ch open-drain output	—	1	0Note 1	0	—	0	0/1													
	ANI17	Analog input	—	×	1	0	—	1	×	×	—	✓	✓	—	—	—	—	—	—	—	—	—
	TS26	I/O	—	×	0Note 1	1	—	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	EI00	Input	—	×	0Note 1	0	—	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TI00	Input	—	×	0Note 1	0	—	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TxD1	Output	—	0/1	0Note 1	0	—	0	1	×	—	✓	✓	✓	✓	✓	✓	—	—	—	—	—
P01	P01	Input	—	—	0Note 1	0	0	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	—	0Note 1	0	0	0	0/1	—	TO00 = 0											
	ANI16	Analog input	—	—	1	0	0	1	×	—	×	✓	✓	—	—	—	—	—	—	—	—	—
	TS27	I/O	—	—	0Note 1	1	0	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	EI01	Input	—	—	0Note 1	0	0	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	EO01	Output	—	—	0Note 1	0	1	0	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TO00	Output	—	—	0Note 1	0	0	0	0	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	RxD1	Input	—	—	0Note 1	0	0	1	×	—	×	✓	✓	✓	✓	✓	✓	—	—	—	—	—
P02	P02	Input	—	×	0	0	—	1	×	×	—	—	—	—	—	—	—	✓	✓	✓	✓	✓
		Output	—	0	0	0	—	0	0/1	TxD1 = 1 SO10 = 1 Note 4	—											
		N-ch open-drain output	—	1	0	0	—	0	0/1													
	ANI17	Analog input	—	×	1	0	—	1	×	×	—	—	—	—	—	—	—	✓	✓	✓	✓	✓
	TS28	I/O	—	×	0	1	—	1	×	×	—	—	—	—	—	—	—	✓	✓	✓	✓	✓
	TxD1	Output	PIOR5 = 0 Note 3	0/1	0	0	—	0	1	×	—	—	—	—	—	—	—	✓	✓	✓	✓	✓
	SO10	Output	PIOR5 = 0 Note 3	0/1	0	0	—	0	1	×	—	—	—	—	—	—	—	—	✓	✓	✓	✓
P03	P03	Input	—	×	0	0	—	1	×	×	—	—	—	—	—	—	—	✓	✓	✓	✓	✓
		Output	—	0	0	0	—	0	0/1	SDA10 = 1 Note 4	—											
		N-ch open-drain output	—	1	0	0	—	0	0/1													
	ANI16	Analog input	—	×	1	0	—	1	×	×	—	—	—	—	—	—	—	✓	✓	✓	✓	✓
	TS29	I/O	—	×	0	1	—	1	×	×	—	—	—	—	—	—	—	✓	✓	✓	✓	✓
	SI10	Input	PIOR5 = 0 Note 3	×	0	0	—	1	×	×	—	—	—	—	—	—	—	—	✓	✓	✓	✓
	RxD1	Input	PIOR5 = 0 Note 3	×	0	0	—	1	×	×	—	—	—	—	—	—	—	✓	✓	✓	✓	✓
	SDA10	I/O	PIOR5 = 0 Note 3	1	0	0	—	0	1	×	—	—	—	—	—	—	—	—	✓	✓	✓	✓

Note 1. This setting is only applicable in the R7F100G products with 30 and 32 pins.

Note 2. This setting is only applicable in the R7F100G products with 30 to 48 pins.

Note 3. This setting is only applicable in the R7F100G products with 100 and 128 pins.

Note 4. This setting is only applicable in the R7F100G products with 64 to 128 pins.

Table 4 - 9 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F100G Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin R7F100G Products with 128-Kbyte Flash Memory (2/22)

Pin Name	Function Used		PIOR	POMxx	PMCAxx	PMCTxx	PMCExx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin	80-pin	100-pin	128-pin
	Function Name	I/O								SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock											
P04	P04	Input	—	×	—	—	—	1	×	×	—	—	—	—	—	—	—	—	✓	✓	✓	✓
		Output	—	0	—	—	—	0	0/1	SCK10/ SCL10 = 1	—											
		N-ch open-drain output	—	1	—	—	—	0	0/1													
	SCK10	Input	PIOR5 = 0 **Note 1**	×	—	—	—	1	×	×	—	—	—	—	—	—	—	—	✓	✓	✓	✓
		Output	PIOR5 = 0 **Note 1**	0/1	—	—	—	0	1	×	—	—	—	—	—	—	—	—	✓	✓	✓	✓
	SCL10	Output	PIOR5 = 0 **Note 1**	0/1	—	—	—	0	1	×	—	—	—	—	—	—	—	—	✓	✓	✓	✓
P05	P05	Input	—	—	—	0	—	1	×	—	×	—	—	—	—	—	—	—	✓	✓	✓	✓
		Output	—	—	—	0	—	0	0/1	—	TO05 = 0**Note 2**											
	TS10	I/O	—	—	—	1	—	1	×	—	×	—	—	—	—	—	—	—	✓	✓	✓	✓
	TI05	Input	PIOR0 = 0 **Note 2**	—	—	0	—	1	×	—	×	—	—	—	—	—	—	—	✓	✓	—	—
	TO05	Output	PIOR0 = 0 **Note 2**	—	—	0	—	0	0	—	×	—	—	—	—	—	—	—	✓	✓	—	—
P06	P06	Input	—	—	—	0	—	1	×	—	×	—	—	—	—	—	—	—	✓	✓	✓	✓
		Output	—	—	—	0	—	0	0/1	—	TO06 = 0**Note 2** CLKA0 = 0**Note 2**											
	TS11	I/O	—	—	—	1	—	1	×	—	×	—	—	—	—	—	—	—	✓	✓	✓	✓
	TI06	Input	PIOR0 = 0	—	—	0	—	1	×	—	×	—	—	—	—	—	—	—	✓	✓	—	—
	TO06	Output	PIOR0 = 0	—	—	0	—	0	0	—	CLKA0 = 0	—	—	—	—	—	—	—	✓	✓	—	—
	CLKA0	Output	—	—	—	0	—	0	0	—	TO06 = 0	—	—	—	—	—	—	—	✓	✓	—	—
P07	P07	Input	—	—	—	—	—	1	×	—	—	—	—	—	—	—	—	—	—	—	—	✓
		Output	—	—	—	—	—	0	0/1	—	—											

Note 1. This setting is only applicable in the R7F100G products with 100 and 128 pins.

Note 2. This setting is only applicable in the R7F100G products with 64 and 80 pins.

Table 4 - 9 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F100G Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin R7F100G Products with 128-Kbyte Flash Memory (3/22)

Pin Name	Function Used		PIOR	POMxx	PMCAxx	PMCExx	CCDE	CCSx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin	80-pin	100-pin	128-pin
	Function Name	I/O									SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock											
P10	P10	Input	—	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	—	0	—	—	0	0/1	SCK00/ SCL00 = 1	(TO07) = 0											
		N-ch open-drain output	—	1	—	0	—	—	0	0/1													
	EI10	Input	—	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	EO10	Output	—	0/1	—	1	—	—	0	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SCK00	Input	PIOR1 = 0	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	PIOR1 = 0	0/1	—	0	—	—	0	1	×	(TO07) = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SCL00	Output	PIOR1 = 0	0/1	—	0	—	—	0	1	×	(TO07) = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(TI07)	Input	PIOR0 = 1	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(TO07)	Output	PIOR0 = 1	0	—	0	—	—	0	0	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P11	P11	Input	—	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	—	0	—	—	0	0/1	SDA00 = 1	(TO06) = 0											
		N-ch open-drain output	—	1	—	0	—	—	0	0/1													
	EI11	Input	—	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	EO11	Output	—	0/1	—	1	—	—	0	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SI00	Input	PIOR1 = 0	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	RxD0	Input	PIOR1 = 0	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SDA00	I/O	PIOR1 = 0	1	—	0	—	—	0	1	×	(TO06) = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(TI06)	Input	PIOR0 = 1	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(TO06)	Output	PIOR0 = 1	0	—	0	—	—	0	0	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P12	P12	Input	—	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	—	0	—	—	0	0/1	SO00/ TxD0 = 1	(TO05) = 0											
		N-ch open-drain output	—	1	—	0	—	—	0	0/1													
	EI12	Input	—	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	EO12	Output	—	0/1	—	1	—	—	0	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SO00	Output	PIOR1 = 0	0/1	—	0	—	—	0	1	×	(TO05) = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TxD0	Output	PIOR1 = 0	0/1	—	0	—	—	0	1	×	(TO05) = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(INTP5)	Input	PIOR4 = 1	×	—	0	—	—	1	×	×	×	—	—	—	—	—	—	—	✓	✓	✓	✓
	(TI05)	Input	PIOR0 = 1	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(TO05)	Output	PIOR0 = 1	0	—	0	—	—	0	0	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 4 - 9 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F100G Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin R7F100G Products with 128-Kbyte Flash Memory (4/22)

Pin Name	Function Used		PIOR	POMxx	PMCAxx	PMCExx	CCDE	CCSx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin	80-pin	100-pin	128-pin
	Function Name	I/O									SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock											
P13	P13	Input	—	×	0	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0	—	—	0	0/1	TxD2/ SO20 = 1	(TO04) = 0 (SDAA0) = 0											
		N-ch open-drain output	—	1	0	0	—	—	0	0/1													
	EO13	Output	—	0/1	0	1	—	—	0	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TxD2	Output	PIOR1 = 0	0/1	0	0	—	—	0	1	×	(TO04) = 0 (SDAA0) = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SO20	Output	PIOR1 = 0	0/1	0	0	—	—	0	1	×	(TO04) = 0 (SDAA0) = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(SDAA0)	I/O	PIOR2 = 1	1	0	0	—	—	0	0	×	(TO04) = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(TI04)	Input	PIOR0 = 1	×	0	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(TO04)	Output	PIOR0 = 1	0	0	0	—	—	0	0	×	(SDAA0) = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	IVREF1	Analog input	—	×	1	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P14	P14	Input	—	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	—	0	—	—	0	0/1	SDA20 = 1	VCOUT1 = 0 (TO03) = 0 (SCLA0) = 0											
		N-ch open-drain output	—	1	—	0	—	—	0	0/1													
	EO14	Output	—	0/1	—	1	—	—	0	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	VCOUT1	Output	—	0	—	0	—	—	0	0	×	(TO03) = 0 (SCLA0) = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	RxD2	Input	PIOR1 = 0	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SI20	Input	PIOR1 = 0	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SDA20	I/O	PIOR1 = 0	1	—	0	—	—	0	1	×	VCOUT1 = 0 (TO03) = 0 (SCLA0) = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(SCLA0)	I/O	PIOR2 = 1	1	—	0	—	—	0	0	×	VCOUT1 = 0 (TO03) = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(TI03)	Input	PIOR0 = 1	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(TO03)	Output	PIOR0 = 1	0	—	0	—	—	0	0	×	VCOUT1 = 0 (SCLA0) = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P15	P15	Input	—	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	—	0	—	—	0	0/1	SCK20/ SCL20 = 1	PCLBUZ1 = 0 **Note** (TO02) = 0											
		N-ch open-drain output	—	1	—	0	—	—	0	0/1													
	EO15	Output	—	0/1	—	1	—	—	0	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	PCLBUZ1	Output	—	0	—	0	—	—	0	0	×	(TO02) = 0	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
	SCK20	Input	PIOR1 = 0	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	PIOR1 = 0	0/1	—	0	—	—	0	1	×	PCLBUZ1 = 0 **Note** (TO02) = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SCL20	Output	PIOR1 = 0	0/1	—	0	—	—	0	1	×	PCLBUZ1 = 0 **Note** (TO02) = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(TI02)	Input	PIOR0 = 1	×	—	0	—	—	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(TO02)	Output	PIOR0 = 1	0	—	0	—	—	0	0	×	PCLBUZ1 = 0 **Note**	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Note This setting is only applicable in the R7F100G products with 30 to 52 pins.

Table 4 - 9 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F100G Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin R7F100G Products with 128-Kbyte Flash Memory (5/22)

Pin Name	Function Used		PIOR	POMxx	PMCAxx	PMCExx	CCDE	CCSx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin	80-pin	100-pin	128-pin
	Function Name	I/O									SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock											
P16	P16	Input	—	—	—	0	CCDE00 = 0	CCS02 to CCS00 = xxxB	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	—	—	0	CCDE00 = 0	CCS02 to CCS00 = xxxB	0	0/1	—	TO01 = 0											
	CCD00	Output	—	—	—	0	CCDE00 = 1	CCS02 to CCS00 = 001B to 011B	0	0	—	TO01 = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	EO16	Output	—	—	—	1	CCDE00 = 0	CCS02 to CCS00 = xxxB	0	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TI01	Input	—	—	—	0	CCDE00 = 0	CCS02 to CCS00 = xxxB	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TO01	Output	—	—	—	0	CCDE00 = 0	CCS02 to CCS00 = xxxB	0	0	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	INTP5	Input	PIOR4 = 0 **Note**	—	—	0	CCDE00 = 0	CCS02 to CCS00 = xxxB	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(SI00)	Input	PIOR1 = 1	—	—	0	CCDE00 = 0	CCS02 to CCS00 = xxxB	1	×	—	×	—	—	—	—	—	—	—	✓	✓	✓	✓
	(RxD0)	Input	PIOR1 = 1	—	—	0	CCDE00 = 0	CCS02 to CCS00 = xxxB	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P17	P17	Input	—	×	—	0	CCDE01 = 0	CCS02 to CCS00 = xxxB	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	—	0	CCDE01 = 0	CCS02 to CCS00 = xxxB	0	0/1	(TxD0) = 1 (SO00) = 1 **Note**	TO02 = 0											
		N-ch open-drain output	—	1	—	0	CCDE01 = 0	CCS02 to CCS00 = xxxB	0	0/1													
	CCD01	Output	—	0/1	—	0	CCDE01 = 1	CCS02 to CCS00 = 001B to 011B	0	0	×	TO02 = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	EO17	Output	—	0/1	—	1	CCDE01 = 0	CCS02 to CCS00 = xxxB	0	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TI02	Input	PIOR0 = 0	×	—	0	CCDE01 = 0	CCS02 to CCS00 = xxxB	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TO02	Output	PIOR0 = 0	0	—	0	CCDE01 = 0	CCS02 to CCS00 = xxxB	0	0	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(TxD0)	Output	PIOR1 = 1	0/1	—	0	CCDE01 = 0	CCS02 to CCS00 = xxxB	0	1	×	TO02 = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	(SO00)	Output	PIOR1 = 1	0/1	—	0	CCDE01 = 0	CCS02 to CCS00 = xxxB	0	1	×	TO02 = 0	—	—	—	—	—	—	—	✓	✓	✓	✓

Note This setting is only applicable in the R7F100G products with 64 to 128 pins.

Table 4 - 9 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F100G Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin R7F100G Products with 128-Kbyte Flash Memory (6/22)

Pin Name	Function Used		PMCAxx	PMCTxx	PMxx	Pxx	30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin	80-pin	100-pin	128-pin
	Function Name	I/O															
P20	P20	Input	0	—	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	0	—	0	0/1											
	ANI0	Analog input	1	—	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	AVREFP	Reference voltage	1	—	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	EI20	Input	0	—	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P21	P21	Input	0	—	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	0	—	0	0/1											
	ANI1	Analog input	1	—	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	AVREFM	Reference voltage	1	—	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	EI21	Input	0	—	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P22	P22	Input	0	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	0	0	0	0/1											
	ANI2	Analog input	1	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	ANO0	Analog output	1	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TS20	I/O	0	1	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	EI22	Input	0	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P23	P23	Input	0	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	0	0	0	0/1											
	ANI3	Analog input	1	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	ANO1	Analog output	1	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	IVREF0	Analog input	1	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TS21	I/O	0	1	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	EI23	Input	0	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P24	P24	Input	0	0	1	×	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	0	0	0	0/1											
	ANI4	Analog input	1	0	1	×	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TS22	I/O	0	1	1	×	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
P25	P25	Input	0	0	1	×	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	0	0	0	0/1											
	ANI5	Analog input	1	0	1	×	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TS23	I/O	0	1	1	×	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
P26	P26	Input	0	0	1	×	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓
		Output	0	0	0	0/1											
	ANI6	Analog input	1	0	1	×	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓
	TS24	I/O	0	1	1	×	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓
P27	P27	Input	0	0	1	×	—	—	—	—	✓	✓	✓	✓	✓	✓	✓
		Output	0	0	0	0/1											
	ANI7	Analog input	1	0	1	×	—	—	—	—	✓	✓	✓	✓	✓	✓	✓
	TS25	I/O	0	1	1	×	—	—	—	—	✓	✓	✓	✓	✓	✓	✓

Table 4 - 9 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F100G Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin R7F100G Products with 128-Kbyte Flash Memory (7/22)

Pin Name	Function Used		PIOR	POMxx	PMCAxx	PMCTxx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin	80-pin	100-pin	128-pin
	Function Name	I/O							SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock											
P30	P30	Input	—	—	—	0	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	—	—	0	0	0/1	SCK11/ SCL11 = 1 **Note 1**	VCOUT0 = 0 RTC1HZ = 0											
	TSCAP	—	—	—	—	1	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	EI30	Input	—	—	—	0	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	VCOUT0	Output	—	—	—	0	0	0	×	RTC1HZ = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	INTP3	Input	PIOR5 = 0 **Note 2**	—	—	0	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	RTC1HZ	Output	—	—	—	0	0	0	×	VCOUT0 = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SCK11	Input	—	—	—	0	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
		Output	—	—	—	0	0	1	×	VCOUT0 = 0 RTC1HZ = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
	SCL11	Output	—	—	—	0	0	1	×	VCOUT0 = 0 RTC1HZ = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
P31	P31	Input	—	—	—	0	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	—	—	0	0	0/1	—	TO03 = 0 PCLBUZ0 = 0**Note 3** (PCLBUZ0) = 0**Note 4**											
	TS01	I/O	—	—	—	1	1	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	EI31	Input	—	—	—	0	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TI03	Input	PIOR0 = 0	—	—	0	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TO03	Output	PIOR0 = 0	—	—	0	0	0	—	PCLBUZ0 = 0**Note 3** (PCLBUZ0) = 0**Note 4**	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	INTP4	Input	PIOR5 = 0 **Note 2**	—	—	0	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	PCLBUZ0	Output	—	—	—	0	0	0	—	TO03 = 0	✓	✓	✓	✓	✓	—	—	—	—	—	—
	(PCLBUZ0)	Output	PIOR3 = 1	—	—	0	0	0	—	TO03 = 0	—	—	—	—	—	✓	✓	✓	✓	✓	✓
P32	P32	Input	—	—	—	—	1	×	—	×	—	—	—	—	—	—	—	—	—	—	✓
		Output	—	—	—	—	0	0/1	—	CLKA1 = 0											
	CLKA1	Output	—	—	—	—	0	0	—	×	—	—	—	—	—	—	—	—	—	—	✓
P33	P33	Input	—	—	—	—	1	×	—	—	—	—	—	—	—	—	—	—	—	—	✓
		Output	—	—	—	—	0	0/1	—	—											
	RxDA1	Input	—	—	—	—	1	×	—	—	—	—	—	—	—	—	—	—	—	—	✓
P34	P34	Input	—	×	—	—	1	×	×	—	—	—	—	—	—	—	—	—	—	—	✓
		Output	—	0	—	—	0	0/1	TxDA1 = 1	—											
		N-ch open-drain output	—	1	—	—	0	0/1													
	TxDA1	Output	—	0/1	—	—	0	1	×	—	—	—	—	—	—	—	—	—	—	—	✓
P35 to P37	P35 to P37	Input	—	—	0	—	1	—	—	—	—	—	—	—	—	—	—	—	—	—	✓
		Output	—	—	0	—	0	0/1	—	—											
	ANI23 to ANI21	Analog input	—	—	1	—	1	×	—	—	—	—	—	—	—	—	—	—	—	—	✓

Note 1. This setting is only applicable in the R7F100G products with 30 to 100 pins.

Note 2. This setting is only applicable in the R7F100G products with 100 and 128 pins.

Note 3. This setting is only applicable in the R7F100G products with 30 to 44 pins.

Note 4. This setting is only applicable in the R7F100G products with 48 to 128 pins.

Table 4 - 9 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F100G Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin R7F100G Products with 128-Kbyte Flash Memory (8/22)

Pin Name	Function Used		PIOR	POMxx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin	80-pin	100-pin	128-pin
	Function Name	I/O					SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock											
P40	P40	Input	—	—	1	×	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	—	0	0/1	—	—											
P41	P41	Input	—	—	1	×	—	×	—	—	—	—	✓	✓	✓	✓	✓	✓	✓
		Output	—	—	0	0/1	—	TO07 = 0 Note 1											
	RxDA1	Input	—	—	1	×	—	×	—	—	—	—	✓	✓	✓	✓	✓	✓	—
	TI07	Input	PIOR0 = 0	—	1	×	—	×	—	—	—	—	✓	✓	✓	✓	✓	—	—
	TO07	Output	PIOR0 = 0	—	0	0	—	×	—	—	—	—	✓	✓	✓	✓	✓	—	—
P42	P42	Input	—	×	1	×	—	×	—	—	—	—	—	—	—	✓	✓	✓	✓
		Output	—	0	0	0/1	TxDA1 = 1 Note 2	TO04 = 0											
		N-ch open-drain output	—	1	0	0/1													
	TxDA1	Output	—	0/1	0	1	×	TO04 = 0	—	—	—	—	—	—	—	✓	✓	✓	—
	TI04	Input	PIOR0 = 0	×	1	×	—	×	—	—	—	—	—	—	—	✓	✓	✓	✓
	TO04	Output	PIOR0 = 0	0	0	0	×	×	—	—	—	—	—	—	—	✓	✓	✓	✓
P43	P43	Input	—	×	1	×	×	×	—	—	—	—	—	—	—	✓	✓	✓	✓
		Output	—	0	0	0/1	SCK01/ SCL01 = 1 Note 3	CLKA1 = 0 Note 2											
		N-ch open-drain output	—	1	0	0/1													
	SCK01	Input	—	×	1	×	×	×	—	—	—	—	—	—	—	—	✓	✓	✓
		Output	—	0/1	0	1	×	CLKA1 = 0 Note 4	—	—	—	—	—	—	—	—	✓	✓	✓
	SCL01	Output	—	0/1	0	1	×	CLKA1 = 0 Note 4	—	—	—	—	—	—	—	—	✓	✓	✓
	CLKA1	Output	—	0	0	0	×	×	—	—	—	—	—	—	—	✓	✓	✓	—
P44	P44	Input	—	×	1	×	×	—	—	—	—	—	—	—	—	—	✓	✓	✓
		Output	—	0	0	0/1	SDA01 = 1	—											
		N-ch open-drain output	—	1	0	0/1													
	SI01	Input	—	×	1	×	×	—	—	—	—	—	—	—	—	—	✓	✓	✓
	SDA01	I/O	—	1	0	1	×	—	—	—	—	—	—	—	—	—	✓	✓	✓
P45	P45	Input	—	×	1	×	×	—	—	—	—	—	—	—	—	—	✓	✓	✓
		Output	—	0	0	0/1	SO01 = 1	—											
		N-ch open-drain output	—	1	0	0/1													
	SO01	Output	—	0/1	0	1	×	—	—	—	—	—	—	—	—	—	✓	✓	✓
P46	P46	Input	—	—	1	×	—	×	—	—	—	—	—	—	—	—	—	✓	✓
		Output	—	—	0	0/1	—	TO05 = 0											
	INTP1	Input	PIOR5 = 0	—	1	×	—	×	—	—	—	—	—	—	—	—	—	✓	✓
	TI05	Input	PIOR0 = 0	—	1	×	—	×	—	—	—	—	—	—	—	—	—	✓	✓
	TO05	Output	PIOR0 = 0	—	0	0	—	×	—	—	—	—	—	—	—	—	—	✓	✓
P47	P47	Input	—	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓	✓
		Output	—	—	0	0/1	—	—											
	INTP2	Input	—	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓	✓

Note 1. This setting is only applicable in the R7F100G products with 44 to 80 pins.

Note 2. This setting is only applicable in the R7F100G products with 64 to 100 pins.

Note 3. This setting is only applicable in the R7F100G products with 80 to 128 pins.

Note 4. This setting is only applicable in the R7F100G products with 80 and 100 pins.

Table 4 - 9 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F100G Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin R7F100G Products with 128-Kbyte Flash Memory (9/22)

Pin Name	Function Used		PIOR	POMxx	PMCTxx	PMCExx	CCDE	CCSx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin	80-pin	100-pin	128-pin
	Function Name	I/O									SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock											
P50	P50	Input	—	×	0	0	CCDE03 = 0	CCS02 to CCS00 = xxxB	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0	CCDE03 = 0	CCS02 to CCS00 = xxxB	0	0/1	SDA11 = 1 **Note**	—											
		N-ch open-drain output	—	1	0	0	CCDE03 = 0	CCS02 to CCS00 = xxxB	0	0/1													
	TS00	I/O	—	×	1	0	CCDE03 = 0	CCS02 to CCS00 = xxxB	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	EI50	Input	—	×	0	0	CCDE03 = 0	CCS02 to CCS00 = xxxB	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	EO50	Output	—	0/1	0	1	CCDE03 = 0	CCS02 to CCS00 = xxxB	0	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	CCD03	Output	—	0/1	0	0	CCDE03 = 1	CCS02 to CCS00 = 001B to 011B	0	0	×	—	✓	✓	✓	✓	—	✓	✓	✓	✓	✓	✓
	INTP1	Input	—	×	0	0	CCDE03 = 0	CCS02 to CCS00 = xxxB	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
	SI11	Input	—	×	0	0	CCDE03 = 0	CCS02 to CCS00 = xxxB	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
	SDA11	I/O	—	1	0	0	CCDE03 = 0	CCS02 to CCS00 = xxxB	0	1	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
P51	P51	Input	—	—	—	0	CCDE02 = 0	CCS02 to CCS00 = xxxB	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	—	—	0	CCDE02 = 0	CCS02 to CCS00 = xxxB	0	0/1	SO11 = 1 **Note**	—											
	EI51	Input	—	—	—	0	CCDE02 = 0	CCS02 to CCS00 = xxxB	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	EO51	Output	—	—	—	1	CCDE02 = 0	CCS02 to CCS00 = xxxB	0	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	CCD02	Output	—	—	—	0	CCDE02 = 1	CCS02 to CCS00 = 001B to 011B	0	0	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	INTP2	Input	—	—	—	0	CCDE02 = 0	CCS02 to CCS00 = xxxB	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
	SO11	Output	—	—	—	0	CCDE02 = 0	CCS02 to CCS00 = xxxB	0	1	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—

Note This setting is only applicable in the R7F100G products with 30 to 100 pins.

Table 4 - 9 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F100G Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin R7F100G Products with 128-Kbyte Flash Memory (10/22)

Pin Name	Function Used		PIOR	POMxx	PMCTxx	PMCExx	CCDE	CCSx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin	80-pin	100-pin	128-pin
	Function Name	I/O									SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock											
P52	P52	Input	—	×	—	—	—	—	1	×	×	—	—	—	—	—	—	—	—	✓	✓	✓	✓
		Output	—	0	—	—	—	—	0	0/1	SO31 = 1 **Note**	—											
		N-ch open-drain output	—	1	—	—	—	—	0	0/1										—			
	SO31	Output	—	0/1	—	—	—	—	0	1	×	—	—	—	—	—	—	—	—	—	✓	✓	✓
	(INTP10)	Input	PIOR1 = 1	×	—	—	—	—	1	×	×	—	—	—	—	—	—	—	—	✓	—	—	—
P53	P53	Input	—	×	—	—	—	—	1	×	×	—	—	—	—	—	—	—	—	✓	✓	✓	✓
		Output	—	0	—	—	—	—	0	0/1	SDA31 = 1 **Note**	—											
		N-ch open-drain output	—	1	—	—	—	—	0	0/1										—			
	SI31	Input	—	×	—	—	—	—	1	×	×	—	—	—	—	—	—	—	—	—	✓	✓	✓
	SDA31	I/O	—	1	—	—	—	—	0	1	×	—	—	—	—	—	—	—	—	—	✓	✓	✓
	(INTP11)	Input	PIOR1 = 1	×	—	—	—	—	1	×	×	—	—	—	—	—	—	—	—	✓	—	—	—
P54	P54	Input	—	×	—	—	—	—	1	×	×	—	—	—	—	—	—	—	—	✓	✓	✓	✓
		Output	—	0	—	—	—	—	0	0/1	SCK31/ SCL1 = 1 **Note**	—											
		N-ch open-drain output	—	1	—	—	—	—	0	0/1										—			
	SCK31	Input	—	×	—	—	—	—	1	×	×	—	—	—	—	—	—	—	—	—	✓	✓	✓
		Output	—	0/1	—	—	—	—	0	1	×	—	—	—	—	—	—	—	—	—	✓	✓	✓
	SCL31	Output	—	0/1	—	—	—	—	0	1	×	—	—	—	—	—	—	—	—	—	✓	✓	✓
P55	P55	Input	—	×	—	—	—	—	1	×	×	×	—	—	—	—	—	—	—	✓	✓	✓	✓
		Output	—	0	—	—	—	—	0	0/1	(SCK00) = 1	(PCLBUZ1) = 0											
		N-ch open-drain output	—	1	—	—	—	—	0	0/1													
	(PCLBUZ1)	Output	PIOR4 = 1	0	—	—	—	—	0	0	×	×	—	—	—	—	—	—	—	✓	✓	✓	✓
	(SCK00)	Input	PIOR1 = 1	×	—	—	—	—	1	×	×	×	—	—	—	—	—	—	—	✓	✓	✓	✓
		Output	PIOR1 = 1	0/1	—	—	—	—	0	1	×	(PCLBUZ1) = 0	—	—	—	—	—	—	—	✓	✓	✓	✓
P56	P56	Input	—	—	—	—	—	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓	✓
		Output	—	—	—	—	—	—	0	0/1	—	—											
	(INTP1)	Input	PIOR5 = 1	—	—	—	—	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓	✓
P57	P57	Input	—	—	—	—	—	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓	✓
		Output	—	—	—	—	—	—	0	0/1	—	—											
	(INTP3)	Input	PIOR5 = 1	—	—	—	—	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓	✓

Note This setting is only applicable in the R7F100G products with 80 to 128 pins.

Table 4 - 9 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F100G Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin R7F100G Products with 128-Kbyte Flash Memory (11/22)

Pin Name	Function Used		PIOR	PMCTxx	PMCExx	CCDE	CCSx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin	80-pin	100-pin	128-pin
	Function Name	I/O								SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock											
P60	P60	Input		—	0	CCDE04 = 0	CCS42 to CCS40 = 000B	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		N-ch open-drain output (withstand voltage of 6 V)		—	0	CCDE04 = 0	CCS42 to CCS40 = 000B	0	0/1	—	SCLA0 = 0											
	EO60	Output	—	—	1	CCDE04 = 0	CCS42 to CCS40 = 000B	0	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	CCD04	Output	—	—	0	CCDE04 = 1	CCS42 to CCS40 = 001B to 100B	0	0	—	SCLA0 = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SCLA0	I/O	PIOR2 = 0	—	0	CCDE04 = 0	CCS42 to CCS40 = 000B	0	0	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P61	P61	Input		—	0	CCDE05 = 0	CCS52 to CCS50 = 000B	1	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		N-ch open-drain output (withstand voltage of 6 V)		—	0	CCDE05 = 0	CCS52 to CCS50 = 000B	0	0/1	—	SDAA0 = 0											
	EO61	Output	—	—	1	CCDE05 = 0	CCS52 to CCS50 = 000B	0	×	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	CCD05	Output	—	—	0	CCDE05 = 1	CCS52 to CCS50 = 001B to 100B	0	0	—	SDAA0 = 0	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SDAA0	I/O	PIOR2 = 0	—	0	CCDE05 = 0	CCS52 to CCS50 = 000B	0	0	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
P62	P62	Input	—	—	—	CCDE06 = 0	CCS62 to CCS60 = 000B	1	×	—	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		N-ch open-drain output (withstand voltage of 6 V)	—	—	—	CCDE06 = 0	CCS62 to CCS60 = 000B	0	0/1	—	SCLA1 = 0**Note**											
	CCD06	Output	—	—	—	CCDE06 = 1	CCS62 to CCS60 = 001B to 100B	0	0	—	SCLA1 = 0**Note**	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SCLA1	I/O	—	—	—	CCDE06 = 0	CCS62 to CCS60 = 000B	0	0	—	×	—	—	—	—	✓	✓	✓	✓	✓	✓	✓
P63	P63	Input	—	—	—	CCDE07 = 0	CCS72 to CCS70 = 000B	1	×	—	×	—	—	—	—	✓	✓	✓	✓	✓	✓	✓
		N-ch open-drain output (withstand voltage of 6 V)	—	—	—	CCDE07 = 0	CCS72 to CCS70 = 000B	0	0/1	—	SDAA1 = 0											
	CCD07	Output	—	—	—	CCDE07 = 1	CCS72 to CCS70 = 001B to 100B	0	0	—	SDAA1 = 0	—	—	—	—	✓	✓	✓	✓	✓	✓	✓
	SDAA1	I/O	—	—	—	CCDE07 = 0	CCS72 to CCS70 = 000B	0	0	—	×	—	—	—	—	✓	✓	✓	✓	✓	✓	✓

Note This setting is only applicable in the R7F100G products with 44 to 128 pins.

Table 4 - 9 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F100G Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin R7F100G Products with 128-Kbyte Flash Memory (12/22)

Pin Name	Function Used		PIOR	PMCTxx	PMCExx	CCDE	CCSx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin	80-pin	100-pin	128-pin
	Function Name	I/O								SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock											
P64	P64	Input	—	0	—	—	—	1	×	—	×	—	—	—	—	—	—	—	—	✓	✓	✓
		Output	—	0	—	—	—	0	0/1	—	TO10 = 0											
	TS12	I/O	—	1	—	—	—	1	×	—	×	—	—	—	—	—	—	—	—	✓	✓	✓
	TI10	Input	—	0	—	—	—	1	×	—	×	—	—	—	—	—	—	—	—	✓	✓	✓
	TO10	Output	—	0	—	—	—	0	0	—	×	—	—	—	—	—	—	—	—	✓	✓	✓
P65	P65	Input	—	0	—	—	—	1	×	—	×	—	—	—	—	—	—	—	—	✓	✓	✓
		Output	—	0	—	—	—	0	0/1	—	TO11 = 0											
	TS13	I/O	—	1	—	—	—	1	×	—	×	—	—	—	—	—	—	—	—	✓	✓	✓
	TI11	Input	—	0	—	—	—	1	×	—	×	—	—	—	—	—	—	—	—	✓	✓	✓
	TO11	Output	—	0	—	—	—	0	0	—	×	—	—	—	—	—	—	—	—	✓	✓	✓
P66	P66	Input	—	0	—	—	—	1	×	—	×	—	—	—	—	—	—	—	—	✓	✓	✓
		Output	—	0	—	—	—	0	0/1	—	TO12 = 0											
	TS14	I/O	—	1	—	—	—	1	×	—	×	—	—	—	—	—	—	—	—	✓	✓	✓
	TI12	Input	—	0	—	—	—	1	×	—	×	—	—	—	—	—	—	—	—	✓	✓	✓
	TO12	Output	—	0	—	—	—	0	0	—	×	—	—	—	—	—	—	—	—	✓	✓	✓
P67	P67	Input	—	0	—	—	—	1	×	—	×	—	—	—	—	—	—	—	—	✓	✓	✓
		Output	—	0	—	—	—	0	0/1	—	TO13 = 0											
	TS15	I/O	—	1	—	—	—	1	×	—	×	—	—	—	—	—	—	—	—	✓	✓	✓
	TI13	Input	—	0	—	—	—	1	×	—	×	—	—	—	—	—	—	—	—	✓	✓	✓
	TO13	Output	—	0	—	—	—	0	0	—	×	—	—	—	—	—	—	—	—	✓	✓	✓

Table 4 - 9 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F100G Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin R7F100G Products with 128-Kbyte Flash Memory (13/22)

Pin Name	Function Used		PIOR	POMxx	PMCTxx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin	80-pin	100-pin	128-pin
	Function Name	I/O						SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock											
P70	P70	Input	—	—	0	1	×	×	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	—	0	0	0/1	SCK21/SCL21 = 1 Note 1	—											
	TS02	I/O	—	—	1	1	×	×	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	RIN0	Input	—	—	0	1	×	×	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	KR0	Input	—	—	0	1	×	×	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓
	SCK21	Input	—	—	0	1	×	×	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	—	0	0	1	×	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SCL21	Output	—	—	0	0	1	×	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
P71	P71	Input	—	×	0	1	×	×	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0	0/1	SDA21 = 1	—											
		N-ch open-drain output	—	1	0	0	0/1													
	TS03	I/O	—	×	1	1	×	×	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	KR1	Input	—	×	0	1	×	×	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓
	SI21	Input	—	×	0	1	×	×	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	SDA21	I/O	—	1	0	0	1	×	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	RxDA0	Input	—	×	0	1	×	×	—	—	—	✓	✓	✓	✓	✓	✓	✓	—	—
P72	P72	Input	—	×	0	1	×	×	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0	0/1	SO21 = 1 TxDA0 = 1	—											
		N-ch open-drain output	—	1	0	0	0/1													
	TS04	I/O	—	×	1	1	×	×	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	KR2	Input	—	×	0	1	×	×	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓
	SO21	Output	—	0/1	0	0	1	TxDA0 = 1	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TxDA0	Output	—	0/1	0	0	1	SO21 = 1	—	—	—	✓	✓	✓	✓	✓	✓	✓	—	—
P73	P73	Input	—	—	0	1	×	×	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	—	0	0	0/1	SO01 = 1 Note 2	—											
	TS05	I/O	—	—	1	1	×	×	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓
	KR3	Input	—	—	0	1	×	×	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓
	SO01	Output	—	—	0	0	1	×	—	—	—	—	—	—	✓	✓	✓	—	—	—

Note 1. This setting is only applicable in the R7F100G products with 36 to 128 pins.

Note 2. This setting is only applicable in the R7F100G products with 48 to 64 pins.

Table 4 - 9 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F100G Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin R7F100G Products with 128-Kbyte Flash Memory (14/22)

Pin Name	Function Used		PIOR	POMxx	PMCTxx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin	80-pin	100-pin	128-pin
	Function Name	I/O						SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock											
P74	P74	Input	—	×	0	1	×	×	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0	0/1	SDA01 = 1 Note 1	—											
		N-ch open-drain output	—	1	0	0	0/1													
	TS06	I/O	—	×	1	1	×	×	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓
	KR4	Input	—	×	0	1	×	×	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓
	INTP8	Input	PIOR5 = 0 Note 2	×	0	1	×	×	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓
	SI01	Input	—	×	0	1	×	×	—	—	—	—	—	—	✓	✓	✓	—	—	—
	SDA01	I/O	—	1	0	0	1	×	—	—	—	—	—	—	✓	✓	✓	—	—	—
P75	P75	Input	—	—	0	1	×	×	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓
		Output	—	—	0	0	0/1	SCK01/SCL01 = 1 Note 1	—											
	TS07	I/O	—	—	1	1	×	×	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓
	KR5	Input	—	—	0	1	×	×	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓
	INTP9	Input	PIOR5 = 0 Note 2	—	0	1	×	×	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓
	SCK01	Input	—	—	0	1	×	×	—	—	—	—	—	—	✓	✓	✓	—	—	—
		Output	—	—	0	0	1	×	—	—	—	—	—	—	✓	✓	✓	—	—	—
	SCL01	Output	—	—	0	0	1	×	—	—	—	—	—	—	✓	✓	✓	—	—	—
P76	P76	Input	—	—	0	1	×	—	—	—	—	—	—	—	—	✓	✓	✓	✓	✓
		Output	—	—	0	0	0/1	—	—											
	TS08	I/O	—	—	1	1	×	—	—	—	—	—	—	—	—	✓	✓	✓	✓	✓
	KR6	Input	—	—	0	1	×	—	—	—	—	—	—	—	—	✓	✓	✓	✓	✓
	INTP10	Input	PIOR1 = 0	—	0	1	×	—	—	—	—	—	—	—	—	✓	✓	✓	✓	✓
	(RxD2)	Input	PIOR1 = 1	—	0	1	×	—	—	—	—	—	—	—	—	✓	✓	✓	✓	✓
P77	P77	Input	—	—	0	1	×	×	—	—	—	—	—	—	—	✓	✓	✓	✓	✓
		Output	—	—	0	0	0/1	(TxD2) = 1	—											
	TS09	I/O	—	—	1	1	×	×	—	—	—	—	—	—	—	✓	✓	✓	✓	✓
	KR7	Input	—	—	0	1	×	×	—	—	—	—	—	—	—	✓	✓	✓	✓	✓
	INTP11	Input	PIOR1 = 0	—	0	1	×	×	—	—	—	—	—	—	—	✓	✓	✓	✓	✓
	(TxD2)	Output	PIOR1 = 1	—	0	0	1	×	—	—	—	—	—	—	—	✓	✓	✓	✓	✓

Note 1. This setting is only applicable in the R7F100G products with 48 to 64 pins.

Note 2. This setting is only applicable in the R7F100G products with 100 and 128 pins.

Table 4 - 9 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F100G Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin R7F100G Products with 128-Kbyte Flash Memory (15/22)

Pin Name	Function Used		PIOR	POMxx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin	80-pin	100-pin	128-pin
	Function Name	I/O					SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock											
P80	P80	Input	—	×	1	×	×	—	—	—	—	—	—	—	—	—	—	✓	✓
		Output	—	0	0	0/1	(SCK10)/ (SCL10) = 1	—											
		N-ch open-drain output	—	1	0	0/1													
	(SCK10)	Input	PIOR5 = 1	×	1	×	×	—	—	—	—	—	—	—	—	—	—	✓	✓
		Output	PIOR5 = 1	0/1	0	1	×	—	—	—	—	—	—	—	—	—	—	✓	✓
	(SCL10)	Output	PIOR5 = 1	0/1	0	1	×	—	—	—	—	—	—	—	—	—	—	✓	✓
P81	P81	Input	—	×	1	×	×	—	—	—	—	—	—	—	—	—	—	✓	✓
		Output	—	0	0	0/1	(SDA10) = 1	—											
		N-ch open-drain output	—	1	0	0/1													
	(SI10)	Input	PIOR5 = 1	×	1	×	×	—	—	—	—	—	—	—	—	—	—	✓	✓
	(RxD1)	Input	PIOR5 = 1	×	1	×	×	—	—	—	—	—	—	—	—	—	—	✓	✓
	(SDA10)	I/O	PIOR5 = 1	1	0	1	×	—	—	—	—	—	—	—	—	—	—	✓	✓
P82	P82	Input	—	×	1	×	×	—	—	—	—	—	—	—	—	—	—	✓	✓
		Output	—	0	0	0/1	(SO10)/ (TxD1) = 1	—											
		N-ch open-drain output	—	1	0	0/1													
	(SO10)	Output	PIOR5 = 1	0/1	0	1	×	—	—	—	—	—	—	—	—	—	—	✓	✓
	(TxD1)	Output	PIOR5 = 1	0/1	0	1	×	—	—	—	—	—	—	—	—	—	—	✓	✓
P83	P83	Input	—	×	1	×	×	—	—	—	—	—	—	—	—	—	—	✓	✓
		Output	—	0	0	0/1	TxDA0 = 1	—											
		N-ch open-drain output	—	1	0	0/1													
	TxDA0	Output	—	0/1	0	1	×	—	—	—	—	—	—	—	—	—	—	✓	✓
P84	P84	Input	—	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓	✓
		Output	—	—	0	0/1	—	—											
	RxDA0	Input	—	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓	✓
	(INTP6)	Input	PIOR5 = 1	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓	✓
P85	P85	Input	—	—	1	×	—	×	—	—	—	—	—	—	—	—	—	✓	✓
		Output	—	—	0	0/1	—	CLKA0 = 0											
	CLKA0	Output	—	—	0	0	—	×	—	—	—	—	—	—	—	—	—	✓	✓
	(INTP7)	Input	PIOR5 = 1	—	1	×	—	×	—	—	—	—	—	—	—	—	—	✓	✓
P86	P86	Input	—	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓	✓
		Output	—	—	0	0/1	—	—											
	(INTP8)	Input	PIOR5 = 1	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓	✓
P87	P87	Input	—	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓	✓
		Output	—	—	0	0/1	—	—											
	(INTP9)	Input	PIOR5 = 1	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓	✓

Table 4 - 9 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F100G Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin R7F100G Products with 128-Kbyte Flash Memory (16/22)

Pin Name	Function Used		PIOR	POMxx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin	80-pin	100-pin	128-pin
	Function Name	I/O					SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock											
P90 to P94	P90 to P94	Input	—	—	1	×	—	—	—	—	—	—	—	—	—	—	—	—	✓
		Output	—	—	0	0/1	—	—											
P95	P95	Input	—	—	1	×	×	—	—	—	—	—	—	—	—	—	—	—	✓
		Output	—	—	0	0/1	SCK11/ SCL11 = 1	—											
	SCK11	Input	—	—	1	×	×	—	—	—	—	—	—	—	—	—	—	—	✓
		Output	—	—	0	1	×	—	—	—	—	—	—	—	—	—	—	—	✓
	SCL11	Output	—	—	0	1	×	—	—	—	—	—	—	—	—	—	—	—	✓
P96	P96	Input	—	×	1	×	×	—	—	—	—	—	—	—	—	—	—	—	✓
		Output	—	0	0	0/1	SDA11 = 1	—											
		N-ch open-drain output	—	1	0	0/1													
	SI11	Input	—	×	1	×	×	—	—	—	—	—	—	—	—	—	—	—	✓
	SDA11	I/O	—	1	0	1	×	—	—	—	—	—	—	—	—	—	—	—	✓
P97	P97	Input	—	—	1	×	×	—	—	—	—	—	—	—	—	—	—	—	✓
		Output	—	—	0	0/1	SO11 = 1	—											
	SO11	Output	—	—	0	1	×	—	—	—	—	—	—	—	—	—	—	—	✓

Table 4 - 9 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F100G Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin R7F100G Products with 128-Kbyte Flash Memory (17/22)

Pin Name	Function Used		PIOR	PMCAxx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin	80-pin	100-pin	128-pin
	Function Name	I/O					SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock											
P100	P100	Input	—	0	1	×	—	—	—	—	—	—	—	—	—	—	✓	✓	✓
		Output	—	0	0	0/1	—	—											
	ANI20	Analog input	—	1	1	×	—	—	—	—	—	—	—	—	—	—	✓	✓	✓
P101	P101	Input	—	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓	✓
		Output	—	—	0	0/1	—	—											
P102	P102	Input	—	—	1	×	—	×	—	—	—	—	—	—	—	—	—	✓	✓
		Output	—	—	0	0/1	—	TO06 = 0											
	TI06	Input	PIOR0 = 0	—	1	×	—	×	—	—	—	—	—	—	—	—	—	✓	✓
	TO06	Output	PIOR0 = 0	—	0	0	—	×	—	—	—	—	—	—	—	—	—	✓	✓
P103	P103	Input	—	—	1	×	—	×	—	—	—	—	—	—	—	—	—	—	✓
		Output	—	—	0	0/1	—	TO14 = 0											
	TI14	Input	—	—	1	×	—	×	—	—	—	—	—	—	—	—	—	—	✓
	TO14	Output	—	—	0	0	—	×	—	—	—	—	—	—	—	—	—	—	✓
P104	P104	Input	—	—	1	×	—	×	—	—	—	—	—	—	—	—	—	—	✓
		Output	—	—	0	0/1	—	TO15 = 0											
	TI15	Input	—	—	1	×	—	×	—	—	—	—	—	—	—	—	—	—	✓
	TO15	Output	—	—	0	0	—	×	—	—	—	—	—	—	—	—	—	—	✓
P105	P105	Input	—	—	1	×	—	×	—	—	—	—	—	—	—	—	—	—	✓
		Output	—	—	0	0/1	—	TO16 = 0											
	TI16	Input	—	—	1	×	—	×	—	—	—	—	—	—	—	—	—	—	✓
	TO16	Output	—	—	0	0	—	×	—	—	—	—	—	—	—	—	—	—	✓
P106	P106	Input	—	—	1	×	—	×	—	—	—	—	—	—	—	—	—	—	✓
		Output	—	—	0	0/1	—	TO17 = 0											
	TI17	Input	—	—	1	×	—	×	—	—	—	—	—	—	—	—	—	—	✓
	TO17	Output	—	—	0	0	—	×	—	—	—	—	—	—	—	—	—	—	✓

Table 4 - 9 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F100G Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin R7F100G Products with 128-Kbyte Flash Memory (18/22)

Pin Name	Function Used		PIOR	POMxx	PMCAxx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin	80-pin	100-pin	128-pin
	Function Name	I/O						SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock											
P110	P110	Input	—	—	—	1	×	—	—	—	—	—	—	—	—	—	—	✓	✓	✓
		Output	—	—	—	0	0/1	—	—											
	(INTP10)	Input	PIOR1 = 1	—	—	1	×	—	—	—	—	—	—	—	—	—	—	✓	✓	✓
P111	P111	Input	—	—	—	1	×	—	—	—	—	—	—	—	—	—	—	✓	✓	✓
		Output	—	—	—	0	0/1	—	—											
	(INTP11)	Input	PIOR1 = 1	—	—	1	×	—	—	—	—	—	—	—	—	—	—	✓	✓	✓
P112 to P114	P112 to P114	Input	—	—	—	1	×	—	—	—	—	—	—	—	—	—	—	—	—	✓
		Output	—	—	—	0	0/1	—	—											
P115 to P117	P115 to P117	Input	—	—	0	1	×	—	—	—	—	—	—	—	—	—	—	—	—	✓
		Output	—	—	0	0	0/1	—	—											
	ANI24 to ANI26	Analog input	—	—	1	1	×	—	—	—	—	—	—	—	—	—	—	—	—	✓
P120	P120	Input	—	×	0	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	0	0	0	0/1	TxDA1 = 1[Note]	—											
		N-ch open-drain output	—	1	0	0	0/1													
	ANI19	Analog input	—	×	1	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	IVCMP1	Analog input	—	×	1	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	EI120	Input	—	×	0	1	×	×	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	TxDA1	Output	—	0/1	0	0	1	×	—	—	—	—	—	✓	✓	✓	—	—	—	—

Note This setting is only applicable in the R7F100G products with 44 to 52 pins.

Table 4 - 9 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F100G Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin R7F100G Products with 128-Kbyte Flash Memory (19/22)

Pin Name	Function Used		CMC		PMxx	Pxx	30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin	80-pin	100-pin	128-pin
	Function Name	I/O	EXCLK, OSCSEL, EXCLKS, OSCSELS	XTSEL													
P121	P121	Input	00xxB/10xxB/11xxB	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
			xx00B/xx10B/xx11B	1Note													
		Output	00xxB/10xxB/11xxB	0	0	0/1											
			xx00B/xx10B/xx11B	1Note													
	EI121	Input	00xxB/10xxB/11xxB	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
			xx00B/xx10B/xx11B	1Note													
	VBAT	—	01xxB	0	1	×	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓
	X1	—	01xxB	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	XT1	—	xx01B	1	1	×	✓	✓	✓	—	—	—	—	—	—	—	—
P122	P122	Input	00xxB/10xxB	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
			xx00B/xx10B	1Note													
		Output	00xxB/10xxB	0	0	0/1											
			xx00B/xx10B	1Note													
	EI122	Input	00xxB/10xxB	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
			xx00B/xx10B	1Note													
	X2	—	01xxB	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	XT2	—	xx01B	1	1	×	✓	✓	✓	—	—	—	—	—	—	—	—
	EXCLK	Input	11xxB	0	1	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	EXCLKS	Input	xx11B	1	1	×	✓	✓	✓	—	—	—	—	—	—	—	—
P123	P123	Input	xx00B/xx10B/xx11B	0	—	×	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓
	XT1	—	xx01B	0	—	×	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓
P124	P124	Input	xx00B/xx10B	0	—	×	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓
	XT2	—	xx01B	0	—	×	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓
	EXCLKS	Input	xx11B	0	—	×	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓

Note This setting is only applicable in the R7F100G products with 30 to 36 pins.

Table 4 - 9 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F100G Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin R7F100G Products with 128-Kbyte Flash Memory (20/22)

Pin Name	Function Used		PMxx	Pxx	30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin	80-pin	100-pin	128-pin
	Function Name	I/O													
P125 to P127	P125 to P127	Input	1	×	—	—	—	—	—	—	—	—	—	—	✓
		Output	0	0/1											
P130	P130	Output	—	0/1	—	—	—	—	—	✓	✓	✓	✓	✓	✓
P137	P137	Input	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	EI137	Input	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	INTP0	Input	—	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 4 - 9 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F100G Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin R7F100G Products with 128-Kbyte Flash Memory (21/22)

Pin Name	Function Used		PIOR	POMxx	PMCAxx	PMxx	Pxx	Multiplexed Function Output		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin	80-pin	100-pin	128-pin
	Function Name	I/O						SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock											
P140	P140	Input	—	—	—	1	×	—	×	—	—	—	—	—	✓	✓	✓	✓	✓	✓
		Output	—	—	—	0	0/1	—	PCLBUZ0 = 0											
	PCLBUZ0	Output	PIOR3 = 0	—	—	0	0	—	×	—	—	—	—	—	✓	✓	✓	✓	✓	✓
	INTP6	Input	PIOR5 = 0 **Note**	—	—	1	×	—	×	—	—	—	—	—	✓	✓	✓	✓	✓	✓
P141	P141	Input	—	—	—	1	×	—	×	—	—	—	—	—	—	—	✓	✓	✓	✓
		Output	—	—	—	0	0/1	—	PCLBUZ1 = 0											
	PCLBUZ1	Output	PIOR4 = 0	—	—	0	0	—	×	—	—	—	—	—	—	—	✓	✓	✓	✓
	INTP7	Input	PIOR5 = 0 **Note**	—	—	1	×	—	×	—	—	—	—	—	—	—	✓	✓	✓	✓
P142	P142	Input	—	×	—	1	×	×	—	—	—	—	—	—	—	—	—	✓	✓	✓
		Output	—	0	—	0	0/1	SCK30/ SCL30 = 1	—											
		N-ch open-drain output	—	1	—	0	0/1													
	SCK30	Input	—	×	—	1	×	×	—	—	—	—	—	—	—	—	—	✓	✓	✓
		Output	—	0/1	—	0	1	×	—	—	—	—	—	—	—	—	—	✓	✓	✓
	SCL30	Output	—	0/1	—	0	1	×	—	—	—	—	—	—	—	—	—	✓	✓	✓
P143	P143	Input	—	×	—	1	×	×	—	—	—	—	—	—	—	—	—	✓	✓	✓
		Output	—	0	—	0	0/1	SDA30 = 1	—											
		N-ch open-drain output	—	1	—	0	0/1													
	SI30	Input	—	×	—	1	×	×	—	—	—	—	—	—	—	—	—	✓	✓	✓
	RxD3	Input	—	×	—	1	×	×	—	—	—	—	—	—	—	—	—	✓	✓	✓
	SDA30	I/O	—	1	—	0	1	×	—	—	—	—	—	—	—	—	—	✓	✓	✓
P144	P144	Input	—	×	—	1	×	×	—	—	—	—	—	—	—	—	—	✓	✓	✓
		Output	—	0	—	0	0/1	SO30/ TxD3 = 1	—											
		N-ch open-drain output	—	1	—	0	0/1													
	SO30	Output	—	0/1	—	0	1	×	—	—	—	—	—	—	—	—	—	✓	✓	✓
	TxD3	Output	—	0/1	—	0	1	×	—	—	—	—	—	—	—	—	—	✓	✓	✓
P145	P145	Input	—	—	—	1	×	—	×	—	—	—	—	—	—	—	—	—	✓	✓
		Output	—	—	—	0	0/1	—	TO07 = 0											
	TI07	Input	PIOR0 = 0	—	—	1	×	—	×	—	—	—	—	—	—	—	—	—	✓	✓
	TO07	Output	PIOR0 = 0	—	—	0	0	—	×	—	—	—	—	—	—	—	—	—	✓	✓
P146	P146	Input	—	—	—	1	×	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓
		Output	—	—	—	0	0/1	—	—											
	(INTP4)	Input	PIOR5 = 1	—	—	1	×	—	—	—	—	—	—	—	—	—	—	—	✓	✓
P147	P147	Input	—	—	0	1	×	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Output	—	—	0	0	0/1	—	—											
	ANI18	Analog input	—	—	1	1	×	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	IVCMP0	Analog input	—	—	1	1	×	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	EI147	Input	—	—	0	1	×	—	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Note This setting is only applicable in the R7F100G products with 100 and 128 pins.

Table 4 - 9 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F100G Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin R7F100G Products with 128-Kbyte Flash Memory (22/22)

Pin Name	Function Used		PMCAxx	PMCTxx	PMxx	Pxx	30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin	80-pin	100-pin	128-pin
	Function Name	I/O															
P150	P150	Input	0	0	1	×	—	—	—	—	—	—	—	—	✓	✓	✓
		Output	0	0	0	0/1											
	ANI8	Analog input	1	0	1	×	—	—	—	—	—	—	—	—	✓	✓	✓
	TS30	I/O	0	1	1	×	—	—	—	—	—	—	—	—	✓	✓	✓
P151	P151	Input	0	0	1	×	—	—	—	—	—	—	—	—	✓	✓	✓
		Output	0	0	0	0/1											
	ANI9	Analog input	1	0	1	×	—	—	—	—	—	—	—	—	✓	✓	✓
	TS31	I/O	0	1	1	×	—	—	—	—	—	—	—	—	✓	✓	✓
P152	P152	Input	0	0	1	×	—	—	—	—	—	—	—	—	✓	✓	✓
		Output	0	0	0	0/1											
	ANI10	Analog input	1	0	1	×	—	—	—	—	—	—	—	—	✓	✓	✓
	TS32	I/O	0	1	1	×	—	—	—	—	—	—	—	—	✓	✓	✓
P153	P153	Input	0	0	1	×	—	—	—	—	—	—	—	—	✓	✓	✓
		Output	0	0	0	0/1											
	ANI11	Analog input	1	0	1	×	—	—	—	—	—	—	—	—	✓	✓	✓
	TS33	I/O	0	1	1	×	—	—	—	—	—	—	—	—	✓	✓	✓
P154	P154	Input	0	0	1	×	—	—	—	—	—	—	—	—	—	✓	✓
		Output	0	0	0	0/1											
	ANI12	Analog input	1	0	1	×	—	—	—	—	—	—	—	—	—	✓	✓
	TS34	I/O	0	1	1	×	—	—	—	—	—	—	—	—	—	✓	✓
P155	P155	Input	0	0	1	×	—	—	—	—	—	—	—	—	—	✓	✓
		Output	0	0	0	0/1											
	ANI13	Analog input	1	0	1	×	—	—	—	—	—	—	—	—	—	✓	✓
	TS35	I/O	0	1	1	×	—	—	—	—	—	—	—	—	—	✓	✓
P156	P156	Input	0	0	1	×	—	—	—	—	—	—	—	—	—	✓	✓
		Output	0	0	0	0/1											
	ANI14	Analog input	1	0	1	×	—	—	—	—	—	—	—	—	—	✓	✓

Table 4 - 10 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F104G Products (1/14)

Pin Name	Function Used		PIOR	POMxx	PMCAxx	PMCTxx	PMCExx	PMxx	Pxx	Multiplexed Function Output		32-pin	44-pin	48-pin
	Function Name	I/O								SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock			
P00	P00	Input	—	×	—	0Note	—	1	×	×	—	✓	✓	✓
		Output	—	0	—	0Note	—	0	0/1	×	—			
		N-ch open-drain output	—	1	—	0Note	—	0	0/1					
	TS26	I/O	—	×	—	1Note	—	1	×	×	—	×	✓	✓
	EI00	Input	—	×	—	0Note	—	1	×	×	—	✓	✓	✓
	TI00	Input	—	×	—	0Note	—	1	×	×	—	✓	✓	✓
P01	P01	Input	—	—	—	0Note	0	1	×	—	×	×	✓	✓
		Output	—	—	—	0Note	0	0	0/1	—	TO00 = 0			
	TS27	I/O	—	—	—	1Note	0	1	×	—	×	×	✓	✓
	EI01	Input	—	—	—	0Note	0	1	×	—	×	×	✓	✓
	EO01	Output	—	—	—	0Note	1	0	×	—	—	×	✓	✓
	TO00	Output	—	—	—	0Note	0	0	0	—	×	×	✓	✓
P02	P02	Input	—	×	0	0Note	—	1	×	×	—	✓	✓	✓
		Output	—	0	0	0Note	—	0	0/1	TxD1 = 1	—			
		N-ch open-drain output	—	1	0	0Note	—	0	0/1					
	ANI17	Analog input	—	×	1	0Note	—	1	×	×	—	✓	×	✓
	TS28	I/O	—	×	0	1Note	—	1	×	×	—	×	✓	✓
	TxD1	Output	—	0/1	0	0Note	—	0	1	×	—	✓	✓	✓
P03	P03	Input	—	×	0	0Note	—	1	×	×	—	✓	✓	✓
		Output	—	0	0	0Note	—	0	0/1	RxD1 = 1	—			
		N-ch open-drain output	—	1	0	0Note	—	0	0/1					
	ANI16	Analog input	—	×	1	0Note	—	1	×	×	—	✓	×	✓
	TS29	I/O	—	×	0	1Note	—	1	×	×	—	×	✓	✓
	RxD1	Input	—	×	0	0Note	—	1	×	×	—	✓	✓	✓
P05	P05	Input	—	—	—	0	—	1	×	—	×	✓	✓	✓
		Output	—	—	—	0	—	0	0/1	—	TO05 = 0			
	TS10	I/O	—	—	—	1	—	1	×	—	×	✓	✓	✓
	TI05	Input	PIOR0 = 0	—	—	0	—	1	×	—	×	✓	✓	✓
	TO05	Output	PIOR0 = 0	—	—	0	—	0	0	—	×	✓	✓	✓
P06	P06	Input	—	—	—	0	—	1	×	—	×	✓	✓	✓
		Output	—	—	—	0	—	0	0/1	—	TO06 = 0 CLKA0 = 0			
	TS11	I/O	—	—	—	1	—	1	×	—	×	✓	✓	✓
	TI06	Input	PIOR0 = 0	—	—	0	—	1	×	—	×	✓	✓	✓
	TO06	Output	PIOR0 = 0	—	—	0	—	0	0	—	CLKA0 = 0	✓	✓	✓
	CLKA0	Output	—	—	—	0	—	0	0	—	TO06 = 0	✓	✓	✓

Note This setting is only applicable in the R7F104G products with 44 and 48 pins.

Table 4 - 10 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F104G Products (2/14)

Pin Name	Function Used		PIOR	POMxx	PMCAxx	PMCExx	CCDE	CCSx	PMxx	Pxx	Multiplexed Function Output		32-pin	44-pin	48-pin
	Function Name	I/O									SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock			
P10	P10	Input	—	×	—	0	—	—	1	×	×	×	✓	✓	✓
		Output	—	0	—	0	—	—	0	0/1	SCK00/SCL00 = 1	(TO07) = 0			
		N-ch open-drain output	—	1	—	0	—	—	0	0/1					
	EI10	Input	—	×	—	0	—	—	1	×	×	×	✓	✓	✓
	EO10	Output	—	0/1	—	1	—	—	0	×	×	×	✓	✓	✓
	SCK00	Input	PIOR1 = 0	×	—	0	—	—	1	×	×	×	✓	✓	✓
		Output	PIOR1 = 0	0/1	—	0	—	—	0	1	×	(TO07) = 0	✓	✓	✓
	SCL00	Output	PIOR1 = 0	0/1	—	0	—	—	0	1	×	(TO07) = 0	✓	✓	✓
	(TI07)	Input	PIOR0 = 1	×	—	0	—	—	1	×	×	×	✓	✓	✓
	(TO07)	Output	PIOR0 = 1	0	—	0	—	—	0	0	×	×	✓	✓	✓
P11	P11	Input	—	×	—	0	—	—	1	×	×	×	✓	✓	✓
		Output	—	0	—	0	—	—	0	0/1	SDA00 = 1	(TO06) = 0			
		N-ch open-drain output	—	1	—	0	—	—	0	0/1					
	EI11	Input	—	×	—	0	—	—	1	×	×	×	✓	✓	✓
	EO11	Output	—	0/1	—	1	—	—	0	×	×	×	✓	✓	✓
	SI00	Input	PIOR1 = 0	×	—	0	—	—	1	×	×	×	✓	✓	✓
	RxD0	Input	PIOR1 = 0	×	—	0	—	—	1	×	×	×	✓	✓	✓
	SDA00	I/O	PIOR1 = 0	1	—	0	—	—	0	1	×	(TO06) = 0	✓	✓	✓
	(TI06)	Input	PIOR0 = 1	×	—	0	—	—	1	×	×	×	✓	✓	✓
	(TO06)	Input	PIOR0 = 1	0	—	0	—	—	0	0	×	×	✓	✓	✓
P12	P12	Input	—	×	—	0	—	—	1	×	×	×	✓	✓	✓
		Output	—	0	—	0	—	—	0	0/1	SO00/TxD0 = 1	(TO05) = 0			
		N-ch open-drain output	—	1	—	0	—	—	0	0/1					
	EI12	Input	—	×	—	0	—	—	1	×	×	×	✓	✓	✓
	EO12	Output	—	0/1	—	1	—	—	0	×	×	×	✓	✓	✓
	SO00	Output	PIOR1 = 0	0/1	—	0	—	—	0	1	×	(TO05) = 0	✓	✓	✓
	TxD0	Output	PIOR1 = 0	0/1	—	0	—	—	0	1	×	(TO05) = 0	✓	✓	✓
	(INTP5)	Input	PIOR4 = 1	×	—	0	—	—	1	×	×	×	✓	✓	✓
	(TI05)	Input	PIOR0 = 1	×	—	0	—	—	1	×	×	×	✓	✓	✓
	(TO05)	Output	PIOR0 = 1	0	—	0	—	—	0	0	×	×	✓	✓	✓
P13	P13	Input	—	×	0	0	—	—	1	×	×	×	×	✓	✓
		Output	—	0	0	0	—	—	0	0/1	TxD2 = 1 Note 1	(TO04) = 0 (SDAA0) Note 2 = 0			
		N-ch open-drain output	—	1	0	0	—	—	0	0/1					
	EO13	Output	—	0/1	0	1	—	—	0	×	×	×	×	✓	✓
	TxD2	Output	PIOR1 = 0	0/1	0	0	—	—	0	1	×	(TO04) = 0 (SDAA0) = 0	×	×	✓
	(SDAA0)	I/O	PIOR2 = 1	1	0	0	—	—	0	0	×	(TO04) = 0	×	×	✓
	(TI04)	Input	PIOR0 = 1	×	0	0	—	—	1	×	×	×	×	✓	✓
	(TO04)	Output	PIOR0 = 1	0	0	0	—	—	0	0	×	(SDAA0) Note 2 = 0	×	✓	✓

Note 1. When an input clock signal is to be supplied to SAU1 (when the SAU1EN bit of the PER0 register is set to 1) in the R7F104G product with 44 pins, be sure to set the SO10 bit of the SO1 register to 1.

Note 2. Be sure to set the IICE0 bit of the IICCTL00 register to 0 in the R7F104G product with 44 pins.

Table 4 - 10 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F104G Products (3/14)

Pin Name	Function Used		PIOR	POMxx	PMCAxx	PMCExx	CCDE	CCSx	PMxx	Pxx	Multiplexed Function Output		32-pin	44-pin	48-pin
	Function Name	I/O									SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock			
P14	P14	Input	—	×	—	0	—	—	1	×	×	×	×	×	✓
		Output	—	0	—	0	—	—	0	0/1	**Note 1**	**Note 2** (TO03) = 0 (SCLA0) = 0			
		N-ch open-drain output	—	1	—	0	—	—	0	0/1					
	EO14	Output	—	0/1	—	1	—	—	0	×	×	×	×	×	✓
	RxD2	Input	PIOR1 = 0	×	—	0	—	—	1	×	×	×	×	×	✓
	(SCLA0)	I/O	PIOR2 = 1	1	—	0	—	—	0	0	×	**Note 2** (TO03) = 0	×	×	✓
	(TI03)	Input	PIOR0 = 1	×	—	0	—	—	1	×	×	×	×	×	✓
	(TO03)	Output	PIOR0 = 1	0	—	0	—	—	0	0	×	**Note 2** (SCLA0) = 0	×	×	✓
P16	P16	Input	—	—	—	0	CCDE00 = 0	CCS02 to CCS00 = xxxB	1	×	×	×	✓	✓	✓
		Output	—	—	—	0	CCDE00 = 0	CCS02 to CCS00 = xxxB	0	0/1	×	TO01 = 0			
	CCD00	Output	—	—	—	0	CCDE00 = 1	CCS02 to CCS00 = 001B to 011B	0	0	×	TO01 = 0	✓	✓	✓
	EO16	Output	—	—	—	1	CCDE00 = 0	CCS02 to CCS00 = xxxB	0	×	×	×	✓	✓	✓
	TI01	Input	—	—	—	0	CCDE00 = 0	CCS02 to CCS00 = xxxB	1	×	×	×	✓	✓	✓
	TO01	Output	—	—	—	0	CCDE00 = 0	CCS02 to CCS00 = xxxB	0	0	×	×	✓	✓	✓
	INTP5	Input	PIOR4 = 0	—	—	0	CCDE00 = 0	CCS02 to CCS00 = xxxB	1	×	×	×	✓	✓	✓
	(RxD0)	Input	PIOR1 = 1	—	—	0	CCDE00 = 0	CCS02 to CCS00 = xxxB	1	×	×	×	✓	✓	✓
P17	P17	Input	—	×	—	0	CCDE01 = 0	CCS02 to CCS00 = xxxB	1	×	×	×	✓	✓	✓
		Output	—	0	—	0	CCDE01 = 0	CCS02 to CCS00 = xxxB	0	0/1	(TxD0) = 1	TO02 = 0			
		N-ch open-drain output	—	1	—	0	CCDE01 = 0	CCS02 to CCS00 = xxxB	0	0/1					
	CCD01	Output	—	0/1	—	0	CCDE01 = 1	CCS02 to CCS00 = 001B to 011B	0	0	×	TO02 = 0	✓	✓	✓
	EO17	Output	—	0/1	—	1	CCDE01 = 0	CCS02 to CCS00 = xxxB	0	×	×	×	✓	✓	✓
	TI02	Input	PIOR0 = 0	×	—	0	CCDE01 = 0	CCS02 to CCS00 = xxxB	1	×	×	×	✓	✓	✓
	TO02	Output	PIOR0 = 0	0	—	0	CCDE01 = 0	CCS02 to CCS00 = xxxB	0	0	×	×	✓	✓	✓
	(TxD0)	Output	PIOR1 = 1	0/1	—	0	CCDE01 = 0	CCS02 to CCS00 = xxxB	0	1	×	×	✓	✓	✓

Note 1. When an input clock signal is to be supplied to SAU1 (when the SAU1EN bit of the PER0 register is set to 1) in the R7F104G product with 48 pins, be sure to set the SO10 bit of the SO1 register to 1.

Note 2. Be sure to set the C1OE bit of the COMPOCR register to 0 in the R7F104G product with 48 pins.

Table 4 - 10 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F104G Products (4/14)

Pin Name	Function Used		PMCAxx	PMCTxx	PMxx	Pxx	32-pin	44-pin	48-pin
	Function Name	I/O							
P20	P20	Input	0	—	1	×	✓	✓	✓
		Output	0	—	0	0/1			
	ANI0	Analog input	1	—	1	×	✓	✓	✓
	AVREFP	Reference voltage	1	—	1	×	✓	✓	✓
	EI20	Input	0	—	1	×	✓	✓	✓
P21	P21	Input	0	—	1	×	✓	✓	✓
		Output	0	—	0	0/1			
	ANI1	Analog input	1	—	1	×	✓	✓	✓
	AVREFM	Reference voltage	1	—	1	×	✓	✓	✓
	EI21	Input	0	—	1	×	✓	✓	✓
P22	P22	Input	0	0Note	1	×	✓	✓	✓
		Output	0	0Note	0	0/1			
	ANI2	Analog input	1	0Note	1	×	✓	✓	✓
	ANO0	Analog output	1	0Note	1	×	✓	✓	✓
	TS20	I/O	0	1Note	1	×	×	✓	✓
	EI22	Input	0	0Note	1	×	✓	✓	✓
P23	P23	Input	0	0Note	1	×	✓	✓	✓
		Output	0	0Note	0	0/1			
	ANI3	Analog input	1	0Note	1	×	✓	✓	✓
	ANO1	Analog output	1	0Note	1	×	✓	✓	✓
	IVREF0	Analog input	1	0Note	1	×	✓	✓	✓
	TS21	I/O	0	1Note	1	×	×	✓	✓
	EI23	Input	0	0Note	1	×	✓	✓	✓
P24	P24	Input	0	0Note	1	×	×	✓	✓
		Output	0	0Note	0	0/1			
	ANI4	Analog input	1	0Note	1	×	×	✓	✓
	TS22	I/O	0	1Note	1	×	×	✓	✓
P25	P25	Input	0	0Note	1	×	×	✓	✓
		Output	0	0Note	0	0/1			
	ANI5	Analog input	1	0Note	1	×	×	✓	✓
	TS23	I/O	0	1Note	1	×	×	✓	✓
P26	P26	Input	0	0Note	1	×	×	✓	✓
		Output	0	0Note	0	0/1			
	ANI6	Analog input	1	0Note	1	×	×	✓	✓
	TS24	I/O	0	1Note	1	×	×	✓	✓

Note This setting is only applicable in the R7F104G products with 44 and 48 pins.

Table 4 - 10 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F104G Products (5/14)

Pin Name	Function Used		PMCAxx	PMCTxx	PMxx	Pxx	32-pin	44-pin	48-pin
	Function Name	I/O							
P27	P27	Input	0	0Note	1	×	×	✓	✓
		Output	0	0Note	0	0/1			
	ANI7	Analog input	1	0Note	1	×	×	✓	✓
	TS25	I/O	0	1Note	1	×	×	✓	✓

Note This setting is only applicable in the R7F104G products with 44 and 48 pins.

Table 4 - 10 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F104G Products (6/14)

Pin Name	Function Used		PIOR	PMCTxx	PMxx	Pxx	Multiplexed Function Output		32-pin	44-pin	48-pin
	Function Name	I/O					SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock			
P30	P30	Input	—	0	1	×	×	×	✓	✓	✓
		Output	—	0	0	0/1	SCL11 = 1	VCOUT0 = 0 RTC1HZ = 0Note			
	TSCAP	—	—	1	1	×	×	×	✓	✓	✓
	EI30	Input	—	0	1	×	×	×	✓	✓	✓
	VCOUT0	Output	—	0	0	0	×	RTC1HZ = 0Note	✓	✓	✓
	INTP3	Input	—	0	1	×	×	×	✓	✓	✓
	RTC1HZ	Output	—	0	0	0	×	VCOUT0 = 0	×	✓	✓
	SCL11	Output	—	0	0	1	×	VCOUT0 = 0 RTC1HZ = 0Note	✓	✓	✓
P31	P31	Input	—	0	1	×	—	×	✓	✓	✓
		Output	—	0	0	0/1	—	TO03 = 0 (PCLBUZ0) = 0			
	TS01	I/O	—	1	1	×	×	×	✓	✓	✓
	EI31	Input	—	0	1	×	—	×	✓	✓	✓
	TI03	Input	PIOR0 = 0	0	1	×	—	×	✓	✓	✓
	TO03	Output	PIOR0 = 0	0	0	0	—	(PCLBUZ0) = 0	✓	✓	✓
	INTP4	Input	—	0	1	×	—	×	✓	✓	✓
	(PCLBUZ0)	Output	PIOR3 = 1	0	0	0	—	TO03 = 0	✓	✓	✓

Note Be sure to set the RCLOE1 bit of the RTCC0 register to 0 in the R7F104G product with 32 pins.

Table 4 - 10 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F104G Products (7/14)

Pin Name	Function Used		PIOR	POMxx	PMxx	Pxx	Multiplexed Function Output		32-pin	44-pin	48-pin
	Function Name	I/O					SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock			
P40	P40	Input	—	—	1	×	—	—	✓	✓	✓
		Output	—	—	0	0/1	—	—			
P41	P41	Input	—	—	1	×	—	×	×	×	✓
		Output	—	—	0	0/1	—	TO07 = 0			
	TI07	Input	PIOR0 = 0	—	1	×	—	×	×	×	✓
	TO07	Output	PIOR0 = 0	—	0	0	—	×	×	×	✓

Table 4 - 10 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F104G Products (8/14)

Pin Name	Function Used		PIOR	POMxx	PMCTxx	PMCExx	CCDE	CCSx	PMxx	Pxx	Multiplexed Function Output		32-pin	44-pin	48-pin
	Function Name	I/O									SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock			
P50	P50	Input	—	×	0	0	CCDE03 = 0	CCS02 to CCS00 = xxxB	1	×	×	×	✓	✓	✓
		Output	—	0	0	0	CCDE03 = 0	CCS02 to CCS00 = xxxB	0	0/1	SDA11 = 1	—			
		N-ch open-drain output	—	1	0	0	CCDE03 = 0	CCS02 to CCS00 = xxxB	0	0/1		—			
	TS00	I/O	—	×	1	0	CCDE03 = 0	CCS02 to CCS00 = xxxB	1	×	×	—	✓	✓	✓
	EI50	Input	—	×	0	0	CCDE03 = 0	CCS02 to CCS00 = xxxB	1	×	×	—	✓	✓	✓
	EO50	Output	—	0/1	0	1	CCDE03 = 0	CCS02 to CCS00 = xxxB	0	×	×	—	✓	✓	✓
	CCD03	Output	—	0/1	0	0	CCDE03 = 1	CCS02 to CCS00 = 001B to 011B	0	0	×	—	✓	✓	✓
	INTP1	Input	—	×	0	0	CCDE03 = 0	CCS02 to CCS00 = xxxB	1	×	×	—	✓	✓	✓
	SDA11	I/O	—	1	0	0	CCDE03 = 0	CCS02 to CCS00 = xxxB	0	1	×	—	✓	✓	✓

Table 4 - 10 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F104G Products (9/14)

Pin Name	Function Used		PIOR	PMCTxx	PMxx	Pxx	Multiplexed Function Output		32-pin	44-pin	48-pin
	Function Name	I/O					SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock			
P64	P64	Input	—	0	1	×	—	×	×	✓	✓
		Output	—	0	0	0/1	—	TO10 = 0			
	TS12	I/O	—	1	1	×	—	×	×	✓	✓
	TI10	Input	—	0	1	×	—	×	×	✓	✓
	TO10	Output	—	0	0	0	—	×	×	✓	✓
P65	P65	Input	—	0	1	×	—	×	×	✓	✓
		Output	—	0	0	0/1	—	TO11 = 0			
	TS13	I/O	—	1	1	×	—	×	×	✓	✓
	TI11	Input	—	0	1	×	—	×	×	✓	✓
	TO11	Output	—	0	0	0	—	×	×	✓	✓
P66	P66	Input	—	0	1	×	—	×	×	×	✓
		Output	—	0	0	0/1	—	TO12 = 0			
	TS14	I/O	—	1	1	×	—	×	×	×	✓
	TI12	Input	—	0	1	×	—	×	×	×	✓
	TO12	Output	—	0	0	0	—	×	×	×	✓

Table 4 - 10 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F104G Products (10/14)

Pin Name	Function Used		PIOR	POMxx	PMCTxx	PMxx	Pxx	Multiplexed Function Output		32-pin	44-pin	48-pin
	Function Name	I/O						SAU and UARTA outputs except a UARTA clock	Other outputs and a UARTA clock			
P70	P70	Input	—	—	0	1	×	×	—	✓	✓	✓
		Output	—	—	0	0	0/1	SCK21/SCL21 = 1	—			
	TS02	I/O	—	—	1	1	×	×	—	✓	✓	✓
	RIN0	Input	—	—	0	1	×	×	—	✓	✓	✓
	KR0	Input	—	—	0	1	×	×	—	✓	✓	✓
	SCK21	Input	—	—	0	1	×	×	—	✓	✓	✓
		Output	—	—	0	0	1	×	—	✓	✓	✓
	SCL21	Output	—	—	0	0	1	×	—	✓	✓	✓
P71	P71	Input	—	×	0	1	×	×	—	✓	✓	✓
		Output	—	0	0	0	0/1	SDA21 = 1	—			
		N-ch open-drain output	—	1	0	0	0/1					
	TS03	I/O	—	×	1	1	×	×	—	✓	✓	✓
	KR1	Input	—	×	0	1	×	×	—	✓	✓	✓
	SI21	Input	—	×	0	1	×	×	—	✓	✓	✓
	SDA21	I/O	—	1	0	0	1	×	—	✓	✓	✓
	RxDA0	Input	—	×	0	1	×	×	—	✓	✓	✓
P72	P72	Input	—	×	0	1	×	×	—	✓	✓	✓
		Output	—	0	0	0	0/1	SO21 = 1 TxDA0 = 1	—			
		N-ch open-drain output	—	1	0	0	0/1					
	TS04	I/O	—	×	1	1	×	×	—	✓	✓	✓
	KR2	Input	—	×	0	1	×	×	—	✓	✓	✓
	SO21	Output	—	0/1	0	0	1	TxDA0 = 1	—	✓	✓	✓
	TxDA0	Output	—	0/1	0	0	1	SO21 = 1	—	✓	✓	✓
P73	P73	Input	—	—	0	1	×	×	—	✓	✓	✓
		Output	—	—	0	0	0/1	**Note**	—			
	TS05	I/O	—	—	1	1	×	×	—	✓	✓	✓
	KR3	Input	—	—	0	1	×	×	—	✓	✓	✓
P76	P76	Input	—	—	0	1	×	—	—	✓	✓	✓
		Output	—	—	0	0	0/1	—	—	✓	✓	✓
	TS08	I/O	—	—	1	1	×	—	—	✓	✓	✓
	KR6	Input	—	—	0	1	×	—	—	✓	✓	✓
	INTP10	Input	PIOR1 = 0	—	0	1	×	—	—	✓	✓	✓
	(RxD2)	Input	PIOR1 = 1	—	0	1	×	—	—	✓	✓	✓
P77	P77	Input	—	—	0	1	×	—	—	✓	✓	✓
		Output	—	—	0	0	0/1	(TxD2) = 1	—	✓	✓	✓
	TS09	I/O	—	—	1	1	×	—	—	✓	✓	✓
	KR7	Input	—	—	0	1	×	—	—	✓	✓	✓
	INTP11	Input	PIOR1 = 0	—	0	1	×	—	—	✓	✓	✓
	(TxD2)	Output	PIOR1 = 1	—	0	0	1	—	—	✓	✓	✓

Note When an input clock signal is to be supplied to SAU0 (when the SAU0EN bit of the PER0 register is set to 1) in the R7F104G product with 32 pins, be sure to set the SO01 bit of the SO0 register to 1.

Table 4 - 10 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F104G Products (11/14)

Pin Name	Function Used		CMC		PMxx	Pxx	32-pin	44-pin	48-pin
	Function Name	I/O	EXCLK, OSCSEL, EXCLKS, OSCSELS	XTSEL					
P121	P121	Input	00xxB/10xxB/11xxB	0	1	×	✓	✓	✓
		Output	00xxB/10xxB/11xxB	0	0	0/1			
	EI121	Input	00xxB/10xxB/11xxB	0	1	×	✓	✓	✓
	VBAT	—	01xxB	0	1	×	✓	✓	✓
	X1	—	01xxB	0	1	×	✓	✓	✓
P122	P122	Input	00xxB/10xxB	0	1	×	✓	✓	✓
		Output	00xxB/10xxB	0	0	0/1			
	EI122	Input	00xxB/10xxB	0	1	×	✓	✓	✓
	X2	—	01xxB	0	1	×	✓	✓	✓
	EXCLK	Input	11xxB	0	1	×	✓	✓	✓
P123	P123	Input	xx00B/xx10B/xx11B	0	—	×	×	✓	✓
	XT1	—	xx01B	0	—	×	×	✓	✓
P124	P124	Input	xx00B/xx10B	0	—	×	×	✓	✓
	XT2	—	xx01B	0	—	×	×	✓	✓
	EXCLKS	Input	xx11B	0	—	×	×	✓	✓

Table 4 - 10 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F104G Products (12/14)

Pin Name	Function Used		PMxx	Pxx	32-pin	44-pin	48-pin
	Function Name	I/O					
P130	P130	Output	—	0/1	×	×	✓
P137	P137	Input	—	×	✓	✓	✓
	EI137	Input	—	×	✓	✓	✓
	INTP0	—	—	×	✓	✓	✓

Table 4 - 10 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F104G Products (13/14)

Pin Name	Function Used		PIOR	POMxx	PMCAxx	PMxx	Pxx	32-pin	44-pin	48-pin
	Function Name	I/O								
P147	P147	Input	—	—	0	1	×	✓	✓	✓
		Output	—	—	0	0	0/1			
	ANI18	Analog input	—	—	1	1	×	✓	×	×
	IVCMP0	Analog input	—	—	1	1	×	✓	✓	✓
	EI147	Input	—	—	0	1	×	✓	✓	✓

Table 4 - 10 Examples of Register and Output Latch Settings for Multiplexed Functions of R7F104G Products (14/14)

Pin Name	Function Used		PMCAxx	PMCTxx	PMxx	Pxx	32-pin	44-pin	48-pin
	Function Name	I/O							
P150	P150	Input	0	0	1	×	×	✓	✓
		Output	0	0	0	0/1			
	TS30	I/O	0	1	1	×	×	✓	✓
P153	P153	Input	0	0	1	×	×	✓	✓
		Output	0	0	0	0/1			
	TS33	I/O	0	1	1	×	×	✓	✓

4.6 Points for Caution When Using Port Function

4.6.1 Point for Caution on 1-bit memory manipulation instruction for port register (Pxx)

When a 1-bit memory manipulation instruction is executed for a port that has both input and output pins, the output latch values of input port pins that are not selected by the instruction may also be changed. Accordingly, we recommend rewriting the output latch values before switching port pins from inputs to outputs.

Example: When the output on P10 is changed from low to high by a 1-bit memory manipulation instruction under the following conditions, the output latch value of port 1 changes to FFH.

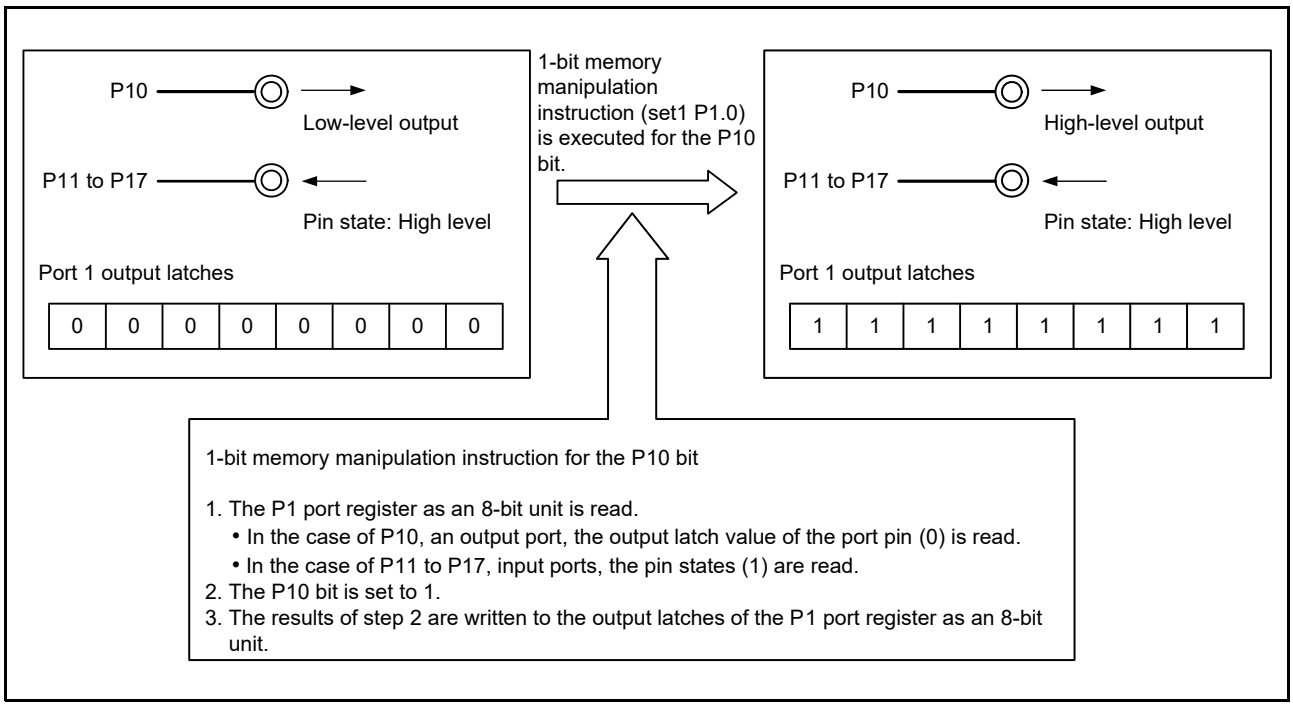
- P10 is set for output.
- P11 to P17 are set for inputs and the signals are at the high level.
- The output latch value of port 1 is 00H.

When the setting of the PMmn bit is 1, writing to the Pxx register updates the output latches, and reading from the Pxx register returns the pin states. A 1-bit memory manipulation instruction is executed in the following order in this MCU.

1. The Pxx register is read as an 8-bit unit.
2. The target bit is modified.
3. The Pxx register is overwritten as an 8-bit unit.

In step 1, the output latch value (0) of P10 set as an output, is read, while the states of P11 to P17 set as inputs are read. If the signals on P11 to P17 are at the high level, the value read is FEH. The value is changed to FFH in step 2. FFH is written to the output latches in step 3.

Figure 4 - 18 1-bit Memory Manipulation Instruction for 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 multiplexed functions must be set to the initial state of the pin so as to prevent conflicting outputs. This also applies to the functions assigned by using the peripheral I/O redirection register (PIOR). For details about the multiplexed function output, see **4.5 Register Settings When Using Multiplexed Function**. No specific setting is required for input pins because the output of their multiplexed 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.

Section 5 Operation State Control

The operating voltage, operating timing, and operating current of the internal circuit are optimized using operating modes of the flash memory. Select an appropriate operating mode of the flash memory in accord with the operating voltage range and clock frequencies of the MCU. The operating mode of the flash memory set by the option byte is selected for operation immediately after a reset is released. The mode can be changed by setting of the relevant register.

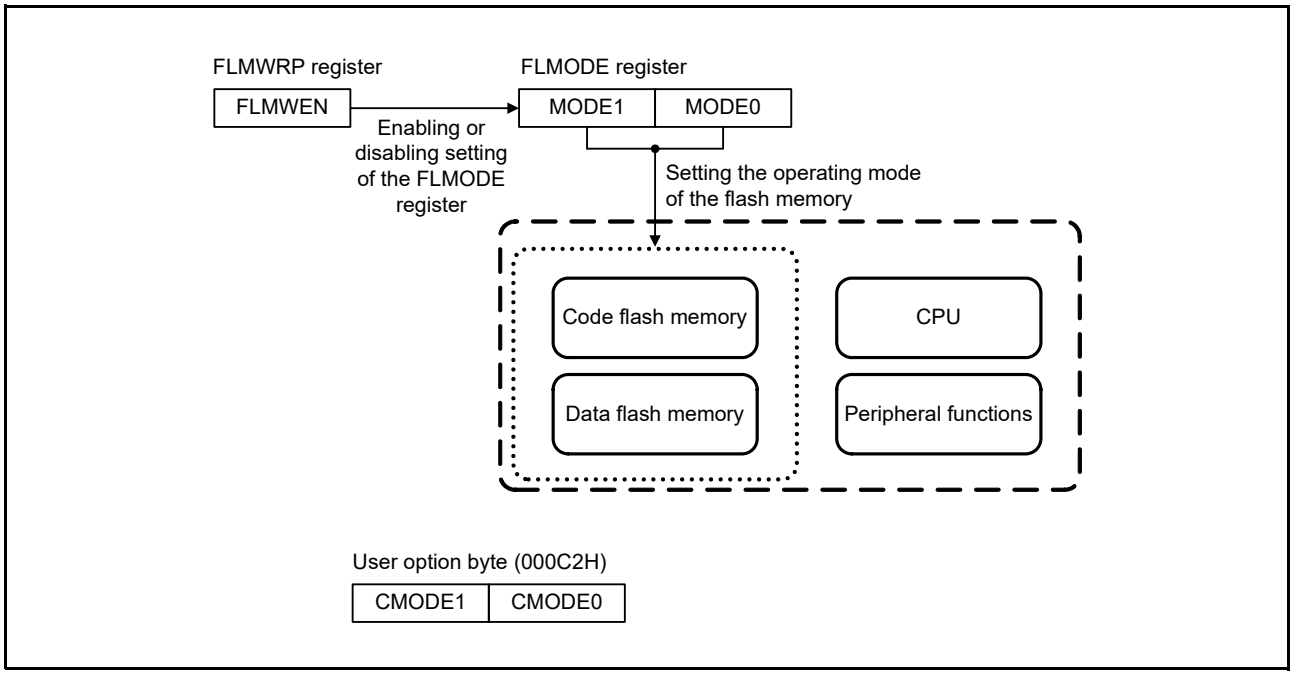
5.1 Configuration of Operation State Control

Operation state control is supported by the following hardware blocks.

Table 5 - 1 Configuration of Operation State Control

Item	Configuration
Option byte	• Address of the user option byte: 000C2H
Control registers	• Flash operating mode select register (FLMODE) • Flash operating mode protect register (FLMWRP)

Figure 5 - 1 Block Diagram of Operation State Control



There are the following four operating modes of the flash memory.

- HS (high-speed main) mode
- LS (low-speed main) mode
- LP (low-power main) mode
- SUB mode

The MCU can be operated efficiently by setting these operating modes of the flash memory according to MCU operating conditions. **Table 5 - 2** lists the features of each operating mode of the flash memory.

Table 5 - 2 Features of Each Operating Mode of the Flash Memory

Operating Mode of the Flash Memory	Recommended Operating Range		Description
HS (high-speed main) mode	1.6 to 1.8 V	1 to 4 MHz (Rewriting of the flash memory is not possible.)	High-speed CPU operation (at 32 MHz (max.)) is possible in this mode. Suitable when CPU processing capacity is required.
	1.6 to 1.8 V	1 to 2 MHz	
	1.8 to 5.5 V	1 to 32 MHz	
LS (low-speed main) mode	1.6 to 1.8 V	1 to 4 MHz (Rewriting of the flash memory is not possible.)	The operating current and CPU operation processing (at 24 MHz (max.)) are well-balanced in this mode.
	1.6 to 1.8 V	1 to 2 MHz	
	1.8 to 5.5 V	1 to 24 MHz	
LP (low-power main) mode	1.6 to 5.5 V	1 to 2 MHz (Rewriting of the flash memory is not possible.)	The CPU operates at 1 to 2 MHz in this mode. Low operating current is realized at 1 to 2 MHz.
SUB mode	1.6 to 5.5 V	32.768 kHz (Rewriting of the flash memory is not possible.)	CPU operation is driven by the subsystem clock.**Note** This mode enables low-power operation.

Note The subsystem clock can be derived from the subsystem clock X (f_{SX}) or the low-speed on-chip oscillator clock (f_{IL}).

5.2 Registers for Controlling the Operation State Control

The following registers are used to control the operation state control.

- Flash operating mode select register (FLMODE)
- Flash operating mode protect register (FLMWRP)

5.2.1 Flash operating mode select register (FLMODE)

The FLMODE is an 8-bit register used to control operating modes of the flash memory. This register can be set by a 1-bit or 8-bit memory manipulation instruction. Note that the value of this register cannot be changed when the FLMWEN bit in the flash operating mode protect register (FLMWRP) is 0. The settings of the MODE1 and MODE0 bits are updated to those of the CMODE1 and CMODE0 bits in the user option byte at address 000C2H following a reset.

Figure 5 - 2 Format of Flash Operating Mode Select Register (FLMODE)

Address: F00AAH
After reset: 40H/80H/C0H**Note**
R/W: R/W

Symbol	<7>	<6>	5	4	3	2	1	0
FLMODE	MODE1	MODE0	0	0	0	0	0	0

MODE1	MODE0	Selection of operating mode of the flash memory
0	0	Setting prohibited
0	1	LP (low-power main) mode (Selectable when 1 MHz ≤ f_{CLK} ≤ 2 MHz in LS mode)
1	0	LS (low-speed main) mode (Selectable when 1 MHz ≤ f_{CLK} ≤ 24 MHz in HS mode or in LP mode)
1	1	HS (high-speed main) mode (Selectable when in LS mode)

Note The initial value of the FLMODE register is set to the value of the MODE1 and MODE0 bits updated with the set value of the CMODE1 and CMODE0 bits in the user option byte (address: 000C2H).

Caution 1. The value of the FLMODE register can be changed when the FLMWEN bit in the flash operating mode protect register (FLMWRP) is 1. After the value of the FLMODE register is changed, set the FLMWEN bit to 0.

Caution 2. Operation is in SUB mode when the setting of the CSS bit in the system clock control register (CKC) is 1 (operation of the CPU and peripheral functions is driven by the subsystem clock) regardless of the settings of the MODE1 and MODE0 bits.

Caution 3. Do not change the values of the MODE1 and MODE0 bits by using the DTC or SMS.

Caution 4. When changing the operating mode of the flash memory, make sure that operation is possible within the voltage range and operating frequency range in the changed operating mode of the flash memory before changing the mode.

(Cautions are listed on the next page.)

Caution 5. When the operating mode of the flash memory is changed by the MODE1 and MODE0 bits, the CPU is placed in the wait state for the following time until the mode changes. Interrupt requests are held pending during this wait period.
Times for switching between the operating modes of the flash memory

Switching between the Operating Modes of the Flash Memory	Transition Time
LS (low-speed main) mode → HS (high-speed main) mode	225 cycles[Note]
LP (low-power main) mode → LS (low-speed main) mode	10 cycles[Note]
LS (low-speed main) mode → LP (low-power main) mode	10 cycles[Note]
HS (high-speed main) mode → LS (low-speed main) mode	30 cycles[Note]

Note The cycle of the CPU/peripheral hardware clock (f_{CLK})

Caution 6. When rewriting the FLMODE register, insert at least one clock cycle of the CPU/peripheral hardware clock (f_{CLK}) after having rewritten the FLMODE register. Do not write to the FLMODE register successively.

Caution 7. Do not change the FLMODE register while rewriting the flash memory.

Caution 8. Writing new values to the FLMODE register should be done while the snooze mode sequencer is stopped, that is, while the setting of the SMSSTART bit is 0, or the settings of the SMSSTAT bit and the SMSTRGWAIT bit are 0 and 1, respectively.

Caution 9. Before changing the operating mode of the flash memory, set the DFLEN bit in the data flash control register (DFLCTL) to 1 to enable access to the data flash memory.

Caution 10. Before rewriting the contents of the code flash memory or data flash memory area by self-programming, be sure to place the MCU in HS (high-speed main) or LS (low-speed main) mode.

5.2.2 Flash operating mode protect register (FLMWRP)

The FLMWRP is an 8-bit register used to control access to the flash operating mode select register. This register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 5 - 3 Format of Flash Operating Mode Protect Register (FLMWRP)

Address: F00ABH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	<0>
FLMWRP	0	0	0	0	0	0	0	FLMWEN

FLMWEN	Control of flash operating mode select register (FLMODE)
0	Rewriting the FLMODE register is disabled.
1	Rewriting the FLMODE register is enabled.

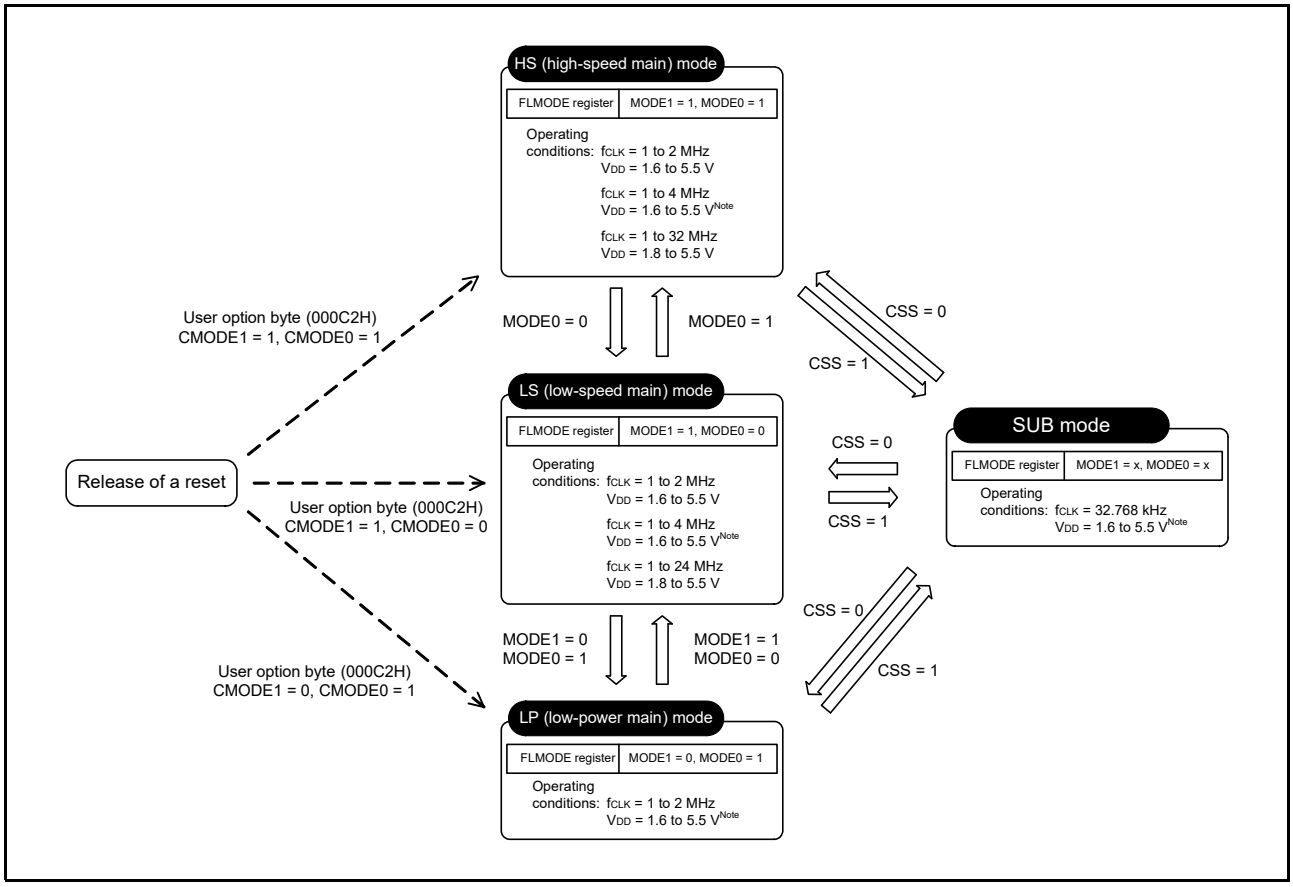
5.3 Initial Setting of Operating Modes of the Flash Memory

The user option byte (000C2H) is used to set the initial state of the operating mode of the flash memory and the high-speed on-chip oscillator after a reset is released. Set an appropriate operating mode of the flash memory according to the V_{DD} voltage and the high-speed on-chip oscillator frequency at a reset release. When a reset is released, the values of the CMODE1 and CMODE0 bits are reflected in the MODE1 and MODE0 bits in the flash operating mode select register (FLMODE) and the values of the FRQSEL3 to FRQSEL0 bits are reflected in the high-speed on-chip oscillator frequency select register (HOCODIV). For details on the user option byte (000C2H), see **Section 32 Option Bytes**.

5.4 Transitions between Operating Modes of the Flash Memory

The settings of the CMODE1 and CMODE0 bits in the user option byte (000C2H) determine the initial operating mode of the flash memory immediately after a reset is released. The initial state can be selected from among the HS (high-speed main) mode, LS (low-speed main) mode, or LP (low-power main) mode. The values of the CMODE1 and CMODE0 bits are reflected in the MODE1 and MODE0 bits in the flash operating mode select register (FLMODE). After that, the operating mode of the flash memory can be changed by changing the value of the FLMODE register during CPU operation. Setting the CSS bit in the CKC register to 1 enables automatically placing the MCU in SUB mode.

Figure 5 - 4 State Transitions between Operating Modes of the Flash Memory



Note Rewriting of the flash memory is not possible.

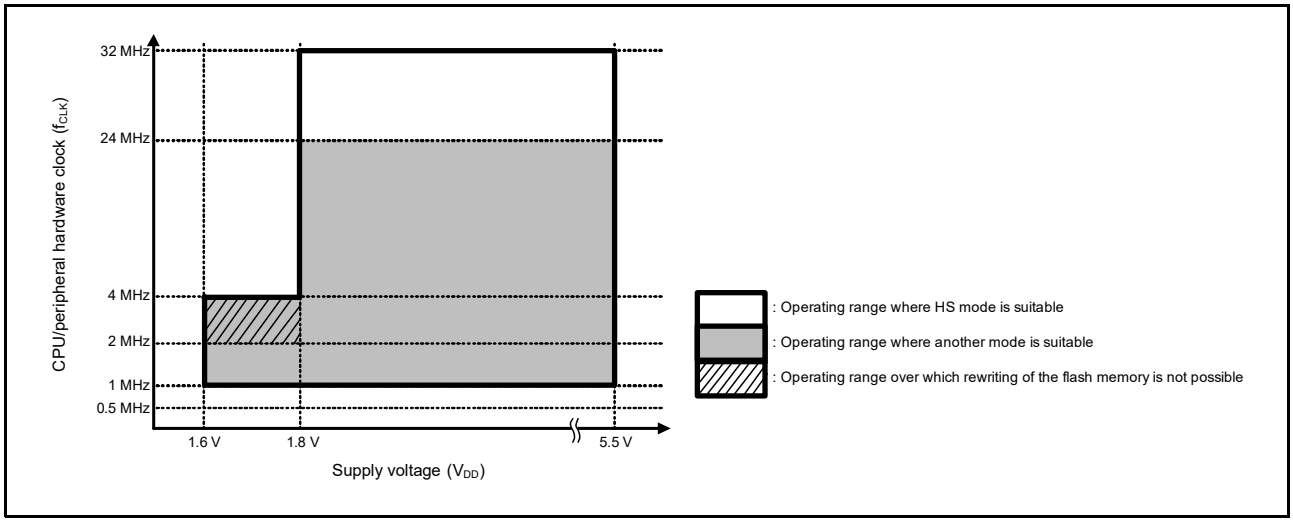
Caution **When a reset is applied while the MCU operates, operation always starts in the operating mode of the flash memory set by the option byte after a reset release. Therefore, make sure that operation does not start outside the operating voltage range when a reset is released by setting the LVD detection voltage to at least the operating voltage range of the operating mode of the flash memory set in the option byte.**

5.5 Details of Operating Modes of the Flash Memory

5.5.1 Details of HS (high-speed main) mode

HS (high-speed main) mode is suitable for applications that require CPU high-speed processing. The CPU can run in HS mode immediately after release from the reset state. Transition to this mode from LS (low-speed main) mode is possible. The suitable operating range for HS mode is when the supply voltage satisfies 1.8 V ≤ V_{DD} ≤ 5.5 V and the operating frequency satisfies 24 MHz < f_{CLK} ≤ 32 MHz. When 24 MHz or a lower frequency is used for operation, another mode can be used as the suitable operating mode of the flash memory.

Figure 5 - 5 Operating Range in HS Mode

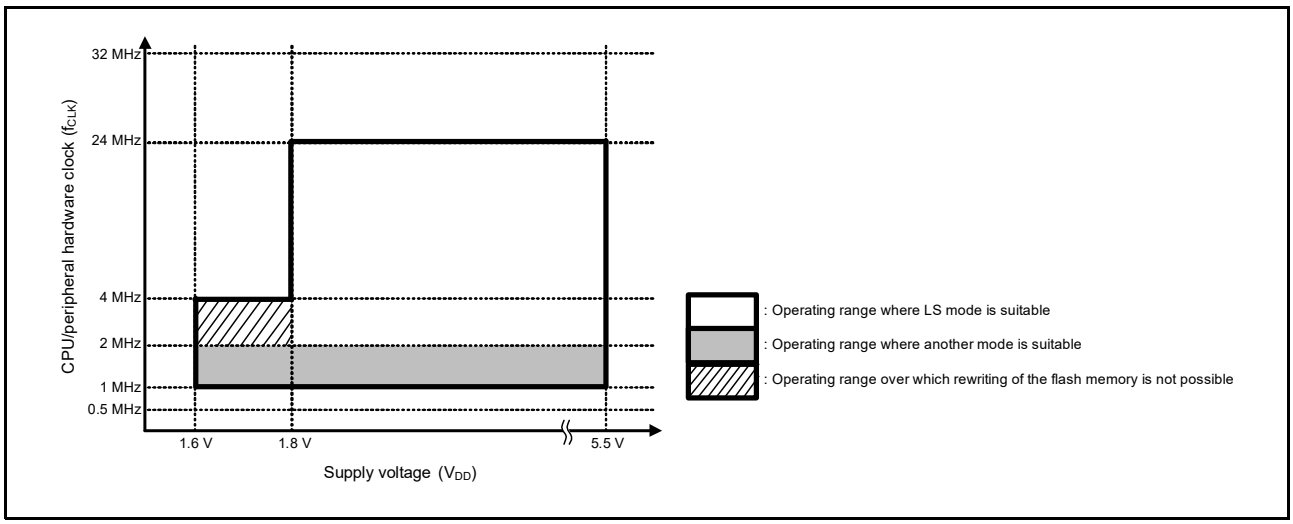


5.5.2 Details of LS (low-speed main) mode

LS (low-speed main) mode supports performance in terms of both CPU processing capacity and operating voltage, and is suitable for applications that require low-power consumption and are capable of running at 2 to 24 MHz. The CPU can run in LS mode immediately after release from the reset state. Transition to this mode from HS (high-speed main) mode or LP (low-power main) mode is possible. For a transition from HS mode to LS mode, make sure that the operating frequency satisfies 1 MHz ≤ f_{CLK} ≤ 24 MHz. The suitable operating range for LS mode is when the supply voltage satisfies 1.8 V ≤ V_{DD} ≤ 5.5 V and the operating frequency satisfies 2 MHz < f_{CLK} ≤ 24 MHz, or when the supply voltage satisfies 1.6 V ≤ V_{DD} < 1.8 V and the operating frequency satisfies 2 MHz < f_{CLK} ≤ 4 MHz**Note**.

Note The operating frequency must satisfy 1 MHz ≤ f_{CLK} ≤ 2 MHz when rewriting of the flash memory is required.

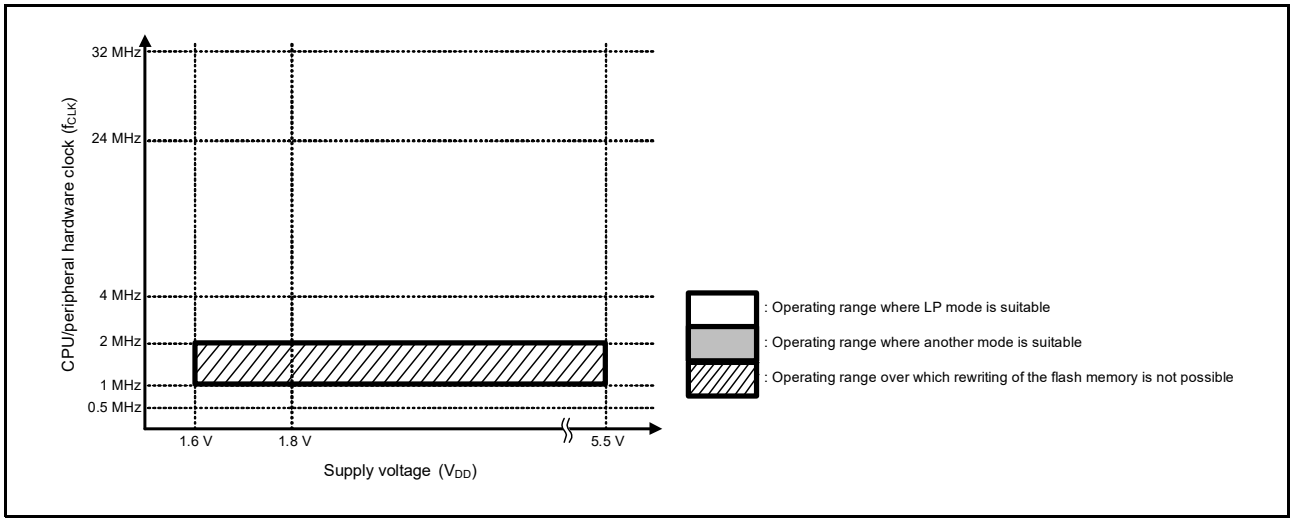
Figure 5 - 6 Operating Range in LS Mode



5.5.3 Details of LP (low-power main) mode

In LP (low-power main) mode, the CPU operates with a low voltage and at a frequency from 1 to 2 MHz. The CPU can run in LP mode immediately after release from the reset state. Transition to this mode from LS (low-speed main) mode is possible. For a transition from LS mode to LP mode, make sure that the operating frequency satisfies 1 MHz ≤ f_{CLK} ≤ 2 MHz. The suitable operating range for LP mode is when the supply voltage satisfies 1.6 V ≤ V_{DD} ≤ 5.5 V and the operating frequency satisfies 1 MHz ≤ f_{CLK} ≤ 2 MHz. When rewriting of flash memory is required, place the CPU in LS mode.

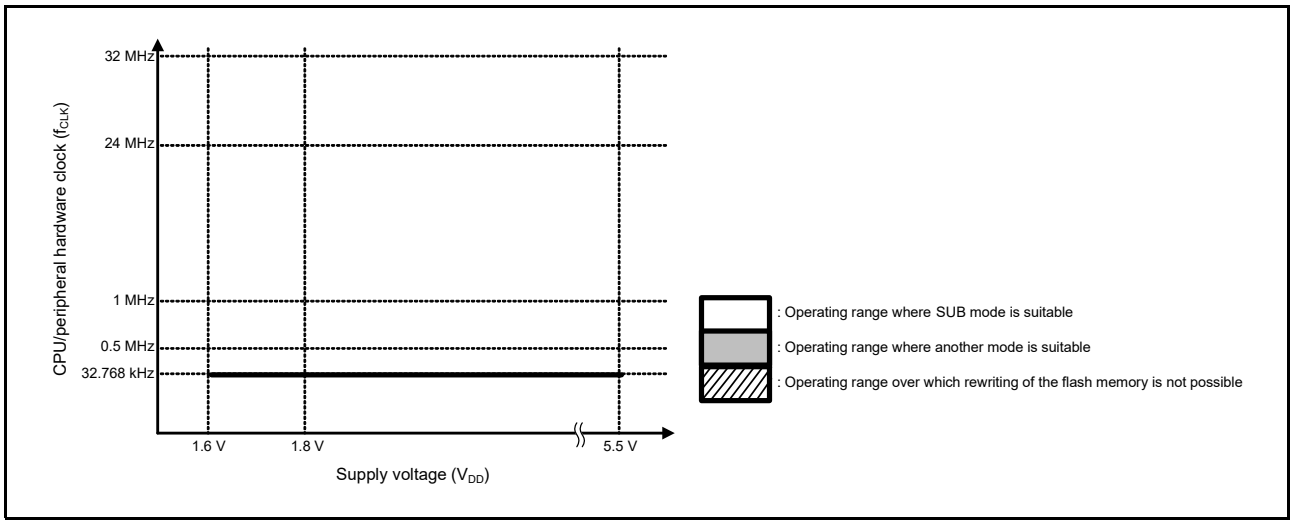
Figure 5 - 7 Operating Range in LP Mode



5.5.4 Details on SUB mode

In SUB mode, the CPU operates at a frequency of 32.768 kHz. Transition to SUB mode from HS (high-speed main) mode, LS (low-speed main) mode, or LP (low-power main) mode is possible. Setting the CSS bit in the system clock control register (CKC) to 1 places the CPU in SUB mode. Rewriting of the flash memory is not possible in SUB mode. When rewriting of flash memory is required, place the CPU in HS mode or LS mode.

Figure 5 - 8 Operating Range in SUB Mode



Section 6 Clock Generator

6.1 Functions of Clock Generator

The clock generator generates clocks to be supplied to the CPU and peripheral hardware. The following kinds of system clocks and clock oscillators are selectable.

1. Main system clock
   <1> X1 oscillator
   This circuit oscillates a clock of f_X = 1 to 20 MHz by connecting a resonator to the X1 and X2 pins. Oscillation can be stopped by executing a STOP instruction or setting of the MSTOP bit of the clock operation status control register (CSC).
   <2> High-speed on-chip oscillator
   The frequency of oscillation for f_{IH} can be selected from among 32, 24, 16, 12, 8, 6, 4, 3, 2, and 1 MHz (typ.) by using a user option byte (000C2H). After release from the reset state, the CPU always starts operating with this high-speed on-chip oscillator clock. Oscillation can be stopped by executing a STOP instruction or setting of the HIOSTOP bit of the CSC register. The frequency specified by using the user option byte can be changed by using the high-speed on-chip oscillator frequency select register (HOCODIV). For details about the frequency, see **Figure 6 - 11 Format of High-speed On-chip Oscillator Frequency Select Register (HOCODIV)**. The frequencies that can be specified for the high-speed on-chip oscillator by using the user option byte and the high-speed on-chip oscillator frequency select register (HOCODIV) are shown below.

Power Supply Voltage	Oscillation Frequency (MHz)									
	1	2	3	4	6	8	12	16	24	32
1.8 V ≤ V_{DD} ≤ 5.5 V	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
1.6 V ≤ V_{DD} ≤ 5.5 V	✓	✓	✓	✓	—	—	—	—	—	—

   <3> Middle-speed on-chip oscillator
   The frequency of oscillation for f_{IM} can be selected from among 4, 2, and 1 MHz (typ.) by using the MOCODIV1 and MOCODIV0 bits of the MOCODIV register. Oscillation can be stopped by executing a STOP instruction or clearing of the MIOEN bit of the CSC register.

An external main system clock (f_{EX} = 1 to 20 MHz) can also be supplied from the EXCLK/X2/P122 pin. External main system clock input can be disabled by executing a STOP instruction or setting of the MSTOP bit.

As the main system clock, a high-speed system clock (X1 clock or external main system clock) or a main on-chip oscillator clock (high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock) can be selected by setting of the MCM0 and MCM1 bits of the system clock control register (CKC).

Note that the usable frequency range of the CPU/peripheral hardware clock depends on the settings for the operating mode of the flash memory and V_{DD} power supply voltage. When the main system clock is to be used as the CPU/peripheral hardware clock, select the frequency of the main system clock to suit the operating mode of the flash memory specified in the CMODE1 and CMODE0 bits of a user option byte (000C2H) (see **Section 32 Option Bytes**) or the flash operating mode select register (FLMODE).

2. Subsystem clock

<1> XT1 oscillator

This circuit oscillates a clock of f_{XT} = 32.768 kHz by connecting a 32.768-kHz resonator to the XT1 and XT2 pins. Oscillation can be stopped by setting the XTSTOP bit 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. External subsystem clock input can be disabled by the setting of the XTSTOP bit.

<2> Low-speed on-chip oscillator

This circuit oscillates a clock of f_{IL} = 32.768 kHz (typ.).

The low-speed on-chip oscillator operates when either of the following conditions is met.

- The value of one or more of the following bits is 1: WDTON bit of a user option byte (000C0H), WUTMMCK0 bit of the subsystem clock supply mode control register (OSMC), and SELLOSC bit of the subsystem clock select register (CKSEL)
  However, if a HALT or STOP instruction is executed when WDTON = 1, WUTMMCK0 = 0, SELLOSC = 0 and the WDSTBYON bit of the user option byte (000C0H) is 0, this oscillator stops oscillating.
- f_{IL} is selected as the source clock for use in waiting by the SNOOZE mode sequencer.

Remark f_X: X1 clock oscillation frequency
f_{IH}: High-speed on-chip oscillator clock frequency (32 MHz max.)
f_{IM}: Middle-speed on-chip oscillator clock frequency (4 MHz max.)
f_{EX}: External main system clock frequency
f_{XT}: XT1 clock oscillation frequency
f_{EXS}: External subsystem clock frequency
f_{IL}: Low-speed on-chip oscillator clock frequency

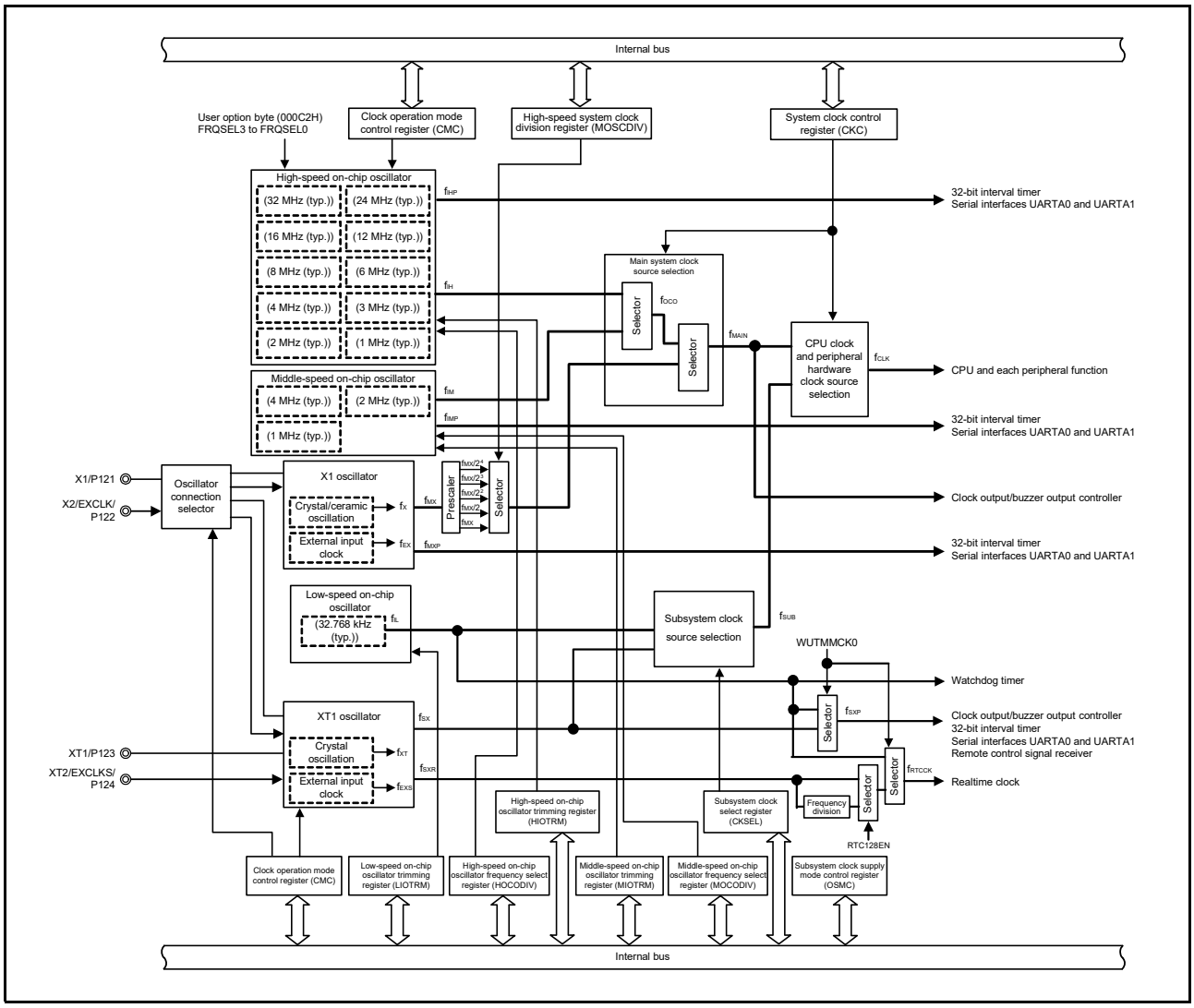
6.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 6 - 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 (PER0, PER1) • Subsystem clock supply mode control register (OSMC) • Subsystem clock select register (CKSEL) • High-speed on-chip oscillator frequency select register (HOCODIV) • Middle-speed on-chip oscillator frequency select register (MOCODIV) • High-speed system clock division register (MOSCDIV) • High-speed on-chip oscillator trimming register (HIOTRM) • Middle-speed on-chip oscillator trimming register (MIOTRM) • Low-speed on-chip oscillator trimming register (LIOTRM) • Standby mode release setting register (WKUPMD)
Oscillators	• X1 oscillator • XT1 oscillator • High-speed on-chip oscillator • Middle-speed on-chip oscillator • Low-speed on-chip oscillator

Figure 6 - 1 Block Diagram of Clock Generator



Remark fX: X1 clock oscillation frequency
fEX: External main system clock frequency
fIH: High-speed on-chip oscillator clock frequency (32 MHz max.)
fIHP: High-speed on-chip oscillator peripheral clock frequency
(32 MHz when FRQSEL3 = 1, 24 MHz when FRQSEL3 = 0)
fIM: Middle-speed on-chip oscillator clock frequency (4 MHz max.)
fIMP: Middle-speed on-chip oscillator peripheral clock frequency (4 MHz)
fMX: High-speed system clock frequency
fMXP: High-speed peripheral clock frequency
fMAIN: Main system clock frequency
fXT: XT1 clock oscillation frequency
fEXS: External subsystem clock frequency
fSX: Subsystem clock X frequency
fSXR: Subsystem clock XR frequency
fRTCCK: Operating clock for the realtime clock controller
fSXP: Low-speed peripheral clock frequency
fSUB: Subsystem clock frequency
fCLK: CPU/peripheral hardware clock frequency
fIL: Low-speed on-chip oscillator clock frequency
foco: Main on-chip oscillator clock frequency (fIH or fIM)

6.3 Registers for Controlling the Clock Generator

The following registers are used to control the clock generator.

- Clock operation mode control register (CMC)
- System clock control register (CKC)
- Clock operation status control register (CSC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)
- Peripheral enable registers 0, 1 (PER0, PER1)
- Subsystem clock supply mode control register (OSMC)
- Subsystem clock select register (CKSEL)
- High-speed on-chip oscillator frequency select register (HOCODIV)
- Middle-speed on-chip oscillator frequency select register (MOCODIV)
- High-speed system clock division register (MOSCDIV)
- High-speed on-chip oscillator trimming register (HIOTRM)
- Middle-speed on-chip oscillator trimming register (MIOTRM)
- Low-speed on-chip oscillator trimming register (LIOTRM)
- Standby mode release setting register (WKUPMD)

Caution The inclusion of registers and bit functions depends on the product. Be sure to set bits to which no functions are allocated to their initial values.

6.3.1 Clock operation mode control register (CMC)

The CMC 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 the gain of the oscillator. The CMC register can be written only once by an 8-bit memory manipulation instruction after release from the reset state. This register can be read by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 6 - 2 Format of Clock Operation Mode Control Register (CMC) (1/2)

Address: FFFA0H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	EXCLKS **Note 1**	OSCSELS **Note 2**	XTSEL **Note 3**	AMPHS1	AMPHS0	AMPH

30- to 36-pin R7F100G products

XTSEL **Note 4**	EXCLK	OSCSEL	EXCLKS **Note 4**	OSCSELS **Note 4**	System clock pin operation mode	X1/P121/ XT1 pin	X2/EXCLK/ P122/XT2/ EXCLKS pin
0	0	0	0	0	Port mode	Port	Port
0	0	1	0	0	X1 oscillation mode	Crystal/ceramic resonator connection	
0	1	0	0	0	Port mode	Port	Port
0	1	1	0	0	External clock input mode	Port	External clock EXCLK input
1	0	0	0	0	Port mode	Port	Port
1	0	0	0	1	XT1 oscillation mode	Crystal resonator connection	
1	0	0	1	0	Port mode	Port	Port
1	0	0	1	1	External clock input mode	Port	External clock EXCLKS input
Other than above					Setting prohibited		

40- to 128-pin R7F100G and 32- to 48-pin R7F104G products

EXCLK	OSCSEL	High-speed system clock pin operation mode	X1/P121 pin	X2/EXCLK/P122 pin
0	0	Port mode	Port	Port
0	1	X1 oscillation mode	Crystal/ceramic resonator connection	
1	0	Port mode	Port	Port
1	1	External clock input mode	Port	External clock input

Figure 6 - 2 Format of Clock Operation Mode Control Register (CMC) (2/2)

EXCLKS **Note 4**	OSCSELS **Note 4**	Subsystem clock pin operation mode	XT1/P123 pin	XT2/EXCLKS/P124 pin
0	0	Input port mode	Input port	Input port
0	1	XT1 oscillation mode	Crystal resonator connection	
1	0	Input port mode	Input port	Input port
1	1	External clock input mode	Input port	External clock input

AMPHS1 **Note 4**	AMPHS0 **Note 4**	Selection of the oscillation mode of the XT1 oscillator
0	0	Low power consumption oscillation 1 **Note 5**
0	1	Normal oscillation
1	0	Low power consumption oscillation 2 **Note 5**
1	1	Low power consumption oscillation 3 **Note 5**

AMPH	Control of the X1 clock oscillation frequency
0	1 MHz ≤ fx ≤ 10 MHz
1	10 MHz < fx ≤ 20 MHz

Note 1. Be sure to leave the EXCLKS bit set to 0 in the 32-pin R7F104G product.

Note 2. Be sure to set the OSCSELS bit to 1 in the 32-pin R7F104G product.

Note 3. Writing to the XTSEL bit is only possible in the 30- to 36-pin R7F100G products. Be sure to leave this bit set to 0 in the 40- to 128-pin R7F100G and 32- to 48-pin R7F104G products.

Note 4. The EXCLKS, OSCSELS, XTSEL, AMPHS1, and AMPHS0 bits are only initialized by a power-on reset; they retain the values when a reset caused by another source occurs.

Note 5. The gain and operating current of the XT1 clock oscillator decrease in the following order: low power consumption oscillation 1 > low power consumption oscillation 2 > low power consumption oscillation 3.

(Cautions and Remark are listed on the next page.)

Caution 1. **The CMC register can be written only once by an 8-bit memory manipulation instruction after release from the reset state. Even if you intend to use the CMC register with its initial value (00H), be sure to write 00H to the register after release from the reset state as a precaution against malfunctions (since returning the value to 00H after erroneously having written a value other than 00H to it is not possible).**

Caution 2. **After release from the reset state, set the CMC register before X1 or XT1 oscillation is started by setting the clock operation status control register (CSC).**

Caution 3. **Be sure to set the AMPH bit to 1 if the X1 clock oscillation frequency exceeds 10 MHz.**

Caution 4. **Make the settings of the AMPH, AMPHS1, and AMPHS0 bits while f_{IH} is selected as f_{CLK} after release from the reset state (before f_{CLK} is switched to f_{MX} or f_{SUB}).**

Caution 5. **Count the oscillation stabilization time of f_{XT} by software.**

Caution 6. **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.**
- **When using low power consumption oscillation 2 (AMPHS1 and AMPHS0 = 10B) or low power consumption oscillation 3 (AMPHS1 and AMPHS0 = 11B) as the XT1 oscillator mode, sufficiently evaluate the resonator as described in 6.7 Resonator and Oscillator Constants, before using it in either of these modes.**
- **Make the wiring runs between the XT1 and XT2 pins and the resonator as short as possible to minimize the parasitic capacitance and wiring resistance. Take care with this particularly when low power consumption oscillation 2 (AMPHS1 and AMPHS0 = 10B) or low power consumption oscillation 3 (AMPHS1 and AMPHS0 = 11B) 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 voltage as VSS as much as possible near the XT1 oscillator.**
- **Be sure that the signal lines between the XT1 and XT2 pins and the resonator do not cross the other signal lines. Do not route the wiring near a signal line through which a strong 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.**

Caution 7. **When selecting the XT1 oscillation mode for the P121/X1/XT1 and P122/X2/EXCLK/XT2/EXCLKS pins while the value of the XTSEL bit is 1 in a 30- to 36-pin R7F100G product, make sure that V_{DD} is no less than 2.4 V.**

Remark fx: X1 clock oscillation frequency

6.3.2 System clock control register (CKC)

The CKC register is used to select a CPU/peripheral hardware clock and a main system clock. The CKC register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 6 - 3 Format of System Clock Control Register (CKC)

Address: FFFA4H
After reset: 00H
R/W: R/W**Note 1**

Symbol	<7>	<6>	<5>	<4>	3	2	<1>	<0>
CKC	CLS	CSS	MCS	MCM0	0	0	MCS1	MCM1

CLS	State of the CPU/peripheral hardware clock (f_{CLK})
0	Main system clock (f_{MAIN})
1	Subsystem clock (f_{SUB})

CSS**Note 2**	Selection of the CPU/peripheral hardware clock (f_{CLK})
0	Main system clock (f_{MAIN})
1	Subsystem clock (f_{SUB})

MCS	State of the main system clock (f_{MAIN})
0	Main on-chip oscillator clock (f_{OCO})
1	High-speed system clock (f_{MX})

MCM0 **Note 2**	Main system clock (f_{MAIN}) operation control
0	Selects the main on-chip oscillator clock (f_{OCO}) as the main system clock (f_{MAIN})
1	Selects the high-speed system clock (f_{MX}) as the main system clock (f_{MAIN})

MCS1	State of the main on-chip oscillator clock (f_{OCO})
0	High-speed on-chip oscillator clock
1	Middle-speed on-chip oscillator clock

MCM1 **Note 2**	Main on-chip oscillator clock (f_{OCO}) operation control
0	High-speed on-chip oscillator clock
1	Middle-speed on-chip oscillator clock

Note 1. Bits 7, 5, and 1 are only readable.
Note 2. Changing the value of the MCM1 and MCM0 bits is prohibited while the CSS bit is set to 1.

(Cautions and Remark are listed on the next page.)

Caution 1. Be sure to leave bits 3 and 2 set to 0.

Caution 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 realtime clock, clock output/buzzer output, 32-bit interval timer, and watchdog timer) is also changed at the same time. Consequently, stop each peripheral function when changing the CPU/peripheral hardware clock.

Caution 3. If the subsystem clock is used as the peripheral hardware clock, the operations of the A/D converter and serial interface IICA are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the sections describing each item of peripheral hardware as well as Section 37 Electrical Characteristics.

Caution 4. When switching the CPU clock between the main system clock and subsystem clock, accessing the data flash memory is prohibited until the CLS bit changes after a clock is selected by the CSS bit.

Caution 5. When switching the CPU clock between the main system clock and subsystem clock, executing the HALT or STOP instruction is prohibited until the CLS bit changes after a clock is selected by the CSS bit.

Remark f_{IH}: High-speed on-chip oscillator clock frequency (32 MHz max.)
f_{MX}: High-speed system clock frequency
f_{MAIN}: Main system clock frequency
f_{SUB}: Subsystem clock frequency
f_{OCO}: Main on-chip oscillator clock frequency (f_{IH} or f_{IM})

6.3.3 Clock operation status control register (CSC)

The CSC register is used to control the operations of the high-speed system clock, high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock, and subsystem clock (except the low-speed on-chip oscillator clock). The CSC register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is C0H.

Figure 6 - 4 Format of Clock Operation Status Control Register (CSC)

Address: FFFA1H
After reset: C0H
R/W: R/W

Symbol	<7>	<6>	5	4	3	2	<1>	<0>
CSC	MSTOP	XTSTOP **Note 1**	0	0	0	0	MIOEN	HIOSTOP

MSTOP	High-speed system clock operation control		
	X1 oscillation mode	External clock input mode	Port mode
0	The X1 oscillator runs.	An external clock signal on the EXCLK pin is effective.	I/O port
1	The X1 oscillator is stopped.	An external clock signal on the EXCLK pin is ineffective.	

XTSTOP **Note 2**	Subsystem clock operation control		
	XT1 oscillation mode	External clock input mode	Port mode
0	The XT1 oscillator runs.	An external clock signal on the EXCLKS pin is effective.	Input port
1	The XT1 oscillator is stopped.	An external clock signal on the EXCLKS pin is ineffective.	

MIOEN	Middle-speed on-chip oscillator clock operation control
0	The middle-speed on-chip oscillator is stopped.
1	The middle-speed on-chip oscillator runs.

HIOSTOP	High-speed on-chip oscillator clock operation control
0	The high-speed on-chip oscillator runs.
1	The high-speed on-chip oscillator is stopped.

Note 1. Be sure to leave the XTSTOP bit set to 1 in the 32-pin R7F104G product.

Note 2. The XTSTOP bit is only initialized by a power-on reset; it retains its value when a reset caused by another source occurs.

(Cautions are listed on the next page.)

Caution 1. After release from the reset state, set the clock operation mode control register (CMC) before setting the CSC register.

Caution 2. Set the oscillation stabilization time select register (OSTS) before setting the MSTOP bit to 0 after release from the reset state. Note that if the OSTS register is being used with its default settings, the OSTS register is not required to be set here.

Caution 3. When starting X1 oscillation by using the MSTOP bit, check the oscillation stabilization time of the X1 clock by using the oscillation stabilization time counter status register (OSTC).

Caution 4. When starting XT1 oscillation by using the XTSTOP bit, include code to wait for oscillation of the subsystem clock to become stable.

Caution 5. Do not stop the clock selected for the CPU/peripheral hardware clock (f_{CLK}) with the CSC register.

Caution 6. When stopping the clock, confirm the conditions before clock oscillation is stopped. For details on how to stop the clock, see Table 6 - 8 Conditions Before the Clock Oscillation Is Stopped and Flag Settings.

6.3.4 Oscillation stabilization time counter status register (OSTC)

The OSTC register indicates the counter value by the X1 clock oscillation stabilization time counter. The X1 clock oscillation stabilization time can be checked in the following cases:

- If the X1 clock starts oscillation while the main on-chip oscillator clock or subsystem clock is in use as the CPU clock.
- If entry to and then release from the STOP mode proceed while the main on-chip oscillator clock is in use as the CPU clock and the X1 clock is oscillating.

The OSTC register can be read by a 1-bit or 8-bit memory manipulation instruction. The value of this register is 00H following a reset, STOP instruction, or the MSTOP bit of the clock operation status control register (CSC) being set to 1.

Remark The oscillation stabilization time counter starts counting in the following cases.

- When oscillation of the X1 clock starts (EXCLK = 0 and OSCSEL = 1 in the CMC register → MSTOP = 0)
- When the STOP mode is released

Figure 6 - 5 Format of Oscillation Stabilization Time Counter Status Register (OSTC)

Address: FFFA2H
After reset: 00H
R/W: R

Symbol	7	6	5	4	3	2	1	0
OSTC	MOST8	MOST9	MOST10	MOST11	MOST13	MOST15	MOST17	MOST18

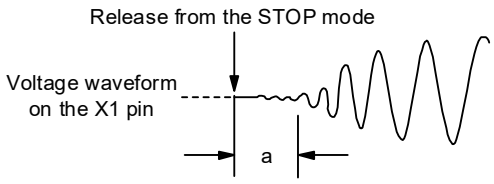
MOST8	MOST9	MOST10	MOST11	MOST13	MOST15	MOST17	MOST18	State of the oscillation stabilization time		
									fx = 10 MHz	fx = 20 MHz
0	0	0	0	0	0	0	0	Less than 2^8/fx	Less than 25.6 μs	Less than 12.8 μs
1	0	0	0	0	0	0	0	2^8/fx min.	25.6 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	2^9/fx min.	51.2 μs min.	25.6 μs min.
1	1	1	0	0	0	0	0	2^{10}/fx min.	102 μs min.	51.2 μs min.
1	1	1	1	0	0	0	0	2^{11}/fx min.	204 μs min.	102 μs min.
1	1	1	1	1	0	0	0	2^{13}/fx min.	819 μs min.	409 μs min.
1	1	1	1	1	1	0	0	2^{15}/fx min.	3.27 ms min.	1.63 ms min.
1	1	1	1	1	1	1	0	2^{17}/fx min.	13.1 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	2^{18}/fx min.	26.2 ms min.	13.1 ms min.

Caution 1. After the above time has elapsed, the bits are set to 1 in order from the MOST8 bit and remain 1.

Caution 2. The value counted by the OSTC register will only have reached the oscillation stabilization time setting in 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 counter value which is to be checked by using the OSTC register.

- **If the X1 clock starts oscillation while the main on-chip oscillator clock or subsystem clock is in use as the CPU clock.**
- **If entry to and then release from the STOP mode proceed while the main on-chip oscillator clock is in use as the CPU clock and the X1 clock is oscillating.**
  Therefore, note that the value counted by the OSTC register will only have reached the oscillation stabilization time setting in the OSTS register after release from the STOP mode.

Caution 3. The X1 clock oscillation stabilization time does not include the time until clock oscillation starts ("a" below).



Remark fx: X1 clock oscillation frequency

6.3.5 Oscillation stabilization time select register (OSTS)

The OSTS register is used to select the X1 clock oscillation stabilization time. When the X1 clock is made to oscillate by clearing the MSTOP bit to start operation of the X1 oscillator, actual operation is automatically delayed for the time set in the OSTS register. Use the oscillation stabilization time counter status register (OSTC) to confirm that the specified oscillation stabilization time has elapsed when the CPU clock is switched from the main on-chip oscillator clock or the subsystem clock to the X1 clock or entry to and then release from the STOP mode proceed while the main on-chip oscillator clock is in use as the CPU clock and the X1 clock is oscillating. The OSTC register can be used to check the counter value when counting has reached the time set beforehand in the OSTS register. The OSTS register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 07H.

Figure 6 - 6 Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFFA3H
After reset: 07H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	Selection of the oscillation stabilization time		
				fx = 10 MHz	fx = 20 MHz
0	0	0	2^8/fx	25.6 μs	12.8 μs
0	0	1	2^9/fx	51.2 μs	25.6 μs
0	1	0	2^{10}/fx	102 μs	51.2 μs
0	1	1	2^{11}/fx	204 μs	102 μs
1	0	0	2^{13}/fx	819 μs	409 μs
1	0	1	2^{15}/fx	3.27 ms	1.63 ms
1	1	0	2^{17}/fx	13.1 ms	6.55 ms
1	1	1	2^{18}/fx	26.2 ms	13.1 ms

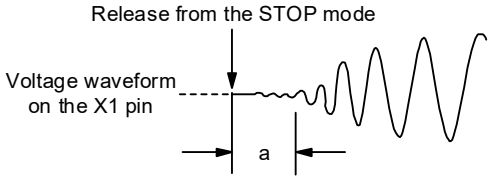
Caution 1. Change the setting of the OSTS register before setting the MSTOP bit of the clock operation status control register (CSC) to 0.

Caution 2. The value counted by the OSTC register will only have reached the oscillation stabilization time setting in the OSTS register. In the following cases, set the oscillation stabilization time of the OSTS register to the value greater than the counter value which is to be checked by using the OSTC register after the oscillation starts.

- **If the X1 clock starts oscillation while the main on-chip oscillator clock or subsystem clock is in use as the CPU clock.**
- **If entry to and then release from the STOP mode proceed while the main on-chip oscillator clock is in use as the CPU clock and the X1 clock is oscillating.**
  Therefore, note that the value counted by the OSTC register will only have reached the oscillation stabilization time setting in the OSTS register after release from the STOP mode.

(Caution and Remark are listed on the next page.)

Caution 3. The X1 clock oscillation stabilization time does not include the time until clock oscillation starts (“a” below).



Remark fx: X1 clock oscillation frequency

6.3.6 Peripheral enable registers 0, 1 (PER0, PER1)

The PER0 and PER1 registers are used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. To use any of the on-chip peripheral modules listed below, the clock supplies to which are controlled by these registers, set the corresponding bit to 1 before making the initial settings of the on-chip peripheral module.

- Realtime clock
- Serial interface IICA
- A/D converter
- Serial array unit
- Timer array unit
- D/A converter
- SNOOZE mode sequencer
- Comparator
- 32-bit interval timer
- DTC
- Serial interface UARTA
- Remote control signal receiver
- Capacitive sensing unit

The PER0 and PER1 registers can be set by a 1-bit or 8-bit memory manipulation instruction. The value of each of these registers following a reset is 00H.

Figure 6 - 7 Format of Peripheral Enable Register 0 (PER0) (1/2)

Address: F00F0H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCWEN	IICA1EN Note 1	ADCEN	IICA0EN Note 2	SAU1EN	SAU0EN	TAU1EN Note 3	TAU0EN

RTCWEN	Control of access to the realtime clock
0	• The SFRs used by the realtime clock cannot be written. • The realtime clock can operate.
1	• The SFRs used by the realtime clock can be read and written. • The realtime clock can operate.

IICA1EN	Control of supply of an input clock to the IICA1 serial interface
0	Stops supply of an input clock. • The SFRs used by the IICA1 serial interface cannot be written.Note 4
1	Enables supply of an input clock. • The SFRs used by the IICA1 serial interface can be read and written.

ADCEN	Control of supply of an input clock to the A/D converter
0	Stops supply of an input clock. • The SFRs used by the A/D converter cannot be written.Note 4
1	Enables supply of an input clock. • The SFRs used by the A/D converter can be read and written.

IICA0EN	Control of supply of an input clock to the IICA0 serial interface
0	Stops supply of an input clock. • The SFRs used by the IICA0 serial interface cannot be written.Note 4
1	Enables supply of an input clock. • The SFRs used by the IICA0 serial interface can be read and written.

SAU1EN	Control of supply of an input clock to serial array unit 1
0	Stops supply of an input clock. • The SFRs used by serial array unit 1 cannot be written.Note 4
1	Enables supply of an input clock. • The SFRs used by serial array unit 1 can be read and written.

SAU0EN	Control of supply of an input clock to serial array unit 0
0	Stops supply of an input clock. • The SFRs used by serial array unit 0 cannot be written.Note 4
1	Enables supply of an input clock. • The SFRs used by serial array unit 0 can be read and written.

Figure 6 - 7 Format of Peripheral Enable Register 0 (PER0) (2/2)

TAU1EN	Control of supply of an input clock to timer array unit 1
0	Stops supply of an input clock. • The SFRs used by timer array unit 1 cannot be written.Note 4
1	Enables supply of an input clock. • The SFRs used by timer array unit 1 can be read and written.

TAU0EN	Control of supply of an input clock to timer array unit 0
0	Stops supply of an input clock. • The SFRs used by timer array unit 0 cannot be written.Note 4
1	Enables supply of an input clock. • The SFRs used by timer array unit 0 can be read and written.

Note 1. This bit is only present in the 44- to 128-pin R7F100G products.

Note 2. This bit is only present in the 30- to 128-pin R7F100G and 48-pin R7F104G products.

Note 3. This bit is only present in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products.

Note 4. When an SFR is read, the value returned is 00H or 0000H regardless of its setting.

Caution 1. Be sure to leave the following bits set to 0.
Bits 6 and 1 in the 30- to 40-pin R7F100G products
Bit 1 in the 44- to 64-pin R7F100G products
Bits 6, 4, and 1 in the 32-pin R7F104G product
Bits 6 and 4 in the 44-pin R7F104G product
Bit 6 in the 48-pin R7F104G product

Caution 2. Do not change the value of a bit of the PER0 register while operation of the corresponding on-chip peripheral module is enabled. Only change a value while the corresponding on-chip peripheral module is stopped.

Figure 6 - 8 Format of Peripheral Enable Register 1 (PER1) (1/2)

Address: F00FAH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER1	DACEN	SMSEN	CMPEN	TML32EN	DTCEN	UTAEN Note 1	REMCEN Note 2	CTSUEN

DACEN	Control of supply of an input clock to the D/A converter
0	Stops supply of an input clock. • The SFRs used by the D/A converter cannot be written.Note 3
1	Enables supply of an input clock. • The SFRs used by the D/A converter can be read and written.

SMSEN	Control of supply of an input clock to the SNOOZE mode sequencer
0	Stops supply of an input clock. • The SFRs used by the SNOOZE mode sequencer cannot be written.Note 3
1	Enables supply of an input clock. • The SFRs used by the SNOOZE mode sequencer can be read and written.

CMPEN	Control of supply of an input clock to the comparator
0	Stops supply of an input clock. • The SFRs used by the comparator cannot be written.Note 3
1	Enables supply of an input clock. • The SFRs used by the comparator can be read and written.

TML32EN	Control of supply of an input clock to the 32-bit interval timer
0	Stops supply of an input clock. • The SFRs used by the 32-bit interval timer cannot be written.Note 3
1	Enables supply of an input clock. • The SFRs used by the 32-bit interval timer can be read and written.

DTCEN	Control of supply of an input clock to the DTC
0	Stops supply of an input clock. • The DTC cannot operate.
1	Enables supply of an input clock. • The DTC can operate.

UTAEN	Control of supply of an input clock to the UARTAn serial interface (n = 0, 1)
0	Stops supply of an input clock. • The SFRs used by the UARTAn serial interface cannot be written.Note 3
1	Enables supply of an input clock. • The SFRs used by the UARTAn serial interface can be read and written.

Figure 6 - 8 Format of Peripheral Enable Register 1 (PER1) (2/2)

REMCEN	Control of supply of an input clock to the remote control signal receiver
0	Stops supply of an input clock. • The SFRs used by the remote control signal receiver cannot be written.Note 3
1	Enables supply of an input clock. • The SFRs used by the remote control signal receiver can be read and written.

CTSUEN	Control of supply of an input clock to the capacitive sensing unit
0	Stops supply of an input clock. • The SFRs used by the capacitive sensing unit cannot be written.Note 3
1	Enables supply of an input clock. • The SFRs used by the capacitive sensing unit can be read and written.

Note 1. This bit is only present in the 36- to 128-pin R7F100G and 32- to 48-pin R7F104G products.

Note 2. This bit is only present in the 32- to 128-pin R7F100G and 32- to 48-pin R7F104G products.

Note 3. When an SFR is read, the value returned is 00H or 0000H regardless of its setting.

Caution 1. Be sure to leave the following bits set to 0.
Bits 2 and 1 in the 30-pin R7F100G products
Bit 2 in the 32-pin R7F100G products

Caution 2. Do not change the value of a bit of the PER1 register while operation of the corresponding on-chip peripheral module is enabled. Only change a value while the corresponding on-chip peripheral module is stopped.

6.3.7 Subsystem clock supply mode control register (OSMC)

The OSMC register is used to reduce power consumption by stopping clock functions that are not required. If the RTCLPC bit is set to 1, power consumption can be reduced, because clock supply to the peripheral functions except the realtime clock is stopped in STOP mode or in HALT mode while the CPU is operating with the subsystem clock. The OSMC register can also be used to select the operating clock for the realtime clock, clock output/buzzer output controller, 32-bit interval timer, serial interfaces UARTA0 and UARTA1, and remote control signal receiver. The OSMC register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is undefined**Note**.

Note The values of the RTCLPC, WUTMMCK0, and HIPREC bits are respectively 0, 0, and 1.

Figure 6 - 9 Format of Subsystem Clock Supply Mode Control Register (OSMC)

Address: F00F3H
After reset: Undefined**Note 1**
R/W: R/W**Note 2**

Symbol	<7>	6	5	<4>	3	2	1	<0>
OSMC	RTCLPC	0	0	WUTMMCK0	×	×	0	HIPREC

RTCLPC **Note 3**	Setting in STOP mode or in HALT mode while the CPU is operating with subsystem clock X
0	Enables supply of subsystem clock X to peripheral functions. (See **Table 23 - 1 Operating States in HALT Mode When the Main System Clock Is Used as the CPU Clock** to **Table 23 - 4 Operating States in SNOOZE Mode** for peripheral functions whose operations are enabled.)
1	Stops supply of the subsystem clock to peripheral functions other than the realtime clock.

WUTMMCK0	Selection of the operating clock for the realtime clock, 32-bit interval timer, serial interfaces UARTA0 and UARTA1, remote control signal receiver, and clock output/buzzer output controller
0	Subsystem clock X
1	Low-speed on-chip oscillator clock**Notes 4, 5**

HIPREC	State of the high-speed on-chip oscillator clock**Notes 6, 7**
0	The high-speed on-chip oscillator clock is being started at high speed and waiting for the precision of its oscillation to become stable is in progress.**Note 8**
1	The high-speed on-chip oscillator clock is operating with high precision.

Note 1. The values of the RTCLPC, WUTMMCK0, and HIPREC bits are respectively 0, 0, and 1.
Note 2. Bits 3, 2, and 0 are only readable. Writing to these bits is ignored.
Note 3. When the subsystem clock X is selected (SELLOSC = 0) by the SELLOSC bit of the CKSEL register and RTCLPC is set to 1, the subsystem clock (f_{SUB}) is stopped. However, when the low-speed on-chip oscillator clock is selected (SELLOSC = 1) and RTCLPC is set to 1, the subsystem clock (f_{SUB}) is not stopped.
Note 4. Do not set the WUTMMCK0 bit to 1 while the subsystem clock X is oscillating.
Note 5. Switching between the subsystem clock X and the low-speed on-chip oscillator clock can be enabled by the WUTMMCK0 bit only when all of the realtime clock, 32-bit interval timer, serial interfaces UARTA0 and UARTA1, remote control signal receiver, and clock output/buzzer output function are stopped.
Note 6. Undefined while the high-speed on-chip oscillator is stopped.

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

Note 7. For frequency accuracy of the high-speed on-chip oscillator clock, see **Section 37 Electrical Characteristics**.

Note 8. On release from the STOP mode while the setting for starting the high-speed on-chip oscillator at high speed is in place (WKUPMD.FWKUP = 1), the high-speed on-chip oscillator clock runs at low precision while it is starting up. After the oscillation accuracy stabilization time, the value of the HIPREC bit changes automatically to 1. The table below shows the frequency of the high-speed on-chip oscillator when FRQSEL3 = 0 and HIPREC = 0.

Setting of FRQSEL2 or HOCODIV2	Setting of FRQSEL1 or HOCODIV1	Setting of FRQSEL0 or HOCODIV0	Frequency of the High-speed On-chip Oscillator
0	0	0	16 MHz
0	0	1	8 MHz
0	1	0	4 MHz
0	1	1	2 MHz
1	×	×	Setting prohibited

Caution 1. Be sure to leave bits 6, 5, and 1 set to 0.

Caution 2. Do not execute a STOP instruction when HIPREC = 0.

Remark ×: Don’t care

6.3.8 Subsystem clock select register (CKSEL)

The CKSEL register is used to select the subsystem clock X or low-speed on-chip oscillator clock as the subsystem clock. The CKSEL register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 6 - 10 Format of Subsystem Clock Select Register (CKSEL)

Address: FFFA7H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	<0>
CKSEL	0	0	0	0	0	0	0	SELLOSC

SELLOSC	Selection of the subsystem clock X or low-speed on-chip oscillator clock
0	Subsystem clock X
1	Low-speed on-chip oscillator clock[Note]

Note Do not set SELLOSC to 1 when the subsystem clock X (f_{SX}) or XR (f_{SXR}) is operating.

Caution When changing the value of the SELLOSC bit, be sure to set the CSS bit to 0 (selecting f_{MAIN}) and confirm that the value of the CLS bit is 0 before doing so.

6.3.9 High-speed on-chip oscillator frequency select register (HOCODIV)

The HOCODIV register is used to change the frequency of the high-speed on-chip oscillator set in a user option byte (000C2H). Note that the selectable frequencies depend on the value of the FRQSEL3 bit of the user option byte (000C2H). The HOCODIV register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is that set in the FRQSEL2 to FRQSEL0 bits of the user option byte (000C2H).

Figure 6 - 11 Format of High-speed On-chip Oscillator Frequency Select Register (HOCODIV)

Address: F00A8H
After reset: The value set in the FRQSEL2 to FRQSEL0 bits of a user option byte (000C2H)
R/W: 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 the high-speed on-chip oscillator clock frequency	
			FRQSEL3 = 0	FRQSEL3 = 1
0	0	0	f_{IH} = 24 MHz	f_{IH} = 32 MHz
0	0	1	f_{IH} = 12 MHz	f_{IH} = 16 MHz
0	1	0	f_{IH} = 6 MHz	f_{IH} = 8 MHz
0	1	1	f_{IH} = 3 MHz	f_{IH} = 4 MHz
1	0	0	Setting prohibited	f_{IH} = 2 MHz
1	0	1	Setting prohibited	f_{IH} = 1 MHz
Other than above			Setting prohibited	

Caution 1. Set the HOCODIV register while ensuring that the voltage is within the usable range for the operating mode of the flash memory set in the flash operating mode select register (FLMODE) both before and after the frequency change. For details about the FLMODE register, see 5.2.1 Flash operating mode select register (FLMODE).

Caution 2. Set the HOCODIV register with the high-speed on-chip oscillator clock (f_{IH}) selected as the CPU/peripheral hardware clock (f_{CLK}).

Caution 3. After use of the HOCODIV register to change the frequency, actually switching of the frequency only proceeds after the following transition times have elapsed.

- **The CPU/peripheral hardware clock continues to operate at the frequency before the change for up to three cycles.**
- **Up to three cycles of waiting are required for the CPU/peripheral hardware clock to be at the post-change frequency.**

Note that even if a value set in the HOCODIV register is the same as its current value, up to three cycles of waiting following the setting are required for stable operation of the CPU/peripheral hardware clock.

6.3.10 Middle-speed on-chip oscillator frequency select register (MOCODIV)

The MOCODIV register is used to select the frequency of the middle-speed on-chip oscillator. The MOCODIV register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 6 - 12 Format of Middle-speed On-chip Oscillator Frequency Select Register (MOCODIV)

Address: F00F2H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
MOCODIV	0	0	0	0	0	0	MOCODIV1	MOCODIV0

MOCODIV1	MOCODIV0	Selection of the middle-speed on-chip oscillator clock frequency
0	0	4 MHz
0	1	2 MHz
1	0	1 MHz
1	1	Setting prohibited

Caution Set the MOCODIV register while ensuring that the voltage is within the usable range for the operating mode of the flash memory set in the flash operating mode select register (FLMODE) both before and after the frequency change.

6.3.11 High-speed system clock division register (MOSCDIV)

The MOSCDIV register is used to select the division ratio of the high-speed system clock. The MOSCDIV register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 6 - 13 Format of High-speed System Clock Division Register (MOSCDIV)

Address: F0214H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
MOSCDIV	0	0	0	0	0	MOSCDIV2	MOSCDIV1	MOSCDIV0

MOSCDIV2	MOSCDIV1	MOSCDIV0	Selected division ratio for the high-speed system clock	f_{MX} = 20 MHz
0	0	0	f_{MX}	20 MHz
0	0	1	f_{MX}/2	10 MHz
0	1	0	f_{MX}/4	5 MHz
0	1	1	f_{MX}/8	2.5 MHz
1	0	0	f_{MX}/16	1.25 MHz
Other than above			Setting prohibited	

Caution Set the MOSCDIV register while ensuring that the voltage is within the usable range for the operating mode of the flash memory set in the flash operating mode select register (FLMODE) both before and after the frequency change.

6.3.12 High-speed on-chip oscillator trimming register (HIOTRM)

The HIOTRM register is used to adjust the accuracy of the high-speed on-chip oscillator. The accuracy of the high-speed on-chip oscillator frequency can be adjusted through self-measurement of the frequency by using a timer with high accuracy (timer array unit or 32-bit interval timer) for external clock input or in other ways. The HIOTRM register can be set 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 6 - 14 Format of High-speed On-chip Oscillator Trimming Register (HIOTRM)

Address: F00A0H
After reset: **Note**
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
HIOTRM	0	0	HIOTRM5	HIOTRM4	HIOTRM3	HIOTRM2	HIOTRM1	HIOTRM0

HIOTRM5	HIOTRM4	HIOTRM3	HIOTRM2	HIOTRM1	HIOTRM0	High-speed on-chip oscillator
0	0	0	0	0	0	Minimum speed
0	0	0	0	0	1	
0	0	0	0	1	0	
0	0	0	0	1	1	
0	0	0	1	0	0	
• • •						
1	1	1	1	1	0	
1	1	1	1	1	1	Maximum speed

Note The value of this register following a reset is that adjusted at shipment.

Remark 1. The HIOTRM register holds a six-bit value used to adjust the high-speed on-chip oscillator with an increment of 1 corresponding to an increase of frequency by about 0.05%.

Remark 2. For the usage example of the HIOTRM register, see the **Application Note for RL78/G23 High-speed On-chip Oscillator (HOCO) Clock Frequency Correction (R01AN5555)**.

6.3.13 Middle-speed on-chip oscillator trimming register (MIOTRM)

The MIOTRM register is used to adjust the accuracy of the middle-speed on-chip oscillator. The accuracy of the middle-speed on-chip oscillator frequency can be adjusted through self-measurement of the frequency by using a timer with high accuracy (timer array unit or 32-bit interval timer) for external clock input or in other ways. The MIOTRM register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 90H.

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 6 - 15 Format of Middle-speed On-chip Oscillator Trimming Register (MIOTRM)

Address: F0212H
After reset: 90H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
MIOTRM	MIOTRM7	MIOTRM6	MIOTRM5	MIOTRM4	MIOTRM3	MIOTRM2	MIOTRM1	MIOTRM0

MIOTRM 7	MIOTRM 6	MIOTRM 5	MIOTRM 4	MIOTRM 3	MIOTRM 2	MIOTRM 1	MIOTRM 0	Middle-speed on-chip oscillator
0	0	0	0	0	0	0	0	Minimum speed
0	0	0	0	0	0	0	1	↑
								↑
1	0	0	0	1	1	1	1	↑
1	0	0	1	0	0	0	0	Initial value
1	0	0	1	0	0	0	1	↓
								↓
1	1	1	1	1	1	1	0	↓
1	1	1	1	1	1	1	1	Maximum speed

Remark For details about the accuracy adjustment resolution of the middle-speed on-chip oscillator clock, see **Section 37 Electrical Characteristics**.

6.3.14 Low-speed on-chip oscillator trimming register (LIOTRM)

The LIOTRM register is used to adjust the accuracy of the low-speed on-chip oscillator. The accuracy of the low-speed on-chip oscillator frequency can be adjusted through self-measurement of the frequency by using a timer with high accuracy (timer array unit or 32-bit interval timer) for external clock input or in other ways. The LIOTRM register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 80H.

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 6 - 16 Format of Low-speed On-chip Oscillator Trimming Register (LIOTRM)

Address: F0213H
After reset: 80H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
LIOTRM	LIOTRM7	LIOTRM6	LIOTRM5	LIOTRM4	LIOTRM3	LIOTRM2	LIOTRM1	LIOTRM0

LIOTRM7	LIOTRM6	LIOTRM5	LIOTRM4	LIOTRM3	LIOTRM2	LIOTRM1	LIOTRM0	Low-speed on-chip oscillator
0	0	0	0	0	0	0	0	Minimum speed
0	0	0	0	0	0	0	1	↑
0	1	1	1	1	1	1	1	
1	0	0	0	0	0	0	0	Initial value
1	0	0	0	0	0	0	1	↓
1	1	1	1	1	1	1	0	
1	1	1	1	1	1	1	1	Maximum speed

Remark For details about the accuracy adjustment resolution of the low-speed on-chip oscillator clock, see **Section 37 Electrical Characteristics**.

6.3.15 Standby mode release setting register (WKUPMD)

The WKUPMD register is used to set the operation when the standby mode is released. The WKUPMD register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 6 - 17 Format of Standby Mode Release Setting Register (WKUPMD)

Address: F0215H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	<0>
WKUPMD	0	0	0	0	0	0	0	FWKUP

FWKUP	Setting for starting the high-speed on-chip oscillator at the times of release from STOP mode and of transitions to SNOOZE mode**Notes 1, 2**
0	Starting is at normal speed.**Note 3**
1	Starting is at high speed.**Note 3**

Note 1. This setting is only available when the high-speed on-chip oscillator is selected for the CPU clock.

Note 2. This register is initialized on release from STOP mode in response to the generation of a reset signal, so starting of the high-speed on-chip oscillator is at normal speed.

Note 3. For the activation time, see **Section 23 Standby Function**. The accuracy of the high-speed on-chip oscillator frequency depends on whether starting of the high-speed on-chip oscillator is at normal speed or at high speed. See **Section 37 Electrical Characteristics**.

6.4 System Clock Oscillator

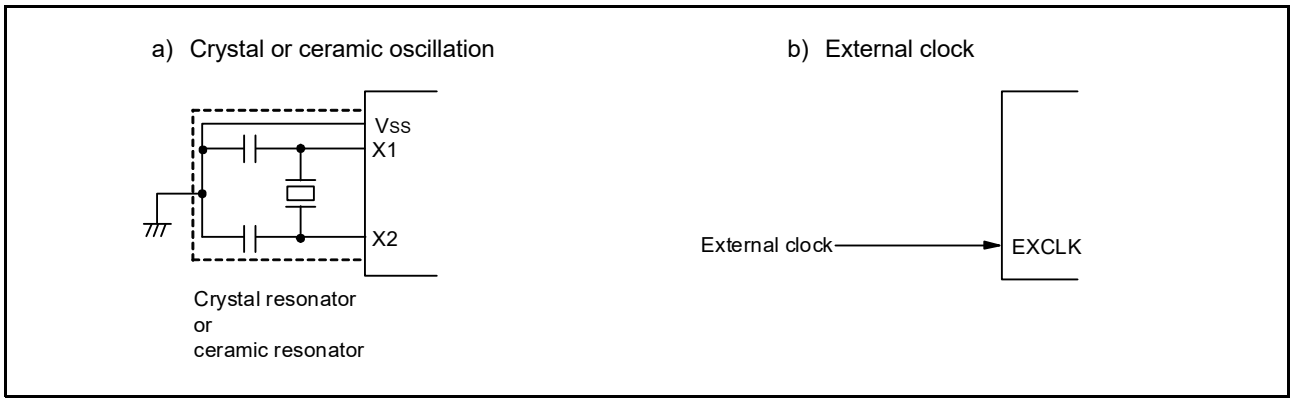
6.4.1 X1 oscillator

The X1 oscillator oscillates with a crystal resonator or ceramic resonator (1 to 20 MHz) connected to the X1 and X2 pins. An external clock can also be input. In this case, input the clock signal to the EXCLK pin. To use the X1 oscillator, set the EXCLK and OSCSEL bits of the clock operation mode control register (CMC) as follows.

- Crystal or ceramic oscillation: EXCLK and OSCSEL = 01B
- External clock input: EXCLK and OSCSEL = 11B

When the X1 oscillator is not used, set the port mode (EXCLK and OSCSEL = 00B). When the pins are not used as port pins, either, see **Table 2 - 2 Connections of Unused Pins**. **Figure 6 - 18** shows the examples of external circuits for the X1 oscillator.

Figure 6 - 18 Examples of External Circuits for the X1 Oscillator



Caution When using the X1 oscillator, wire as follows in the area enclosed in broken lines in Figure 6 - 18 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 strong fluctuating current flows.**
- **Always apply the same voltage to the ground point for capacitors in the oscillator as that on Vss. Do not ground the capacitors to a ground pattern through which a strong current flows.**
- **Do not bring a signal line out from the oscillator.**

6.4.2 XT1 oscillator

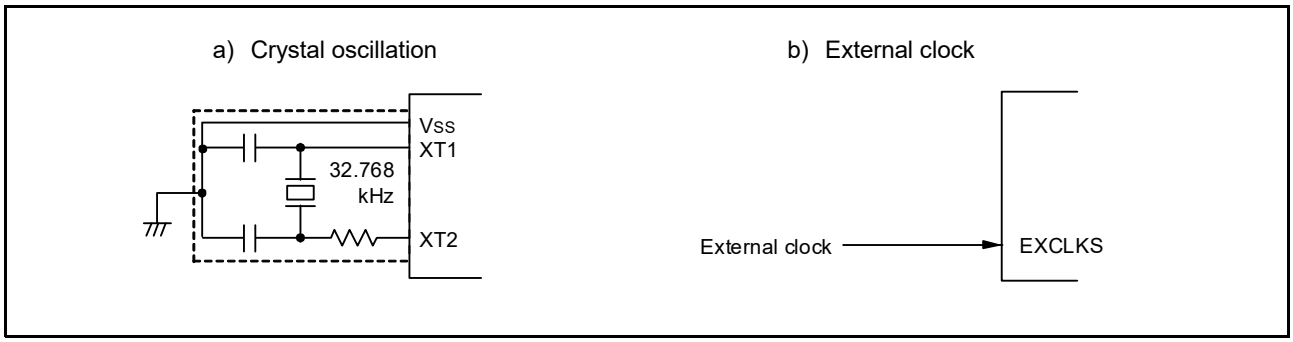
The XT1 oscillator oscillates with a crystal resonator (32.768 kHz (typ.)) connected to the XT1 and XT2 pins. To use the XT1 oscillator, set the OSCSELS bit of the clock operation mode control register (CMC) to 1. An external clock can also be input. In this case, input the clock signal to the EXCLKS pin. To use the XT1 oscillator, set the EXCLKS and OSCSELS bits of the clock operation mode control register (CMC) as follows.

- Crystal oscillation: EXCLKS and OSCSELS = 01B
- External clock input: EXCLKS and OSCSELS = 11B

When the XT1 oscillator is not used, set the input port mode (EXCLKS and OSCSELS = 00B). When XT1 oscillator is not used, and the pins are not used as input port pins, either, see **Table 2 - 2 Connections of Unused Pins**.

Figure 6 - 19 shows the examples of external circuits for the XT1 oscillator.

Figure 6 - 19 Examples of External Circuits for the XT1 Oscillator



Caution 1. When using the XT1 oscillator, wire as follows in the area enclosed in broken lines in Figure 6 - 19 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 strong fluctuating current flows.**
- **Always apply the same voltage to the ground point for capacitors in the oscillator as that on Vss. Do not ground the capacitors to a ground pattern through which a strong current flows.**
- **Do not bring a signal line out from the oscillator.**

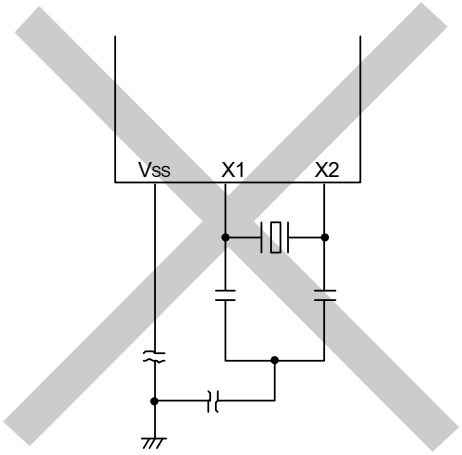
Caution 2. The XT1 oscillator is a circuit with low amplification in order to achieve low power consumption. Note the following points when designing the circuit.

- **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.**
- **When using low power consumption oscillation 2 (AMPHS1 and AMPHS0 = 10B) or low power consumption oscillation 3 (AMPHS1 and AMPHS0 = 11B) as the XT1 oscillator mode, sufficiently evaluate the resonator as described in 6.7 Resonator and Oscillator Constants, before using it in either of these modes.**
- **Make the wiring runs between the XT1 and XT2 pins and the resonator as short as possible to minimize the parasitic capacitance and wiring resistance. Take care with this particularly when low power consumption oscillation 2 (AMPHS1 and AMPHS0 = 10B) or low power consumption oscillation 3 (AMPHS1 and AMPHS0 = 11B) 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 voltage as Vss as much as possible near the XT1 oscillator.**
- **Be sure that the signal lines between the XT1 and XT2 pins and the resonator do not cross the other signal lines. Do not route the wiring near a signal line through which a strong 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.**

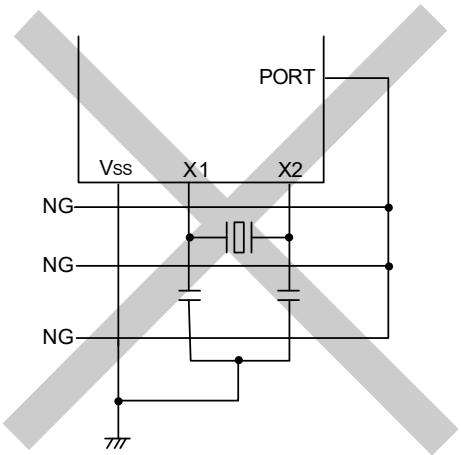
Figure 6 - 20 shows examples of incorrect resonator connection.

Figure 6 - 20 Examples of Incorrect Resonator Connection (1/2)

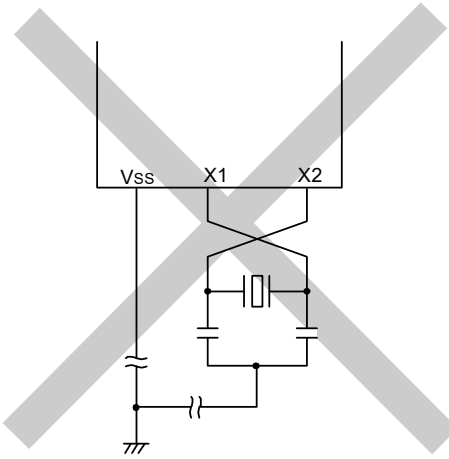
a) The wiring runs of the oscillator are too long.



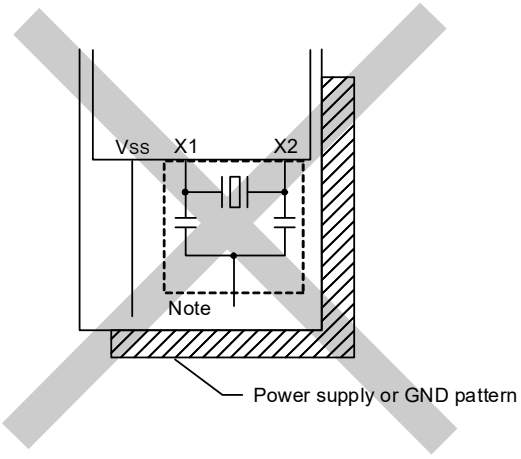
b) The wiring runs of the oscillator cross signal lines.



c) The wiring runs for the X1 and X2 signals cross each other.



d) The power supply or GND pattern is under the wiring runs for the X1 and X2 signals.

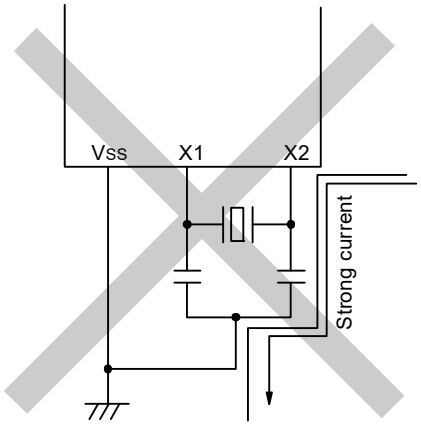


Note Do not place a power supply or GND pattern under the wiring section (section enclosed in broken lines in the figure) of the X1 and X2 pins and the resonator in a multi-layer board or double-sided board. Do not configure a layout that will cause capacitance elements and affect the oscillation characteristics.

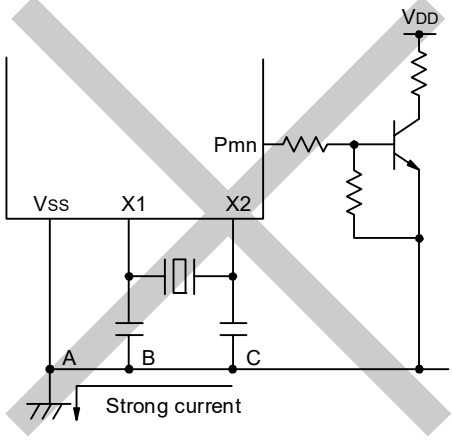
Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert a resistor in series on the XT2 side.

Figure 6 - 20 Examples of Incorrect Resonator Connection (2/2)

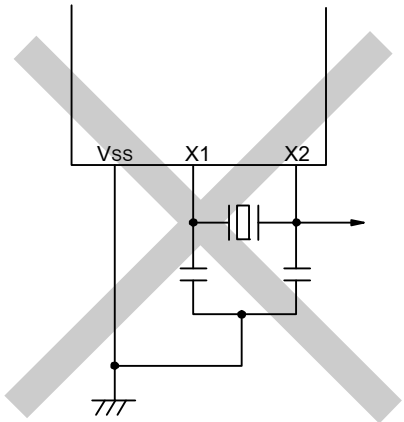
e) A signal line for the oscillator and a signal line through which a strong fluctuating current flows are close to each other.



f) Current flows through the ground line of the oscillator (making the voltage at points A, B, and C fluctuate).



g) A signal line is brought out.



Caution **When the wiring runs for the X2 and X1 signals are in parallel, crosstalk noise from the X2 signal line may be imposed on the X1 signal, resulting in malfunctions.**

Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert a resistor in series on the XT2 side.

6.4.3 High-speed on-chip oscillator

This MCU incorporates a high-speed on-chip oscillator. The frequency can be selected from among 32, 24, 16, 12, 8, 6, 4, 3, 2, and 1 MHz by using the user option byte (000C2H). Oscillation can be controlled by the HIOSTOP bit of the clock operation status control register (CSC). The high-speed on-chip oscillator automatically starts oscillating after release from the reset state.

6.4.4 Middle-speed on-chip oscillator

This MCU incorporates a middle-speed on-chip oscillator. Oscillation can be controlled by the MIOEN bit of the clock operation status control register (CSC).

6.4.5 Low-speed on-chip oscillator

This MCU incorporates a low-speed on-chip oscillator. The low-speed on-chip oscillator operates when any of the following conditions is met.

- The watchdog timer is operating.
- The value of one or both of the WUTMMCK0 bit of the subsystem clock supply mode control register (OSMC) and the SELLOSC bit of the subsystem clock select register (CKSEL) is 1.
- f_{IL} is selected as the source clock for use in waiting by the SNOOZE mode sequencer.

The low-speed on-chip oscillator is stopped when the watchdog timer is stopped and both WUTMMCK0 and SELLOSC are set to 0.

6.5 Operations of the Clock Generator

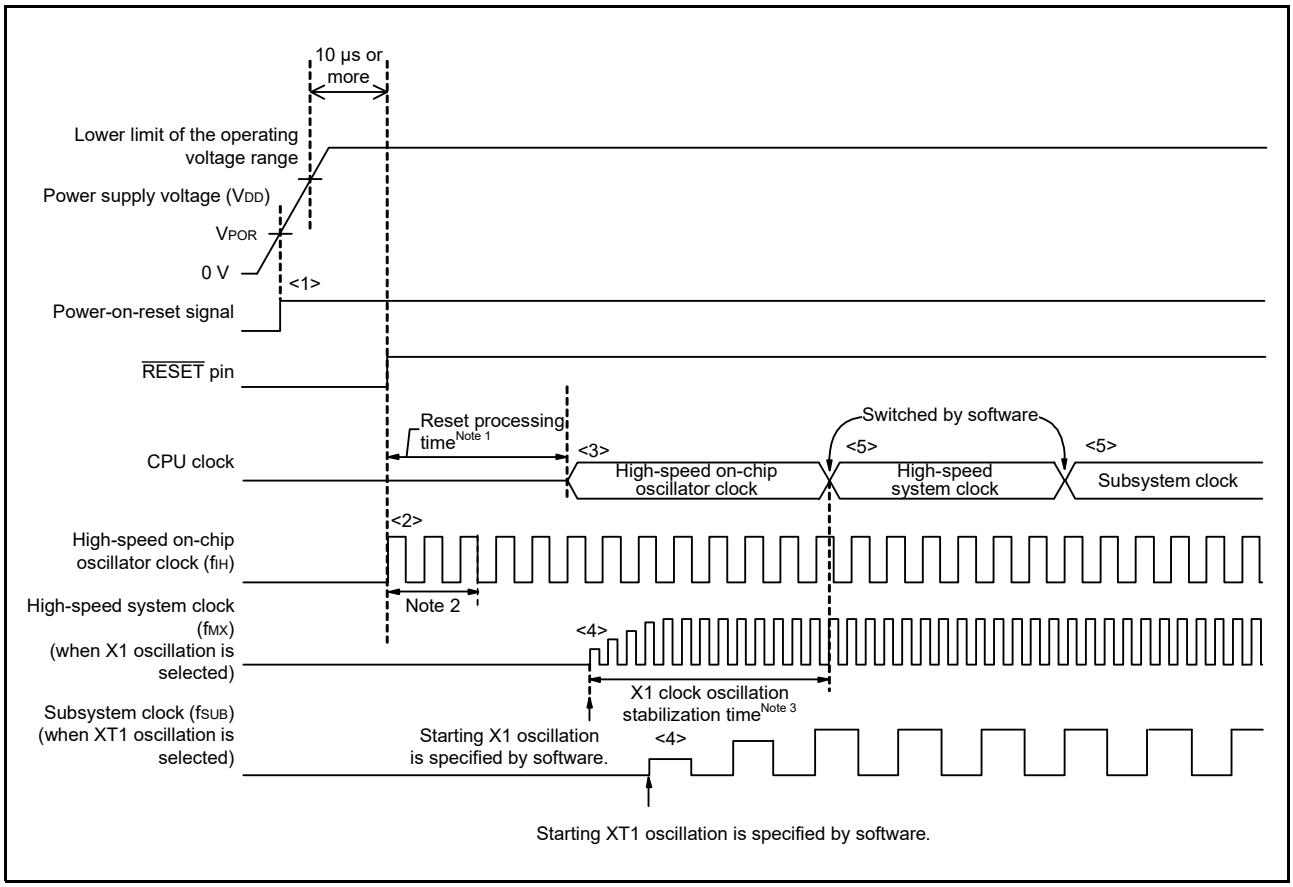
The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode (see **Figure 6 - 1 Block Diagram of Clock Generator**).

- Main system clocks (f_{MAIN})
  - High-speed system clocks (f_{MX})
    X1 clock (f_X)
    External main system clock (f_{EX})
  - High-speed on-chip oscillator clock (f_{IH})
  - Middle-speed on-chip oscillator clock (f_{IM})
- Subsystem clocks (f_{SUB})
  - XT1 clock (f_{XT})
  - External subsystem clock (f_{EXS})
  - Low-speed on-chip oscillator clock (f_{IL})
- CPU/peripheral hardware clock (f_{CLK})
- Subsystem clock X (f_{SX})
- Peripheral clocks
  - High-speed on-chip oscillator peripheral clock (f_{IHP})
  - Middle-speed on-chip oscillator peripheral clock (f_{IMP})
  - High-speed peripheral clock (f_{MXP})
  - Low-speed peripheral clock (f_{SXP})
  - Subsystem clock XR (f_{SXR})

In this MCU, operation of the CPU is driven by the high-speed on-chip oscillator clock after release from the reset state.

Figure 6 - 21 shows the clock generator operation when power supply voltage is turned on.

Figure 6 - 21 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 circuit. Note that the reset state is maintained after a reset by the voltage detection circuit or an external reset until the voltage reaches the range of operating voltage described in **37.4 AC Characteristics** (the above figure is an example when the external reset is in use).

<2> When the reset is released, the high-speed on-chip oscillator automatically starts oscillation.

<3> The CPU starts operation with the high-speed on-chip oscillator clock after waiting for the voltage to become stable and a reset processing have been performed after release from the reset state.

<4> Set the start of oscillation of the X1 or XT1 clock via software (see **6.6.2 Example of setting the X1 oscillator clock** or **6.6.3 Example of setting the XT1 oscillator clock**).

<5> When switching the CPU clock to the X1 or XT1 clock, wait for the clock oscillation to become stable, and then make the setting for switching via software (see **6.6.2 Example of setting the X1 oscillator clock** or **6.6.3 Example of setting the XT1 oscillator clock**).

Note 1. For the reset processing time, see **Section 25 Power-on-reset Circuit (POR)**.

Note 2. The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

Note 3. When releasing a reset, confirm the oscillation stabilization time for the X1 clock using the oscillation stabilization time counter status register (OSTC).

Caution Waiting for the oscillation stabilization time is not required when external clock input through the EXCLK pin is used.

6.6 Controlling Clocks

6.6.1 Example of setting the high-speed on-chip oscillator

After release from the reset state, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock as the source. The frequency of the high-speed on-chip oscillator can be selected from among 32, 24, 16, 12, 8, 6, 4, 3, 2, and 1 MHz by using the FRQSEL3 to FRQSEL0 bits of a user option byte (000C2H). In addition, the frequency can be changed by the high-speed on-chip oscillator frequency select register (HOCODIV).

[User option byte setting]

Address: 000C2H

	7	6	5	4	3	2	1	0
User option byte (000C2H)	CMODE1 0/1	CMODE0 0/1	1	0	FRQSEL3 0/1	FRQSEL2 0/1	FRQSEL1 0/1	FRQSEL0 0/1

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator
1	0	0	0	32 MHz
0	0	0	0	24 MHz
1	0	0	1	16 MHz
0	0	0	1	12 MHz
1	0	1	0	8 MHz
0	0	1	0	6 MHz
1	0	1	1	4 MHz
0	0	1	1	3 MHz
1	1	0	0	2 MHz
1	1	0	1	1 MHz
Other than above				Setting prohibited

[High-speed on-chip oscillator frequency select register (HOCODIV) setting]

Address: F00A8H

Symbol	7	6	5	4	3	2	1	0
HOCODIV	0	0	0	0	0	HOCODIV2	HOCODIV1	HOCODIV0

HOCODIV2	HOCODIV1	HOCODIV0	Selection of the high-speed on-chip oscillator clock frequency	
			FRQSEL3 = 0	FRQSEL3 = 1
0	0	0	f_{IH} = 24 MHz	f_{IH} = 32 MHz
0	0	1	f_{IH} = 12 MHz	f_{IH} = 16 MHz
0	1	0	f_{IH} = 6 MHz	f_{IH} = 8 MHz
0	1	1	f_{IH} = 3 MHz	f_{IH} = 4 MHz
1	0	0	Setting prohibited	f_{IH} = 2 MHz
1	0	1	Setting prohibited	f_{IH} = 1 MHz
Other than above			Setting prohibited	

6.6.2 Example of setting the X1 oscillator clock

After release from the reset state, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock as the source. To subsequently change the source clock to the X1 oscillator clock, set and start the oscillator by using the oscillation stabilization time select register (OSTS), clock operation mode control register (CMC), and clock operation status control register (CSC), and wait for the oscillation to become stable by using the oscillation stabilization time counter status register (OSTC). After the oscillation becomes stable, set the X1 oscillator clock as the source of f_{CLK} by using the system clock control register (CKC).

[Register settings] Set the registers in the order of <1> to <5> below.

<1> Set the OSCSEL bit of the CMC register to 1 to make the X1 oscillator operate. Also set the AMPH bit to 1 if f_X is more than 10 MHz.

	7	6	5	4	3	2	1	0
CMC	EXCLK 0	OSCSEL 1	EXCLKS 0	OSCSELS 0	XTSEL 0	AMPHS1 0	AMPHS0 0	AMPH 0/1

<2> Use the OSTS register to select the oscillation stabilization time for the X1 oscillator after release from the STOP mode.
Example: Set the register as shown below if waiting is to be for at least 102 μs with a 10-MHz resonator.

	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2 0	OSTS1 1	OSTS0 0

<3> Clear the MSTOP bit of the CSC register to 0 to start X1 oscillation.

	7	6	5	4	3	2	1	0
CSC	MSTOP 0	XTSTOP 1	0	0	0	0	MIOEN 0	HIOSTOP 0

<4> Use the OSTC register to wait for oscillation of the X1 oscillator to become stable.
Example: Wait until counting has reached the following value if waiting is to be for at least 102 μs with a 10-MHz resonator.

	7	6	5	4	3	2	1	0
OSTC	MOST8 1	MOST9 1	MOST10 1	MOST11 0	MOST13 0	MOST15 0	MOST17 0	MOST18 0

<5> Use the MCM0 bit of the CKC register to specify the X1 oscillator clock as the source of the CPU/peripheral hardware clock.

	7	6	5	4	3	2	1	0
CKC	CLS 0	CSS 0	MCS 0	MCM0 1	0	0	MCS1 0	MCM1 0

Caution When using the system clock control register (CKC) to change the main system clock (f_{MAIN}), do so while the voltage is within the usable range for the operating mode of the flash memory set in the user option byte (000C2H) and in the flash operating mode select register (FLMODE) both before and after changing the clock. For details about the FLMODE register, see 5.2.1 Flash operating mode select register (FLMODE).

6.6.3 Example of setting the XT1 oscillator clock

After release from the reset state, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock as the source. To subsequently change the source clock to the XT1 oscillator clock, set and start the oscillator by using the subsystem clock supply mode control register (OSMC), clock operation mode control register (CMC), and clock operation status control register (CSC), and set the XT1 oscillator clock as the source of f_{CLK} by using the system clock control register (CKC).

[Register settings] Set the registers in the order of <1> to <5> below.

<1> To select only running the realtime clock with the subsystem clock (for ultra-low current) in STOP mode or in HALT mode while the CPU is operating with the subsystem clock, set the RTCLPC bit to 1.

	7	6	5	4	3	2	1	0
OSMC	RTCLPC 0/1	0	0	WUTMMCK0 0	×	×	0	HIPREC 1

<2> Set the OSCSELS bit of the CMC register to 1 to make the XT1 oscillator operate. Also set the XTSEL bit to 1 in a product with 30 to 36 pins.

	7	6	5	4	3	2	1	0
CMC	EXCLK 0	OSCSEL 0	EXCLKS 0	OSCSELS 1	XTSEL 0/1	AMPHS1 0/1	AMPHS0 0/1	AMPH 0

AMPHS1 and AMPHS0 bits: These bits are used to specify the oscillation mode of the XT1 oscillator.

<3> Clear the XTSTOP bit of the CSC register to 0 to start XT1 oscillation.

	7	6	5	4	3	2	1	0
CSC	MSTOP 1	XTSTOP 0	0	0	0	0	MIOEN 0	HIOSTOP 0

<4> Include code to wait for oscillation of the subsystem clock to become stable by using a timer or in other ways.

<5> Use the CSS bit of the CKC register to specify the XT1 oscillator clock as the source of the CPU/peripheral hardware clock.

	7	6	5	4	3	2	1	0
CKC	CLS 0	CSS 1	MCS 0	MCM0 0	0	0	MCS1 0	MCM1 0

Remark ×: Undefined

6.6.4 State transitions of the CPU clock

Figure 6 - 22 shows the state transitions of the CPU clock.

Figure 6 - 22 State Transitions of the CPU Clock

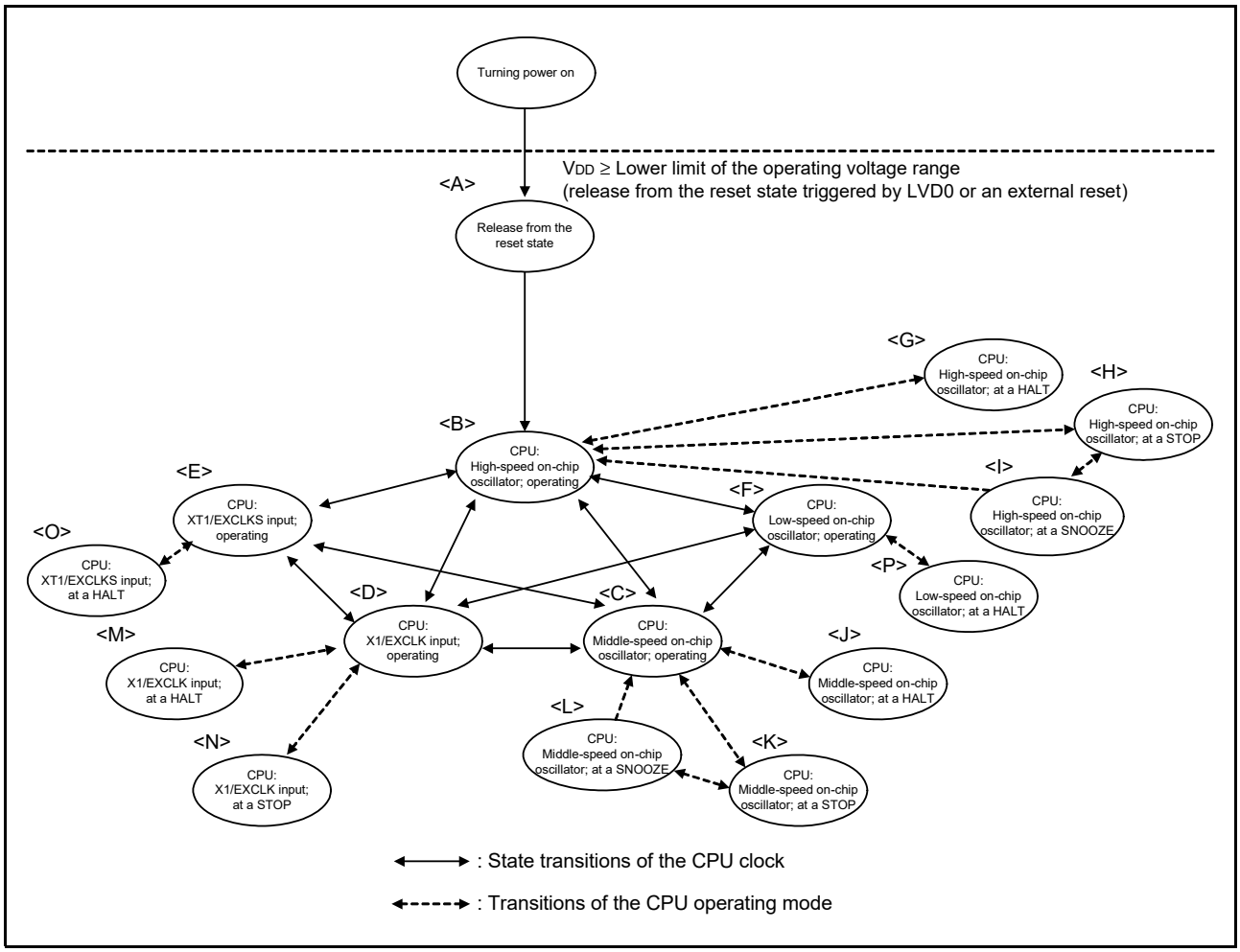


Table 6 - 2 shows examples of transitions of the CPU clock and SFR settings.

Table 6 - 2 Examples of Transitions of the CPU Clock and SFR Settings (1/3)

1. Transition to state <B> where the CPU is operating with the high-speed on-chip oscillator clock after release from the reset state <A>
   Scope of state transitions: <A> → <B>

Clock after Transition	SFR Setting
High-speed on-chip oscillator	SFR setting not required (default state after release from the reset state)

2. Transitions to state <B> where the CPU is operating with the high-speed on-chip oscillator clock
   Scope of state transitions: <C> → <B>, <D> → <B>, <E> → <B>, <F> → <B>

(Sequence of setting the SFRs) →

SFR to be Set / Clock after Transition	CSC Register	Waiting for Oscillation Stabilization	CKC Register		
	HIOSTOP		CSS	MCM0	MCM1
High-speed on-chip oscillator	0	5 μs	0	0	0

Not required if the high-speed on-chip oscillator clock is already running.

3. Transitions to state <C> where the CPU is operating with the middle-speed on-chip oscillator clock
   Scope of state transitions: <B> → <C>, <D> → <C>, <E> → <C>, <F> → <C>

(Sequence of setting the SFRs) →

SFR to be Set / Clock after Transition	CSC Register	Waiting for Oscillation Stabilization	CKC Register		
	MIOEN		CSS	MCM0	MCM1
Middle-speed on-chip oscillator	1	1 μs	0	0	1

Not required if the middle-speed on-chip oscillator clock is already running.

Remark <A> to <P> in **Table 6 - 2** correspond to <A> to <P> in **Figure 6 - 22 State Transitions of the CPU Clock**.

Table 6 - 2 Examples of Transitions of the CPU Clock and SFR Settings (2/3)

4. Transitions to state <D> where the CPU is operating with the high-speed system clock
   Scope of state transitions: <B> → <D>, <C> → <D>, <E> → <D>Note 1, <F> → <D>

(Sequence of setting the SFRs) →

SFR to be Set / Clock after Transition	CMC RegisterNote 2			OSTS Register	CSC Register	OSTC Register	CKC Register	
	EXCLK	OSCSEL	AMPH		MSTOP		CSS	MCM0
X1 clock: 1 MHz ≤ fx ≤ 10 MHz	0	1	0	Note 3	0	Must be checked	0	1
X1 clock: 10 MHz < fx ≤ 20 MHz	0	1	1	Note 3	0	Must be checked	0	1
External main clock	1	1	×	Note 3	0	Need not be checked	0	1

Not required if these bits are already set.

Not required if the high-speed system clock is already running.

Note 1. Products with 30 to 36 pins do not support this transition.

Note 2. The clock operation mode control register (CMC) can be changed only once after release from the reset state. This register setting is not necessary if it has already been set. In the products with 30 to 36 pins, set XTSEL to 0.

Note 3. Set the oscillation stabilization time as follows.

- Desired oscillation stabilization time counted by the oscillation stabilization time counter status register (OSTC) ≤ Oscillation stabilization time set by the oscillation stabilization time select register (OSTS)

Caution **Set a clock after the supply voltage has reached the operating voltage range for the clock to be set (see Section 37 Electrical Characteristics).**

5. Transitions to state <E> where the CPU is operating with the subsystem clock
   Scope of state transitions: <B> → <E>, <C> → <E>, <D> → <E>Note 4

(Sequence of setting the SFRs) →

SFR to be Set / Clock after Transition	CMC RegisterNote 5				CSC Register	Waiting for Oscillation Stabilization	CKC Register
	EXCLKS	OSCSELS	AMPHS1	AMPHS0	XTSTOP		CSS
XT1 clock	0	1	0/1	0/1	0	Required	1
External subsystem clock	1	1	×	×	0	Required	1

Not required if these bits are already set.

Not required if the subsystem clock is already running.

Note 4. Products with 30 to 36 pins do not support this transition.

Note 5. The clock operation mode control register (CMC) can be written only once by an 8-bit memory manipulation instruction after release from the reset state. In the products with 30 to 36 pins, set XTSEL to 1.

Remark 1. ×: Don't care

Remark 2. <A> to <P> in **Table 6 - 2** correspond to <A> to <P> in **Figure 6 - 22 State Transitions of the CPU Clock**.

Table 6 - 2 Examples of Transitions of the CPU Clock and SFR Settings (3/3)

6. Transitions to state <F> where the CPU is operating with the low-speed on-chip oscillator clock
   Scope of state transitions: <B> → <F>, <C> → <F>, <D> → <F>

(Sequence of setting the SFRs) →

SFR to be Set / Clock after Transition	CKSEL	Waiting for Oscillation Accuracy Stabilization	CKC Register
	SELLOSC		CSS
Low-speed on-chip oscillator	1	80 μs	1

Not required if the low-speed on-chip oscillator clock is already running.

7. Transitions from the CPU operating mode <B>, <C>, <D>, <E>, or <F> to the HALT mode <G>, <J>, <M>, <O>, or <P>
   Scope of state transitions: <B> → <G>, <C> → <J>, <D> → <M>, <E> → <O>, <F> → <P>

Mode after Transition	Description
HALT mode	Executing a HALT instruction

8. Transitions from the CPU operating mode <B>, <C>, or <D> to the STOP mode <H>, <K>, or <N>
   Scope of state transitions: <B> → <H>, <C> → <K>, <D> → <N>

(Setting sequence) →

Mode after Transition	Description			
STOP mode	Stopping peripheral functions that cannot operate in STOP mode	Setting the OSTS register	Confirming the value of the HIPREC bit is 1.	Executing a STOP instruction

When STOP or SNOOZE mode is released
- FWKUP = 0 if the high-speed on-chip oscillator is started at normal speed.
- FWKUP = 1 if the high-speed on-chip oscillator is started at high speed.

Only required if the CPU is shifting from the state of operating with the high-speed on-chip oscillator to the STOP mode.

Only required if the CPU is shifting from the state of operating with the high-speed system clock to the STOP mode.

Only required if the high-speed on-chip oscillator is started at high speed.

9. Transitions between a STOP mode <H> or <K> and SNOOZE mode <I> or <L>
   For details about the setting for switching from the STOP mode to the SNOOZE mode, see **Section 23 Standby Function** and descriptions of the SNOOZE mode functions of the on-chip peripheral modules.

Remark <A> to <P> in **Table 6 - 2** correspond to <A> to <P> in **Figure 6 - 22 State Transitions of the CPU Clock**.

6.6.5 Conditions before changing the CPU clock and processing after changing the CPU clock

Table 6 - 3 shows the conditions before changing the CPU clock and processing after changing the CPU clock.

Table 6 - 3 Changing the CPU Clock (1/4)

CPU Clock		Conditions before Change	Processing after Change
Before Change	After Change		
High-speed on-chip oscillator clock	Middle-speed on-chip oscillator clock	The middle-speed on-chip oscillator is operating. • MIOEN = 1	Operating current can be reduced by stopping the high-speed on-chip oscillator (HIOSTOP = 1) after checking that the CPU clock is changed.
	X1 clock	X1 oscillation has become stable. • OSCSEL = 1, EXCLK = 0, MSTOP = 0, XTSEL = 0**Note 1** • The oscillation stabilization time has elapsed.	
	External main system clock	External clock input from the EXCLK pin is enabled. • OSCSEL = 1, EXCLK = 1, MSTOP = 0, XTSEL = 0**Note 1**	
	XT1 clock	XT1 oscillation has become stable. • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0, XTSEL = 1**Note 1** • The oscillation stabilization time has elapsed.	
	External subsystem clock	External clock input from the EXCLKS pin is enabled. • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0, XTSEL = 1**Note 1**	
	Low-speed on-chip oscillator clock	The low-speed on-chip oscillator is selected. • SELLOSC = 1	
Middle-speed on-chip oscillator clock	High-speed on-chip oscillator clock	The high-speed on-chip oscillator is operating. • HIOSTOP = 0	Operating current can be reduced by stopping the middle-speed on-chip oscillator (MIOEN = 0) after checking that the CPU clock is changed.
	X1 clock	X1 oscillation has become stable. • OSCSEL = 1, EXCLK = 0, MSTOP = 0, XTSEL = 0**Note 1** • The oscillation stabilization time has elapsed.	
	External main system clock	External clock input from the EXCLK pin is enabled. • OSCSEL = 1, EXCLK = 1, MSTOP = 0, XTSEL = 0**Note 1**	
	XT1 clock	XT1 oscillation has become stable. • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0, XTSEL = 1**Note 1** • The oscillation stabilization time has elapsed.	
	External subsystem clock	External clock input from the EXCLKS pin is enabled. • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0, XTSEL = 1**Note 1**	
	Low-speed on-chip oscillator clock	The low-speed on-chip oscillator is selected. • SELLOSC = 1	

Table 6 - 3 Changing the CPU Clock (2/4)

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 is enabled. • HIOSTOP = 0 • The oscillation stabilization time has elapsed.	X1 oscillation can be stopped (MSTOP = 1) after checking that the CPU clock is changed.
	Middle-speed on-chip oscillator clock	The middle-speed on-chip oscillator is operating. • MIOEN = 1	
	External main system clock	Transition is not possible.	—
	XT1 clock**Note 2**	XT1 oscillation has become stable. • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • The oscillation stabilization time has elapsed.	X1 oscillation can be stopped (MSTOP = 1) after checking that the CPU clock is changed.
	External subsystem clock**Note 2**	External clock input from the EXCLKS pin is enabled. • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	
	Low-speed on-chip oscillator clock	The XT1 oscillator is stopped. The low-speed on-chip oscillator is selected. • SELLOSC = 1	
External main system clock	High-speed on-chip oscillator clock	The high-speed on-chip oscillator is enabled. • HIOSTOP = 0 • The oscillation stabilization time has elapsed.	External main system clock input can be disabled (MSTOP = 1) after checking that the CPU clock is changed.
	Middle-speed on-chip oscillator clock	The middle-speed on-chip oscillator is operating. • MIOEN = 1	
	X1 clock	Transition is not possible.	—
	XT1 clock**Note 2**	XT1 oscillation has become stable. • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • The oscillation stabilization time has elapsed.	External main system clock input can be disabled (MSTOP = 1) after checking that the CPU clock is changed.
	External subsystem clock**Note 2**	External clock input from the EXCLKS pin is enabled. • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	
	Low-speed on-chip oscillator clock	The XT1 oscillator is stopped. The low-speed on-chip oscillator is selected. • SELLOSC = 1	

Table 6 - 3 Changing the CPU Clock (3/4)

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 is operating and the high-speed on-chip oscillator clock is selected as the main system clock. • HIOSTOP = 0, MCS = 0, MCS1 = 0	XT1 oscillation can be stopped (XTSTOP = 1) after checking that the CPU clock is changed.
	Middle-speed on-chip oscillator clock	The middle-speed on-chip oscillator is operating and the middle-speed on-chip oscillator clock is selected as the main system clock. • MIOEN = 1, MCS = 0, MCS1 = 1	
	X1 clock**Note 2**	X1 oscillation has become stable and the high-speed system clock is selected as the main system clock. • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • The oscillation stabilization time has elapsed. • MCS = 1	
	External main system clock**Note 2**	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 • MCS = 1	
	External subsystem clock	Transition is not possible.	—
	Low-speed on-chip oscillator clock		
External subsystem clock	High-speed on-chip oscillator clock	The high-speed on-chip oscillator is operating and the high-speed on-chip oscillator clock is selected as the main system clock. • HIOSTOP = 0, MCS = 0, MCS1 = 0	External subsystem clock input can be disabled (XTSTOP = 1) after checking that the CPU clock is changed.
	Middle-speed on-chip oscillator clock	The middle-speed on-chip oscillator is operating and the middle-speed on-chip oscillator clock is selected as the main system clock. • MIOEN = 1, MCS = 0, MCS1 = 1	
	X1 clock**Note 2**	X1 oscillation has become stable and the high-speed system clock is selected as the main system clock. • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • The oscillation stabilization time has elapsed. • MCS = 1	
	External main system clock**Note 2**	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 • MCS = 1	
	XT1 clock	Transition is not possible.	—
	Low-speed on-chip oscillator clock		

Table 6 - 3 Changing the CPU Clock (4/4)

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 is operating and the high-speed on-chip oscillator clock is selected as the main system clock. • HIOSTOP = 0, MCS = 0, MCS1 = 0	—
	Middle-speed on-chip oscillator clock	The middle-speed on-chip oscillator is operating and the middle-speed on-chip oscillator clock is selected as the main system clock. • MIOEN = 1, MCS = 0, MCS1 = 1	
	X1 clock	X1 oscillation has become stable and the high-speed system clock is selected as the main system clock. • OSCSEL = 1, EXCLK = 0, MSTOP = 0, XTSEL = 0**Note 1** • The oscillation stabilization time has elapsed. • 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, XTSEL = 0**Note 1** • MCS = 1	
	XT1 clock	Transition is not possible.	—
	External subsystem clock		

Note 1. Writing to the XTSEL bit is only possible in the products with 30 to 36 pins.

Note 2. Switching to this clock is only possible in the products with 40 to 128 pins.

6.6.6 Time required for switchover of the CPU clock and main system clock

By setting the CSS, MCM0, and MCM1 bits of the system clock control register (CKC), the CPU clock can be switched between the main system clock and subsystem clock, the main system clock can be switched between the on-chip oscillator clock and high-speed system clock, and the on-chip oscillator clock can be switched between the high-speed on-chip oscillator clock and middle-speed on-chip oscillator clock. In actual operation, the clock is not switched immediately after writing to the CKC register; the CPU continues to operate with the prior clock for several clock cycles after writing proceeds (see **Table 6 - 4** to **Table 6 - 7**). Whether the source of the CPU clock is the main system clock or subsystem clock can be ascertained from the CLS bit of the CKC register. Whether the source of the main system clock is the high-speed system clock or main on-chip oscillator clock can be ascertained from the MCS bit of the CKC register. Whether the source of the main on-chip oscillator clock is the high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock can be ascertained from the MCS1 bit of the CKC register. When the CPU clock is switched, the peripheral hardware clock is also switched.

Table 6 - 4 Maximum Time Required for Main System Clock Switchover

Clock A	Switching Directions	Clock B	Remark
f_{OCO}	⟷	f_{MX}	See **Table 6 - 5**.
f_{IH}	⟷	f_{IM}	See **Table 6 - 6**.
f_{MAIN}	⟷	f_{SUB}	See **Table 6 - 7**.

Table 6 - 5 Maximum Number of Clock Cycles Required for f_{OCO} ↔ f_{MX}

Set Value before Switchover		Set Value after Switchover	
MCM0		MCM0	
		0 ($f_{MAIN} = f_{OCO}$)	1 ($f_{MAIN} = f_{MX}$)
0 ($f_{MAIN} = f_{OCO}$)	$f_{MX} \geq f_{OCO}$		2 cycles
	$f_{MX} < f_{OCO}$		2 f_{OCO}/f_{MX} cycles
1 ($f_{MAIN} = f_{MX}$)	$f_{MX} \geq f_{OCO}$	2 f_{MX}/f_{OCO} cycles	
	$f_{MX} < f_{OCO}$	2 cycles	

Table 6 - 6 Maximum Number of Clock Cycles Required for f_{IH} ↔ f_{IM}

Set Value before Switchover		Set Value after Switchover	
MCM1		MCM1	
		0 ($f_{OCO} = f_{IH}$)	1 ($f_{OCO} = f_{IM}$)
0 ($f_{OCO} = f_{IH}$)	$f_{IM} \geq f_{IH}$		2 cycles
	$f_{IM} < f_{IH}$		2 f_{IH}/f_{IM} cycles
1 ($f_{OCO} = f_{IM}$)	$f_{IM} \geq f_{IH}$	2 f_{IM}/f_{IH} cycles	
	$f_{IM} < f_{IH}$	2 cycles	

Table 6 - 7 Maximum Number of Clock Cycles Required for f_{MAIN} ↔ f_{SUB}

Set Value before Switchover	Set Value after Switchover	
CSS	CSS	
	0 (f_{CLK} = f_{MAIN})	1 (f_{CLK} = f_{SUB})
0 (f_{CLK} = f_{MAIN})		1 + 2 f_{MAIN}/f_{SUB} cycles
1 (f_{CLK} = f_{SUB})	3 cycles	

Remark 1. The number of clock cycles listed in **Table 6 - 5** to **Table 6 - 7** is the number of cycles of the CPU clock before switchover.

Remark 2. Calculate the number of clock cycles in **Table 6 - 5** to **Table 6 - 7** by rounding up the number after the decimal position.

Example: When switching the main system clock from the high-speed on-chip oscillator clock (8 MHz selected) to the high-speed system clock (in this case, f_{IH} = 8 MHz, f_{MX} = 10 MHz)

1 + f_{IH}/f_{MX} = 1 + 8/10 = 1 + 0.8 = 1.8 → 2 cycles

6.6.7 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. When stopping the clock, confirm the conditions before clock oscillation is stopped.

Table 6 - 8 Conditions Before the Clock Oscillation Is Stopped and Flag Settings

Clock	Conditions Before Clock Oscillation is Stopped (External Clock Input is Disabled)	Flag Settings of the SFRs
High-speed on-chip oscillator clock	MCS1 = 1, MCS = 1 or CLS = 1 (The CPU is operating with a clock other than the high-speed on-chip oscillator clock.)	HIOSTOP = 1
Middle-speed on-chip oscillator clock	MCS1 = 0, MCS = 1 or CLS = 1 (The CPU is operating with a clock other than the middle-speed on-chip oscillator clock.)	MIOEN = 0
X1 clock External main system clock	MCS = 0 or CLS = 1 (The CPU is operating with a clock other than the high-speed system clock.)	MSTOP = 1
XT1 clock External subsystem clock	CLS = 0 (The CPU is operating with a clock other than the subsystem clock.)	XTSTOP = 1
Low-speed on-chip oscillator clock**Note**	CLS = 0 (The CPU is operating with a clock other than the low-speed on-chip oscillator clock.)	SELLOSC = 0 WUTMMCK0 = 0

Note The low-speed on-chip oscillator clock is not stopped while the WDT is operating or when f_{IL} is selected as the source clock for use in waiting by the SNOOZE mode sequencer.

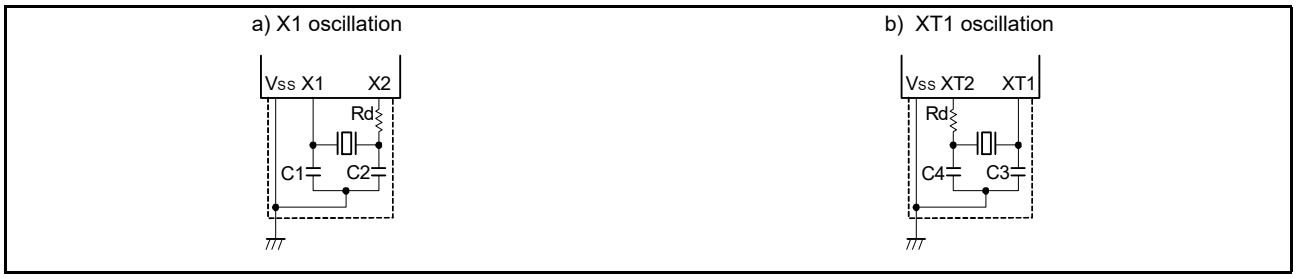
6.7 Resonator and Oscillator Constants

For the resonators for which operation has been verified and their oscillator constants (reference values), see the target product page on the Renesas Web site.

Caution 1. The constants for these oscillator circuits are reference values based on specific environments set up for evaluation by the manufacturers. For actual applications, request evaluation by the manufacturer of the oscillator circuit mounted on a board. Furthermore, if you are switching from a different product to this microcontroller, and whenever you change the board, again request evaluation by the manufacturer of the oscillator circuit mounted on the new board.

Caution 2. The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use this MCU so that the internal operation conditions are within the specifications of the DC and AC characteristics.

Figure 6 - 23 Examples of External Circuits



Section 7 Timer Array Unit (TAU)

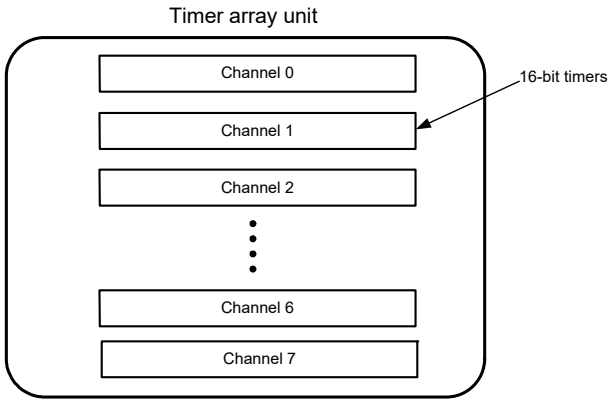
The number of units or channels of the timer array unit differs, depending on the product.

Units	Channels	R7F100G			R7F104G		
		30- to 64-pin	80- and 100-pin	128-pin	32-pin	44-pin	48-pin
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	—	—	✓	—	—	—

Caution 1. Which of the timer I/O pins are present and absent depends on the product. See Table 7 - 2 Timer I/O Pins Provided in Each Product for details.

Caution 2. Most of the following descriptions in this section use the 128-pin R7F100G 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 (see **7.8.1**) • Square wave output (see **7.8.1**) • External event counter (see **7.8.2**) • Divider**Note** (see **7.8.3**) • Input pulse interval measurement (see **7.8.4**) • Measurement of low- or high-level width of input signal (see **7.8.5**) • Delay counter (see **7.8.6**)	• One-shot pulse output (see **7.9.1**) • PWM output (see **7.9.2**) • Multiple PWM output (see **7.9.3**)

Note Only channel 0 of unit 0 supports this function.

Each of the 16-bit timer channels 1 and 3 of units 0 and 1 can be used as two 8-bit timers, represented by the higher- and lower-order bytes. The following functions can be realized by using channels 1 and 3 as 8-bit timers:

- Interval timer (higher or lower 8-bit timer)/square wave output (only for lower 8-bit timer)
- External event counter (only for lower 8-bit timer)
- Delay counter (only for lower 8-bit timer)

Channel 7 of unit 0 can be used to realize LIN-bus communication operating in combination with UART2 of the serial array unit.

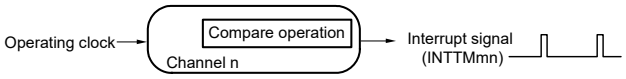
7.1 Functions of Timer Array Unit

Timer array unit has the following functions.

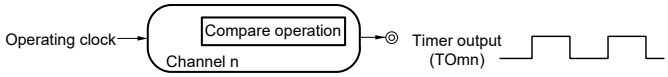
7.1.1 Independent channel operation function

By operating a channel independently, it can be used for the following seven purposes without being affected by the operation mode of other channels.

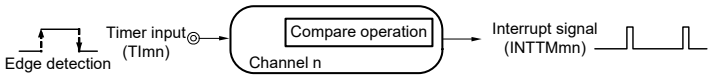
1. Interval timer
   Each timer of a unit can be used as a reference timer that generates an interrupt (INTTMmn) at fixed intervals.



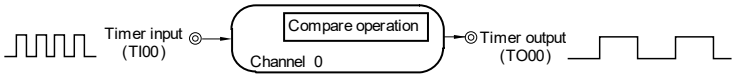
2. Square wave output
   A toggle operation is performed each time INTTMmn interrupt is generated and a square wave with a duty factor of 50% is output from a timer output pin (TOmn).



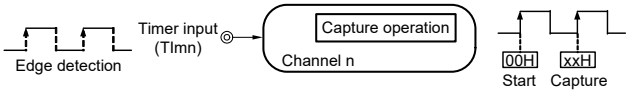
3. External event counter
   Each timer of a unit can be used as an event counter that generates an interrupt when the number of the valid edges of a signal input to the timer input pin (TImn) has reached a specific value.



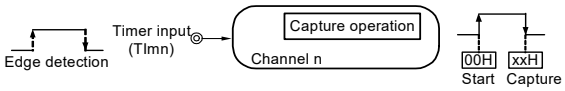
4. Divider function (only for channel 0 of unit 0)
   A clock input from a timer input pin (TI00) is divided and output from an output pin (TO00).



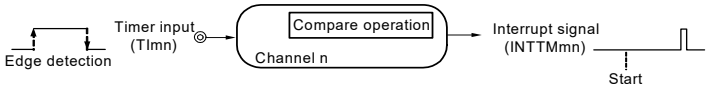
5. Input pulse interval measurement
   Counting is started by the valid edge of a pulse signal input to a timer input pin (TImn). The count value of the timer is captured at the valid edge of the next pulse. In this way, the interval of the input pulse can be measured.



6. Measurement of low- or high-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 low- or high-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.



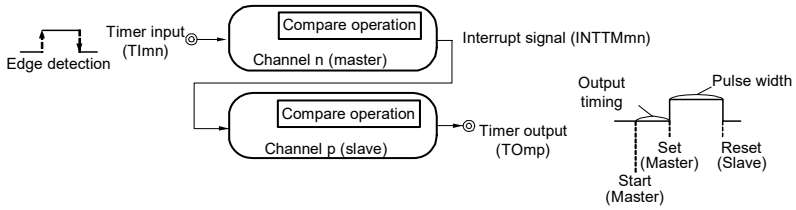
Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

Remark 2. Which of the timer I/O pins for channels 0 to 7 are present and absent depends on the product. See **Table 7 - 2 Timer I/O Pins Provided in Each Product** for details.

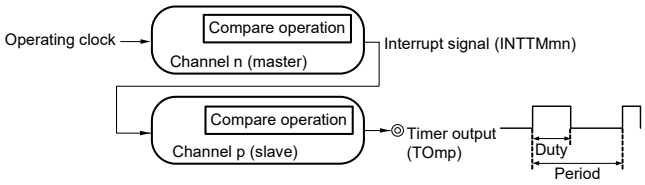
7.1.2 Simultaneous channel operation function

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 three purposes.

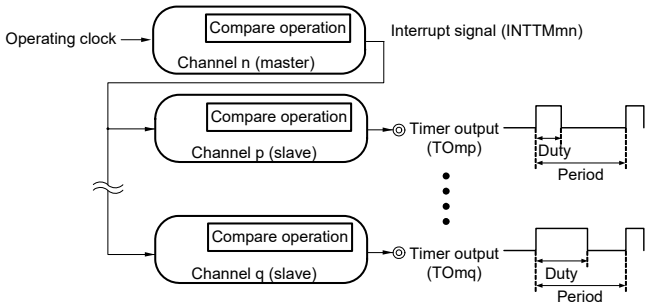
1. One-shot pulse output
   Two channels are used as a set to generate a one-shot pulse with a specified output timing and a specified pulse width.



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.



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 **For details about the rules of simultaneous channel operation function, see 7.4.1 Basic specifications of the 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)

7.1.3 8-bit timer function (only for channels 1 and 3)

The 8-bit timer function allows the use of a 16-bit timer configured by cascade-connecting two 8-bit timer channels. This function can only be used with channels 1 and 3.

Caution **There are several rules for using the 8-bit timer function. For details, see 7.4.2 Basic specifications of the 8-bit timer function (only for channels 1 and 3).**

7.1.4 LIN-bus supporting function (only for channel 7 of unit 0)

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 (RxD2) of UART2 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 break field
   The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD2) of UART2 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 break field.
3. Measurement of pulse width of sync field
   After a break field is detected, the low-level width and high-level width of the signal input to the serial data input pin (RxD2) of UART2 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 **7.3.15 Input switch control register (ISC)** and **7.8.5 Operation for input signal low- and high-level width measurement**.

7.2 Configuration of Timer Array Unit

Timer array unit includes the following hardware.

Table 7 - 1 Configuration of Timer Array Unit

Item	Configuration
Timer/counter	Timer counter register mn (TCRmn)
Register	Timer data register mn (TDRmn)
Timer input	TI00 to TI07, TI10 to TI17[Note 1], RxD2 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) • Peripheral reset control register 0 (PRR0) • 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 output enable register m (TOEm) • Timer output register m (TOm) • Timer output level register m (TOLm) • Timer output mode register m (TOMm)
	<Registers of each channel> • Timer mode register mn (TMRmn) • Timer status register mn (TSRmn) • Input switch control register (ISC) • Noise filter enable registers 1, 2 (NFEN1, NFEN2) • Port mode registers (PMxx)[Note 2] • Port registers (Pxx)[Note 2] • Port output mode registers (POMxx)[Note 2] • Port mode control A registers (PMCAxx)[Note 2] • Port mode control T registers (PMCTxx)[Note 2] • Port mode control E registers (PMCExx)[Note 2] • Port function output enable registers (PFOEx)[Note 2]

Note 1. Which of the timer I/O pins for channels 0 to 7 are present and absent depends on the product. See **Table 7 - 2 Timer I/O Pins Provided in Each Product** for details.

Note 2. The port mode registers (PMxx), port registers (Pxx), port output mode registers (POMxx), port mode control A registers (PMCAxx), port mode control T registers (PMCTxx), port mode control E registers (PMCExx), and port function output enable register 0 (PFOE0) to be set differ depending on the product. For details, see **4.5.4 Examples of register settings for port and multiplexed functions**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

Which of the timer I/O pins for each timer array unit channel are present and absent depends on the product.

Table 7 - 2 Timer I/O Pins Provided in Each Product

Timer array unit channels		I/O Pins of Each Product										
		R7F100G								R7F104G		
		30- to 36-pin	40-pin	44- and 48-pin	52-pin	64-pin	80-pin	100-pin	128-pin	32-pin	44-pin	48-pin
Unit 0	Channel 0	TI00, TO00								TI00	TI00, TO00	
	Channel 1	TI01/TO01								TI01/TO01		
	Channel 2	TI02/TO02								TI02/TO02		
	Channel 3	TI03/TO03								TI03/TO03		
	Channel 4	(TI04/TO04)				TI04/TO04				—	(TI04/TO04)	
	Channel 5	(TI05/TO05)				TI05/TO05				TI05/TO05		
	Channel 6	(TI06/TO06)				TI06/TO06				TI06/TO06		
	Channel 7	(TI07/TO07)		TI07/TO07						(TI07/TO07)		TI07/TO07
Unit 1	Channel 0	—	—	—	—	—	TI10/TO10			—	TI10/TO10	
	Channel 1	—	—	—	—	—	TI11/TO11			—	TI11/TO11	
	Channel 2	—	—	—	—	—	TI12/TO12			—	—	TI12/TO12
	Channel 3	—	—	—	—	—	TI13/TO13			—	—	—
	Channel 4	—	—	—	—	—	—	—	TI14/TO14	—	—	—
	Channel 5	—	—	—	—	—	—	—	TI15/TO15	—	—	—
	Channel 6	—	—	—	—	—	—	—	TI16/TO16	—	—	—
	Channel 7	—	—	—	—	—	—	—	TI17/TO17	—	—	—

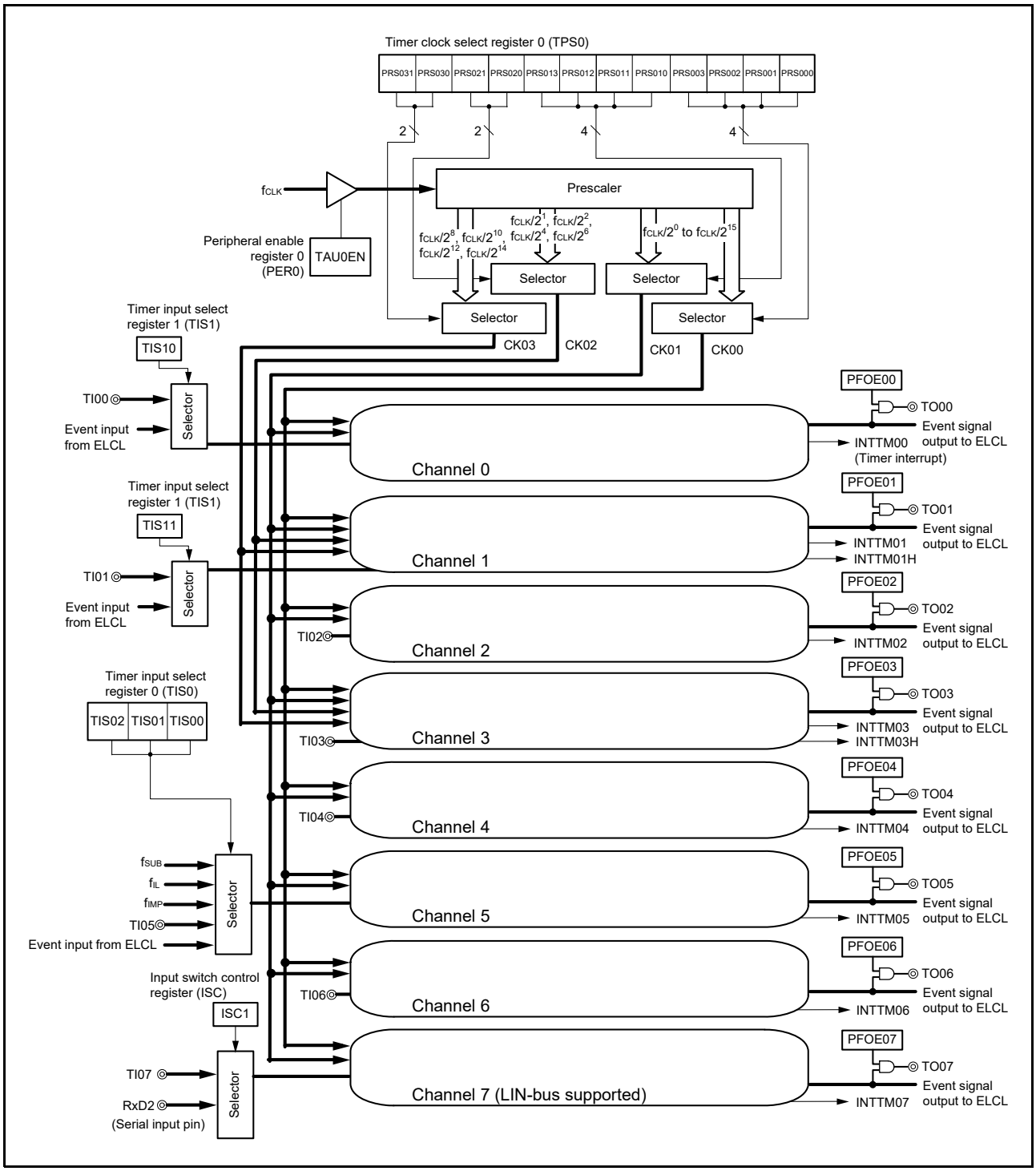
Remark 1. When timer input and timer output are shared by the same pin, either only timer input or only timer output can be used.

Remark 2. —: The channel is not available.

Remark 3. Pins in parentheses are selected when the setting of bit 0 of the peripheral I/O redirection register (PIOR) is 1.

Figure 7 - 1 shows the block diagram of timer array unit 0.

Figure 7 - 1 Block Diagram of Timer Array Unit 0 for the 64-pin R7F100G Products as an Example



Remark fSUB: Subsystem clock frequency
fIL: Low-speed on-chip oscillator clock frequency
fIMP: Middle-speed on-chip oscillator peripheral clock frequency

Figure 7 - 2 Internal Block Diagram of Channel 0 of Timer Array Unit 0

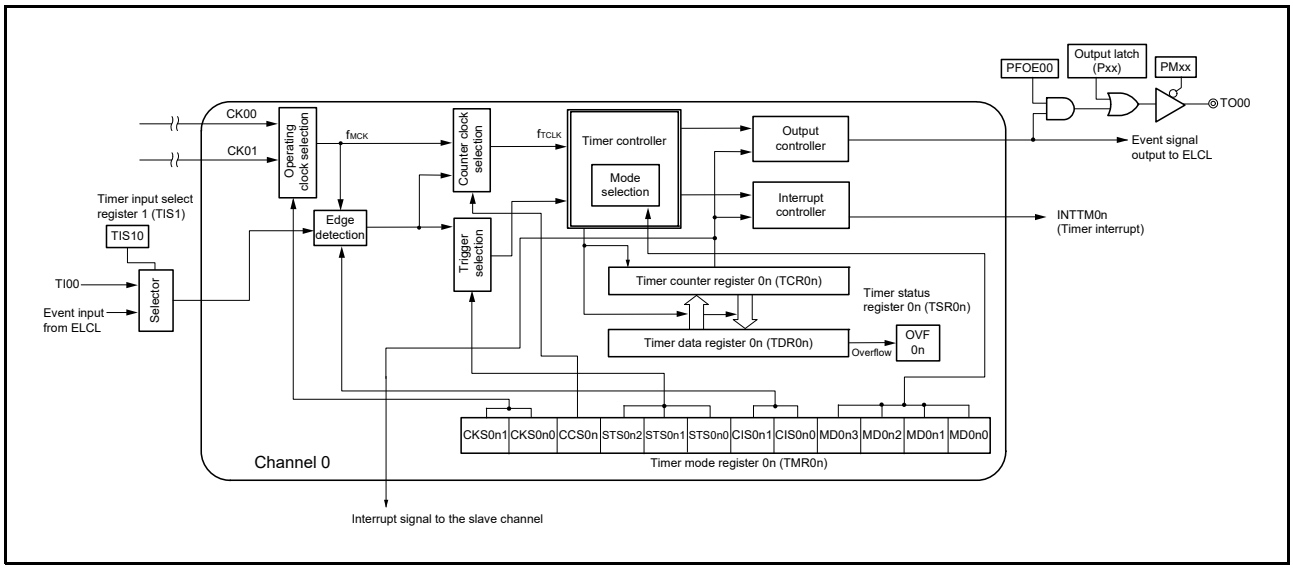


Figure 7 - 3 Internal Block Diagram of Channel 1 of Timer Array Unit 0

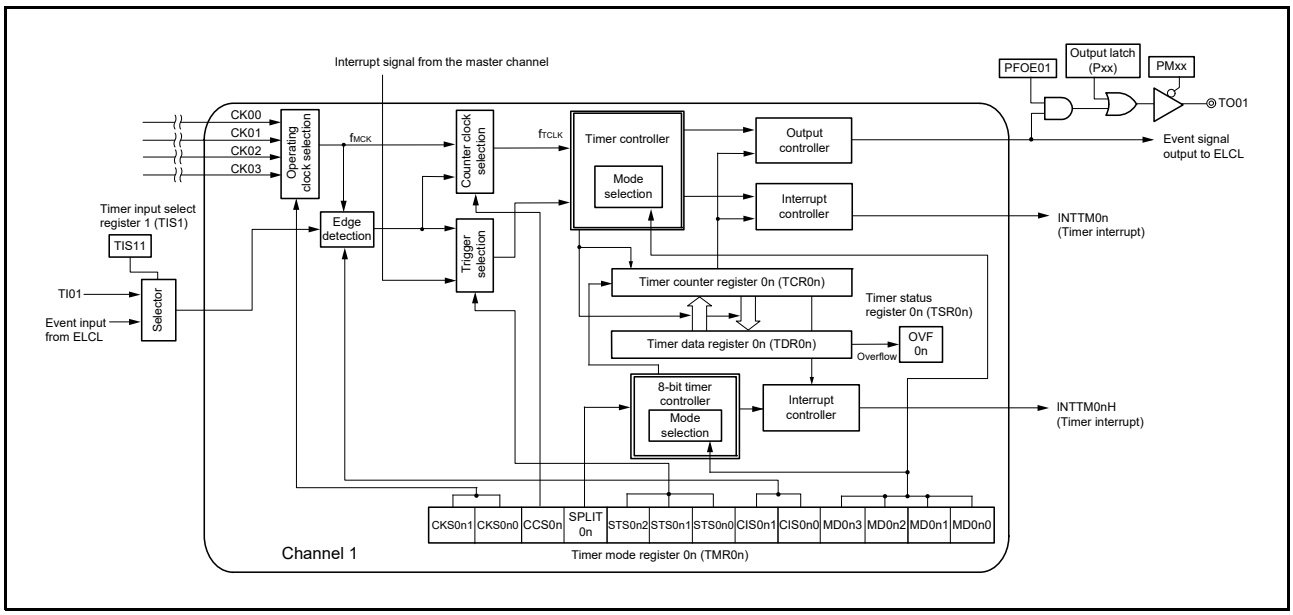
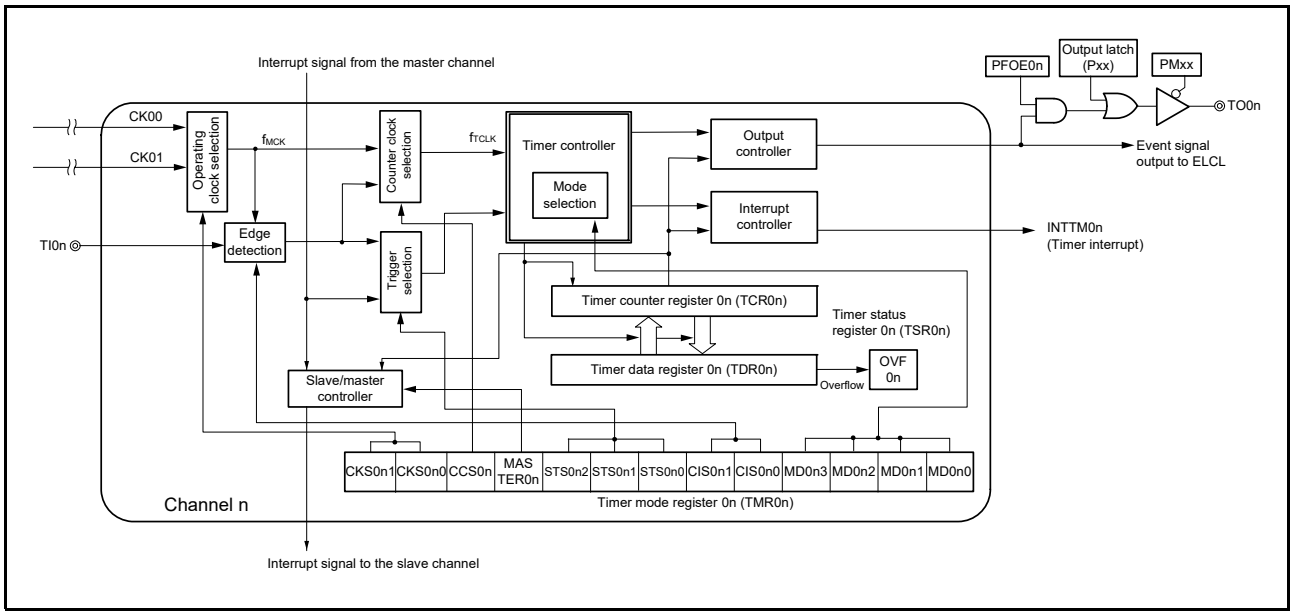


Figure 7 - 4 Internal Block Diagram of Channels 2, 4, and 6 of Timer Array Unit 0



Remark n = 2, 4, 6

Figure 7 - 5 Internal Block Diagram of Channel 3 of Timer Array Unit 0

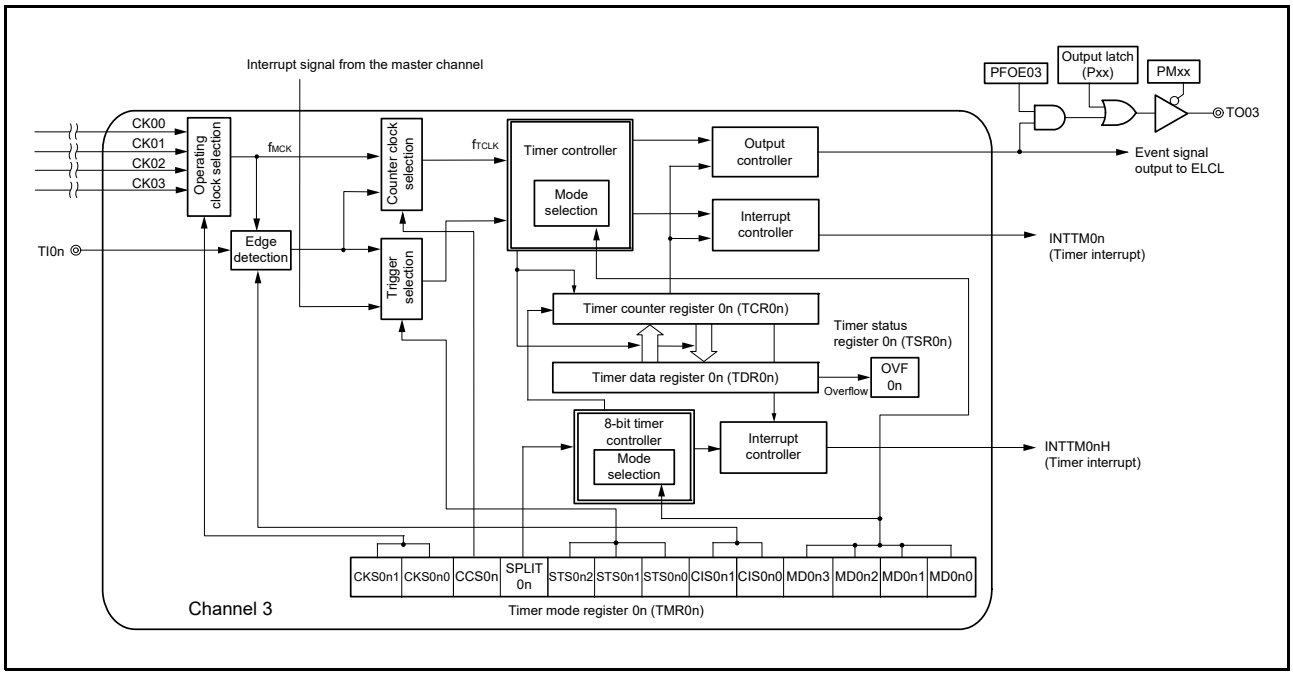


Figure 7 - 6 Internal Block Diagram of Channel 5 of Timer Array Unit 0

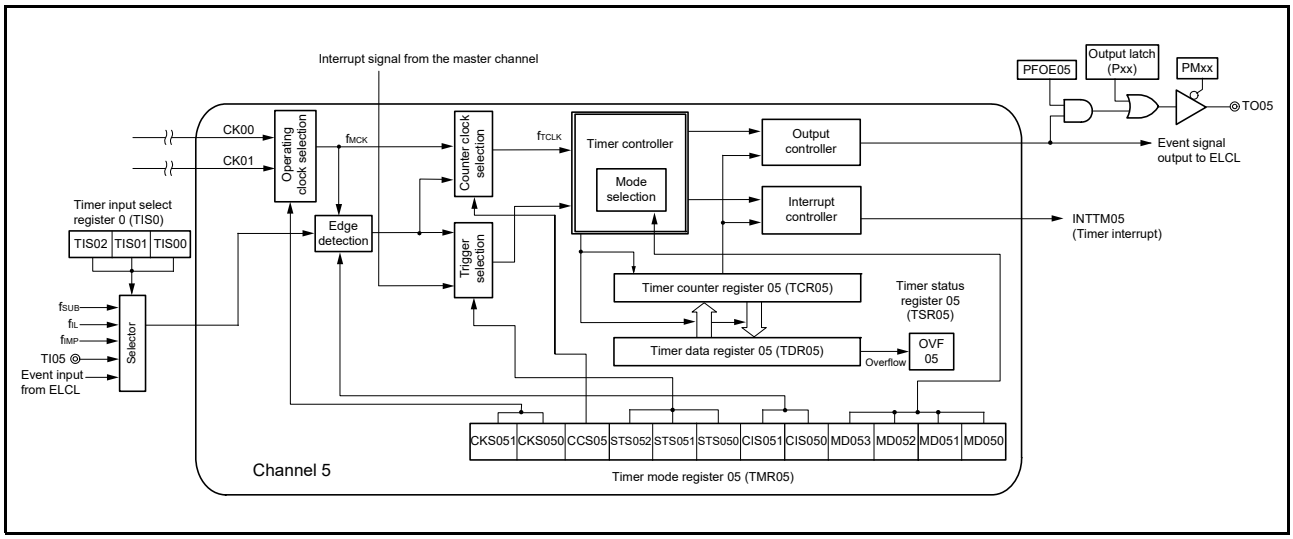
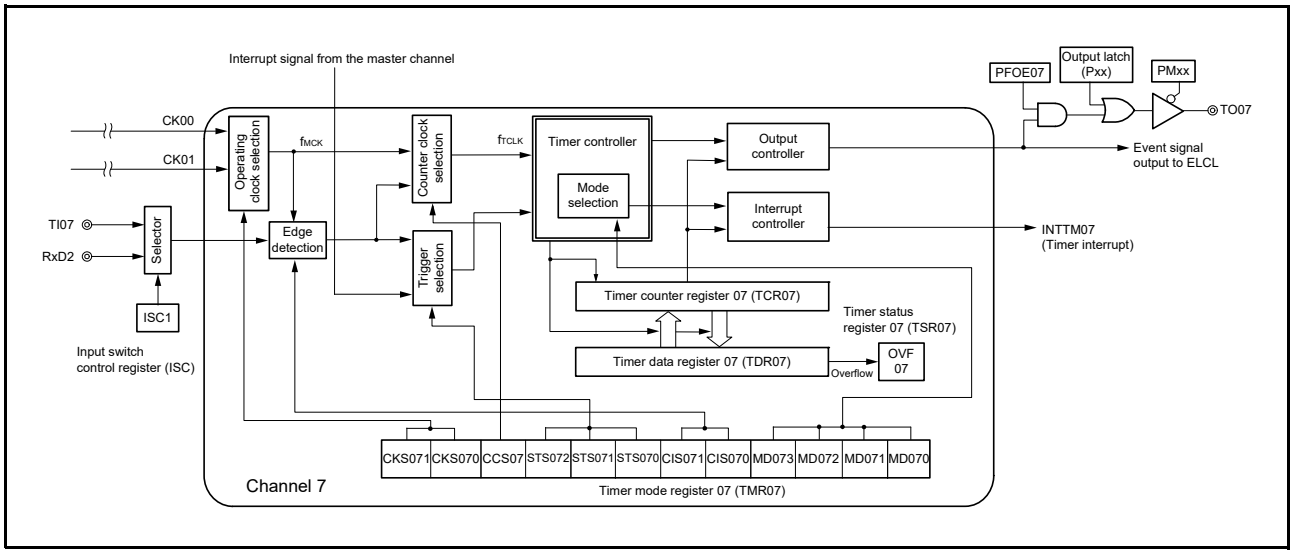


Figure 7 - 7 Internal Block Diagram of Channel 7 of Timer Array Unit 0

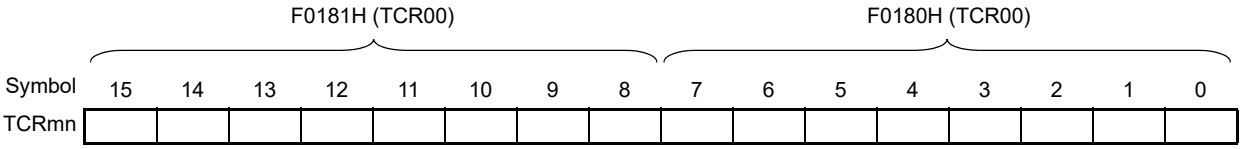


7.2.1 Timer counter register mn (TCRmn)

The TCRmn register is a 16-bit read-only register and is used to count the number of input clock cycles. The value of this counter is incremented or decremented in synchronization with the rising edge of each cycle of the counter 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) (see **7.3.4 Timer mode register mn (TMRmn)**).

Figure 7 - 8 Format of Timer Counter Register mn (TCRmn)

Address: F0180H, F0181H (TCR00) to F018EH, F018FH (TCR07), F01C0H, F01C1H (TCR10) to F01CEH, F01CFH (TCR17)
After reset: FFFFH
R/W: R



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

The count value can be read by reading timer counter register mn (TCRmn). The count value is set to FFFFH in the following cases.

- When the reset signal is generated
- When the TAUmRES bit of peripheral reset control register 0 (PRR0) 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 counter 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 value read from the TCRmn register varies depending on the change to the operation mode and operating state as shown in the table below.

Table 7 - 3 Timer Counter Register mn (TCRmn) Read Value in Various Operation Modes

Operation Mode	Count Mode	Value Read from the Timer Counter Register mn (TCRmn)Note			
		Value when the operation mode is changed after releasing reset	Value when count operation is temporarily stopped (TTmn = 1)	Value when the operation mode is changed after count operation was temporarily stopped (TTmn = 1)	Value when waiting for a start trigger after one count
Interval timer mode	Countdown	FFFFH	Value when counting is stopped	Undefined	—
Capture mode	Count-up	0000H	Value when counting is stopped	Undefined	—
Event counter mode	Countdown	FFFFH	Value when counting is stopped	Undefined	—
One-count mode	Countdown	FFFFH	Value when counting is stopped	Undefined	FFFFH
Capture & one-count mode	Count-up	0000H	Value when counting is stopped	Undefined	Captured 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)

7.2.2 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). The value of the TDRmn register 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 read and write 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. The value of this register following a reset is 0000H.

Figure 7 - 9 Format of Timer Data Register mn (TDRmn) (n = 0, 2, 4 to 7)

Address: FFF18H, FFF19H (TDR00), FFF64H, FFF65H (TDR02), FFF68H, FFF69H (TDR04) to FFF6EH, FFF6FH (TDR07), FFF70H, FFF71H (TDR10), FFF74H, FFF75H (TDR12), FFF78H, FFF79H (TDR14) to FFF7EH, FFF7FH (TDR17)
After reset: 0000H
R/W: R/W

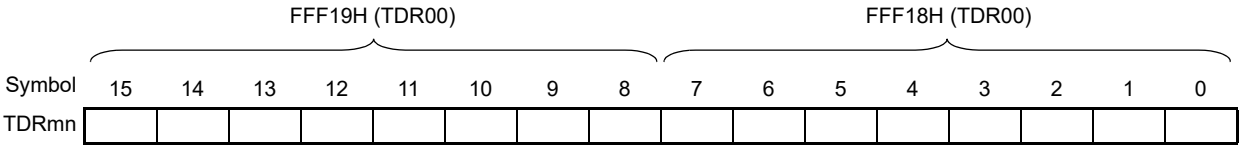
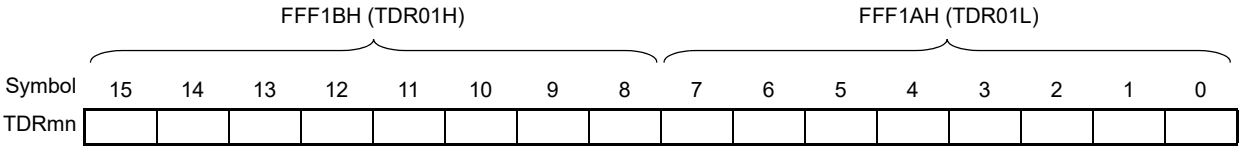


Figure 7 - 10 Format of Timer Data Register mn (TDRmn) (n = 1, 3)

Address: FFF1AH, FFF1BH (TDR01), FFF66H, FFF67H (TDR03), FFF72H, FFF73H (TDR11), FFF76H, FFF77H (TDR13)
After reset: 0000H
R/W: R/W



i) When timer data register mn (TDRmn) is used as compare register
Counting down is started from the value set to the TDRmn register. When the count value reaches 0000H, an interrupt signal (INTTMmn) is generated. The TDRmn register holds its value until it is rewritten.

ii) When timer data register mn (TDRmn) is used as capture register
The count value of timer counter 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).

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.**

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

7.3 Registers for Controlling the Timer Array Unit

The following registers are used to control the timer array unit.

- Peripheral enable register 0 (PER0)
- Peripheral reset control register 0 (PRR0)
- Timer clock select register m (TPSm)
- Timer mode register mn (TMRmn)
- Timer status register mn (TSRmn)
- Timer channel enable status register m (TEm)
- Timer channel start register m (TSm)
- Timer channel stop register m (TTm)
- Timer input select register 0 (TIS0)
- Timer input select register 1 (TIS1)
- Timer output enable register m (TOEm)
- Timer output register m (TOm)
- Timer output level register m (TOLm)
- Timer output mode register m (TOMm)
- Input switch control register (ISC)
- Noise filter enable registers 1, 2 (NFEN1, NFEN2)
- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- Port function output enable registers (PFOEx)

Caution Which registers and bits are included depends on the product. Be sure to set bits that are not mounted to their initial values.

7.3.1 Peripheral enable register 0 (PER0)

The PER0 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. If timer array unit 0 is to be used, be sure to set the TAU0EN bit of this register to 1. If timer array unit 1 is to be used, be sure to set the TAU1EN bit of this register to 1. The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 7 - 11 Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCWEN	IICA1EN **Note 1**	ADCEN	IICA0EN **Note 2**	SAU1EN	SAU0EN	TAU1EN **Note 3**	TAU0EN

TAU1EN	Control of supply of an input clock to timer array unit 1
0	Stops supply of an input clock. • The SFRs used by timer array unit 1 cannot be written.**Note 4**
1	Enables supply of an input clock. • The SFRs used by timer array unit 1 can be read and written.

TAU0EN	Control of supply of an input clock to timer array unit 0
0	Stops supply of an input clock. • The SFRs used by timer array unit 0 cannot be written.**Note 4**
1	Enables supply of an input clock. • The SFRs used by timer array unit 0 can be read and written.

Note 1. This bit is only present in the 44- to 128-pin R7F100G products.
Note 2. This bit is only present in the 30- to 128-pin R7F100G and 48-pin R7F104G products.
Note 3. This bit is only present in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products.
Note 4. When an SFR used by a timer array unit is read, the value returned is 00H or 0000H regardless of the setting value.

(Cautions are listed on the next page.)

Caution 1. Be sure to leave the following bits set to 0.
Bits 6 and 1 in the 30- to 40-pin R7F100G products
Bit 1 in the 44- to 64-pin R7F100G products
Bits 6, 4, and 1 in the 32-pin R7F104G product
Bits 6 and 4 in the 44-pin R7F104G product
Bit 6 in the 48-pin R7F104G product

Caution 2. When setting a timer array unit, start by setting the following registers while the setting of the TAUmEN bit is 1.

- **Timer clock select register m (TPSm)**
- **Timer mode register mn (TMRmn)**
- **Timer status register mn (TSRmn)**
- **Timer channel enable status register m (TEm)**
- **Timer channel start register m (TSm)**
- **Timer channel stop register m (TTm)**
- **Timer output enable register m (TOEm)**
- **Timer output register m (TOm)**
- **Timer output level register m (TOLm)**
- **Timer output mode register m (TOMm)**

If the setting of the TAUmEN bit is 0, each register which controls the timer array unit is restored to 00H and writing to any of those registers is ignored. Note, however, writing to the following registers is valid.

- **Timer input select registers 0, 1 (TIS0, TIS1)**
- **Input switch control register (ISC)**
- **Noise filter enable registers 1, 2 (NFEN1, NFEN2)**
- **Port mode control A registers (PMCA0, PMCA1)**
- **Port mode control T registers (PMCT0, PMCT3, PMCT6)**
- **Port mode control E registers (PMCE0, PMCE1)**
- **Port mode registers (PM0, PM1, PM3, PM4, PM6, PM10, PM14)**
- **Port registers (P0, P1, P3, P4, P6, P10, P14)**
- **Port function output enable register 0 (PFOE0)**

7.3.2 Peripheral reset control register 0 (PRR0)

The PRR0 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. Set the TAU0RES and TAU1RES bits of this register to 1 to place timer array units 0 and 1 in the reset state, respectively. The PRR0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 7 - 12 Format of Peripheral Reset Control Register 0 (PRR0)

Address: F00F1H
After reset: 00H
R/W: R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PRR0	0	IICA1RES **Note 1**	ADCRES	IICA0RES **Note 2**	SAU1RES	SAU0RES	TAU1RES **Note 3**	TAU0RES

TAU1RES	Control resetting of timer array unit 1
0	Timer array unit 1 is released from the reset state.
1	Timer array unit 1 is in the reset state. • The SFRs for use with timer array unit 1 are initialized.

TAU0RES	Control resetting of timer array unit 0
0	Timer array unit 0 is released from the reset state.
1	Timer array unit 0 is in the reset state. • The SFRs for use with timer array unit 0 are initialized.

Note 1. This bit is only present in the 44- to 128-pin R7F100G products.
Note 2. This bit is only present in the 30- to 128-pin R7F100G and 48-pin R7F104G products.
Note 3. This bit is only present in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products.

Caution 1. Be sure to leave bit 7 set to 0.
Caution 2. Be sure to leave the following bits set to 0.
Bits 6 and 1 in the 30- to 40-pin R7F100G products
Bit 1 in the 44- to 64-pin R7F100G products
Bits 6, 4, and 1 in the 32-pin R7F104G product
Bits 6 and 4 in the 44-pin R7F104G product
Bit 6 in the 48-pin R7F104G product

7.3.3 Timer clock select register m (TPSm)

The TPSm register is a 16-bit register used to select two types or four types of operating clocks (CKm0, CKm1, CKm2, CKm3) that are commonly supplied to each channel. CKm0 is selected by using bits 3 to 0 of the TPSm register, and CKm1 is selected by using bits 7 to 4 of the TPSm register. In addition, only for channels 1 and 3, CKm2 and CKm3 can be also selected. CKm2 is selected by using bits 9 and 8 of the TPSm register, and CKm3 is selected by using bits 13 and 12 of the TPSm register. 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 operating clock (CKSmn1, CKSmn0 = 00B) 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 operating clock (CKSmn1, CKSmn0 = 10B) are stopped (TEmn = 0).
If the PRSm20 and PRSm21 bits can be rewritten (n = 1, 3):
All channels for which CKm2 is selected as the operating clock (CKSmn1, CKSmn0 = 01B) are stopped (TEmn = 0).
If the PRSm30 and PRSm31 bits can be rewritten (n = 1, 3):
All channels for which CKm3 is selected as the operating clock (CKSmn1, CKSmn0 = 11B) are stopped (TEmn = 0).

The TPSm register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 7 - 13 Format of Timer Clock Select Register m (TPSm) (1/2)

Address: F01B6H, F01B7H (TPS0), F01F6H, F01F7H (TPS1)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TPSm	0	0	PRSm31	PRSm30	0	0	PRSm21	PRSm20

7	6	5	4	3	2	1	0
PRSm13	PRSm12	PRSm11	PRSm10	PRSm03	PRSm02	PRSm01	PRSm00

PRS mk3	PRS mk2	PRS mk1	PRS mk0	Selection of operating clock (CKmk)Note (k = 0, 1)				
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 32 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz	32 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	8 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	4 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz	2 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156 kHz	313 kHz	625 kHz	1 MHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz	500 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	250 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	62.5 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	31.3 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	15.6 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz	3.91 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz	1.95 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61.0 Hz	153 Hz	305 Hz	610 Hz	977 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop timer array unit (TTm = 00FFH). The timer array unit must also be stopped if the operating clock (f_{MCK}) or the valid edge of the signal input from the TImn pin is selected.

Caution 1. Be sure to leave bits 15, 14, 11, and 10 set to 0.

Caution 2. If f_{CLK} (no division) is selected as the operating clock (CKmk) and TDRmn is set to 0000H (m = 0, 1, n = 0 to 7), interrupt requests output from timer array units cannot be used.

Remark 1. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 2. Waveform of the clock to be selected in the TPSm register which becomes high level for one period of f_{CLK} from its rising edge (m = 0, 1). For details, see **7.5.1 Counter clock (f_{TCLK})**.

Figure 7 - 13 Format of Timer Clock Select Register m (TPSm) (2/2)

PRS m21	PRS m20	Selection of operating clock (CKm2)Note					
			f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 32 MHz
0	0	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz
0	1	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	8 MHz
1	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz	2 MHz
1	1	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz	500 kHz

PRS m31	PRS m30	Selection of operating clock (CKm3)Note					
			f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 32 MHz
0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz
0	1	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	31.3 kHz
1	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz
1	1	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz	1.95 kHz

Note 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}) or the valid edge of the signal input from the TImn pin is selected.

Caution Be sure to leave bits 15, 14, 11, and 10 set to 0.

By using channels 1 and 3 in the 8-bit timer mode and specifying CKm2 or CKm3 as the operating clock, the interval times shown in **Table 7 - 4** can be achieved by using the interval timer function.

Table 7 - 4 Interval Times Available for Operating Clock CKSm2 or CKSm3

Clock		Interval time[Note] (f_{CLK} = 32 MHz)			
		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%.

Remark 1. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 2. For details of a signal of $f_{CLK}/2^j$ (j = 2 to 15) selected with the TPSm register, see **7.5.1 Counter clock (f_{TCLK})**.

7.3.4 Timer mode register mn (TMRmn)

The TMRmn register sets an operation mode of channel n. This register is used to select the operating clock (f_{MCK}), select the counter 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 **7.8 Independent Channel Operation Function of Timer Array Unit** and **7.9 Simultaneous Channel Operation Function of Timer Array Unit**. The TMRmn register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Caution The bit function assigned to bit 11 of the TMRmn register depends on the channel.
TMRm2, TMRm4, TMRm6: MASTERmn bit (n = 2, 4, 6)
TMRm1, TMRm3: SPLITmn bit (n = 1, 3)
TMRm0, TMRm5, TMRm7: Fixed to 0

Figure 7 - 14 Format of Timer Mode Register mn (TMRmn) (1/3)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07), F01D0H, F01D1H (TMR10) to F01DEH, F01DFH (TMR17)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TMRmn (n = 2, 4, 6)	CKSmn1	CKSmn0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0
	7	6	5	4	3	2	1	0
	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8
TMRmn (n = 1, 3)	CKSmn1	CKSmn0	0	CCSmn	SPLITmn	STSmn2	STSmn1	STSmn0
	7	6	5	4	3	2	1	0
	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8
TMRmn (n = 0, 5, 7)	CKSmn1	CKSmn0	0	CCSmn	0Note 1	STSmn2	STSmn1	STSmn0
	7	6	5	4	3	2	1	0
	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

CKSmn1	CKSmn0	Selection of operating clock (f_{MCK}) of channel n
0	0	Operating clock CKm0 set by timer clock select register m (TPSm)
0	1	Operating clock CKm2 set by timer clock select register m (TPSm)
1	0	Operating clock CKm1 set by timer clock select register m (TPSm)
1	1	Operating clock CKm3 set by timer clock select register m (TPSm)
Operating clock (f_{MCK}) is used by the edge detector. A counter clock (f_{TCLK}) and a sampling clock are generated depending on the setting of the CCSmn bit. The operating clocks CKm2 and CKm3 can only be selected for channels 1 and 3.		

CCSmn	Selection of counter clock (f_{TCLK}) of channel n
0	Operating clock (f_{MCK}) specified by the CKSmn1 and CKSmn0 bits
1	Valid edge of input signal input from the TImn pin • In the case of unit 0: In channels 0 to 4, valid edge of input signal selected by TIS1n In channel 5, valid edge of input signal selected by TIS0 In channel 7, valid edge of input signal selected by ISC
Counter clock (f_{TCLK}) is used for the counter, output controller, and interrupt controller.	

Figure 7 - 14 Format of Timer Mode Register mn (TMRmn) (2/3)

Bit 11 of TMRmn (n = 2, 4, 6)

MASTERmn	Selection between using channel n independently or simultaneously with other channels (as a slave or master channel)
0	Channel n operates in the independent channel operation function, or as a slave channel in the simultaneous channel operation function.
1	Channel n operates as a master channel in the simultaneous channel operation function.
Only channel 2, 4, or 6 can be set as a master channel by setting the corresponding MASTERmn bit to 1. This bit for channels 0, 5, and 7 is fixed to 0. In any case, channel 0 would operate as a master channel regardless of the bit setting. Set the corresponding MASTERmn bit to 0 for a channel that is to be used independently.	

Bit 11 of TMRmn (n = 1, 3)

SPLITmn	Selection of 8 or 16-bit timer operation for channels 1 and 3
0	Channel n operates as a 16-bit timer; that is, it operates in the independent channel operation function, or as a slave channel in the simultaneous channel operation function.
1	Channel n operates as an 8-bit timer.

STSmn2	STSmn1	STSmn0	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

CISmn1	CISmn0	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 STSmn2 to STSmn0 is other than 010B, set CISmn1 and CISmn0 to 10B.		

Figure 7 - 14 Format of Timer Mode Register mn (TMRmn) (3/3)

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 low- or high-level width of input signal	Counting up
Other than above			Setting prohibited		
The operation in each mode varies depending on the MDmn0 bit. See the table below for details.					

Operation mode set by the MDmn3 to MDmn1 bits (see the table above)	MDmn0	Setting of starting counting and interrupt
• Interval timer mode (000B) • Capture mode (010B)	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 (011B)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• One-count mode[Note 2] (100B)	0	Start trigger is invalid during counting operation. At that time, interrupt is not generated.
	1	Start trigger is valid during counting operation[Note 3]. At that time, interrupt is not generated.
• Capture & one-count mode (110B)	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.
Other than above		Setting prohibited

Note 1. Bit 11 is a read-only bit and fixed to 0. Writing to this bit is ignored.

Note 2. In one-count mode, interrupt output (INTTMmn) when starting a count operation and TOmn output are not controlled.

Note 3. If the start trigger (TSmn = 1) is issued during operation, the counter is initialized, and recounting is started (does not occur the interrupt request).

Caution 1. Be sure to leave bits 13, 5, and 4 set to 0.

Caution 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 CKSmn1 and CKSmn0 bits (f_{MCK}) or the valid edge of the signal input from the TImn pin is selected as the counter clock (f_{TCLK}).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

7.3.5 Timer status register mn (TSRmn)

The TSRmn register indicates the overflow state of the counter of channel n. The TSRmn register is valid only in the capture mode (MDmn3 to MDmn1 = 010B) and capture & one-count mode (MDmn3 to MDmn1 = 110B). See **Table 7 - 5** for the operation of the OVF bit in each operation mode and set/clear conditions. The TSRmn register can be read by a 16-bit memory manipulation instruction. The 8 lower-order bits of a TSRmn register can be handled as the TSRmnL register, which can be read by an 8-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 7 - 15 Format of Timer Status Register mn (TSRmn)

Address: F01A0H, F01A1H (TSR00) to F01AEH, F01AFH (TSR07), F01E0H, F01E1H (TSR10) to F01EEH, F01EFH (TSR17)

After reset: 0000H

R/W: R

Symbol	15	14	13	12	11	10	9	8
TSRmn	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	OVF

OVF	Counter overflow state 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.	

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

Table 7 - 5 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.

7.3.6 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 8 lower-order bits of a TEm register can be handled as the TEmL register, which can be read by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 7 - 16 Format of Timer Channel Enable Status Register m (TEm)

Address: F01B0H, F01B1H (TE0), F01F0H, F01F1H (TE1)
After reset: 0000H
R/W: R

Symbol	15	14	13	12	11	10	9	8
TEm	0	0	0	0	TEHm3	0	TEHm1	0

7	6	5	4	3	2	1	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 enabled or stopped state 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.	

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

7.3.7 Timer channel start register m (TSm)

The TSm register is used to initialize the timer counter register mn (TCRmn) and to start counting by the given channel. Setting a bit of this register to 1 leads to the corresponding bit of the timer channel enable status register m (TEm) being set to 1. The TSmn, TSHm1, and TSHm3 bits are immediately cleared when operation is enabled; that is, after the TEmn, TEHm1, and TEHm3 bits are set to 1. The TSm register can be set by a 16-bit memory manipulation instruction. The 8 lower-order bits of a TSm register can be handled as the TSmL register, which can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 7 - 17 Format of Timer Channel Start Register m (TSm)

Address: F01B2H, F01B3H (TS0), F01F2H, F01F3H (TS1)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TSm	0	0	0	0	TSHm3	0	TSHm1	0

7	6	5	4	3	2	1	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 counting is enabled. In the counter-operation enabled state, counting by the TCRm3 register starts in the interval timer mode. See **Table 7 - 6** in **7.5.2 Timing of the start of counting**.

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 counting is enabled. In the counter-operation enabled state, counting by the TCRm1 register starts in the interval timer mode. See **Table 7 - 6** in **7.5.2 Timing of the start of counting**.

TSmn	Trigger to enable operation (start operation) of channel n
0	No trigger operation
1	The TEmn bit is set to 1 and counting is enabled. In the counter-operation enabled state, the timer operation at the start of counting by the TCRmn register depends on the operation mode. See **Table 7 - 6** in **7.5.2 Timing of the start of counting**. For channel 1 or 3 in the 8-bit timer mode, the TSm1 or TSm3 bit respectively is the trigger to enable operation (start operation) of the lower 8-bit timer.

(Cautions and Remarks are listed on the next page.)

Caution 1. Be sure to leave bits 15 to 12, 10, and 8 set to 0.

Caution 2. In the following cases, an interval of waiting as stated below is required from setting of the timer mode register mn (TMRmn) until setting of the TSmn (TSHm1, TSHm3) bit to 1.

- **The setting of the CCSmn bit of the TMRmn register is changed from 0 to 1.**
- **The setting of the STSmn2 to STSmn0 bits is changed from 000B to 001B or from 000B to 010B.**

The required intervals of waiting depend on whether the TImn pin noise filter is enabled or disabled and are as follows.

- **Four cycles of the operating clock (f_{MCK}) when the TImn pin noise filter is enabled (TNFENmn = 1)**
- **Two cycles of the operating clock (f_{MCK}) when the TImn pin noise filter is disabled (TNFENmn = 0)**

Remark 1. When the TSm register is read, 0 is always read.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

7.3.8 Timer channel stop register m (TTm)

The TTm register is used to stop counting by the given channel. Setting a bit of this register to 1 leads to the corresponding bit of the timer channel enable status register m (TEm) being cleared to 0. The TTmn, TTHm1, and TTHm3 bits are immediately cleared when operation is stopped; that is, after the TEmn, TEHm1, and TEHm3 bits are cleared to 0. The TTm register can be set by a 16-bit memory manipulation instruction. The 8 lower-order bits of a TTm register can be handled as the TTmL register, which can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 7 - 18 Format of Timer Channel Stop Register m (TTm)

Address: F01B4H, F01B5H (TT0), F01F4H, F01F5H (TT1)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TTm	0	0	0	0	TTHm3	0	TTHm1	0

7	6	5	4	3	2	1	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	The TEHm3 bit is cleared to 0 and counting stops.

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	The TEHm1 bit is cleared to 0 and counting stops.

TTmn	Trigger to stop operation of channel n
0	No trigger operation
1	The TEmn bit is cleared to 0 and counting stops. For channel 1 or 3 in the 8-bit timer mode, the TTm1 or TTm3 bit respectively is the trigger to stop operation of the lower 8-bit timer.

Caution Be sure to leave bits 15 to 12, 10, and 8 set to 0.

Remark 1. When the TTm register is read, 0 is always read.
Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

7.3.9 Timer input select register 0 (TIS0)

The TIS0 register is used to select the channel 5 of unit 0 timer input. The TIS0 register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 7 - 19 Format of Timer Input Select Register 0 (TIS0)

Address: F0074H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	0	0	TIS02	TIS01	TIS00

TIS02	TIS01	TIS00	Selection of timer input used with channel 5
0	0	0	Input signal of timer input pin (TI05)
0	0	1	Event input signal from ELCL
0	1	1	Middle-speed on-chip oscillator peripheral clock (f_{IMP})
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
1	0	1	Subsystem clock (f_{SUB})
Other than above			Setting prohibited

Caution Make sure that both the low-level and high-level widths of timer input to be selected are no less than 1/f_{MCK} + 10 ns. Therefore, when selecting f_{SUB} as f_{CLK} (the CSS bit of the CKC register = 1), the TIS02 bit cannot be set to 1.

7.3.10 Timer input select register 1 (TIS1)

The TIS1 register is used to select channels 0 and 1 of unit 0 timer input. The TIS1 register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 7 - 20 Format of Timer Input Select Register 1 (TIS1)

Address: F0075H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TIS1	0	0	0	0	0	0	TIS11	T1S10

TIS1n	Selection of timer input used with channel n
0	Timer input pin (TI0n)
1	Event input signal from ELCL

Remark n = 0, 1

7.3.11 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. The 8 lower-order bits of a TOEm register can be handled as the TOEmL register, which can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 7 - 21 Format of Timer Output Enable Register m (TOEm)

Address: F01BAH, F01BBH (TOE0), F01FAH, F01FBH (TOE1)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TOEm	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TOEm7	TOEm6	TOEm5	TOEm4	TOEm3	TOEm2	TOEm1	TOEm0

TOEmn	Enabling/disabling timer output for channel n
0	Disables timer output. The corresponding TOmn bit does not reflect timer operation with this setting, so the output level of a TOmn bit is fixed to the level written to the TOm register. Writing to the TOmn bit is enabled and the level set in the TOmn bit is output from the TOmn pin.
1	Enables timer output. The corresponding TOmn bit reflects timer operation with this setting, so the output waveform is generated. Writing to the TOmn bit is ignored.

Caution **Be sure to leave bits 15 to 8 set to 0.**

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

7.3.12 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 on 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. To use the P00/TI00, P01/TO00, P16/TI01/TO01, P17/TI02/TO02, P31/TI03/TO03, P42/TI04/TO04, P46/TI05/TO05, P102/TI06/TO06, P145/TI07/TO07, P64/TI10/TO10, P65/TI11/TO11, P66/TI12/TO12, P67/TI13/TO13, P103/TI14/TO14, P104/TI15/TO15, P105/TI16/TO16, or P106/TI17/TO17 pin as a port function pin, set the corresponding TOmn bit to 0. The TOm register can be set by a 16-bit memory manipulation instruction. The 8 lower-order bits of a TOm register can be handled as the TOmL register, which can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 7 - 22 Format of Timer Output Register m (TOm)

Address: F01B8H, F01B9H (TO0), F01F8H, F01F9H (TO1)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TOm	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	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 leave bits 15 to 8 set to 0.**

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

7.3.13 Timer output level register m (TOLm)

The TOLm register 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. The 8 lower-order bits of a TOLm register can be handled as the TOLmL register, which can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 7 - 23 Format of Timer Output Level Register m (TOLm)

Address: F01BCH, F01BDH (TOL0), F01FCH, F01FDH (TOL1)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TOLm	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
TOLm7	TOLm6	TOLm5	TOLm4	TOLm3	TOLm2	TOLm1	0

TOLmn	Control of timer output level of channel n
0	Positive logic output (active-high)
1	Negative logic output (active-low)

Caution Be sure to leave bits 15 to 8, and 0 set to 0.

Remark 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.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

7.3.14 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. The 8 lower-order bits of a TOMm register can be handled as the TOMmL register, which can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 7 - 24 Format of Timer Output Mode Register m (TOMm)

Address: F01BEH, F01BFH (TOM0), F01FEH, F01FFH (TOM1)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
TOMm	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	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 toggled output by timer interrupt request signal (INTTMmn))
1	Slave channel output mode (output is set by the timer interrupt request signal (INTTMmn) of the master channel, and reset by the timer interrupt request signal (INTTM0p) of the slave channel)

Caution **Be sure to leave bits 15 to 8, and 0 set to 0.**

Remark 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, see **7.4.1 Basic specifications of the simultaneous channel operation function**.)

7.3.15 Input switch control register (ISC)

The ISC1 and ISC0 bits of the ISC register are used to implement LIN-bus communication operation by using channel 7 in association with the serial array unit. When the ISC1 bit is set to 1, the input signal of the serial data input pin (RxD2) is selected as a timer input signal. The ISC register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 7 - 25 Format of Input Switch Control Register (ISC)

Address: F0073H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ISC	ISC7	ISC6	ISC5	ISC4	ISC3	ISC2	ISC1	ISC0

ISC1	Switching channel 7 input of timer array unit
0	Uses the input signal of the TI07 pin as a timer input (normal operation).
1	Input signal of the RxD2 pin is used as timer input (detects the wakeup signal and measures the low-level width of the break field and the pulse width of the sync field).

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 RxD2 pin as an external interrupt (wakeup signal detection).

Remark When the LIN-bus communication function is used, select the input signal of the RxD2 pin by setting ISC1 to 1.

7.3.16 Noise filter enable registers 1, 2 (NFEN1, NFEN2)

The NFEN1 and NFEN2 registers are 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 enabled, after synchronization with the operating clock (f_{MCK}) for the target channel, whether the signal keeps the same value for two clock cycles is detected. When the noise filter is disabled, the input signal is only synchronized with the operating clock (f_{MCK}) for the target channel[Note]. The NFEN1 and NFEN2 registers can be set by a 1-bit or 8-bit memory manipulation instruction. The value of each of these registers following a reset is 00H.

Note For details, see the following sections.

- **7.5.1 Counter clock (f_{TCLK}), 2.When valid edge of input signal via the TImn pin is selected (CCSmn = 1)**
- **7.5.2 Timing of the start of counting**
- **7.7 Timer Input (TImn) Control**

Figure 7 - 26 Format of Noise Filter Enable Register 1 (NFEN1)

Address: F0071H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
NFEN1	TNFEN07	TNFEN06	TNFEN05	TNFEN04	TNFEN03	TNFEN02	TNFEN01	TNFEN00

TNFEN07	Enabling/disabling use of the noise filter for the TI07 pin[Note]
0	Disables use of the noise filter for the TI07 pin.
1	Enables use of the noise filter for the TI07 pin.

TNFEN06	Enabling/disabling use of the noise filter for the TI06 pin
0	Disables use of the noise filter for the TI06 pin.
1	Enables use of the noise filter for the TI06 pin.

TNFEN05	Enabling/disabling use of the noise filter for the TI05 pin
0	Disables use of the noise filter for the TI05 pin.
1	Enables use of the noise filter for the TI05 pin.

TNFEN04	Enabling/disabling use of the noise filter for the TI04 pin
0	Disables use of the noise filter for the TI04 pin.
1	Enables use of the noise filter for the TI04 pin.

TNFEN03	Enabling/disabling use of the noise filter for the TI03 pin
0	Disables use of the noise filter for the TI03 pin.
1	Enables use of the noise filter for the TI03 pin.

TNFEN02	Enabling/disabling use of the noise filter for the TI02 pin
0	Disables use of the noise filter for the TI02 pin.
1	Enables use of the noise filter for the TI02 pin.

TNFEN01	Enabling/disabling use of the noise filter for the TI01 pin
0	Disables use of the noise filter for the TI01 pin.
1	Enables use of the noise filter for the TI01 pin.

TNFEN00	Enabling/disabling use of the noise filter for the TI00 pin
0	Disables use of the noise filter for the TI00 pin.
1	Enables use of the noise filter for the TI00 pin.

Note The applicable pin can be switched by setting the ISC1 bit of the ISC register.
ISC1 = 0: Whether or not to use the noise filter of the TI07 pin can be selected.
ISC1 = 1: Whether or not to use the noise filter of the RxD2 pin can be selected.

Remark Which of the timer I/O pins for channels 0 to 7 are present and absent depends on the product. See **Table 7 - 2 Timer I/O Pins Provided in Each Product** for details.

Figure 7 - 27 Format of Noise Filter Enable Register 2 (NFEN2)

Address: F0072H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
NFEN2	TNFEN17	TNFEN16	TNFEN15	TNFEN14	TNFEN13	TNFEN12	TNFEN11	TNFEN10

TNFEN17	Enabling/disabling use of the noise filter for the TI17 pin
0	Disables use of the noise filter for the TI17 pin.
1	Enables use of the noise filter for the TI17 pin.

TNFEN16	Enabling/disabling use of the noise filter for the TI16 pin
0	Disables use of the noise filter for the TI16 pin.
1	Enables use of the noise filter for the TI16 pin.

TNFEN15	Enabling/disabling use of the noise filter for the TI15 pin
0	Disables use of the noise filter for the TI15 pin.
1	Enables use of the noise filter for the TI15 pin.

TNFEN14	Enabling/disabling use of the noise filter for the TI14 pin
0	Disables use of the noise filter for the TI14 pin.
1	Enables use of the noise filter for the TI14 pin.

TNFEN13	Enabling/disabling use of the noise filter for the TI13 pin
0	Disables use of the noise filter for the TI13 pin.
1	Enables use of the noise filter for the TI13 pin.

TNFEN12	Enabling/disabling use of the noise filter for the TI12 pin
0	Disables use of the noise filter for the TI12 pin.
1	Enables use of the noise filter for the TI12 pin.

TNFEN11	Enabling/disabling use of the noise filter for the TI11 pin
0	Disables use of the noise filter for the TI11 pin.
1	Enables use of the noise filter for the TI11 pin.

TNFEN10	Enabling/disabling use of the noise filter for the TI10 pin
0	Disables use of the noise filter for the TI10 pin.
1	Enables use of the noise filter for the TI10 pin.

Remark Which of the timer I/O pins for channels 0 to 7 are present and absent depends on the product. See **Table 7 - 2 Timer I/O Pins Provided in Each Product** for details.

7.3.17 Registers for controlling the port functions multiplexed with the inputs and outputs of timers

Set the following registers to control the port functions multiplexed with the inputs and outputs of timer array units.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- Port function output enable registers (PFOEx)

For details, see the following sections.

- **4.3.1 Port mode registers (PMxx)**
- **4.3.2 Port registers (Pxx)**
- **4.3.5 Port output mode registers (POMxx)**
- **4.3.7 Port mode control A registers (PMCAxx)**
- **4.3.8 Port mode control T registers (PMCTxx)**
- **4.3.9 Port mode control E registers (PMCExx)**
- **4.3.15 Port function output enable registers (PFOEx)**

For examples of register settings for controlling port functions, see **4.5.4 Examples of register settings for port and multiplexed functions**.

7.4 Basic Specifications of the Timer Array Unit

7.4.1 Basic specifications of the simultaneous channel operation function

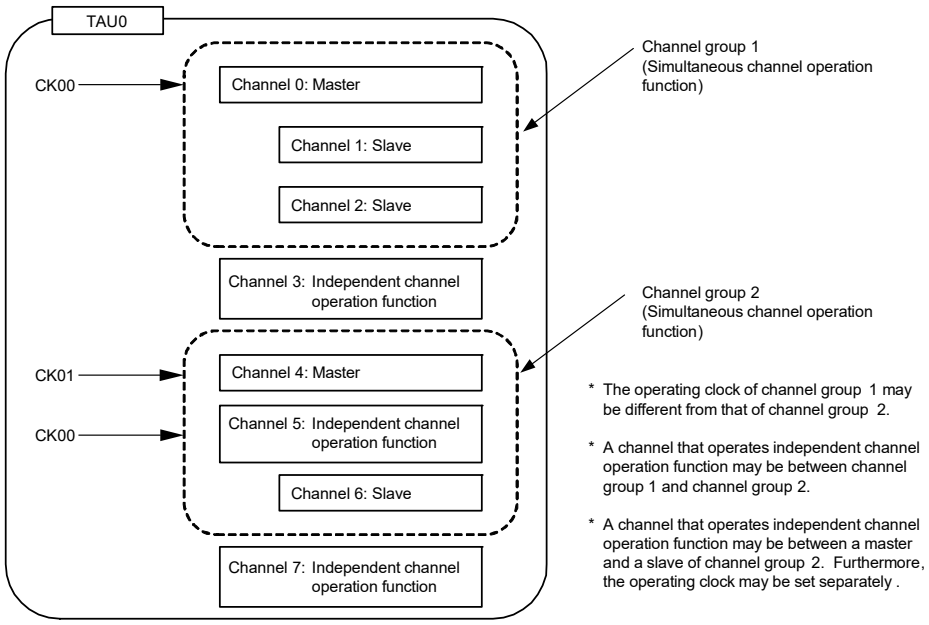
When simultaneously using multiple channels, namely, a combination of a master channel (a reference timer mainly counting the cycle) and slave channels (timers operating according to the master channel). The following lists the basic specifications of this function.

1. Only an even channel (channels 0, 2, 4, and 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. For example, if channel 2 is set as a master channel, channel 3 and those that follow (i.e., channels 3, 4, 5, etc.) 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. For example, if channels 0 and 4 are set as master channels, channels 1 to 3 can be set as the slave channels of master channel 0, whereas channels 5 to 7 cannot be set as the slave channels 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 CKSmn1 and CKSmn0 bits (bits 15 and 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 counter clock to the lower channels.
8. A slave channel can use INTTMmn (interrupt), a start software trigger, or the counter clock of the master channel as a source clock, but cannot transmit its own INTTMmn (interrupt), start software trigger, or counter clock to channels with lower channel numbers.
9. A master channel cannot use INTTMmn (interrupt), a start software trigger, or the counter clock from the other higher master channel as a source clock.
10. To simultaneously start channels that operate in combination, the operation enable (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 operation 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. In the timer mode register m0 (TMRm0), the MASTERm0 bit for selecting a slave or master channel is not assigned to bit 11, which is fixed to 0 instead. Note, however, that channel 0 operates as a master channel during simultaneous operation.

The specifications 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 specifications of the simultaneous channel operation function in **7.4.1 Basic specifications of the simultaneous channel operation function** do not apply to the channel groups.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

Example


TAU0
CK00
Channel 0: Master
Channel 1: Slave
Channel 2: Slave
Channel 3: Independent channel operation function
CK01
CK00
Channel 4: Master
Channel 5: Independent channel operation function
Channel 6: Slave
Channel 7: Independent channel operation function
Channel group 1 (Simultaneous channel operation function)
Channel group 2 (Simultaneous channel operation function)
* The operating clock of channel group 1 may be different from that of channel group 2.
* A channel that operates independent channel operation function may be between channel group 1 and channel group 2.
* A channel that operates independent channel operation function may be between a master and a slave of channel group 2. Furthermore, the operating clock may be set separately .

7.4.2 Basic specifications of the 8-bit timer function (only for channels 1 and 3)

The 8-bit timer function allows the use of a 16-bit timer configured by cascade-connecting two 8-bit timer channels. This function can only be used with channels 1 and 3. The following lists the basic specifications of this function.

1. This 8-bit timer function (cascade connection to form a 16-bit timer) can only be used with channels 1 and 3.
2. To use the two channels as 8-bit timers, set the SPLITmn bits of timer mode registers mn (TMRmn) to 1.
3. The higher-numbered 8-bit timer can be used as an interval timer.
4. INTTMm1H and INTTMm3H interrupts are output at the start of operation by the higher-numbered 8-bit timer (which is the same as the operation when MDmn0 is set to 1).
5. Selection of the operating clock for the higher-numbered 8-bit timer is in accord with the setting of the CKSmn1 and CKSmn0 bits of the TMRmn register for the lower-numbered 8-bit timer.
6. Set the TSHm1 and TSHm3 bits to start operation of the higher-numbered 8-bit timer as an 8-bit timer, and the TTHm1 and TTHm3 bits to stop operation. The channel state can be checked by reading the TEHm1 and TEHm3 bits.
7. Operation of the lower-numbered 8-bit timer is in accord with the setting of the TMRmn register, through which any of the following three functions is selectable.
   - Interval timer square wave output
   - External event counter
   - Delay counter
8. Set the TSm1 and TSm3 bits to start operation of the lower-numbered 8-bit timer as an 8-bit timer, and the TTm1 and TTm3 bits to stop operation. The channel state can be checked by reading the TEm1 and TEm3 bits.
9. When the channels are to be used as a 16-bit timer, setting the TSHm1, TSHm3, TTHm1, and TTHm3 bits has no effect. Set the TSm1, TSm3, TTm1, and TTm3 bits to start operation by both channels 1 and 3. These trigger bits do not affect the TEHm3 and TEHm1 bits.
10. The simultaneous operations (one-shot pulse, PWM, and multiple PWM) cannot be used when the 8-bit timer functions are selected by the SPLITmn bits.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3)

7.5 Operations of Counters

7.5.1 Counter clock (fTCLK)

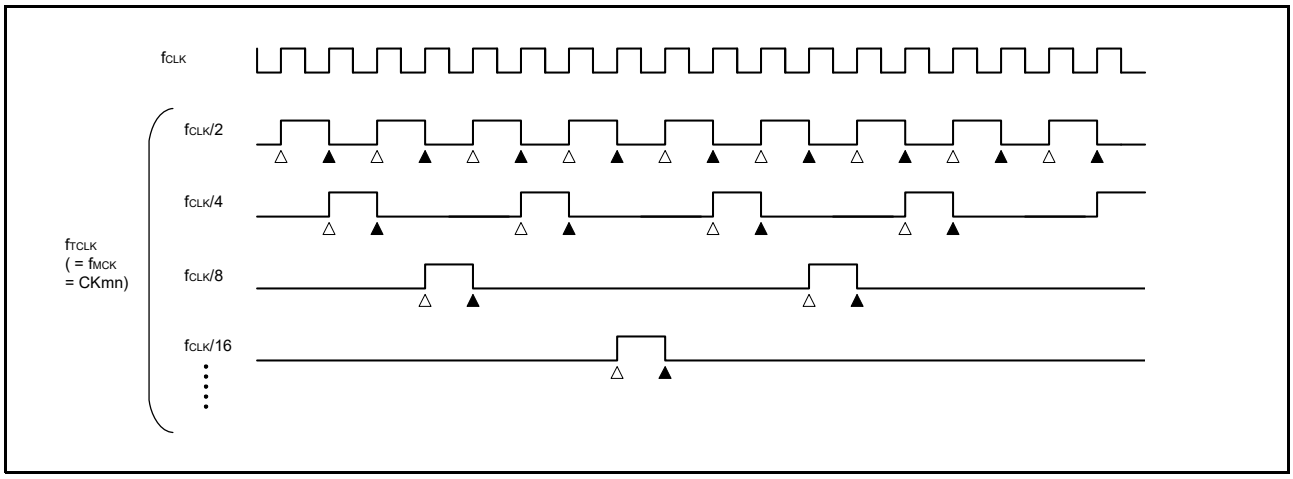
The counter clock (fTCLK) of the timer array unit can be selected between following by CCSmn bit of timer mode register mn (TMRmn).

- Operating clock (fMCK) specified by the CKSmn1 and CKSmn0 bits
- Valid edge of input signal input from the TImn pin

Because the timer array unit is designed to operate in synchronization with fCLK, the timings of the counter clock (fTCLK) are shown below.

1. When operating clock (fMCK) specified by the CKSmn1 and CKSmn0 bits is selected (CCSmn = 0)
   The counter clock (fTCLK) is between fCLK to fCLK/2^{15} by setting of timer clock select register m (TPSm). When a divided fCLK is selected, however, the clock selected in TPSmn register, but a signal which becomes high level for one period of fCLK from its rising edge. When a fCLK is selected, fixed to high level. Counting of timer counter register mn (TCRmn) delayed by one period of fCLK from rising edge of the counter clock, because of synchronization with fCLK. But, this is described as “counting at rising edge of the counter clock”, as a matter of convenience.

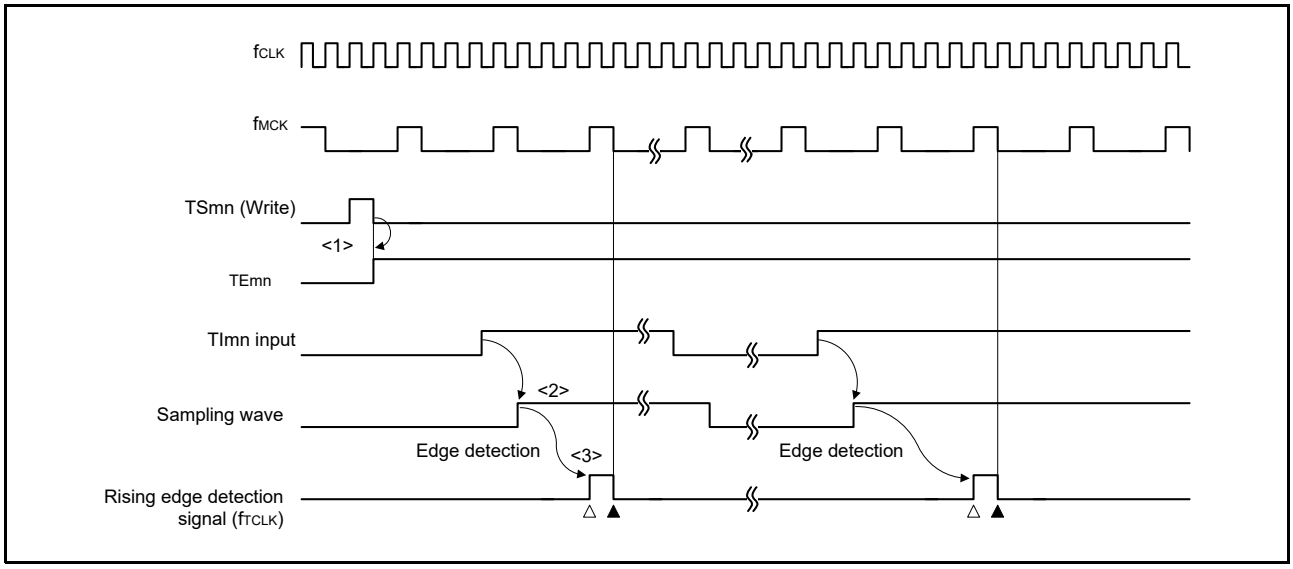
Figure 7 - 28 Timing of fCLK and Counter Clock (fTCLK) (When CCSmn = 0)



Remark 1. △ : Rising edge of the counter clock
▲ : Synchronization, increment/decrement of counter
Remark 2. fCLK: CPU/peripheral hardware clock

2. When valid edge of input signal via the TImn pin is selected (CCSmn = 1)
   The counter clock (f_{TCLK}) becomes the signal that detects valid edge of input signal via the TImn pin and synchronizes next rising f_{MCK}. The counter clock (f_{TCLK}) is delayed for 1 to 2 period of f_{MCK} from the input signal via the TImn pin (when a noise filter is used, the delay becomes 3 to 4 clock). Counting of timer counter register mn (TCRmn) delayed by one period of f_{CLK} from rising edge of the counter clock, because of synchronization with f_{CLK}. But, this is described as “counting at valid edge of input signal via the TImn pin”, as a matter of convenience.

Figure 7 - 29 Timing of f_{CLK} and Counter 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 (counter clock) is output.

Remark 1. △ : Rising edge of the counter clock
▲ : Synchronization, increment/decrement of counter

Remark 2. f_{CLK}: CPU/peripheral hardware clock
f_{MCK}: Operating clock of channel n

Remark 3. The waveforms of the input signals via TImn pin of the input pulse interval measurement, the measurement of low- and high-level width of input signal, the delay counter, and the one-shot pulse output are the same.

7.5.2 Timing of the start of counting

Timer counter register mn (TCRmn) becomes enabled to operation by setting of TSmn bit of timer channel start register m (TSm). **Table 7 - 6** shows operations from the counter-operation enabled state to the start of counting by a timer counter register mn (TCRmn).

Table 7 - 6 Operations from the Counter-Operation Enabled State to the Start of Counting by a Timer Counter Register mn (TCRmn)

Timer operation mode	Operation when TSmn = 1 is set
Interval timer mode	No operation is carried out from start trigger detection (TSmn=1) until counter clock generation. The first counter clock loads the value of the TDRmn register to the TCRmn register and the subsequent counter clock performs count down operation (see **7.5.3 Operations of counters, 1. Operation in interval timer mode**).
Event counter mode	Writing 1 to the TSmn bit loads the value of the TDRmn register to the TCRmn register. If detect edge of TImn input. The subsequent counter clock performs count down operation (see **7.5.3 Operations of counters, 2. Operation in event counter mode**).
Capture mode	No operation is carried out from start trigger detection (TSmn = 1) until counter clock generation. The first counter clock loads 0000H to the TCRmn register and the subsequent counter clock performs count up operation (see **7.5.3 Operations of counters**, **3. Operation in 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 counter clock generation. The first counter clock loads the value of the TDRmn register to the TCRmn register and the subsequent counter clock performs count down operation (see **7.5.3 Operations of counters, 4. Operation in one-count mode**).
Capture & one-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0). No operation is carried out from start trigger detection until counter clock generation. The first counter clock loads 0000H to the TCRmn register and the subsequent counter clock performs count up operation (see **7.5.3 Operations of counters, 5. Operation in capture & one-count mode (high-level width measurement)**).

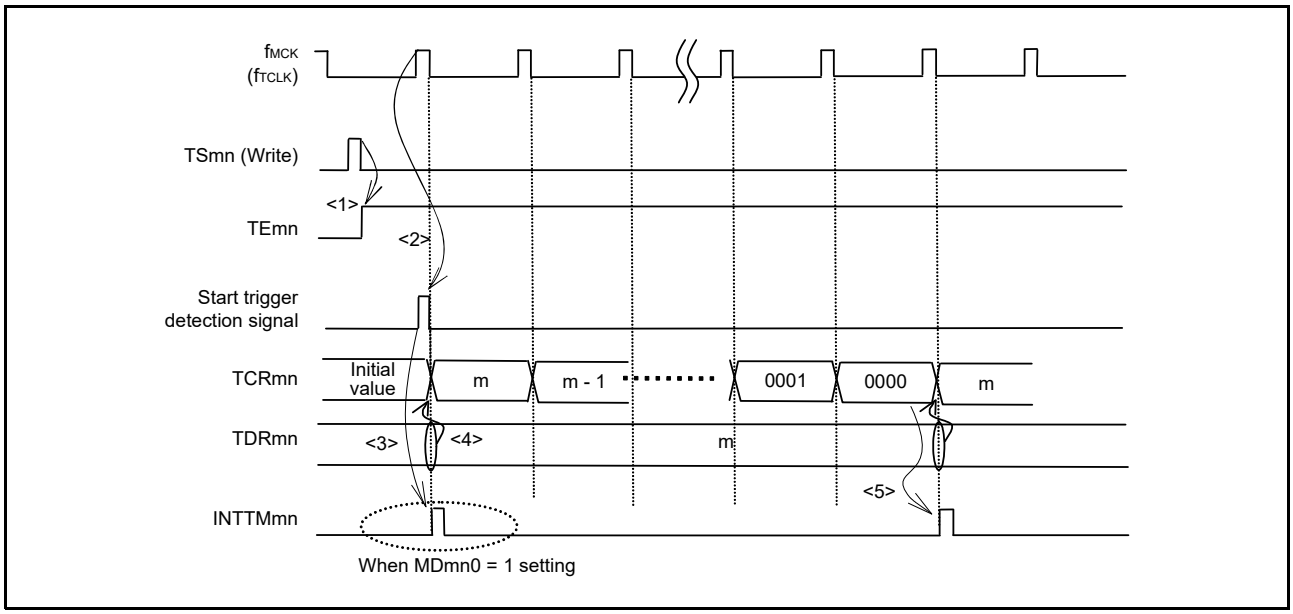
7.5.3 Operations of counters

Here, the counter operation in each mode is explained.

1. Operation in interval timer mode

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit. Timer counter register mn (TCRmn) holds the initial value until counter clock generation.
<2> A start trigger is generated at the first counter 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 counter 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 7 - 30 Timing during Operation in Interval Timer Mode



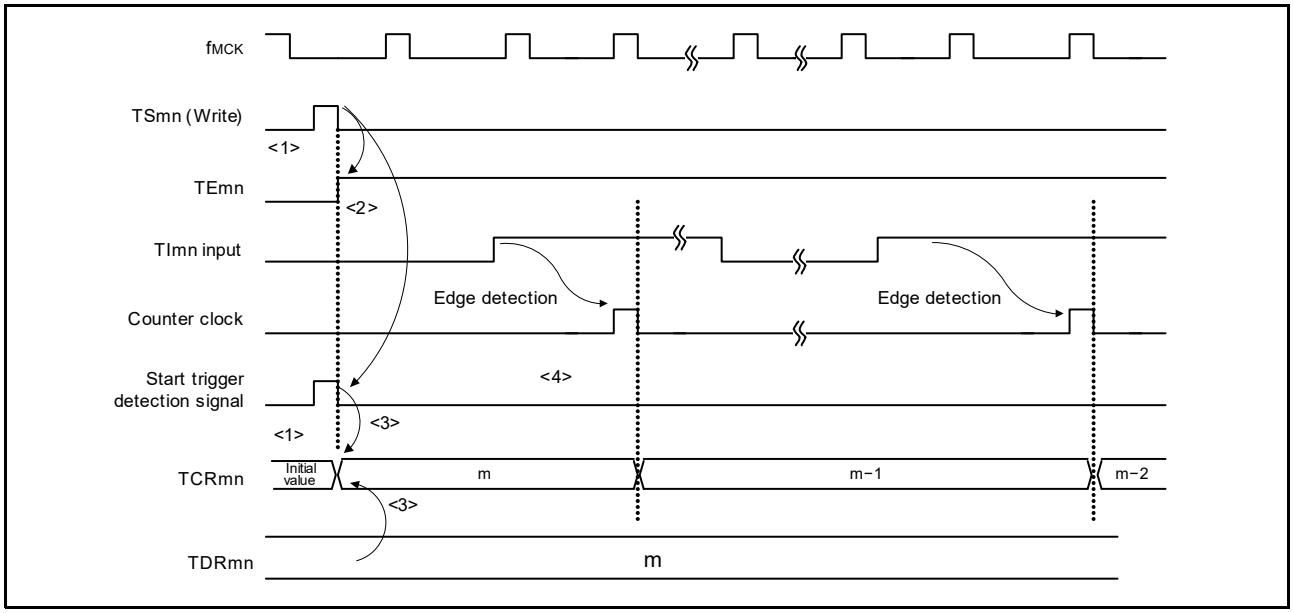
Caution **In the first cycle operation of counter clock after writing the TSmn bit, an error at a maximum of one clock is generated since count start delays until counter clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.**

Remark fMCK, the start trigger detection signal, and INTTMmn become active between one clock in synchronization with fCLK.

2. Operation in event counter mode

<1> Timer counter 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 counter clock of the valid edge of the TImn input.

Figure 7 - 31 Timing during Operation in Event Counter Mode

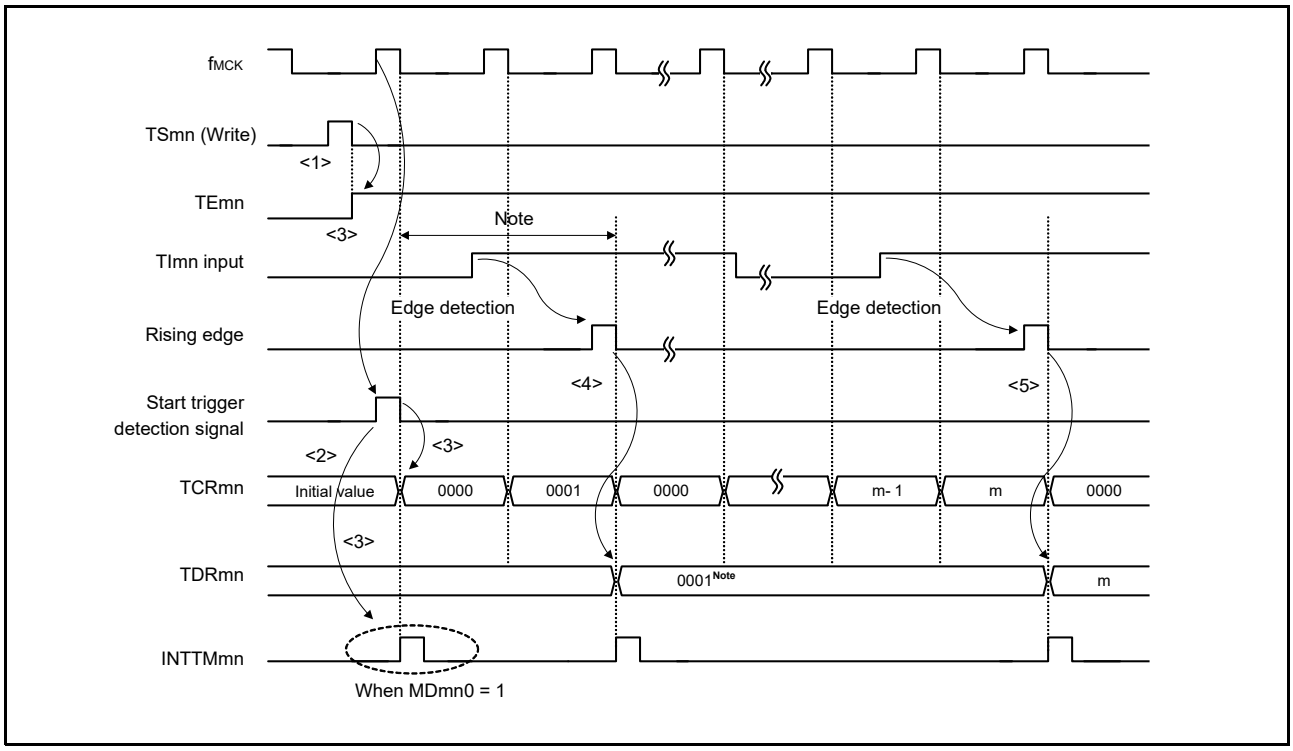


Remark **Figure 7 - 31** shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the counter clock (fMCK).

3. Operation in capture mode (input pulse interval measurement)

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.
<2> Timer counter register mn (TCRmn) holds the initial value until counter clock generation.
<3> A start trigger is generated at the first counter 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 captured value is meaningless. 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 7 - 32 Timing during Operation 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.

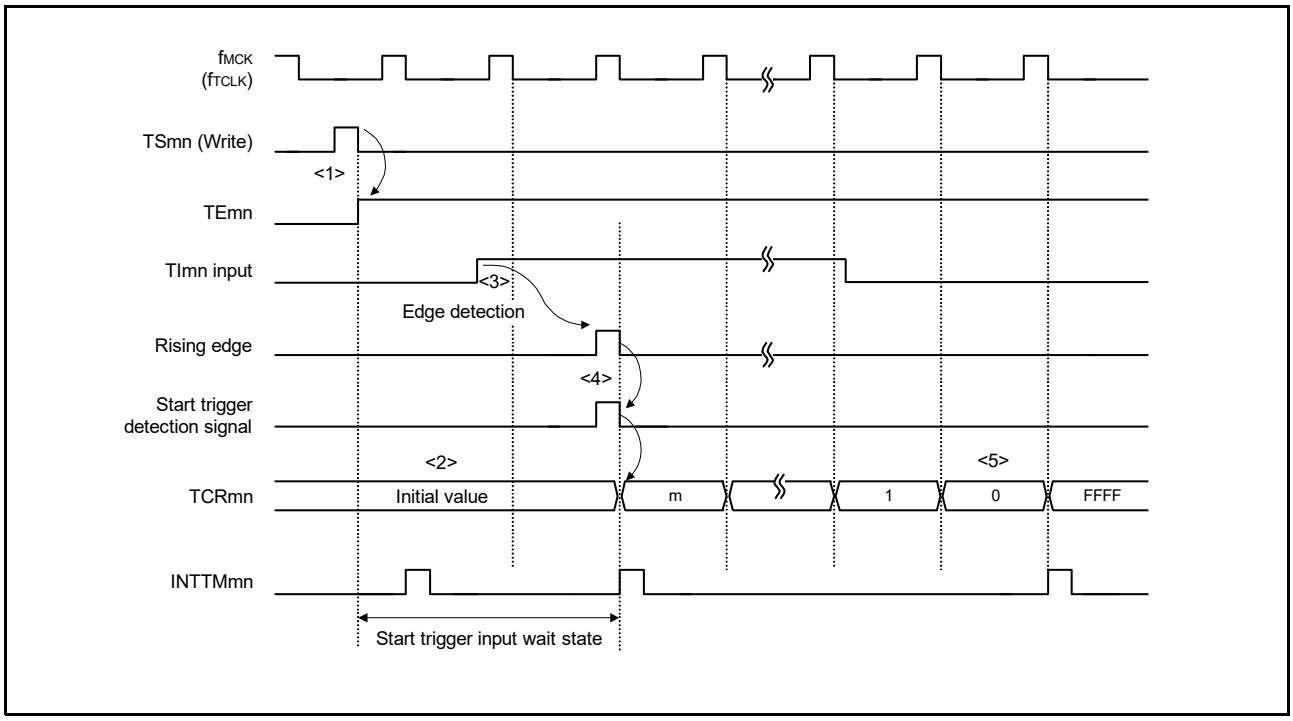
Caution **In the first cycle operation of counter clock after writing the TSmn bit, an error at a maximum of one clock is generated since count start delays until counter clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.**

Remark **Figure 7 - 32** 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 counter clock (fMCK).

4. Operation in one-count mode

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.
<2> Timer counter 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 7 - 33 Timing during Operation in One-count Mode

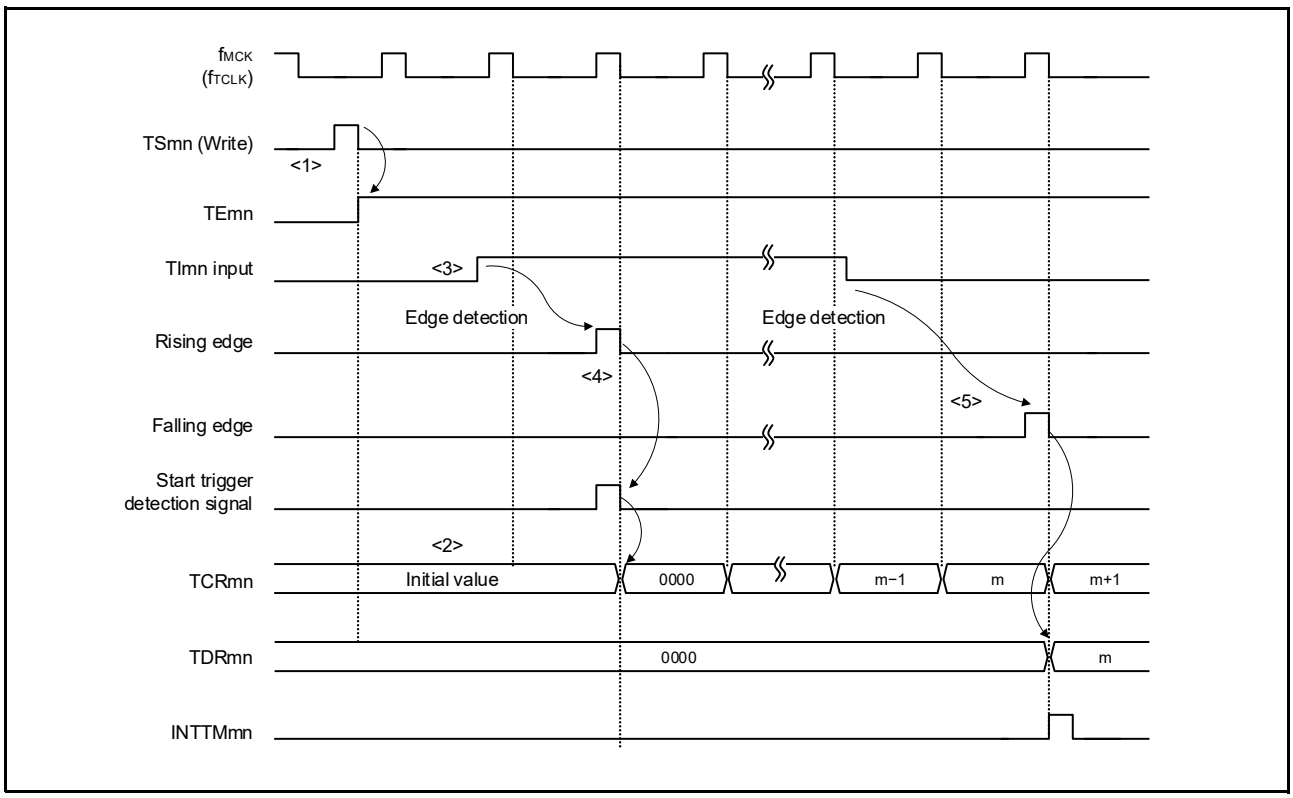


Remark **Figure 7 - 33** shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the counter clock (fMCK).

5. Operation in 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 counter 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 7 - 34 Timing during Operation in Capture & One-count Mode (High-level Width Measurement)

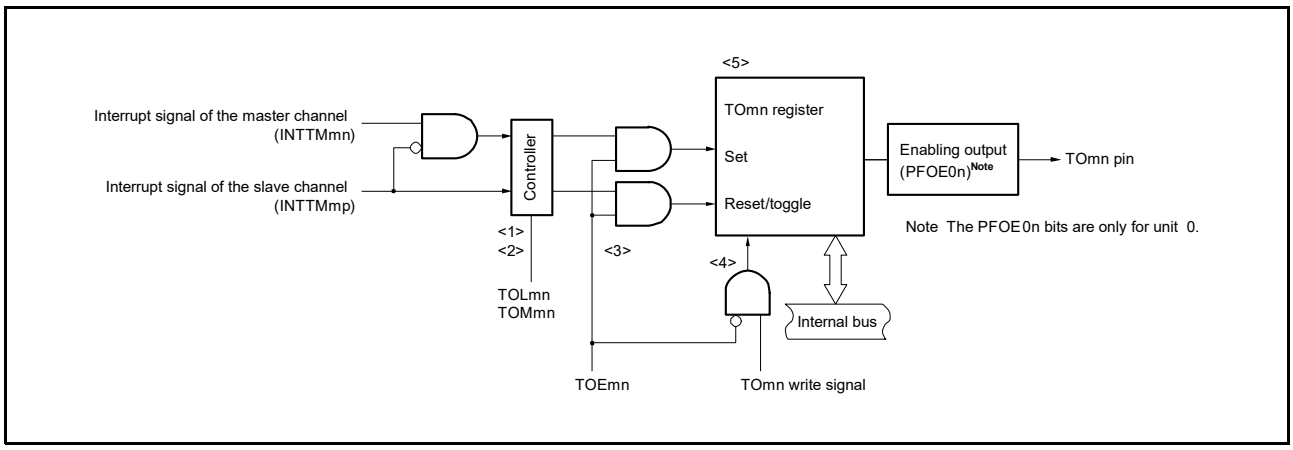


Remark **Figure 7 - 34** shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the counter clock (fMCK).

7.6 Channel Output (TOmn Pin) Control

7.6.1 TOmn pin output circuit configuration

Figure 7 - 35 Output Circuit Configuration



The following describes the TOmn pin output circuit.

<1> When TOMmn = 0 (master channel output mode), the set value of timer output level register m (TOLm) is ignored and only INTTM0p (slave channel timer interrupt) is transmitted to timer output register m (TOm).

<2> When TOMmn = 1 (slave channel output mode), both INTTMmn (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TOm register. At this time, the TOLm register becomes valid and the signals are controlled as follows:

- When TOLmn = 0: Positive logic output (INTTMmn → set, INTTM0p → reset)
- When TOLmn = 1: Negative logic output (INTTMmn → reset, INTTM0p → set)

When INTTMmn and INTTM0p are simultaneously generated, (0% output of PWM), INTTM0p (reset signal) takes priority, and INTTMmn (set signal) is masked.

<3> While timer output is enabled (TOEmn = 1), INTTMmn (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TOm register. Writing to the TOm register (TOmn write signal) becomes invalid. When TOEmn = 1, the TOmn pin output never changes with signals other than interrupt signals. To initialize the TOmn pin output level, it is necessary to set timer operation is stopped (TOEmn = 0) and to write a value to the TOm register.

<4> While timer output is disabled (TOEmn = 0), writing to the TOmn bit to the target channel (TOmn write signal) becomes valid. When timer output is disabled (TOEmn = 0), neither INTTMmn (master channel timer interrupt) nor INTTM0p (slave channel timer interrupt) is transmitted to the TOm register.

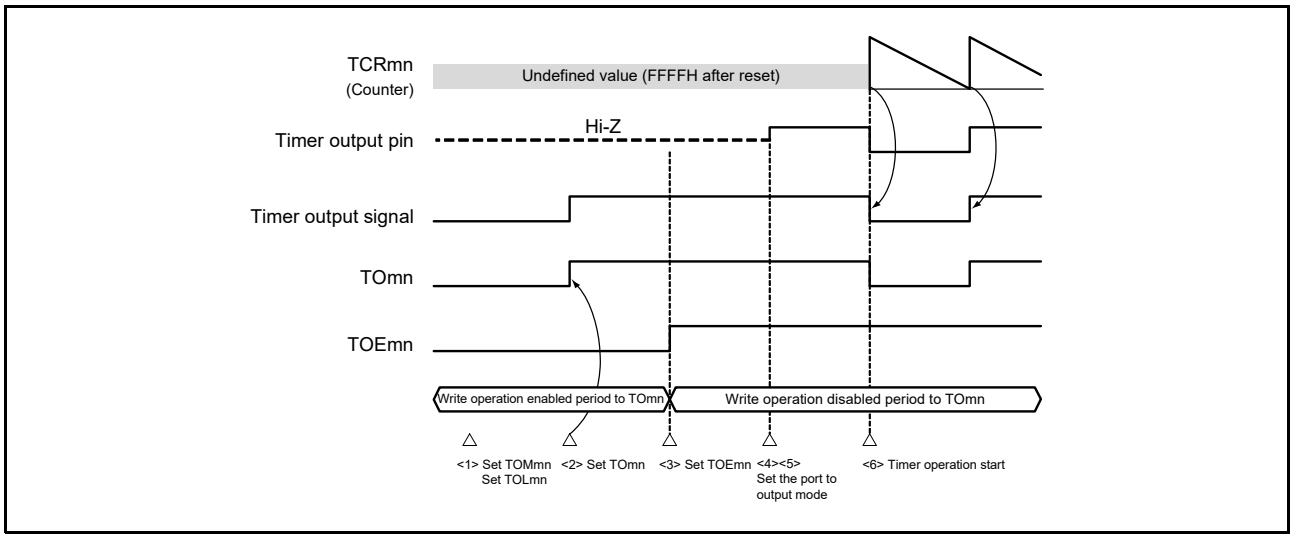
<5> The TOm register can always be read, and the TOmn pin output level can be checked.

Remark 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

7.6.2 TOmn pin output setting

The following figure shows the procedure and state transitions from the initial settings of a TOmn output pin to the start of timer operation.

Figure 7 - 36 State Transitions from the Settings for Timer Output to the Start of Timer Operation



<1> The operation mode of timer output is set.

- TOMmn bit (0: Master channel output mode, 1: Slave channel output mode)
- TOLmn bit (0: Positive logic output, 1: Negative logic output)

<2> The timer output signal is set to the initial state 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 is set to digital I/O by the port mode control A register (PMCAxx), port mode control T register (PMCTxx), port mode control E register (PMCExx), and port function output enable register (PFOEx) (see **7.3.17 Registers for controlling the port functions multiplexed with the inputs and outputs of timers**).

<5> The port I/O setting is set to output (see **7.3.17 Registers for controlling the port functions multiplexed with the inputs and outputs of timers**).

<6> The timer operation is enabled (TSmn = 1).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

7.6.3 Cautions on channel output operation

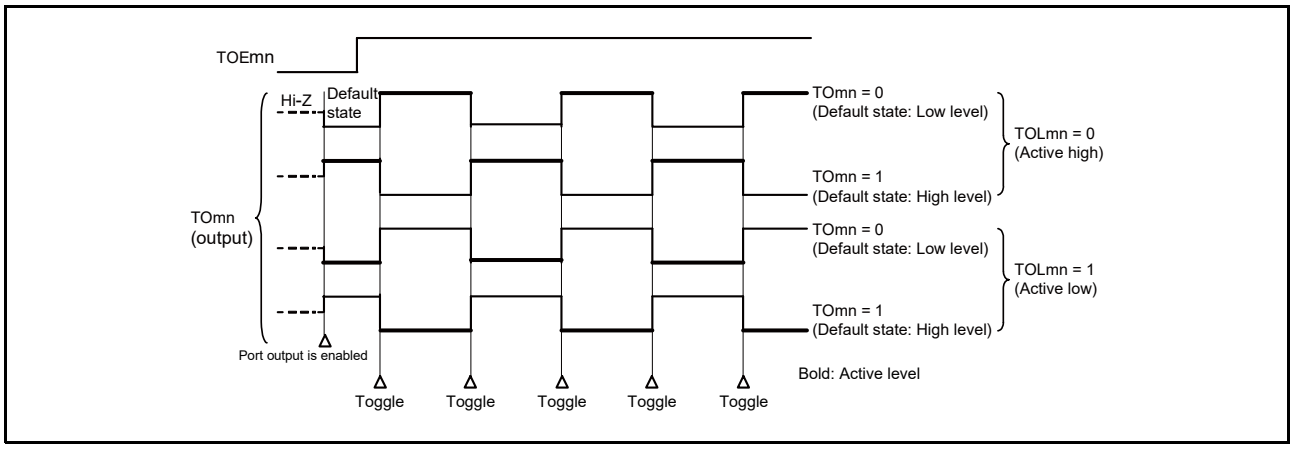
1. Changing values set in the registers TOm, TOEm, and TOLm during timer operation
   Since the timer operations (operations of timer counter register mn (TCRmn) and timer data register mn (TDRmn)) are independent of the TOmn output circuit and changing the values set in timer output register m (TOm), timer output enable register m (TOEm), and timer output level register m (TOLm) 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 **7.7 Timer Input (TImn) Control** and **7.8 Independent Channel Operation Function of Timer Array Unit**. When the values set to the TOEm, 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.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

2. Default level of TOmn pin and output level after timer operation start
   The change in the output level of the TOmn pin when timer output register m (TOm) is written while timer output is disabled (TOEmn = 0), the initial level is changed, and then timer output is enabled (TOEmn = 1) before port output is enabled, is shown below.

   a) When operation starts with master channel output mode (TOMmn = 0) setting
      The setting of timer output level register m (TOLm) is invalid when master channel output mode (TOMmn = 0).
      When the timer operation starts after setting the default level, the toggle signal is generated and the output level of the TOmn pin is inverted.

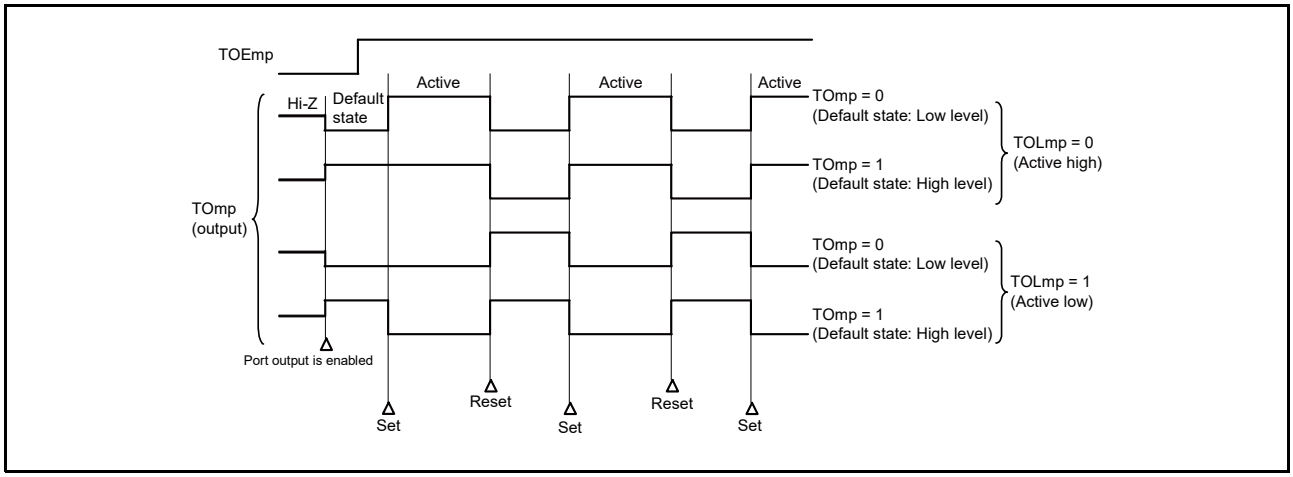
Figure 7 - 37 TOmn Pin Output States with Toggled Output (TOMmn = 0)



Remark 1. Toggle: Toggle signal to invert the output on the TOmn pin
Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

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 7 - 38 TOmp Pin Output States with PWM Output (TOMmp = 1)



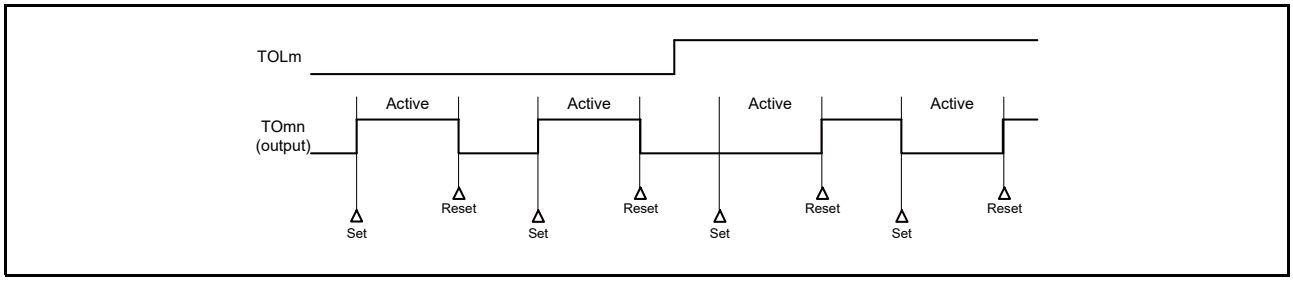
Remark 1. Set: The output signal of the TOmp pin changes from inactive level to active level.
Reset: The output signal of the TOmp pin changes from active level to inactive level.

Remark 2. m: Unit number (m = 0, 1), p: Channel number (p = 1 to 7)

3. Operation of TOmn pin in slave channel output mode (TOMmn = 1)

a) When the relevant bit of timer output level register m (TOLm) is 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 7 - 39 Operation When the Relevant Bit of the TOLm Register Is Changed during Timer Operation



Remark 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.

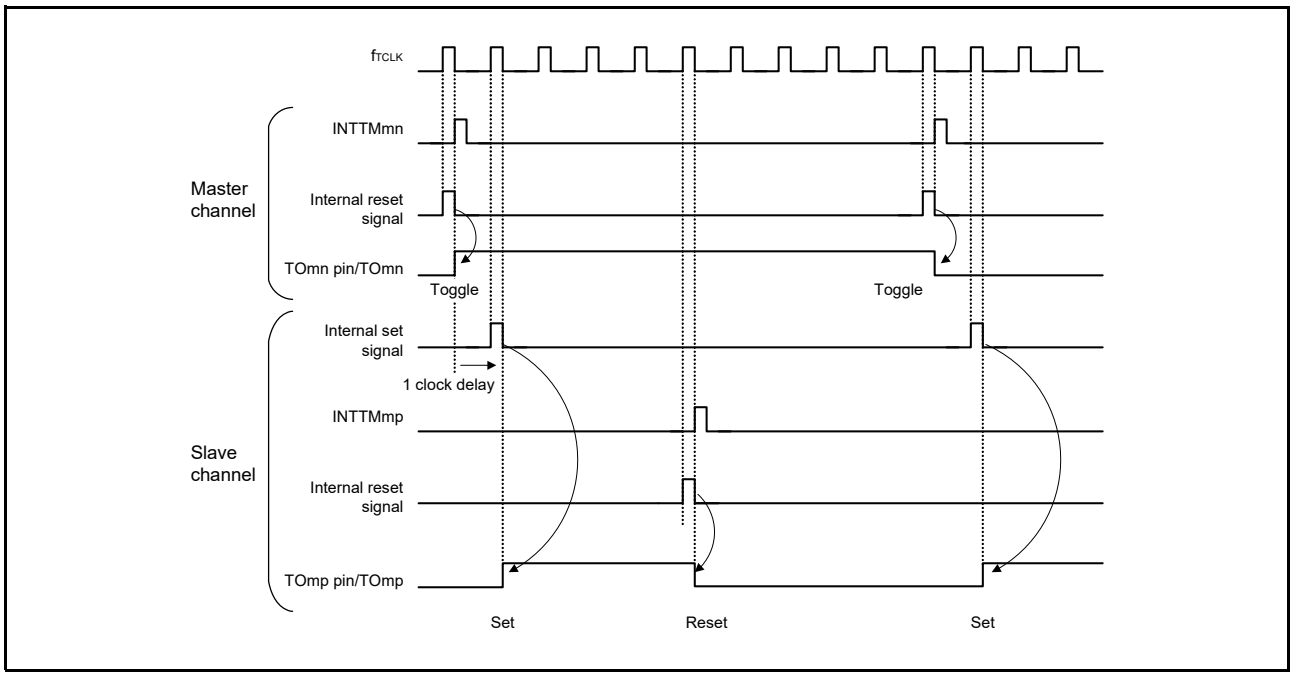
Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

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 counter 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 7 - 40** shows the states of operation following set and reset signals when the master and slave channels are set as follows.

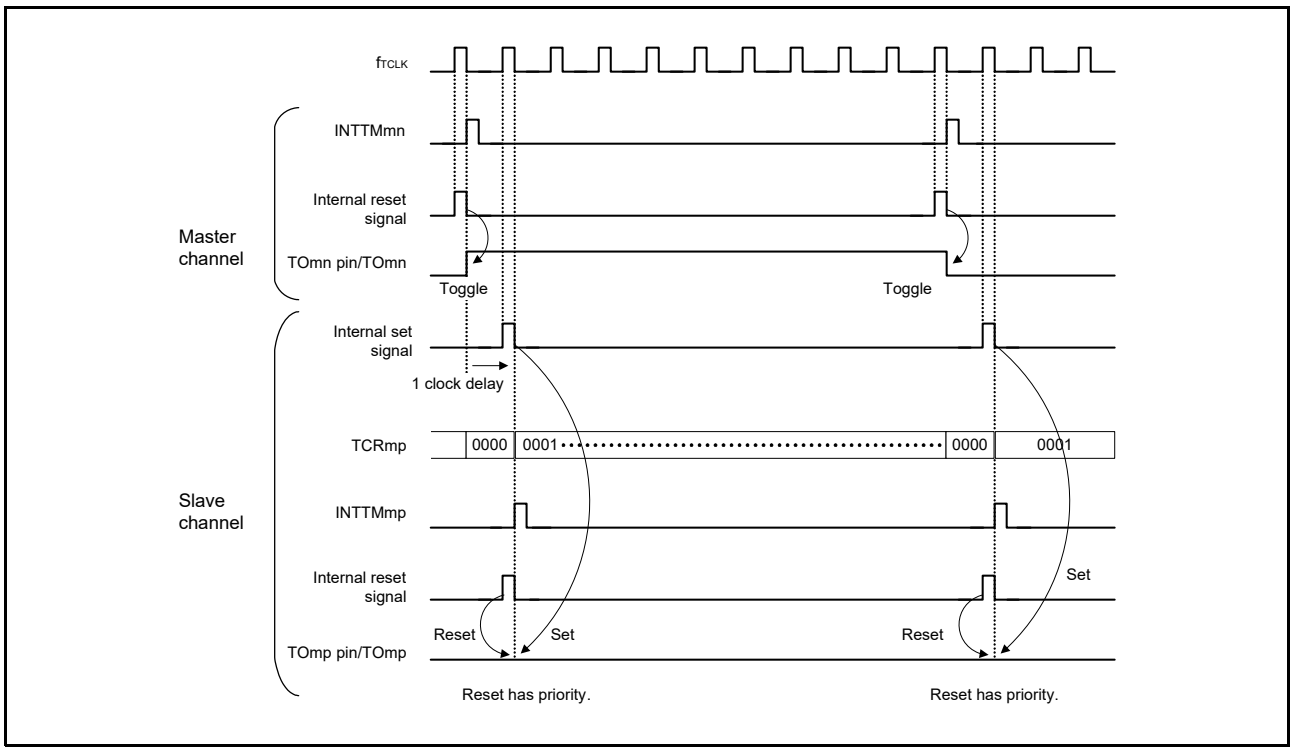
- Master channel: TOEmn = 1, TOMmn = 0, TOLmn = 0
- Slave channel: TOEmp = 1, TOMmp = 1, TOLmp = 0

Figure 7 - 40 States of Operation following Set and Reset Signals

1. Basic timing during operation



2. Timing during operation with the duty cycle set to 0%



(Remarks are listed on the next page.)

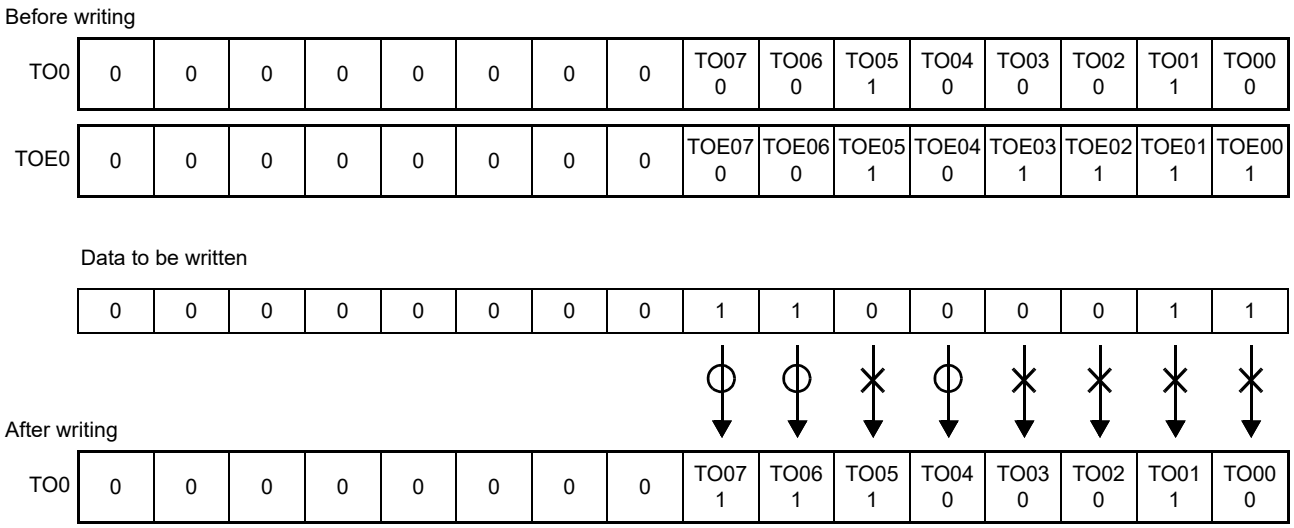
Remark 1. Internal reset signal: TOmn pin reset/toggle signal
Internal set signal: TOmn pin set signal

Remark 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

7.6.4 Collective manipulation of TOmn bit

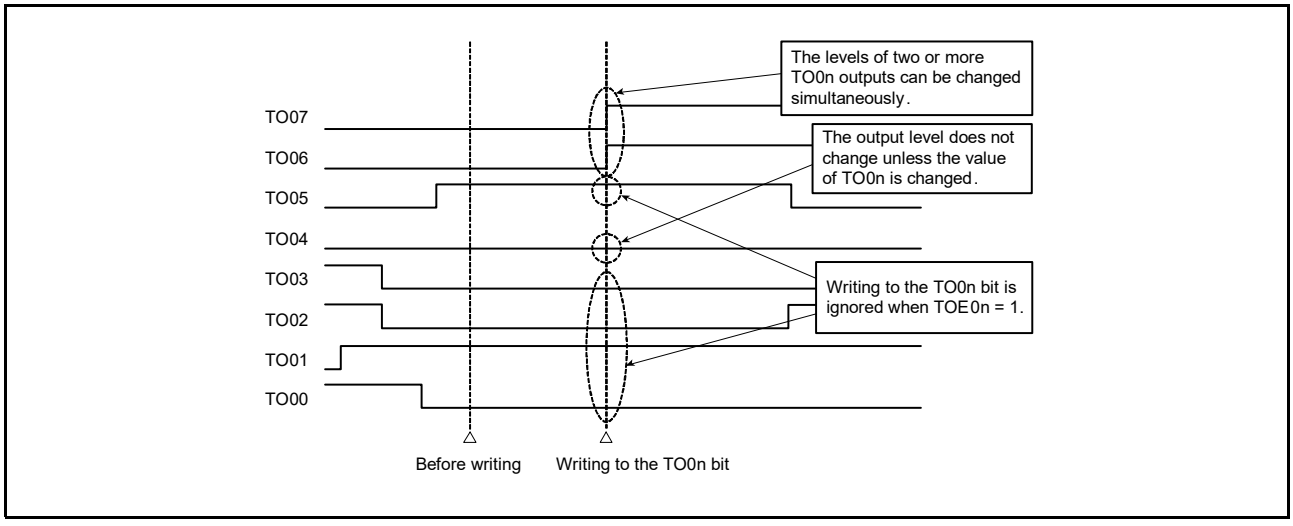
In timer output register m (TOm), the setting bits for all the channels are located in one register in the same way as timer channel start register m (TSm). Therefore, the TOmn bit of all the channels can be manipulated collectively. Only the desired bits can also be manipulated by enabling writing only to the TOmn bits (TOEmn = 0) that correspond to the relevant bits of the channel used to perform output (TOmn).

Figure 7 - 41 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 7 - 42 TO0n Pin States by Collective Manipulation of TO0n Bits



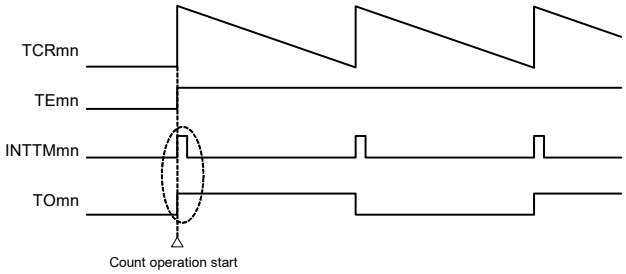
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

7.6.5 Timer interrupts and TOmn outputs when counting is started

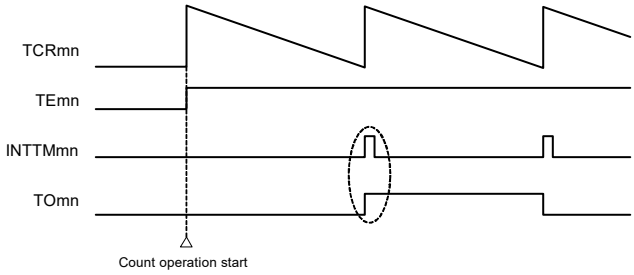
In the interval timer mode or capture mode, the MDmn0 bit in timer mode register mn (TMRmn) sets whether or not to generate a timer interrupt at count start. When MDmn0 is set to 1, the count operation start timing can be known by the timer interrupt (INTTMmn) generation. In the other modes, neither timer interrupt at count operation start nor TOmn output is controlled. **Figure 7 - 43** shows operation examples when the interval timer mode (TOEmn = 1, TOMmn = 0) is set.

Figure 7 - 43 Examples of the Operation of Timer Interrupts and TOmn Outputs When Counting Is Started

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.

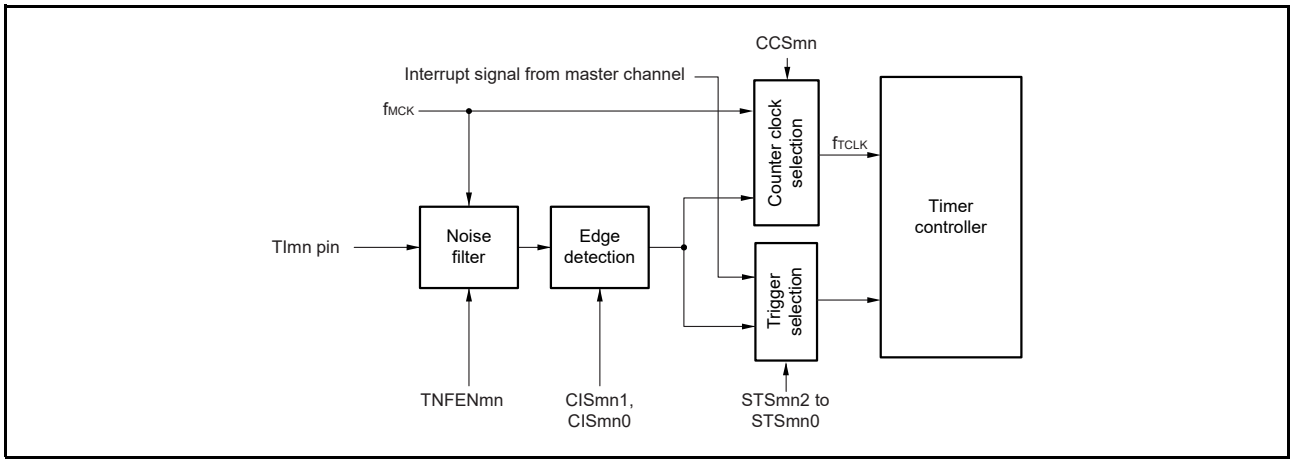
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

7.7 Timer Input (TImn) Control

7.7.1 TImn input circuit configuration

A signal is input from a timer input pin, goes through a noise filter and an edge detector, and is sent to a timer controller. Enable the noise filter for the pin in need of noise removal. The following shows the configuration of the input circuit.

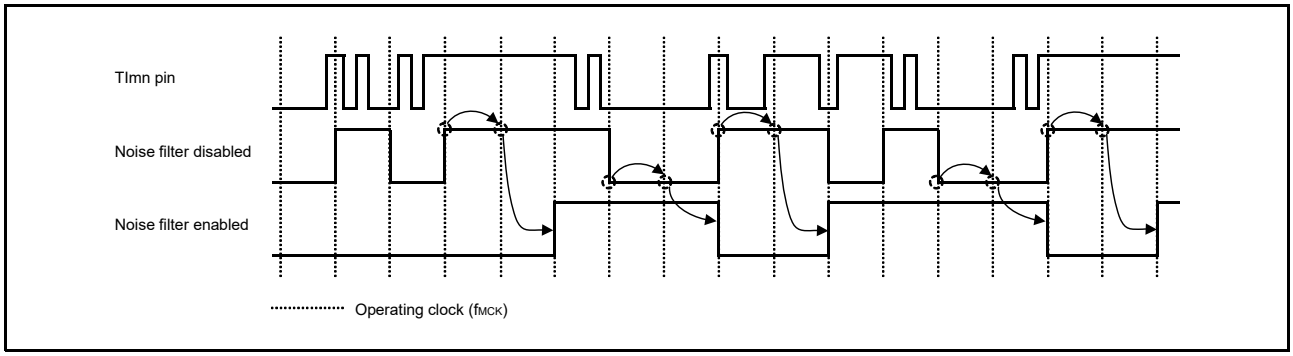
Figure 7 - 44 Input Circuit Configuration



7.7.2 Noise filter

When the noise filter is disabled, the input signal is only synchronized with the operating clock (fMCK) for channel n. When the noise filter is enabled, after synchronization with the operating clock (fMCK) for channel n, whether the signal keeps the same value for two clock cycles is detected. The following shows differences in waveforms output from the noise filter between when the noise filter is enabled and disabled.

Figure 7 - 45 Sampling Waveforms through TImn Input Pin with Noise Filter Enabled and Disabled



7.7.3 Cautions on channel input operation

When a timer input pin is set as unused, the operating clock is not supplied to the noise filter. Therefore, after settings are made to use the timer input pin, the following wait time is necessary before a trigger is specified to enable operation of the channel corresponding to the timer input pin.

1. Noise filter is disabled
   When the CCSmn, STSmn1, and STSmn0 bits in the timer mode register mn (TMRmn) are all 0 and then one of them is set to 1, wait for at least two cycles of the operating clock (f_{MCK}), and then set the operation enable trigger bit in the timer channel start register (TSm).

2. Noise filter is enabled
   When the CCSmn, STSmn1, and STSmn0 bits in the timer mode register mn (TMRmn) are all 0 and then one of them is set to 1, wait for at least four cycles of the operating clock (f_{MCK}), and then set the operation enable trigger bit in the timer channel start register (TSm).

7.8 Independent Channel Operation Function of Timer Array Unit

7.8.1 Operation as an interval timer or for 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 counter clock × (Set value of TDRmn + 1)

2. Operation for 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.

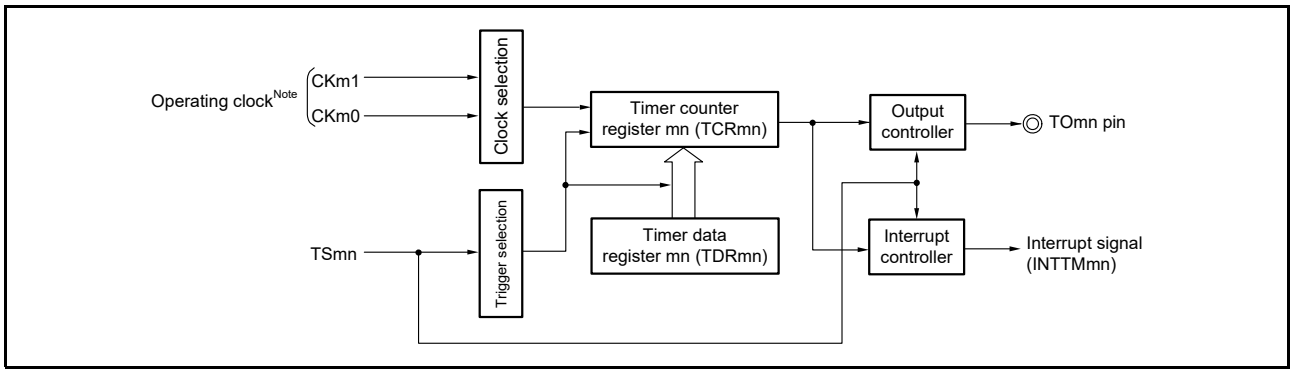
Period of square wave output from TOmn = Period of counter clock × (Set value of TDRmn + 1) × 2

Frequency of square wave output from TOmn = Frequency of counter clock/{(Set value of TDRmn + 1) × 2}

   Timer counter 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 counter clock after any of the operation enable (start) trigger bits (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 the output on TOmn is not toggled. If the MDmn0 bit of the TMRmn register is 1, INTTMmn is output and the output on TOmn is toggled. After that, the TCRmn register count down in synchronization with the counter clock. When TCRmn = 0000H, INTTMmn is output and the output on TOmn is toggled at the next counter 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.

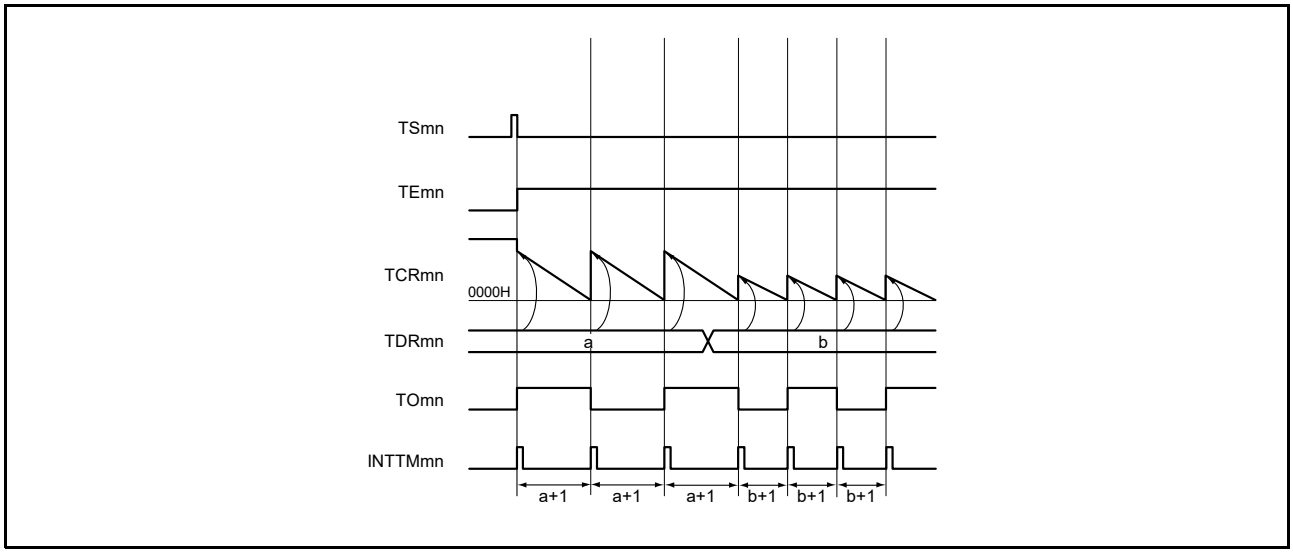
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

Figure 7 - 46 Block Diagram for Operation as an Interval Timer or for Square Wave Output



Note For channels 1 and 3, the clock can be selected as CKm0, CKm1, CKm2, or CKm3.

Figure 7 - 47 Example of Basic Timing during Operation as an Interval Timer or for Square Wave Output (MDmn0 = 1)

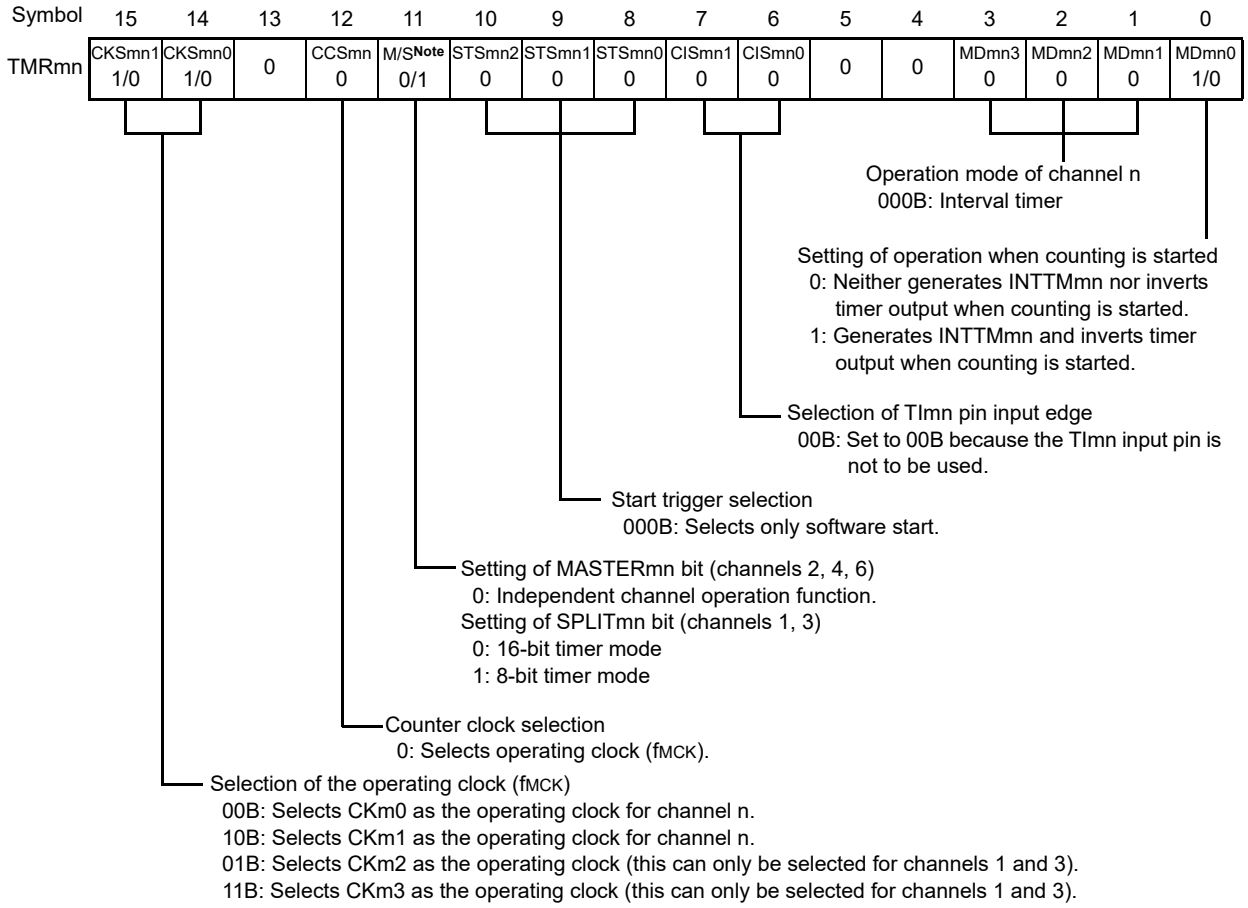


Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

Remark 2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TCRmn: Timer counter register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)
TOmn: TOmn pin output signal

Figure 7 - 48 Example of Register Settings for Operation as an Interval Timer or for 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: Set this bit 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 TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

Figure 7 - 49 Procedure for Operations When the Interval Timer or Square Wave Output Function Is to Be Used

	Software Operation	Hardware State
TAU default setting		Power-off state (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 state. 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 Sets the TOMmn bit of timer output mode register m (TOMm) to 0 (master channel output mode). Sets 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.
	Sets 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.	TEmn (TEHm1, TEHm3) = 1, and count operation starts. Value of the TDRmn register is loaded to timer counter register mn (TCRmn). 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.	TEmn (TEHm1, TEHm3) = 0, and count operation stops. The TCRmn register holds count value and stops. The TOmn output is not initialized and retains its current state.
	The TOEmn bit is set to 0 and value is set to the TOmn bit. →	The TOmn pin outputs the TOmn bit set level.
TAU stop	To hold the TOmn pin output level Sets 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 set to 0. → Set the TAUmRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit. →	This stops supply of the input clock to timer array unit m. 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.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

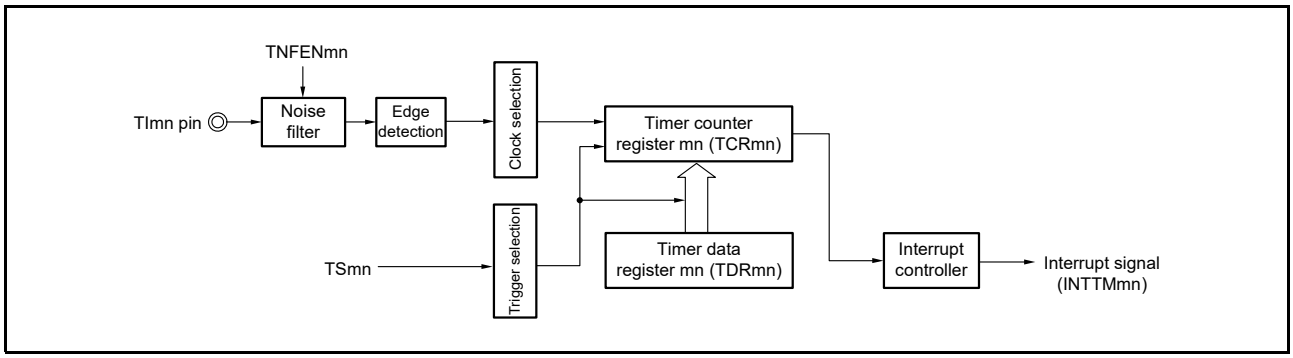
7.8.2 Operation as an external event counter

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

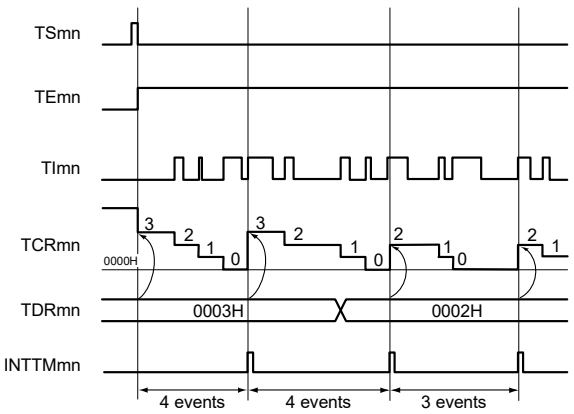
Timer counter register mn (TCRmn) operates as a down counter in the event counter mode. The TCRmn register loads the value of timer data register mn (TDRmn) by setting any operation enable (start) trigger bit (TSmn) of timer channel start register m (TSm) to 1. The TCRmn register counts down each time 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. Instead of using the TImn pin input, a channel specified for the external event counter function can also use the timer input selected in the TIS0 or TIS1 register as its input source to drive counting.

Figure 7 - 50 Block Diagram for Operation as an External Event Counter



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

Figure 7 - 51 Example of Basic Timing during Operation as an External Event Counter

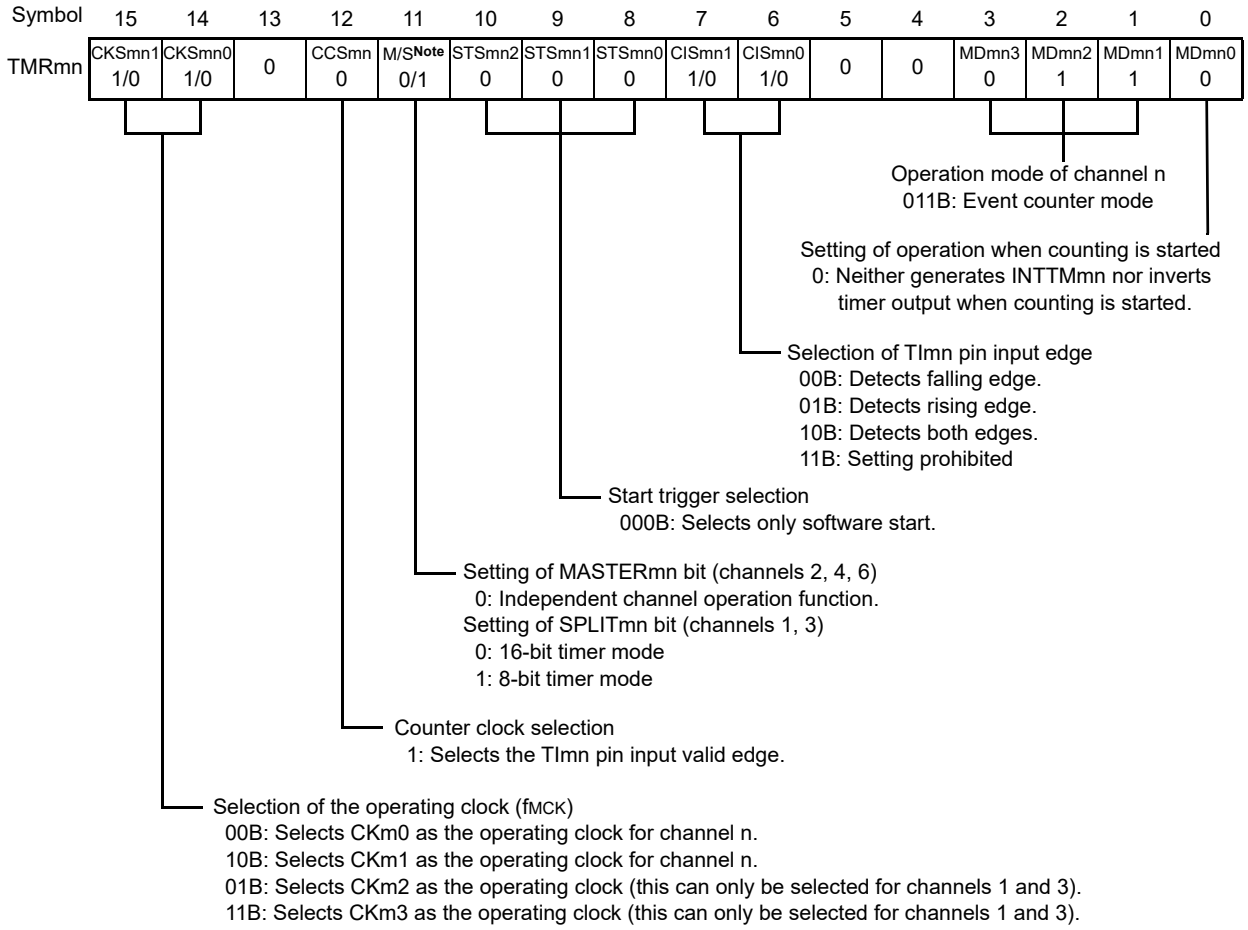


Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

Remark 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 counter register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)

Figure 7 - 52 Example of Register Settings in External Event Counter Mode

a) Timer mode register mn (TMRmn)



b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops the TOmn output operation by counting operation.

d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Set this bit 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 TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

Figure 7 - 53 Procedure for Operations When the External Event Counter Function Is to Be Used

	Software Operation	Hardware State
TAU default setting		Power-off state (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 state. 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). Sets 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.	TEmn = 1, and count operation starts. Value of the TDRmn register is loaded to timer counter 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.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops.
TAU stop	The TAUmEN bit of the PER0 register is set to 0. → Set the TAUmRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit. →	This stops supply of the input clock to timer array unit m. All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

7.8.3 Operation as a frequency divider (only for channel 0 of unit 0)

The timer array unit can be used as a frequency divider that divides a clock input to the TI00 pin and outputs the result from the TO00 pin. The divided clock frequency output from TO00 can be calculated by the following expression.

- When rising edge/falling edge is selected:
  Divided clock frequency = Input clock frequency/{(Set value of TDR00 + 1) × 2}
- When both edges are selected:
  Divided clock frequency ≈ Input clock frequency/(Set value of TDR00 + 1)

Timer counter register 00 (TCR00) operates as a down counter in the interval timer mode. After the operation enable (start) trigger bit (TS00) of timer channel start register 0 (TS0) is set to 1, the TCR00 register loads the value of timer data register 00 (TDR00) when the TI00 valid edge is detected. If the MD000 bit of timer mode register 00 (TMR00) is 0 at this time, INTTM00 is not output and the output on TO00 is not toggled. If the MD000 bit of timer mode register 00 (TMR00) is 1, INTTM00 is output and the output on TO00 is toggled. After that, the TCR00 register counts down at the valid edge of the TI00 pin. When TCR00 = 0000H, it toggles the output on TO00. At the same time, the TCR00 register loads the value of the TDR00 register again, and continues counting. If detection of both the edges of the TI00 pin is selected, the duty factor error of the input clock affects the divided clock period of the TO00 output. The period of the TO00 output clock includes a sampling error of one period of the operating clock.

Clock period of TO00 output = Ideal TO00 output clock period ± Operating clock period (error)

The TDR00 register can be rewritten at any time. The new value of the TDR00 register becomes valid during the next count period.

Figure 7 - 54 Block Diagram for Operation as a Frequency Divider

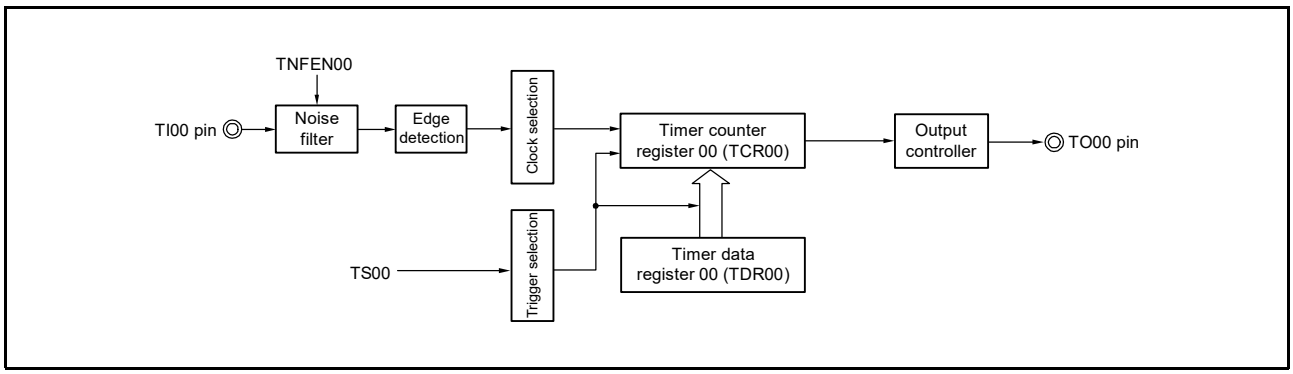
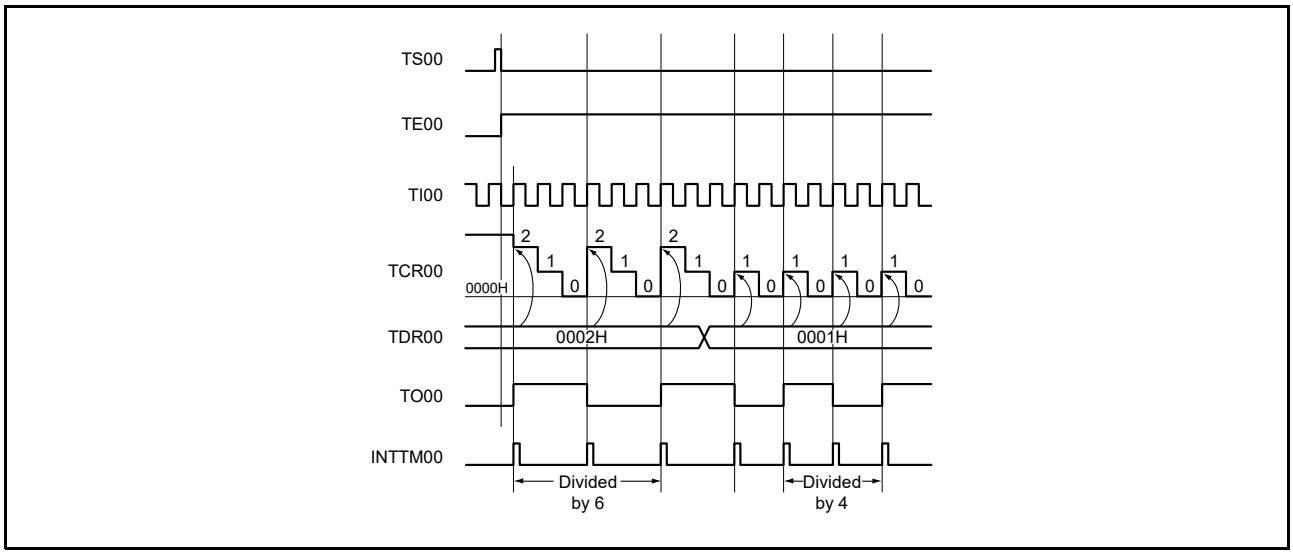


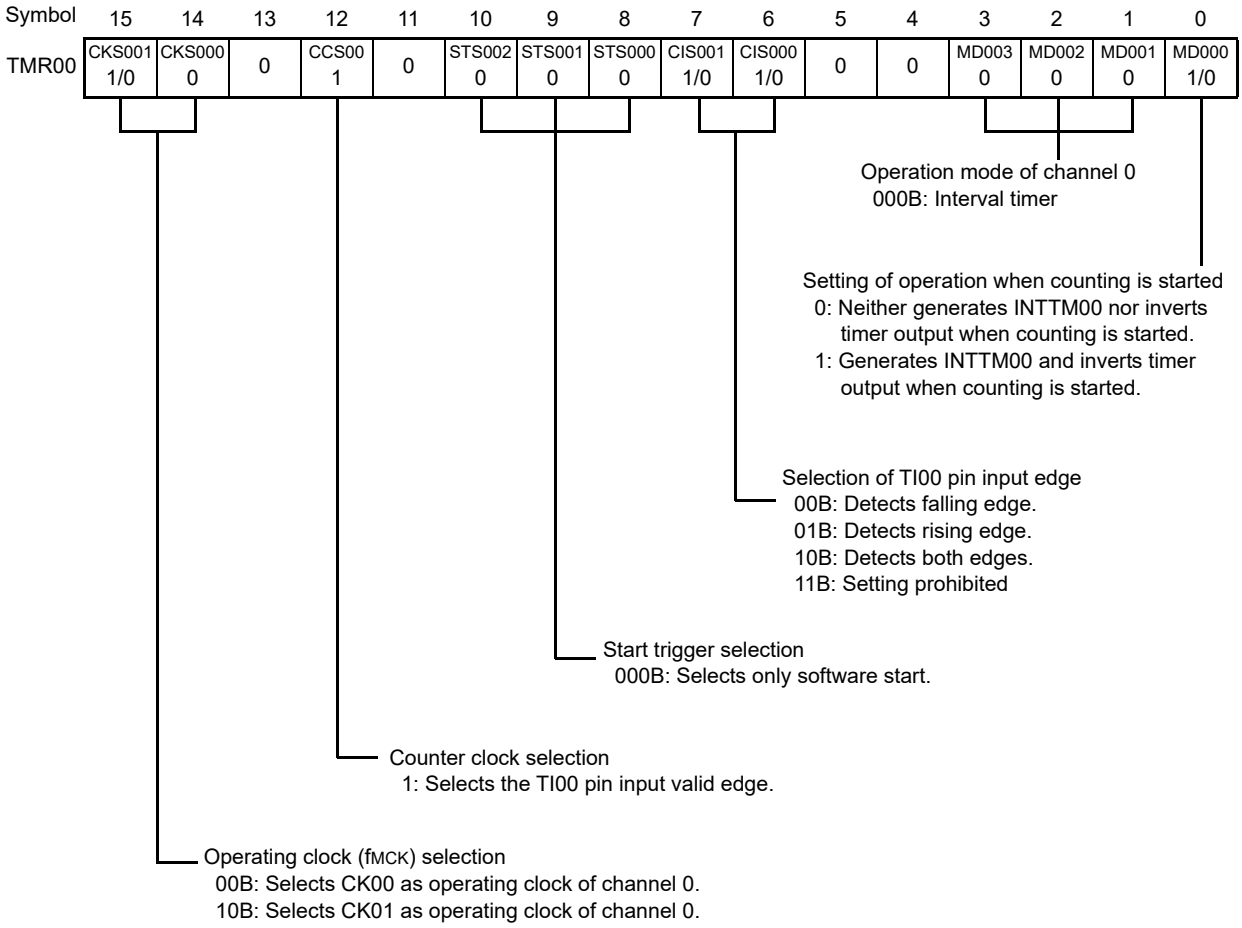
Figure 7 - 55 Example of Basic Timing during Operation as a Frequency Divider (MD000 = 1)



Remark TS00: Bit 0 of timer channel start register 0 (TS0)
TE00: Bit 0 of timer channel enable status register 0 (TE0)
TI00: TI00 pin input signal
TCR00: Timer counter register 00 (TCR00)
TDR00: Timer data register 00 (TDR00)
TO00: TO00 pin output signal

Figure 7 - 56 Example of Register Settings for Operation as a Frequency Divider

a) Timer mode register 00 (TMR00)



b) Timer output register 0 (TO0)

	Bit 0	
TO0	TO00 1/0	0: Outputs 0 from TO00. 1: Outputs 1 from TO00.

c) Timer output enable register 0 (TOE0)

	Bit 0	
TOE0	TOE00 1/0	0: Stops the TO00 output operation by counting operation. 1: Enables the TO00 output operation by counting operation.

d) Timer output level register 0 (TOL0)

	Bit 0	
TOL0	TOL00 0	0: Set this bit to 0 when master channel output mode (TOM00 = 0).

e) Timer output mode register 0 (TOM0)

	Bit 0	
TOM0	TOM00 0	0: Sets master channel output mode.

Figure 7 - 57 Procedure for Operations When the Frequency Divider Function Is to Be Used (1/2)

	Software Operation	Hardware State
TAU default setting		Power-off state (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Power-on state. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 0 (TPS0). Determines clock frequencies of CK00 to CK03.	
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 0n (TMR0n) (determines operation mode of channel and selects the detection edge). Sets interval (period) value to timer data register 00 (TDR00).	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets the TOM00 bit of timer output mode register 0 (TOM0) to 0 (master channel output mode). Sets the TOL00 bit to 0. Sets the TO00 bit and determines default level of the TO00 output. →	The TO00 pin goes into Hi-Z output state. The TO00 default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOE00 bit to 1 and enables operation of TO00. →	TO00 does not change because channel stops operating.
	Sets the port register and port mode register to 0. →	The TO00 pin outputs the TO00 set level.

Figure 7 - 57 Procedure for Operations When the Frequency Divider Function Is to Be Used (2/2)

		Software Operation	Hardware State
Operation is resumed	Operation start	Sets the TOE00 bit to 1 (only when operation is resumed). Sets the TS00 bit to 1. The TS00 bit automatically returns to 0.	TE00 = 1, and count operation starts. Value of the TDR00 register is loaded to timer counter register 00 (TCR00). INTTM00 is generated and TO00 performs toggle operation if the MD000 bit of the TMR00 register is 1.
	During operation	Set value of the TDR00 register can be changed. The TCR00 register can always be read. The TSR00 register is not used. Set values of the TO0 and TOE0 registers can be changed. Set values of the TMR00 register, TOM00, and TOL00 bits cannot be changed.	Counter (TCR00) counts down. When count value reaches 0000H, the value of the TDR00 register is loaded to the TCR00 register again, and the count operation is continued. By detecting TCR00 = 0000H, INTTM00 is generated and TO00 performs toggle operation. After that, the above operation is repeated.
	Operation stop	The TT00 bit is set to 1. The TT00 bit automatically returns to 0.	TE00 = 0, and count operation stops. The TCR00 register holds count value and stops. The TO00 output is not initialized and retains its current state.
		The TOE00 bit is set to 0 and value is set to the TO00 bit.	The TO00 pin outputs the TO00 set level.
	TAU stop	To hold the TO00 pin output level Sets the TO00 bit to 0 after the value to be held is set to the port register. When holding the TO00 pin output level is not necessary Setting not required.	The TO00 pin output level is held by port function.
		The TAU0EN bit of the PER0 register is set to 0. Set the TAUmRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit.	This stops supply of the input clock to timer array unit m. 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).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

7.8.4 Operation for input pulse interval measurement

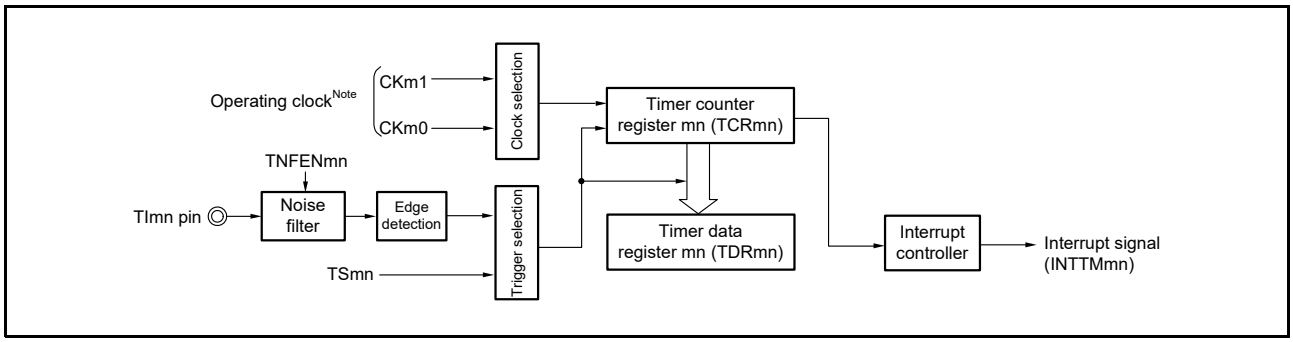
The count value can be captured at the TImn valid edge and the interval of the pulse input to TImn can be measured. In addition, the count value can be captured by using software operation (TSmn = 1) as a capture trigger while the TEmn bit is set to 1. The pulse interval can be calculated by the following expression.

TImn input pulse interval = Period of counter clock × ((10000H × TSRmn.OVF) + (Captured 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 cycle of the operating clock occurs.

Timer counter register mn (TCRmn) operates as an up counter in the capture mode. When the operation enable (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 counter 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 state 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. Instead of using the TImn pin input, an input pulse interval can also be measured by using the timer input selected in the TIS0 or TIS1 register or the software operation (TSmn = 1) as a start trigger and a capture trigger.

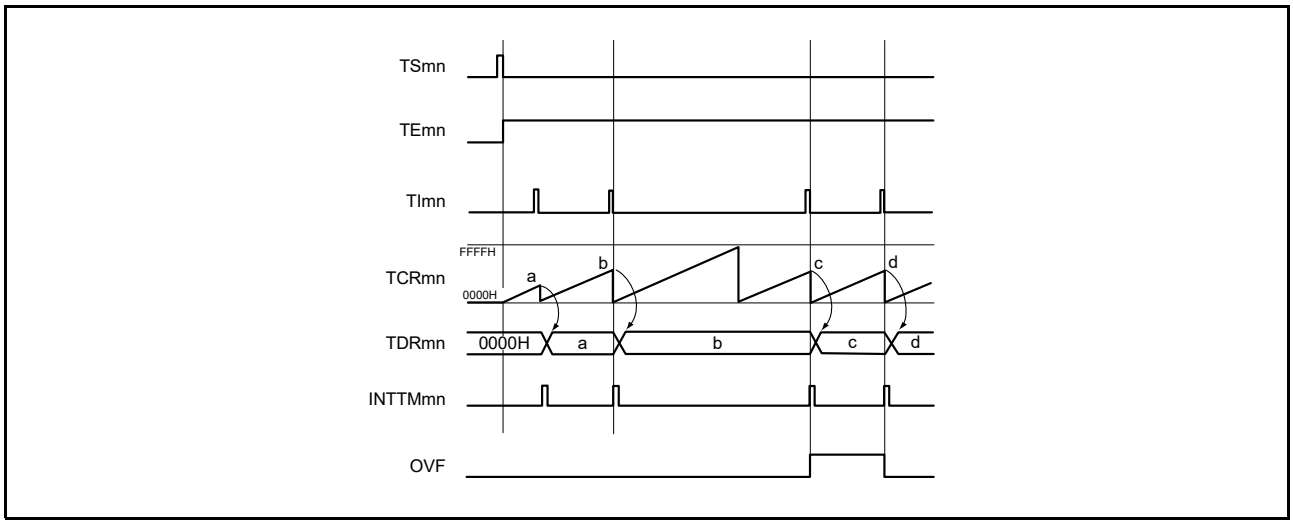
Figure 7 - 58 Block Diagram for Operation for Input Pulse Interval Measurement



Note For channels 1 and 3, the clock can be selected as CKm0, CKm1, CKm2, or CKm3.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

Figure 7 - 59 Example of Basic Timing during Operation for Input Pulse Interval Measurement (MDmn0 = 0)

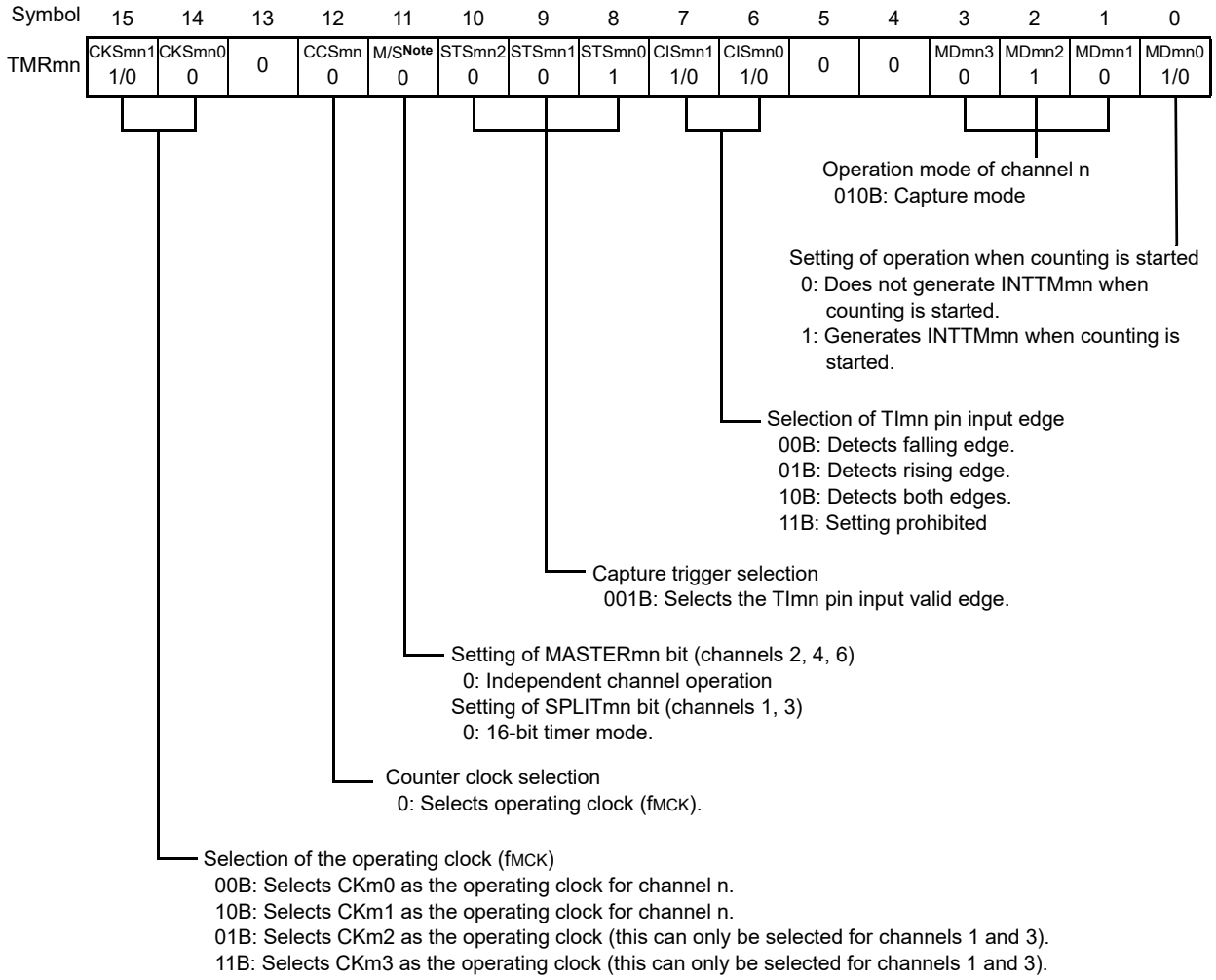


Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

Remark 2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TImn: TImn pi n input signal
TCRmn: Timer counter register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)
OVF: Bit 0 of timer status register mn (TSRmn)

Figure 7 - 60 Example of Register Settings for Operation for Input Pulse Interval Measurement

a) Timer mode register mn (TMRmn)



b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops TOmn output operation by counting operation.

d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Set this bit 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 Remark are listed on the next page.)

Note TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

Figure 7 - 61 Procedure for Operations When the Input Pulse Interval Measurement Function Is to Be Used

	Software Operation	Hardware State
TAU default setting		Power-off state (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 state. 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.	TEmn = 1, and count operation starts. Timer counter register mn (TCRmn) is cleared to 0000H. 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 valid edge of the TImn pin input is detected or the TSmn bit is set to 1, 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.	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 set to 0. → Set the TAUmRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit. →	This stops supply of the input clock to timer array unit m. All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

7.8.5 Operation for input signal low- and high-level width measurement

Caution When using a channel to implement the LIN-bus, set the ISC1 bit of the input switch control register (ISC) to 1. In the following descriptions, read TImn as RxD2.

By starting counting at one edge of the TImn pin input and capturing the number of counts at another edge, the signal width (low and high levels) of TImn can be measured. The signal width of TImn can be calculated by the following expression.

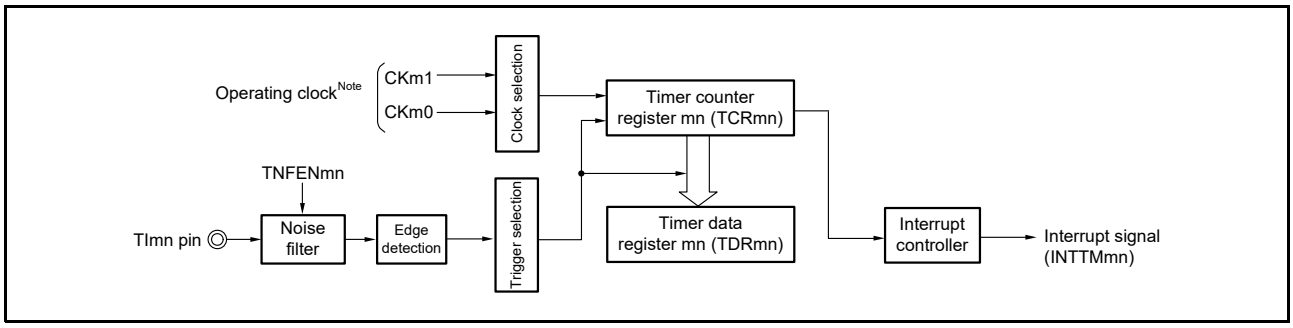
Signal width of TImn input = Period of counter clock × ((10000H × TSRmn.OVF) + (Captured value of TDRmn + 1))

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 cycle of the operating clock occurs.

Timer counter register mn (TCRmn) operates as an up counter in the capture & one-count mode. When the operation enable (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 state 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 counter 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 state 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 state 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 low-level width or high-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. Instead of the TImn pin input, the timer input selected in the TIS0 or TIS1 register can also be used as a start edge and a capture edge.

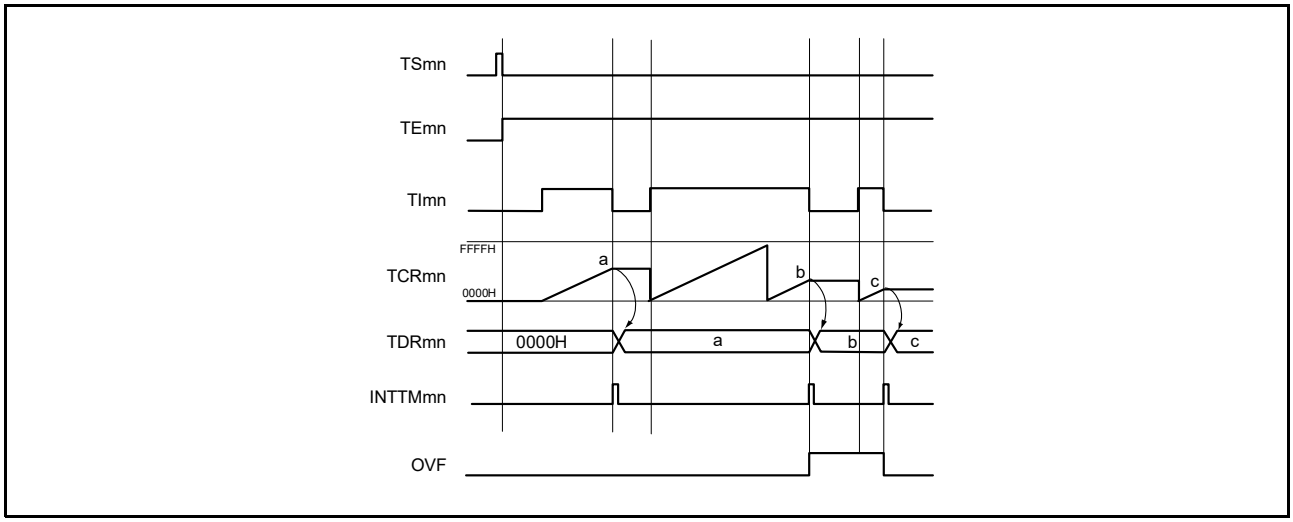
- CISmn1 and CISmn0 bits of the TMRmn register = 10B: Low-level width is measured.
- CISmn1 and CISmn0 bits of the TMRmn register = 11B: High-level width is measured.

Figure 7 - 62 Block Diagram for Operation for Input Signal Low- and High-level Width Measurement



Note For channels 1 and 3, the clock can be selected as CKm0, CKm1, CKm2, or CKm3.

Figure 7 - 63 Example of Basic Timing during Operation for Input Signal Low- and High-level Width Measurement

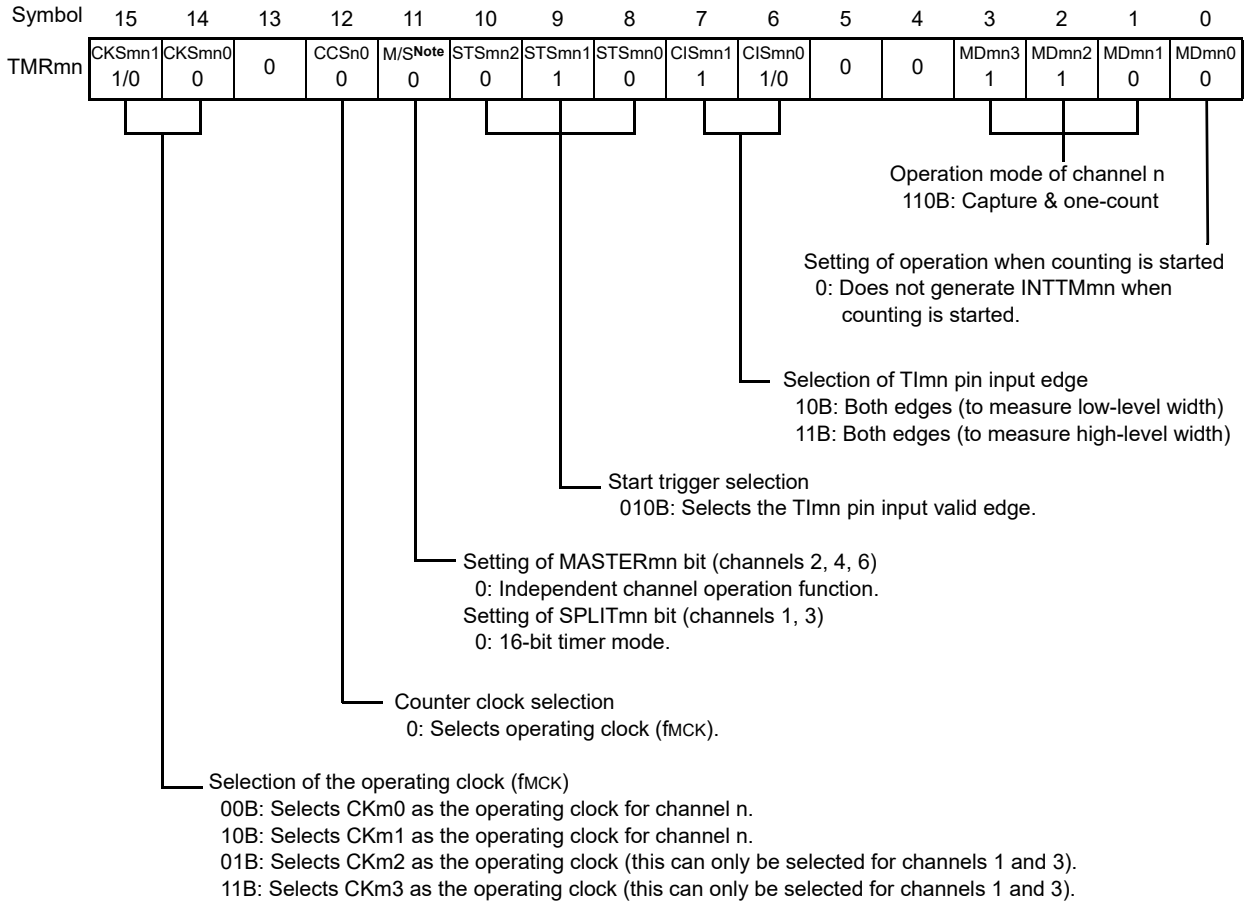


Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

Remark 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 counter register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)
OVF: Bit 0 of timer status register mn (TSRmn)

Figure 7 - 64 Example of Register Settings for Operation for Input Signal Low- and High-level Width Measurement

a) Timer mode register mn (TMRmn)



b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops the TOmn output operation by counting operation.

d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Set this bit 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 TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

Figure 7 - 65 Procedure for Operations When the Input Signal Low- and High-level Width Measurement Function Is to Be Used

	Software Operation	Hardware State
TAU default setting		Power-off state (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 state. 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 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.	TEmn = 1, and the TImn pin start edge detection wait state is set.
	Detects the TImn pin input count start valid edge. →	Clears timer counter register mn (TCRmn) to 0000H and starts counting up.
During operation	Set value of 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 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.	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 set to 0. → Set the TAUmRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit. →	This stops supply of the input clock to timer array unit m. All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

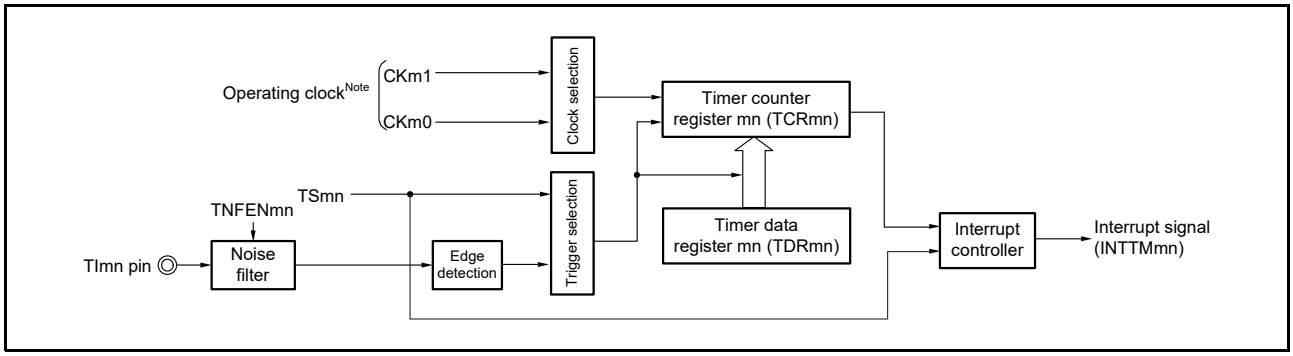
7.8.6 Operation as a delay counter

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 is also possible to start counting down and generate INTTMmn (timer interrupt) at any interval by setting TSmn to 1 by software while TEmn = 1. The interrupt generation period can be calculated by the following expression.

Generation period of INTTMmn (timer interrupt) = Period of counter clock × (Set value of TDRmn + 1)

Timer counter register mn (TCRmn) operates as a down counter in the one-count mode. When the operation enable (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 state is set. Timer counter 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 counter 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. Instead of using the TImn pin input, a channel specified for the delay counter function can also use the timer input selected in the TIS0 or TIS1 register or the software operation (TSmn = 1) as a start trigger for the function.

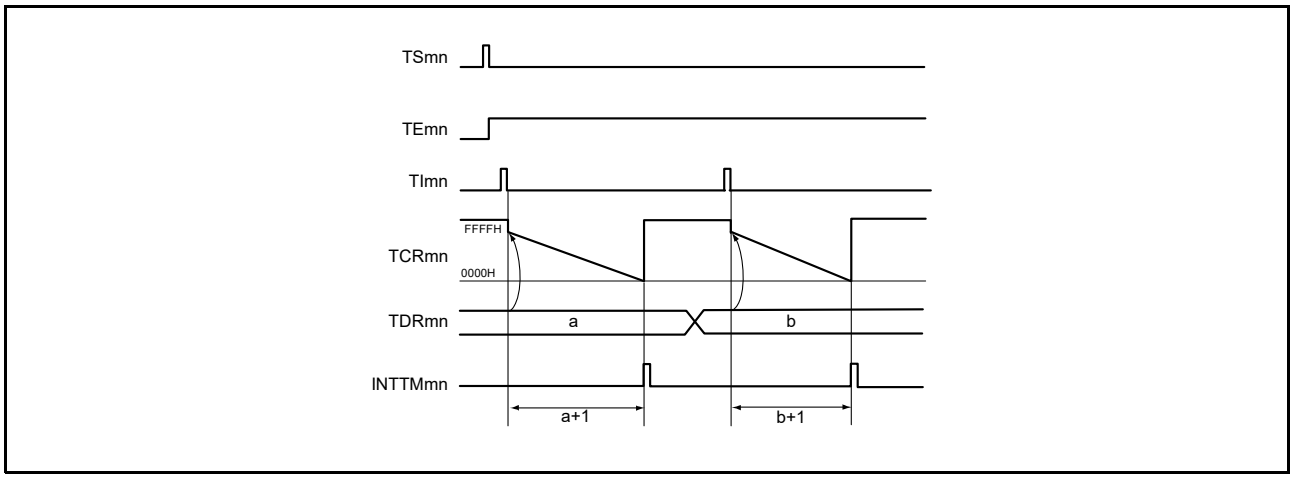
Figure 7 - 66 Block Diagram for Operation as a Delay Counter



Note For channels 1 and 3, the clock can be selected as CKm0, CKm1, CKm2, or CKm3.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

Figure 7 - 67 Example of Basic Timing during Operation as a Delay Counter

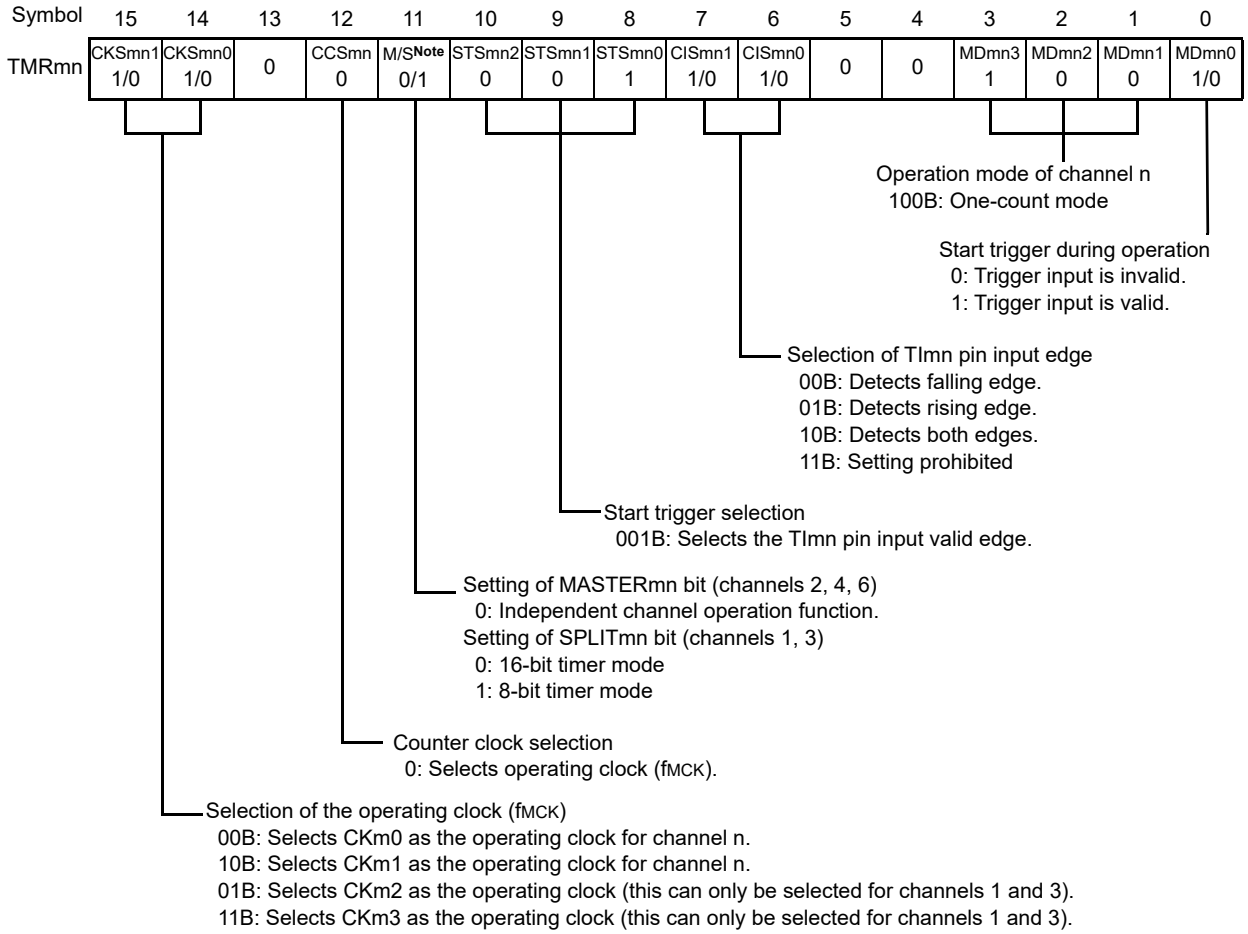


Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

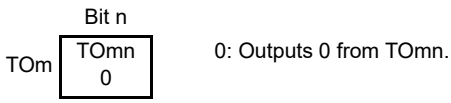
Remark 2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TImn: TImn pin input signal
TCRmn: Timer counter register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)

Figure 7 - 68 Example of Register Settings for Operation as a Delay Counter

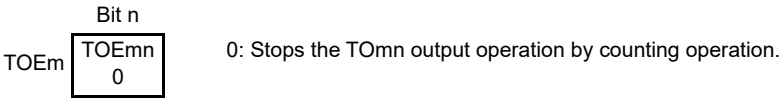
a) Timer mode register mn (TMRmn)



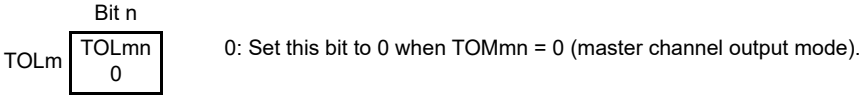
b) Timer output register m (TOm)



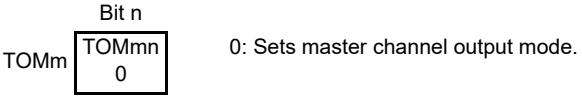
c) Timer output enable register m (TOEm)



d) Timer output level register m (TOLm)



e) Timer output mode register m (TOMm)



Note TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

Figure 7 - 69 Procedure for Operations When the Delay Counter Function Is to Be Used

	Software Operation	Hardware State
TAU default setting		Power-off state (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 state. 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). Sets 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.	TEmn = 1, and the start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit is set to 1) wait state is set.
	The counter starts counting down by the next start trigger detection. • Detects the TImn pin input valid edge. → • Sets the TSmn bit to 1 by the software.	Value of the TDRmn register is loaded to the timer counter 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 the count value of TCRmn reaches 0000H, the INTTMmn output is generated, and the count operation stops until the next start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit is set to 1).
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops.
TAU stop	The TAUmEN bit of the PER0 register is set to 0. → Set the TAUmRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit. →	This stops supply of the input clock to timer array unit m. All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7)

7.9 Simultaneous Channel Operation Function of Timer Array Unit

7.9.1 Operation for the 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.

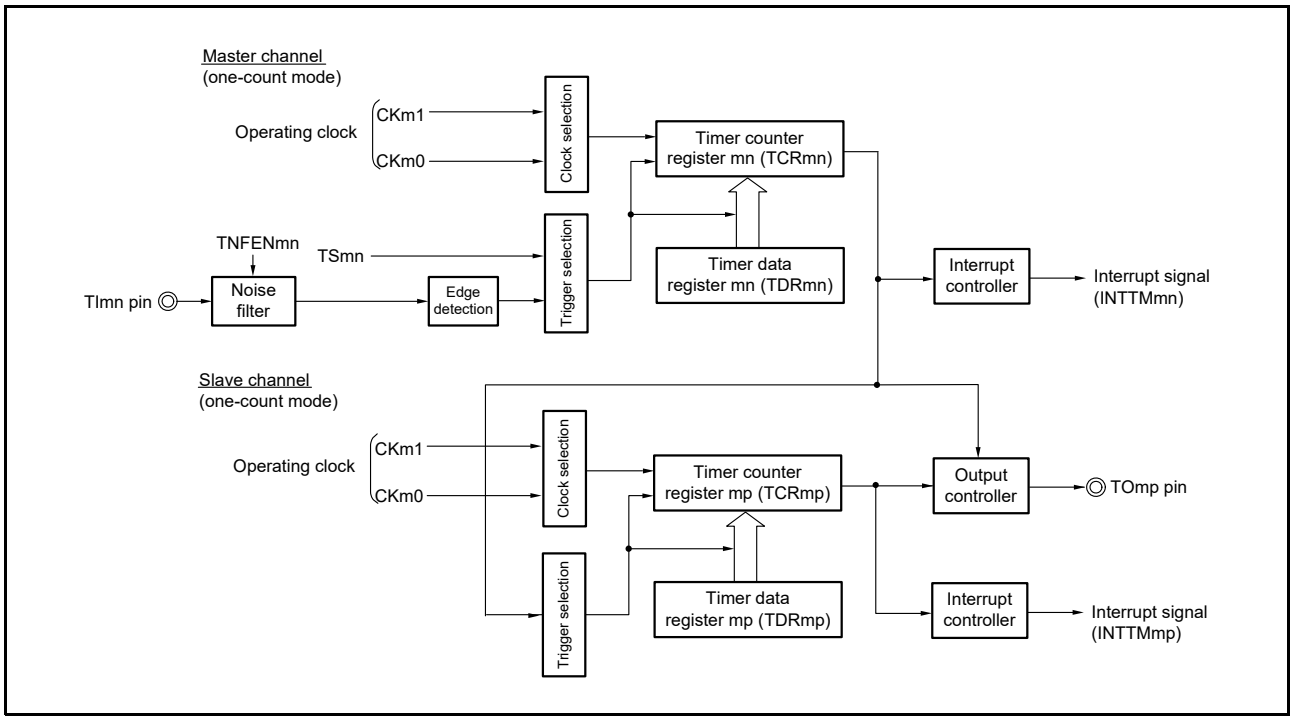
Delay time = {Set value of TDRmn (master) + 2} × Counter clock period Pulse width = {Set value of TDRmp (slave)} × Counter clock period

The master channel operates in the one-count mode and counts the delays. Timer counter 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 counter 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 counter 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 timer input selected in the TIS1 register or 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.**

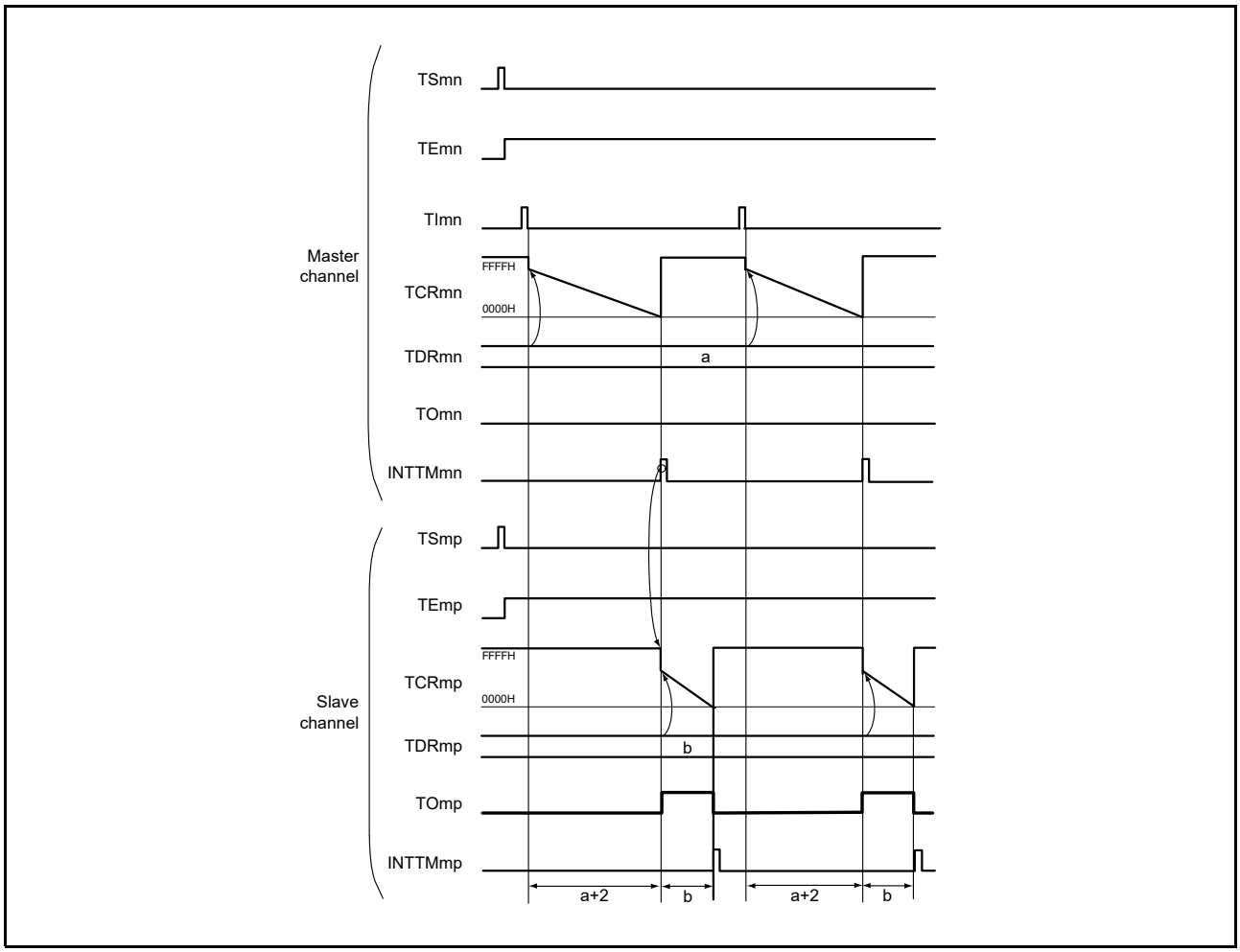
Remark m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 7 - 70 Block Diagram for Operation for the One-shot Pulse Output Function



Remark m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 7 - 71 Example of Basic Timing during Operation for the One-shot Pulse Output Function

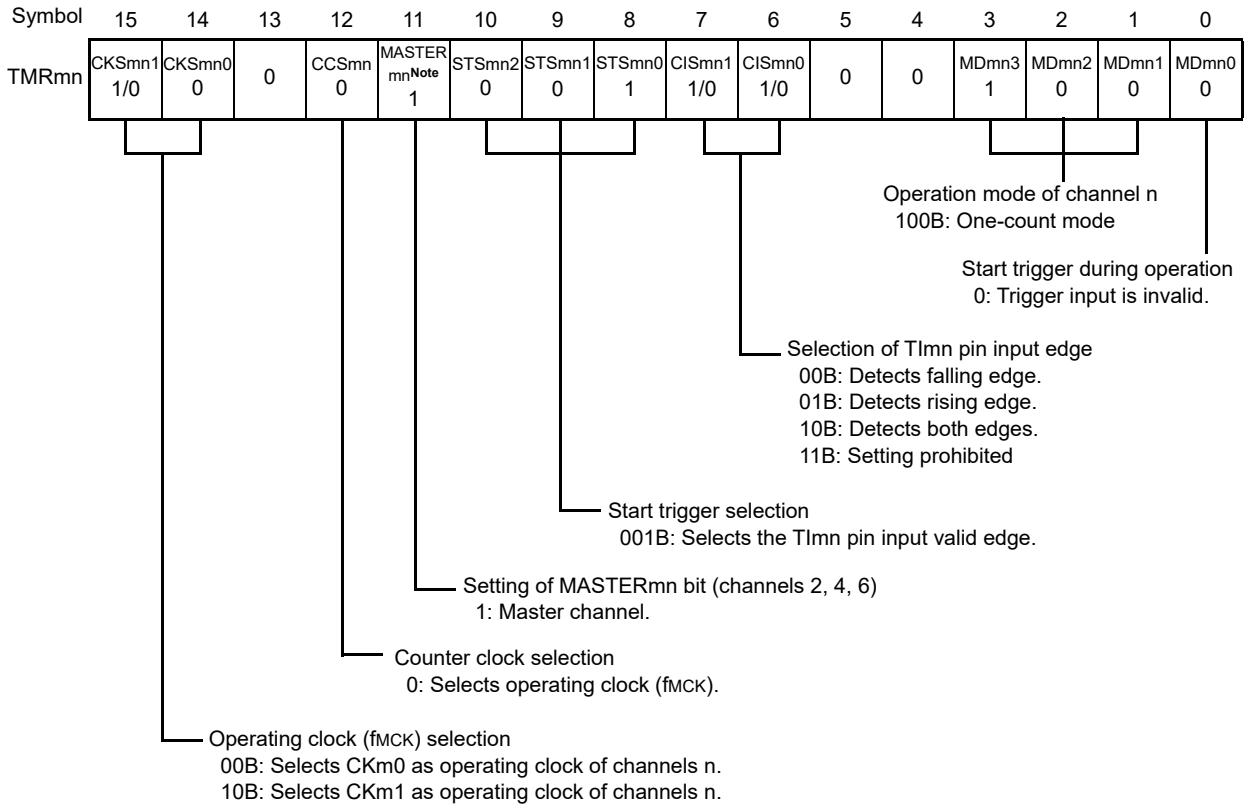


Remark 1. m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Remark 2. TSmn, TSmp: Bits n and p of timer channel start register m (TSm)
TEmn, TEmp: Bits n and p of timer channel enable status register m (TEm)
TImn, TImp: Signals on the TImn and TImp input pins
TCRmn, TCRmp: Timer counter registers mn, mp (TCRmn, TCRmp)
TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)
TOmn, TOmp: Signals on the TOmn and TOmp output pins

Figure 7 - 72 Example of Register Settings for the Master Channel When the One-shot Pulse Output Function Is to Be Used

a) Timer mode register mn (TMRmn)



b) Timer output register m (TOm)

	Bit n
TOm	TOmn 0

0: Outputs 0 from TOmn.

c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn 0

0: Stops the TOmn output operation by counting operation.

d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn 0

0: Set this bit to 0 when TOMmn = 0 (master channel output mode).

e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn 0

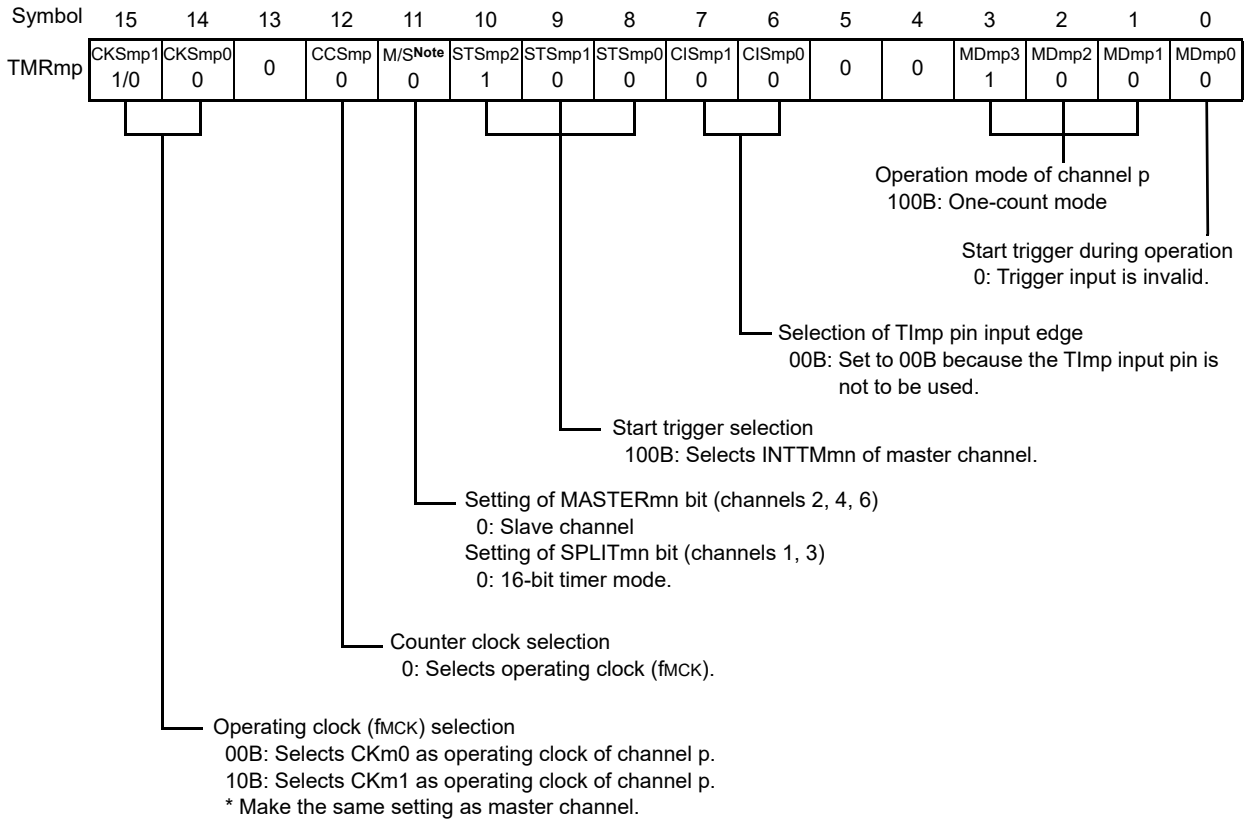
0: Sets master channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmn = 1
TMRm0: Fixed to 0

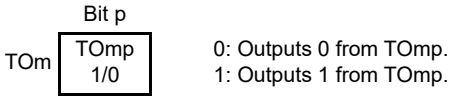
Remark m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)

Figure 7 - 73 Example of Register Settings for the Slave Channel When the One-shot Pulse Output Function Is to Be Used

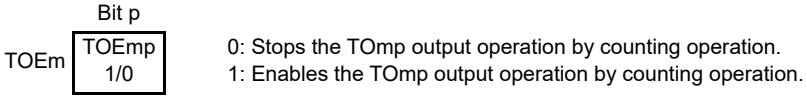
a) Timer mode register mp (TMRmp)



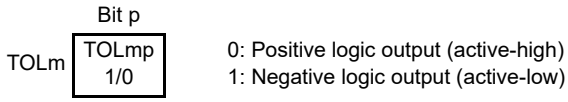
b) Timer output register m (TOm)



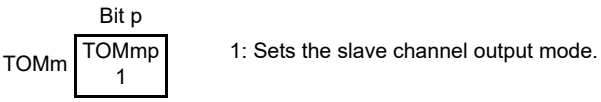
c) Timer output enable register m (TOEm)



d) Timer output level register m (TOLm)



e) Timer output mode register m (TOMm)



Note TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmp bit
TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 7 - 74 Procedure for Operations When the One-shot Pulse Output Function Is to Be Used (1/2)

	Software Operation	Hardware State
TAU default setting		Power-off state (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 state. 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 and CKm1.	
Channel default setting	Sets the corresponding bit of the noise filter enable registers 1, 2 (NFEN1, NFEN2) to 1. 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. → Sets the TOEmp bit to 1 and enables operation of TOmp. → Sets the port register and port mode register to 0. →	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. TOmp does not change because channel stops operating. The TOmp pin outputs the TOmp set level.

Figure 7 - 74 Procedure for Operations When the One-shot Pulse Output Function Is to Be Used (2/2)

	Software Operation	Hardware State
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.	The TEmn and TEmp bits are set to 1 and the master channel enters the start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit of the master channel is set to 1) wait state. Counter stops operating.
	Count operation of the master channel is started by start trigger detection of the master channel. • Detects the TImn pin input valid edge. • Sets the TSmn bit of the master channel to 1 by software[Note].	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 by slave channel can be changed.	Master channel loads the value of the TDRmn register to timer counter register mn (TCRmn) by the start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit of the master channel is set to 1), 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 start trigger detection. 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 counter 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.	TEmn, TEmp = 0, and count operation stops. The TCRmn and TCRmp registers hold count value and stop. The TOmp output is not initialized and retains its current state.
	The TOEmp bit of slave channel is set 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 Sets 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 set to 0. → Set the TAUmRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit. →	This stops supply of the input clock to timer array unit m. All circuits are initialized and SFR of each channel is also initialized. (The TOmn and TOmp bits are cleared to 0 and the TOmn and TOmp pins are set to port mode.)

Operation is resumed. <R>

Note Do not set the TSmn bit of the slave channel to 1.

Remark m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

7.9.2 Operation for the 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} × Counter 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}

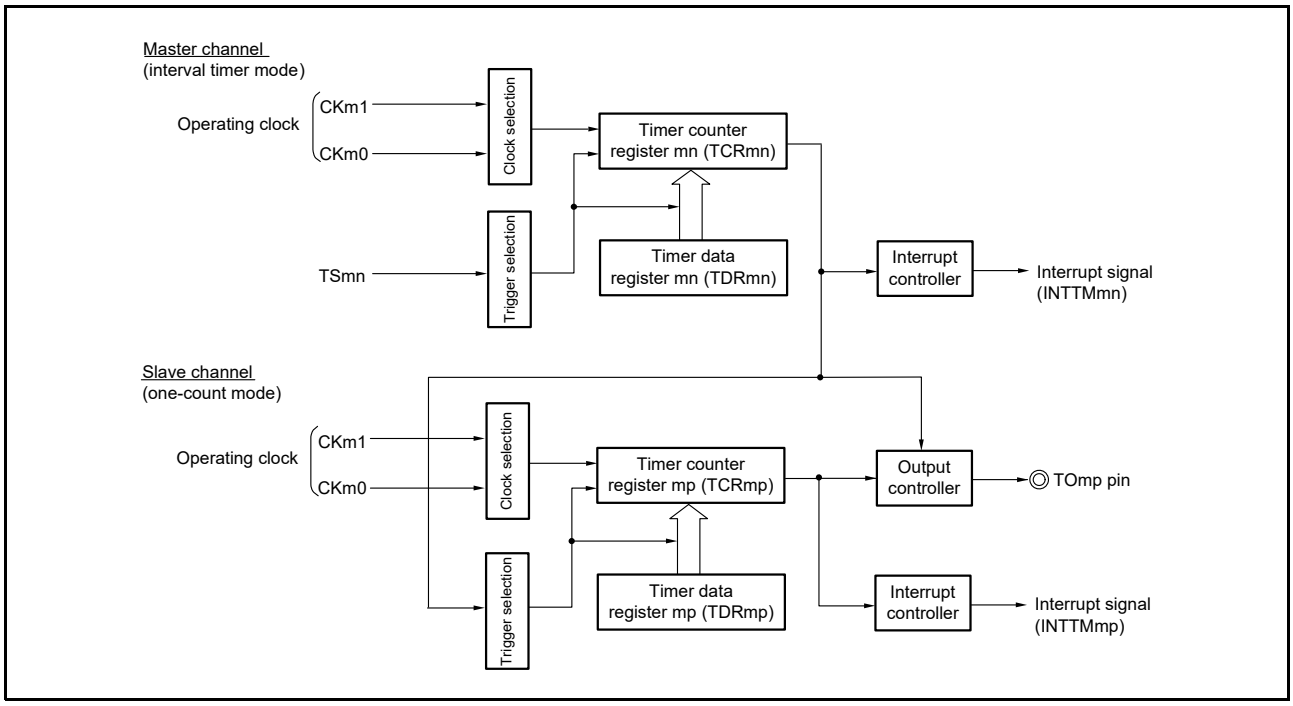
Remark The duty factor exceeds 100% if the set value of TDRmp (slave) > (set value of TDRmn (master) + 1), it summarizes to 100% output.

The master channel operates in the interval timer mode. If the operation enable (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 counter register mn (TCRmn), and the counter counts down in synchronization with the counter 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 operation 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.**

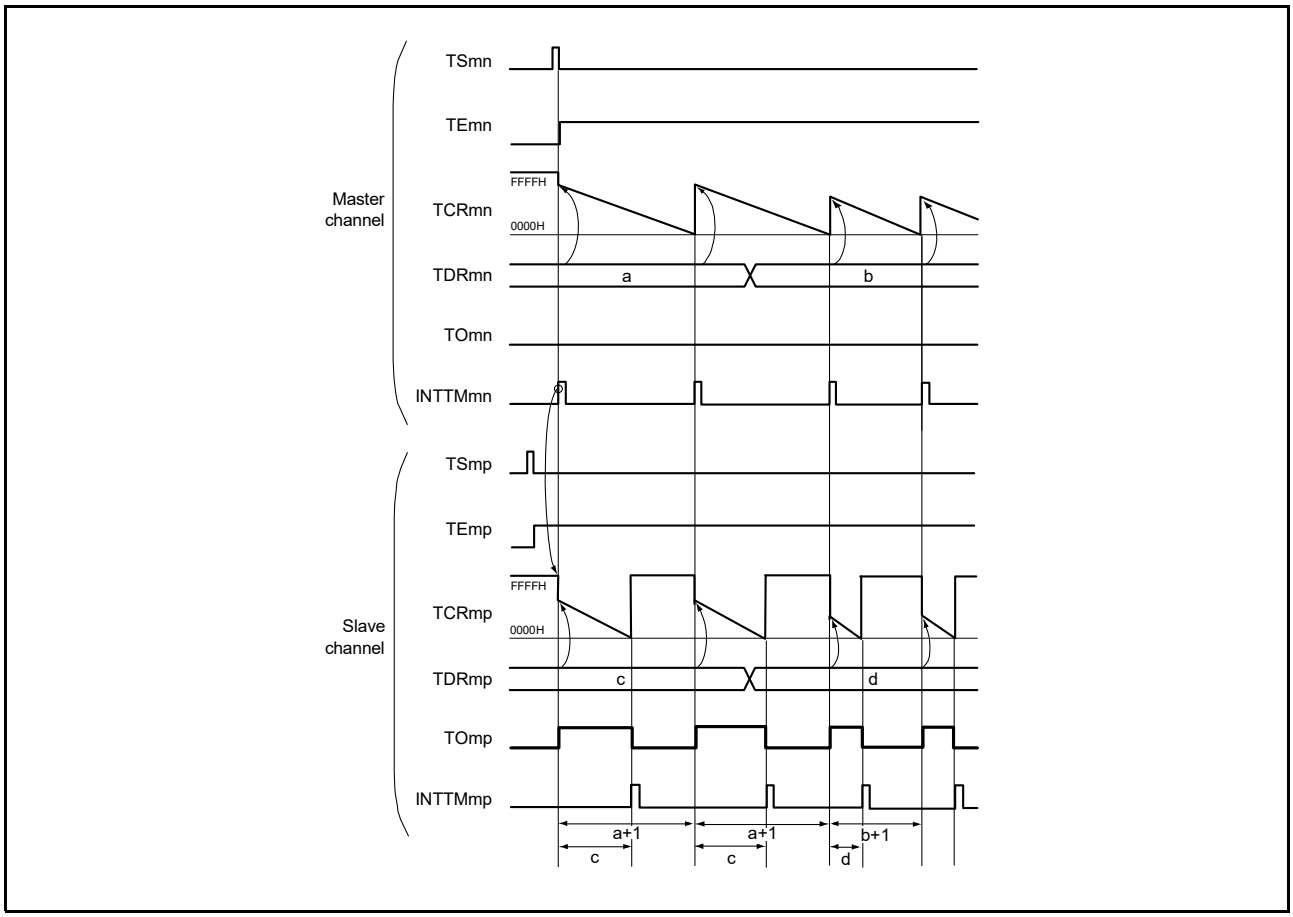
Remark m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 7 - 75 Block Diagram for Operation for the PWM Function



Remark m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 7 - 76 Example of Basic Timing during Operation for the PWM Function

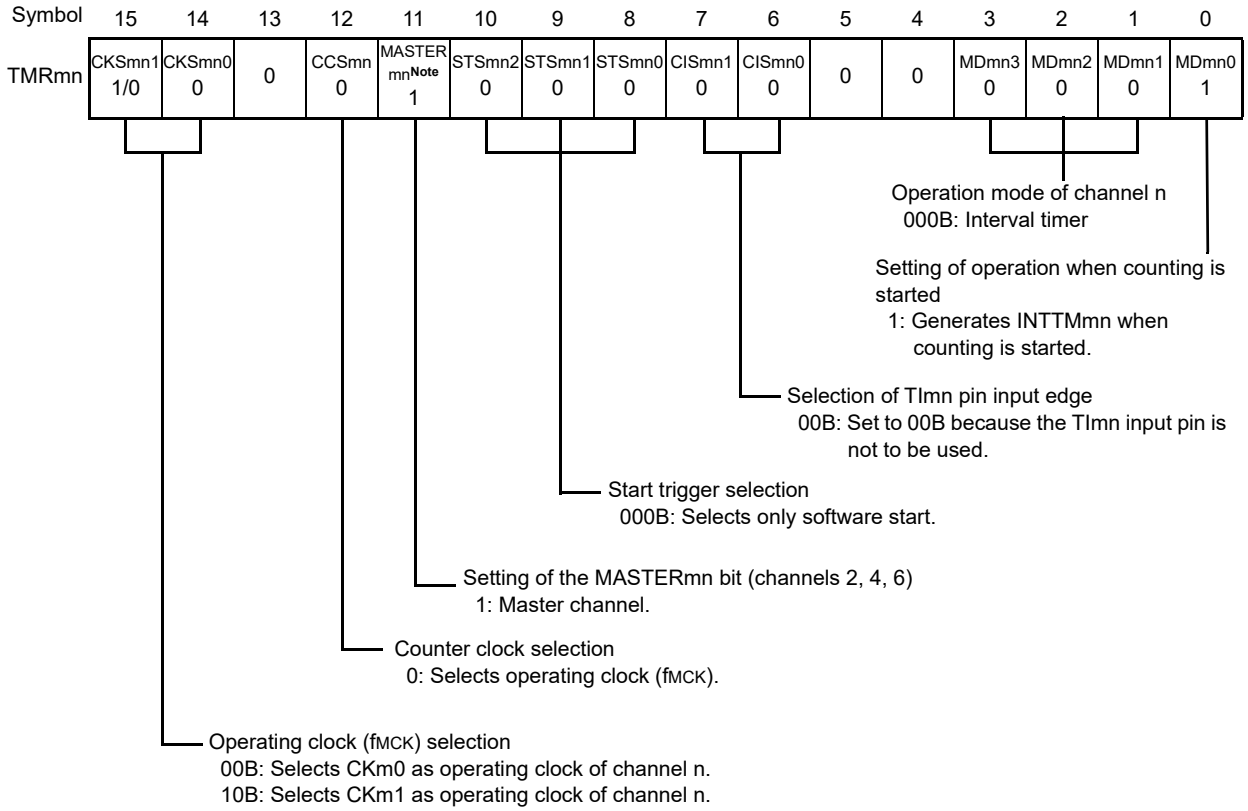


Remark 1. m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Remark 2. TSmn, TSmp: Bits n and p of timer channel start register m (TSm)
TEmn, TEmp: Bits n and p of timer channel enable status register m (TEm)
TCRmn, TCRmp: Timer counter registers mn, mp (TCRmn, TCRmp)
TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)
TOmn, TOmp: Signals on the TOmn and TOmp output pins

Figure 7 - 77 Example of Register Settings for the Master Channel When the PWM Function Is to Be Used

a) Timer mode register mn (TMRmn)



b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops the TOmn output operation by counting operation.

d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Set this bit to 0 when TOMmn = 0 (master channel output mode).

e) Timer output mode register m (TOMm)

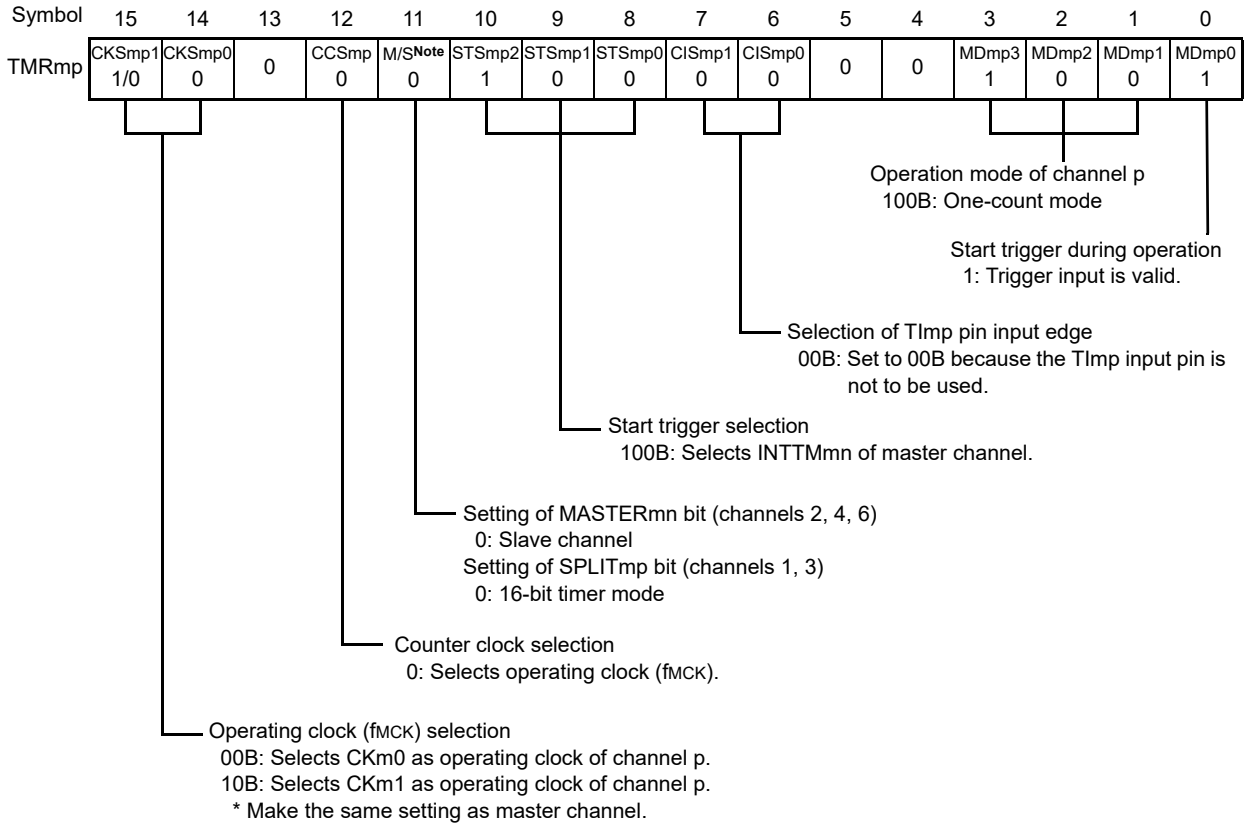
	Bit n	
TOMm	TOMmn 0	0: Sets master channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmn = 1
TMRm0: Fixed to 0

Remark m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)

Figure 7 - 78 Example of Register Settings for the Slave Channel When the PWM Function Is to Be 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 TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmp bit
TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 7 - 79 Procedure for Operations When the PWM Function Is to Be Used (1/2)

	Software Operation	Hardware State
TAU default setting		Power-off state (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 state. 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 and CKm1.	
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. →	TOmp does not change because channel stops operating.
	Sets the port register and port mode register to 0. →	The TOmp pin outputs the TOmp set level.

Figure 7 - 79 Procedure for Operations When the PWM Function Is to Be Used (2/2)

Operation is resumed.

	Software Operation	Hardware State
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.	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 counter 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 counter 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.	TEmn, TEmp = 0, and count operation stops. The TCRmn and TCRmp registers hold count value and stop. The TOmp output is not initialized and retains its current state.
	The TOEmp bit of slave channel is set 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 Sets 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 set to 0. → Set the TAUmRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit. →	This stops supply of the input clock to timer array unit m. All circuits are initialized and SFR of each channel is also initialized. (The TOmn and TOmp bits are cleared to 0 and the TOmn and TOmp pins are set to port mode.)

Remark m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

7.9.3 Operation for the 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} × Counter clock period
Duty factor 1 [%] = {Set value of TDRmp (slave 1)}/{Set value of TDRmn (master) + 1} × 100
Duty factor 2 [%] = {Set value of TDRmq (slave 2)}/{Set value of TDRmn (master) + 1} × 100

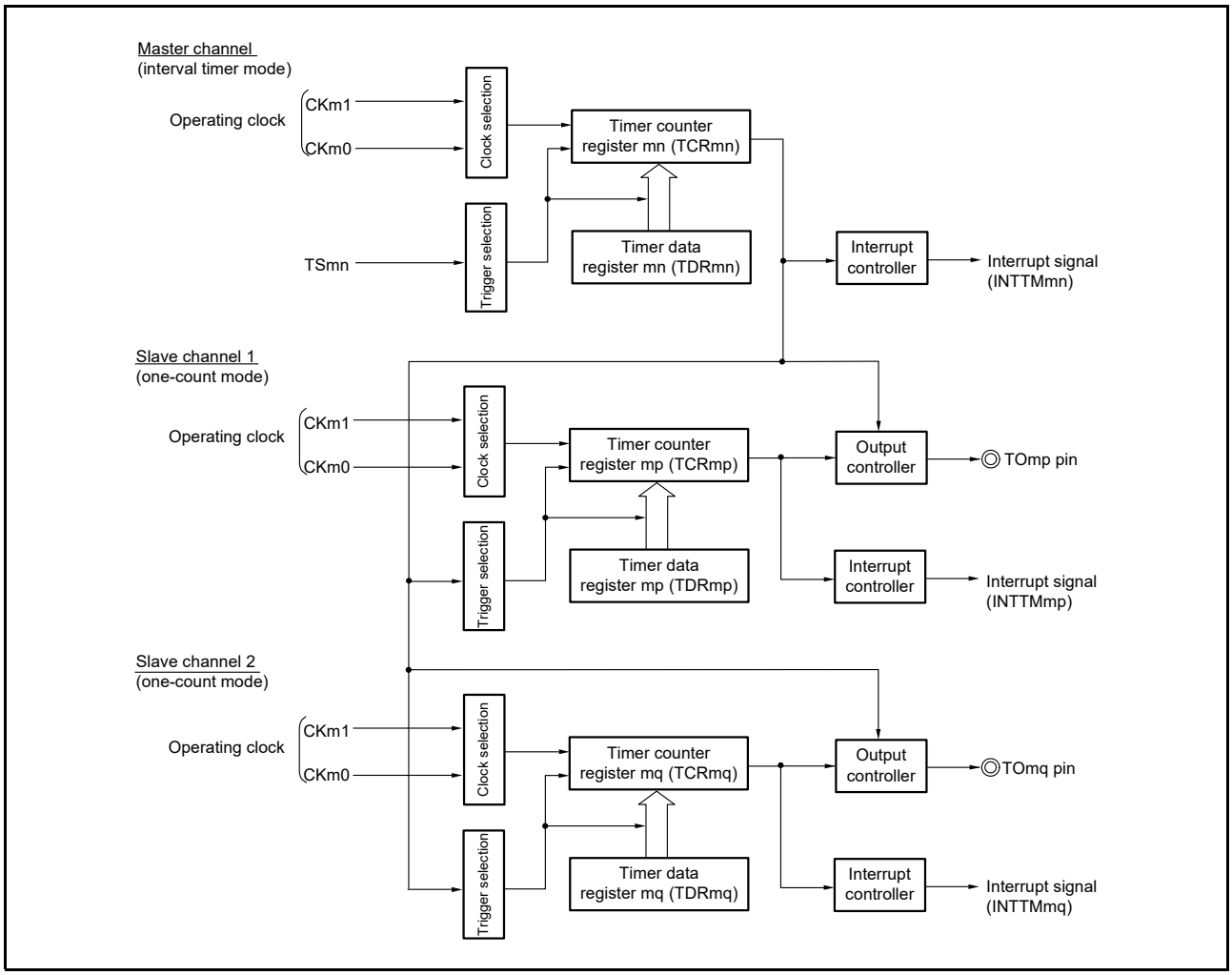
Remark Although the duty factor exceeds 100% if the set value of TDRmp (slave 1) > {set value of TDRmn (master) + 1} or if the {set value of TDRmq (slave 2)} > {set value of TDRmn (master) + 1}, it is summarized into 100% output.

Timer counter 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 counter 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 counter 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 seven 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).**

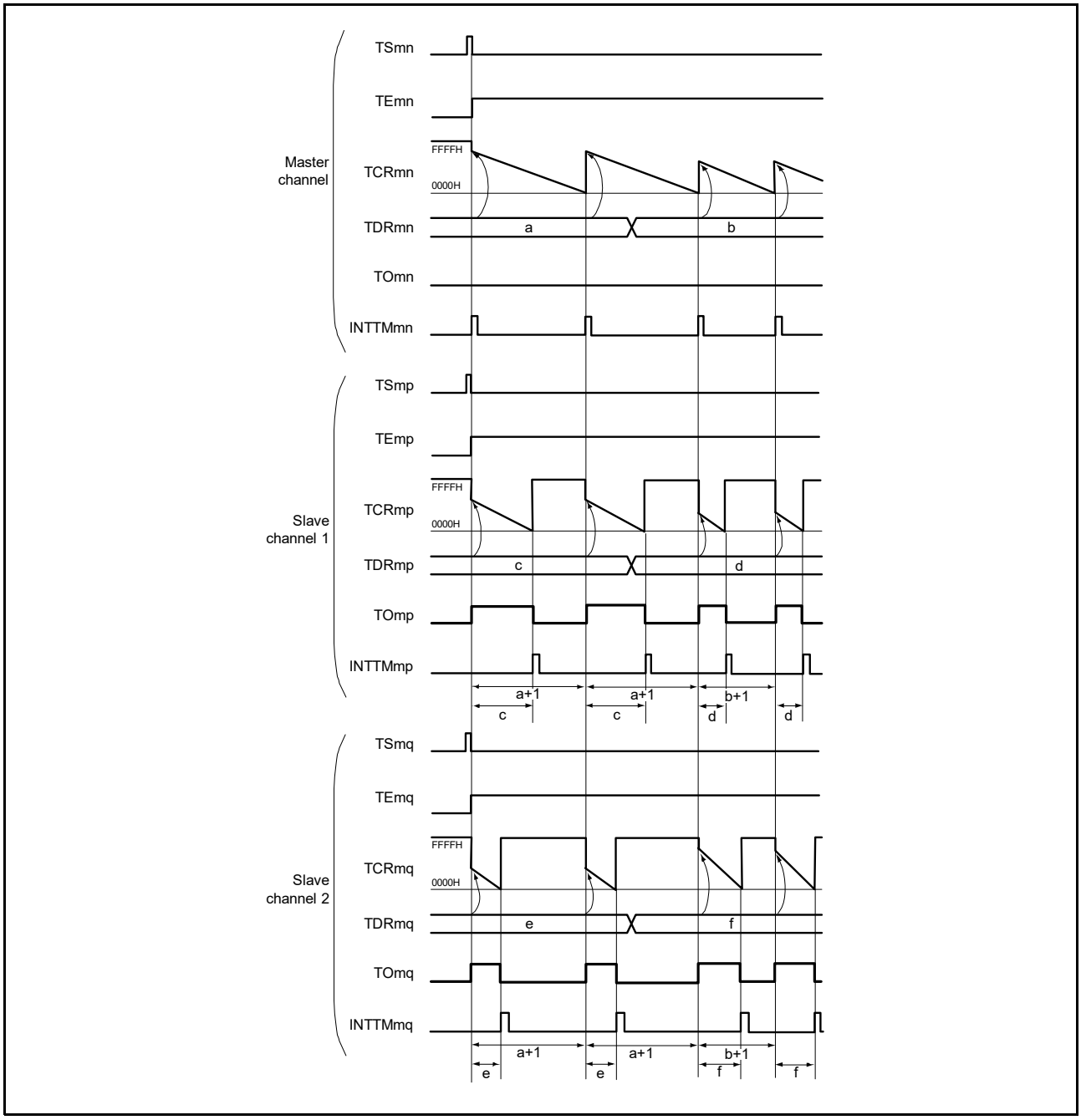
Remark m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 7 (Where p and q are integers)

Figure 7 - 80 Block Diagram for Operation for the Multiple PWM Output Function (for Two Types of PWM Output)



Remark m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 7 (Where p and q are integers)

Figure 7 - 81 Example of Basic Timing during Operation for the Multiple PWM Output Function (for Two Types of PWM Output)

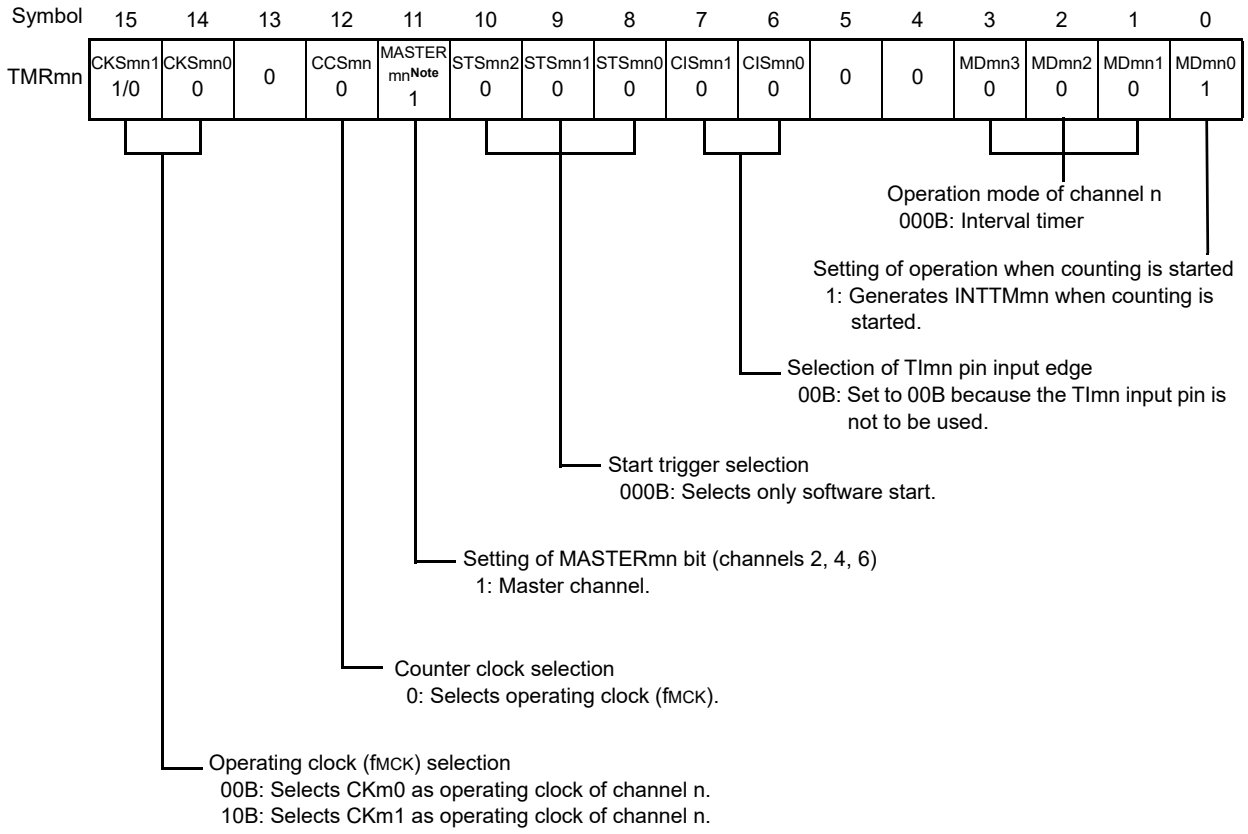


Remark 1. m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 7 (Where p and q are integers)

Remark 2. TSmn, TSmp, TSmq: Bits n, p, and q of timer channel start register m (TSm)
TEmn, TEmp, TEmq: Bits n, p, and q of timer channel enable status register m (TEm)
TCRmn, TCRmp, TCRmq: Timer counter registers mn, mp, mq (TCRmn, TCRmp, TCRmq)
TDRmn, TDRmp, TDRmq: Timer data registers mn, mp, mq (TDRmn, TDRmp, TDRmq)
TOmn, TOmp, TOmq: Signals on the TOmn, TOmp, and TOmq output pins

Figure 7 - 82 Example of Register Settings for the Master Channel When the Multiple PWM Output Function Is to Be Used

a) Timer mode register mn (TMRmn)



b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops the TOmn output operation by counting operation.

d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Set this bit to 0 when TOMmn = 0 (master channel output mode).

e) Timer output mode register m (TOMm)

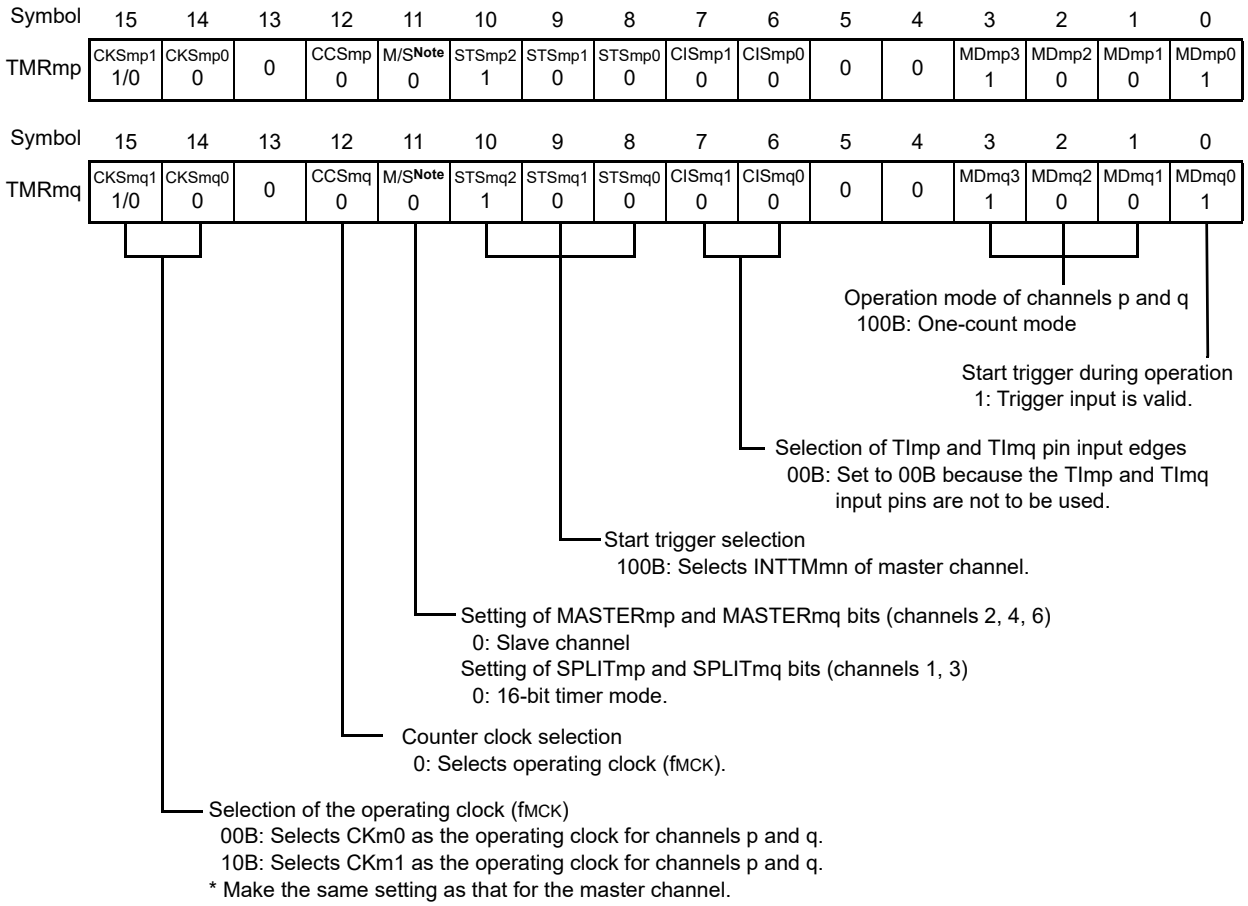
	Bit n	
TOMm	TOMmn 0	0: Sets master channel output mode.

Note TMRm2, TMRm4, TMRm6: MASTERmn = 1
TMRm0: Fixed to 0

Remark m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4)

Figure 7 - 83 Example of Register Settings for the Slave Channel When the Multiple PWM Output Function Is to Be Used (for Two Types of PWM Output)

a) Timer mode registers mp and mq (TMRmp and TMRmq)



b) Timer output register m (TOm)

	Bit q	Bit p	
TOm	TOmq 1/0	TOmp 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 1/0	TOEmp 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 1/0	TOLmp 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 1	TOMmp 1	1: Sets the slave channel output mode.

(Note and Remark are listed on the next page.)

Note TMRm2, TMRm4, TMRm6: MASTERmp and MASTERmq bits
TMRm1, TMRm3: SPLITmp and SPLITmq bits
TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 7 (Where p and q are integers)

Figure 7 - 84 Procedure for Operations When the Multiple PWM Output Function Is to Be Used (for Two Types of PWM Output) (1/2)

	Software Operation	Hardware State
TAU default setting		Power-off state (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 state. 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 and CKm1.	
Channel default setting	Sets timer mode registers mn, mp, 0q (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). Sets the TOLmp and TOLmq bits. 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.
	Sets the port register and port mode register to 0. →	The TOmp and TOmq pins output the TOmp and TOmq set levels.

Figure 7 - 84 Procedure for Operations When the Multiple PWM Output Function Is to Be Used (for Two Types of PWM Output) (2/2)

Operation is resumed.

	Software Operation	Hardware State
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. → The TSmn, TSmp, and TSmq bits automatically return to 0.	TEmn = 1, TEmp, TEmq = 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, 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 TSR0q registers are not used.	The counter of the master channel loads the TDRmn register value to timer counter 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 counter 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 counter 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. → The TTmn, TTmp, and TTmq bits automatically return to 0.	TEmn, TEmp, TEmq = 0, and count operation stops. The TCRmn, TCRmp, and TCRmq registers hold count value and stop. The TOmp and TOmq outputs are not initialized and retain their current states.
	The TOEmp and TOEmq bits of slave channels are set 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 Sets the TOmp and TOmq bits to 0 after the value to be held is set to the port register. → When holding the TOmp and TOmq pin output levels are not necessary Setting not required	The TOmp and TOmq pin output levels are held by port function.
	The TAUmEN bit of the PER0 register is set to 0. → Set the TAUmRES bit of the PRR0 register to 1 to initialize all circuits of the timer array unit. →	This stops supply of the input clock to timer array unit m. All circuits are initialized and SFR of each channel is also initialized. (The TOmn and TOmp bits are cleared to 0 and the TOmn and TOmp pins are set to port mode.)

Remark m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2, 4)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 7 (Where p and q are integers)

7.10 Cautions When Using Timer Array Unit

7.10.1 Cautions when using timer output

Pins may be assigned multiplexed timer output and other multiplexed functions. The assignment depends on the product. If you intend to use a timer output, set the outputs from all other multiplexed pin functions to their initial values. For details, see **4.5 Register Settings When Using Multiplexed Function**.

7.10.2 Point for caution when a timer output is to be used as an event input for the ELCL

The timer outputs (TO00 to TO07) of channels 0 to 7 of timer array unit 0 can be used as event inputs for the logic and event link controller (ELCL). When the timer output signal is to be used for event input to the ELCL but not for output to the TO0m pin, set the corresponding bit in port function output enable register 0 (PFOE0) to 0. For details, see **4.3.15 Port function output enable registers (PFOEx)**.

Section 8 Realtime Clock (RTC)

8.1 Functions of Realtime Clock

The realtime clock has the following features.

- Capable of counting years, months, days of the week, dates, hours, minutes, and seconds, for up to 99 years
- Fixed-cycle interrupt (with period selectable from among 0.5 of a second, 1 second, 1 minute, 1 hour, 1 day, or 1 month)
- Alarm interrupt (alarm set by day of week, hour, and minute)
- Pin output function of 1 Hz

The realtime clock interrupt signal (INTRTC) can be used to wake up the MCU from the HALT or STOP mode, or to trigger transitions of the A/D converter to the SNOOZE mode.

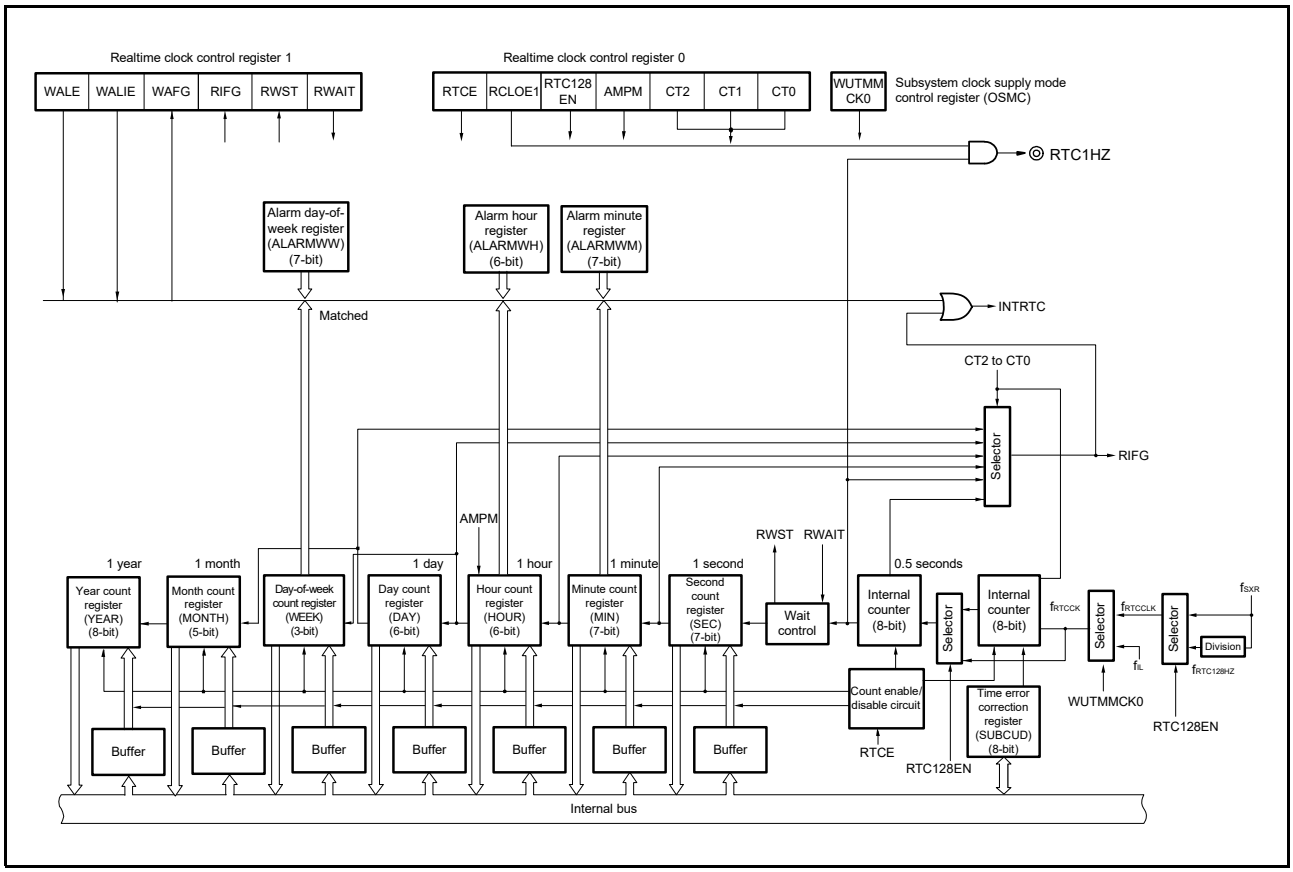
8.2 Configuration of the Realtime Clock

The realtime clock includes the following hardware blocks.

Table 8 - 1 Configuration of the Realtime Clock

Item	Configuration
Counter	Internal counter (16 bits)
Control registers	• Peripheral enable register 0 (PER0) • Subsystem clock supply mode control register (OSMC) • Realtime clock control register 0 (RTCC0) • Realtime clock control register 1 (RTCC1) • Second count register (SEC) • Minute count register (MIN) • Hour count register (HOUR) • Day count register (DAY) • Day-of-week count register (WEEK) • Month count register (MONTH) • Year count register (YEAR) • Time error correction register (SUBCUD) • Alarm minute register (ALARMWM) • Alarm hour register (ALARMWH) • Alarm day-of-week register (ALARMWW)

Figure 8 - 1 Block Diagram of the Realtime Clock



Caution **The count of years, months, weeks, days, hours, minutes, and seconds can only proceed when a subsystem clock (f_{SXR} = 32.768 kHz) is selected as the operating clock of the realtime clock. When the low-speed on-chip oscillator clock (f_{IL} = 32.768 kHz) is selected, only the fixed-cycle interrupt is available.**

8.3 Registers for Controlling the Realtime Clock

The following registers are used to control the realtime clock.

- Peripheral enable register 0 (PER0)
- Subsystem clock supply mode control register (OSMC)
- Realtime clock control register 0 (RTCC0)
- Realtime clock control register 1 (RTCC1)
- Second count register (SEC)
- Minute count register (MIN)
- Hour count register (HOUR)
- Day count register (DAY)
- Day-of-week count register (WEEK)
- Month count register (MONTH)
- Year count register (YEAR)
- Time error correction register (SUBCUD)
- Alarm minute register (ALARMWM)
- Alarm hour register (ALARMWH)
- Alarm day-of-week register (ALARMWW)
- Port mode registers (PMxx)
- Port registers (Pxx)
- Port mode control T registers (PMCTxx)

The following shows the register states depending on reset sources.

Reset Source	System-related Registers[Note 1]	Calendar-related Registers[Note 2]
POR	Reset	Not reset
External reset	Retained	Retained
WDT	Retained	Retained
TRAP	Retained	Retained
LVD	Retained	Retained
Other internal reset sources	Retained	Retained

Note 1. RTCC0, RTCC1, and SUBCUD

Note 2. SEC, MIN, HOUR, DAY, WEEK, MONTH, YEAR, ALARMWM, ALARMWH, and ALARMWW

Assertion of the reset signal does not reset the SEC, MIN, HOUR, DAY, WEEK, MONTH, YEAR, ALARMWM, ALARMWH, or ALARMWW register. Initialize all the registers after power on.

8.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. If the realtime clock is to be used, be sure to set the RTCWEN bit to 1. The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 8 - 2 Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCWEN	IICA1EN **Note 1**	ADCEN	IICA0EN **Note 2**	SAU1EN	SAU0EN	TAU1EN **Note 3**	TAU0EN

RTCWEN	Control of access to the realtime clock
0	• The SFRs used by the realtime clock cannot be written. • The realtime clock can operate.
1	• The SFRs used by the realtime clock can be read and written. • The realtime clock can operate.

Note 1. This bit is only present in the 44- to 128-pin R7F100G products.
Note 2. This bit is only present in the 30- to 128-pin R7F100G and 48-pin R7F104G products.
Note 3. This bit is only present in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products.

Caution 1. Be sure to leave the following bits set to 0.
Bits 6 and 1 in the 30- to 40-pin R7F100G products
Bit 1 in the 44- to 64-pin R7F100G products
Bits 6, 4, and 1 in the 32-pin R7F104G product
Bits 6 and 4 in the 44-pin R7F104G product
Bit 6 in the 48-pin R7F104G product

Caution 2. When the realtime clock is to be used, start by setting the RTCWEN bit to 1 and then set the following registers once oscillation of the counter clock (f_{RTCCK}) has become stable.
- **Realtime clock control register 0 (RTCC0)**
- **Realtime clock control register 1 (RTCC1)**
- **Second count register (SEC)**
- **Minute count register (MIN)**
- **Hour count register (HOUR)**
- **Day count register (DAY)**
- **Day-of-week count register (WEEK)**
- **Month count register (MONTH)**
- **Year count register (YEAR)**
- **Time error correction register (SUBCUD)**
- **Alarm minute register (ALARMWM)**
- **Alarm hour register (ALARMWH)**
- **Alarm day-of-week register (ALARMWW)**

If the setting of RTCWEN is 0, attempted writing to the control registers of the realtime clock is ignored, and 00H is read. Note, however, writing to the following registers is valid.
- **Subsystem clock supply mode control register (OSMC)**
- **Port mode register (PM3)**
- **Port register (P3)**

Caution 3. The subsystem clock supply to peripheral functions other than the realtime clock can be stopped in STOP mode or HALT mode when the subsystem clock is in use, by setting the RTCLPC bit of the subsystem clock supply mode control register (OSMC) to 1.

8.3.2 Subsystem clock supply mode control register (OSMC)

The WUTMMCK0 bit can be used to select the control clock (f_{RTCCK}) for the realtime clock. The RTCLPC bit of this register can be used to reduce the power consumption by disabling supply of the clock signal to peripheral functions that are not in use. For details about setting the RTCLPC bit, see **Section 6 Clock Generator**. The OSMC register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is undefined**Note**.

Note The values of the RTCLPC, WUTMMCK0, and HIPREC bits are respectively 0, 0, and 1.

Figure 8 - 3 Format of Subsystem Clock Supply Mode Control Register (OSMC)

Address: F00F3H
After reset: Undefined**Note 1**
R/W: R/W**Note 2**

Symbol	<7>	6	5	<4>	3	2	1	<0>
OSMC	RTCLPC	0	0	WUTMMCK0	×	×	0	HIPREC

WUTMMCK0	Selection of the operating clock (f_{RTCCK}) for the control block of the realtime clock
0	Subsystem clock XR (f_{SXR}) or $f_{RTC128HZ}$ ($f_{RTC128HZ}$ is selected when the setting of the RTC128EN bit is 1.)
1	Low-speed on-chip oscillator clock (f_{IL})**Notes 3, 4**

Note 1. The values of the RTCLPC, WUTMMCK0, and HIPREC bits are respectively 0, 0, and 1.
Note 2. Bits 3, 2, and 0 are only readable, and writing to them is ignored.
Note 3. Setting the WUTMMCK0 bit to 1 is prohibited while the subsystem clock X is oscillating.
Note 4. Switching between the subsystem clock and the low-speed on-chip oscillator clock by using the WUTMMCK0 bit is only possible while operations of the realtime clock, 32-bit interval timer, serial interfaces UARTA0 and UARTA1, remote control signal receiver, and clock output/buzzer output controller are all stopped.

Caution 1. Be sure to leave bits 6, 5, and 1 set to 0.
Caution 2. Counting of years, months, weeks, days, hours, minutes, and seconds can only proceed when the subsystem clock XR (f_{SXR} = 32.768 kHz) or $f_{RTC128HZ}$ is selected as the operating clock for the control block of the realtime clock. When the low-speed on-chip oscillator clock (f_{IL} = 32.768 kHz) is selected, only the fixed-cycle interrupt is available.

Remark ×: Don't care

8.3.3 Realtime clock control register 0 (RTCC0)

The RTCC0 is an 8-bit register that is used to start or stop the realtime clock operation, control the RTC1HZ pin, and set a 12- or 24-hour system and the fixed-cycle interrupt. The RTCC0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following an internal reset by the power-on reset circuit is 00H.

Figure 8 - 4 Format of Realtime Clock Control Register 0 (RTCC0) (1/2)

Address: F022BH
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	3	2	1	0
RTCC0	RTCE	0	RCLOE1 **Note 1**	RTC128EN	AMPM	CT2	CT1	CT0

RTCE**Note 2**	Realtime 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).

RTC128EN	Selection of the operating clock for the realtime clock (f_{RTCCLK})
0	32.768 kHz
1	128 Hz
• Setting this bit to 1 enables the realtime clock to operate with the 128-Hz clock for lower-power operation. • Time error correction cannot be used when the setting of this bit is 1. • The WUTMMCK0 bit in the OSMC register should be set to 0 when setting this bit to 1.	

AMPM	Selection of 12-/24-hour system
0	12-hour system (a.m. and p.m. are displayed.)
1	24-hour system
• Rewrite this bit value after setting the RWAIT bit of the realtime 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 8 - 2** shows the time (hour) digits indicated according to the setting of this bit.	

Figure 8 - 4 Format of Realtime Clock Control Register 0 (RTCC0) (2/2)

CT2	CT1	CT0	Fixed-cycle interrupt (INTRTC) selection
0	0	0	Does not use fixed-cycle interrupt.
0	0	1	Once per 0.5 s (synchronized with second count up)
0	1	0	Once per second (same time as second count up)
0	1	1	Once per minute (second 00 of every minute)
1	0	0	Once an hour (minute 00 and second 00 of every hour)
1	0	1	Once a day (hour 00, minute 00, and second 00 of every day)
1	1	×	Once a month (Day 1, hour 00 a.m., minute 00, and second 00 of every month)
To change the values of the CT2 to CT0 bits while counting is in progress (RTCE = 1), rewrite the values of the CT2 to CT0 bits after disabling interrupt processing of INTRTC by using the interrupt mask flag register. Furthermore, after rewriting the values of the CT2 to CT0 bits, enable interrupt processing after clearing the RIFG and RTCIF flags.			

Note 1. Be sure to leave this bit set to 0 in the 32-pin R7F104G product.

Note 2. To shift to the HALT or STOP mode immediately after setting the RTCE bit to 1, follow the procedure in **Figure 8 - 18 Procedure for Shifting to HALT or STOP Mode after Setting RTCE Bit to 1**.

Caution 1. Be sure to leave bit 6 set to 0.

Caution 2. Do not change the value of the RCLOE1 bit when RTCE is 1.

Caution 3. 1 Hz is not output even if RCLOE1 is set to 1 when RTCE is 0.

Caution 4. Do not use the SNOOZE mode sequencer for access to the RTCC0 register in standby mode.

Remark ×: Don’t care

8.3.4 Realtime clock control register 1 (RTCC1)

The RTCC1 is an 8-bit register that is used to control the alarm interrupt and the wait time of the counter. The RTCC1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following an internal reset by the power-on reset circuit is 00H.

Figure 8 - 5 Format of Realtime Clock Control Register 1 (RTCC1) (1/2)

Address: F022CH
After reset: 00H
R/W: R/W

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 this bit while counting is in progress (RTCE = 1) and WALIE is 1, rewrite this bit after disabling processing of the INTRTC interrupt by using the interrupt mask flag register. Furthermore, clear the WAFG and RTCIF flags after rewriting this bit. When setting any of the alarm-related registers (WALIE flag of realtime clock control register 1 (RTCC1), the alarm minute register (ALARMWM), the alarm hour register (ALARMWH), and the alarm day-of-week register (ALARMWW)), set this bit to 0.	

WALIE	Control of alarm interrupt (INTRTC)
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 only valid when WALE is 1 and is set to 1 one cycle of f_{RTCCK} after matching of the alarm is detected. This flag is cleared when 0 is written to it. Writing 1 to it is invalid.	

RIFG	Fixed-cycle 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.	

RWST	Wait status flag of realtime clock**Note 1**
0	Counting is in progress.
1	Counter values are readable and writable.
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.	

Figure 8 - 5 Format of Realtime Clock Control Register 1 (RTCC1) (2/2)

RWAIT	Wait control of realtime clock
0	Counting proceeds.
1	Stops the SEC to YEAR counters. Counter values are readable and writable.
This bit controls the operation of the counter. Be sure to write 1 to this bit to read or write the counter value. As the 16-bit internal counter continues to run, return the value of this bit to 0 on completion of reading or writing within one second. When reading or writing to the counter is required while generation of the alarm interrupt is enabled, first set the CT2 to CT0 bits in the RTCC0 register to 010B (generating the fixed-cycle interrupt once per second). Then, complete the processing from setting the RWAIT bit to 1 to setting it to 0 before generation of the next fixed-cycle interrupt. After setting this bit to 1, it takes up to one cycle of f_{RTCCK} until the counter value can be actually read or written (RWST = 1).**Notes 2, 3** When the internal counter (16 bits) overflows while the setting of this bit is 1, an indicator of the counter having overflowed is retained after RWAIT has become 0, after which counting up continues. Note that, when the second count register has been written to, the overflow is not retained.	

Note 1. Bit 1 is only readable.

Note 2. When the RWAIT bit is set to 1 within one cycle of f_{RTCCK} clock after setting the RTCE bit to 1, the setting of the RWST bit actually becoming 1 may take up to two cycles of the operating clock (f_{RTCCK}).

Note 3. When the RWAIT bit is set to 1 within one cycle of f_{RTCCK} clock after release from the standby mode (HALT, STOP, or SNOOZE mode), the setting of the RWST bit actually becoming 1 may take up to two cycles of the operating clock (f_{RTCCK}).

Caution 1. Note that using a 1-bit memory manipulation instruction for writing to the RTCC1 register may lead to clearing of the RIFG and WAFG flags. Therefore, when writing to the RTCC1 register, be sure to use an 8-bit memory manipulation instruction. To prevent the RIFG and WAFG flags from being cleared during writing, disable writing by setting 1 to the corresponding bit. However, if the RIFG and WAFG flags are not in use and a change to the value does not matter, using a 1-bit memory manipulation instruction for writing to the RTCC1 register does not create a problem.

Caution 2. Do not use the SNOOZE mode sequencer for access to the RTCC1 register in standby mode.

Remark 1. Fixed-cycle interrupts and alarm 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.

Remark 2. The internal counter (16 bits) is cleared when the second count register (SEC) is written.

8.3.5 Second count register (SEC)

The SEC is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of seconds. This counter is incremented each time the internal counter (16-bit) overflows. When data are written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTCCK} later. Set a decimal value of 00 to 59 to this register in BCD code. The SEC register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Figure 8 - 6 Format of Second Count Register (SEC)

Address: F0220H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1

Caution 1. When reading from or writing to this register while the counter is in operation (RTCE = 1), follow the procedures described in 8.4.3 Reading from and writing to the counters of the realtime clock.

Caution 2. Do not use the SNOOZE mode sequencer for access to the SEC register in standby mode.

Remark The internal counter (16 bits) is cleared when the second count register (SEC) is written.

8.3.6 Minute count register (MIN)

The MIN is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of minutes. This counter is incremented each time the second counter overflows. When data are written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTCCK} 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. The MIN register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Figure 8 - 7 Format of Minute Count Register (MIN)

Address: F0221H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1

Caution 1. When reading from or writing to this register while the counter is in operation (RTCE = 1), follow the procedures described in 8.4.3 Reading from and writing to the counters of the realtime clock.

Caution 2. Do not use the SNOOZE mode sequencer for access to the MIN register in standby mode.

8.3.7 Hour count register (HOUR)

The HOUR is an 8-bit register that takes a value of 0 to 23 or 1 to 12 and 21 to 32 (decimal) and indicates the count value of hours. This counter is incremented each time the minute counter overflows. When data are written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTCCK} 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 the AMPM bit of realtime 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. The HOUR register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Figure 8 - 8 Format of Hour Count Register (HOUR)

Address: F0222H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
HOUR	0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1

Caution 1. The HOUR20 bit of the HOUR register indicates AM (0) or PM (1) when AMPM = 0 (that is, when the 12-hour system is selected).

Caution 2. When reading from or writing to this register while the counter is in operation (RTCE = 1), follow the procedures described in 8.4.3 Reading from and writing to the counters of the realtime clock.

Caution 3. Do not use the SNOOZE mode sequencer for access to the HOUR register in standby mode.

Table 8 - 2 shows the relationship between the setting value of the AMPM bit, the hour count register (HOUR) value, and time.

Table 8 - 2 Displayed Time Digits

24-hour Display (RTCC0.AMPM = 1)		12-hour Display (RTCC0.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.

8.3.8 Day count register (DAY)

The DAY is an 8-bit register that takes a value of 1 to 31 (decimal) and indicates the count value of days. This counter is incremented each time the hour counter overflows. Counting by the date counter proceeds as shown below.

- 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 are written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTCCK} 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. The DAY register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Figure 8 - 9 Format of Day Count Register (DAY)

Address: F0224H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1

Caution 1. When reading from or writing to this register while the counter is in operation (RTCE = 1), follow the procedures described in 8.4.3 Reading from and writing to the counters of the realtime clock.

Caution 2. Do not use the SNOOZE mode sequencer for access to the DAY register in standby mode.

8.3.9 Day-of-week count register (WEEK)

The WEEK is an 8-bit register that takes a value of 0 to 6 (decimal) and indicates the count value of days of the week. This counter is incremented in synchronization with the date counter. When data are written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTCCK} later. Set a decimal value of 00 to 06 to this register in BCD code. The WEEK register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Figure 8 - 10 Format of Day-of-week Count Register (WEEK)

Address: F0223H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
WEEK	0	0	0	0	0	WEEK4	WEEK2	WEEK1

Caution 1. The value corresponding to the month count register (MONTH) or the day count register (DAY) is not stored in the day-of-week count register (WEEK) automatically. After release from the reset state, set the day-of-week count register as follows.

Day	WEEK
Sunday	00H
Monday	01H
Tuesday	02H
Wednesday	03H
Thursday	04H
Friday	05H
Saturday	06H

Caution 2. When reading from or writing to this register while the counter is in operation (RTCE = 1), follow the procedures described in 8.4.3 Reading from and writing to the counters of the realtime clock.

Caution 3. Do not use the SNOOZE mode sequencer for access to the WEEK register in standby mode.

8.3.10 Month count register (MONTH)

The MONTH is an 8-bit register that takes a value of 1 to 12 (decimal) and indicates the count value of months. This counter is incremented each time the day counter overflows. When data are written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTCCK} 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. The MONTH register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Figure 8 - 11 Format of Month Count Register (MONTH)

Address: F0225H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

Caution 1. When reading from or writing to this register while the counter is in operation (RTCE = 1), follow the procedures described in 8.4.3 Reading from and writing to the counters of the realtime clock.
Caution 2. Do not use the SNOOZE mode sequencer for access to the MONTH register in standby mode.

8.3.11 Year count register (YEAR)

The YEAR is an 8-bit register that takes a value of 0 to 99 (decimal) and indicates the value of the counter of years. This counter is incremented each time the month count register (MONTH) overflows. Values 00, 04, 08, …, 92, and 96 indicate a leap year. When data are written to this register, it is written to a buffer and then to the counter up to two cycles of f_{RTCCK} 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. The YEAR register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Figure 8 - 12 Format of Year Count Register (YEAR)

Address: F0226H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

Caution 1. When reading from or writing to this register while the counter is in operation (RTCE = 1), follow the procedures described in 8.4.3 Reading from and writing to the counters of the realtime clock.
Caution 2. Do not use the SNOOZE mode sequencer for access to the YEAR register in standby mode.

8.3.12 Time error correction register (SUBCUD)

This register is used to correct the time with high accuracy when it is running slow or fast by adjusting the value that is considered an overflow from the internal counter (16 bits) to the second count register (SEC) (reference value: 7FFFH). The SUBCUD register can be set by an 8-bit memory manipulation instruction. The value of this register following an internal reset by the power-on reset circuit is 00H.

Figure 8 - 13 Format of Time Error Correction Register (SUBCUD)

Address: F0227H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
SUBCUD	DEV	F6	F5	F4	F3	F2	F1	F0

DEV	Setting of time error correction timing
0	Corrects time error when the second digits are at 00, 20, or 40 (every 20 seconds).
1	Corrects time 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	

F6	Setting of time 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 (F6, F5, F4, F3, F2, F1, F0) = (*, 0, 0, 0, 0, 0, *), the time error is not corrected. * is 0 or 1. /F5 to /F0 are the inverted values of the corresponding bits (000011B when 111100B). Range of correction value: (when F6 = 0) 2, 4, 6, 8, … , 120, 122, 124 (when F6 = 1) –2, –4, –6, –8, … , –120, –122, –124	

The range of value that can be corrected by using the time 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 quantization error	±1.53 ppm	±0.51 ppm
Minimum resolution	±3.05 ppm	±1.02 ppm

Caution 1. Time error correction cannot be used in the 128-Hz operating mode (RTC128EN = 1); it can only proceed if the setting of RTC128EN is 0.

Caution 2. Do not use the SNOOZE mode sequencer for access to the SUBCUD register in standby mode.

Remark If a correctable range is –63.1 ppm or lower and 63.1 ppm or higher, set DEV to 0.

8.3.13 Alarm minute register (ALARMWM)

This register is used to set minutes of alarm. The ALARMWM register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

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 8 - 14 Format of Alarm Minute Register (ALARMWM)

Address: F0228H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

Caution Do not use the SNOOZE mode sequencer for access to the ALARMWM register in standby mode.

8.3.14 Alarm hour register (ALARMWH)

This register is used to set hours of alarm. The ALARMWH register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

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 8 - 15 Format of Alarm Hour Register (ALARMWH)

Address: F0229H
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWH	0	0	WH20	WH10	WH8	WH4	WH2	WH1

Caution 1. The WH20 bit of the ALARMWH register indicates AM (0) or PM (1) when AMPM = 0 (the 12-hour system is selected).
Caution 2. Do not use the SNOOZE mode sequencer for access to the ALARMWH register in standby mode.

8.3.15 Alarm day-of-week register (ALARMWW)

This register is used to set days of the week of alarm. The ALARMWW register can be set by an 8-bit memory manipulation instruction. This register is not initialized by a reset signal.

Figure 8 - 16 Format of Alarm Day-of-week Register (ALARMWW)

Address: F022AH
After reset: Undefined
R/W: 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 of week							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

Caution Do not use the SNOOZE mode sequencer for access to the ALARMWW register in standby mode.

8.3.16 Registers for controlling the port functions multiplexed with the realtime clock output

Set the following registers to control the port functions multiplexed with the realtime clock output.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Port mode control T registers (PMCTxx)

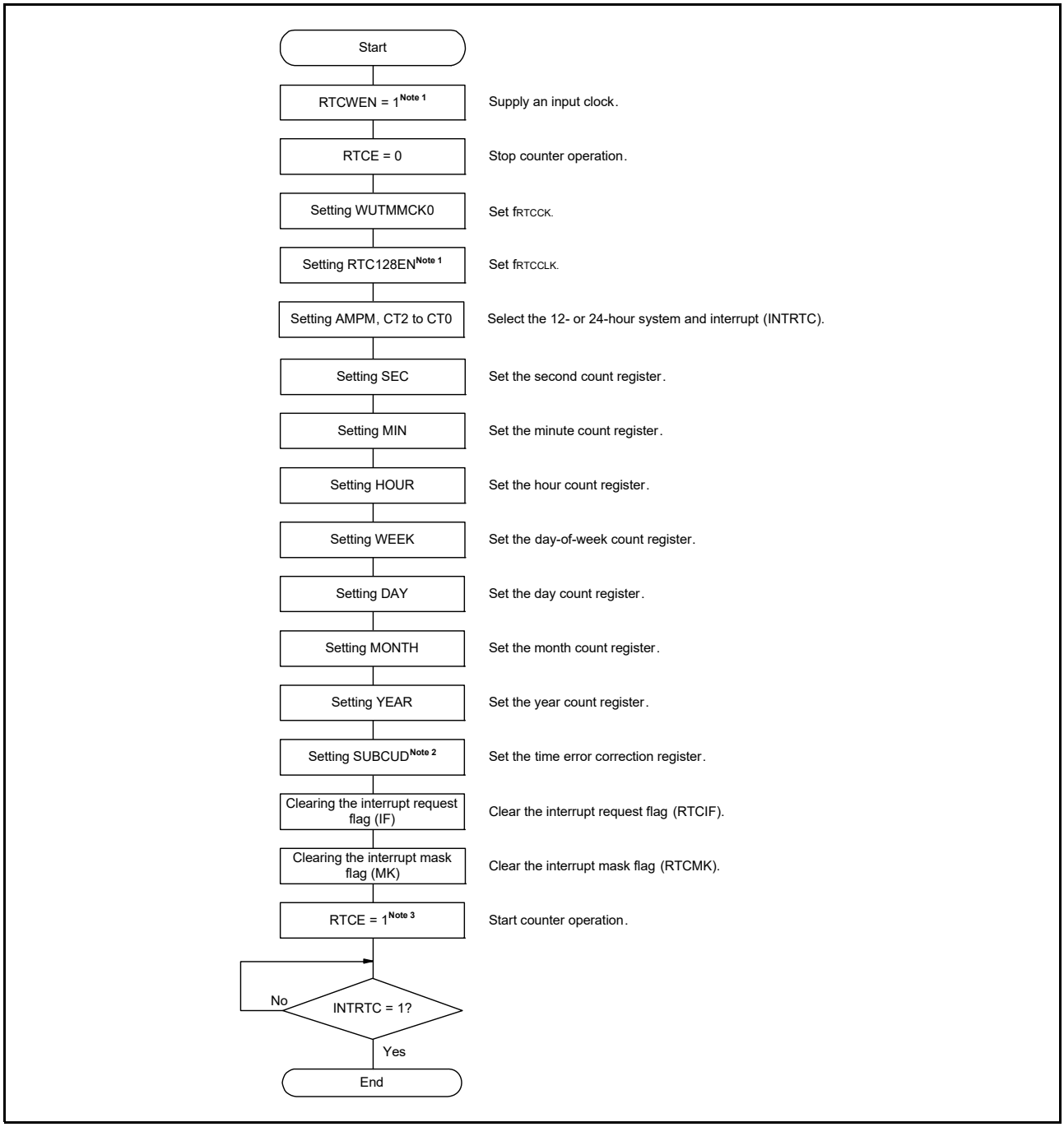
For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.2 Port registers (Pxx)**, and **4.3.8 Port mode control T registers (PMCTxx)**.

For examples of register settings for controlling port functions, see **4.5.4 Examples of register settings for port and multiplexed functions**.

8.4 Operations of the Realtime Clock

8.4.1 Starting the realtime clock operation

Figure 8 - 17 Procedure for Starting the Realtime Clock Operation



Note 1. First set the RTCWEN bit to 1 and set the RTC128EN bit as desired, while oscillation of the count clock (f_{RTCCK}) is stable.

Note 2. Set up the SUBCUD register only if the time error must be corrected. For details about how to calculate the correction value, see **8.4.6 Example of time error correction by the realtime clock**. Time error correction cannot be used while the setting of the RTC128EN bit is 1.

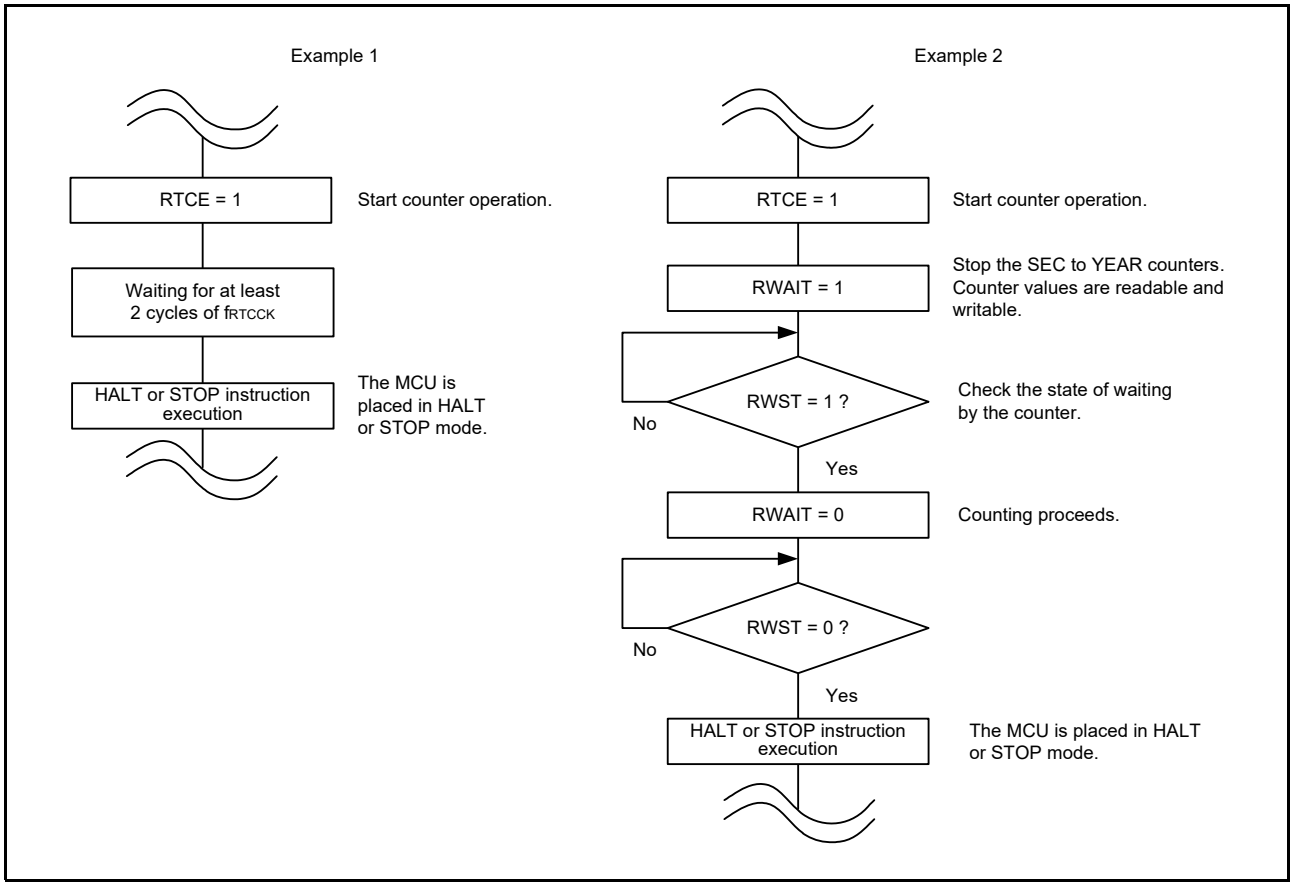
Note 3. Confirm the procedure described in **8.4.2 Shifting to HALT or STOP mode after starting operation** when shifting to HALT or STOP mode without waiting for INTRTC = 1 after RTCE = 1.

8.4.2 Shifting to HALT or STOP mode after starting operation

Take either of the steps listed below when shifting to HALT or STOP mode immediately after setting the RTCE bit to 1. Note that any of these steps is not required when shifting to HALT or STOP mode after the INTRTC interrupt has occurred.

- Make a transition to HALT or STOP mode when at least two counter clock cycles (f_{RTCCK}) have elapsed after setting the RTCE bit to 1 (see **Figure 8 - 18**, **Example 1**).
- After setting the RTCE bit to 1 and then setting the RWAIT bit to 1, poll the RWST bit to check if it has become 1 yet. After setting the RWAIT bit to 0 and polling the RWST bit to check if it has become 0 yet, a transition to HALT or STOP mode will proceed (see **Figure 8 - 18**, **Example 2**).

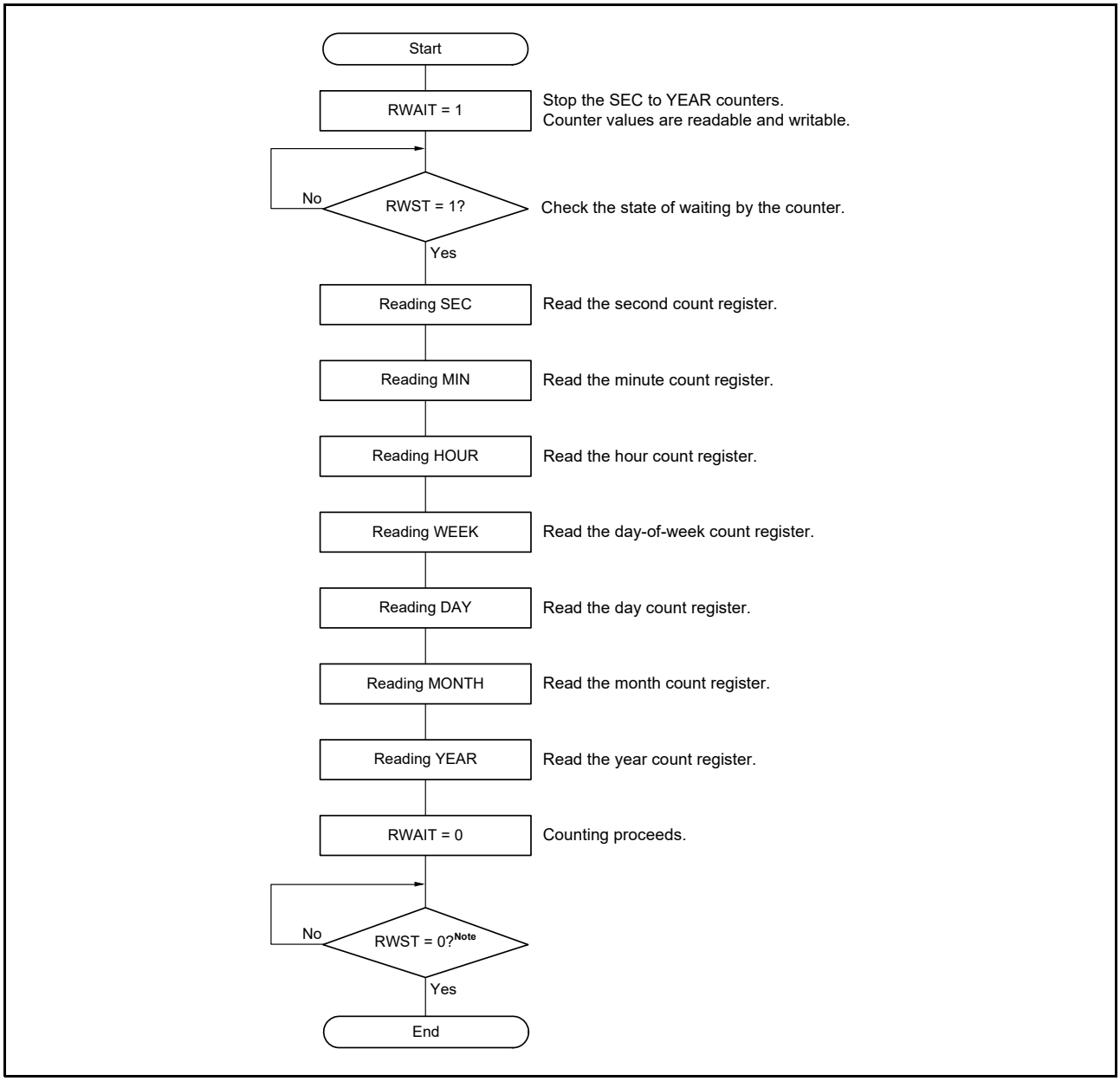
Figure 8 - 18 Procedure for Shifting to HALT or STOP Mode after Setting RTCE Bit to 1



8.4.3 Reading from and writing to the counters of the realtime clock

Read from or write to the counter after setting RWAIT to 1 first. Set RWAIT to 0 after completion of reading from or writing to the counter. When the alarm interrupt is in use, read from or write to the counters according to the procedures shown in **Figure 8 - 20** and **Figure 8 - 22**.

Figure 8 - 19 Procedure for Reading from the Realtime Clock

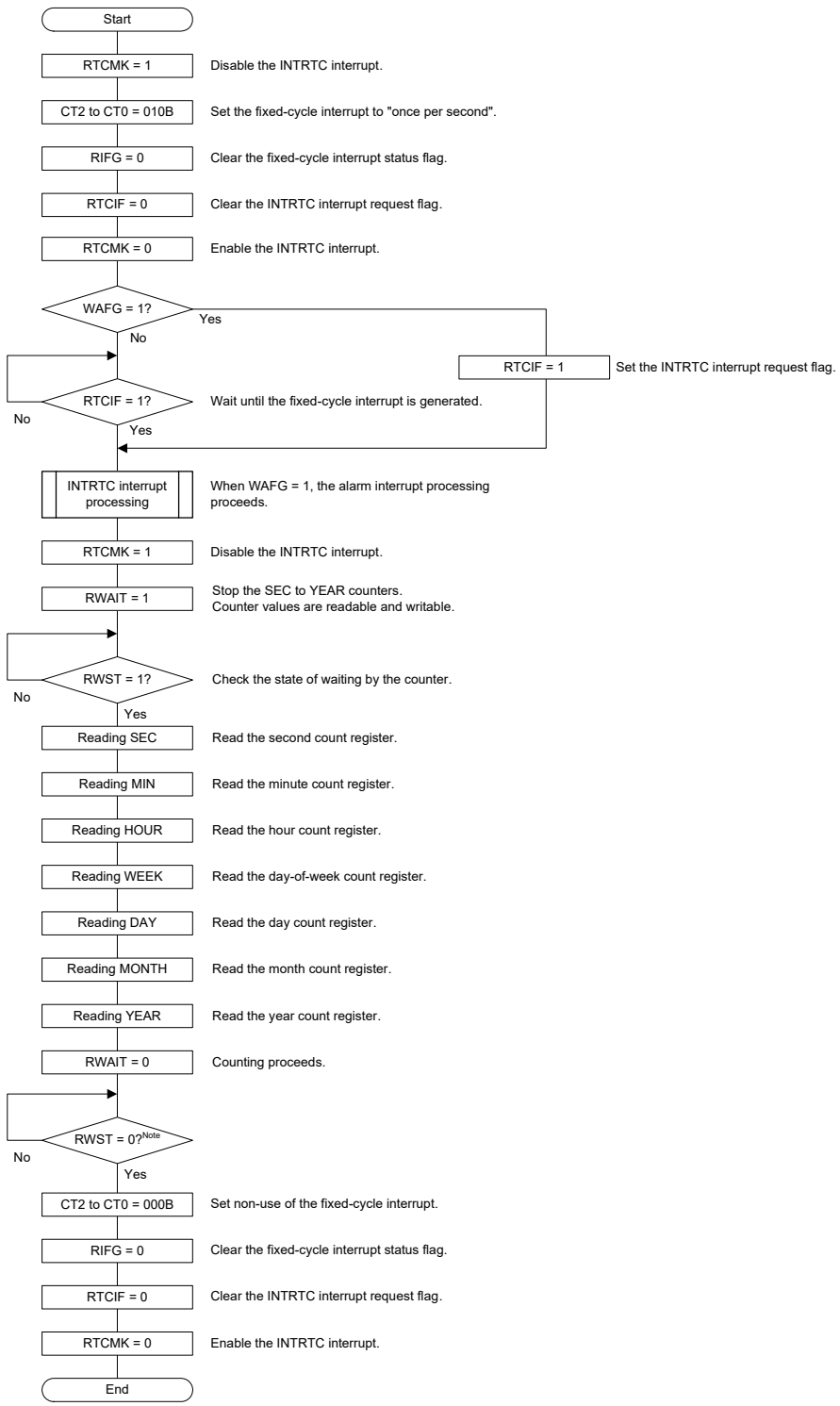


Note Be sure to confirm that RWST = 0 before shifting to the HALT or STOP mode.

Caution **Complete the series of process of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.**

Remark The second count register (SEC), minute count register (MIN), hour count register (HOUR), day-of-week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be read in any sequence. All the registers do not have to be read and only some registers may be read.

Figure 8 - 20 Procedure for Reading from the Realtime Clock When the Alarm Interrupt Is in Use

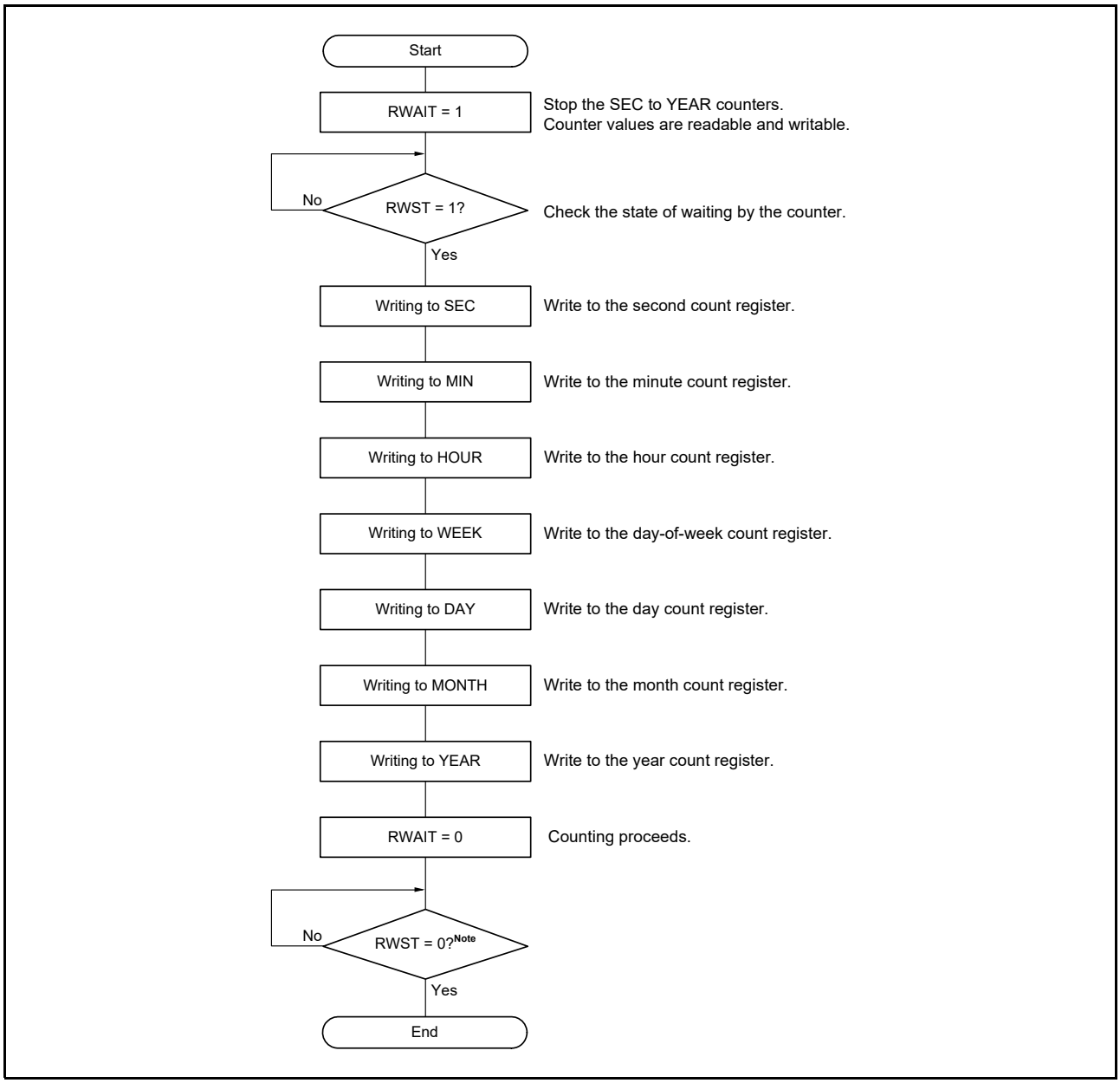


Note Be sure to confirm that RWST = 0 before shifting to the HALT or STOP mode.

Caution **Complete the parts of the process from the start of INTRTC interrupt processing to clearing the RWAIT bit to 0 within 1 second.**

Remark The second count register (SEC), minute count register (MIN), hour count register (HOUR), day-of-week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be read in any sequence. All the registers do not have to be read and only some registers may be read.

Figure 8 - 21 Procedure for Writing to the Realtime Clock



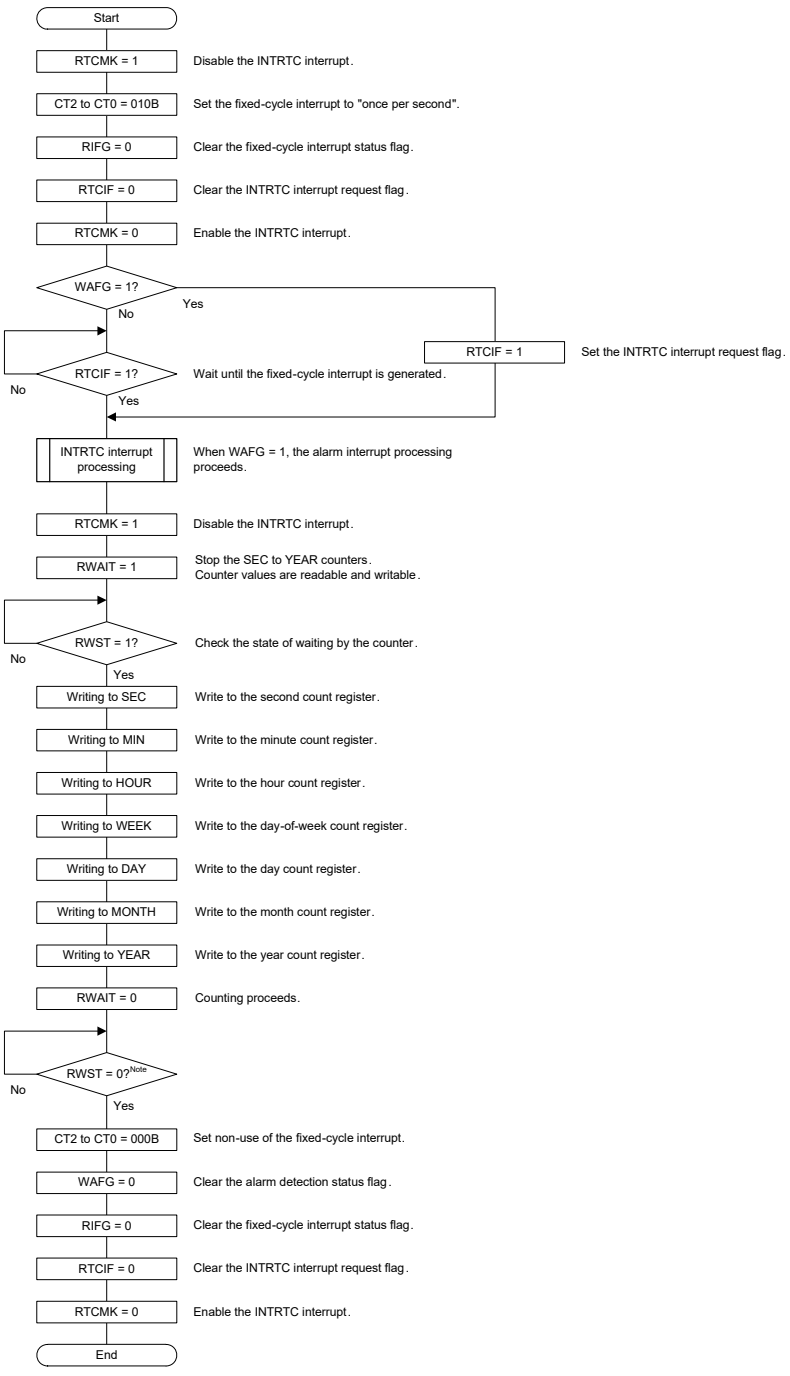
Note Be sure to confirm that RWST = 0 before shifting to the HALT or STOP mode.

Caution 1. Complete the series of operations of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.

Caution 2. When changing the values of the SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR registers while counting is in progress (RTCE = 1), rewrite the registers after disabling interrupt processing of INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG, and RTCIF flags after rewriting the registers.

Remark The second count register (SEC), minute count register (MIN), hour count register (HOUR), day-of-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.

Figure 8 - 22 Procedure for Writing to the Realtime Clock When the Alarm Interrupt Is in Use



Note Be sure to confirm that RWST = 0 before shifting to the HALT or STOP mode.

Caution 1. Complete the parts of the process from the start of INTRTC interrupt processing to clearing the RWAIT bit to 0 within 1 second.

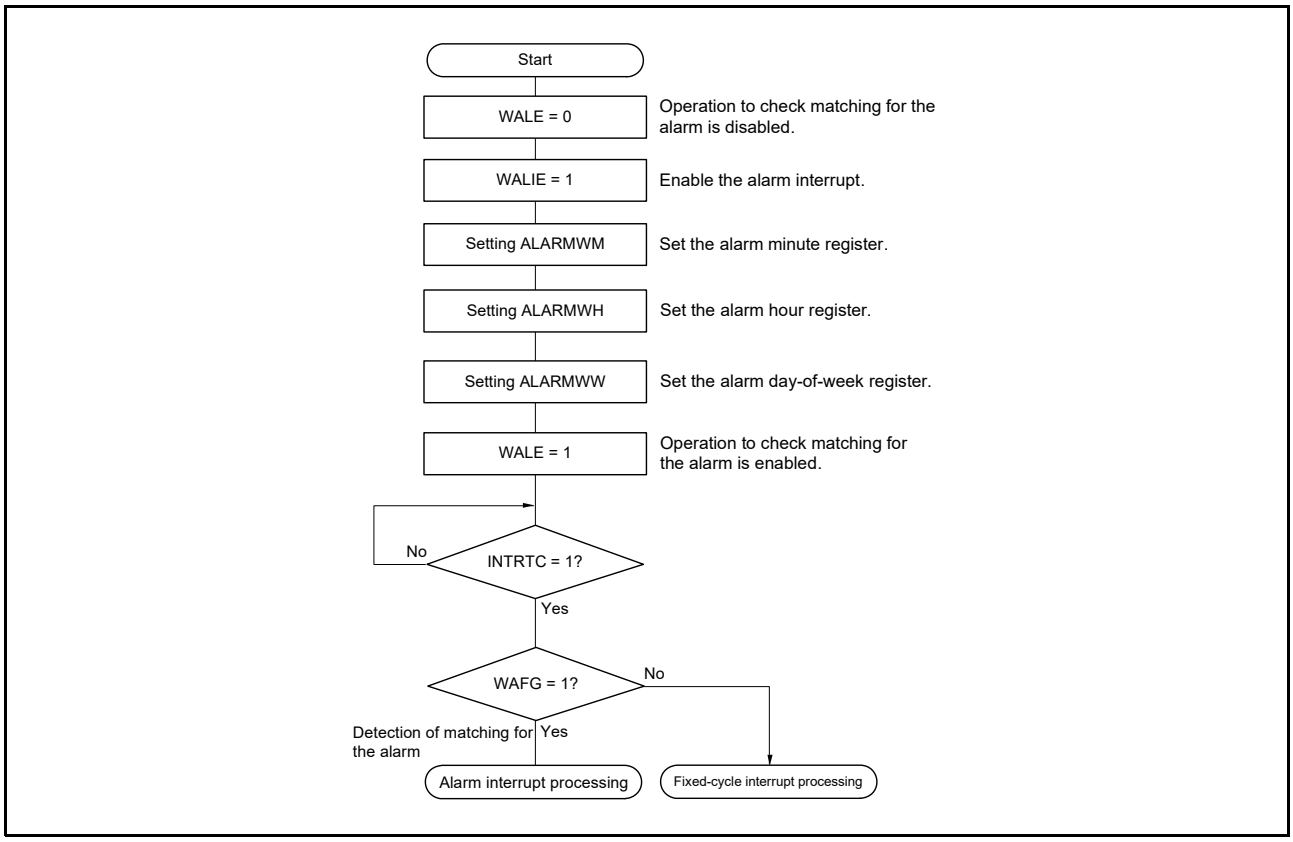
Caution 2. When changing the values of the SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR registers while counting is in progress (RTCE = 1), rewrite the registers after disabling interrupt processing of INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG and RTCIF flags after rewriting the registers.

Remark The second count register (SEC), minute count register (MIN), hour count register (HOUR), day-of-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.

8.4.4 Setting alarm by the realtime clock

Set time of alarm after setting WALE to 0 (alarm operation invalid) first.

Figure 8 - 23 Alarm Processing Procedure

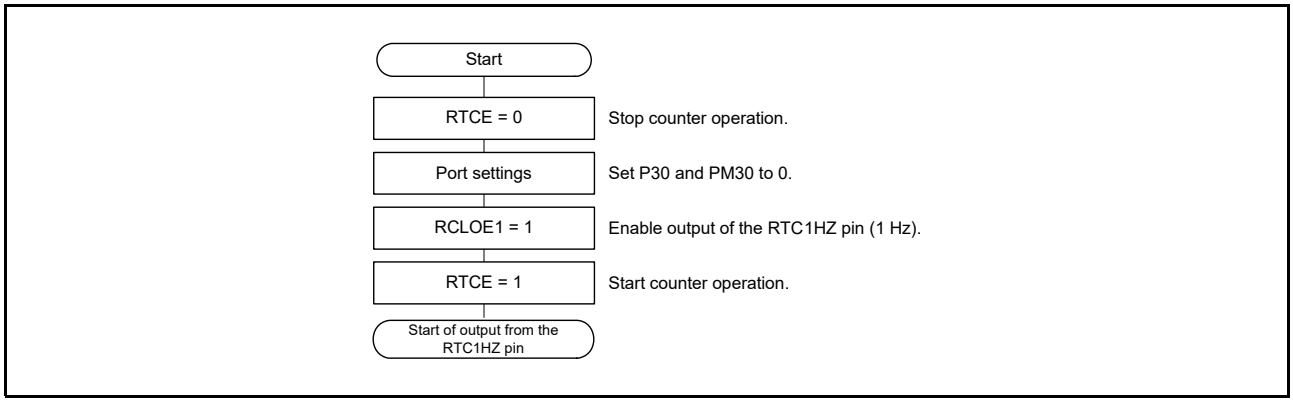


Remark 1. The alarm minute register (ALARMWM), alarm hour register (ALARMWH), and alarm day-of-week register (ALARMWW) may be written in any sequence.

Remark 2. Fixed-cycle interrupts and alarm interrupts use the same interrupt source (INTRTC). To use these two types of interrupts at the same time, the source of the interrupt can be identified by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) when an INTRTC is generated.

8.4.5 1 Hz output by the realtime clock

Figure 8 - 24 1 Hz Output Setting Procedure



Caution 1. First set the RTCWEN bit to 1, while oscillation of the count clock (f_{RTCCK}) is stable.

Caution 2. The setting procedure above is only for the 30- to 128-pin R7F100G and 44- and 48-pin R7F104G products.

8.4.6 Example of time error correction by the realtime clock

Time can be corrected with high accuracy when it is slow or fast, by setting a value to the time error correction register.

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 –63.1 ppm or less, or 63.1 ppm or more.

DEV = 0

Correction value[Note] = Number counted for error correction in 1 minute ÷ 3
= (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60 ÷ 3

DEV = 1

Correction value[Note] = Number counted for error correction in 1 minute
= (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60

Note The correction value is the time error correction value calculated by using bits 6 to 0 of the time error correction register (SUBCUD).
(When F6 = 0) Correction value = {(F5, F4, F3, F2, F1, F0) – 1} × 2
(When F6 = 1) Correction value = – {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2
When (F6, F5, F4, F3, F2, F1, F0) is (*, 0, 0, 0, 0, 0, *), where "*" is 0 or 1, time error is not corrected.
/F5 to /F0 are bit-inverted values (000011B when 111100B).

Remark 1. The correction value is 2, 4, 6, 8, … 120, 122, 124 or –2, –4, –6, –8, … –120, –122, –124.

Remark 2. The oscillation frequency is a value of the count clock (f_{RTCCK}). It can be calculated from the output frequency of the RTC1HZ pin × 32768 when the time error correction register is set to its initial value (00H).

Remark 3. The target frequency is the frequency resulting after correction performed by using the time error correction register.

Correction example 1

Example of correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz – 131.2 ppm)

[Measuring the oscillation frequency]

To measure the oscillation frequency[Note] of each product, a signal at about 32.768 kHz can be output from the
<R> PCLBUZ0 pin or a signal at about 1 Hz can be output from the RTC1HZ pin when the clock error correction register (SUBCUD) is set to its initial value (00H).

Note See **8.4.5 1 Hz output by the realtime clock** for the setting procedure for output of about 1 Hz from the RTC1HZ pin, and see **10.4 Operations of the Clock Output/Buzzer Output Controller** for the setting procedure for output of about 32 kHz from the PCLBUZ0 pin.

[Calculating the correction value]

When the output frequency from the PCLBUZ0 pin is 32772.3 Hz:

Given that the target frequency is 32768 Hz (32772.3 Hz -131.2 ppm) and the extent of correction is -131.2 ppm (not in the range below -63.1 ppm), set DEV to 0. Accordingly, the expression for calculating the correction value when DEV is 0 is applicable.

Correction value = Number counted for error correction in 1 minute ÷ 3
= (Oscillation frequency ÷ target frequency – 1) × 32768 × 60 ÷ 3
= (32772.3 ÷ 32768 – 1) × 32768 × 60 ÷ 3
= 86

[Calculating the values to be set to (F6 to F0)]

When the correction value is 86:

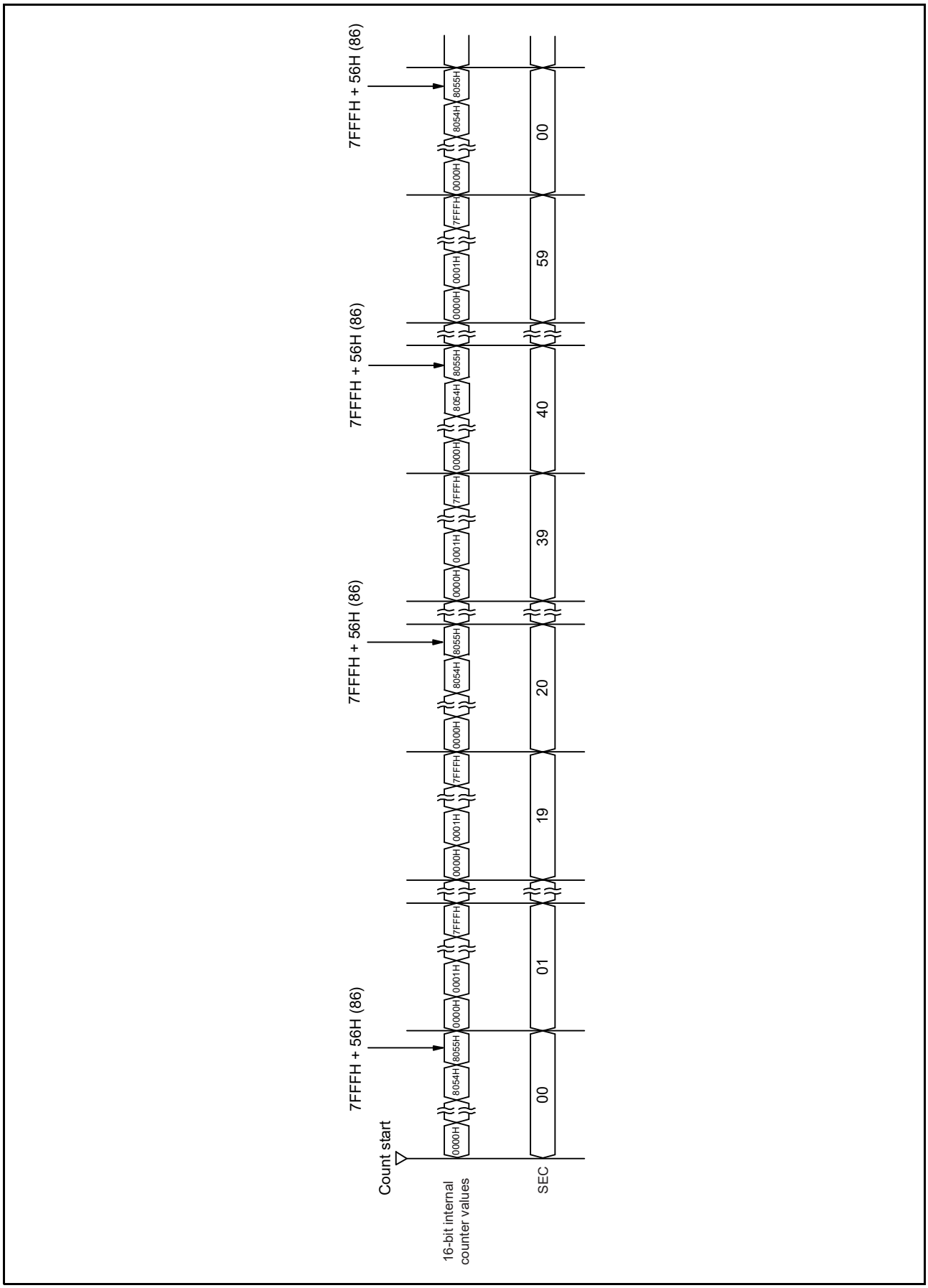
If the correction value is 0 or larger (the clock is running slow), set F6 to 0. Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

{(F5, F4, F3, F2, F1, F0) – 1} × 2 = 86
(F5, F4, F3, F2, F1, F0) = 44
(F5, F4, F3, F2, F1, F0) = (1, 0, 1, 1, 0, 0)

Consequently, when correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz – 131.2 ppm), setting the correction register such that DEV is 0 and the correction value is 86 (bits 6 to 0 of the SUBCUD register: 0101100B) results in the desired frequency of 32768 Hz (error of 0 ppm).

Figure 8 - 25 shows the operation for correction when the value of (DEV, F6, F5, F4, F3, F2, F1, F0) is (0, 0, 1, 0, 1, 1, 0, 0).

Figure 8 - 25 Operation for Correction When the Value of (DEV, F6, F5, F4, F3, F2, F1, F0) = (0, 0, 1, 0, 1, 1, 0, 0)



Correction example 2

Example of correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm)

[Measuring the oscillation frequency]

To measure the oscillation frequency[Note] of each product, a signal at about 1 Hz can be output from the RTC1HZ pin when the clock error correction register (SUBCUD) is set to its initial value (00H).

Note See **8.4.5 1 Hz output by the realtime clock** for the setting procedure for output of about 1 Hz from the RTC1HZ pin.

[Calculating the correction value]

When the output frequency from the RTC1HZ pin is 0.9999817 Hz:
Oscillation frequency = 32768 × 0.9999817 ≈ 32767.4 Hz
Given that the target frequency is 32768 Hz (32767.4 Hz + 18.3 ppm), set DEV to 1. Accordingly, the expression for calculating the correction value when DEV is 1 is applicable.

Correction value = Number counted for error correction in 1 minute
= (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60
= (32767.4 ÷ 32768 – 1) × 32768 × 60
= –36

[Calculating the values to be set to (F6 to F0)]

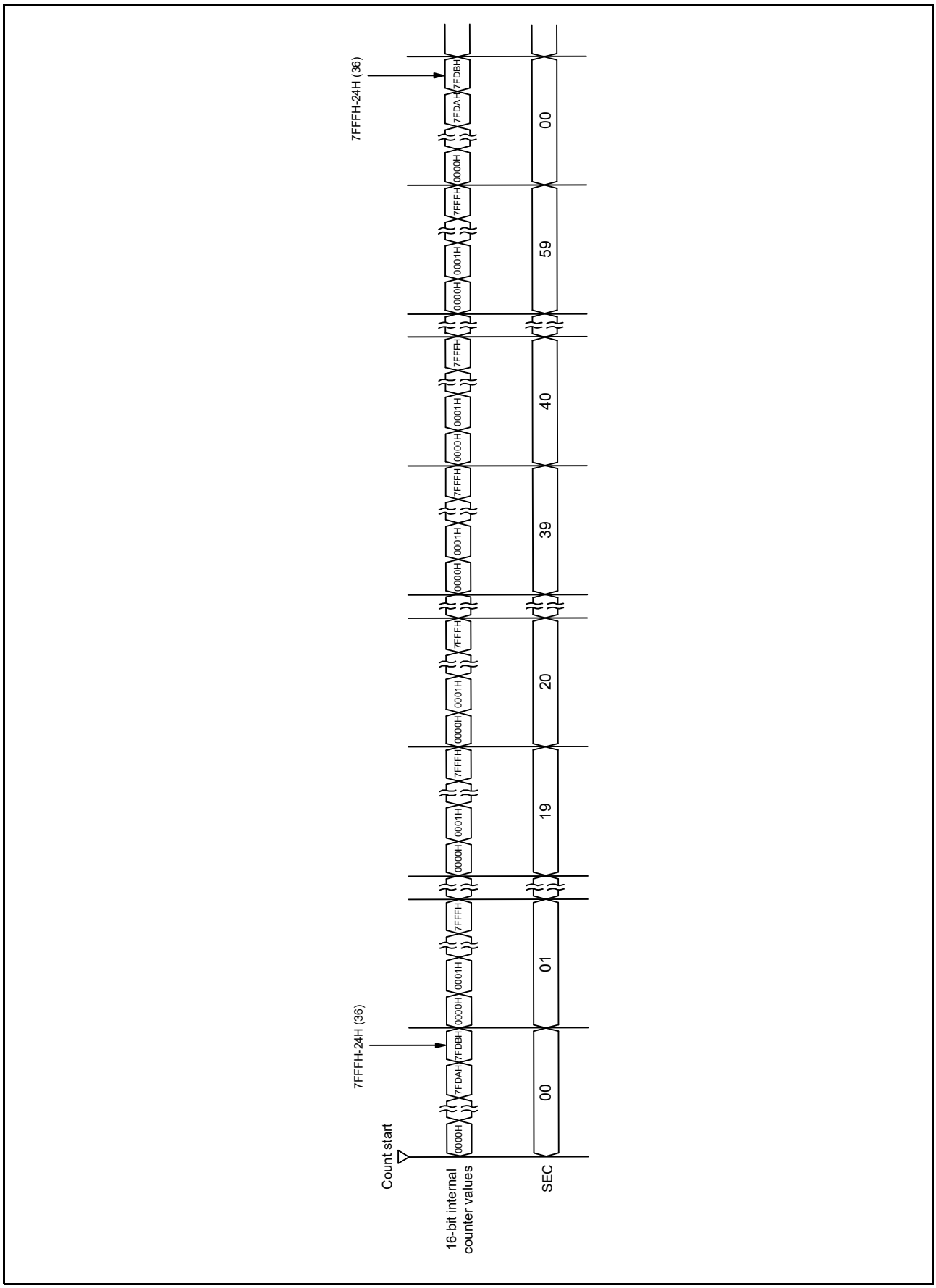
When the correction value is –36:
If the correction value is 0 or less (the clock is running fast), set F6 to 1. Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

– {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2 = –36
(/F5, /F4, /F3, /F2, /F1, /F0) = 17
(/F5, /F4, /F3, /F2, /F1, /F0) = (0, 1, 0, 0, 0, 1)
(F5, F4, F3, F2, F1, F0) = (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 6 to 0 of the SUBCUD register: 1101110B) results in the desired frequency of 32768 Hz (error of 0 ppm).

Figure 8 - 26 shows the operation for correction when the value of (DEV, F6, F5, F4, F3, F2, F1, F0) is (1, 1, 1, 0, 1, 1, 1, 0).

Figure 8 - 26 Operation for Correction When the Value of (DEV, F6, F5, F4, F3, F2, F1, F0) = (1, 1, 1, 0, 1, 1, 1, 0)



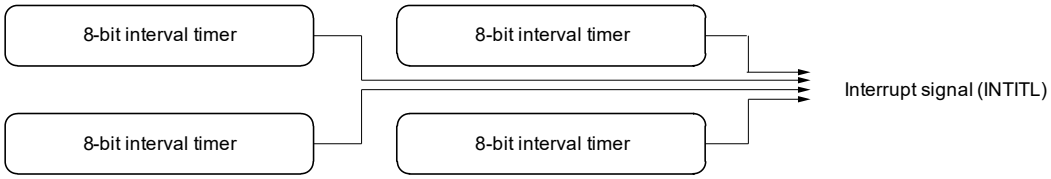
Section 9 32-bit Interval Timer (TML32)

The 32-bit interval timer is made up of four 8-bit interval timers (referred to as channels 0 to 3). Each is capable of operating independently and in that case they all have the same functions. Two 8-bit interval timer channels can be connected to operate as a 16-bit interval timer. Four 8-bit interval timer channels can be connected to operate as a 32-bit interval timer.

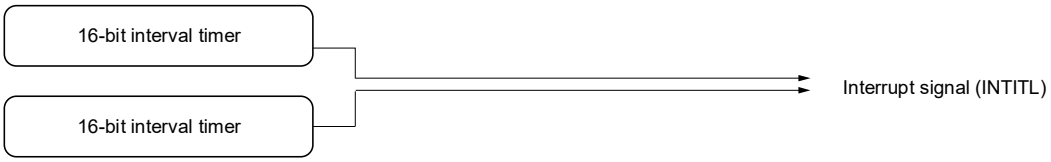
9.1 Overview

The 32-bit interval timer operates with the f_{MXP}, f_{SXP}, f_{IHP}, or f_{IMP} clock or the event input from the ELCL, which is asynchronous with the CPU operation. The 32-bit interval timer has the following functions.

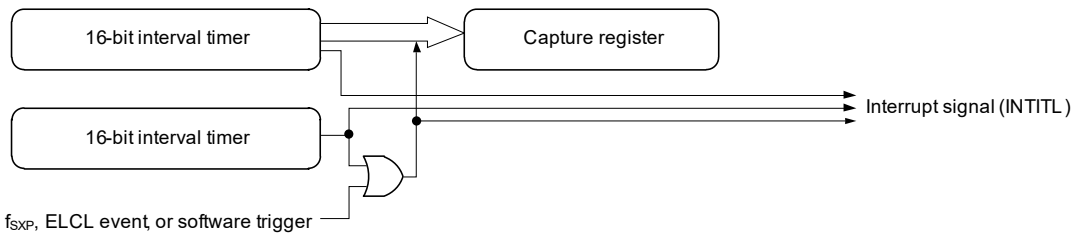
1. 8-bit counter mode
   The 8-bit timers can operate as four 8-bit interval timers that generate interrupts (INTITL) at fixed intervals.



2. 16-bit counter mode
   The 8-bit timers can operate as two 16-bit interval timers that generate interrupts (INTITL) at fixed intervals.



3. 16-bit capture mode
   The 8-bit timers can operate as two 16-bit interval timers that generate interrupts (INTITL) at fixed intervals. One 16-bit interval timer has the capture function, and the other can be used for generating a trigger. The value of one of the 16-bit interval timers can also be stored in the capture register in response to a selected capture trigger.



4. 32-bit counter mode
   The 8-bit timers can operate as a 32-bit interval timer that generates interrupts (INTITL) at fixed intervals.

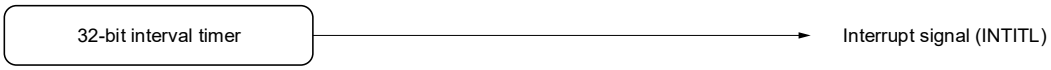


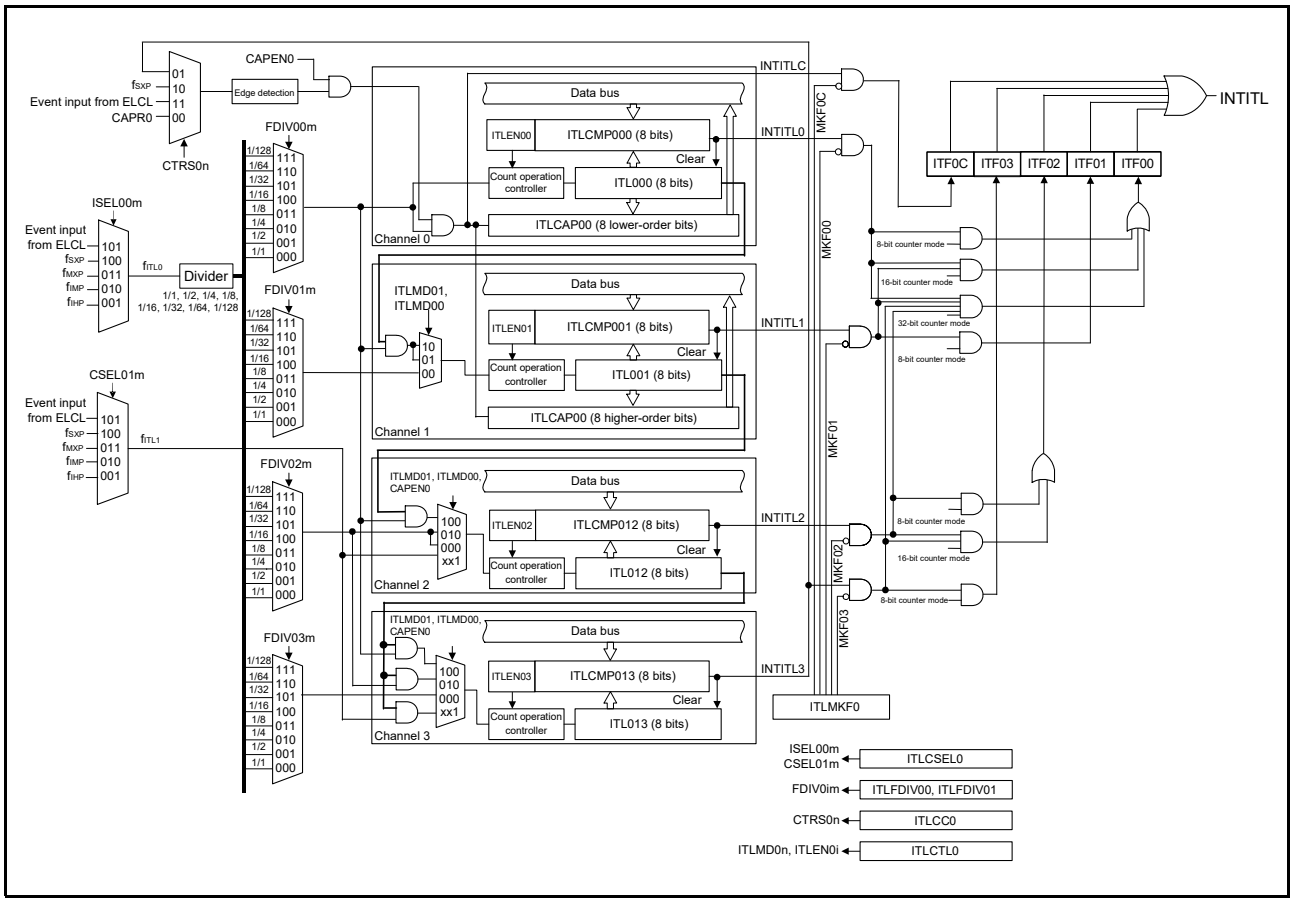
Table 9 - 1 lists the specifications of the 32-bit interval timer operations and **Figure 9 - 1** shows a block diagram of the 32-bit interval timer.

Table 9 - 1 Specifications of 32-bit Interval Timer

Item	Description
Count source (operating clock)	• f_{MXP} • f_{SXP} • f_{IHP} • f_{IMP} • Event input from the ELCL
Capture clock (Selectable sources for counting by the timer which can generate a capture trigger)	• f_{MXP} • f_{SXP} • f_{IHP} • f_{IMP} • Event input from the ELCL
Frequency division ratio	• 1/1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128
Operating mode	• 8-bit counter mode Channels 0 to 3 independently operate as 8-bit counters. • 16-bit counter mode The combinations of channels 0 and 1 and channels 2 and 3 are cascade-connectable to operate as two 16-bit counters. • 32-bit counter mode Channels 0 to 3 are connected to operate as a 32-bit counter. • 16-bit capture mode Channels 0 and 1 are connected to operate as a 16-bit counter using the count source, channels 2 and 3 are connected to operate as a 16-bit counter using the capture clock, and the connected counters are used for capture operation.
Interrupt	• Five interrupt sources are integrated and output as the single interrupt signal (INTITL). - Output when the counter value in any of channels 0 to 3 matches the compare value. - Output when the capturing of the counter value is completed in the capture mode.

Remark f_{MXP}: High-speed peripheral clock frequency
f_{SXP}: Low-speed peripheral clock frequency
f_{IHP}: High-speed on-chip oscillator peripheral clock frequency
f_{IMP}: Middle-speed on-chip oscillator peripheral clock frequency (4 MHz)
Event input from the ELCL: The ELCL function is used to generate this signal from a peripheral module such as a timer, serial communications interface, or port pin. For details, see **Section 20 Logic and Event Link Controller (ELCL)**.

Figure 9 - 1 Block Diagram of 32-bit Interval Timer



Remark 1. ITL000, ITL001, ITL012, ITL013: 8-bit counters

In the 16-bit counter mode, the counters in channels 0 and 1 are connected (ITL000 + ITL001) and the counters in channels 2 and 3 are connected (ITL012 + ITL013). In the 32-bit counter mode, the counters in channels 0 to 3 are connected (ITL000 + ITL001 + ITL012 + ITL013).

ISEL00m, CSEL01m: Bits in the ITLCSEL0 register
FDIV0im: Bits in the ITLFDIV0n register
CTRS0n: Bits in the ITLCC0 register
ITLMD0n, ITLEN0i: Bits in the ITLCTL0 register

Remark 2. n = 0, 1; m = 0 to 2; i = 0 to 3

9.2 Registers for Controlling the 32-bit Interval Timer

The following registers are used to control the 32-bit interval timer.

- Peripheral enable register 1 (PER1)
- Peripheral reset control register 1 (PRR1)
- Interval timer compare registers 0mn (ITLCMP0mn)
- Interval timer compare registers 0n (ITLCMP0n)
- Interval timer capture register 00 (ITLCAP00)
- Interval timer control register (ITLCTL0)
- Interval timer clock select register 0 (ITLCSEL0)
- Interval timer frequency division register 0 (ITLFDIV00)
- Interval timer frequency division register 1 (ITLFDIV01)
- Interval timer capture control register 0 (ITLCC0)
- Interval timer status register (ITLS0)
- Interval timer match detection mask register (ITLMKF0)

9.2.1 Peripheral enable register 1 (PER1)

This register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. If the 32-bit interval timer is to be used, be sure to set the TML32EN bit of this register to 1. The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 9 - 2 Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER1	DACEN	SMSEN	CMPEN	TML32EN	DTCEN	UTAEN **Note 1**	REMCEN **Note 2**	CTSUEN

TML32EN	Control of supply of an input clock to the 32-bit interval timer
0	Stops supply of an input clock. • The SFRs used by the 32-bit interval timer cannot be written.**Note 3**
1	Enables supply of an input clock. • The SFRs used by the 32-bit interval timer can be read and written.

Note 1. This bit is only present in the 36- to 128-pin R7F100G and 32- to 48-pin R7F104G products.

Note 2. This bit is only present in the 32- to 128-pin R7F100G and 32- to 48-pin R7F104G products.

Note 3. When an SFR used by the 32-bit interval timer is read, the value returned is 00H or 0000H regardless of the setting value.

Caution Be sure to leave the following bits set to 0.
Bits 2 and 1 in the 30-pin R7F100G products
Bit 2 in the 32-pin R7F100G products

9.2.2 Peripheral reset control register 1 (PRR1)

The PRR1 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. Set the TML32RES bit of this register to 1 to reset the 32-bit interval timer. The PRR1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 9 - 3 Format of Peripheral Reset Control Register 1 (PRR1)

Address: F00FBH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	3	2	<1>	<0>
PRR1	DACRES	SMSRES	CMPRES	TML32RES	0	0	REMCRES **Note**	CTSURES

TML32RES	Control resetting of the 32-bit interval timer
0	The 32-bit interval timer is released from the reset state.
1	The 32-bit interval timer is in the reset state. • The SFRs for use with the 32-bit interval timer are initialized.

Note This bit is only present in the 32- to 128-pin R7F100G and 32- to 48-pin R7F104G products.

Caution 1. Be sure to leave bits 3 and 2 set to 0.
Caution 2. Be sure to leave bit 1 in the 30-pin R7F100G products set to 0.

9.2.3 Interval timer compare registers 0mn (ITLCMP0mn)

These are compare value registers used in the 8-bit counter mode. These registers can be set by an 8-bit memory manipulation instruction. The value of each ITLCMP0mn register following a reset is FFH. A value from 01H to FFH can be specified. Setting of 00H is prohibited. These registers hold values to be compared with the ITL000 to ITL013 counter values.

Figure 9 - 4 Format of Interval Timer Compare Registers 0mn (ITLCMP0mn)

Address: F0360H (ITLCMP000), F0361H (ITLCMP001), F0362H (ITLCMP012), F0363H (ITLCMP013)
After reset: FFH
R/W: R/W**Note**

Symbol	7	6	5	4	3	2	1	0
ITLCMP0mn								

Note Write to the ITLCMP000 to ITLCMP013 registers while the settings of the ITLEN00 to ITLEN03 bits in the ITLCTL0 register are 0, respectively.

Remark mn = 00, 01, 12, 13

9.2.4 Interval timer compare registers 0n (ITLCMP0n)

These are compare value registers used in the 16-bit or 32-bit counter mode. These registers can be set by a 16-bit memory manipulation instruction. The value of each ITLCMP0n register following a reset is FFFFH. A value from 0001H to FFFFH can be specified. Setting of 0000H is prohibited. These registers hold values to be compared with the ITL0n counter values. When the setting of the ITLMD01 and ITLMD00 bits in the ITLCTL0 register is 10B, these registers are used as compare registers in the 32-bit counter mode; specify the upper 16-bit compare value in the ITLCMP01 register and the lower 16-bit compare value in the ITLCMP00 register.

Figure 9 - 5 Format of Interval Timer Compare Registers 0n (ITLCMP0n)

Address: F0360H, F0361H (ITLCMP00), F0362H, F0363H (ITLCMP01)
After reset: FFFFH
R/W: R/W**Note**

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ITLCMP0n																

Note Write to the ITLCMP00 register while the ITLEN00 bit in the ITLCTL0 register is 0. Write to the ITLCMP01 register while the ITLEN02 bit in the ITLCTL0 register is 0 in the 16-bit counter mode or while the ITLEN00 bit in the ITLCTL0 register is 0 in the 32-bit counter mode.

Remark n = 0, 1

9.2.5 Interval timer capture register 00 (ITLCAP00)

This register holds 16-bit captured values when the interval timers are operating in the 16-bit counter mode. The ITLCAP00 register can be read by a 16-bit manipulation instruction. The value of this register following a reset is 0000H. The value of the 16-bit counter (ITL000 + ITL001) is stored in the ITLCAP00 register in response to the capture trigger selected in the ITLCC0 register when the CAPEN0 bit in the ITLCC0 register is 1. When an interrupt on compare match with the ITLCMP01 register is to be used, select the counter clock in the ITLCSEL0 register and set the comparison value in the ITLCMP01 register.

Figure 9 - 6 Format of Interval Timer Capture Register 00 (ITLCAP00)

Address: F0364H, F0365H
After reset: 0000H
R/W: R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ITLCAP00																

9.2.6 Interval timer control register (ITLCTL0)

This register is used to start or stop counting by the interval timer and to select the 8-bit, 16-bit, or 32-bit counter mode.
This register can be set by a 1-bit or 8-bit manipulation instruction. The value of this register following a reset is 00H.

Figure 9 - 7 Format of Interval Timer Control Register (ITLCTL0) (1/2)

Address: F0366H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	<3>	<2>	<1>	<0>
ITLCTL0	ITLMD01	ITLMD00	0	0	ITLEN03	ITLEN02	ITLEN01	ITLEN00

ITLMD01	ITLMD00	Selection of 8-bit, 16-bit, or 32-bit counter mode[Note 1]
0	0	The interval timer operates in the 8-bit counter mode.
0	1	The interval timer operates in the 16-bit counter mode (channel 0 is connected with channel 1 and channel 2 is connected with channel 3).
1	0	The interval timer operates in the 32-bit counter mode (channels 0 to 3 are connected).
1	1	Setting prohibited

ITLEN03	8-bit counter mode: ITL013 count enable[Note 2]
0	Counting stops.
1	Counting begins.
In the 8-bit counter mode, writing 1 to this bit starts up-counting in the ITL013 counter and writing 0 stops it. In the 16-bit counter mode, set this bit to 0. In the 32-bit counter mode, set this bit to 0.	

ITLEN02	8-bit counter mode: ITL012 count enable[Note 2] 16-bit counter mode: ITL012 + ITL013 count enable[Note 2]
0	Counting stops.
1	Counting begins.
In the 8-bit counter mode, writing 1 to this bit starts up-counting in the ITL012 counter and writing 0 stops it. In the 16-bit counter mode, writing 1 to this bit starts up-counting in the ITL012 + ITL013 counter and writing 0 stops it. In the 32-bit counter mode, set this bit to 0.	

ITLEN01	8-bit counter mode: ITL001 count enable[Note 2]
0	Counting stops.
1	Counting begins.
In the 8-bit counter mode, writing 1 to this bit starts up-counting in the ITL001 counter and writing 0 stops it. In the 16-bit counter mode, set this bit to 0. In the 32-bit counter mode, set this bit to 0.	

Figure 9 - 7 Format of Interval Timer Control Register (ITLCTL0) (2/2)

ITLEN00	8-bit counter mode: ITL000 count enable[Note 2] 16-bit counter mode: ITL000 + ITL001 count enable[Note 2] 32-bit counter mode: ITL000 + ITL001 + ITL012 + ITL013 count enable[Note 2]
0	Counting stops.
1	Counting begins.
In the 8-bit counter mode, writing 1 to this bit starts up-counting in the ITL000 counter and writing 0 stops it. In the 16-bit counter mode, writing 1 to this bit starts up-counting in the ITL000 + ITL001 counter and writing 0 stops it. In the 32-bit counter mode, writing 1 to this bit starts up-counting in the ITL000 + ITL001 + ITL012 + ITL013 counter and writing 0 stops it.	

Note 1. To change the counter mode specified in the ITLMD01 and ITLMD00 bits, be sure to only write to the given bits while the ITLEN03 to ITLEN00 bits are all 0.

Note 2. When any of the ITLEN03 to ITLEN00 bits is set to 0, the corresponding counter is cleared to 0 without synchronization with the counter clock.

Mode	ITLMD01	ITLMD00	ITLEN03	ITLEN02	ITLEN01	ITLEN00	Target Counter
8-bit counter mode[Note 1]	0	0				✓	ITL000
					✓		ITL001
				✓			ITL012
			✓				ITL013
16-bit counter mode[Note 2]	0	1	Always set to 0.		Always set to 0.	✓	ITL000+ITL001
			Always set to 0.	✓	Always set to 0.		ITL012+ITL013
32-bit counter mode	1	0	Always set to 0.	Always set to 0.	Always set to 0.	✓	ITL000+ITL001+ITL012+ITL013

Note 1. Two or more bits of ITLEN03 to ITLEN00 can be set to 1 or 0 at the same time.

Note 2. The ITLEN02 and ITLEN00 bits can be set to 1 or 0 at the same time.

Remark ✓: Enables counting in the target counter.

9.2.7 Interval timer clock select register 0 (ITLCSEL0)

This register is used to select the count source for the interval timer. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 9 - 8 Format of Interval Timer Clock Select Register 0 (ITLCSEL0)

Address: F0367H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ITLCSEL0	0	CSEL012	CSEL011	CSEL010	0	ISEL002	ISEL001	ISEL000

CSEL012	CSEL011	CSEL010	Selection of interval timer counter clock for capturing (f_{ITL1})Note
0	0	0	Counting stops.
0	0	1	f_{IHP}
0	1	0	f_{IMP}
0	1	1	f_{MXP}
1	0	0	f_{SXP}
1	0	1	Event input from the ELCL
Others			Setting prohibited

ISEL002	ISEL001	ISEL000	Selection of interval timer counter clock (f_{ITL0})Note
0	0	0	Counting stops.
0	0	1	f_{IHP}
0	1	0	f_{IMP}
0	1	1	f_{MXP}
1	0	0	f_{SXP}
1	0	1	Event input from the ELCL
Others			Setting prohibited

Note Be sure to only write to the CSEL012 to CSEL010 bits and ISEL002 to ISEL000 bits while the ITLEN03 to ITLEN00 bits in the ITLCTL0 register are all 0.

Remark f_{MXP}: High-speed peripheral clock frequency
f_{SXP}: Low-speed peripheral clock frequency
f_{IHP}: High-speed on-chip oscillator peripheral clock frequency
f_{IMP}: Middle-speed on-chip oscillator peripheral clock frequency
Event input from the ELCL: The ELCL function is used to generate this signal from a peripheral module such as a timer, serial communications interface, or port pin. For details, see **Section 20 Logic and Event Link Controller (ELCL)**.

9.2.8 Interval timer frequency division register 0 (ITLFDIV00)

This register is used to select the counter clock for the interval timer. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 9 - 9 Format of Interval Timer Frequency Division Register 0 (ITLFDIV00)

Address: F0368H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ITLFDIV00	0	FDIV012	FDIV011	FDIV010	0	FDIV002	FDIV001	FDIV000

FDIV012	FDIV011	FDIV010	8-bit counter mode: Counter clock for ITL001Note 1
0	0	0	f_{ITL0}
0	0	1	$f_{ITL0}/2$
0	1	0	$f_{ITL0}/4$
0	1	1	$f_{ITL0}/8$
1	0	0	$f_{ITL0}/16$
1	0	1	$f_{ITL0}/32$
1	1	0	$f_{ITL0}/64$
1	1	1	$f_{ITL0}/128$
In the 8-bit counter mode, ITL001 counts cycles of the counter clock specified in the FDIV012 to FDIV010 bits. In the 16-bit counter mode, set these bits to 000B. In the 32-bit counter mode, set these bits to 000B.			

FDIV002	FDIV001	FDIV000	8-bit counter mode: Counter clock for ITL000Note 2 16-bit counter mode: Counter clock for ITL000 + ITL001Note 2 32-bit counter mode: Counter clock for ITL000 + ITL001 + ITL012 + ITL013Note 2
0	0	0	f_{ITL0}
0	0	1	$f_{ITL0}/2$
0	1	0	$f_{ITL0}/4$
0	1	1	$f_{ITL0}/8$
1	0	0	$f_{ITL0}/16$
1	0	1	$f_{ITL0}/32$
1	1	0	$f_{ITL0}/64$
1	1	1	$f_{ITL0}/128$
In the 8-bit counter mode, ITL000 counts cycles of the counter clock specified in the FDIV002 to FDIV000 bits. In the 16-bit counter mode, ITL000 + ITL001 counts cycles of the counter clock specified in the FDIV002 to FDIV000 bits. In the 32-bit counter mode, ITL000 +ITL001 + ITL012 + ITL013 counts cycles of the counter clock specified in the FDIV002 to FDIV000 bits.			

Note 1. In the 8-bit counter mode, be sure to only write to the FDIV012 to FDIV010 bits while the ITLEN01 bit in the ITLCTL0 register is 0.

Note 2. Be sure to only write to the FDIV002 to FDIV000 bits while the ITLEN00 bit in the ITLCTL0 register is 0.

9.2.9 Interval timer frequency division register 1 (ITLFDIV01)

This register is used to select the counter clock for the interval timer. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 9 - 10 Format of Interval Timer Frequency Division Register 1 (ITLFDIV01)

Address: F0369H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ITLFDIV01	0	FDIV032	FDIV031	FDIV030	0	FDIV022	FDIV021	FDIV020

FDIV032	FDIV031	FDIV030	8-bit counter mode: Counter clock for ITL013Note 1
0	0	0	f_{ITL0}
0	0	1	$f_{ITL0}/2$
0	1	0	$f_{ITL0}/4$
0	1	1	$f_{ITL0}/8$
1	0	0	$f_{ITL0}/16$
1	0	1	$f_{ITL0}/32$
1	1	0	$f_{ITL0}/64$
1	1	1	$f_{ITL0}/128$
In the 8-bit counter mode, ITL013 counts cycles of the counter clock specified in the FDIV032 to FDIV030 bits. In the 16-bit counter mode, set these bits to 000B. In the 32-bit counter mode, set these bits to 000B.			

FDIV022	FDIV021	FDIV020	8-bit counter mode: Counter clock for ITL012Note 2 16-bit counter mode: Counter clock for ITL012 and ITL013Note 2
0	0	0	f_{ITL0}
0	0	1	$f_{ITL0}/2$
0	1	0	$f_{ITL0}/4$
0	1	1	$f_{ITL0}/8$
1	0	0	$f_{ITL0}/16$
1	0	1	$f_{ITL0}/32$
1	1	0	$f_{ITL0}/64$
1	1	1	$f_{ITL0}/128$
In the 8-bit counter mode, ITL012 counts cycles of the counter clock specified in the FDIV022 to FDIV020 bits. In the 16-bit counter mode, ITL012 + ITL013 counts cycles of the counter clock specified in the FDIV022 to FDIV020 bits. In the 32-bit counter mode, these bits are not used; write 000B to them.			

Note 1. In the 8-bit counter mode, be sure to only write to the FDIV032 to FDIV030 bits while the ITLEN03 bit in the ITLCTL0 register is 0.

Note 2. In the 8-bit or 16-bit counter mode, be sure to only write to the FDIV022 to FDIV020 bits while the ITLEN02 bit in the ITLCTL0 register is 0.

9.2.10 Interval timer capture control register 0 (ITLCC0)

This register is used to enable or disable the capture function of the interval timer, specify whether to hold or clear the capture completion flag, set up the software trigger, and select the capture trigger. This register can be set by a 1-bit or 8-bit manipulation instruction. The value of this register following a reset is 00H.

Figure 9 - 11 Format of Interval Timer Capture Control Register 0 (ITLCC0)

Address: F036AH
After reset: 00H
R/W: R/W

Symbol	7	<6>	5	<4>	3	2	1	0
ITLCC0	CAPEN0	CAPF0CR	CAPF0	CAPR0	CAPC0CR	0	CTRS01	CTRS00

CAPEN0	Capture enable[Note 1]
0	Capturing is disabled.
1	Capturing is enabled.

CAPF0CR	Capture completion flag clear[Note 2]
0	The value of the capture completion flag CAPF0 is held.
1	The value of the capture completion flag CAPF0 is cleared.

CAPF0	Capture completion flag[Note 3]
0	Capturing has not been completed.
1	Capturing has been completed. This flag is set to 1 after generation of the capture trigger selected in the CTRS01 and CTRS00 bits and capture of the data in the ITLCAP00 register. Writing 1 to the CAPF0CR bit clears this flag to 0.

CAPR0	Software capture trigger[Notes 4, 5]
0	Trigger operation does not proceed.
1	A software trigger for capturing is generated.

CAPC0CR	Selection of the 16-bit counter (ITL000 + ITL001) clearing after capturing[Note 6]
0	The 16-bit counter (ITL000 + ITL001) is retained after the completion of capturing.
1	The 16-bit counter (ITL000 + ITL001) is cleared after the completion of capturing.

CTRS01	CTRS00	Selection of capture trigger[Notes 5, 7]
0	0	Software trigger
0	1	Interrupt on compare match with the ITLCMP01 register[Note 8]
1	0	f_{SXP} (rising edge)
1	1	Event input from ELCL (rising edge)

(Notes are listed on the next page.)

Note 1. Be sure to only write to the CAPEN0 bit while the ITLEN03 to ITLEN00 bits in the ITLCTL0 register are all 0.

Note 2. The CAPF0CR bit is always read as 0.

Note 3. Only bit 5 is readable.

Note 4. The CAPR0 bit is always read as 0.

Note 5. In the capture operation, the interval at which the capture trigger is generated should be two or more cycles of the counter clock.

Note 6. Be sure to only write to the CAPC0CR bit while the ITLEN03 to ITLEN00 bits in the ITLCTL0 register are all 0.

Note 7. Be sure to only write to the CTRS01 and CTRS00 bits while the ITLEN03 to ITLEN00 bits in the ITLCTL0 register are all 0.

Note 8. When the interrupt on compare match with ITLCMP01 is selected as a capture trigger, the compare match detection flag for channel 2 (ITF02) and capture detection flag (ITF0C) are set on capture of the counter value. When only using the capture detection flag, set the ITLMKF0 register to mask the compare match detection flag for channel 2.

9.2.11 Interval timer status register (ITLS0)

This is a status register for the interval timer. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H. When the value of the ITL0mn register (mn = 00, 01, 12, 13) matches the value specified in the ITLCMP0mn, ITLCMP00, and ITLCMP01 registers, the compare-match flag for the corresponding channel is set. When a capture trigger is generated while the CAPEN0 bit in the ITLCC0 register is 1, the capture detection flag is set after the value of the ITL0n counter is stored in the ITLCAP00 register. The values of the ITF0C and ITF03 to ITF00 bits in this register are ORed and output as the interrupt request signal (INTITL). **Table 9 - 2** shows the conditions for setting the status flags in each counter mode selected by the ITLMD01 and ITLMD00 bits.

Figure 9 - 12 Format of Interval Timer Status Register (ITLS0)

Address: F036BH
After reset: 00H
R/W: R/W**Note**

Symbol	7	6	5	4	3	2	1	0
ITLS0	0	0	0	ITF0C	ITF03	ITF02	ITF01	ITF00

ITF0C	Capture detection flag
0	Completion of capturing has not been detected.
1	Completion of capturing has been detected.

ITF03	Compare match detection flag for channel 3
0	A compare match signal has not been detected in channel 3.
1	A compare match signal has been detected in channel 3.

ITF02	Compare match detection flag for channel 2
0	A compare match signal has not been detected in channel 2.
1	A compare match signal has been detected in channel 2.

ITF01	Compare match detection flag for channel 1
0	A compare match signal has not been detected in channel 1.
1	A compare match signal has been detected in channel 1.

ITF00	Compare match detection flag for channel 0
0	A compare match signal has not been detected in channel 0.
1	A compare match signal has been detected in channel 0.

Note Writing 1 to each bit is ignored. To clear the ITF0C or ITF0i bit (i = 0 to 3), write 0 to the desired bit and 1 to the other bits by using an 8-bit memory manipulation instruction.

Caution 1. If clearing any of the ITF0C, ITF03, ITF02, ITF01, ITF00 flag bits to 0 does not lead to the value of the ITLS0 register becoming 00H, an interrupt request (INTITL) is generated and the interrupt request flag (ITLIF) is set to 1.

Caution 2. To clear a flag bit in the ITLS0 register to 0, only write 0 to a bit that has the setting 1. This is because writing 0 to a bit that has the setting 0 may make detecting a compare match signal or capture detection signal generated at the same time as the writing of 0 impossible. For example, when the ITF01 flag bit is set to 1, write 00011101B to the ITLS0 register to clear the ITF01 flag bit.

Table 9 - 2 Conditions for Setting the Status Flags in Each Counter Mode

Mode	ITLMD01	ITLMD00	CAPEN0	Status Flag	Conditions for Setting Status Flag
8-bit counter mode	0	0	×	ITF00	The next rising edge of the counter clock following a match between the ITLCMP000 and ITL000 values
			×	ITF01	The next rising edge of the counter clock following a match between the ITLCMP001 and ITL001 values
			×	ITF02	The next rising edge of the counter clock following a match between the ITLCMP012 and ITL012 values
			×	ITF03	The next rising edge of the counter clock following a match between the ITLCMP013 and ITL013 values
16-bit counter mode	0	1	×	ITF00	The next rising edge of the counter clock following a match between the ITLCMP00 and ITL000 + ITL001 values
			×	ITF02	The next rising edge of the counter clock following a match between the ITLCMP01 and ITL012 + ITL013 values
			1	ITF0C	The ITL000 + ITL001 value is stored in ITLCAP00 after a capture trigger is generated.
32-bit counter mode	1	0	×	ITF00	The next rising edge of the counter clock following a match between the ITLCMP00 + ITLCMP01 and ITL000 + ITL001 + ITL012 + ITL013 values

Remark ×: Don’t care

9.2.12 Interval timer match detection mask register (ITLMKF0)

This register is used to enable or disable setting of each effective bit in the interval timer status register (ITLS0) to 1. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H. Setting an MKF0C or MKF03 to MKF00 bit to 1 masks the corresponding status flag (ITF0C, ITF03 to ITF00), after which the given flag is not set to 1 even if a compare match with a compare register or capture completion is detected. Since the status flag will not be set to 1, masking also prevents generation of the interval detection interrupt (INTITL).

Figure 9 - 13 Format of Interval Timer Match Detection Mask Register (ITLMKF0)

Address: F036CH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ITLMKF0	0	0	0	MKF0C	MKF03	MKF02	MKF01	MKF00

MKF0C	Mask for capture detection status flag[Note]
0	The ITF0C flag is not masked.
1	The ITF0C flag is masked.

MKF03	Mask for compare match status flag for channel 3[Note]
0	The ITF03 flag is not masked.
1	The ITF03 flag is masked.

MKF02	Mask for compare match status flag for channel 2[Note]
0	The ITF02 flag is not masked.
1	The ITF02 flag is masked.

MKF01	Mask for compare match status flag for channel 1[Note]
0	The ITF01 flag is not masked.
1	The ITF01 flag is masked.

MKF00	Mask for compare match status flag for channel 0[Note]
0	The ITF00 flag is not masked.
1	The ITF00 flag is masked.

Note Setting all functional bits to 1 for masking prevents setting of the corresponding bits in the ITLS0 register. This in turn prevents software detection of compare matches and completion of capture. When compare match for any of channels 0 to 3 is to be used, be sure to set the bit corresponding to the given status flag to 0 so that the flag is not masked. For the state of completion of capture, on the other hand, the CAPF0 flag in the interval timer capture control register 0 (ITLCC0) can be used to detect this even when the MKF0C bit is set to 1 to mask the ITF0C flag.

9.3 Operation

9.3.1 Counter mode settings

The 32-bit interval timer has three counter modes: the 8-bit counter mode, 16-bit counter mode, and 32-bit counter mode. **Table 9 - 3** to **Table 9 - 5** show the registers and settings for use in the 8-bit counter mode, 16-bit counter mode, and 32-bit counter mode, respectively.

Table 9 - 3 Registers and Settings Used in the 8-bit Counter Mode

Register	Bit	Setting
Interval timer compare registers 0mn (ITLCMP0mn)	Bits 7 to 0	Specify 8-bit compare values for channels 0 to 3.
Interval timer control register 0 (ITLCTL0)	ITLEN00	Specify whether to start or stop counting in channel 0.
	ITLEN01	Specify whether to start or stop counting in channel 1.
	ITLEN02	Specify whether to start or stop counting in channel 2.
	ITLEN03	Specify whether to start or stop counting in channel 3.
	ITLMD00	Set to 0.
	ITLMD01	Set to 0.
Interval timer frequency division registers n (ITLFDIV0n)	FDIV002 to FDIV000	Select the counter clock for channel 0.
	FDIV012 to FDIV010	Select the counter clock for channel 1.
	FDIV022 to FDIV020	Select the counter clock for channel 2.
	FDIV032 to FDIV030	Select the counter clock for channel 3.
Interval timer clock select register 0 (ITLCSEL0)	ISEL002 to ISEL000	Select the counter clock for the interval timer.
	CSEL012 to CSEL010	Set to 000B.
Interval timer capture control register 0 (ITLCC0)	Bits 7 to 0	Set to 0.

Remark mn = 00, 01, 12, 13; n = 0, 1

Table 9 - 4 Registers and Settings Used in the 16-bit Counter Mode

Register	Bit	Setting
Interval timer compare registers 0n (ITLCMP0n)	Bits 15 to 0	Specify 16-bit compare values for channels 0 and 1, and channels 2 and 3.
Interval timer control register 0 (ITLCTL0)	ITLEN00	Specify whether to start or stop counting in channels 0 and 1.
	ITLEN01	Set to 0.
	ITLEN02	Specify whether to start or stop counting in channels 2 and 3.
	ITLEN03	Set to 0.
	ITLMD00	Set to 1.
	ITLMD01	Set to 0.
Interval timer frequency division registers n (ITLFDIV0n)	FDIV002 to FDIV000	Select the counter clock for channels 0 and 1.
	FDIV012 to FDIV010	Set to 000B.
	FDIV022 to FDIV020	Select the counter clock for channels 2 and 3.
	FDIV032 to FDIV030	Set to 000B.
Interval timer clock select register 0 (ITLCSEL0)	ISEL002 to ISEL000	Select the counter clock for the interval timer.
	CSEL012 to CSEL010	Set to 000B.
Interval timer capture control register 0 (ITLCC0)	Bits 7 to 0	Set to 0.

Remark n = 0, 1

Table 9 - 5 Registers and Settings Used in the 32-bit Counter Mode

Register	Bit	Setting
Interval timer compare registers 0n (ITLCMP0n)	Bits 15 to 0	Specify a compare value in the 32-bit counter mode. Specify the lower 16 bits of the compare value in channels 0 and 1 (ITLCMP00) and the upper 16 bits of the compare value in channels 2 and 3 (ITLCMP01).
Interval timer control register 0 (ITLCTL0)	ITLEN00	Specify whether to start or stop counting in channels 0 to 3.
	ITLEN01	Set to 0.
	ITLEN02	Set to 0.
	ITLEN03	Set to 0.
	ITLMD00	Set to 0.
	ITLMD01	Set to 1.
Interval timer frequency division registers n (ITLFDIV0n)	FDIV002 to FDIV000	Select the counter clock for channels 0 to 3.
	FDIV012 to FDIV010	Set to 000B.
	FDIV022 to FDIV020	Set to 000B.
	FDIV032 to FDIV030	Set to 000B.
Interval timer clock select register 0 (ITLCSEL0)	ISEL002 to ISEL000	Select the counter clock for the interval timer.
	CSEL012 to CSEL010	Set to 000B.
Interval timer capture control register 0 (ITLCC0)	Bits 7 to 0	Set to 0.

Remark n = 0, 1

9.3.2 Capture mode settings

When the 16-bit capture mode is to be used for channels 0 and 1, the counter value is stored in interval timer capture register 00 (ITLCAP00) in response to a selected capture trigger.

Table 9 - 6 Registers and Settings Used in the 16-bit Capture Mode

Register	Bit	Setting
Interval timer compare register 00 (ITLCMP00)	Bits 15 to 0	Specify 16-bit compare values for channels 0 and 1.
Interval timer compare register 01 (ITLCMP01)**Note**	Bits 15 to 0	Specify 16-bit compare values for channels 2 and 3.
Interval timer control register 0 (ITLCTL0)	ITLEN00	Specify whether to start or stop counting in channels 0 and 1.
	ITLEN01	Set to 0.
	ITLEN02	Specify whether to start or stop counting in channels 2 and 3.
	ITLEN03	Set to 0.
	ITLMD00	Set to 1.
	ITLMD01	Set to 0.
Interval timer frequency division registers n (ITLFDIV0n)	FDIV002 to FDIV000	Select the counter clock for channel 0.
	FDIV012 to FDIV010	Set to 000B.
	FDIV022 to FDIV020	Set to 000B.
	FDIV032 to FDIV030	Set to 000B.
Interval timer clock select register 0 (ITLCSEL0)	ISEL002 to ISEL000	Select the counter clock for the interval timer in channels 0 and 1.
	CSEL012 to CSEL010	Select the counter clock for the interval timer for capturing in channels 2 and 3.
Interval timer capture control register 0 (ITLCC0)	CAPEN0	Set to 1.
	CAPC0CR	Specify whether to clear or hold the counter value in channels 0 and 1 after the completion of capturing.
	CTRS01 and CTRS00	Select a capture trigger.

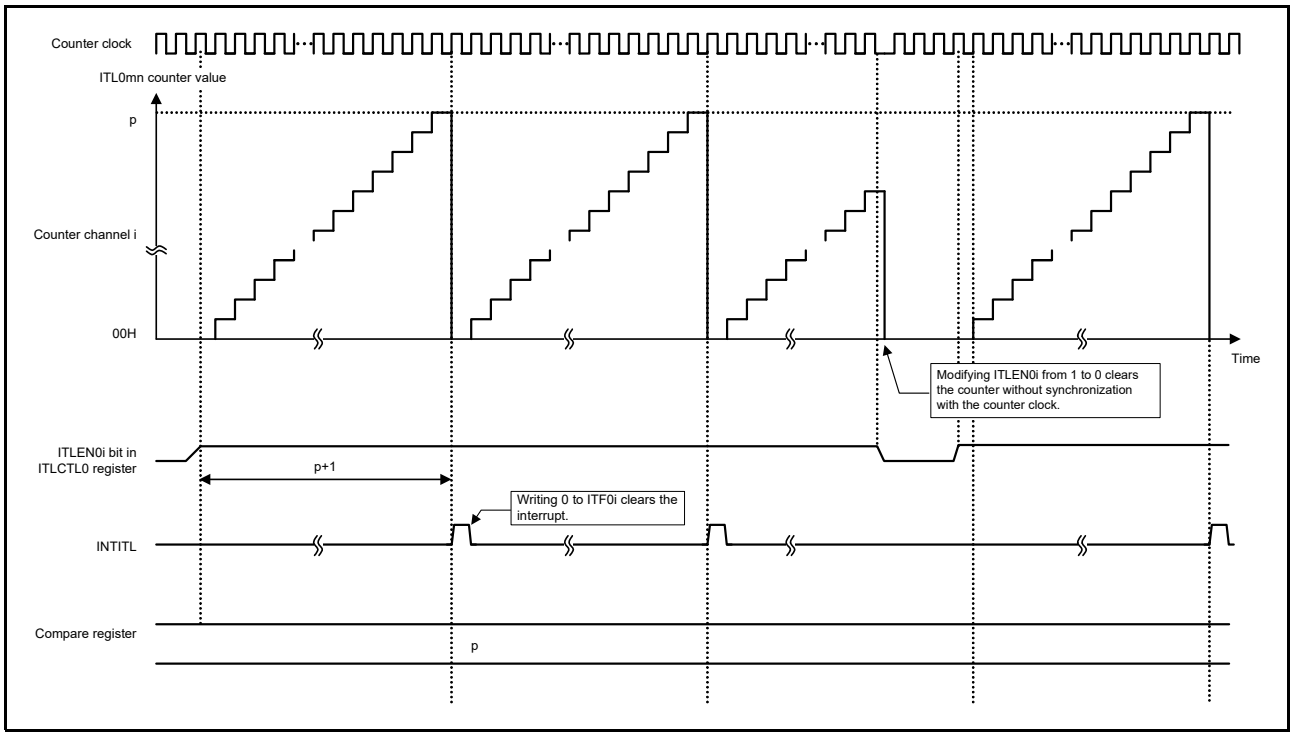
Note Channels 2 and 3 can only be used in the 16-bit counter mode when an interrupt on compare match with ITLCMP01 is not to be used as a capture trigger.

Remark n = 0, 1

9.3.3 Timer operation

The ITL0mn counter counts up cycles of the counting clock specified in the interval timer frequency division registers 0 and 1 (ITLFDIV00 and ITLFDIV01). An interrupt request signal (INTITL) is generated on the counting of the next clock cycle after the value of the counter matches the comparison value. The interrupt request signal (INTITL) remains high until the value of the ITLS0 register becomes 00H. While the interrupt request signal (INTITL) is high, neither the generation of a further interrupt request (INTITL) nor setting of the interrupt request flag (ITLIF) proceeds even if a compare match or capture completion is detected for an operating channel. Setting the ITLEN03 to ITLEN00 bits to 0 clears the counter value.

Figure 9 - 14 Example of Timer Operation in the 8-bit Counter Mode



Remark 1. mn = 00, 01, 12, 13; i = 0 to 3

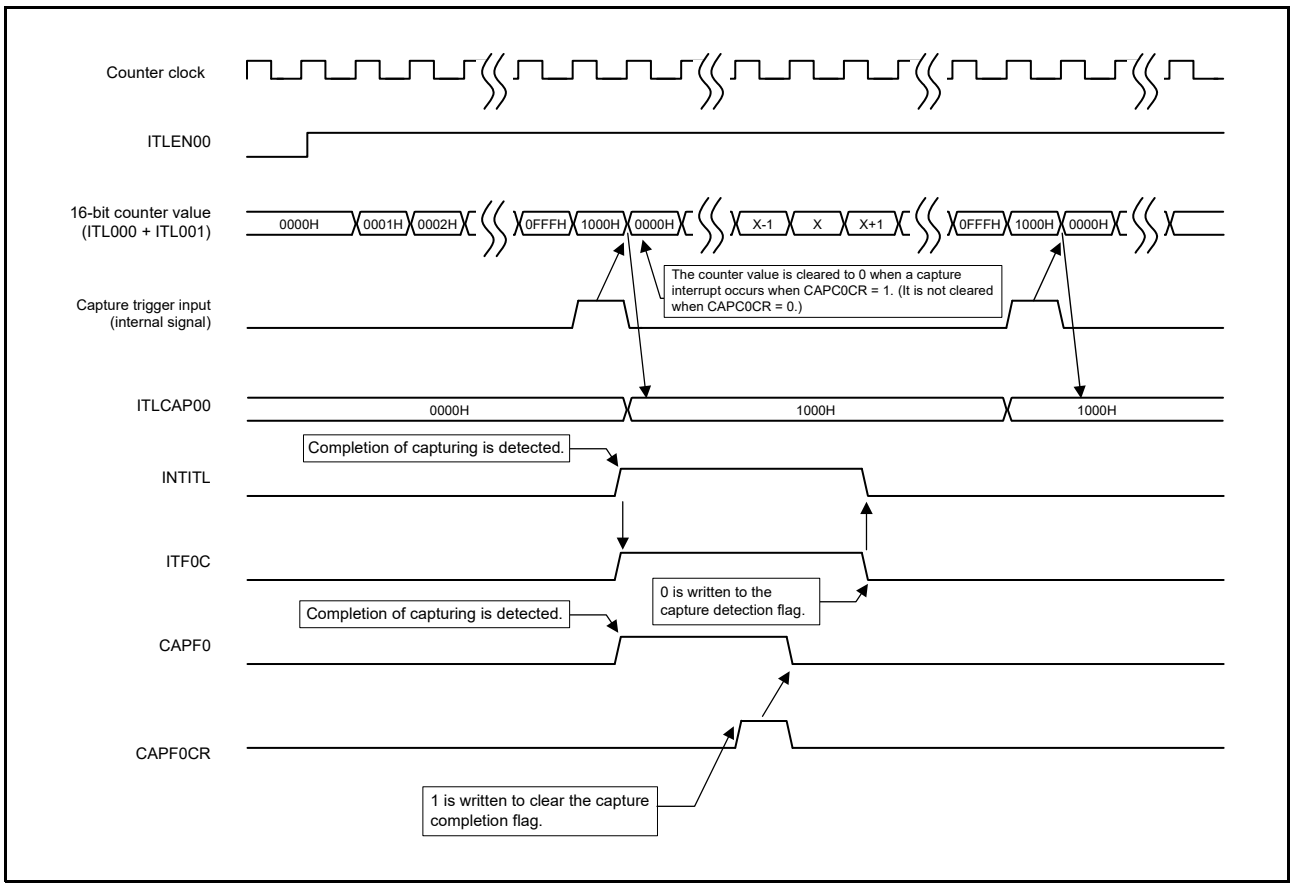
Remark 2. p: Set value of the ITLCMP0mn register

9.3.4 Capture operation

When the setting of the CAPEN0 bit in the interval timer capture control register 0 (ITLCC0) is 1, the value of the 16-bit counter (ITL000 + ITL001) is stored in interval timer capture register 00 (ITLCAP00) in response to the capture trigger specified in the ITLCC0 register. The capture trigger is selectable from among the interrupt on compare match with the ITLCMP01 register, fSXP, an event signal input from the ELCL, and a software trigger (setting the CAPR0 bit to 1). To use the interrupt on compare match with the ITLCMP01 register as the capture trigger, set interval timer clock select register 0 (ITLCSEL0) to select the clock for counting, and set interval timer compare register 01 (ITLCMP01) to specify the comparison value. When using fSXP, an event signal input from the ELCL, or a software trigger (setting the CAPR0 bit to 1) as a capture trigger, channels 2 and 3 can be used in the 16-bit counter mode. After a capture trigger is input and the counter value is stored in interval timer capture register 00, the interrupt request signal (INTITL) is output, the capture completion flag (CAPF0) and capture detection flag (ITF0C) are set to 1, and the flag values are retained until they are explicitly cleared **Note**. The CAPF0 flag can be cleared by setting the CAPF0CR bit to 1. The ITF0C flag in the ITLS0 register can be cleared by writing 0 to it. Since capture operations operate with the counter clock, the interval at which the capture trigger is generated should be at least five cycles of the counter clock. If a capture trigger is generated again within two cycles of the counter clock after an earlier capture trigger was generated, the CAPF0 bit may not be set.

Note If the value of the ITLS0 register is other than 00H, neither interrupt operation nor setting of the interrupt request flag (ITLIF) will proceed even when the ITF0C flag is set to 1 because the interrupt request signal (INTITL) is kept at the high level.

Figure 9 - 15 Example of Capture Operation



When the counter value matches the comparison value while the CAPC0CR bit in the ITLCC0 register is set to 1 (the mode where the 16-bit counter (ITL000 + ITL001) is cleared following the completion of capture), counting of the next clock cycle clears the counter value. Note that the ITF00 flag is set when the 16-bit counter (ITL000 + ITL001) matches the comparison value before a capture trigger is input. The counter value is not cleared in this way if the CAPC0CR bit is set to 0 (the mode where the 16-bit counter (ITL000 + ITL001) retains its value after the completion of capture). The ITF00 flag is set when the 16-bit counter (ITL000 + ITL001) matches the comparison value.

9.3.5 Interrupt

Table 9 - 7 shows the interrupt sources in the 8-bit, 16-bit, and 32-bit counter modes. The ITF0C and ITF03 to ITF00 bits are interrupt status flags in the ITLS0 register. When any of the interrupt status flag is set, an interrupt request signal (INTITL) is output.

Table 9 - 7 Interrupt Sources in the 8-bit, 16-bit, and 32-bit Counter Modes

Interrupt Source	Interrupt Condition in the 8-bit Counter Mode	Interrupt Condition in the 16-bit Counter Mode	Interrupt Condition in the 32-bit Counter Mode
ITF00	Next rising edge of the counter clock after a compare match in channel 0	Next rising edge of the counter clock after a compare match in channels 0 and 1	Next rising edge of the counter clock after a compare match
ITF01	Next rising edge of the counter clock after a compare match in channel 1	Not generated	Not generated
ITF02	Next rising edge of the counter clock after a compare match in channel 2	Next rising edge of the counter clock after a compare match in channels 2 and 3	Not generated
ITF03	Next rising edge of the counter clock after a compare match in channel 3	Not generated	Not generated
ITF0C	Not generated; this is the case when the setting of ITLCC0 is 00H.	Timing of storing the counter value in the capture register after a capture trigger is input	Not generated; this is the case when the setting of ITLCC0 is 00H.

If the value of the ITLS0 register is other than 00H, the interrupt request signal (INTITL) is kept at the high level. Accordingly, neither the generation of a further interrupt request (INTITL) nor setting of the interrupt request flag (ITLIF) will proceed, even when a compare match or completion of capture is detected for an operating channel. However, if the value of the ITLS0 register is not 00H after any bit in the ITLS0 register is set to 0 by an 8-bit memory manipulation instruction, a low-level pulse signal is output on the INTITL pin and the interrupt request flag (ITLIF) is set to 1. Accordingly, clearing a status flag in the ITLS0 register to 0 while ITLIF = 0 during vector interrupt processing or other processing enables the detection of an interrupt in response to another status bit having the setting 1. **Figure 9 - 16** shows the relationship between clearing of the detection flags and the interval detection interrupt signal.

The following describes the operation shown in **Figure 9 - 16**. When a compare match in channel 1 is detected while the value of the ITLS0 register is 00H, the ITF01 flag is set to 1 and the interval detection interrupt signal (INTITL) is driven high. While the interval detection interrupt signal (INTITL) is kept at the high level, neither the generation of a further interrupt request (INTITL) nor setting of the interrupt request flag (ITLIF) will proceed, even when a compare match or completion of capture is detected for an operating channel. Note that if another detection flag is set to 1 immediately before clearing the ITF0x (x = 0 to 3, C) flag bit to 0, the output of the INTITL pin temporarily goes to the low level after clearing of the given ITF0x flag bit, leading to setting of the interrupt request flag (ITLIF) to 1.

<1> The ITF01 flag is set to 1 in response to a compare match in channel 1 and the interval detection interrupt signal (INTITL) and interrupt request flag (ITLIF) are driven high. The interval detection interrupt processing is executed after clearing of the interrupt request flag (ITLIF) to 0.
<2> Check which detection flag in the ITLS0 register is set to 1 from within the interval detection interrupt processing. In the case shown in **Figure 9 - 16**, the ITF01 and ITF00 flags being set to 1 can be confirmed.
<3> Clear the ITF01 and ITF00 flags detected in step 2 by using an 8-bit memory manipulation instruction to write 00011100B to the ITLS0 register so that its value becomes 00H.**Note**
<4> The respective processing sequences in response to the ITF01 and ITF00 flags being set to 1 are then executed.**Note**

<5> The ITF01 flag is set to 1 in response to a further compare match in channel 1 and the interval detection interrupt signal (INTITL) and interrupt request flag (ITLIF) are driven high. The interval detection interrupt processing is executed after clearing of the interrupt request flag (ITLIF) to 0.

<6> Check which detection flag in the ITLS0 register is set to 1 from within the interval detection interrupt processing. In the case shown in **Figure 9 - 16**, the ITF01 flag being set to 1 can be confirmed.

<7> Clear the ITF01 flag detected in step 6 by using an 8-bit memory manipulation instruction to write 00011101B to the ITLS0 register so that its value becomes 00H. Though the ITF00 flag is also set to 1 in response to the compare match in channel 0 at this time, the ITF00 flag is not cleared because the processing for the flag does not proceed.

<8> As the ITF00 flag is set to 1 at the time the ITF01 flag is cleared to 0 in step 7, the INTITL signal is temporarily driven low and the interrupt request flag (ITLIF) is set to 1. If the interrupt enable flag (IE) is not cleared to 0 at this time, this interrupt request is held pending.

<9> The processing in response to the ITF01 flag being set to 1 is then executed.

<10> As the interrupt request flag (ITLIF) is still set to 1 after return from the processing in response to the ITF01 flag being set to 1, clear the interrupt request flag (ITLIF) to 0, after which the interval detection interrupt processing that was held pending proceeds.

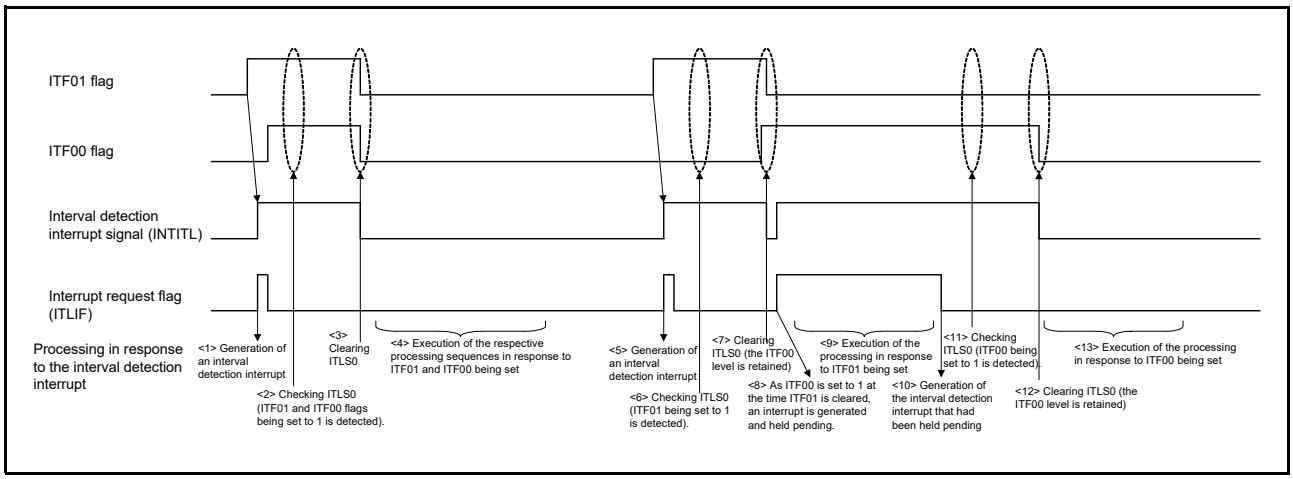
<11> Check which detection flag in the ITLS0 register is set to 1 from within the interval detection interrupt processing. In the case shown in **Figure 9 - 16**, the ITF00 flag being set to 1 can be confirmed.

<12> Clear the ITF00 flag detected in step 11 by using an 8-bit memory manipulation instruction to write 00011101B to the ITLS0 register so that its value becomes 00H.

<13> The processing in response to the ITF00 flag being set to 1 is then executed.

Note Missing an interrupt source can also be prevented by repeating the processing for clearing an interrupt source per flag.

Figure 9 - 16 Example of Clearing the Detected Flags



9.3.6 Interval timer setting procedures

The following shows the procedures for setting up the 32-bit interval timer.

Figure 9 - 17 Procedure for Starting the 32-bit Interval Timer

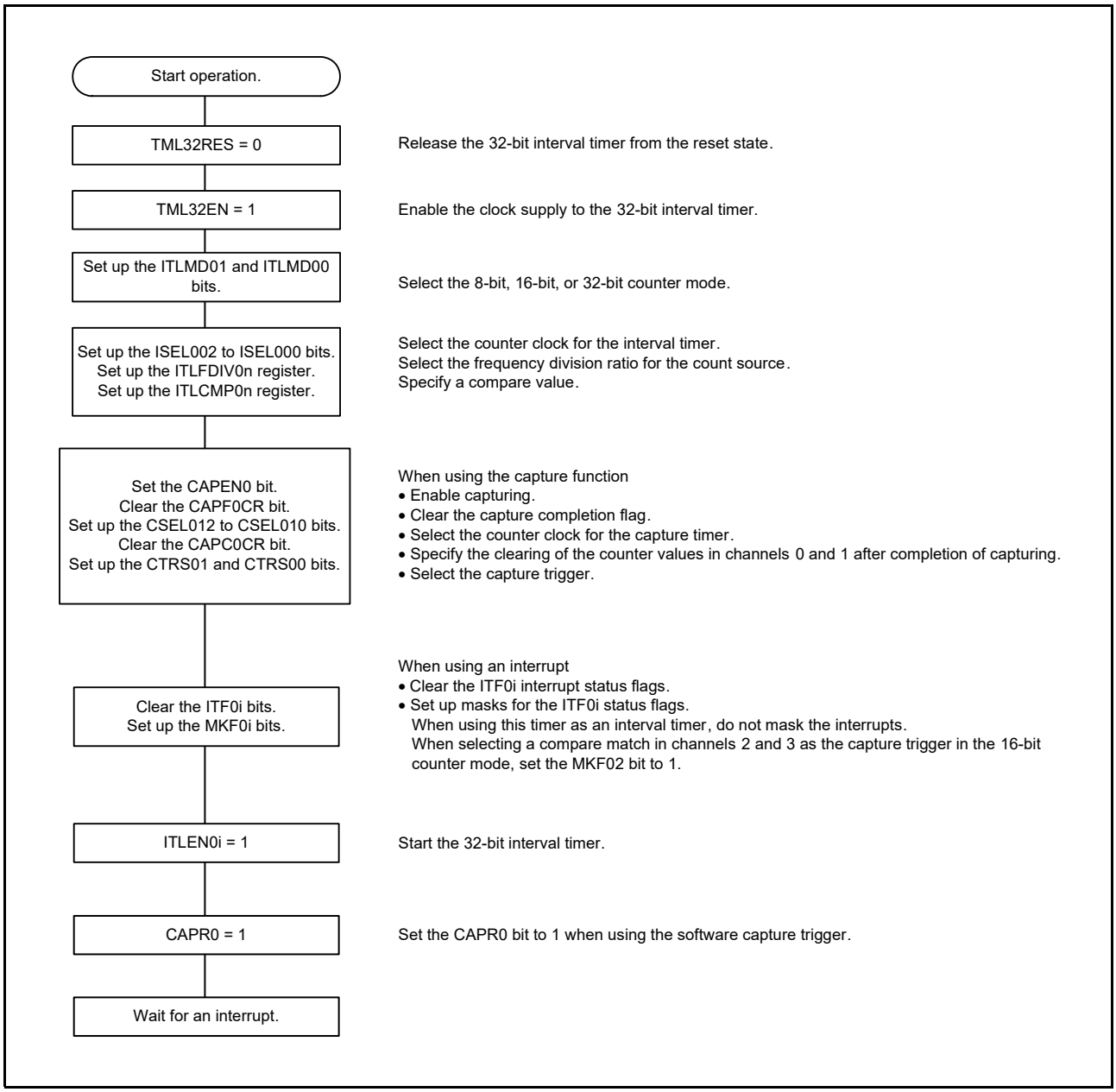


Figure 9 - 18 Procedure for Stopping the 32-bit Interval Timer

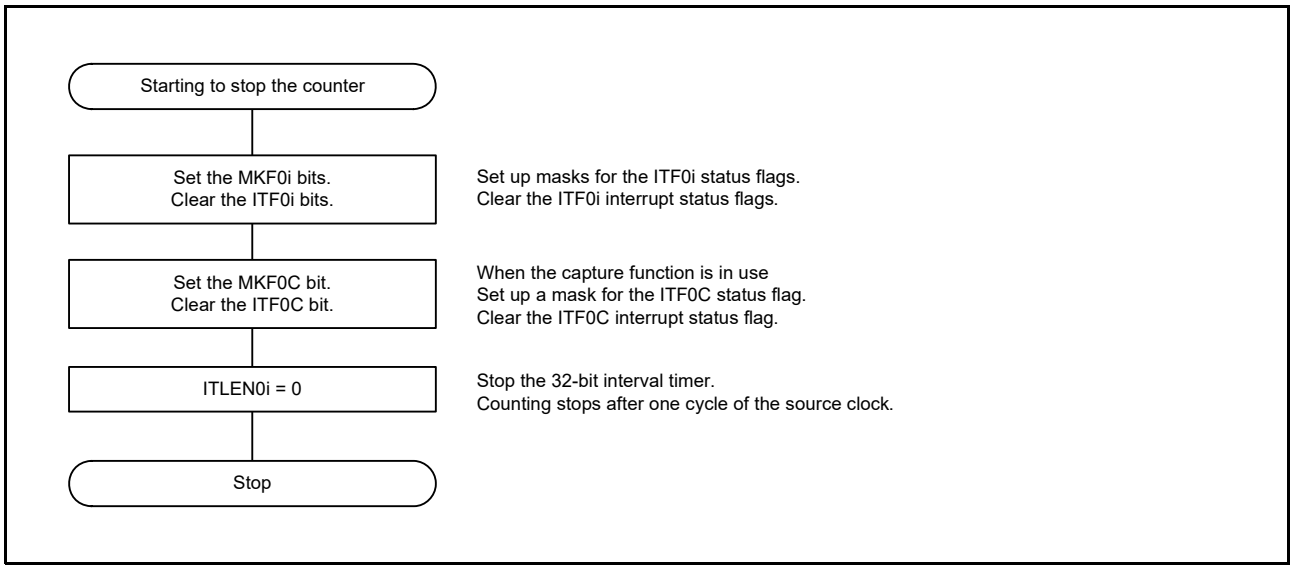


Figure 9 - 19 Procedure for Changing the Operating Mode of the 32-bit Interval Timer

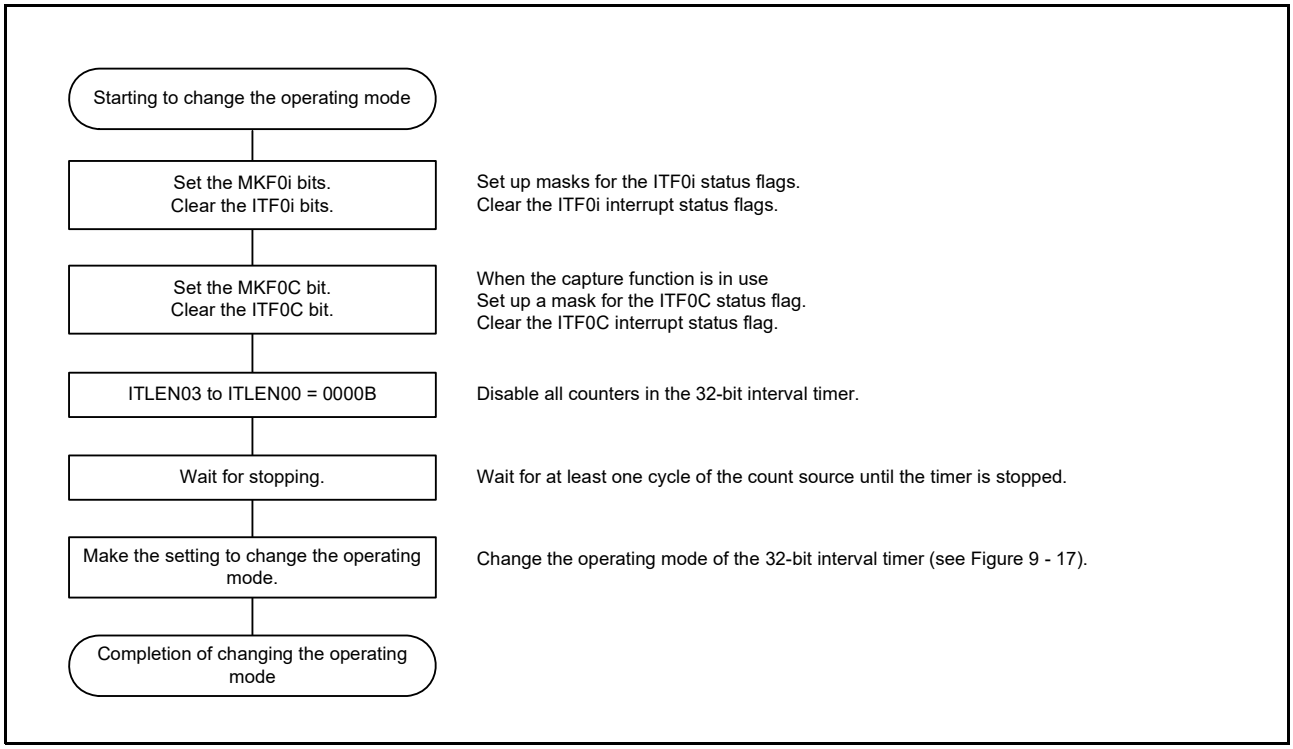


Figure 9 - 20 Procedure for Resetting the 32-bit Interval Timer

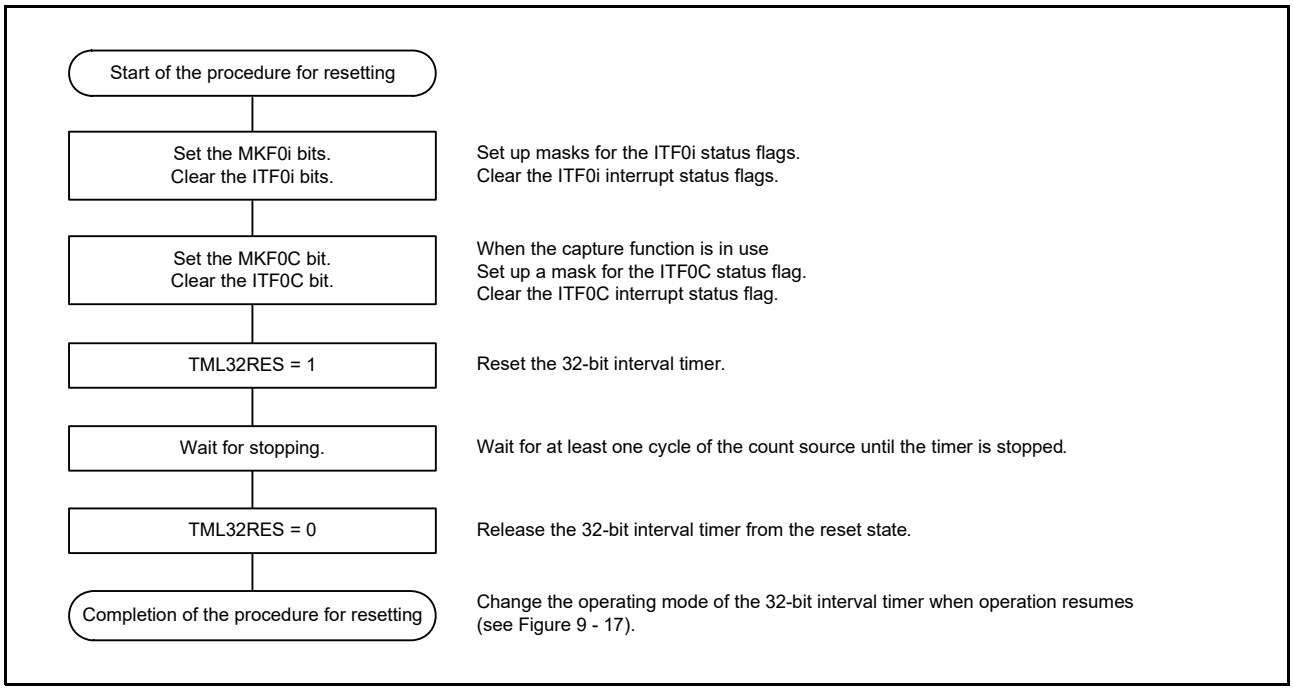


Figure 9 - 21 Procedure for Starting Event Input from the ELCL

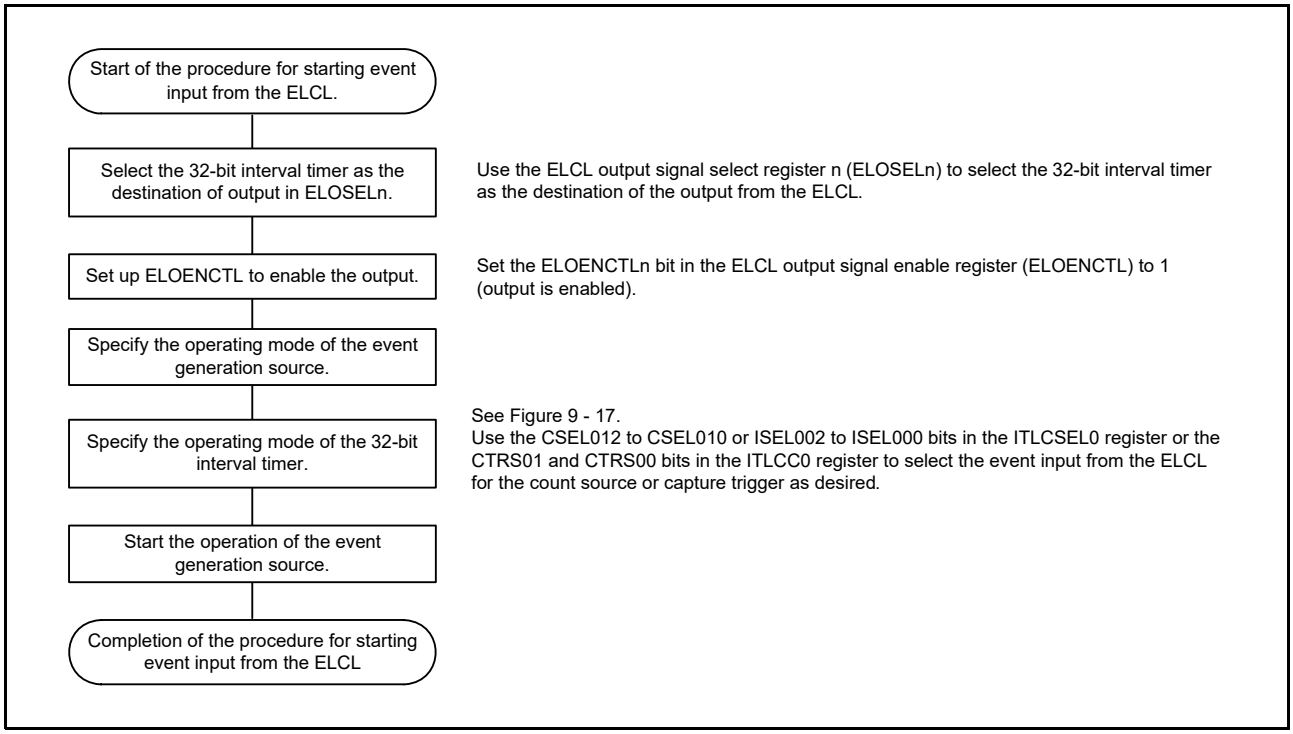
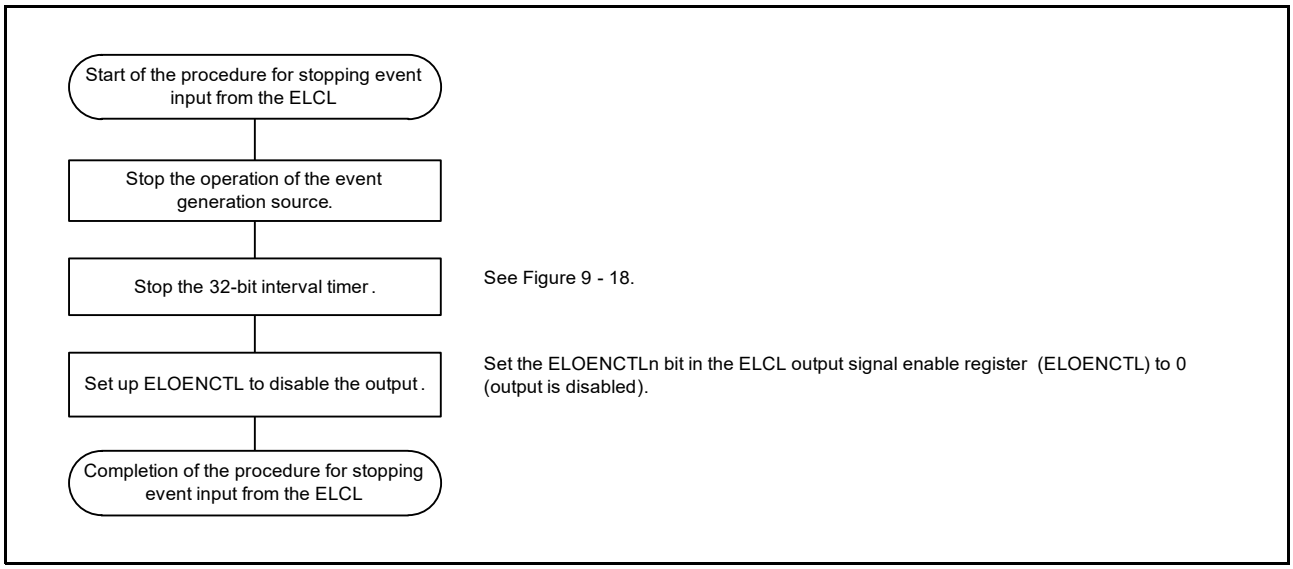


Figure 9 - 22 Procedure for Stopping Event Input from the ELCL



Section 10 Clock Output/Buzzer Output Controller (PCLBUZ)

The number of clock or buzzer output pins depends on the product.

Clock or Buzzer Output Pins	R7F100G	R7F104G
	30- to 128-pin	32- to 48-pin
PCLBUZ0	✓	✓
PCLBUZ1	✓	—

Caution Most of the following descriptions in this section use the 64-pin R7F100G products as an example.

10.1 Functions of Clock Output/Buzzer Output Controller

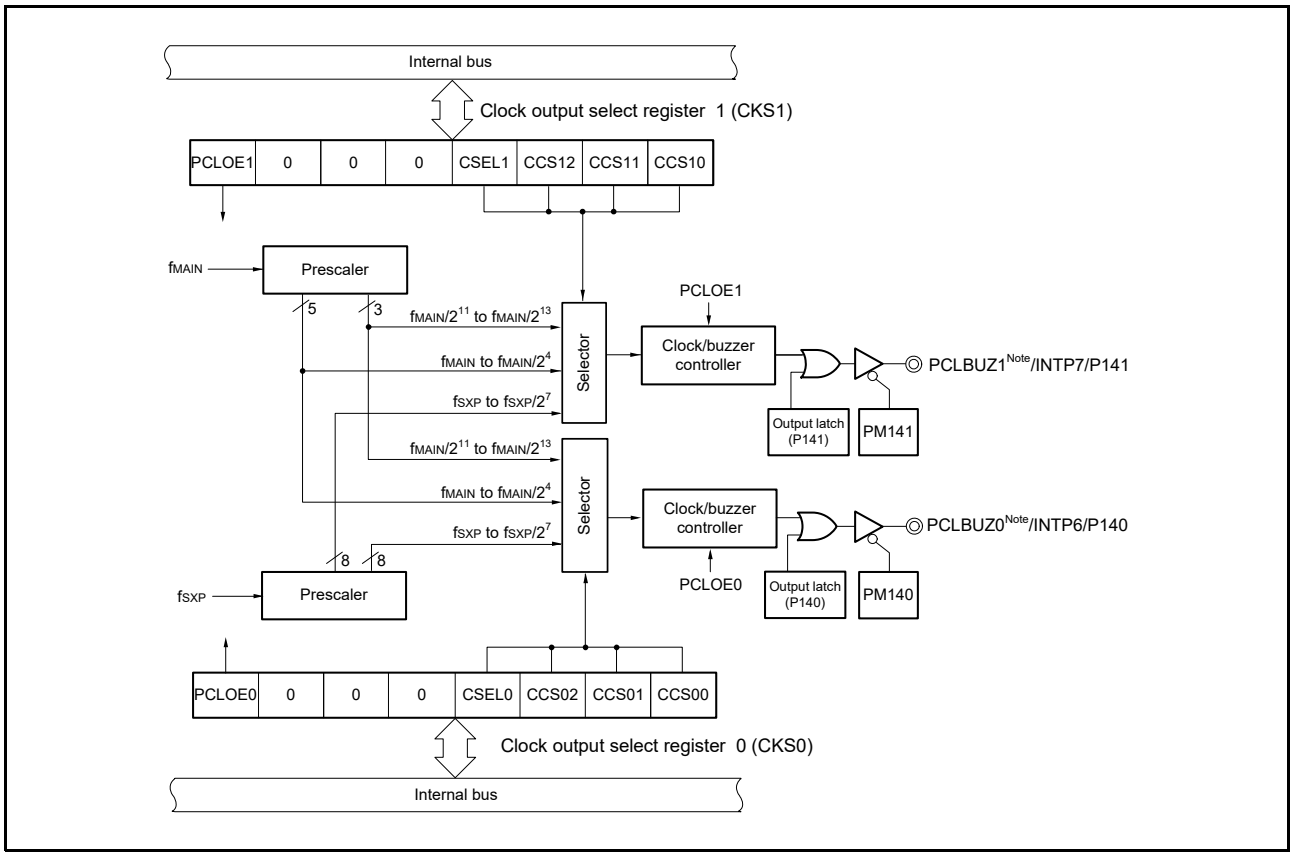
In clock output, the controller outputs a clock signal for supply to peripheral ICs. In buzzer output, the controller outputs a square wave at the buzzer frequency. This module has two output channels (PCLBUZn) and each of them can be specified to output a clock or buzzer signal. Switching of the output is handled by clock output select registers n (CKSn). **Figure 10 - 1** shows the block diagram of the clock output/buzzer output controller.

Caution Output of the low-speed peripheral clock (f_{SXP}) from the PCLBUZn pin is not possible when the following conditions are both satisfied:

- **The setting of the RTCLPC bit in the subsystem clock supply mode control register (OSMC) is 1.**
- **Operation is in the HALT mode with the subsystem clock (f_{SUB}) selected as the CPU clock.**

Remark n = 0, 1

Figure 10 - 1 Block Diagram of Clock Output/Buzzer Output Controller



Note For output frequencies available from PCLBUZ0 and PCLBUZ1, refer to **37.4 AC Characteristics**.

Remark The configuration of the clock or buzzer output pins shown in the above diagram is for the case of 64- to 128-pin R7F100G products with PIOR3 = 0 and PIOR4 = 0.

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 n (CKSn) • Port mode registers (PMxx) • Port registers (Pxx) • Port mode control T registers (PMCTxx) • Port mode control E registers (PMCExx)

10.3 Registers for Controlling the Clock Output/Buzzer Output Controller

The following registers are used to control the clock output/buzzer output controller.

- Clock output select registers n (CKSn)
- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)

10.3.1 Clock output select registers n (CKSn)

These registers enable or disable the output from the clock or buzzer frequency output pin (PCLBUZn), and set the output clock. Use the CKSn register to select the clock to be output from the PCLBUZn pin. The CKSn registers are set by a 1-bit or 8-bit memory manipulation instruction. The value of each CKSn register following a reset is 00H.

Figure 10 - 2 Format of Clock Output Select Registers n (CKSn)

Address: FFFA5H (CKS0), FFFA6H (CKS1)
After reset: 00H
R/W: R/W

Symbol	<7>	6	5	4	3	2	1	0
CKSn	PCLOEn	0	0	0	CSELn	CCSn2	CCSn1	CCSn0

PCLOEn	PCLBUZn pin output enable/disable specification
0	Output disabled
1	Output enabled

CSELn	CCSn2	CCSn1	CCSn0		PCLBUZn pin output clock selection			
					f_{MAIN} = 5 MHz	f_{MAIN} = 10 MHz	f_{MAIN} = 20 MHz	f_{MAIN} = 32 MHz
0	0	0	0	f_{MAIN}	5 MHz**Note**	10 MHz**Note**	Setting prohibited	Setting prohibited
0	0	0	1	$f_{MAIN}/2$	2.5 MHz	5 MHz**Note**	10 MHz**Note**	16 MHz**Note**
0	0	1	0	$f_{MAIN}/2^2$	1.25 MHz	2.5 MHz	5 MHz**Note**	8 MHz**Note**
0	0	1	1	$f_{MAIN}/2^3$	625 kHz	1.25 MHz	2.5 MHz	4 MHz
0	1	0	0	$f_{MAIN}/2^4$	312.5 kHz	625 kHz	1.25 MHz	2 MHz
0	1	0	1	$f_{MAIN}/2^{11}$	2.44 kHz	4.88 kHz	9.77 kHz	15.63 kHz
0	1	1	0	$f_{MAIN}/2^{12}$	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz
0	1	1	1	$f_{MAIN}/2^{13}$	610 Hz	1.22 kHz	2.44 kHz	3.91 kHz
1	0	0	0	f_{SXP}	32.768 kHz			
1	0	0	1	$f_{SXP}/2$	16.384 kHz			
1	0	1	0	$f_{SXP}/2^2$	8.192 kHz			
1	0	1	1	$f_{SXP}/2^3$	4.096 kHz			
1	1	0	0	$f_{SXP}/2^4$	2.048 kHz			
1	1	0	1	$f_{SXP}/2^5$	1.024 kHz			
1	1	1	0	$f_{SXP}/2^6$	512 Hz			
1	1	1	1	$f_{SXP}/2^7$	256 Hz			

Note The selectable output clock frequency depends on the power supply voltage (V_{DD}). See **37.4 AC Characteristics** for details.

Caution 1. Change the output clock after disabling clock output (PCLOEn = 0).

Caution 2. To shift to STOP mode when the main system clock is selected (CSELn = 0), set PCLOEn = 0 before executing the STOP instruction. When the subsystem clock is selected (CSELn = 1), PCLOEn = 1 can be set because the clock can be output while the RTCLPC bit of the subsystem clock supply mode control (OSMC) register is set to 0 and moreover while STOP mode is set.

Caution 3. Output of the low-speed peripheral clock (f_{SXP}) from the PCLBUZn pin is not possible when the following conditions are both satisfied:

- **The setting of the RTCLPC bit in the subsystem clock supply mode control register (OSMC) is 1.**
- **Operation is in the HALT mode with the subsystem clock (f_{SUB}) selected as the CPU clock.**

Remark 1. n = 0, 1

Remark 2. f_{MAIN}: Main system clock frequency
f_{SUB}: Subsystem clock frequency
f_{SXP}: Low-speed peripheral clock frequency

10.3.2 Registers for controlling the port functions multiplexed with the clock or buzzer outputs

Set the following registers to control the port functions multiplexed with the clock or buzzer outputs.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)

For details, see the following sections.

- **4.3.1 Port mode registers (PMxx)**
- **4.3.2 Port registers (Pxx)**
- **4.3.5 Port output mode registers (POMxx)**
- **4.3.8 Port mode control T registers (PMCTxx)**
- **4.3.9 Port mode control E registers (PMCExx)**

For examples of setting the registers for controlling the port function, see **4.5.4 Examples of register settings for port and multiplexed functions**.

10.4 Operations of the Clock Output/Buzzer Output Controller

This module has two output channels (PCLBUZn) and each of them can be specified to output a clock or buzzer signal. The PCLBUZ0 and PCLBUZ1 pins respectively output the clock or buzzer signal specified by the clock output select registers 0 and 1 (CKS0 and CKS1).

10.4.1 Operation of output pins

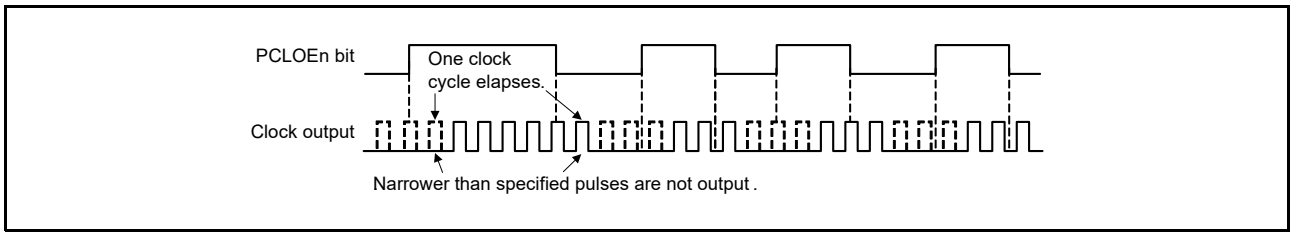
Follow the steps below to enable output from a PCLBUZn pin.

1. Set the corresponding bits of the following registers to 0 to select the port pin for use as a PCLBUZn pin.
   - Port mode register (PMxx)
   - Port register (Pxx)
   - Port mode control T register (PMCTxx)
   - Port mode control E register (PMCEx)
2. Select the output frequency with the CSELn and CCSn2 to CCSn0 bits of the clock output select register n (CKSn) for a PCLBUZn pin (output is still disabled).
3. Set the PCLOEn bit of the CKSn register to 1 to enable clock/buzzer output.

Remark 1. The controller used for the clock output starts or stops output one clock cycle after that in which the PCLOEn bit was set to enable or disable the output, respectively. Pulses that are narrower than the specified width are not output at those times. **Figure 10 - 3** shows the relationship between the setting of the PCLOEn bit and the timing of clock output.

Remark 2. n = 0, 1

Figure 10 - 3 Timing of the Clock Output from a PCLBUZn Pin



10.5 Point for Caution When the Clock Output/Buzzer Output Controller is to be Used

When the main system clock is selected for the PCLBUZn pin output (CKSn.CSELn = 0), if STOP mode is entered within 1.5 clock cycles of output from a PCLBUZn pin having been disabled (CKSn.PCLOEn = 0), the width of a pulse being output from a PCLBUZn pin at that time becomes shorter.

Section 11 Watchdog Timer (WDT)

11.1 Functions of Watchdog Timer

Counting by the watchdog timer is set by the user option byte (000C0H). The watchdog timer operates with the low-speed on-chip oscillator clock (f_{IL}) divided by 2 (1/2 f_{IL}). The watchdog timer is used to detect program malfunctions. If a malfunction is detected, an internal reset signal is generated. Any among the following cases is considered a program malfunction.

- The watchdog timer counter overflows.
- A 1-bit memory manipulation instruction is used to write to the watchdog timer enable register (WDTE).
- A value other than ACH is written to the WDTE register.
- Writing to the WDTE register proceeds while the window is closed.

When a reset occurs due to the watchdog timer, the WDTRF bit of the reset control flag register (RESF) is set to 1. For details of the RESF register, see **Section 24 Reset Function**. When 75% of the overflow time + 1/4 f_{IL} 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)

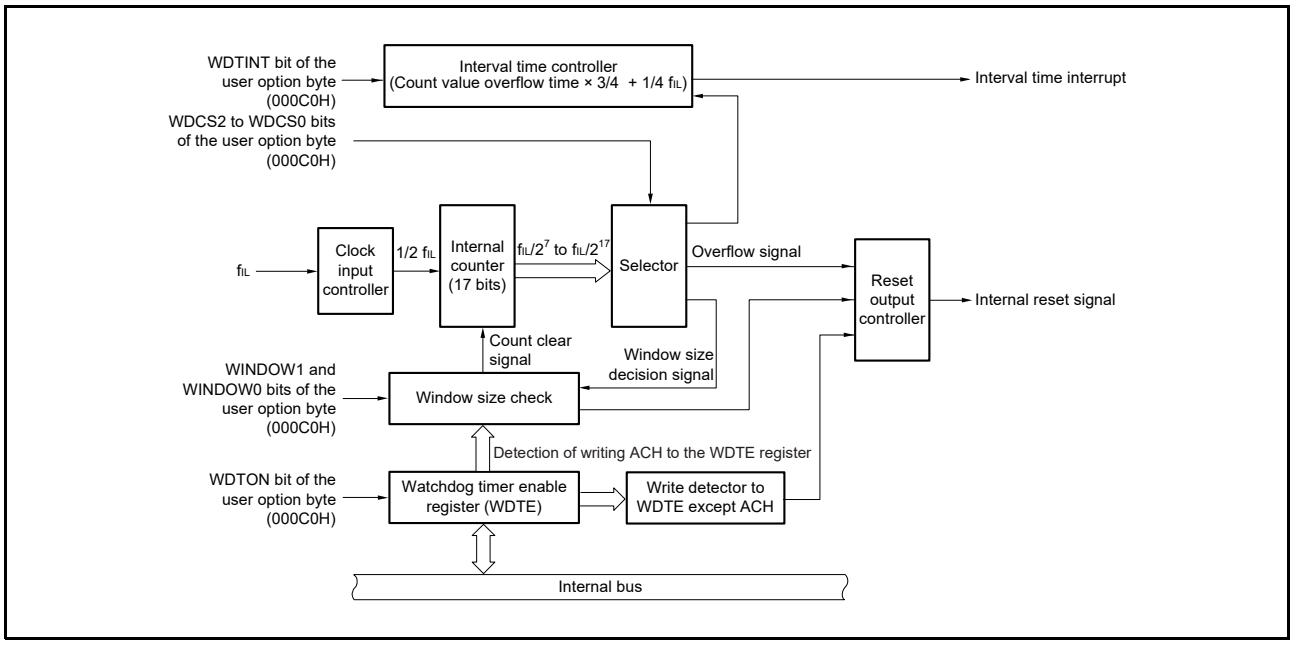
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the user option byte.

Table 11 - 2 Settings of the User Option Byte and Watchdog Timer

Setting of Watchdog Timer	User 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 mode)	Bit 0 (WDSTBYON)

Remark For the user option byte, see **Section 32 Option Bytes**.

Figure 11 - 1 Block Diagram of Watchdog Timer



Remark fIL: Low-speed on-chip oscillator clock

11.3 Register for Controlling the Watchdog Timer

The following register is used to control the watchdog timer.

- Watchdog timer enable register (WDTE)

11.3.1 Watchdog timer enable register (WDTE)

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. The value of this register following a reset is 9AH or 1AH**Note**.

Note The value of the WDTE register following a reset depends on the setting of the WDTON bit of the user option byte (000C0H). To enable counting by the watchdog timer, set the WDTON bit to 1.

Figure 11 - 2 Format of Watchdog Timer Enable Register (WDTE)

Address: FFFABH
After reset: 9AH/1AH**Note**
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
WDTE								

WDTON Bit Setting Value	Value of the WDTE Register following a Reset
0 (Watchdog timer counting disabled)	1AH
1 (Watchdog timer counting enabled)	9AH

Note The value of the WDTE register following a reset depends on the setting of the WDTON bit of the user option byte (000C0H). To enable counting by the watchdog timer, set the WDTON bit to 1.

Caution 1. If a value other than ACH is written to the WDTE register, an internal reset signal is generated.
Caution 2. If a 1-bit memory manipulation instruction is executed for the WDTE register, an internal reset signal is generated.
Caution 3. The value read from the WDTE register is 9AH or 1AH. Note that the written value (ACH) is not read.

Remark When the setting of the WDTON bit in the user option byte (000C0H) is 1 (watchdog timer operation enabled), the illicit memory access detection is enabled even if the setting of the IAWEN bit in the IAWCTL register is 0 (the detection of illicit memory accesses disabled).

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 user option byte (000C0H).
   - Enable counting by the watchdog timer by setting bit 4 (WDTON) of the user option byte (000C0H) to 1 (the counter starts operating after release from the reset state). For details, see **Section 32 Option Bytes**.

WDTON	Watchdog Timer Counter
0	Counter operation disabled (counting stopped after release from the reset state)
1	Counter operation enabled (counting started after release from the reset state)

   - Set the time at which the counter is to overflow by using bits 3 to 1 (WDCS2 to WDCS0) of a user option byte (000C0H). For details, see **11.4.2** and **Section 32 Option Bytes**.
   - Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the user option byte (000C0H). For details, see **11.4.3** and **Section 32 Option Bytes**.
2. After release from the reset state, 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 user option byte, the watchdog timer is cleared and starts counting again.
4. After that, writing to the WDTE register the second and subsequent times must proceed while the window is open. If the WDTE register is written during a window close period, an internal reset signal is generated.
5. If the time at which an overflow is to occur elapses without ACH having been written to the WDTE register, an internal reset signal is generated. An internal reset signal is also generated in the following cases.
   - If a 1-bit memory manipulation instruction is executed on the WDTE register
   - If data other than ACH are written to the WDTE register

Caution 1. When data are written to the watchdog timer enable register (WDTE) for the first time after release from the reset state, the watchdog timer 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 starts counting again.

Caution 2. After ACH is written to the WDTE register, an error of up to 4 cycles of the clock at f_{IL} may occur before the watchdog timer is cleared.

Caution 3. The watchdog timer can be cleared immediately before the counter value overflows.

Caution 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) of the user option byte (000C0H).

	WDSTBYON = 0	**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 or STOP mode 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 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 user option byte (000C0H). If an overflow occurs, an internal reset signal is generated. The present count is cleared and the watchdog timer starts counting again by writing ACH to the watchdog timer enable register (WDTE) during the window open period before the overflow time. The following overflow times can be set.

Table 11 - 3 Setting of Overflow Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer (f_{IL} = 37.683 kHz (max.))
0	0	0	$2^7/f_{IL}$ (3.39 ms)
0	0	1	$2^8/f_{IL}$ (6.79 ms)
0	1	0	$2^9/f_{IL}$ (13.58 ms)
0	1	1	$2^{10}/f_{IL}$ (27.17 ms)
1	0	0	$2^{12}/f_{IL}$ (108.69 ms)
1	0	1	$2^{14}/f_{IL}$ (434.78 ms)**Note**
1	1	0	$2^{15}/f_{IL}$ (869.56 ms)**Note**
1	1	1	$2^{17}/f_{IL}$ (3478.26 ms)**Note**

Note Using the watchdog timer under the following conditions may lead to the generation of an interval interrupt (INTWDTI) after one cycle of the watchdog timer clock once the watchdog timer counter has been cleared.

- The watchdog timer interval interrupt is in use.
- ACH is written to the WDTE register (FFFABH) when the watchdog timer counter has reached or exceeded 75% of the overflow time.

This interrupt can be masked by clearing the watchdog timer counter through steps <1> to <5> below.

<1> Set the WDTIMK bit of the interrupt mask flag register 0 (MK0L) to 1 before clearing the watchdog timer counter.
<2> Clear the watchdog timer counter.
<3> Wait for at least 80 μs.
<4> Clear the WDTIIF bit of the interrupt request flag register 0 (IF0L) to 0.
<5> Clear the WDTIMK bit of the interrupt mask flag register 0 (MK0L) to 0.

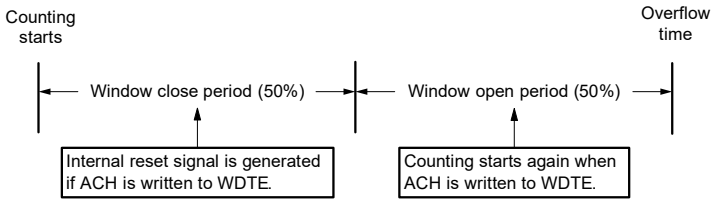
Remark f_{IL}: Low-speed on-chip oscillator clock frequency

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 user 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 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 are written to the WDTE register for the first time after release from the reset state, the watchdog timer 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 starts counting again.

The window open period can be set is as follows.

Table 11 - 4 Setting Window Open Period of Watchdog Timer

WINDOW1	WINDOW0	Window Open Period of Watchdog Timer
0	1	50%
1	1	100%
Others		Setting prohibited

Remark If the overflow time is set to $2^{10}/f_{IL}$, the times over which the window is open and closed are as follows.

	Setting of Window Open Period	
	50%	100%
Window close time	0 to 18.38 ms	None
Window open time	18.38 to 27.17 ms	0 to 27.17 ms

<When window open period is 50%>

- Overflow time:
  $2^{10}/f_{IL}$ (max.) = 2^{10}/37.683 kHz = 27.17 ms
- Window close time:
  0 to $2^{10}/f_{IL}$ (min.) × (1 - 0.5) = 0 to 2^{10}/27.852 kHz × 0.5 = 0 to 18.38 ms
- Window open time:
  $2^{10}/f_{IL}$ (min.) × (1 - 0.5) to $2^{10}/f_{IL}$ (max.) = 2^{10}/27.853 kHz × 0.5 to 2^{10}/37.683 kHz = 18.38 to 27.17 ms

11.4.4 Setting watchdog timer interval interrupt

Depending on the setting of bit 7 (WDTINT) of a user option byte (000C0H), an interval interrupt (INTWDTI) can be generated when 75% of the overflow time + 1/4 f_{IL} is reached.

Table 11 - 5 Setting of Watchdog Timer Interval Interrupt

WDTINT	Use of Watchdog Timer Interval Interrupt
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% of the overflow time + 1/4 f_{IL} 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.

Section 12 A/D Converter (ADC)

The number of analog input channels of the A/D converter depends on the product.

	R7F100G							
	30- and 32-pin	36-pin	40-pin	44- and 48-pin	52- and 64-pin	80-pin	100-pin	128-pin
Number of the analog input channels	8 (ANI0 to ANI3, ANI16 to ANI19)	8 (ANI0 to ANI5, ANI18, ANI19)	9 (ANI0 to ANI6, ANI18, ANI19)	10 (ANI0 to ANI7, ANI18, ANI19)	12 (ANI0 to ANI7, ANI16 to ANI19)	17 (ANI0 to ANI11, ANI16 to ANI20)	20 (ANI0 to ANI14, ANI16 to ANI20)	26 (ANI0 to ANI14, ANI16 to ANI26)

	R7F104G		
	32-pin	44-pin	48-pin
Number of the analog input channels	7 (ANI0 to ANI3, ANI16 to ANI18)	8 (ANI0 to ANI7)	10 (ANI0 to ANI7, ANI16, ANI17)

12.1 Function of A/D Converter

The A/D converter is used to convert analog input signals into digital values, and is configured to control up to 26 channels of A/D converter analog inputs (ANI0 to ANI14 and ANI16 to ANI26). 12-bit, 10-bit, or 8-bit resolution can be selected by the ADTYP1 and ADTYP0 bits of A/D converter mode register 2 (ADM2). The A/D converter has the following function.

- 12-bit/10-bit/8-bit resolution A/D conversion
  12-bit, 10-bit, or 8-bit resolution A/D conversion proceeds repeatedly for one analog input channel selected from ANI0 to ANI14 and ANI16 to ANI26. Each time an A/D conversion operation is complete, an interrupt request signal (INTAD) is generated in the select mode.

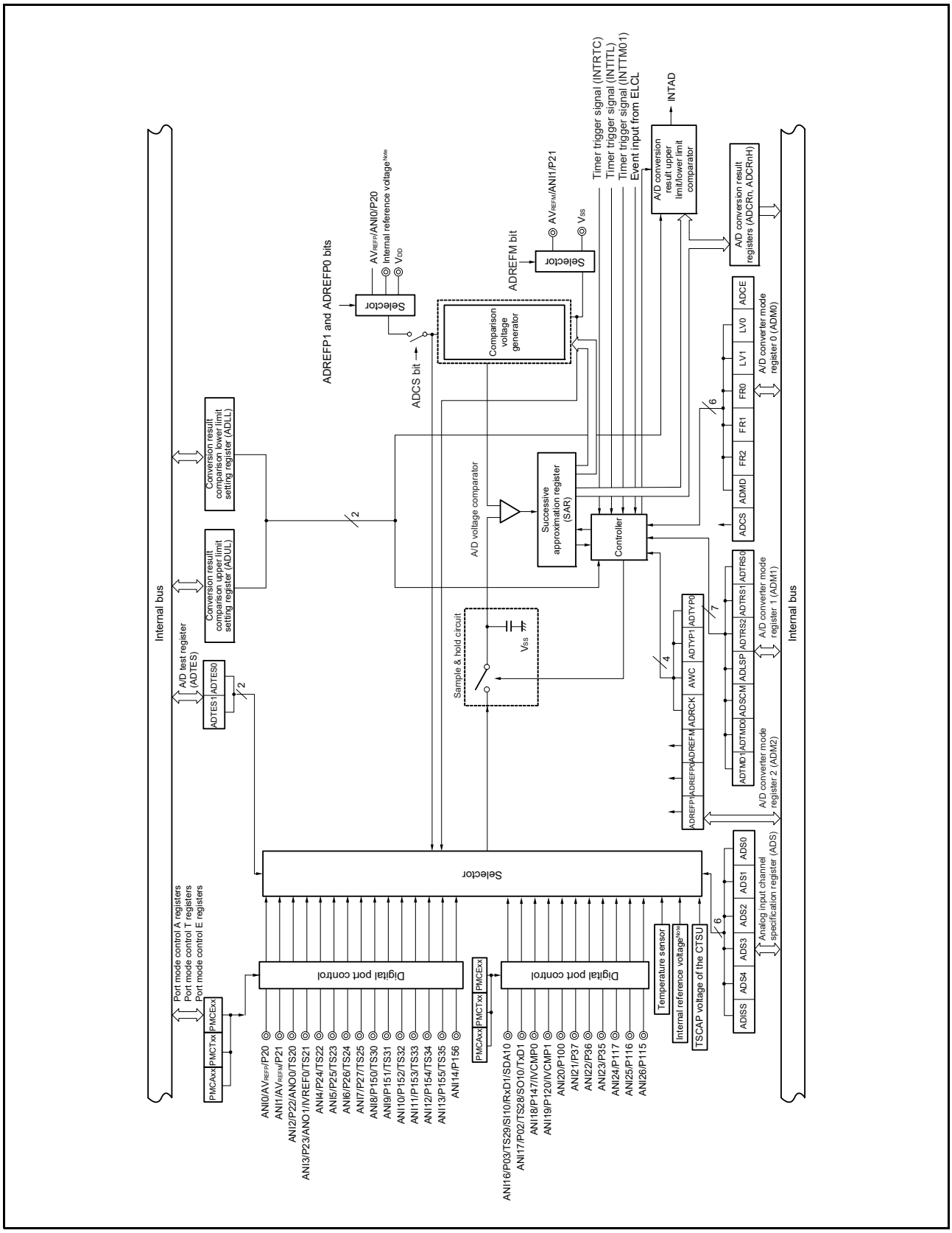
Various A/D conversion modes can be specified by using the mode combinations below.

Trigger mode	Software trigger no-wait mode	Conversion is started by setting the ADCE bit to 1 by software, and then setting the ADCS bit to 1 after the A/D power supply stabilization wait time has passed.
	Software trigger wait mode	The power is turned on by setting the ADCS bit to 1 by software while A/D conversion is stopped and conversion is then started automatically after the A/D power supply stabilization wait time has passed.
	Hardware trigger no-wait mode	Conversion is started by detecting a hardware trigger.
	Hardware trigger wait mode	The power to the A/D converter is turned on by detecting a hardware trigger while the power is off and the A/D converter is in the conversion standby state, and conversion is then started automatically after the A/D power supply stabilization wait time has passed. When using the SNOOZE mode function, specify the hardware trigger wait mode.
Channel selection mode	Select mode	A/D conversion is performed on the analog input of one selected channel.
	Scan mode	A/D conversion is performed on the analog input of four channels in order. Four consecutive channels can be selected from ANI0 to ANI14 as analog input channels.
Conversion operation mode	One-shot conversion mode	A/D conversion is performed on the selected channel once.
	Sequential conversion mode	A/D conversion is sequentially performed on the selected channels until it is stopped by software.

Operation voltage mode**Note**	Sampling clock cycles	
Normal 1	43 f_{AD}	Set the number of sampling clock cycles so that the sampling capacitor is sufficiently charged in accord with the output impedance of the analog input source.
Normal 2	160 f_{AD}	
Low voltage 1	53 f_{AD}	
Low voltage 2	80 f_{AD}	

Note The operation mode that can be selected differs depending on the analog input channel, V_{DD} voltage, AV_{REFP} voltage, trigger mode, and f_{CLK}. See **Table 12 - 3 A/D Conversion Time Selection** for details.

Figure 12 - 1 Block Diagram of A/D Converter



Note For the internal reference voltage, see **Section 37 Electrical Characteristics**.

Remark 1. Analog input pins in this figure are for a 128-pin product.

Remark 2. n = 0 to 3

12.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

1. ANI0 to ANI14 and ANI16 to ANI26 pins
   These are the analog input pins of the 26 channels of the A/D converter. They input analog signals to be converted into digital signals. Pins other than the one selected as the analog input pin can be used as I/O port pins.
2. Sample & hold (S&H) circuit
   The sample & hold circuit samples each of the analog input voltages sequentially sent from the input circuit, and sends them to the A/D voltage comparator. This circuit also holds the sampled analog input voltage during A/D conversion.
3. A/D voltage comparator
   This A/D voltage comparator compares the voltage generated from the voltage tap of the comparison voltage generator with the analog input voltage. If the analog input voltage is found to be greater than the reference voltage (1/2 AV_{REF}) as a result of the comparison, the most significant bit (MSB) of the successive approximation register (SAR) is set. If the analog input voltage is less than the reference voltage (1/2 AV_{REF}), the MSB of the SAR is reset. After that, bit 10 of the SAR register is automatically set, and the next comparison is made. The voltage tap of the comparison voltage generator is selected by the value of bit 11, to which the result has already been set.

   - Bit 11 = 0: (1/4 AV_{REF})
   - Bit 11 = 1: (3/4 AV_{REF})

   The voltage tap of the comparison voltage generator and the analog input voltage are compared and bit 10 of the SAR register is manipulated in accord with the result of the comparison.

   - Analog input voltage ≥ Voltage tap of comparison voltage generator: Bit 10 = 1
   - Analog input voltage ≤ Voltage tap of comparison voltage generator: Bit 10 = 0

   Comparison is continued like this to bit 0 of the SAR register.

4. Comparison voltage generator
   The comparison voltage generator generates the voltage to be compared with the input from an analog input pin.
5. Successive approximation register (SAR)
   The SAR is used to set voltage tap data whose values from the comparison voltage generator match the voltage values of the analog input pins, one bit at a time starting from the most significant bit (MSB). If data is set in the SAR register all the way to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register (conversion results) are held in the A/D conversion result register (ADCRn). When all the specified A/D conversion operations have ended, an A/D conversion end interrupt request signal (INTAD) is generated.
6. 12-bit/10-bit A/D conversion result register (ADCRn)
   Each time A/D conversion is complete, the conversion result is loaded from the successive approximation register, and then operation is performed as follows:
   When A/D conversion with 12-bit resolution is selected, it holds the A/D conversion result in its lower 12 bits (the higher 4 bits are fixed to 0).
   When A/D conversion with 10-bit resolution is selected, it holds the A/D conversion result in its higher 10 bits (the lower 6 bits are fixed to 0).

7. 8-bit A/D conversion result register (ADCRnH)
   The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCRnH register holds the higher 8 bits of the A/D conversion result.
8. Controller
   This circuit controls the conversion time of an analog input signal that is to be converted into a digital signal, as well as starting and stopping of the conversion operation. When A/D conversion has been completed, this controller generates INTAD through the A/D conversion result upper limit/lower limit comparator.
9. AV_{REFP} pin
   This pin is used to input an external reference voltage (AV_{REFP}). To use AV_{REFP} as the positive reference voltage of the A/D converter, set the ADREFP1 and ADREFP0 bits of A/D converter mode register 2 (ADM2) to 0 and 1, respectively. The analog signals input to ANI2 to ANI14 and ANI16 to ANI26 are converted to digital signals based on the voltage applied between AV_{REFP} and the negative reference voltage (AV_{REFM}/V_{SS}). Selecting V_{DD} or the internal reference voltage as the positive reference voltage of the A/D converter is also possible.
10. AV_{REFM} pin
    This pin is used to input an external reference voltage (AV_{REFM}). To use AV_{REFM} as the negative reference voltage of the A/D converter, set the ADREFM bit of the ADM2 register to 1. Selecting V_{SS} as the negative reference voltage of the A/D converter is also possible.

Remark AV_{REF}: The positive reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage, and V_{DD}. For the internal reference voltage, see **Section 37 Electrical Characteristics**.

12.3 Registers for Controlling the A/D Converter

The following registers are used to control the A/D converter.

- Peripheral enable register 0 (PER0)
- Peripheral reset control register 0 (PRR0)
- A/D converter mode register 0 (ADM0)
- A/D converter mode register 1 (ADM1)
- A/D converter mode register 2 (ADM2)
- 12-bit/10-bit A/D conversion result register (ADCRn)
- 8-bit A/D conversion result register (ADCRnH)
- Analog input channel specification register (ADS)
- Conversion result comparison upper limit setting register (ADUL)
- Conversion result comparison lower limit setting register (ADLL)
- A/D test register (ADTES)
- Port mode registers (PMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)

12.3.1 Peripheral enable register 0 (PER0)

The PER0 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. If the A/D converter is to be used, be sure to set the ADCEN bit of this register to 1. The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 12 - 2 Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCWEN	IICA1EN Note 1	ADCEN	IICA0EN Note 2	SAU1EN	SAU0EN	TAU1EN Note 3	TAU0EN

ADCEN	Control of supply of an input clock to the A/D converter
0	Stops supply of an input clock. • The SFRs used by the A/D converter cannot be written.Note 4
1	Enables supply of an input clock. • The SFRs used by the A/D converter can be read and written.

Note 1. This bit is only present in the 44- to 128-pin R7F100G products.
Note 2. This bit is only present in the 30- to 128-pin R7F100G and 48-pin R7F104G products.
Note 3. This bit is only present in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products.
Note 4. When an SFR is read, the value returned is 00H or 0000H regardless of its setting.

Caution 1. Be sure to leave the following bits set to 0.
Bits 6 and 1 in the 30- to 40-pin R7F100G products
Bit 1 in the 44- to 64-pin R7F100G products
Bits 6, 4, and 1 in the 32-pin R7F104G product
Bits 6 and 4 in the 44-pin R7F104G product
Bit 6 in the 48-pin R7F104G product

Caution 2. When setting the A/D converter, start by setting the following registers while the setting of the ADCEN bit is 1.
- **A/D converter mode register 0 (ADM0)**
- **A/D converter mode register 1 (ADM1)**
- **A/D converter mode register 2 (ADM2)**
- **12-bit/10-bit A/D conversion result register (ADCRn)**
- **8-bit A/D conversion result register (ADCRnH)**
- **Analog input channel specification register (ADS)**
- **Conversion result comparison upper limit setting register (ADUL)**
- **Conversion result comparison lower limit setting register (ADLL)**
- **A/D test register (ADTES)**

If the setting of the ADCEN bit is 0, each register which controls the A/D converter is restored to 00H and writing to any of those registers is ignored. Note, however, writing to the following registers is valid.
- **Port mode registers (PM0, PM2, PM3, PM10, PM11, PM12, PM14, and PM15)**
- **Port mode control A registers (PMCA0, PMCA2, PMCA3, PMCA10, PMCA11, PMCA12, PMCA14, and PMCA15)**
- **Port mode control T registers (PMCT0, PMCT2, and PMCT15)**
- **Port mode control E register (PMCE0)**

12.3.2 Peripheral reset control register 0 (PRR0)

The PRR0 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. Set the ADCRES bit of this register to 1 to place the A/D converter in the reset state. The PRR0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 12 - 3 Format of Peripheral Reset Control Register 0 (PRR0)

Address: F00F1H
After reset: 00H
R/W: R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PRR0	0	IICA1RES **Note 1**	ADCRES	IICA0RES **Note 2**	SAU1RES	SAU0RES	TAU1RES **Note 3**	TAU0RES

ADCRES	Control resetting of the A/D converter
0	The A/D converter is released from the reset state.
1	The A/D converter is in the reset state. • The SFRs for use with the A/D converter are initialized.

Note 1. This bit is only present in the 44- to 128-pin R7F100G products.
Note 2. This bit is only present in the 30- to 128-pin R7F100G and 48-pin R7F104G products.
Note 3. This bit is only present in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products.

Caution 1. Be sure to leave bit 7 set to 0.
Caution 2. Be sure to leave the following bits set to 0.
Bits 6 and 1 in the 30- to 40-pin R7F100G products
Bit 1 in the 44- to 64-pin R7F100G products
Bits 6, 4, and 1 in the 32-pin R7F104G product
Bits 6 and 4 in the 44-pin R7F104G product
Bit 6 in the 48-pin R7F104G product

12.3.3 A/D converter mode register 0 (ADM0)

This register sets the time for converting analog input to digital data, and starts and stops conversion. The ADM0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 12 - 4 Format of A/D Converter Mode Register 0 (ADM0)

Address: FFF30H
After reset: 00H
R/W: R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM0	ADCS	ADMD	FR2 **Note 1**	FR1 **Note 1**	FR0 **Note 1**	LV1 **Note 1**	LV0 **Note 1**	ADCE

ADCS	A/D conversion operation control
0	Stops conversion operation. [When read] Conversion is stopped or in standby.
1	Enables conversion operation. [When read] While in the software trigger no-wait mode: Conversion in progress While in the software trigger wait mode: A/D power supply stabilization wait state or conversion in progress While in the hardware trigger no-wait mode: Conversion is enabled. While in the hardware trigger wait mode: A/D power supply stabilization wait state or conversion in progress

ADMD	Specification of the A/D conversion channel selection mode
0	Select mode
1	Scan mode

ADCE	A/D voltage comparator operation control**Note 2**
0	Stops A/D voltage comparator operation.
1	Enables A/D voltage comparator operation.

Note 1. For details of the FR2 to FR0, LV1, LV0 bits, and A/D conversion, see **Table 12 - 3 A/D Conversion Time Selection**.

Note 2. While in the software trigger no-wait mode or hardware trigger no-wait mode, the operation of the A/D voltage comparator is controlled by the ADCS and ADCE bits, and it takes 1 μs + 2 cycles of the conversion clock (f_{AD}) from the start of operation for the operation to become stable. Therefore, after having set the ADCE bit to 1, wait for at least 1 μs + 2 cycles of the conversion clock (f_{AD}) before setting the ADCS bit to 1; this ensures that the results of conversion will be valid from the very first conversion. When the ADCS bit is set to 1 while the setting of the ADCE bit is 0, A/D conversion starts after the stabilization wait time has passed. If the ADCS bit is set to 1 without the wait, ignore the data from the first conversion.

Caution 1. Change the ADMD, FR2 to FR0, LV1, and LV0 bits while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 2. Setting change from ADCS = 1 and ADCE = 1 to ADCS = 1 and ADCE = 0 is prohibited.

Caution 3. Do not change the ADCS and ADCE bits from 0 to 1 at the same time by using an 8-bit memory manipulation instruction. Be sure to follow the procedure described in 12.7 Flows of Setting the A/D Converter Modes.

Caution 4. Following stoppage of conversion by setting the ADCS and ADCE bits to 0 from the conversion standby or conversion state, wait for at least 5 μs before restoring the values of the bits to 1. Note that, when changing the settings of bits ADMD, FR2 to FR0, LV1, and LV0, start by setting the ADCS and ADCE bits to 0, then wait for at least 0.2 μs before changing the rest of the bits.

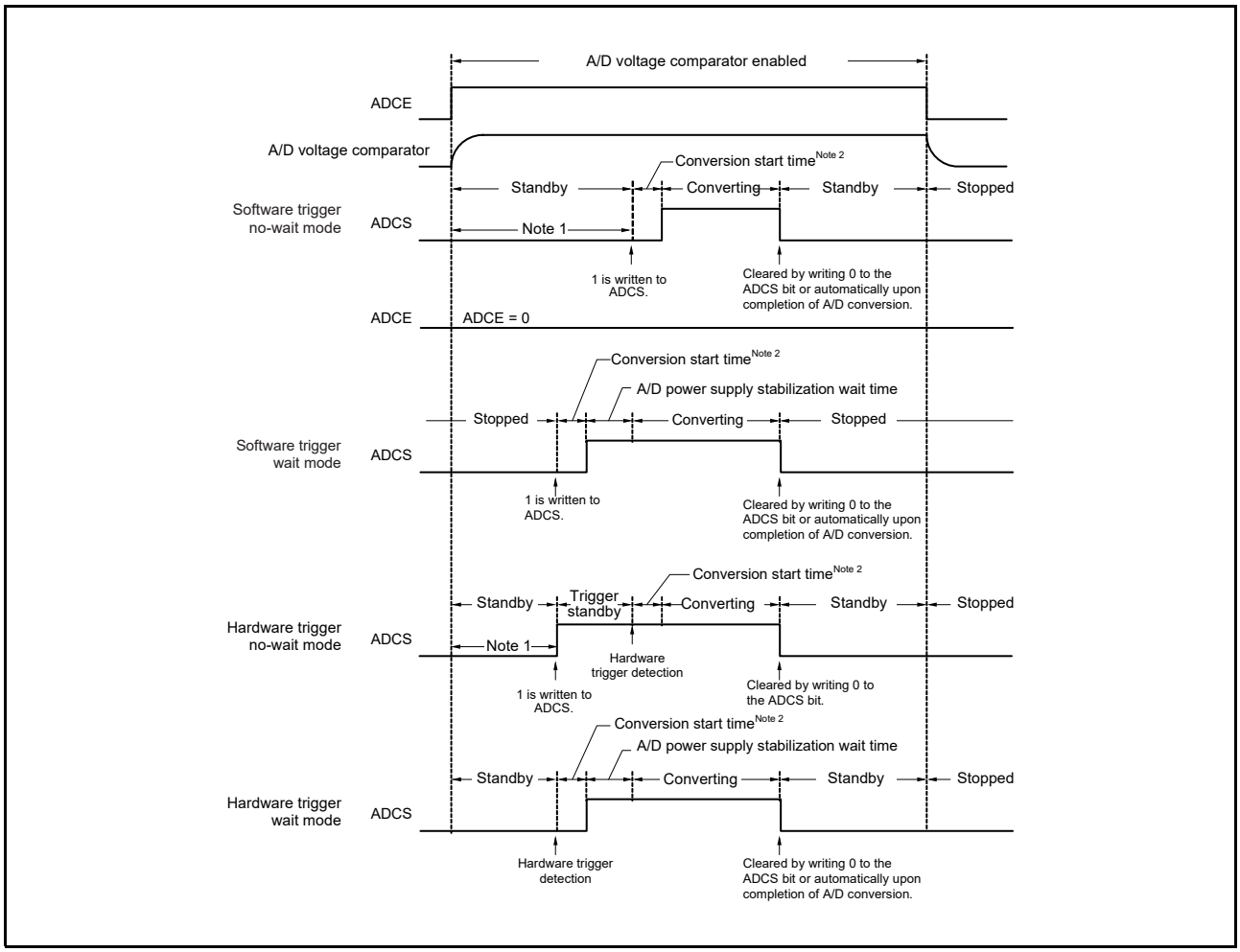
Table 12 - 1 Settings of ADCS and ADCE Bits

ADCS	ADCE	A/D Conversion Operation
0	0	Stopped
0	1	Standby
1	0	Conversion in progress in software trigger wait mode or stopped
1	1	Conversion in progress other than in software trigger wait mode

Table 12 - 2 Conditions for Setting and Clearing the ADCS Bit

A/D Conversion Mode			Set Conditions	Clear Conditions
Software trigger no-wait mode	Select mode	Sequential conversion mode	1 is written to ADCS.	• 0 is written to ADCS.
		One-shot conversion mode		• 0 is written to ADCS. • A/D conversion is complete.
	Scan mode	Sequential conversion mode		• 0 is written to ADCS.
		One-shot conversion mode		• 0 is written to ADCS. • Conversion is complete on the specified four channels.
Software trigger wait mode	Select mode	Sequential conversion mode		• 0 is written to ADCS.
		One-shot conversion mode		• 0 is written to ADCS. • A/D conversion is complete.
	Scan mode	Sequential conversion mode		• 0 is written to ADCS.
		One-shot conversion mode		• 0 is written to ADCS. • Conversion is complete on the specified four channels.
Hardware trigger no-wait mode	Select mode	Sequential conversion mode		• 0 is written to ADCS.
		One-shot conversion mode		• 0 is written to ADCS.
	Scan mode	Sequential conversion mode		• 0 is written to ADCS.
		One-shot conversion mode		• 0 is written to ADCS.
Hardware trigger wait mode	Select mode	Sequential conversion mode	A hardware trigger is input.	• 0 is written to ADCS.
		One-shot conversion mode		• 0 is written to ADCS. • A/D conversion is complete.
	Scan mode	Sequential conversion mode		• 0 is written to ADCS.
		One-shot conversion mode		• 0 is written to ADCS. • Conversion is complete on the specified four channels.

Figure 12 - 5 Timing of Operations When the A/D Voltage Comparator Is Used



Note 1. While in the software trigger no-wait mode or hardware trigger no-wait mode, the interval between setting the ADCE bit to 1 and setting the ADCS bit to 1 must be 1 μs + 2 cycles of the conversion clock (f_{AD}) or longer to allow stabilization of the internal circuit.

Note 2. The following shows the maximum time to start conversion.

ADM1	ADM0			Conversion Clock (f_{AD})	Conversion Start Time (Number of f_{CLK} Clock Cycles)	
ADLSP	FR2	FR1	FR0		Software Trigger No-wait Mode/ Hardware Trigger No-wait Mode	Software Trigger Wait Mode/ Hardware Trigger Wait Mode
0	0	0	0	f_{CLK}/32	31	1
0	0	0	1	f_{CLK}/16	15	1
0	0	1	0	f_{CLK}/8	7	1
0	0	1	1	f_{CLK}/4	3	1
0	1	0	0	f_{CLK}/2	1	1
0	1	0	1	f_{CLK}	1	1
1	0	1	1	f_{CLK}/4	3	1
1	1	0	0	f_{CLK}/2	1	1
1	1	0	1	f_{CLK}	1	1

<R> However, in the second and subsequent conversions in sequential conversion mode and in conversion of the channels specified as scan 1, 2, and 3 in scan mode, the A/D converter is already operating, so the A/D power supply will also be stable after detection of a hardware trigger.

Caution 1. If using the hardware trigger wait mode, setting the ADCS bit to 1 is prohibited (but the bit is automatically switched to 1 when the hardware trigger signal is detected). However, it is possible to clear the ADCS bit to 0 to specify the A/D conversion standby state.

Caution 2. While in the one-shot conversion mode of the hardware trigger no-wait mode, the ADCS bit is not automatically cleared to 0 when A/D conversion is complete. Instead, 1 is retained.

Caution 3. Only rewrite the value of the ADCE bit when ADCS = 0 (while in the conversion stopped/conversion standby state).

Caution 4. To complete A/D conversion, specify at least the following time as the hardware trigger interval:
Hardware trigger no-wait mode: 2 f_{CLK} clock cycles + conversion start time + A/D conversion time
Hardware trigger wait mode: 2 f_{CLK} clock cycles + conversion start time + A/D power supply stabilization wait time + A/D conversion time + 5 µs

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 12 - 3 A/D Conversion Time Selection (1/8)

<R> 1. A/D conversion starts regardless of the state in terms of A/D power supply stability.
Normal modes 1 and 2 in software trigger no-wait mode with the select mode selected and hardware trigger no-wait mode with the select mode selected

A/D Converter Mode Register 0 A/D Converter Mode Register 1						Mode	Conversion Clock (f_{AD})	Number of Clock Cycles for Conversion Start Delay	Number of Clock Cycles for Conversion	Number of Clock Cycles for Interrupt Output Delay	A/D Conversion Time (Conversion Start Delay Time + Conversion Time + Interrupt Output Delay Time)					
(ADM1)	(ADM0)											2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V				
ADLSP	FR2	FR1	FR0	LV1	LV0							f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	0	0	0	Normal 1	f_{CLK}/32	1 f_{AD}	64 f_{AD}	1 f_{AD}	2112/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	66 μs
0	0	0	1				f_{CLK}/16	1 f_{AD}	64 f_{AD}	1 f_{AD}	1056/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	66 μs	33 μs
0	0	1	0				f_{CLK}/8	1 f_{AD}	64 f_{AD}	1 f_{AD}	528/f_{CLK}	Setting prohibited	Setting prohibited	66 μs	33 μs	16.5 μs
0	0	1	1				f_{CLK}/4	1 f_{AD}	64 f_{AD}	1 f_{AD}	264/f_{CLK}	Setting prohibited	Setting prohibited	33 μs	16.5 μs	8.25 μs
0	1	0	0				f_{CLK}/2	1 f_{AD}	64 f_{AD}	1 f_{AD}	132/f_{CLK}	Setting prohibited	Setting prohibited	16.5 μs	8.25 μs	4.125 μs
0	1	0	1				f_{CLK}	1 f_{AD}	64 f_{AD}	1 f_{AD}	66/f_{CLK}	Setting prohibited	Setting prohibited	8.25 μs	4.125 μs	2.0625 μs
1	0	1	1				f_{CLK}/4	1 f_{AD}	64 f_{AD}	1 f_{AD}	264/f_{CLK}	Setting prohibited	66 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	1 f_{AD}	64 f_{AD}	1 f_{AD}	132/f_{CLK}	Setting prohibited	33 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	1 f_{AD}	64 f_{AD}	1 f_{AD}	66/f_{CLK}	66 μs	16.5 μs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									
0	0	0	0	0	1	Normal 2	f_{CLK}/32	1 f_{AD}	181 f_{AD}	1 f_{AD}	5856/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	183 μs
0	0	0	1				f_{CLK}/16	1 f_{AD}	181 f_{AD}	1 f_{AD}	2928/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	183 μs	91.5 μs
0	0	1	0				f_{CLK}/8	1 f_{AD}	181 f_{AD}	1 f_{AD}	1464/f_{CLK}	Setting prohibited	Setting prohibited	183 μs	91.5 μs	45.75 μs
0	0	1	1				f_{CLK}/4	1 f_{AD}	181 f_{AD}	1 f_{AD}	732/f_{CLK}	Setting prohibited	Setting prohibited	91.5 μs	45.75 μs	22.875 μs
0	1	0	0				f_{CLK}/2	1 f_{AD}	181 f_{AD}	1 f_{AD}	366/f_{CLK}	Setting prohibited	Setting prohibited	45.75 μs	22.875 μs	11.4375 μs
0	1	0	1				f_{CLK}	1 f_{AD}	181 f_{AD}	1 f_{AD}	183/f_{CLK}	Setting prohibited	Setting prohibited	22.875 μs	11.4375 μs	5.71875 μs
1	0	1	1				f_{CLK}/4	1 f_{AD}	181 f_{AD}	1 f_{AD}	732/f_{CLK}	Setting prohibited	183 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	1 f_{AD}	181 f_{AD}	1 f_{AD}	366/f_{CLK}	Setting prohibited	91.5 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	1 f_{AD}	181 f_{AD}	1 f_{AD}	183/f_{CLK}	183 μs	45.75 μs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									

Caution 1. The A/D conversion time must be selected within the relevant ranges of the conversion clock (f_{AD}) and conversion times (t_{CONV}) described in 37.6.1 Characteristics of the A/D converter.

Caution 2. Rewrite the FR2 to FR0, LV1, and LV0 bits to different values while conversion is stopped (ADCS = 0, ADCE = 0). When conversion is to be stopped while the A/D converter is on standby or is operating, wait for at least 0.2 μs before setting bits FR2 to FR0, LV1, and LV0.

Caution 3. The above conversion times do not include the conversion start time. Add the conversion start time to obtain the time for the first conversion. The conversion times also do not take clock frequency errors into account. Consider clock frequency errors when selecting conversion times.

Caution 4. Use normal mode 2 when the internal reference voltage or temperature sensor output voltage is selected as the target for A/D conversion.

Caution 5. When the internal reference voltage is selected as the positive reference voltage, normal modes 1 and 2 cannot be used. Use low-voltage mode 1 or 2.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 12 - 3 A/D Conversion Time Selection (2/8)

<R> 2. A/D conversion starts regardless of the state in terms of A/D power supply stability.
Low-voltage modes 1 and 2 in software trigger no-wait mode with the select mode selected and hardware trigger no-wait mode with the select mode selected

A/D Converter Mode Register 0 A/D Converter Mode Register 1						Mode	Conversion Clock (f_{AD})	Number of Clock Cycles for Conversion Start Delay	Number of Clock Cycles for Conversion	Number of Clock Cycles for Interrupt Output Delay	A/D Conversion Time (Conversion Start Delay Time + Conversion Time + Interrupt Output Delay Time)				
(ADM1)	(ADM0)											1.6 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	1.6 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	1.8 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	2.7 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V
ADLSP	FR2	FR1	FR0	LV1	LV0							f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	0	1	0	Low voltage 1	$f_{CLK}/32$	1 f_{AD}	80 f_{AD}	1 f_{AD}	2624/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	82 μs
0	0	0	1				$f_{CLK}/16$	1 f_{AD}	80 f_{AD}	1 f_{AD}	1312/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	82 μs	41 μs
0	0	1	0				$f_{CLK}/8$	1 f_{AD}	80 f_{AD}	1 f_{AD}	656/f_{CLK}	Setting prohibited	Setting prohibited	82 μs	41 μs	20.5 μs
0	0	1	1				$f_{CLK}/4$	1 f_{AD}	80 f_{AD}	1 f_{AD}	328/f_{CLK}	Setting prohibited	Setting prohibited	41 μs	20.5 μs	10.25 μs
0	1	0	0				$f_{CLK}/2$	1 f_{AD}	80 f_{AD}	1 f_{AD}	164/f_{CLK}	Setting prohibited	Setting prohibited	20.5 μs	10.25 μs	5.125 μs
0	1	0	1				f_{CLK}	1 f_{AD}	80 f_{AD}	1 f_{AD}	82/f_{CLK}	Setting prohibited	Setting prohibited	10.25 μs	5.125 μs	Setting prohibited
1	0	1	1				$f_{CLK}/4$	1 f_{AD}	80 f_{AD}	1 f_{AD}	328/f_{CLK}	Setting prohibited	82 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				$f_{CLK}/2$	1 f_{AD}	80 f_{AD}	1 f_{AD}	164/f_{CLK}	Setting prohibited	41 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	1 f_{AD}	80 f_{AD}	1 f_{AD}	82/f_{CLK}	82 μs	20.5 μs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									
0	0	0	0	1	1	Low voltage 2	$f_{CLK}/32$	1 f_{AD}	107 f_{AD}	1 f_{AD}	3488/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	109 μs
0	0	0	1				$f_{CLK}/16$	1 f_{AD}	107 f_{AD}	1 f_{AD}	1744/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	109 μs	54.5 μs
0	0	1	0				$f_{CLK}/8$	1 f_{AD}	107 f_{AD}	1 f_{AD}	872/f_{CLK}	Setting prohibited	Setting prohibited	109 μs	54.5 μs	27.25 μs
0	0	1	1				$f_{CLK}/4$	1 f_{AD}	107 f_{AD}	1 f_{AD}	436/f_{CLK}	Setting prohibited	Setting prohibited	54.5 μs	27.25 μs	13.625 μs
0	1	0	0				$f_{CLK}/2$	1 f_{AD}	107 f_{AD}	1 f_{AD}	218/f_{CLK}	Setting prohibited	Setting prohibited	27.25 μs	13.625 μs	6.8125 μs
0	1	0	1				f_{CLK}	1 f_{AD}	107 f_{AD}	1 f_{AD}	109/f_{CLK}	Setting prohibited	Setting prohibited	13.625 μs	6.8125 μs	Setting prohibited
1	0	1	1				$f_{CLK}/4$	1 f_{AD}	107 f_{AD}	1 f_{AD}	436/f_{CLK}	Setting prohibited	109 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				$f_{CLK}/2$	1 f_{AD}	107 f_{AD}	1 f_{AD}	218/f_{CLK}	Setting prohibited	54.5 μs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	1 f_{AD}	107 f_{AD}	1 f_{AD}	109/f_{CLK}	109 μs	27.25 μs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									

Caution 1. The A/D conversion time must be selected within the relevant ranges of the conversion clock (f_{AD}) and conversion times (t_{CONV}) described in 37.6.1 Characteristics of the A/D converter.

Caution 2. Rewrite the FR2 to FR0, LV1, and LV0 bits to different values while conversion is stopped (ADCS = 0, ADCE = 0). When conversion is to be stopped while the A/D converter is on standby or is operating, wait for at least 0.2 μs before setting bits FR2 to FR0, LV1, and LV0.

Caution 3. The above conversion times do not include the conversion start time. Add the conversion start time to obtain the time for the first conversion. The conversion times also do not take clock frequency errors into account. Consider clock frequency errors when selecting conversion times.

(Cautions and Remark are listed on the next page.)

Caution 4. When the internal reference voltage or temperature sensor output voltage is selected as the target for A/D conversion, use low-voltage mode 2 and use a conversion clock (f_{AD}) with a frequency no greater than 16 MHz.

Caution 5. When the internal reference voltage is selected as the positive reference voltage, the conversion clock (f_{AD}) must be in the range from 1 to 2 MHz.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 12 - 3 A/D Conversion Time Selection (3/8)

<R> 3. The stabilization wait time for the A/D power supply is automatically secured before A/D conversion starts.
Normal modes 1 and 2 in software trigger wait mode with the select mode selected and hardware trigger wait mode with the select mode selected**Note 1**

A/D Converter Mode Register 0 A/D Converter Mode Register 1						Mode	Conversion Clock (f_{AD})	Number of Clock Cycles for A/D Power Supply Stabilization Wait	Number of Clock Cycles for Conversion	Number of Clock Cycles for Interrupt Output Delay **Note 2**	A/D Conversion Time (A/D Power Supply Stabilization Wait Time + Conversion Time + Interrupt Output Delay Time)					
(ADM1)	(ADM0)											2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V				
ADLSP	FR2	FR1	FR0	LV1	LV0							f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	0	0	0	Normal 1	f_{CLK}/32	4 f_{AD}	64 f_{AD}	4 f_{AD}	2304/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	72 µs
0	0	0	1				f_{CLK}/16	4 f_{AD}	64 f_{AD}	4 f_{AD}	1152/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	72 µs	36 µs
0	0	1	0				f_{CLK}/8	6 f_{AD}	64 f_{AD}	4 f_{AD}	592/f_{CLK}	Setting prohibited	Setting prohibited	74 µs	37 µs	18.5 µs
0	0	1	1				f_{CLK}/4	10 f_{AD}	64 f_{AD}	4 f_{AD}	312/f_{CLK}	Setting prohibited	Setting prohibited	39 µs	19.5 µs	9.75 µs
0	1	0	0				f_{CLK}/2	18 f_{AD}	64 f_{AD}	4 f_{AD}	172/f_{CLK}	Setting prohibited	Setting prohibited	21.5 µs	10.75 µs	5.375 µs
0	1	0	1				f_{CLK}	34 f_{AD}	64 f_{AD}	4 f_{AD}	102/f_{CLK}	Setting prohibited	Setting prohibited	12.75 µs	6.375 µs	3.1875 µs
1	0	1	1				f_{CLK}/4	4 f_{AD}	64 f_{AD}	4 f_{AD}	288/f_{CLK}	Setting prohibited	72 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	4 f_{AD}	64 f_{AD}	4 f_{AD}	144/f_{CLK}	Setting prohibited	36 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	6 f_{AD}	64 f_{AD}	4 f_{AD}	74/f_{CLK}	74 µs	18.5 µs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									
0	0	0	0	0	1	Normal 2	f_{CLK}/32	4 f_{AD}	181 f_{AD}	4 f_{AD}	6048/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	189 µs
0	0	0	1				f_{CLK}/16	4 f_{AD}	181 f_{AD}	4 f_{AD}	3024/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	189 µs	94.5 µs
0	0	1	0				f_{CLK}/8	6 f_{AD}	181 f_{AD}	4 f_{AD}	1528/f_{CLK}	Setting prohibited	Setting prohibited	191 µs	95.5 µs	47.75 µs
0	0	1	1				f_{CLK}/4	10 f_{AD}	181 f_{AD}	4 f_{AD}	780/f_{CLK}	Setting prohibited	Setting prohibited	97.5 µs	48.75 µs	24.375 µs
0	1	0	0				f_{CLK}/2	18 f_{AD}	181 f_{AD}	4 f_{AD}	406/f_{CLK}	Setting prohibited	Setting prohibited	50.75 µs	25.375 µs	12.6875 µs
0	1	0	1				f_{CLK}	34 f_{AD}	181 f_{AD}	4 f_{AD}	219/f_{CLK}	Setting prohibited	Setting prohibited	27.375 µs	13.6875 µs	6.84375 µs
1	0	1	1				f_{CLK}/4	4 f_{AD}	181 f_{AD}	4 f_{AD}	756/f_{CLK}	Setting prohibited	189 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	4 f_{AD}	181 f_{AD}	4 f_{AD}	378/f_{CLK}	Setting prohibited	94.5 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	6 f_{AD}	181 f_{AD}	4 f_{AD}	191/f_{CLK}	191 µs	47.75 µs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									

<R> **Note 1.** In the second and subsequent conversions in sequential conversion mode, the A/D converter is already operating, so the A/D power supply will also be stable. Accordingly, the A/D conversion times after detection of a hardware trigger in such cases are the same as those listed in **Table 12 - 3 A/D Conversion Time Selection (1/8)**.

Note 2. The value in this column is applicable when the one-shot conversion mode is selected. When the sequential conversion mode is selected, the number of clock cycles is shortened by 3 cycles of the conversion clock (f_{AD}).

<R> **Caution 1. The A/D conversion time must be selected within the relevant ranges of the conversion clock (f_{AD}) and conversion times (t_{CONV}) described in 37.6.1 Characteristics of the A/D converter. Note that the conversion time (t_{CONV}) does not include the A/D power supply stabilization wait time.**

(Cautions and Remark are listed on the next page.)

Caution 2. **Rewrite the FR2 to FR0, LV1, and LV0 bits to different values while conversion is stopped (ADCS = 0, ADCE = 0). When conversion is to be stopped while the A/D converter is on standby or is operating, wait for at least 0.2 μs before setting bits FR2 to FR0, LV1, and LV0.**

Caution 3. **The above conversion times do not include the conversion start time. Add the conversion start time to obtain the time for the first conversion. The conversion times also do not take clock frequency errors into account. Consider clock frequency errors when selecting conversion times.**

Caution 4. **The conversion times in hardware trigger wait mode include the A/D power supply stabilization wait time from the time the hardware trigger is detected. The conversion times in software trigger wait mode include the A/D power supply stabilization wait time from the time the ADCS bit is set to 1.**

Caution 5. **Use normal mode 2 when the internal reference voltage or temperature sensor output voltage is selected as the target for A/D conversion.**

Caution 6. **When the internal reference voltage is selected as the positive reference voltage, normal modes 1 and 2 cannot be used. Use low-voltage mode 1 or 2.**

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 12 - 3 A/D Conversion Time Selection (4/8)

<R> 4. The stabilization wait time for the A/D power supply is automatically secured before A/D conversion starts.
Low-voltage modes 1 and 2 in software trigger wait mode with the select mode selected and hardware trigger wait mode with the select mode selected**Note 1**

A/D Converter Mode Register 0 A/D Converter Mode Register 1						Mode	Conversion Clock (f_{AD})	Number of Clock Cycles for A/D Power Supply Stabilization Wait	Number of Clock Cycles for Conversion	Number of Clock Cycles for Interrupt Output Delay **Note 2**	A/D Conversion Time (A/D Power Supply Stabilization Wait Time + Conversion Time + Interrupt Output Delay Time)				
(ADM1)	(ADM0)											1.6 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	1.6 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	1.8 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	2.7 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V
ADLSP	FR2	FR1	FR0	LV1	LV0							f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	0	1	0	Low voltage 1	f_{CLK}/32	4 f_{AD}	80 f_{AD}	4 f_{AD}	2816/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	88 µs
0	0	0	1				f_{CLK}/16	4 f_{AD}	80 f_{AD}	4 f_{AD}	1408/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	88 µs	44 µs
0	0	1	0				f_{CLK}/8	6 f_{AD}	80 f_{AD}	4 f_{AD}	720/f_{CLK}	Setting prohibited	Setting prohibited	90 µs	45 µs	22.5 µs
0	0	1	1				f_{CLK}/4	10 f_{AD}	80 f_{AD}	4 f_{AD}	376/f_{CLK}	Setting prohibited	Setting prohibited	47 µs	23.5 µs	11.75 µs
0	1	0	0				f_{CLK}/2	18 f_{AD}	80 f_{AD}	4 f_{AD}	204/f_{CLK}	Setting prohibited	Setting prohibited	25.5 µs	12.75 µs	6.375 µs
0	1	0	1				f_{CLK}	34 f_{AD}	80 f_{AD}	4 f_{AD}	118/f_{CLK}	Setting prohibited	Setting prohibited	14.75 µs	7.375 µs	Setting prohibited
1	0	1	1				f_{CLK}/4	4 f_{AD}	80 f_{AD}	4 f_{AD}	352/f_{CLK}	Setting prohibited	88 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	4 f_{AD}	80 f_{AD}	4 f_{AD}	176/f_{CLK}	Setting prohibited	44 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	6 f_{AD}	80 f_{AD}	4 f_{AD}	90/f_{CLK}	90 µs	22.5 µs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									
0	0	0	0	1	1	Low voltage 2	f_{CLK}/32	4 f_{AD}	107 f_{AD}	4 f_{AD}	3680/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	115 µs
0	0	0	1				f_{CLK}/16	4 f_{AD}	107 f_{AD}	4 f_{AD}	1840/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	115 µs	57.5 µs
0	0	1	0				f_{CLK}/8	6 f_{AD}	107 f_{AD}	4 f_{AD}	936/f_{CLK}	Setting prohibited	Setting prohibited	117 µs	58.5 µs	29.25 µs
0	0	1	1				f_{CLK}/4	10 f_{AD}	107 f_{AD}	4 f_{AD}	484/f_{CLK}	Setting prohibited	Setting prohibited	60.5 µs	30.25 µs	15.125 µs
0	1	0	0				f_{CLK}/2	18 f_{AD}	107 f_{AD}	4 f_{AD}	258/f_{CLK}	Setting prohibited	Setting prohibited	32.25 µs	16.125 µs	8.0625 µs
0	1	0	1				f_{CLK}	34 f_{AD}	107 f_{AD}	4 f_{AD}	145/f_{CLK}	Setting prohibited	Setting prohibited	18.125 µs	9.0625 µs	Setting prohibited
1	0	1	1				f_{CLK}/4	4 f_{AD}	107 f_{AD}	4 f_{AD}	460/f_{CLK}	Setting prohibited	115 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	4 f_{AD}	107 f_{AD}	4 f_{AD}	230/f_{CLK}	Setting prohibited	57.5 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	6 f_{AD}	107 f_{AD}	4 f_{AD}	117/f_{CLK}	117 µs	29.25 µs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									

<R> **Note 1.** In the second and subsequent conversions in sequential conversion mode, the A/D converter is already operating, so the A/D power supply will also be stable. Accordingly, the A/D conversion times after detection of a hardware trigger in such cases are the same as those listed in **Table 12 - 3 A/D Conversion Time Selection (2/8)**.

Note 2. The value in this column is applicable when the one-shot conversion mode is selected. When the sequential conversion mode is selected, the number of clock cycles is shortened by 3 cycles of the conversion clock (f_{AD}).

<R> **Caution 1. The A/D conversion time must be selected within the relevant ranges of the conversion clock (f_{AD}) and conversion times (t_{CONV}) described in 37.6.1 Characteristics of the A/D converter. Note that the conversion time (t_{CONV}) does not include the A/D power supply stabilization wait time.**

(Cautions and Remark are listed on the next page.)

Caution 2. **Rewrite the FR2 to FR0, LV1, and LV0 bits to different values while conversion is stopped (ADCS = 0, ADCE = 0). When conversion is to be stopped while the A/D converter is on standby or is operating, wait for at least 0.2 μs before setting bits FR2 to FR0, LV1, and LV0.**

Caution 3. **The above conversion times do not include the conversion start time. Add the conversion start time to obtain the time for the first conversion. The conversion times also do not take clock frequency errors into account. Consider clock frequency errors when selecting conversion times.**

Caution 4. **The conversion times in hardware trigger wait mode include the A/D power supply stabilization wait time from the time the hardware trigger is detected. The conversion times in software trigger wait mode include the A/D power supply stabilization wait time from the time the ADCS bit is set to 1.**

Caution 5. **When the internal reference voltage or temperature sensor output voltage is selected as the target for A/D conversion, use low-voltage mode 2 and use a conversion clock (f_{AD}) with a frequency no greater than 16 MHz.**

Caution 6. **When the internal reference voltage is selected as the positive reference voltage, the conversion clock (f_{AD}) must be in the range from 1 to 2 MHz.**

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 12 - 3 A/D Conversion Time Selection (5/8)

<R> 5. A/D conversion starts regardless of the state in terms of A/D power supply stability.
Normal modes 1 and 2 in software trigger no-wait mode with the scan mode selected and hardware trigger no-wait mode with the scan mode selected

A/D Converter Mode Register 0 A/D Converter Mode Register 1						Mode	Conversion Clock (f_{AD})	Number of Clock Cycles for Conversion Start Delay	Number of Clock Cycles for Conversion	Number of Clock Cycles for Interrupt Output Delay	A/D Conversion Time (Conversion Start Delay Time + Conversion Time × 4 + Interrupt Output Delay Time)				
(ADM1)	(ADM0)											2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V				
ADLSP	FR2	FR1	FR0	LV1	LV0							f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	0	0	0	Normal 1	f_{CLK}/32	1 f_{AD}	64 f_{AD}	1 f_{AD}	8256/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	258 µs
0	0	0	1				f_{CLK}/16	1 f_{AD}	64 f_{AD}	1 f_{AD}	4128/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	258 µs	129 µs
0	0	1	0				f_{CLK}/8	1 f_{AD}	64 f_{AD}	1 f_{AD}	2064/f_{CLK}	Setting prohibited	Setting prohibited	258 µs	129 µs	64.5 µs
0	0	1	1				f_{CLK}/4	1 f_{AD}	64 f_{AD}	1 f_{AD}	1032/f_{CLK}	Setting prohibited	Setting prohibited	129 µs	64.5 µs	32.25 µs
0	1	0	0				f_{CLK}/2	1 f_{AD}	64 f_{AD}	1 f_{AD}	516/f_{CLK}	Setting prohibited	Setting prohibited	64.5 µs	32.25 µs	16.125 µs
0	1	0	1				f_{CLK}	1 f_{AD}	64 f_{AD}	1 f_{AD}	258/f_{CLK}	Setting prohibited	Setting prohibited	32.25 µs	16.125 µs	8.0625 µs
1	0	1	1				f_{CLK}/4	1 f_{AD}	64 f_{AD}	1 f_{AD}	1032/f_{CLK}	Setting prohibited	258 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	1 f_{AD}	64 f_{AD}	1 f_{AD}	516/f_{CLK}	Setting prohibited	129 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	1 f_{AD}	64 f_{AD}	1 f_{AD}	258/f_{CLK}	258 µs	64.5 µs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									
0	0	0	0	0	1	Normal 2	f_{CLK}/32	1 f_{AD}	181 f_{AD}	1 f_{AD}	23232/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	726 µs
0	0	0	1				f_{CLK}/16	1 f_{AD}	181 f_{AD}	1 f_{AD}	11616/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	726 µs	363 µs
0	0	1	0				f_{CLK}/8	1 f_{AD}	181 f_{AD}	1 f_{AD}	5808/f_{CLK}	Setting prohibited	Setting prohibited	726 µs	363 µs	181.5 µs
0	0	1	1				f_{CLK}/4	1 f_{AD}	181 f_{AD}	1 f_{AD}	2904/f_{CLK}	Setting prohibited	Setting prohibited	363 µs	181.5 µs	90.75 µs
0	1	0	0				f_{CLK}/2	1 f_{AD}	181 f_{AD}	1 f_{AD}	1452/f_{CLK}	Setting prohibited	Setting prohibited	181.5 µs	90.75 µs	45.375 µs
0	1	0	1				f_{CLK}	1 f_{AD}	181 f_{AD}	1 f_{AD}	726/f_{CLK}	Setting prohibited	Setting prohibited	90.75 µs	45.375 µs	22.6875 µs
1	0	1	1				f_{CLK}/4	1 f_{AD}	181 f_{AD}	1 f_{AD}	2904/f_{CLK}	Setting prohibited	726 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	1 f_{AD}	181 f_{AD}	1 f_{AD}	1452/f_{CLK}	Setting prohibited	363 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	1 f_{AD}	181 f_{AD}	1 f_{AD}	726/f_{CLK}	726 µs	181.5 µs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									

Caution 1. The A/D conversion time must be selected within the relevant ranges of the conversion clock (f_{AD}) and conversion times (t_{CONV}) described in 37.6.1 Characteristics of the A/D converter.

Caution 2. Rewrite the FR2 to FR0, LV1, and LV0 bits to different values while conversion is stopped (ADCS = 0, ADCE = 0). When conversion is to be stopped while the A/D converter is on standby or is operating, wait for at least 0.2 µs before setting bits FR2 to FR0, LV1, and LV0.

Caution 3. The above conversion times do not include the conversion start time. Add the conversion start time to obtain the time for the first conversion. The conversion times also do not take clock frequency errors into account. Consider clock frequency errors when selecting conversion times.

Caution 4. Use normal mode 2 when the internal reference voltage or temperature sensor output voltage is selected as the target for A/D conversion.

Caution 5. When the internal reference voltage is selected as the positive reference voltage, normal modes 1 and 2 cannot be used. Use low-voltage mode 1 or 2.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 12 - 3 A/D Conversion Time Selection (6/8)

<R> 6. A/D conversion starts regardless of the state in terms of A/D power supply stability.
Low-voltage modes 1 and 2 in software trigger no-wait mode with the scan mode selected and hardware trigger no-wait mode with the scan mode selected

A/D Converter Mode Register 0 A/D Converter Mode Register 1						Mode	Conversion Clock (f_{AD})	Number of Clock Cycles for Conversion Start Delay	Number of Clock Cycles for Conversion	Number of Clock Cycles for Interrupt Output Delay	A/D Conversion Time (Conversion Start Delay Time + Conversion Time × 4 + Interrupt Output Delay Time)				
(ADM1)	(ADM0)											1.6 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	1.6 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	1.8 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	2.7 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V
ADLSP	FR2	FR1	FR0	LV1	LV0							f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	0	1	0	Low voltage 1	$f_{CLK}/32$	1 f_{AD}	80 f_{AD}	1 f_{AD}	10304/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	322 µs
0	0	0	1				$f_{CLK}/16$	1 f_{AD}	80 f_{AD}	1 f_{AD}	5152/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	322 µs	161 µs
0	0	1	0				$f_{CLK}/8$	1 f_{AD}	80 f_{AD}	1 f_{AD}	2576/f_{CLK}	Setting prohibited	Setting prohibited	322 µs	161 µs	80.5 µs
0	0	1	1				$f_{CLK}/4$	1 f_{AD}	80 f_{AD}	1 f_{AD}	1288/f_{CLK}	Setting prohibited	Setting prohibited	161 µs	80.5 µs	40.25 µs
0	1	0	0				$f_{CLK}/2$	1 f_{AD}	80 f_{AD}	1 f_{AD}	644/f_{CLK}	Setting prohibited	Setting prohibited	80.5 µs	40.25 µs	20.125 µs
0	1	0	1				f_{CLK}	1 f_{AD}	80 f_{AD}	1 f_{AD}	322/f_{CLK}	Setting prohibited	Setting prohibited	40.25 µs	20.125 µs	Setting prohibited
1	0	1	1				$f_{CLK}/4$	1 f_{AD}	80 f_{AD}	1 f_{AD}	1288/f_{CLK}	Setting prohibited	322 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				$f_{CLK}/2$	1 f_{AD}	80 f_{AD}	1 f_{AD}	644/f_{CLK}	Setting prohibited	161 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	1 f_{AD}	80 f_{AD}	1 f_{AD}	322/f_{CLK}	322 µs	80.5 µs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									
0	0	0	0	1	1	Low voltage 2	$f_{CLK}/32$	1 f_{AD}	107 f_{AD}	1 f_{AD}	13760/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	430 µs
0	0	0	1				$f_{CLK}/16$	1 f_{AD}	107 f_{AD}	1 f_{AD}	6880/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	430 µs	215 µs
0	0	1	0				$f_{CLK}/8$	1 f_{AD}	107 f_{AD}	1 f_{AD}	3440/f_{CLK}	Setting prohibited	Setting prohibited	430 µs	215 µs	107.5 µs
0	0	1	1				$f_{CLK}/4$	1 f_{AD}	107 f_{AD}	1 f_{AD}	1720/f_{CLK}	Setting prohibited	Setting prohibited	215 µs	107.5 µs	53.75 µs
0	1	0	0				$f_{CLK}/2$	1 f_{AD}	107 f_{AD}	1 f_{AD}	860/f_{CLK}	Setting prohibited	Setting prohibited	107.5 µs	53.75 µs	26.875 µs
0	1	0	1				f_{CLK}	1 f_{AD}	107 f_{AD}	1 f_{AD}	430/f_{CLK}	Setting prohibited	Setting prohibited	53.75 µs	26.875 µs	Setting prohibited
1	0	1	1				$f_{CLK}/4$	1 f_{AD}	107 f_{AD}	1 f_{AD}	1720/f_{CLK}	Setting prohibited	430 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				$f_{CLK}/2$	1 f_{AD}	107 f_{AD}	1 f_{AD}	860/f_{CLK}	Setting prohibited	215 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	1 f_{AD}	107 f_{AD}	1 f_{AD}	430/f_{CLK}	430 µs	107.5 µs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									

Caution 1. The A/D conversion time must be selected within the relevant ranges of the conversion clock (f_{AD}) and conversion times (t_{CONV}) described in 37.6.1 Characteristics of the A/D converter.

Caution 2. Rewrite the FR2 to FR0, LV1, and LV0 bits to different values while conversion is stopped (ADCS = 0, ADCE = 0). When conversion is to be stopped while the A/D converter is on standby or is operating, wait for at least 0.2 µs before setting bits FR2 to FR0, LV1, and LV0.

Caution 3. The above conversion times do not include the conversion start time. Add the conversion start time to obtain the time for the first conversion. The conversion times also do not take clock frequency errors into account. Consider clock frequency errors when selecting conversion times.

(Cautions and Remark are listed on the next page.)

Caution 4. When the internal reference voltage or temperature sensor output voltage is selected as the target for A/D conversion, use low-voltage mode 2 and use a conversion clock (f_{AD}) with a frequency no greater than 16 MHz.

Caution 5. When the internal reference voltage is selected as the positive reference voltage, the conversion clock (f_{AD}) must be in the range from 1 to 2 MHz.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 12 - 3 A/D Conversion Time Selection (7/8)

<R> 7. The stabilization wait time for the A/D power supply is automatically secured before A/D conversion starts.
Normal modes 1 and 2 in software trigger wait mode with the scan mode selected and hardware trigger wait mode with the scan mode selected**Note 1**

A/D Converter Mode Register 0 A/D Converter Mode Register 1						Mode	Conversion Clock (f_{AD})	Number of Clock Cycles for A/D Power Supply Stabilization Wait	Number of Clock Cycles for Conversion	Number of Clock Cycles for Interrupt Output Delay **Note 2**	A/D Conversion Time (A/D Power Supply Stabilization Wait Time + Conversion Time × 4 + Interrupt Output Delay Time)					
(ADM1)	(ADM0)											2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V				
ADLSP	FR2	FR1	FR0	LV1	LV0							f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	0	0	0	Normal 1	$f_{CLK}/32$	4 f_{AD}	64 f_{AD}	4 f_{AD}	8448/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	264 µs
0	0	0	1				$f_{CLK}/16$	4 f_{AD}	64 f_{AD}	4 f_{AD}	4224/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	264 µs	132 µs
0	0	1	0				$f_{CLK}/8$	6 f_{AD}	64 f_{AD}	4 f_{AD}	2128/f_{CLK}	Setting prohibited	Setting prohibited	266 µs	133 µs	66.5 µs
0	0	1	1				$f_{CLK}/4$	10 f_{AD}	64 f_{AD}	4 f_{AD}	1080/f_{CLK}	Setting prohibited	Setting prohibited	135 µs	67.5 µs	33.75 µs
0	1	0	0				$f_{CLK}/2$	18 f_{AD}	64 f_{AD}	4 f_{AD}	556/f_{CLK}	Setting prohibited	Setting prohibited	69.5 µs	34.75 µs	17.375 µs
0	1	0	1				f_{CLK}	34 f_{AD}	64 f_{AD}	4 f_{AD}	294/f_{CLK}	Setting prohibited	Setting prohibited	36.75 µs	18.375 µs	9.1875 µs
1	0	1	1				$f_{CLK}/4$	4 f_{AD}	64 f_{AD}	4 f_{AD}	1056/f_{CLK}	1056 µs	264 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				$f_{CLK}/2$	4 f_{AD}	64 f_{AD}	4 f_{AD}	528/f_{CLK}	528 µs	132 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	6 f_{AD}	64 f_{AD}	4 f_{AD}	266/f_{CLK}	266 µs	66.5 µs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									
0	0	0	0	0	1	Normal 2	$f_{CLK}/32$	4 f_{AD}	181 f_{AD}	4 f_{AD}	23424/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	732 µs
0	0	0	1				$f_{CLK}/16$	4 f_{AD}	181 f_{AD}	4 f_{AD}	11712/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	732 µs	366 µs
0	0	1	0				$f_{CLK}/8$	6 f_{AD}	181 f_{AD}	4 f_{AD}	5872/f_{CLK}	Setting prohibited	Setting prohibited	734 µs	367 µs	183.5 µs
0	0	1	1				$f_{CLK}/4$	10 f_{AD}	181 f_{AD}	4 f_{AD}	2952/f_{CLK}	Setting prohibited	Setting prohibited	369 µs	184.5 µs	92.25 µs
0	1	0	0				$f_{CLK}/2$	18 f_{AD}	181 f_{AD}	4 f_{AD}	1492/f_{CLK}	Setting prohibited	Setting prohibited	186.5 µs	93.25 µs	46.625 µs
0	1	0	1				f_{CLK}	34 f_{AD}	181 f_{AD}	4 f_{AD}	762/f_{CLK}	Setting prohibited	Setting prohibited	95.25 µs	47.625 µs	23.8125 µs
1	0	1	1				$f_{CLK}/4$	4 f_{AD}	181 f_{AD}	4 f_{AD}	2928/f_{CLK}	Setting prohibited	732 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				$f_{CLK}/2$	4 f_{AD}	181 f_{AD}	4 f_{AD}	1464/f_{CLK}	Setting prohibited	366 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	6 f_{AD}	181 f_{AD}	4 f_{AD}	734/f_{CLK}	734 µs	183.5 µs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									

<R> **Note 1.** In conversion of the channels specified as scan 1, 2, and 3 in scan mode, the A/D converter is already operating, so the A/D power supply will also be stable. Accordingly, the A/D conversion times after detection of a hardware trigger in such cases are the same as those listed in **Table 12 - 3 A/D Conversion Time Selection (5/8)**.

Note 2. The value in this column is applicable when the one-shot conversion mode is selected. When the sequential conversion mode is selected, the number of clock cycles is shortened by 3 cycles of the conversion clock (f_{AD}).

<R> **Caution 1. The A/D conversion time must be selected within the relevant ranges of the conversion clock (f_{AD}) and conversion times (t_{CONV}) described in 37.6.1 Characteristics of the A/D converter. Note that the conversion time (t_{CONV}) does not include the A/D power supply stabilization wait time.**

(Cautions and Remark are listed on the next page.)

Caution 2. **Rewrite the FR2 to FR0, LV1, and LV0 bits to different values while conversion is stopped (ADCS = 0, ADCE = 0). When conversion is to be stopped while the A/D converter is on standby or is operating, wait for at least 0.2 μs before setting bits FR2 to FR0, LV1, and LV0.**

Caution 3. **The above conversion times do not include the conversion start time. Add the conversion start time to obtain the time for the first conversion. The conversion times also do not take clock frequency errors into account. Consider clock frequency errors when selecting conversion times.**

Caution 4. **The conversion times in hardware trigger wait mode include the A/D power supply stabilization wait time from the time the hardware trigger is detected. The conversion times in software trigger wait mode include the A/D power supply stabilization wait time from the time the ADCS bit is set to 1.**

Caution 5. **Use normal mode 2 when the internal reference voltage or temperature sensor output voltage is selected as the target for A/D conversion.**

Caution 6. **When the internal reference voltage is selected as the positive reference voltage, normal modes 1 and 2 cannot be used. Use low-voltage mode 1 or 2.**

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 12 - 3 A/D Conversion Time Selection (8/8)

<R> 8. The stabilization wait time for the A/D power supply is automatically secured before A/D conversion starts.
Low-voltage modes 1 and 2 in software trigger wait mode with the scan mode selected and hardware trigger wait mode with the scan mode selected**Note 1**

A/D Converter Mode Register 0 A/D Converter Mode Register 1						Mode	Conversion Clock (f_{AD})	Number of Clock Cycles for A/D Power Supply Stabilization Wait	Number of Clock Cycles for Conversion	Number of Clock Cycles for Interrupt Output Delay **Note 2**	A/D Conversion Time (A/D Power Supply Stabilization Wait Time + Conversion Time × 4 + Interrupt Output Delay Time)				
(ADM1)	(ADM0)											1.6 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	1.6 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	1.8 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V	2.7 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V
ADLSP	FR2	FR1	FR0	LV1	LV0							f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	0	1	0	Low voltage 1	f_{CLK}/32	4 f_{AD}	80 f_{AD}	4 f_{AD}	10496/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	328 µs
0	0	0	1				f_{CLK}/16	4 f_{AD}	80 f_{AD}	4 f_{AD}	5248/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	328 µs	164 µs
0	0	1	0				f_{CLK}/8	6 f_{AD}	80 f_{AD}	4 f_{AD}	2640/f_{CLK}	Setting prohibited	Setting prohibited	330 µs	165 µs	82.5 µs
0	0	1	1				f_{CLK}/4	10 f_{AD}	80 f_{AD}	4 f_{AD}	1336/f_{CLK}	Setting prohibited	Setting prohibited	167 µs	83.5 µs	41.75 µs
0	1	0	0				f_{CLK}/2	18 f_{AD}	80 f_{AD}	4 f_{AD}	684/f_{CLK}	Setting prohibited	Setting prohibited	85.5 µs	42.75 µs	21.375 µs
0	1	0	1				f_{CLK}	34 f_{AD}	80 f_{AD}	4 f_{AD}	358/f_{CLK}	Setting prohibited	Setting prohibited	44.75 µs	22.375 µs	Setting prohibited
1	0	1	1				f_{CLK}/4	4 f_{AD}	80 f_{AD}	4 f_{AD}	1312/f_{CLK}	Setting prohibited	328 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	4 f_{AD}	80 f_{AD}	4 f_{AD}	656/f_{CLK}	Setting prohibited	164 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	6 f_{AD}	80 f_{AD}	4 f_{AD}	330/f_{CLK}	330 µs	82.5 µs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									
0	0	0	0	1	1	Low voltage 2	f_{CLK}/32	4 f_{AD}	107 f_{AD}	4 f_{AD}	13952/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	436 µs
0	0	0	1				f_{CLK}/16	4 f_{AD}	107 f_{AD}	4 f_{AD}	6976/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	436 µs	218 µs
0	0	1	0				f_{CLK}/8	6 f_{AD}	107 f_{AD}	4 f_{AD}	3504/f_{CLK}	Setting prohibited	Setting prohibited	438 µs	219 µs	109.5 µs
0	0	1	1				f_{CLK}/4	10 f_{AD}	107 f_{AD}	4 f_{AD}	1768/f_{CLK}	Setting prohibited	Setting prohibited	221 µs	110.5 µs	55.25 µs
0	1	0	0				f_{CLK}/2	18 f_{AD}	107 f_{AD}	4 f_{AD}	900/f_{CLK}	Setting prohibited	Setting prohibited	112.5 µs	56.25 µs	28.125 µs
0	1	0	1				f_{CLK}	34 f_{AD}	107 f_{AD}	4 f_{AD}	466/f_{CLK}	Setting prohibited	Setting prohibited	58.25 µs	29.125 µs	Setting prohibited
1	0	1	1				fCLK/4	4 f_{AD}	107 f_{AD}	4 f_{AD}	1744/f_{CLK}	Setting prohibited	436 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	0				f_{CLK}/2	4 f_{AD}	107 f_{AD}	4 f_{AD}	872/f_{CLK}	Setting prohibited	218 µs	Setting prohibited	Setting prohibited	Setting prohibited
1	1	0	1				f_{CLK}	6 f_{AD}	107 f_{AD}	4 f_{AD}	438/f_{CLK}	438 µs	109.5 µs	Setting prohibited	Setting prohibited	Setting prohibited
Other than the above							Setting prohibited									

<R> **Note 1.** In conversion of the channels specified as scan 1, 2, and 3 in scan mode, the A/D converter is already operating, so the A/D power supply will also be stable. Accordingly, the A/D conversion times after detection of a hardware trigger in such cases are the same as those listed in **Table 12 - 3 A/D Conversion Time Selection (6/8)**.

Note 2. The value in this column is applicable when the one-shot conversion mode is selected. When the sequential conversion mode is selected, the number of clock cycles is shortened by 3 cycles of the conversion clock (f_{AD}).

<R> **Caution 1. The A/D conversion time must be selected within the relevant ranges of the conversion clock (f_{AD}) and conversion times (t_{CONV}) described in 37.6.1 Characteristics of the A/D converter. Note that the conversion time (t_{CONV}) does not include the A/D power supply stabilization wait time.**

(Cautions and Remark are listed on the next page.)

Caution 2. **Rewrite the FR2 to FR0, LV1, and LV0 bits to different values while conversion is stopped (ADCS = 0, ADCE = 0). When conversion is to be stopped while the A/D converter is on standby or is operating, wait for at least 0.2 μs before setting bits FR2 to FR0, LV1, and LV0.**

Caution 3. **The above conversion times do not include the conversion start time. Add the conversion start time to obtain the time for the first conversion. The conversion times also do not take clock frequency errors into account. Consider clock frequency errors when selecting conversion times.**

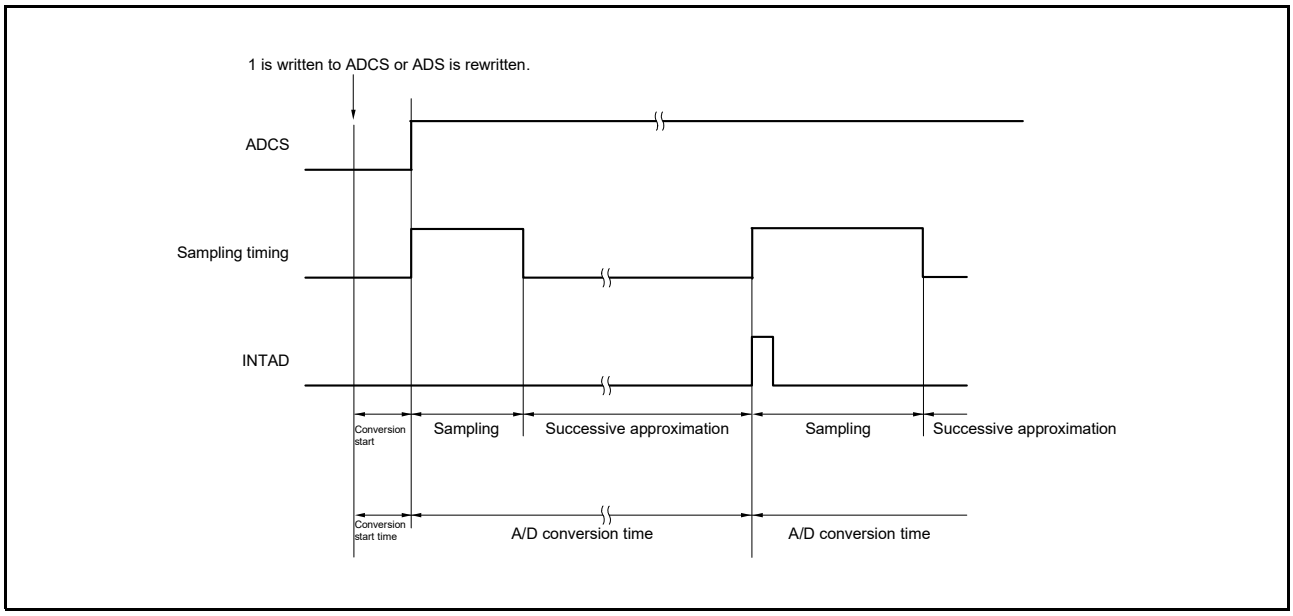
Caution 4. **The conversion times in hardware trigger wait mode include the A/D power supply stabilization wait time from the time the hardware trigger is detected. The conversion times in software trigger wait mode include the A/D power supply stabilization wait time from the time the ADCS bit is set to 1.**

Caution 5. **When the internal reference voltage or temperature sensor output voltage is selected as the target for A/D conversion, use low-voltage mode 2 and use a conversion clock (f_{AD}) with a frequency no greater than 16 MHz.**

Caution 6. **When the internal reference voltage is selected as the positive reference voltage, the conversion clock (f_{AD}) must be in the range from 1 to 2 MHz.**

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Figure 12 - 6 A/D Converter Sampling and A/D Conversion Timing for Software Trigger No-wait Mode



12.3.4 A/D converter mode register 1 (ADM1)

This register is used to specify the A/D conversion trigger, conversion mode, and hardware trigger signal. The ADM1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 12 - 7 Format of A/D Converter Mode Register 1 (ADM1)

Address: FFF32H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ADM1	ADTMD1	ADTMD0	ADSCM	0	ADLSP	ADTRS2	ADTRS1	ADTRS0

ADTMD1	ADTMD0	Selection of the A/D conversion trigger mode
0	×	Software trigger no-wait mode or software trigger wait mode
1	0	Hardware trigger no-wait mode
1	1	Hardware trigger wait mode

ADLSP	fCLK input frequency setting
0	4 MHz < fCLK ≤ 32 MHz
1	1 MHz ≤ fCLK ≤ 4 MHz

ADSCM	Specification of the A/D conversion mode
0	Sequential conversion mode
1	One-shot conversion mode

ADTRS2	ADTRS1	ADTRS0	Selection of the hardware trigger signal
0	0	0	End of counting or capture in channel 1 of timer array unit 0 interrupt signal (INTTM01)
0	1	0	Realtime clock interrupt signal (INTRTC)
0	1	1	32-bit interval timer interrupt signal (INTITL)
1	0	0	Event input from ELCL
Other than the above			Setting prohibited

Caution 1. Only rewrite the value of the ADM1 register while conversion operation is stopped (ADCS = 0, ADCE = 0).

Caution 2. To complete A/D conversion, specify at least the following time as the hardware trigger interval:
Hardware trigger no-wait mode: 2 fCLK clock cycles + conversion start time + A/D conversion time
Hardware trigger wait mode: 2 fCLK clock cycles + conversion start time + A/D power supply stabilization wait time + A/D conversion time + 5 μs

Caution 3. In modes other than the SNOOZE mode, input of the next INTRTC or INTITL will not be recognized as a valid hardware trigger for up to four fCLK cycles after the first INTRTC or INTITL is input.

Remark 1. ×: Don't care

Remark 2. fCLK: CPU/peripheral hardware clock frequency

12.3.5 A/D converter mode register 2 (ADM2)

This register is used to select the positive and negative reference voltages of the A/D converter, select the range of the A/D conversion results that causes interrupt generation, select the resolution, and specify whether to use the SNOOZE mode. The ADM2 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 12 - 8 Format of A/D Converter Mode Register 2 (ADM2) (1/2)

Address: F0010H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	<3>	<2>	<1>	<0>
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	ADTYP1	ADTYP0

ADREFP1	ADREFP0	Selection of the positive reference voltage source of the A/D converter
0	0	Supplied from V_{DD}
0	1	Supplied from P20/AV_{REFP}/ANI0
1	0	Supplied from the internal reference voltage**Note 1**
1	1	Discharged

- Use the following procedures to rewrite the ADREFP1 and ADREFP0 bits.
  <1> Set ADCE = 0.
  <2> Wait for at least 0.2 μs.
  <3> Set ADREFP1 and ADREFP0 to 11B.
  This step is only required when the value of ADREFP1 and ADREFP0 is changed to 10B.
  <4> Reference voltage discharge time: 1 μs
  This step is only required when the value of ADREFP1 and ADREFP0 is changed to 10B.
  <5> Change the value of the ADREFP1 and ADREFP0 bits.
  <6> Wait for reference voltage stabilization time A.
  <7> Set ADCE = 1.
  <8> Wait for reference voltage stabilization time B.

  Reference voltage stabilization times A and B
  Stabilization of the reference voltage will require the following waiting times; the duration of time A depends on the settings of ADREFP1 and ADREFP0.
  - ADREFP1 and ADREFP0 = 10B
    A = 5 μs
    B = 1 μs + 2 cycles of the conversion clock (f_{AD})
  - ADREFP1 and ADREFP0 = 00B or 01B
    A = 4.8 μs
    B = 1 μs + 2 cycles of the conversion clock (f_{AD})

  Note that the waiting time of A = 4.8 μs is not required when you set the reference voltage for the first time.

  After waiting for reference voltage stabilization time B, start the A/D conversion.
- When ADREFP1 and ADREFP0 are set to 10B, the temperature sensor output voltage and internal reference voltage**Note 1** cannot be A/D converted. Be sure to proceed with A/D conversion while ADISS = 0.

ADREFM	Selection of the negative reference voltage source of the A/D converter
0	Supplied from V_{SS}
1	Supplied from P21/AV_{REFM}/ANI1

Figure 12 - 8 Format of A/D Converter Mode Register 2 (ADM2) (2/2)

ADRCK	Control of the range of the A/D conversion results that causes interrupt generation
0	The interrupt signal (INTAD) is generated when the result of A/D conversion falls within area 1, where ADLL register ≤ ADCRn register ≤ ADUL register.
1	The interrupt signal (INTAD) is generated when the result of A/D conversion falls within area 2 or 3, which are defined as follows. Area 2: ADCRn register < ADLL register Area 3: ADUL register < ADCRn register
Figure 12 - 9 shows the ranges (areas 1 to 3) in which the interrupt signal (INTAD) is generated.	

AWC	Specification of the SNOOZE mode
0	Do not use the SNOOZE mode.
1	Use the SNOOZE mode.

When there is a hardware trigger signal in the STOP mode, the STOP mode is exited, and A/D conversion is performed without operating the CPU (the SNOOZE mode).

- The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock is selected for the CPU/peripheral hardware clock (f_{CLK}). If any other clock is selected, specifying this mode is prohibited.
- When using the SNOOZE mode, set AWC to 0 in software trigger wait mode, and set AWC to 1 in hardware trigger wait mode.
- Using the SNOOZE mode in the software trigger no-wait mode or hardware trigger no-wait mode is prohibited.

<R>

- Using the SNOOZE mode in hardware trigger wait mode with sequential conversion is prohibited.
- When using the SNOOZE mode, specify a hardware trigger interval of at least “shift time to SNOOZE mode**Note 2** + conversion start time + A/D power supply stabilization wait time + A/D conversion time + 2 f_{CLK} clock cycles + 5 μs”.
- Even when using the SNOOZE mode, be sure to set the AWC bit to 0 in normal operation and change it to 1 just before shifting to STOP mode.
  Also, be sure to change the AWC bit to 0 after returning from STOP mode to normal operation.
  If the AWC bit is left set to 1, A/D conversion will not start normally in spite of the subsequent SNOOZE mode or normal operation.

ADTYP1	ADTYP0	Selection of the A/D conversion resolution
0	0	10-bit resolution
0	1	8-bit resolution
1	0	12-bit resolution
1	1	Setting prohibited

Note 1. For the internal reference voltage, see **Section 37 Electrical Characteristics**.

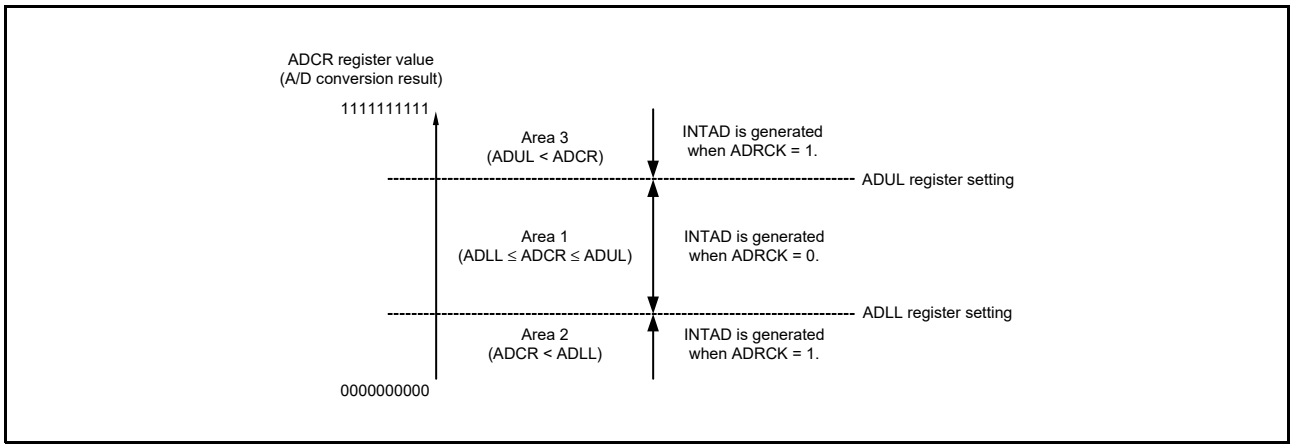
Note 2. See “Transition time from STOP mode to SNOOZE mode” in **23.3.3 SNOOZE mode**.

Caution 1. Only rewrite the value of the ADM2 register while conversion operation is stopped (ADCS = 0, ADCE = 0).

Caution 2. Do not set the ADREFP1 bit to 1 when shifting to STOP mode, or to HALT mode while the CPU is operating with the subsystem clock. When the internal reference voltage is selected (ADREFP1, ADREFP0 = 10B), the A/D converter reference voltage current (I_{ADREF}) indicated in 37.3.2 Characteristics of the supply current will be added.

Caution 3. When using AV_{REFP} and AV_{REFM}, specify ANI0 and ANI1 as the analog input channels and specify input mode by using a port mode register (PMxx).

Figure 12 - 9 Range of Interrupt Signals Generated under Control by the ADRCK Bit



<R> **Remark** The result of A/D conversion is only stored in the ADCRn and ADCRnH registers when INTAD is generated.

12.3.6 12-bit/10-bit A/D conversion result register (ADCRn)

ADCRn is a 16-bit register that holds the A/D conversion result. When A/D conversion with 12-bit resolution is selected, the higher 4 bits are fixed to 0. When A/D conversion with 10-bit resolution is selected, the lower 6 bits are fixed to 0. Each time A/D conversion is complete, the conversion result is loaded from the successive approximation register (SAR). The ADCRn register can be read by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H. In select mode, the conversion results are stored in the ADCR and ADCR0 registers**Note**. In scan mode, the conversion results of scan 0 are stored in the ADCR and ADCR0 registers, and the conversion results of scan 1 to 3 are stored in the ADCR1 to ADCR3 registers**Note**.

Note If the A/D conversion result is outside the range specified by using the A/D conversion result comparison function (the value specified by the ADM2.ADRCK bit and ADUL and ADLL registers), the result is not stored. For details, see **Figure 12 - 9**.

Figure 12 - 10 Format of 12-bit/10-bit A/D Conversion Result Register (ADCRn)

Address: FFF1FH, FFF1EH (ADCR)**Note**, F0021H, F0020H (ADCR0)**Note**, F0023H, F0022H (ADCR1), F0025H, F0024H (ADCR2), F0027H, F0026H (ADCR3)

After reset: 0000H

R/W: R

• When A/D conversion with 12-bit resolution is selected

	FFF1FH (for ADCR)								FFF1EH (for ADCR)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCRn	0	0	0	0												

• When A/D conversion with 10-bit resolution is selected

	FFF1FH (for ADCR)								FFF1EH (for ADCR)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCRn											0	0	0	0	0	0

Note The contents of the ADCR register are stored in the ADCR0 register.

Caution 1. When 8-bit resolution A/D conversion is selected (ADM2.ADTYP1, ADM2.ADTYP0 = 01B) and the ADCRn register is read, 0 is read from the bits other than the higher 8 bits.

Caution 2. When the ADCRn register is accessed in 16-bit units, and A/D conversion with 10-bit resolution is selected, the higher 10 bits of the conversion result are read in order starting at bit 15 of the ADCRn register. When A/D conversion with 12-bit resolution is selected, the higher 12 bits of the conversion result are read in order starting at bit 11 of the ADCRn register.

Caution 3. The contents of the ADCRn register may become undefined when writing to any of the following registers.

- **A/D converter mode register 0 (ADM0)**
- **Analog input channel specification register (ADS)**
- **Port mode control A registers (PMCA0, PMCA2, PMCA3, PMCA10, PMCA11, PMCA12, PMCA14, PMCA15)**
- **Port mode control T registers (PMCT0, PMCT2, PMCT15)**
- **Port mode control E register (PMCE0)**

Read the conversion result following conversion completion before writing to any of these registers. Otherwise, the correct conversion result may not be obtained.

Remark n = 0 to 3

12.3.7 8-bit A/D conversion result register (ADCRnH)

ADCRnH is an 8-bit register that holds the A/D conversion result. The higher 8 bits of 12-bit resolution are stored**Note**. The ADCRnH register can be read by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Note If the A/D conversion result is outside the range specified by using the A/D conversion result comparison function (the value specified by the ADM2.ADRCK bit and ADUL and ADLL registers), the result is not stored. For details, see **Figure 12 - 9**.

Figure 12 - 11 Format of 8-bit A/D Conversion Result Register (ADCRnH)

Address: FFF1FH (ADCRH)**Note**, F0021H (ADCR0H)**Note**, F0023H (ADCR1H), F0025H (ADCR2H), F0027H (ADCR3H)
After reset: 00H
R/W: R

Symbol	7	6	5	4	3	2	1	0
ADCRnH								

Note The contents of the ADCRH register are stored in the ADCR0H register.

Caution **The contents of the ADCRnH register may become undefined when writing to any of the following registers.**
- **A/D converter mode register 0 (ADM0)**
- **Analog input channel specification register (ADS)**
- **Port mode control A registers (PMCA0, PMCA2, PMCA3, PMCA10, PMCA11, PMCA12, PMCA14, PMCA15)**
- **Port mode control T registers (PMCT0, PMCT2, PMCT15)**
- **Port mode control E register (PMCE0)**

Read the conversion result following conversion completion before writing to any of these registers. Otherwise, the correct conversion result may not be obtained.

Remark n = 0 to 3

12.3.8 Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted. The ADS register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 12 - 12 Format of Analog Input Channel Specification Register (ADS)

Address: FFF31H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

Select mode (ADM0.ADMD = 0) (1/2)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P20/ANI0/AVREFP pin
0	0	0	0	0	1	ANI1	P21/ANI1/AVREFM pin
0	0	0	0	1	0	ANI2	P22/ANI2 pin
0	0	0	0	1	1	ANI3	P23/ANI3 pin
0	0	0	1	0	0	ANI4	P24/ANI4 pin
0	0	0	1	0	1	ANI5	P25/ANI5 pin
0	0	0	1	1	0	ANI6	P26/ANI6 pin
0	0	0	1	1	1	ANI7	P27/ANI7 pin
0	0	1	0	0	0	ANI8	P150/ANI8 pin
0	0	1	0	0	1	ANI9	P151/ANI9 pin
0	0	1	0	1	0	ANI10	P152/ANI10 pin
0	0	1	0	1	1	ANI11	P153/ANI11 pin
0	0	1	1	0	0	ANI12	P154/ANI12 pin
0	0	1	1	0	1	ANI13	P155/ANI13 pin
0	0	1	1	1	0	ANI14	P156/ANI14 pin
0	1	0	0	0	0	ANI16	P03/ANI16 pin**Note 1**
0	1	0	0	0	1	ANI17	P02/ANI17 pin**Note 2**
0	1	0	0	1	0	ANI18	P147/ANI18 pin
0	1	0	0	1	1	ANI19	P120/ANI19 pin
0	1	0	1	0	0	ANI20	P100/ANI20 pin
0	1	0	1	0	1	ANI21	P37/ANI21 pin
0	1	0	1	1	0	ANI22	P36/ANI22 pin
0	1	0	1	1	1	ANI23	P35/ANI23 pin
0	1	1	0	0	0	ANI24	P117/ANI24 pin
0	1	1	0	0	1	ANI25	P116/ANI25 pin
0	1	1	0	1	0	ANI26	P115/ANI26 pin
0	1	1	1	1	0	—	TSCAP voltage of the CTSU
1	0	0	0	0	0	—	Temperature sensor output voltage

Select mode (ADM0.ADMD = 0) (2/2)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
1	0	0	0	0	1	—	Internal reference voltage Note 3
Other than the above						Setting prohibited	

Scan mode (ADM0.ADMD = 1)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel			
						Scan 0	Scan 1	Scan 2	Scan 3
0	0	0	0	0	0	ANI0	ANI1	ANI2	ANI3
0	0	0	0	0	1	ANI1	ANI2	ANI3	ANI4
0	0	0	0	1	0	ANI2	ANI3	ANI4	ANI5
0	0	0	0	1	1	ANI3	ANI4	ANI5	ANI6
0	0	0	1	0	0	ANI4	ANI5	ANI6	ANI7
0	0	0	1	0	1	ANI5	ANI6	ANI7	ANI8
0	0	0	1	1	0	ANI6	ANI7	ANI8	ANI9
0	0	0	1	1	1	ANI7	ANI8	ANI9	ANI10
0	0	1	0	0	0	ANI8	ANI9	ANI10	ANI11
0	0	1	0	0	1	ANI9	ANI10	ANI11	ANI12
0	0	1	0	1	0	ANI10	ANI11	ANI12	ANI13
0	0	1	0	1	1	ANI11	ANI12	ANI13	ANI14
Other than the above						Setting prohibited			

Note 1. P01/ANI16 pin in the 30- and 32-pin R7F100G products

Note 2. P00/ANI17 pin in the 30- and 32-pin R7F100G products

Note 3. For the internal reference voltage, see **Section 37 Electrical Characteristics**.

Caution 1. Be sure to leave bits 6 and 5 set to 0.

Caution 2. Set the port that is specified as the analog input by a PMCAxx register to the input mode by using a port mode register (PM0, PM2, PM3, PM10 to PM12, PM14, or PM15).

Caution 3. When specifying an input channel by the ADS register, do not select the pin that is specified as digital I/O by a port mode control A register (PMCA0, PMCA2, PMCA3, PMCA10, PMCA11, PMCA12, PMCA14, or PMCA15), port mode control T register (PMCT0, PMCT2, or PMCT15), or port mode control E register (PMCE0).

Caution 4. Rewrite the value of the ADISS bit while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 5. If using AVREFP as the positive reference voltage of the A/D converter, do not select ANI0 as an A/D conversion channel.

Caution 6. If using AVREFM as the negative reference voltage of the A/D converter, do not select ANI1 as an A/D conversion channel.

(Cautions are listed on the next page.)

Caution 7. When the setting of the ADISS bit is 1, the internal reference voltage cannot be used for the positive reference voltage. After the ADISS bit is set to 1, the initial conversion result cannot be used. For the setting flow, see 12.7.5 Example of settings when temperature sensor output voltage, internal reference voltage, or TSCAP voltage of the CTSU is selected in the software trigger no-wait mode and one-shot conversion mode. For details about the internal reference voltage, see Section 37 Electrical Characteristics.

Caution 8. Do not set the ADISS bit to 1 when shifting to STOP mode, or to HALT mode while the CPU is operating with the subsystem clock. When the ADISS bit is set to 1, the A/D converter reference voltage current (I_{ADREF}) indicated in 37.3.2 Characteristics of the supply current will be added.

Caution 9. When the setting of the ADISS bit is 1, the hardware trigger wait mode and one-shot conversion mode cannot be used at the same time.

Caution 10. When the setting of the ADISS bit is 1, the software trigger wait mode and one-shot conversion mode cannot be used at the same time.

12.3.9 Conversion result comparison upper limit setting register (ADUL)

The ADUL register sets the upper limit value of the range of A/D conversion results that causes interrupt generation. The A/D conversion results and ADUL register value are compared, and generation of the interrupt signal (INTAD) is controlled in the range specified for the ADRCK bit of A/D converter mode register 2 (ADM2). For details, see **Figure 12 - 9 Range of Interrupt Signals Generated under Control by the ADRCK Bit**. The ADUL register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is FFH.

Figure 12 - 13 Format of Conversion Result Comparison Upper Limit Setting Register (ADUL)

Address: F0011H
After reset: FFH
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ADUL	ADUL7	ADUL6	ADUL5	ADUL4	ADUL3	ADUL2	ADUL1	ADUL0

Caution 1. When A/D conversion with 10-bit resolution is selected, the value of the ADCRn[15:8] bits in the A/D conversion result register is compared with the value in the ADUL register. When A/D conversion with 12-bit resolution is selected, the value of the ADCRn[11:4] bits in the A/D conversion result register is compared with the value in the ADUL register.
Caution 2. Only write a new value to the ADUL register while conversion is stopped (ADCS = 0, ADCE = 0).
Caution 3. The setting of the ADUL register must be greater than that of the ADLL register.

12.3.10 Conversion result comparison lower limit setting register (ADLL)

The ADLL register sets the lower limit value of the range of A/D conversion results that causes interrupt generation. The A/D conversion results and ADLL register value are compared, and generation of the interrupt signal (INTAD) is controlled in the range specified for the ADRCK bit of A/D converter mode register 2 (ADM2). For details, see **Figure 12 - 9 Range of Interrupt Signals Generated under Control by the ADRCK Bit**. The ADLL register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 12 - 14 Format of Conversion Result Comparison Lower Limit Setting Register (ADLL)

Address: F0012H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ADLL	ADLL7	ADLL6	ADLL5	ADLL4	ADLL3	ADLL2	ADLL1	ADLL0

Caution 1. When A/D conversion with 10-bit resolution is selected, the value of the ADCRn[15:8] bits in the A/D conversion result register is compared with the value in the ADLL register. When A/D conversion with 12-bit resolution is selected, the value of the ADCRn[11:4] bits in the A/D conversion result register is compared with the value in the ADLL register.
Caution 2. Only write a new value to the ADLL register while conversion is stopped (ADCS = 0, ADCE = 0).
Caution 3. The setting of the ADLL register must be smaller than that of the ADUL register.

12.3.11 A/D test register (ADTES)

This register is used to select the positive reference voltage or negative reference voltage for the converter, an analog input channel (ANIxx), the temperature sensor output voltage, the internal reference voltage**Note**, or the TSCAP voltage of the CTSU as the target for A/D conversion. When using this register to test the converter, set as follows.

- For zero-scale measurement, select the negative reference voltage as the target for conversion.
- For full-scale measurement, select the positive reference voltage as the target for conversion.

The ADTES register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Note For the internal reference voltage, see **Section 37 Electrical Characteristics**.

Figure 12 - 15 Format of A/D Test Register (ADTES)

Address: F0013H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	0	0	ADTES1	ADTES0

ADTES1	ADTES0	A/D conversion target
0	0	ANIxx, temperature sensor output voltage, internal reference voltage**Note**, or TSCAP voltage of the CTSU (set by the analog input channel specification register (ADS))
1	0	Negative reference voltage (selected by the ADREFM bit of the ADM2 register)
1	1	Positive reference voltage (selected by the ADREFP1 and ADREFP0 bits of the ADM2 register)
Other than the above		Setting prohibited

Note For the internal reference voltage, see **Section 37 Electrical Characteristics**.

Caution Be sure to leave bits 7 to 2 set to 0.

12.3.12 Registers for controlling the port functions multiplexed with the analog inputs of the A/D converter

Set the following registers to control the port functions multiplexed with the analog inputs of the A/D converter.

- Port mode registers (PMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)

For details, see the following sections.

- **4.3.1 Port mode registers (PMxx)**
- **4.3.7 Port mode control A registers (PMCAxx)**
- **4.3.8 Port mode control T registers (PMCTxx)**
- **4.3.9 Port mode control E registers (PMCExx)**

For examples of register settings for controlling port functions, see **4.5.4 Examples of register settings for port and multiplexed functions**.

12.4 A/D Converter Operations

The A/D converter conversion operations are described below.

<1> The voltage input to the selected analog input channel is sampled by the sample & hold (S&H) circuit.

<2> When sampling has been done for a certain time, the S&H circuit is placed in the hold state and the sampled voltage is held until the A/D conversion operation has ended.

<3> Bit 11 of the successive approximation register (SAR) is set to 1. The series resistor string voltage tap is set to (1/2) × AVREF by the tap selector.

<4> The voltage difference between the series resistor string voltage tap and sampled voltage is compared by the voltage comparator. If the analog input is greater than (1/2) × AVREF, the MSB of the SAR register remains set to 1. If the analog input is smaller than (1/2) × AVREF, the MSB is reset to 0.

<5> Next, bit 10 of the SAR register is automatically set to 1, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected in accord with the value of bit 11 already compared, as described below.

- Bit 11 = 1: (3/4) × AVREF
- Bit 11 = 0: (1/4) × AVREF

The voltage tap and sampled voltage are compared and bit 10 of the SAR register is manipulated as follows.

- Sampled voltage ≥ Voltage tap: Bit 10 = 1
- Sampled voltage < Voltage tap: Bit 10 = 0

<6> Comparison is continued in this way up to bit 0 of the SAR register.

<7> When the comparison of 12 bits is completed, a series of comparison results remains in the SAR register, and the
<R> result value is transferred to the A/D conversion result registers (ADCRn and ADCRnH) and then latched**Note 1**. At the same time, the A/D conversion end interrupt request signal (INTAD) can also be generated**Note 1**.

<8> Repeat steps <1> to <7>, until the ADCS bit is cleared to 0**Note 2**. To stop the A/D converter, clear the ADCS bit to 0.

Note 1. If the A/D conversion result is outside the A/D conversion result range specified by the ADM2.ADRCK bit and the ADUL and ADLL registers, the A/D conversion end interrupt request signal (INTAD) is not generated and no A/D conversion results are stored in the ADCRn and ADCRnH registers. For details, see **Figure 12 - 9 Range of Interrupt Signals Generated under Control by the ADRCK Bit**.

Note 2. While in the sequential conversion mode, the ADCS flag is not automatically cleared to 0. This flag is not automatically cleared to 0 while in the one-shot conversion mode of the hardware trigger no-wait mode, either. Instead, 1 is retained.

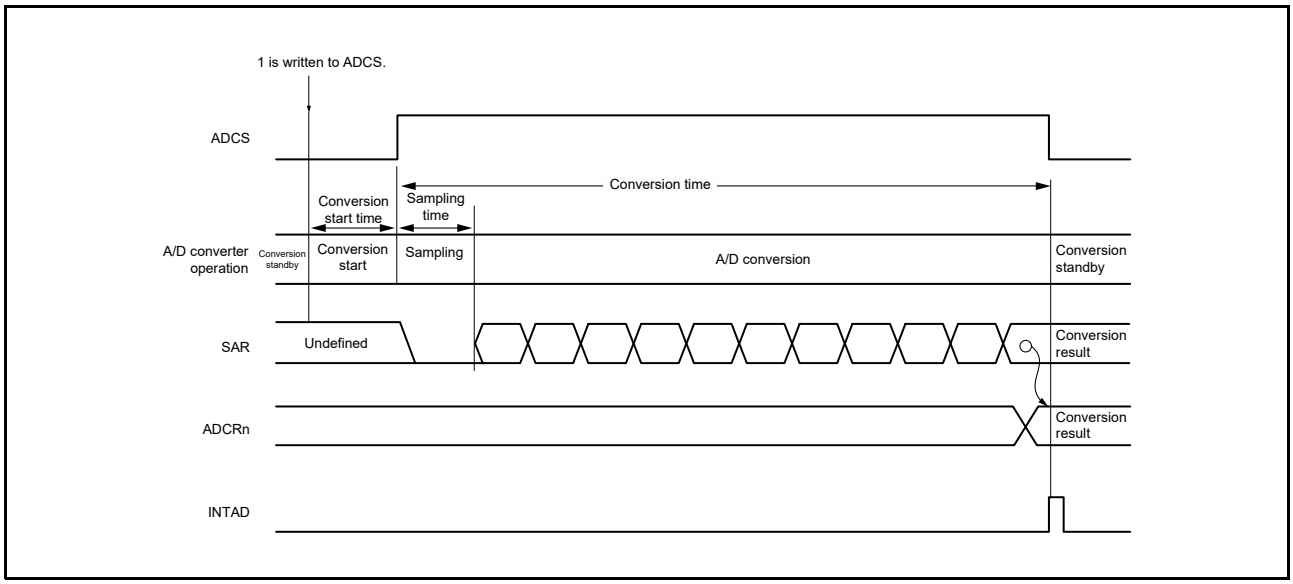
Remark 1. Two types of the A/D conversion result registers are available.

- ADCRn register (16 bits) to hold 12-bit/10-bit A/D conversion value
- ADCRnH register (8 bits) to hold 8-bit A/D conversion value

Remark 2. AVREF: The positive reference voltage of the A/D converter. This can be selected from AVREFP, the internal reference voltage, and VDD. For the internal reference voltage, see **Section 37 Electrical Characteristics**.

Remark 3. n = 0 to 3

Figure 12 - 16 Operation of the A/D Converter in the Software Trigger No-wait Mode



In one-shot conversion mode, the ADCS bit is automatically cleared to 0 after completion of A/D conversion. In sequential conversion mode, A/D conversion operations proceed continuously until the software clears the ADCS bit of A/D converter mode register 0 (ADM0) to 0. Writing to the analog input channel specification register (ADS) during A/D conversion aborts the current A/D conversion and starts a new conversion for the analog input newly specified by the ADS register. Any data from the aborted conversion are discarded. The values of the A/D conversion result registers (ADCRn and ADCRnH) following a reset are 0000H and 00H, respectively.

12.5 Input Voltage and Conversion Results

The relationship between the analog voltage input to the analog input pins (ANI0 to ANI14, ANI16 to ANI26) and the theoretical A/D conversion result (stored in the 12-bit/10-bit A/D conversion result register (ADCRn)) is shown by the following expression.

$$ADCRn = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 4096 + 0.5\right)$$

or

$$(ADCRn - 0.5) \times \frac{AV_{REF}}{4096} \leq V_{AIN} < (ADCRn + 0.5) \times \frac{AV_{REF}}{4096}$$

where, INT( ): Function which returns integer part of value in parentheses

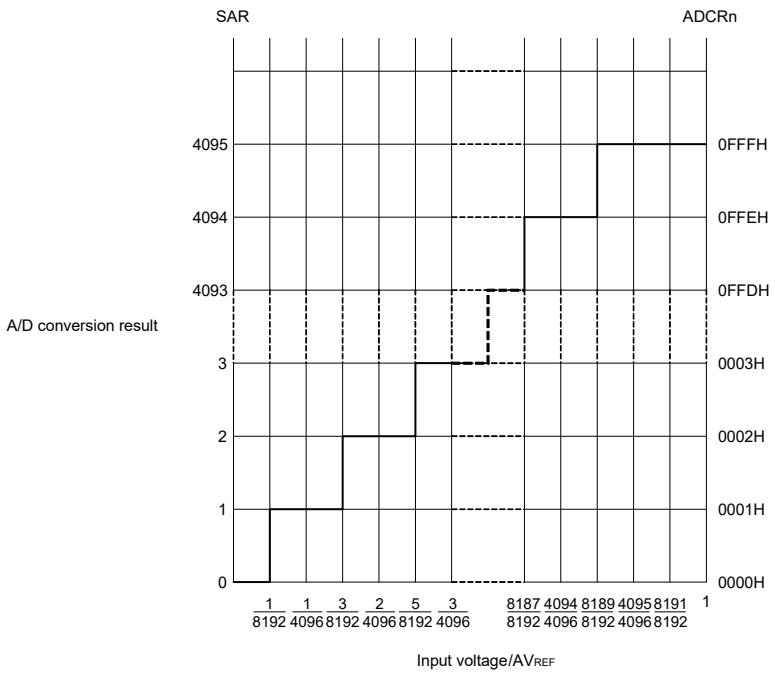
V_{AIN}: Analog input voltage

AV_{REF}: AV_{REF} pin voltage

ADCRn: 12-bit/10-bit A/D conversion result register (ADCRn) value

Figure 12 - 17 shows the relationship between analog input voltage and A/D conversion result.

Figure 12 - 17 Relationship Between Analog Input Voltage and A/D Conversion Result



Remark AV_{REF}: The positive reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage, and V_{DD}. For the internal reference voltage, see **Section 37 Electrical Characteristics**.

12.6 Operating Modes of the A/D Converter

This section describes the operation of the A/D converter modes. For the procedures to set each of the modes, see **12.7 Flows of Setting the A/D Converter Modes**.

12.6.1 Software trigger no-wait mode: Select mode and sequential conversion mode

<1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 while conversion is stopped places the A/D converter in the standby state.

<2> To allow stabilization of the internal circuit, apply software control for a wait of 1 μs + 2 cycles of the conversion clock (f_{AD}), then set the ADCS bit in the ADM0 register to 1 to start A/D conversion of the analog input specified by the analog input channel specification register (ADS).

<3> Once A/D conversion is complete, the result of conversion is stored in the A/D conversion result registers (ADCR, ADCRH, ADCR0, and ADCR0H), and an A/D conversion end interrupt request signal (INTAD) is generated. The next A/D conversion starts at the same time.

<4> Overwriting the ADCS bit with 1 during conversion aborts the current A/D conversion and starts a new conversion. Any data from the aborted conversion are discarded.

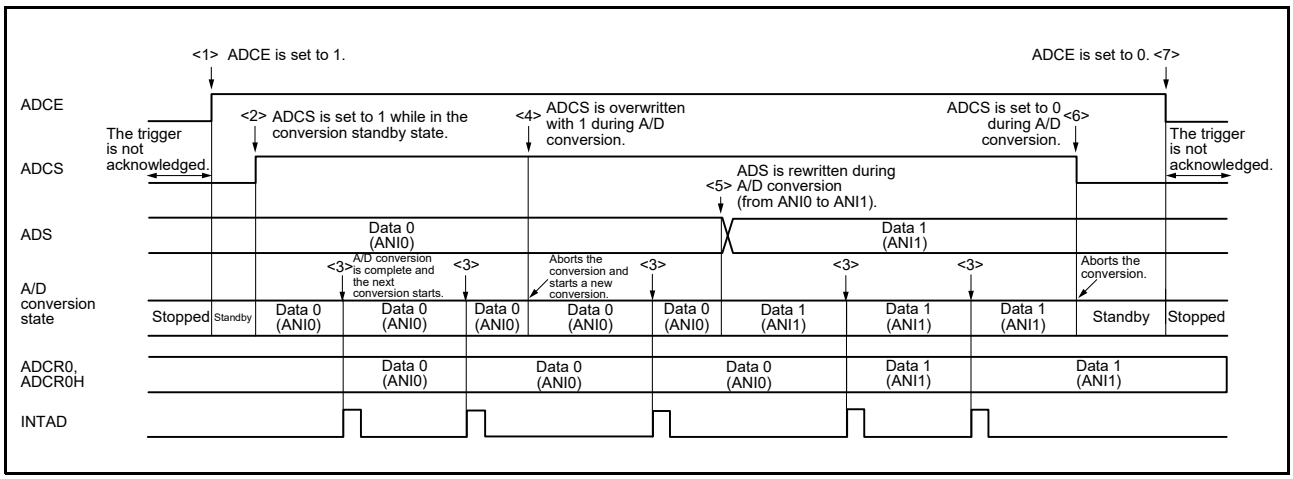
<5> Writing to the ADS register during conversion aborts the current A/D conversion and starts a new conversion for the analog input newly specified by the ADS register. Any data from the aborted conversion are discarded.

<6> Setting the ADCS bit to 0 during conversion aborts the current A/D conversion, and places the A/D converter in the standby state.

<7> Setting the ADCE bit to 0 while the A/D converter is in the standby state places the converter in the stopped state.

Caution In software trigger no-wait mode, a hardware trigger does not start A/D conversion.

Figure 12 - 18 Example of Timing of Operations in the Software Trigger No-wait Mode: Select Mode and Sequential Conversion Mode



Caution When <4> or <5> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when the stabilization wait time for the A/D power supply is automatically secured before A/D conversion starts in software trigger wait mode or hardware trigger wait mode. For details, see Table 12 - 3 A/D Conversion Time Selection (3/8) and Table 12 - 3 A/D Conversion Time Selection (4/8).

12.6.2 Software trigger no-wait mode: Select mode and one-shot conversion mode

<1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 while conversion is stopped places the A/D converter in the standby state.

<2> To allow stabilization of the internal circuit, apply software control for a wait of 1 μs + 2 cycles of the conversion clock (f_{AD}), then set the ADCS bit in the ADM0 register to 1 to start A/D conversion of the analog input specified by the analog input channel specification register (ADS).

<3> Once A/D conversion is complete, the result of conversion is stored in the A/D conversion result registers (ADCR, ADCRH, ADCR0, and ADCR0H), and an A/D conversion end interrupt request signal (INTAD) is generated.

<4> Once A/D conversion is complete, the ADCS bit is automatically cleared to 0, and the converter is placed in the standby state.

<5> Overwriting the ADCS bit with 1 during conversion aborts the current A/D conversion and starts a new conversion. Any data from the aborted conversion are discarded.

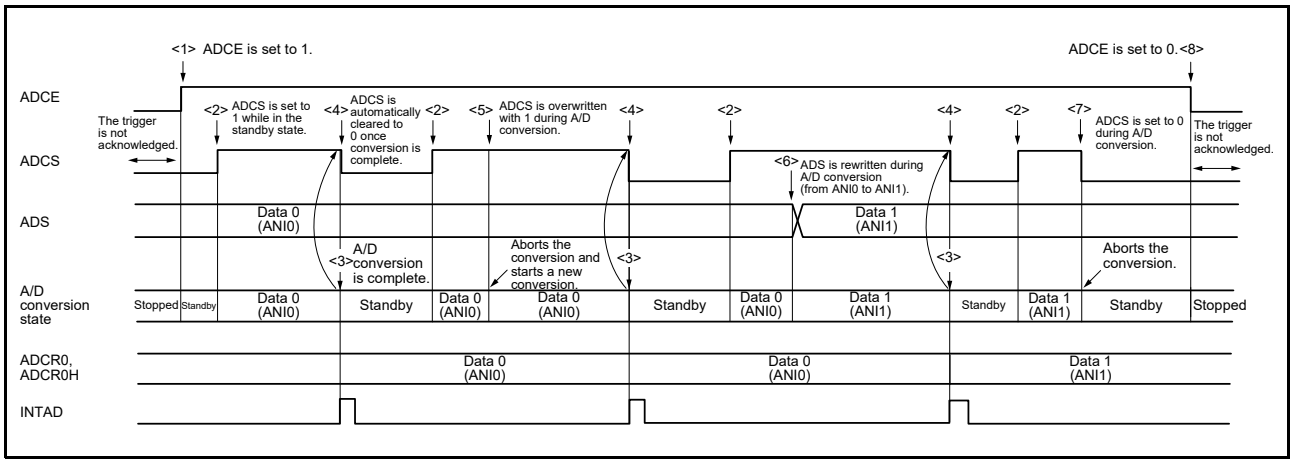
<6> Writing to the ADS register during conversion aborts the current A/D conversion and starts a new conversion for the analog input newly specified by the ADS register. Any data from the aborted conversion are discarded.

<7> Setting the ADCS bit to 0 during conversion aborts the current A/D conversion, and places the A/D converter in the standby state.

<8> Setting the ADCE bit to 0 while the A/D converter is in the standby state places the converter in the stopped state.

Caution In software trigger no-wait mode, a hardware trigger does not start A/D conversion.

Figure 12 - 19 Example of Timing of Operations in the Software Trigger No-wait Mode: Select Mode and One-shot Conversion Mode



Caution When <5> or <6> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when the stabilization wait time for the A/D power supply is automatically secured before A/D conversion starts in software trigger wait mode or hardware trigger wait mode. For details, see Table 12 - 3 A/D Conversion Time Selection (3/8) and Table 12 - 3 A/D Conversion Time Selection (4/8).

12.6.3 Software trigger no-wait mode: Scan mode and sequential conversion mode

<1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 while conversion is stopped places the A/D converter in the standby state.

<2> To allow stabilization of the internal circuit, apply software control for a wait of 1 μs + 2 cycles of the conversion clock (f_{AD}), then set the ADCS bit in the ADM0 register to 1 to start A/D conversion on the four analog input channels specified as scan 0 to scan 3 by the analog input channel specification register (ADS). A/D conversion proceeds sequentially, starting with the channel specified as scan 0.

<3> The four analog input channels are sequentially converted. Each time a conversion is completed, the result of
<R> conversion is stored in the A/D conversion result registers (ADCRn and ADCRnH), and an A/D conversion end interrupt request signal (INTAD) is generated immediately after A/D conversion of the four channels is complete. On completion of conversion for all four channels, the next A/D conversion of the specified channels is automatically started.

<4> Overwriting the ADCS bit with 1 during conversion aborts the current A/D conversion and starts a new conversion from the first channel. Any data from the aborted conversion are discarded.

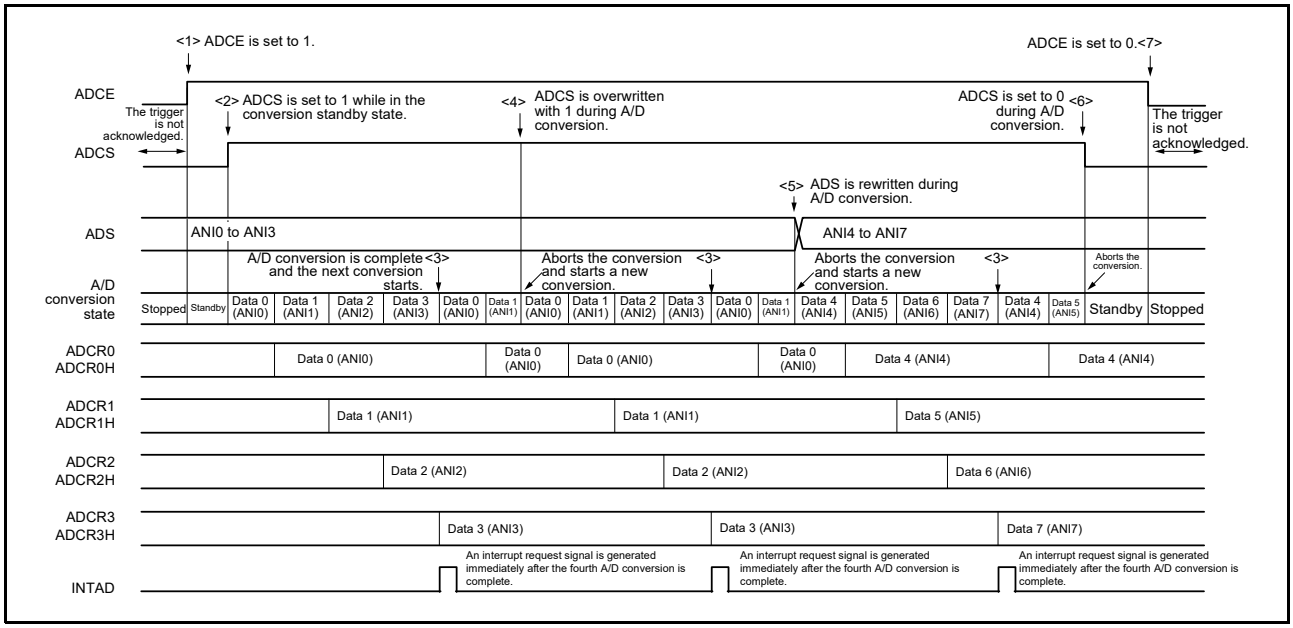
<5> Writing to the ADS register during conversion aborts the current A/D conversion and starts a new conversion from the first channel newly specified by the ADS register. Any data from the aborted conversion are discarded.

<6> Setting the ADCS bit to 0 during conversion aborts the current A/D conversion, and places the A/D converter in the standby state.

<7> Setting the ADCE bit to 0 while the A/D converter is in the standby state places the converter in the stopped state.

Caution In software trigger no-wait mode, a hardware trigger does not start A/D conversion.

Figure 12 - 20 Example of Timing of Operations in the Software Trigger No-wait Mode: Scan Mode and Sequential Conversion Mode



Caution When <4> or <5> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when the stabilization wait time for the A/D power supply is automatically secured before A/D conversion starts in software trigger wait mode or hardware trigger wait mode. For details, see Table 12 - 3 A/D Conversion Time Selection (3/8) and Table 12 - 3 A/D Conversion Time Selection (4/8).

12.6.4 Software trigger no-wait mode: Scan mode and one-shot conversion mode

<1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 while conversion is stopped places the A/D converter in the standby state.

<2> To allow stabilization of the internal circuit, apply software control for a wait of 1 μs + 2 cycles of the conversion clock (f_{AD}), then set the ADCS bit in the ADM0 register to 1 to start A/D conversion on the four analog input channels specified as scan 0 to scan 3 by the analog input channel specification register (ADS). A/D conversion proceeds sequentially, starting with the channel specified as scan 0.

<3> The four analog input channels are sequentially converted. Each time a conversion is completed, the result of
<R> conversion is stored in the A/D conversion result registers (ADCRn and ADCRnH), and an A/D conversion end interrupt request signal (INTAD) is generated immediately after A/D conversion of the four channels is complete.

<4> Once A/D conversion for all four channels is complete, the ADCS bit is automatically cleared to 0, and the converter is placed in the standby state.

<5> Overwriting the ADCS bit with 1 during conversion aborts the current A/D conversion and starts a new conversion from the first channel. Any data from the aborted conversion are discarded.

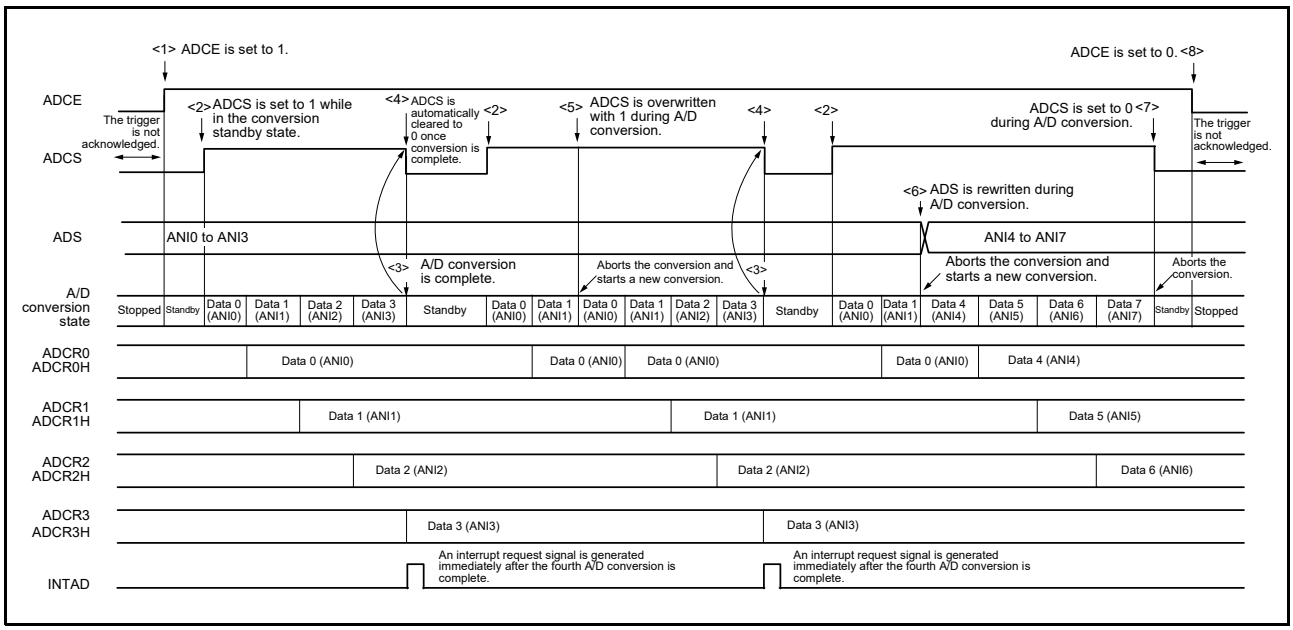
<6> Writing to the ADS register during conversion aborts the current A/D conversion and starts a new conversion from the first channel newly specified by the ADS register. Any data from the aborted conversion are discarded.

<7> Setting the ADCS bit to 0 during conversion aborts the current A/D conversion, and places the A/D converter in the standby state.

<8> Setting the ADCE bit to 0 while the A/D converter is in the standby state places the converter in the stopped state.

Caution In software trigger no-wait mode, a hardware trigger does not start A/D conversion.

Figure 12 - 21 Example of Timing of Operations in the Software Trigger No-wait Mode: Scan Mode and One-shot Conversion Mode



Caution When <5> or <6> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when the stabilization wait time for the A/D power supply is automatically secured before A/D conversion starts in software trigger wait mode or hardware trigger wait mode. For details, see Table 12 - 3 A/D Conversion Time Selection (3/8) and Table 12 - 3 A/D Conversion Time Selection (4/8).

12.6.5 Software trigger wait mode: Select mode and sequential conversion mode

<1> To shift to software trigger wait mode, the ADCE bit of A/D converter mode register 0 (ADM0) must be set to 0 (stopped state).

<2> Setting the ADCS bit to 1 while conversion is stopped starts A/D conversion of the analog input specified by the analog input channel specification register (ADS) in software trigger wait mode.

<3> Once A/D conversion is complete, the result of conversion is stored in the A/D conversion result registers (ADCR, ADCRH, ADCR0, and ADCR0H), and an A/D conversion end interrupt request signal (INTAD) is generated. The next A/D conversion starts at the same time.

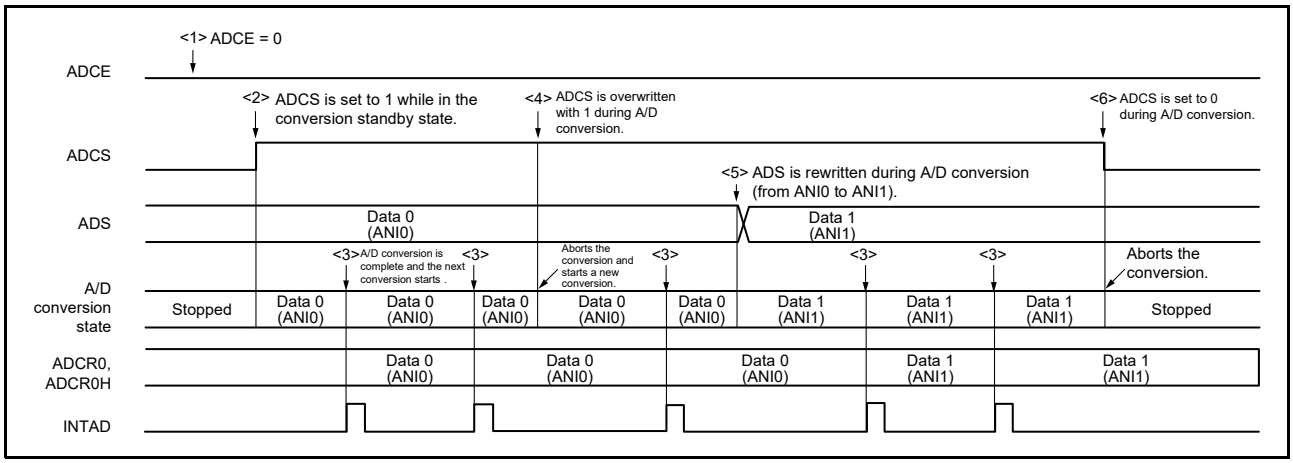
<4> Overwriting the ADCS bit with 1 during conversion aborts the current A/D conversion and starts a new conversion. Any data from the aborted conversion are discarded.

<5> Writing to the ADS register during conversion aborts the current A/D conversion and starts a new conversion for the analog input newly specified by the ADS register. Any data from the aborted conversion are discarded.

<6> Setting the ADCS bit to 0 during conversion aborts the current A/D conversion, and places the A/D converter in the stopped state.

Caution In software trigger wait mode, a hardware trigger does not start A/D conversion.

Figure 12 - 22 Example of Timing of Operations in the Software Trigger Wait Mode: Select Mode and Sequential Conversion Mode



Caution When <4> or <5> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed since the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when the stabilization wait time for the A/D power supply is automatically secured before A/D conversion starts in software trigger wait mode or hardware trigger wait mode. For details, see Table 12 - 3 A/D Conversion Time Selection (3/8) and Table 12 - 3 A/D Conversion Time Selection (4/8).

12.6.6 Software trigger wait mode: Select mode and one-shot conversion mode

<1> To shift to software trigger wait mode, the ADCE bit of A/D converter mode register 0 (ADM0) must be set to 0 (stopped state).

<2> Setting the ADCS bit to 1 while conversion is stopped starts A/D conversion of the analog input specified by the analog input channel specification register (ADS) in software trigger wait mode.

<3> Once A/D conversion is complete, the result of conversion is stored in the A/D conversion result registers (ADCR, ADCRH, ADCR0, and ADCR0H), and an A/D conversion end interrupt request signal (INTAD) is generated.

<4> Once A/D conversion is complete, the ADCS bit is automatically cleared to 0, and the converter is placed in the stopped state.

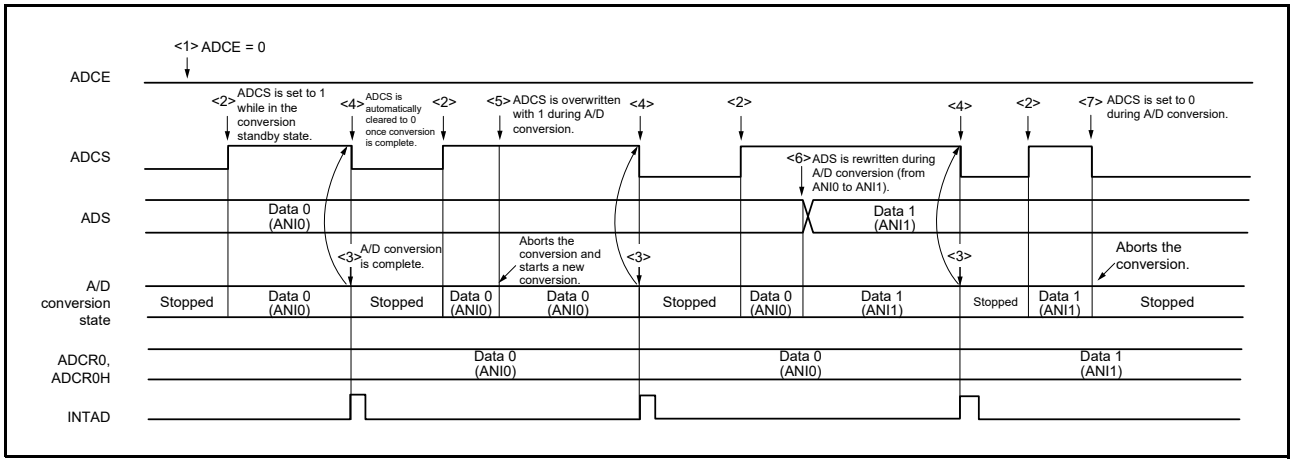
<5> Overwriting the ADCS bit with 1 during conversion aborts the current A/D conversion and starts a new conversion. Any data from the aborted conversion are discarded.

<6> Writing to the ADS register during conversion aborts the current A/D conversion and starts a new conversion for the analog input newly specified by the ADS register. Any data from the aborted conversion are discarded.

<7> Setting the ADCS bit to 0 during conversion aborts the current A/D conversion, and places the A/D converter in the stopped state.

Caution In software trigger wait mode, a hardware trigger does not start A/D conversion.

Figure 12 - 23 Example of Timing of Operations in the Software Trigger Wait Mode: Select Mode and One-shot Conversion Mode



Caution 1. When <5> or <6> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed since the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when the stabilization wait time for the A/D power supply is automatically secured before A/D conversion starts in software trigger wait mode or hardware trigger wait mode. For details, see Table 12 - 3 A/D Conversion Time Selection (3/8) and Table 12 - 3 A/D Conversion Time Selection (4/8).

Caution 2. The setting of ADISS being 1 (the input source is temperature sensor output voltage or internal reference voltage) cannot be used in the software trigger wait mode: select mode and one-shot conversion mode.

12.6.7 Software trigger wait mode: Scan mode and sequential conversion mode

<1> To shift to software trigger wait mode, the ADCE bit of A/D converter mode register 0 (ADM0) must be set to 0 (stopped state).

<2> Setting the ADCS bit to 1 while conversion is stopped starts A/D conversion on the four analog input channels specified as scan 0 to scan 3 by the analog input channel specification register (ADS) in software trigger wait mode. A/D conversion proceeds sequentially, starting with the channel specified as scan 0.

<3> The four analog input channels are sequentially converted. Each time a conversion is completed, the result of
<R> conversion is stored in the A/D conversion result registers (ADCRn and ADCRnH), and an A/D conversion end interrupt request signal (INTAD) is generated immediately after A/D conversion of the four channels is complete. On completion of conversion for all four channels, the next A/D conversion of the specified channels is automatically started.

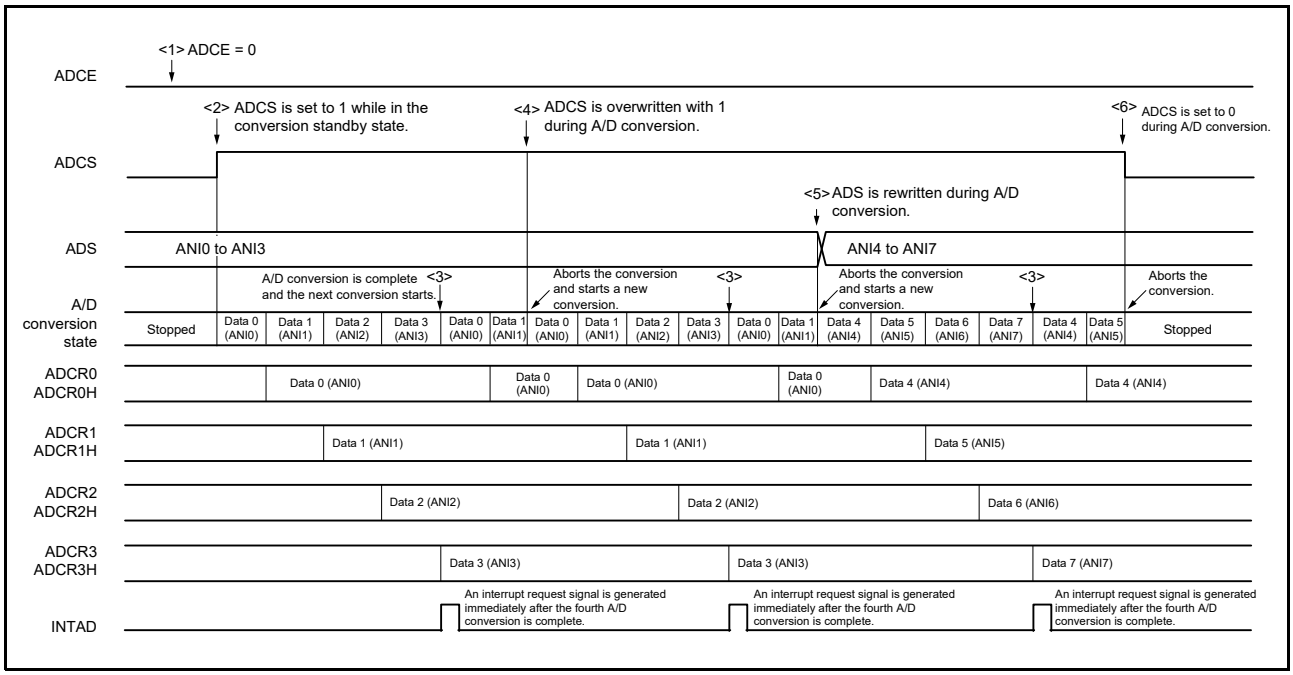
<4> Overwriting the ADCS bit with 1 during conversion aborts the current A/D conversion and starts a new conversion. Any data from the aborted conversion are discarded.

<5> Writing to the ADS register during conversion aborts the current A/D conversion and starts a new conversion from the first channel newly specified by the ADS register. Any data from the aborted conversion are discarded.

<6> Setting the ADCS bit to 0 during conversion aborts the current A/D conversion, and places the A/D converter in the stopped state.

Caution In software trigger wait mode, a hardware trigger does not start A/D conversion.

Figure 12 - 24 Example of Timing of Operations in the Software Trigger Wait Mode: Scan Mode and Sequential Conversion Mode



Caution When <4> or <5> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed since the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when the stabilization wait time for the A/D power supply is automatically secured before A/D conversion starts in software trigger wait mode or hardware trigger wait mode. For details, see Table 12 - 3 A/D Conversion Time Selection (3/8) and Table 12 - 3 A/D Conversion Time Selection (4/8).

12.6.8 Software trigger wait mode: Scan mode and one-shot conversion mode

<1> To shift to software trigger wait mode, the ADCE bit of A/D converter mode register 0 (ADM0) must be set to 0 (stopped state).

<2> Setting the ADCS bit to 1 while conversion is stopped starts A/D conversion on the four analog input channels specified as scan 0 to scan 3 by the analog input channel specification register (ADS) in software trigger wait mode. A/D conversion proceeds sequentially, starting with the channel specified as scan 0.

<3> The four analog input channels are sequentially converted. Each time a conversion is completed, the result of
<R> conversion is stored in the A/D conversion result registers (ADCRn and ADCRnH), and an A/D conversion end interrupt request signal (INTAD) is generated immediately after A/D conversion of the four channels is complete.

<4> Once A/D conversion for all four channels is complete, the ADCS bit is automatically cleared to 0, and the converter is placed in the stopped state.

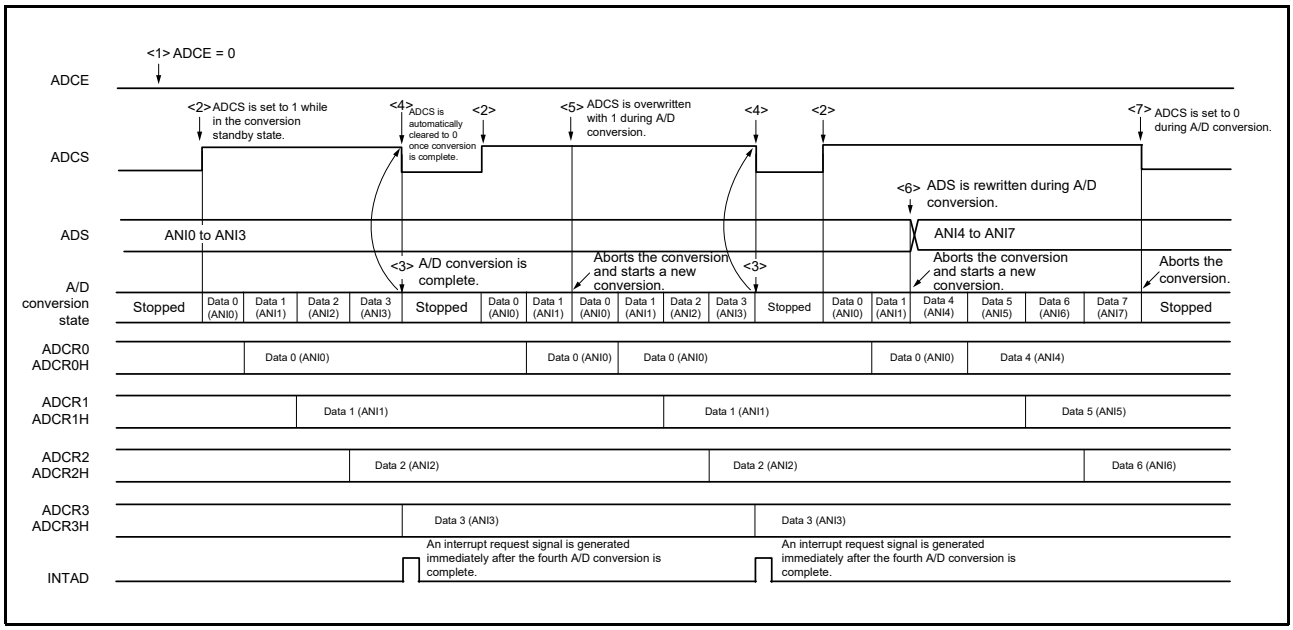
<5> Overwriting the ADCS bit with 1 during conversion aborts the current A/D conversion and starts a new conversion from the first channel. Any data from the aborted conversion are discarded.

<6> Writing to the ADS register during conversion aborts the current A/D conversion and starts a new conversion from the first channel newly specified by the ADS register. Any data from the aborted conversion are discarded.

<7> Setting the ADCS bit to 0 during conversion aborts the current A/D conversion, and places the A/D converter in the stopped state.

Caution In software trigger wait mode, a hardware trigger does not start A/D conversion.

Figure 12 - 25 Example of Timing of Operations in the Software Trigger Wait Mode: Scan Mode and One-shot Conversion Mode



Caution When <5> or <6> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed since the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when the stabilization wait time for the A/D power supply is automatically secured before A/D conversion starts in software trigger wait mode or hardware trigger wait mode. For details, see Table 12 - 3 A/D Conversion Time Selection (3/8) and Table 12 - 3 A/D Conversion Time Selection (4/8).

12.6.9 Hardware trigger no-wait mode: Select mode and sequential conversion mode

<1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 while conversion is stopped places the A/D converter in the standby state.

<2> To allow stabilization of the internal circuit, apply software control for a wait of 1 μs + 2 cycles of the conversion clock (f_{AD}), then set the ADCS bit in the ADM0 register to 1 to place the A/D converter in the hardware trigger standby state. Note that, while in this state, A/D conversion does not start even if the ADCS bit is set to 1.

<3> Input of a hardware trigger while the ADCS bit is 1 starts A/D conversion of the analog input specified by the analog input channel specification register (ADS).

<4> Once A/D conversion is complete, the result of conversion is stored in the A/D conversion result registers (ADCR, ADCRH, ADCR0, and ADCR0H), and an A/D conversion end interrupt request signal (INTAD) is generated. The next A/D conversion starts at the same time.

<5> Input of a hardware trigger during conversion aborts the current A/D conversion and starts a new conversion. Any data from the aborted conversion are discarded.

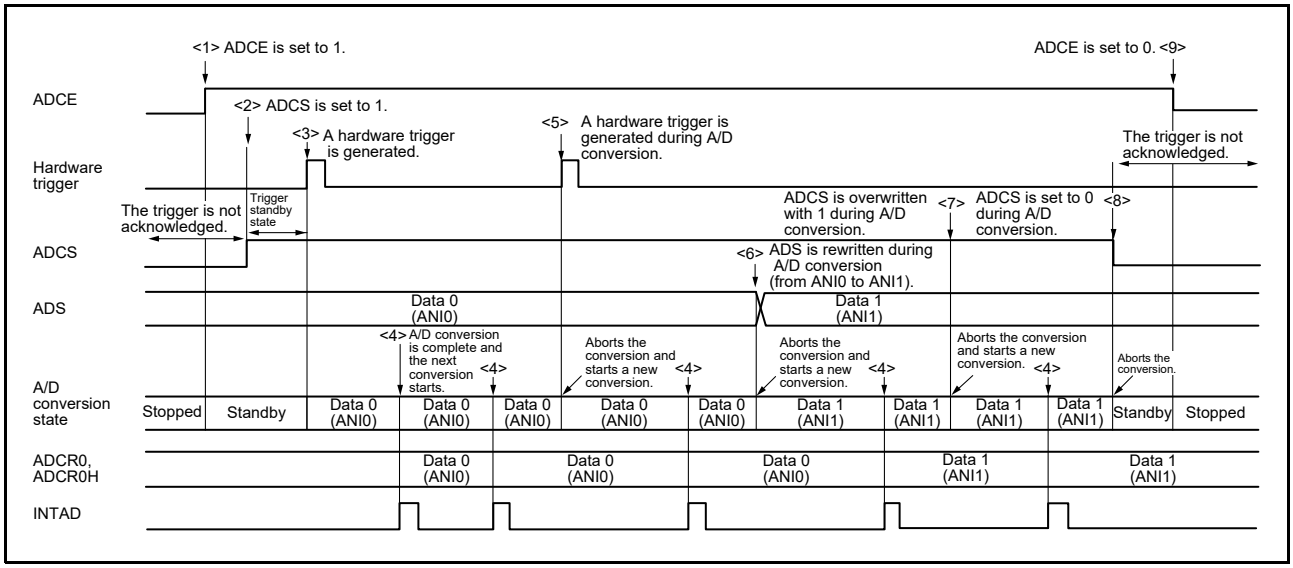
<6> Writing to the ADS register during conversion aborts the current A/D conversion and starts a new conversion for the analog input newly specified by the ADS register. Any data from the aborted conversion are discarded.

<7> Overwriting the ADCS bit with 1 during conversion aborts the current A/D conversion and starts a new conversion. Any data from the aborted conversion are discarded.

<8> Setting the ADCS bit to 0 during conversion aborts the current A/D conversion, and places the A/D converter in the standby state. While the ADCS bit is 0, input of a hardware trigger is ignored, and A/D conversion does not start in response.

<9> Setting the ADCE bit to 0 while the A/D converter is in the standby state places the converter in the stopped state.

Figure 12 - 26 Example of Timing of Operations in the Hardware Trigger No-wait Mode: Select Mode and Sequential Conversion Mode



Caution **When <5>, <6>, or <7> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when the stabilization wait time for the A/D power supply is automatically secured before A/D conversion starts in software trigger wait mode or hardware trigger wait mode. For details, see Table 12 - 3 A/D Conversion Time Selection (3/8) and Table 12 - 3 A/D Conversion Time Selection (4/8).**

12.6.10 Hardware trigger no-wait mode: Select mode and one-shot conversion mode

<1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 while conversion is stopped places the A/D converter in the standby state.

<2> To allow stabilization of the internal circuit, apply software control for a wait of 1 μs + 2 cycles of the conversion clock (f_{AD}), then set the ADCS bit in the ADM0 register to 1 to place the A/D converter in the hardware trigger standby state. Note that, while in this state, A/D conversion does not start even if the ADCS bit is set to 1.

<3> Input of a hardware trigger while the ADCS bit is 1 starts A/D conversion of the analog input specified by the analog input channel specification register (ADS).

<4> Once A/D conversion is complete, the result of conversion is stored in the A/D conversion result registers (ADCR, ADCRH, ADCR0, and ADCR0H), and an A/D conversion end interrupt request signal (INTAD) is generated.

<5> Once A/D conversion is complete, the converter is placed in the standby state with the ADCS bit remaining set to 1.

<6> Input of a hardware trigger during conversion aborts the current A/D conversion and starts a new conversion. Any data from the aborted conversion are discarded.

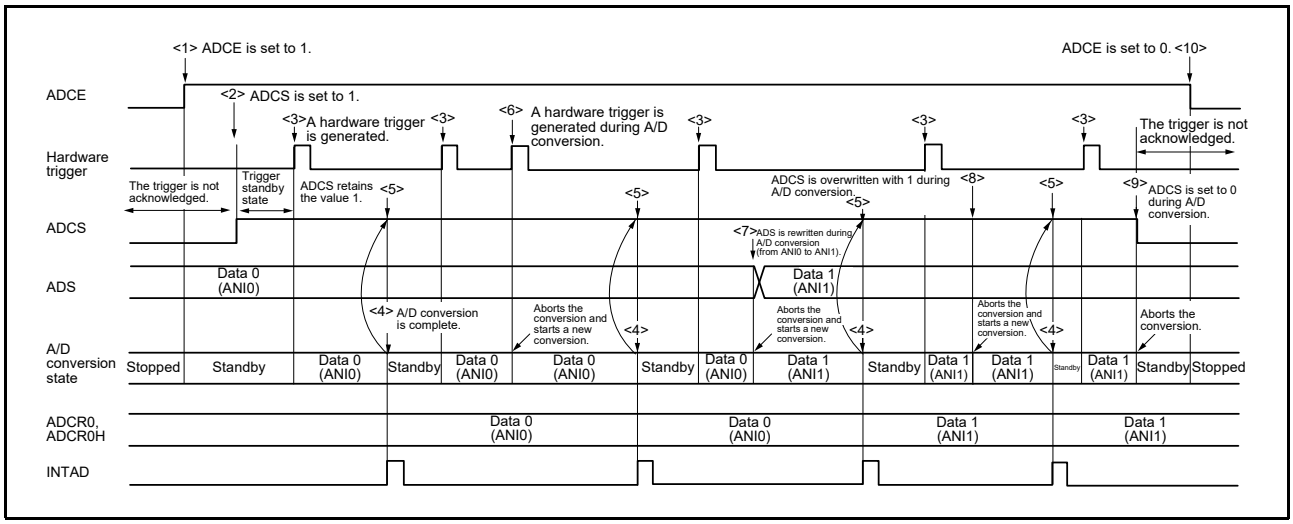
<7> Writing to the ADS register during conversion aborts the current A/D conversion and starts a new conversion for the analog input newly specified by the ADS register. Any data from the aborted conversion are discarded.

<8> Overwriting the ADCS bit with 1 during conversion aborts the current A/D conversion and starts a new conversion. Any data from the aborted conversion are discarded.

<9> Setting the ADCS bit to 0 during conversion aborts the current A/D conversion, and places the A/D converter in the standby state. While the ADCS bit is 0, input of a hardware trigger is ignored, and A/D conversion does not start in response.

<10> Setting the ADCE bit to 0 while the A/D converter is in the standby state places the converter in the stopped state.

Figure 12 - 27 Example of Timing of Operations in the Hardware Trigger No-wait Mode: Select Mode and One-shot Conversion Mode



Caution When <6>, <7>, or <8> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when the stabilization wait time for the A/D power supply is automatically secured before A/D conversion starts in software trigger wait mode or hardware trigger wait mode. For details, see Table 12 - 3 A/D Conversion Time Selection (3/8) and Table 12 - 3 A/D Conversion Time Selection (4/8).

12.6.11 Hardware trigger no-wait mode: Scan mode and sequential conversion mode

<1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 while conversion is stopped places the A/D converter in the standby state.

<2> To allow stabilization of the internal circuit, apply software control for a wait of 1 μs + 2 cycles of the conversion clock (f_{AD}), then set the ADCS bit in the ADM0 register to 1 to place the A/D converter in the hardware trigger standby state. Note that, while in this state, A/D conversion does not start even if the ADCS bit is set to 1.

<3> Input of a hardware trigger while the ADCS bit is 1 starts A/D conversion on the four analog input channels specified as scan 0 to scan 3 by the analog input channel specification register (ADS). A/D conversion proceeds sequentially, starting with the channel specified as scan 0.

<4> The four analog input channels are sequentially converted. Each time a conversion is completed, the result of <R> conversion is stored in the A/D conversion result registers (ADCRn and ADCRnH), and an A/D conversion end interrupt request signal (INTAD) is generated immediately after A/D conversion of the four channels is complete. On completion of conversion for all four channels, the next A/D conversion of the specified channels is automatically started.

<5> Input of a hardware trigger during conversion aborts the current A/D conversion and starts a new conversion from the first channel. Any data from the aborted conversion are discarded.

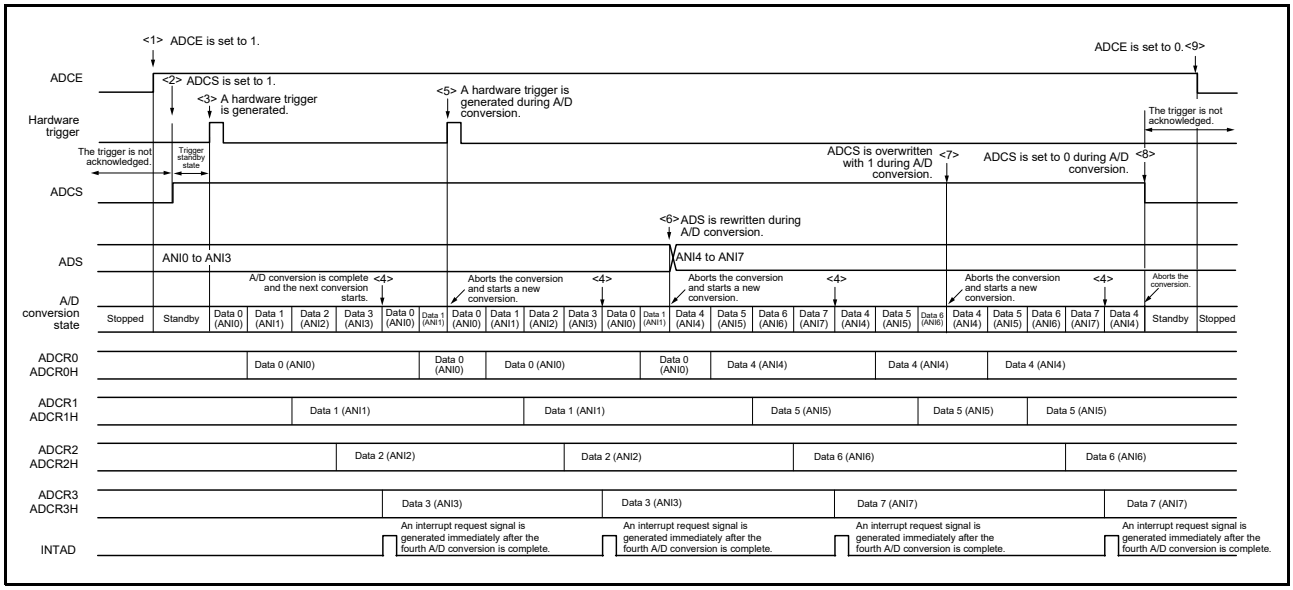
<6> Writing to the ADS register during conversion aborts the current A/D conversion and starts a new conversion from the first channel newly specified by the ADS register. Any data from the aborted conversion are discarded.

<7> Overwriting the ADCS bit with 1 during conversion aborts the current A/D conversion and starts a new conversion. Any data from the aborted conversion are discarded.

<8> Setting the ADCS bit to 0 during conversion aborts the current A/D conversion, and places the A/D converter in the standby state. While the ADCS bit is 0, input of a hardware trigger is ignored, and A/D conversion does not start in response.

<9> Setting the ADCE bit to 0 while the A/D converter is in the standby state places the converter in the stopped state.

Figure 12 - 28 Example of Timing of Operations in the Hardware Trigger No-wait Mode: Scan Mode and Sequential Conversion Mode



Caution When <5>, <6>, or <7> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when the stabilization wait time for the A/D power supply is automatically secured before A/D conversion starts in software trigger wait mode or hardware trigger wait mode. For details, see Table 12 - 3 A/D Conversion Time Selection (3/8) and Table 12 - 3 A/D Conversion Time Selection (4/8).

12.6.12 Hardware trigger no-wait mode: Scan mode and one-shot conversion mode

<1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 while conversion is stopped places the A/D converter in the standby state.

<2> To allow stabilization of the internal circuit, apply software control for a wait of 1 μs + 2 cycles of the conversion clock (f_{AD}), then set the ADCS bit in the ADM0 register to 1 to place the A/D converter in the hardware trigger standby state. Note that, while in this state, A/D conversion does not start even if the ADCS bit is set to 1.

<3> Input of a hardware trigger while the ADCS bit is 1 starts A/D conversion on the four analog input channels specified as scan 0 to scan 3 by the analog input channel specification register (ADS). A/D conversion proceeds sequentially, starting with the channel specified as scan 0.

<4> The four analog input channels are sequentially converted. Each time a conversion is completed, the result of <R> conversion is stored in the A/D conversion result registers (ADCRn and ADCRnH), and an A/D conversion end interrupt request signal (INTAD) is generated immediately after A/D conversion of the four channels is complete.

<5> Once A/D conversion for all four channels is complete, the converter is placed in the standby state with the ADCS bit remaining set to 1.

<6> Input of a hardware trigger during conversion aborts the current A/D conversion and starts a new conversion from the first channel. Any data from the aborted conversion are discarded.

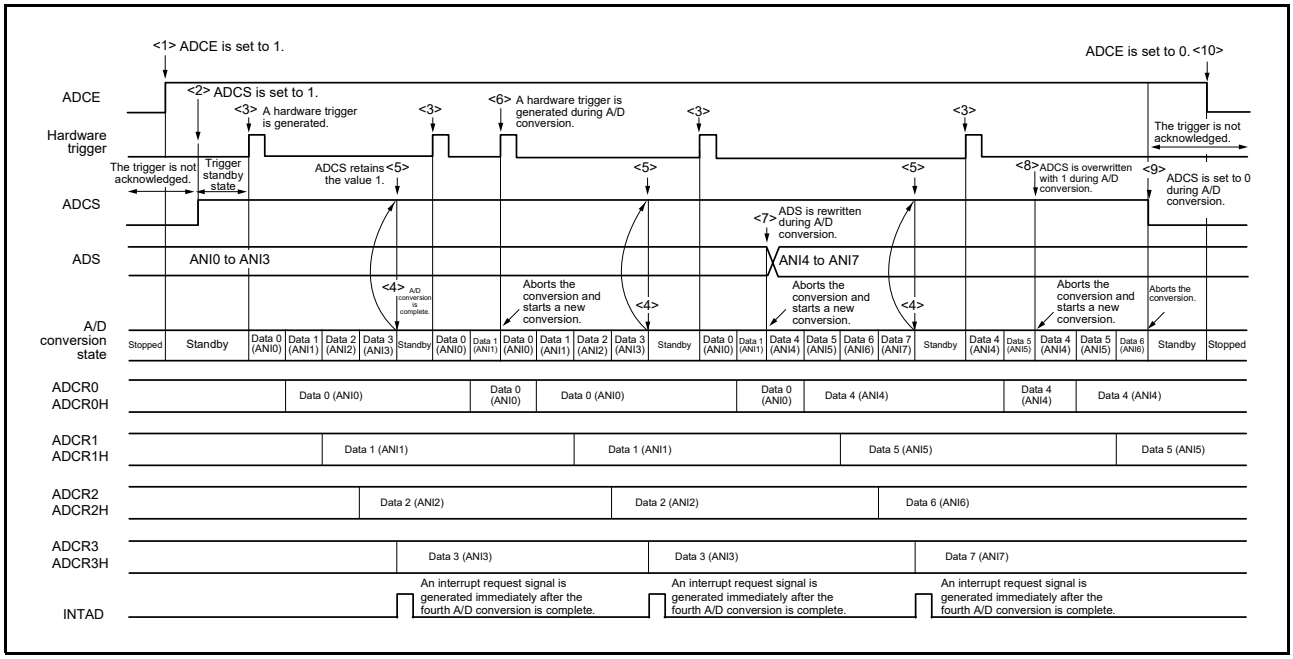
<7> Writing to the ADS register during conversion aborts the current A/D conversion and starts a new conversion from the first channel newly specified by the ADS register. Any data from the aborted conversion are discarded.

<8> Overwriting the ADCS bit with 1 during conversion aborts the current A/D conversion and starts a new conversion from the first channel. Any data from the aborted conversion are discarded.

<9> Setting the ADCS bit to 0 during conversion aborts the current A/D conversion, and places the A/D converter in the standby state. While the ADCS bit is 0, input of a hardware trigger is ignored, and A/D conversion does not start in response.

<10> Setting the ADCE bit to 0 while the A/D converter is in the standby state places the converter in the stopped state.

Figure 12 - 29 Example of Timing of Operations in the Hardware Trigger No-wait Mode: Scan Mode and One-shot Conversion Mode



Caution When <6>, <7>, or <8> is detected while conversion is in progress, conversion is automatically restarted from the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when the stabilization wait time for the A/D power supply is automatically secured before A/D conversion starts in software trigger wait mode or hardware trigger wait mode. For details, see Table 12 - 3 A/D Conversion Time Selection (3/8) and Table 12 - 3 A/D Conversion Time Selection (4/8).

12.6.13 Hardware trigger wait mode: Select mode and sequential conversion mode

<1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 while conversion is stopped places the A/D converter in the hardware trigger standby state.

<2> Input of a hardware trigger while in the hardware trigger standby state starts A/D conversion of the analog input specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 in response to the input of a hardware trigger.

<3> Once A/D conversion is complete, the conversion result is stored in the A/D conversion result registers (ADCR, ADCRH, ADCR0, and ADCR0H), an A/D conversion end interrupt request signal (INTAD) is generated, and the next A/D conversion automatically starts without a hardware trigger, simultaneously.

<4> Input of a hardware trigger during conversion aborts the current A/D conversion and starts a new conversion. Any data from the aborted conversion are discarded.

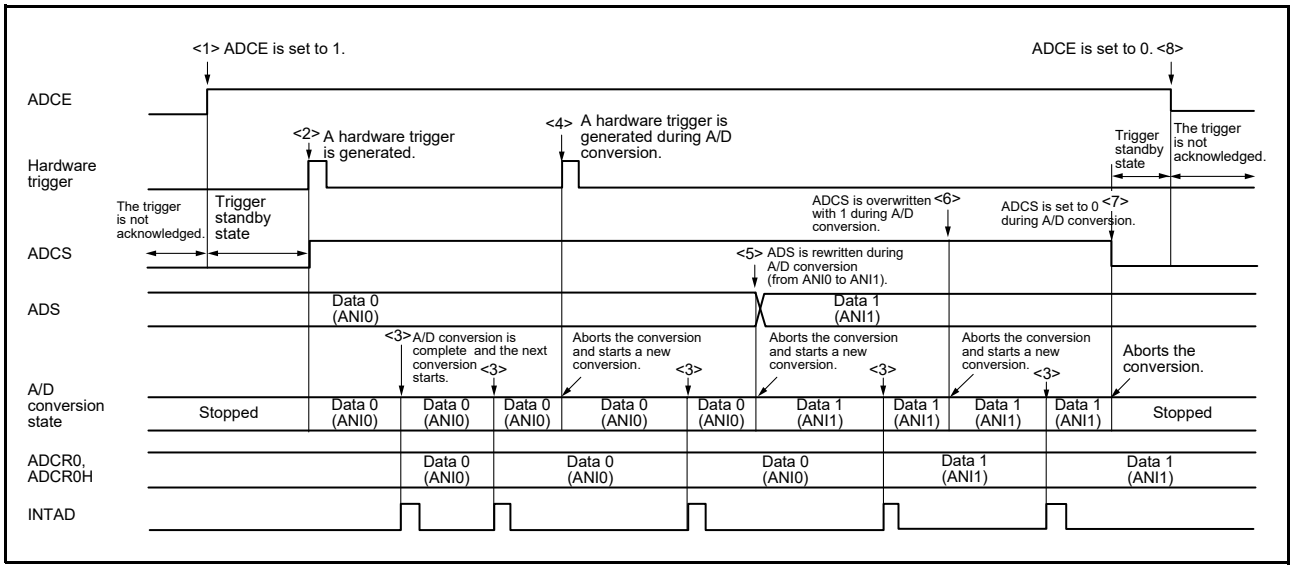
<5> Writing to the ADS register during conversion aborts the current A/D conversion and starts a new conversion for the analog input newly specified by the ADS register. Any data from the aborted conversion are discarded.

<6> Overwriting the ADCS bit with 1 during conversion aborts the current A/D conversion and starts a new conversion. Any data from the aborted conversion are discarded.

<7> Setting the ADCS bit to 0 during conversion aborts the current A/D conversion, and places the A/D converter in the hardware trigger standby state.

<8> Setting the ADCE bit to 0 while the A/D converter is in the hardware trigger standby state places the converter in the stopped state. While the ADCE bit is 0, input of a hardware trigger is ignored, and A/D conversion does not start in response.

Figure 12 - 30 Example of Timing of Operations in the Hardware Trigger Wait Mode: Select Mode and Sequential Conversion Mode



Caution When <4>, <5>, or <6> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed since the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when the stabilization wait time for the A/D power supply is automatically secured before A/D conversion starts in software trigger wait mode or hardware trigger wait mode. For details, see Table 12 - 3 A/D Conversion Time Selection (3/8) and Table 12 - 3 A/D Conversion Time Selection (4/8).

12.6.14 Hardware trigger wait mode: Select mode and one-shot conversion mode

<1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 while conversion is stopped places the A/D converter in the hardware trigger standby state.

<2> Input of a hardware trigger while in the hardware trigger standby state starts A/D conversion of the analog input specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 in response to the input of a hardware trigger.

<3> Once A/D conversion is complete, the conversion result is stored in the A/D conversion result registers (ADCR, ADCRH, ADCR0, and ADCR0H), and an A/D conversion end interrupt request signal (INTAD) is generated.

<4> Once A/D conversion is complete, the ADCS bit is automatically cleared to 0, and the converter is placed in the standby state.

<5> Input of a hardware trigger during conversion aborts the current A/D conversion and starts a new conversion. Any data from the aborted conversion are discarded.

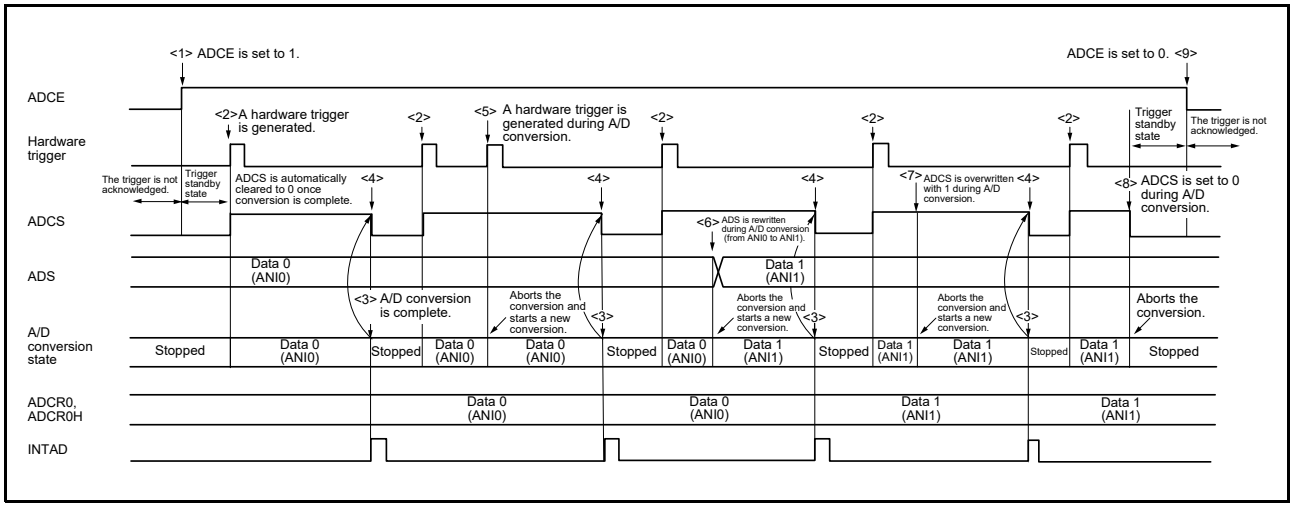
<6> Writing to the ADS register during conversion aborts the current A/D conversion and starts a new conversion for the analog input newly specified by the ADS register. Any data from the aborted conversion are discarded.

<7> Overwriting the ADCS bit with 1 during conversion aborts the current A/D conversion and starts a new conversion. Any data from the aborted conversion are discarded.

<8> Setting the ADCS bit to 0 during conversion aborts the current A/D conversion, and places the A/D converter in the hardware trigger standby state.

<9> Setting the ADCE bit to 0 while the A/D converter is in the hardware trigger standby state places the converter in the stopped state. While the ADCE bit is 0, input of a hardware trigger is ignored, and A/D conversion does not start in response.

Figure 12 - 31 Example of Timing of Operations in the Hardware Trigger Wait Mode: Select Mode and One-shot Conversion Mode



Caution 1. When <5>, <6>, or <7> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed since the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when the stabilization wait time for the A/D power supply is automatically secured before A/D conversion starts in software trigger wait mode or hardware trigger wait mode. For details, see Table 12 - 3 A/D Conversion Time Selection (3/8) and Table 12 - 3 A/D Conversion Time Selection (4/8).

Caution 2. The setting of ADISS being 1 (the input source is temperature sensor output voltage or internal reference voltage) cannot be used in the hardware trigger wait mode: select mode and one-shot conversion mode.

12.6.15 Hardware trigger wait mode: Scan mode and sequential conversion mode

<1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 while conversion is stopped places the A/D converter in the hardware trigger standby state.

<2> Input of a hardware trigger while in the hardware trigger standby state starts A/D conversion on the four analog input channels specified as scan 0 to scan 3 by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 in response to the input of a hardware trigger. A/D conversion proceeds sequentially, starting with the channel specified as scan 0.

<3> The four analog input channels are sequentially converted. Each time a conversion is completed, the result of
<R> conversion is stored in the A/D conversion result registers (ADCRn and ADCRnH), and an A/D conversion end interrupt request signal (INTAD) is generated immediately after A/D conversion of the four channels is complete. On completion of conversion for all four channels, the next A/D conversion of the specified channels is automatically started.

<4> Input of a hardware trigger during conversion aborts the current A/D conversion and starts a new conversion from the first channel. Any data from the aborted conversion are discarded.

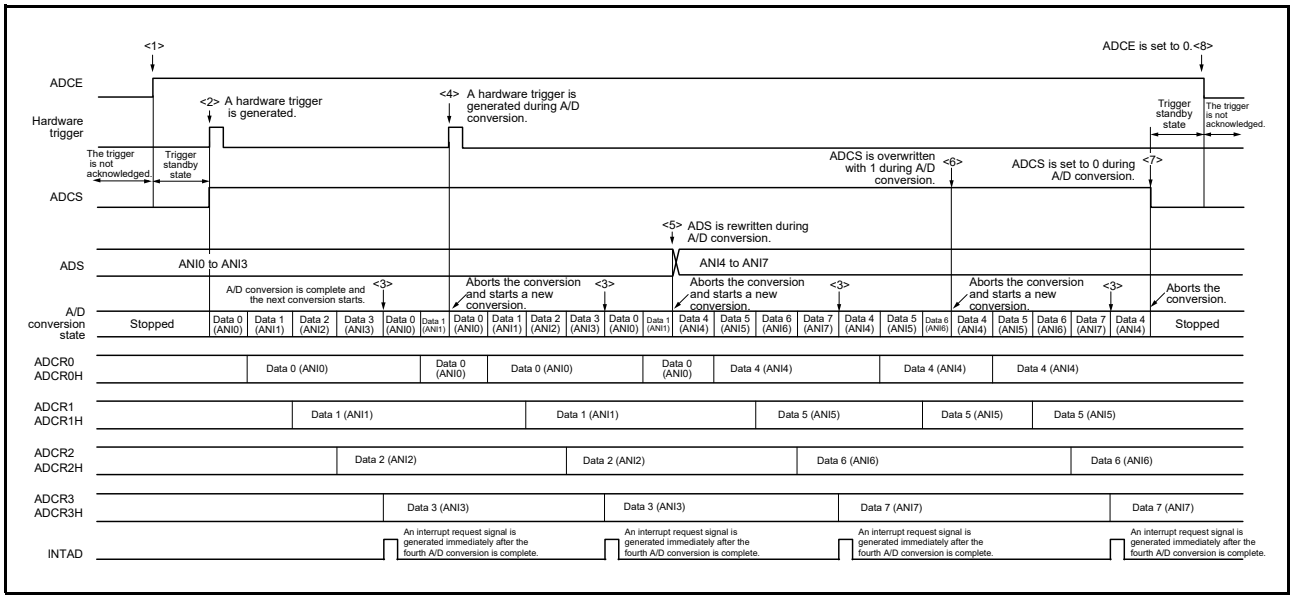
<5> Writing to the ADS register during conversion aborts the current A/D conversion and starts a new conversion from the first channel newly specified by the ADS register. Any data from the aborted conversion are discarded.

<6> Overwriting the ADCS bit with 1 during conversion aborts the current A/D conversion and starts a new conversion. Any data from the aborted conversion are discarded.

<7> Setting the ADCS bit to 0 during conversion aborts the current A/D conversion, and places the A/D converter in the hardware trigger standby state.

<8> Setting the ADCE bit to 0 while the A/D converter is in the hardware trigger standby state places the converter in the stopped state. While the ADCE bit is 0, input of a hardware trigger is ignored, and A/D conversion does not start in response.

Figure 12 - 32 Example of Timing of Operations in the Hardware Trigger Wait Mode: Scan Mode and Sequential Conversion Mode



Caution When <4>, <5>, or <6> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed since the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when the stabilization wait time for the A/D power supply is automatically secured before A/D conversion starts in software trigger wait mode or hardware trigger wait mode. For details, see Table 12 - 3 A/D Conversion Time Selection (3/8) and Table 12 - 3 A/D Conversion Time Selection (4/8).

12.6.16 Hardware trigger wait mode: Scan mode and one-shot conversion mode

<1> Setting the ADCE bit of A/D converter mode register 0 (ADM0) to 1 while conversion is stopped places the A/D converter in the hardware trigger standby state.

<2> Input of a hardware trigger while in the hardware trigger standby state starts A/D conversion on the four analog input channels specified as scan 0 to scan 3 by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 in response to the input of a hardware trigger. A/D conversion proceeds sequentially, starting with the channel specified as scan 0.

<3> The four analog input channels are sequentially converted. Each time a conversion is completed, the result of
<R> conversion is stored in the A/D conversion result registers (ADCRn and ADCRnH), and an A/D conversion end interrupt request signal (INTAD) is generated immediately after A/D conversion of the four channels is complete.

<4> Once A/D conversion is complete, the ADCS bit is automatically cleared to 0, and the converter is placed in the stopped state.

<5> Input of a hardware trigger during conversion aborts the current A/D conversion and starts a new conversion from the first channel. Any data from the aborted conversion are discarded.

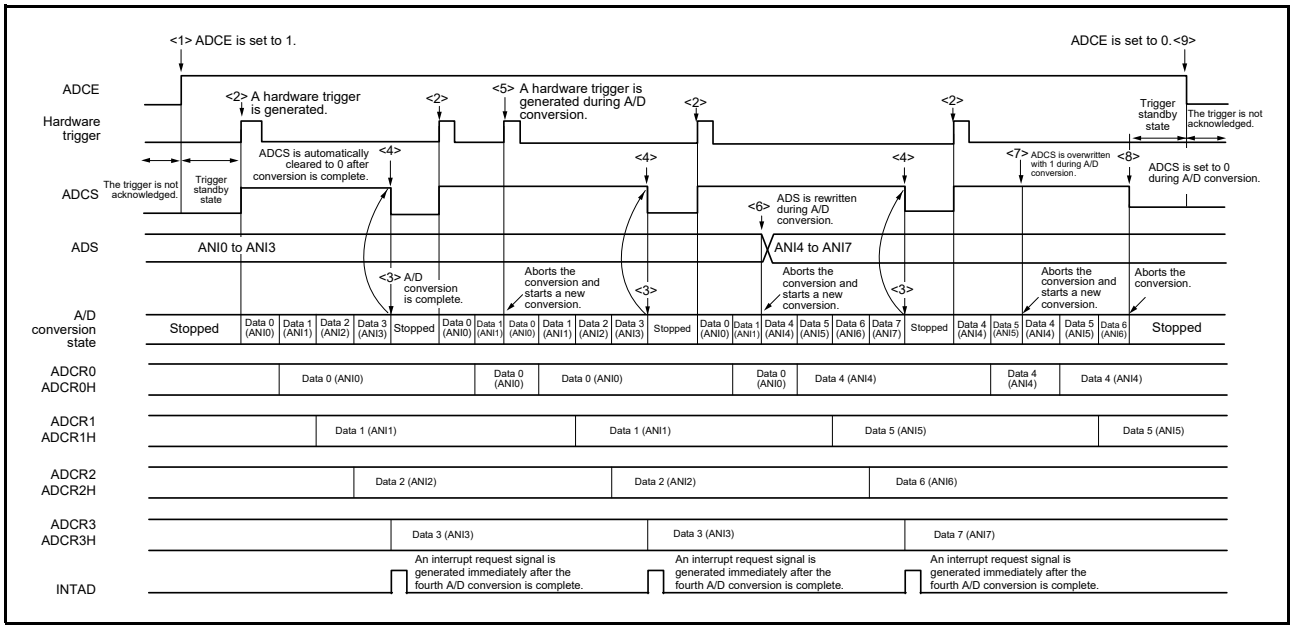
<6> Writing to the ADS register during conversion aborts the current A/D conversion and starts a new conversion from the first channel newly specified by the ADS register. Any data from the aborted conversion are discarded.

<7> Overwriting the ADCS bit with 1 during conversion aborts the current A/D conversion and starts a new conversion. Any data from the aborted conversion are discarded.

<8> Setting the ADCS bit to 0 during conversion aborts the current A/D conversion, and places the A/D converter in the hardware trigger standby state.

<9> Setting the ADCE bit to 0 while the A/D converter is in the hardware trigger standby state places the converter in the stopped state. While the ADCE bit is 0, input of a hardware trigger is ignored, and A/D conversion does not start in response.

Figure 12 - 33 Example of Timing of Operations in the Hardware Trigger Wait Mode: Scan Mode and One-shot Conversion Mode



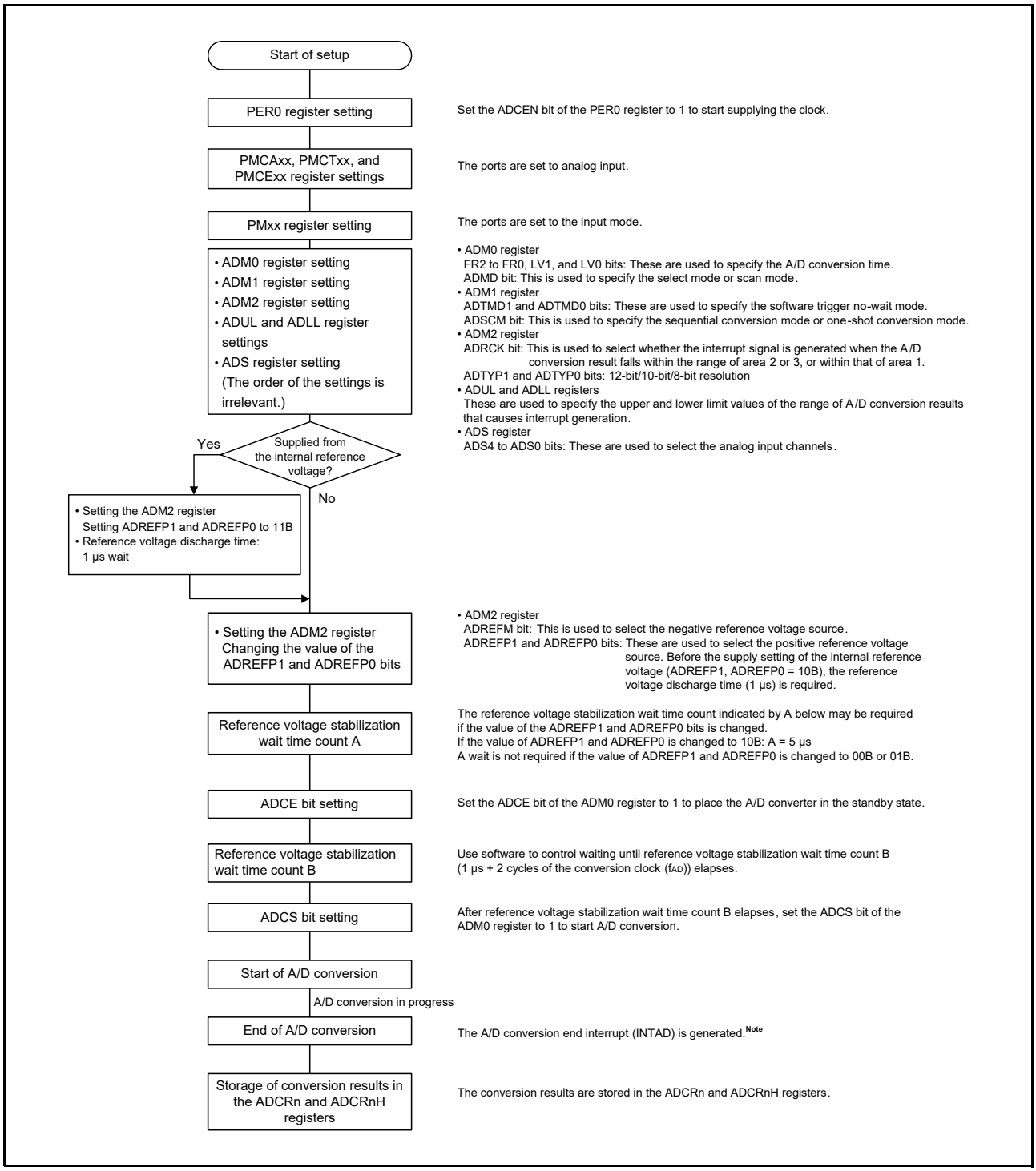
Caution When <5>, <6>, or <7> is detected during conversion operation, conversion is restarted automatically after the stabilization wait time has passed since the rising edge of the next cycle of the conversion clock (f_{AD}). The conversion time at the first conversion operation restarted is the same as that when the stabilization wait time for the A/D power supply is automatically secured before A/D conversion starts in software trigger wait mode or hardware trigger wait mode. For details, see Table 12 - 3 A/D Conversion Time Selection (3/8) and Table 12 - 3 A/D Conversion Time Selection (4/8).

12.7 Flows of Setting the A/D Converter Modes

This section describes the flows of setting each of the A/D converter modes.

12.7.1 Settings in software trigger no-wait mode

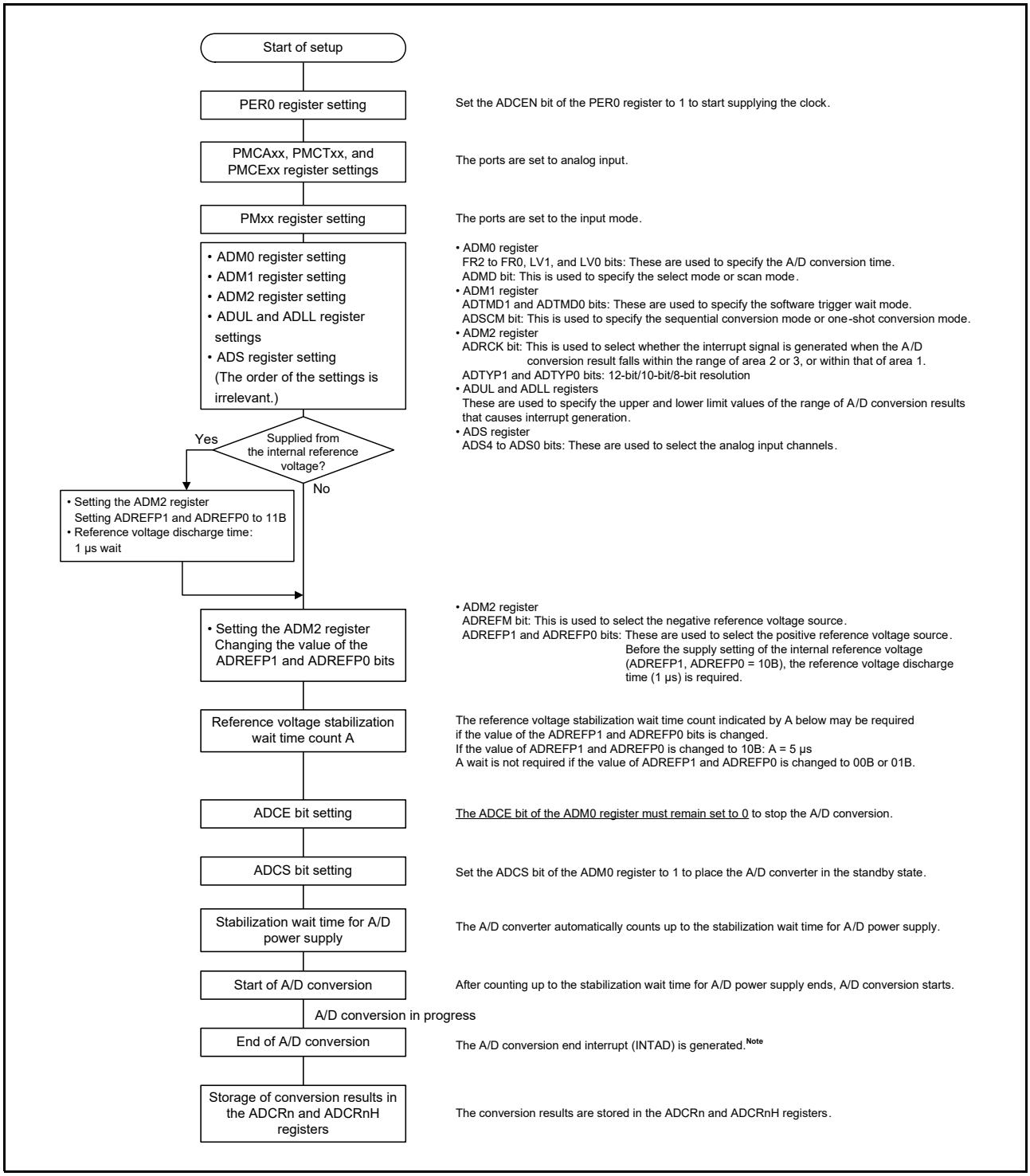
Figure 12 - 34 Settings in Software Trigger No-wait Mode



Note Depending on the settings of the ADRCK bit and ADUL and ADLL registers, the interrupt signal may not be generated. In such cases, the results of A/D conversion are not stored in the ADCRn and ADCRnH registers.

12.7.2 Settings in software trigger wait mode

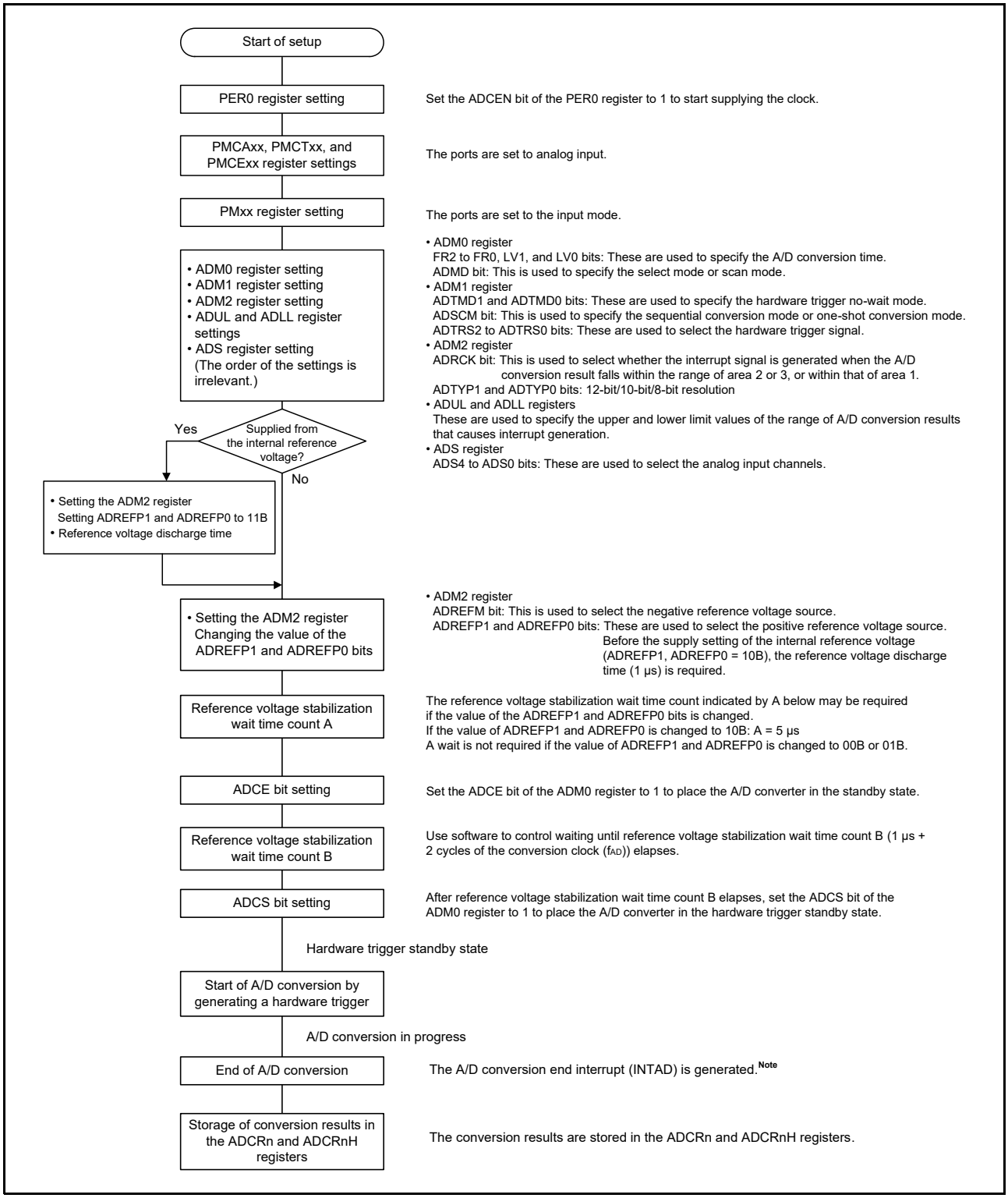
Figure 12 - 35 Settings in Software Trigger Wait Mode



Note Depending on the settings of the ADRCK bit and ADUL and ADLL registers, the interrupt signal may not be generated. In such cases, the results of A/D conversion are not stored in the ADCRn and ADCRnH registers.

12.7.3 Settings in hardware trigger no-wait mode

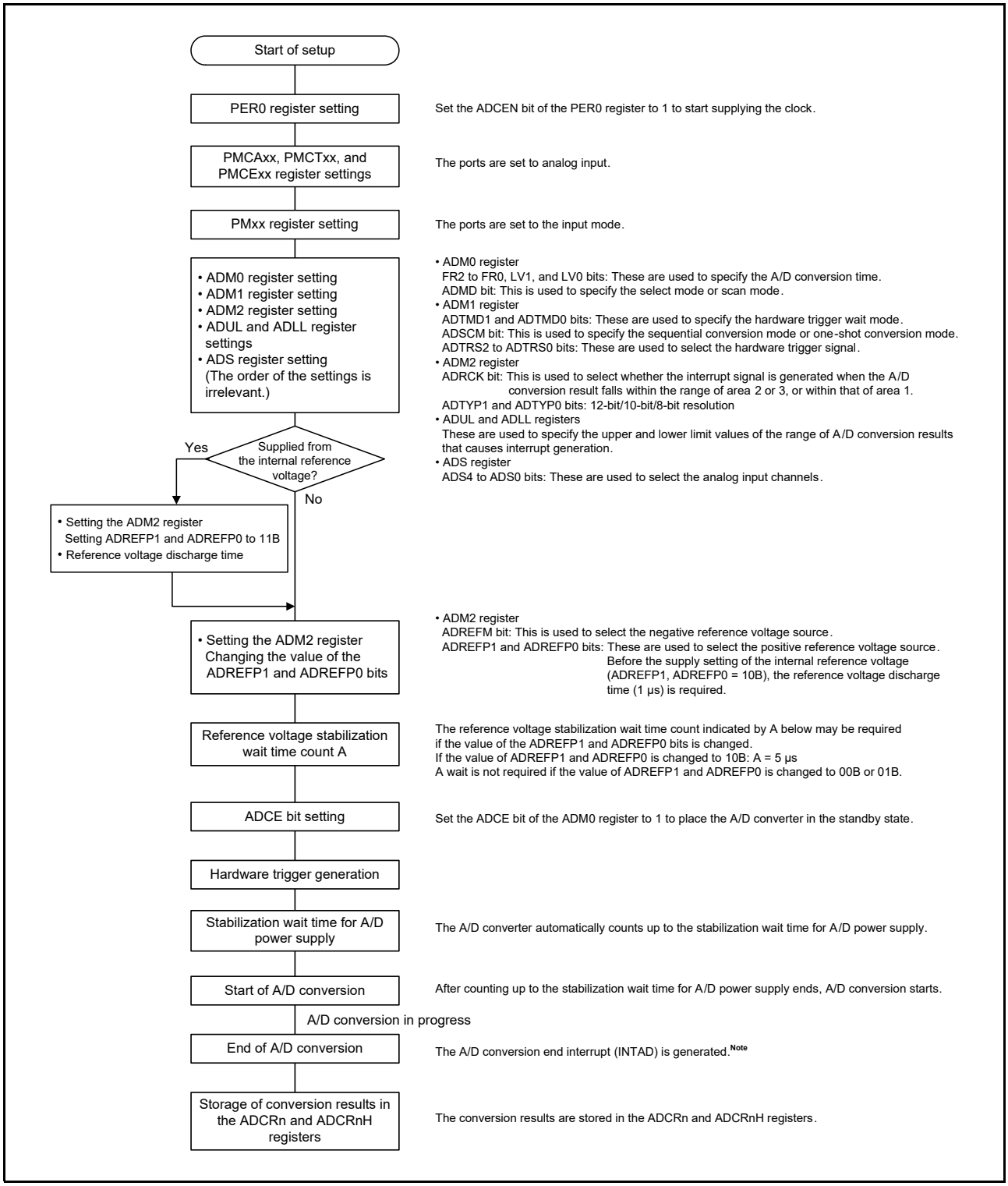
Figure 12 - 36 Settings in Hardware Trigger No-wait Mode



Note Depending on the settings of the ADRCK bit and ADUL and ADLL registers, the interrupt signal may not be generated. In <R> such cases, the results of A/D conversion are not stored in the ADCRn and ADCRnH registers.

12.7.4 Settings in hardware trigger wait mode

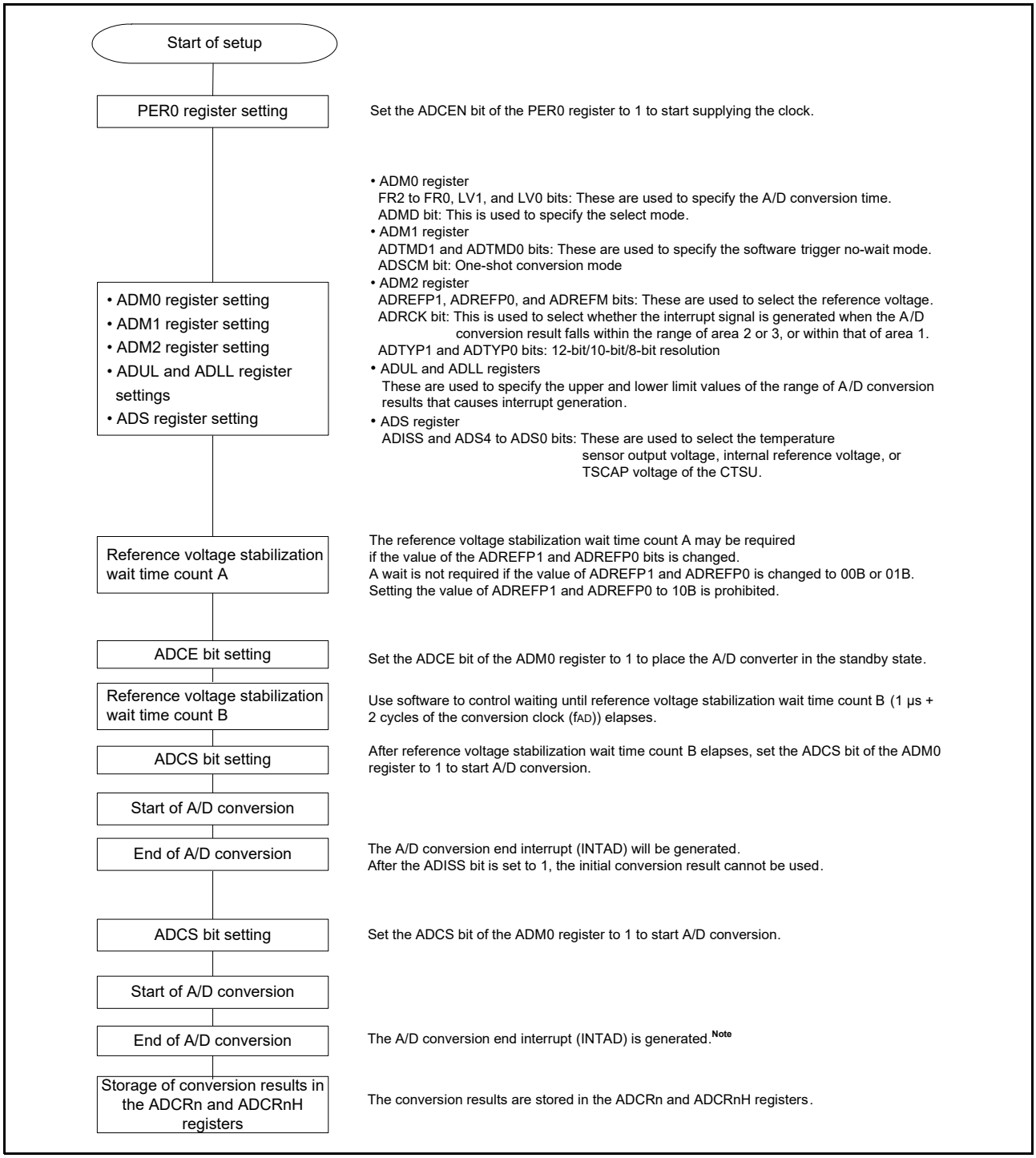
Figure 12 - 37 Settings in Hardware Trigger Wait Mode



Note Depending on the settings of the ADRCK bit and ADUL and ADLL registers, the interrupt signal may not be generated. In <R> such cases, the results of A/D conversion are not stored in the ADCRn and ADCRnH registers.

12.7.5 Example of settings when temperature sensor output voltage, internal reference voltage, or TSCAP voltage of the CTSU is selected in the software trigger no-wait mode and one-shot conversion mode

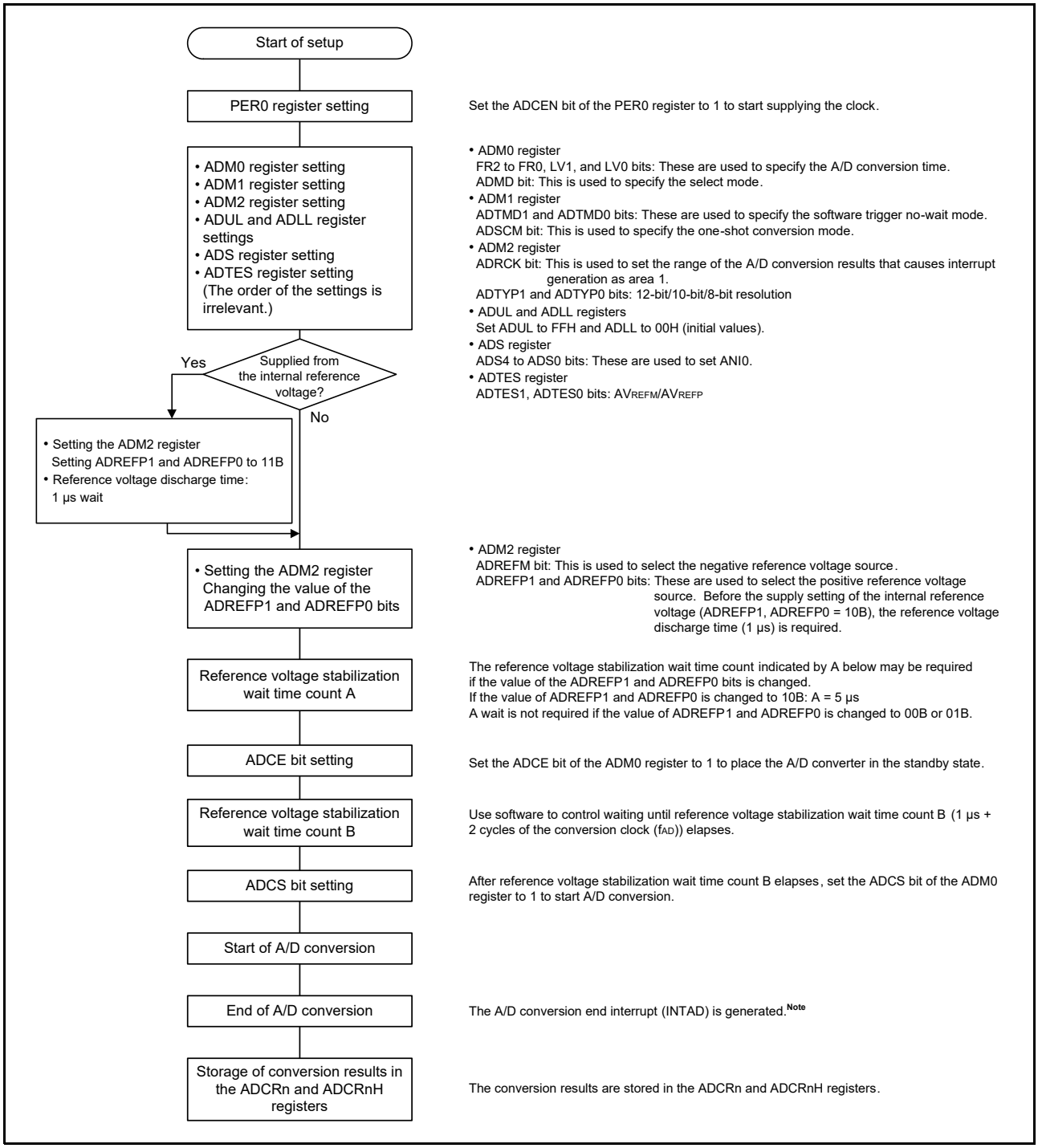
Figure 12 - 38 Settings When Selecting the Temperature Sensor Output Voltage, Internal Reference Voltage, or TSCAP Voltage of the CTSU



Note Depending on the settings of the ADRCK bit and ADUL and ADLL registers, the interrupt signal may not be generated. In <R> such cases, the results of A/D conversion are not stored in the ADCRn and ADCRnH registers.

12.7.6 Settings in test mode

Figure 12 - 39 Settings in Test Mode



Note Depending on the settings of the ADRCK bit and ADUL and ADLL registers, the interrupt signal may not be generated. In such cases, the results of A/D conversion are not stored in the ADCRn and ADCRnH registers.

Caution For the procedure for testing the A/D converter, see 27.3.10 Testing of the A/D converter.

12.8 SNOOZE Mode

In the SNOOZE mode, A/D conversion is triggered by inputting a software trigger or a hardware trigger in the STOP mode. Normally, A/D conversion is stopped while in the STOP mode, but, by using the SNOOZE mode, A/D conversion can be performed without operating the CPU. This is effective for reducing the operating current.

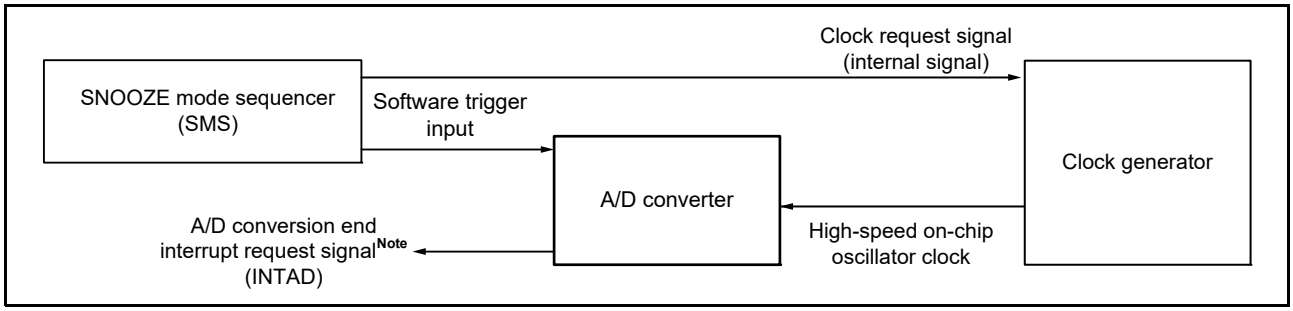
12.8.1 A/D conversion by inputting a software trigger

In the SNOOZE mode, A/D conversion is triggered by inputting a software trigger. A software trigger generated by the SNOOZE mode sequencer (SMS) is used as an input trigger for A/D conversion. When performing A/D conversion by inputting a software trigger in SNOOZE mode, the software trigger wait mode can only be used, with the following conversion modes:

- Select mode with one-shot conversion
- Select mode with sequential conversion
- Scan mode with one-shot conversion
- Scan mode with sequential conversion

Caution The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock is selected for f_{CLK}.

Figure 12 - 40 Block Diagram When Using SNOOZE Mode in Software Trigger Wait Mode



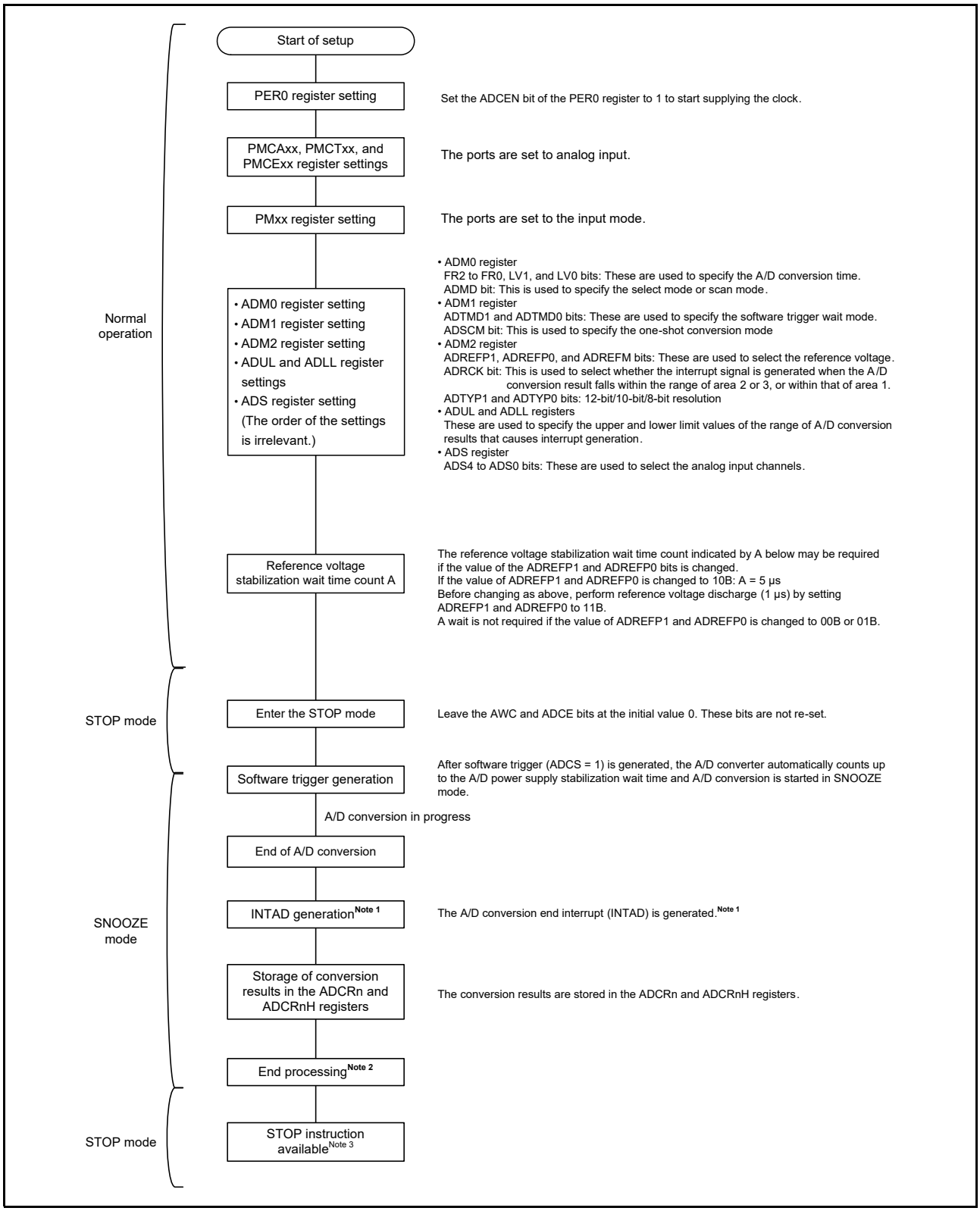
When using the SNOOZE mode, the initial setting of each register is specified before switching to the STOP mode. For details about these settings, see **12.7.2 Settings in software trigger wait mode**. If a software trigger (ADCS = 1) is input after switching to the STOP mode, the high-speed on-chip oscillator clock is supplied to the A/D converter. After supplying this clock, the A/D converter automatically counts up to the A/D power supply stabilization wait time, and then A/D conversion starts.

Note Clear the setting of the A/D conversion result comparison function (ADM2.ADRCK bit, and ADUL and ADLL registers) to the initial value so that an interrupt signal is generated each time conversion is completed.

- One-shot conversion mode (select/scan)
  When the STOP instruction is executed after the A/D conversion is completed and the interrupt request signal (INTAD) is generated, the supplied clock can be stopped.

- Sequential conversion mode (select/scan)
  When the ADCS bit is written with 0 after the A/D conversion is completed and the interrupt request signal (INTAD) is generated, the sequential conversion mode is stopped. After that, when the STOP instruction is executed after 2 cycles of the conversion clock (f_{AD}) have elapsed, the supplied clock can be stopped.

Figure 12 - 41 Flow of Setting up the SNOOZE Mode for Use with Software-triggered Conversion



<R>

Note 1. Clear the ADRCK bit, and ADUL and ADLL registers to the initial value in the initial setting so that an interrupt request signal (INTAD) is generated each time A/D conversion is completed.

Note 2. Sequential conversion mode requires the end processing. Write 0 to the ADCS bit after INTAD is generated. Then, after 2 cycles of the conversion clock (f_{AD}) have elapsed, the STOP instruction can be executed.

Note 3. If a software trigger is input after the STOP instruction, A/D conversion operation is again performed in the SNOOZE mode.

12.8.2 A/D conversion by inputting a hardware trigger

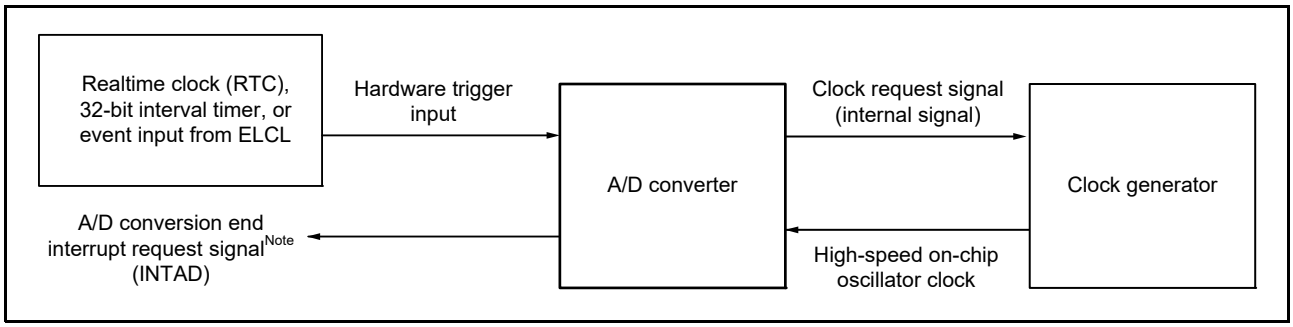
In the SNOOZE mode, A/D conversion is triggered by inputting a hardware trigger. When performing A/D conversion by inputting a hardware trigger in SNOOZE mode, the hardware trigger wait mode can only be used, with the following conversion modes:

- Select mode with one-shot conversion
- Scan mode with one-shot conversion

If the A/D conversion result range is specified using the ADUL and ADLL registers, A/D conversion results can be judged at a certain interval of time. Using this function enables power supply voltage monitoring and input key judgment based on A/D inputs.

Caution The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock is selected for f_{CLK}.

Figure 12 - 42 Block Diagram When Using SNOOZE Mode in Hardware Trigger Wait Mode



When using the SNOOZE mode, the initial setting of each register is specified before switching to the STOP mode. For details about these settings, see **Figure 12 - 45 Flow of Setting up the SNOOZE Mode for Use with Hardware-triggered Conversion**. Just before moving to STOP mode, set the AWC bit of A/D converter mode register 2 (ADM2) to 1. After the initial settings are specified, set the ADCE bit of A/D converter mode register 0 (ADM0) to 1. If a hardware trigger is input after switching to the STOP mode, the high-speed on-chip oscillator clock is supplied to the A/D converter. After supplying this clock, the A/D converter automatically counts up to the A/D power supply stabilization wait time, and then A/D conversion starts. The SNOOZE mode operation after A/D conversion is complete differs depending on whether an interrupt signal is generated**Note**.

<R> **Note** Depending on the setting of the A/D conversion result comparison function (ADM2.ADRCK bit, and ADUL and ADLL registers), the interrupt signal may not be generated.
When the 32-bit interval timer interrupt signal (INTITL) is selected as a hardware trigger, the detection flag in the ITLS0 register has to be cleared each time a 32-bit interval timer interrupt signal (INTITL) is generated. Therefore, clear the setting of the A/D conversion result comparison function (ADRCK bit, and ADUL and ADLL registers) to the initial value so that an A/D conversion end interrupt request signal (INTAD) is generated upon completion of A/D conversion.

Caution Select the hardware trigger signal from among the realtime clock interrupt signal (INTRTC), 32-bit interval timer interrupt signal (INTITL), and event input from the ELCL.

1. If an interrupt is generated after A/D conversion is complete

<R> If the A/D conversion result value is inside the range of values specified by the A/D conversion result comparison function (which is set up by using the ADRCK bit, and ADUL and ADLL registers), the A/D conversion end interrupt request signal (INTAD) is generated.

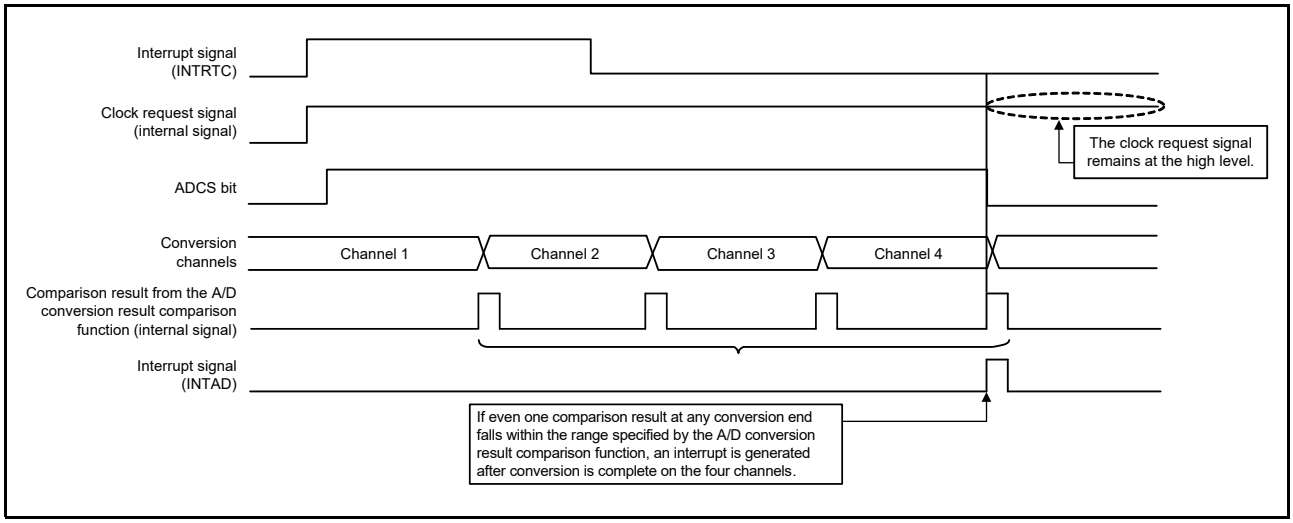
- While in the select mode

  When A/D conversion is complete and an A/D conversion end interrupt request signal (INTAD) is generated, the A/D converter returns to normal operation mode from SNOOZE mode. At this time, be sure to set the AWC bit of A/D converter mode register 2 (ADM2) to 0. If the AWC bit is left set to 1, A/D conversion will not start normally in the subsequent SNOOZE or normal operation mode.

- While in the scan mode

  If even one value of the A/D conversion results of the four channels falls within the range specified by the A/D conversion result comparison function, and A/D conversion end interrupt request signal (INTAD) is generated, the A/D converter switches from the SNOOZE mode to the normal operation mode. At this time, be sure to set the AWC bit of A/D converter mode register 2 (ADM2) to 0. If the AWC bit is left set to 1, A/D conversion will not start normally in the subsequent SNOOZE or normal operation mode.

Figure 12 - 43 Example of Operation When Interrupt Is Generated Once A/D Conversion Is Complete in the Scan Mode



2. If no interrupt is generated after A/D conversion is complete
   If the A/D conversion result value is outside the range of values specified by the A/D conversion result comparison function (which is set up by using the ADRCK bit, and ADUL and ADLL registers), the A/D conversion end interrupt request signal (INTAD) is not generated.

   - While in the select mode
     If the A/D conversion end interrupt request signal (INTAD) is not generated after A/D conversion is complete, the clock request signal (an internal signal) is automatically set to the low level, and supplying the high-speed on-chip oscillator clock stops. If a hardware trigger is input later, A/D conversion work is again performed in the SNOOZE mode.

   - While in the scan mode
     If the A/D conversion result values of the four channels do not fall within the range specified by the A/D conversion result comparison function even once, and the A/D conversion end interrupt request signal (INTAD) is not generated, the clock request signal (an internal signal) is automatically set to the low level after A/D conversion of the four channels is complete, and supplying the high-speed on-chip oscillator clock stops. If a hardware trigger is input later, A/D conversion work is again performed in the SNOOZE mode.

Figure 12 - 44 Example of Operation When No Interrupt Is Generated Once A/D Conversion Is Complete in the Scan Mode

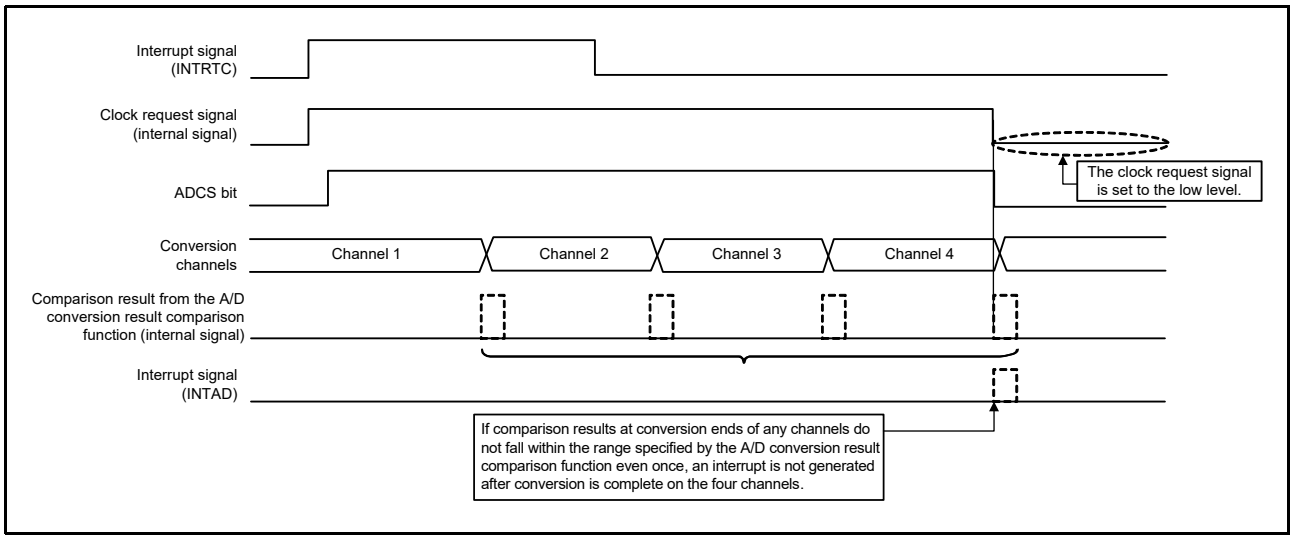
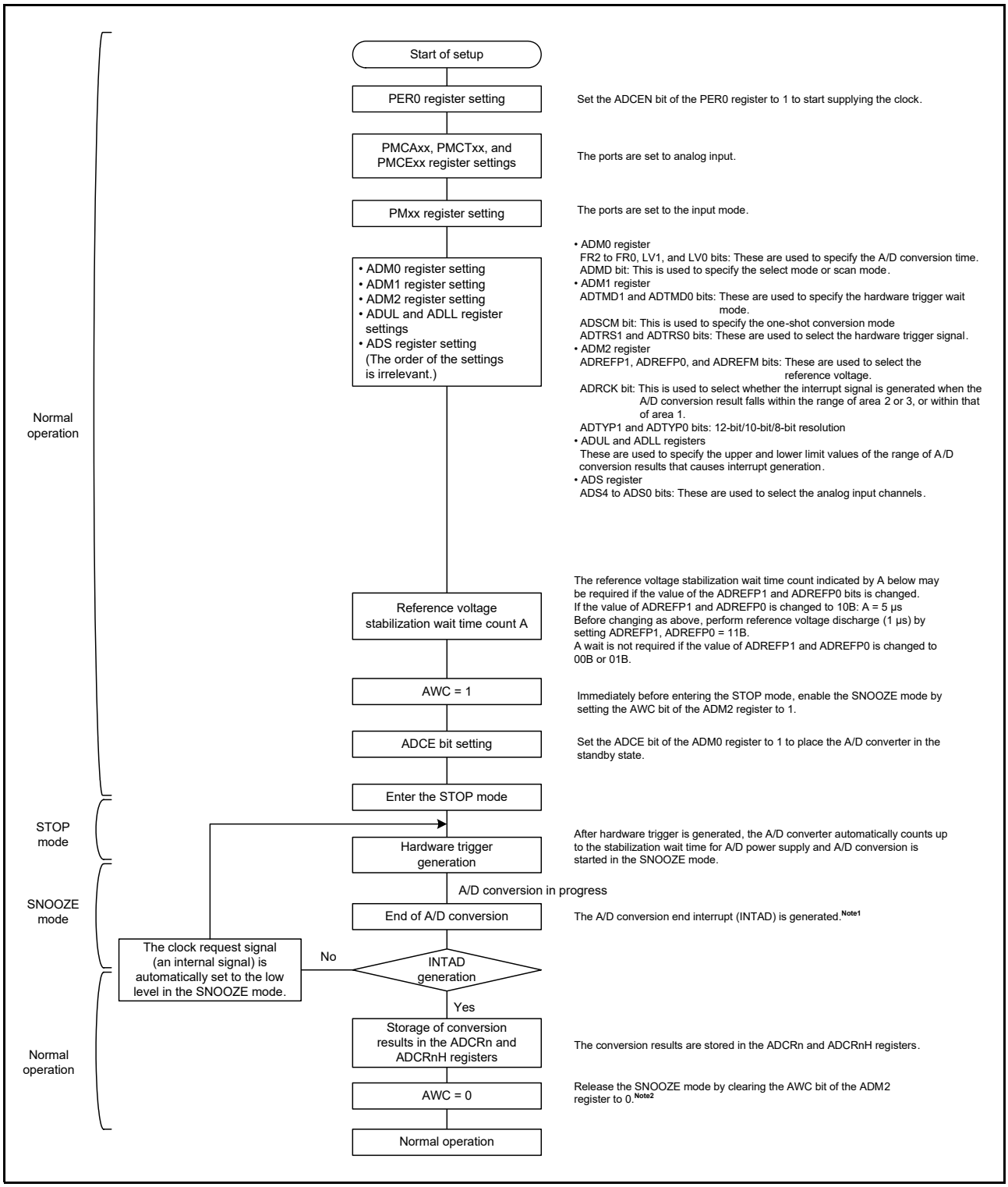


Figure 12 - 45 Flow of Setting up the SNOOZE Mode for Use with Hardware-triggered Conversion



<R> **Note 1.** If the A/D conversion end interrupt request signal (INTAD) is not generated depending on the settings of the ADRCK bit and ADUL and ADLL registers, the result is not stored in the ADCRn and ADCRnH registers. The A/D converter enters the STOP mode again. If a hardware trigger is input later, A/D conversion operation is again performed in the SNOOZE mode.

Note 2. If the AWC bit is left set to 1, A/D conversion will not start normally in the subsequent SNOOZE or normal operation mode. Be sure to set the AWC bit to 0.

12.9 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

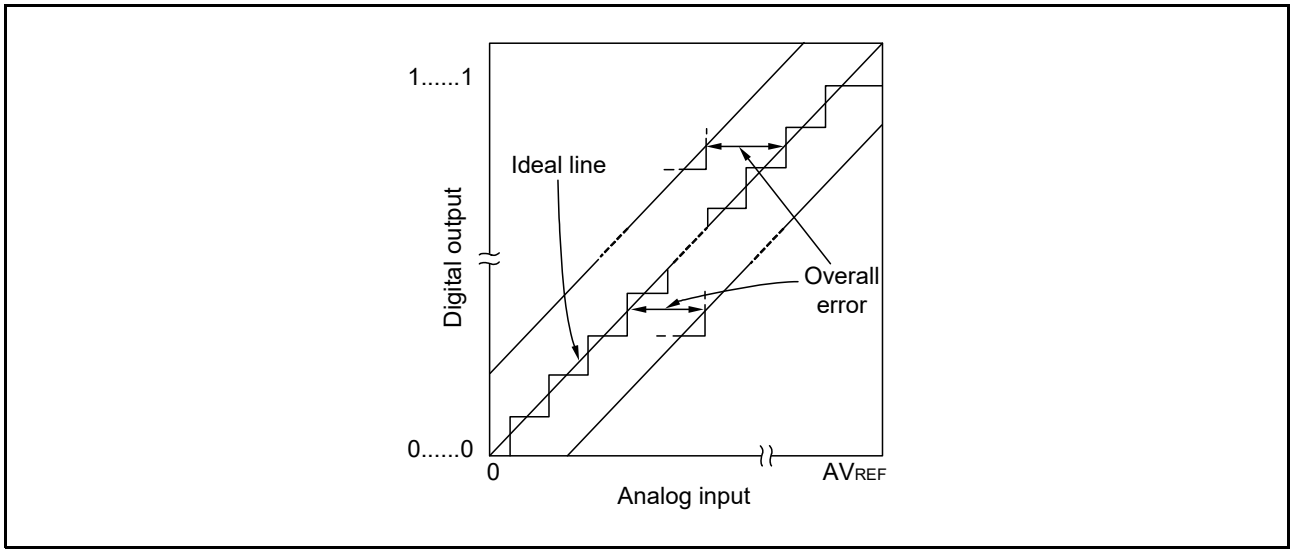
1. Resolution
   This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1 LSB (Least Significant Bit). The percentage of 1 LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

   1 LSB is as follows when the resolution is 12 bits.

   $$1\ LSB = 1/2^{12} = 1/4096$$
   $$\approx 0.024\%FSR$$

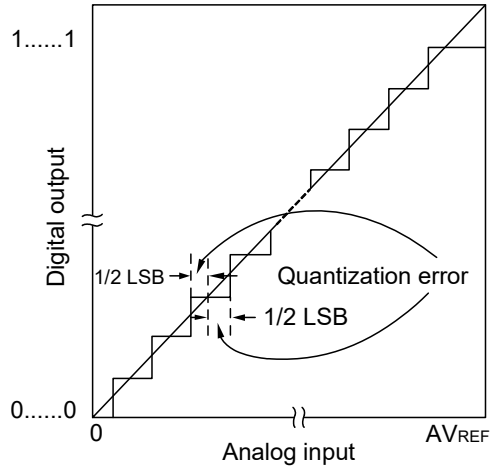
   Accuracy has no relation to resolution, but is determined by overall error.
2. Overall error
   This shows the maximum error value between the actual measured value and the theoretical value. Zero-scale error, full-scale error, integral linearity error, differential linearity error, and combinations of these express the overall error. Note that the quantization error is not included in the overall error in the characteristics table.

Figure 12 - 46 Overall Error



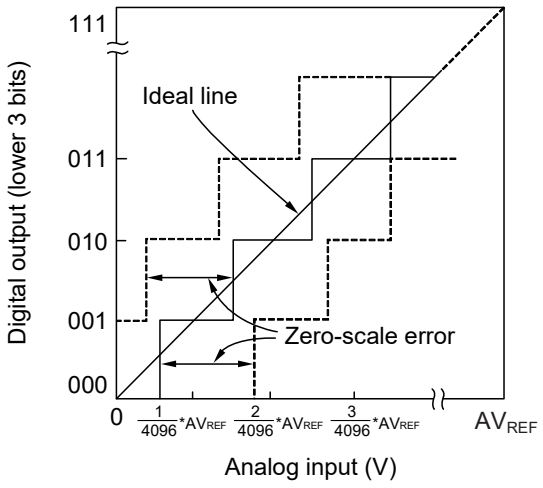
3. Quantization error
   When analog values are converted to digital values, a ±1/2 LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2 LSB is converted to the same digital code, so a quantization error cannot be avoided. Note that the quantization error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

Figure 12 - 47 Quantization Error



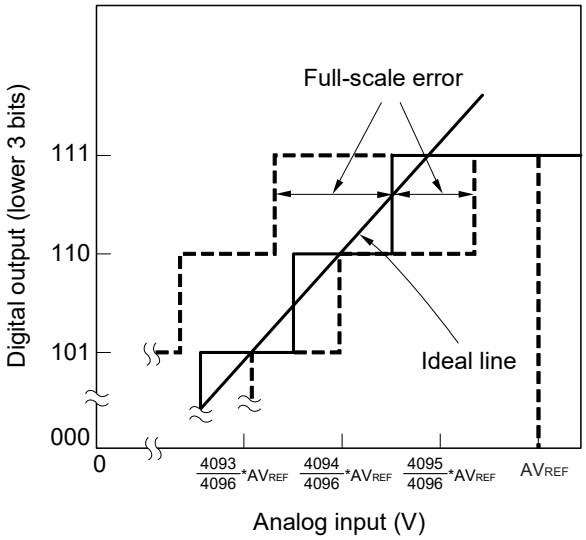
4. Zero-scale error
   This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2 LSB) when the digital output changes from 0.....000 to 0.....001. If the actual measurement value is greater than the theoretical value, it shows the difference between the actual measurement value of the analog input voltage and the theoretical value (3/2 LSB) when the digital output changes from 0……001 to 0……010.

Figure 12 - 48 Zero-scale Error



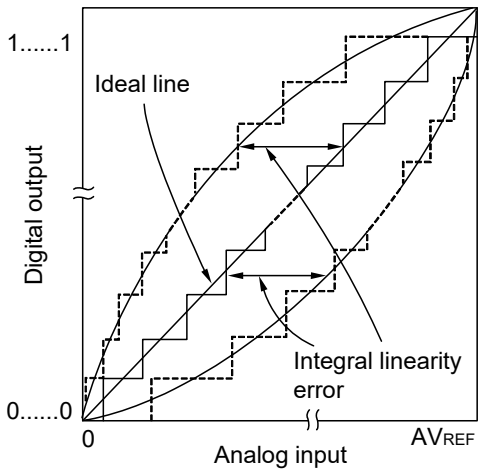
5. Full-scale error
   This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (Full-scale – 3/2 LSB) when the digital output changes from 1.....110 to 1.....111.

Figure 12 - 49 Full-scale Error



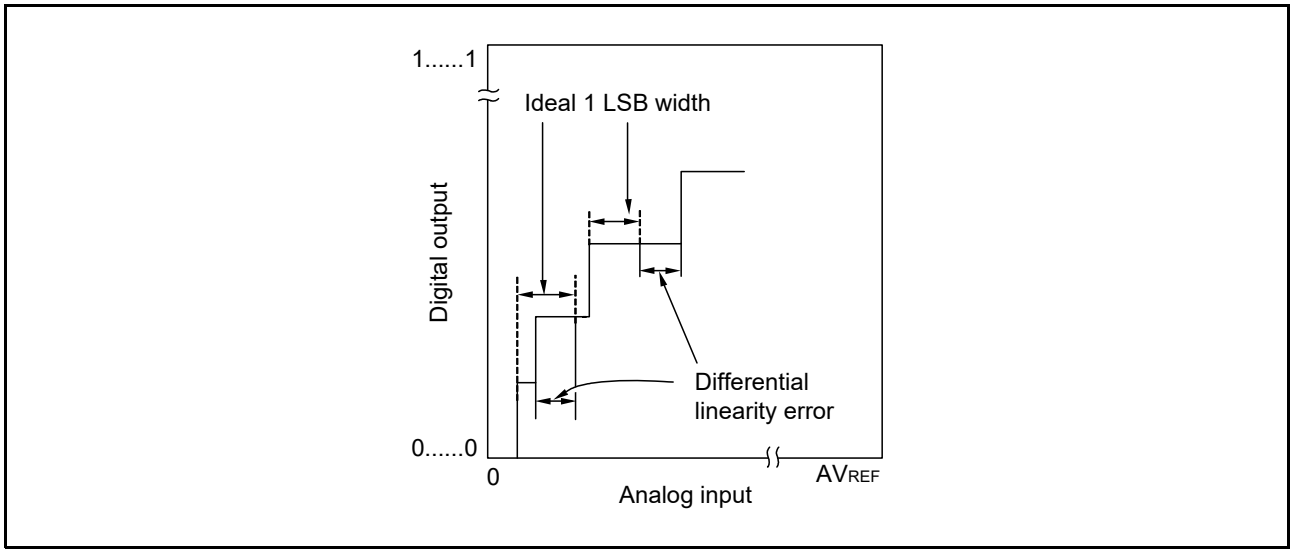
6. Integral linearity error
   This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

Figure 12 - 50 Integral Linearity Error

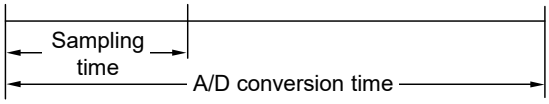


7. Differential linearity error
   While the ideal width of code output is 1 LSB, this indicates the difference between the actual measurement value and the ideal value of the width of output code.

Figure 12 - 51 Differential Linearity Error



8. Conversion time
   This expresses the time from the start of sampling to when the digital output is obtained. The sampling time is included in the conversion time in the characteristics table.
9. Sampling time
   This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.



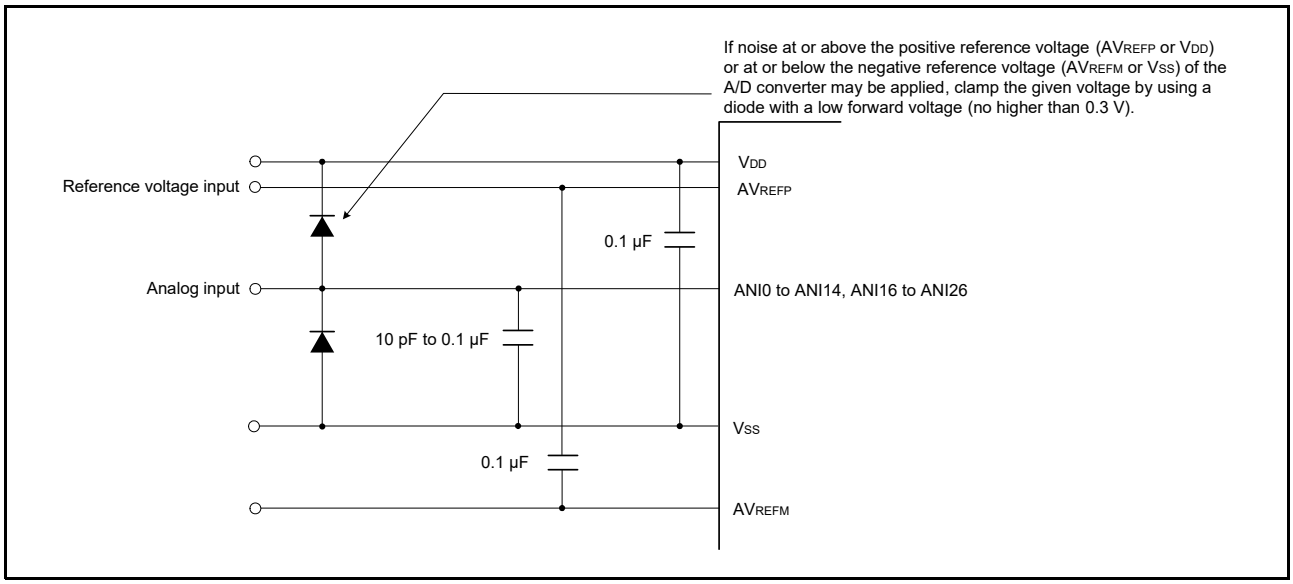
12.10 Points for Caution When the A/D Converter is to be Used

1. Operating current in STOP mode
   Shift to STOP mode after stopping the A/D converter (by setting the ADCS bit of A/D converter mode register 0 (ADM0) to 0). The operating current can be reduced by setting the ADCE bit of the ADM0 register to 0 at the same time. To restart from the standby state, clear the ADIF bit of interrupt request flag register 1H (IF1H) to 0 and start operation.
2. Input range of ANI0 to ANI14 and ANI16 to ANI26 pins
   Observe the rated range of the ANI0 to ANI14 and ANI16 to ANI26 pins input voltage. If a voltage exceeding V_{DD} and AV_{REFP} or a voltage lower than V_{SS} and AV_{REFM} (even in the range of absolute maximum ratings) is input to an analog input channel, the converted value of that channel becomes undefined. In addition, the converted values of the other channels may also be affected. When internal reference voltage is selected as the positive reference
<R> voltage of the A/D converter, do not input a voltage higher than the internal reference voltage to a pin selected by the ADS register. However, it is no problem that a voltage higher than the internal reference voltage is input to a pin not selected by the ADS register.

   Caution For details about the internal reference voltage, see Section 37 Electrical Characteristics.

3. Conflicting operations
   <1> Conflict between the write access to the A/D conversion result register (ADCRn or ADCRnH) at the end of conversion and the read access to the ADCRn or ADCRnH register by instruction
   Reading from the ADCRn or ADCRnH register has the higher priority. After the read operation, the new conversion result is written to the ADCRn or ADCRnH registers.
   <2> Conflict between the conversion result being stored in the A/D conversion result register (ADCRn or ADCRnH) at the end of conversion and the write access to A/D converter mode register 0 (ADM0) or the analog input channel specification register (ADS) by instruction
   Writing to the ADM0 and ADS registers has the higher priority. Writing to the ADCRn and ADCRnH registers is not performed, nor is the conversion end interrupt signal (INTAD) generated.
4. Noise countermeasures
   To maintain the 12-bit/10-bit resolution, attention must be paid to noise input to the AV_{REFP}, V_{DD}, ANI0 to ANI14, and ANI16 to ANI26 pins.
   <1> Connect a capacitor with a low equivalent resistance and a good frequency response (capacitance of about 0.1 μF) via the shortest possible run of relatively thick wiring to the V_{DD} and AV_{REFP} pins.
   <2> The higher the output impedance of the analog input source, the greater the influence. To reduce the noise, connecting an external capacitor as shown in **Figure 12 - 52** is recommended.
   <3> Do not switch these pins with other pins during conversion.
   <4> The accuracy is improved if the HALT mode is set immediately after the start of conversion.

Figure 12 - 52 Connections of V_{DD}, AV_{REFP}, and Analog Input Pins

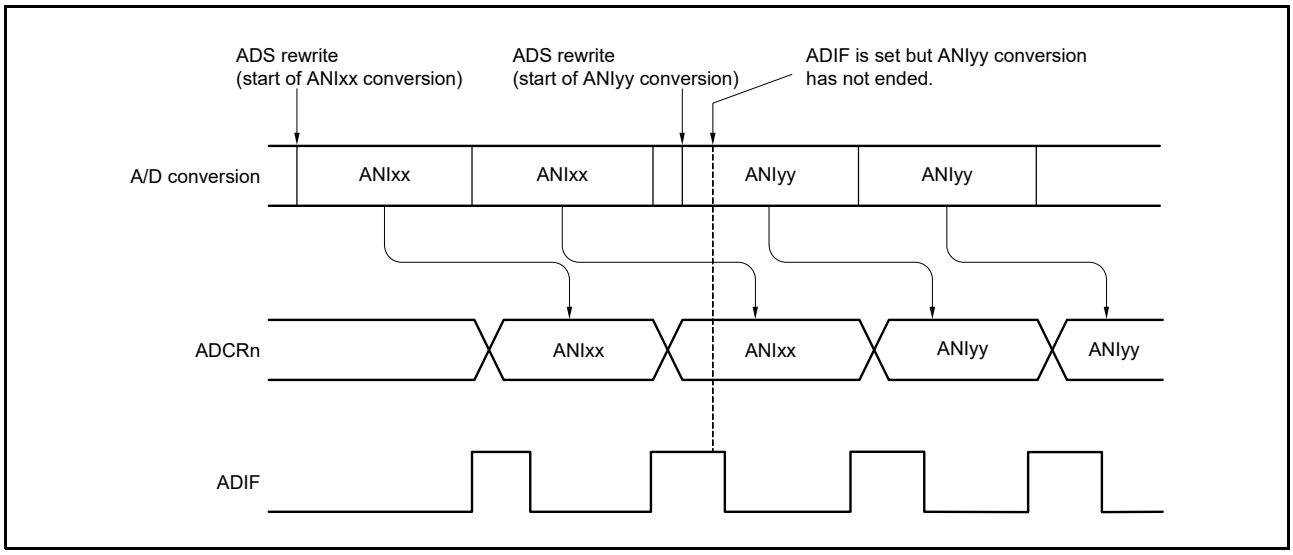


5. Analog input (ANIxx) pins
   - <1> The analog input pins (ANI0 to ANI14 and ANI16 to ANI26) are also used as input port pins (P02, P03, P20 to P27, P35 to P37, P100, P115 to P117, P120, P147, and P150 to P156). Do not change the output values for the port-pin functions P02, P03, P20 to P27, P35 to P37, P100, P115 to P117, P120, P147, and P150 to P156 while A/D conversion of the signals on the ANI0 to ANI14 or ANI16 to ANI26 pins is selected and conversion is in progress, since doing so may lower the precision of the results of conversion.
   - <2> If a pin adjacent to a pin that is being A/D converted is used as a digital I/O port pin, the A/D conversion result might differ from the expected value due to a coupling noise. Be sure to avoid the input or output of digital signals and signals with similarly sharp transitions during conversion.
6. Input impedance of analog input (ANIxx) pins

   This A/D converter charges a sampling capacitor for sampling during sampling time. Therefore, only a leakage current flows when sampling is not in progress, and a current that charges the capacitor flows during sampling. Consequently, the input impedance fluctuates depending on whether sampling is in progress. To make sure that sampling is effective, however, we recommend using the converter with analog input sources that have output impedances no greater than 1 kΩ. If a source has a higher output impedance, lengthen the sampling time or connect a larger capacitor (with a value of about 0.1 µF) to the pin from among ANI0 to ANI14 and ANI16 to ANI26 to which the source is connected (see **Figure 12 - 52**). The sampling capacitor may be being charged while the setting of the ADCS bit is 0 and immediately after sampling is restarted and so is not defined at these times. Accordingly, the state of conversion is undefined after charging starts in the next round of conversion after the value <R> of the ADCS bit has been 0 or when conversion is repeated. Thus, to secure full charging regardless of the size of fluctuations in the analog signal, ensure that the output impedances of the sources of analog inputs are low or secure sufficient time for the completion of sampling.
7. Interrupt request flag (ADIF)

   The interrupt request flag (ADIF) is not cleared to 0 even if the analog input channel specification register (ADS) is changed. Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and ADIF flag for the pre-change analog input may have been set before the ADS register is rewritten. When reading the ADIF flag immediately after rewriting to the ADS register, note that the ADIF flag is set although A/D conversion for the post-change analog input has not ended. When A/D conversion is stopped and then resumed, clear ADIF flag to 0 before the A/D conversion operation is resumed.

Figure 12 - 53 Timing of A/D Conversion End Interrupt Request Generation



8. Conversion results just after A/D conversion start

<R> While in the software trigger no-wait mode or hardware trigger no-wait mode, the first A/D conversion value immediately after A/D conversion starts may not fall within the rating range if the ADCS bit is set to 1 within 1 μs + 2 cycles of the conversion clock (f_{AD}) after the ADCE bit was set to 1. Take measures such as polling the A/D conversion end interrupt request signal (INTAD) and removing the first conversion result.

9. A/D conversion result register (ADCRn, ADCRnH) read operation

The contents of the ADCRn and ADCRnH registers may become undefined when writing to any of the following registers.

- A/D converter mode register 0 (ADM0)
- Analog input channel specification register (ADS)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)

Read the conversion result following conversion completion before writing to any of these registers. Otherwise, the correct conversion result may not be obtained.

10. Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 12 - 54 Internal Equivalent Circuit of ANIxx Pin

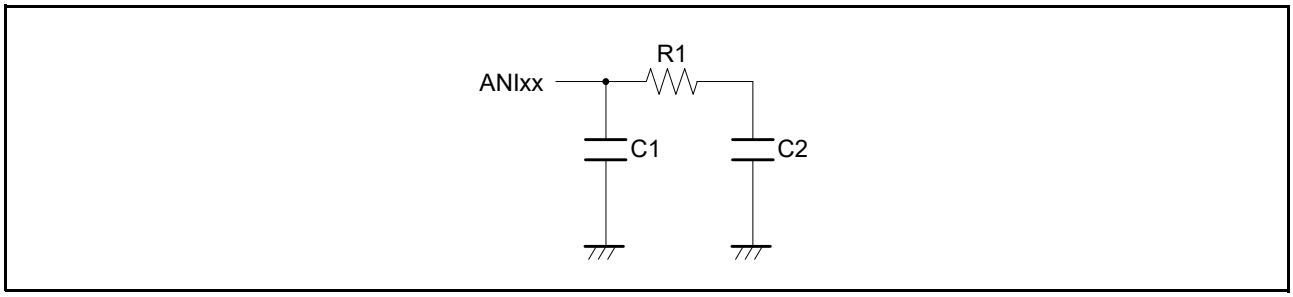


Table 12 - 4 Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AV_{REFP}, V_{DD}	ANIxx pin	R1 [kΩ]	C1 [pF]	C2 [pF]
2.4 V ≤ V_{DD} ≤ 5.5 V	ANI0 to ANI14	11	8	9
	ANI16 to ANI26	12	8	10
1.8 V ≤ V_{DD} < 2.4 V	ANI0 to ANI14	55	8	9
	ANI16 to ANI26	60	8	10
1.6 V ≤ V_{DD} < 1.8 V	ANI0 to ANI14	110	8	9
	ANI16 to ANI26	120	8	10

Remark The resistance and capacitance values are not guaranteed values.

11. Starting the A/D converter

Start the A/D converter after the AV_{REFP} and V_{DD} voltages stabilize.

Section 13 D/A Converter (DAC)

13.1 Functions of D/A Converter

The D/A converter is an 8-bit resolution converter that converts digital inputs into analog signals. It is used to control analog outputs for two independent channels (ANOi). The D/A converter has the following features.

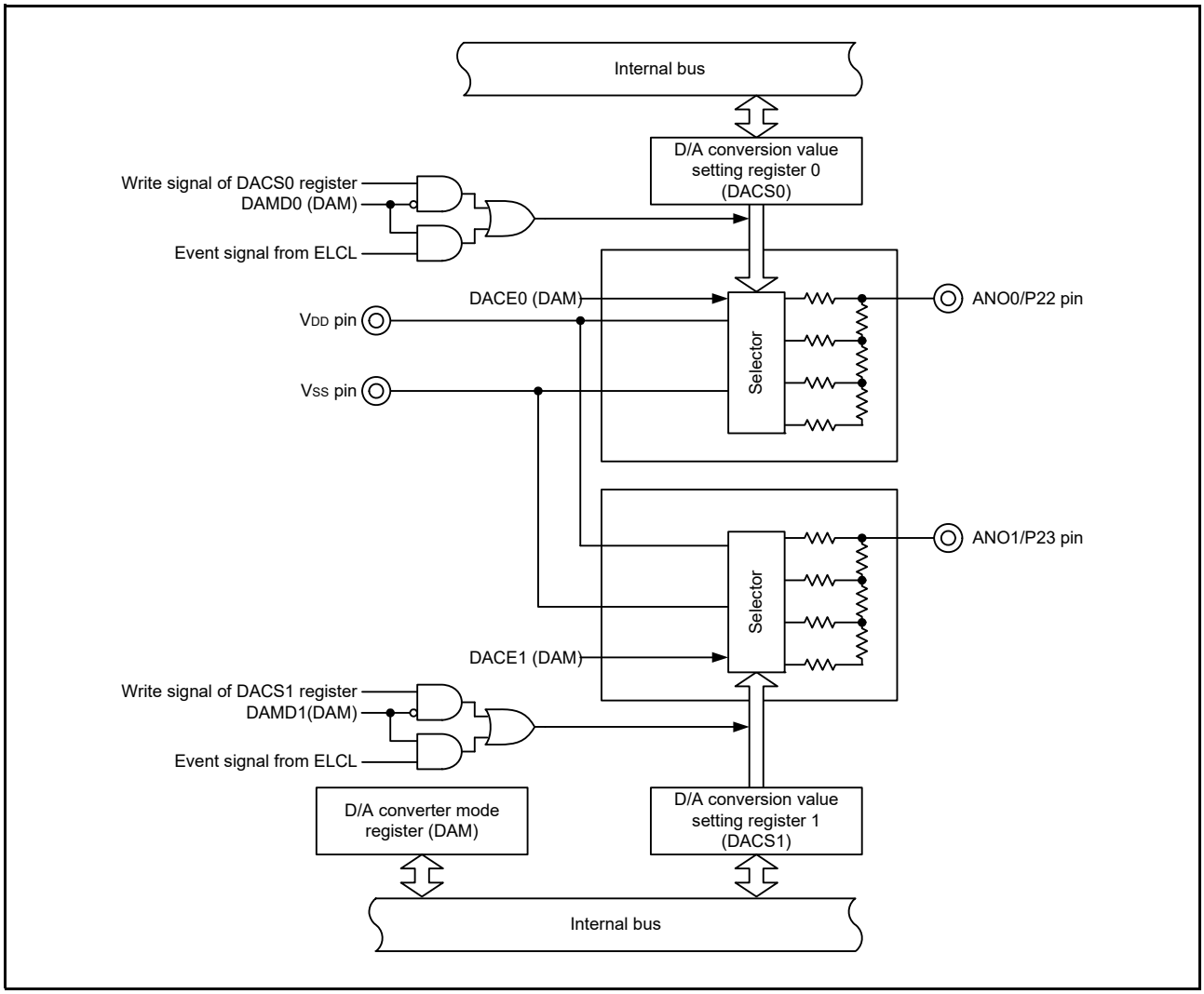
- 8-bit resolution × 2 channels
- R-2R ladder method
- Output analog voltage
  - 8-bit resolution: V_{DD} × m8/256, where m8 is the value set in the DACSi register.
- Operation mode
  - Normal operation mode
  - Realtime output mode

Remark i = 0, 1

13.2 Configuration of D/A Converter

Figure 13 - 1 shows the block diagram of D/A converter.

Figure 13 - 1 Block Diagram of D/A Converter



13.3 Registers for Controlling the D/A Converter

The following registers are used to control the D/A converter.

- Peripheral enable register 1 (PER1)
- Peripheral reset control register 1 (PRR1)
- D/A converter mode register (DAM)
- D/A conversion value setting register i (DACSi)
- Output signal select registers n (ELOSELn)
- Output signal enable register (ELOENCTL)
- Port mode registers (PMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)

13.3.1 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. If the D/A converter is to be used, be sure to set the DACEN bit of this register to 1. The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 13 - 2 Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH
After reset: 00H
R/W: R/W

<R>

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER1	DACEN	SMSEN	CMPEN	TML32EN	DTCEN	UTAEN **Note 1**	REMCEN **Note 2**	CTSUEN

DACEN	Control of supply of an input clock to the D/A converter
0	Stops supply of an input clock. • The SFRs used by the D/A converter cannot be written.**Note 3**
1	Enables supply of an input clock. • The SFRs used by the D/A converter can be read and written.

Note 1. This bit is only present in the 36- to 128-pin R7F100G and 32- to 48-pin R7F104G products.
Note 2. This bit is only present in the 32- to 128-pin R7F100G and 32- to 48-pin R7F104G products.
Note 3. When an SFR used by the D/A converter is read, the value returned is 00H regardless of the setting value.

Caution 1. Be sure to leave the following bits set to 0.
Bits 2 and 1 in the 30-pin R7F100G products
Bit 2 in the 32-pin R7F100G products
Caution 2. When setting the D/A converter, start by setting the DACEN bit to 1. If the setting of the DACEN bit is 0, attempted writing to the control registers of the D/A converter is ignored, and the value read is 00H. Note, however, writing to the port mode register (PM2) and port register (P2) is valid.

13.3.2 Peripheral reset control register 1 (PRR1)

The PRR1 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. Set the DACRES bit of this register to 1 to place the D/A converter in the reset state. The PRR1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 13 - 3 Format of Peripheral Reset Control Register 1 (PRR1)

Address: F00FBH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	3	2	<1>	<0>
PRR1	DACRES	SMSRES	CMPRES	TML32RES	0	0	REMCRES **Note**	CTSURES

DACRES	Control resetting of the D/A converter
0	The D/A converter is released from the reset state.
1	The D/A converter is in the reset state. • The SFRs for use with the D/A converter are initialized.

Note This bit is only present in the 32- to 128-pin R7F100G and 32- to 48-pin R7F104G products.

Caution 1. Be sure to leave bits 3 and 2 set to 0.
Caution 2. Be sure to leave bit 1 in the 30-pin R7F100G products set to 0.

13.3.3 D/A converter mode register (DAM)

This register controls the operation of the D/A converter. The DAM register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 13 - 4 Format of D/A Converter Mode Register (DAM)

Address: F0332H
After reset: 00H
R/W: R/W

Symbol	7	6	<5>	<4>	3	2	1	0
DAM	0	0	DACE1	DACE0	0	0	DAMD1	DAMD0

DACEi	D/A conversion operation control
0	Stops D/A conversion operation
1	Enables D/A conversion operation

DAMDi	D/A converter operation mode selection
0	Normal operation mode
1	Realtime output mode

Remark i = 0, 1

13.3.4 D/A conversion value setting register i (DACSi)

This register is used to set the analog voltage value to be output to the ANOi pins when the D/A converter is used.The DACSi register can be read by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 13 - 5 Format of D/A Conversion Value Setting Register i (DACSi)

Address: F0330H (DACS0), F0331H (DACS1)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
DACSi	DACSi7	DACSi6	DACSi5	DACSi4	DACSi3	DACSi2	DACSi1	DACSi0

Remark The relation between the resolution and analog output voltage (VANOi) of the D/A converter channel i is as follows.
$VANOi = V_{DD} \times (DACSi)/256$

When the D/A converter is not used, set the DACEi bit to 0 (stopping the D/A conversion) and set the DACSi register to 00H to prevent current from flowing into the R-2R resistor ladder to reduce unnecessary current consumption.

Remark i = 0, 1

13.3.5 Registers for controlling the event output from the logic and event link controller

D/A conversion in the realtime output mode is triggered by an event signal from the logic and event link controller. For details, see **20.3.23 Output signal select registers n (ELOSELn)** and **20.3.24 Output signal enable register (ELOENCTL).**

13.3.6 Registers for controlling the port functions multiplexed with the analog outputs of the D/A converter

Set the following registers to control the port functions multiplexed with the analog outputs of the D/A converter.

- Port mode registers (PMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)

For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.7 Port mode control A registers (PMCAxx)**, and **4.3.8 Port mode control T registers (PMCTxx)**.

For examples of setting the registers for controlling the port function, see **4.5.4 Examples of register settings for port and multiplexed functions**.

13.4 Operations of D/A Converter

13.4.1 Operation in normal operation mode

D/A conversion is triggered by writing to the DACSi register. An example of the settings is given 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> Set the corresponding port bits of the port mode control A register (PMCAxx) to 1 for operation as analog pins.
<3> Set the DAMDi bit of the D/A converter mode register (DAM) to 0 (normal operation mode).
<4> Set the analog voltage value to be output to the ANOi pin to the D/A conversion value setting register i (DACSi).

Steps <1> and <4> complete the initial settings.

<5> Set the DACEj bit of the DAM0 register to 1 to start D/A conversion. Once the settling time has elapsed, the analog voltage set in step <4> is output to the ANOi pin.
<6> To proceed with subsequent D/A conversions, write to the DACSi register.

The previous D/A conversion result is retained until the next D/A conversion proceeds. Setting of the DACEj bit of the DAM register to 0 stops D/A conversion.

Caution 1. Even if 1, 0, and then 1 is set to the DACEi bit, the analog voltage set by the DACSi register is output to the ANOi pin when a settling time has elapsed after 1 is set for the last time.
Caution 2. If the DACSi register is rewritten during the settling time, D/A conversion is aborted and reconversion by using the rewritten values starts.
Caution 3. Set the DACRES bit of the PRR1 register to 1 to initialize all circuits of the D/A converter.

Remark i = 0, 1

13.4.2 Operation in realtime output mode

D/A conversion on each channel is triggered by the event signals from the ELCL. An example of the settings is given below.

<1> Set the DACRES bit of the peripheral reset control register 1 (PRR1) to 0 to release the D/A converter from the reset state.
<2> 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.
<3> Set the corresponding port bits of the port mode control A register (PMCA) to analog pins.
<4> Set the DAMDi bit of the D/A converter mode register (DAM) to 0 (normal operation mode).
<5> Set the analog voltage value to be output to the ANOi pin to the D/A conversion value setting register i (DACSi).
<6> Set the DACEi bit of the DAM0 register to 1 to start D/A conversion. Once the settling time has elapsed, the analog voltage set in step <3> is output to the ANOi pin.
<7> Use the registers (ELOSELn and ELOENCTL) that control the event signal output from the logic and event link controller to set the trigger signal for use in realtime output mode.
<8> Set the DAMDi bit of the DAM register to 1 (realtime output mode).
<9> Start the operation of the event source.

Steps <1> to <9> completes initial settings.

<10> Upon generation of the trigger signals used for realtime output mode, D/A conversion starts and the analog voltage set in step <5> will be output to the ANOi pin once the settling time has elapsed.

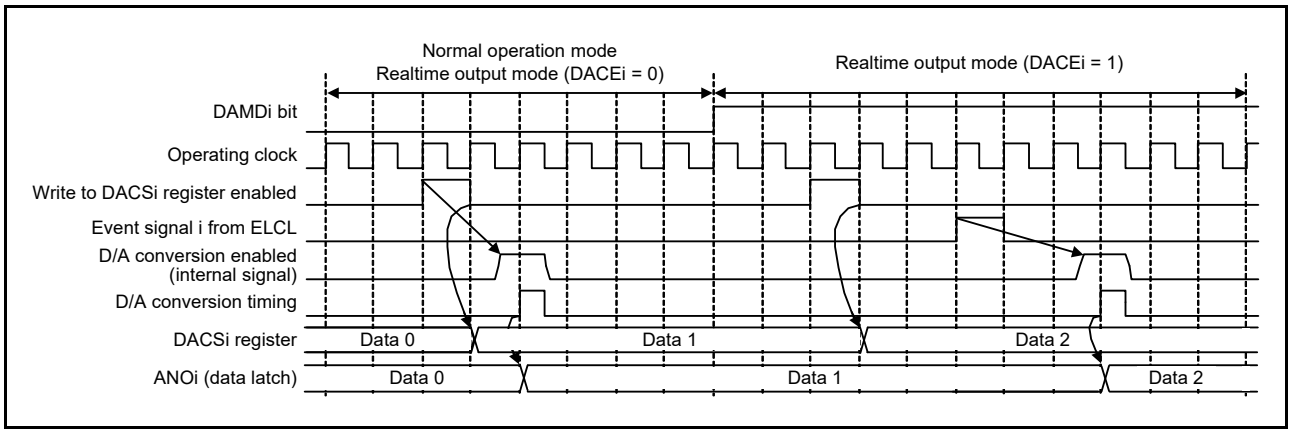
Set the analog voltage value to be output to the ANOi pin, to the DACSi register before starting the next D/A conversion (trigger signal used for realtime output mode is generated). Setting of the DACEi bit of the DAM register to 0 stops D/A conversion.

Caution 1. Even if 1, 0, and then 1 is set to the DACEi bit, the analog voltage set by the DACSi register is output to the ANOi pin when a settling time has elapsed after 1 is set for the last time.
Caution 2. Set the interval between each generation of the trigger signal used for realtime output mode of the same channel to longer than the settling time. If a trigger signal used for realtime output mode is generated during the settling time, D/A conversion is aborted and reconversion starts.
Caution 3. Set the interval between each generation of the trigger signal used for realtime output mode of the same channel to longer than the three clocks of f_{CLK}. When a trigger is generated consecutively at intervals of three or fewer f_{CLK} clock cycles, only the first signal triggers D/A conversion.
Caution 4. Set the DACRES bit of the PRR1 register to 1 to initialize all circuits of the D/A converter.

13.4.3 Timing for outputting D/A conversion value

Figure 13 - 6 shows the timing for outputting D/A conversion value.

Figure 13 - 6 Timing for Outputting D/A Conversion Value



Remark i = 0, 1

- Normal operation mode and realtime output mode when conversion is disabled
  The value is written to the data latch after one cycle of the operating clock when the DACSi register is written.
- Realtime output mode when conversion operation is enabled
  The value is written to the data latch (output from the ANOi pin) after three cycles of the operating clock when the event signal from the ELCL is accepted.

13.5 Points for Caution when the D/A Converter is to be Used

Observe the following cautions when using the D/A converter.

1. The digital port I/O functions which are multiplexed with the functions of the ANOi pins do not operate if the port settings in the port mode control A register (PMCAxx) are for analog pins. When the port register (P2) is read while the ports are set to analog pins by using the PMCAxx register, 0 is read in the input mode and the set value of the P2 register is read in the output mode. If the digital output mode is set, no output data is output to pins.
2. The operation of the D/A converter continues in the HALT and STOP modes. To lower the power consumption, therefore, clear the DACEi bit to 0, and execute the HALT or STOP instruction after stopping the operation of the D/A converter.

   Remark i = 0, 1

3. To stop the realtime output mode (including when changing to normal operation 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 DACEi and DAMDi to 0.
   - After setting bits DACEi and DAMDi, set the DACEN bit of the PER1 register to 0 (stopping supply of the input clock to the DAC). Note that setting the DACEN bit to 0 does not initialize the D/A converter. Use the DACRES bit of the PRR1 register to initialize the D/A converter and its SFRs.
4. Do not start A/D conversion by using analog input pin functions that are multiplexed with the ANOi pins while D/A conversion is enabled.
5. In realtime output mode, set the value of the DACSi register before a trigger signal used for realtime output mode is generated. Do not change the set value of the DACSi 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 ANOi pin. If the input impedance of the load is low, insert a follower amplifier between the load and the ANOi 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 STOP mode while realtime output mode is enabled, disable linking of ELC events before entering STOP mode.

Section 14 Comparator (CMP)

14.1 Functions of Comparator

The comparator has the following functions.

- Comparator high-speed mode, or comparator low-speed mode can be selected.
- The external reference voltage input and the internal reference voltage[Note 1] or D/A converter output[Note 2] are selectable as the reference voltage.
- The canceling width of the noise canceling digital filter can be selected.
- An interrupt signal can be generated by detecting an active edge of the comparator output.
- An event signal can be output to the logic and event link controller (ELCL) by detecting an active edge of the comparator output.

Note 1. For the internal reference voltage, see **Section 37 Electrical Characteristics**.

Note 2. The internal reference voltage and D/A converter output 0 are selectable for comparators 0 and 1, respectively.

Caution Comparator 1 is not present in the R7F104G products.

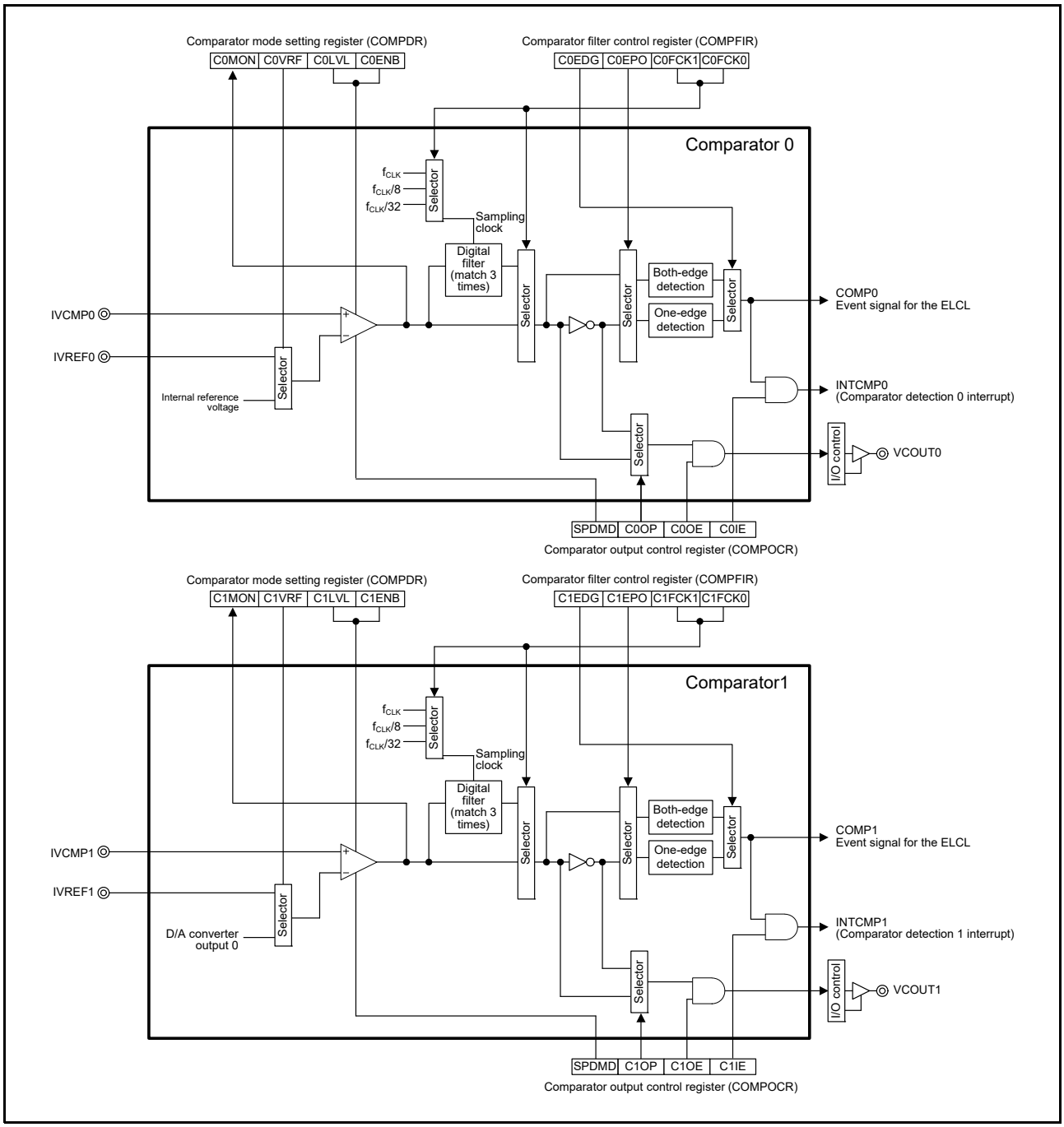
The number of the comparator channels depends on the product.

	30- to 128-pin R7F100G products	32- to 48-pin R7F104G products
Number of Channels	2	1

14.2 Configuration of Comparator

Figure 14 - 1 shows the comparator block diagram.

Figure 14 - 1 Comparator Block Diagram



14.3 Registers for Controlling the Comparator

The following registers are used to control the comparator.

- Peripheral enable register 1 (PER1)
- Peripheral reset control register 1 (PRR1)
- Comparator mode setting register (COMPMDR)
- Comparator filter control register (COMPFIR)
- Comparator output control register (COMPOCR)
- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)

14.3.1 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. If the comparator is to be used, be sure to set the CMPEN bit of this register to 1. The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 14 - 2 Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER1	DACEN	SMSEN	CMPEN	TML32EN	DTCEN	UTAEN Note 1	REMCEN Note 2	CTSUEN

CMPEN	Control of an input clock to the comparator
0	Stops supply of an input clock. • The SFRs used by the comparator cannot be written.Note 3
1	Enables supply of an input clock. • The SFRs used by the comparator can be read and written.

Note 1. This bit is only present in the 36- to 128-pin R7F100G and 32- to 48-pin R7F104G products.
Note 2. This bit is only present in the 32- to 128-pin R7F100G and 32- to 48-pin R7F104G products.
Note 3. When an SFR used by the comparator is read, the value returned is 00H regardless of the setting value.

Caution 1. Be sure to leave the following bits set to 0.
Bits 2 and 1 in the 30-pin R7F100G products
Bit 2 in the 32-pin R7F100G product

Caution 2. When setting the comparator, start by setting the CMPEN bit to 1. If the setting of the CMPEN bit is 0, attempted writing to the control registers of the comparator is ignored. Note, however, writing to the following registers is valid.
- **Port mode control A registers (PMCA1, PMCA2, PMCA12, PMCA14)**
- **Port mode registers (PM1, PM2, PM3, PM12, PM14)**
- **Port registers (P1, P2, P3, P12, P14)**

14.3.2 Peripheral reset control register 1 (PRR1)

The PRR1 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. Set the CMPRES bit of this register to 1 to place the comparator in the reset state. The PRR1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 14 - 3 Format of Peripheral Reset Control Register 1 (PRR1)

Address: F00FBH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	3	2	<1>	<0>
PRR1	DACRES	SMSRES	CMPRES	TML32RES	0	0	REMCRES **Note**	CTSURES

CMPRES	Control resetting of the comparator
0	The comparator is released from the reset state.
1	The comparator is in the reset state. • The SFRs for use with the comparator are initialized.

Note This bit is only present in the 32- to 128-pin R7F100G and 32- to 48-pin R7F104G products.

Caution 1. Be sure to leave bits 2 and 3 set to 0.
Caution 2. Be sure to leave bit 1 in the 30-pin R7F100G products set to 0.

14.3.3 Comparator mode setting register (COMPMDR)

The COMPMDR register is used to make the various settings of comparators 0 and 1, such as selection of the reference voltage, starting or stopping comparing operation, and also indicates the results of comparison. The COMPMDR register can be set by a 1-bit or 8-bit memory manipulation instruction. Note that the CnMON bit is read-only. The value of this register following a reset is 00H.

Figure 14 - 4 Format of Comparator Mode Setting Register (COMPMDR) (1/2)

Address: F0340H
After reset: 00H
R/W: R/W

Symbol	7	6	5	<4>	3	2	1	<0>
COMPMDR	C1MON **Note 1**	C1VRF **Note 1**	C1LVL **Note 1**	C1ENB **Note 1**	C0MON	C0VRF	C0LVL	C0ENB

C1MON	Comparator 1 monitor flag**Notes 2, 3**
0	IVCMP1 < comparator 1 reference voltage (IVREF1 or D/A converter output 0)
1	IVCMP1 > comparator 1 reference voltage (IVREF1 or D/A converter output 0)

C1VRF	Selection of comparator 1 reference voltage
0	Supply through the IVREF1 pin
1	Supply through D/A converter output 0

C1LVL	Selection of comparator 1 reference voltage range
0	0 to V_{DD} - 1.4 V
1	1.4 V to V_{DD}

C1ENB	Comparator 1 operation enable
0	Comparator 1 operation disabled
1	Comparator 1 operation enabled

C0MON	Comparator 0 monitor flag**Notes 2, 3**
0	IVCMP0 < comparator 0 reference voltage (IVREF0 or internal reference voltage)
1	IVCMP0 > comparator 0 reference voltage (IVREF0 or internal reference voltage)

C0VRF	Selection of comparator 0 reference voltage**Note 4**
0	Supply through the IVREF0 pin
1	Supply through the internal reference voltage

C0LVL	Selection of comparator 0 reference voltage range**Note 4**
0	0 to V_{DD} - 1.4 V
1	1.4 V to V_{DD}

Figure 14 - 4 Format of Comparator Mode Setting Register (COMPMDR) (2/2)

C0ENB	Comparator 0 operation enable
0	Comparator 0 operation disabled
1	Comparator 0 operation enabled

Note 1. This bit is only present in the 30- to 128-pin R7F100G products.

Note 2. The value is 0 immediately after a reset is released. However, the value is undefined when C0ENB is set to 0 and C1ENB is set to 0 after operation of the comparator is enabled once.

Note 3. Bits 7 and 3 are read-only. The value written to these bits is ignored.

Note 4. When the internal reference voltage (C0VRF = 1) is to be used as the comparator 0 reference voltage, C0LVL = 0 and V_{DD} ≥ (internal reference voltage (max.) + 1.4 V) must be satisfied.

14.3.4 Comparator filter control register (COMPFIR)

The COMPFIR register is used to select the valid edge for use with the comparator interrupt signals and whether or not to use the digital filter. If rejecting noise is required, set the CnFCK1 and CnFCK0 bits to enable the digital filters. While a digital filter is enabled, the filter conveys the input level after detecting matches for three cycles of the sampling clock for use with the filter. The COMPFIR register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 14 - 5 Format of Comparator Filter Control Register (COMPFIR)

Address: F0341H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
COMPFIR	C1EDG **Note 1**	C1EPO **Note 1**	C1FCK1 **Note 1**	C1FCK0 **Note 1**	C0EDG	C0EPO	C0FCK1	C0FCK0

C1EDG	Comparator 1 edge detection selection **Note 2**
0	Interrupt request by comparator 1 one-edge detection
1	Interrupt request by comparator 1 both-edge detection

C1EPO	Comparator 1 active edge switching **Note 2**
0	Interrupt request at the rising edge of comparator 1
1	Interrupt request at the falling edge of comparator 1

C1FCK1	C1FCK0	Comparator 1 filter selection **Notes 2, 3**
0	0	No comparator 1 filter
0	1	Comparator 1 filter enabled, sampling at f_{CLK}
1	0	Comparator 1 filter enabled, sampling at $f_{CLK}/8$
1	1	Comparator 1 filter enabled, sampling at $f_{CLK}/32$

C0EDG	Comparator 0 edge detection selection **Note 4**
0	Interrupt request by comparator 0 one-edge detection
1	Interrupt request by comparator 0 both-edge detection

C0EPO	Comparator 0 active edge switching **Note 4**
0	Interrupt request at the rising edge of comparator 0
1	Interrupt request at the falling edge of comparator 0

C0FCK1	C0FCK0	Comparator 0 filter selection **Notes 3, 4**
0	0	No comparator 0 filter
0	1	Comparator 0 filter enabled, sampling at f_{CLK}
1	0	Comparator 0 filter enabled, sampling at $f_{CLK}/8$
1	1	Comparator 0 filter enabled, sampling at $f_{CLK}/32$

(Notes are listed on the next page.)

Note 1. This bit is only present in the 30- to 128-pin R7F100G products.

Note 2. Changing the C1FCK0 and C1FCK1 bit field or the C1EPO or C1EDG bit may lead to the generation of an event signal for the ELCL and a comparator 1 interrupt request. Only change these bits while the output from comparator 1 is not selected as an input signal to the ELCL and the comparator 1 interrupt is masked. After that, to use the comparator 1 interrupt, clear the CMPIF1 bit to 0 and then clear the CMPMK1 bit to 0 to enable the interrupt. If the C1FCK1 and C1FCK0 bits are changed from 00B (no comparator 1 filter) to a value other than 00B (comparator 1 filter enabled), allow the time for sampling four times to elapse until the filter output is updated, and then use the comparator 1 interrupt request or the event signal for the ELCL.

Note 3. When using the comparators in STOP mode, make the settings for non-use of the filters (C0FCK1 and C0FCK0 = 00B, C1FCK1 and C1FCK0 = 00B).

Note 4. Changing the C0FCK1 and C0FCK0 bit field or the C0EPO or C0EDG bit may lead to the generation of an event signal for the ELCL and a comparator 0 interrupt request. Only change these bits while the output from comparator 0 is not selected as an input signal to the ELCL and the comparator 0 interrupt is masked. After that, to use the comparator 0 interrupt, clear the CMPIF0 bit to 0 and then clear the CMPMK0 bit to 0 to enable the interrupt. If the C0FCK1 and C0FCK0 bits are changed from 00B (no comparator 0 filter) to a value other than 00B (comparator 0 filter enabled), allow the time for sampling four times to elapse until the filter output is updated, and then use the comparator 0 interrupt request or the event signal for the ELCL.

14.3.5 Comparator output control register (COMPOCR)

The COMPOCR register is used to select items including the response speed of the comparators, control of the VCOUTn outputs, and enabling or disabling the interrupt request signals. The COMPOCR register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Remark n = 0, 1

Figure 14 - 6 Format of Comparator Output Control Register (COMPOCR)

Address: F0342H
After reset: 00H
R/W: R/W

Symbol	7	6	<5>	<4>	3	2	<1>	<0>
COMPOCR	SPDMD	C1OP **Note 1**	C1OE **Note 1**	C1IE **Note 1**	0	C0OP	C0OE	C0IE

SPDMD	Comparator speed selection **Note 2**
0	Comparator low-speed mode
1	Comparator high-speed mode

C1OP	VCOUT1 output active level selection
0	Comparator 1 output is output as VCOUT1.
1	Comparator 1 output is inverted for output as VCOUT1.

C1OE	VCOUT1 pin output enable
0	Comparator 1 VCOUT1 pin output disabled
1	Comparator 1 VCOUT1 pin output enabled

C1IE	Comparator 1 interrupt request enable **Note 3**
0	Comparator 1 interrupt request disabled
1	Comparator 1 interrupt request enabled

C0OP	VCOUT0 output active level selection
0	Comparator 0 output is output as VCOUT0.
1	Comparator 0 output is inverted for output as VCOUT0.

C0OE	VCOUT0 pin output enable
0	Comparator 0 VCOUT0 pin output disabled
1	Comparator 0 VCOUT0 pin output enabled

C0IE	Comparator 0 interrupt request enable **Note 4**
0	Comparator 0 interrupt request disabled
1	Comparator 0 interrupt request enabled

Note 1. This bit is only present in the 30- to 128-pin R7F100G products.
Note 2. When rewriting the SPDMD bit, be sure to set the CiENB bit in the COMPMDR register to 0 in advance.

(Notes and Remark are listed on the next page.)

Note 3. If the C1IE bit is changed from 0 (interrupt request disabled) to 1 (interrupt request enabled), since the CMPIF1 bit in the interrupt request flag register 2H (IF2H) may be set to 1 (interrupt requested), clear the CMPIF1 bit in the interrupt request flag register 2H (IF2H) to 0 before using an interrupt.

Note 4. If the C0IE bit is changed from 0 (interrupt request disabled) to 1 (interrupt request enabled), since the CMPIF0 bit in the interrupt request flag register 2H (IF2H) may be set to 1 (interrupt requested), clear the CMPIF0 bit in the interrupt request flag register 2H (IF2H) to 0 before using an interrupt.

Remark n = 0, 1

14.3.6 Registers for controlling the port functions multiplexed with the analog inputs and outputs of the comparator

Set the following registers to control the port functions multiplexed with the analog inputs and outputs of the comparator.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)

For details, see the following sections.

- **4.3.1 Port mode registers (PMxx)**
- **4.3.2 Port registers (Pxx)**
- **4.3.5 Port output mode registers (POMxx)**
- **4.3.7 Port mode control A registers (PMCAxx)**
- **4.3.8 Port mode control T registers (PMCTxx)**
- **4.3.9 Port mode control E registers (PMCExx)**

For examples of setting the registers for controlling port functions, see **4.5.4 Examples of register settings for port and multiplexed functions**.

14.4 Operation

Comparators i (i = 0, 1) can run independently of each other. The procedures for settings and the type of operations the settings produce are the same. **Table 14 - 1** lists the steps in the procedure for setting the comparator-related registers.

Table 14 - 1 Procedure for Setting the Comparator Related Registers

Step	Register	Bit	Setting Value
1	PRR1	CMPRES	0 (releasing the comparators from the reset state)
2	PER1	CMPEN	1 (enabling input clock supply)
3	PMCAxx	PMCA13, PMCA23, PMCA120, PMCA147	Select the function of pins IVCMPi and IVREFi. • Set the PMCA13**Note 1**, PMCA23, PMCA120**Note 1**, and PMCA147 bits to 1 (analog functions). • Set the PM13**Note 1**, PM23, PM120**Note 1**, and PM147 bits to 1 (input mode).
	PMxx	PM13, PM23, PM120, PM147	
4	COMPOCR	SPDMD	Select the comparator response speed (0: Low-speed mode, 1: High-speed mode). **Note 2**
5	COMPMDR	CiVRF	Select the reference voltage. For i = 0, 0: Input to the IVREF0 pin, 1: Internal reference voltage For i = 1, 0: Input to the IVREF1 pin, 1: D/A converter output 0
		CiLVL	Select the reference input voltage range. (0: 0 to V_{DD} - 1.4 V, 1: 1.4 V to V_{DD})
6	COMPMDR	CiENB	1 (enabling operation)
7	Wait for comparator stabilization time (t_{CMP}).		
8 9	COMPFIR	CiFCK1, CiFCK0	Select whether the digital filter is used or not and the sampling clock.
		CiEPO, CiEDG	Select the condition for edge detection of interrupt requests (rising edge/falling edge/both edges).
10	COMPOCR	CiOP, CiOE	Set the VCOUTi output (select the active level and enable or disable output).
		CiIE	Set enabling or disabling of interrupt request output.
11	Set the active level and enabling or disabling of the VCOUTi output. See **14.4.4 Comparator i output**.		
12	PR02H	CMPPR0i, CMPPR1i	When using an interrupt: Select the interrupt priority level.
13	MK2H	CMPMKi	When using an interrupt: Select masking of the interrupt.
14	IF2H	CMPIFi	When using an interrupt: 0 (no interrupt requested: initialization) **Note 3**

Note 1. This bit is only present in the 30- to 128-pin R7F100G products.

Note 2. This setting cannot be made independently for comparator 0 and comparator 1.

Note 3. Initialize the interrupt flag after having set the comparator, as an unintended interrupt may otherwise occur in the interval until operation becomes stable.

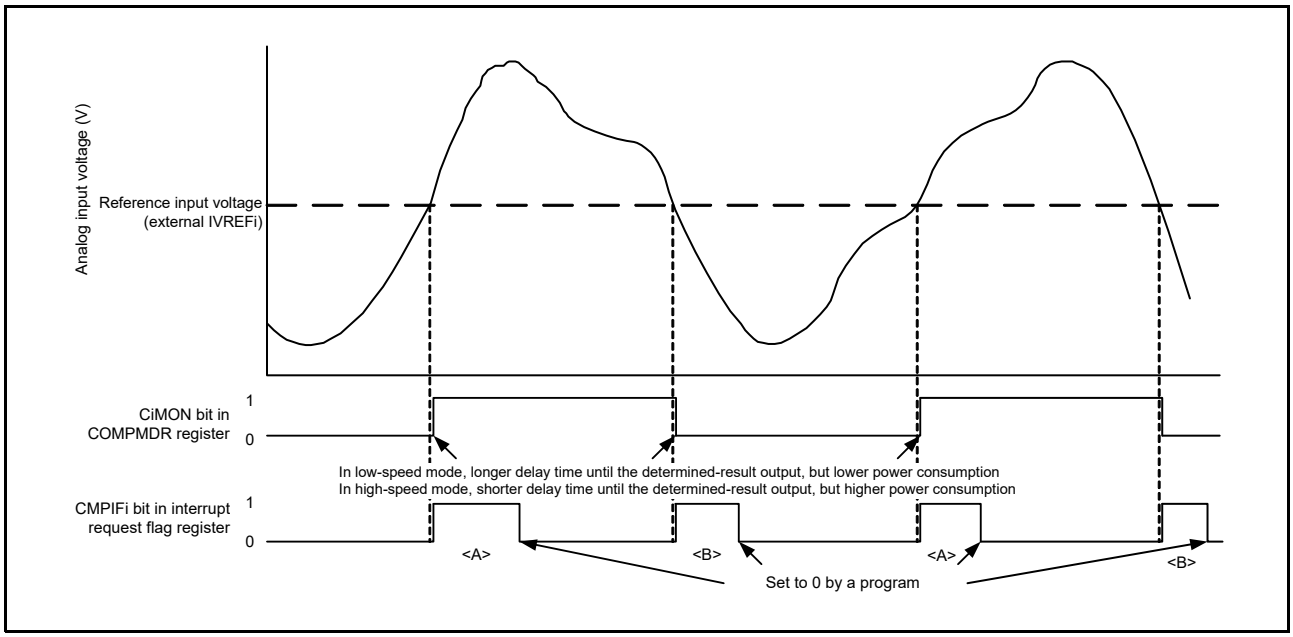
Caution 1. To initialize all circuits of comparators i, set the CMPRES bit in the PRR1 register to 1.

Caution 2. Comparator 1 is not present in the R7F104G products.

Remark i = 0, 1

Figure 14 - 7 shows an operation example of comparator i. In both low-speed mode and high-speed mode, the CiMON bit in the COMPMDR register is set to 1 when the analog input voltage is higher than the reference input voltage, and the CiMON bit is set to 0 when the analog input voltage is lower than the reference input voltage. The input voltage to the IVREFi pin is selected for use as the reference voltage.

Figure 14 - 7 Example of the Comparator i Operation



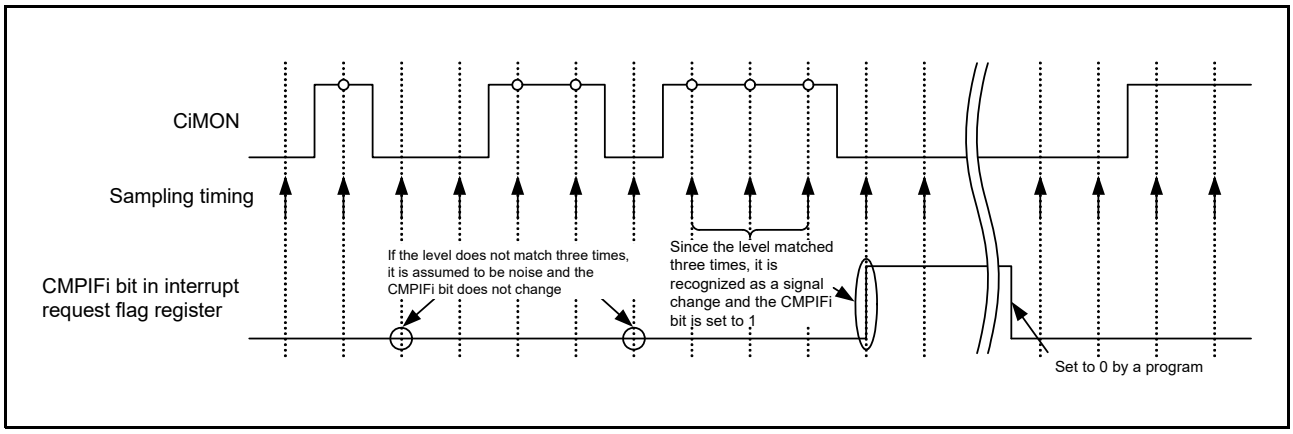
Caution **The above diagram applies when CiFCK1 and CiFCK0 in the COMPFIR register = 00B (no digital filter) and CiEDG = 1 (both-edge detection). When CiEDG = 0 and CiEPO = 0 (rising edge), CMPIFi changes as shown by <A> only. When CiEDG = 0 and CiEPO = 1 (detection of the falling edge), CMPIFi changes as shown by <B> only.**

Remark i = 0, 1

14.4.1 Comparator i digital filter

Comparator i contains a digital filter. The sampling clock can be selected by bits CiFCK1 and CiFCK0 in the COMPFIR register. The comparator i output signal is sampled every sampling clock, and when the level matches three times, that value is determined as the digital filter output at the next sampling clock. **Figure 14 - 8** shows the comparator i digital filter and interrupt operation example.

Figure 14 - 8 Comparator i Digital Filter and Interrupt Operation Example



Caution The above operation example applies when bits CiFCK1 and CiFCK0 in the COMPFIR register is 01B, 10B, or 11B (digital filter enabled).

14.4.2 Comparator i interrupts

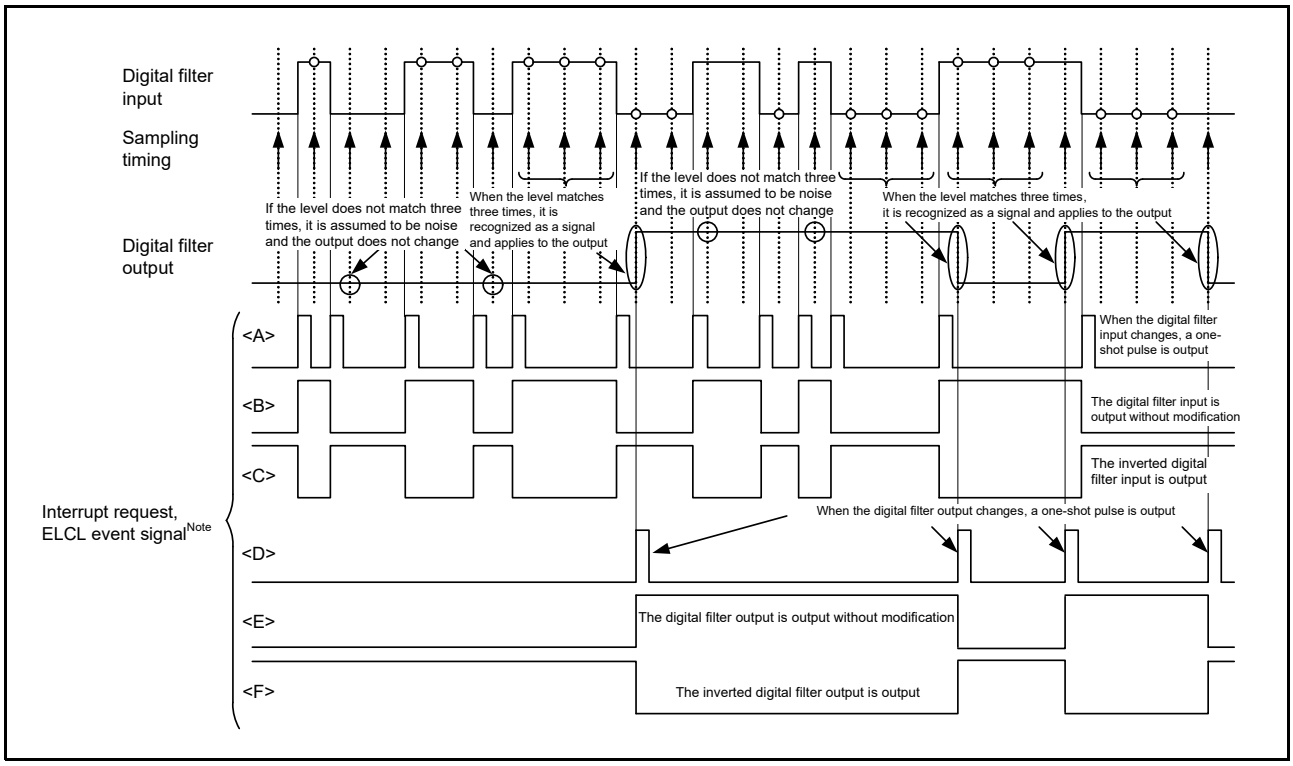
The comparator generates interrupt requests from two sources, comparator 0 and comparator 1. The comparator i interrupt each uses a priority level specification flag, an interrupt mask flag, an interrupt request flag, and a single vector. When using the comparator i interrupt, set the CiIE bit in the COMPOCR register to 1 (interrupt request enabled). The condition for interrupt request generation can be set by the COMPFIR register. The comparator outputs can also be passed through the digital filter. Three different sampling clocks can be selected for the digital filter. For details on the register setting and interrupt request generation, refer to **14.3.4 Comparator filter control register (COMPFIR)** and **14.3.5 Comparator output control register (COMPOCR)**.

Remark i = 0, 1

14.4.3 Event signal output for the logic and event link controller (ELCL)

An event signal for the ELCL is generated by detecting the edge for the digital filter output set by the COMPFIR register, which is the same as the condition for interrupt request generation. However, unlike interrupt requests, the event signal for the ELCL is always output regardless of the setting of the CiIE bit in the COMPOCR register. Set the output signal select registers n (ELOSELn) and output signal enable register (ELOENCTL) of the ELCL to select the event output destination and to stop linking events.

Figure 14 - 9 Digital Filter and Interrupt Request/Event Signal Output for the ELCL



Note When the CiIE bit is 1, the same waveform is generated for an interrupt request and an ELCL event signal.
When the CiIE bit is 0, the value is fixed at 0 for an interrupt request only.

The waveforms of <A>, <B>, and <C> are shown for an operation example when the CiFCK bits in the COMPFIR register are 00B (no digital filter). The waveforms <D>, <E>, and <F> are shown for an operation example when the CiFCK bits in the COMPFIR register are 01B, 10B, or 11B (digital filter enabled). <A> and <D> apply when the CiEDG bit is set to 1 (both-edge detection), <B> and <E> when the CiEDG bit is 0 and the CiEPO bit is 0 (rising edge), and <C> and <F> when the CiEDG bit is 0 and the CiEPO bit is 1 (detection of the falling edge).

Remark i = 0, 1

14.4.4 Comparator i output

The comparison result from the comparator can be output to external pins. Bits CiOP and CiOE in the COMPOCR register can be used to set the active level (non-inverted or inverted output) and enable or disable output. For the correspondence between the register setting and the comparator output, refer to **14.3.5 Comparator output control register (COMPOCR)**. To output the comparator comparison result to the VCOUTi output pin, use the following procedure to set the ports. Note that the ports are set to input after reset.

<1> Set the mode for the comparator (steps 1 to 9 as listed in **Table 14 - 1 Procedure for Setting the Comparator Related Registers**).
<2> Select the active level of the VCOUT0 or VCOUT1 output and enable the output (step 11 as listed in **Table 14 - 1 Procedure for Setting the Comparator Related Registers**).
<3> Set the bit in the port mode control T register (PMCTxx) or port mode control E register (PMCExx) corresponding to the VCOUTi output pin to 0.
<4> Set the corresponding port register bit for the VCOUTi output pin to 0.
<5> Set the corresponding port mode register for the VCOUTi output pin to output (start outputting from the pin).

Remark i = 0, 1

14.4.5 Stopping or supplying the clock to the comparator

To stop the comparator clock by setting peripheral enable register 1 (PER1), use the following procedure:

<1> Set the CiIE bit in the COMPOCR register to 0 (disable a comparator interrupt).
<2> Set the CiENB bit in the COMPMDR register to 0 (stop the comparator).
<3> Set the CMPIFi flag in the IF2H register to 0 (clear any unintended interrupt before stopping the comparator).
<4> Set the CMPEN bit in the PER1 register to 0.

When the clock is stopped by setting PER1, all the internal registers in the comparator are not initialized. The values of these registers are retained.

Caution **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, check the monitor flag (CiMON) of the comparator before enabling DTC activation as required.**

- **The comparator is set to an interrupt request on one-edge detection (CiEDG = 0), an interrupt request at the rising edge for the comparator (CiEPO = 0), and IVCMP > comparator n reference voltage**
- **The comparator is set to an interrupt request on one-edge detection (CiEDG = 0), an interrupt request at the falling edge for the comparator (CiEPO = 1), and IVCMP < comparator i reference voltage**

Remark i = 0, 1

Section 15 Serial Array Unit (SAU)

A single serial array unit has up to four serial channels. Each channel can achieve 3-wire serial (simplified SPI or CSI[Note]), UART, and simplified I2C communication. The following tables show the assignment of functions to each channel in this MCU.

Note Although the CSI function is generally called SPI, it is also called CSI in this product, so it is referred to as such in this manual.

30- and 32-pin R7F100G products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I2C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting the LIN bus)	IIC20
	1	—		—

36- to 44-pin R7F100G products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I2C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting the LIN bus)	IIC20
	1	CSI21		IIC21

48- and 52-pin R7F100G products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I2C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting the LIN bus)	IIC20
	1	CSI21		IIC21

64-pin R7F100G products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting the LIN bus)	IIC20
	1	CSI21		IIC21

80- to 128-pin R7F100G products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting the LIN bus)	IIC20
	1	CSI21		IIC21
	2	CSI30	UART3	IIC30
	3	CSI31		IIC31

32- to 48-pin R7F104G products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	—		IIC11
1	0	—	UART2 (supporting the LIN bus)	—
	1	CSI21		IIC21

When “UART0” is used for channels 0 and 1 of the unit 0, CSI00 and CSI01 cannot be used, but CSI10, UART1, or IIC10 in channels 1 and 3 can be used.

Caution Most of the following descriptions in this section use the units and channels of the 128-pin R7F100G products as an example.

15.1 Functions of Serial Array Unit

Each serial interface supported by this MCU has the following features.

15.1.1 Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31)

Data is transmitted or received in synchronization with the serial clock (SCK) output from the master. 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 **15.5 Operation of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31) Communication**.

[Data transmission/reception]

- Data length of 7 or 8 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable

[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**Note** During master communication: Max. f_{CLK}/2 (CSI00 only)
  Max. f_{CLK}/4
  During slave communication: Max. f_{MCK}/6

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

Furthermore, only the following simplified SPIs (CSIs) support the SNOOZE mode. In the SNOOZE mode, data can be received without CPU processing upon detecting SCK input in the STOP mode.

- 30- to 64-pin R7F100G and 32- to 48-pin R7F104G products: CSI00
- 80- to 128-pin R7F100G products: CSI00 and CSI20

Note Set up the transfer rate within a range satisfying the SCK cycle time (t_{KCY}). For details, see **Section 37 Electrical Characteristics**.

Caution Use a general-purpose port pin to send a chip select signal when required.

15.1.2 UART (UART0 to UART3)

This is a start-stop synchronization communication function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel). The LIN bus can be implemented by using timer array unit with an external interrupt (INTP0). For details about the settings, see **15.6 Operation of UART (UART0 to UART3) Communication**.

[Data transmission/reception]

- Data length of 7, 8, or 9 bits**Note**
- MSB/LSB first selectable
- Level setting of transmit/receive data and select of reverse
- Parity bit appending and parity check functions
- Stop bit appending

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

Furthermore, only the following UARTs support reception in the SNOOZE mode. In the SNOOZE mode, data can be received without CPU processing upon detecting RxD input in the STOP mode.

- UART0 in the 30- to 64-pin R7F100G and 32-pin R7F104G products
- UART0 and UART2 in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products

UART2 (channels 0 and 1 of unit 1) only supports the LIN bus in the 30- to 128-pin R7F100G and 32- to 48-pin R7F104G products.

[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

Note Only the following UARTs support the 9-bit data length.

- UART0 in the 30- to 64-pin R7F100G and 32-pin R7F104G products
- UART0 and UART2 in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products

15.1.3 Simplified I2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31)

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This simplified I2C is designed for single communication with a device such as EEPROM, flash memory, or A/D converter, and therefore, it functions only as a master. Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed. For details about the settings, see **15.8 Operation of Simplified I2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31) 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
- Clock stretch detection

Note When receiving the last data, ACK will not be output if 0 is written to the SOEmn bit (serial output enable register m (SOEm)) and serial communication data output is stopped. See **15.8.3 Data reception**, **2. Processing flow** for details.

Remark 1. To use an I2C bus of full function, see **Section 16 Serial Interface IICA (IICA)**.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

15.2 Configuration of Serial Array Unit

The serial array unit includes the following registers, and input and output pins.

Table 15 - 1 Configuration of Serial Array Unit

Item	Configuration
Shift register	8 or 9 bits[Note 1]
Buffer register	Lower 8 bits or 9 bits of serial data register mn (SDRmn)[Notes 1, 2]
Serial clock I/O	SCK00, SCK01, SCK10, SCK11, SCK20, SCK21, SCK30, SCK31 pins (for simplified SPI), SCL00, SCL01, SCL10, SCL11, SCL20, SCL21, SCL30, SCL31 pins (for simplified I^2C)
Serial data input	SI00, SI01, SI10, SI11, SI20, SI21, SI30, SI31 pins (for simplified SPI), RxD0, RxD1, RxD3 pins (for UART), RxD2 pin (for UART supporting the LIN bus)
Serial data output	SO00, SO01, SO10, SO11, SO20, SO21, SO30, SO31 pins (for simplified SPI), TxD0, TxD1, TxD3 pins (for UART), TxD2 pin (for UART supporting the LIN bus)
Serial data I/O	SDA00, SDA01, SDA10, SDA11, SDA20, SDA21, SDA30, SDA31 pins (for simplified I^2C)
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Peripheral reset control register 0 (PRR0) • 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 standby control register m (SSCm) • 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 (PIMxx) • Port output mode registers (POMxx) • Port mode control A registers (PMCAxx) • Port mode control T registers (PMCTxx) • Port mode control E registers (PMCExx) • Port mode registers (PMxx) • Port registers (Pxx) • Port function output enable register (PFOE1) • UART loopback select register (ULBS)

Note 1. The number of bits used as the shift register and buffer register differs depending on the unit and channel.
- 30- to 64-pin R7F100G and 32-pin R7F104G products and mn = 00, 01: Lower 9 bits
- 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products and mn = 00, 01, 10, 11: Lower 9 bits
- Other than above: Lower 8 bits

Note 2. 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: SIOp (CSIp data register)
- UARTq reception: RXDq (UARTq receive data register)
- UARTq transmission: TXDq (UARTq transmit data register)
- IICr communication: SIOr (IICr data register)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31)
q: UART number (q = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21, 30, 31)

Figure 15 - 1 shows the block diagram of serial array unit 0.

Figure 15 - 1 Block Diagram of Serial Array Unit 0

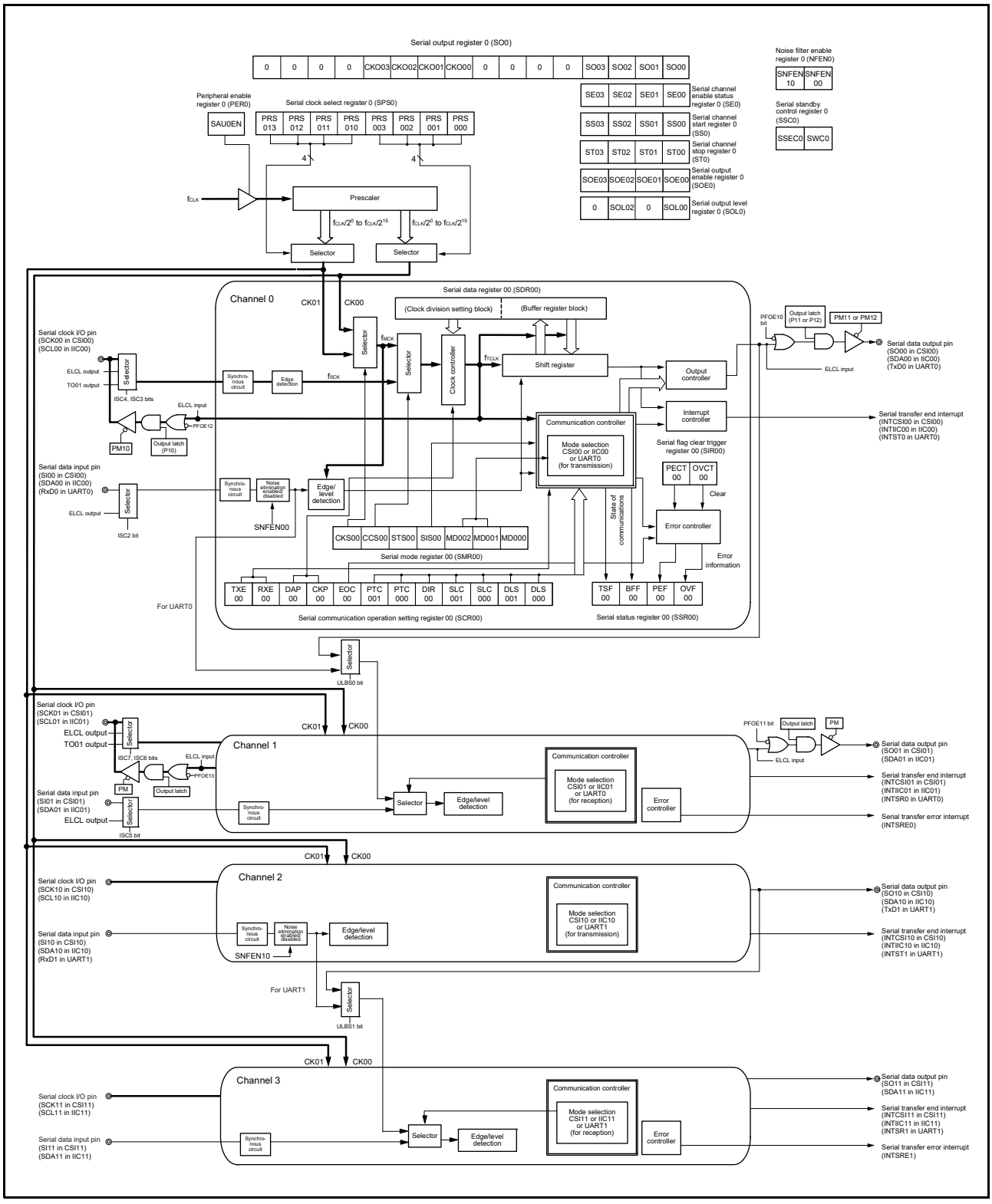
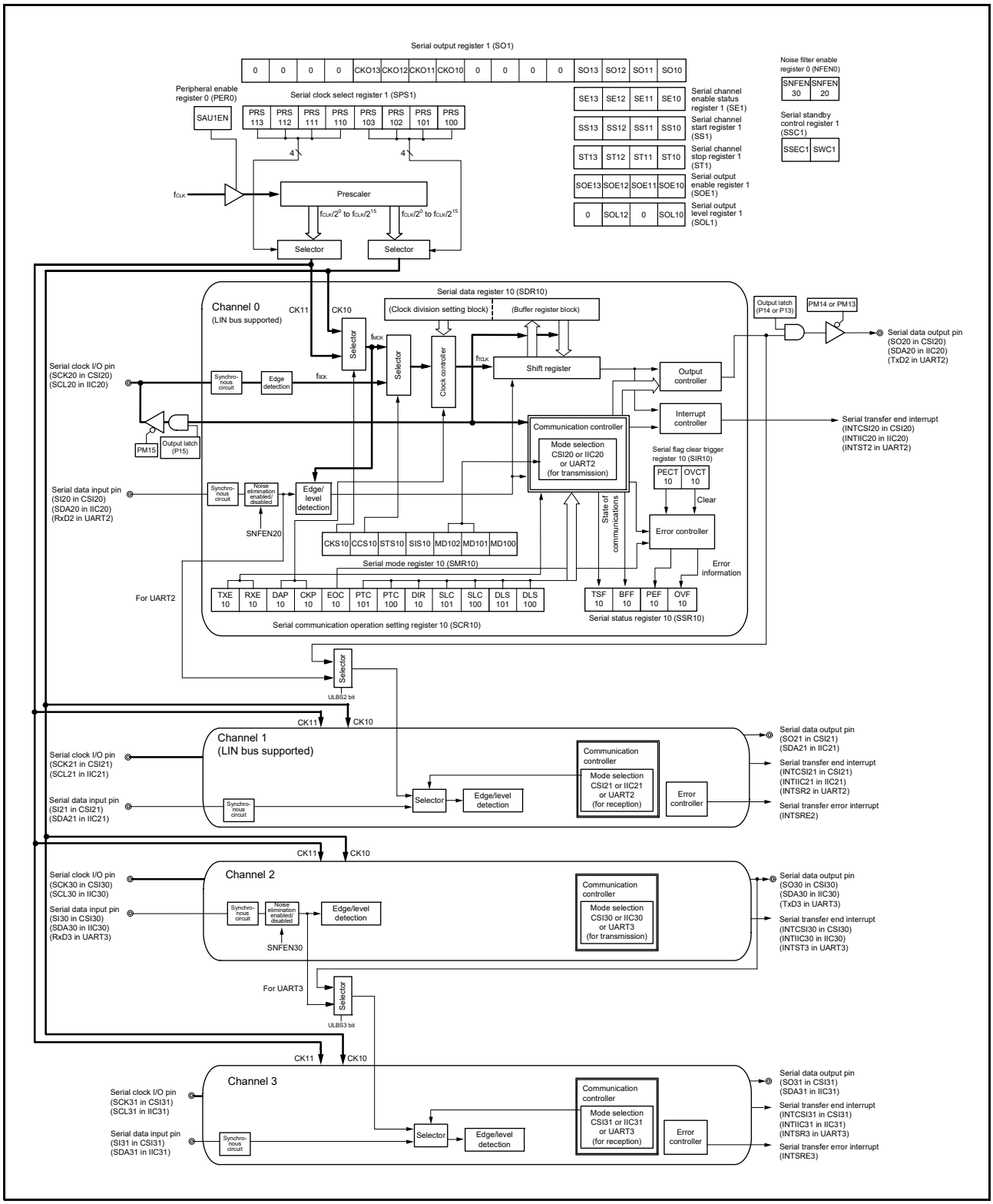


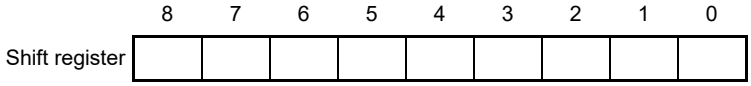
Figure 15 - 2 shows the block diagram of serial array unit 1.

Figure 15 - 2 Block Diagram of Serial Array Unit 1



15.2.1 Shift register

This is a 9-bit register that converts parallel data into serial data or vice versa. In case of the UART communication of nine bits of data, nine bits (bits 8 to 0) are used[Note]. 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 lower 8 or 9 bits of serial data register mn (SDRmn).



Note Only the following UARTs support the 9-bit data length.
- UART0 in the 30- to 64-pin R7F100G and 32-pin R7F104G products
- UART0 and UART2 in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products

15.2.2 Lower 8 or 9 bits of the serial data register mn (SDRmn)

The SDRmn is the transmit/receive data register (16 bits) of channel n. Bits 8 to 0 (lower 9 bits)[Note 1] or bits 7 to 0 (lower 8 bits) function as a transmit/receive buffer register, and bits 15 to 9 are used as a register that sets the division ratio of the operating clock (f_{MCK}). When data is received, parallel data converted by the shift register is stored in the lower 8 or 9 bits. When data is to be transmitted, set transmit data to be transferred to the shift register to the lower 8 or 9 bits. The data stored in the lower 8 or 9 bits of this register is as follows, depending on the setting of bits 0 and 1 (DLSmn0, DLSmn1) 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 the SDRmn register)
- 8-bit data length (stored in bits 0 to 7 of the SDRmn register)
- 9-bit data length (stored in bits 0 to 8 of the SDRmn register)[Note 1]

The SDRmn register can be read or written in 16-bit units. The lower 8 or 9 bits of the SDRmn register can be read or written[Note 2] as the following SFR, depending on the communication mode.

- CSIp communication: SIOp (CSIp data register)
- UARTq reception: RXDq (UARTq receive data register)
- UARTq transmission: TXDq (UARTq transmit data register)
- IICr communication: SIOr (IICr data register)

The value of each SDRmn register following a reset is 0000H.

Note 1. Only the following UARTs support the 9-bit data length.
- UART0 in the 30- to 64-pin R7F100G and 32-pin R7F104G products
- UART0 and UART2 in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products

Note 2. When operation is stopped (SEmn = 0), do not rewrite the SDRmn[7:0] bits by an 8-bit memory manipulation instruction (SDRmn[15:9] are all cleared to 0).

Remark 1. After data is received, 0 is stored in bits 0 to 8 in bit portions that exceed the data length.

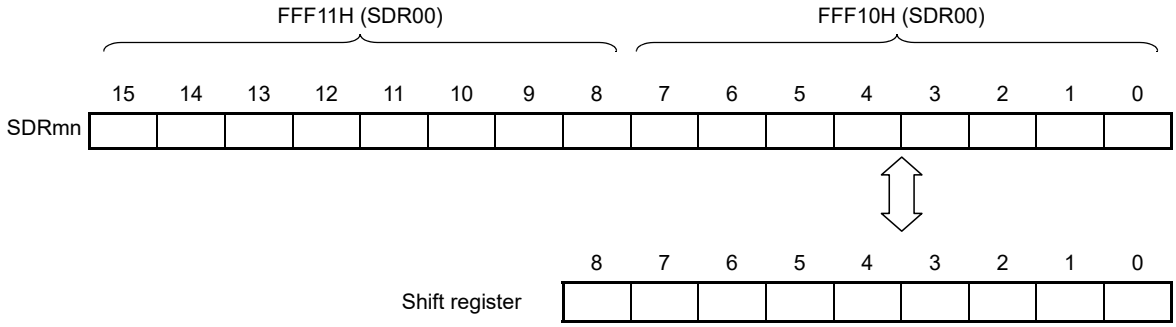
Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31) q: UART number (q = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21, 30, 31)

Figure 15 - 3 Format of Serial Data Register mn (SDRmn) (mn = 00, 01, 10, 11)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01)
FFF48H, FFF49H (SDR10)Note, FFF4AH, FFF4BH (SDR11)Note

After reset: 0000H

R/W: R/W



Note This register is only present in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products.

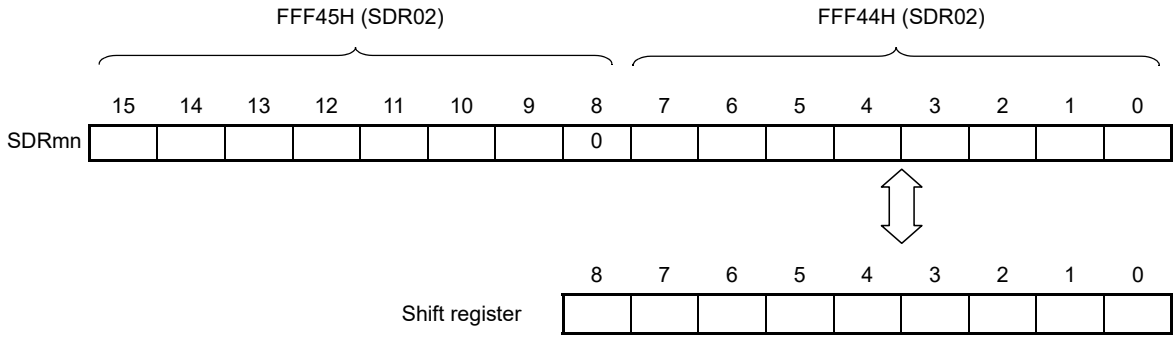
Remark For the function of the higher 7 bits of the SDRmn register, see **15.3 Registers for Controlling the Serial Array Unit**.

Figure 15 - 4 Format of Serial Data Register mn (SDRmn) (mn = 02, 03, 10, 11, 12, 13)

Address: FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03),
FFF48H, FFF49H (SDR10)Note 1, FFF4AH, FFF4BH (SDR11)Note 1
FFF14H, FFF15H (SDR12)Note 2, FFF16H, FFF17H (SDR13)Note 2

After reset: 0000H

R/W: R/W



Note 1. This register is only present in the 30- to 64-pin R7F100G and 32-pin R7F104G products.

Note 2. This register is only present in the 80- to 128-pin R7F100G products.

Caution **Be sure to leave bit 8 set to 0.**

Remark For the function of the higher 7 bits of the SDRmn register, see **15.3 Registers for Controlling the Serial Array Unit**.

15.3 Registers for Controlling the Serial Array Unit

The following registers are used to control the serial array unit.

- Peripheral enable register 0 (PER0)
- Peripheral reset control register 0 (PRR0)
- Serial clock select register m (SPSm)
- Serial mode register mn (SMRmn)
- Serial communication operation setting register mn (SCRmn)
- Serial data register mn (SDRmn)
- Serial flag clear trigger register mn (SIRmn)
- Serial status register mn (SSRmn)
- Serial channel start register m (SSm)
- Serial channel stop register m (STm)
- Serial channel enable status register m (SEm)
- Serial output enable register m (SOEm)
- Serial output register m (SOm)
- Serial output level register m (SOLm)
- Serial standby control register m (SSCm)
- Input switch control register (ISC)
- Noise filter enable register 0 (NFEN0)
- Port mode registers (PMxx)
- Port registers (Pxx)
- Port input mode registers (PIMxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- Port function output enable registers (PFOEx)
- UART loopback select register (ULBS)

15.3.1 Peripheral enable register 0 (PER0)

The PER0 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. If serial array unit 0 is to be used, be sure to set the SAU0EN bit of this register to 1. If serial array unit 1 is to be used, be sure to set the SAU1EN bit of this register to 1. The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of the PER0 register following a reset is 00H.

Figure 15 - 5 Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCWEN	IICA1EN Note 1	ADCEN	IICA0EN Note 2	SAU1EN	SAU0EN	TAU1EN Note 3	TAU0EN

SAUmEN	Control of supply of an input clock to serial array unit m
0	Stops supply of an input clock. • The SFRs used by serial array unit m cannot be written.Note 4
1	Enables supply of an input clock. • The SFRs used by serial array unit m can be read and written.

Note 1. This bit is only present in the 44- to 128-pin R7F100G products.
Note 2. This bit is only present in the 30- to 128-pin R7F100G and 48-pin R7F104G products.
Note 3. This bit is only present in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products.
Note 4. When an SFR used by serial array unit m is read, the value returned is 00H or 0000H regardless of the setting value.

Caution 1. Be sure to leave the following bits set to 0.
Bits 6 and 1 in the R7F100G products with 30 to 40 pins
Bit 1 in the R7F100G products with 44 to 64 pins
Bits 6, 4, and 1 in the R7F104G product with 32 pins
Bits 6 and 4 in the R7F104G product with 44 pins
Bit 6 in the R7F104G product with 48 pins

(Caution is listed on the next page.)

Caution 2. When setting serial array unit m, start by setting the following registers while the setting of the SAUmEN bit is 1.

- **Port function output enable register 1 (PFOE1)**
- **Serial clock select register m (SPSm)**
- **Serial mode register mn (SMRmn)**
- **Serial communication operation setting register mn (SCRmn)**
- **Serial data register mn (SDRmn)**
- **Serial flag clear trigger register mn (SIRmn)**
- **Serial status register mn (SSRmn)**
- **Serial channel start register m (SSm)**
- **Serial channel stop register m (STm)**
- **Serial channel enable status register m (SEm)**
- **Serial output enable register m (SOEm)**
- **Serial output level register m (SOLm)**
- **Serial output register m (SOm)**
- **Serial standby control register m (SSCm)**

If the setting of the SAUmEN bit is 0, writing to any of the registers which control serial array unit m is ignored. Note, however, writing to the following registers is valid.

- **Input switch control register (ISC)**
- **Noise filter enable register 0 (NFEN0)**
- **Port input mode registers (PIM0, PIM1, PIM4, PIM5, PIM8, PIM14)**
- **Port output mode registers (POM0, POM1, POM4, POM5, POM7 to POM9, POM14)**
- **Port mode control A registers (PMCA0, PMCA3, PMCA14)**
- **Port mode control T registers (PMCT0, PMCT3)**
- **Port mode control E register (PMCE0)**
- **Port mode registers (PM0, PM1, PM3 to PM5, PM7 to PM9, PM14)**
- **Port registers (P0, P1, P3 to P5, P7 to P9, P14)**

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

15.3.2 Peripheral reset control register 0 (PRR0)

The PRR0 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. Set the SAU0RES and SAU1RES bits to 1 to reset serial array units 0 and 1, respectively. The PRR0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of the PRR0 register following a reset is 00H.

Figure 15 - 6 Format of Peripheral Reset Control Register 0 (PRR0)

Address: F00F1H
After reset: 00H
R/W: R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PRR0	0	IICA1RES **Note 1**	ADCRES	IICA0RES **Note 2**	SAU1RES	SAU0RES	TAU1RES **Note 3**	TAU0RES

SAUmRES	Control resetting of serial array unit m
0	Serial array unit m is released from the reset state.
1	Serial array unit m is in the reset state. • The SFRs for use with serial array unit m are initialized.

Note 1. This bit is only present in the 44- to 128-pin R7F100G products.
Note 2. This bit is only present in the 30- to 128-pin R7F100G and 48-pin R7F104G products.
Note 3. This bit is only present in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products.

Caution 1. Be sure to leave bit 7 set to 0.
Caution 2. Be sure to leave the following bits set to 0.
Bits 6 and 1 in the R7F100G products with 30 to 40 pins
Bit 1 in the R7F100G products with 44 to 64 pins
Bits 6, 4, and 1 in the R7F104G product with 32 pins
Bits 6 and 4 in the R7F104G product with 44 pins
Bit 6 in the R7F104G product with 48 pins

15.3.3 Serial clock select register m (SPSm)

The SPSm is a 16-bit register that is used to select two types of operating 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. The 8 lower-order bits of an SPSm register can be handled as the SPSmL register, which can be set by an 8-bit memory manipulation instruction. The value of each SPSm register following a reset is 0000H.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 15 - 7 Format of Serial Clock Select Register m (SPSm)

Address: F0126H, F0127H (SPS0), F0166H, F0167H (SPS1)**Note**
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SPSm	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
PRSm13	PRSm12	PRSm11	PRSm10	PRSm03	PRSm02	PRSm01	PRSm00

PRS mk3	PRS mk2	PRS mk1	PRS mk0	Selection of operating clock (CKmk)**Note**				
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 32 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz	32 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	8 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	4 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz	2 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156 kHz	313 kHz	625 kHz	1 MHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz	500 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	250 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	62.5 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	31.3 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	15.6 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz	3.91 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz	1.95 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61 Hz	153 Hz	305 Hz	610 Hz	977 Hz

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) = 000FH) the operation of the serial array unit (SAU).

Caution Be sure to leave bits 15 to 8 set to 0.

(Remarks are listed on the next page.)

Remark 1. f_{CLK}: CPU/peripheral hardware clock frequency
Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)
Remark 3. k = 0, 1

15.3.4 Serial mode register mn (SMRmn)

The SMRmn register is used to set an operation mode of channel n. It is also used to select an operating clock (fMCK), specify whether the serial clock (fSCK) may be input or not, set a start trigger, the operating mode (as simplified SPI or CSI, UART, or simplified I2C), 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. The SMRmn register can be set by a 16-bit memory manipulation instruction. The value of each SMRmn register following a reset is 0020H.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 15 - 8 Format of Serial Mode Register mn (SMRmn) (1/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03), F0150H, F0151H (SMR10) to F0156H, F0157H (SMR13)**Note 1**

After reset: 0020H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SMRmn	CKSmn	CCSmn	0	0	0	0	0	STSmn **Note 2**

7	6	5	4	3	2	1	0
0	SISmn0 **Note 2**	1	0	0	MDmn2	MDmn1	MDmn0

CKSmn	Selection of operating clock (fMCK) of channel n
0	Operating clock CKm0 set by the SPSm register
1	Operating clock CKm1 set by the SPSm register
Operating clock (fMCK) 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 (fTCLK) is generated.	

CCSmn	Selection of transfer clock (fTCLK) of channel n
0	Divided operating clock fMCK specified by the CKSmn bit
1	Clock input fSCK from the SCKp pin (slave transfer in simplified SPI or CSI mode)
Transfer clock fTCLK is used for the shift register, communication controller, output controller, interrupt controller, and error controller. When CCSmn = 0, the division ratio of operating clock (fMCK) 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 simplified SPI or CSI, UART transmission, and simplified I2C).
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.	

Figure 15 - 8 Format of Serial Mode Register mn (SMRmn) (2/2)

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.

MDmn2	MDmn1	Setting of operation mode of channel n
0	0	Simplified SPI (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 continuous transmission, set this bit to 1 and write the next transmit data when SDRmn data has run out.	

Note 1. SMR00 to SMR03, SMR10, and SMR11 are present in all products.
SMR12 and SMR13 are only present in the R7F100G products with 80 to 128 pins.

Note 2. This bit is only present in the SMR01, SMR03, SMR11, and SMR13 registers.

Caution **Be sure to leave bits 13 to 9, 7, 4, and 3 (or bits 13 to 6, 4, and 3 for the SMR00, SMR02, SMR10, or SMR12 register) set to 0. Be sure to leave bit 5 set to 1.**

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), q: UART number (q = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21, 30, 31)

15.3.5 Serial communication operation setting register mn (SCRmn)

The SCRmn is a communication operation setting register of channel n. It is used to set a data transmission/reception mode, phase of data and clock, whether an error signal is to be masked or not, parity bit, start bit, stop bit, and data length. Rewriting the SCRmn register is prohibited when the register is in operation (when SEmn = 1). The SCRmn register can be set by a 16-bit memory manipulation instruction. The value of each SCRmn register following a reset is 0087H.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 15 - 9 Format of Serial Communication Operation Setting Register mn (SCRmn) (1/2)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03), F0158H, F0159H (SCR10) to F015EH, F015FH (SCR13)**Note 1**

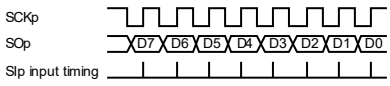
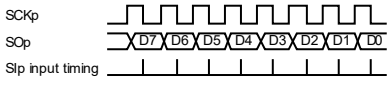
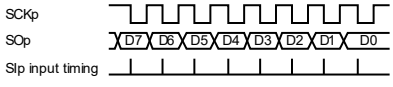
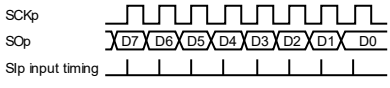
After reset: 0087H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn	0	EOCmn	PTCmn1	PTCmn0

7	6	5	4	3	2	1	0
DIRmn	0	SLCmn1 **Note 2**	SLCmn0	0	1	DLSmn1 **Note 3**	DLSmn0

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 simplified SPI (CSI) mode	Type
0	0		1
0	1		2
1	0		3
1	1		4
Set DAPmn and CKPmn to 00B in the UART mode and simplified I^2C mode.			

EOCmn	Mask control of error interrupt signal (INTSREx (x = 0 to 3))
0	Disables generation of error interrupt INTSREx (INTSRx is generated).
1	Enables generation of error interrupt INTSREx (INTSRx is not generated if an error occurs).
Set EOCmn to 0 in the simplified SPI (CSI) mode, simplified I^2C mode, and during UART transmission**Note 4**.	

Figure 15 - 9 Format of Serial Communication Operation Setting Register mn (SCRmn) (2/2)

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 5**.	No parity judgment
1	0	Outputs even parity.	Judged as even parity.
1	1	Outputs odd parity.	Judges as odd parity.
Set PTCmn1 and PTCmn to 00B in the simplified SPI (CSI) mode and simplified I^2C mode.			

DIRmn	Selection of data transfer sequence in simplified SPI (CSI) and UART modes
0	Inputs/outputs data with MSB first.
1	Inputs/outputs data with LSB first.
Set DIRmn to 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, 02, 10, 12 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 = 01B) during UART reception and in the simplified I^2C mode. Set no stop bit (SLCmn1, SLCmn0 = 00B) in the simplified SPI (CSI) mode. Set 1 bit (SLCmn1, SLCmn0 = 01B) or 2 bits (SLCmn1, SLCmn0 = 10B) during UART transmission.		

DLSmn1	DLSmn0	Setting of data length in simplified SPI (CSI) and UART modes
0	1	9-bit data length (stored in bits 0 to 8 of the SDRmn register) (settable in UART mode only)
1	0	7-bit data length (stored in bits 0 to 6 of the SDRmn register)
1	1	8-bit data length (stored in bits 0 to 7 of the SDRmn register)
Other than above		Setting prohibited
Set DLSmn1 and DLSmn0 to 11B in the simplified I^2C mode.		

Note 1. SCR00 to SCR03, SCR10, and SCR11 are present in all products.
SCR12 and SCR13 are only present in the R7F100G products with 80 to 128 pins.

Note 2. This bit is only present in the SCR00, SCR02, SCR10, and SCR12 registers.

Note 3. This bit is only present in the following registers; in other registers, it is fixed to 1.
- SCR00 and SCR01 registers
- SCR10 and SCR11 registers of the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products

Note 4. An INTSREx error interrupt might occur unless the setting of EOCmn in CSImn is 0.

Note 5. 0 is appended to the data value regardless of the contents.

Caution **Be sure to leave bits 11, 6, and 3 set to 0. Also, leave bit 5 of the SCR01, SCR03, SCR11, or SCR13 register set to 0. Be sure to leave bit 2 set to 1.**

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31)

15.3.6 Serial data register mn (SDRmn)

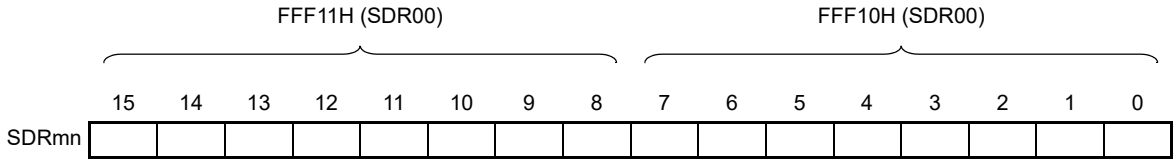
The SDRmn is the transmit/receive data register (16 bits) of channel n. Bits 8 to 0 (lower 9 bits) of the SDR00, SDR01, SDR10**Note 1**, and SDR11**Note 1** registers or bits 7 to 0 (lower 8 bits) of the SDR02, SDR03, SDR10**Note 2**, SDR11**Note 2**, SDR12**Note 3**, and SDR13**Note 3** registers function as a transmit/receive buffer register, and bits 15 to 9 (higher 7 bits) are used as a register that sets the division ratio of the operating clock (fMCK). If the CCSmn bit of serial mode register mn (SMRmn) is set to 0, the clock set by dividing the operating clock by bits 15 to 9 (higher 7 bits) of the SDRmn register is used as the transfer clock. If the CCSmn bit of serial mode register mn (SMRmn) is set to 1, set bits 15 to 9 (upper 7 bits) of the SDR00, SDR01, SDR10**Note 1**, and SDR11**Note 1** registers to 0000000B. The input clock fSCK (slave transfer in simplified SPI or CSI mode) from the SCKp pin is used as the transfer clock. The lower 8 or 9 bits of the SDRmn register function as a transmit/receive buffer register. During reception, the parallel data converted by the shift register is stored in the lower 8 or 9 bits, and during transmission, the data to be transmitted to the shift register is set to the lower 8 or 9 bits. The SDRmn register can be read or written in 16-bit units. However, the higher 7 bits can only be written or read when the operation is stopped (SEmn = 0). During operation (SEmn = 1), a value is written only to the lower 8 or 9 bits of the SDRmn register. When the SDRmn register is read during operation, the higher 7 bits are always read as 0. The value of each SDRmn register following a reset is 0000H.

Note 1. This register is only present in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products.
Note 2. This register is only present in the 30- to 64-pin R7F100G and 32-pin R7F104G products.
Note 3. This register is only present in the 80- to 128-pin R7F100G products.

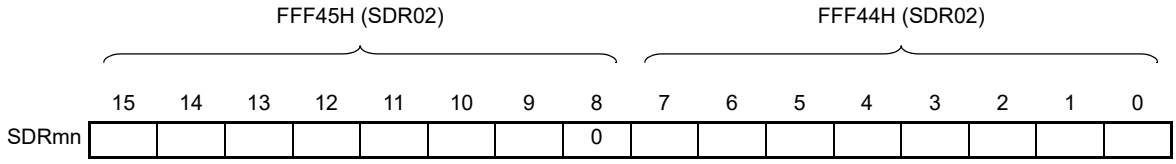
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 15 - 10 Format of Serial Data Register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01)
FFF48H, FFF49H (SDR10)Note 1, FFF4AH, FFF4BH (SDR11)Note 1
After reset: 0000H
R/W: R/W



Address: FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03)
FFF48H, FFF49H (SDR10)Note 2, FFF4AH, FFF4BH (SDR11)Note 2
FFF14H, FFF15H (SDR12)Note 3, FFF16H, FFF17H (SDR13)Note 3
After reset: 0000H
R/W: R/W



SDRmn[15:9]							Transfer clock setting by dividing the operating clock
0	0	0	0	0	0	0	fMCK/2
0	0	0	0	0	0	1	fMCK/4
0	0	0	0	0	1	0	fMCK/6
0	0	0	0	0	1	1	fMCK/8
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	1	1	1	0	fMCK/254
1	1	1	1	1	1	1	fMCK/256

Note 1. This register is only present in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products.
Note 2. This register is only present in the 30- to 64-pin R7F100G and 32-pin R7F104G products.
Note 3. This register is only present in the 80- to 128-pin R7F100G products.

Caution 1. Be sure to leave bit 8 of each of the following registers set to 0.
SDR02, SDR03, SDR12, and SDR13 registers
SDR10 and SDR11 registers of the 30- to 64-pin R7F100G and 32-pin R7F104G products
Caution 2. Setting SDRmn[15:9] to 0000000B or 0000001B is prohibited when UART is used.
Caution 3. Setting SDRmn[15:9] to 0000000B is prohibited when simplified I²C is used. Set SDRmn[15:9] to 0000001B or greater.
Caution 4. When operation is stopped (SEmn = 0), do not rewrite the SDRmn[7:0] bits by an 8-bit memory manipulation instruction (the SDRmn[15:9] bits are all cleared to 0).

Remark 1. For the function of the lower 8 or 9 bits of the SDRmn register, see **15.2 Configuration of Serial Array Unit**.
Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

15.3.7 Serial flag clear trigger register mn (SIRmn)

The SIRmn 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 serial status register mn is cleared to 0. Because the SIRmn is a trigger register, it is cleared immediately when the corresponding bit of the SSRmn register is cleared. The SIRmn register can be set by a 16-bit memory manipulation instruction. The 8 lower-order bits of an SIRmn register can be handled as the SIRmnL register, which can be set by an 8-bit memory manipulation instruction. The value of each SIRmn register following a reset is 0000H.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 15 - 11 Format of Serial Flag Clear Trigger Register mn (SIRmn)

Address: F0108H, F0109H (SIR00) to F010EH, F010FH (SIR03),
F0148H, F0149H (SIR10) to F014EH, F014FH (SIR13)**Note 1**

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SIRmn	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	FECTmn **Note 2**	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.

Note 1. SIR00 to SIR03, SIR10, and SIR11 are present in all products.
SIR12 and SIR13 are only present in the R7F100G products with 80 to 128 pins.

Note 2. This bit is only present in the SIR01, SIR03, SIR11, and SIR13 registers.

Caution Be sure to leave bits 15 to 3 (or bits 15 to 2 for the SIR00, SIR02, SIR10, or SIR12 register) set to 0.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Remark 2. When the SIRmn register is read, 0000H is always read.

15.3.8 Serial status register mn (SSRmn)

The SSRmn register indicates the state of communications and occurrence of errors for 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 8 lower-order bits of an SSRmn register can be handled as the SSRmnL register, which can be read by an 8-bit memory manipulation instruction. The value of each SSRmn register following a reset is 0000H.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 15 - 12 Format of Serial Status Register mn (SSRmn) (1/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03),
F0140H, F0141H (SSR10) to F0146H, F0147H (SSR13)**Note 1**

After reset: 0000H

R/W: R

Symbol	15	14	13	12	11	10	9	8
SSRmn	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	TSFmn	BFFmn	0	0	FEFmn **Note 2**	PEFmn	OVFmn

TSFmn	Flag indicating the state of communications for 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	Flag indicating the state of the buffer register for 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 condition>
- Transmit data is written to the SDRmn register while the TXEmn bit of the SCRmn register is set to 1 (transmission or transmission/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/reception mode in each communication mode).
- A reception error occurs.

Figure 15 - 12 Format of Serial Status Register mn (SSRmn) (2/2)

FEFmn	Framing error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception).
<Clear conditions> • 1 is written to the FECTmn bit of the SIRmn register. <Set condition> • A stop bit is not detected when UART reception ends.	

PEFmn	Parity/ACK error detection flag of channel n
0	No error occurs.
1	Parity error occurs (during UART reception) or ACK is not detected (during I^2C transmission).
<Clear conditions> • 1 is written to the PECTmn bit of the SIRmn register. <Set condition> • 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 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 conditions> • 1 is written to the OVCTmn bit of the SIRmn register. <Set condition> • 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/reception mode in each communication mode). • Transmit data is not ready for slave transmission or transmission/reception in simplified SPI (CSI) mode.	

Note 1. SSR00 to SSR03, SSR10, and SSR11 are present in all products.
SSR12 and SSR13 are only present in the R7F100G products with 80 to 128 pins.

Note 2. This bit is only present in the SSR01, SSR03, SSR11, and SSR13 registers.

Caution 1. 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.

Caution 2. When the simplified SPI (CSI) is handling reception in the SNOOZE mode (SWCm = 1), the OVFmn and BFFmn flags will not change.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

15.3.9 Serial channel start register m (SSm)

The SSm is a trigger register that is used to enable starting communication/count by each channel. When 1 is written to a bit (SSmn) of this register, 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. The SSm register can be set by a 16-bit memory manipulation instruction. The 8 lower-order bits of an SSm register can be handled as the SSmL register, which can be set by a 1-bit or 8-bit memory manipulation instruction. The value of each SSm register following a reset is 0000H.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 15 - 13 Format of Serial Channel Start Register m (SSm)

Address: F0122H, F0123H (SS0)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SS0	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	SS03	SS02	SS01	SS00

Address: F0162H, F0163H (SS1)Note
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SS1	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	SS13	SS12	SS11	SS10

SSmn	Operation start trigger of channel n
0	No trigger operation
1	Set the SEmn bit to 1 to place the channel in the communications waiting state.Note

Note Setting an SSmn bit to 1 during communications stops communications through channel n and places the channel in the waiting state. At this time, the values of the control registers and shift register, the states of the SCKmn and SOmn pins, and the values of the FEFmn, PEFmn, and OVFmn flags are retained.

Caution 1. Be sure to leave the following bits set to 0.
Bits 15 to 4 of the SS0 register
Bits 15 to 2 of the SS1 register for the 30- to 64-pin R7F100G and 32- to 48-pin R7F104G products
Bits 15 to 4 of the SS1 register for the 80- to 128-pin R7F100G products

Caution 2. For the UART reception, set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after at least 4 fMCK clock cycles have elapsed.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)
Remark 2. When the SSm register is read, 0000H is always read.

15.3.10 Serial channel stop register m (STm)

The STm is a trigger register that is used to enable stopping communication/count by each channel. When 1 is written to a bit (STmn) of this register, 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. The STm register can be set by a 16-bit memory manipulation instruction. The 8 lower-order bits of an STm register can be handled as the STmL register, which can be set by a 1-bit or 8-bit memory manipulation instruction. The value of each STm register following a reset is 0000H.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 15 - 14 Format of Serial Channel Stop Register m (STm)

Address: F0124H, F0125H (ST0)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
ST0	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	ST03	ST02	ST01	ST00

Address: F0164H, F0165H (ST1)**Note**
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
ST1	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	ST13	ST12	ST11	ST10

STmn	Operation stop trigger of channel n
0	No trigger operation
1	Clears the SEmn bit to 0 and stops the communication operation**Note**.

Note The values of the control registers and shift register, the states of the SCKmn and SOmn pins, and the values of the FEFmn, PEFmn, and OVFmn flags are retained.

Caution **Be sure to leave the following bits set to 0.**
Bits 15 to 4 of the ST0 register
Bits 15 to 2 of the ST1 register for the 30- to 64-pin R7F100G and 32- to 48-pin R7F104G products
Bits 15 to 4 of the ST1 register for the 80- to 128-pin R7F100G products

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)
Remark 2. When the STm register is read, 0000H is always read.

15.3.11 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 to a bit of serial channel start register m (SSm), the corresponding bit of this register is set to 1. When 1 is written to a bit of serial channel stop register m (STm), the corresponding bit is cleared to 0. For channel n whose operation is enabled, the value of the CKOmn bit of serial output register m (SOm) to be described later cannot be rewritten by software, and a value reflected by a communication operation is output from the serial clock pin. For channel n whose operation is stopped, the value of the CKOmn bit of the SOm register can be set by software and is output from the serial clock pin. In this way, any waveform, such as that of a start condition/stop condition, can be created by software. The SEm register can be read by a 16-bit memory manipulation instruction. The 8 lower-order bits of an SEm register can be handled as the SEmL register, which can be read by a 1-bit or 8-bit memory manipulation instruction. The value of each SEm register following a reset is 0000H.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 15 - 15 Format of Serial Channel Enable Status Register m (SEm)

Address: F0120H, F0121H (SE0)
After reset: 0000H
R/W: R

Symbol	15	14	13	12	11	10	9	8
SE0	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	SE03	SE02	SE01	SE00

Address: F0160H, F0161H (SE1)
After reset: 0000H
R/W: R

Symbol	15	14	13	12	11	10	9	8
SE1	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	SE13	SE12	SE11	SE10

SEmn	Indication of whether operation of channel n is enabled or stopped.
0	Operation stops
1	Operation is enabled.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

15.3.12 Serial output enable register m (SOEm)

The SOEm register is used to enable or stop output of the serial communication operation of each channel. For channel n whose serial output is enabled, the value of the SOmn bit of serial output register m (SOm) to be described later cannot be rewritten by software, 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, such as that of a start condition/stop condition, can be created by software. The SOEm register can be set by a 16-bit memory manipulation instruction. The 8 lower-order bits of an SOEm register can be handled as the SOEmL register, which can be set by a 1-bit or 8-bit memory manipulation instruction. The value of each SOEm register following a reset is 0000H.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 15 - 16 Format of Serial Output Enable Register m (SOEm)

Address: F012AH, F012BH (SOE0)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SOE0	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	SOE03	SOE02	SOE01	SOE00

Address: F016AH, F016BH (SOE1)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SOE1	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	SOE13	SOE12	SOE11	SOE10

SOEmn	Serial output enable/stop of channel n
0	Stops output by serial communication operation.
1	Enables output by serial communication operation.

Caution **Be sure to leave the following bits set to 0.**
Bits 15 to 4 of the SOE0 register
Bits 15 to 2 of the SOE1 register for the 30- to 64-pin R7F100G and 32- to 48-pin R7F104G products
Bits 15 to 4 of the SOE1 register for the 80- to 128-pin R7F100G products

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

15.3.13 Serial output register m (SOm)

The SOm 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. The SOm register can be set by a 16-bit memory manipulation instruction. The value of each SOm register following a reset is 0F0FH.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 15 - 17 Format of Serial Output Register m (SOm)

Address: F0128H, F0129H (SO0)
After reset: 0F0FH
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SO0	0	0	0	0	CKO03	CKO02	CKO01	CKO00

7	6	5	4	3	2	1	0
0	0	0	0	SO03	SO02	SO01	SO00

Address: F0168H, F0169H (SO1)
After reset: 0F0FH**Note**
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SO1	0	0	0	0	CKO13	CKO12	CKO11	CKO10

7	6	5	4	3	2	1	0
0	0	0	0	SO13	SO12	SO11	SO10

CKOmn	Serial clock output of channel n
0	Serial clock output value is 0.
1	Serial clock output value is 1.

SOmn	Serial data output of channel n
0	Serial data output value is 0.
1	Serial data output value is 1.

Note The value of the SO1 register following a reset is 0303H in the products below.
- 30- to 64-pin R7F100G products with 128 or fewer Kbytes of code flash memory
- 32-pin R7F104G product

(Cautions and Remark are listed on the next page.)

Caution 1. Be sure to leave bits 15 to 12 and 7 to 4 of the SO0 register set to 0.
Be sure to leave the following bits of the SO1 register set to 0, depending on the product:
- **Bits 15 to 10 and 7 to 2**
  - **30- to 64-pin R7F100G products with 128 or fewer Kbytes of code flash memory**
  - **32-pin R7F104G product**
- **Bits 15 to 12 and 7 to 4**
  - **30- to 64-pin R7F100G products with at least 192 Kbytes of code flash memory**
  - **80- to 128-pin R7F100G products**
  - **44- and 48-pin R7F104G products**

Caution 2. Leave the SO01 bit of the SO0 register set to 1 in the 32-pin R7F104G product. Leave the SO10 bit of the SO1 register set to 1 in the 44-pin R7F104G product.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

15.3.14 Serial output level register m (SOLm)

The SOLm register 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 the bit corresponding the channel used in the simplified SPI (CSI) mode or simplified 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). The SOLm register can be set by a 16-bit memory manipulation instruction. The 8 lower-order bits of an SOLm register can be handled as the SOLmL register, which can be set by an 8-bit memory manipulation instruction. The value of each SOLm register following a reset is 0000H.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 15 - 18 Format of Serial Output Level Register m (SOLm)

Address: F0134H, F0135H (SOL0)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SOL0	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	SOL02	0	SOL00

Address: F0174H, F0175H (SOL1)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SOL1	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	SOL12	0	SOL10

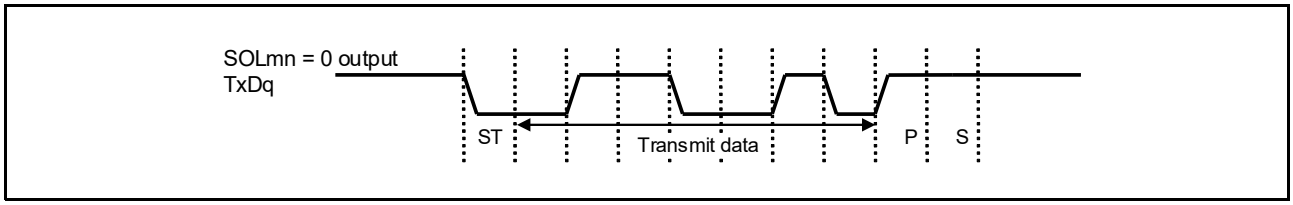
SOLmn	Selects inversion of the level of the transmit data of channel n in UART mode
0	Communication data is output as is.
1	Communication data is inverted and output.

Caution **Be sure to leave the following bits set to 0.**
Bits 15 to 3 and 1 of the SOL0 register
Bits 15 to 1 of the SOL1 register for the 30- to 64-pin R7F100G and 32- to 48-pin R7F104G products
Bits 15 to 3 and 1 of the SOL1 register for the 80- to 128-pin R7F100G products

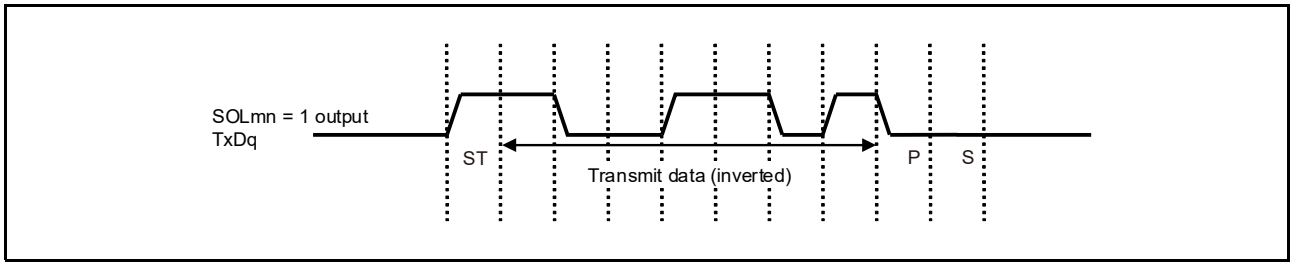
Figure 15 - 19 shows examples in which the level of transmit data is reversed during UART transmission.

Figure 15 - 19 Examples of Reverse Transmit Data

a) Non-reverse Output (SOLmn = 0)



b) Reverse Output (SOLmn = 1)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

15.3.15 Serial standby control register m (SSCm)

The SSC0 register is used to control the startup of reception (the SNOOZE mode) while in the STOP mode when receiving CSI00 or UART0 serial data. The SSC1[Note] register is used to control the startup of reception (the SNOOZE mode) while in the STOP mode when receiving CSI20 or UART2 serial data. The SSCm register can be set by a 16-bit memory manipulation instruction. The 8 lower-order bits of an SSCm register can be handled as the SSCmL register, which can be set by an 8-bit memory manipulation instruction. The value of each SSCm register following a reset is 0000H.

Caution The maximum transfer rate in the SNOOZE mode is as follows.

- **When using CSI00, CSI20: Up to 1 Mbps**
- **When using UART0, UART2: Up to 115.2 kbps (when setting FWKUP = 1, f_{CLK} = f_{IH} (32 MHz))**

Remark m: Unit number (m = 0, 1)

Figure 15 - 20 Format of Serial Standby Control Register m (SSCm)

Address: F0138H, F0139H (SSC0), F0178H, F0179H (SSC1)[Note]
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SSCm	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	SSECm	SWCm

SSECm	Selection of whether to enable or disable the generation of communication error interrupts in the SNOOZE mode
0	Enable the generation of error interrupts (INTSRE0/INTSRE2).
1	Disable the generation of error interrupts (INTSRE0/INTSRE2).

- The SSECm bit can be set to 1 or 0 only when both the SWCm and EOCmn bits are set to 1 during UART reception in the SNOOZE mode. In other cases, set the SSECm bit to 0.
- Setting SSECm and SWCm to 10B is prohibited.

SWCm	Setting of the SNOOZE mode
0	Do not use the SNOOZE mode function.
1	Use the SNOOZE mode function.

- When there is a hardware trigger signal in the STOP mode, the STOP mode is exited, and simplified SPI (CSI) or UART reception is performed without operating the CPU (the SNOOZE mode).
- The SNOOZE mode function can only be specified when the high-speed on-chip oscillator clock or medium-speed on-chip oscillator clock is selected for the CPU/peripheral hardware clock (f_{CLK}). If any other clock is selected, specifying this mode is prohibited.
- Even when using SNOOZE mode, be sure to set the SWCm bit to 0 in normal operation mode and change it to 1 just before shifting to STOP mode. Also, be sure to change the SWCm bit to 0 after returning from STOP mode to normal operation mode.

Note The SSC1 register is only present in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products.

Remark m: Unit number (m = 0, 1)

Table 15 - 2 Interrupt in UART Reception Operation in SNOOZE Mode

EOCmn Bit	SSECm Bit	Reception Ended Successfully	Reception Ended in an Error
0	0	INTSRx is generated.	INTSRx is generated.
0	1	INTSRx is generated.	INTSRx is generated.
1	0	INTSRx is generated.	INTSREx is generated.
1	1	INTSRx is generated.	No interrupt is generated.

15.3.16 Input switch control register (ISC)

The ISC1 and ISC0 bits of the ISC register are used to realize a LIN-bus communication by UART2 in coordination with an external interrupt and the timer array unit. The ISC2 bit is used to select the serial data input source of CSI00/UART0. The ISC4 and ISC3 bits are used to select the serial clock input source of CSI00. The ISC7 to ISC5 bits are used to select the serial data input source and serial clock input source of CSI01. When bit 0 is set to 1, the input signal of the serial data input (RxD2) pin is selected as an external interrupt (INTP0) that can be used to detect a wakeup signal. When bit 1 is set to 1, the input signal of the serial data input (RxD2) pin is selected as a timer input, so that wake up signal can be detected, the low-level width of the break field, and the pulse width of the sync field can be measured by the timer. When bit 2 is set to 1, the ELCL output signal can be selected as the serial data input source of CSI00/UART0. Bits 3 and 4 can be used to select the SCK00 pin input, ELCL output signal, or TO01 output signal as the serial clock input source of CSI00. When bit 5 is set to 1, the ELCL output signal can be selected as the serial data input source of CSI01. Bits 6 and 7 can be used to select the SCK01 pin input, ELCL output signal, or TO01 output signal as the serial clock input source of CSI01. The ISC register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of the ISC register following a reset is 00H.

Figure 15 - 21 Format of Input Switch Control Register (ISC)

Address: F0073H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ISC	ISC7	ISC6	ISC5	ISC4	ISC3	ISC2	ISC1	ISC0

ISC7	ISC6	Switch of the serial clock input source of CSI01Note 1
0	0	Input signal of the SCK01 pin (normal operation)
0	1	ELCL output signal
1	0	TO01 output signal
1	1	Setting prohibited

ISC5	Switch of the serial data input source of CSI01Note 2
0	Input signal of the SI01 pin (normal operation)
1	ELCL output signal

ISC4	ISC3	Switch of the serial clock input source of CSI00Note 3
0	0	Input signal of the SCK00 pin (normal operation)
0	1	ELCL output signal
1	0	TO01 output signal
1	1	Setting prohibited

ISC2	Switch of the serial data input of CSI00/UART0Note 4
0	Input signal of the SI00/RxD0 pin (normal operation)
1	ELCL output signal

ISC1	Switching channel 7 input of timer array unit
0	Uses the input signal of the TI07 pin as a timer input (normal operation).
1	Input signal of the RxD2 pin is used as timer input (detects the wakeup signal and measures the low-level width of the break field and the pulse width of the sync field).

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 RxD2 pin as an external interrupt (wakeup signal detection).

Note 1. When UART mode or simplified I^2C mode is to be selected for channel 1, set the ISC7 and ISC6 bits to 0.
Note 2. When UART mode or simplified I^2C mode is to be selected for channel 1, set the ISC5 bit to 0.
Note 3. When UART mode or simplified I^2C mode is to be selected for channel 0, set the ISC4 and ISC3 bits to 0.
Note 4. When simplified I^2C mode is to be selected for channel 0, set the ISC2 bit to 0.

15.3.17 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 simplified SPI (CSI) or simplified I^2C communication, by setting 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, after synchronization is performed with the operating clock (f_{MCK}) of the target channel, 2-clock match detection is performed. When the noise filter is disabled, only synchronization is performed with the operating clock (f_{MCK}) of the target channel. The NFEN0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of the NFEN0 register following a reset is 00H.

Figure 15 - 22 Format of Noise Filter Enable Register 0 (NFEN0)

Address: F0070H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
NFEN0	0	SNFEN30	0	SNFEN20	0	SNFEN10	0	SNFEN00

SNFEN30	Enabling/disabling use of the noise filter for the RxD3 pin
0	Disables use of the noise filter for the RxD3 pin.
1	Enables use of the noise filter for the RxD3 pin.
Set SNFEN30 to 1 to use the RxD3 pin. Set SNFEN30 to 0 to use the other than RxD3 pin.	

SNFEN20	Enabling/disabling use of the noise filter for the RxD2 pin
0	Disables use of the noise filter for the RxD2 pin.
1	Enables use of the noise filter for the RxD2 pin.
Set SNFEN20 to 1 to use the RxD2 pin. Set SNFEN20 to 0 to use the other than RxD2 pin.	

SNFEN10	Enabling/disabling use of the noise filter for the RxD1 pin
0	Disables use of the noise filter for the RxD1 pin.
1	Enables use of the noise filter for the RxD1 pin.
Set the SNFEN10 bit to 1 to use the RxD1 pin. Set SNFEN10 to 0 to use the other than RxD1 pin.	

SNFEN00	Enabling/disabling use of the noise filter for the RxD0 pin
0	Disables use of the noise filter for the RxD0 pin.
1	Enables use of the noise filter for the RxD0 pin.
Set the SNFEN00 bit to 1 to use the RxD0 pin. Set SNFEN00 to 0 to use the other than RxD0 pin.	

Caution Be sure to leave the following bits set to 0.
Bits 7 to 5, 3, and 1 for the 30- to 64-pin R7F100G and 32- to 48-pin R7F104G products
Bits 7, 5, 3, and 1 for the 80- to 128-pin R7F100G products

15.3.18 Registers for controlling the port functions multiplexed with the inputs and outputs of the serial array unit

Set the following registers to control the port functions multiplexed with the inputs and outputs of the serial array unit.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Port input mode registers (PIMxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)
- Port function output enable registers (PFOEx)

For details, see the following sections.

- **4.3.1 Port mode registers (PMxx)**
- **4.3.2 Port registers (Pxx)**
- **4.3.4 Port input mode registers (PIMxx)**
- **4.3.5 Port output mode registers (POMxx)**
- **4.3.7 Port mode control A registers (PMCAxx)**
- **4.3.8 Port mode control T registers (PMCTxx)**
- **4.3.9 Port mode control E registers (PMCExx)**
- **4.3.15 Port function output enable registers (PFOEx)**

For examples of register settings for controlling port functions, see **4.5.4 Examples of register settings for port and multiplexed functions**. To communicate with an external device operating at a different voltage (1.8 V, 2.5 V or 3 V), see **4.4.5 Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching I/O buffers**.

15.3.19 UART loopback select register (ULBS)

The ULBS register is used to enable the UART loopback function. This register has bits to individually control UART channels. When the bit corresponding to each channel is set to 1, the UART loopback function is selected, and output from the transmission shift register is looped back to the reception shift register. The ULBS register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of the ULBS register following a reset is 00H.

Figure 15 - 23 Format of UART Loopback Select Register (ULBS)

Address: F0079H
After reset: 00H
R/W: R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
ULBS	0	0	ULBS5	ULBS4	ULBS3	ULBS2	ULBS1	ULBS0

ULBSn	Selection of the UART loopback function
0	Inputs the states of the RxD0 to RxD3 pins of serial array unit UART0 to UART3 to the reception shift register.
1	Loops back output from the transmission shift register to the reception shift register.

Caution **Be sure to leave bits 7 and 6 set to 0.**

Remark n = 0 to 3

15.4 Operation Stop Mode

Each serial interface of serial array unit has the operation stop mode. In this mode, serial communication cannot be executed, thus reducing the power consumption. In addition, the pin for serial interface can be used as port function pins in this mode.

15.4.1 Stopping the operation by units

The stopping of the operation by units is set by using peripheral enable register 0 (PER0). The PER0 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. To stop the operation of serial array unit 0, be sure to set the SAU0EN bit to 0. To stop the operation of serial array unit 1, be sure to set the SAU1EN bit to 0.

Figure 15 - 24 Peripheral Enable Register 0 (PER0) Setting When Stopping the Operation by Units

a) Peripheral enable register 0 (PER0): Only set the bit corresponding to SAUm to be stopped to 0.

	7	6	5	4	3	2	1	0
PER0	RTCWEN ×	IICA1EN **Note 1** ×	ADCEN ×	IICA0EN **Note 2** ×	SAU1EN 0/1	SAU0EN 0/1	TAU1EN **Note 3** ×	TAU0EN ×

Control of SAUm input clock
0: Stops supply of input clock
1: Supplies input clock

Note 1. This bit is only present in the 44- to 128-pin R7F100G products.
Note 2. This bit is only present in the 30- to 128-pin R7F100G and 48-pin R7F104G products.
Note 3. This bit is only present in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products.

Caution 1. Be sure to leave the following bits set to 0.
Bits 6 and 1 in the R7F100G products with 30 to 40 pins
Bit 1 in the R7F100G products with 44 to 64 pins
Bits 6, 4, and 1 in the R7F104G product with 32 pins
Bits 6 and 4 in the R7F104G product with 44 pins
Bit 6 in the R7F104G product with 48 pins

Caution 2. If SAUmEN = 0, writing to the registers which control serial array unit m is ignored, and, even if the register is read, only the initial 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 (PIM0, PIM1, PIM4, PIM5, PIM8, PIM14)**
- **Port output mode registers (POM0, POM1, POM4, POM5, POM7 to POM9, POM14)**
- **Port mode control A registers (PMCA0, PMCA3, PMCA14)**
- **Port mode control T registers (PMCT0, PMCT3)**
- **Port mode control E register (PMCE0)**
- **Port mode registers (PM0, PM1, PM3 to PM5, PM7 to PM9, PM14)**
- **Port registers (P0, P1, P3 to P5, P7 to P9, P14)**

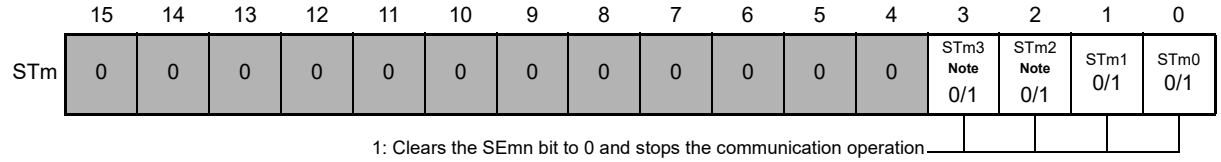
Remark ×: Bits not used with serial array units (depending on the settings of other peripheral functions)
0/1: Set to 0 or 1 depending on the user's application.

15.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

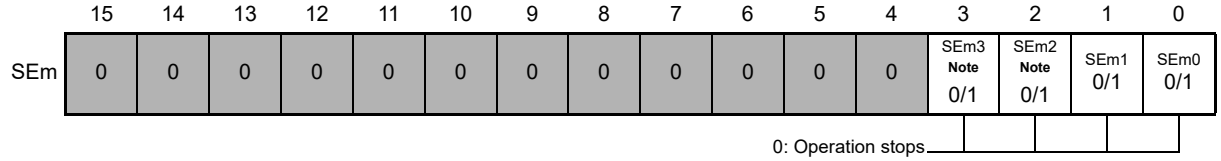
Figure 15 - 25 Each Register Setting When Stopping the Operation by Channels

a) Serial channel stop register m (STm): The STm is a trigger register that is used to enable stopping communication/count by each channel.



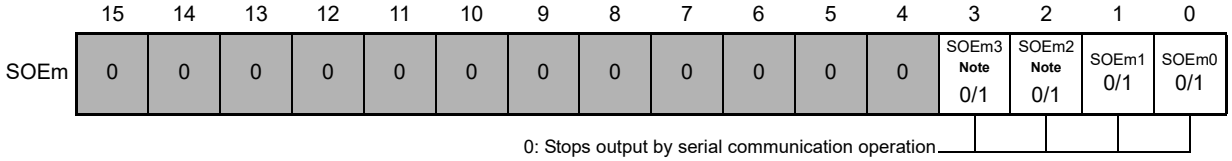
* Because the STmn bit is a trigger bit, it is cleared immediately when SEmn = 0.

b) Serial channel enable status register m (SEm): This register indicates whether data transmission/reception operation of each channel is enabled or stopped.



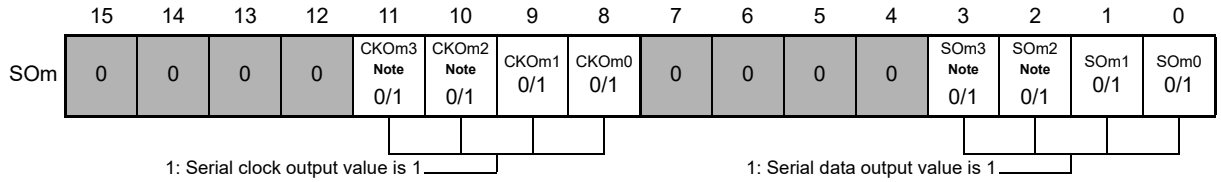
* The SEm is a read-only status register, whose operation is stopped by using the STm register. With a channel whose operation is stopped, the value of the CKOmn bit of the SOm register can be set by software.

c) Serial output enable register m (SOEm): This register is used to enable or stop output of the serial communication operation of each channel.



* For channel n, whose serial output is stopped, the SOmn bit value of the SOm register can be set by software.

d) Serial output register m (SOm): The SOm is a buffer register for serial output of each channel.



* When using pins corresponding to each channel as port function pins, set the corresponding CKOmn, SOmn bits to 1.

Note These bits are only present in the R7F100G products with 80 to 128 pins of serial array unit 1.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Remark 2. ■: Set to the initial value. 0/1: Set to 0 or 1 depending on the user's application.

15.5 Operation of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31) Communication

The simplified SPI allows clocked communications using three lines: serial clock (SCK), serial input (SI), and serial output (SO).

[Data transmission/reception]

- Data length of 7 or 8 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable

[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**Note** During master communication: Max. f_{CLK}/2 (CSI00 only)
  Max. f_{CLK}/4
  During slave communication: Max. f_{MCK}/6

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

Furthermore, only the following simplified SPIs (CSIs) support the SNOOZE mode. In the SNOOZE mode, data can be received without CPU processing upon detecting SCK input in the STOP mode.

- 30- to 64-pin R7F100G and 32- to 48-pin R7F104G products: CSI00
- 80- to 128-pin R7F100G products: CSI00, CSI20

Note Set up the transfer rate within a range satisfying the SCK cycle time (t_{KCY}). For details, see **Section 37 Electrical Characteristics**.

Caution Use a general-purpose port pin to send a chip select signal when required.

The channels supporting simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31) are channels 0 to 3 of both SAU0 and SAU1.

30- and 32-pin R7F100G products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting the LIN bus)	IIC20
	1	—		—

36- to 44-pin R7F100G products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting the LIN bus)	IIC20
	1	CSI21		IIC21

48- and 52-pin R7F100G products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting the LIN bus)	IIC20
	1	CSI21		IIC21

64-pin R7F100G products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting the LIN bus)	IIC20
	1	CSI21		IIC21

80- to 128-pin R7F100G products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting the LIN bus)	IIC20
	1	CSI21		IIC21
	2	CSI30	UART3	IIC30
	3	CSI31		IIC31

32- to 48-pin R7F104G products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	—		IIC11
1	0	—	UART2 (supporting the LIN bus)	—
	1	CSI21		IIC21

Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31) performs the following seven types of communication operations.

- Master transmission (See **15.5.1**.)
- Master reception (See **15.5.2**.)
- Master transmission/reception (See **15.5.3**.)
- Slave transmission (See **15.5.4**.)
- Slave reception (See **15.5.5**.)
- Slave transmission/reception (See **15.5.6**.)
- SNOOZE mode function (See **15.5.7**.)

15.5.1 Master transmission

Master transmission is that this MCU outputs a transfer clock and transmits data to another device.

Simplified SPI	CSI00	CSI01	CSI10	CSI11	CSI20	CSI21	CSI30	CSI31
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 2 of SAU1	Channel 3 of SAU1
Pins used	SCK00, SO00	SCK01, SO01	SCK10, SO10	SCK11, SO11	SCK20, SO20	SCK21, SO21	SCK30, SO30	SCK31, SO31
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21	INTCSI30	INTCSI31
	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 or 8 bits							
Transfer rate**Note**	Max. $f_{CLK}/2$ [Hz] (CSI00 only), $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK}: CPU/peripheral hardware clock frequency							
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data output starts from the start of the operation of the serial clock. • DAPmn = 1: Data output starts half a clock cycle before the start of the serial clock operation.							
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse							
Data direction	MSB or LSB first							

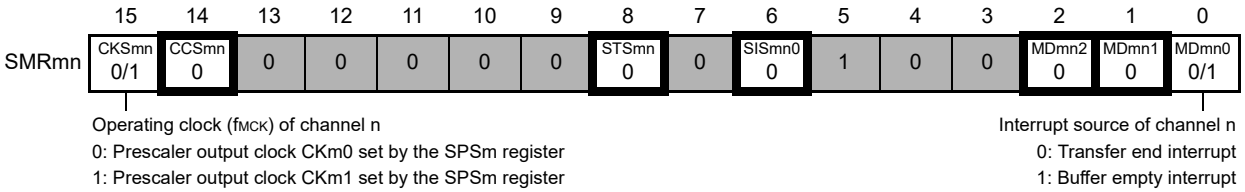
Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 37 Electrical Characteristics**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

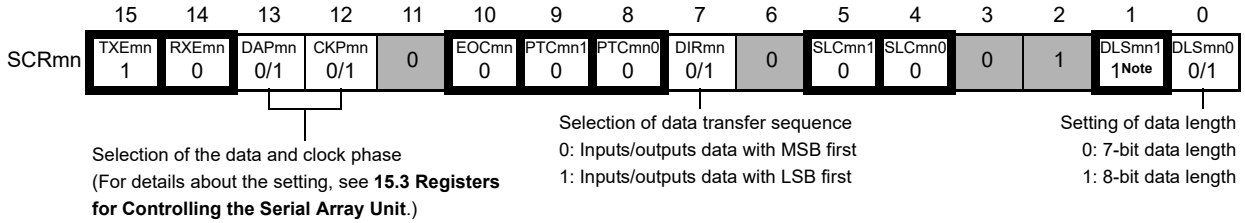
1. Register setting

Figure 15 - 26 Example of Contents of Registers for Master Transmission of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31)

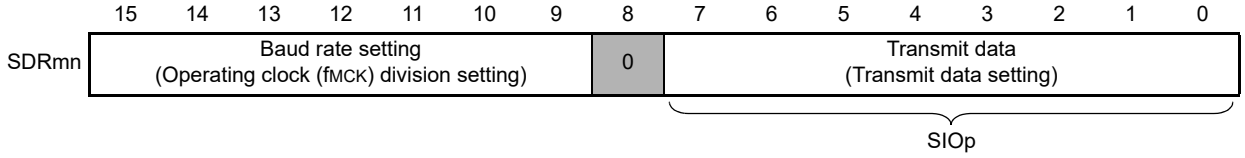
a) Serial mode register mn (SMRmn)



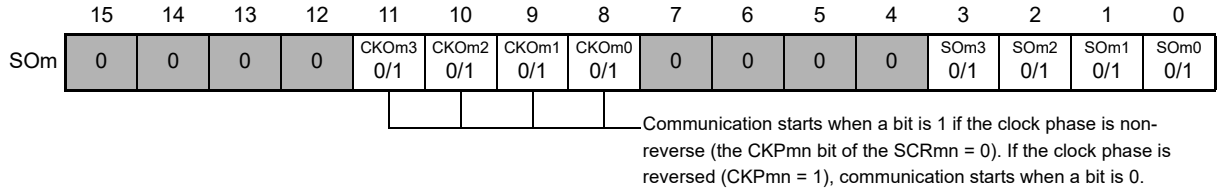
b) Serial communication operation setting register mn (SCRmn)



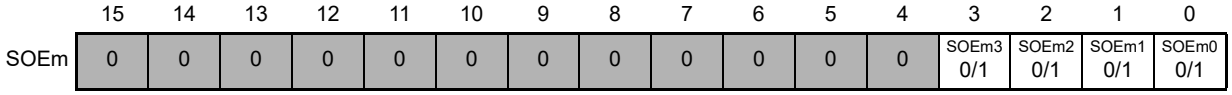
c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



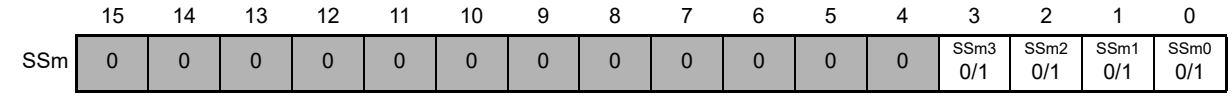
d) Serial output register m (SOm): Set only the bit of the target channel.



e) Serial output enable register m (SOEm): Set only the bit of the target channel to 1.



f) Serial channel start register m (SSm): Set only the bit of the target channel to 1.



Note This bit is only present in the following registers; in other registers, it is fixed to 1.
- SCR00 and SCR01 registers
- SCR10 and SCR11 registers of the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products

(Remarks are listed on the next page.)

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Remark 2. ■: Setting is fixed in the simplified SPI (CSI) master transmission mode, ■: Set to the initial value.
0/1: Set to 0 or 1 depending on the user's application.

2. Operation procedure

Figure 15 - 27 Initial Setting Procedure for Master Transmission

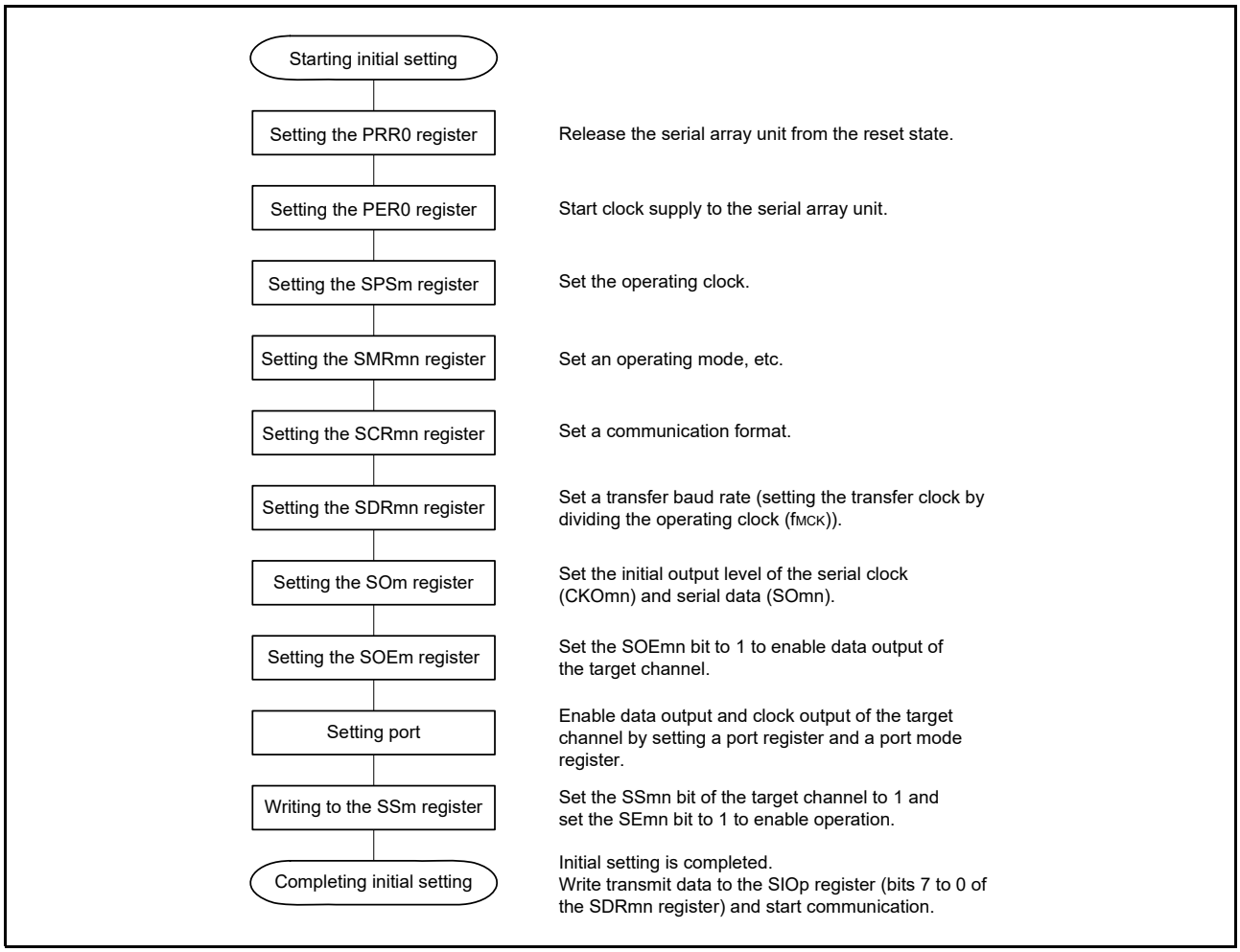


Figure 15 - 28 Procedure for Stopping Master Transmission

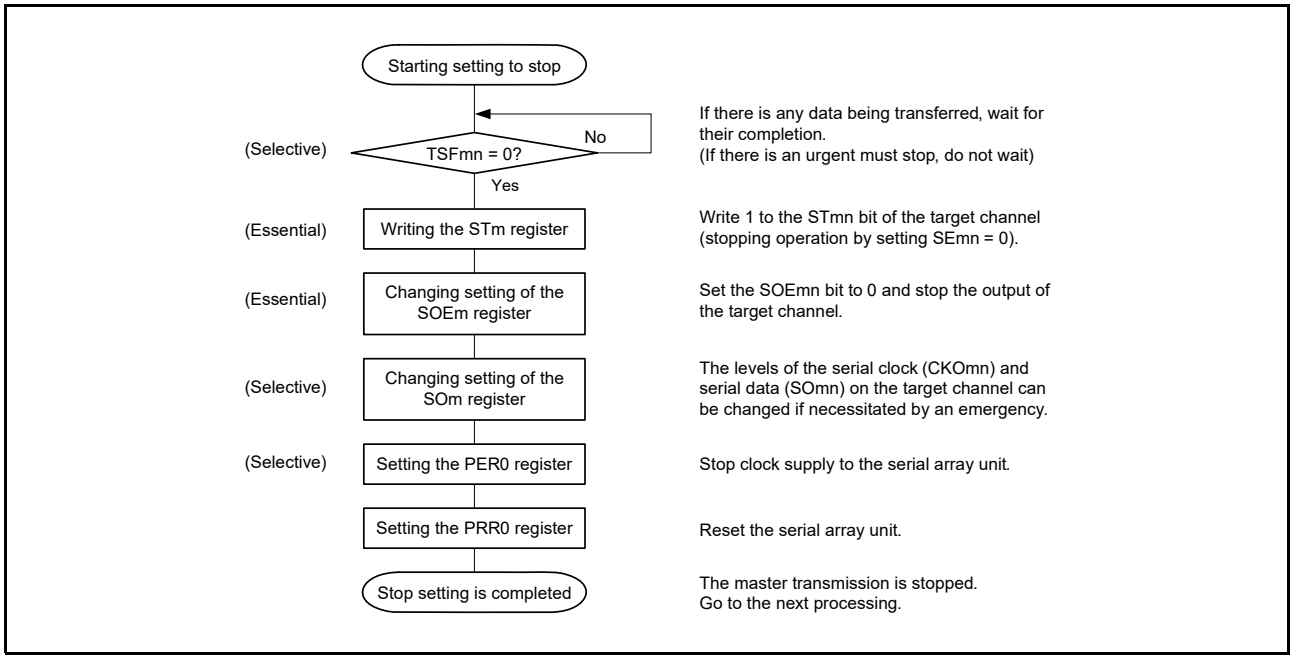
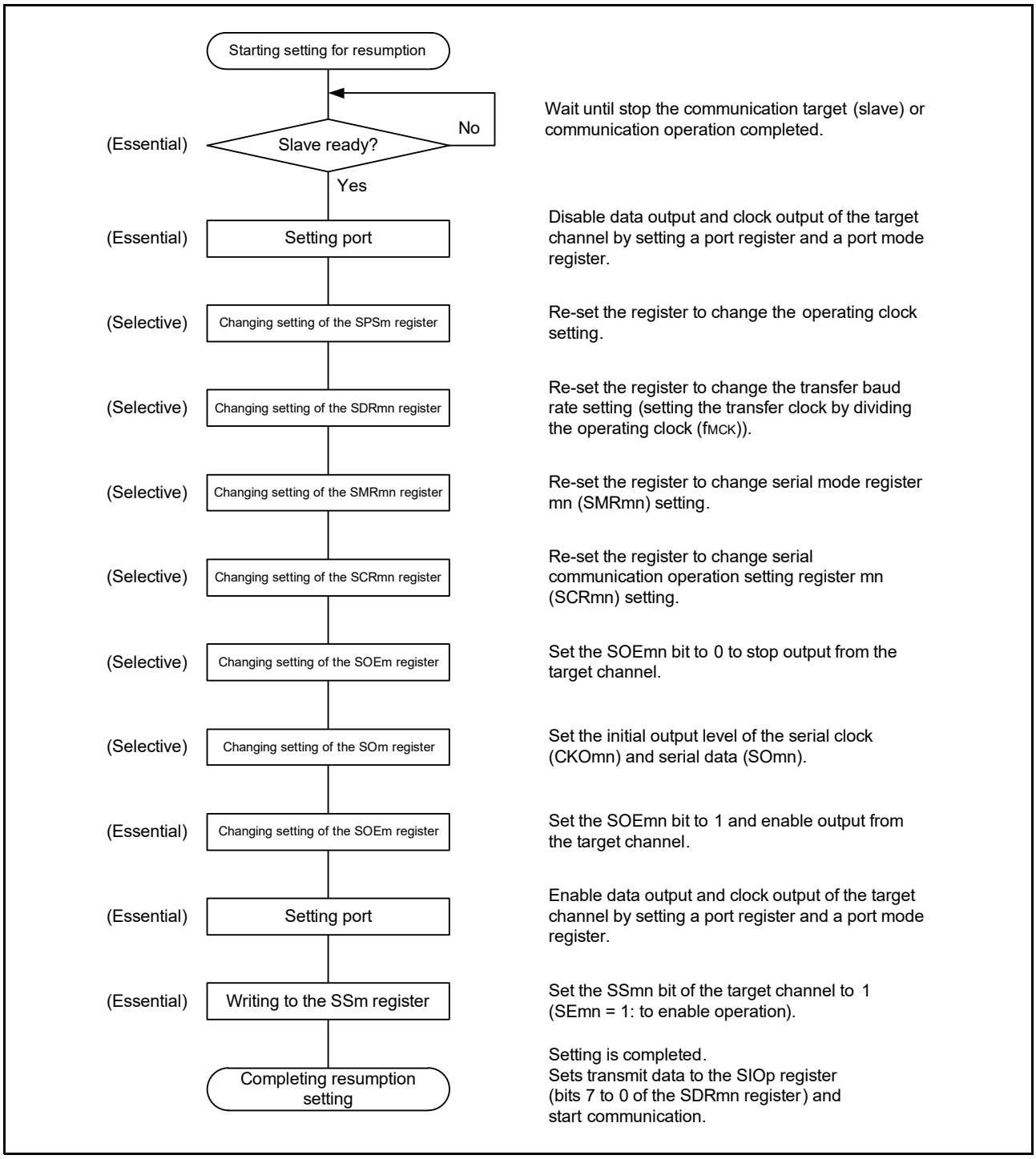


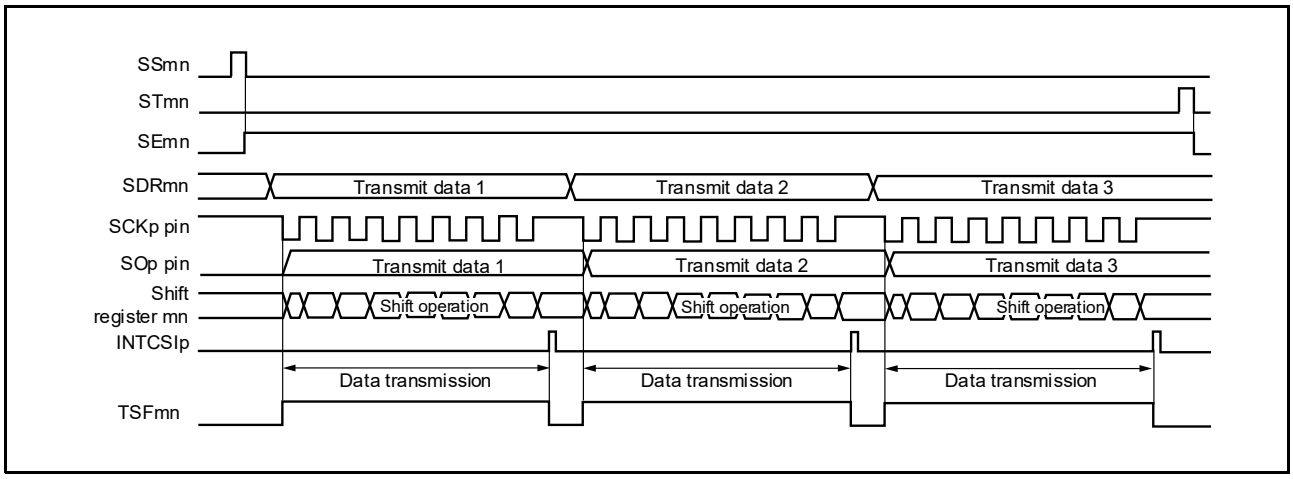
Figure 15 - 29 Procedure for Resuming Master Transmission



Remark If the PRR0 register is rewritten while stopping the communication to reset the serial array unit, wait until the communication target (slave) stops or communication finishes, and then perform initialization instead of restarting the communication.

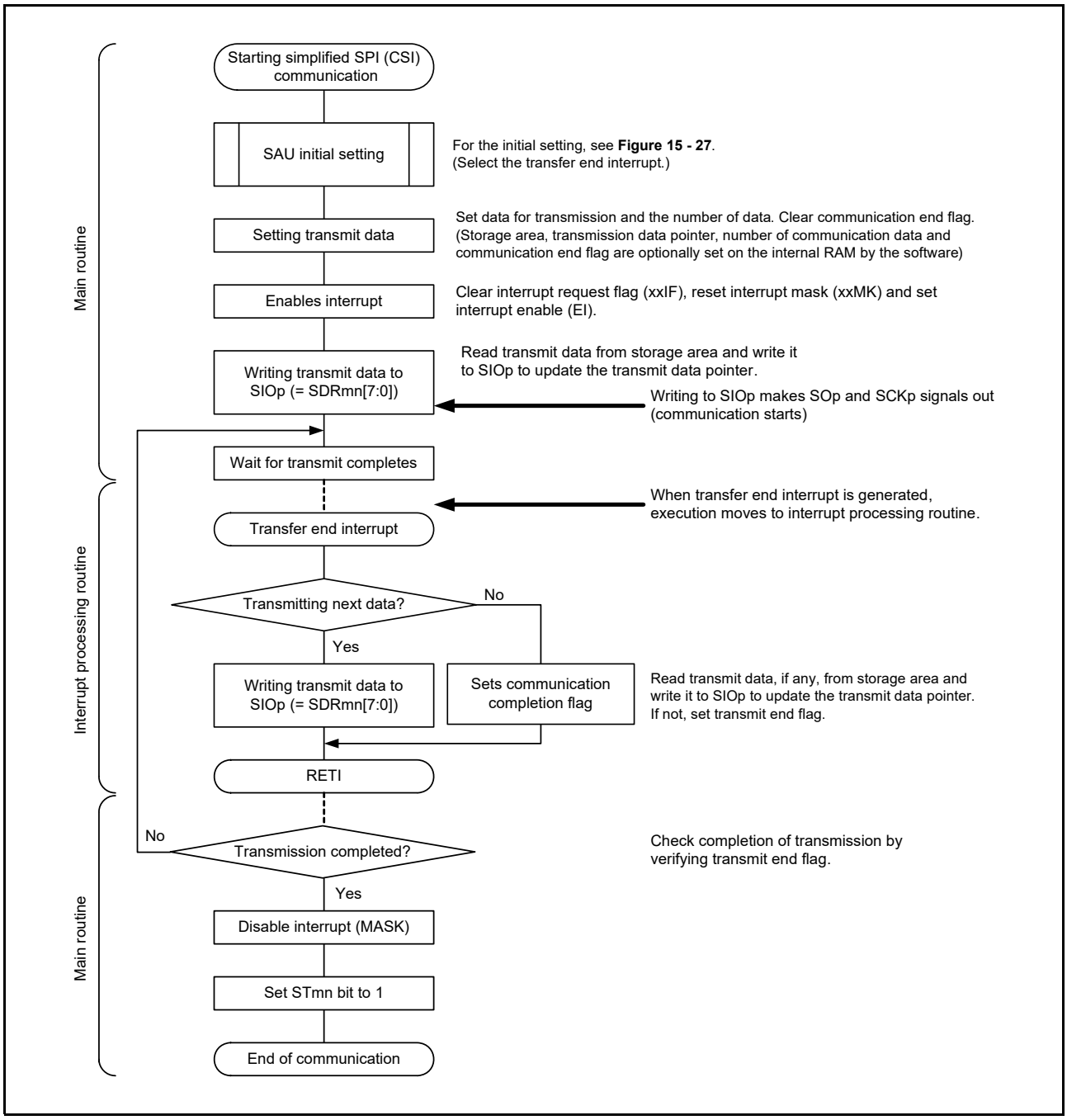
3. Processing flow (in single-transmission mode)

Figure 15 - 30 Timing Chart of Master Transmission (in Single-transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)



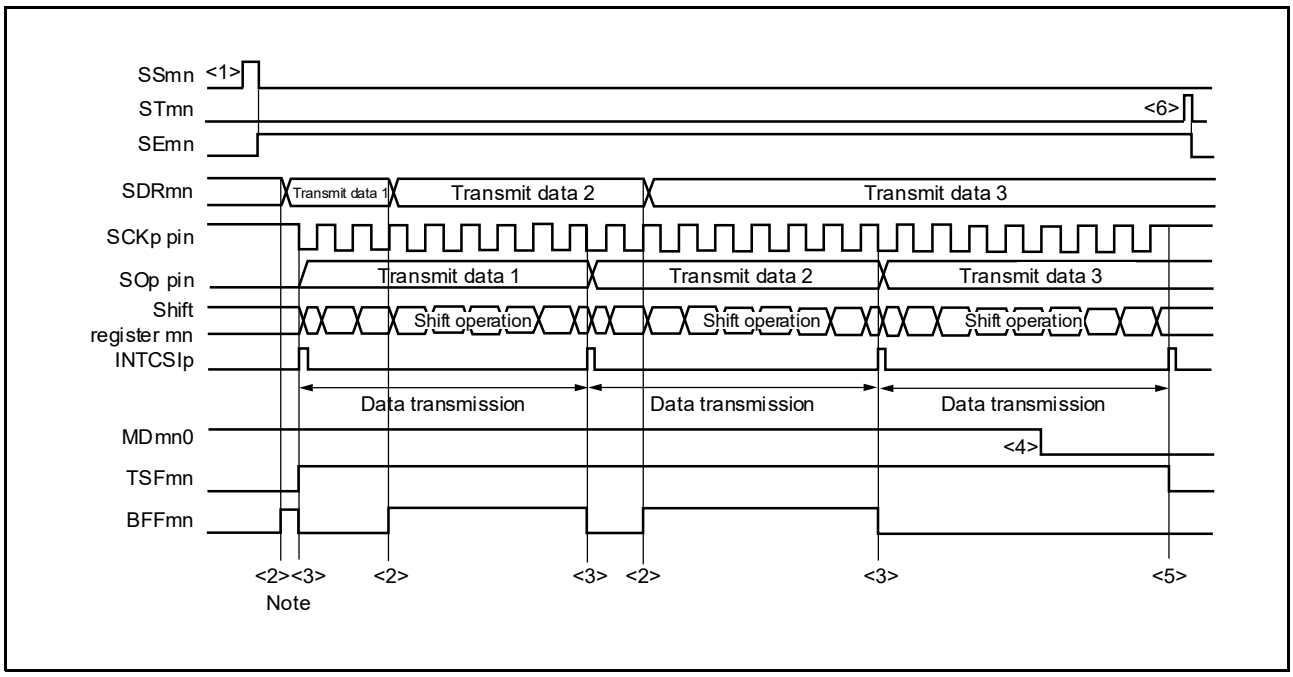
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Figure 15 - 31 Flowchart of Master Transmission (in Single-transmission Mode)



4. Processing flow (in continuous transmission mode)

Figure 15 - 32 Timing Chart of Master Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)

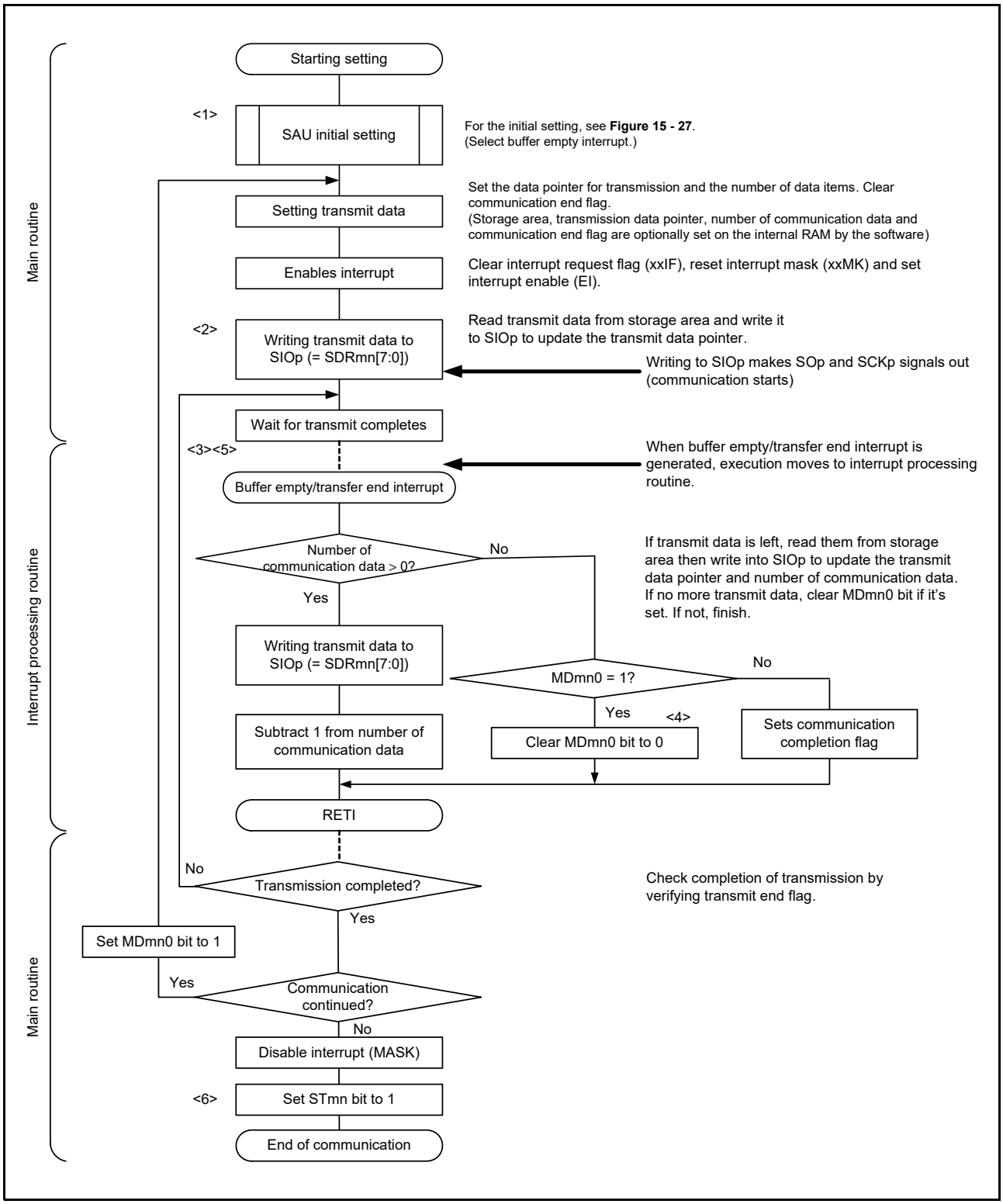


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution **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 has been rewritten before the transfer end interrupt of the last transmit data.**

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Figure 15 - 33 Flowchart of Master Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 15 - 32 Timing Chart of Master Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)**.

15.5.2 Master reception

Master reception is that this MCU outputs a transfer clock and receives data from another device.

Simplified SPI	CSI00	CSI01	CSI10	CSI11	CSI20	CSI21	CSI30	CSI31
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 2 of SAU1	Channel 3 of SAU1
Pins used	SCK00, SI00	SCK01, SI01	SCK10, SI10	SCK11, SI11	SCK20, SI20	SCK21, SI21	SCK30, SI30	SCK31, SI31
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21	INTCSI30	INTCSI31
	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 or 8 bits							
Transfer rate**Note**	Max. $f_{CLK}/2$ [Hz] (CSI00 only), $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK}: CPU/peripheral hardware clock frequency							
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data input starts from the start of the operation of the serial clock. • DAPmn = 1: Data input starts half a clock cycle before the start of the serial clock operation.							
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse							
Data direction	MSB or LSB first							

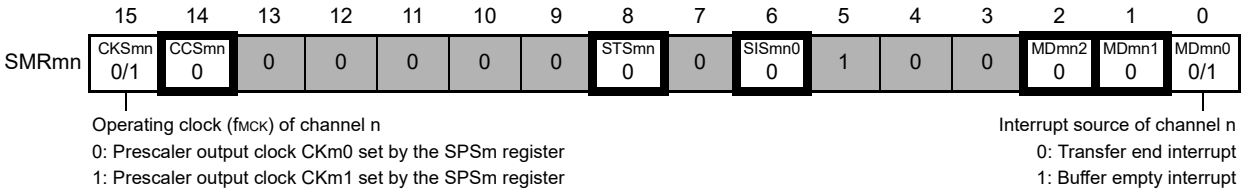
Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 37 Electrical Characteristics**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

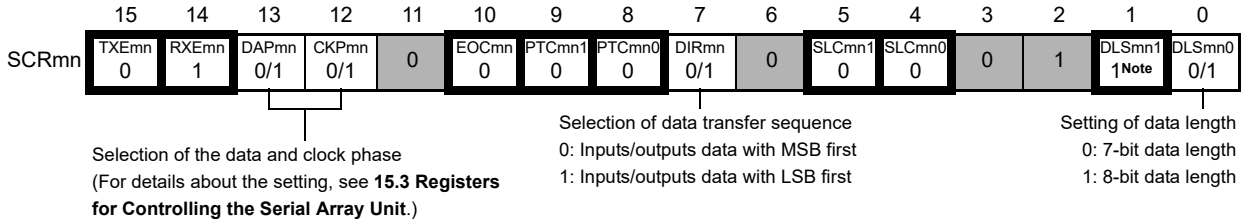
1. Register setting

Figure 15 - 34 Example of Contents of Registers for Master Reception of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31)

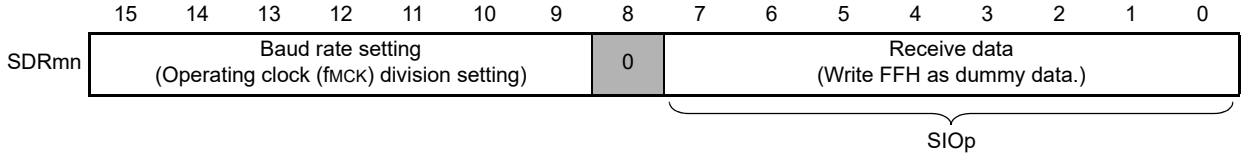
a) Serial mode register mn (SMRmn)



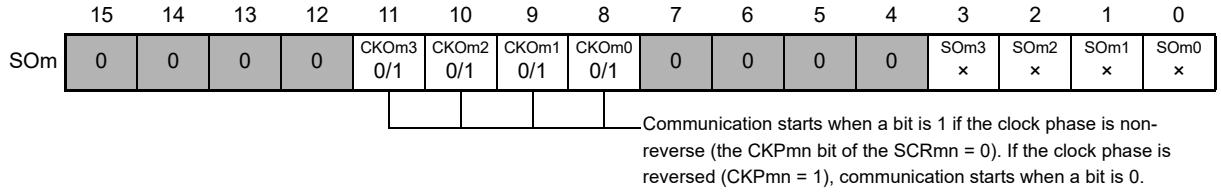
b) Serial communication operation setting register mn (SCRmn)



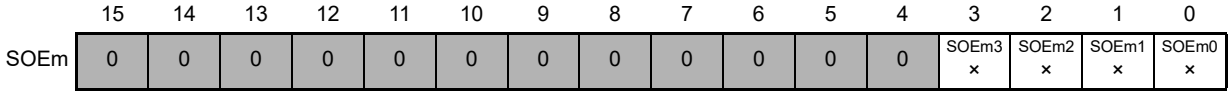
c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



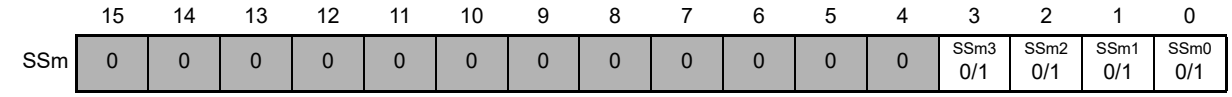
d) Serial output register m (SOm): Set only the bit of the target channel.



e) Serial output enable register m (SOEm): This register is not used in this mode.



f) Serial channel start register m (SSm): Set only the bit of the target channel to 1.



Note This bit is only present in the following registers; in other registers, it is fixed to 1.
- SCR00 and SCR01 registers
- SCR10 and SCR11 registers of the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products

(Remarks are listed on the next page.)

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Remark 2. ■: Setting is fixed in the simplified SPI (CSI) master reception mode, ■: Set to the initial value.
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the user's application.

2. Operation procedure

Figure 15 - 35 Initial Setting Procedure for Master Reception

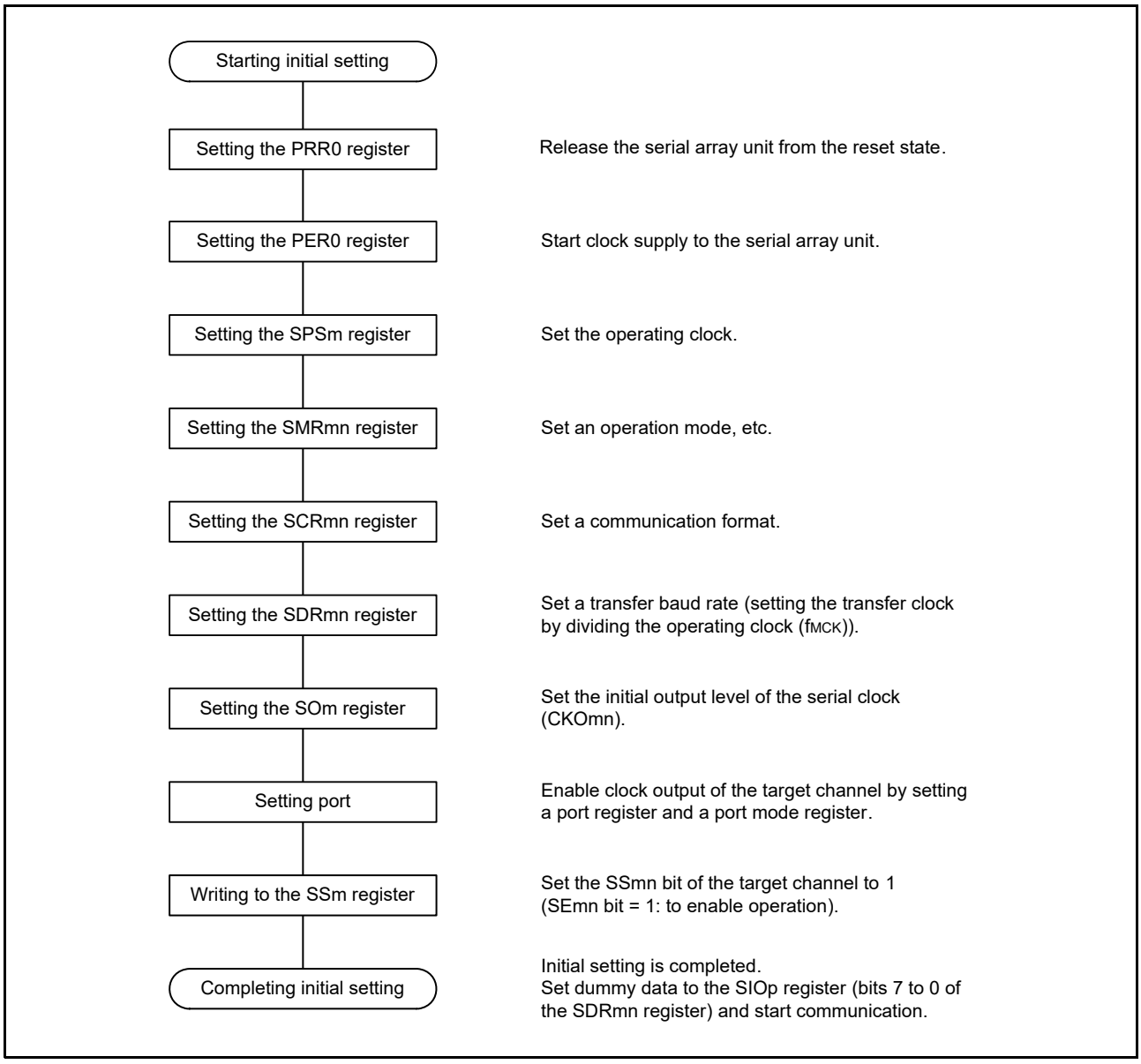


Figure 15 - 36 Procedure for Stopping Master Reception

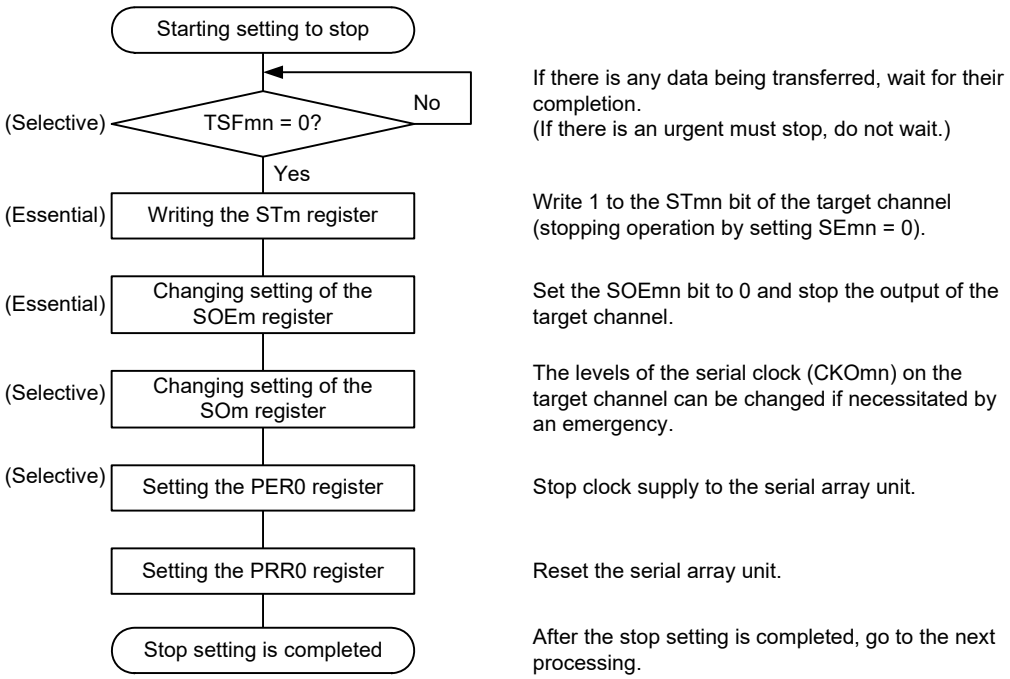
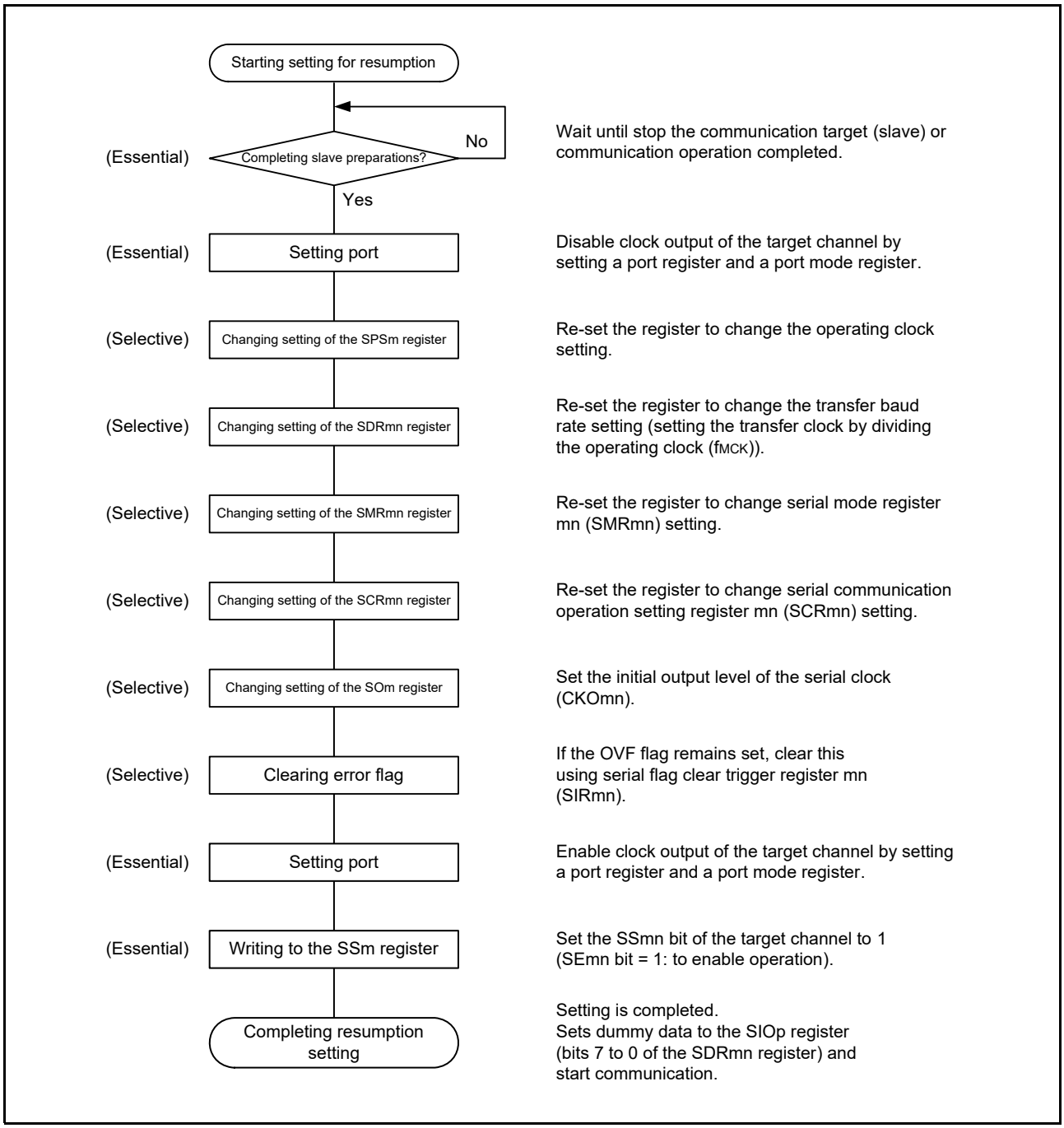


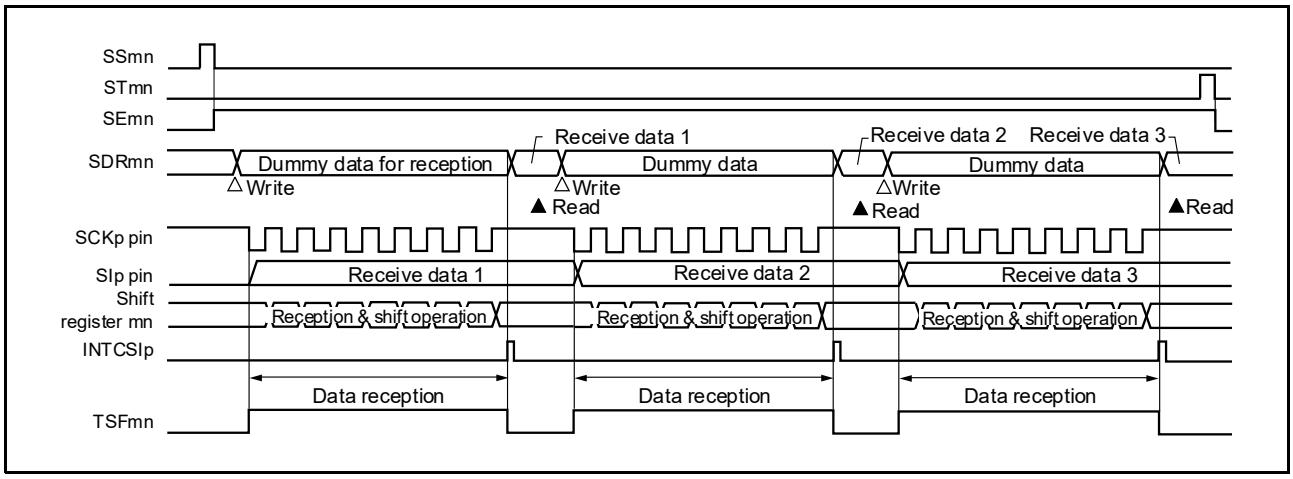
Figure 15 - 37 Procedure for Resuming Master Reception



Remark If the PRR0 register is rewritten while stopping the communication to reset the serial array unit, wait until the communication target (slave) stops or communication finishes, and then perform initialization instead of restarting the communication.

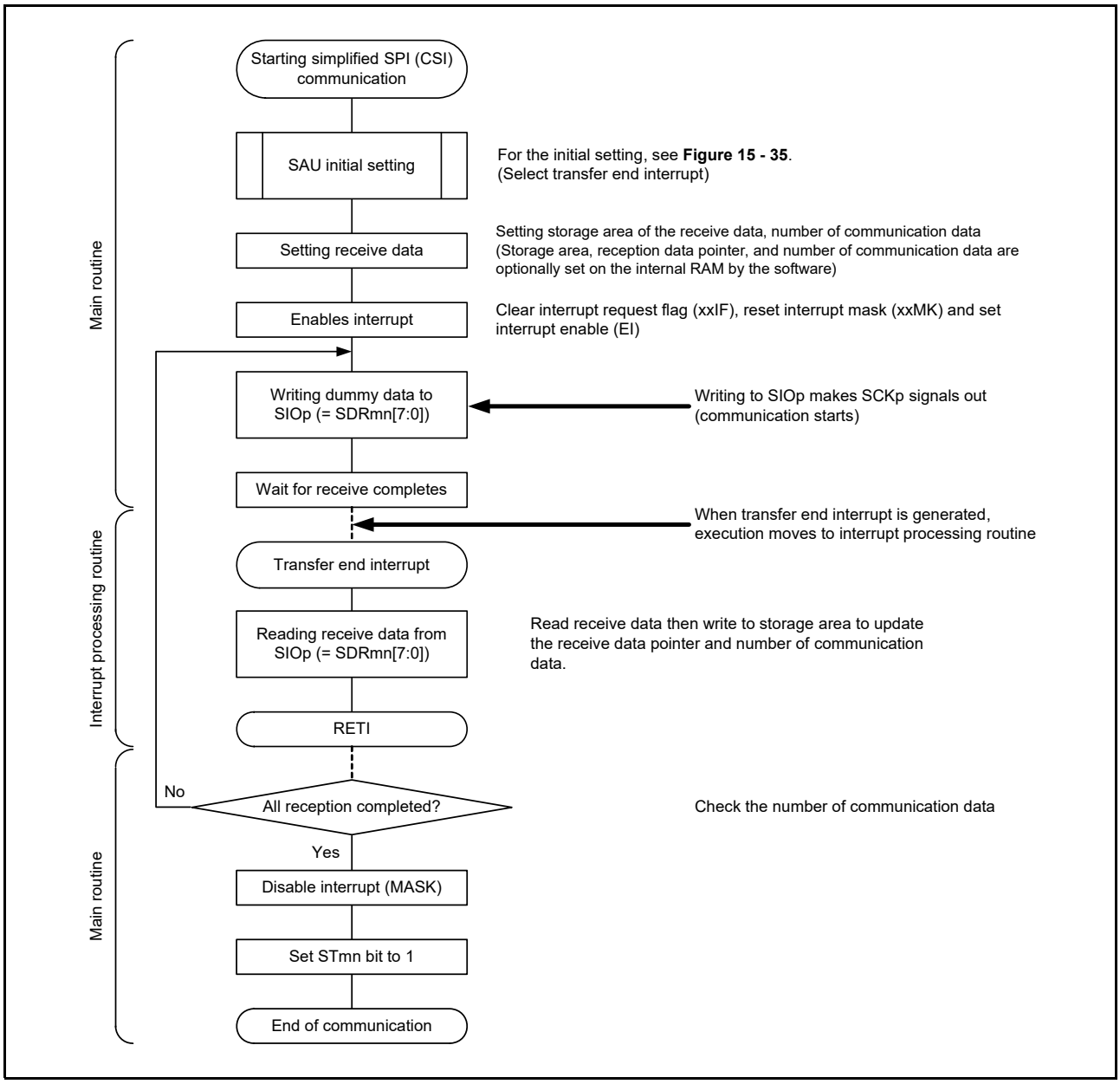
3. Processing flow (in single-reception mode)

Figure 15 - 38 Timing Chart of Master Reception (in Single-reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



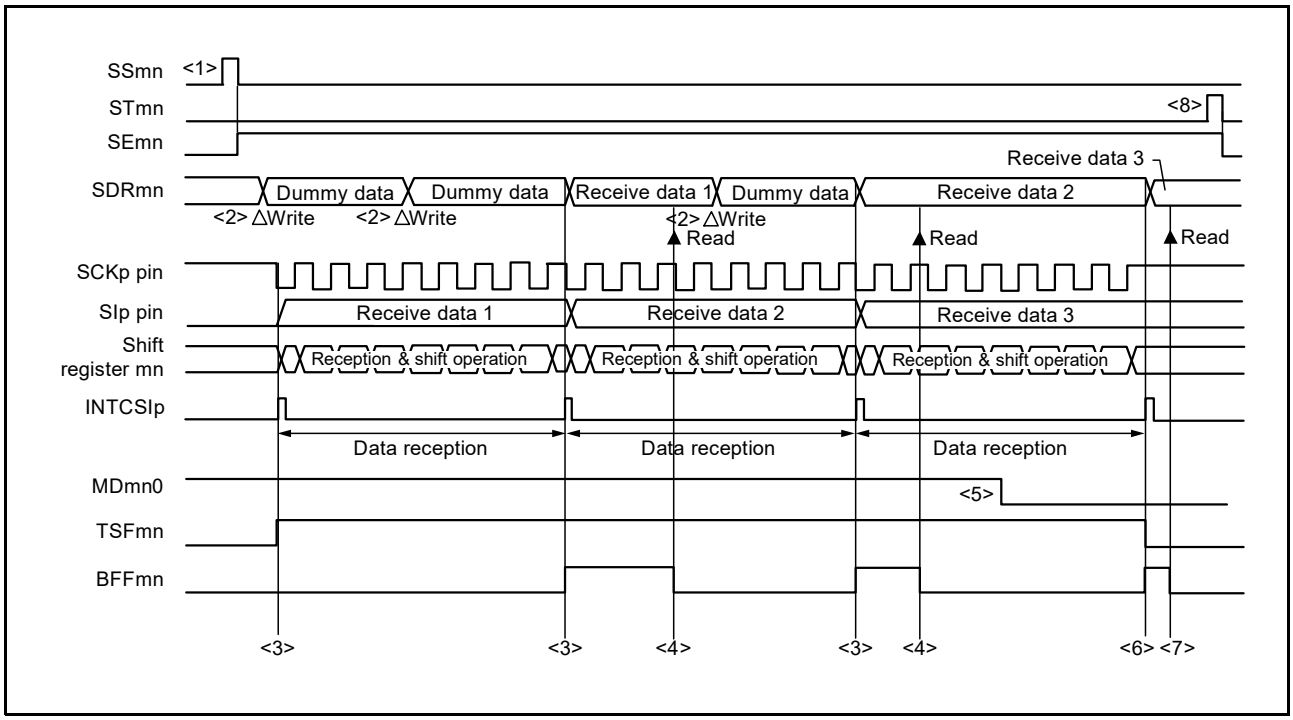
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Figure 15 - 39 Flowchart of Master Reception (in Single-reception Mode)



4. Processing flow (in continuous reception mode)

Figure 15 - 40 Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)

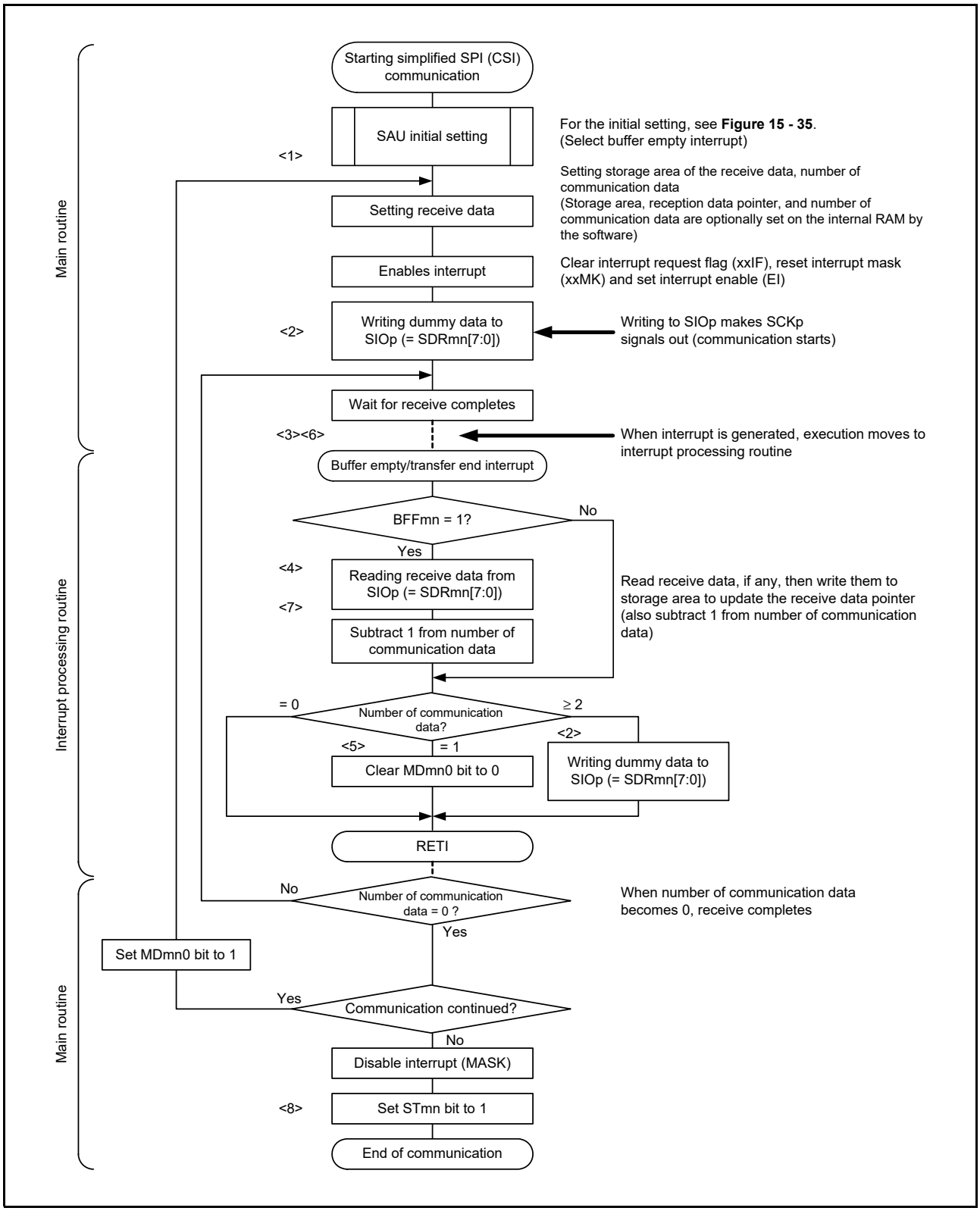


Caution **The MDmn0 bit can be rewritten even during operation. However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.**

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 15 - 41 Flowchart of Master Reception (in Continuous Reception Mode)**.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Figure 15 - 41 Flowchart of Master Reception (in Continuous Reception Mode)



Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 15 - 40 Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)**.

15.5.3 Master transmission/reception

Master transmission/reception is that this MCU outputs a transfer clock and transmits/receives data to/from other device.

Simplified SPI	CSI00	CSI01	CSI10	CSI11	CSI20	CSI21	CSI30	CSI31
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 2 of SAU1	Channel 3 of SAU1
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01	SCK10, SI10, SO10	SCK11, SI11, SO11	SCK20, SI20, SO20	SCK21, SI21, SO21	SCK30, SI30, SO30	SCK31, SI31, SO31
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21	INTCSI30	INTCSI31
	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 or 8 bits							
Transfer rate**Note**	Max. $f_{CLK}/2$ [Hz] (CSI00 only), $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK}: CPU/peripheral hardware clock frequency							
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data I/O starts at the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock cycle before the start of the serial clock operation.							
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse							
Data direction	MSB or LSB first							

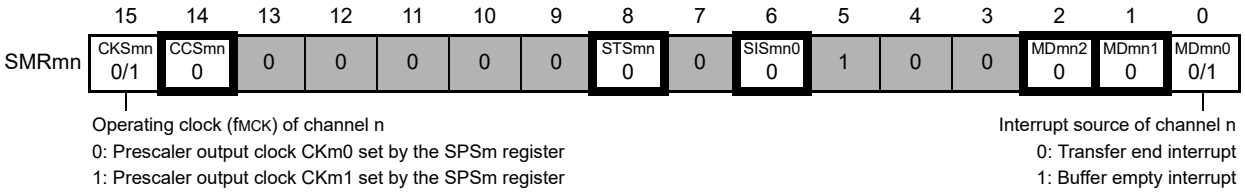
Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 37 Electrical Characteristics**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

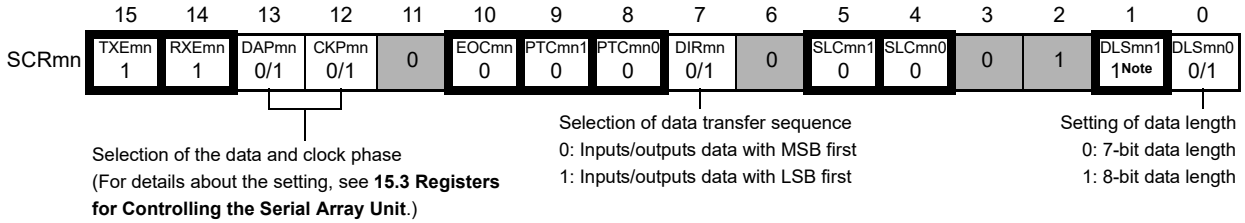
1. Register setting

Figure 15 - 42 Example of Contents of Registers for Master Transmission/Reception of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31)

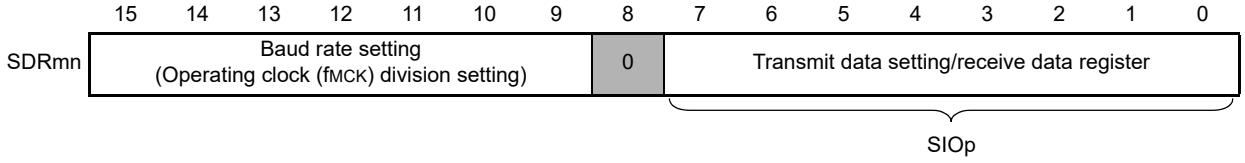
a) Serial mode register mn (SMRmn)



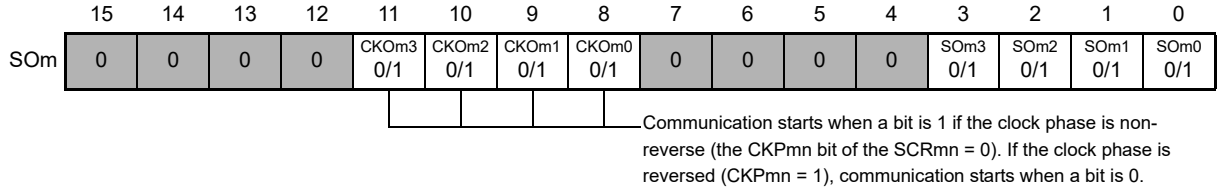
b) Serial communication operation setting register mn (SCRmn)



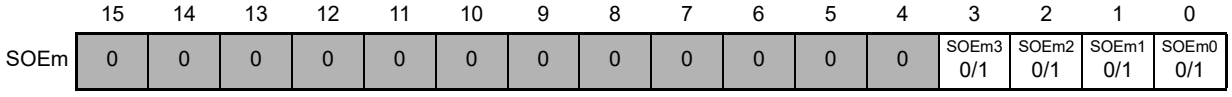
c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



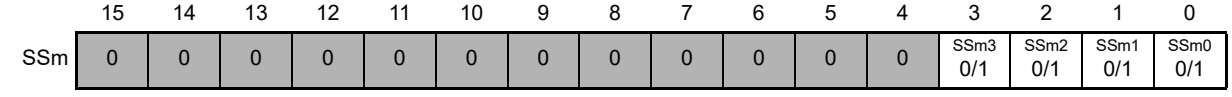
d) Serial output register m (SOm): Set only the bit of the target channel.



e) Serial output enable register m (SOEm): Set only the bit of the target channel to 1.



f) Serial channel start register m (SSm): Set only the bit of the target channel to 1.



Note This bit is only present in the following registers; in other registers, it is fixed to 1.
- SCR00 and SCR01 registers
- SCR10 and SCR11 registers of the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products

(Remarks are listed on the next page.)

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Remark 2. ■: Setting is fixed in the simplified SPI (CSI) master transmission/reception mode, ■: Set to the initial value. 0/1: Set to 0 or 1 depending on the user's application.

2. Operation procedure

Figure 15 - 43 Initial Setting Procedure for Master Transmission/Reception

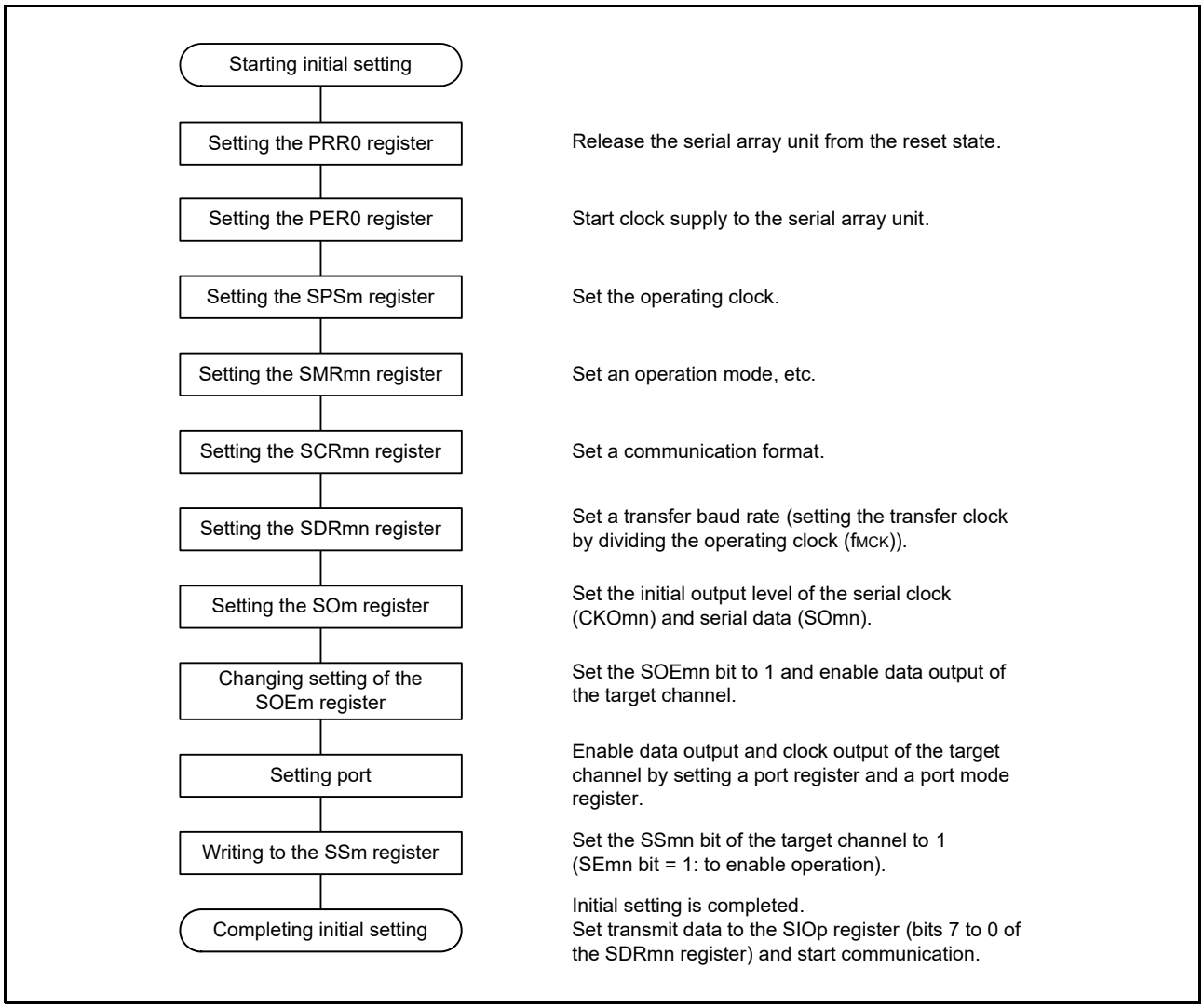


Figure 15 - 44 Procedure for Stopping Master Transmission/Reception

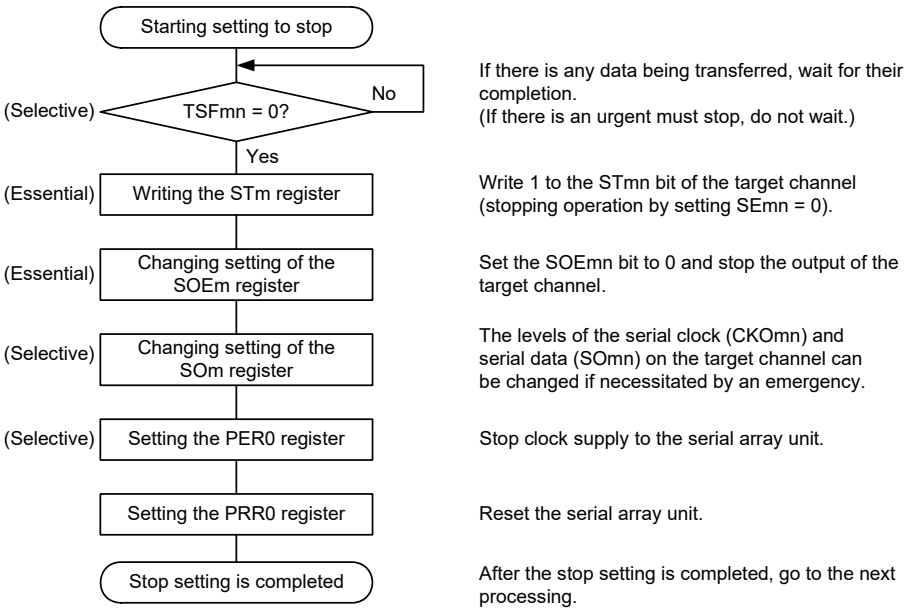
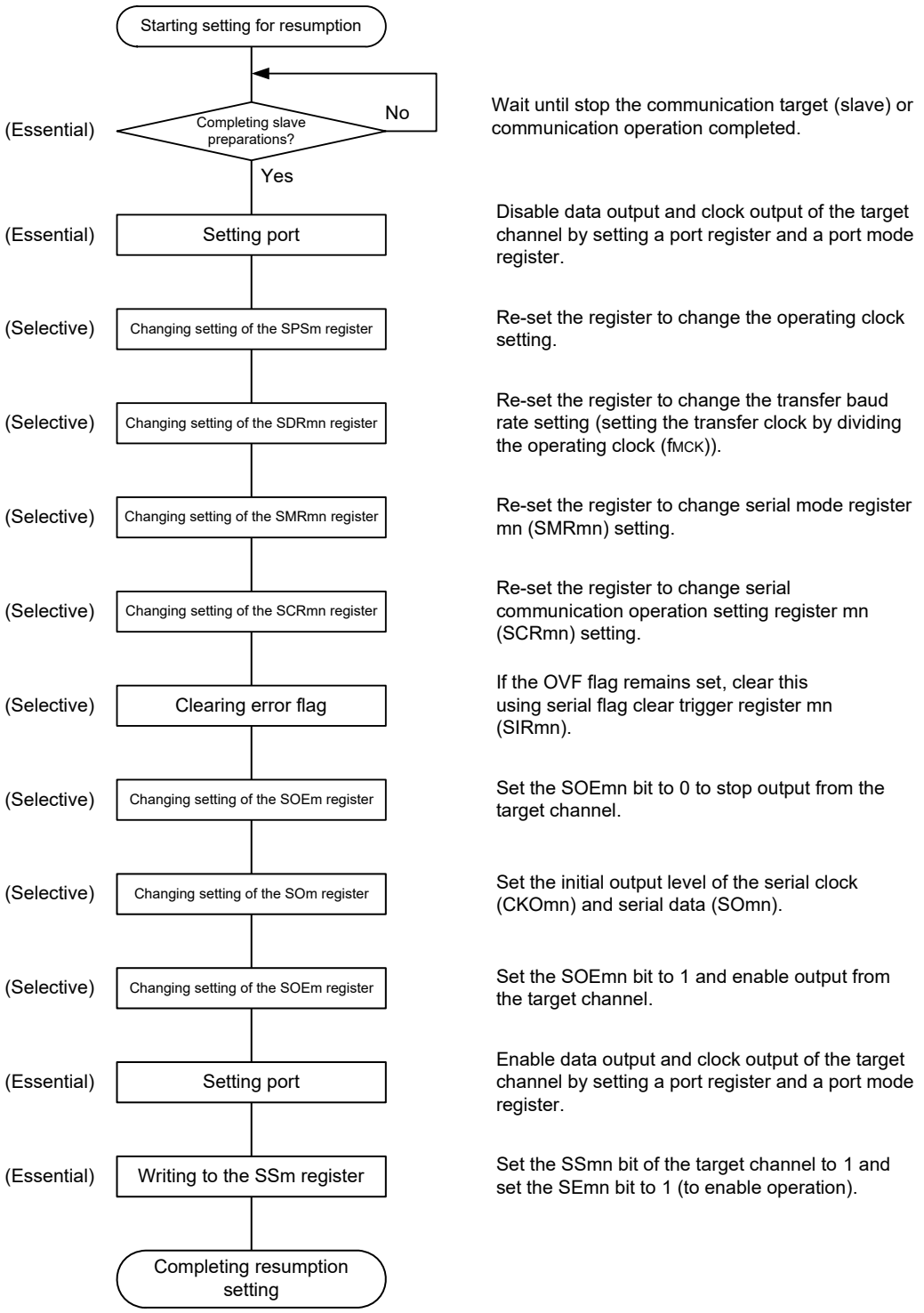
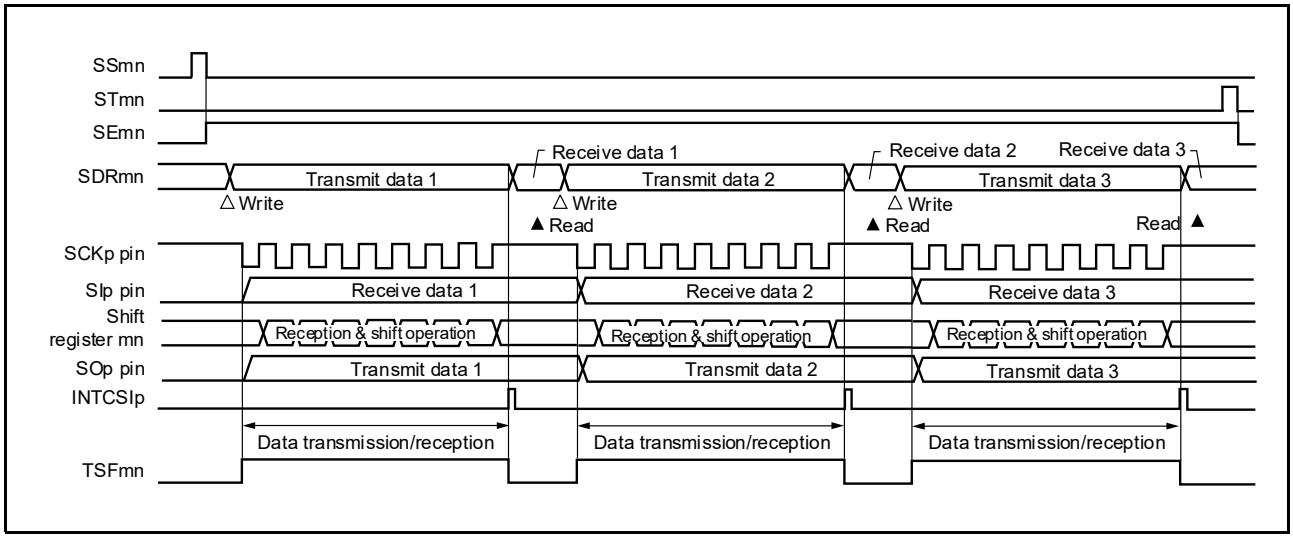


Figure 15 - 45 Procedure for Resuming Master Transmission/Reception



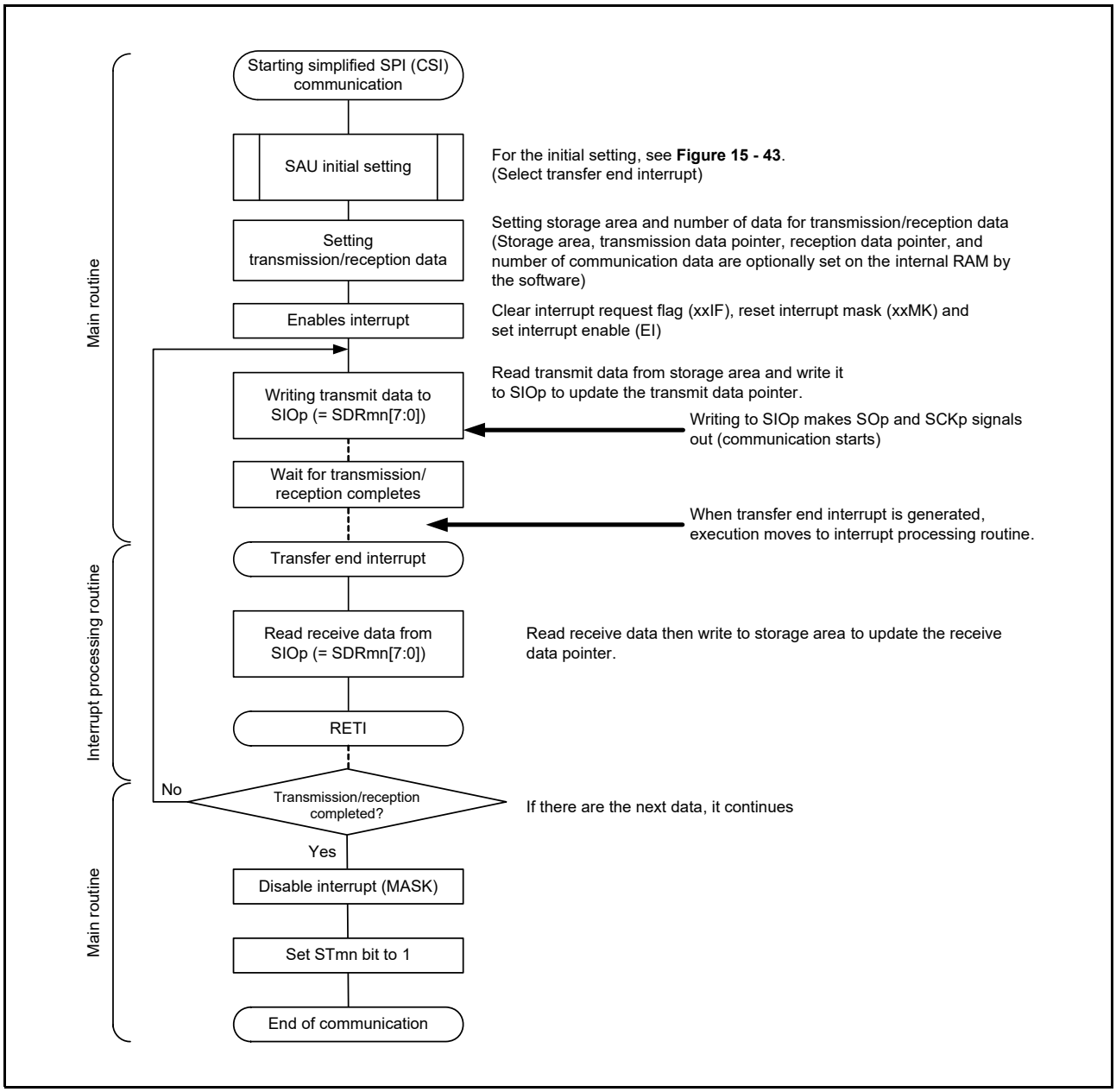
3. Processing flow (in single-transmission/reception mode)

Figure 15 - 46 Timing Chart of Master Transmission/Reception (in Single-transmission/reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



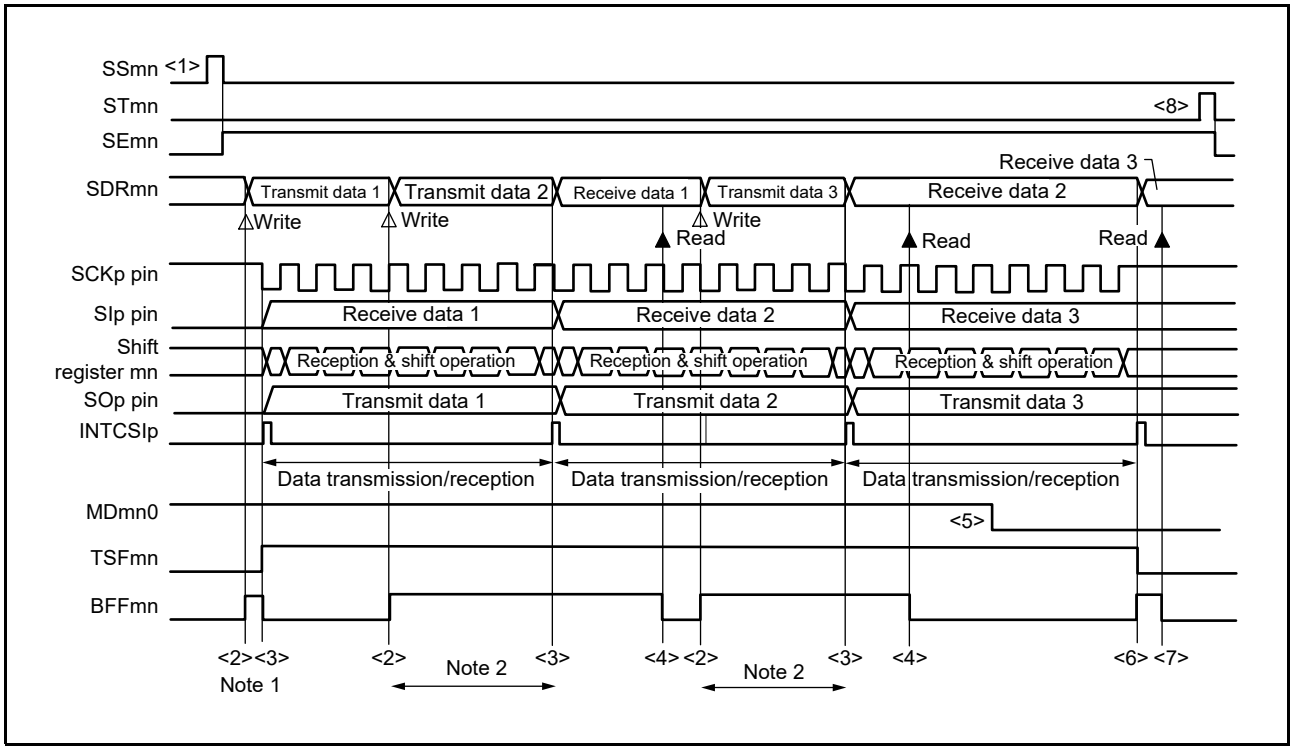
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Figure 15 - 47 Flowchart of Master Transmission/Reception (in Single-transmission/reception Mode)



4. Processing flow (in continuous transmission/reception mode)

Figure 15 - 48 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Note 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.

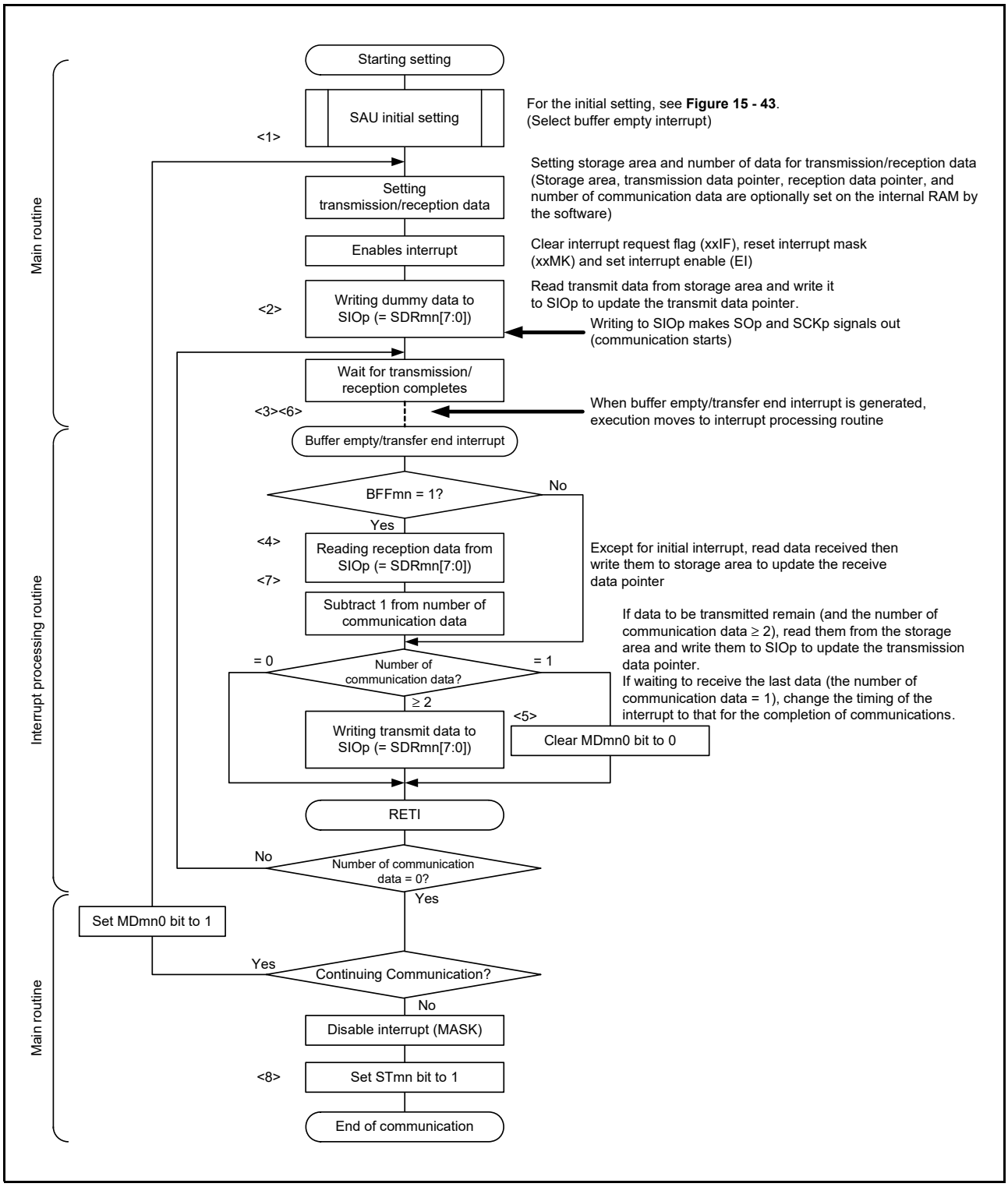
Note 2. The transmit data can be read by reading the SDRmn register during this period. Reading this register does not affect the transfer operation.

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 has been rewritten before the transfer end interrupt of the last transmit data.**

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 15 - 49 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Figure 15 - 49 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 15 - 48 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)**.

15.5.4 Slave transmission

Slave transmission is that this MCU transmits data to another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00	CSI01	CSI10	CSI11	CSI20	CSI21	CSI30	CSI31
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 2 of SAU1	Channel 3 of SAU1
Pins used	SCK00, SO00	SCK01, SO01	SCK10, SO10	SCK11, SO11	SCK20, SO20	SCK21, SO21	SCK30, SO30	SCK31, SO31
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21	INTCSI30	INTCSI31
	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 or 8 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 operation of the serial clock. • DAPmn = 1: Data output starts half a clock cycle before the start of the serial clock operation.							
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse							
Data direction	MSB or LSB first							

Note 1. Because the external serial clock input to the SCK00, SCK01, SCK10, SCK11, SCK20, SCK21, SCK30, and SCK31 pins is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz].

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 37 Electrical Characteristics**.

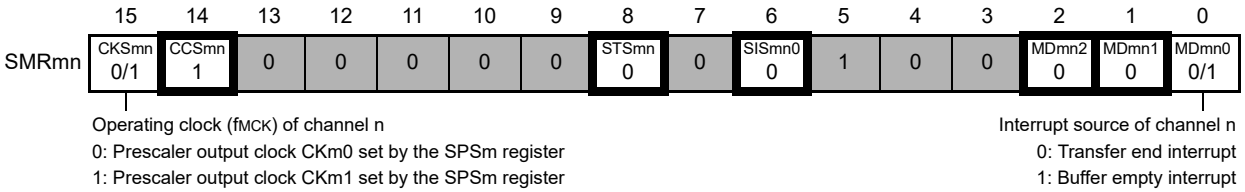
Remark 1. f_{MCK}: Operating clock frequency of target channel
f_{SCK}: Serial clock frequency

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

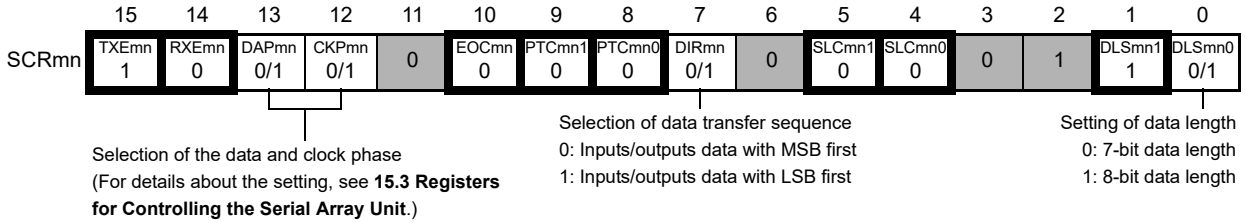
1. Register setting

Figure 15 - 50 Example of Contents of Registers for Slave Transmission of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31)

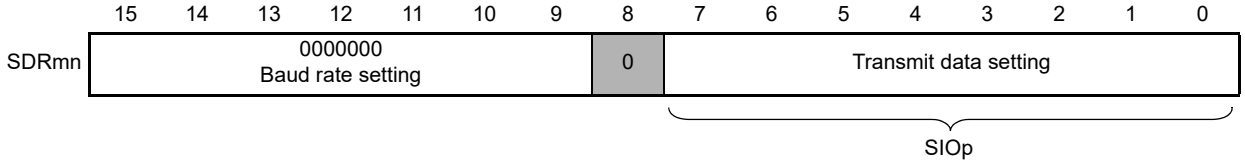
a) Serial mode register mn (SMRmn)



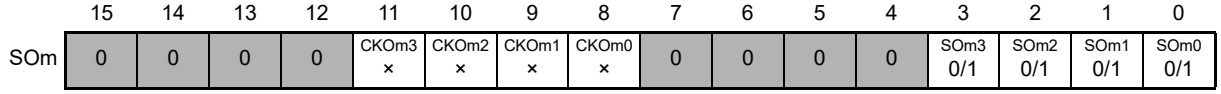
b) Serial communication operation setting register mn (SCRmn)



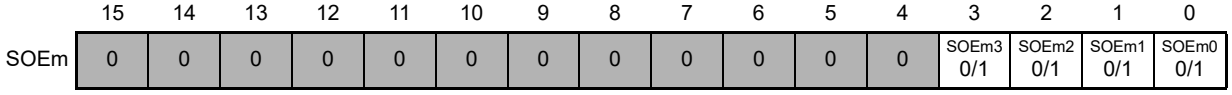
c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



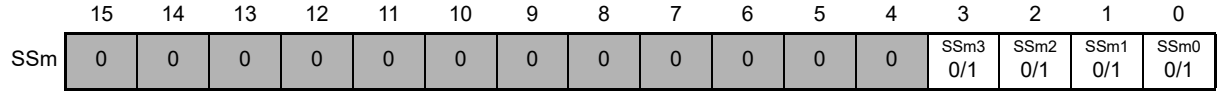
d) Serial output register m (SOm): Set only the bit of the target channel.



e) Serial output enable register m (SOEm): Set only the bit of the target channel to 1.



f) Serial channel start register m (SSm): Set only the bit of the target channel to 1.



Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Remark 2. ■: Setting is fixed in the simplified SPI (CSI) slave transmission mode, ■: Set to the initial value.
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the user's application.

2. Operation procedure

Figure 15 - 51 Initial Setting Procedure for Slave Transmission

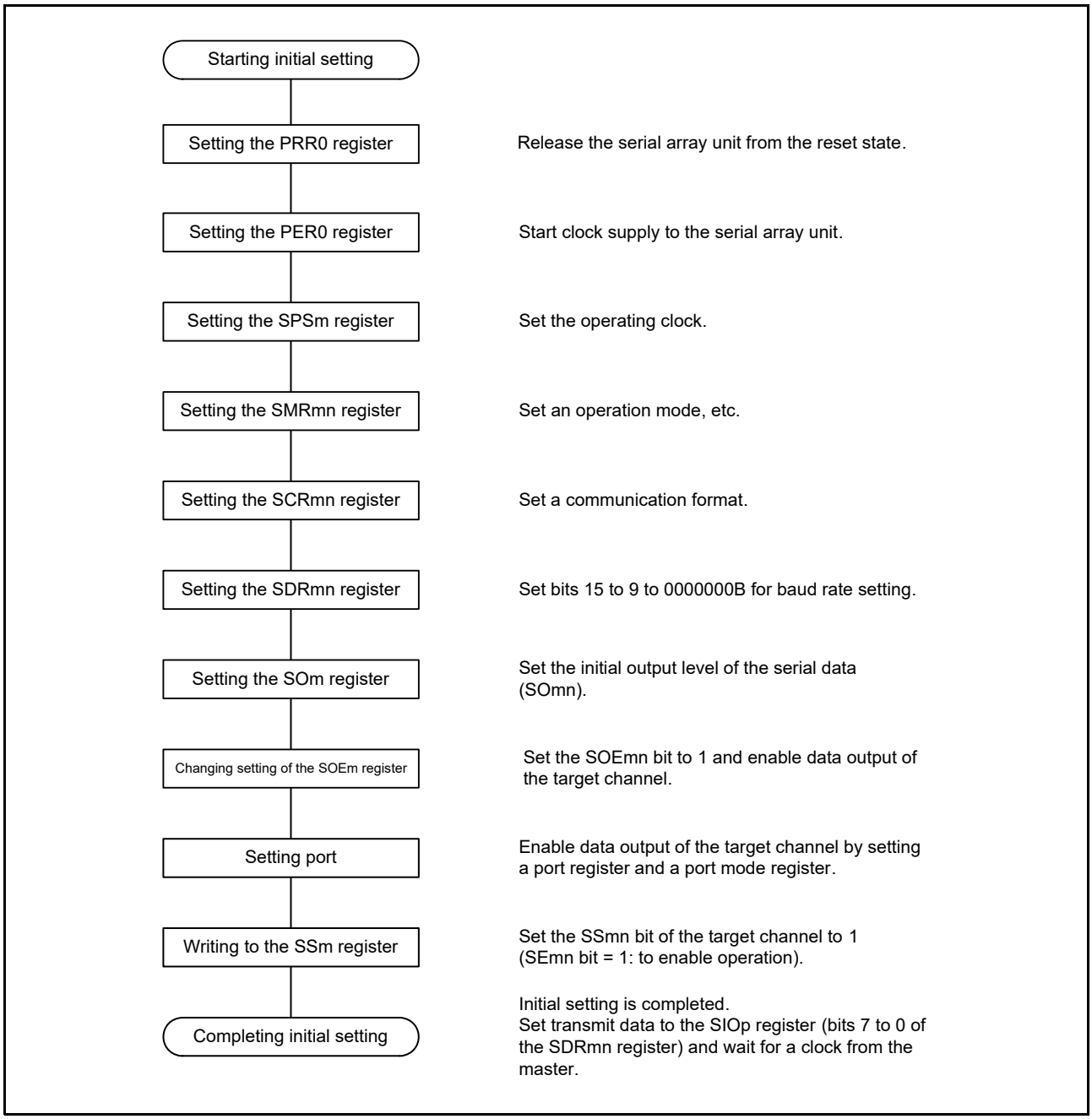


Figure 15 - 52 Procedure for Stopping Slave Transmission

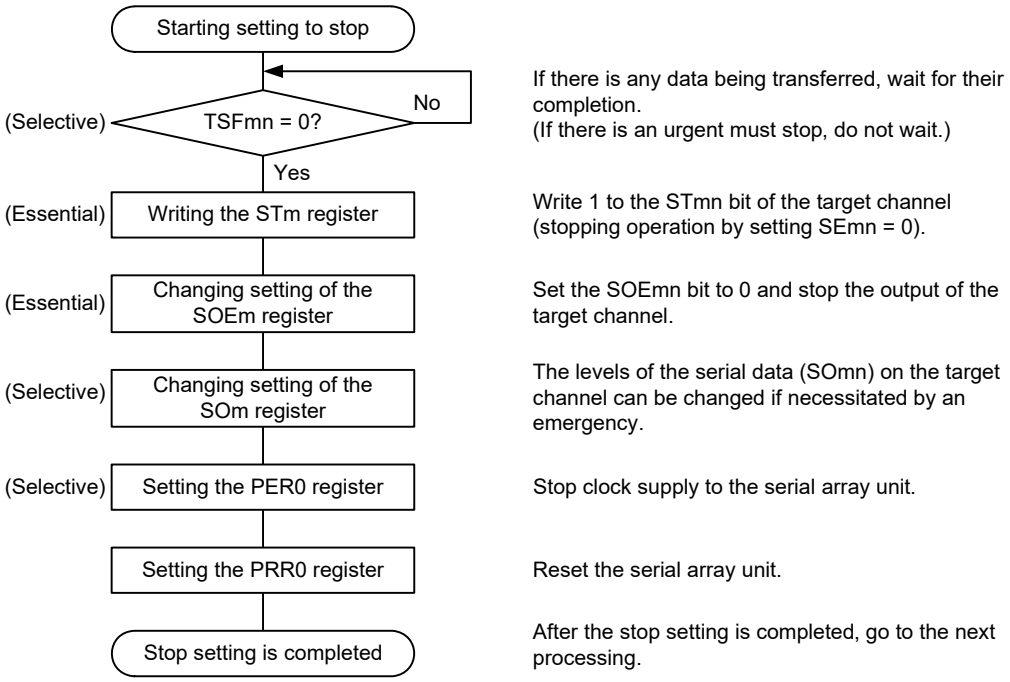
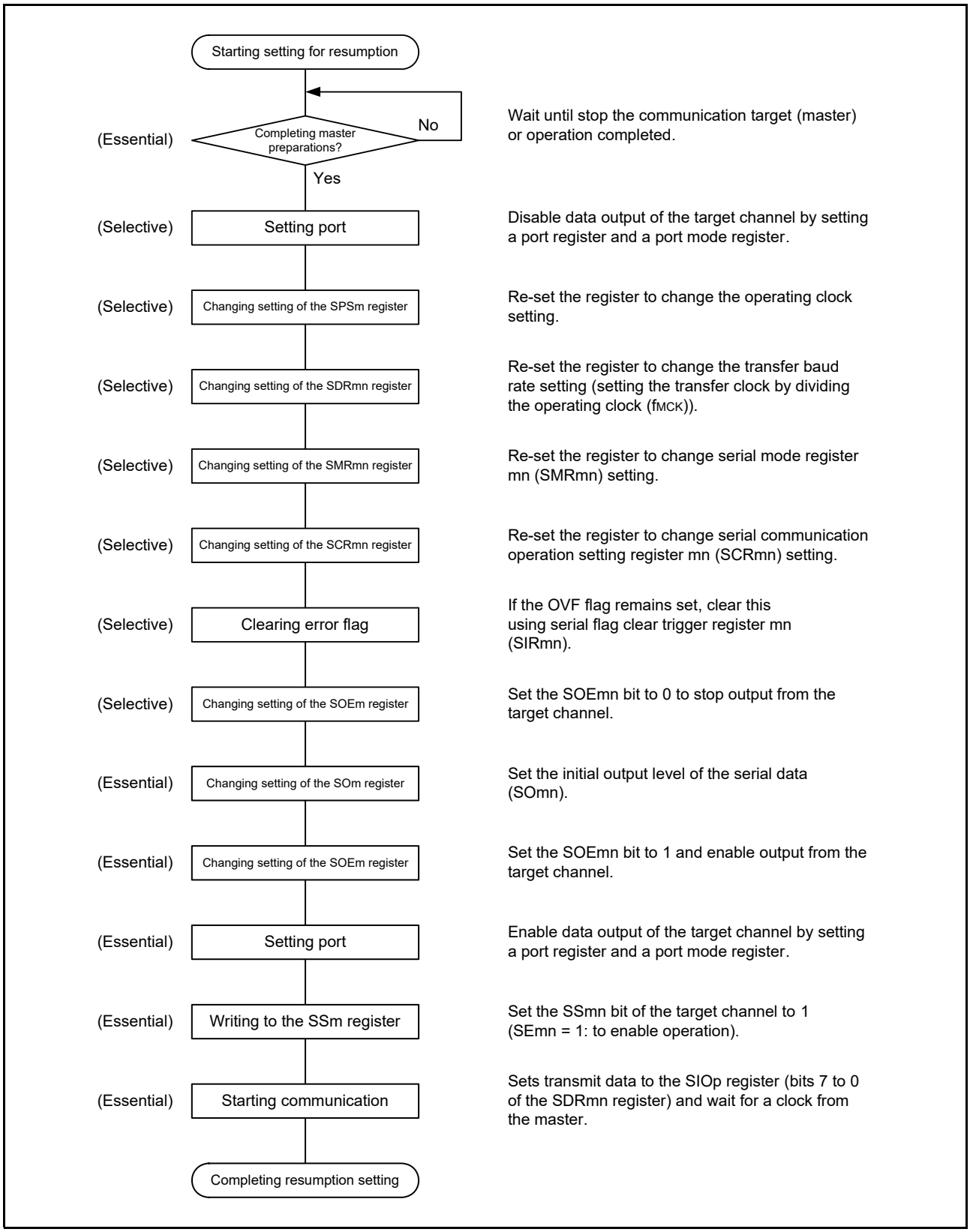


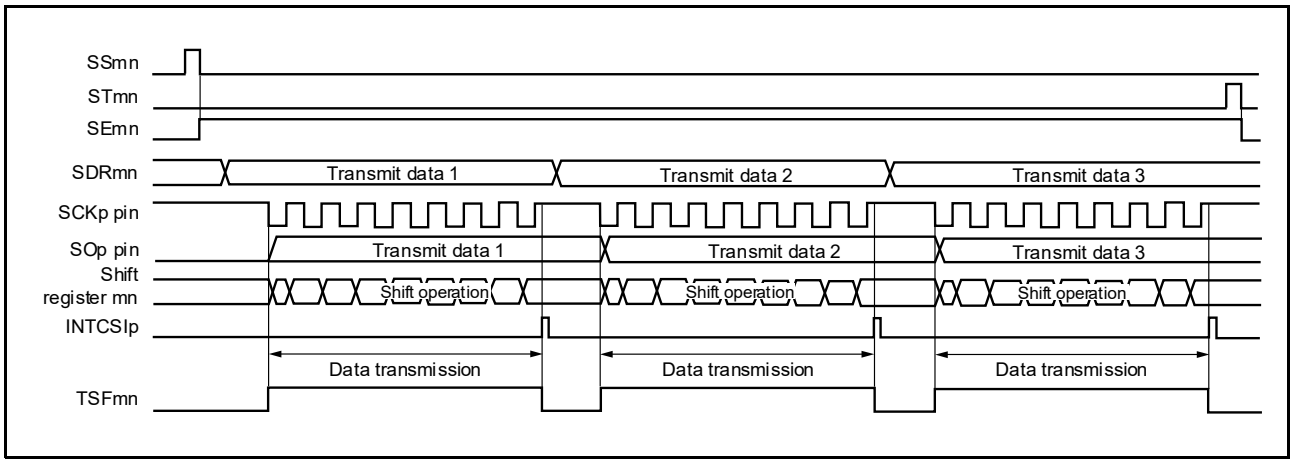
Figure 15 - 53 Procedure for Resuming Slave Transmission



Remark If the PRR0 register is rewritten while stopping the communication to reset the serial array unit, wait until the communication target (master) stops or communication finishes, and then perform initialization instead of restarting the communication.

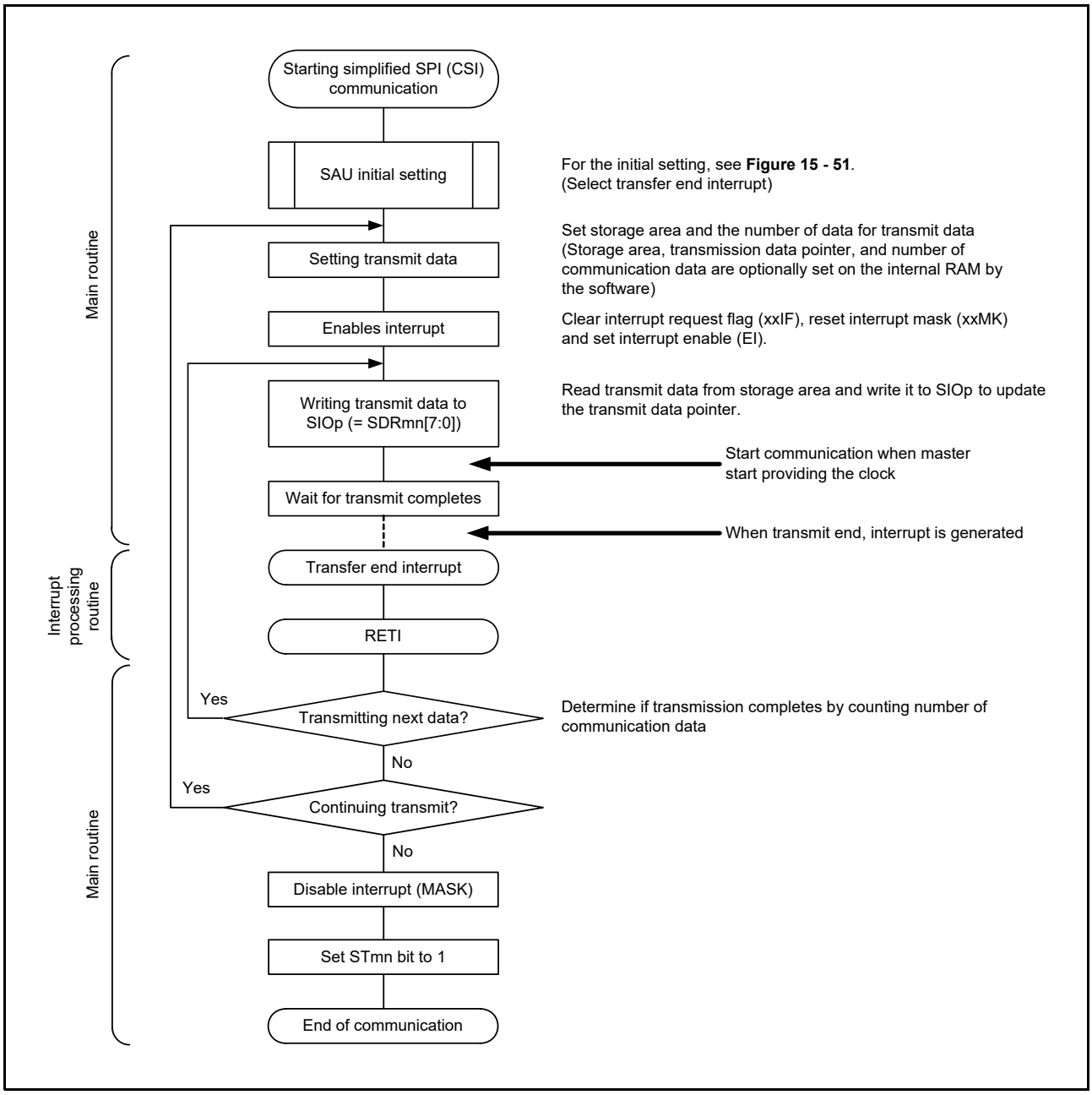
3. Processing flow (in single-transmission mode)

Figure 15 - 54 Timing Chart of Slave Transmission (in Single-transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)



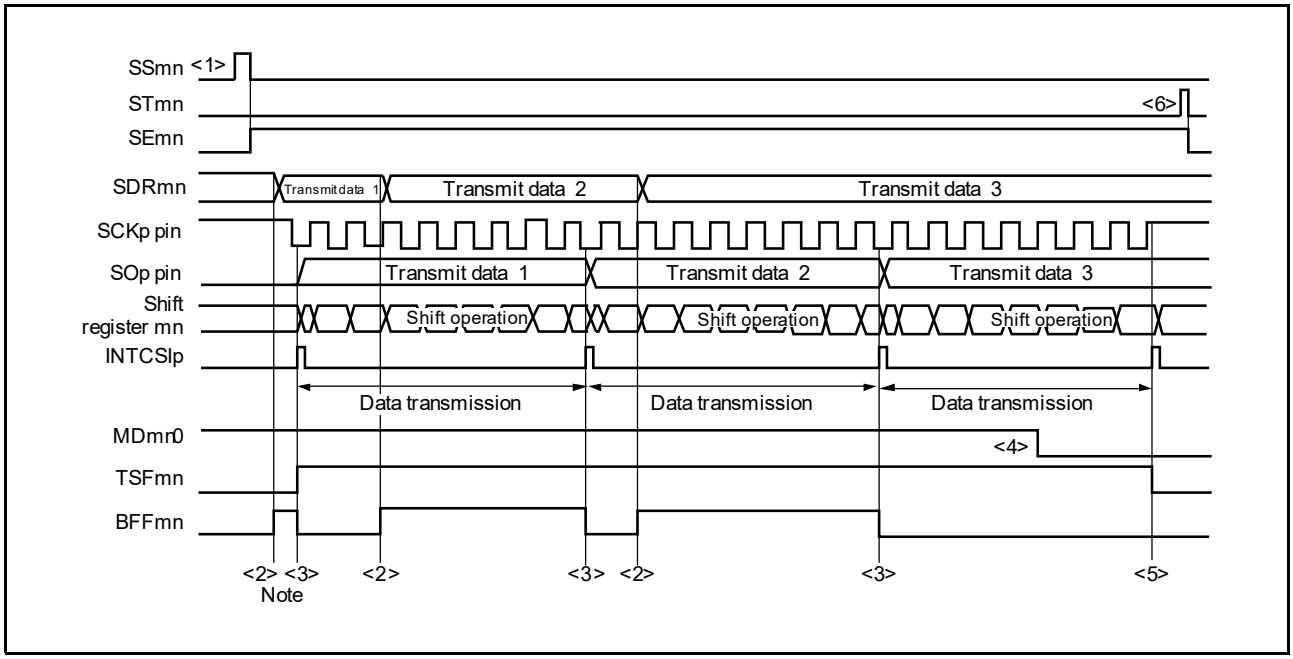
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Figure 15 - 55 Flowchart of Slave Transmission (in Single-transmission Mode)



4. Processing flow (in continuous transmission mode)

Figure 15 - 56 Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)

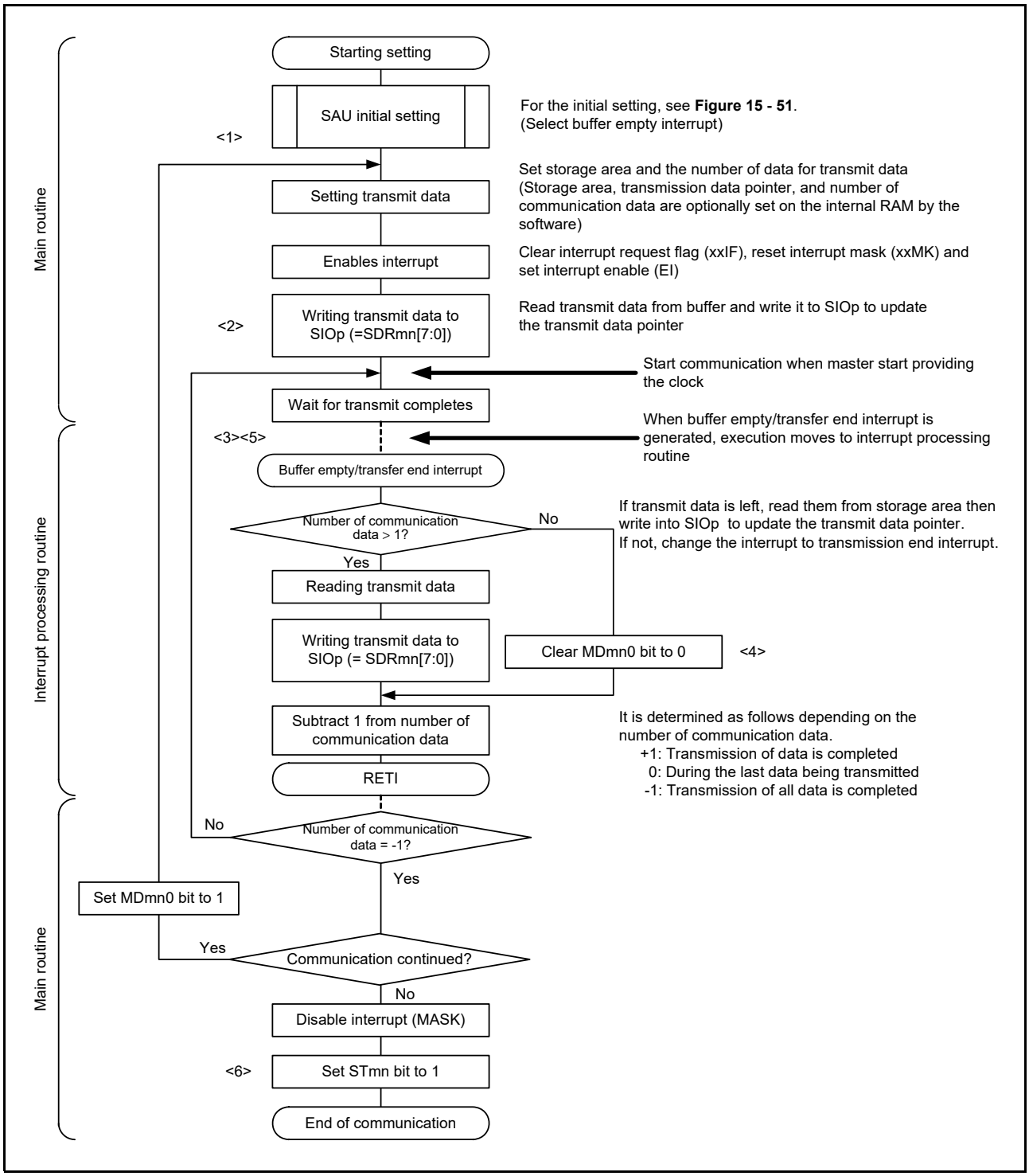


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution **The MDmn0 bit of serial mode register mn (SSRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.**

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Figure 15 - 57 Flowchart of Slave Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 15 - 56 Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)**.

15.5.5 Slave reception

Slave reception is that this MCU receives data from another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00	CSI01	CSI10	CSI11	CSI20	CSI21	CSI30	CSI31
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 2 of SAU1	Channel 3 of SAU1
Pins used	SCK00, SI00	SCK01, SI01	SCK10, SI10	SCK11, SI11	SCK20, SI20	SCK21, SI21	SCK30, SI30	SCK31, SI31
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21	INTCSI30	INTCSI31
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)							
Error detection flag	Overrun error detection flag (OVFmn) only							
Transfer data length	7 or 8 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 operation of the serial clock. • DAPmn = 1: Data input starts half a clock cycle before the start of the serial clock operation.							
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse							
Data direction	MSB or LSB first							

Note 1. Because the external serial clock input to the SCK00, SCK01, SCK10, SCK11, SCK20, SCK21, SCK30, and SCK31 pins is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz].

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 37 Electrical Characteristics**.

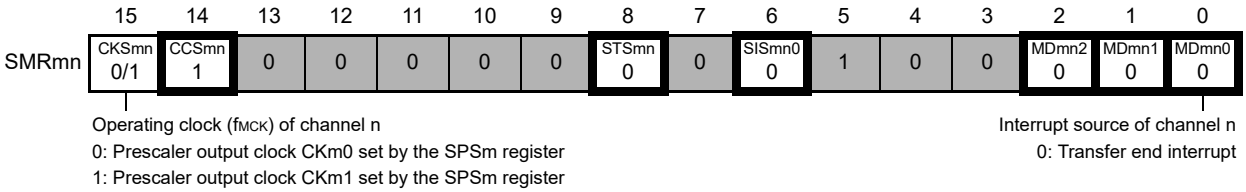
Remark 1. f_{MCK}: Operating clock frequency of target channel
f_{SCK}: Serial clock frequency

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

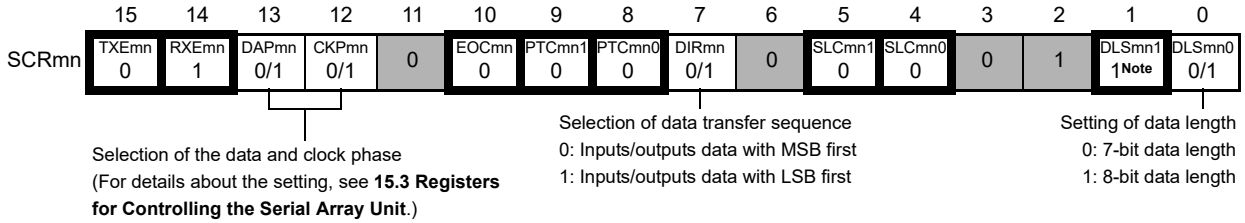
1. Register setting

Figure 15 - 58 Example of Contents of Registers for Slave Reception of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31)

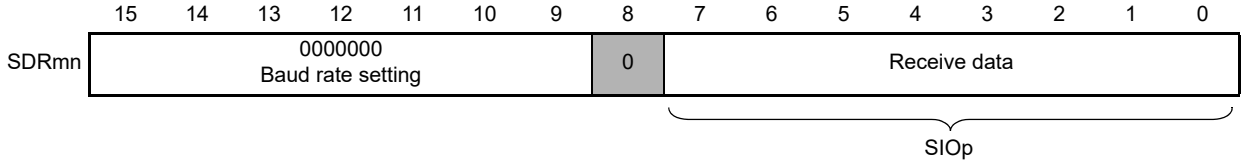
a) Serial mode register mn (SMRmn)



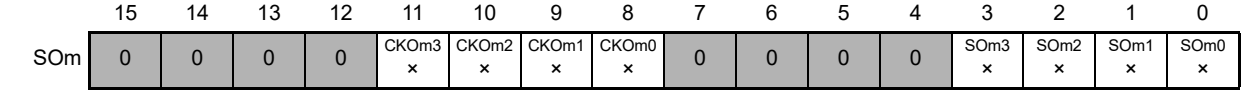
b) Serial communication operation setting register mn (SCRmn)



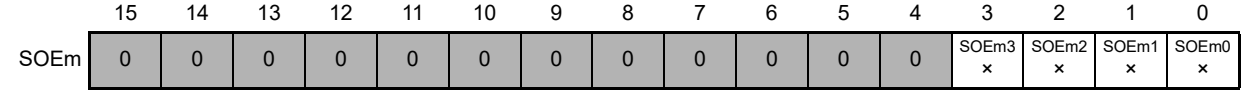
c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



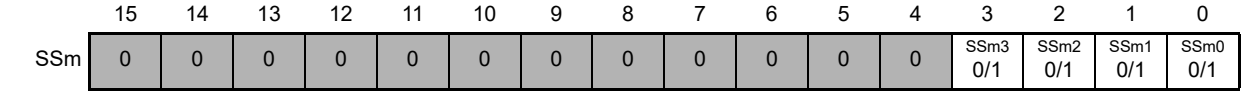
d) Serial output register m (SOm): This register is not used in this mode.



e) Serial output enable register m (SOEm): This register is not used in this mode.



f) Serial channel start register m (SSm): Set only the bit of the target channel to 1.



Note This bit is only present in the following registers; in other registers, it is fixed to 1.
- SCR00 and SCR01 registers
- SCR10 and SCR11 registers of the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products

(Remarks are listed on the next page.)

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Remark 2. ■: Setting is fixed in the slave reception mode of the simplified SPI (CSI), ■: Set to the initial value.
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the user's application.

2. Operation procedure

Figure 15 - 59 Initial Setting Procedure for Slave Reception

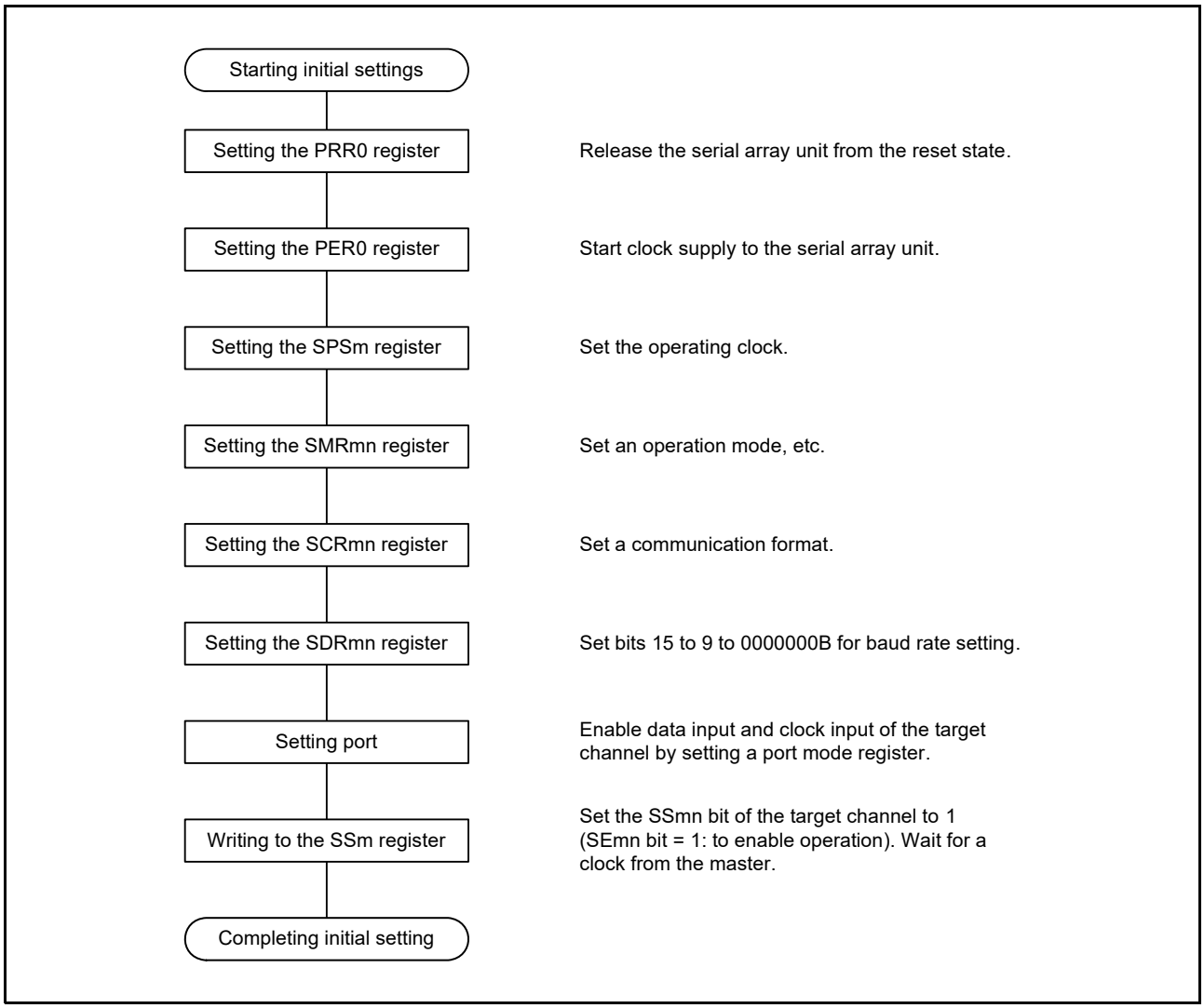


Figure 15 - 60 Procedure for Stopping Slave Reception

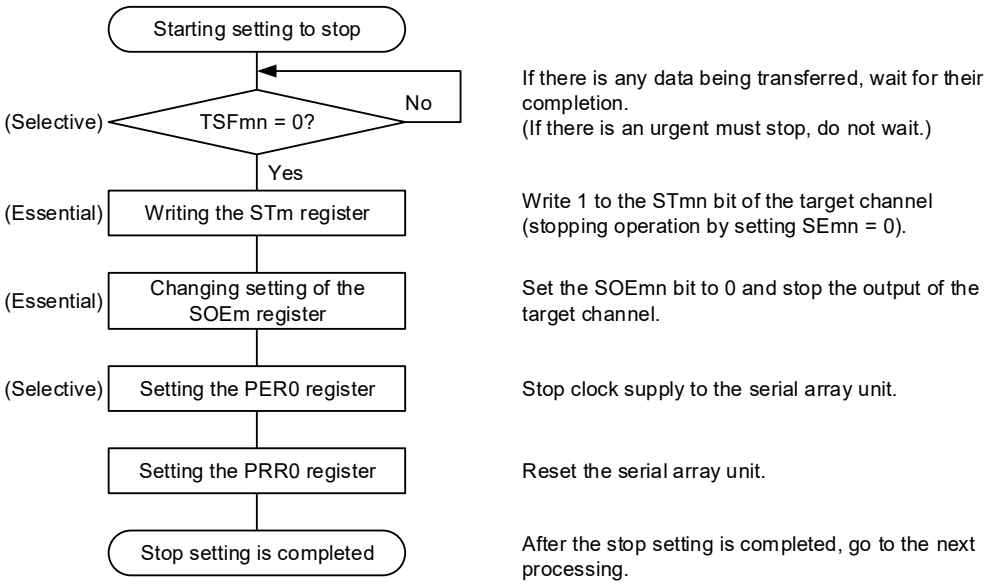
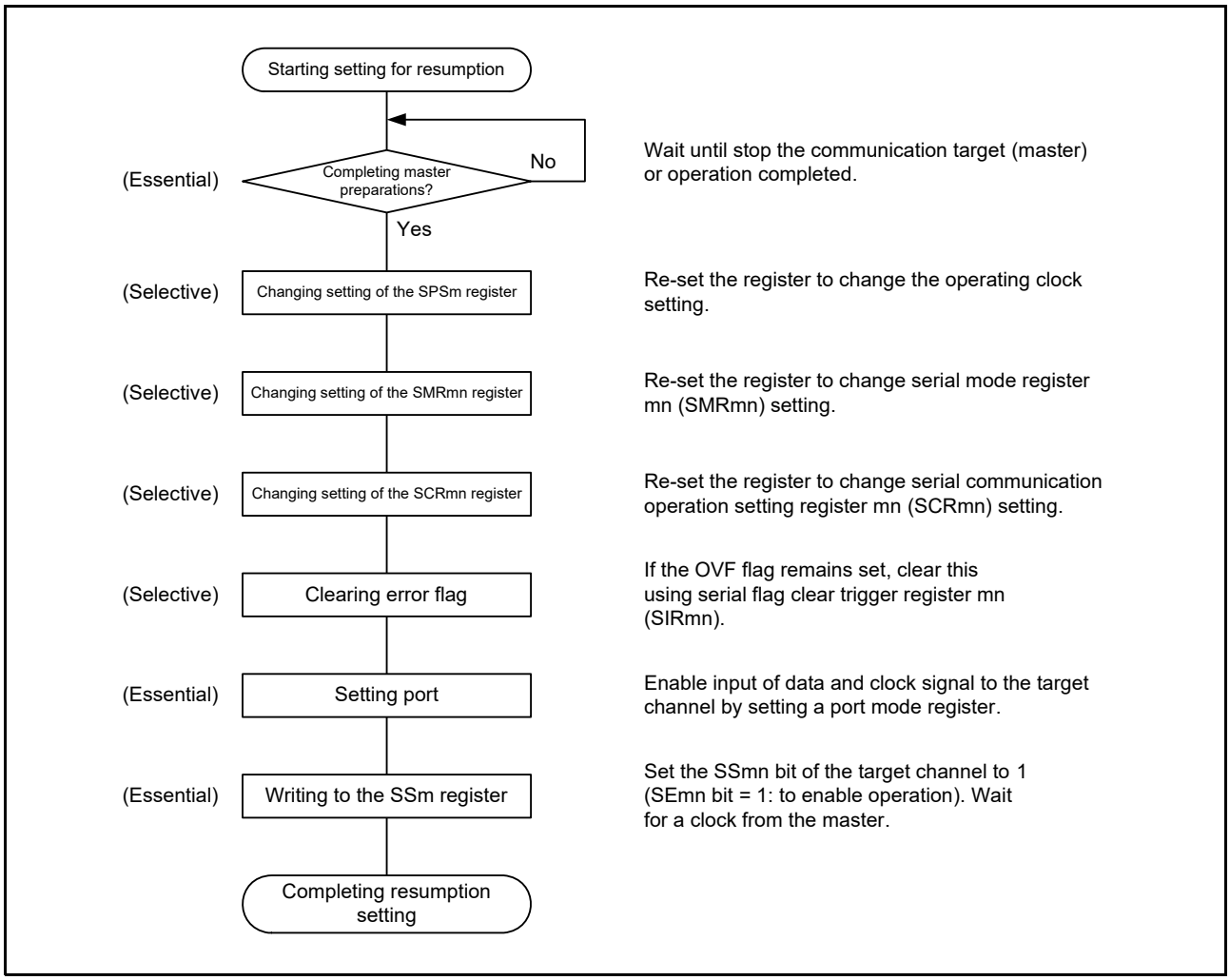


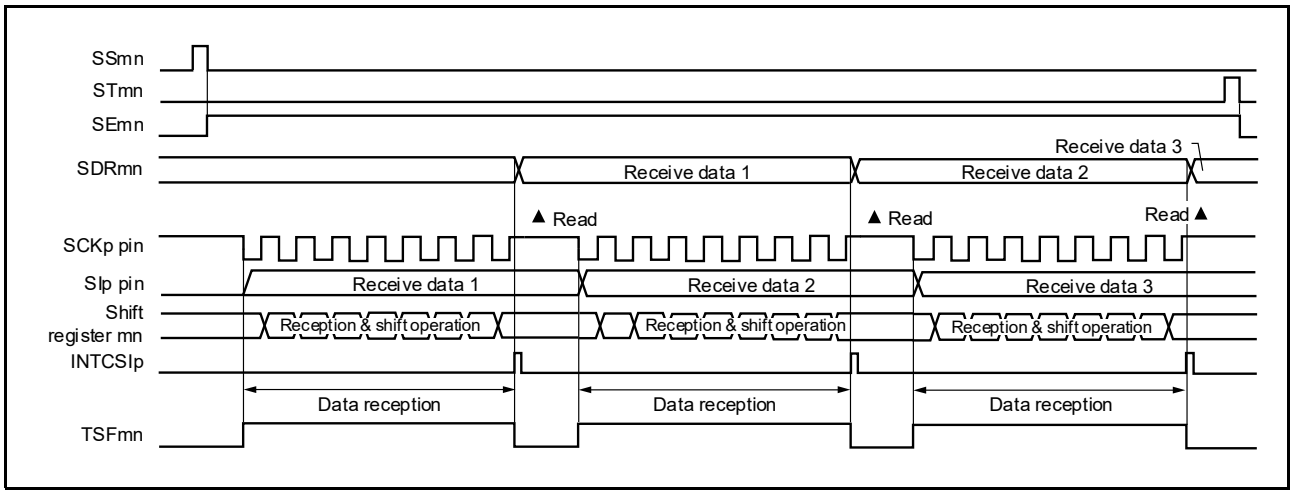
Figure 15 - 61 Procedure for Resuming Slave Reception



Remark If the PRR0 register is rewritten while stopping the communication to reset the serial array unit, wait until the communication target (master) stops or communication finishes, and then perform initialization instead of restarting the communication.

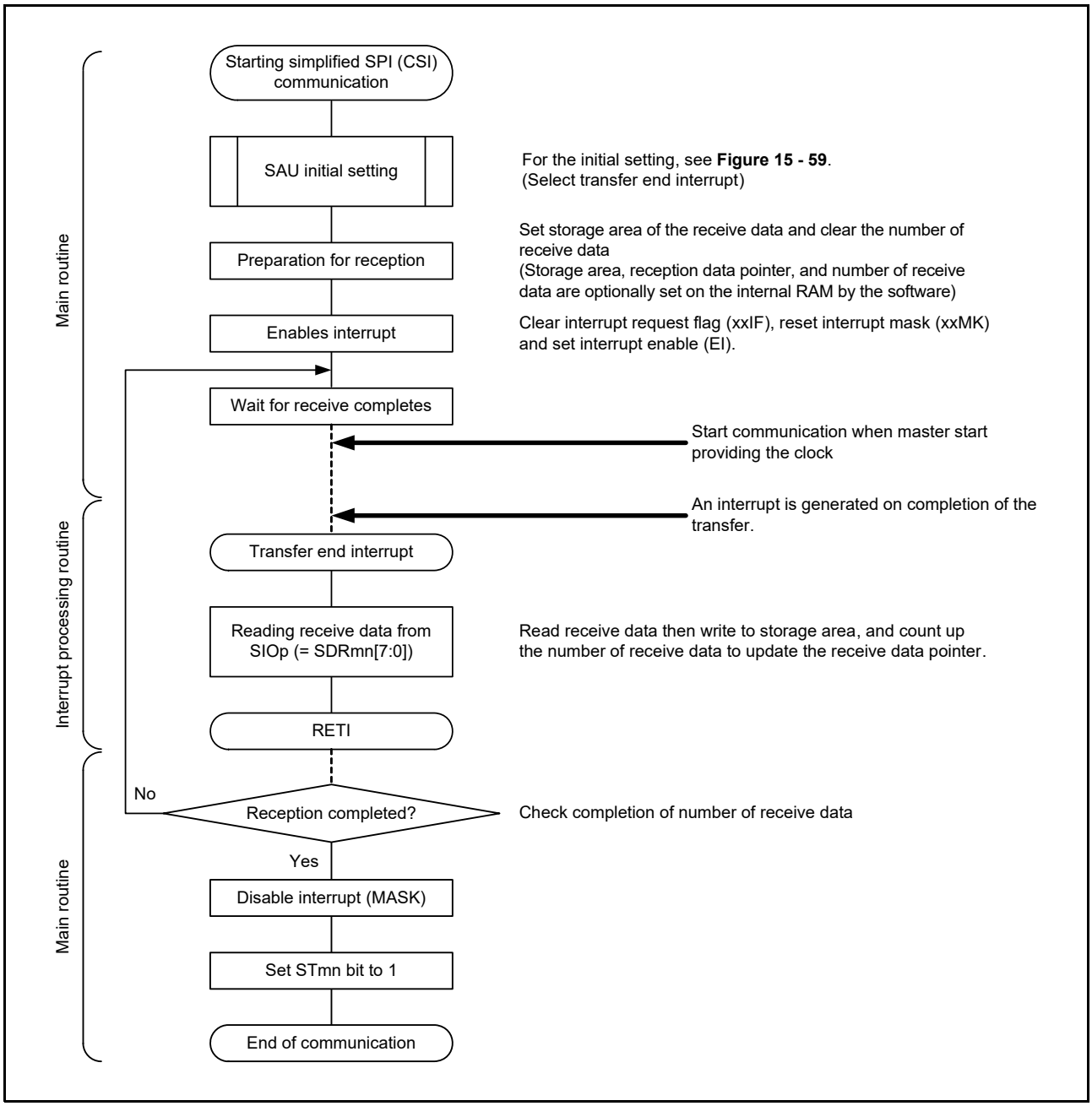
3. Processing flow (in single-reception mode)

Figure 15 - 62 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 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Figure 15 - 63 Flowchart of Slave Reception (in Single-reception Mode)



15.5.6 Slave transmission/reception

Slave transmission/reception is that this MCU transmits/receives data to/from another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00	CSI01	CSI10	CSI11	CSI20	CSI21	CSI30	CSI31
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 2 of SAU1	Channel 3 of SAU1
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01	SCK10, SI10, SO10	SCK11, SI11, SO11	SCK20, SI20, SO20	SCK21, SI21, SO21	SCK30, SI30, SO30	SCK31, SI31, SO31
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11	INTCSI20	INTCSI21	INTCSI30	INTCSI31
	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 or 8 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 at the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock cycle before the start of the serial clock operation.							
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse							
Data direction	MSB or LSB first							

Note 1. Because the external serial clock input to the SCK00, SCK01, SCK10, SCK11, SCK20, SCK21, SCK30, and SCK31 pins is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz].

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 37 Electrical Characteristics**.

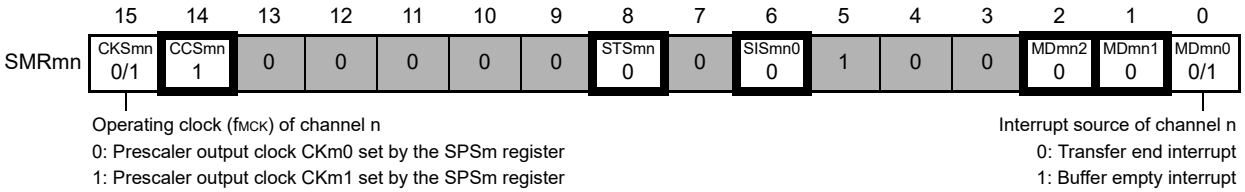
Remark 1. f_{MCK}: Operating clock frequency of target channel
f_{SCK}: Serial clock frequency

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

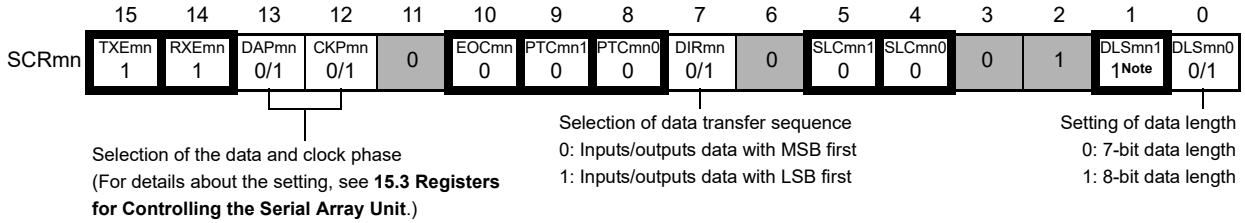
1. Register setting

Figure 15 - 64 Example of Contents of Registers for Slave Transmission/Reception of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31)

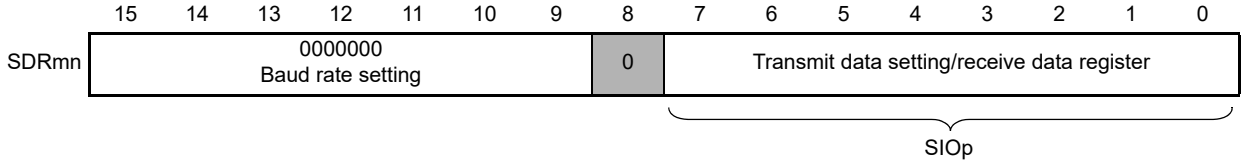
a) Serial mode register mn (SMRmn)



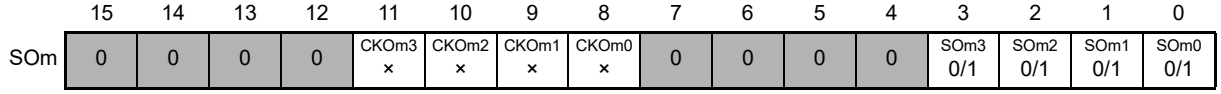
b) Serial communication operation setting register mn (SCRmn)



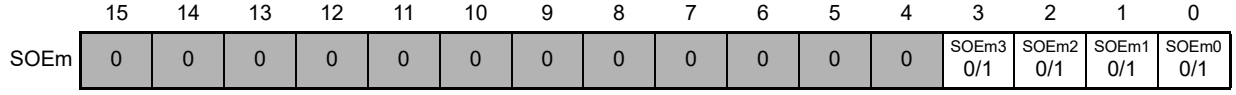
c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



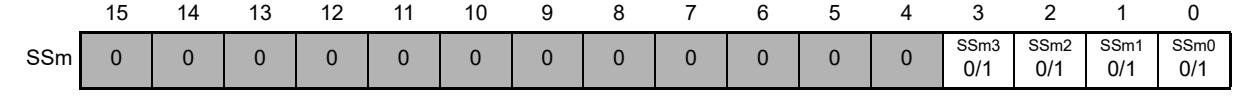
d) Serial output register m (SOm): Set only the bit of the target channel.



e) Serial output enable register m (SOEm): Set only the bit of the target channel to 1.



f) Serial channel start register m (SSm): Set only the bit of the target channel to 1.



Note This bit is only present in the following registers; in other registers, it is fixed to 1.
- SCR00 and SCR01 registers
- SCR10 and SCR11 registers of the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products

Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

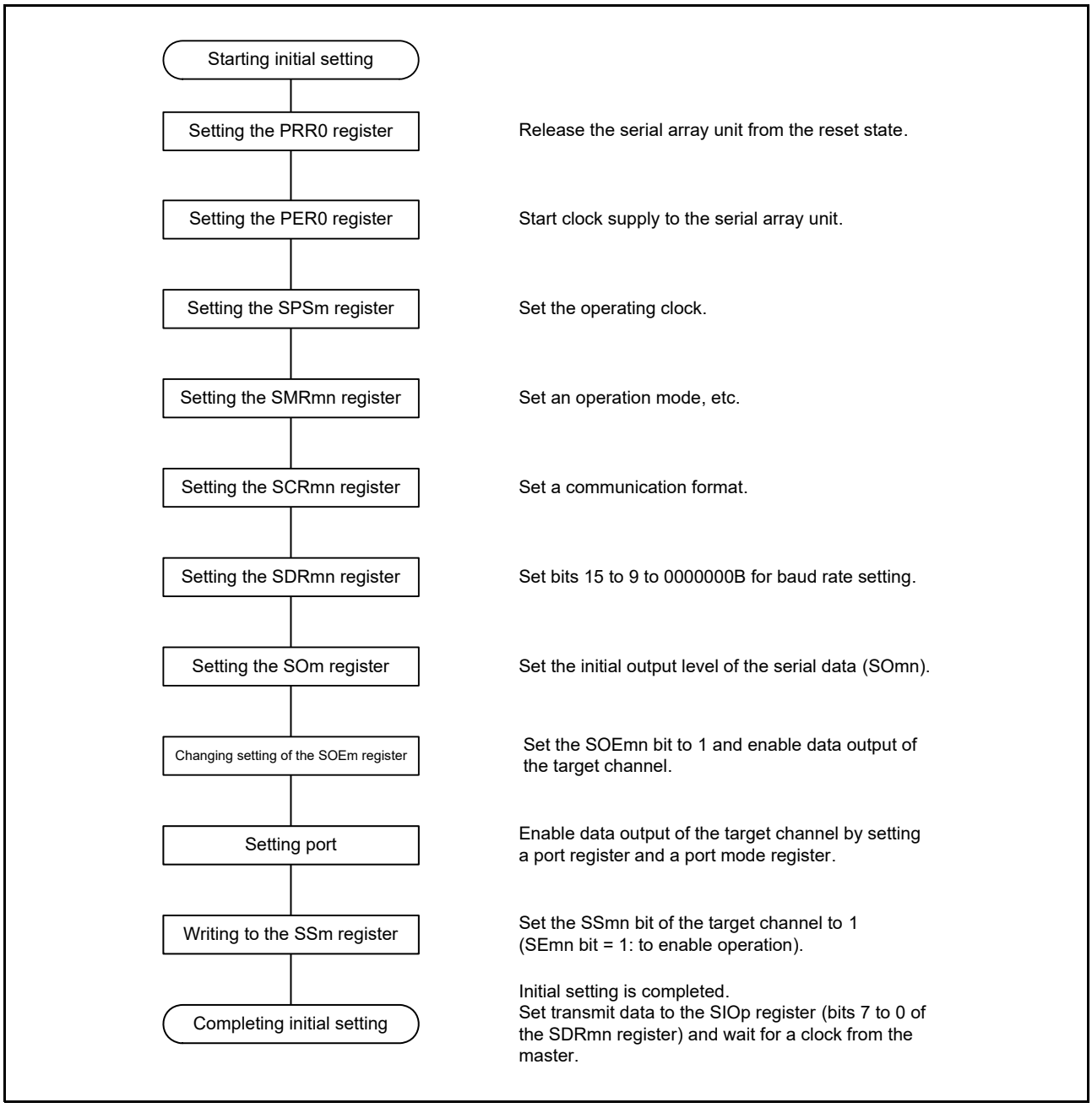
(Remarks are listed on the next page.)

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Remark 2. ■: Setting is fixed in the simplified SPI (CSI) slave transmission/reception mode, ■: Set to the initial value.
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the user's application.

2. Operation procedure

Figure 15 - 65 Initial Setting Procedure for Slave Transmission/Reception



Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Figure 15 - 66 Procedure for Stopping Slave Transmission/Reception

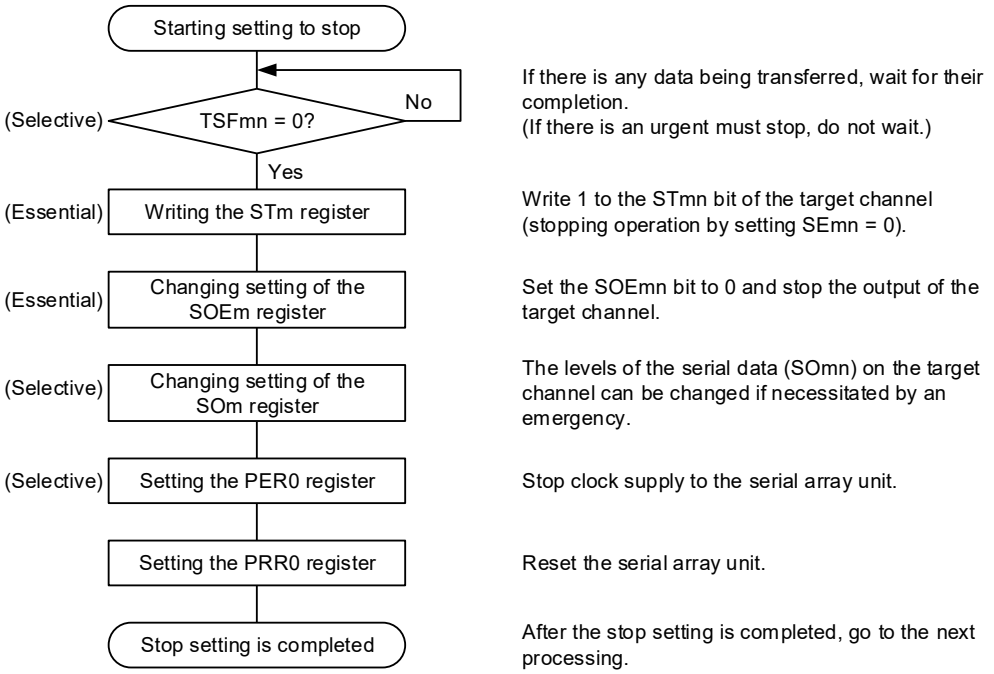
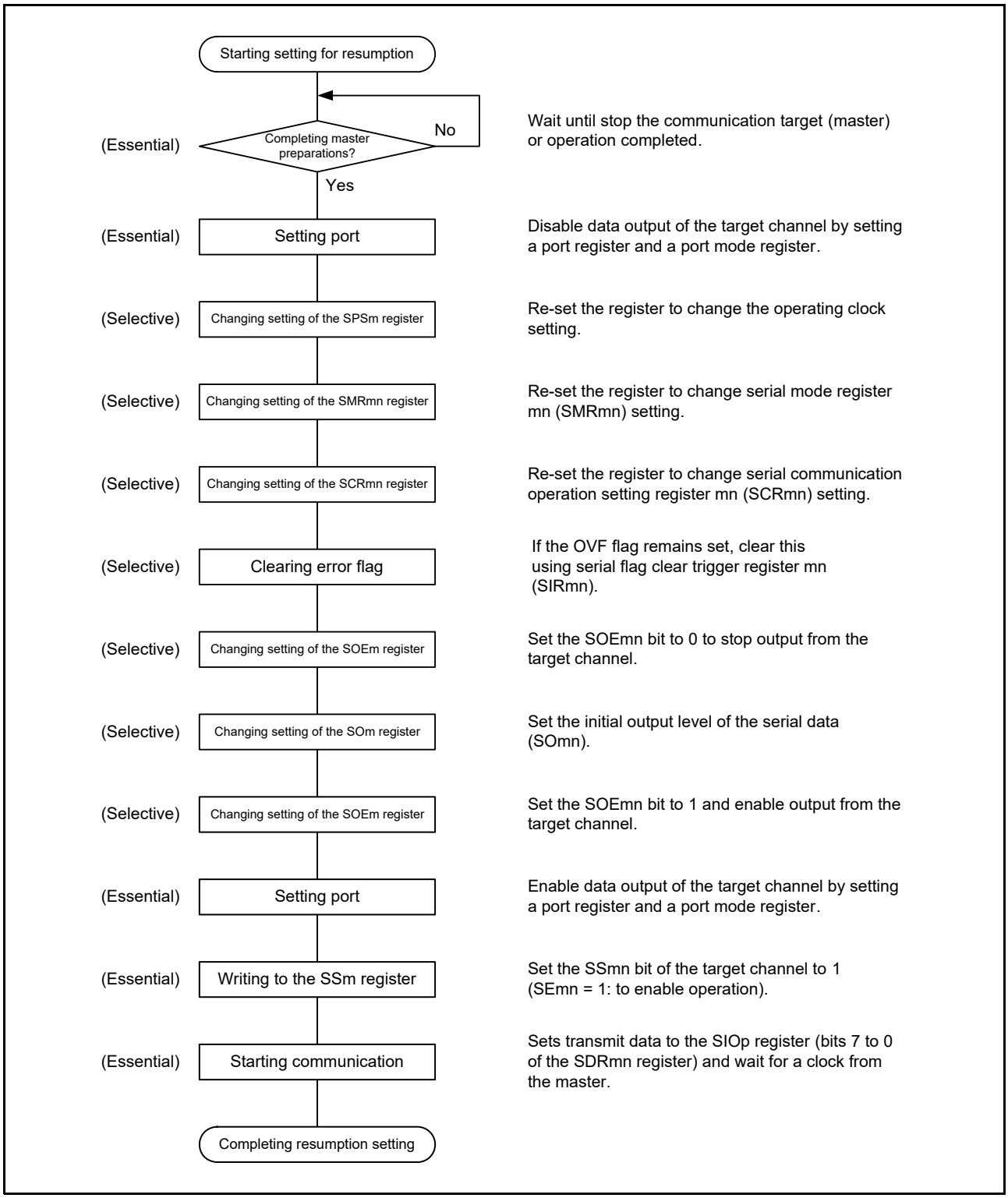


Figure 15 - 67 Procedure for Resuming Slave Transmission/Reception

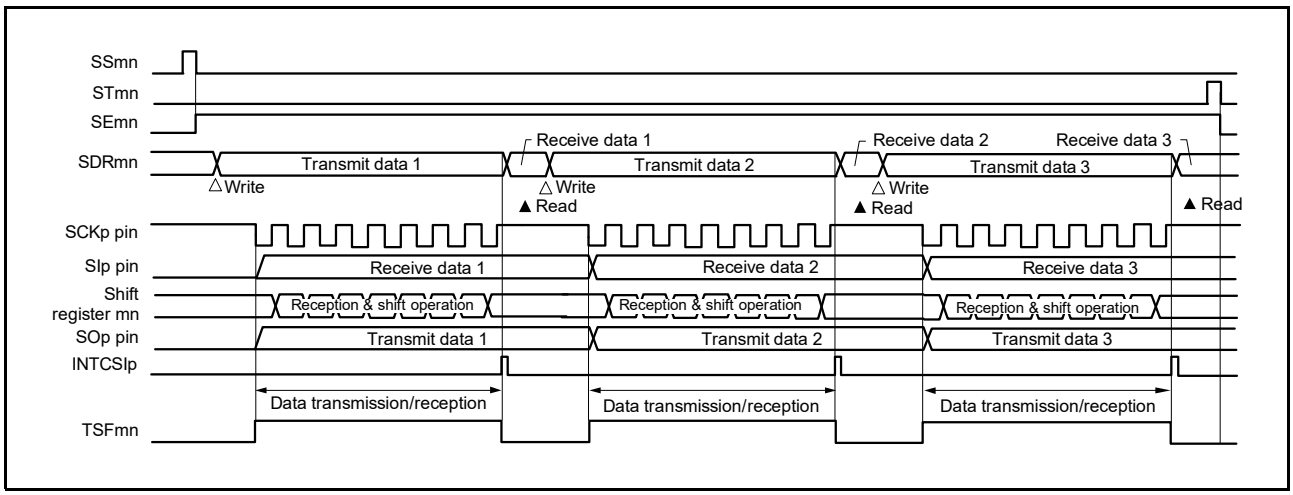


Caution 1. Be sure to set transmit data to the SIOp register before the clock from the master is started.

Caution 2. If the PRR0 register is rewritten while stopping the communication to reset the serial array unit, wait until the communication target (master) stops or communication finishes, and then perform initialization instead of restarting the communication.

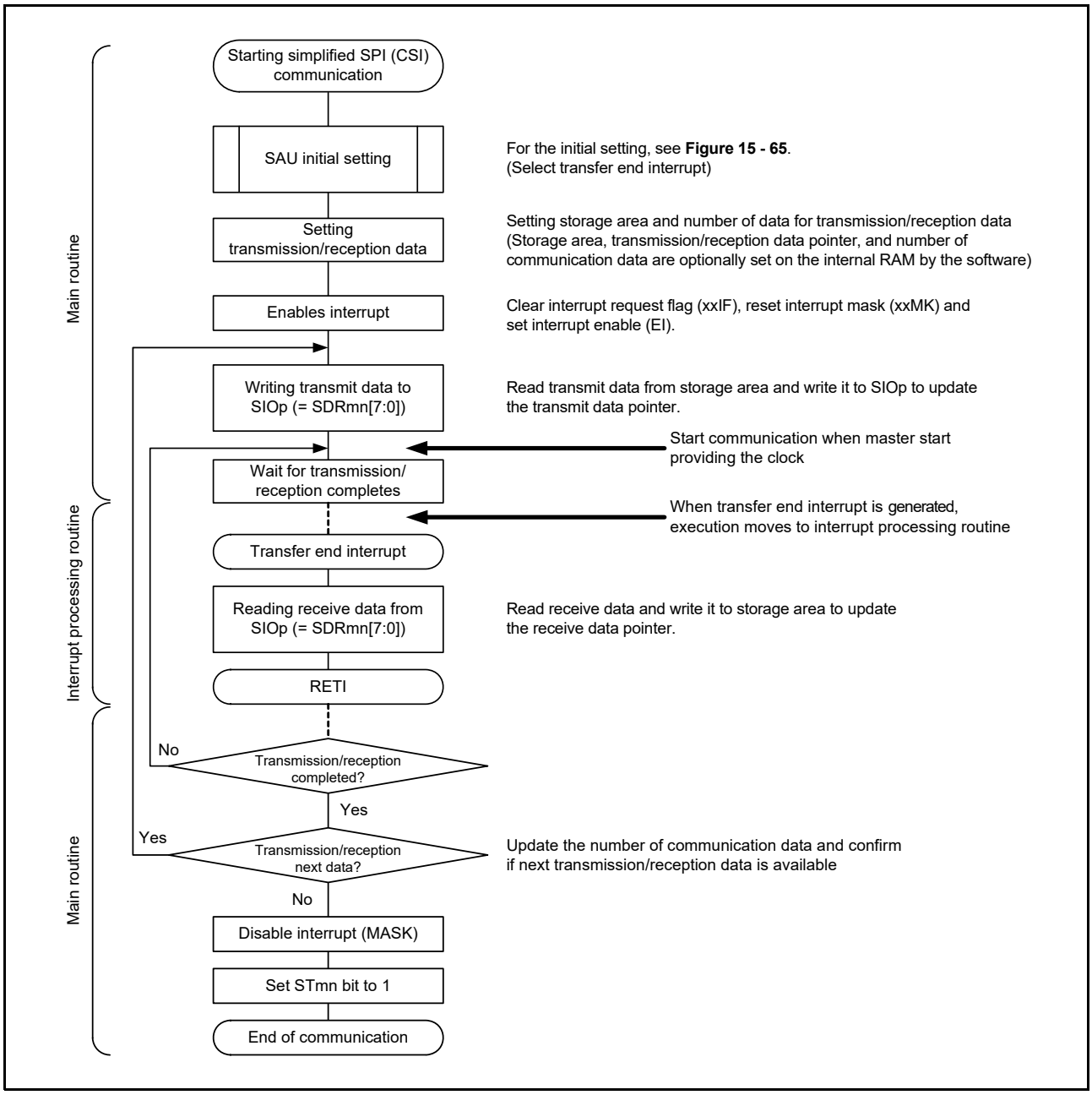
3. Processing flow (in single-transmission/reception mode)

Figure 15 - 68 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 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

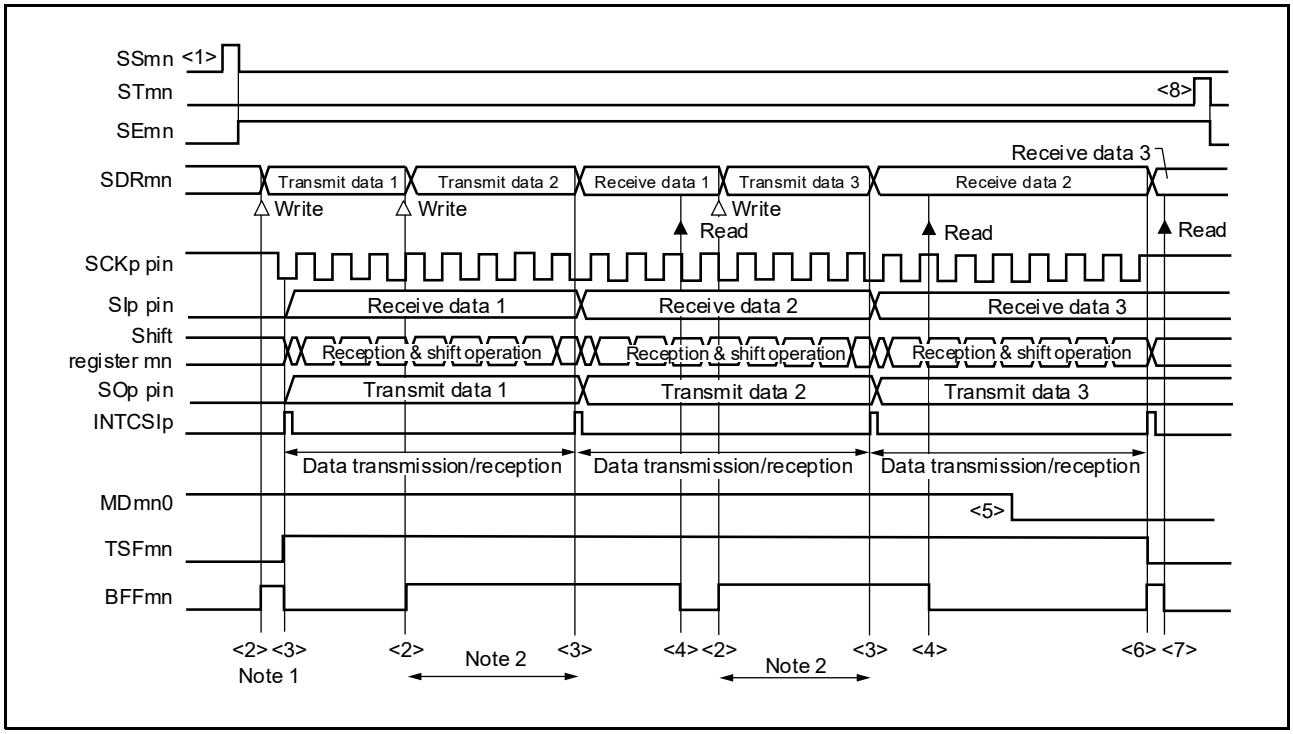
Figure 15 - 69 Flowchart of Slave Transmission/Reception (in Single-transmission/reception Mode)



Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

4. Processing flow (in continuous transmission/reception mode)

Figure 15 - 70 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Note 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.

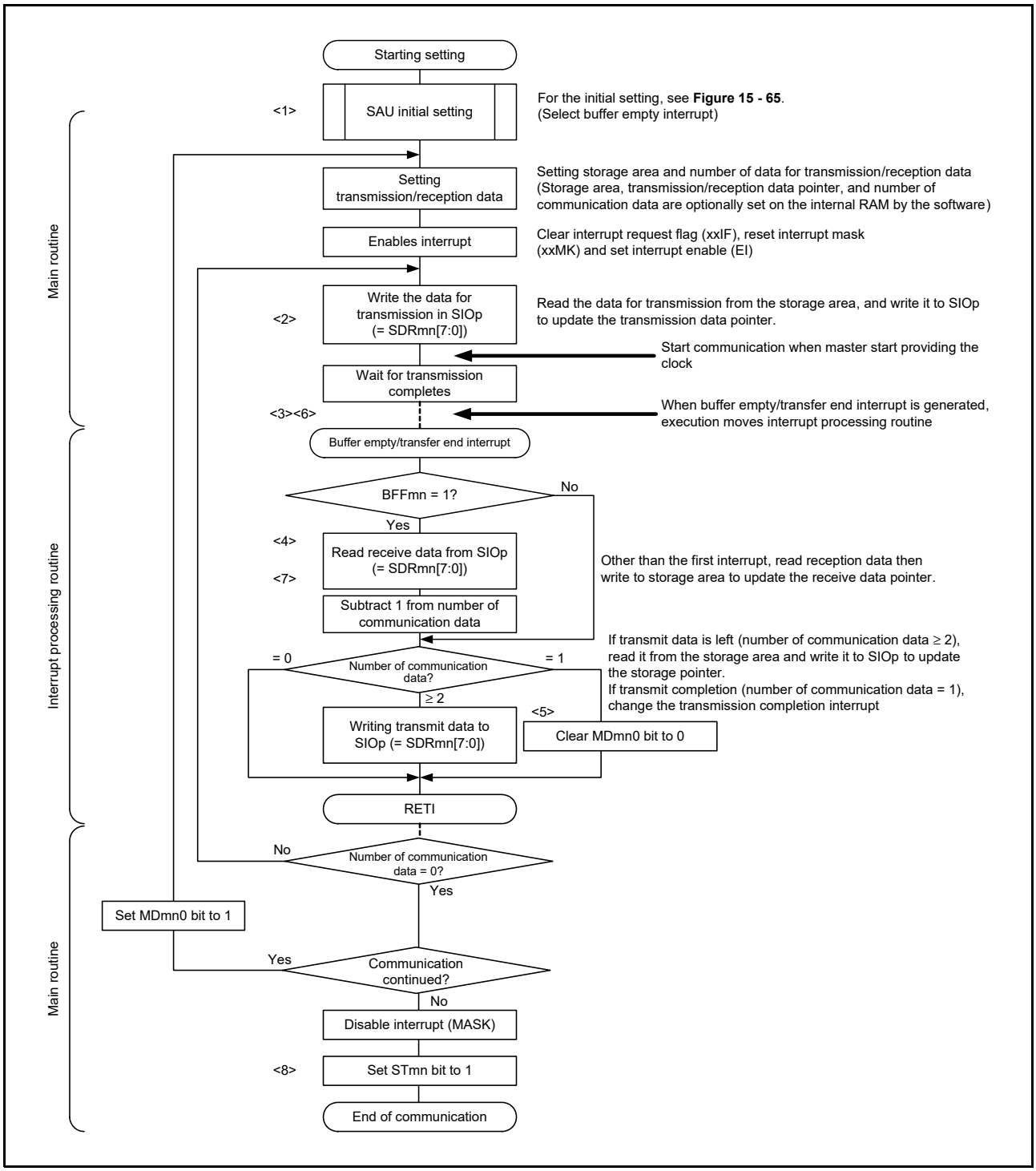
Note 2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution **The MDmn0 bit of serial mode register mn (SSRmn) 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.**

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 15 - 71 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Figure 15 - 71 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)



Caution **Be sure to set transmit data to the SIOp register before the clock from the master is started.**

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 15 - 70 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)**.

15.5.7 SNOOZE mode

The SNOOZE mode makes the simplified SPI (CSI) perform reception operations upon SCKp pin input detection while in the STOP mode. Normally the simplified SPI (CSI) stops communication in the STOP mode. However, using the SNOOZE mode enables the simplified SPI (CSI) to perform reception operations without CPU operation upon detection of the SCKp pin input. Only the following simplified SPIs (CSIs) can be set for the SNOOZE mode.

- CSI00 in the 30- to 64-pin R7F100G and 32- to 48-pin R7F104G products
- CSI00 and CSI20 in the 80- to 128-pin R7F100G products

When using the simplified SPI (CSI) in SNOOZE mode, make the following setting before switching to the STOP mode. See **Figure 15 - 73 Flowchart of SNOOZE Mode Operation (Once Startup)** and **Figure 15 - 75 Flowchart of SNOOZE Mode Operation (Continuous Startup)**.

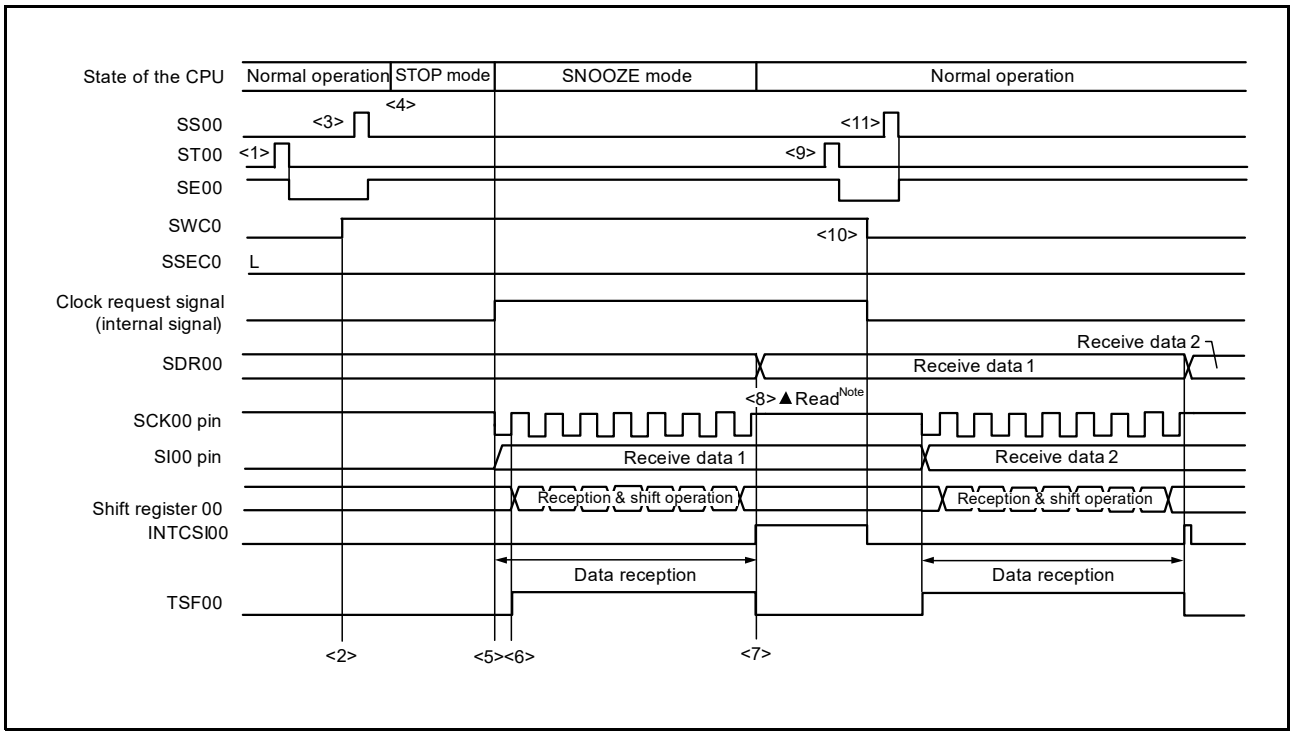
- When using the SNOOZE mode function, set the SWCm bit of serial standby control register m (SSCm) to 1 just before switching to the STOP mode. After the initial setting has been completed, set the SSm0 bit of serial channel start register m (SSm) to 1.
- The CPU shifts to the SNOOZE mode on detecting the valid edge of the SCKp signal following a transition to the STOP mode. A CSIp starts reception on detecting input of the serial clock on the SCKp pin.

Caution 1. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock or medium-speed on-chip oscillator clock is selected for f_{CLK}.

Caution 2. The maximum transfer rate when using CSIp in the SNOOZE mode is 1 Mbps.

1. SNOOZE mode operation (once startup)

Figure 15 - 72 Timing Chart of SNOOZE Mode Operation (Once Startup) (Type 1: DAPmn = 0, CKPmn = 0)



Note Only read received data while SWCm = 1 and before the next valid edge of the SCKp pin input is detected.

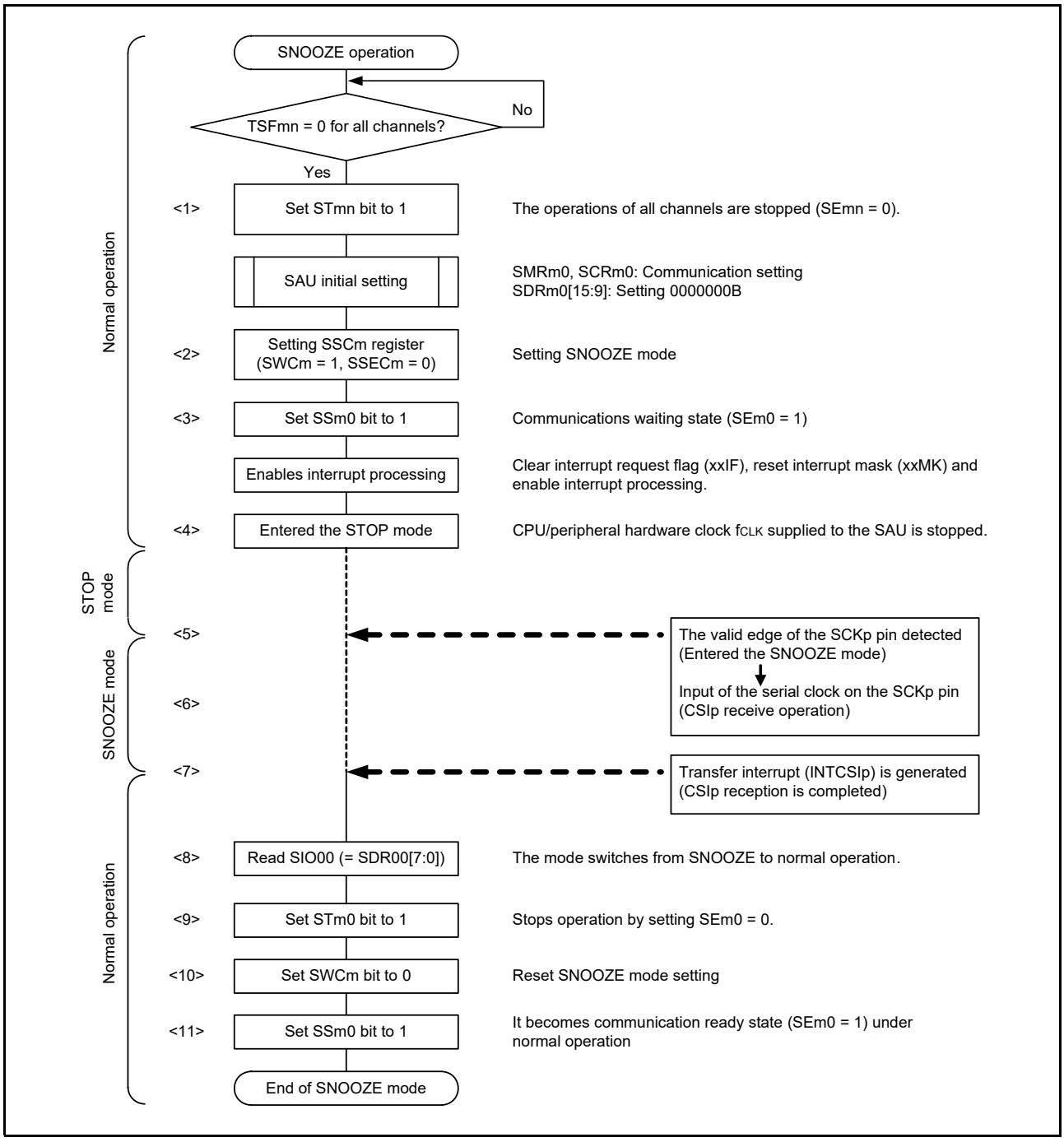
Caution 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (the SEm0 bit is cleared and the operation stops). After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).

Caution 2. When SWCm = 1, the BFFm0 and OVFm0 flags will not change.

Remark 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 15 - 73 Flowchart of SNOOZE Mode Operation (Once Startup)**.

Remark 2. 30- and 32-pin R7F100G products
m: Unit number (m = 0), n: Channel number (n = 0, 3), p: CSI number (p = 00)
36- to 52-pin R7F100G products
m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), p: CSI number (p = 00)
64-pin R7F100G products
m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00)
80- to 128-pin R7F100G products
m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 20)
32- to 48-pin R7F104G products
m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00)

Figure 15 - 73 Flowchart of SNOOZE Mode Operation (Once Startup)

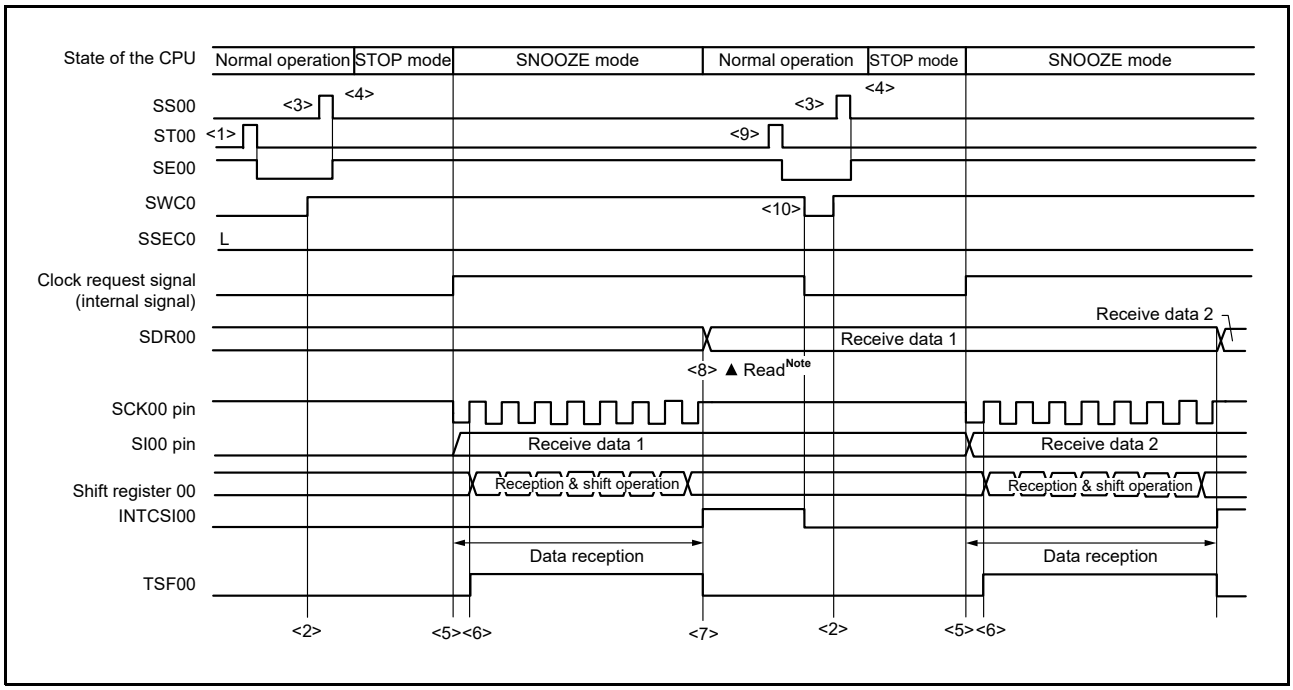


Remark 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 15 - 72 Timing Chart of SNOOZE Mode Operation (Once Startup) (Type 1: DAPmn = 0, CKPmn = 0)**.

Remark 2. 30- and 32-pin R7F100G products
m: Unit number (m = 0), n: Channel number (n = 0, 3), p: CSI number (p = 00)
36- to 52-pin R7F100G products
m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), p: CSI number (p = 00)
64-pin R7F100G products
m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00)
80- to 128-pin R7F100G products
m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 20)
32- to 48-pin R7F104G products
m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00)

2. SNOOZE mode operation (continuous startup)

Figure 15 - 74 Timing Chart of SNOOZE Mode Operation (Continuous Startup) (Type 1: DAPmn = 0, CKPmn = 0)



Note Only read received data while SWCm = 1 and before the next valid edge of the SCKp pin input is detected.

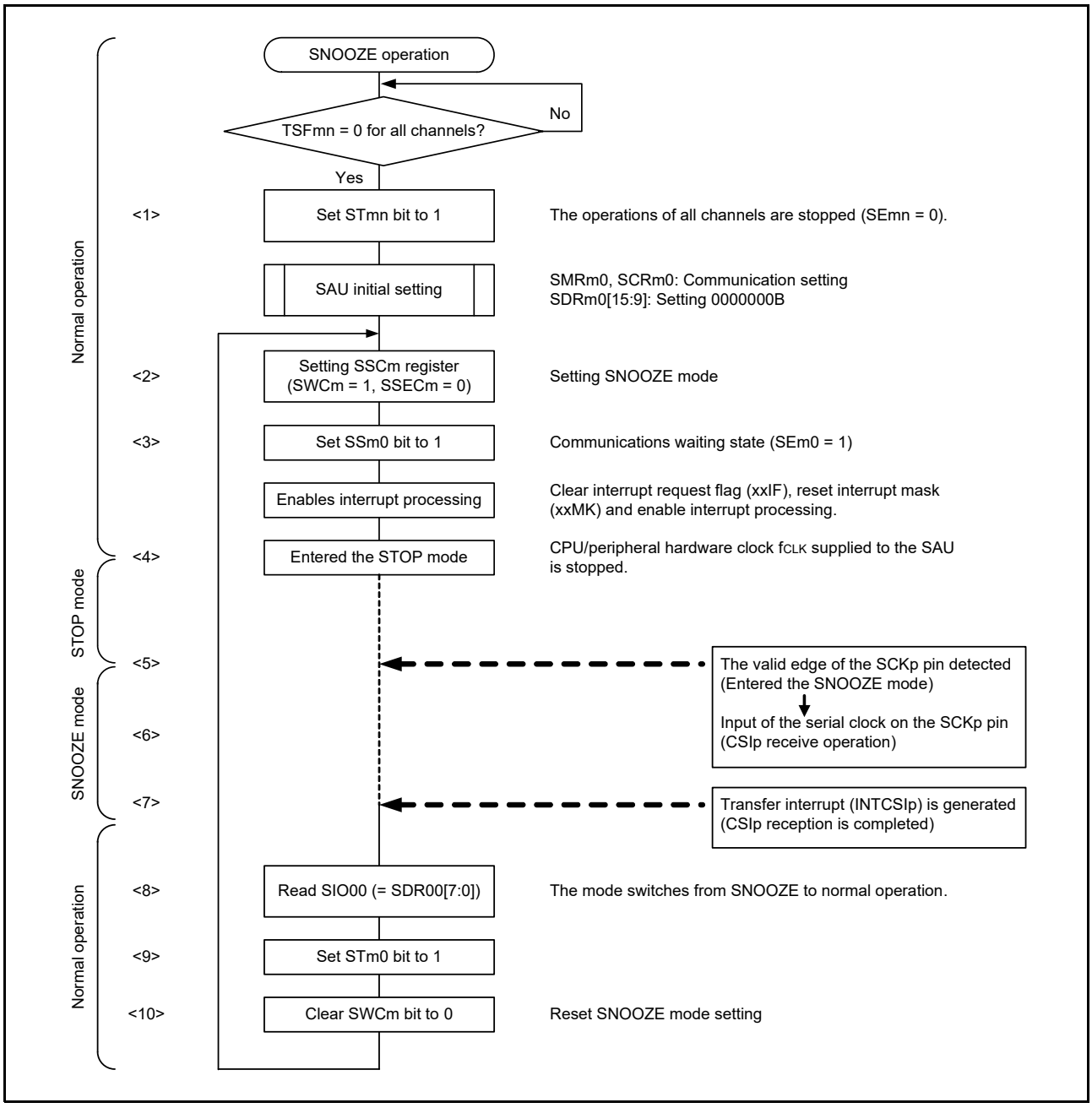
Caution 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (the SEm0 bit is cleared and the operation stops). After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).

Caution 2. When SWCm = 1, the BFFm0 and OVFm0 flags will not change.

Remark 1. <1> to <10> in the figure correspond to <1> to <10> in **Figure 15 - 75 Flowchart of SNOOZE Mode Operation (Continuous Startup)**.

Remark 2. 30- and 32-pin R7F100G products
m: Unit number (m = 0), n: Channel number (n = 0, 3), p: CSI number (p = 00)
36- to 52-pin R7F100G products
m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), p: CSI number (p = 00)
64-pin R7F100G products
m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00)
80- to 128-pin R7F100G products
m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 20)
32- to 48-pin R7F104G products
m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00)

Figure 15 - 75 Flowchart of SNOOZE Mode Operation (Continuous Startup)



Remark 1. <1> to <10> in the figure correspond to <1> to <10> in **Figure 15 - 74 Timing Chart of SNOOZE Mode Operation (Continuous Startup) (Type 1: DAPmn = 0, CKPmn = 0)**.

Remark 2. 30- and 32-pin R7F100G products
m: Unit number (m = 0), n: Channel number (n = 0, 3), p: CSI number (p = 00)
36- to 52-pin R7F100G products
m: Unit number (m = 0, 1), n: Channel number (n = 0, 1, 3), p: CSI number (p = 00)
64-pin R7F100G products
m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00)
80- to 128-pin R7F100G products
m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 20)
32- to 48-pin R7F104G products
m: Unit number (m = 0, 1), n: Channel number (n = 0, 1), p: CSI number (p = 00)

15.5.8 Calculating transfer clock frequency

The transfer clock frequency for simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31) communication can be calculated by the following expressions.

1. Master

(Transfer clock frequency) = {Operating clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [Hz]

2. Slave

(Transfer clock frequency) = {Frequency of serial clock (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 operating clock (f_{MCK}) is determined by serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn).

Table 15 - 3 Selection of Operating Clock for Simplified SPI (1/2)

SMRmn Register	SPSm Register								Operating Clock (f_{MCK})[Note]	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 32 MHz
0	×	×	×	×	0	0	0	0	f_{CLK}	32 MHz
	×	×	×	×	0	0	0	1	$f_{CLK}/2$	16 MHz
	×	×	×	×	0	0	1	0	$f_{CLK}/2^2$	8 MHz
	×	×	×	×	0	0	1	1	$f_{CLK}/2^3$	4 MHz
	×	×	×	×	0	1	0	0	$f_{CLK}/2^4$	2 MHz
	×	×	×	×	0	1	0	1	$f_{CLK}/2^5$	1 MHz
	×	×	×	×	0	1	1	0	$f_{CLK}/2^6$	500 kHz
	×	×	×	×	0	1	1	1	$f_{CLK}/2^7$	250 kHz
	×	×	×	×	1	0	0	0	$f_{CLK}/2^8$	125 kHz
	×	×	×	×	1	0	0	1	$f_{CLK}/2^9$	62.5 kHz
	×	×	×	×	1	0	1	0	$f_{CLK}/2^{10}$	31.25 kHz
	×	×	×	×	1	0	1	1	$f_{CLK}/2^{11}$	15.63 kHz
	×	×	×	×	1	1	0	0	$f_{CLK}/2^{12}$	7.81 kHz
	×	×	×	×	1	1	0	1	$f_{CLK}/2^{13}$	3.91 kHz
	×	×	×	×	1	1	1	0	$f_{CLK}/2^{14}$	1.95 kHz
	×	×	×	×	1	1	1	1	$f_{CLK}/2^{15}$	977 Hz

Table 15 - 3 Selection of Operating Clock for Simplified SPI (2/2)

SMRmn Register	SPSm Register								Operating 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
1	0	0	0	0	×	×	×	×	f_{CLK}	32 MHz
	0	0	0	1	×	×	×	×	$f_{CLK}/2$	16 MHz
	0	0	1	0	×	×	×	×	$f_{CLK}/2^2$	8 MHz
	0	0	1	1	×	×	×	×	$f_{CLK}/2^3$	4 MHz
	0	1	0	0	×	×	×	×	$f_{CLK}/2^4$	2 MHz
	0	1	0	1	×	×	×	×	$f_{CLK}/2^5$	1 MHz
	0	1	1	0	×	×	×	×	$f_{CLK}/2^6$	500 kHz
	0	1	1	1	×	×	×	×	$f_{CLK}/2^7$	250 kHz
	1	0	0	0	×	×	×	×	$f_{CLK}/2^8$	125 kHz
	1	0	0	1	×	×	×	×	$f_{CLK}/2^9$	62.5 kHz
	1	0	1	0	×	×	×	×	$f_{CLK}/2^{10}$	31.25 kHz
	1	0	1	1	×	×	×	×	$f_{CLK}/2^{11}$	15.63 kHz
	1	1	0	0	×	×	×	×	$f_{CLK}/2^{12}$	7.81 kHz
	1	1	0	1	×	×	×	×	$f_{CLK}/2^{13}$	3.91 kHz
	1	1	1	0	×	×	×	×	$f_{CLK}/2^{14}$	1.95 kHz
	1	1	1	1	×	×	×	×	$f_{CLK}/2^{15}$	977 Hz
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) = 000FH) the operation of the serial array unit (SAU).

Remark 1. ×: Don't care

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

15.5.9 Procedure for processing errors that occurred during simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31) communication

Table 15 - 4 describes the procedure for processing errors that occurred during simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31) communication.

Table 15 - 4 Processing Procedure in Case of Overrun Error

Software Manipulation	State of the Hardware	Remark
Reads serial data register mn (SDRmn).	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		The error type is identified and the read value is used to clear the error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn).	The error flag is cleared.	The error only during reading can be cleared, 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 to 3), mn = 00 to 03, 10 to 13

The serial data output (SO00, SO01) and serial clock output (SCK00, SCK01) of channels 0 and 1 of serial array unit 0 can be selected as input for the logic and event link controller (ELCL). When the serial data output and serial clock output signals are to be used for event input to the ELCL but not for output to the SO0m and SCK0m pins, set the corresponding bit in port function output enable register 0 (PFOE0) to 0. See **4.3.15 Port function output enable registers (PFOEx)** for details.

15.6 Operation of UART (UART0 to UART3) Communication

This is a start-stop synchronization communication 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 asynchronous communication 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 UART2, timer array unit 0 (channel 7), and an external interrupt (INTP0).

[Data transmission/reception]

- Data length of 7, 8, or 9 bits**Note**
- MSB/LSB first selectable
- Level setting of transmit/receive data (selecting whether to reverse the level)
- Parity bit appending and parity check functions
- Stop bit appending, stop bit check function

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

Furthermore, only the following UARTs support reception in the SNOOZE mode. In the SNOOZE mode, data can be received without CPU processing upon detecting RxD input in the STOP mode.

- UART0 in the 30- to 64-pin R7F100G and 32-pin R7F104G products
- UART0 and UART2 in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products

UART2 (channels 0 and 1 of unit 1) only supports the LIN bus in the 30- to 128-pin R7F100G and 32- to 48-pin R7F104G products.

[LIN-bus functions]

- Wakeup signal detection
- Break field (BF) detection
- Sync field measurement, baud rate calculation

} Using the external interrupt (INTP0) and the timer array unit 0 (channel 7)

Note Only the following UARTs support the 9-bit data length.

- UART0 in the 30- to 64-pin R7F100G and 32-pin R7F104G products
- UART0 and UART2 in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products

When the medium-speed on-chip oscillator clock (f_{IM}) or low-speed on-chip oscillator clock (f_{IL}) is selected for f_{CLK}, use the medium-speed on-chip oscillator trimming register (MIOTRM) or the low-speed on-chip oscillator trimming register (LIOTRM) to correct the accuracy of the oscillation frequency.

UART0 uses channels 0 and 1 of SAU0.
UART1 uses channels 2 and 3 of SAU0.
UART2 uses channels 0 and 1 of SAU1.
UART3 uses channels 2 and 3 of SAU1.

30- and 32-pin R7F100G products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting the LIN bus)	IIC20
	1	—		—

36- to 44-pin R7F100G products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting the LIN bus)	IIC20
	1	CSI21		IIC21

48- and 52-pin R7F100G products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting the LIN bus)	IIC20
	1	CSI21		IIC21

64-pin R7F100G products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting the LIN bus)	IIC20
	1	CSI21		IIC21

80- to 128-pin R7F100G products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting the LIN bus)	IIC20
	1	CSI21		IIC21
	2	CSI30	UART3	IIC30
	3	CSI31		IIC31

32- to 48-pin R7F104G products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	—		IIC11
1	0	—	UART2 (supporting the LIN bus)	—
	1	CSI21		IIC21

Select a single function for each channel. Only the selected function is possible. If UART0 is selected for channels 0 and 1 of unit 0, for example, the CSI00 and CSI01 functions cannot be used. At this time, however, channel 2 or 3 of the same unit can be used for a function other than UART0, such as CSI10, UART1, and IIC10.

Caution When using a serial array unit for UART, both the transmitter side (even-numbered channel) and the receiver side (odd-numbered channel) can only be used for UART.

UART performs the following four types of communication operations.

- UART transmission (See **15.6.1**.)
- UART reception (See **15.6.2**.)
- LIN transmission (UART2 only) (See **15.7.1**.)
- LIN reception (UART2 only) (See **15.7.2**.)

15.6.1 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	UART2	UART3
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1	Channel 2 of SAU1
Pins used	TxD0	TxD1	TxD2	TxD3
Interrupt	INTST0	INTST1	INTST2	INTST3
	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, 8, or 9 bits**Note 1**			
Transfer rate**Note 2**	Max. f_{MCK}/6 [bps] (SDRmn[15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]			
Data phase	Non-reverse 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 bit			
Data direction	MSB or LSB first			

Note 1. Only the following UARTs support the 9-bit data length.
- UART0 in the 30- to 64-pin R7F100G and 32-pin R7F104G products
- UART0 and UART2 in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 37 Electrical Characteristics**.

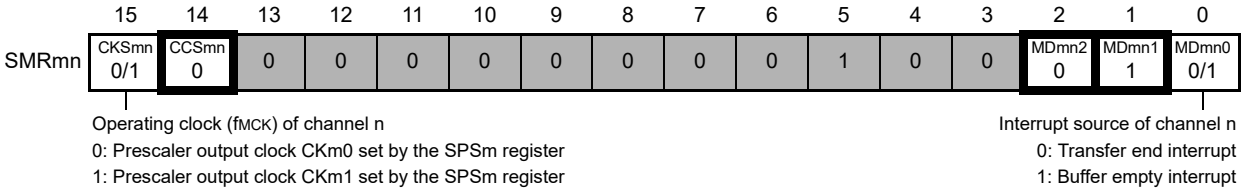
Remark 1. f_{MCK}: Operating clock frequency of target channel
f_{CLK}: CPU/peripheral hardware clock frequency

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12

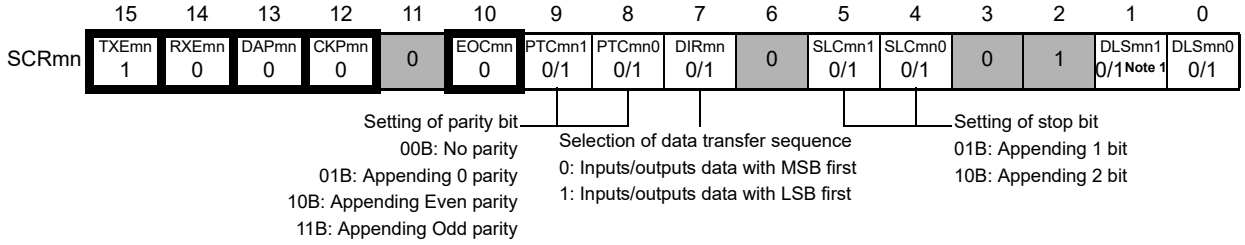
1. Register setting

Figure 15 - 76 Example of Contents of Registers for UART Transmission of UART (UART0 to UART3)

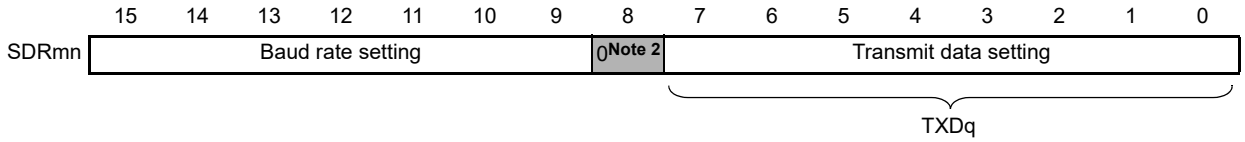
a) Serial mode register mn (SMRmn)



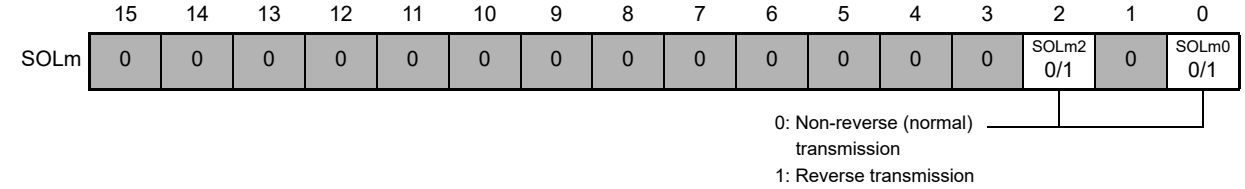
b) Serial communication operation setting register mn (SCRmn)



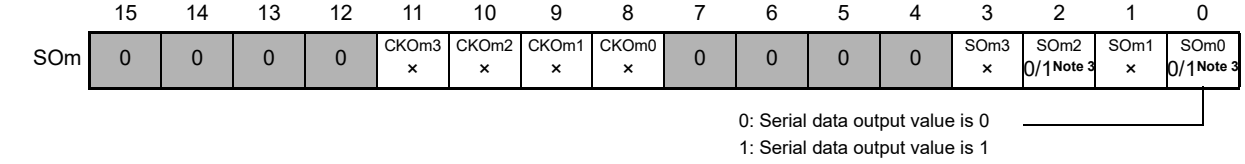
c) Serial data register mn (SDRmn) (lower 8 bits: TXDq)



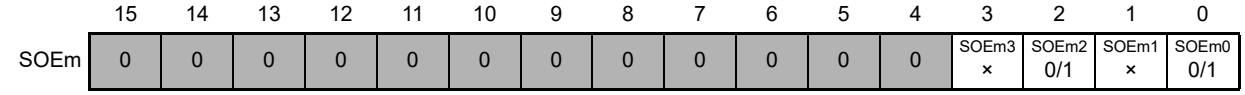
d) Serial output level register m (SOLm): Set only the bit of the target channel.



e) Serial output register m (SOm): Set only the bit of the target channel.



f) Serial output enable register m (SOEm): Set only the bit of the target channel to 1.



g) Serial channel start register m (SSm): Set only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

Note 1. This bit is only present in the following registers; in other registers, it is fixed to 1.
- SCR00 and SCR01 registers
- SCR10 and SCR11 registers of the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products

Note 2. When UART performs 9-bit communication, bits 0 to 8 of the SDRm0 register are used as the transmission data specification area. Only the following UARTs support the 9-bit data length.
- UART0 in the 30- to 64-pin R7F100G and 32-pin R7F104G products
- UART0 and UART2 in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products

Note 3. Before transmission is started, be sure to set this bit to 1 when the SOLmn bit of the target channel is set to 0, and set this bit 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.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 3), mn = 00, 02, 10, 12

Remark 2. ■: Setting is fixed in the UART transmission mode, ■: Set to the initial value.
0/1: Set to 0 or 1 depending on the user's application.

2. Operation procedure

Figure 15 - 77 Initial Setting Procedure for UART Transmission

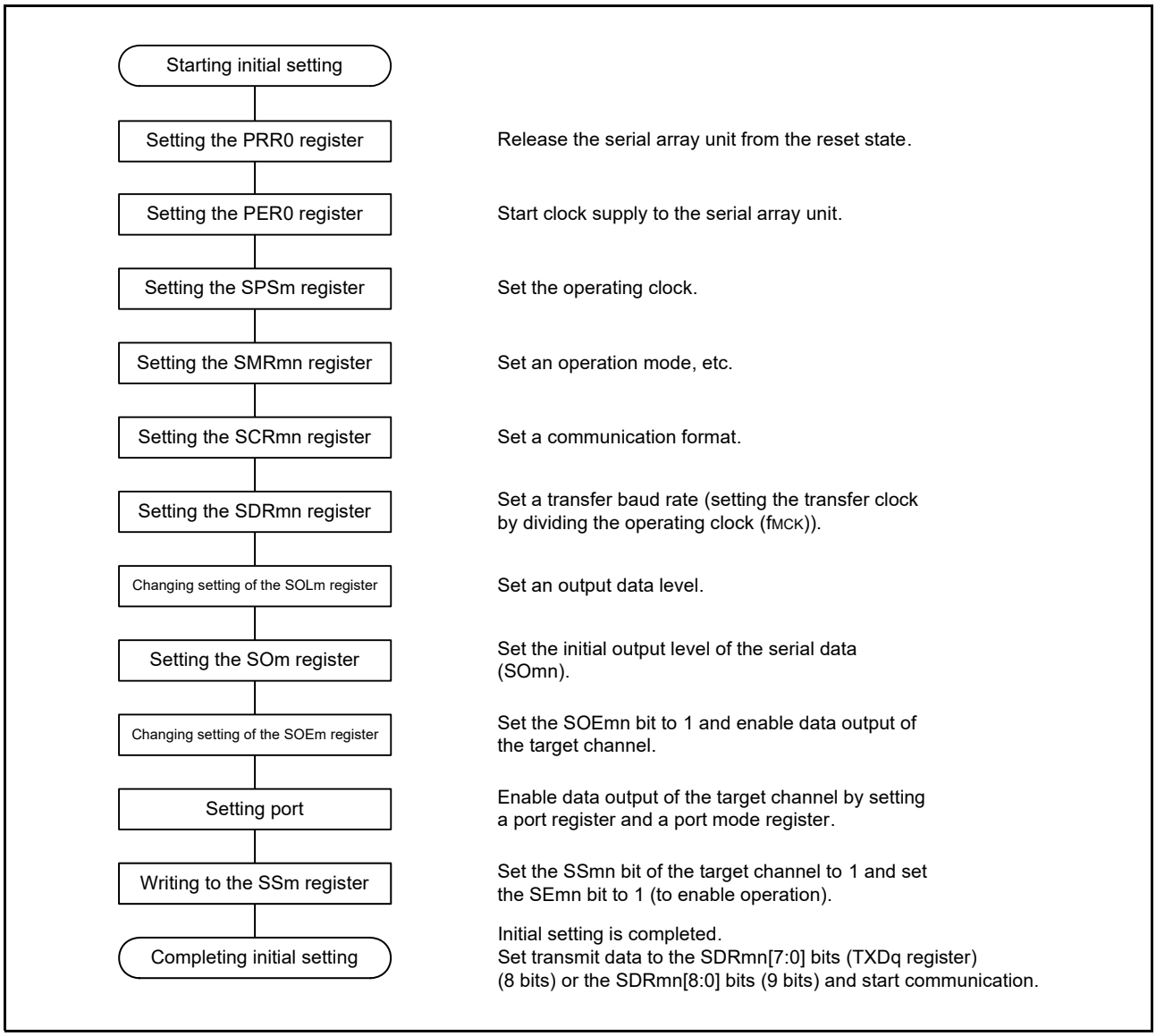


Figure 15 - 78 Procedure for Stopping UART Transmission

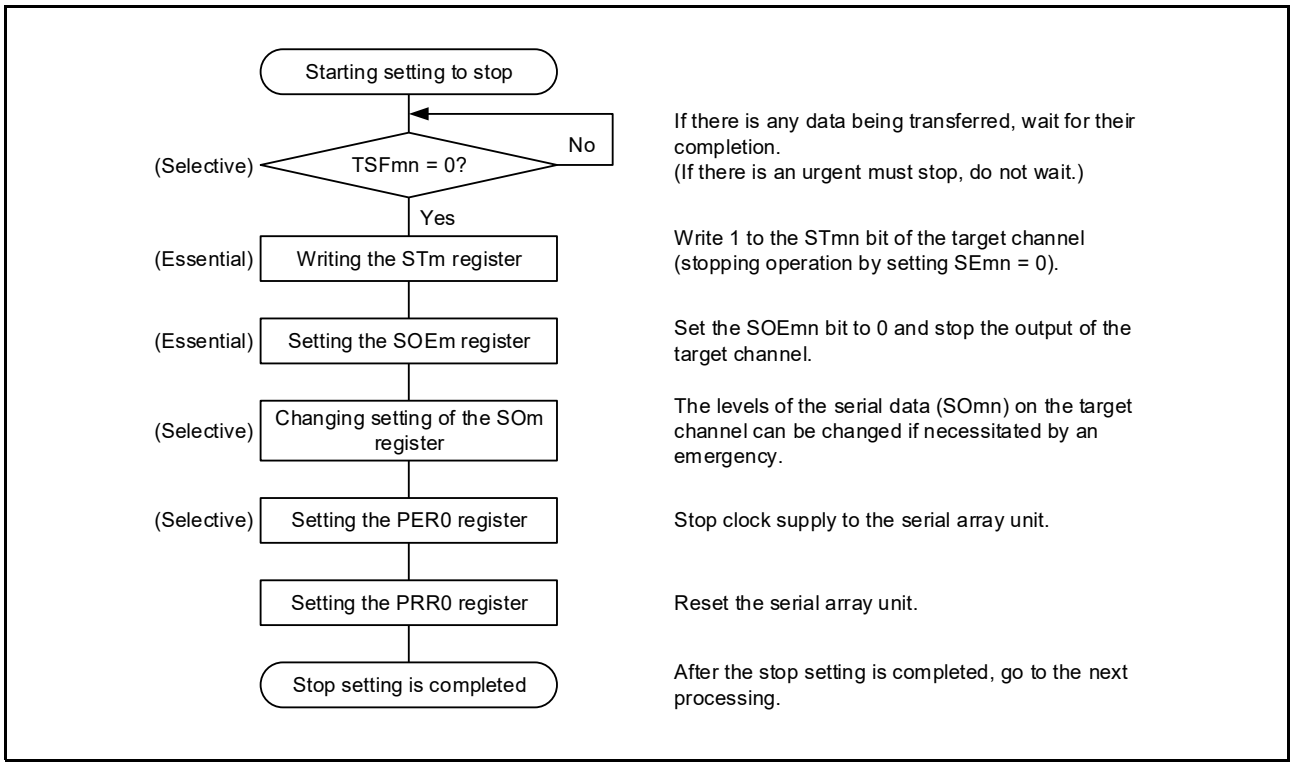
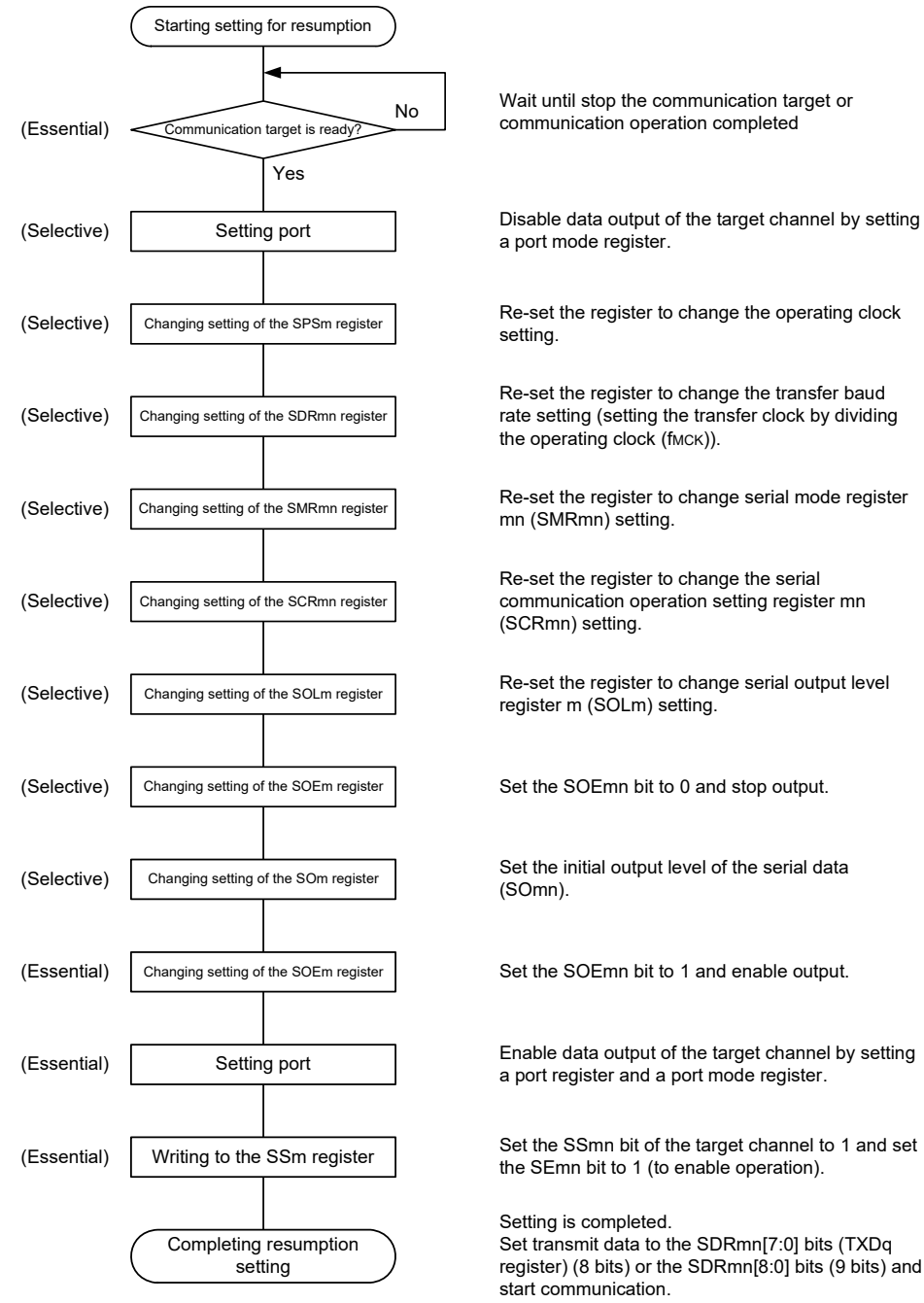


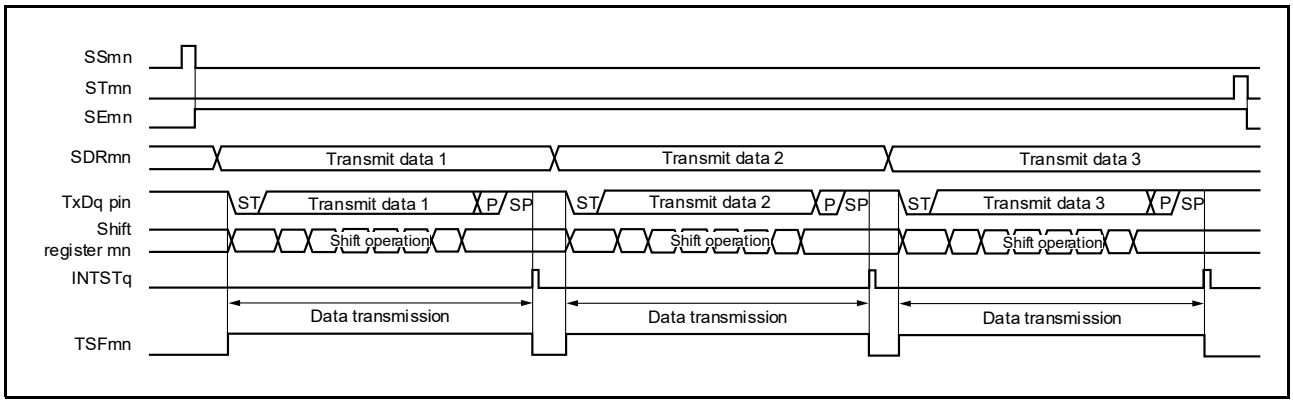
Figure 15 - 79 Procedure for Resuming UART Transmission



Remark If the PRR0 register is rewritten while stopping the communication to reset the serial array unit, wait until the communication target stops or communication finishes, and then perform initialization instead of restarting the communication.

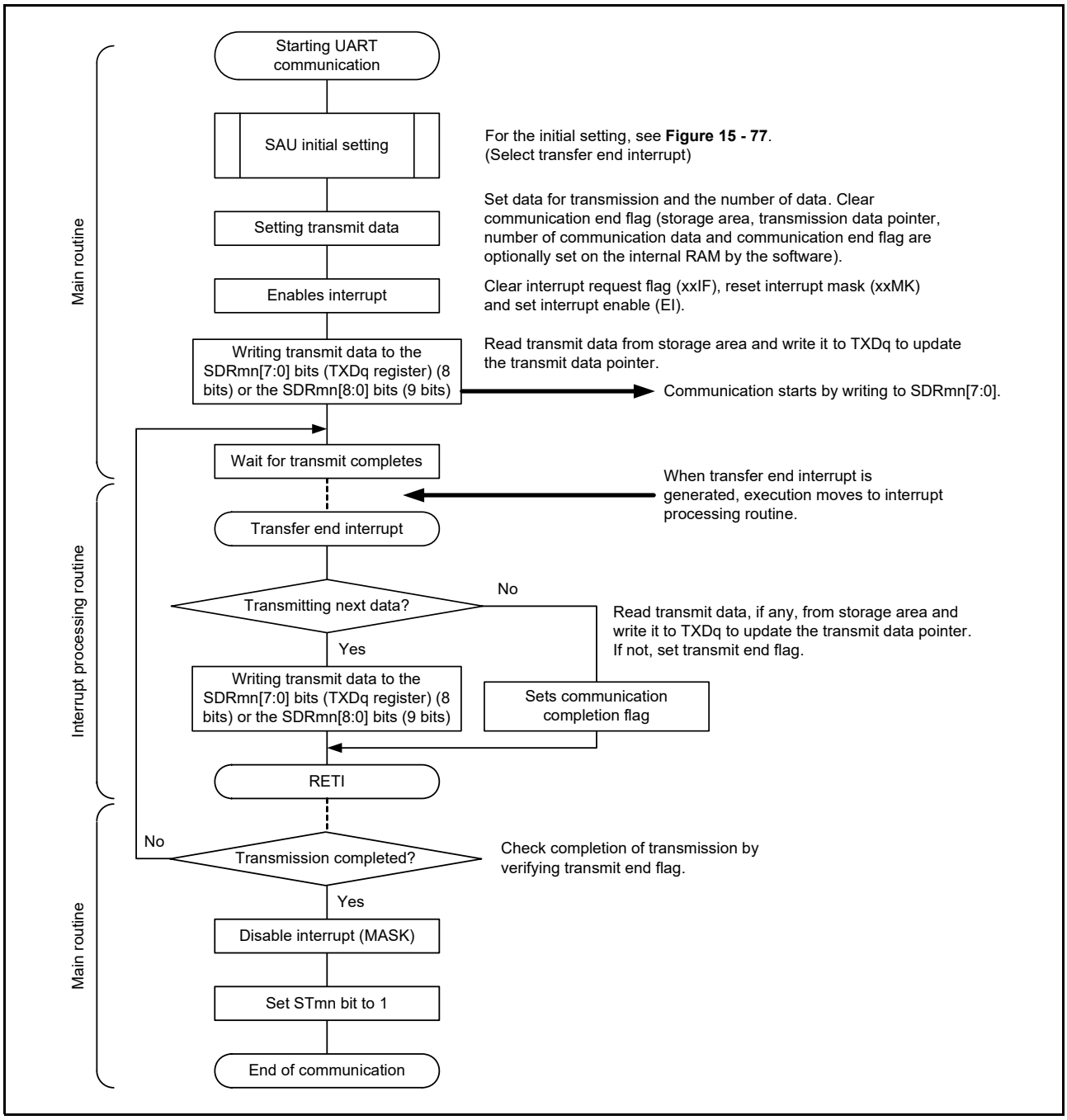
3. Processing flow (in single-transmission mode)

Figure 15 - 80 Timing Chart of UART Transmission (in Single-transmission Mode)



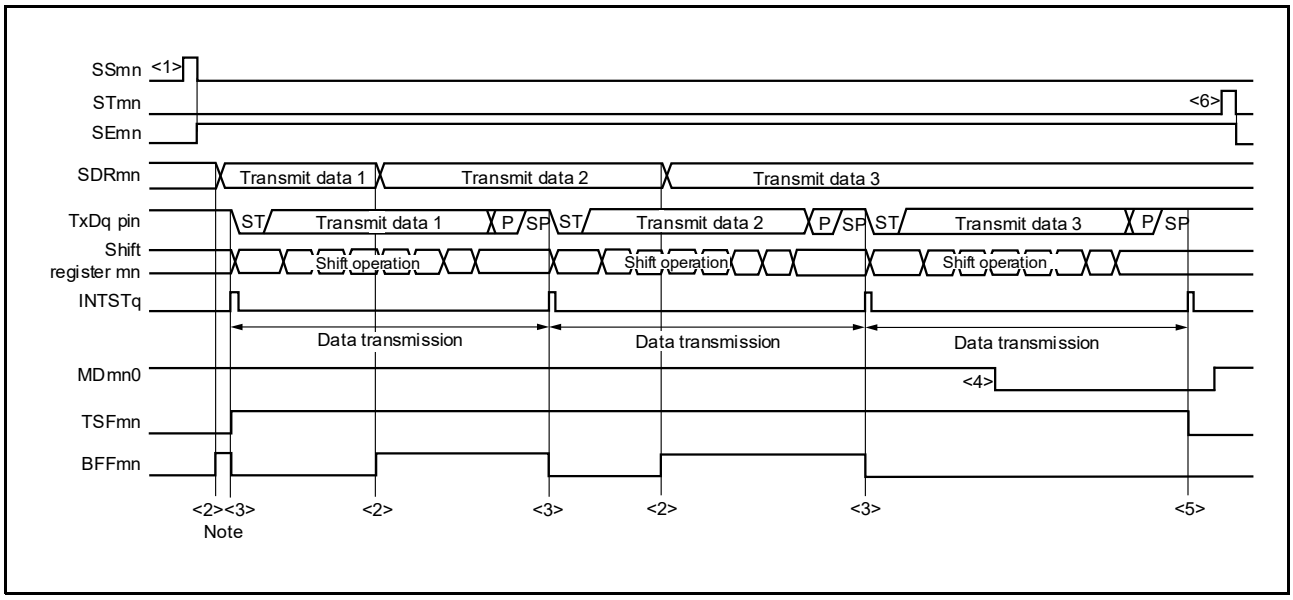
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 3), mn = 00, 02, 10, 12

Figure 15 - 81 Flowchart of UART Transmission (in Single-transmission Mode)



4. Processing flow (in continuous transmission mode)

Figure 15 - 82 Timing Chart of UART Transmission (in Continuous Transmission Mode)

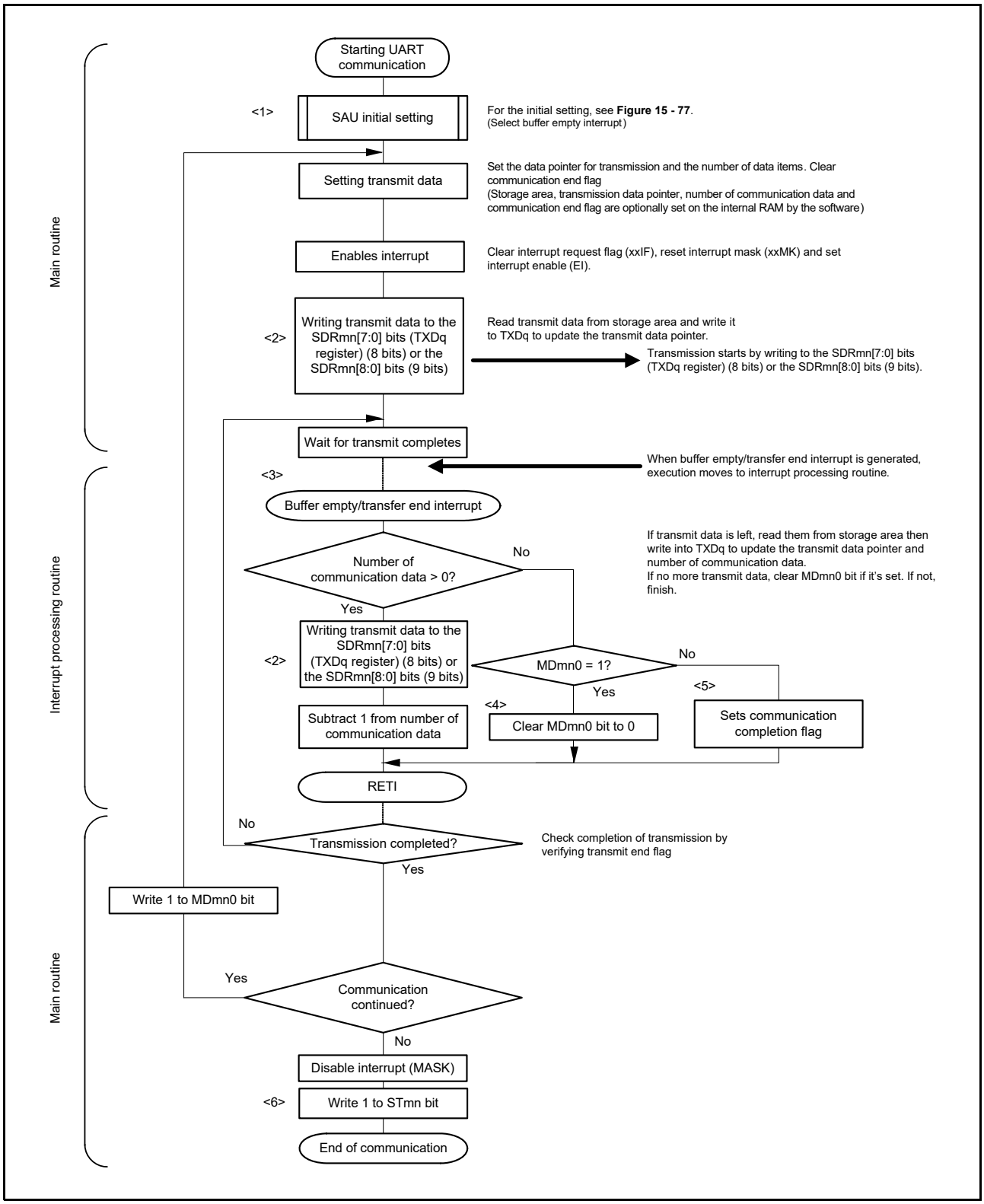


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

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 has been rewritten before the transfer end interrupt of the last transmit data.**

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 3),
mn = 00, 02, 10, 12

Figure 15 - 83 Flowchart of UART Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 15 - 82 Timing Chart of UART Transmission (in Continuous Transmission Mode)**.

15.6.2 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	UART2	UART3
Target channel	Channel 1 of SAU0	Channel 3 of SAU0	Channel 1 of SAU1	Channel 3 of SAU1
Pins used	RxD0	RxD1	RxD2	RxD3
Interrupt	INTSR0	INTSR1	INTSR2	INTSR3
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error interrupt	INTSRE0	INTSRE1	INTSRE2	INTSRE3
Error detection flag	• Framing error detection flag (FEFmn) • Parity error detection flag (PEFmn) • Overrun error detection flag (OVFmn)			
Transfer data length	7, 8, or 9 bits**Note 1**			
Transfer rate**Note 2**	Max. f_{MCK}/6 [bps] (SDRmn[15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]			
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)			
Parity bit	The following selectable • No parity bit (no parity check) • No parity judgment (0 parity) • Even parity check • Odd parity check			
Stop bit	Appending 1 bit			
Data direction	MSB or LSB first			

Note 1. Only the following UARTs support the 9-bit data length.
- UART0 in the 30- to 64-pin R7F100G and 32-pin R7F104G products
- UART0 and UART2 in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 37 Electrical Characteristics**.

Remark 1. f_{MCK}: Operating clock frequency of target channel
f_{CLK}: CPU/peripheral hardware clock frequency

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11, 13

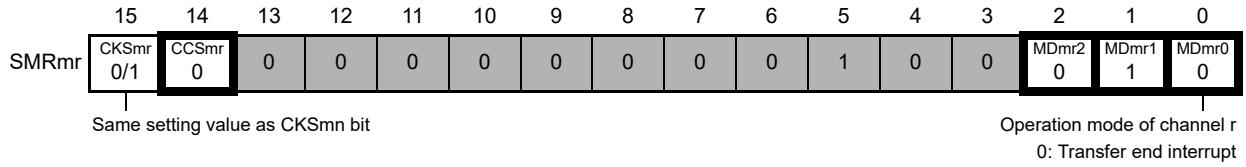
1. Register setting

Figure 15 - 84 Example of Contents of Registers for UART Reception of UART (UART0 to UART3)

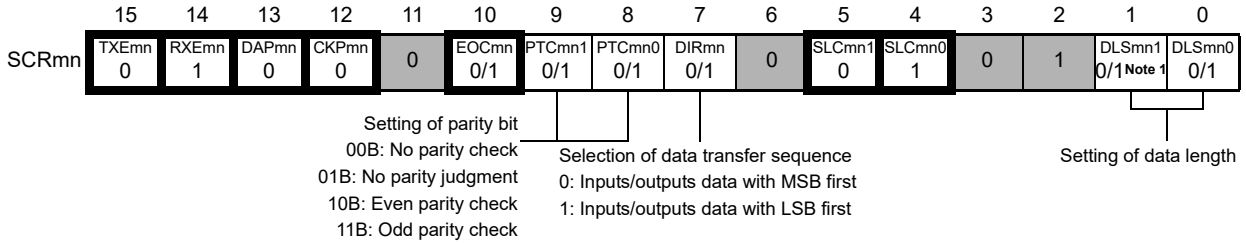
a) Serial mode register mn (SMRmn)



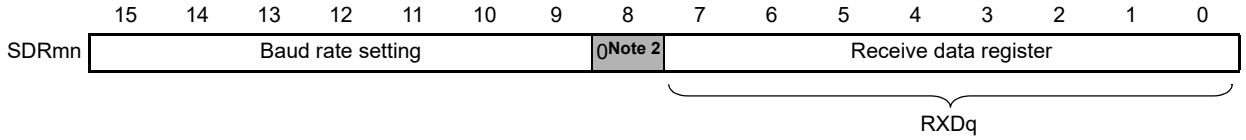
b) Serial mode register mr (SMRmr)



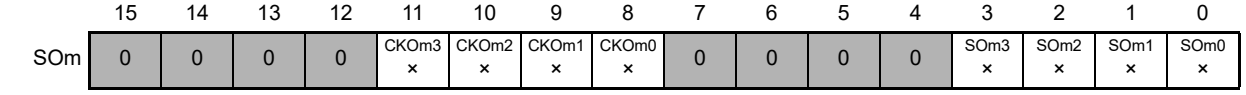
c) Serial communication operation setting register mn (SCRmn)



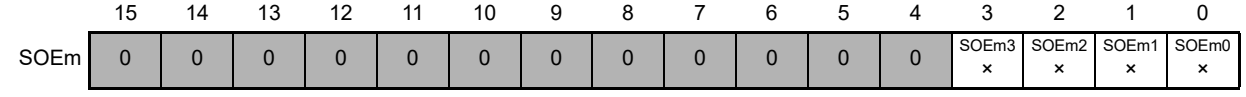
d) Serial data register mn (SDRmn) (lower 8 bits: RXDq)



e) Serial output register m (SOm): This register is not used in this mode.



f) Serial output enable register m (SOEm): This register is not used in this mode.



g) Serial channel start register m (SSm): Set only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 ×	SSm1 0/1	SSm0 ×

Note 1. This bit is only present in the following registers; in other registers, it is fixed to 1.
- SCR01 register
- SCR11 register for the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products

Note 2. When UART performs 9-bit communication, bits 0 to 8 of the SDRm1 register are used as the reception data specification area. Only the following UARTs support the 9-bit data length.
- UART0 in the 30- to 64-pin R7F100G and 32-pin R7F104G products
- UART0 and UART2 in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products

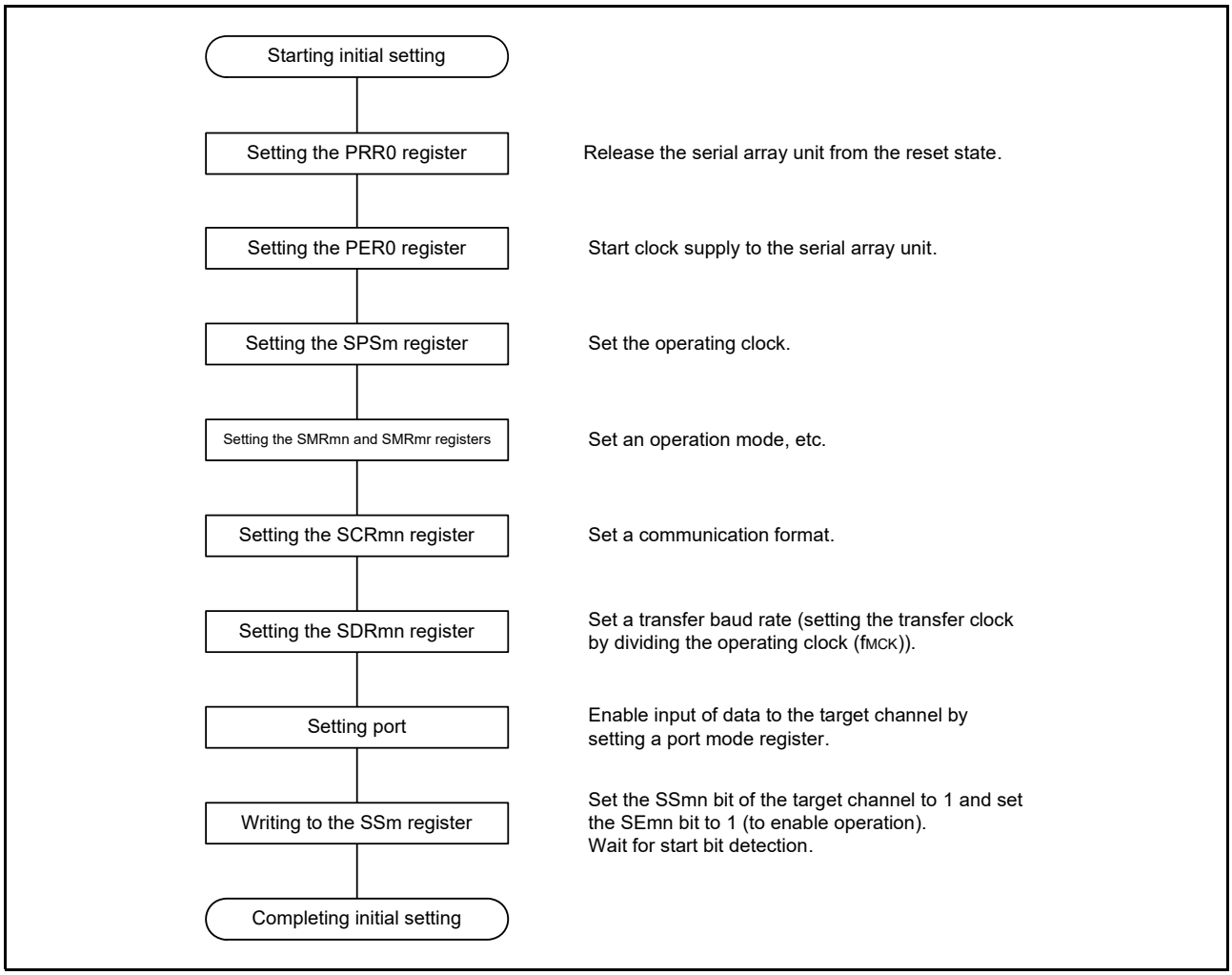
Caution **For the UART reception, be sure to set the SMRmr register of channel r to UART transmission mode that is to be paired with channel n.**

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), r: Channel number (r = n – 1), q: UART number (q = 0 to 3)
mn = 01, 03, 11, 13

Remark 2. ■: Setting is fixed in the UART reception mode, ■: Set to the initial value.
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the user's application.

2. Operation procedure

Figure 15 - 85 Initial Setting Procedure for UART Reception



Caution Set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after at least 4 fMCK clock cycles have elapsed.

Figure 15 - 86 Procedure for Stopping UART Reception

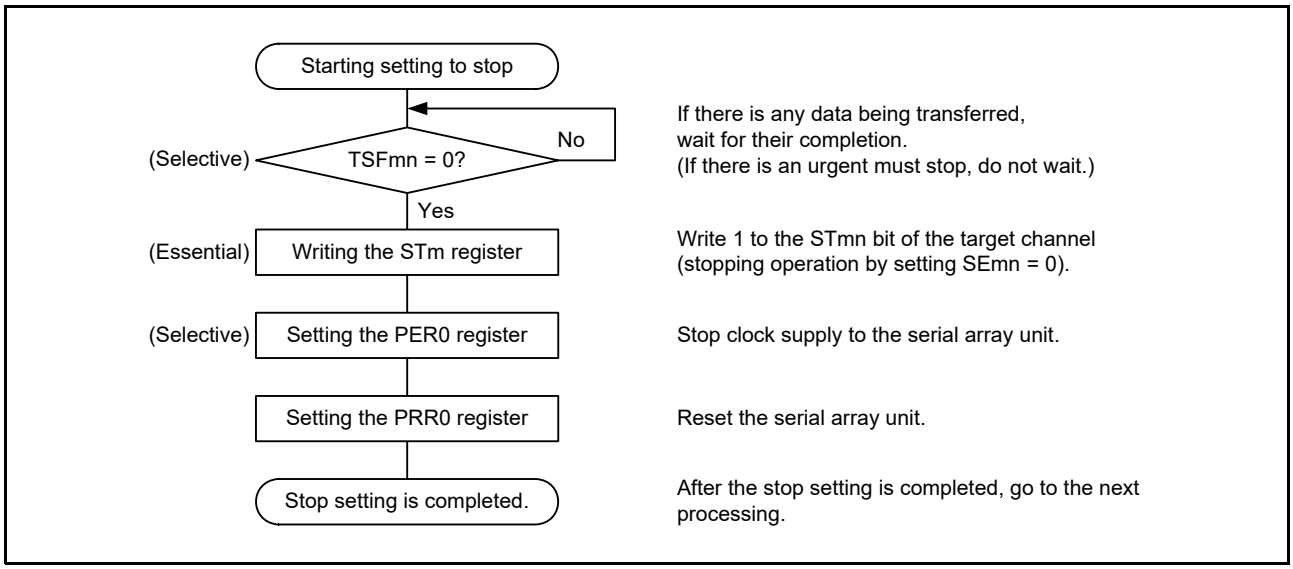
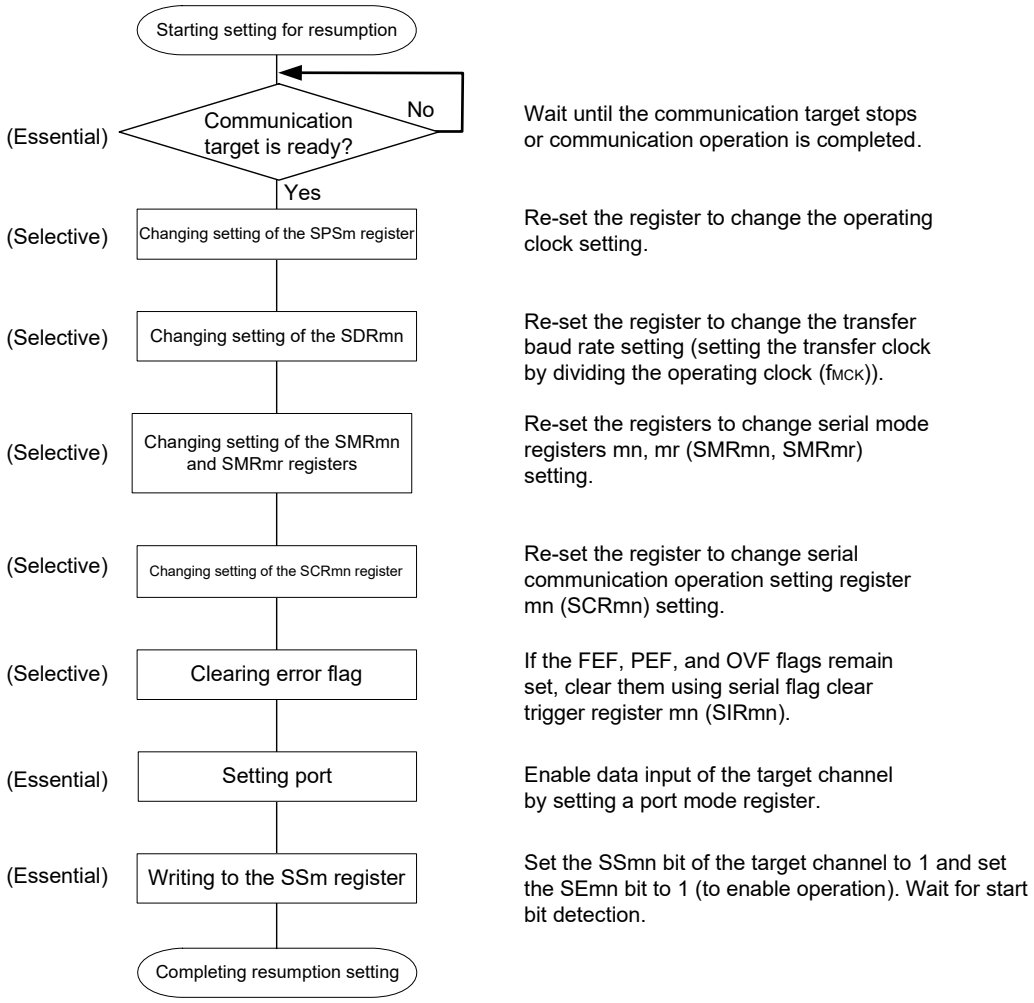


Figure 15 - 87 Procedure for Resuming UART Reception

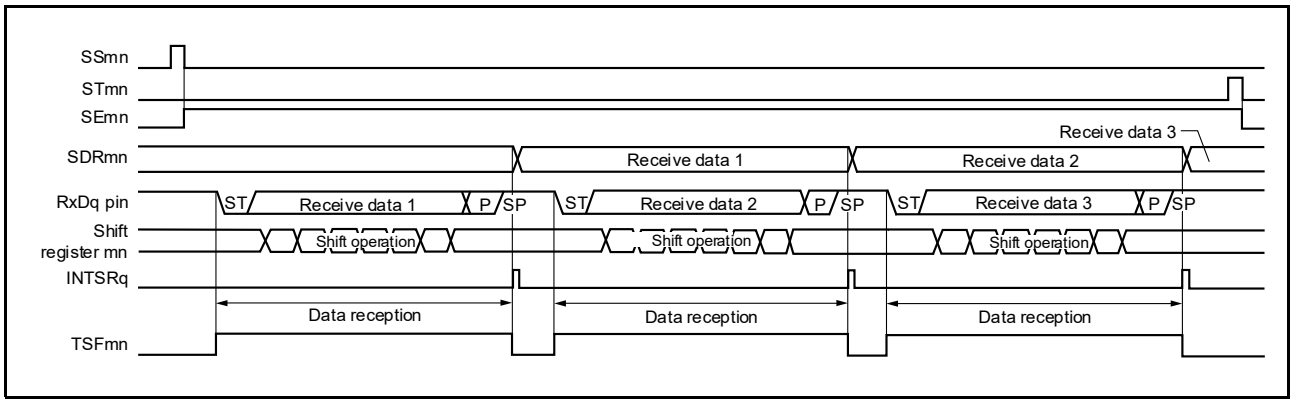


Caution **Set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after at least 4 fMCK clocks have elapsed.**

Remark If the PRR0 register is rewritten while stopping the communication to reset the serial array unit, wait until the communication target stops or communication finishes, and then perform initialization instead of restarting the communication.

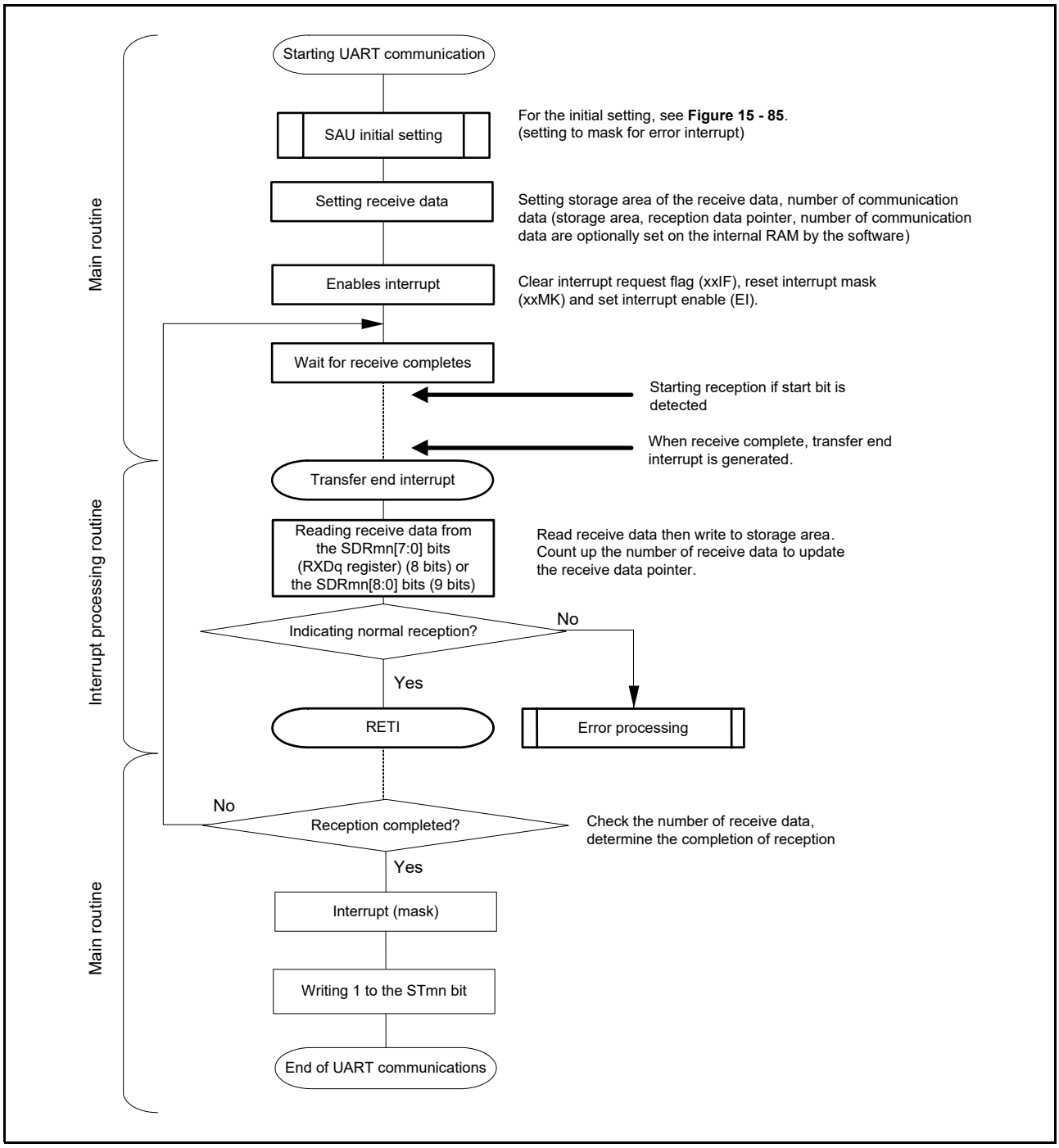
3. Processing flow

Figure 15 - 88 Timing Chart of UART Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), r: Channel number (r = n – 1), q: UART number (q = 0 to 3)
mn = 01, 03, 11, 13

Figure 15 - 89 Flowchart of UART Reception



15.6.3 SNOOZE mode

The SNOOZE mode makes the UART perform reception operations upon RxDq pin input detection while in the STOP mode. Normally the UART stops communication in the STOP mode. However, using the SNOOZE mode enables the UART to perform reception operations without CPU operation on detection of input on the RxDq pin. Only the following UARTs can be set up for reception in the SNOOZE mode.

- UART0 in the 30- to 64-pin R7F100G and 32-pin R7F104G products
- UART0 and UART2 in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products

When using UARTq in the SNOOZE mode, make the following settings before entering the STOP mode. See **Figure 15 - 92 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)** and **Figure 15 - 94 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)**.

- In the SNOOZE mode, the baud rate setting for UART reception needs to be changed to a value different from that in normal operation. Set the SPSm register and bits 15 to 9 of the SDRmn register with reference to **Table 15 - 5**.
- Set the EOCmn and SSECm bits. This is for enabling or stopping generation of an error interrupt (INTSREq) when a communication error occurs.
- When using the SNOOZE mode function, set the SWCm bit of serial standby control register m (SSCm) to 1 just before switching to the STOP mode. After the initial setting has been completed, set the SSm1 bit of serial channel start register m (SSm) to 1.
- A UARTq starts reception in SNOOZE mode on detecting input of the start bit on the RxDq pin following a transition of the CPU to the STOP mode.

Caution 1. The SNOOZE mode can only be used when the high-speed on-chip oscillator clock or medium-speed on-chip oscillator clock is selected for f_{CLK}. When the medium-speed on-chip oscillator clock is selected, use the medium-speed on-chip oscillator trimming register (MIOTRM) to correct the accuracy of the oscillation frequency.

Caution 2. The maximum transfer rate in the SNOOZE mode is 115.2 kbps (when setting FWKUP = 1, $f_{CLK} = f_{IH}$ (32 MHz)). When FWKUP is set to 1, f_{CLK} cannot be set to a value other than f_{IH} = 32 MHz.

Caution 3. When SWCm = 1, UARTq can be used only when the reception operation is started in the STOP mode. When used simultaneously with another SNOOZE mode function or interrupt, if the reception operation is started in a state other than the STOP mode, such as those given below, data may not be received correctly and a framing error or parity error may be generated.

- **When after the SWCm bit has been set to 1, the reception operation is started before the STOP mode is entered**
- **When the reception operation is started while another function is in the SNOOZE mode**
- **When after returning from the STOP mode to normal operation due to an interrupt or other cause, the reception operation is started before the SWCm bit is returned to 0**

Caution 4. If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFmn, FEFmn, or OVFmn flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFmn, FEFmn, and OVFmn flags before setting the SWCm bit to 1 and read the value in bits 7 to 0 (RXDq register) of the SDRm1 register.

Caution 5. The CPU shifts from the STOP mode to the SNOOZE mode on detecting the valid edge of the RxDq signal. Note, however, that reception through the UART channel may not start and the CPU may remain in the SNOOZE mode if an input pulse on the RxDq pin is too short to be detected as a start bit. In such cases, data may not be received correctly, and this may lead to a framing error or parity error in the next UART reception.

Remark 30- to 64-pin R7F100G and 32-pin R7F104G products
m: Unit number (m = 0), n: Channel number (n = 1), q = UART number (q = 0)
80- to 128-pin R7F100G and 44- and 48-pin R7F104G products
m: Unit number (m = 0, 1), n: Channel number (n = 1), q = UART number (q = 0, 2)

Table 15 - 5 Baud Rate Setting for UART Reception in SNOOZE Mode

When starting of the high-speed on-chip oscillator is at normal speed (WKUPMD.FWKUP = 0)

Baud Rate	High-speed On-chip Oscillator (f_{IH})	Operating Clock (f_{MCK})	SDRmn[15:9]	Maximum Permissible Value	Minimum Permissible Value
4800 bps	32 MHz ±1%**Note**	$f_{CLK}/2^5$	106	1.45%	-1.67%
	24 MHz ±1%**Note**	$f_{CLK}/2^5$	79	1.77%	-1.37%
9600 bps	32 MHz ±1%**Note**	$f_{CLK}/2^4$	106	1.45%	-1.67%
	24 MHz ±1%**Note**	$f_{CLK}/2^4$	79	1.77%	-1.37%

When starting of the high-speed on-chip oscillator is at high speed (WKUPMD.FWKUP = 1)

Baud Rate	High-speed On-chip Oscillator (f_{IH})	Operating Clock (f_{MCK})	SDRmn[15:9]	Maximum Permissible Value	Minimum Permissible Value
4800 bps	32 MHz ±1%**Note**	$f_{CLK}/2^5$	106	1.45%	-1.67%
9600 bps		$f_{CLK}/2^4$	106	1.45%	-1.67%
19200 bps		$f_{CLK}/2^3$	106	1.45%	-1.67%
31250 bps		$f_{CLK}/2^3$	65	1.05%	-2.06%
38400 bps		$f_{CLK}/2^2$	106	1.45%	-1.67%
76800 bps		$f_{CLK}/2$	106	1.45%	-1.67%
115200 bps		$f_{CLK}/2$	70	1.93%	-1.21%

Note When the accuracy of the clock frequency of the high-speed on-chip oscillator is ±1.5% or ±2.0%, the permissible range becomes smaller as shown below.

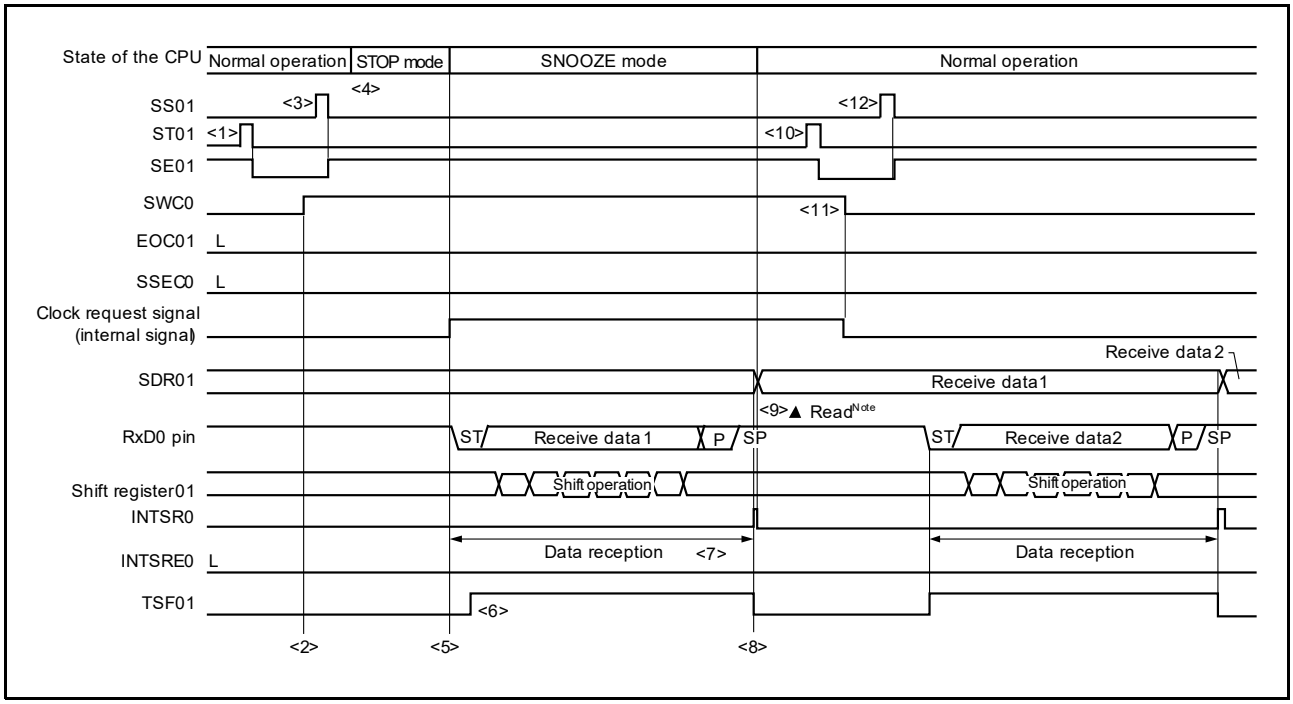
- In the case of f_{IH} ±1.5%, perform (Maximum permissible value – 0.5%) and (Minimum permissible value + 0.5%) to the values in the above table.
- In the case of f_{IH} ±2.0%, perform (Maximum permissible value – 1.0%) and (Minimum permissible value + 1.0%) to the values in the above table.

Remark 1. The maximum permissible value and minimum permissible value are permissible values for the baud rate in UART reception. The baud rate on the transmitting side should be set to fall inside this range.

Remark 2. mn = 01, 11

1. SNOOZE mode operation (EOCm1 = 0, SSECm = 0/1)
   Because of the setting of EOCm1 = 0, even though a communication error occurs, an error interrupt (INTSREq) is not generated, regardless of the setting of the SSECm bit. A transfer end interrupt (INTSRq) will be generated.

Figure 15 - 90 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)



Note Read the received data when SWCm = 1.

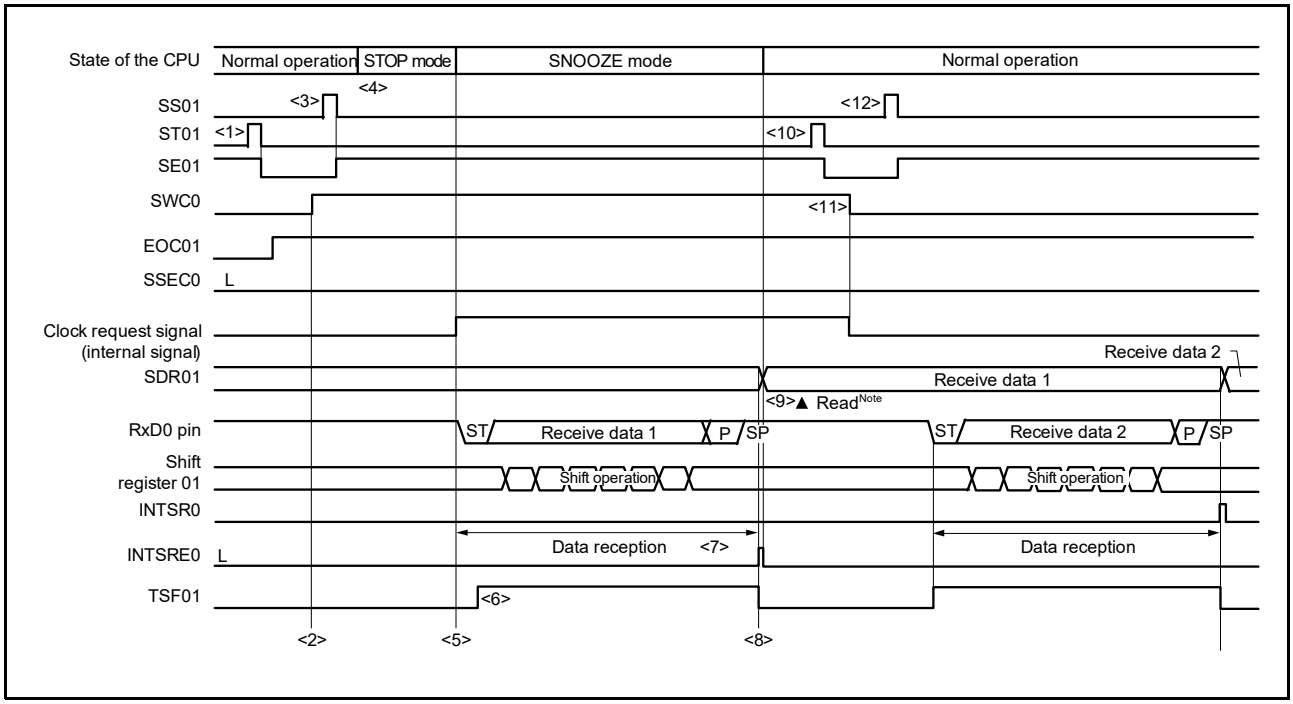
Caution **Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (the SEm1 bit is cleared and the operation stops). After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).**

Remark 1. <1> to <12> in the figure correspond to <1> to <12> in **Figure 15 - 92 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)**.

Remark 2. 30- to 64-pin R7F100G and 32-pin R7F104G products
m: Unit number (m = 0), q: UART number (p = 0)
80- to 128-pin R7F100G and 44- and 48-pin R7F104G products
m: Unit number (m = 0, 1), q: UART number (p = 0, 2)

2. SNOOZE mode operation (EOCm1 = 1, SSECm = 0: Error interrupt (INTSREq) generation is enabled)
   Because EOCm1 = 1 and SSECm = 0, an error interrupt (INTSREq) is generated when a communication error occurs.

Figure 15 - 91 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)



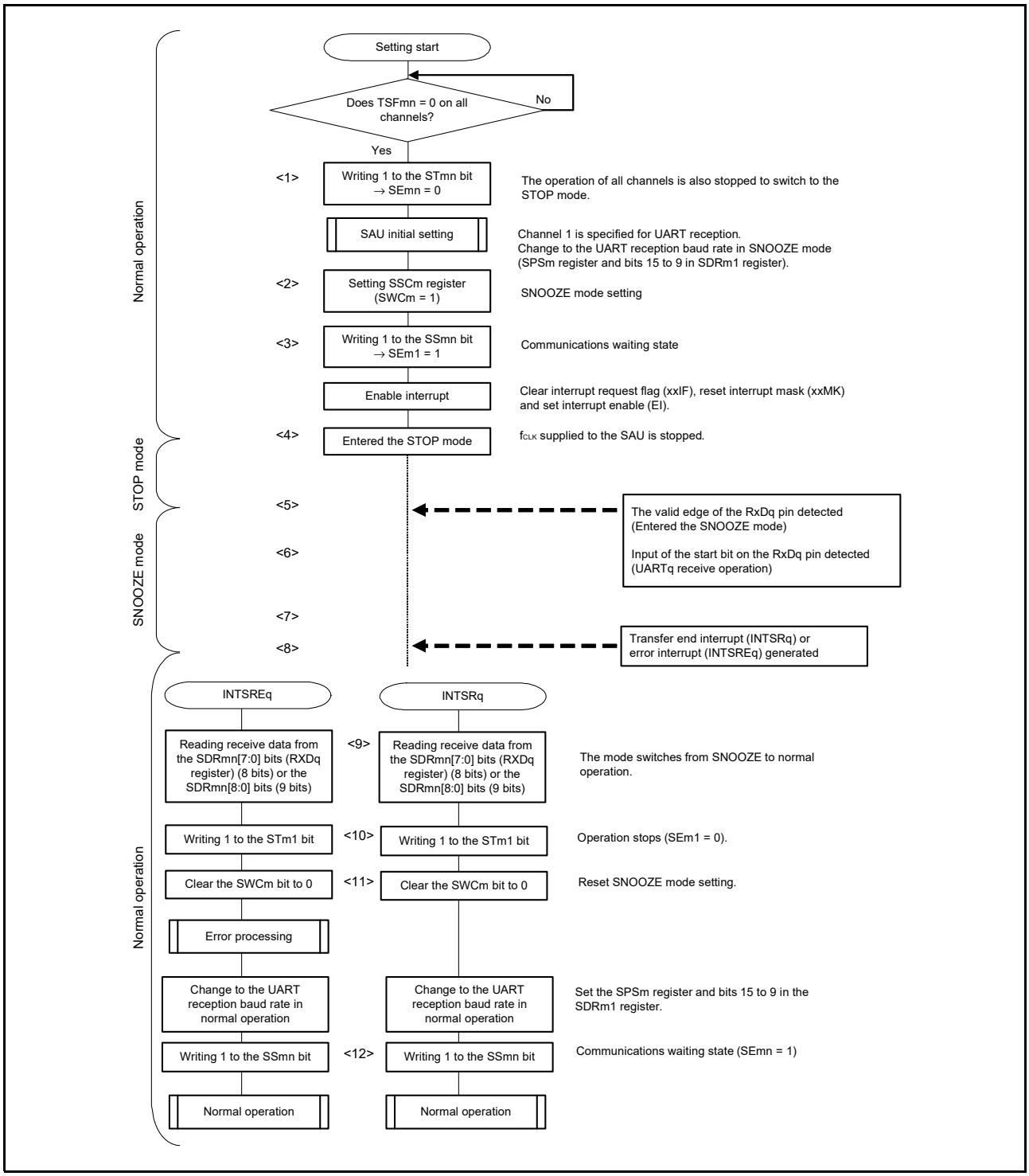
Note Read the received data when SWCm = 1.

Caution **Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (the SEm1 bit is cleared and the operation stops). After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).**

Remark 1. <1> to <12> in the figure correspond to <1> to <12> in **Figure 15 - 92 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)**.

Remark 2. 30- to 64-pin R7F100G and 32-pin R7F104G products
m: Unit number (m = 0), q: UART number (p = 0)
80- to 128-pin R7F100G and 44- and 48-pin R7F104G products
m: Unit number (m = 0, 1), q: UART number (p = 0, 2)

Figure 15 - 92 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)

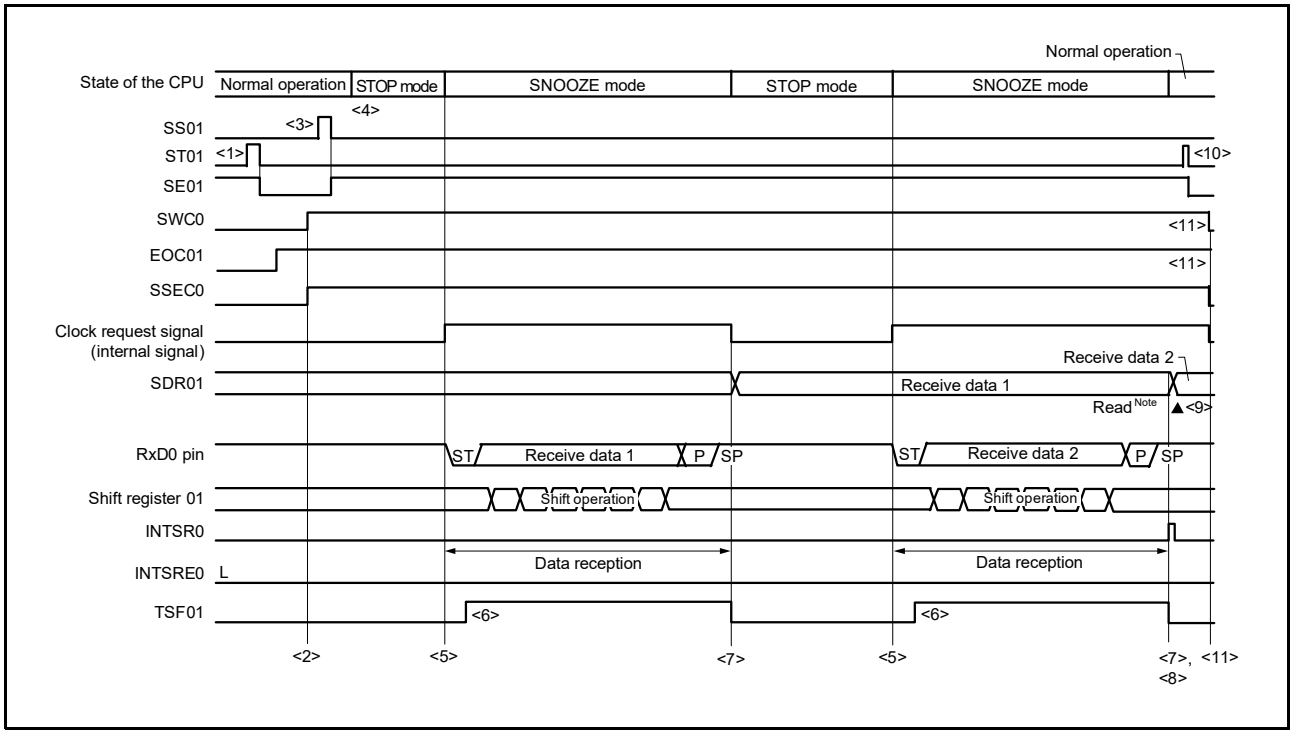


Remark 1. <1> to <12> in the figure correspond to <1> to <12> in **Figure 15 - 90 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)** and **Figure 15 - 91 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)**.

Remark 2. 30- to 64-pin R7F100G and 32-pin R7F104G products
m: Unit number (m = 0), n: Channel number (n = 0 to 3), q: UART number (p = 0)
80- to 128-pin R7F100G and 44- and 48-pin R7F104G products
m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), q: UART number (p = 0, 2)

3. SNOOZE mode operation (EOCm1 = 1, SSECm = 1: Error interrupt (INTSREq) generation is stopped)
   Because EOCm1 = 1 and SSECm = 1, an error interrupt (INTSREq) is not generated when a communication error occurs.

Figure 15 - 93 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)



Note Read the received data when SWCm = 1.

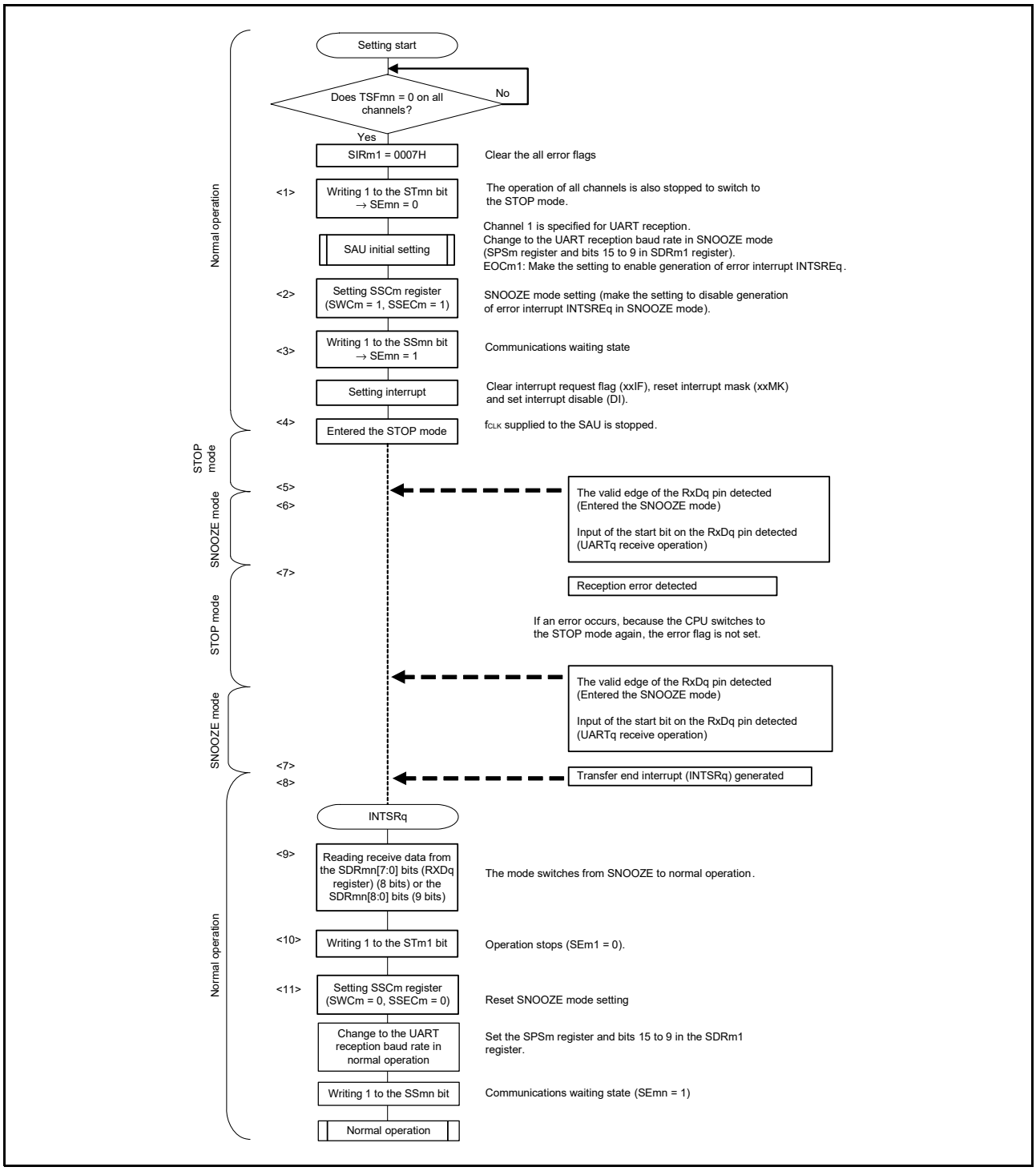
Caution 1. **Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (the SEm1 bit is cleared and the operation stops). After the receive operation completes, also clear the SWCm bit to 0 (SNOOZE mode release).**

Caution 2. **If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFm1, FEFm1, or OVFm1 flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFm1, FEFm1, and OVFm1 flags before setting the SWCm bit to 1 and read the value in SDRm1[7:0] (RXDq register) (8 bits) or SDRm1[8:0] (9 bits).**

Remark 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 15 - 94 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)**.

Remark 2. 30- to 64-pin R7F100G and 32-pin R7F104G products
m: Unit number (m = 0), q: UART number (p = 0)
80- to 128-pin R7F100G and 44- and 48-pin R7F104G products
m: Unit number (m = 0, 1), q: UART number (p = 0, 2)

Figure 15 - 94 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)



Caution **If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFm1, FEFm1, or OVFm1 flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFm1, FEFm1, and OVFm1 flags before setting the SWCm bit to 1 and read the value in SDRm1[7:0] (RXDq register) (8 bits) or SDRm1[8:0] (9 bits).**

Remark 1. <1> to <11> in the figure correspond to <1> to <11> in **Figure 15 - 93 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)**.

Remark 2. 30- to 64-pin R7F100G and 32-pin R7F104G products
m: Unit number (m = 0), n: Channel number (n = 0 to 3), q: UART number (p = 0)
80- to 128-pin R7F100G and 44- and 48-pin R7F104G products
m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), q: UART number (p = 0, 2)

15.6.4 Calculating baud rate

1. Baud rate calculation expression

The baud rate for UART (UART0 to UART3) communication can be calculated by the following expressions.

(Baud rate) = {Operating 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.

Remark 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.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

The operating clock (f_{MCK}) is determined by serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn).

Table 15 - 6 Selection of Operating Clock for UART

SMRmn Register	SPSm Register								Operating Clock (f_{MCK})[Note]	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 32 MHz
0	×	×	×	×	0	0	0	0	f_{CLK}	32 MHz
	×	×	×	×	0	0	0	1	$f_{CLK}/2$	16 MHz
	×	×	×	×	0	0	1	0	$f_{CLK}/2^2$	8 MHz
	×	×	×	×	0	0	1	1	$f_{CLK}/2^3$	4 MHz
	×	×	×	×	0	1	0	0	$f_{CLK}/2^4$	2 MHz
	×	×	×	×	0	1	0	1	$f_{CLK}/2^5$	1 MHz
	×	×	×	×	0	1	1	0	$f_{CLK}/2^6$	500 kHz
	×	×	×	×	0	1	1	1	$f_{CLK}/2^7$	250 kHz
	×	×	×	×	1	0	0	0	$f_{CLK}/2^8$	125 kHz
	×	×	×	×	1	0	0	1	$f_{CLK}/2^9$	62.5 kHz
	×	×	×	×	1	0	1	0	$f_{CLK}/2^{10}$	31.25 kHz
	×	×	×	×	1	0	1	1	$f_{CLK}/2^{11}$	15.63 kHz
	×	×	×	×	1	1	0	0	$f_{CLK}/2^{12}$	7.81 kHz
	×	×	×	×	1	1	0	1	$f_{CLK}/2^{13}$	3.91 kHz
	×	×	×	×	1	1	1	0	$f_{CLK}/2^{14}$	1.95 kHz
	×	×	×	×	1	1	1	1	$f_{CLK}/2^{15}$	977 Hz
1	0	0	0	0	×	×	×	×	f_{CLK}	32 MHz
	0	0	0	1	×	×	×	×	$f_{CLK}/2$	16 MHz
	0	0	1	0	×	×	×	×	$f_{CLK}/2^2$	8 MHz
	0	0	1	1	×	×	×	×	$f_{CLK}/2^3$	4 MHz
	0	1	0	0	×	×	×	×	$f_{CLK}/2^4$	2 MHz
	0	1	0	1	×	×	×	×	$f_{CLK}/2^5$	1 MHz
	0	1	1	0	×	×	×	×	$f_{CLK}/2^6$	500 kHz
	0	1	1	1	×	×	×	×	$f_{CLK}/2^7$	250 kHz
	1	0	0	0	×	×	×	×	$f_{CLK}/2^8$	125 kHz
	1	0	0	1	×	×	×	×	$f_{CLK}/2^9$	62.5 kHz
	1	0	1	0	×	×	×	×	$f_{CLK}/2^{10}$	31.25 kHz
	1	0	1	1	×	×	×	×	$f_{CLK}/2^{11}$	15.63 kHz
	1	1	0	0	×	×	×	×	$f_{CLK}/2^{12}$	7.81 kHz
	1	1	0	1	×	×	×	×	$f_{CLK}/2^{13}$	3.91 kHz
	1	1	1	0	×	×	×	×	$f_{CLK}/2^{14}$	1.95 kHz
	1	1	1	1	×	×	×	×	$f_{CLK}/2^{15}$	977 Hz
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) = 000FH) the operation of the serial array unit (SAU).

Remark 1. ×: Don't care

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

2. Baud rate error during transmission

The baud rate error of UART (UART0 to UART3) communication during transmission can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

(Baud rate error) = (Calculated baud rate value) ÷ (Target baud rate) × 100 - 100 [%]

Here is an example of setting a UART baud rate at f_{CLK} = 32 MHz.

UART Baud Rate (Target Baud Rate)	f_{CLK} = 32 MHz			
	Operating Clock (f_{MCK})	SDRmn[15:9]	Calculated Baud Rate	Error from Target Baud Rate
300 bps	$f_{CLK}/2^9$	103	300.48 bps	+0.16%
600 bps	$f_{CLK}/2^8$	103	600.96 bps	+0.16%
1200 bps	$f_{CLK}/2^7$	103	1201.92 bps	+0.16%
2400 bps	$f_{CLK}/2^6$	103	2403.85 bps	+0.16%
4800 bps	$f_{CLK}/2^5$	103	4807.69 bps	+0.16%
9600 bps	$f_{CLK}/2^4$	103	9615.38 bps	+0.16%
19200 bps	$f_{CLK}/2^3$	103	19230.8 bps	+0.16%
31250 bps	$f_{CLK}/2^3$	63	31250.0 bps	±0.0%
38400 bps	$f_{CLK}/2^2$	103	38461.5 bps	+0.16%
76800 bps	$f_{CLK}/2$	103	76923.1 bps	+0.16%
153600 bps	f_{CLK}	103	153846 bps	+0.16%
312500 bps	f_{CLK}	50	313725.5 bps	+0.39%

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10, 12

3. Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UART0 to UART3) communication can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

$$\text{(Maximum receivable baud rate)} = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times Brate$$

$$\text{(Minimum receivable baud rate)} = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times Brate$$

Brate: Calculated baud rate value at the reception side (See **15.6.4 Calculating baud rate**, **1. Baud rate calculation expression**.)

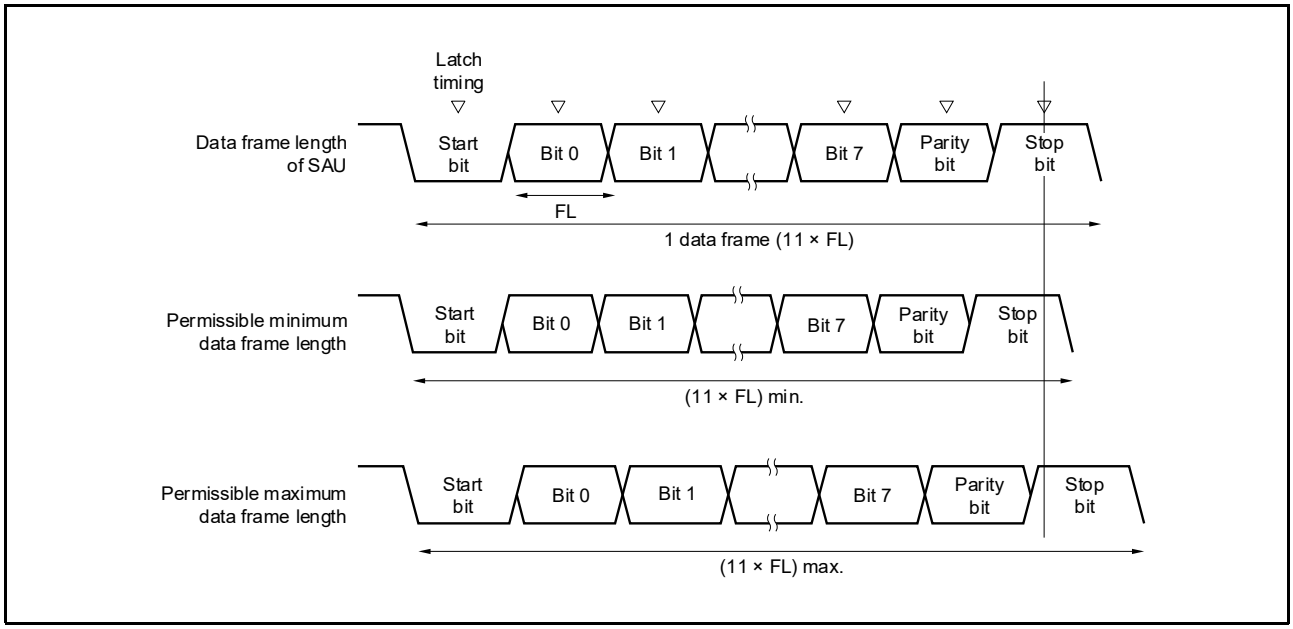
k: SDRmn[15:9] + 1

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, 3), mn = 01, 03, 11, 13

Figure 15 - 95 Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



The timing of latching receive data is determined by the division ratio set by bits 15 to 9 of serial data register mn (SDRmn) after the start bit is detected. If the last data (stop bit) is received before this latch timing, the data can be correctly received.

15.6.5 Procedure for processing errors that occurred during UART (UART0 to UART3) communication

Table 15 - 7 and **Table 15 - 8** describe the procedure for processing errors that occurred during UART (UART0 to UART3) communication.

Table 15 - 7 Processing Procedure in Case of Parity Error or Overrun Error

Software Manipulation	State of the Hardware	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		The error type is identified and the read value is used to clear the error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	The error flag is cleared.	Only the error generated during reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.

Table 15 - 8 Processing Procedure in Case of Framing Error

Software Manipulation	State of the Hardware	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		The error type is identified and the read value is used to clear the error flag.
Writes serial flag clear trigger register mn (SIRmn). →	The error flag is cleared.	Only the error generated during reading can be cleared, 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.	
Synchronization with other party of communication		Synchronization with the other party of communication is re-established and communication is resumed because it is considered that a framing error has occurred because the start bit has been shifted.
Sets the SSmn bit of serial channel start register m (SSm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate.	

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

15.7 LIN Communication Operation

15.7.1 LIN transmission

Of UART transmission, UART2 supports LIN communication. For LIN transmission, channel 0 of unit 1 is used.

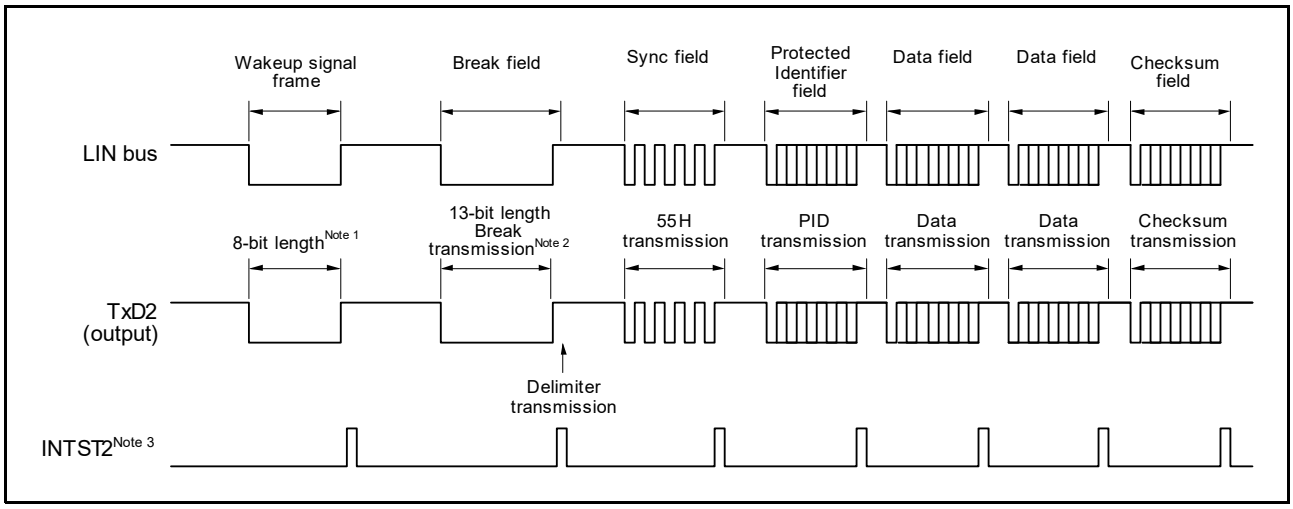
UART	UART0	UART1	UART2	UART3
Support of LIN communication	Not supported	Not supported	Supported	Not supported
Target channel	—	—	Channel 0 of SAU1	—
Pins used	—	—	TxD2	—
Interrupt	—	—	INTST2	—
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	None			
Transfer data length	8 bits			
Transfer rate**Note**	Max. f_{MCK}/6 [bps] (SDR10[15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]			
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)			
Parity bit	No parity bit			
Stop bit	Appending 1 bit			
Data direction	LSB first			

Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 37 Electrical Characteristics**. In general, 2.4, 9.6, or 19.2 kbps is often used in LIN communication.

Remark f_{MCK}: Operating clock frequency of target channel
f_{CLK}: CPU/peripheral hardware clock frequency

LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol designed to reduce the cost of an automobile network. Communication of LIN is single-master communication and up to 15 slaves can be connected to one master. The slaves are used to control switches, actuators, and sensors, which are connected to the master via LIN. Usually, the master is connected to a network such as CAN (Controller Area Network). A LIN bus is a single-wire bus to which nodes are connected via transceiver conforming to ISO9141. According to the protocol of LIN, the master transmits a frame by attaching baud rate information to it. A slave receives this frame and corrects a baud rate error from the master. If the baud rate error of a slave is within ±15%, communication can be established.
Figure 15 - 96 outlines a transmission operation of LIN.

Figure 15 - 96 Transmission Operation of LIN



Note 1. Set the baud rate in accordance with the wakeup signal regulations and transmit data of 80H.

Note 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.

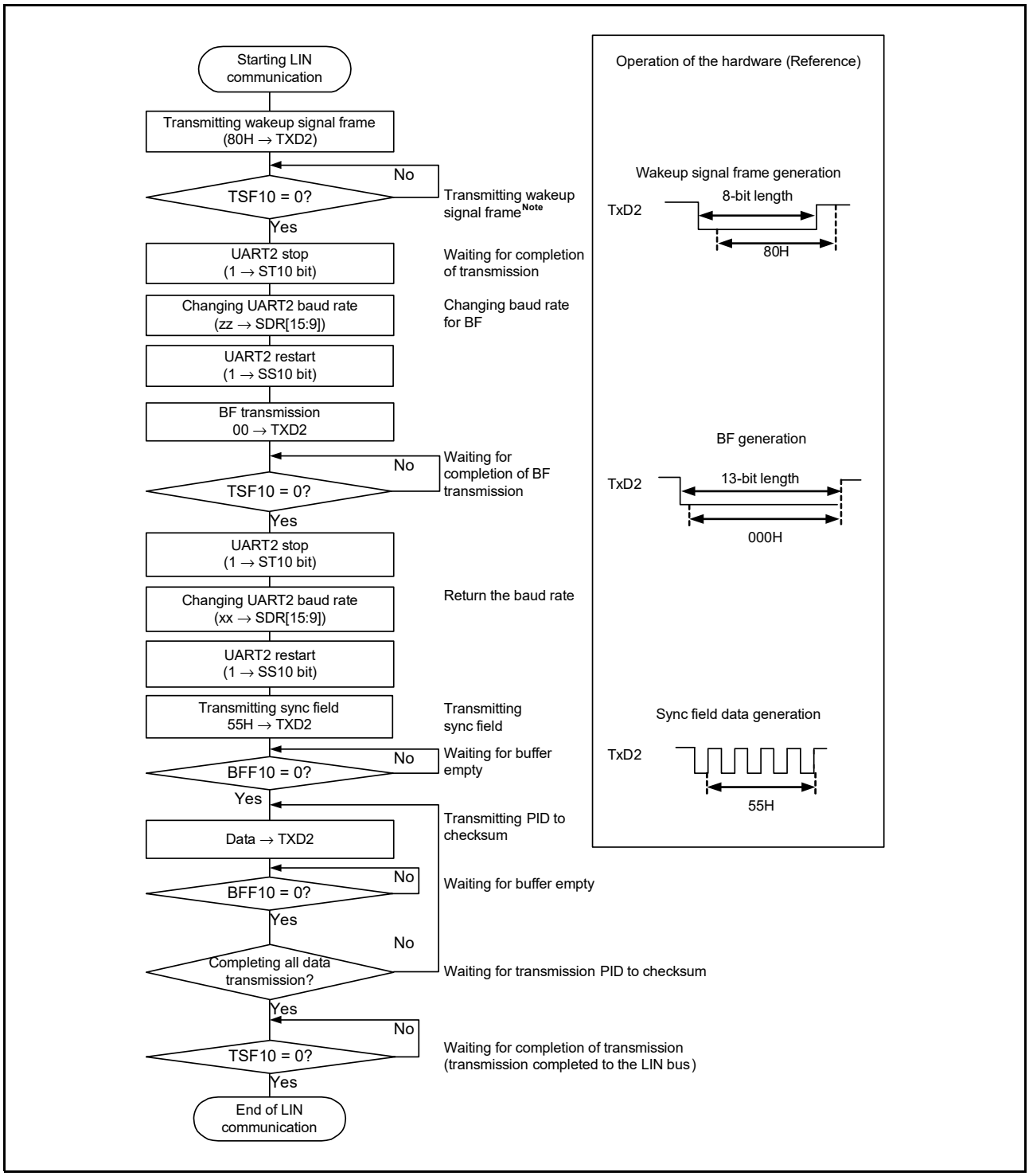
(Baud rate of break field) = 9/13 × N

By transmitting data of 00H at this baud rate, a break field is generated.

Note 3. INTST2 is output upon completion of transmission. INTST2 is also output at BF transmission.

Remark The interval between fields is controlled by software.

Figure 15 - 97 Flowchart for LIN Transmission



Note This is only required if the LIN bus is being started from sleep mode.

Remark This flow assumes that the initial setting of the UART is completed and transmission is enabled.

15.7.2 LIN reception

Of UART reception, UART2 of the 30- to 128-pin R7F100G and 32- to 48-pin R7F104G products support LIN communication. For LIN reception, channel 1 of unit 1 is used.

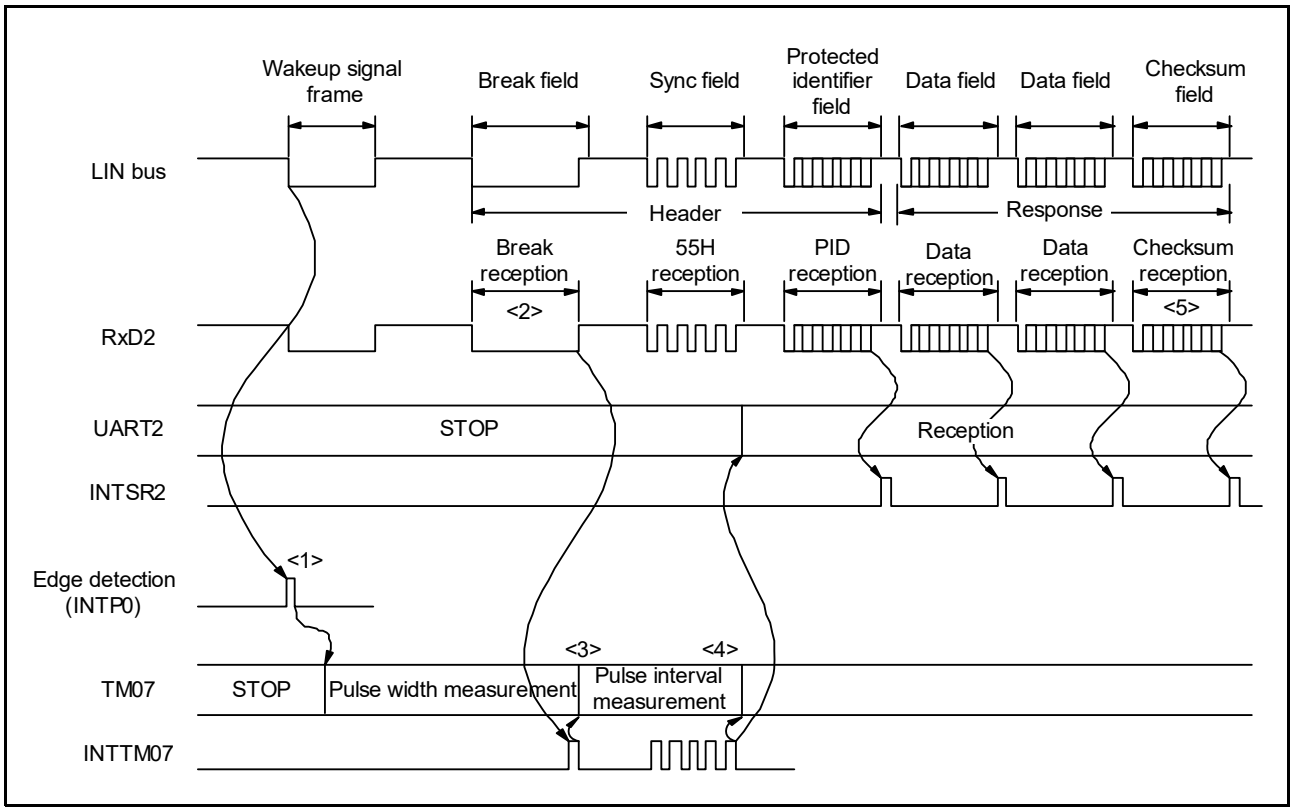
UART	UART0	UART1	UART2	UART3
Support of LIN communication	Not supported	Not supported	Supported	Not supported
Target channel	—	—	Channel 1 of SAU1	—
Pins used	—	—	RxD2	—
Interrupt	—	—	INTSR2	—
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error interrupt	—	—	INTSRE2	—
Error detection flag	• Framing Error detection flag (FEF11) • Overrun Error detection flag (OVF11)			
Transfer data length	8 bits			
Transfer rate**Note**	Max. f_{MCK}/6 [bps] (SDR11[15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]			
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)			
Parity bit	No parity bit (The parity bit is not checked.)			
Stop bit	Check the first bit			
Data direction	LSB first			

Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 37 Electrical Characteristics**.

Remark f_{MCK}: Operating clock frequency of target channel
f_{CLK}: CPU/peripheral hardware clock frequency

Figure 15 - 98 outlines a reception operation of LIN.

Figure 15 - 98 Reception Operation of LIN



Here is the flow of reception processing.

<1> The wakeup signal is detected by detecting an edge on the INTP0 interrupt pin. When the wakeup signal is detected, set TM07 to the pulse width measurement function to measure the low-level width of the BF signal. Then wait for BF signal reception.

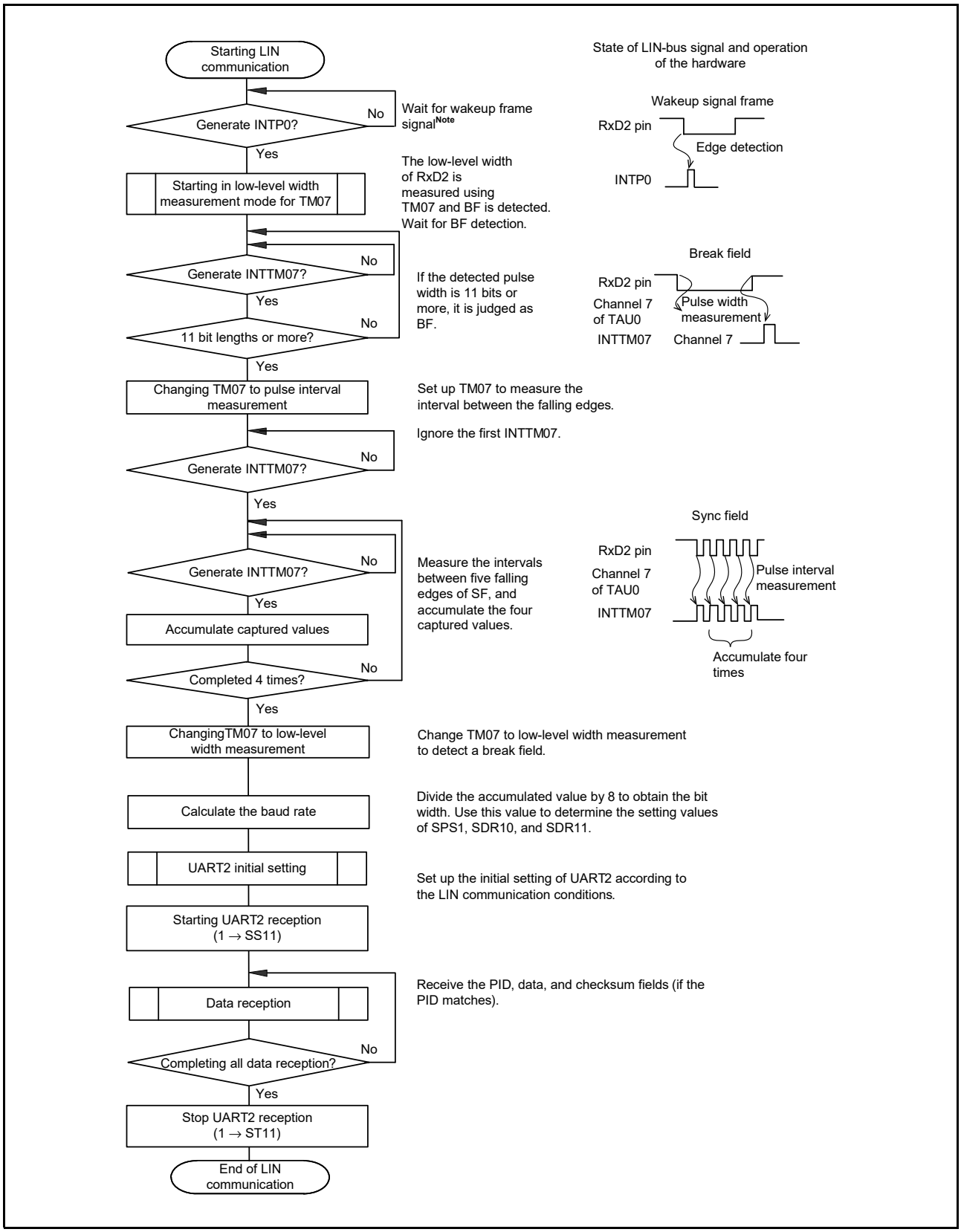
<2> TM07 starts measuring the low-level width upon detection of the falling edge of the BF signal, and then captures the data upon detection of the rising edge of the BF signal. The captured data is used to judge whether it is the BF signal.

<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 RxD2 signal in the Sync field four times. (see **7.8.4 Operation for input pulse interval measurement**.)

<4> Calculate a baud rate error from the bit interval of sync field (SF). Stop UART2 once and adjust (re-set) the baud rate.

<5> The checksum field should be distinguished by software. In addition, processing to initialize UART2 after the checksum field is received and to wait for reception of BF should also be performed by software.

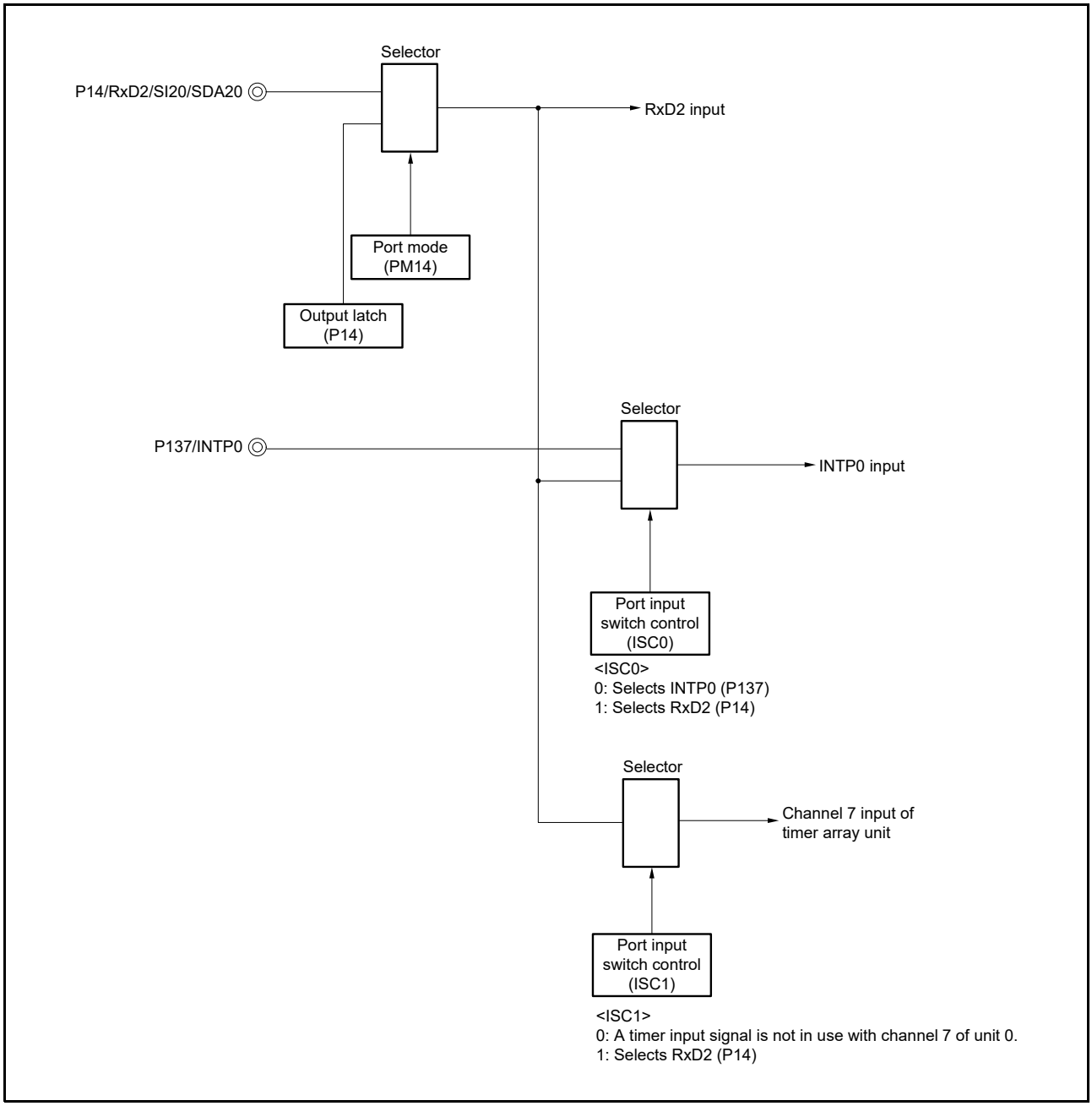
Figure 15 - 99 Flowchart of LIN Reception



Note This is only required if the LIN bus is in sleep mode.

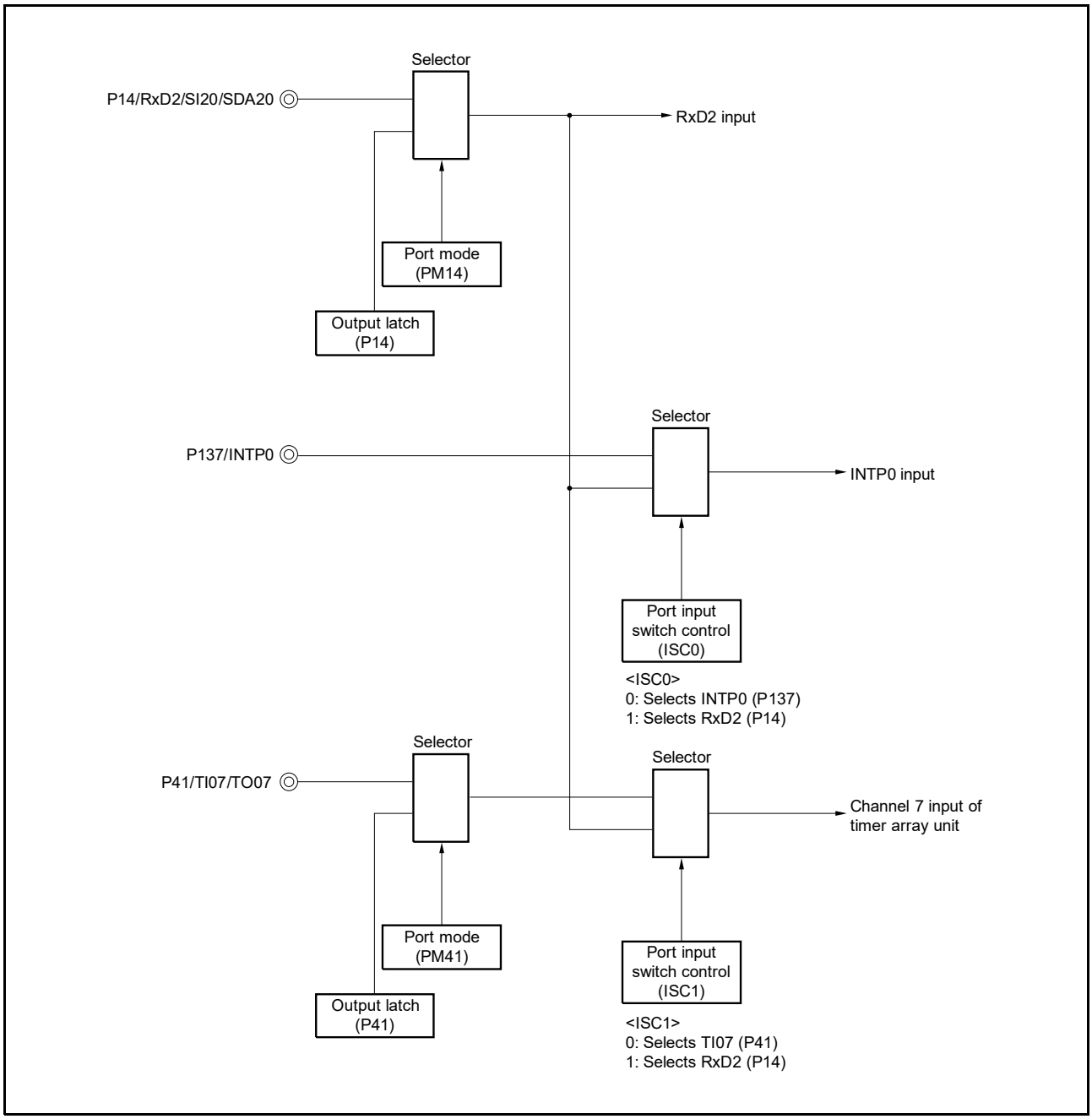
Figure 15 - 100 and **Figure 15 - 101** show the configuration of ports used for LIN reception. The wakeup signal transmitted from the master of LIN is received by detecting an edge of an external interrupt (INTP0). The length of the sync field transmitted from the master can be measured by using the external event capture operation of the timer array unit 0 to calculate a baud-rate error. By using the port input switching control (the ISC0 and ISC1 bits), the signal input to the reception port (RxD2) can be used as an external interrupt (INTP0) or sent to the timer array unit without additional external connections.

Figure 15 - 100 Port Configuration for Reception of LIN in the R7F100G Products with 30 to 40 Pins



Remark ISC0, ISC1: Bits 0 and 1 of the input switch control register (ISC) (See **Figure 15 - 21**.)

Figure 15 - 101 Port Configuration for Reception of LIN in the R7F100G Products with 44 to 64 Pins



Remark ISC0, ISC1: Bits 0 and 1 of the input switch control register (ISC) (See **Figure 15 - 21**.)

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, break field detection.
  Usage: To detect the length of the sync field (SF) and divide it by the number of bits in order to detect a baud rate error. (The interval of the edge input to RxD2 is measured in the capture mode.)
  To measure the low-level width to detect the break field (BF).
- Channels 0 and 1 (UART2) of serial array unit 1 (SAU1)

15.8 Operation of Simplified I2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31) 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 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. Operate the control registers by software for setting the start and stop conditions while observing the specifications of the I2C bus line.

[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.)
- Generation of start condition and stop condition for software

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- Overrun error
- ACK error

* [Functions not supported by simplified I2C]

- Slave transmission, slave reception
- Multi-master function (arbitration loss detection function)
- Clock stretch detection

Note When receiving the last data, ACK will not be output if 0 is written to the SOEmn (SOEm register) bit and serial communication data output is stopped. See **15.8.3 Data reception**, **2. Processing flow** for details.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

The channels supporting simplified I2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31) are channels 0 to 3 of both SAU0 and SAU1.

30- and 32-pin R7F100G products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I2C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting the LIN bus)	IIC20
	1	—		—

36- to 44-pin R7F100G products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I2C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting the LIN bus)	IIC20
	1	CSI21		IIC21

48- and 52-pin R7F100G products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I2C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting the LIN bus)	IIC20
	1	CSI21		IIC21

64-pin R7F100G products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I2C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting the LIN bus)	IIC20
	1	CSI21		IIC21

80- to 128-pin R7F100G products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11
1	0	CSI20	UART2 (supporting the LIN bus)	IIC20
	1	CSI21		IIC21
	2	CSI30	UART3	IIC30
	3	CSI31		IIC31

32- to 48-pin R7F104G products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	—		IIC11
1	0	—	UART2 (supporting the LIN bus)	—
	1	CSI21		IIC21

Simplified I²C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31) performs the following four types of communication operations.

- Address field transmission (See **15.8.1**.)
- Data transmission (See **15.8.2**.)
- Data reception (See **15.8.3**.)
- Stop condition generation (See **15.8.4**.)

15.8.1 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	IIC20	IIC21	IIC30	IIC31
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 2 of SAU1	Channel 3 of SAU1
Pins used	SCL00, SDA00Note 1	SCL01, SDA01Note 1	SCL10, SDA10Note 1	SCL11, SDA11Note 1	SCL20, SDA20Note 1	SCL21, SDA21Note 1	SCL30, SDA30Note 1	SCL31, SDA31Note 1
Interrupt	INTIIC00	INTIIC01	INTIIC10	INTIIC11	INTIIC20	INTIIC21	INTIIC30	INTIIC31
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)							
Error detection flag	ACK 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 rateNote 2	Max. f_{MCK}/4 [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operating clock frequency of target channel However, the following condition must be satisfied in each mode of I^2C. • Max. 1 MHz (fast mode plus) • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)							
Data level	Non-reverse output (default: high level)							
Parity bit	No parity bit							
Stop bit	Appending 1 bit (for ACK transmission/reception timing)							
Data direction	MSB first							

Note 1. To perform communication via simplified I^2C, set the N-ch open-drain output [withstand voltage of V_{DD} (30- to 52-pin R7F100G and 32- to 48-pin R7F104G products)/withstand voltage of EV_{DD} (64- to 128-pin R7F100G products)] mode (POMxx = 1) with the port output mode register (POMxx). See **4.3 Registers for Controlling the Port Function** and **4.5 Register Settings When Using Multiplexed Function** for details. To communicate with an external device operating at a different voltage through IIC00, IIC10, IIC20, IIC30, and IIC31, set the N-ch open-drain output [withstand voltage of V_{DD} (30- to 52-pin R7F100G and 32- to 48-pin R7F104G products)/withstand voltage of EV_{DD} (64- to 128-pin R7F100G products)] mode (POMxx = 1) also for the clock input/output pins (SCL00, SCL10, SCL20, SCL30, SCL31). See **4.4.5 Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching I/O buffers** for details.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 37 Electrical Characteristics**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

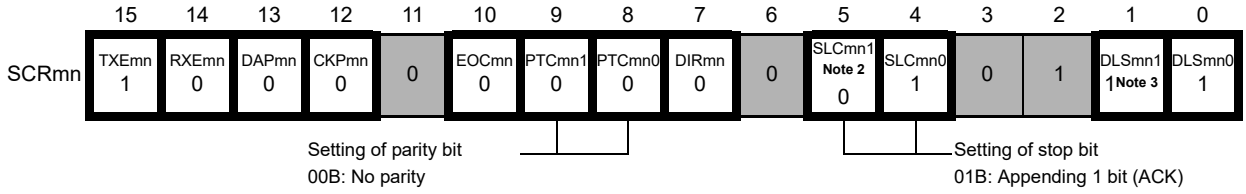
1. Register setting

Figure 15 - 102 Example of Contents of Registers for Address Field Transmission of Simplified I2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31)

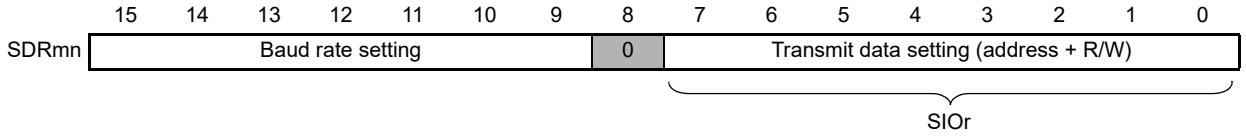
a) Serial mode register mn (SMRmn)



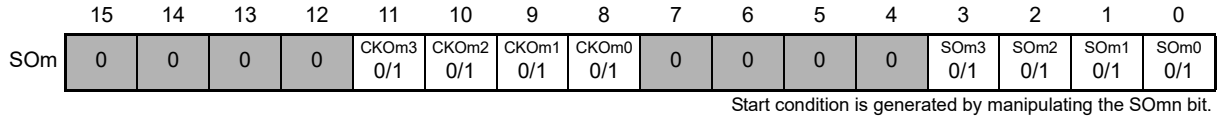
b) Serial communication operation setting register mn (SCRmn)



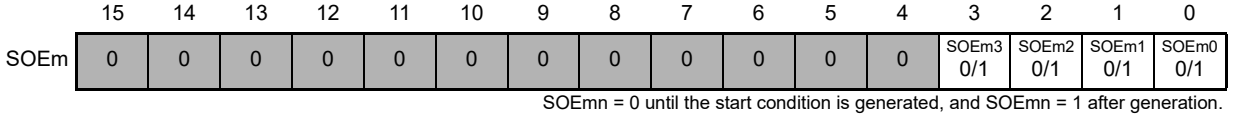
c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)



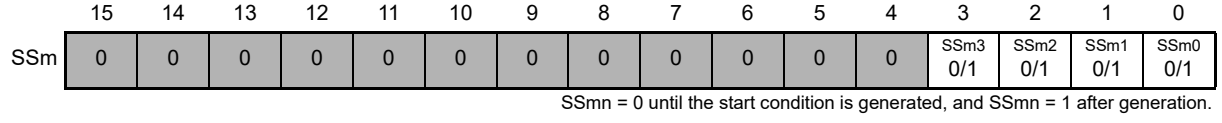
d) Serial output register m (SOm)



e) Serial output enable register m (SOEm)



f) Serial channel start register m (SSm): Set only the bit of the target channel to 1.



Note 1. This bit is only present in the SMR00, SMR03, SMR11, and SMR13 registers.

Note 2. This bit is only present in the SCR00, SCR02, SCR10, and SCR12 registers.

(Note and Remarks are listed on the next page.)

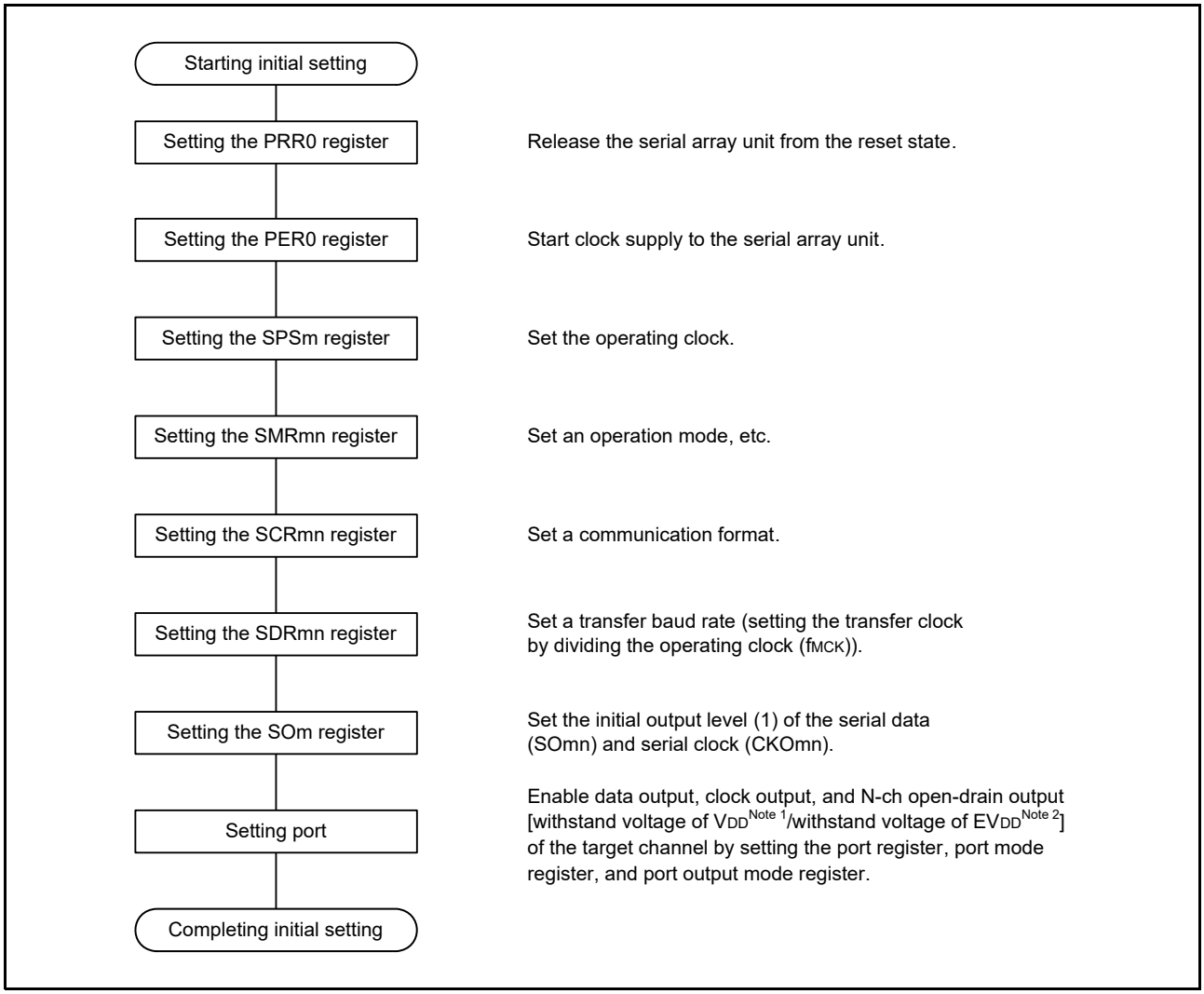
Note 3. This bit is only present in the following registers; in other registers, it is fixed to 1.
- SCR00 and SCR01 registers
- SCR10 and SCR11 registers of the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Remark 2. ■: Setting is fixed in the IIC mode, ▒: Set to the initial value.
0/1: Set to 0 or 1 depending on the user's application.

2. Operation procedure

Figure 15 - 103 Initial Setting Procedure for Simplified I2C Address Field Transmission

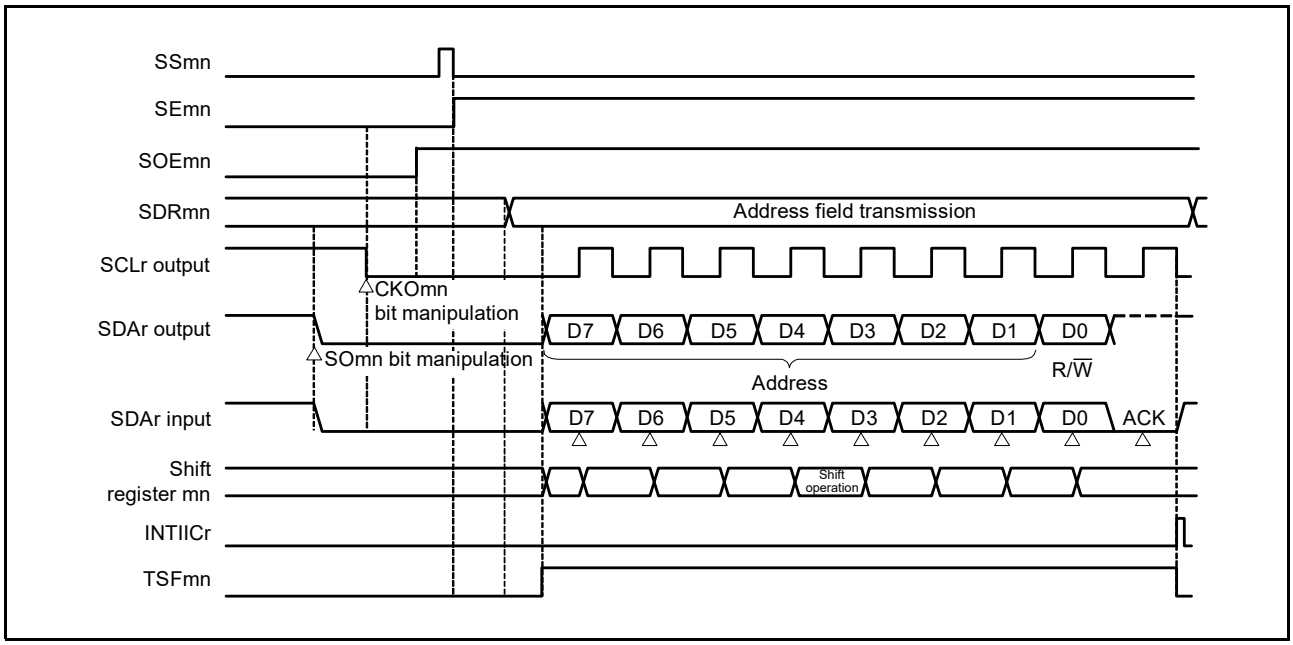


Note 1. 30- to 52-pin R7F100G and 32- to 48-pin R7F104G products

Note 2. 64- to 128-pin R7F100G products

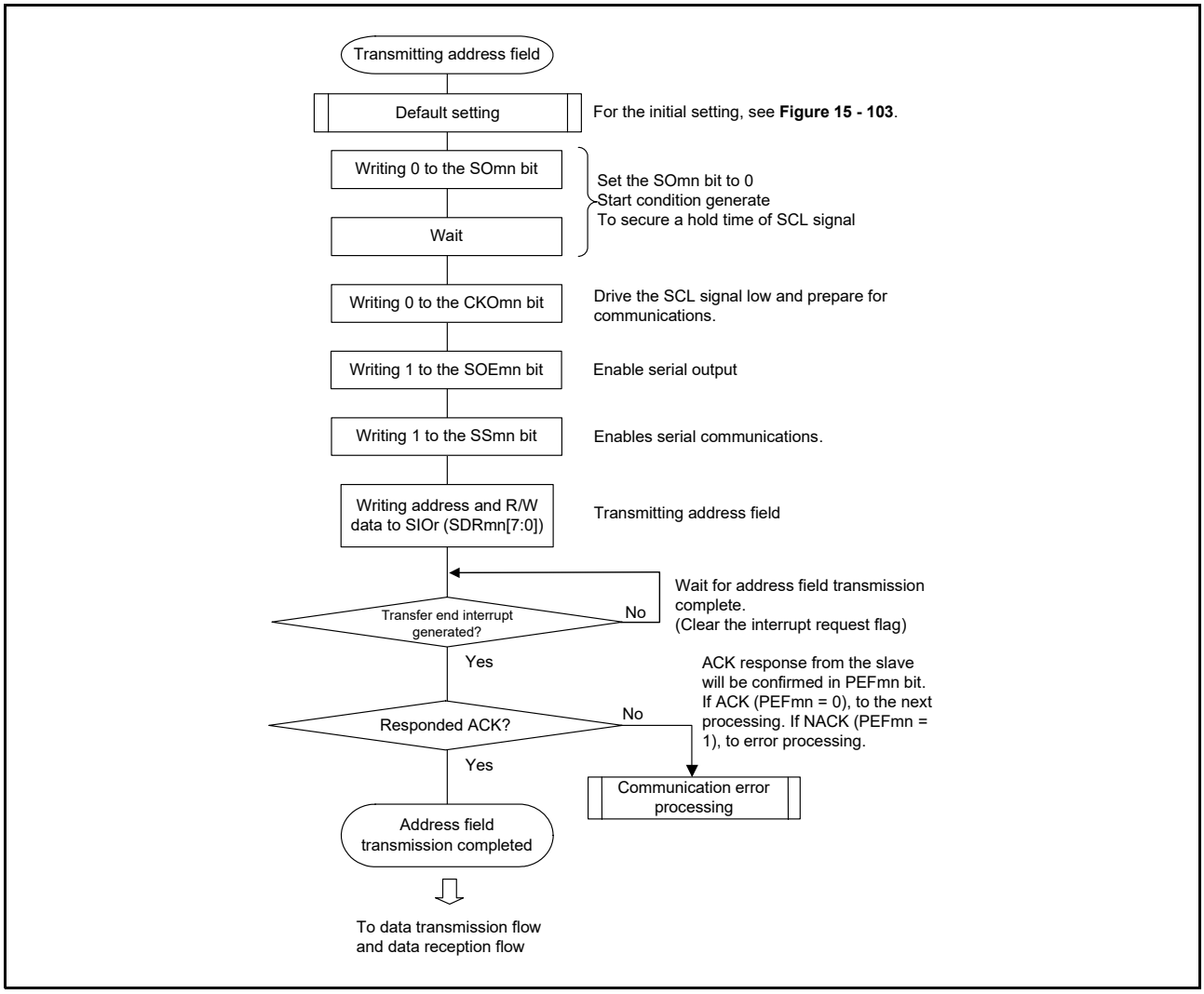
3. Processing flow

Figure 15 - 104 Timing Chart of Address Field Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Figure 15 - 105 Flowchart of Simplified I2C Address Field Transmission



15.8.2 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 I2C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21	IIC30	IIC31
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 2 of SAU1	Channel 3 of SAU1
Pins used	SCL00, SDA00Note 1	SCL01, SDA01Note 1	SCL10, SDA10Note 1	SCL11, SDA11Note 1	SCL20, SDA20Note 1	SCL21, SDA21Note 1	SCL30, SDA30Note 1	SCL31, SDA31Note 1
Interrupt	INTIIC00	INTIIC01	INTIIC10	INTIIC11	INTIIC20	INTIIC21	INTIIC30	INTIIC31
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)							
Error detection flag	ACK error flag (PEFmn)							
Transfer data length	8 bits							
Transfer rateNote 2	Max. fMCK/4 [Hz] (SDRmn[15:9] = 1 or more) fMCK: Operating clock frequency of target channel However, the following condition must be satisfied in each mode of I2C. • Max. 1 MHz (fast mode plus) • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)							
Data level	Non-reverse output (default: high level)							
Parity bit	No parity bit							
Stop bit	Appending 1 bit (for ACK reception timing)							
Data direction	MSB first							

Note 1. To perform communication via simplified I2C, set the N-ch open-drain output [withstand voltage of VDD (30- to 52-pin R7F100G and 32- to 48-pin R7F104G products)/withstand voltage of EVDD (64- to 128-pin R7F100G products)] mode (POMxx = 1) with the port output mode register (POMxx). For details, see **4.3 Registers for Controlling the Port Function** and **4.5 Register Settings When Using Multiplexed Function**. To communicate with an external device operating at a different voltage through IIC00, IIC10, IIC20, IIC30, and IIC31, set the N-ch open-drain output [withstand voltage of VDD (30- to 52-pin R7F100G and 32- to 48-pin R7F104G products)/withstand voltage of EVDD (64- to 128-pin R7F100G products)] mode (POMxx = 1) also for the clock input/output pins (SCL00, SCL10, SCL20, SCL30, SCL31). For details, see **4.4.5 Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching I/O buffers**.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 37 Electrical Characteristics**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

1. Register setting

Figure 15 - 106 Example of Contents of Registers for Data Transmission of Simplified I²C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31)

a) Serial mode register mn (SMRmn): Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0 Note 1	0	SISmn0 0 Note 1	1	0	0	MDmn2 1	MDmn1 0	MDmn0 0

b) Serial communication operation setting register mn (SCRmn): Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0	CKPmn 0	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0	0	SLCmn1 0 Note 2	SLCmn0 1	0	1	DLSmn1 1 Note 3	DLSmn0 1

c) Serial data register mn (SDRmn) (lower 8 bits: SIOr): During data transmission/reception, valid only lower 8-bits (SIOr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting Note 4							0	Transmit data setting							
									SIOr							

d) Serial output register m (SOm): Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	CKOm3 0/1 Note 5	CKOm2 0/1 Note 5	CKOm1 0/1 Note 5	CKOm0 0/1 Note 5	0	0	0	0	SOm3 0/1 Note 5	SOm2 0/1 Note 5	SOm1 0/1 Note 5	SOm0 0/1 Note 5

e) Serial output enable register m (SOEm): Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	SOEm3 1	SOEm2 1	SOEm1 1	SOEm0 1

f) Serial channel start register m (SSm): Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 0/1	SSm1 0/1	SSm0 0/1

Note 1. This bit is only present in the SMR01, SMR03, SMR11, and SMR13 registers.

Note 2. This bit is only present in the SCR00, SCR02, SCR10, and SCR12 registers.

Note 3. This bit is only present in the following registers; in other registers, it is fixed to 1.
- SCR00 and SCR01 registers
- SCR10 and SCR11 registers of the 80- to 128-pin R7F100G and 32- to 48-pin R7F104G products

Note 4. Because the setting is completed by address field transmission, setting is not required.

Note 5. The value varies depending on the communication data during communication operation.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Remark 2. ■: Setting is fixed in the IIC mode, ■: Set to the initial value.
0/1: Set to 0 or 1 depending on the user's application.

2. Processing flow

Figure 15 - 107 Timing Chart of Data Transmission

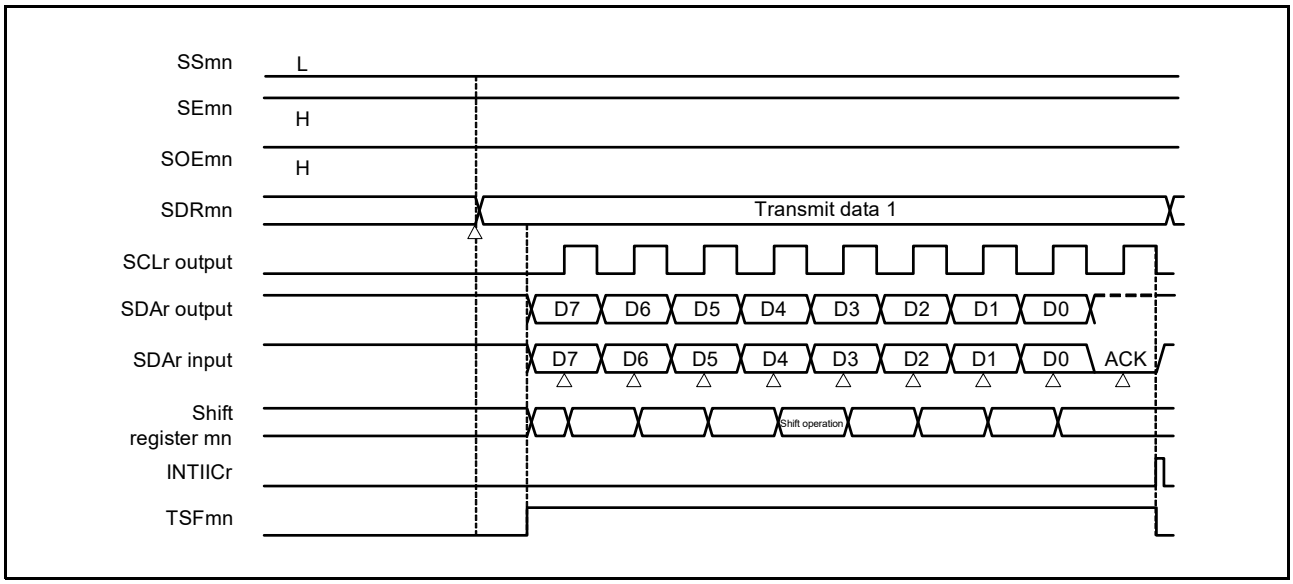
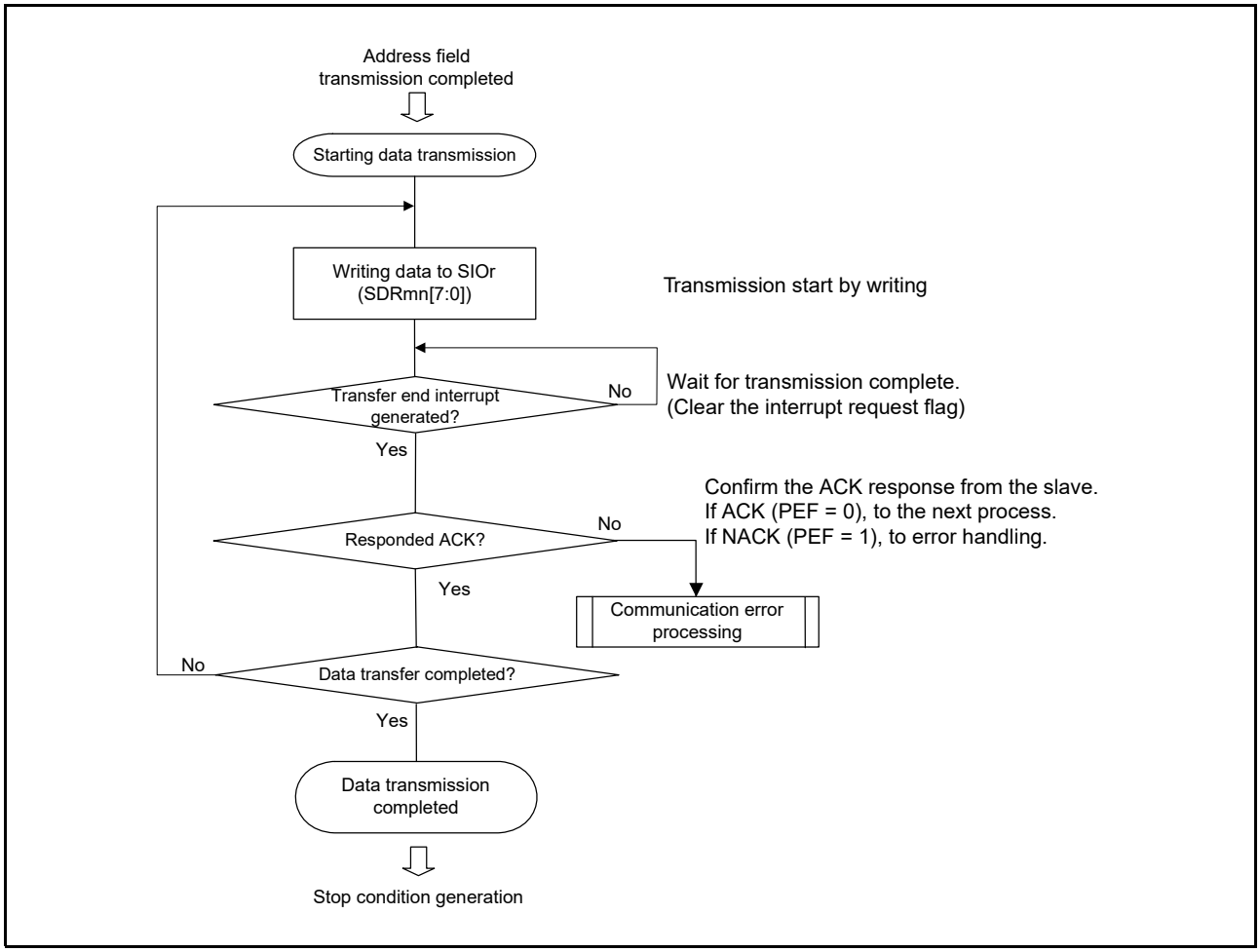


Figure 15 - 108 Flowchart of Simplified I2C Data Transmission



15.8.3 Data reception

Data reception is an operation to receive data from the target for transfer (slave) after transmission of an address field. After all data are received from the slave, a stop condition is generated and the bus is released.

Simplified I²C	IIC00	IIC01	IIC10	IIC11	IIC20	IIC21	IIC30	IIC31
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1	Channel 1 of SAU1	Channel 2 of SAU1	Channel 3 of SAU1
Pins used	SCL00, SDA00Note 1	SCL01, SDA01Note 1	SCL10, SDA10Note 1	SCL11, SDA11Note 1	SCL20, SDA20Note 1	SCL21, SDA21Note 1	SCL30, SDA30Note 1	SCL31, SDA31Note 1
Interrupt	INTIIC00	INTIIC01	INTIIC10	INTIIC11	INTIIC20	INTIIC21	INTIIC30	INTIIC31
	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 rateNote 2	Max. fMCK/4 [Hz] (SDRmn[15:9] = 1 or more) fMCK: Operating clock frequency of target channel However, the following condition must be satisfied in each mode of I²C. • Max. 1 MHz (fast mode plus) • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)							
Data level	Non-reverse output (default: high level)							
Parity bit	No parity bit							
Stop bit	Appending 1 bit (ACK transmission)							
Data direction	MSB first							

Note 1. To perform communication via simplified I²C, set the N-ch open-drain output [withstand voltage of VDD (30- to 52-pin R7F100G and 32- to 48-pin R7F104G products)/withstand voltage of EVDD (64- to 128-pin R7F100G products)] mode (POMxx = 1) with the port output mode register (POMxx). For details, see **4.3 Registers for Controlling the Port Function** and **4.5 Register Settings When Using Multiplexed Function**. To communicate with an external device operating at a different voltage through IIC00, IIC10, IIC20, IIC30, and IIC31, set the N-ch open-drain output [withstand voltage of VDD (30- to 52-pin R7F100G and 32- to 48-pin R7F104G products)/withstand voltage of EVDD (64- to 128-pin R7F100G products)] mode (POMxx = 1) also for the clock input/output pins (SCL00, SCL10, SCL20, SCL30, SCL31). For details, see **4.4.5 Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching I/O buffers**.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **Section 37 Electrical Characteristics**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

1. Register setting

Figure 15 - 109 Example of Contents of Registers for Data Reception of Simplified I²C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31)

a) Serial mode register mn (SMRmn): Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0 Note 1	0	SISmn0 0 Note 1	1	0	0	MDmn2 1	MDmn1 0	MDmn0 0

b) Serial communication operation setting register mn (SCRmn): Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0	CKPmn 0	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0	0	SLCmn1 0 Note 2	SLCmn0 1	0	1	DLSmn1 1 Note 3	DLSmn0 1

c) Serial data register mn (SDRmn) (lower 8 bits: SIOr): The lower 8 bits (SIOr) are only valid during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting Note 4							0	Dummy transmit data setting (FFH)							
									SIOr							

d) Serial output register m (SOm): Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	CKOm3 0/1 Note 5	CKOm2 0/1 Note 5	CKOm1 0/1 Note 5	CKOm0 0/1 Note 5	0	0	0	0	SOm3 0/1 Note 5	SOm2 0/1 Note 5	SOm1 0/1 Note 5	SOm0 0/1 Note 5

e) Serial output enable register m (SOEm): Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	SOEm3 0/1	SOEm2 0/1	SOEm1 0/1	SOEm0 0/1

f) Serial channel start register m (SSm): Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 0/1	SSm1 0/1	SSm0 0/1

Note 1. This bit is only present in the SMR01, SMR03, SMR11, and SMR13 registers.

Note 2. This bit is only present in the SCR00, SCR02, SCR10, and SCR12 registers.

Note 3. This bit is only present in the following registers; in other registers, it is fixed to 1.
- SCR00 and SCR01 registers
- SCR10 and SCR11 registers of the 80- to 128-pin R7F100G and 32- to 48-pin R7F104G products

Note 4. Because the setting is completed by address field transmission, setting is not required.

Note 5. The value varies depending on the communication data during communication operation.

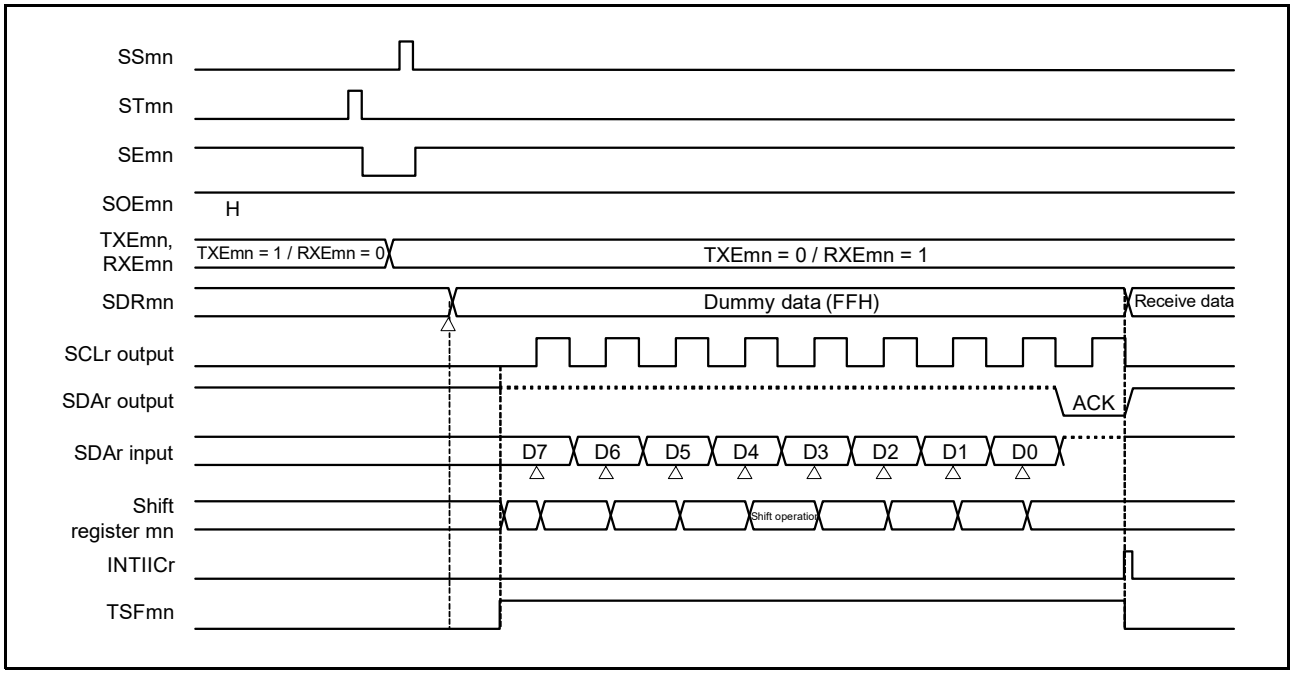
Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Remark 2. ■: Setting is fixed in the IIC mode, ■: Set to the initial value.
0/1: Set to 0 or 1 depending on the user's application.

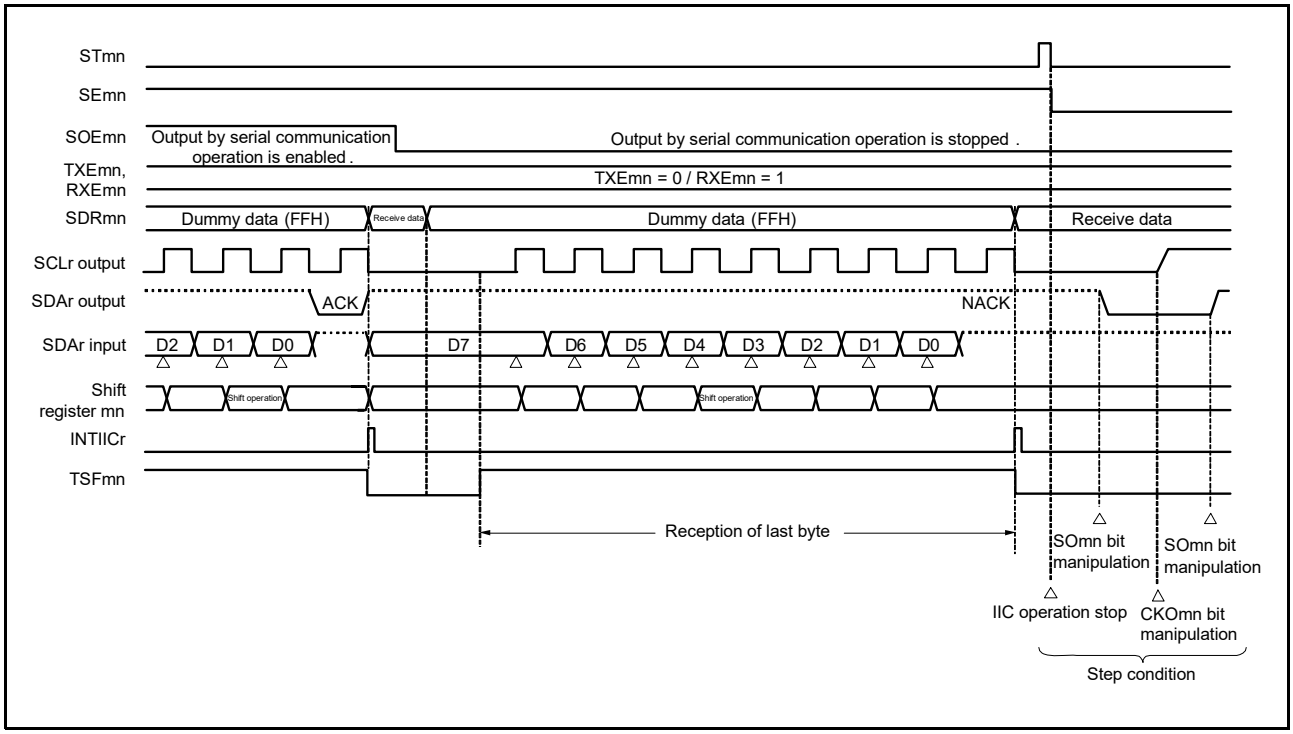
2. Processing flow

Figure 15 - 110 Timing Chart of Data Reception

a) When starting data reception

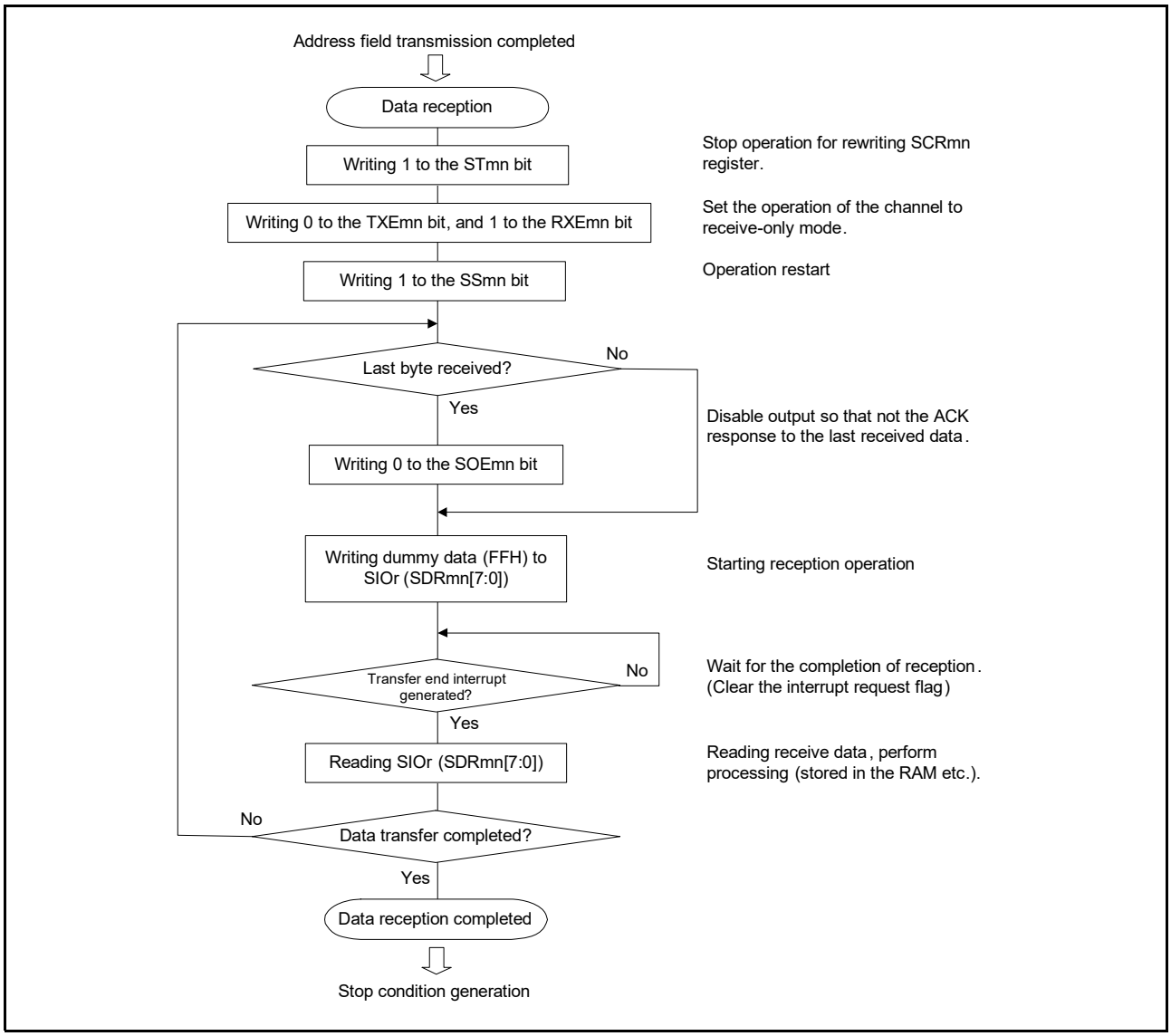


b) When receiving last data



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Figure 15 - 111 Flowchart of Data Reception



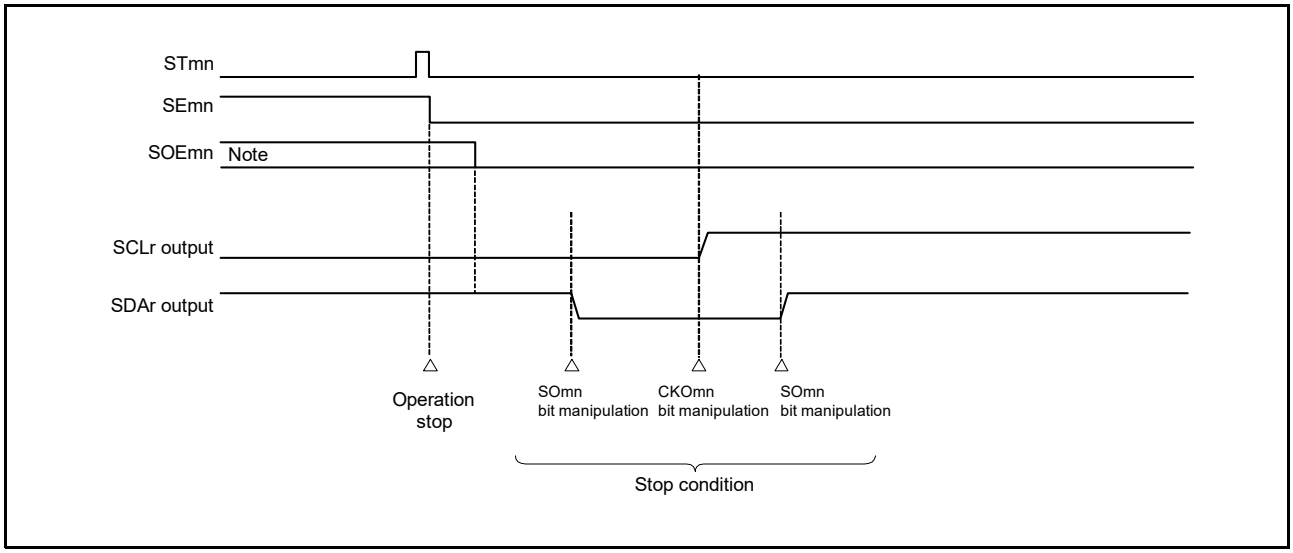
Caution **ACK is not output when the last data is received (NACK). Communication is then completed by setting 1 in the STmn bit of serial channel stop register m (STm) to stop operation and generating a stop condition.**

15.8.4 Stop condition generation

After all data are transmitted to or received from the target slave, a stop condition is generated and the bus is released.

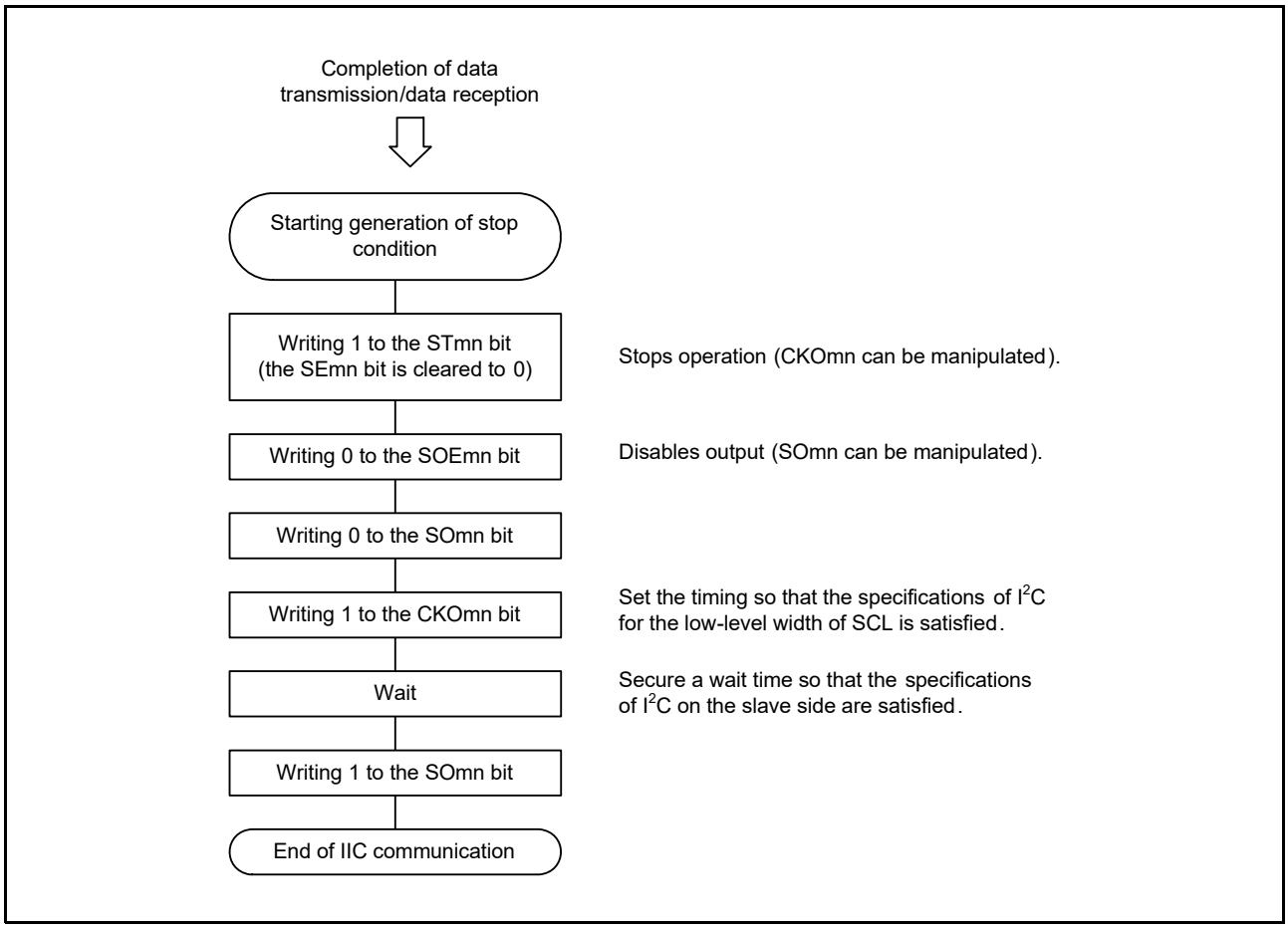
1. Processing flow

Figure 15 - 112 Timing Chart of Stop Condition Generation



Note During a receive operation, the SOEmn bit of serial output enable register m (SOEm) is set to 0 before receiving the last data.

Figure 15 - 113 Flowchart of Stop Condition Generation



15.8.5 Calculating transfer rate

The transfer rate for simplified I²C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31) communication can be calculated by the following expressions.

(Transfer rate) = {Operating clock (fMCK) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2

<R> **Caution SDRmn[15:9] must not be set to 0000000B. Set SDRmn[15:9] to 0000001B or greater. The duty ratio of the SCL signal output by the simplified I²C is 50%. The I²C bus specifications define that the low-level width of the SCL signal is longer than the high-level width. If 400 kbps (fast mode) or 1 Mbps (fast mode plus) is specified, therefore, the low-level width of the SCL output signal becomes shorter than the value specified in the I²C bus specifications. Make sure that the SDRmn[15:9] value satisfies the I²C bus specifications.**

Remark 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.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

The operating clock (fMCK) is determined by serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn).

Table 15 - 9 Selection of Operating Clock for Simplified I²C

SMRmn Register	SPSm Register								Operating Clock (f_{MCK})Note	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 32 MHz
0	×	×	×	×	0	0	0	0	f_{CLK}	32 MHz
	×	×	×	×	0	0	0	1	$f_{CLK}/2$	16 MHz
	×	×	×	×	0	0	1	0	$f_{CLK}/2^2$	8 MHz
	×	×	×	×	0	0	1	1	$f_{CLK}/2^3$	4 MHz
	×	×	×	×	0	1	0	0	$f_{CLK}/2^4$	2 MHz
	×	×	×	×	0	1	0	1	$f_{CLK}/2^5$	1 MHz
	×	×	×	×	0	1	1	0	$f_{CLK}/2^6$	500 kHz
	×	×	×	×	0	1	1	1	$f_{CLK}/2^7$	250 kHz
	×	×	×	×	1	0	0	0	$f_{CLK}/2^8$	125 kHz
	×	×	×	×	1	0	0	1	$f_{CLK}/2^9$	62.5 kHz
	×	×	×	×	1	0	1	0	$f_{CLK}/2^{10}$	31.25 kHz
	×	×	×	×	1	0	1	1	$f_{CLK}/2^{11}$	15.63 kHz
1	0	0	0	0	×	×	×	×	f_{CLK}	32 MHz
	0	0	0	1	×	×	×	×	$f_{CLK}/2$	16 MHz
	0	0	1	0	×	×	×	×	$f_{CLK}/2^2$	8 MHz
	0	0	1	1	×	×	×	×	$f_{CLK}/2^3$	4 MHz
	0	1	0	0	×	×	×	×	$f_{CLK}/2^4$	2 MHz
	0	1	0	1	×	×	×	×	$f_{CLK}/2^5$	1 MHz
	0	1	1	0	×	×	×	×	$f_{CLK}/2^6$	500 kHz
	0	1	1	1	×	×	×	×	$f_{CLK}/2^7$	250 kHz
	1	0	0	0	×	×	×	×	$f_{CLK}/2^8$	125 kHz
	1	0	0	1	×	×	×	×	$f_{CLK}/2^9$	62.5 kHz
	1	0	1	0	×	×	×	×	$f_{CLK}/2^{10}$	31.25 kHz
	1	0	1	1	×	×	×	×	$f_{CLK}/2^{11}$	15.63 kHz
Other than above									Setting prohibited	

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Remark 1. ×: Don't care

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10 to 13

Here is an example of setting an I^2C transfer rate where $f_{MCK} = f_{CLK} = 32$ MHz.

I^2C Transfer Mode (Desired Transfer Rate)	f_{CLK} = 32 MHz			
	Operating 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}	41	380 kHz	5.0%**Note**
1 MHz	f_{CLK}	18	0.84 MHz	16.0%**Note**

Note The error cannot be set to about 0% because the duty ratio of the SCL signal is 50%.

15.8.6 Procedure for processing errors that occurred during simplified I2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31) communication

Table 15 - 10 and **Table 15 - 11** describe the procedure for processing errors that occurred during simplified I2C (IIC00, IIC01, IIC10, IIC11, IIC20, IIC21, IIC30, IIC31) communication.

Table 15 - 10 Processing Procedure in Case of Overrun Error

Software Manipulation	State of the Hardware	Remark
Reads serial data register mn (SDRmn).	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		The error type is identified and the read value is used to clear the error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn).	The error flag is cleared.	Only the error during reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.

Table 15 - 11 Processing Procedure in Case of ACK Error in Simplified I2C Mode

Software Manipulation	State of the Hardware	Remark
Reads serial status register mn (SSRmn).		The error type is identified and the read value is used to clear the error flag.
Writes serial flag clear trigger register mn (SIRmn).	The error flag is cleared.	Only the error during reading can be cleared, 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.	The 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 a stop condition.		
Creates a 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 to 3), r: IIC number (r = 00, 01, 10, 11, 20, 21, 30, 31), mn = 00 to 03, 10 to 13

Section 16 Serial Interface IICA (IICA)

The number of the serial interface IICA channels depends on the product.

	R7F100G		R7F104G	
	30- to 40-pin Products	44- to 128-pin Products	32- and 44-pin Products	48-pin Product
Number of channels	1	2	—	1

16.1 Functions of Serial Interface IICA

The serial interface IICA has the following three modes.

1. Operation stop mode
   This mode is used when serial transfers are not performed. The operating power can be reduced in this mode.

2. I^2C bus mode (multi-master supported)
   This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCLAn) line and a serial data bus (SDAAn) line. This mode complies with the I^2C bus format and the master device can send start conditions, addresses, transfer directions, acknowledges (ACK), data, and stop conditions to the slave devices, via the serial data bus. The slave device automatically detects these states and data by hardware. This function can simplify the part of application program that controls the I^2C bus. Since the SCLAn and SDAAn 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 (INTIICAn) when an extension code from the master device or the local address has been received while in STOP mode. This can be set by using the WUPn bit of IICA control register n1 (IICCTLn1). The all-address match function is enabled by setting the SVADISn bit of the IICCTLn1 register to 1, allowing any received address is to be determined as a matched address.

Remark n = 0, 1

Figure 16 - 1 shows a block diagram of serial interface IICA0.

Figure 16 - 1 Block Diagram of Serial Interface IICA0

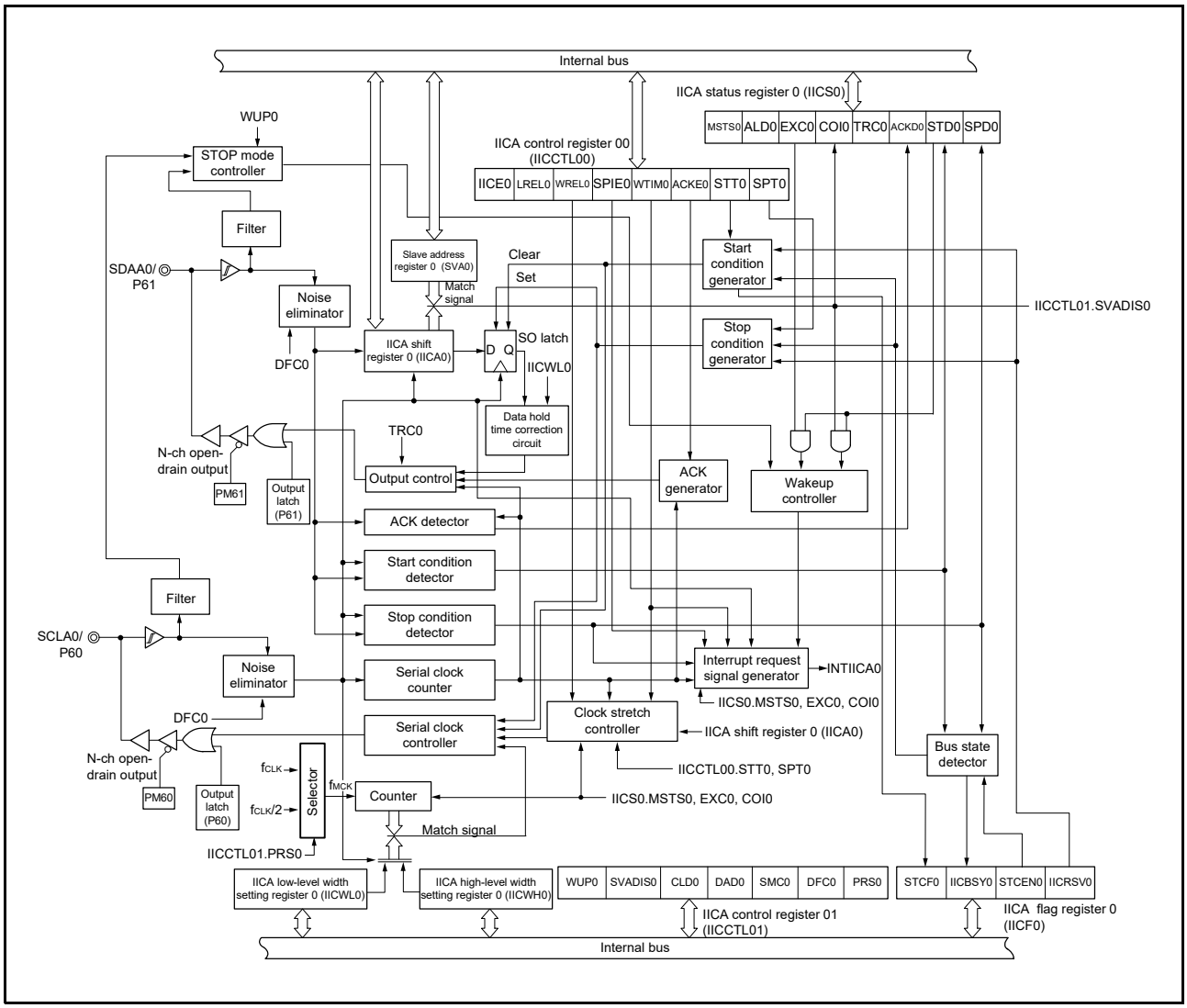
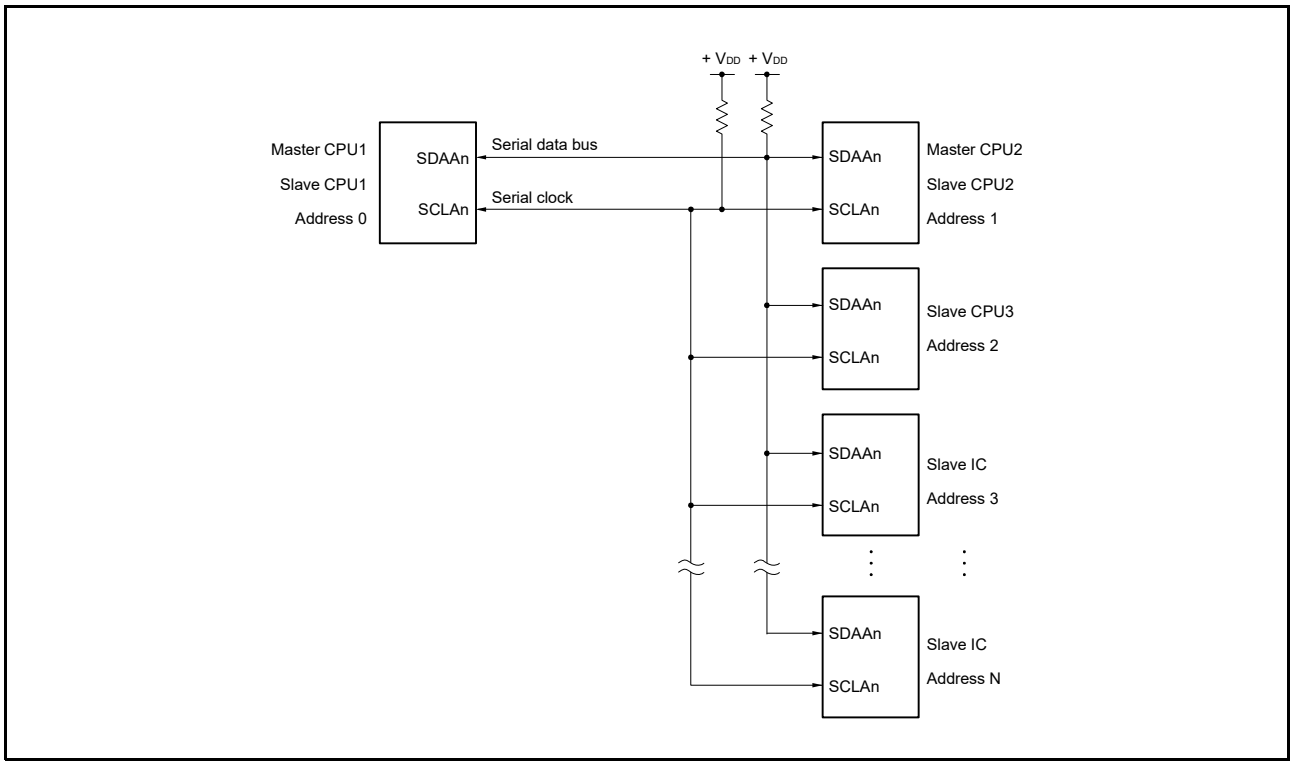


Figure 16 - 2 shows an example of the serial bus configuration using the I2C bus.

Figure 16 - 2 Example of the Serial Bus Configuration Using the I2C Bus



Remark n = 0, 1

16.2 Configuration of Serial Interface IICA

Serial interface IICA includes the following hardware.

Table 16 - 1 Configuration of Serial Interface IICA

Item	Configuration
Registers	• IICA shift register n (IICAn) • Slave address register n (SVAn)
Control registers	• Peripheral enable register 0 (PER0) • Peripheral reset control register 0 (PRR0) • IICA control register n0 (IICCTLn0) • IICA status register n (IICSn) • IICA flag register n (IICFn) • IICA control register n1 (IICCTLn1) • IICA low-level width setting register n (IICWLn) • IICA high-level width setting register n (IICWHn) • Port mode registers (PMxx) • Port registers (Pxx)

Remark n = 0, 1

1. IICA shift register n (IICAn)
   The IICAn register is used to convert 8-bit serial data to 8-bit parallel data and vice versa in synchronization with the serial clock. The IICAn register can be used for both transmission and reception. The actual transmit and receive operations can be controlled by writing to and reading from the IICAn register. Release serial interface IICA from the clock stretch state and start data transfer by writing data to the IICAn register during the clock stretch period. The IICAn register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 16 - 3 Format of IICA Shift Register n (IICAn)

Address: FFF50H (IICA0), FFF54H (IICA1)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
IICAn								

Caution 1. Do not write data to the IICAn register during data transfer.
Caution 2. Write to or read from the IICAn register only during the clock stretch period. Accessing the IICAn register in a communication state other than during the clock stretch period is prohibited. When the device serves as the master, however, the IICAn register can be written only once after the communication trigger bit (IICCTLn0.STTn) is set to 1.
Caution 3. When communication is reserved, write data to the IICAn register after the interrupt triggered by a stop condition is detected.

Remark n = 0, 1

2. Slave address register n (SVAn)
   This register holds seven bits (A6, A5, A4, A3, A2, A1, and A0) of the local address when in slave mode. The SVAn register can be set by an 8-bit memory manipulation instruction. However, rewriting to this register is prohibited while STDn = 1 (while the start condition is detected). The value of this register following a reset is 00H.

Figure 16 - 4 Format of Slave Address Register n (SVAn)

Address: F0234H (SVA0), F023CH (SVA1)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
SVAn	A6	A5	A4	A3	A2	A1	A0	0 **Note**

Note Bit 0 is fixed to 0.

Remark n = 0, 1

3. SO latch
   The SO latch is used to retain the output level of the SDAAn pin.

4. Wakeup controller
   This circuit generates an interrupt request signal (INTIICAn) when the received address matches the address value set to the slave address register n (SVAn), when any address is received while the all-address match function is enabled, or when an extension code is received.

5. Serial clock counter
   This counter counts the serial clock cycles 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 (INTIICAn). 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 IICCTLn0.WTIMn bit)
   - Interrupt request generated when a stop condition is detected (set by the IICCTLn0.SPIEn bit)

7. Serial clock controller
   In master mode, this circuit generates the serial clock, which is output via the SCLAn pin.

8. Clock stretch controller
   This circuit controls the timing of clock stretching.

9. ACK generator, stop condition detector, start condition detector, and ACK detector
   These circuits generate or detect each state.

10. Data hold time correction circuit
    This circuit generates the hold time for data after the falling edge of the serial clock.

11. Start condition generator
    This circuit generates a start condition when the IICCTLn0.STTn bit is set to 1. However, while communication reservations are disabled (IICFn.IICRSVn = 1) and the bus is busy (IICFn.IICBSYn = 1), start condition requests are ignored and the IICFn.STCFn flag is set to 1.

12. Stop condition generator
    This circuit generates a stop condition when the IICCTLn0.SPTn bit is set to 1.

13. Bus state detector
    This circuit detects whether or not the bus is released by detecting start conditions and stop conditions. Note that the bus state cannot be detected immediately after the IICA operation has been enabled. Set the initial state of the bus state detector by the IICFn.STCENn bit.

Remark n = 0, 1

16.3 Registers for Controlling Serial Interface IICA

The following registers are used to control serial interface IICA.

- Peripheral enable register 0 (PER0)
- Peripheral reset control register 0 (PRR0)
- IICA control register n0 (IICCTLn0)
- IICA status register n (IICSn)
- IICA flag register n (IICFn)
- IICA control register n1 (IICCTLn1)
- IICA low-level width setting register n (IICWLn)
- IICA high-level width setting register n (IICWHn)
- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control E registers (PMCExx)

16.3.1 Peripheral enable register 0 (PER0)

The PER0 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. When serial interface IICA0 is to be used, be sure to set the IICA0EN bit of this register to 1. When serial interface IICA1 is to be used, be sure to set the IICA1EN bit of this register to 1. The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 16 - 5 Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCWEN	IICA1EN Note 1	ADCEN	IICA0EN Note 2	SAU1EN	SAU0EN	TAU1EN Note 3	TAU0EN

IICAnEN	Control of supply of an input clock to the serial interface IICAn
0	Stops supply of an input clock. • The SFRs used by the serial interface IICAn cannot be written.Note 4
1	Enables supply of an input clock. • The SFRs used by the serial interface IICAn can be read and written.

Note 1. This bit is only present in the 44- to 128-pin R7F100G products.
Note 2. This bit is only present in the 30- to 128-pin R7F100G and 48-pin R7F104G products.
Note 3. This bit is only present in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products.
Note 4. When an SFR used by the serial interface IICAn is read, the value returned is 00H or 0000H regardless of the setting value.

Caution 1. Be sure to leave the following bits set to 0.
Bits 6 and 1 in the 30- to 40-pin R7F100G products
Bit 1 in the 44- to 64-pin R7F100G products
Bits 6, 4, and 1 in the 32-pin R7F104G product
Bits 6 and 4 in the 44-pin R7F104G product
Bit 6 in the 48-pin R7F104G product

Caution 2. When setting a serial interface IICA, start by setting the following registers while the setting of the IICAnEN bit is 1.
- **IICA control register n0 (IICCTLn0)**
- **IICA flag register n (IICFn)**
- **IICA status register n (IICSn)**
- **IICA control register n1 (IICCTLn1)**
- **IICA low-level width setting register n (IICWLn)**
- **IICA high-level width setting register n (IICWHn)**
- **IICA shift register n (IICAn)**
- **Slave address register n (SVAn)**

If the setting of the IICAnEN bit is 0, each register which controls the serial interface IICA is restored to its initial value and writing to any of those registers is ignored. Note, however, writing to the following registers is valid.
- **Port mode register (PM6)**
- **Port register (P6)**

Caution 3. While the operation of a peripheral function is enabled, do not switch the setting of the corresponding bit in the PER0 register. When switching the setting by the PER0 register, each peripheral function assigned to the PER0 register must be stopped.

Remark n = 0, 1

16.3.2 Peripheral reset control register 0 (PRR0)

The PRR0 register is used to control resetting of the on-chip peripheral modules. Use this register to control reset or release from the reset for the target on-chip peripheral modules. The PRR0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 16 - 6 Format of Peripheral Reset Control Register 0 (PRR0)

Address: F00F1H
After reset: 00H
R/W: R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PRR0	0	IICA1RES **Note 1**	ADCRES	IICA0RES **Note 2**	SAU1RES	SAU0RES	TAU1RES **Note 3**	TAU0RES

IICAnRES	Control resetting of the serial interface IICAn
0	Serial interface IICAn is released from the reset state.
1	Serial interface IICAn is in the reset state. • The SFRs for use with the serial interface IICAn are initialized.

Note 1. This bit is only present in the 44- to 128-pin R7F100G products.
Note 2. This bit is only present in the 30- to 128-pin R7F100G and 48-pin R7F104G products.
Note 3. This bit is only present in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products.

Caution 1. Be sure to leave bit 7 set to 0.
Caution 2. Be sure to leave the following bits set to 0.
Bits 6 and 1 in the 30- to 40-pin R7F100G products
Bit 1 in the 44- to 64-pin R7F100G products
Bits 6, 4, and 1 in the 32-pin R7F104G product
Bits 6 and 4 in the 44-pin R7F104G product
Bit 6 in the 48-pin R7F104G product

Remark n = 0, 1

16.3.3 IICA control register n0 (IICCTLn0)

This register is used to enable or disable the I2C operations, set the timing of clock stretching, and set other I2C operations. The IICCTLn0 register can be set by a 1-bit or 8-bit memory manipulation instruction. Note that bits SPIEn, WTIMn, and ACKEn must be set while the setting of IICEn is 0 or this module is in the clock stretch state. These bits can be set at the same time as setting the IICEn bit 1. The value of this register following a reset is 00H.

Remark n = 0, 1

Figure 16 - 7 Format of IICA Control Register n0 (IICCTLn0) (1/4)

Address: F0230H (IICCTL00), F0238H (IICCTL10)
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICCTLn0	IICEn	LRELn	WRELn	SPIEn	WTIMn	ACKEn	STTn	SPTn

IICEn	I2C operation enable
0	Stop operation. Reset the IICA status register n (IICSn).Note 1 Stop internal operation.
1	Enable operation.
Be sure to set this bit to 1 while the SCLAn and SDAAn lines are at high level.	
Condition for clearing (IICEn = 0)	Condition for setting (IICEn = 1)
• Cleared by instruction • Reset	• Set by instruction

LRELn Notes 2, 3	Exit from communications
0	Normal operation
1	IICA exits from the current communications and is in the standby mode. This bit is automatically cleared to 0 after having been set to 1. Its uses include cases in which a locally irrelevant extension code has been received. The SCLAn and SDAAn lines are set to high impedance. The following flags of IICA control register n0 (IICCTLn0) and the IICA status register n (IICSn) are cleared to 0. • STTn, SPTn, MSTSn, EXCn, COIn, TRCn, ACKDn, and STDn
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, extension code reception, or address reception with the all-address match function enabled occurs after the start condition.	
Condition for clearing (LRELn = 0)	Condition for setting (LRELn = 1)
• Automatically cleared after execution • Reset	• Set by instruction

Figure 16 - 7 Format of IICA Control Register n0 (IICCTLn0) (2/4)

WRELn **Notes 2, 3**	Release from the clock stretch state
0	The interface is not released from the clock stretch state.
1	The interface is released from the clock stretch state. After release from the clock stretch state, this bit is automatically cleared to 0.
When the WRELn bit is set (for release from the clock stretch state) during the clock stretch period at the ninth clock pulse in the transmission state (IICSn.TRCn = 1), the SDAAn line goes into the high impedance state (IICSn.TRCn = 0).	
Condition for clearing (WRELn = 0)	Condition for setting (WRELn = 1)
• Automatically cleared after execution • Reset	• Set by instruction

SPIEn**Note 2**	Enable/disable generation of interrupt request when stop condition is detected
0	Disable
1	Enable
If IICCTLn1.WUPn = 1, no stop condition interrupt will be generated even if SPIEn = 1.	
Condition for clearing (SPIEn = 0)	Condition for setting (SPIEn = 1)
• Cleared by instruction • Reset	• Set by instruction

WTIMn **Note 4**	Control of clock stretching and interrupt request generation
0	An interrupt request is generated on the falling edge of the eighth clock cycle. • Master mode: After the output of eight clock pulses, the clock output is set to the low level and clock stretching is set. • Slave mode: After the input of eight clock pulses, the clock is set to the low level and clock stretching is set for the master device.
1	An interrupt request is generated on the falling edge of the ninth clock cycle. • Master mode: After the output of nine clock pulses, the clock output is set to the low level and clock stretching is set. • Slave mode: After the input of nine clock pulses, the clock is set to the low level and clock stretching is set for the master device.
An interrupt is generated on the falling edge of the ninth clock cycle during address transfer independently of the setting of this bit. The setting of this bit is valid when the address transfer is completed. In master mode, clock stretching is inserted at the falling edge of the ninth clock cycle during address transfer. For a slave device that has received a local address, clock stretching is inserted at the falling edge of the ninth clock cycle after an acknowledge (ACK) is issued. However, if the slave device has received an extension code, clock stretching is inserted at the falling edge of the eighth clock cycle. When an address is received while the all-address match function is enabled, clock stretching is inserted at the falling edge of the eighth clock cycle.	
Condition for clearing (WTIMn = 0)	Condition for setting (WTIMn = 1)
• Cleared by instruction • Reset	• Set by instruction

Figure 16 - 7 Format of IICA Control Register n0 (IICCTLn0) (3/4)

ACKEn **Notes 4, 5**	Acknowledgment control
0	Disable acknowledgment.
1	Enable acknowledgment. During the ninth clock period, the SDAAn line is set to low level.

Condition for clearing (ACKEn = 0)	Condition for setting (ACKEn = 1)
• Cleared by instruction • Reset	• Set by instruction

STTn **Notes 2, 6**	Start condition trigger
0	Do not generate a start condition.
1	When bus is released (in standby state, when IICFn.IICBSYn = 0): If this bit is set to 1, a start condition is generated (startup as the master). When a third party is communicating: • When communication reservation function is enabled (IICFn.IICRSVn = 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 (IICFn.IICRSVn = 1) Even if this bit is set to 1, the STTn bit is cleared and the STTn clear flag (IICFn.STCFn) is set to 1. No start condition is generated. In the clock stretch state (for a master device): Generates a restart condition after release from the clock stretch state.

Cautions concerning set timing
- For master reception: Cannot be set to 1 during transfer. Can be set to 1 only during the clock stretch period when the ACKEn 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 clock stretch period that follows output of the ninth clock.
- Cannot be set to 1 at the same time as stop condition trigger (SPTn).
- Once STTn is set to 1, setting it to 1 again before the clear condition is met is not allowed.

Condition for clearing (STTn = 0)	Condition for setting (STTn = 1)
• Cleared by setting the STTn bit to 1 while communication reservation is prohibited. • Cleared by loss in arbitration • Cleared after start condition is generated by master device • Cleared by LRELn = 1 (exit from communications) • When IICEn = 0 (operation stop) • Reset	• Set by instruction

Figure 16 - 7 Format of IICA Control Register n0 (IICCTLn0) (4/4)

SPTn[Note 7]	Stop condition trigger
0	Stop condition is not generated.
1	Stop condition is generated (end of transfer as a master).

Cautions concerning set timing
- For master reception: Cannot be set to 1 during transfer. Can be set to 1 only during the clock stretch period when the ACKEn 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 clock stretch period that follows output of the ninth clock.
- Cannot be set to 1 at the same time as start condition trigger (STTn).
- The SPTn bit can be set to 1 only when in master mode.
- When the WTIMn bit has been cleared to 0, if the SPTn bit is set to 1 during the clock stretch period that follows output of eight clock pulses, note that a stop condition will be generated during the high-level period of the ninth clock after release from the clock stretch state. The WTIMn bit should be changed from 0 to 1 during the clock stretch period following the output of eight clock pulses, and the SPTn bit should be set to 1 during the clock stretch period that follows the output of the ninth clock.
- Once SPTn is set to 1, setting it to 1 again before the clear condition is met is not allowed.

Condition for clearing (SPTn = 0)	Condition for setting (SPTn = 1)
• Cleared by loss in arbitration • Automatically cleared after stop condition is detected • Cleared by LRELn = 1 (exit from communications) • When IICEn = 0 (operation stop) • Reset	• Set by instruction

Note 1. The IICA status register n (IICSn), the STCFn and IICBSYn flags of the IICA flag register n (IICFn), and the CLDn and DADn bits of IICA control register n1 (IICCTLn1) are reset.

Note 2. The setting of this bit has no effect while the setting of IICEn is 0.

Note 3. Reading the LRELn and WRELn bits always returns 0.

Note 4. The signal of this bit is invalid while IICEn is 0. Set this bit during that period.

Note 5. The set value is invalid during address transfer and if the code is not an extension code, and the all-address match function is disabled. When the device serves as a slave and the addresses match, an acknowledgment is generated regardless of the set value.

Note 6. The STTn bit is always read as 0.

Note 7. The SPTn bit is always read as 0.

Caution 1. If the operation of I^2C is enabled (IICEn = 1) when the SCLAn line is high level, the SDAAn line is low level, and the digital filter is turned on (IICCTLn1.DFCn = 1), a start condition will be inadvertently detected immediately. In this case, set the LRELn bit to 1 by using a 1-bit memory manipulation instruction immediately after enabling operation of I^2C (IICEn = 1).

Caution 2. When the IICSn.TRCn bit 1 is set to 1 (transmission state), the WRELn bit of the IICA control register n0 is set to 1 during the ninth clock and the interface is released from the clock stretch state, after which the TRCn bit is cleared (reception state) and the SDAAn line is set to the high impedance state. Release the interface from the clock stretch state while the TRCn bit is 1 (transmission state) by writing to the IICA shift register n.

Remark 1. The SPTn bit becomes 0 when it is read after data setting.

Remark 2. n = 0, 1

16.3.4 IICA status register n (IICSn)

This register indicates the state of the I^2C. The IICSn register can only be read by a 1-bit or 8-bit memory manipulation instruction while the setting of IICCTLn0.STTn is 1 or this module is in the clock stretch state. The value of this register following a reset is 00H.

Caution **Reading the IICSn register while the address match wakeup function is enabled (IICCTLn1.WUPn = 1) in STOP mode is prohibited. When the WUPn bit is changed from 1 to 0 (stopping operation of wakeup function), regardless of the INTIICAn interrupt request signal, a change in the state is not reflected until the next start condition or stop condition is detected. To use the wakeup function, therefore, enable (IICCTLn0.SPIEn = 1) the interrupt generated by detecting a stop condition and read the IICSn register after the interrupt has been detected.**

Remark n = 0, 1

Figure 16 - 8 Format of IICA Status Register n (IICSn) (1/4)

Address: FFF51H (IICS0), FFF55H (IICS1)
After reset: 00H
R/W: R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICSn	MSTSn	ALDn	EXCn	COIn	TRCn	ACKDn	STDn	SPDn

MSTSn	Master status check flag
0	Slave state or communications standby state
1	Master communications state

Condition for clearing (MSTSn = 0)	Condition for setting (MSTSn = 1)
• When a stop condition is detected • When ALDn = 1 (arbitration loss) • Cleared by IICCTLn0.LRELn = 1 (exit from communications) • When IICCTLn0.IICEn changes from 1 to 0 (operation stop) • Reset	• When a start condition is generated

ALDn	Detection of arbitration loss
0	This value indicates either that arbitration is not in progress or that the result of arbitration was a win.
1	This value indicates that the result of arbitration was a loss. The MSTSn flag is cleared when the ALDn bit has this value.

Condition for clearing (ALDn = 0)	Condition for setting (ALDn = 1)
• Automatically cleared after the IICSn register is read**Note 1** • When IICCTLn0.IICEn changes from 1 to 0 (operation stop) • Reset	• When the arbitration result is a “loss”.

Figure 16 - 8 Format of IICA Status Register n (IICSn) (2/4)

EXCn	Detection of extension code reception
0	Extension code was not received.
1	Extension code was received. Or, the all-address match function is enabled.

Condition for clearing (EXCn = 0)	Condition for setting (EXCn = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by IICCTLn0.LRELn = 1 (exit from communications) • When IICCTLn0.IICEn 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). • When an address is received while the all-address match function is enabled (IICCTLn1.SVADISn = 1) (set at the rising edge of the eighth clock).

COIn	Detection of matching addresses
0	Addresses do not match.
1	Addresses match. Or, the all-address match function is enabled.

Condition for clearing (COIn = 0)	Condition for setting (COIn = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by IICCTLn0.LRELn = 1 (exit from communications) • When IICCTLn0.IICEn changes from 1 to 0 (operation stop) • Reset	• When the received address matches the local address (slave address register n (SVAn)) (set at the rising edge of the eighth clock). • When an address is received while the all-address match function is enabled (IICCTLn1.SVADISn = 1) (set at the rising edge of the eighth clock).

Figure 16 - 8 Format of IICA Status Register n (IICSn) (3/4)

TRCn	Detection of transmission or reception
0	Reception is in progress (not transmission). The SDAAn line is set for high impedance.
1	Transmission is in progress. The SOn latch value is output to the SDAAn line (valid from the falling edge of the ninth clock of the first byte).

Condition for clearing (TRCn = 0)	Condition for setting (TRCn = 1)
<Both master and slave> • When a stop condition is detected • Cleared by IICCTLn0.LRELn = 1 (exit from communications) • When IICCTLn0.IICEn changes from 1 to 0 (operation stop) • Cleared by IICCTLn0.WRELn = 1 **Note 2** (release from the clock stretch state) • When the ALDn bit changes from 0 to 1 (arbitration loss) • Reset • When not used for communication (MSTSn, EXCn, COIn = 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 from the master (during address transfer)

ACKDn	Detection of acknowledge (ACK)
0	Acknowledge was not detected.
1	Acknowledge was detected.

Condition for clearing (ACKDn = 0)	Condition for setting (ACKDn = 1)
• When a stop condition is detected • At the rising edge of the first clock of the next byte • Cleared by IICCTLn0.LRELn = 1 (exit from communications) • When IICCTLn0.IICEn changes from 1 to 0 (operation stop) • Reset	• The SDAAn line is low at the rising edge of the ninth clock on the SCLAn line.

Figure 16 - 8 Format of IICA Status Register n (IICSn) (4/4)

STDn	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 (STDn = 0)	Condition for setting (STDn = 1)
• When a stop condition is detected • At the rising edge of the first clock of the byte following address transfer • Cleared by IICCTLn0.LRELn = 1 (exit from communications) • When IICCTLn0.IICEn changes from 1 to 0 (operation stop) • Reset	• When a start condition is detected

SPDn	Detection of stop condition
0	A stop condition has not been detected.
1	A stop condition has been detected. The master transfer has ended, and the bus has been released.

Condition for clearing (SPDn = 0)	Condition for setting (SPDn = 1)
• After this bit is set, at the rising edge of the first clock of the address transfer byte following detection of a start condition • When IICCTLn1.WUPn changes from 1 to 0 • When IICCTLn0.IICEn changes from 1 to 0 (operation stop) • Reset	• When a stop condition is detected

Note 1. The ALDn bit is also cleared when a 1-bit memory manipulation instruction is executed for another bit in the IICSn register. Therefore, when using the ALDn bit, read the data of this bit before the data of the other bits.

Note 2. When the TRCn flag of the IICA status register n (IICSn) is set to 1 (transmission state), setting the WRELn bit of the IICA control register n0 (IICCTLn0) to 1 at the ninth clock releases the interface from the clock stretch state, after which the TRCn flag is cleared (reception state) and the SDAAn line is set to the high impedance state. Release the interface from the clock stretch state while the TRCn bit is 1 (transmission state) by writing to the IICA shift register n.

Remark n = 0, 1

16.3.5 IICA flag register n (IICFn)

This register sets the operation mode of I²C and indicates the state of the I²C bus. The IICFn register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the STTn clear flag (STCFn) and I²C bus status flag (IICBSYn) bits are read-only. The IICRSVn bit can be used to enable or disable the communication reservation. The STCENn bit can be used to set the initial value of the IICBSYn flag. The IICRSVn and STCENn bits can be written only when the operation of I²C is disabled (IICCTLn0.IICEn = 0). The IICFn register is read-only while the operation of the I²C is enabled. The value of this register following a reset is 00H.

Remark n = 0, 1

Figure 16 - 9 Format of IICA Flag Register n (IICFn) (1/2)

Address: FFF52H (IICF0), FFF56H (IICF1)
After reset: 00H
R/W: R/W**Note**

Symbol	<7>	<6>	5	4	3	2	<1>	<0>
IICFn	STCFn	IICBSYn	0	0	0	0	STCENn	IICRSVn

STCFn	STTn clear flag
0	Generate start condition
1	Start condition generation unsuccessful: clear the STTn flag

Condition for clearing (STCFn = 0)	Condition for setting (STCFn = 1)
• Cleared by IICCTLn0.STTn = 1 • When IICCTLn0.IICEn = 0 (operation stop) • Reset	• Generating start condition unsuccessful and the IICCTLn0.STTn bit cleared to 0 when communication reservation is disabled (IICRSVn = 1).

IICBSYn	I²C bus status flag
0	Bus released state (initial communications state when STCENn = 1)
1	Bus communications state (initial communications state when STCENn = 0)

Condition for clearing (IICBSYn = 0)	Condition for setting (IICBSYn = 1)
• When a stop condition is detected • When IICCTLn0.IICEn = 0 (operation stop) • Reset	• When a start condition is detected • Setting of the IICCTLn0.IICEn bit when STCENn = 0

STCENn	Initial start enable trigger
0	After operation is enabled (IICCTLn0.IICEn = 1), enable generation of a start condition upon detection of a stop condition.
1	After operation is enabled (IICCTLn0.IICEn = 1), enable generation of a start condition without detecting a stop condition.

Condition for clearing (STCENn = 0)	Condition for setting (STCENn = 1)
• Cleared by instruction • When a start condition is detected • Reset	• Set by instruction

Figure 16 - 9 Format of IICA Flag Register n (IICFn) (2/2)

IICRSVn	Communication reservation function disable bit
0	Enable communication reservation
1	Disable communication reservation

Condition for clearing (IICRSVn = 0)	Condition for setting (IICRSVn = 1)
• Cleared by instruction • Reset	• Set by instruction

Note Bits 7 and 6 are read-only.

Caution 1. Write to the STCENn bit only when the operation is stopped (IICCTLn0.IICEn = 0).

Caution 2. As the bus is recognized as being in the released state (IICBSYn = 0) regardless of its actual state when STCENn = 1. When generating the first start condition (IICCTLn0.STTn = 1), it is necessary to verify that no third party communications are in progress in order to prevent such communications from being destroyed.

Caution 3. Write to the IICRSVn bit only when the operation is stopped (IICCTLn0.IICEn = 0).

Remark n = 0, 1

16.3.6 IICA control register n1 (IICCTLn1)

This register is used to set the operation mode of I2C and detect the states of the SCLAn and SDAAn pins. The IICCTLn1 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the CLDn and DADn bits are read-only. Set the IICCTLn1 register, except the WUPn bit, while operation of I2C is disabled (IICCTLn0.IICEn = 0). The value of this register following a reset is 00H.

Remark n = 0, 1

Figure 16 - 10 Format of IICA Control Register n1 (IICCTLn1) (1/2)

Address: F0231H (IICCTL01), F0239H (IICCTL11)
After reset: 00H
R/W: R/W**Note 1**

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	1	<0>
IICCTLn1	WUPn	SVADISn	CLDn	DADn	SMCn	DFCn	0	PRSn

WUPn	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 WUPn = 1, execute the STOP instruction at least three cycles of fMCK after setting the WUPn bit to 1. See **Figure 16 - 22 Flow for Setting WUPn to 1**. Clear the WUPn bit to 0 after the address has matched, an address has been received while the all-address match function is enabled, or an extension code has been received. The subsequent communication can be entered by clearing the WUPn bit to 0. The interface must be released from the clock stretch state and transmit data must be written after the WUPn bit has been cleared to 0. The interrupt timing when the address has matched, when an address has been received while the all-address match function is enabled, or when an extension code has been received, while WUPn = 1, is identical to the interrupt timing when WUPn = 0. A delay of the difference of sampling by the clock will occur. Furthermore, when WUPn = 1, a stop condition interrupt is not generated even if IICCTLn0.SPIEn is set to 1.	
Condition for clearing (WUPn = 0)	Condition for setting (WUPn = 1)
• Cleared by instruction (after address match, address reception with the all-address match function enabled, or extension code reception)	• Set by instruction (when the MSTSn, EXCn, and COIn bits are 0, and the STDn bit also 0 (communication not entered))**Note 2**

SVADISn	Address match disabling flag
0	Disables the all-address match function.
1	Enables the all-address match function.
When SVADISn = 1, IICA considers any address as address match, and performs the same operation as that when an extension code is received. Therefore, IICSn.COIn is set to 1, and IICSn.EXCn is set to 1. For details on extension code reception, see **16.5.11 Extension code**.	

Figure 16 - 10 Format of IICA Control Register n1 (IICCTLn1) (2/2)

CLDn	Detection of SCLAn pin level (valid only when IICCTLn0.IICEn = 1)
0	The SCLAn pin was detected at low level.
1	The SCLAn pin was detected at high level.

Condition for clearing (CLDn = 0)	Condition for setting (CLDn = 1)
• When the SCLAn pin is at low level • When IICCTLn0.IICEn = 0 (operation stop) • Reset	• When the SCLAn pin is at high level

DADn	Detection of SDAAn pin level (valid only when IICCTLn0.IICEn = 1)
0	The SDAAn pin was detected at low level.
1	The SDAAn pin was detected at high level.

Condition for clearing (DADn = 0)	Condition for setting (DADn = 1)
• When the SDAAn pin is at low level • When IICCTLn0.IICEn = 0 (operation stop) • Reset	• When the SDAAn pin is at high level

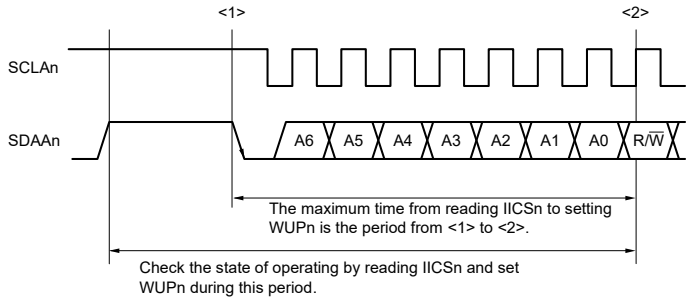
SMCn	Operation mode switching
0	Operates in standard mode (fastest transfer rate: 100 kbps).
1	Operates in fast mode (fastest transfer rate: 400 kbps) or fast mode plus (fastest transfer rate: 1 Mbps).

DFCn	Digital filter operation control
0	Digital filter off.
1	Digital filter on.
Use the digital filter only in fast mode and fast mode plus. The digital filter is used for noise elimination. The transfer clock does not vary, regardless of the DFCn bit being set to 1 or cleared to 0.	

PRSn	Control of IICA operation clock (f_{MCK})
0	Selects f_{CLK} ($1\ MHz \le f_{CLK} \le 20\ MHz$).
1	Selects $f_{CLK}/2$ ($20\ MHz < f_{CLK}$).

Note 1. Bits 5 and 4 are read-only.

Note 2. The state of the IICA status register n (IICSn) must be checked and the WUPn bit must be set during the period shown below.



(Cautions and Remark are listed on the next page.)

Caution 1. **The fastest operation frequency of the IICA operation clock (f_{MCK}) is 20 MHz (max.). Only set the PRSn bit to 1 when f_{CLK} exceeds 20 MHz.**

Caution 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 n = 0, 1

16.3.7 IICA low-level width setting register n (IICWLn)

This register is used to set the low-level width (t_{LOW}) of the SCLAn pin signal that is output by serial interface IICA and to control the SDAAn pin signal. The IICWLn register can be set by an 8-bit memory manipulation instruction. Set the IICWLn register while operation of I^2C is disabled (IICCTLn0.IICEn = 0). For details on setting the IICWLn register, see **16.4.2 Setting transfer clock by using IICWLn and IICWHn registers**. The data hold time is one-quarter of the time set by the IICWLn register. The value of this register following a reset is FFH.

Figure 16 - 11 Format of IICA Low-level Width Setting Register n (IICWLn)

Address: F0232H (IICWL0), F023AH (IICWL1)
After reset: FFH
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
IICWLn								

Remark 1. The minimum cycle of the serial clock is (IICWLn + 1) + (IICWHn + 1).
Remark 2. n = 0, 1

16.3.8 IICA high-level width setting register n (IICWHn)

This register is used to set the high-level width of the SCLAn pin signal that is output by serial interface IICA and to control the SDAAn pin signal. The IICWHn register can be set by an 8-bit memory manipulation instruction. Set the IICWHn register while operation of I^2C is disabled (IICCTLn0.IICEn = 0). For details on setting the IICWHn register, see **16.4.2 Setting transfer clock by using IICWLn and IICWHn registers**. The value of this register following a reset is FFH.

Figure 16 - 12 Format of IICA High-level Width Setting Register n (IICWHn)

Address: F0233H (IICWH0), F023BH (IICWH1)
After reset: FFH
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
IICWHn								

Remark 1. The minimum cycle of the serial clock is (IICWLn + 1) + (IICWHn + 1).
Remark 2. n = 0, 1

16.3.9 Registers for controlling the port functions multiplexed with the IICA serial interface inputs and outputs

Set the following registers to control the port functions multiplexed with the serial interface IICA inputs and outputs.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control E registers (PMCExx)

For details, see the following sections.

- **4.3.1 Port mode registers (PMxx)**
- **4.3.2 Port registers (Pxx)**
- **4.3.5 Port output mode registers (POMxx)**
- **4.3.7 Port mode control A registers (PMCAxx)**
- **4.3.9 Port mode control E registers (PMCExx)**

For examples of register settings for controlling port functions, see **4.5.4 Examples of register settings for port and multiplexed functions**.

16.4 I2C Bus Mode Functions

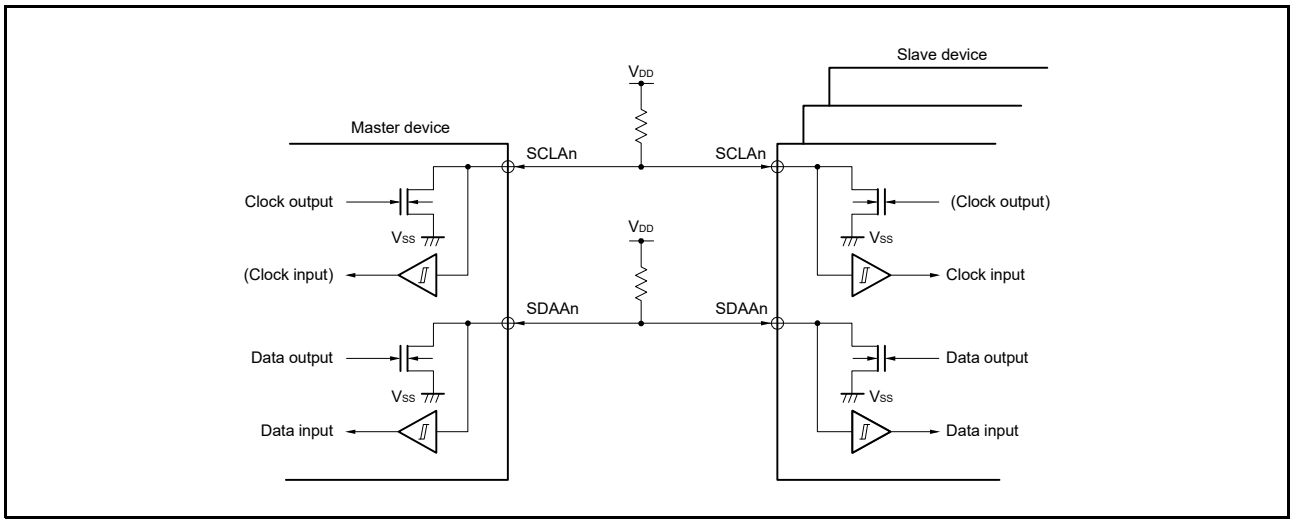
16.4.1 Pin configuration

The serial clock pin (SCLAn) and the serial data bus pin (SDAAn) are configured as follows.

1. SCLAn: 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. SDAAn: This pin is used for serial data input and output. This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open-drain outputs, an external pull-up resistor is required.

Figure 16 - 13 Pin Configuration Diagram



Remark n = 0, 1

16.4.2 Setting transfer clock by using IICWLn and IICWHn registers

1. Setting transfer clock on master side

$$\text{Transfer clock} = \frac{f_{MCK}}{IICWL + IICWH + f_{MCK}(t_R + t_F)}$$

At this time, the optimal setting values of the IICWLn and IICWHn registers are as follows.
(The fractional parts of all setting values are rounded up.)

- When the fast mode

$$IICWLn = \frac{0.52}{\text{Transfer clock}} \times f_{MCK}$$

$$IICWHn = \left(\frac{0.48}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

- When the normal mode

$$IICWLn = \frac{0.47}{\text{Transfer clock}} \times f_{MCK}$$

$$IICWHn = \left(\frac{0.53}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

- When the fast mode plus

$$IICWLn = \frac{0.50}{\text{Transfer clock}} \times f_{MCK}$$

$$IICWHn = \left(\frac{0.50}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

2. Setting IICWLn and IICWHn registers on slave side
(The fractional parts of all setting values are rounded up.)
   - When the fast mode
     $IICWLn = 1.3\ \mu s \times f_{MCK}$
     $IICWHn = (1.2\ \mu s - t_R - t_F) \times f_{MCK}$
   - When the normal mode
     $IICWLn = 4.7\ \mu s \times f_{MCK}$
     $IICWHn = (5.3\ \mu s - t_R - t_F) \times f_{MCK}$
   - When the fast mode plus
     $IICWLn = 0.50\ \mu s \times f_{MCK}$
     $IICWHn = (0.50\ \mu s - t_R - t_F) \times f_{MCK}$

Caution 1. The fastest operation frequency of the IICA operation clock (f_{MCK}) is 20 MHz (max.). Set the PRSn bit of the IICA control register n1 (IICCTLn1) to 1 only when f_{CLK} exceeds 20 MHz.

Caution 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.)

Caution 3. The minimum cycle of the serial clock is (IICWLn + 1) + (IICWHn + 1). Determine the values to be set in the IICWLn and IICWHn registers, taking into account the rising time (t_R) and falling time (t_F) of the SDAAn and SCLAn signals.

(Remarks are listed on the next page.)

Remark 1. Calculate the rise time (t_R) and fall time (t_F) of the SDAAn and SCLAn signals separately, because they differ depending on the pull-up resistor and wire load.

Remark 2. IICWLn: IICA low-level width setting register n
IICWHn: IICA high-level width setting register n
t_F: SDAAn and SCLAn signal falling times
t_R: SDAAn and SCLAn signal rising times
f_{MCK}: IICA operation clock frequency

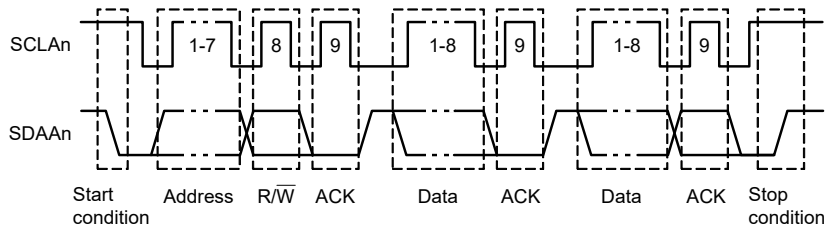
Remark 3. n = 0, 1

16.5 I2C Bus Definitions and Control Methods

This section describes the format of the I2C bus serial data communications and the functions of the signals used.

Figure 16 - 14 shows the transfer timing of the start condition, address, data, and stop condition generated on the I2C serial data bus.

Figure 16 - 14 I2C Bus Serial Data Transfer Timing

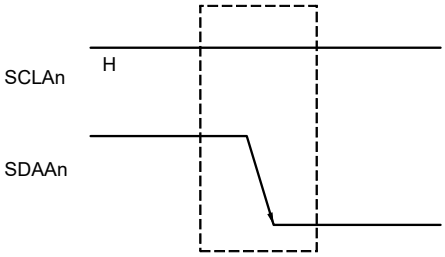


The master device generates the start condition, slave address, and stop condition. The acknowledge (ACK) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data). The serial clock (SCLAn) is continuously output by the master device. For the slave device, the period over which the SCLAn pin is at the low level can be extended and clock stretching can be inserted.

16.5.1 Start conditions

<R> When the SCLAn pin is at high level, changing the SDAAn pin from the high level to the low level generates a start condition. A start condition is a signal that the master device generates to the slave device when starting a serial transfer. When the device is used as a slave, start conditions can be detected.

Figure 16 - 15 Start Conditions



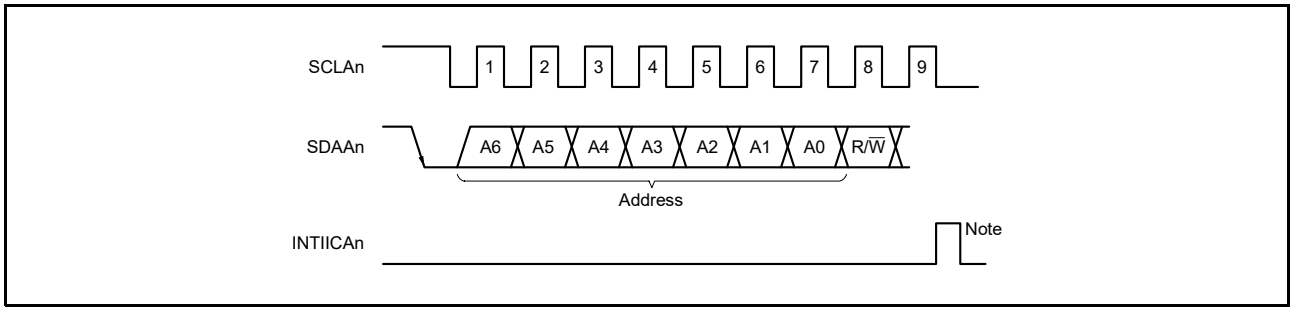
A start condition is output when the STTn bit of the IICA control register n0 (IICCTLn0) is set to 1 after a stop condition has been detected (IICSn.SPDn = 1). When a start condition is detected, the STDn bit of the IICSn register is set to 1.

Remark n = 0, 1

16.5.2 Address

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 n (SVAn). If the address data matches the SVAn register values, the slave device is selected and communicates with the master device until the master device generates a start condition or stop condition.

Figure 16 - 16 Addresses



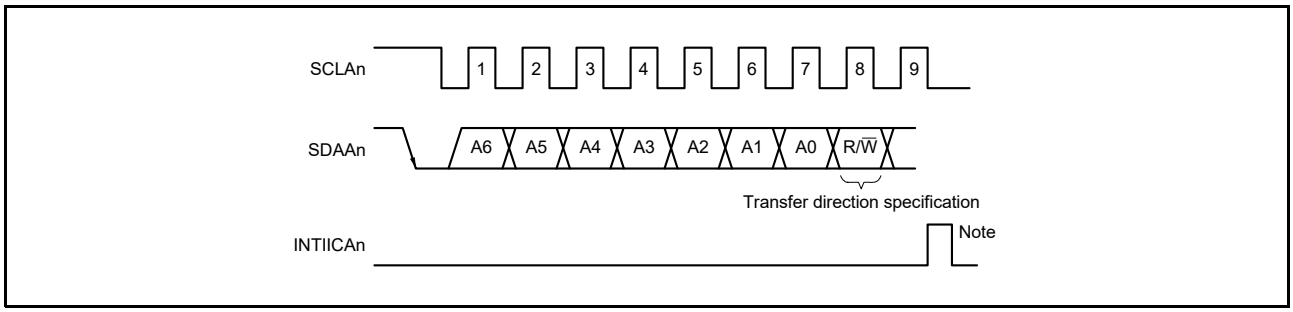
Note INTIICAn is not issued if data other than a local address or extension code is received while the all-address match function is disabled during slave device operation.

Addresses are output when a total of 8 bits consisting of the slave address and the transfer direction described in **16.5.3 Transfer direction specification** are written to the IICA shift register n (IICAn). The received addresses are written to the IICAn register. The slave address is assigned to the higher 7 bits of the IICAn register.

16.5.3 Transfer direction specification

In addition to the 7-bit address data, the master device sends 1 bit that specifies the transfer direction. When this transfer direction specification bit has a value of 0, it indicates that the master device is transmitting data to a slave device. When the transfer direction specification bit has a value of 1, it indicates that the master device is receiving data from a slave device.

Figure 16 - 17 Transfer Direction Specification



Note INTIICAn is not issued if data other than a local address or extension code is received while the all-address match function is disabled during slave device operation.

Remark n = 0, 1

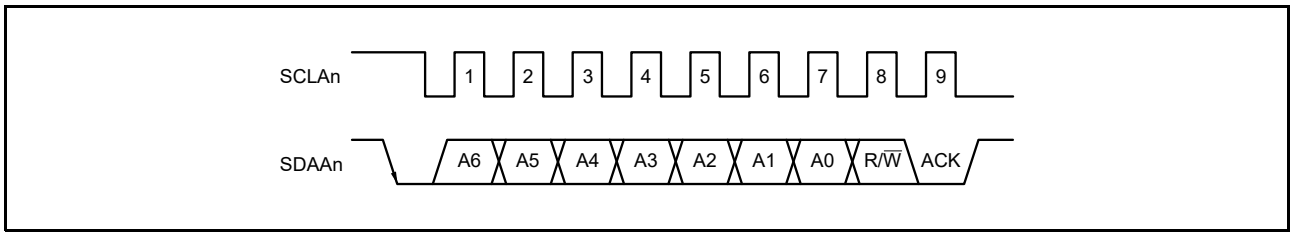
16.5.4 Acknowledge (ACK)

ACK is used to check the state 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 the ACKDn bit of the IICA status register n (IICSn). 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 SDAAn line low at the ninth clock (indicating normal reception). Automatic generation of ACK is enabled by setting the ACKEn bit of the IICA control register n0 (IICCTLn0) to 1. The TRCn bit of the IICSn register is set to the value of the eighth bit that follows 7-bit address information. Usually, set the ACKEn bit to 1 for reception (TRCn = 0). If a slave can receive no more data during reception (TRCn = 0) or does not require the next data item, then the slave must inform the master, by clearing the ACKEn bit to 0, that it will not receive any more data. When the master does not require the next data item during reception (TRCn = 0), it must clear the ACKEn 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 16 - 18 ACK



When the local address is received, ACK is automatically generated, regardless of the value of the ACKEn bit. When an address other than that of the local address is received, ACK is not generated (NACK). When an extension code is received, or when an address is received while the all-address match function is enabled, ACK is generated if the ACKEn bit is set to 1 in advance. How ACK is generated when data is received depends on the setting of the timing of clock stretching as follows.

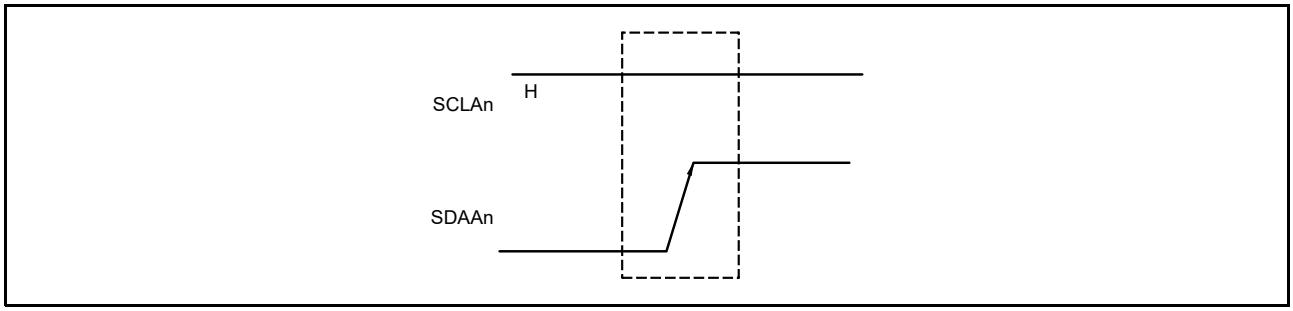
- When 8th cycle clock stretching is selected (IICCTLn0.WTIMn = 0):
  By setting the ACKEn bit to 1 before release from the clock stretch state, ACK is generated at the falling edge of the eighth clock cycle of the SCLAn pin.
- When 9th cycle clock stretching is selected (IICCTLn0.WTIMn = 1):
  ACK is generated if the ACKEn bit is set to 1 in advance.

Remark n = 0, 1

16.5.5 Stop condition

When the SCLAn pin is at high level, changing the SDAAn pin from low level to high level generates a stop condition. A stop condition is a signal that the master device generates to the slave device when serial transfer has been completed. When the device is used as a slave, stop conditions can be detected.

Figure 16 - 19 Stop Condition



A stop condition is generated when the SPTn bit of the IICA control register n0 (IICCTLn0) is set to 1. When the stop condition is detected, the SPTn bit of the IICA status register n (IICSn) is set to 1 and INTIICAn is generated when the SPIEn bit of the IICCTLn0 register is set to 1.

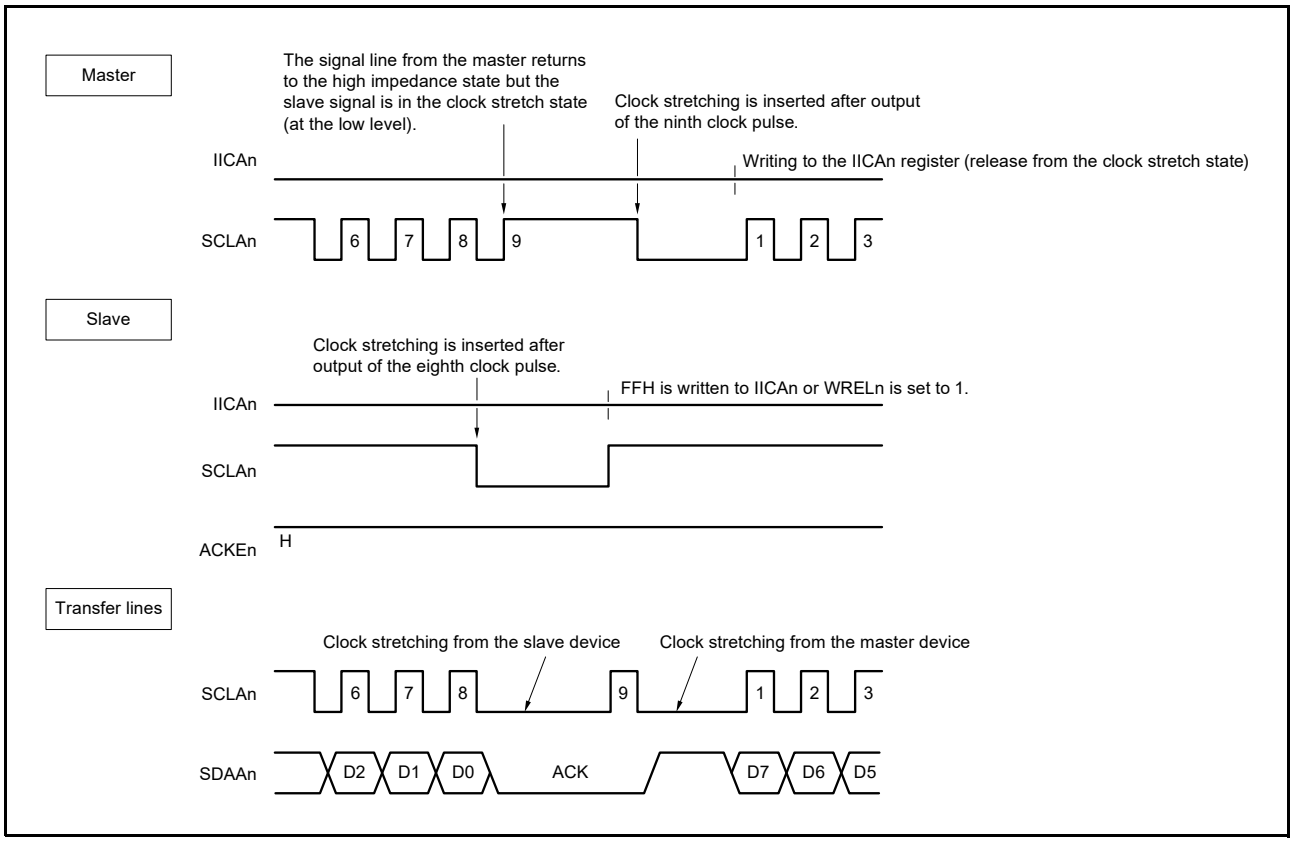
Remark n = 0, 1

16.5.6 Clock stretching

Clock stretching is used to notify the other party in communications that a device (master or slave) is preparing to transmit or receive data (i.e., the interface is in the clock stretch state). Setting the SCLAn pin to the low level indicates the clock stretch state to the other party. When clock stretching is released for both the master and slave devices, the next data transfer can start.

Figure 16 - 20 Clock Stretching (1/2)

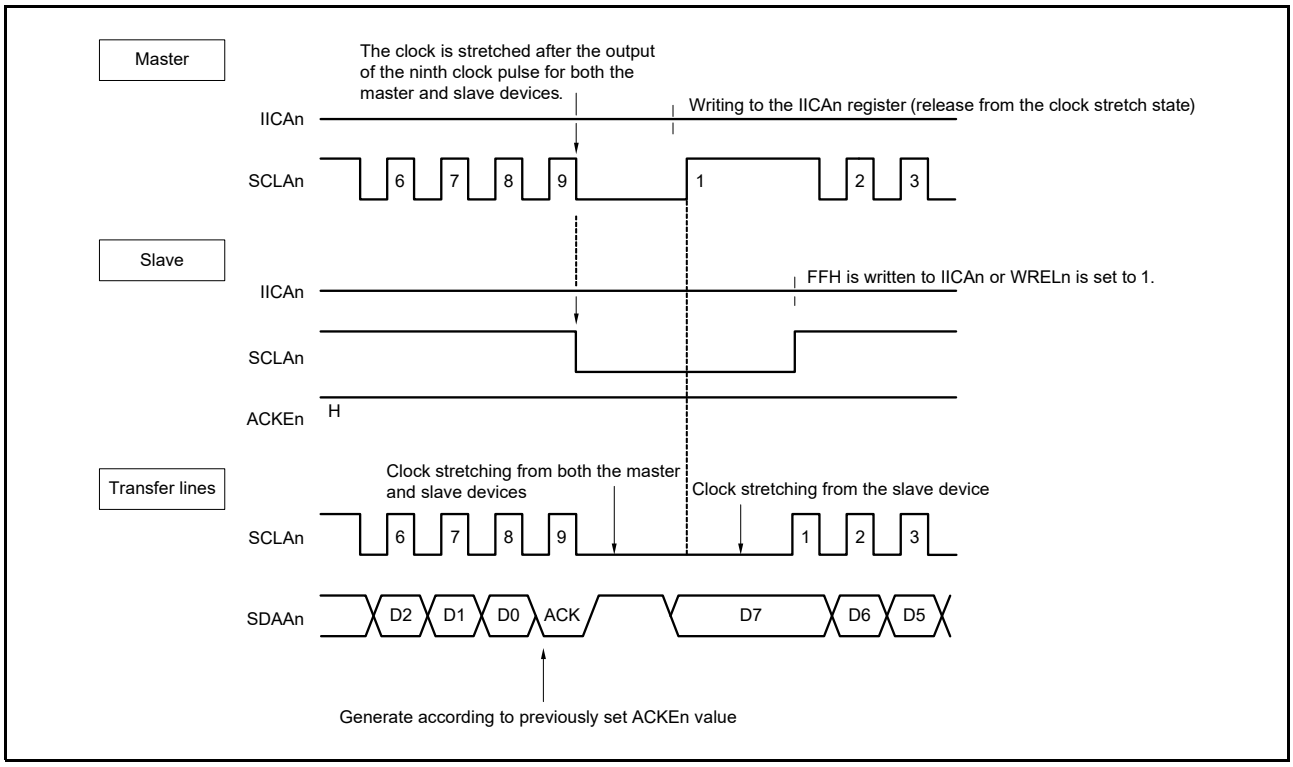
1. When clock stretching is set for the ninth and eighth clock cycles for the master and slave devices, respectively (master: transmission, slave: reception, and ACKEn = 1)



Remark n = 0, 1

Figure 16 - 20 Clock Stretching (2/2)

2. When clock stretching is set for the ninth clock cycle for both the master and slave devices (master: transmission, slave: reception, and ACKEn = 1)



Remark ACKEn: Bit 2 of IICA control register n0 (IICCTLn0)
WRELn: Bit 5 of IICA control register n0 (IICCTLn0)

Clock stretching is automatically generated depending on the setting of the WTIMn bit of the IICA control register n0 (IICCTLn0). Normally, the receiving side releases the clock stretch state when IICCTLn0.WRELn is set to 1 or when FFH is written to the IICA shift register n (IICAn), and the transmitting side releases the clock stretch state when data is written to the IICAn register. The master device can also releases the clock stretch state via either of the following methods.

- By setting IICCTLn0.STTn to 1
- By setting IICCTLn0.SPTn to 1

Remark n = 0, 1

16.5.7 Release from clock stretching

The I^2C interface usually releases the clock stretch state by the following processing.

- Writing data to the IICA shift register n (IICAn)
- Setting the WRELn bit of the IICA control register n0 (IICCTLn0) (release from the clock stretch state)
- Setting the STTn bit of the IICCTLn0 register (generating start condition)**Note**
- Setting the SPTn bit of the IICCTLn0 register (generating stop condition)**Note**

Executing the above processing for release from clock stretching leads to IICA releasing the clock stretch state after which communications are resumed. To release the clock stretch state and transmit data (including addresses), write the data to the IICAn register. To receive data after release from the clock stretch state, or to complete data transmission, set the WRELn bit of the IICCTLn0 register to 1. To generate a restart condition after release from the clock stretch state, set the STTn bit of the IICCTLn0 register to 1. To generate a stop condition after release from the clock stretch state, set the SPTn bit of the IICCTLn0 register to 1. Execute the processing for release only once for each period in the clock stretch state. If, for example, data is written to the IICAn register after release from the clock stretch state by setting the WRELn bit to 1, an incorrect value may be output to SDAAn line because the timing for changing the SDAAn line conflicts with the timing for writing the IICAn register.
In addition to the above, communications are stopped if the IICEn bit is cleared to 0 when communications have been aborted, so that the clock stretch state can be released. If the I^2C bus has deadlocked due to noise, the device can exit from communications by setting the LRELn bit of the IICCTLn0 register to 1, so that the clock stretch state can be released.

Note This setting is only required in the master device.

Caution **If the processing for release from clock stretching is executed when WUPn = 1, the clock stretch state will not be released.**

Remark n = 0, 1

16.5.8 Timing of generation of the interrupt request signal (INTIICAn) and control of clock stretching

The setting of the WTIMn bit of the IICA control register n0 (IICCTLn0) determines the timing by which INTIICAn is generated and controls clock stretching, as shown in **Table 16 - 2**.

Table 16 - 2 INTIICAn Generation Timing and Control of Clock Stretching

WTIMn	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, 3**	9**Note 2**	9**Note 2**	9	9	9

Note 1. In slave operation, the INTIICAn signal is generated and clock stretching occurs at the falling edge of the ninth clock only when the received address matches the address set in slave address register n (SVAn). ACK is generated regardless of the value set to the ACKEn bit of the IICCTLn0 register. For a slave device that has received an extension code, or has received an address while the all-address match function is enabled, INTIICAn occurs at the falling edge of the eighth clock. However, if the address does not match after restart, INTIICAn is generated at the falling edge of the ninth clock cycle, but clock stretching does not occur.

Note 2. If the received address does not match the contents of the slave address register n (SVAn), the all-address match function is disabled, and extension code is not received, neither INTIICAn nor clock stretching occurs.

Note 3. When the setting of the WTIMn bit is 1, for a slave device that has received an extension code, or has received an address while the all-address match function is enabled, INTIICAn and clock stretching occur at the falling edge of the eighth clock as well as on that of the ninth clock.

Remark The numbers in the table indicate the pulses of the serial clock signal. Interrupt requests and control of clock stretching are both synchronized with the falling edge of these clock pulses.

1. During address transmission/reception
   - Slave device operation: The timing of the interrupt and clock stretching depends on the conditions described in **Notes 1** to **3** above.
   - Master device operation: The interrupt and clock stretching occur at the falling edge of the ninth clock cycle, regardless of the setting of the WTIMn bit.
2. During data reception
   - Master/slave device operation: The timing of the interrupt and clock stretching depends on the setting of the WTIMn bit.
3. During data transmission
   - Master/slave device operation: The timing of the interrupt and clock stretching depends on the setting of the WTIMn bit.
4. Release from clock stretching

   The four types of processing for release from clock stretching are as follows.
   - Writing data to the IICA shift register n (IICAn)
   - Setting the WRELn bit of the IICA control register n0 (IICCTLn0) (release from the clock stretch state)
   - Setting the STTn bit of the IICCTLn0 register (generating start condition)**Note**
   - Setting the SPTn bit of the IICCTLn0 register (generating stop condition)**Note**

   When 8th cycle clock stretching has been selected (WTIMn = 0), the presence/absence of ACK generation must be determined before release from the clock stretch state.
5. Detection of stop condition

   INTIICAn is generated when a stop condition is detected (only when SPIEn = 1).

Note This setting is only required in the master device.

Remark n = 0, 1

16.5.9 Address match detection method

In I2C 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 signal (INTIICAn) occurs only when the address set to the slave address register n (SVAn) matches the slave address sent by the master device, when an address is received while the all-address match function is enabled (IICCTLn1.SVADISn = 1), or when an extension code has been received.

16.5.10 Error detection

In I2C bus mode, the state of the serial data bus (SDAAn) during data transmission is captured by the IICA shift register n (IICAn) 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.

Remark n = 0, 1

16.5.11 Extension code

1. When the higher 4 bits of the receive address are either 0000 or 1111, the detection of extension code reception flag (EXCn) is set to 1 for extension code reception and an interrupt request signal (INTIICAn) is issued at the falling edge of the eighth clock. When an address is received while the all-address match function is enabled, it is also determined that an extension code has been received. The local address stored in the slave address register n (SVAn) is not affected.

2. The settings below are specified if 11110xx0 is transferred from the master by using a 10-bit address transfer while the SVAn register is set to 11110xx0 or if an address is received while the all-address match function is enabled. Note that INTIICAn occurs at the falling edge of the eighth clock.
   - Higher four bits of data match or the all-address match function is enabled: IICSn.EXCn = 1
   - Seven bits of data match or the all-address match function is enabled: IICSn.COIn = 1

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 or an address is received with the all-address match function enabled during operation as a slave, 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 IICCTLn0.LRELn to 1 to set the standby mode for the next communication operation.

Table 16 - 3 Bit Definitions of Major Extension Codes

Slave Address	R/W Bit	Description
0 0 0 0 0 0 0	0	General call address
1 1 1 1 0 x x	0	10-bit slave address specification (during address authentication)
1 1 1 1 0 x x	1	10-bit slave address specification (after address match, when read command is issued)

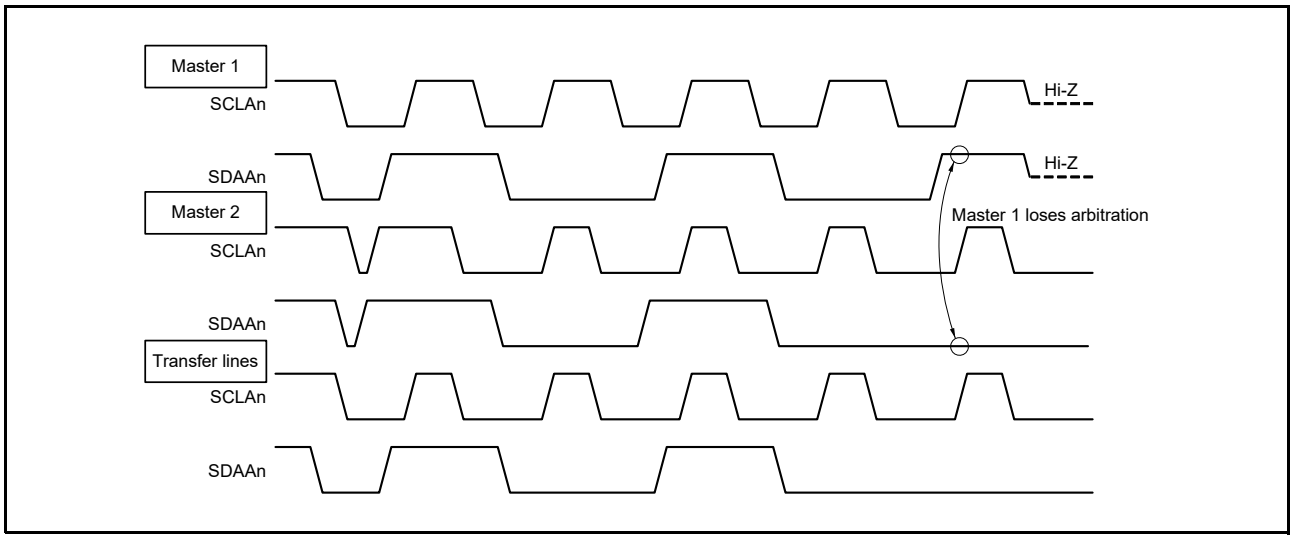
Remark 1. See the I^2C bus specifications issued by NXP Semiconductors for details of extension codes other than those described above.

Remark 2. n = 0, 1

16.5.12 Arbitration

When several master devices simultaneously generate a start condition (when IICCTLn0.STTn is set to 1 before IICSn.STDn 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 (ALDn) in the IICA status register n (IICSn) is set to 1 via the timing by which the arbitration loss occurred, and the SCLAn and SDAAn lines are both set to high impedance, which releases the bus. The arbitration loss is detected by checking ALDn = 1 by software at the timing of the next interrupt request (the eighth or ninth clock cycle, when a stop condition is detected, etc.). For details of interrupt request timing, see **16.5.8 Timing of generation of the interrupt request signal (INTIICAn) and control of clock stretching**.

Figure 16 - 21 Arbitration Timing Example



Remark n = 0, 1

Table 16 - 4 State during Arbitration and Interrupt Request Generation Timing

State 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 SPIEn = 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 SPIEn = 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 SCLAn is at low level while attempting to generate a restart condition	

<R> **Note 1.** When IICCTLn0.WTIMn is 1, an interrupt request is generated at the falling edge of the ninth clock. When WTIMn is 0, an interrupt request is generated at the falling edge of the eighth clock if either the slave address for an extension code is received or an address is received while the all-address match function is enabled.

Note 2. If arbitration may occur, set SPIEn to 1 in master operation.

Remark 1. SPIEn: Bit 4 of IICA control register n0 (IICCTLn0)

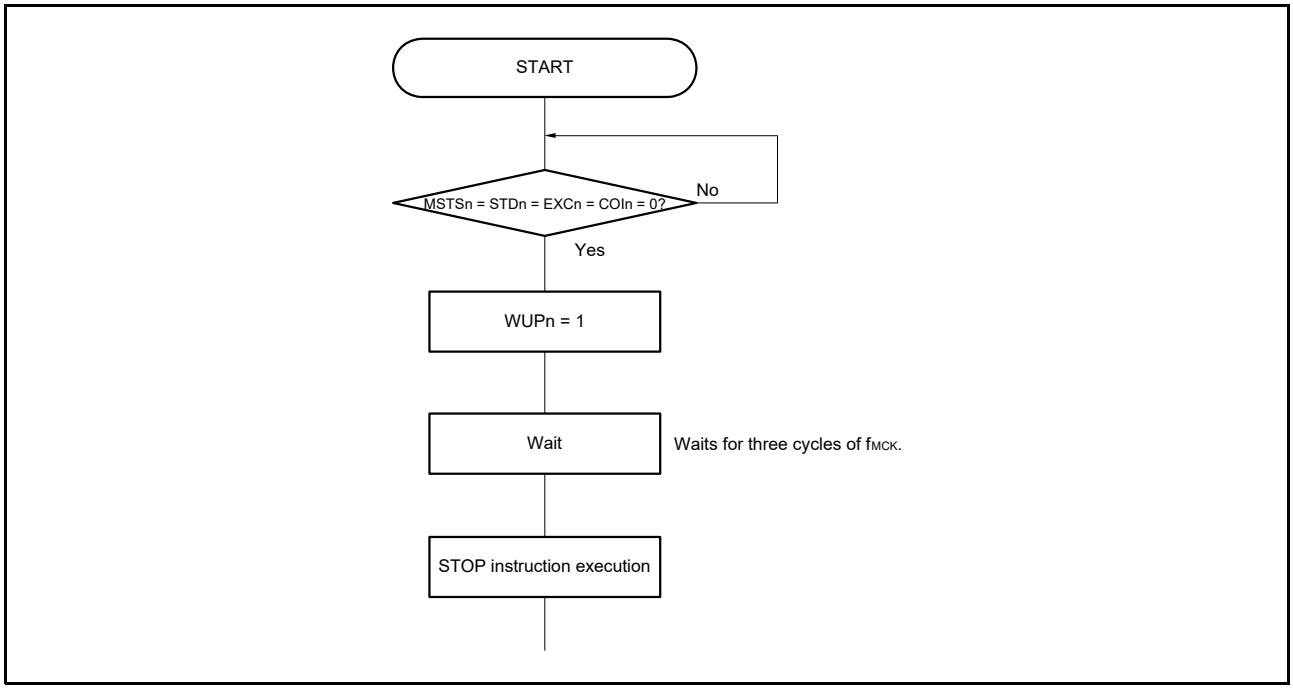
Remark 2. n = 0, 1

16.5.13 Wakeup function

This function generates an interrupt request signal (INTIICAn) in slave mode when the I^2C bus interface receives its own slave address, receives any address while the all-address match function is enabled, or receives an extension code. This function makes processing more efficient by preventing the generation of unnecessary INTIICAn signals when addresses do not match while the all-address match function is disabled. On detection of a start condition, the module enters the wakeup standby mode. Even a master, if it has generated a start condition, enters the wakeup standby state while transmitting an address because the master may become a slave due to a loss in arbitration. To use the wakeup function in the STOP mode, set the WUPn bit to 1. Addresses can be received regardless of the operating clock. An interrupt request signal (INTIICAn) is also generated when the local address is received, any address is received while the all-address match function is enabled, or an extension code is received. Normal operation is made to resume by using an instruction to clear the WUPn bit to 0 after this interrupt has been generated.

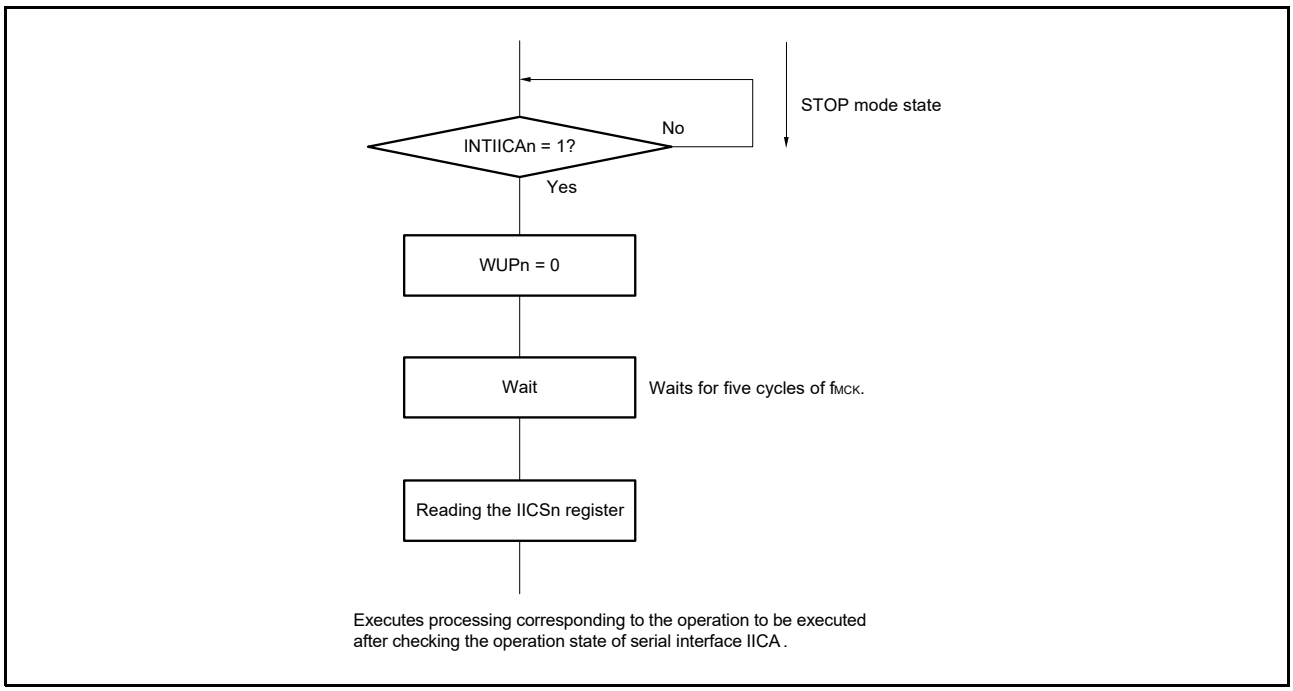
Figure 16 - 22 shows the flow for setting WUPn to 1 and **Figure 16 - 23** shows the flow for setting WUPn to 0 when the address matches or any address or extension code is received while the all-address match function is enabled.

Figure 16 - 22 Flow for Setting WUPn to 1



Remark n = 0, 1

Figure 16 - 23 Flow for Setting WUPn to 0 When the Address Matches or Any Address or Extension Code Is Received While the All-Address Match Function Is Enabled



Use the following flows to perform the processing to release the STOP mode other than by an interrupt request signal (INTIICAn) generated from serial interface IICA.

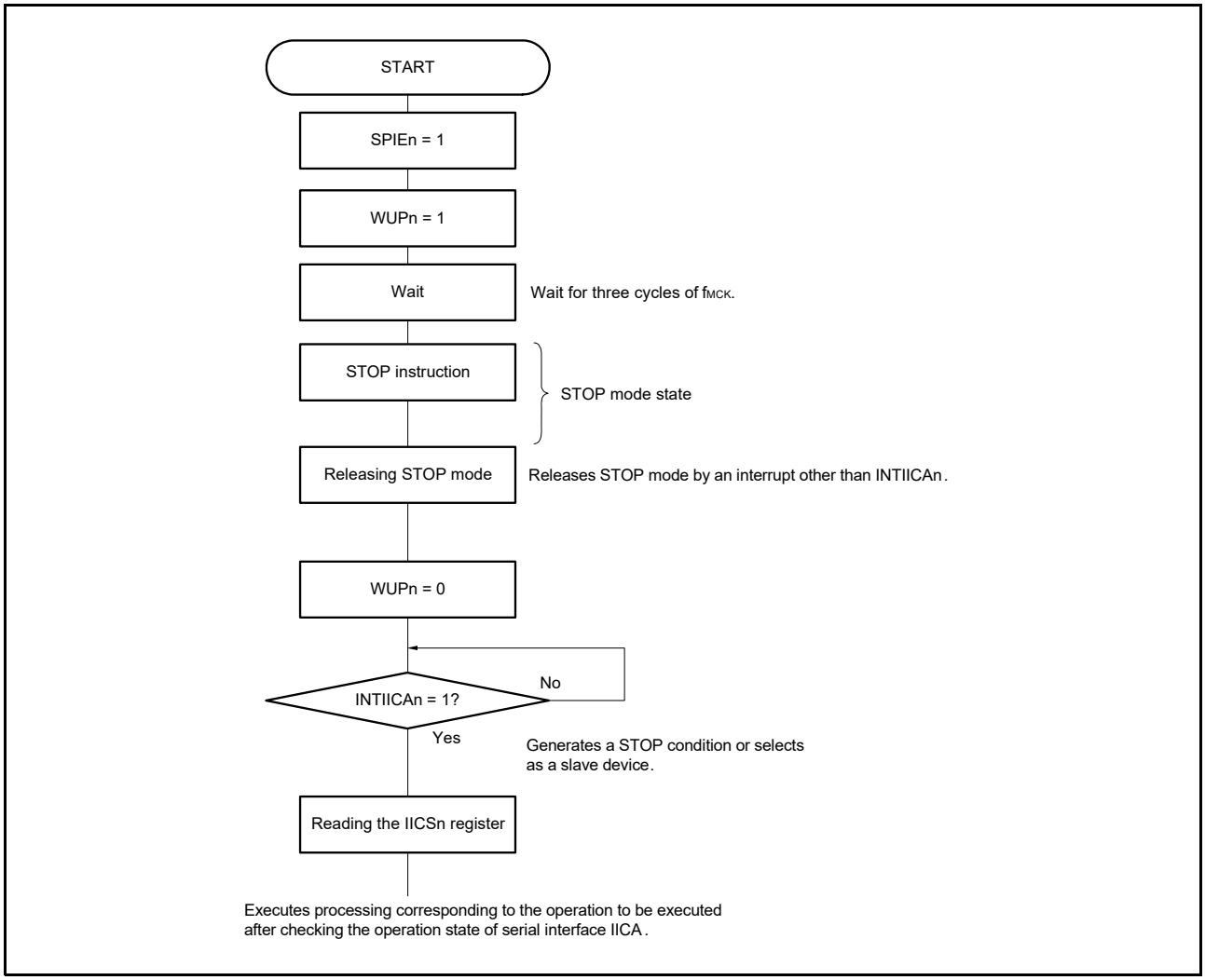
- When operating next IIC communication as master: Flow shown in **Figure 16 - 24**.
- When operating next IIC communication as slave:

  When released by INTIICAn interrupt: Same as the flow in **Figure 16 - 23**.

  When released by other than INTIICAn interrupt: Wait for INTIICAn interrupt with WUPn left set to 1.

Remark n = 0, 1

Figure 16 - 24 Flow for Using This MCU as the Master after STOP Mode Is Released by an Interrupt Other than INTIICAn



Remark n = 0, 1

16.5.14 Communication reservation

1. When communication reservation function is enabled (IICFn.IICRSVn = 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
   - While the all-address match function is disabled, when an extension code is received and slave operation is disabled (ACK is not returned and the bus was released by setting IICCTLn0.LRELn to 1 and exiting from communication)

   If the STTn bit of the IICCTLn0 register is set to 1 while the bus is not used, a start condition is automatically generated and wait state is entered after the bus is released (after a stop condition is detected). If an address is written to the IICA shift register n (IICAn) after the SPIEn bit of the IICCTLn0 register was set to 1, and it was detected by generation of an interrupt request signal (INTIICAn) that the bus was released (detection of the stop condition), then the device automatically starts communication as the master. Data written to the IICAn register before the stop condition is detected is invalid. When the STTn bit has been set to 1, the operation mode (as start condition or as communication reservation) is determined according to the state of the bus.

   - 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 MSTSn flag (bit 7 of the IICA status register n (IICSn)) after the STTn bit is set to 1 and the wait time elapses. Use software to secure the wait time calculated by the following expression.

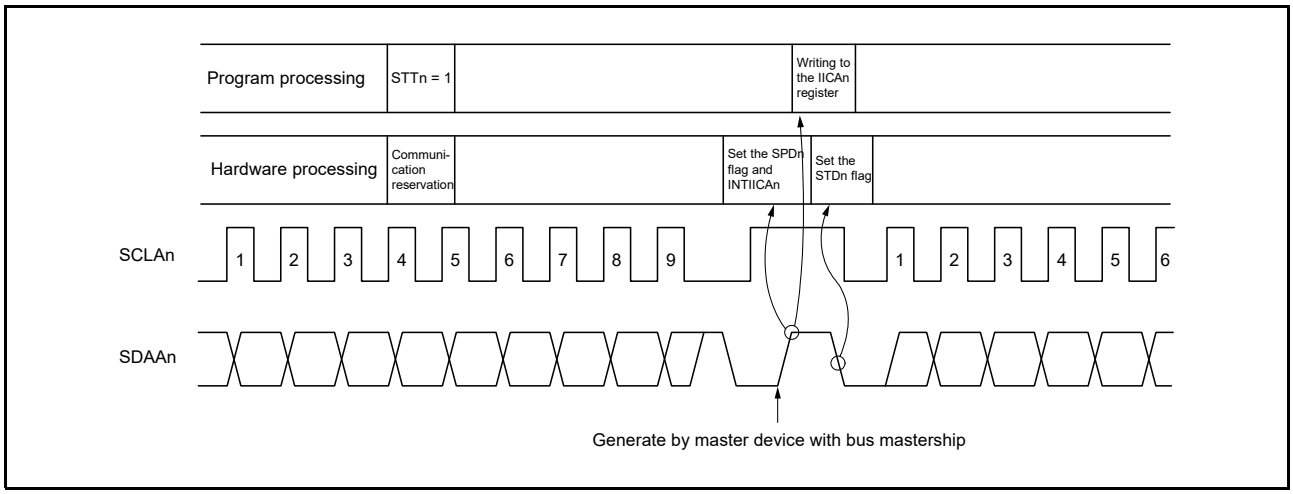
Wait time from setting STTn = 1 to checking the MSTSn flag:
(IICWLn setting value + IICWHn setting value + 4)/f_{MCK} + t_F × 2

   Remark 1. IICWLn: IICA low-level width setting register n
   IICWHn: IICA high-level width setting register n
   t_F: SDAAn and SCLAn signal falling times
   f_{MCK}: IICA operation clock frequency

   Remark 2. n = 0, 1

Figure 16 - 25 shows the communication reservation timing.

Figure 16 - 25 Communication Reservation Timing

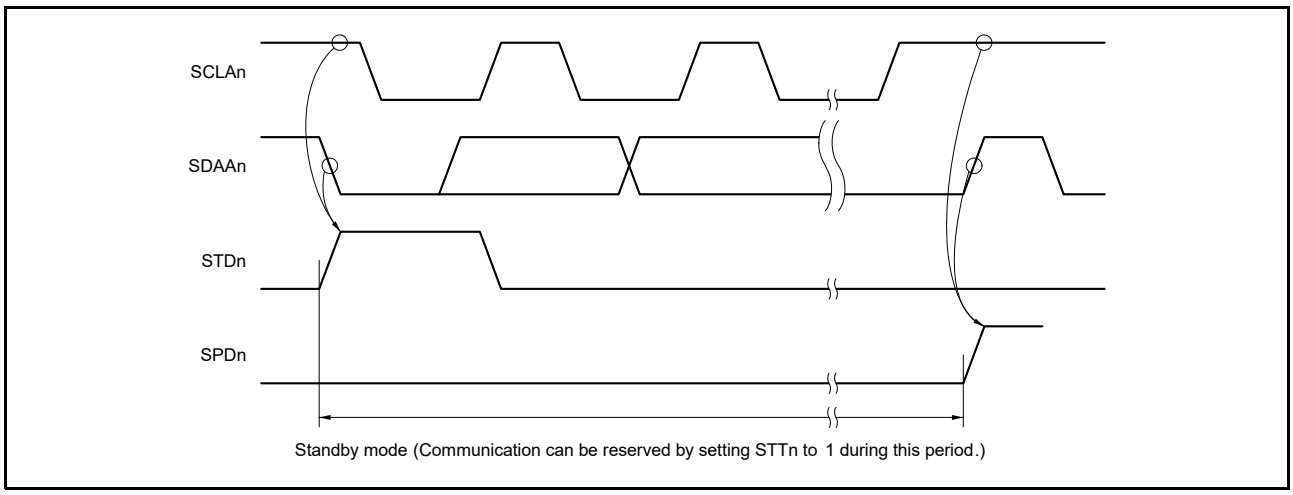


Remark 1. IICAn: IICA shift register n
STTn: Bit 1 of IICA control register n0 (IICCTLn0)
STDn: Bit 1 of IICA status register n (IICSn)
SPDn: Bit 0 of IICA status register n (IICSn)

Remark 2. n = 0, 1

Communication reservations are accepted via the timing shown in **Figure 16 - 26**. After IICSn.STDn is set to 1, a communication reservation can be made by setting IICCTLn0.STTn to 1 before a stop condition is detected.

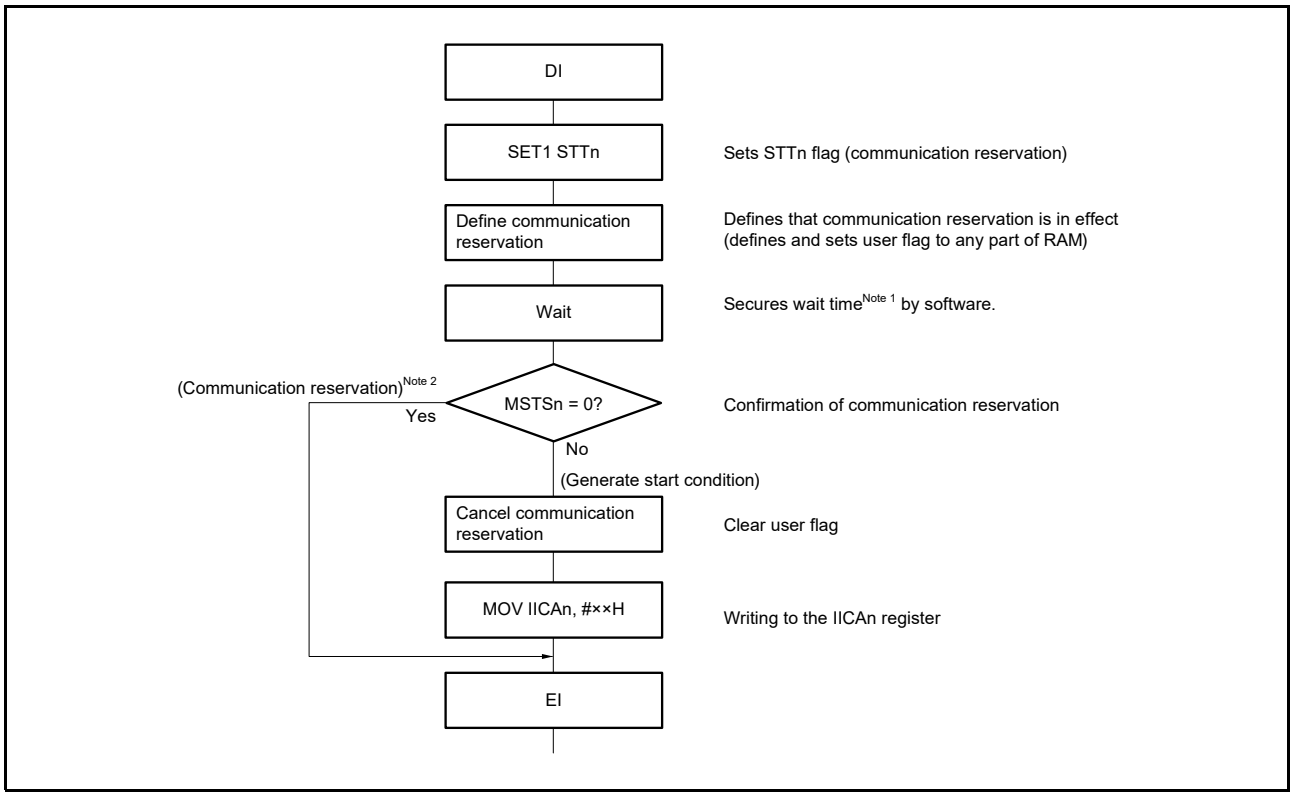
Figure 16 - 26 Timing for Accepting Communication Reservations



Remark n = 0, 1

Figure 16 - 27 shows the communication reservation protocol.

Figure 16 - 27 Communication Reservation Protocol



Note 1. The wait time is calculated as follows.
(IICWLn setting value + IICWHn setting value + 4)/f_{MCK} + t_F × 2

Note 2. The communication reservation operation executes a write to the IICA shift register n (IICAn) when a stop condition interrupt request occurs.

Remark 1. STTn: Bit 1 of IICA control register n0 (IICCTLn0)
MSTSn: Bit 7 of IICA status register n (IICSn)
IICAn: IICA shift register n
IICWLn: IICA low-level width setting register n
IICWHn: IICA high-level width setting register n
t_F: SDAAn and SCLAn signal falling times
f_{MCK}: IICA operation clock frequency

Remark 2. n = 0, 1

2. When communication reservation function is disabled (IICFn.IICRSVn = 1)
   When the STTn bit of the IICA control register n0 (IICCTLn0) 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 states where the bus is not in use consist of the following two states.

   - When arbitration results in neither master nor slave operation
   - While the all-address match function is disabled, when an extension code is received and slave operation is disabled (ACK is not returned and the bus was released by setting IICCTLn0.LRELn to 1 and exiting from communication)

   To confirm whether the start condition was generated or request was rejected, check the STCFn flag of the IICFn register. It takes up to 5 cycles of f_{MCK} until the STCFn flag is set to 1 after setting STTn = 1. Therefore, secure the time by software.

   Remark n = 0, 1

16.5.15 Cautions

1. When STCENn = 0
   Immediately after I^2C operation is enabled (IICEn = 1), the bus is recognized as being in a communications state (IICBSYn = 1) regardless of its actual state. 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 n1 (IICCTLn1).
   <2> Set the IICEn bit of the IICA control register n0 (IICCTLn0) to 1.
   <3> Set the SPTn bit of the IICCTLn0 register to 1.

2. When STCENn = 1
   Immediately after I^2C operation is enabled (IICEn = 1), the bus is recognized as being in the released state (IICBSYn = 0) regardless of its actual state. To generate the first start condition (STTn = 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 SDAAn pin is low and the SCLAn pin is high, the IICA recognizes that the SDAAn pin has gone low (detects a start condition). If the value on the bus at this time can be recognized as an extension code or the all-address match function is enabled, ACK is returned, but this interferes with other I^2C communications. To avoid this, start the IICA in the following sequence.
   <1> Clear the SPIEn bit of the IICCTLn0 register to 0 to disable generation of an interrupt request signal (INTIICAn) when the stop condition is detected.
   <2> Set the IICEn bit of the IICCTLn0 register to 1 to enable the operation of the IICA.
   <3> Wait for detection of the start condition.
   <4> Set the LRELn bit of the IICCTLn0 register to 1 before ACK is returned (4 to 72 cycles of f_{MCK} after setting the IICEn bit to 1), to forcibly disable detection.

4. Setting the STTn and SPTn bits (bits 1 and 0 of the IICCTLn0 register) again after they are set and before they are cleared to 0 is prohibited.

5. When transmission is reserved, set the SPIEn bit (bit 4 of the IICCTLn0 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 n (IICAn) 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 SPIEn bit to 1 when the MSTSn flag of the IICA status register n (IICSn) is detected by software.

Remark n = 0, 1

16.5.16 Communication operations

The following shows three operation procedures with the flowchart.

1. Master operation in single-master system
   The flowchart when using a product of this group 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.

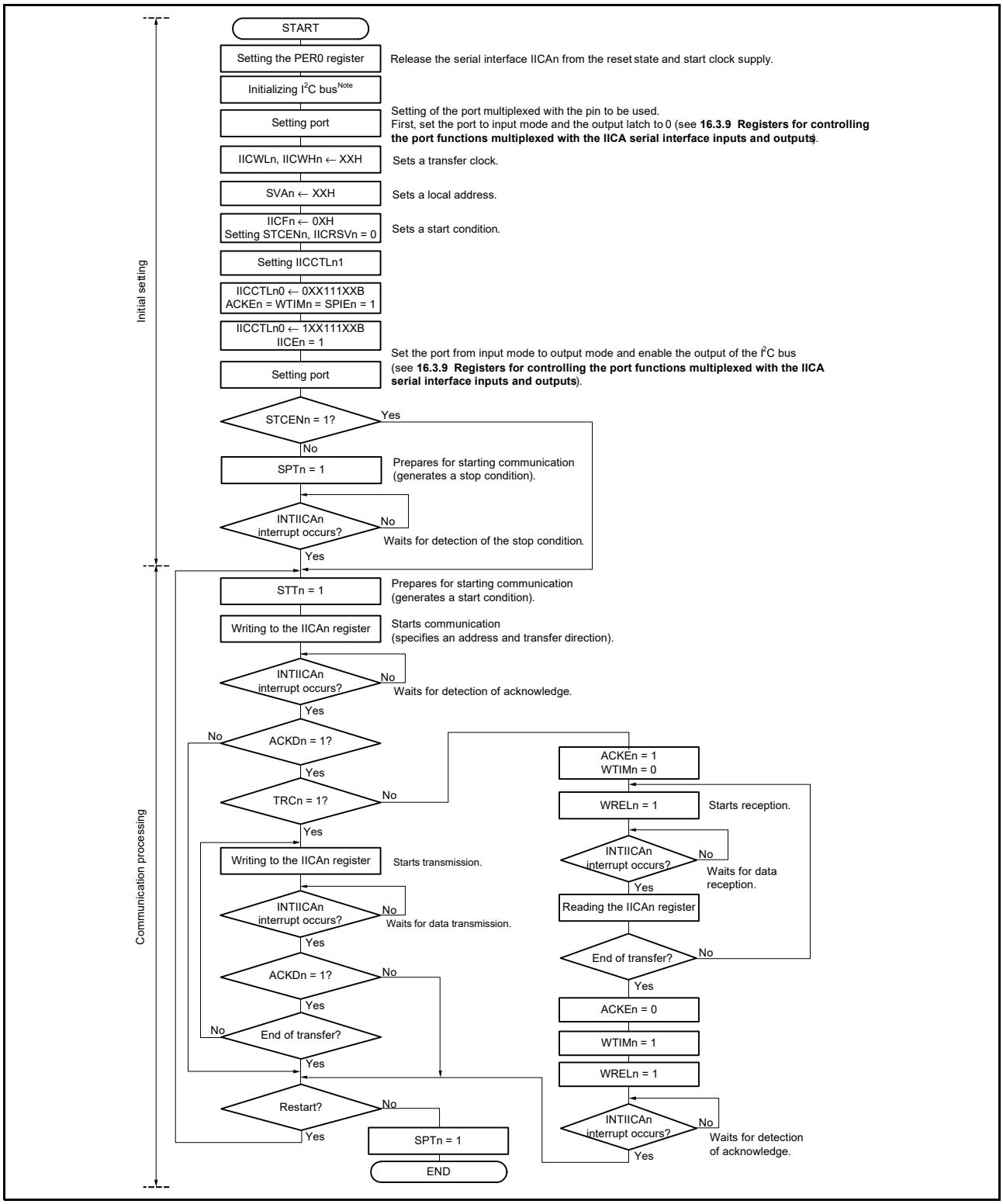
2. Master operation in multi-master system
   In the I^2C bus multi-master system, whether the bus is released or used cannot be judged by the I^2C bus specifications when a device takes part in a communication. Here, when data and clock are at a high level for a certain period (1 frame), a product of this group takes 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 product loses 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 a product of this group is 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 INTIICAn interrupt occurrence (communication waiting). When an INTIICAn interrupt occurs, the communications state is judged and the result is passed as a flag over to the main processing. By checking the flags, necessary communication processing is performed.

Remark n = 0, 1

1. Master operation in single-master system

Figure 16 - 28 Master Operation in Single-master System



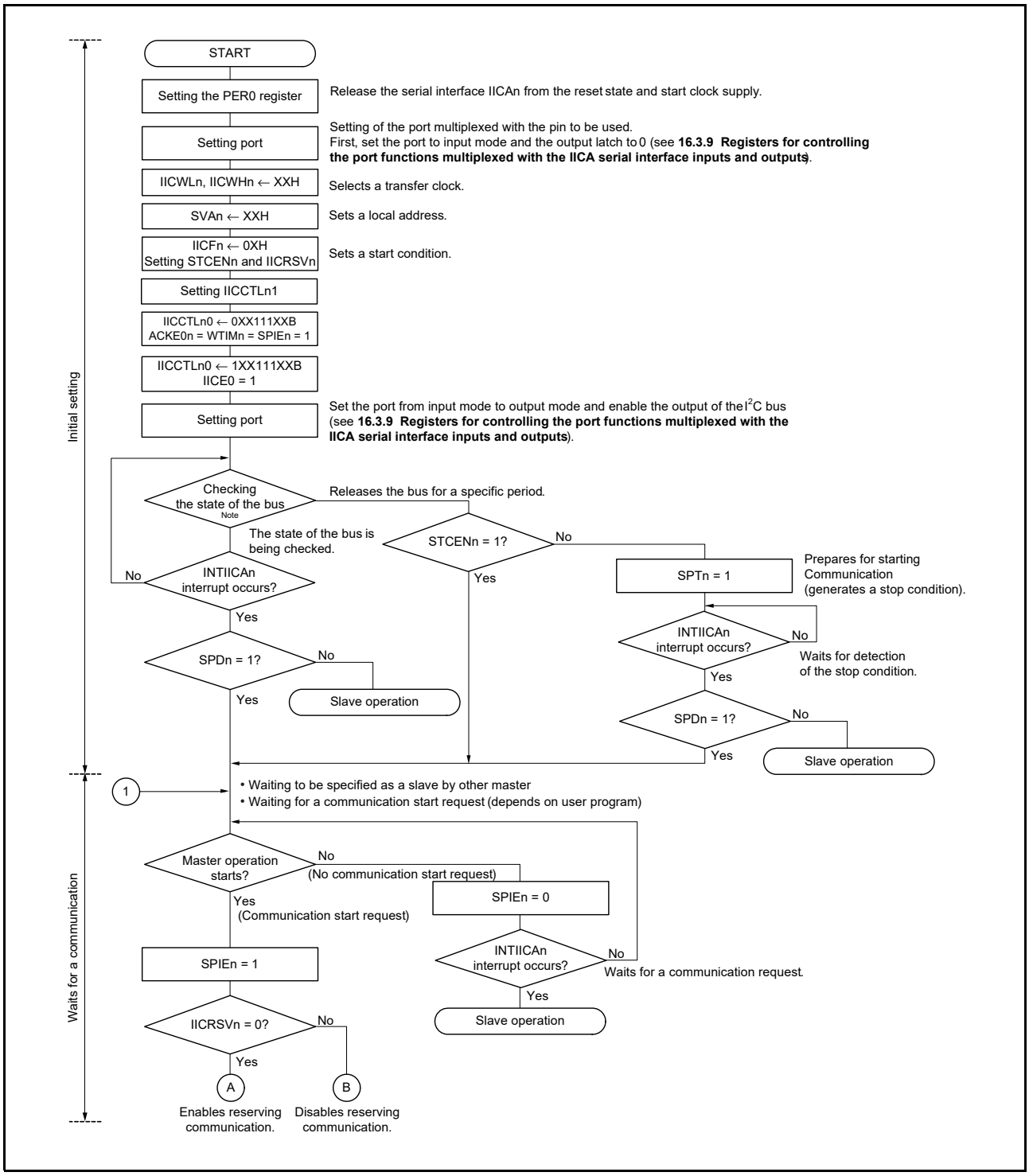
Note Release (SCLAn and SDAAn pins = high level) the I2C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDAAn pin, for example, set the SCLAn pin in the output port mode, and output a clock pulse from the output port until the SDAAn pin is constantly at high level.

Remark 1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

Remark 2. n = 0, 1

2. Master operation in multi-master system

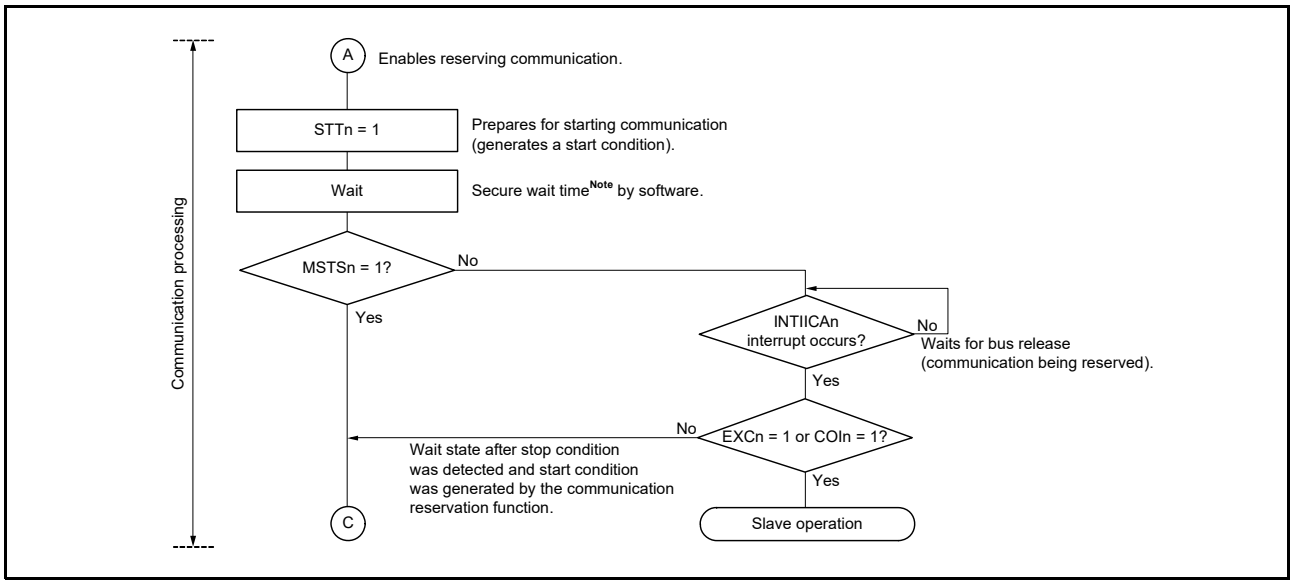
Figure 16 - 29 Master Operation in Multi-master System (1/3)



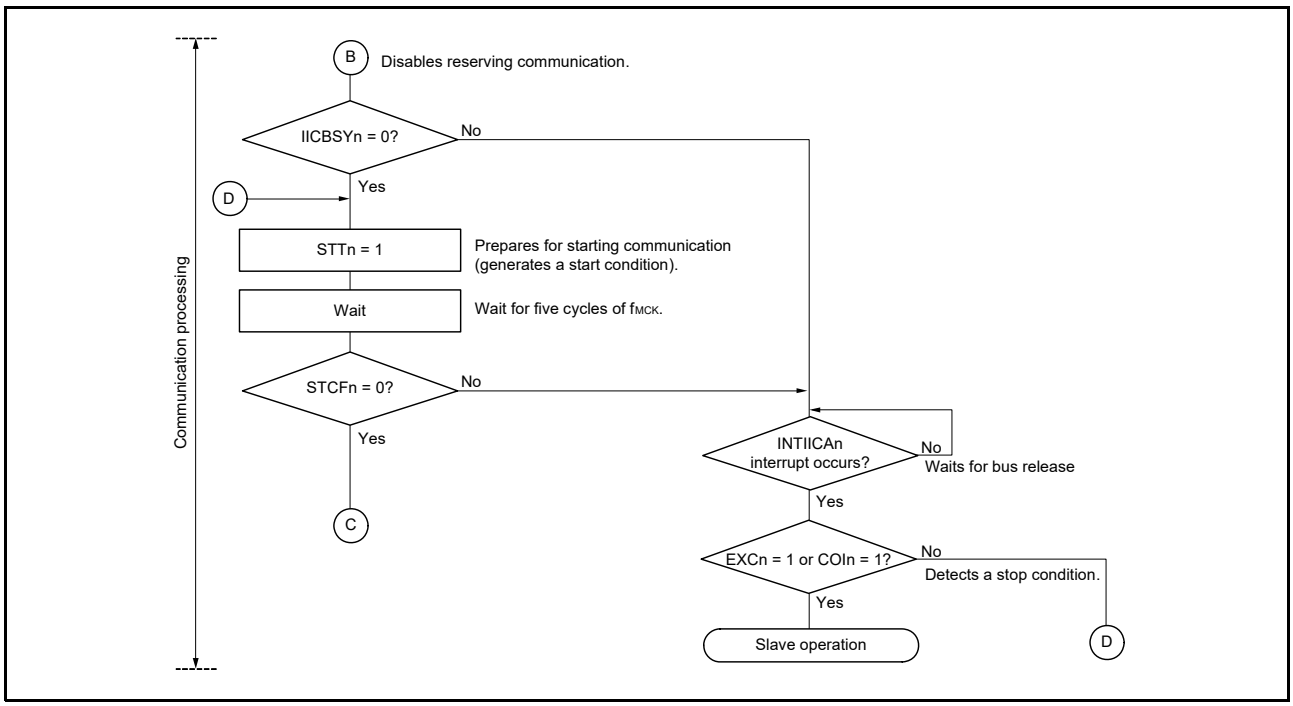
Note Confirm that the bus is released (CLDn = 1, DADn = 1) for a specific period (for example, for a period of one frame). If the SDAAn pin is constantly at low level, decide whether to release the I^2C bus (SCLAn and SDAAn pins = high level) in conformance with the specifications of the product that is communicating.

Remark n = 0, 1

Figure 16 - 29 Master Operation in Multi-master System (2/3)



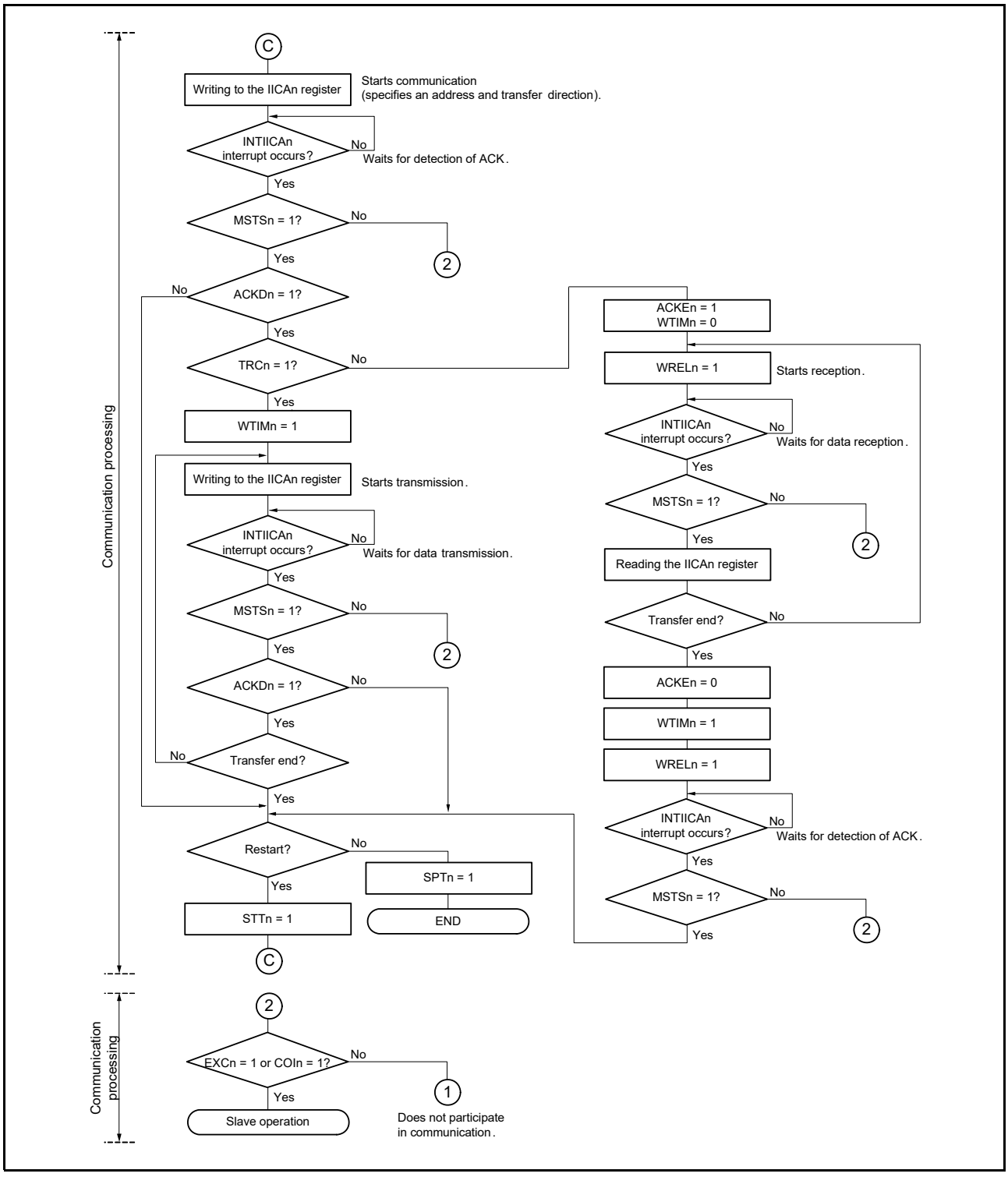
Note The wait time is calculated as follows.
(IICWLn setting value + IICWHn setting value + 4)/f_{MCK} + t_F × 2



Remark 1. IICWLn: IICA low-level width setting register n
IICWHn: IICA high-level width setting register n
t_F: SDAAn and SCLAn signal falling times
f_{MCK}: IICA operation clock frequency

Remark 2. n = 0, 1

Figure 16 - 29 Master Operation in Multi-master System (3/3)



Remark 1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

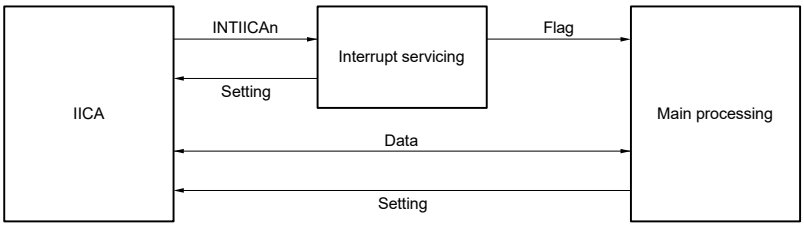
Remark 2. To use the device as a master in a multi-master system, read the MSTSn flag each time interrupt INTIICAn has occurred to check the arbitration result.

Remark 3. To use the device as a slave in a multi-master system, check the state by using the IICA status register n (IICSn) and IICA flag register n (IICFn) each time interrupt INTIICAn has occurred, and determine the processing to be performed next.

Remark 4. n = 0, 1

3. Slave operation

The processing procedure of the slave operation is as follows. Basically, the slave operation is event-driven. Therefore, processing by the INTIICAn interrupt (processing that must substantially change the state of operating such as detection of a stop condition during communication) is necessary. In the following explanation, it is assumed that the all-address match function is disabled and the extension code is not supported for data communication. It is also assumed that the INTIICAn interrupt servicing routine only handles state transition processing, and that the main processing handles actual data communications.



Therefore, data communication processing is performed by preparing the following three flags and passing them to the main processing instead of INTIICAn.

<1> Communication mode flag

This flag indicates the following two communications states.

- Clear mode: State in which data communications are not in progress
- Communication mode: State in which data communications are in progress (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 INTIICAn 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).

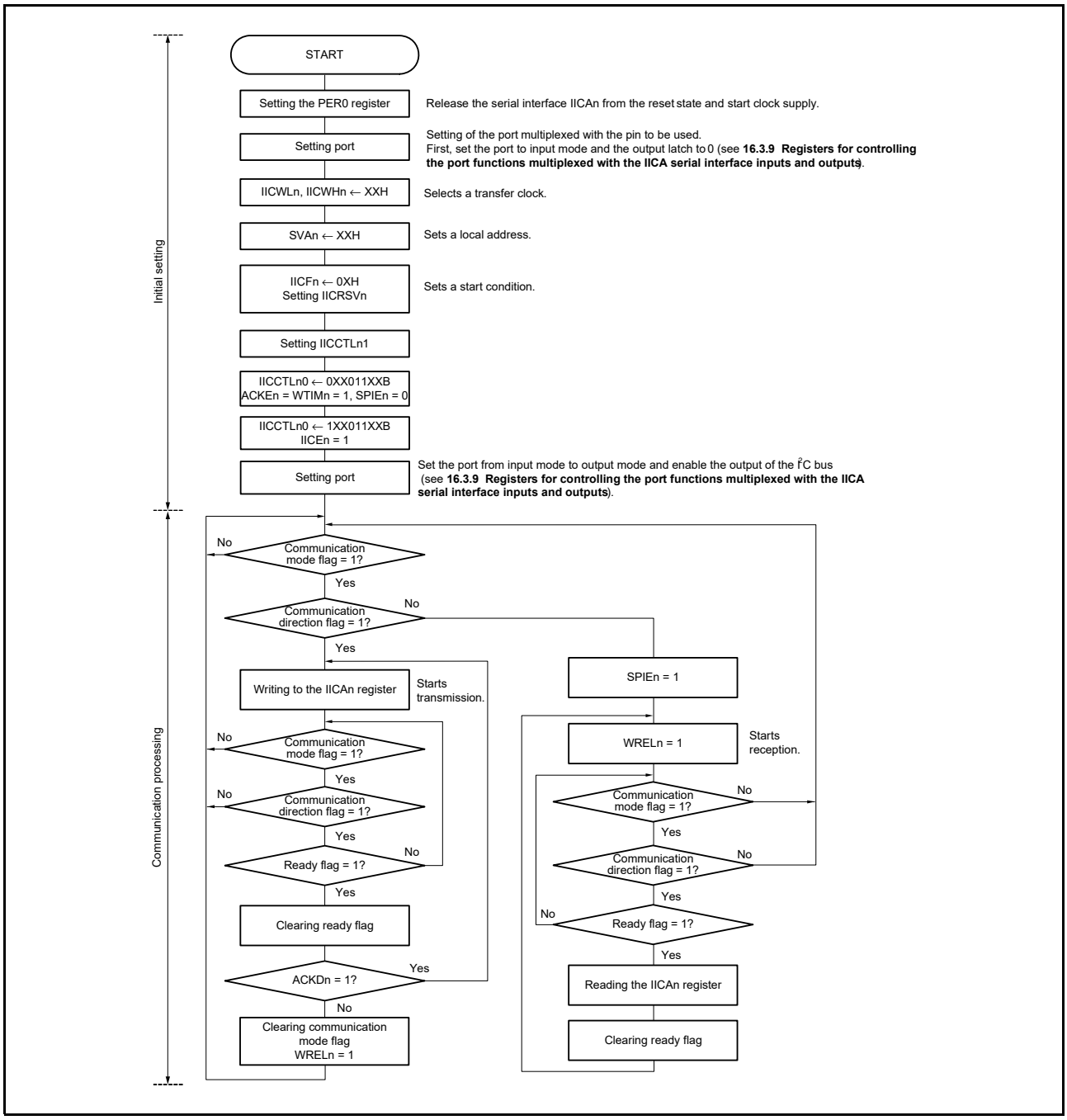
<3> Communication direction flag

This flag indicates the direction of communication. Its value is the same as the TRCn flag.

Remark n = 0, 1

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 state 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 communications state occurs in this way.

Figure 16 - 30 Slave Operation Flowchart (1)



Remark 1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

Remark 2. n = 0, 1

An example of the processing procedure of the slave with the INTIICAn interrupt is explained below (processing is performed assuming that the all-address match function is disabled and no extension code is used). The INTIICAn interrupt checks the state, and the following operations are performed.

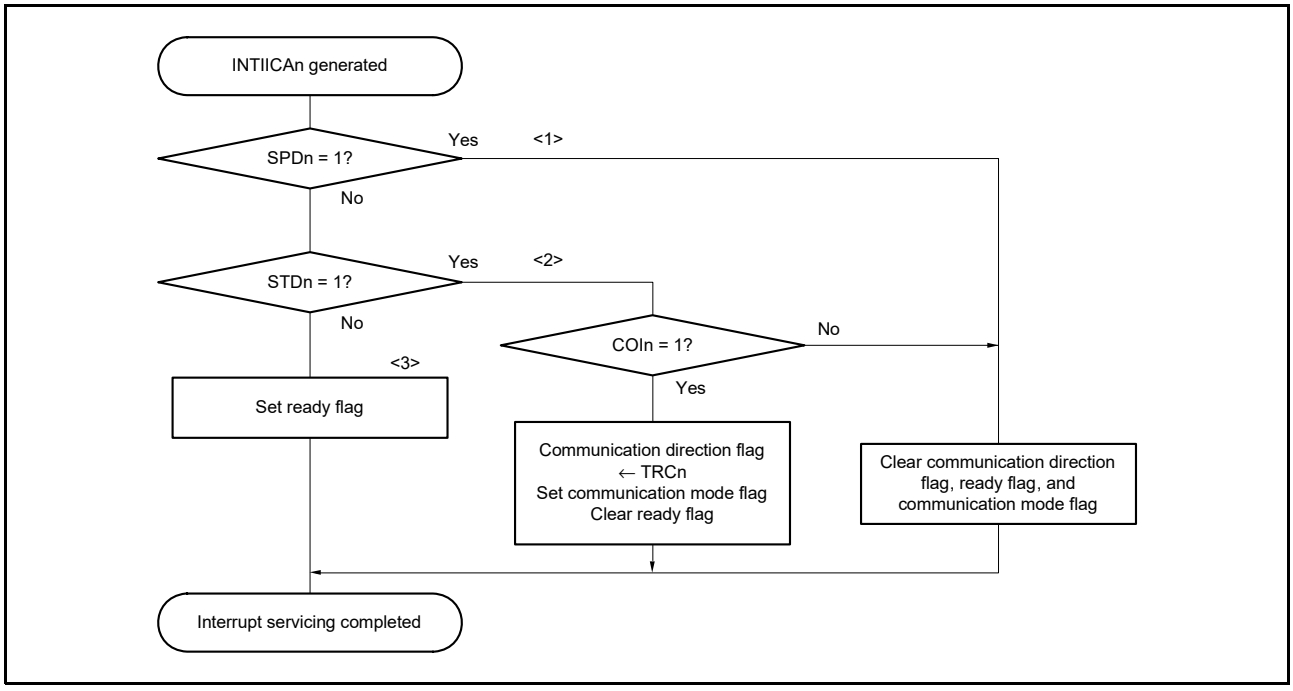
<1> Communication is stopped if the stop condition is issued.

<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 canceled, and processing returns from the interrupt (the ready flag is cleared).

<3> For data transmit/receive, only the ready flag is set. Processing returns from the interrupt with the I^2C bus remaining in the wait state.

Remark <1> to <3> above correspond to <1> to <3> in **Figure 16 - 31 Slave Operation Flowchart (2)**.

Figure 16 - 31 Slave Operation Flowchart (2)



Remark n = 0, 1

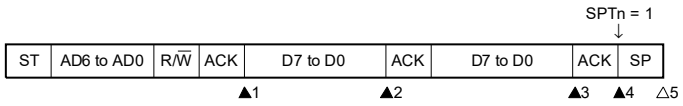
16.5.17 Timing of I2C interrupt request signal (INTIICAn) occurrence

The timing of transmitting or receiving data and generation of interrupt request signal INTIICAn, and the value of the IICA status register n (IICSn) when the INTIICAn signal is generated are shown below.

Remark 1. ST: Start condition
AD6 to AD0: Address
R/$\overline{W}$: Transfer direction specification
ACK: Acknowledge
D7 to D0: Data
SP: Stop condition

Remark 2. n = 0, 1

1. Master device operation
   a) Start ~ Address ~ Data ~ Data ~ Stop (transmission/reception)
      i) When WTIMn = 0

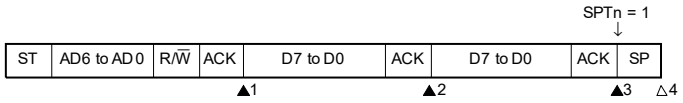


▲1: IICSn = 1000×110B
▲2: IICSn = 1000×000B
▲3: IICSn = 1000×000B (Sets the WTIMn bit to 1)**Note**
▲4: IICSn = 1000××00B (Sets the SPTn bit to 1)
△5: IICSn = 00000001B

Note To generate a stop condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

      ii) When WTIMn = 1



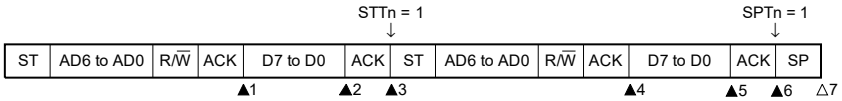
▲1: IICSn = 1000×110B
▲2: IICSn = 1000×100B
▲3: IICSn = 1000××00B (Sets the SPTn bit to 1)
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0, 1

b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

i) When WTIMn = 0



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×000B (Sets the WTIMn bit to 1)**Note 1**

▲3: IICSn = 1000××00B (Clears the WTIMn bit to 0**Note 2**, sets the STTn bit to 1)

▲4: IICSn = 1000×110B

▲5: IICSn = 1000×000B (Sets the WTIMn bit to 1)**Note 3**

▲6: IICSn = 1000××00B (Sets the SPTn bit to 1)

△7: IICSn = 00000001B

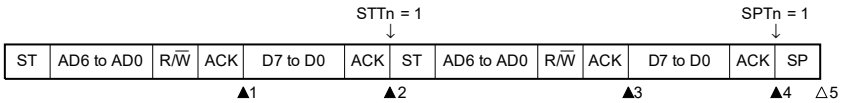
Note 1. To generate a start condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

Note 2. Clear the WTIMn bit to 0 to restore the original setting.

Note 3. To generate a stop condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

ii) When WTIMn = 1



▲1: IICSn = 1000×110B

▲2: IICSn = 1000××00B (Sets the STTn bit to 1)

▲3: IICSn = 1000×110B

▲4: IICSn = 1000××00B (Sets the SPTn bit to 1)

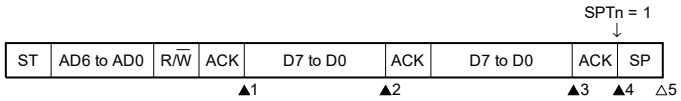
△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0, 1

c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

i) When WTIMn = 0



▲1: IICSn = 1010×110B

▲2: IICSn = 1010×000B

▲3: IICSn = 1010×000B (Sets the WTIMn bit to 1)**Note**

▲4: IICSn = 1010××00B (Sets the SPTn bit to 1)

△5: IICSn = 00000001B

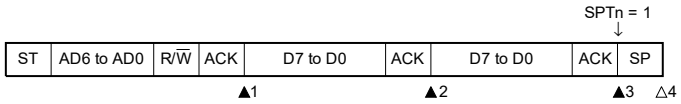
Note To generate a stop condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

ii) When WTIMn = 1



▲1: IICSn = 1010×110B

▲2: IICSn = 1010×100B

▲3: IICSn = 1010××00B (Sets the SPTn bit to 1)

△4: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

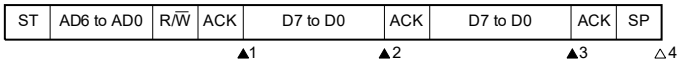
×: Don't care

Remark n = 0, 1

2. Slave device operation (slave address data reception)

a) Start ~ Address ~ Data ~ Data ~ Stop

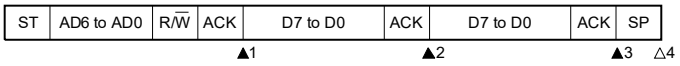
i) When WTIMn = 0



▲1: IICSn = 0001×110B
▲2: IICSn = 0001×000B
▲3: IICSn = 0001×000B
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

ii) When WTIMn = 1



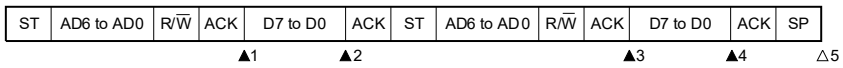
▲1: IICSn = 0001×110B
▲2: IICSn = 0001×100B
▲3: IICSn = 0001××00B
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0, 1

b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

i) When WTIMn = 0 (after restart, matches with SVAn, the all-address match function is disabled)



▲1: IICSn = 0001×110B
▲2: IICSn = 0001×000B
▲3: IICSn = 0001×110B
▲4: IICSn = 0001×000B
△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

ii) When WTIMn = 1 (after restart, matches with SVAn, the all-address match function is disabled)



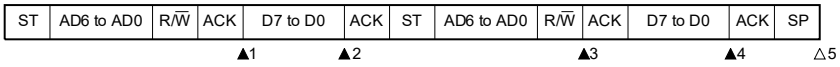
▲1: IICSn = 0001×110B
▲2: IICSn = 0001××00B
▲3: IICSn = 0001×110B
▲4: IICSn = 0001××00B
△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0, 1

c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

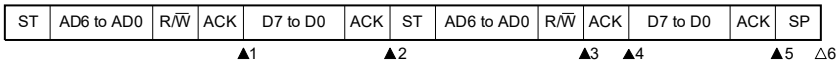
i) When WTIMn = 0 (after restart, does not match address (= extension code, the all-address match function is disabled))



▲1: IICSn = 0001×110B
▲2: IICSn = 0001×000B
▲3: IICSn = 0010×010B
▲4: IICSn = 0010×000B
△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

ii) When WTIMn = 1 (after restart, does not match address (= extension code, the all-address match function is disabled))



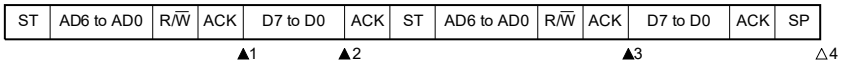
▲1: IICSn = 0001×110B
▲2: IICSn = 0001××00B
▲3: IICSn = 0010×010B
▲4: IICSn = 0010×110B
▲5: IICSn = 0010××00B
△6: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0, 1

d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

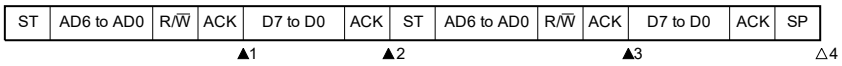
i) When WTIMn = 0 (after restart, does not match address (= not extension code, the all-address match function is disabled))



▲1: IICSn = 0001×110B
▲2: IICSn = 0001×000B
▲3: IICSn = 00000×10B
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

ii) When WTIMn = 1 (after restart, does not match address (= not extension code, the all-address match function is disabled))

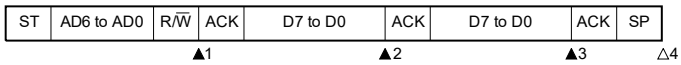


▲1: IICSn = 0001×110B
▲2: IICSn = 0001××00B
▲3: IICSn = 00000×10B
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0, 1

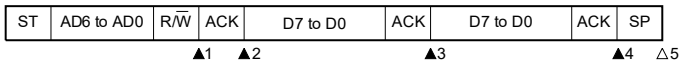
3. Slave device operation (when receiving extension code and the all-address match function is disabled)
   The device is always participating in communication when it receives an extension code.
   a) Start ~ Code ~ Data ~ Data ~ Stop
      i) When WTIMn = 0



▲1: IICSn = 0010×010B
▲2: IICSn = 0010×000B
▲3: IICSn = 0010×000B
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

      ii) When WTIMn = 1



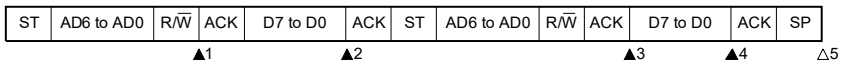
▲1: IICSn = 0010×010B
▲2: IICSn = 0010×110B
▲3: IICSn = 0010×100B
▲4: IICSn = 0010××00B
△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0, 1

b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

i) When WTIMn = 0 (after restart, matches with SVAn, the all-address match function is disabled)



▲1: IICSn = 0010×010B
▲2: IICSn = 0010×000B
▲3: IICSn = 0001×110B
▲4: IICSn = 0001×000B
△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

ii) When WTIMn = 1 (after restart, matches with SVAn, the all-address match function is disabled)



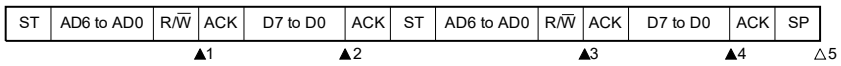
▲1: IICSn = 0010×010B
▲2: IICSn = 0010×110B
▲3: IICSn = 0010××00B
▲4: IICSn = 0001×110B
▲5: IICSn = 0001××00B
△6: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0, 1

c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

i) When WTIMn = 0 (after restart, extension code reception, the all-address match function is disabled)



▲1: IICSn = 0010×010B
▲2: IICSn = 0010×000B
▲3: IICSn = 0010×010B
▲4: IICSn = 0010×000B
△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

ii) When WTIMn = 1 (after restart, extension code reception, the all-address match function is disabled)



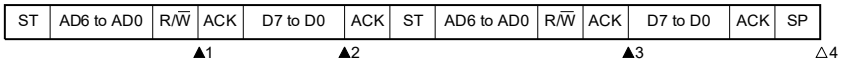
▲1: IICSn = 0010×010B
▲2: IICSn = 0010×110B
▲3: IICSn = 0010××00B
▲4: IICSn = 0010×010B
▲5: IICSn = 0010×110B
▲6: IICSn = 0010××00B
△7: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0, 1

d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

i) When WTIMn = 0 (after restart, does not match address (= not extension code, the all-address match function is disabled))



▲1: IICSn = 0010×010B
▲2: IICSn = 0010×000B
▲3: IICSn = 00000×10B
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

ii) When WTIMn = 1 (after restart, does not match address (= not extension code, the all-address match function is disabled))



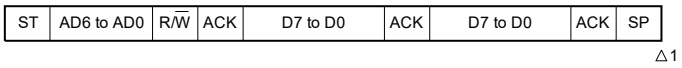
▲1: IICSn = 0010×010B
▲2: IICSn = 0010×110B
▲3: IICSn = 0010××00B
▲4: IICSn = 00000×10B
△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0, 1

4. Operation without communication

a) Start ~ Code ~ Data ~ Data ~ Stop



△1: IICSn = 00000001B

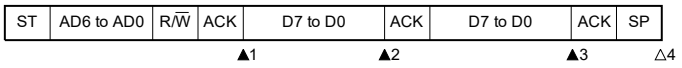
Remark △: Generated only when SPIEn = 1

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 MSTSn flag each time interrupt request signal INTIICAn has occurred to check the arbitration result.

a) When arbitration loss occurs during transmission of slave address data

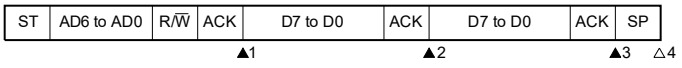
i) When WTIMn = 0



▲1: IICSn = 0101×110B
▲2: IICSn = 0001×000B
▲3: IICSn = 0001×000B
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

ii) When WTIMn = 1



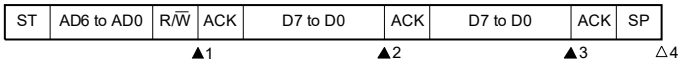
▲1: IICSn = 0101×110B
▲2: IICSn = 0001×100B
▲3: IICSn = 0001××00B
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0, 1

b) When arbitration loss occurs during transmission of extension code (the all-address match function is disabled)

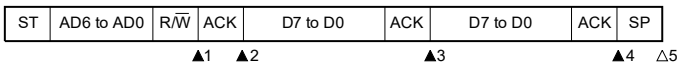
i) When WTIMn = 0



▲1: IICSn = 0110×010B
▲2: IICSn = 0010×000B
▲3: IICSn = 0010×000B
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

ii) When WTIMn = 1

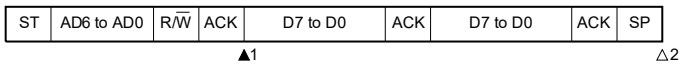


▲1: IICSn = 0110×010B
▲2: IICSn = 0010×110B
▲3: IICSn = 0010×100B
▲4: IICSn = 0010××00B
△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0, 1

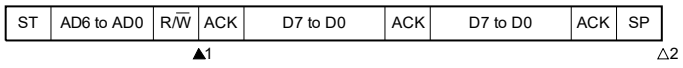
6. Operation when arbitration loss occurs (no communication after arbitration loss)
   When the device is used as a master in a multi-master system, read the MSTSn flag each time interrupt request signal INTIICAn has occurred to check the arbitration result.
   a) When arbitration loss occurs during transmission of slave address data (when WTIMn = 1)



▲1: IICSn = 01000110B
△2: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1

   b) When arbitration loss occurs during transmission of extension code (the all-address match function is disabled)



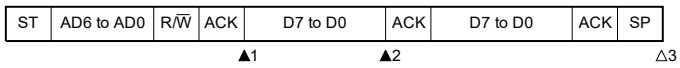
▲1: IICSn = 0110×010B
Sets LRELn = 1 by software
△2: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0, 1

c) When arbitration loss occurs during transmission of data

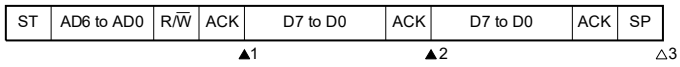
i) When WTIMn = 0



▲1: IICSn = 10001110B
▲2: IICSn = 01000000B
△3: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1

ii) When WTIMn = 1



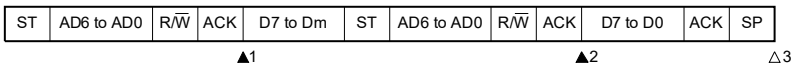
▲1: IICSn = 10001110B
▲2: IICSn = 01000100B
△3: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1

Remark n = 0, 1

d) When loss occurs due to restart condition during data transfer

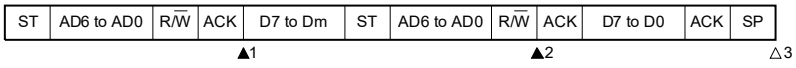
i) Not extension code (Example: unmatches with SVAn, the all-address match function is disabled)



▲1: IICSn = 1000×110B
▲2: IICSn = 01000110B
△3: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care
m = 6 to 0

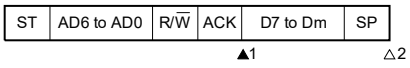
ii) Extension code (the all-address match function is disabled)



▲1: IICSn = 1000×110B
▲2: IICSn = 01100010B
Sets LRELn = 1 by software
△3: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care
m = 6 to 0

e) When loss occurs due to stop condition during data transfer



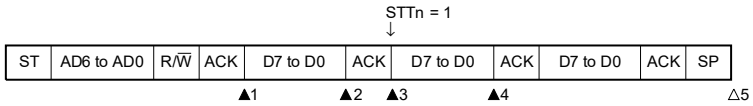
▲1: IICSn = 10000110B
△2: IICSn = 01000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care
m = 6 to 0

Remark n = 0, 1

f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

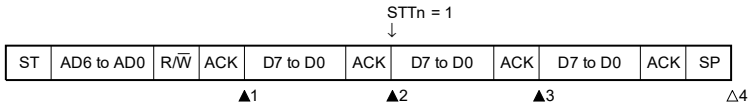
i) When WTIMn = 0



▲1: IICSn = 1000×110B
▲2: IICSn = 1000×000B (Sets the WTIMn bit to 1)
▲3: IICSn = 1000×100B (Clears the WTIMn bit to 0)
▲4: IICSn = 01000000B
△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

ii) When WTIMn = 1



▲1: IICSn = 1000×110B
▲2: IICSn = 1000×100B (Sets the STTn bit to 1)
▲3: IICSn = 01000100B
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0, 1

g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

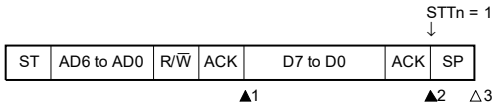
i) When WTIMn = 0



▲1: IICSn = 1000×110B
▲2: IICSn = 1000×000B (Sets the WTIMn bit to 1)
▲3: IICSn = 1000××00B (Sets the STTn bit to 1)
△4: IICSn = 01000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

ii) When WTIMn = 1



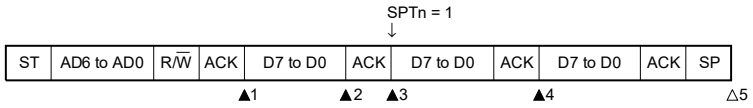
▲1: IICSn = 1000×110B
▲2: IICSn = 1000××00B (Sets the STTn bit to 1)
△3: IICSn = 01000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0, 1

h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

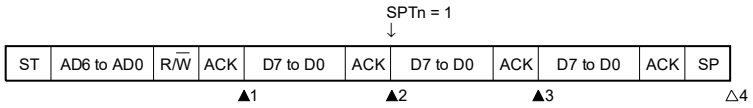
i) When WTIMn = 0



▲1: IICSn = 1000×110B
▲2: IICSn = 1000×000B (Sets the WTIMn bit to 1)
▲3: IICSn = 1000×100B (Clears the WTIMn bit to 0)
▲4: IICSn = 01000100B
△5: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

ii) When WTIMn = 1



▲1: IICSn = 1000×110B
▲2: IICSn = 1000×100B (Sets the SPTn bit to 1)
▲3: IICSn = 01000100B
△4: IICSn = 00000001B

Remark ▲: Always generated
△: Generated only when SPIEn = 1
×: Don't care

Remark n = 0, 1

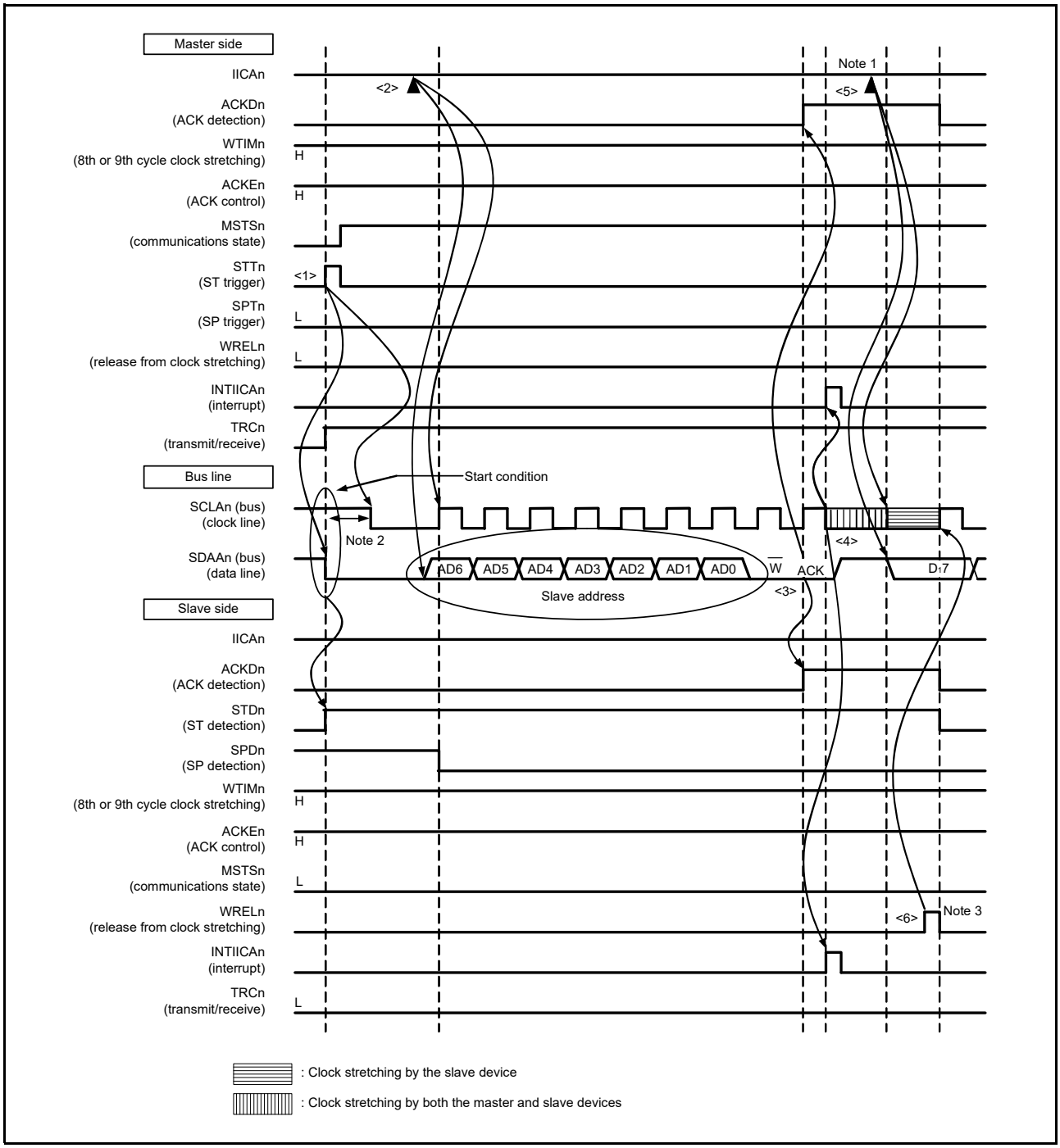
16.6 Timing Charts

In I^2C bus mode, the master selects one target slave device from among multiple slave devices by transmitting a slave address on the serial bus. After the slave address, the master transmits the R/W bit, which indicates the data transfer direction, and starts serial communication with the slave. This direction is reflected in the TRCn flag of IICA status register n (IICSn). **Figure 16 - 32** and **Figure 16 - 33** show timing of the data communication. The IICA shift register n (IICAn)'s shift operation is synchronized with the falling edge of the serial clock (SCLAn). The transmit data is transferred to the SO latch and is output (MSB first) via the SDAAn pin. Data input to the SDAAn pin is captured into IICAn at the rising edge of SCLAn. In the timing charts described in this section, it is assumed that the all-address match function is disabled.

Remark n = 0, 1

Figure 16 - 32 Example of Master-to-Slave Communications with Clock Stretching in the 9th-Cycle for Both the Master and Slave (1/4)

1. Start condition to address to data



Note 1. For release from the clock stretch state during transmission by a master device, write data to the IICAn register instead of setting the WRELn bit.

Note 2. Make sure that the time between the fall of the SDAAn pin signal and the fall of the SCLAn pin signal is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

Note 3. For release from the clock stretch state during reception by a slave device, write FFH to IICAn or set the WRELn bit.

Remark n = 0, 1

The meanings of <1> to <6> in **1. Start condition to address to data** in **Figure 16 - 32** are explained below.

<1> The start condition trigger is set by the master device (STTn = 1) and a start condition (i.e., SCLAn = 1 and SDAAn changes from 1 to 0) is generated once the bus data line goes low (SDAAn). When the start condition is subsequently detected, the master device enters the master device communications state (MSTSn = 1). The master device is ready to communicate once the bus clock line goes low (SCLAn = 0) after the hold time has elapsed.

<2> The master device writes the address + W (transmission) to the IICA shift register n (IICAn) and transmits the slave address.

<3> In the slave device if the address received matches the address (SVAn 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 (ACKDn = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device with the address matching the transmitted slave address sets the clock stretch state (SCLAn = 0) and issues an interrupt (INTIICAn: address match)**Note**.

<5> The master device writes the data to transmit to the IICAn register and releases the clock stretch state set by the master device.

<6> If the slave device releases the clock stretch state (WRELn = 1), the master device starts transferring data to the slave device.

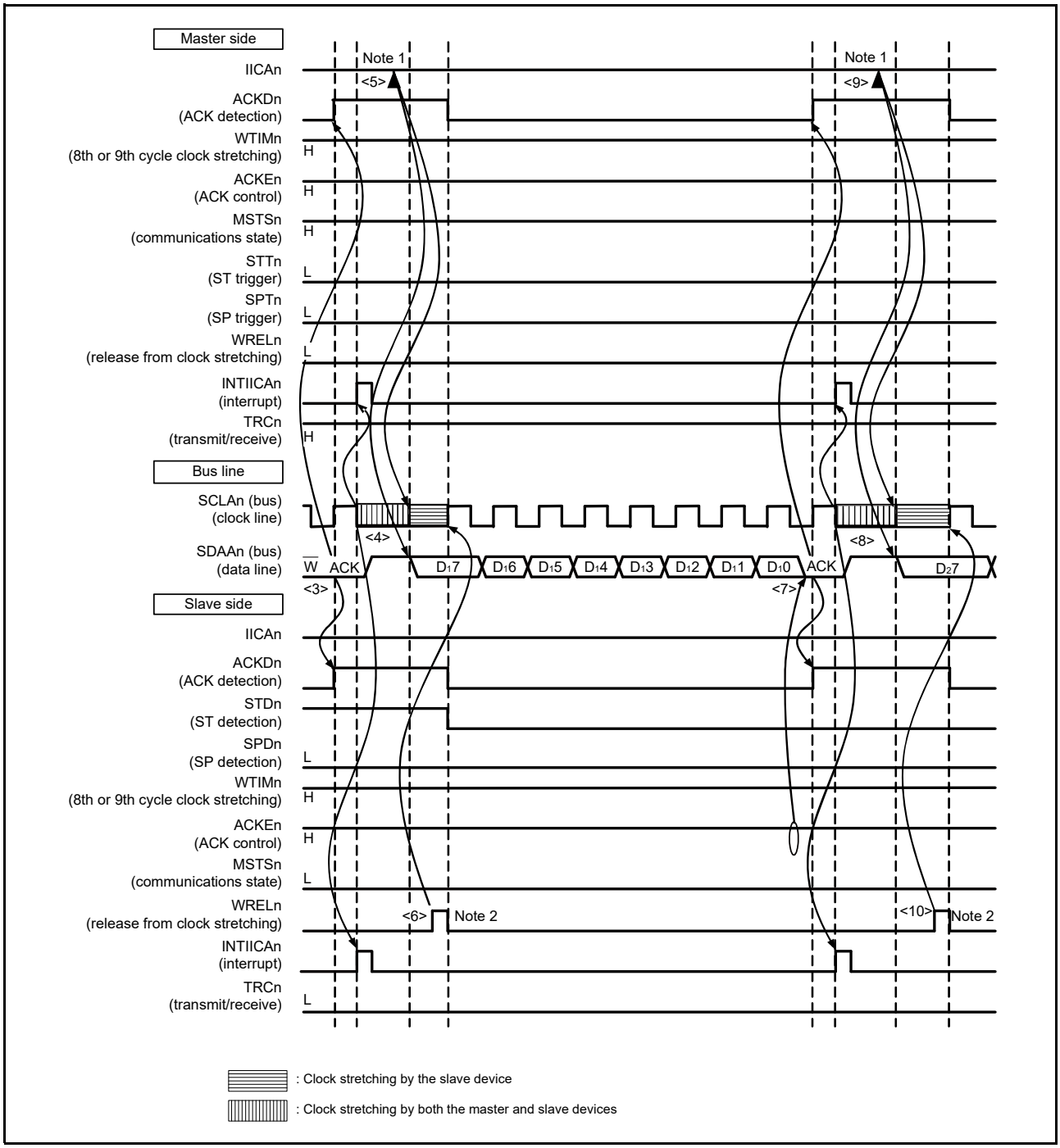
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set the clock stretch state. The master device issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <15> in **Figure 16 - 32** represent the entire procedure for communicating data using the I^2C bus.
1. Start condition to address to data in **Figure 16 - 32**: Steps <1> to <6>
2. Address to data to data in **Figure 16 - 32**: Steps <3> to <10>
3. Data to data to stop condition in **Figure 16 - 32**: Steps <7> to <15>

Remark 2. n = 0, 1

Figure 16 - 32 Example of Master-to-Slave Communications with Clock Stretching in the 9th-Cycle for Both the Master and Slave (2/4)

2. Address to data to data



Note 1. For release from the clock stretch state during transmission by a master device, write data to the IICAn register instead of setting the WRELn bit.

Note 2. For release from the clock stretch state during reception by a slave device, write FFH to IICAn or set the WRELn bit.

Remark n = 0, 1

The meanings of <3> to <10> in **2. Address to data to data** in **Figure 16 - 32** are explained below.

<3> In the slave device if the address received matches the address (SVAn 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 (ACKDn = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device with the address matching the transmitted slave address sets the clock stretch state (SCLAn = 0) and issues an interrupt (INTIICAn: address match)**Note**.

<5> The master device writes the data to transmit to the IICA shift register n (IICAn) and releases the clock stretch state set by the master device.

<6> If the slave device releases the clock stretch state (WRELn = 1), the master device starts transferring data to the slave device.

<7> After data transfer is completed, because of ACKEn = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).

<9> The master device writes the data to transmit to the IICAn register and releases the clock stretch state set by the master device.

<10> The slave device reads the received data and releases the clock stretch state (WRELn = 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: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set the clock stretch state. The master device issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <15> in **Figure 16 - 32** represent the entire procedure for communicating data using the I^2C bus.

1. Start condition to address to data in **Figure 16 - 32**: Steps <1> to <6>

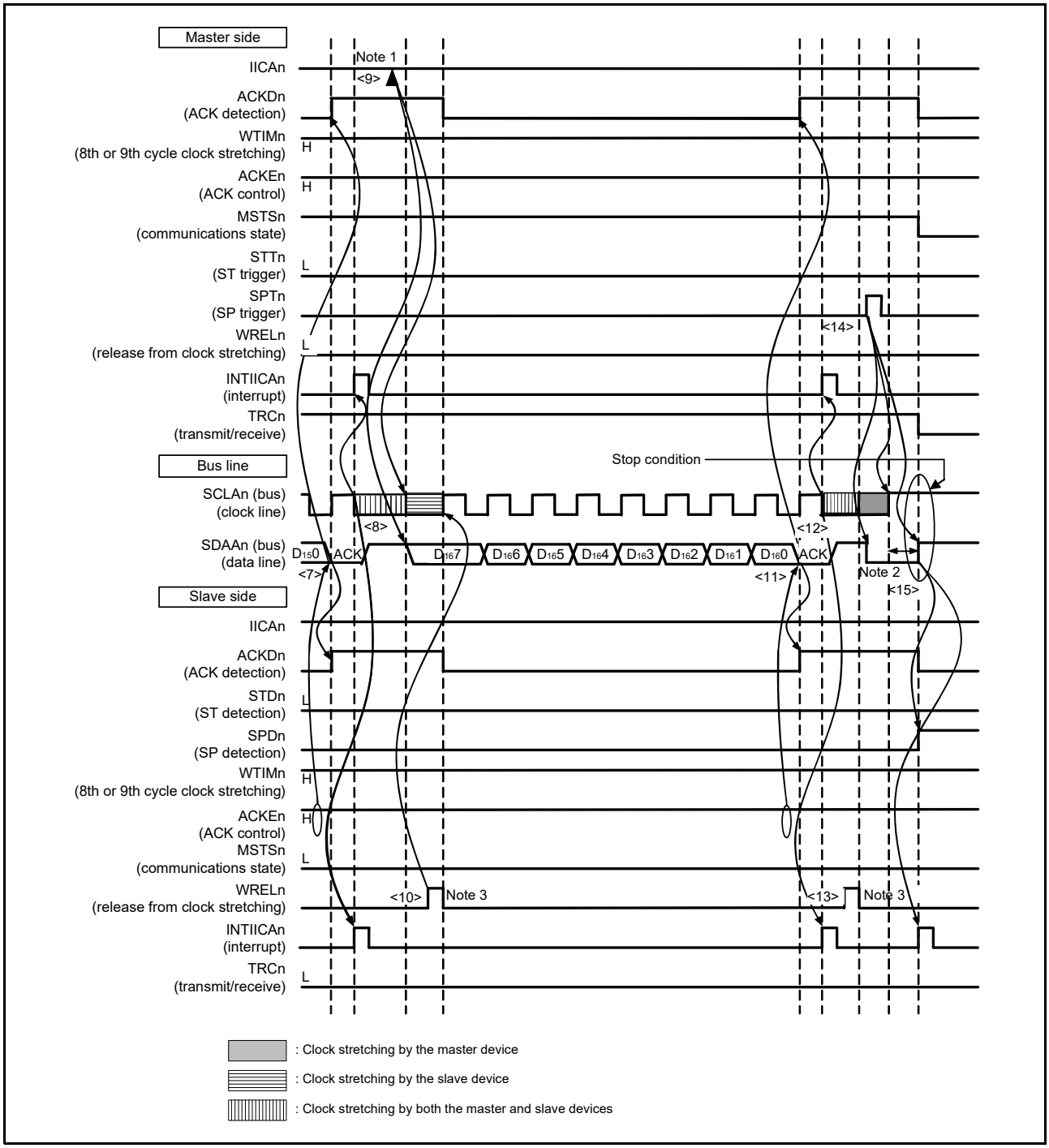
2. Address to data to data in **Figure 16 - 32**: Steps <3> to <10>

3. Data to data to stop condition in **Figure 16 - 32**: Steps <7> to <15>

Remark 2. n = 0, 1

Figure 16 - 32 Example of Master-to-Slave Communications with Clock Stretching in the 9th-Cycle for Both the Master and Slave (3/4)

3. Data to data to stop condition



Note 1. For release from the clock stretch state during transmission by a master device, write data to the IICAn register instead of setting the WRELn bit.

Note 2. Make sure that the time between the rise of the SCLAn 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.

Note 3. For release from the clock stretch state during reception by a slave device, write FFH to IICAn or set the WRELn bit.

Remark n = 0, 1

The meanings of <7> to <15> in **3. Data to data to stop condition** in **Figure 16 - 32** are explained below.

<7> After data transfer is completed, because of ACKEn = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).

<9> The master device writes the data to transmit to the IICA shift register n (IICAn) and releases the clock stretch state set by the master device.

<10> The slave device reads the received data and releases the clock stretch state (WRELn = 1). The master device then starts transferring data to the slave device.

<11> When data transfer is complete, the slave device (ACKEn =1) sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<12> The master device and slave device set the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).

<13> The slave device reads the received data and releases the clock stretch state (WRELn = 1).

<14> By the master device setting a stop condition trigger (SPTn = 1), the bus data line is cleared (SDAAn = 0) and the bus clock line is set (SCLAn = 1). After the stop condition setup time has elapsed, by setting the bus data line (SDAAn = 1), the stop condition is then generated (i.e., SCLAn = 1 and SDAAn changes from 0 to 1).

<15> When a stop condition is generated, the slave device detects the stop condition and issues an interrupt (INTIICAn: stop condition).

Remark 1. <1> to <15> in **Figure 16 - 32** represent the entire procedure for communicating data using the I^2C bus.

1. Start condition to address to data in **Figure 16 - 32**: Steps <1> to <6>

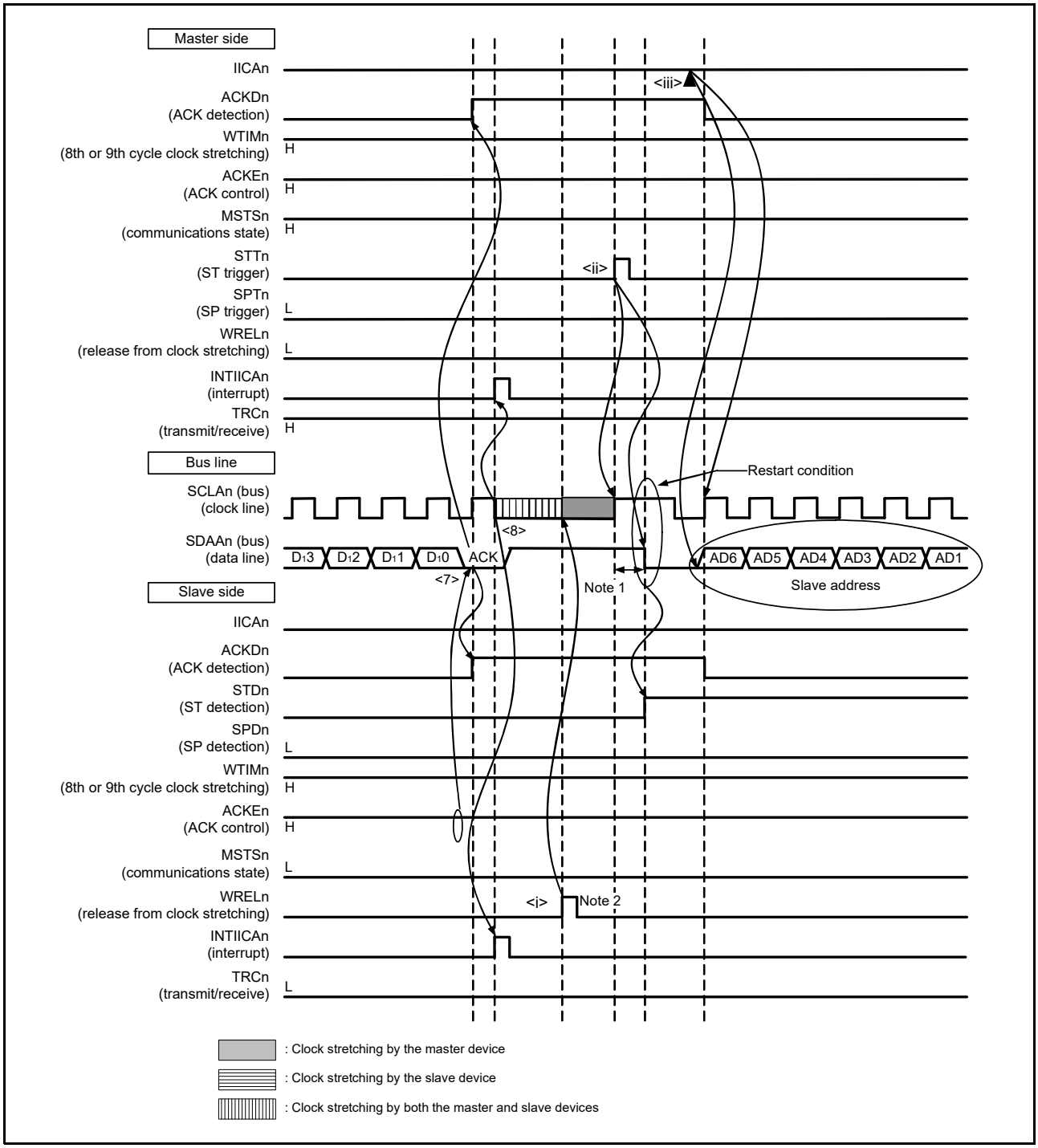
2. Address to data to data in **Figure 16 - 32**: Steps <3> to <10>

3. Data to data to stop condition in **Figure 16 - 32**: Steps <7> to <15>

Remark 2. n = 0, 1

Figure 16 - 32 Example of Master-to-Slave Communications with Clock Stretching in the 9th-Cycle for Both the Master and Slave (4/4)

4. Data to restart condition to address



Note 1. Make sure that the time between the rise of the SCLAn 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.

Note 2. For release from the clock stretch state during reception by a slave device, write FFH to IICAn or set the WRELn bit.

Remark n = 0, 1

The following describes the operations in **4. Data to restart condition to address** in **Figure 16 - 32**. After the operations in steps <7> and <8>, the operations in steps <i> to <iii> are performed. These steps return the processing to step <3>, the data transmission step.

<7> After data transfer is completed, because of ACKEn = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).

<i> The slave device reads the received data and releases the clock stretch state (WRELn = 1).

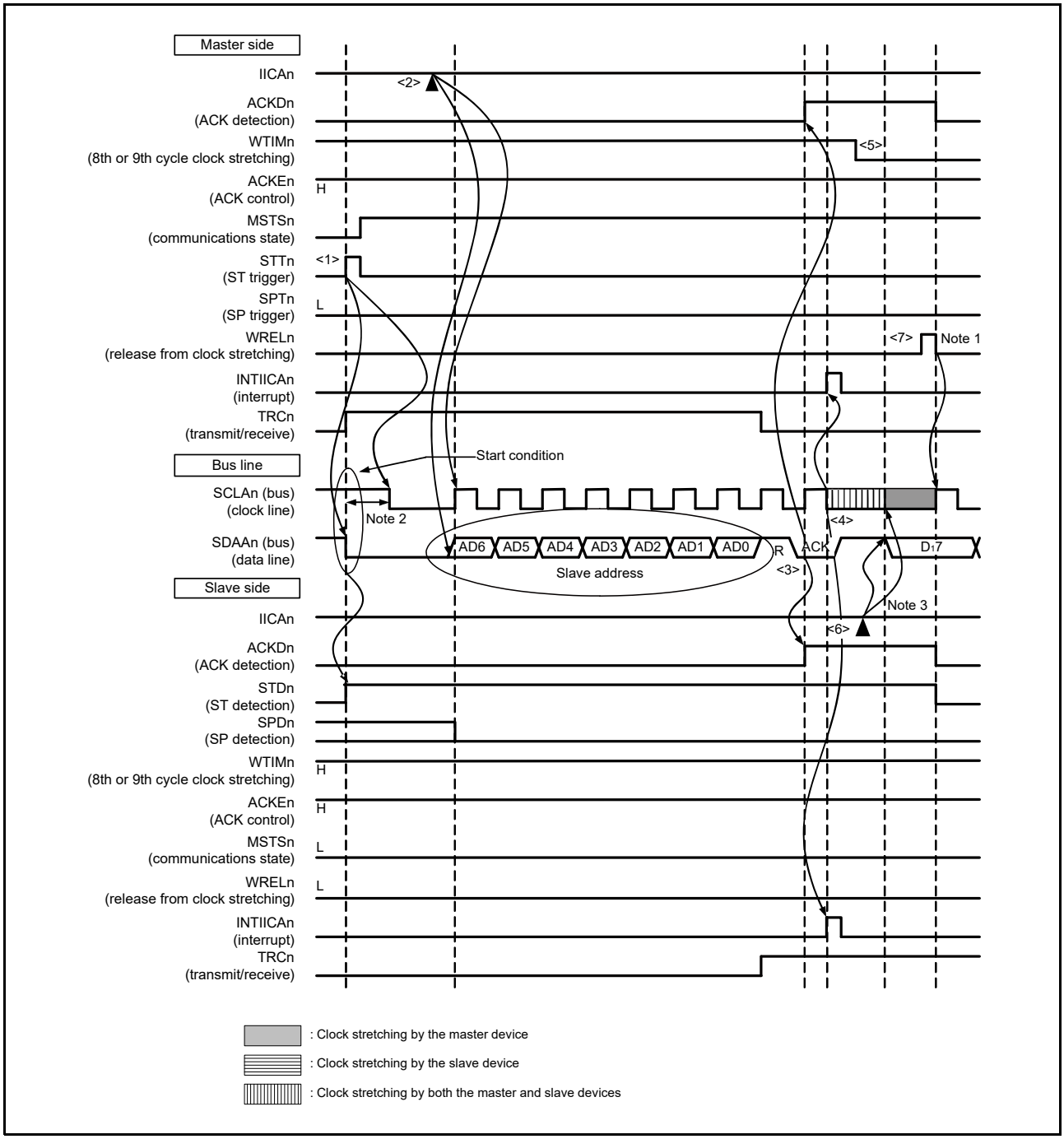
<ii> The start condition trigger is set again by the master device (STTn = 1) and a start condition (i.e., SCLAn = 1 and SDAAn changes from 1 to 0) is generated once the bus clock line goes high (SCLAn = 1) and the bus data line goes low (SDAAn = 0) after the restart condition setup time has elapsed. When the start condition is subsequently detected, the master device is ready to communicate once the bus clock line goes low (SCLAn = 0) after the hold time has elapsed.

<iii> The master device writing the address + R/W (transmission) to the IICA shift register (IICAn) enables the slave address to be transmitted.

Remark n = 0, 1

Figure 16 - 33 Example of Slave-to-Master Communications with Clock Stretching in the 8th-Cycle for the Master and 9th-Cycle for the Slave (1/3)

1. Start condition to address to data



Note 1. For release from the clock stretch state during reception by a master device, write FFH to IICAn or set the WRELn bit.

Note 2. Make sure that the time between the fall of the SDAAn pin signal and the fall of the SCLAn pin signal is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

Note 3. For release from the clock stretch state during transmission by a slave device, write data to the IICAn register instead of setting the WRELn bit.

Remark n = 0, 1

The meanings of <1> to <7> in **1. Start condition to address to data** in **Figure 16 - 33** are explained below.

<1> The start condition trigger is set by the master device (STTn = 1) and a start condition (i.e., SCLAn = 1 and SDAAn changes from 1 to 0) is generated once the bus data line goes low (SDAAn). When the start condition is subsequently detected, the master device enters the master device communications state (MSTSn = 1). The master device is ready to communicate once the bus clock line goes low (SCLAn = 0) after the hold time has elapsed.

<2> The master device writes the address + R (reception) to the IICA shift register n (IICAn) and transmits the slave address.

<3> In the slave device if the address received matches the address (SVAn 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 (ACKDn = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device with the address matching the transmitted slave address sets the clock stretch state (SCLAn = 0) and issues an interrupt (INTIICAn: address match)**Note**.

<5> The timing at which the master device sets the clock stretch state changes to the 8th clock (WTIMn = 0).

<6> The slave device writes the data to transmit to the IICAn register and releases the clock stretch state set by the slave device.

<7> The master device releases the clock stretch state (WRELn = 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: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set the clock stretch state. The master device issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

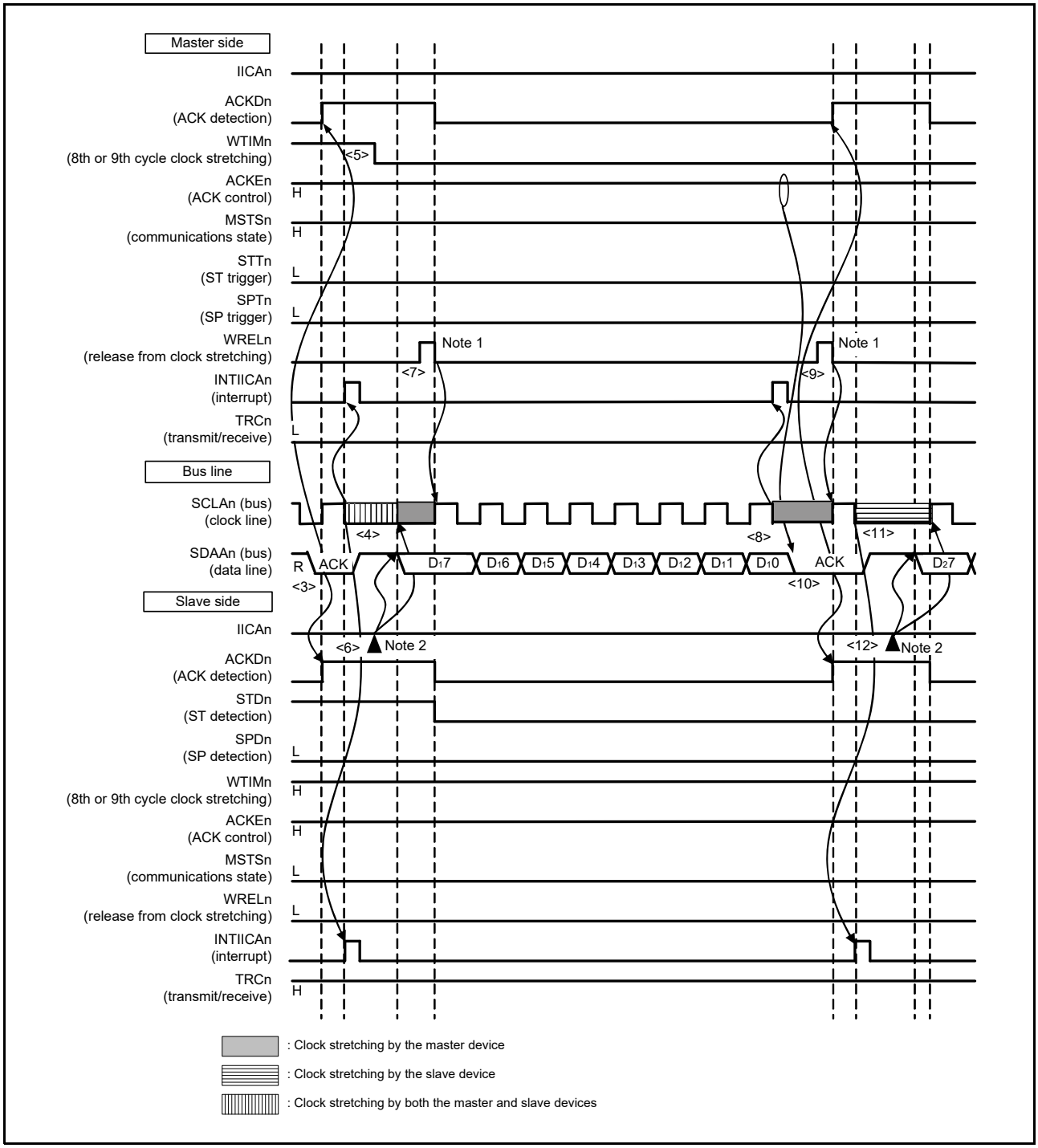
Remark 1. <1> to <19> in **Figure 16 - 33** represent the entire procedure for communicating data using the I^2C bus.
1. Start condition to address to data in **Figure 16 - 33**: Steps <1> to <7>
2. Address to data to data in **Figure 16 - 33**: Steps <3> to <12>
3. Data to data to stop condition in **Figure 16 - 33**: Steps <8> to <19>

Remark 2. n = 0, 1

Figure 16 - 33 Example of Slave-to-Master Communications with Clock Stretching in the 8th-Cycle for the Master and 9th-Cycle for the Slave (2/3)

2. Address to data to data

<R>



Note 1. For release from the clock stretch state during reception by a master device, write FFH to IICAn or set the WRELn bit.

Note 2. For release from the clock stretch state during transmission by a slave device, write data to the IICAn register instead of setting the WRELn bit.

Remark n = 0, 1

The meanings of <3> to <12> in **2. Address to data to data** in **Figure 16 - 33** are explained below.

<3> In the slave device if the address received matches the address (SVAn 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 (ACKDn = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device with the address matching the transmitted slave address sets the clock stretch state (SCLAn = 0) and issues an interrupt (INTIICAn: address match)**Note**.

<5> The master device changes the timing of clock stretching to the 8th clock (WTIMn = 0).

<6> The slave device writes the data to transmit to the IICA shift register n (IICAn) and releases the clock stretch state set by the slave device.

<7> The master device releases the clock stretch state (WRELn = 1) and starts transferring data from the slave device to the master device.

<8> The master device sets the clock stretch state (SCLAn = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICAn: end of transfer). Because of ACKEn = 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 clock stretch state (WRELn = 1).

<10> The ACK is detected by the slave device (ACKDn = 1) at the rising edge of the 9th clock.

<11> The slave device sets the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICAn: end of transfer).

<12> By the slave device writing the data to transmit to the IICAn register, the clock stretch state 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: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set the clock stretch state. The master device issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <19> in **Figure 16 - 33** represent the entire procedure for communicating data using the I^2C bus.

1. Start condition to address to data in **Figure 16 - 33**: Steps <1> to <7>

2. Address to data to data in **Figure 16 - 33**: Steps <3> to <12>

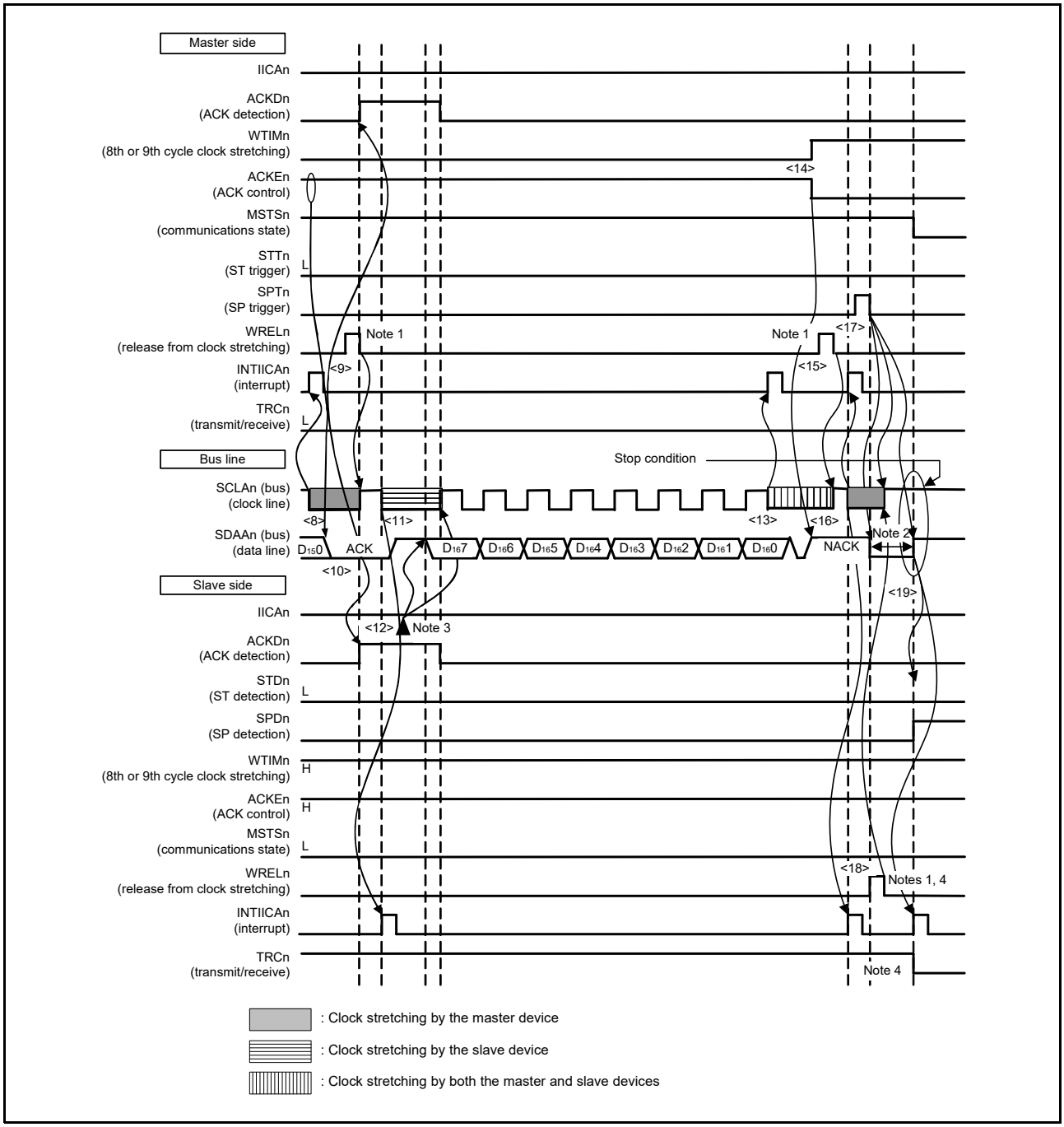
3. Data to data to stop condition in **Figure 16 - 33**: Steps <8> to <19>

Remark 2. n = 0, 1

Figure 16 - 33 Example of Slave-to-Master Communications with Shifted Clock Stretching (8th- to 9th-Cycle) for the Master and 9th-Cycle for the Slave (3/3)

3. Data to data to stop condition

<R>



Note 1. For release from the clock stretch state, write FFH to IICAn or set the WRELn bit.

Note 2. Make sure that the time between the rise of the SCLAn 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.

Note 3. For release from the clock stretch state during transmission by a slave device, write data to the IICAn register instead of setting the WRELn bit.

Note 4. If the clock stretch state during transmission by a slave device is released by setting the WRELn bit, the TRCn flag will be cleared.

Remark n = 0, 1

The meanings of <8> to <19> in **3. Data to data to stop condition** in **Figure 16 - 33** are explained below.

<8> The master device sets the clock stretch state (SCLAn = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICAn: end of transfer). Because of ACKEn = 0 in the master device, the master device then sends an ACK by hardware to the slave device.

<9> The master device reads the received data and releases the clock stretch state (WRELn = 1).

<10> The ACK is detected by the slave device (ACKDn = 1) at the rising edge of the 9th clock.

<11> The slave device sets the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICAn: end of transfer).

<12> By the slave device writing the data to transmit to the IICA register, the clock stretch state 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 (INTIICAn: end of transfer) at the falling edge of the 8th clock, and sets the clock stretch state (SCLAn = 0). Because ACK control (ACKEn = 1) is performed, the bus data line is at the low level (SDAAn = 0) at this stage.

<14> The master device sets NACK as the response (ACKEn = 0) and changes the timing at which it sets the clock stretch state to the 9th clock (WTIMn = 1).

<15> If the master device releases the clock stretch state (WRELn = 1), the slave device detects the NACK (ACK = 0) at the rising edge of the 9th clock.

<16> The master device and slave device set the clock stretch state (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).

<17> When the master device issues a stop condition (SPTn = 1), the bus data line is cleared (SDAAn = 0) and the master device releases the clock stretch state. The master device then waits until the bus clock line is set (SCLAn = 1).

<18> The slave device acknowledges the NACK, halts transmission, and releases the clock stretch state (WRELn = 1) to end communication. Once the slave device releases the clock stretch state, the bus clock line is set (SCLAn = 1).

<19> Once the master device recognizes that the bus clock line is set (SCLAn = 1) and after the stop condition setup time has elapsed, the master device sets the bus data line (SDAAn = 1) and issues a stop condition (i.e., SCLAn = 1 and SDAAn changes from 0 to 1). The slave device detects the generated stop condition and slave device issue an interrupt (INTIICAn: stop condition).

Remark 1. <1> to <19> in **Figure 16 - 33** represent the entire procedure for communicating data using the I^2C bus.

1. Start condition to address to data in **Figure 16 - 33**: Steps <1> to <7>

2. Address to data to data in **Figure 16 - 33**: Steps <3> to <12>

3. Data to data to stop condition in **Figure 16 - 33**: Steps <8> to <19>

Remark 2. n = 0, 1

Section 17 Serial Interface UARTA (UARTA)

The number of channels of the serial interface UARTA depends on the product.

	R7F100G			R7F104G
	30- and 32-pin Products	36- and 40-pin Products	44- to 123-pin Products	32- to 48-pin Products
Number of channels	—	1	2	1

17.1 Overview

The serial interface UARTAn supports the following two modes.

1. Operation stop mode
   This mode is used when serial communication is not executed; power consumption can be reduced.
2. UART mode
   This is a UART mode that supports continuous transmission. The list below outlines the features:

The serial interface UARTAn performs an asynchronous communication. It has the following functions.

- Maximum transfer rate: 153.6 kbps
- Transmission and reception using two pins
  TxDAn: Transmit data output pin
  RxDAn: Receive data input pin
- Character length of transfer data selectable from 5, 7, and 8 bits
- Baud rate arbitrarily settable with the dedicated internal 8-bit baud rate generator
- Transmission and reception independent of each other (full-duplex communication)
- MSB or LSB first transfer selectable
- Inversion control of communication logic level provided
- Transmit clock output is available (CLKAn)
- Operating clock independent of the CPU/peripheral hardware clock (f_{CLK}) selectable

Remark n: Unit number (n = 0, 1)

Figure 17 - 1 shows a block diagram of the serial interface UARTAn and **Table 17 - 1** shows the pin configuration of the serial interface UARTAn.

Figure 17 - 1 Block Diagram of Serial Interface UARTAn

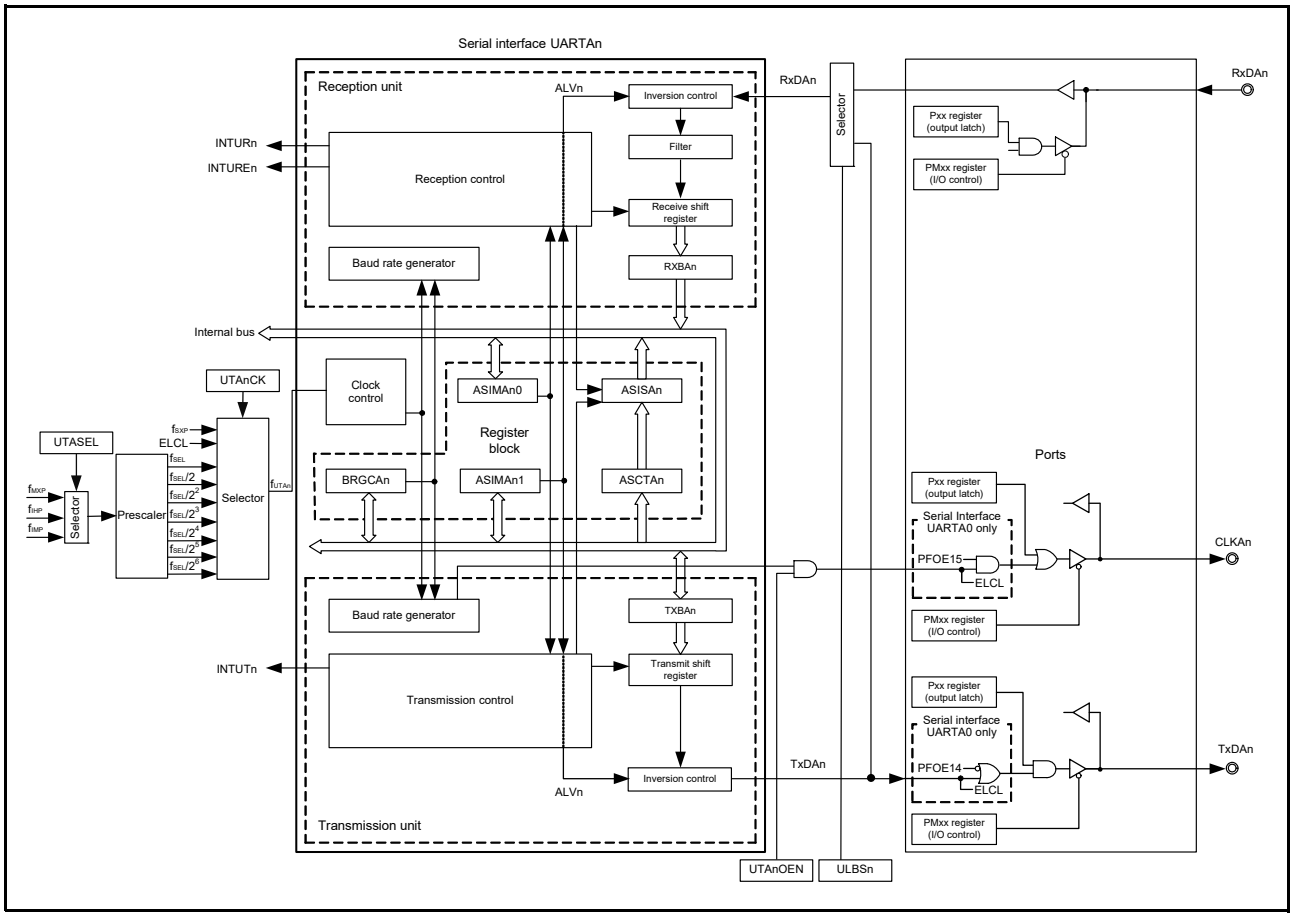


Table 17 - 1 Pin Configuration of Serial Interface UARTAn

Name	I/O	Function
RxDAn	Input	Serial data input signal
TxDAn	Output	Serial data output signal
CLKAn	Output	Serial clock output signal

Remark 1. f_{MXP}: High-speed peripheral clock frequency
f_{IHP}: High-speed on-chip oscillator peripheral clock frequency
f_{IMP}: Middle-speed on-chip oscillator peripheral clock frequency
f_{SXP}: Low-speed peripheral clock frequency
f_{SEL}: Selected clock to be divided for the UARTA
f_{UTAn}: Serial interface UARTAn operation clock
ELCL: Event input from the ELCL

Remark 2. n: Unit number (n = 0, 1)

17.2 Registers for Controlling the Serial Interface UARTA

The following registers are used to control the serial interface UARTA.

- Peripheral enable register 1 (PER1)
- Transmit buffer register n (TXBAn)
- Receive buffer register n (RXBAn)
- Operation mode setting register n0 (ASIMAn0)
- Operation mode setting register n1 (ASIMAn1)
- Baud rate generator control register n (BRGCAn)
- Status register n (ASISAn)
- Status clear trigger register n (ASCTAn)
- UARTA clock select register 0 (UTA0CK)
- UARTA clock select register 1 (UTA1CK)
- UART loopback select register (ULBS)
- Port mode registers (PMxx)
- Port registers (Pxx)
- Port input mode registers (PIMxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port function output enable registers (PFOEx)

17.2.1 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. To use the serial interface UARTAn, be sure to set the UTAEN bit to 1. The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 17 - 2 Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER1	DACEN	SMSEN	CMPEN	TML32EN	DTCEN	UTAEN **Note 1**	REMCEN **Note 2**	CTSUEN

UTAEN	Control of supply of an input clock to the serial interface UARTAn
0	Stops supply of an input clock. • The SFRs used by the serial interface UARTAn cannot be written.**Note 3**
1	Enables supply of an input clock. • The SFRs used by the serial interface UARTAn can be read and written.

Note 1. This bit is only present in the 36- to 128-pin R7F100G and 32- to 48-pin R7F104G products.
Note 2. This bit is only present in the 32- to 128-pin R7F100G and 32- to 48-pin R7F104G products.
Note 3. When an SFR is read, 00H is returned regardless of its setting.

Caution 1. Be sure to leave the following bits set to 0.
Bits 2 and 1 in the 30-pin R7F100G products
Bit 2 in the 32-pin R7F100G products
Caution 2. Make sure that the setting of the UTAEN bit is 1 before using the serial interface UARTAn. If UTAEN = 0, writing to the registers which control the serial interface UARTAn is ignored.

Remark n: Unit number (n = 0, 1)

17.2.2 Transmit buffer register n (TXBAn)

TXBAn is a buffer register for setting transmit data. Transmission starts by writing data for transmission to the TXBAn register. The TXBAn register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is FFH.

Figure 17 - 3 Format of Transmit Buffer Register n (TXBAn)

Address: F0300H (TXBA0), F0308H (TXBA1)
After reset: FFH
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TXBAn								

Bits 7 to 0	Function
—	• When a character length of 8 bits is specified: Data in bits 7 to 0 of TXBAn are transferred. • When a character length of 7 bits is specified: Data in bits 6 to 0 of TXBAn are transferred in either MSB- or LSB-first mode; bit 7 is invalid. • When a character length of 5 bits is specified: Data in bits 4 to 0 of TXBAn are transferred in either MSB- or LSB-first mode; bits 7 to 5 are invalid.

Caution 1. When the TXBFAn flag of the ASISAn register is 1, do not write data for transmission to the TXBAn register.

Caution 2. After setting the TXEAn bit of the ASIMAn0 register to 1, wait for the period of at least one cycle of the serial interface UARTAn operation clock (f_{UTAn}) before setting the first data for transmission in the TXBAn register. If data for transmission is set within one cycle of the serial interface UARTAn operation clock after the TXEAn bit is set to 1, the start of transmission is delayed by one cycle of the serial interface UARTAn operation clock.

Remark 1. Data is transferred from the TXBAn register to this register, and is then transmitted as serial data through the TxDAn pin. In the first transmission, data is transferred from the TXBAn register to this register immediately after data is written to the TXBAn register. In continuous transmission, data is transferred after transmission of one frame and just before generation of the transfer completion interrupt. The transmit shift register cannot be manipulated directly by a program.

Remark 2. n: Unit number (n = 0, 1)

17.2.3 Receive buffer register n (RXBAn)

The RXBAn register holds the parallel data converted by the receive shift register. Every time one byte of data is received, the next receive data is transferred from the receive shift register**Note** to this register. The RXBAn register can be read by an 8-bit memory manipulation instruction. The value of this register following a reset is FFH.

Note The receive shift register converts the serial data that is input through the RxDAn pin to parallel data. The receive shift register cannot be manipulated directly by a program.

Figure 17 - 4 Format of Receive Buffer Register n (RXBAn)

Address: F0301H (RXBA0), F0309H (RXBA1)
After reset: FFH
R/W: R

Symbol	7	6	5	4	3	2	1	0
RXBAn								

Bits 7 to 0	Function
—	• When a character length of 8 bits is specified: Receive data is transferred to bits 7 to 0 of this register. • When a character length of 7 bits is specified: Receive data is transferred to bits 6 to 0 of this register in either MSB- or LSB-first mode; bit 7 is always 0. • When a character length of 5 bits is specified: Receive data is transferred to bits 4 to 0 of this register in either MSB- or LSB-first mode; bits 7 to 5 are always 0.

Caution **If an overrun error (OVEAn) occurs, the data received at that time are not stored in the RXBAn register.**

Remark n: Unit number (n = 0, 1)

17.2.4 Operation mode setting register n0 (ASIMAn0)

The ASIMAn0 register is an 8-bit register that controls serial communication of the serial interface UARTAn. The ASIMAn0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 01H.

Figure 17 - 5 Format of Operation Mode Setting Register n0 (ASIMAn0)

Address: F0302H (ASIMA00), F030AH (ASIMA10)
After reset: 01H
R/W: R/W

Symbol	<7>	<6>	<5>	4	3	2	<1>	<0>
ASIMAn0	UARTAENn	TXEAn	RXEAn	0	0	0	ISSMAn	ISRMAn

UARTAENn Note 1	UART operation enable
0	Disables the UART operation clock. Resets the internal circuits.Note 2
1	Enables the UART operation clock.

TXEAn	Transmission enable
0	Disables transmission. (Resets the transmission circuit.)
1	Enables transmission.

RXEAn	Reception enable
0	Disables reception. (Resets the reception circuit.)
1	Enables reception.

ISSMAn	Transmit interrupt mode select
0	The INTUTn interrupt is generated on completion of transmission.
1	The INTUTn interrupt is generated when the transmit buffer becomes empty. (for continuous transmission)

ISRMAn	Receive interrupt mode select
0	The INTUREn interrupt is generated when a reception error occurs. (INTURn is not generated.)
1	The INTURn interrupt is generated when a reception error occurs. (INTUREn is not generated.)

Note 1. When UARTAENn = 0, the level being output from the TxDAn pin and the level being input from the RxDAn pin are determined according to the setting of the ALVn bit as described below.
- When ALVn = 0, output from the TxDAn pin is high.
- When ALVn = 1, output from the TxDAn pin is low.

Note 2. The ASISAn and RXBAn registers are reset by clearing the UARTAENn bit to 0.

Caution 1. Be sure to leave bits 4 to 2 set to 0.

Caution 2. To start transmission, set the UARTAENn bit to 1 and then set the TXEAn bit to 1. To stop transmission, clear the TXEAn bit to 0 and then clear the UARTAENn bit to 0.

Caution 3. To start reception, set the UARTAENn bit to 1 and then set the RXEAn bit to 1. To stop reception, clear the RXEAn bit to 0 and then clear the UARTAENn bit to 0.

(Cautions and Remark are listed on the next page.)

Caution 4. Follow the procedure below when setting the UARTAENn bit to 1 and then setting the RXEAn bit to 1.
- **When ALVn = 0, the setting must be made while the level being input to the RxDAn pin is high. Otherwise, reception will start at that point and a framing error may occur.**
- **When ALVn = 1, the setting must be made while the level being input to the RxDAn pin is low. Otherwise, reception will start at that point and a framing error may occur.**

Caution 5. The TXEAn and RXEAn bits are synchronized with the serial interface UARTAn operation clock (f_{UTAn}). To enable transmission or reception again, set the TXEAn or RXEAn bit to 1 at least two cycles of the serial interface UARTAn operation clock after clearing the TXEAn or RXEAn bit to 0. If the bit is set to 1 within two cycles of the serial interface UARTAn operation clock after the clearing, the transmission or reception circuit may not be able to be initialized.

Caution 6. After setting TXEAn bit to 1, wait for at least one cycle of the serial interface UARTAn operation clock (f_{UTAn}) before setting the transmit data in the TXBAn register.

Caution 7. Clear the RXEAn bit to 0 before modifying the ISRMAn bit.

Remark n: Unit number (n = 0, 1)

17.2.5 Operation mode setting register n1 (ASIMAn1)

The ASIMAn1 register is an 8-bit register that controls serial communication of the serial interface UARTAn. The ASIMAn1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The ASIMAn1 register must be modified while TXEAn = 0 and RXEAn = 0. The value of this register following a reset is 1AH.

Figure 17 - 6 Format of Operation Mode Setting Register n1 (ASIMAn1)

Address: F0303H (ASIMA01), F030BH (ASIMA11)
After reset: 1AH
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ASIMAn1	0	PSn1	PSn0	CLn1	CLn0	SLn	DIRn	ALVn

PSn1	PSn0	Transmission/reception parity bit setting 1, 0	
		Transmission	Reception
0	0	No parity bit is output.	Data is received without parity.
0	1	0 parity is output.	Data is received with 0 parity.Note
1	0	Odd parity is output.	Check is made for odd parity.
1	1	Even parity is output.	Check is made for even parity.

CLn1	CLn0	Transmission/reception character length setting 1, 0
0	0	Character length of data = 5 bits
0	1	Setting prohibited
1	0	Character length of data = 7 bits
1	1	Character length of data = 8 bits

SLn	Transmission stop bit length setting
0	Stop bit length = 1 bit
1	Stop bit length = 2 bits

DIRn	Transmission/reception order setting
0	MSB first
1	LSB first

ALVn	Transmission/reception level setting
0	Positive logic (wait state = high level, start bit = low level, stop bit = high level)
1	Negative logic (wait state = low level, start bit = high level, stop bit = low level)

Note When “Data is received with 0 parity” is set, parity check is not performed. Accordingly the PEAn bit of the ASISAn register is not set: no reception error interrupts are generated.

Caution 1. Clear both the TXEAn and RXEAn bits to 0 before modifying the ASIMAn1 register.

Caution 2. Reception is always handled as including a stop bit. The setting of the SLn bit does not affect reception.

Remark n: Unit number (n = 0, 1)

17.2.6 Baud rate generator control register n (BRGCAn)

The BRGCAn register sets the frequency divisor for the 8-bit counter in the serial interface UARTAn. The BRGCAn register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is FFH.

Figure 17 - 7 Format of Baud Rate Generator Control Register n (BRGCAn)

Address: F0304H (BRGCA0), F030CH (BRGCA1)
After reset: FFH
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
BRGCAn								

Bits 7 to 0	Function
—	Controls the UART baud rate (serial transfer speed). For details on the settings, see **Table 17 - 2**.

Caution Modify the BRGCAn bits while the TXEAn and RXEAn bits are 0 (in the transmission/reception stopped state).

Table 17 - 2 BRGCAn Settings

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	k	Selection of 8-bit counter output clock
0	0	0	0	0	0	0	×	×	Setting prohibited
0	0	0	0	0	0	1	0	2	$f_{UTAn}/2$
0	0	0	0	0	0	1	1	3	$f_{UTAn}/3$
:	:	:	:	:	:	:	:	:	:
1	1	1	1	1	1	0	0	252	$f_{UTAn}/252$
1	1	1	1	1	1	0	1	253	$f_{UTAn}/253$
1	1	1	1	1	1	1	0	254	$f_{UTAn}/254$
1	1	1	1	1	1	1	1	255	$f_{UTAn}/255$

Caution The baud rate is one half the frequency of the output clock signal from the 8-bit counter.

Remark 1. k: The value set with the BRGCAn register (k = 2, 3, 4, …, 255)
Remark 2. ×: Don't care
For an example of the baud rate setting, see **17.3.4 Baud rate generator, 3. Baud rate calculation, c) Baud rate setting example**.
Remark 3. n: Unit number (n = 0, 1)

17.2.7 Status register n (ASISAn)

The ASISAn register indicates the states in terms of errors and transmission on completion of reception by the serial interface UARTAn. It consists of three error flag bits (PEAn, FEAn, and OVEAn) and two transmission status flag bits (TXBFAn and TXSFAn). The ASISAn register is read-only and can be read by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H. The PEAn, FEAn, and OVEAn bits are initialized by clearing the UARTAENn or RXEAn bit to 0. These bits are also cleared by writing to the corresponding bit of the ASCTAn register. The TXBFAn and TXSFAn bits are initialized by clearing the UARTAENn or TXEAn bit to 0.

Figure 17 - 8 Format of Status Register n (ASISAn) (1/2)

Address: F0305H (ASISA0), F030DH (ASISA1)
After reset: 00H
R/W: R

Symbol	7	6	5	4	3	2	1	0
ASISAn	0	0	TXBFAn	TXSFAn	0	PEAn	FEAn	OVEAn

TXBFAn	Transmit buffer data flag
—	[Setting condition] • Data is written to the TXBAn register. (Data exists in the TXBAn register.) [Clearing conditions] • The UARTAENn or TXEAn bit is cleared to 0. • Data is transferred to the transmit shift register.

TXSFAn	Transmit shift register data flag
—	[Setting condition] • Data is transferred from the TXBAn register. (Data is being transmitted.) [Clearing conditions] • The UARTAENn or TXEAn bit is cleared to 0. • Data is transferred from the transmit shift register and then no subsequent data is transferred from the TXBAn register.

PEAn	Parity error flag
—	[Setting condition] • The parity of the received data does not match the parity bit. [Clearing conditions] • The UARTAENn or RXEAn bit is cleared to 0. • 1 is written to the PECTAn bit.

FEAn	Framing error flag
—	[Setting condition] • A stop bit is not detected when receiving data. [Clearing conditions] • The UARTAENn or RXEAn bit is cleared to 0. • 1 is written to the FECTAn bit.

Figure 17 - 8 Format of Status Register n (ASISAn) (2/2)

OVEAn	Overrun error flag
—	[Setting condition] • The next reception is completed before the receive data in the RXBAn register is read. [Clearing conditions] • The UARTAENn or RXEAn bit is cleared to 0. • 1 is written to the OVECTAn bit.

Caution 1. For continuous transmission, be sure to check that the TXBFAn flag is 0 after writing the first transmit data (the first byte) to the TXBAn register and then write the next transmit data (the second byte) to the TXBAn register. Otherwise, the transmit data become undefined. However, the TXBFAn flag need not be checked when continuous transmission is performed by using the buffer empty interrupt (ISSMAn bit = 1).

Caution 2. When initializing the transmission unit (TXEAn = 0) after completion of continuous transmission, be sure to check that the TXSFAn flag is 0 after the transfer completion interrupt is generated, and then initialize the unit. Otherwise, the transmit data become undefined.

Caution 3. The operation of the PEAn bit depends on the setting of the PSn1 and PSn0 bits of the ASIMAn1 register.

Caution 4. For the receive data, only the first 1 bit of the stop bits is checked regardless of the stop bit length.

Caution 5. When an overrun error occurs, the next receive data is not written to the RXBAn register and discarded.

Remark n: Unit number (n = 0, 1)

17.2.8 Status clear trigger register n (ASCTAn)

The ASCTAn register sets the trigger to clear the states in terms of errors on completion of reception of the serial interface UARTAn. It contains 3 bits of the error clear trigger flags (PECTAn, FECTAn, and OVECTAn). The ASCTAn register can be written by an 8-bit or 1-bit memory manipulation instruction. When the ASCTAn register is read, 00H is always read. The value of this register following a reset is 00H. Writing 1 to the PECTAn, FECTAn, and OVECTAn bits clears the PEAn, FEAn, and OVEAn bits of the ASISAn register, respectively. When writing 0, the corresponding error flags are not cleared.

Figure 17 - 9 Format of Status Clear Trigger Register n (ASCTAn)

Address: F0306H (ASCTA0), F030EH (ASCTA1)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
ASCTAn	0	0	0	0	0	PECTAn	FECTAn	OVECTAn

PECTAn **Note**	Parity error flag clear trigger
0	Does not clear the PEAn flag (the flag is retained).
1	Clears the PEAn flag.

FECTAn **Note**	Framing error flag clear trigger
0	Does not clear the FEAn flag (the flag is retained).
1	Clears the FEAn flag.

OVECTAn **Note**	Overrun error flag clear trigger
0	Does not clear the OVEAn flag (the flag is retained).
1	Clears the OVEAn flag.

Note Reading this bit returns 0.

Caution **After writing 1 to the trigger bit, the corresponding error flag is cleared on the next rising edge of the operating clock (f_{UTAn}). Accordingly, if reading the ASISAn register immediately after writing 1 to the trigger bit, the corresponding error flag may not have been cleared yet.**

Remark n: Unit number (n = 0, 1)

17.2.9 UARTA clock select register 0 (UTA0CK)

The UTA0CK register selects the operating clock of the serial interface UARTA0. The UTASEL1 and UTASEL0 bits select the clock source, f_{SEL}, for the serial interface UARTA0 from f_{MXP}, f_{IH}, and f_{IMP}. The bits from UTA0CK3 to UTA0CK0 select the operating clock for the serial interface UARTA0 from $f_{SEL}/1$ to $f_{SEL}/64$, f_{SXP}, and ELCL. Setting the UTA0OEN bit to 1 selects the serial interface UARTA0 clock output mode, and the operating clock for the serial interface UARTA0 is output from the CLKA0 pin. The UTA0CK register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 17 - 10 Format of UARTA Clock Select Register 0 (UTA0CK)

Address: F0310H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
UTA0CK	UTA0OEN	0	UTASEL1	UTASEL0	UTA0CK3	UTA0CK2	UTA0CK1	UTA0CK0

UTA0OEN **Note**	Serial interface UARTA0 clock output enable
0	Disables CLKA0 output.
1	Enables CLKA0 output.

UTASEL1	UTASEL0	f_{SEL} clock select
0	0	Stop
0	1	f_{MXP}
1	0	f_{IHP}
1	1	f_{IMP}

UTA0CK3	UTA0CK2	UTA0CK1	UTA0CK0	Serial interface UARTA0 operation clock select (f_{UTA0})
0	0	0	0	f_{SEL}
0	0	0	1	$f_{SEL}/2$
0	0	1	0	$f_{SEL}/4$
0	0	1	1	$f_{SEL}/8$
0	1	0	0	$f_{SEL}/16$
0	1	0	1	$f_{SEL}/32$
0	1	1	0	$f_{SEL}/64$
1	0	0	0	f_{SXP}
1	0	0	1	ELCL
Other than above				Setting prohibited

Note Set this bit to 0 in the R7F100G products with 36 to 52 pins because the CLKA0 output pin is not present in the given products.

Caution **This register should be read or written when the TXEA0 and RXEA0 bits are 0 (in the transmission/reception stopped state).**

(Remark is listed on the next page.)

Remark f_{MXP}: High-speed peripheral clock
f_{IHP}: High-speed on-chip oscillator peripheral clock
f_{IMP}: Middle-speed on-chip oscillator peripheral clock
f_{SXP}: Low-speed peripheral clock
f_{SEL}: Selected clock to be divided for the UARTA
ELCL: Event input from the ELCL

17.2.10 UARTA clock select register 1 (UTA1CK)

The UTA1CK register selects the operating clock of the serial interface UARTA1. The bits from UTA1CK3 to UTA1CK0 select the operating clock for the serial interface UARTA1 from f_{SEL}/1 to f_{SEL}/64, f_{SXP}, and ELCL. Setting the UTA1OEN bit to 1 selects the serial interface UARTA1 clock output mode, and the operating clock for the serial interface UARTA1 is output from the CLKA1 pin. The UTA1CK register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 17 - 11 Format of UARTA Clock Select Register 1 (UTA1CK)

Address: F0311H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
UTA1CK	UTA1OEN	0	0	0	UTA1CK3	UTA1CK2	UTA1CK1	UTA1CK0

UTA1OEN **Note**	Serial interface UARTA1 clock output enable
0	Disables CLKA1 output.
1	Enables CLKA1 output.

UTA1CK3	UTA1CK2	UTA1CK1	UTA1CK0	Serial interface UARTA1 operation clock select (f_{UTA1})
0	0	0	0	f_{SEL}
0	0	0	1	f_{SEL}/2
0	0	1	0	f_{SEL}/4
0	0	1	1	f_{SEL}/8
0	1	0	0	f_{SEL}/16
0	1	0	1	f_{SEL}/32
0	1	1	0	f_{SEL}/64
1	0	0	0	f_{SXP}
1	0	0	1	ELCL
Other than above				Setting prohibited

Note Set this bit to 0 in the 44- to 52-pin R7F100G and 32- to 48-pin R7F104G products because the CLKA1 output pin is not present in the given products.

Caution **This register should be read or written when the TXEA1 and RXEA1 bits are 0 (in the transmission/reception stopped state).**

Remark f_{SXP}: Low-speed peripheral clock frequency
f_{SEL}: Selected clock to be divided for the UARTA
ELCL: Event input from the ELCL

17.2.11 UART loopback select register (ULBS)

The ULBS register is used to enable the UART loopback function. This register has a bit to control loopback for each UART. Setting a bit to 1 enables the loopback function for the given UART, with the output from the transmit shift register being looped back to the receive shift register. The ULBS register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 17 - 12 Format of the UART Loopback Select Register (ULBS)

Address: F0079H
After reset: 00H
R/W: R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
ULBS	0	0	ULBS5	ULBS4	ULBS3	ULBS2	ULBS1	ULBS0

ULBS5	Selection of the UART loopback function
0	The state of the RxDA1 pin of the serial interface UARTA1 is input to the receive shift register.
1	The output from the transmit shift register is looped back to the receive shift register.

ULBS4	Selection of the UART loopback function
0	The state of the RxDA0 pin of the serial interface UARTA0 is input to the receive shift register.
1	The output from the transmit shift register is looped back to the receive shift register.

Caution 1. Be sure to leave bits 7 and 6 set to 0.

Caution 2. When using the loopback function in the serial interface UARTA0, set the PFOE14 bit of the port function output enable register 1 (PFOE1) to 1.

Remark The RxDA0 pin can be used only in the 36- to 128-pin R7F100G and 32- to 48-pin R7F104G products. The RxDA1 pin can be used only in the 44- to 128-pin R7F100G products.

17.2.12 Registers for controlling the port functions multiplexed with the inputs and outputs of the UARTA serial interface

Set the following registers to control the port functions multiplexed with the inputs and outputs of the UARTA serial interface.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Port input mode registers (PIMxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port function output enable registers (PFOEx)

For details, see the following sections.

- **4.3.1 Port mode registers (PMxx)**
- **4.3.2 Port registers (Pxx)**
- **4.3.4 Port input mode registers (PIMxx)**
- **4.3.5 Port output mode registers (POMxx)**
- **4.3.7 Port mode control A registers (PMCAxx)**
- **4.3.8 Port mode control T registers (PMCTxx)**
- **4.3.15 Port function output enable registers (PFOEx)**

For examples of register settings for controlling port functions, see **4.5.4 Examples of register settings for port and multiplexed functions**. When connecting an external device operating at a different voltage (1.8 V, 2.5 V, or 3 V), see **4.4.5 Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching I/O buffers**.

17.3 Operation

The serial interface UARTAn operates in either of the following two modes.

- Operation stop mode
- UART mode

17.3.1 Operation stop mode

In the operation stop mode, serial communication is not performed, and thus the power consumption can be reduced. In addition, in this mode, the pins can be used as ordinary port pins. To set the operation stop mode, clear bits 7 to 5 (UARTAENn, TXEAn, RXEAn) of the ASIMAn0 register to 0.
The bus clock is not stopped by the above setting. To completely stop operation, clear the UTAEN bit of the PER1 register to 0 after the above setting.

17.3.2 UART mode

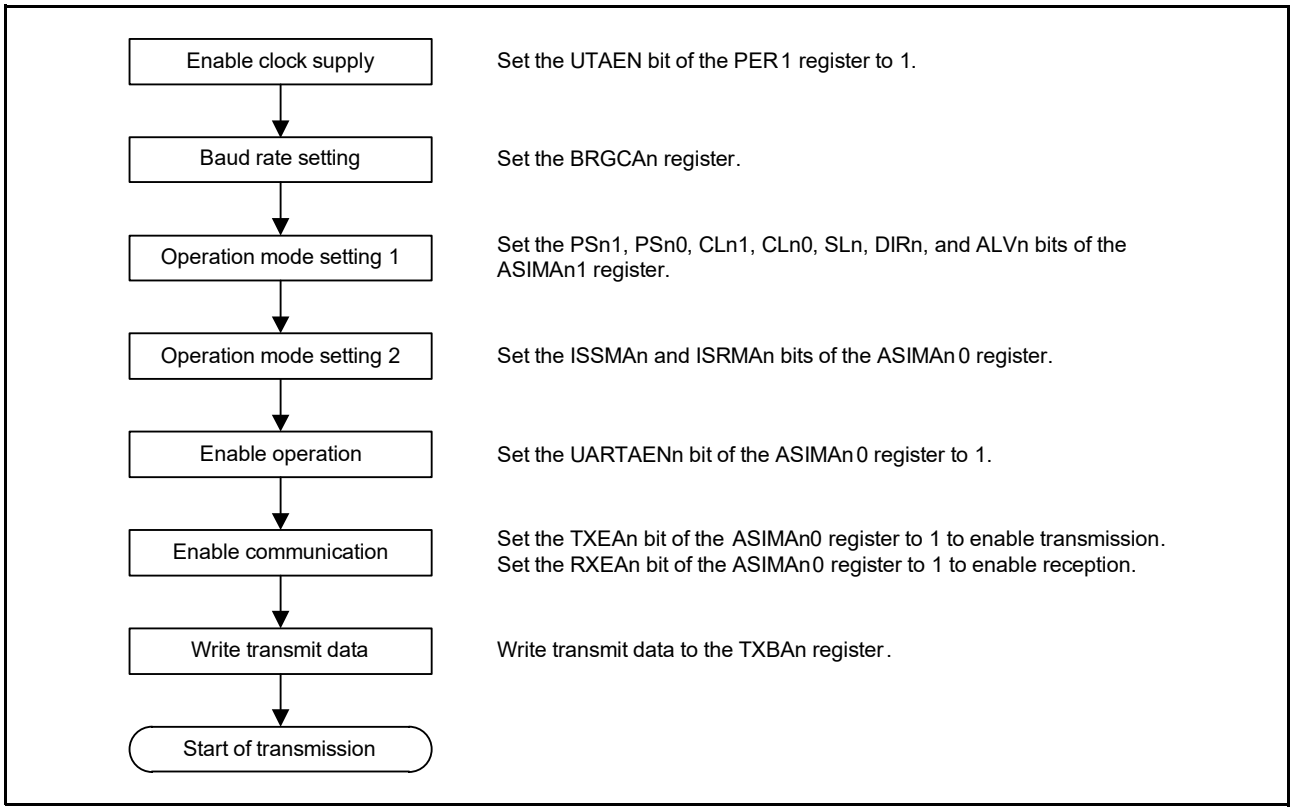
In this mode, one byte of data is transmitted and one byte is received following the start bit. That is, operation is full duplex. A dedicated UART baud rate generator is incorporated, so that communication can be executed at a wide range of baud rates.

Remark n: Unit number (n = 0, 1)

1. Communication procedure

Figure 17 - 13 shows the flowchart of communication procedure.

Figure 17 - 13 Flowchart of Communication Procedure



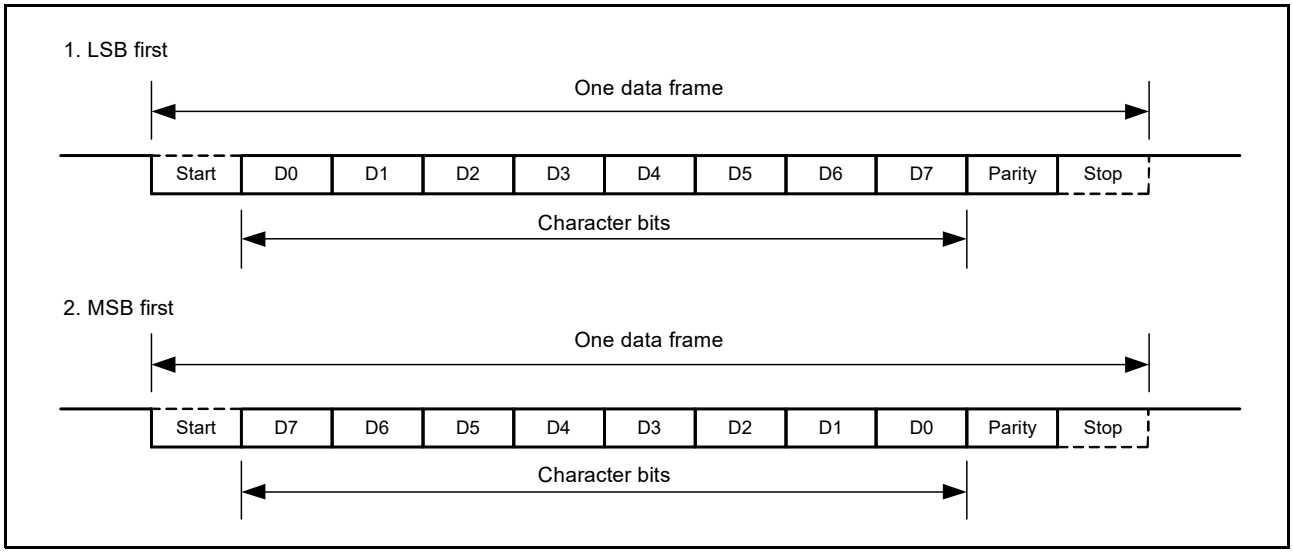
Caution **When a pin is to be used for UART reception, use the port-related register to set the given pin to input mode. When a pin is to be used for UART transmission, use the port register (Pxx) for the given pin to set the initial output level and then use the port-related register to set the pin to output mode. Make these settings before enabling transmission. The port-related registers to be used for the settings stated above differ with the channel and product to be used. For further details on the settings, see 4.5 Register Settings When Using Multiplexed Function. When connecting an external device operating at a different voltage (1.8 V, 2.5 V or 3 V), see 4.4.5 Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching I/O buffers.**

Remark n: Unit number (n = 0, 1)

2. Format and waveform example of transmit/receive data

The following describes the communication data format of the serial interface UARTAn. **Figure 17 - 14** shows the data format.

Figure 17 - 14 Transmit/Receive Data Format



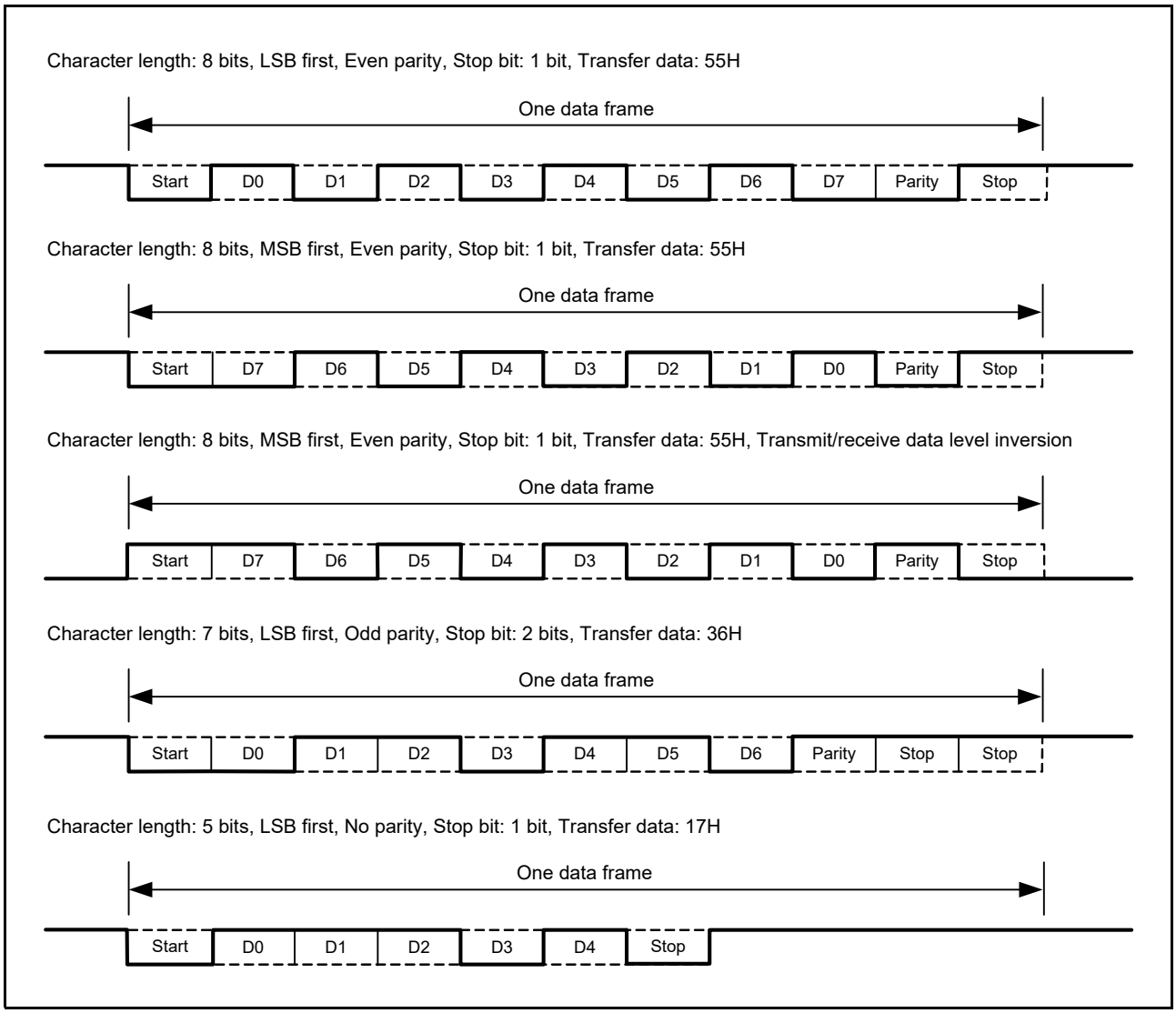
One data frame consists of the following bits.

- Start bit: 1 bit
- Character bits: 5, 7 or 8 bits
- Parity bit: Even parity, odd parity, 0 parity, or no parity
- Stop bit: 1 or 2 bits

The character bit length, the parity, the stop bit length, the transfer direction (LSB/MSB first), and the TxDAn pin output (direct/inverted) in one data frame are specified by the ASIMAn1 register.

Figure 17 - 15 shows the examples of transmit/receive data waveforms.

Figure 17 - 15 Example of Transmit/Receive Data Waveform



3. Parity types and operation

The parity bit is used to detect a bit error in communication data. Usually, the same type of parity bit is used on both the transmitting and reception sides. With even/odd parity, a 1-bit (odd number) error can be detected. With zero/no parity, an error cannot be detected.

a) Even parity

- In transmission

  Data for transmission, including the parity bit, are controlled so that an even number of bits have the value 1. The value of the parity bit is set as follows.

  If the data for transmission have an odd number of bits with the value 1: 1

  If the data for transmission have an even number of bits with the value 1: 0

- In reception

  In the data for reception, including the parity bit, the number of bits with the value 1, is counted. If it is odd, a parity error occurs.

b) Odd parity

- In transmission

  Unlike even parity, data for transmission, including the parity bit, are controlled so that an odd number of bits have the value 1.

  If the data for transmission have an odd number of bits with the value 1: 0

  If the data for transmission have an even number of bits with the value 1: 1

- In reception

  In the data for reception, including the parity bit, the number of bits with the value 1, is counted. If it is even, a parity error occurs.

c) 0 parity

The parity bit is cleared to 0 when data is transmitted, regardless of the transmit data. The parity bit is not detected when the data is received. Therefore, a parity error does not occur regardless of whether the parity bit is 0 or 1.

d) No parity

No parity bit is appended to the transmit data. Reception is performed assuming that there is no parity bit. A parity error does not occur, because there is no parity bit.

4. Normal transmission

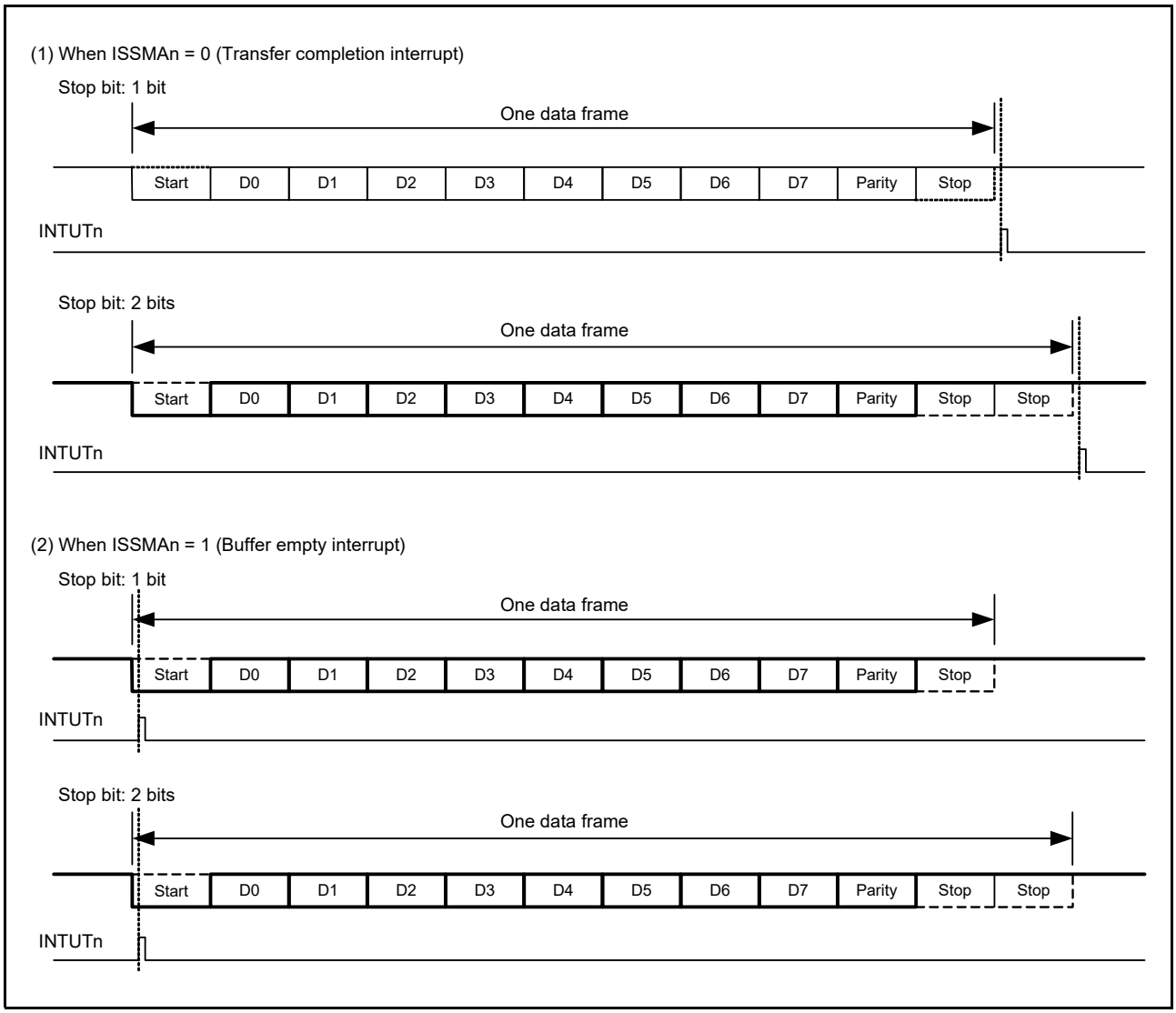
Transmission is enabled by setting the UARTAENn bit of the operation mode setting register n0 (ASIMAn0) to 1 and then setting the TXEAn bit of the ASIMAn0 register to 1. Transmission can be started by writing the data for transmission to the transmit buffer register n (TXBAn). The start bit, parity bit, and stop bit are automatically appended to the data. When transmission is started, the data in the TXBAn register are transferred to the transmit shift register. After that, the transmit data are sequentially output from the transmit shift register to the TxDAn pin in the specified transfer direction. When transmission is completed, the parity and stop bits which are set by the ASIMAn0 register are appended and a transfer completion interrupt request signal (INTUTn) is generated. Transmission is suspended until the next transmit data is written to the TXBAn register.

Figure 17 - 16 shows the timing of the transfer completion interrupt request signal (INTUTn). INTUTn is issued at the following timing.

a) When ISSMAn = 0 (INTUTn functions as a transfer completion interrupt.)
INTUTn is issued after the output of the last stop bit.

b) ISSMAn = 1 (INTUTn functions as a buffer empty interrupt.)
INTUTn is issued when the start bit is output.

Figure 17 - 16 Interrupt Output Timing



5. Continuous transmission

The serial interface UARTAn has two separate registers for continuous transmission: the transmit buffer register n (TXBAn) and the transmit shift register. At the moment the transmit shift register starts a shift operation, the next transmit data can be written to the transmit buffer register n (TXBAn). This operation enables continuous transmission, thereby improving communication rate. Note that continuous transmission is not achieved when writing to the TXBAn register is not completed within the maximum number of clock cycles defined below from generation of the buffer empty interrupt.

Maximum number of clock cycles = Data transfer length × 2k - (2k + 3)
k: The value set with the BRGCAn bits (k = 2, 3, 4, …, 255)

An example of calculating the maximum number of clock cycles is described below.
When the BRGCAn register = 02H (k = 2), start bit = 1 bit, character length = 8 bits, parity used, and stop bit = 1 bit:
The maximum number of clock cycles = Transfer length × 2k - (2k + 3) = 11 × 2 × 2 - (2 × 2 + 3) = 37
(Writing must be completed within 37 cycles of the serial interface UARTAn operating clock (f_{UTAn}).)

Continuous transmission is achieved by the following two methods.

a) Continuous transmission by polling

Continuous transmission is achieved by polling the transmit buffer data flag (TXBFAn bit) and the transmit shift register data flag (TXSFAn bit) of the status register n (ASISAn). When using this method, clear the ISSMAn bit of the operation mode setting register n0 (ASIMAn0) to 0.

- At the start of and during continuous transmission

  At the start of continuous transmission, write the first byte of data to the TXBAn register, check that the transmit buffer data flag (TXBFAn) is 0, and then write the second byte of data. In a similar way, check that the TXBFAn flag is 0 and then write the subsequent data to the TXBAn register.

TXBFAn	Determination flag indicating that writing to TXBAn is enabled or disabled at the start of continuous transmission
0	Writing is enabled.
1	Writing is disabled.

Caution To determine if continuous transmission is enabled or disabled, only check the TXBFAn flag. The TXSFAn flag must not be used for the determination in combination with this flag.

- Completion of continuous transmission

  In continuous transmission, when data in the transmit shift register and the TXBAn register are transmitted after the required number of transmit data are written to the TXBAn register, the continuous transmission is completed. To confirm the completion, check the setting of the transmit shift register data flag (TXSFAn).

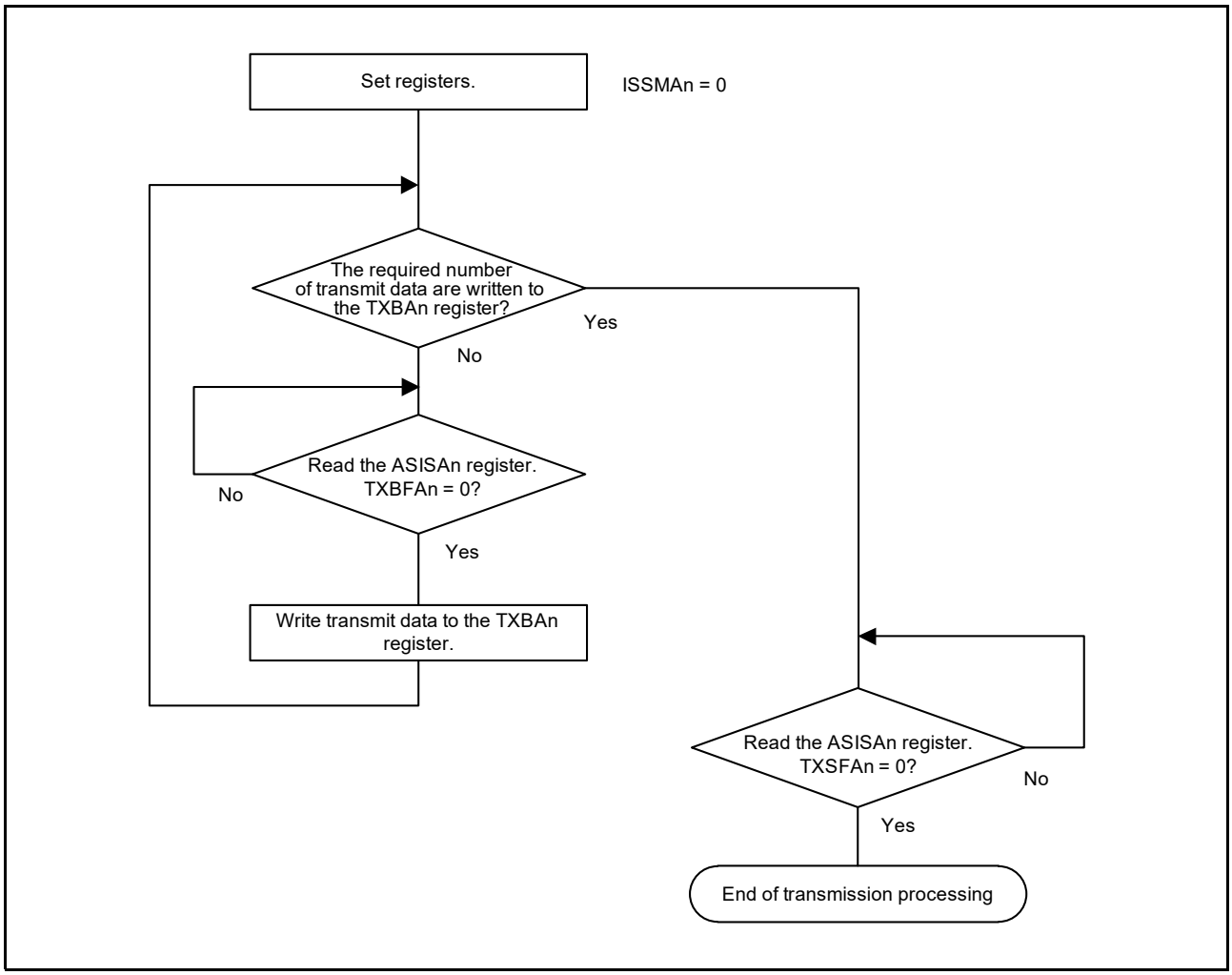
TXSFAn	Confirmation flag indicating whether transmission is in progress or not
0	Transmission is completed.
1	Transmission is in progress.

Caution 1. When initializing the transmission unit after completion of continuous transmission, check that the TXSFAn flag is 0 after the transfer completion interrupt is generated, and then initialize the unit.

Caution 2. During continuous transmission, after transmission of one data frame, the subsequent transmission may be completed before execution of the INTUTn interrupt processing. This can be detected by incorporating the program that counts the number of transmit data and by referencing the TXSFAn flag.

Figure 17 - 17 shows a flow example of continuous transmission processing by polling.

Figure 17 - 17 Flow Example of Continuous Transmission Processing by Polling



b) Continuous transfer by using an interrupt

Continuous transmission is achieved by using the interrupt (INTUTn). An interrupt can be generated when data in the transmit buffer register n (TXBAn) are transferred to the transmit shift register by setting the ISSMAn bit to 1 in the operation mode setting register n0 (ASIMAn0). With this setting, continuous transmission is enabled by writing data to the TXBAn register on occurrence of the buffer empty interrupt. In addition, the transfer completion interrupt can be generated on completion of continuous transmission by clearing the ISSMAn bit to 0 after writing the last transmit data to the TXBAn register.

Figure 17 - 18 shows a flow example of continuous transmission using interrupt.

Figure 17 - 18 Flow Example of Continuous Transmission Using Interrupt

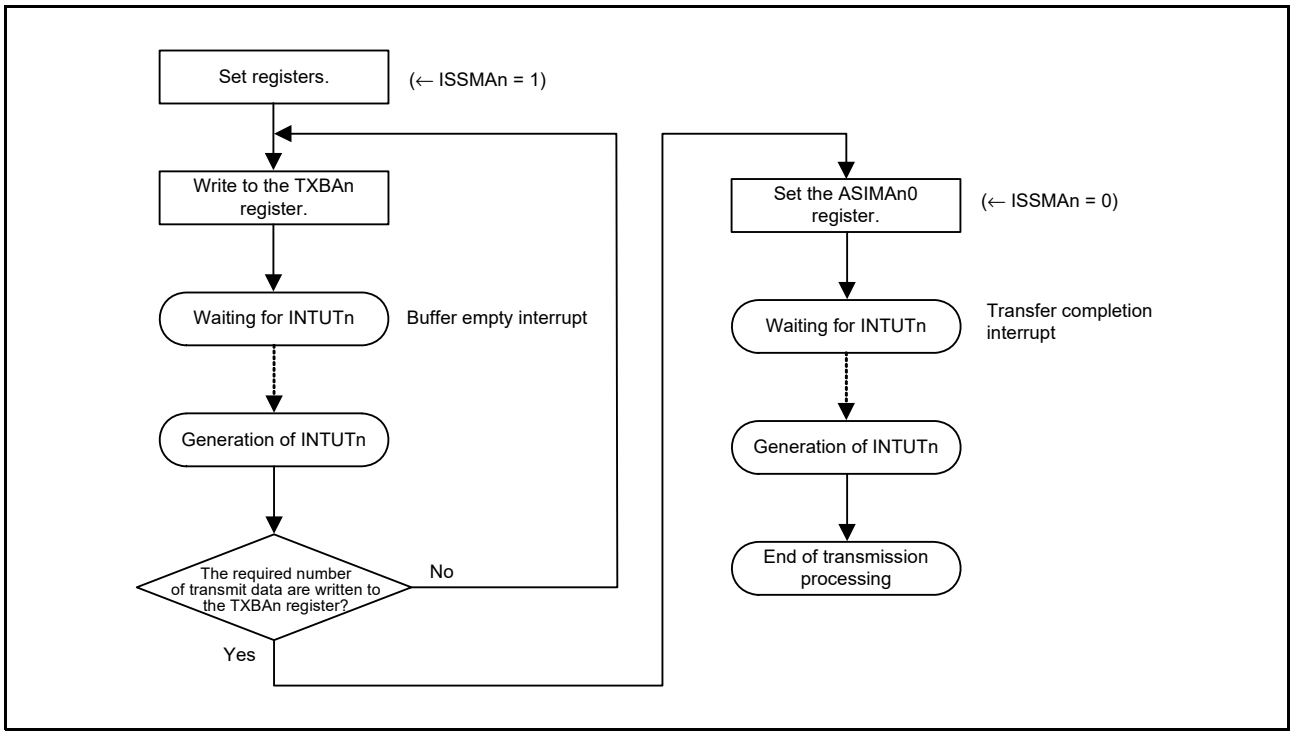
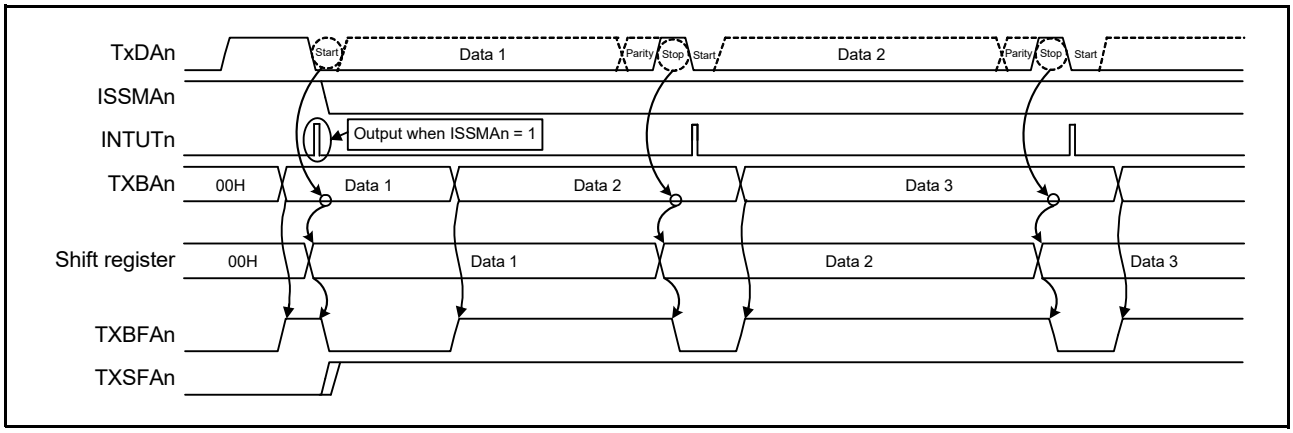


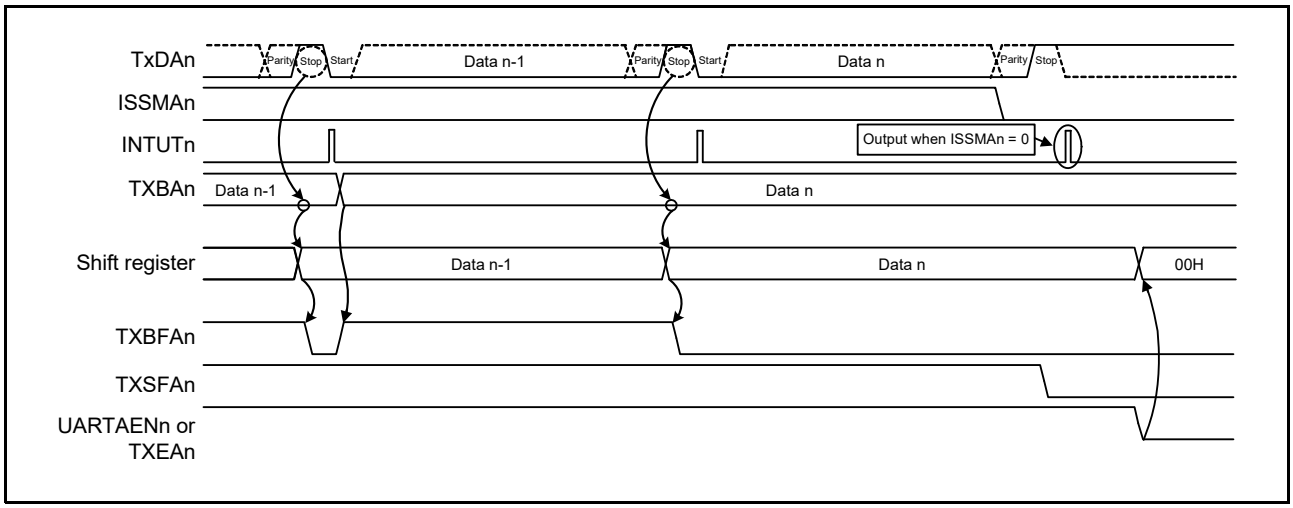
Figure 17 - 19 and **Figure 17 - 20** show the timing charts when the continuous transmission is started and completed, respectively.

Figure 17 - 19 Timing Chart When Continuous Transmission Is Started



Caution When the ASISAn register is read, the value of the TXBFAn and TXSFAn flags is 11B over certain periods. Accordingly, use only the TXBFAn flag to determine if writing is enabled or disabled.

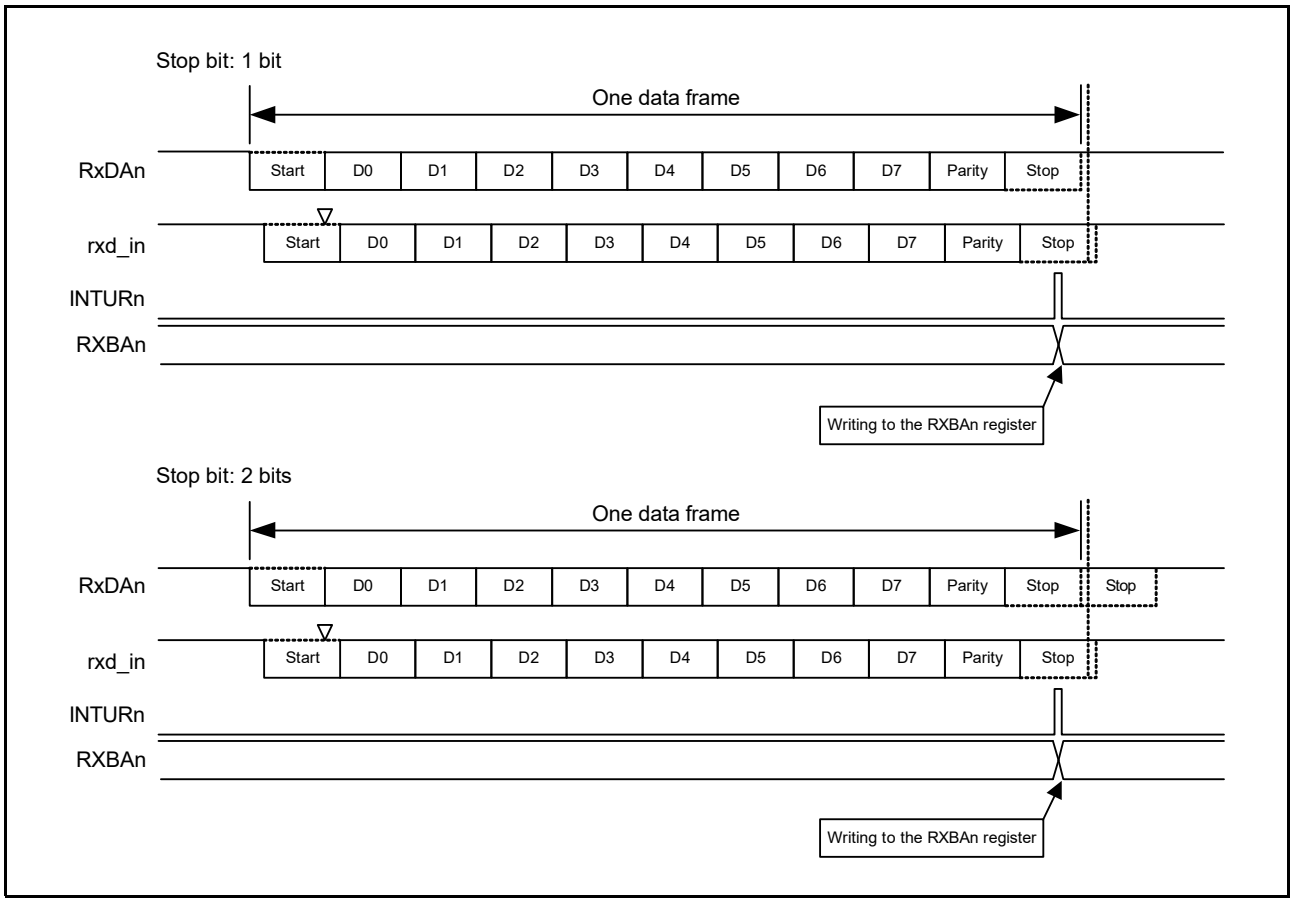
Figure 17 - 20 Timing Chart When Continuous Transmission Is Completed



6. Normal reception

When setting the UARTAENn bit of the operation mode setting register n0 (ASIMAn0) to 1 and then setting the RXEAn bit of the ASIMAn0 register to 1, reception is enabled, and sampling of the input to the RxDAn pin is performed. When the ALVn bit is 0, the 8-bit counter of the baud rate generator starts counting on detection of the falling edge on the RxDAn pin. When the counter reaches the set value of the baud rate generator control register n (BRGCAn), the input to the RxDAn pin is sampled again (at the point indicated with ∇ in **Figure 17 - 21**). If the RxDAn pin is low, it is regarded as a start bit. When the ALVn bit is 1, the 8-bit counter of the baud rate generator starts counting on detection of the rising edge on the RxDAn pin. When the counter reaches the set value of the baud rate generator control register n (BRGCAn), the input to the RxDAn pin is sampled again (at the point indicated with ∇ in **Figure 17 - 21**). If the RxDAn pin is high, it is regarded as a start bit. **Figure 17 - 21** shows the timing chart of receive operation. On detection of a start bit, receive operation is started: serial data is sequentially stored in the receive shift register at a specified baud rate. On reception of a stop bit, the transfer completion interrupt (INTURn) is generated, and at the same time, the data in the receive shift register is written to the receive buffer register n (RXBAn). Note that when an overrun error (OVEAn) occurs, the data received on occurrence of the error is not written to the RXBAn register. When a parity error (PEAn) or a framing error (FEAn) occurs during reception, reception continues until a stop bit is received. After completion of the reception, the reception error interrupt (INTURn/INTUREn) set in the ISRMAn bit is generated. When a reception error occurs, read the status register n (ASISAn) and then read the receive buffer register n (RXBAn) to clear the error flag. If the receive buffer register n (RXBAn) is not read, an overrun error will occur when the next data is received: the reception error state will continue. Reception is always handled as including a stop bit. Accordingly, the second stop bit is ignored.

Figure 17 - 21 Timing of UART Receive Operation



Remark 1. rxd_in is the internal signal generated by latching RxDAn with a noise filter. rxd_in is delayed relative to RxDAn by maximum of 3 cycles of the UART operation clock.

Remark 2. The INTUR output timing in the figure is just an example. The timing relative to RxDAn varies according to the setting of the BRGCAn register.

7. Reception error

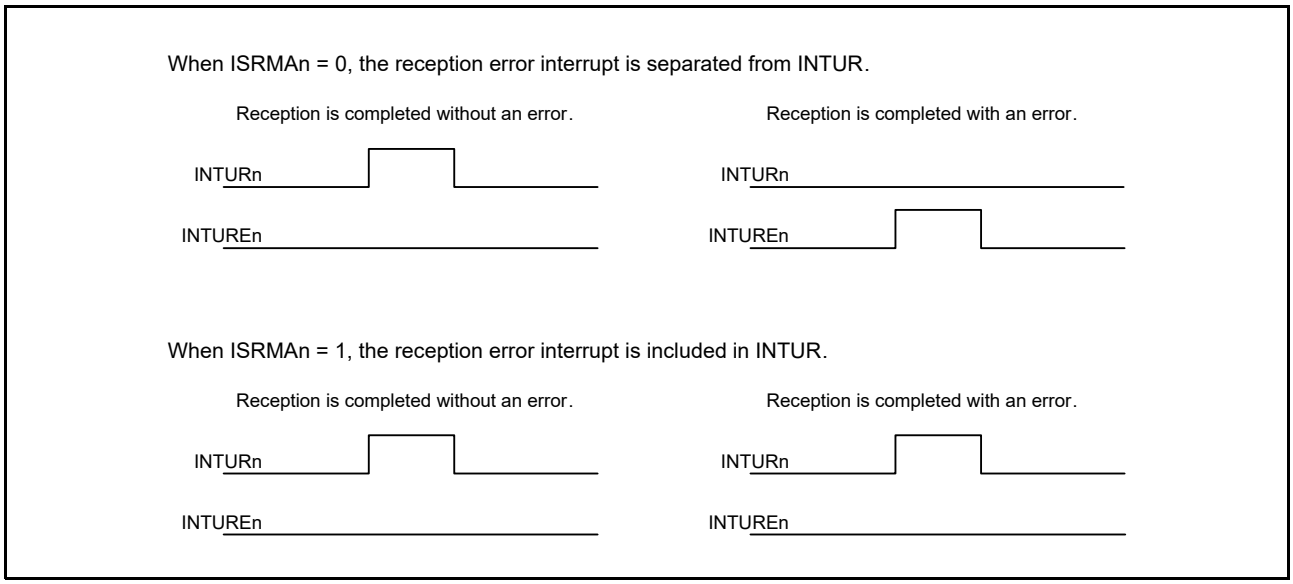
Three types of errors may occur during reception; parity error, framing error, and overrun error. When these errors occur, the corresponding error flag in the status register n (ASISAn) is set, and the reception error interrupt request signal (INTURn or INTUREn) is generated. The type of the reception error can be identified by the reception error interrupt processing routine, which reads and checks the contents of the status register n (ASISAn). The contents of the ASISAn register are cleared to 0 by setting the corresponding bit of the status clear trigger register n (ASCTAn) to 1. **Table 17 - 3** shows the causes of the reception errors.

Table 17 - 3 Causes of Reception Errors

Error Flag	Reception Error	Cause
PEAn	Parity error	The parity specified for reception does not match the parity of receive data.
FEAn	Framing error	No stop bit is detected.
OVEAn	Overrun error	Before the receive data is read from the receive buffer, the next data reception is completed.

Setting the ISRMAn bit of the operation mode setting register n0 (ASIMAn0) to 0 allows the reception error interrupt to be separated from INTURn and allows it to be generated as INTUREn. **Figure 17 - 22** shows the interrupt output waveform which varies depending on the setting of the ISRMAn bit.

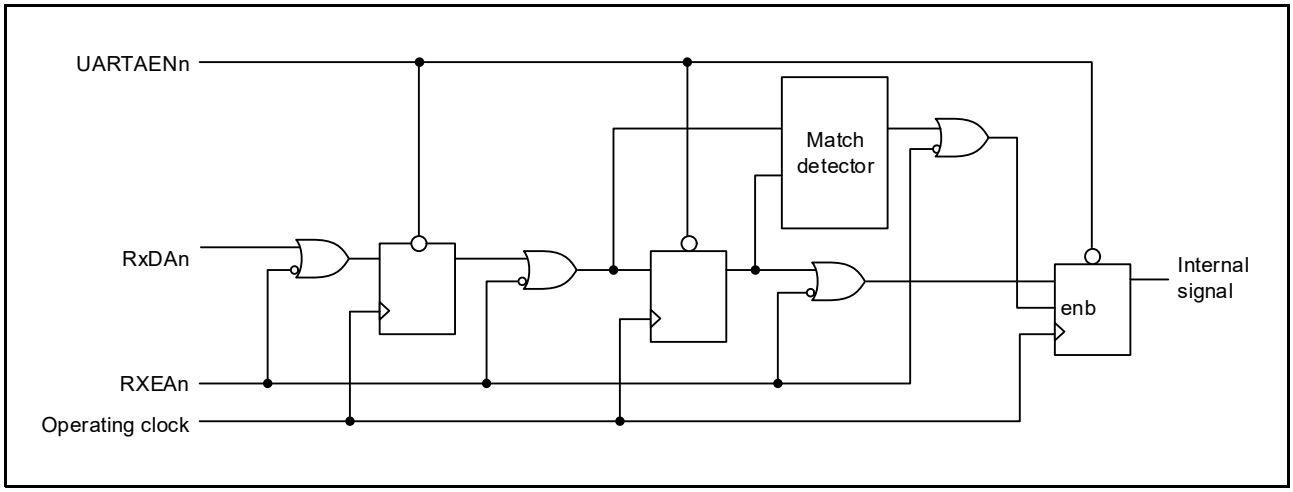
Figure 17 - 22 Various Interrupt Output Waveforms Depending on ISRMAn Setting



17.3.3 Receive data noise filter

This filter samples the receive data (RxDAn), and determines the level when the same level is sampled twice. The receive data is delayed by maximum of three cycles of the operating clock because of the circuit configuration. **Figure 17 - 23** shows the noise filter circuit.

Figure 17 - 23 Noise Filter



Caution 1. When ALVn = 0 (wait state = high level; start bit = low level), the initial value of the receive data (RxDAn) needs to be "high".

Caution 2. When ALVn = 1 (wait state = low level; start bit = high level), the initial value of the receive data (RxDAn) needs to be "low".

Remark n: Unit number (n = 0, 1)

17.3.4 Baud rate generator

The baud rate generator consists of 8-bit programmable counters, and generates a serial clock for transmission/reception of the serial interface UARTAn. An 8-bit counter is provided each for transmission and reception.

1. Configuration of baud rate generator
   a) Serial interface UARTAn operation clock
      When the UARTAENn bit is 1 in the operation mode setting register n0 (ASIMAn0), the serial interface UARTAn operation clock (f_{UTAn}) is supplied to each module. When the UARTAENn bit is 0, the serial interface UARTAn operation clock is fixed to low level.

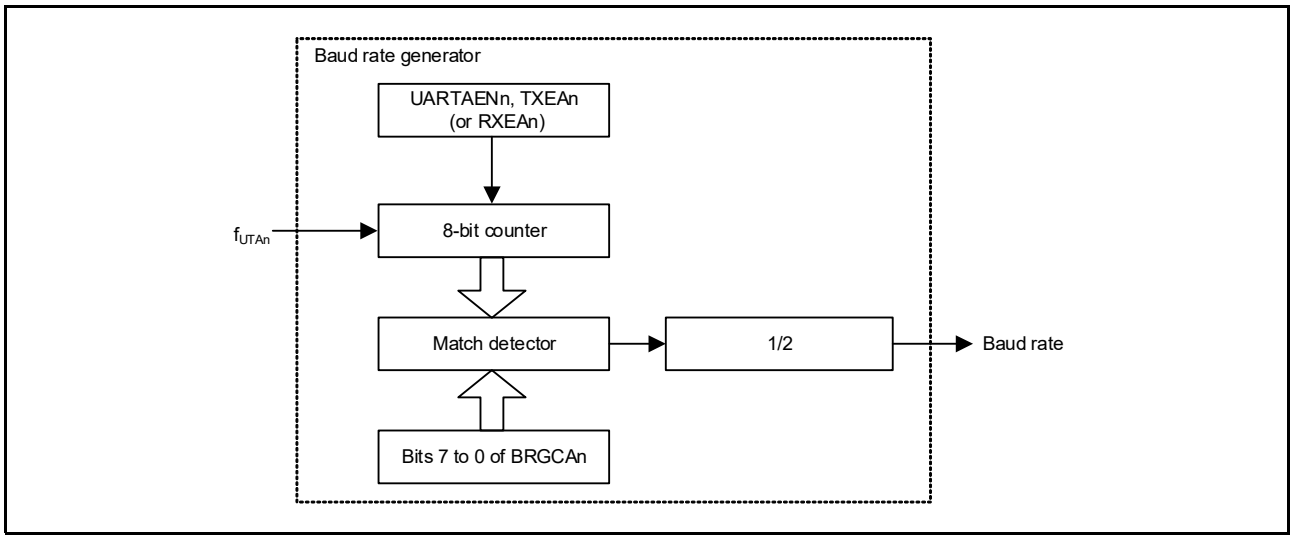
   b) Transmission counter
      This counter is cleared to 0 and stops when the UARTAENn bit is 0 or the TXEAn bit is 0 in the operation mode setting register n0 (ASIMAn0). It starts counting when the UARTAENn bit is 1 and the TXEAn bit is 1. The counter is cleared to 0 when the first transmit data is written to the transmit buffer register n (TXBAn). When continuous transmission is performed, the counter is cleared to 0 again when transmission of one frame of data has been completed. If there is no data to be transmitted next, the counter is not cleared to 0 and continues counting until the UARTAENn or TXEAn bit is cleared to 0. When the UARTAENn bit is 0 or the TXEAn bit is 0 in the ASIMAn0 register, the counter stops at 00H.

   c) Reception counter
      This counter is cleared to 0 and stops when the UARTAENn bit is 0 or the RXEAn bit is 0 in the operation mode setting register n0 (ASIMAn0). It starts counting when the start bit is detected. The counter stops operation after one frame has been received, until the next start bit is detected. When the UARTAENn bit is 0 or the RXEAn bit is 0 in the ASIMAn0 register, the counter stops at 00H.

Remark n: Unit number (n = 0, 1)

Figure 17 - 24 shows the configuration of the baud rate generator.

Figure 17 - 24 Configuration of Baud Rate Generator



2. Generation of serial clock
   A serial clock to be generated can be specified by using the baud rate generator control register n (BRGCAn). The baud rate generator divides the frequency of the input clock signal to the 8-bit counter (f_{UTAn}) by the divisor set by the BRGCAn register. The result of this division is further divided by 2 to produce the serial clock.

3. Baud rate calculation

   a) Baud rate calculation expression
      The baud rate can be calculated by the following expression.
      Baud rate = $f_{UTAn} \div (2 \times k)$ [bps]

      f_{UTAn}: Frequency of operating clock
      k: Value set by bits 7 to 0 of the BRGCAn register (k = 2, 3, 4, …, 255)

   b) Baud rate error
      The baud rate error can be calculated by the following expression.

$$\text{Error} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (correct baud rate)}} - 1 \right) \times 100\ [\%]$$

Caution 1. Keep the baud rate error during transmission to within the permissible error range on the reception side.

Caution 2. Make sure that the baud rate error during reception satisfies the permissible baud rate error range during reception. Permissible baud rate error during reception is described in 17.3.4 Baud rate generator, 3. Baud rate calculation, d) Permissible baud rate range during reception.

c) Baud rate setting example

Table 17 - 4 Set Data of Baud Rate Generator (1/4)

Desired baud rate	In operation with f_{IHP} = 32 MHz (UTASEL1 and UTASEL0 = 10B)													
	No division (UTAnCK3 to UTAnCK0 = 0000B)		×1/2 (UTAnCK3 to UTAnCK0 = 0001B)		×1/4 (UTAnCK3 to UTAnCK0 = 0010B)		×1/8 (UTAnCK3 to UTAnCK0 = 0011B)		×1/16 (UTAnCK3 to UTAnCK0 = 0100B)		×1/32 (UTAnCK3 to UTAnCK0 = 0101B)		×1/64 (UTAnCK3 to UTAnCK0 = 0110B)	
	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate
200 bps	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled		Disabled	
300 bps	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled		Disabled	
600 bps	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled		Disabled	
1200 bps	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled		208	0.16%
2400 bps	Disabled		Disabled		Disabled		Disabled		Disabled		208	0.16%	104	0.16%
4800 bps	Disabled		Disabled		Disabled		Disabled		208	0.16%	104	0.16%	52	0.16%
9600 bps	Disabled		Disabled		Disabled		208	0.16%	104	0.16%	52	0.16%	26	0.16%
19200 bps	Disabled		Disabled		208	0.16%	104	0.16%	52	0.16%	26	0.16%	13	0.16%
38400 bps	Disabled		208	0.16%	104	0.16%	52	0.16%	26	0.16%	13	0.16%	Disabled	
76800 bps	208	0.16%	104	0.16%	52	0.16%	26	0.16%	13	0.16%	Disabled		Disabled	
115200 bps	139	-0.08%	69	0.64%	35	-0.79%	17	2.12%	Disabled		Disabled		Disabled	
153600 bps	104	0.16%	52	0.16%	26	0.16%	13	0.16%	Disabled		Disabled		Disabled	

Remark k: Value set by bits 7 to 0 of the baud rate generator control register n (BRGCAn) (k = 2, 3, 4, ..., 255)
n: Unit number (n = 0, 1)

Table 17 - 4 Set Data of Baud Rate Generator (2/4)

Desired baud rate	In operation with f_{IMP} = 4 MHz (UTASEL1 and UTASEL0 = 11B)													
	No division (UTAnCK3 to UTAnCK0 = 0000B)		×1/2 (UTAnCK3 to UTAnCK0 = 0001B)		×1/4 (UTAnCK3 to UTAnCK0 = 0010B)		×1/8 (UTAnCK3 to UTAnCK0 = 0011B)		×1/16 (UTAnCK3 to UTAnCK0 = 0100B)		×1/32 (UTAnCK3 to UTAnCK0 = 0101B)		×1/64 (UTAnCK3 to UTAnCK0 = 0110B)	
	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate
200 bps	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled		156	0.16%
300 bps	Disabled		Disabled		Disabled		Disabled		Disabled		208	0.16%	104	0.16%
600 bps	Disabled		Disabled		Disabled		Disabled		208	0.16%	104	0.16%	52	0.16%
1200 bps	Disabled		Disabled		Disabled		208	0.16%	104	0.16%	52	0.16%	26	0.16%
2400 bps	Disabled		Disabled		208	0.16%	104	0.16%	52	0.16%	26	0.16%	13	0.16%
4800 bps	Disabled		208	0.16%	104	0.16%	52	0.16%	26	0.16%	13	0.16%	Disabled	
9600 bps	208	0.16%	104	0.16%	52	0.16%	26	0.16%	13	0.16%	Disabled		Disabled	
19200 bps	104	0.16%	52	0.16%	26	0.16%	13	0.16%	Disabled		Disabled		Disabled	
38400 bps	52	0.16%	26	0.16%	13	0.16%	Disabled		Disabled		Disabled		Disabled	
76800 bps	26	0.16%	13	0.16%	Disabled		Disabled		Disabled		Disabled		Disabled	
115200 bps	17	2.12%	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled	
153600 bps	13	0.16%	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled	

Remark k: Value set by bits 7 to 0 of the baud rate generator control register n (BRGCAn) (k = 2, 3, 4, ..., 255)
n: Unit number (n = 0, 1)

Table 17 - 4 Set Data of Baud Rate Generator (3/4)

Desired baud rate	In operation with f_{MXP} = 20 MHz (UTASEL1 and UTASEL0 = 01B)													
	No division (UTAnCK3 to UTAnCK0 = 0000B)		×1/2 (UTAnCK3 to UTAnCK0 = 0001B)		×1/4 (UTAnCK3 to UTAnCK0 = 0010B)		×1/8 (UTAnCK3 to UTAnCK0 = 0011B)		×1/16 (UTAnCK3 to UTAnCK0 = 0100B)		×1/32 (UTAnCK3 to UTAnCK0 = 0101B)		×1/64 (UTAnCK3 to UTAnCK0 = 0110B)	
	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate	k	Error from the desired baud rate
200 bps	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled		Disabled	
300 bps	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled		Disabled	
600 bps	Disabled		Disabled		Disabled		Disabled		Disabled		Disabled		255	2.12%
1200 bps	Disabled		Disabled		Disabled		Disabled		Disabled		255	2.12%	130	0.16%
2400 bps	Disabled		Disabled		Disabled		Disabled		255	2.12%	130	0.16%	65	0.16%
4800 bps	Disabled		Disabled		Disabled		255	2.12%	130	0.16%	65	0.16%	33	-1.36%
9600 bps	Disabled		Disabled		255	2.12%	130	0.16%	65	0.16%	33	-1.36%	16	1.73%
19200 bps	Disabled		255	2.12%	130	0.16%	65	0.16%	33	-1.36%	16	1.73%	8	1.73%
38400 bps	255	2.12%	130	0.16%	65	0.16%	33	-1.36%	16	1.73%	8	1.73%	4	1.73%
76800 bps	130	0.16%	65	0.16%	33	-1.36%	16	1.73%	8	1.73%	4	1.73%	Disabled	
115200 bps	87	-0.22%	43	0.94%	22	-1.36%	11	-1.36%	Disabled		Disabled		Disabled	
153600 bps	65	0.16%	33	-1.36%	16	1.73%	8	1.73%	4	1.73%	Disabled		Disabled	

Remark k: Value set by bits 7 to 0 of the baud rate generator control register n (BRGCAn) (k = 2, 3, 4, ..., 255)
n: Unit number (n = 0, 1)

Table 17 - 4 Set Data of Baud Rate Generator (4/4)

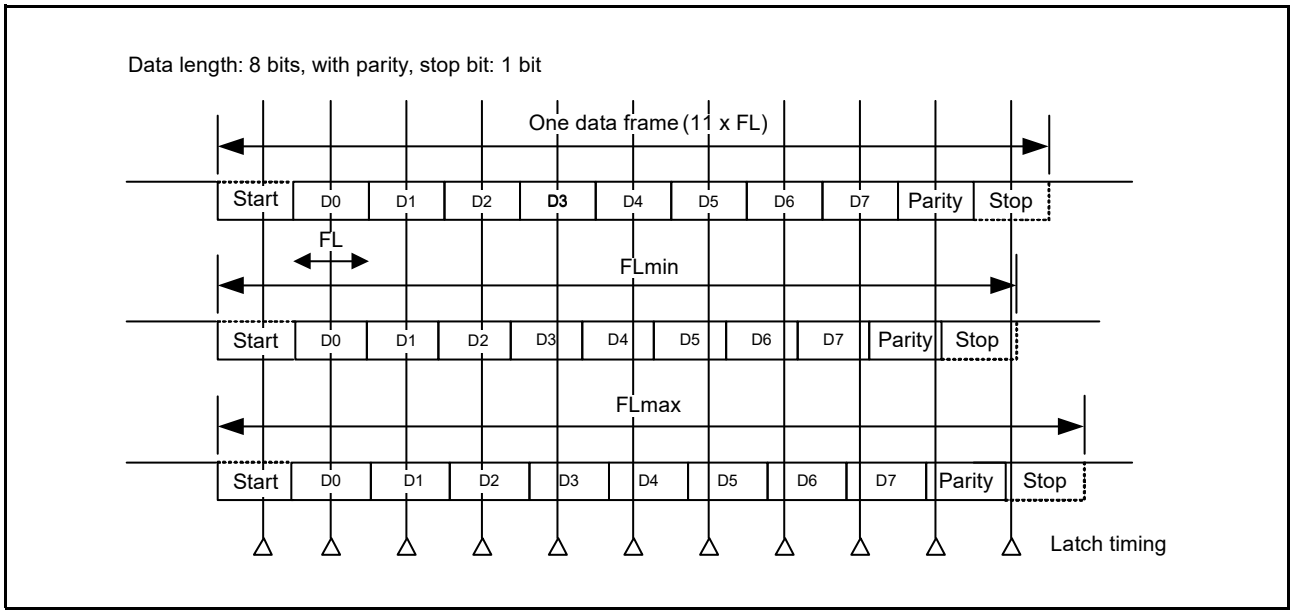
Desired baud rate	In operation with f_{SXP} = 32.768 kHz (UTAnCK3 to UTAnCK0 = 1000B)		In operation with ELCL = 32 MHz (UTAnCK3 to UTAnCK0 = 1001B)	
	k	Error from the desired baud rate	k	Error from the desired baud rate
200 bps	82	-0.10%	Disabled	
300 bps	55	-0.70%	Disabled	
600 bps	27	-1.14%	Disabled	
1200 bps	14	-2.48%	Disabled	
2400 bps	7	-2.48%	Disabled	
4800 bps	Disabled		Disabled	
9600 bps	Disabled		Disabled	
19200 bps	Disabled		Disabled	
38400 bps	Disabled		Disabled	
76800 bps	Disabled		208	0.16%
115200 bps	Disabled		139	-0.08%
153600 bps	Disabled		104	0.16%

Remark k: Value set by bits 7 to 0 of the baud rate generator control register n (BRGCAn) (k = 2, 3, 4, ..., 255)
n: Unit number (n = 0, 1)

d) Permissible baud rate range during reception

Figure 17 - 25 shows the permissible error from the baud rate on the transmitting side during reception.

Figure 17 - 25 Permissible Baud Rate Range during Reception



Caution Be sure to make settings so that the baud rate error during reception is within the permissible error range. Use the calculation expression below to check if the error is within the permissible range.

After the start bit is detected, the latch timing of receive data is determined by the counter specified with the baud rate generator control register n (BRGCAn). If the whole frame including the stop bit has been received before this latching, reception can proceed correctly. Assuming that 11 bits of data are received, the theoretical values can be calculated as follows.

- The relation between 1-bit data length and baud rate
  FL = (Brate) - 1
  Brate: Baud rate of UART
  k: Set value of the BRGCAn register
  FL: 1-bit data length
  Margin of latch timing: 2 clock cycles

- Minimum permissible data frame length (FLmin)

$$k = 3 \text{ to } 255: FLmin = 11 \times FL - \frac{k - 2}{2k} \times FL = \frac{21k + 2}{2k} FL$$

- Maximum permissible baud rate for reception on the transmitting side (BRmax)

$$k = 2: BRmax = Brate + \frac{1}{22k} Brate$$

$$k = 3 \text{ to } 255: BRmax = (FLmin/11)^{-1} = \frac{22k}{21k + 2} Brate$$

- Maximum permissible data frame length (FLmax)

$$k = 3 \text{ to } 255: FLmax = \frac{21k + 2}{20k} FL \times 11$$

- Minimum permissible baud rate for reception on the transmitting side (BRmin)

$$k = 2: BRmin = Brate - \frac{1}{22k} Brate$$

$$k = 3 \text{ to } 255: BRmin = (FLmax/11)^{-1} = \frac{20k}{21k - 2} Brate$$

Table 17 - 5 shows the permissible baud rate errors between the UART and the transmitting side that can be calculated from the above expressions for the minimum and maximum baud rates.

Table 17 - 5 Maximum/Minimum Permissible Baud Rate Error

Division ratio (k)	Maximum permissible baud rate error	Minimum permissible baud rate error
2	+2.27%	–2.27%
4	+2.33%	–2.44%
8	+3.53%	–3.61%
20	+4.27%	–4.31%
50	+4.56%	–4.58%
100	+4.66%	–4.67%
255	+4.72%	–4.73%

Remark 1. The permissible error of reception depends on the number of bits in one frame, input clock frequency, and division ratio (k). The higher the input clock frequency and the division ratio (k), the higher the permissible error.

Remark 2. k: Set value of the BRGCAn register
n: Unit number (n = 0, 1)

17.4 Points for Caution when the Serial Interface UARTA is to be Used

17.4.1 Port setting for the RxDAn pin

When ALVn = 0 (wait state = high level, start bit = low level), the initial value of receive data (RxDAn) must be high.
When ALVn = 1 (wait state = low level, start bit = high level), the initial value of receive data (RxDAn) must be low.
Accordingly, port setting is required for the RxDAn pin before setting the UARTAENn bit to 1.

17.4.2 Serial interfaces

The serial data output (TxDA0) and serial clock output (CLKA0) of the serial interface UARTA0 are selectable as inputs for the logic and event link controller (ELCL). If TxDA0 or CLKA0 is to be used as an input for the ELCL, you need to clear the corresponding bit in port function output enable register 1 (PFOE1) to 0 so that the signal is not output to the TxDA0 and CLKA0 pins. For details, see **4.3.15 Port function output enable registers (PFOEx)**.

17.4.3 Point for caution when selecting the serial interface UARTAn operation clock (f_{UTAn})

When the middle-speed on-chip oscillator peripheral clock (f_{IMP}) is selected for f_{UTAn}, communication may not be executed correctly due to the oscillation frequency accuracy of the middle-speed on-chip oscillator. Adjust the accuracy, therefore, by using the middle-speed on-chip oscillator trimming register (MIOTRM). When the low-speed peripheral clock (f_{SXP}) is selected for f_{UTAn} and the low-speed on-chip oscillator peripheral clock (f_{IL}) is selected for f_{SXP}, communication may not be executed correctly due to the oscillation frequency accuracy of the low-speed on-chip oscillator. Adjust the accuracy, therefore, by using the low-speed on-chip oscillator trimming register (LIOTRM).

Remark n: Unit number (n = 0, 1)

Section 18 Remote Control Signal Receiver (REMC)

The remote control signal receiver is provided in the 32- to 128-pin R7F100G and 32- to 48-pin R7F104G products.

18.1 Remote Control Signal Receiver

The remote control signal receiver can receive data by checking the width and period of an external pulse input signal. **Table 18 - 1** lists the specifications of the remote control signal receiver. **Figure 18 - 1** shows a block diagram of the remote control signal receiver.

Table 18 - 1 Specifications of Remote Control Signal Receiver

Item	Description
Number of provided units	1
External pulse input	RIN0
Operating clock	Low-speed peripheral clock (fSXP) Timer interrupt (INTTM06)
Detection patterns	Header pattern Data 0 pattern Data 1 pattern Special data pattern
Receive buffer	8 bytes (64 bits)
Compare bit count	1 to 16 bits
Interrupt request signal	INTREMC
Interrupt sources	Header pattern match**Note 1** Compare match**Note 1** Completion of data reception**Note 1** Special data pattern match**Note 1** Data 0 pattern or data 1 pattern match Receive buffer full Receive error
Selectable functions	Input signal inversion Digital filter (matching three or two times)**Note 2** Pattern end setting
Low power consumption	The clock supply can be stopped by setting the PER1.REMCEN bit. SNOOZE mode can shift to normal operation mode by using the REMC interrupt request signal.

Note 1. The interrupt mode is selectable from either the normal interrupt mode or the sequential interrupt mode. In the normal interrupt mode, the OR condition of multiple interrupt sources is applied. In the sequential interrupt mode, the AND condition of multiple interrupt sources is applied.

Note 2. The sampling clock of the digital filter is an operating clock selected by the REMCON1.CSRC bit, or fSXP. When transition to the SNOOZE mode is enabled, select the low-speed peripheral clock (fSXP) as the sampling clock of the digital filter.

Figure 18 - 1 Block Diagram of Remote Control Signal Receiver

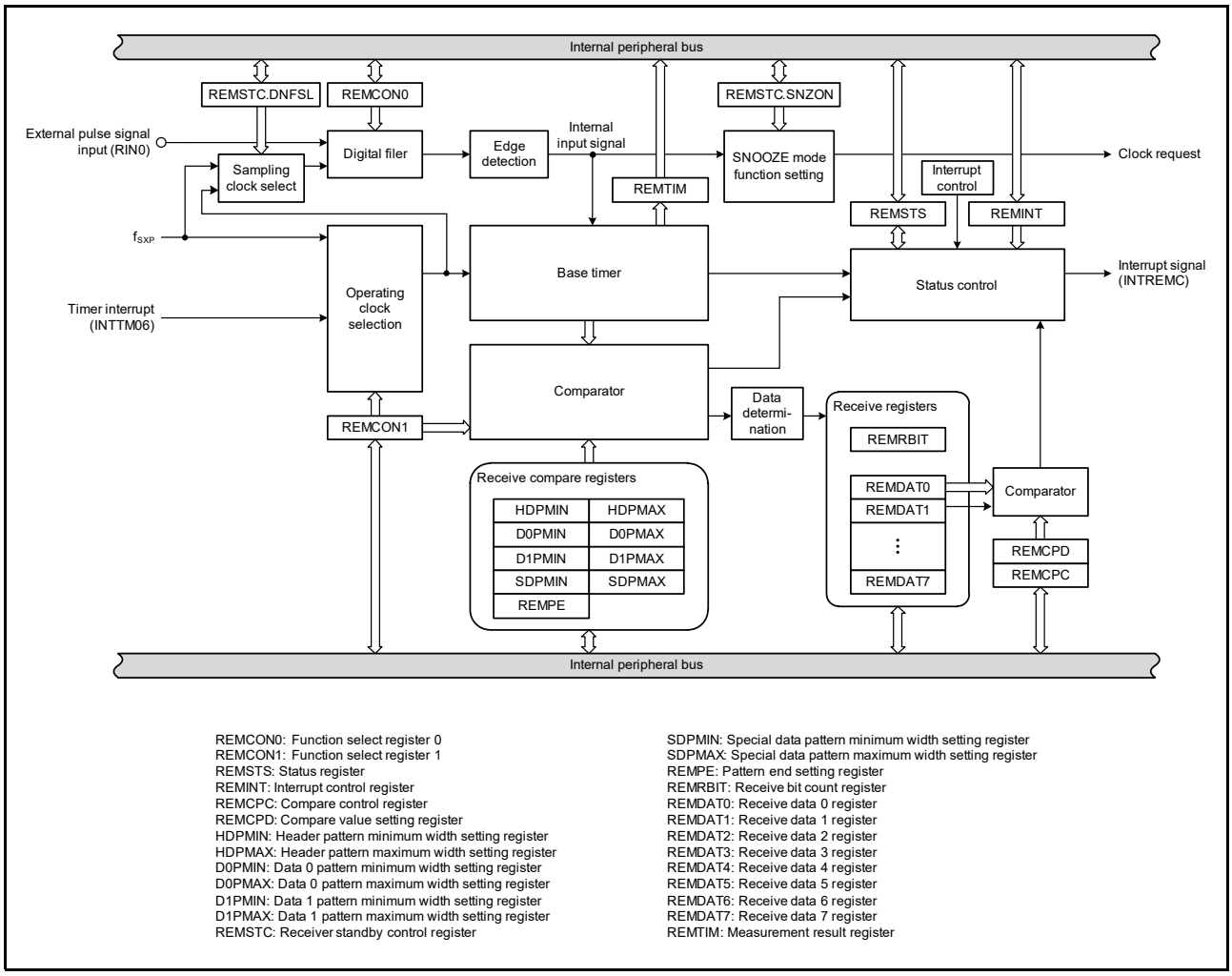


Table 18 - 2 shows the input pin used for the remote control signal receiver.

Table 18 - 2 Pin Configuration of Remote Control Signal Receiver

Pin Name	I/O	Function
RIN0	Input	External pulse signal input

18.2 Registers for Controlling the Remote Control Signal Receiver

The following registers are used to control the remote control signal receiver.

- Peripheral enable register 1 (PER1)
- Peripheral reset control register 1 (PRR1)
- Function select register 0 (REMCON0)
- Function select register 1 (REMCON1)
- Status register (REMSTS)
- Interrupt control register (REMINT)
- Compare control register (REMCPC)
- Compare value setting register (REMCPD)
- Header pattern minimum width setting register (HDPMIN)
- Header pattern maximum width setting register (HDPMAX)
- Data 0 pattern minimum width setting register (D0PMIN)
- Data 0 pattern maximum width setting register (D0PMAX)
- Data 1 pattern minimum width setting register (D1PMIN)
- Data 1 pattern maximum width setting register (D1PMAX)
- Special data pattern minimum width setting register (SDPMIN)
- Special data pattern maximum width setting register (SDPMAX)
- Pattern end setting register (REMPE)
- Receiver standby control register (REMSTC)
- Receive bit count register (REMRBIT)
- Receive data 0 register (REMDAT0)
- Receive data j register (REMDATj)
- Measurement result register (REMTIM)
- Port mode registers (PMxx)
- Port mode control T registers (PMCTxx)

18.2.1 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. To use the remote control signal receiver, be sure to set the REMCEN bit to 1. The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 18 - 2 Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER1	DACEN	SMSEN	CMPEN	TML32EN	DTCEN	UTAEN **Note 1**	REMCEN **Note 2**	CTSUEN

REMCEN	Control of supply of an input clock to the remote control signal receiver
0	Stops supply of an input clock. • The SFRs used by the remote control signal receiver cannot be written.**Note 3**
1	Enables supply of an input clock. • The SFRs used by the remote control signal receiver can be read and written.

Note 1. This bit is only present in the 36- to 128-pin R7F100G and 32- to 48-pin R7F104G products.
Note 2. This bit is only present in the 32- to 128-pin R7F100G and 32- to 48-pin R7F104G products.
Note 3. When an SFR is read, 00H or 0000H is returned regardless of its setting.

Caution Be sure to leave the following bits set to 0.
Bits 2 and 1 in the 30-pin R7F100G products
Bit 2 in the 32-pin R7F100G products

18.2.2 Peripheral reset control register 1 (PRR1)

The PRR1 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. Set the REMCRES bit to 1 to reset the remote control signal receiver. The PRR1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 18 - 3 Format of Peripheral Reset Control Register 1 (PRR1)

Address: F00FBH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	3	2	<1>	<0>
PRR1	DACRES	SMSRES	CMPRES	TML32RES	0	0	REMCRES **Note**	CTSURES

REMCRES	Control resetting of the remote control signal receiver
0	The remote control signal receiver is released from the reset state.
1	The remote control signal receiver is in the reset state. • The SFRs for use with the remote control signal receiver are initialized.

Note This bit is only present in the 32- to 128-pin R7F100G and 32- to 48-pin R7F104G products.

Caution 1. Be sure to leave bits 3 and 2 set to 0.
Caution 2. Be sure to leave bit 1 in the 30-pin R7F100G products set to 0.

18.2.3 Function select register 0 (REMCON0)

The REMCON0 register has flags for indicating the state of the operation of the remote control signal receiver and the level of the input signal, and is used to control the input signal inversion, control the digital filter, and select the receive error capture operation. Rewriting the REMCON0 register is prohibited when the REMC is in operation (when REMCON1.EN = 1). The REMCON0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 18 - 4 Format of Function Select Register 0 (REMCON0) (1/2)

Address: F0640H
After reset: 00H
R/W: R/W**Note 1**

Symbol	7	6	5	4	<3>	2	1	<0>
REMCON0	0	FILSEL **Note 2**	0	EC **Note 2**	INFLG **Note 3**	FIL **Note 2**	INV **Note 2**	ENFLG **Note 3**

FILSEL **Note 2**	Digital Filter Function Select
0	Digital filter for matching three times
1	Digital filter for matching two times

EC**Note 2**	Receive Error Capture Operation Select
0	Captures the data after an error pattern is received.
1	Does not capture the data after an error pattern is received.
This bit can be used to set capture operation to the REMRBIT and REMDATj registers (j = 0 to 7) after an error pattern is received.	

INFLG **Note 3**	Input Signal Flag
0	The level of the internal input signal of the remote control signal receiver is low.
1	The level of the internal input signal of the remote control signal receiver is high.
This flag can be used to confirm the level of the internal input signal of the remote control signal receiver. The confirmed level is the result set by the INV and FIL bits.	

FIL**Note 2**	Digital Filter Enable/Disable Setting
0	Disables the digital filter for matching three or two times.
1	Enables the digital filter for matching three or two times.
This bit enables or disables the digital filter.	

INV**Note 2**	Input Signal Inversion
0	Not inverted
1	Inverted

Figure 18 - 4 Format of Function Select Register 0 (REMCON0) (2/2)

ENFLG **Note 3**	Remote Control Status Flag
0	Stopped
1	Operating
This flag can be used to confirm whether the remote control signal receiver is stopped or operating. This flag changes after zero to one cycle of the operating clock when a value is written to the REMCON1.EN bit.	

Note 1. Bits 3 and 0 are read-only.

Note 2. These bits can be rewritten when the REMCON1.EN bit and the ENFLG flag are both 0 (REMC operation is stopped).

Note 3. These flags become 0 when the REMCON1.EN bit is set to 0.

18.2.4 Function select register 1 (REMCON1)

The REMCON1 register is used to set the operation conditions of the remote control signal receiver such as operating clock, reception mode, and interrupt mode. The REMCON1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 18 - 5 Format of Function Select Register 1 (REMCON1)

Address: F0641H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	<3>	<2>	1	0
REMCON1	0	0	CSRC **Note 1**	0	INTMD **Note 1**	EN	TYP[1:0] **Note 2**	

CSRC**Note 1**	Operating Clock Select
0	Low-speed peripheral clock (f_{SXP})
1	Timer interrupt (INTTM06) Use channel 6 of timer array unit 0 in the interval timer mode.
This bit selects the operating clock for the remote control signal receiver.	

INTMD **Note 1**	Interrupt Mode Select
0	Normal interrupt mode (OR condition)
1	Sequential interrupt mode (AND condition)
This bit can be used to select the interrupt mode. In normal interrupt mode, an interrupt is generated when the OR condition of the interrupt sources enabled (set to 1) in the interrupt control register (REMINT) is met. In sequential interrupt mode, if all the interrupt generation conditions of the sources (compare match, data reception complete, header pattern match, and special data pattern match) whose interrupt enable bit of the REMINT register is set to 1 are met, an interrupt (INTREMC) is generated.	

EN	Remote Control
0	Operation disabled
1	Operation enabled
This bit enables or disables REMC operation. Check the REMCON0.ENFLG flag to determine whether the REMC is operating or not.	

TYP[1:0]**Note 2**		Receive Mode Select
0	0	Format A
0	1	Format B
1	0	Format C
1	1	Setting prohibited
These bits can be used to select the format for capturing the remote control signal waveform. For details of each format, see **18.3.3 Pattern setting**.		

Note 1. These bits can be rewritten when the EN bit and the REMCON0.ENFLG flag are both 0 (REMC is stopped).

Note 2. To rewrite the TYP[1:0] bits when the EN bit or REMCON0.ENFLG flag is 1 (REMC is operating), change the values of these bits one bit at a time.

18.2.5 Status register (REMSTS)

The REMSTS register is used to indicate the state of reception and of errors occurring for the remote control signal receiver. The REMSTS register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 18 - 6 Format of Status Register (REMSTS) (1/3)

Address: F0642H
After reset: 00H
R/W: R/W**Notes 1, 2**

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
REMSTS	SDFLG	D1FLG	D0FLG	HDFLG	BFULFLG **Note 2**	DRFLG	REFLG	CPFLG

SDFLG	Special Data Pattern Match Flag
0	Mismatch
1	Match
<Clearing conditions> • When the DRFLG flag changes from 0 to 1 (next frame reception starts) • When the REFLG flag changes from 0 to 1 • See **Table 18 - 3 Measurement Results and Flags**. <Setting condition> • See **Table 18 - 3 Measurement Results and Flags**.	

D1FLG	Data 1 Pattern Match Flag
0	Mismatch
1	Match
<Clearing conditions> • When the DRFLG flag changes from 0 to 1 (next frame reception starts) • When the REFLG flag changes from 0 to 1 • See **Table 18 - 3 Measurement Results and Flags**. <Setting condition> • See **Table 18 - 3 Measurement Results and Flags**.	

D0FLG	Data 0 Pattern Match Flag
0	Mismatch
1	Match
<Clearing conditions> • When the DRFLG flag changes from 0 to 1 (next frame reception starts) • When the REFLG flag changes from 0 to 1 • See **Table 18 - 3 Measurement Results and Flags**. <Setting condition> • See **Table 18 - 3 Measurement Results and Flags**.	

Figure 18 - 6 Format of Status Register (REMSTS) (2/3)

HDFLG	Header Pattern Match Flag
0	Mismatch
1	Match
<Clearing conditions> • When the DRFLG flag changes from 0 to 1 (next frame reception starts) • When the REFLG flag changes from 0 to 1 • See **Table 18 - 3 Measurement Results and Flags**. <Setting condition> • See **Table 18 - 3 Measurement Results and Flags**.	

BFULFLG **Note 2**	Receive Buffer Full Flag
0	Receive buffer is empty.
1	Receive buffer is full (64 bits received).
<Clearing conditions> • When the HDFLG flag changes from 0 to 1 • When the DRFLG flag changes from 0 to 1 (next frame reception starts) • This flag becomes 0 after one to two cycles of the operating clock when 0 is written to the BFULFLG flag. <Setting condition> • When the value of the REMRBIT register becomes 64	

DRFLG	Data Receiving Flag
0	Waiting for data reception.
1	Data is being received.
This flag indicates the state of receiving the remote control signal. <Clearing conditions> • This flag becomes 0 after one cycle of the operating clock when the value of the base timer is greater than any value of the HDPMAX, D0PMAX, D1PMAX, SDPMAX, and REMPE registers. <Setting condition> • Rising edge of REMC internal input signal (when the REMCON0.INV bit is 0)	

Figure 18 - 6 Format of Status Register (REMSTS) (3/3)

REFLG	Receive Error Flag
0	No error has occurred.
1	An error has occurred.

This flag indicates that a receive error has occurred. The setting conditions differ depending on the value of the REMCON1.TYP[1:0] bits.
<Clearing conditions>
- The header pattern is detected
- When the DRFLG flag changes from 0 to 1 (next frame reception starts)

<Setting conditions>
When the REMCON1.TYP[1:0] bits are 00B (format A):
- The data 0, data 1, or special data pattern is detected prior to receiving the header pattern
- The width between a rising edge and the next rising edge of the input signal is not the header, data 0, data 1, or special data pattern (when the REMCON0.INV bit is 0)
- A conflict occurs between when data reception is completed (timing when the DRFLG flag changes from 1 to 0) and when the input signal level changes (new signal).

When the REMCON1.TYP[1:0] bits are 01B (format B):
- The data 0, data 1, or special data pattern is detected prior to receiving the header pattern
- The width between a falling edge and the next falling edge of the input signal is not the data 0, data 1, or special data pattern (when the REMCON0.INV bit is 0)
- A conflict occurs between when data reception is completed (timing when the DRFLG flag changes from 1 to 0) and when the input signal level changes (new signal).

When the REMCON1.TYP[1:0] bits are 10B (format C):
- The width between a rising edge and the next rising edge of the input signal is not the header, data 0, data 1, or special data pattern (when the REMCON0.INV bit is 0)
- A conflict occurs between when data reception is completed (timing when the DRFLG flag changes from 1 to 0) and when the input signal level changes (new signal).

CPFLG	Compare Match Flag
0	Mismatch
1	Match

This flag indicates the comparison result between the value of the REMCPD register specified by the REMCPC.CPN[3:0] bits and the data to be stored in the REMDAT1 and REMDAT0 registers.
<Clearing conditions>
- When the DRFLG flag changes from 0 to 1 (next frame reception starts)
- When the HDFLG flag changes from 0 to 1

<Setting condition>
- When the value of the REMCPD register matches the value to be stored in the REMDAT1 and REMDAT0 registers (when the setting value of the REMCPC.CPN[3:0] bits is n, bits n to 0 in the REMCPD register match bits n to 0 in the REMDAT1 and REMDAT0 registers)

Note 1. Bits 7 to 4 and 2 to 0 are read-only.

Note 2. Bit 3 can only be written with 0 to clear the flag. However, if this flag is written when changing the REMCON0.INFLG flag, the value read from this flag may become undefined.

Caution 1. If updating and reading data overlap, an undefined value may be read. For details on reading this register, see 18.4.5 Reading registers.

Caution 2. The value of this register becomes 00H when the REMCON1.EN bit is set to 0.

Table 18 - 3 Measurement Results and Flags

Comparison Result between REMTIM Register Value (Measurement Result) and Each Register	Flag Value			
	HDFLG	D0FLG	D1FLG	SDFLG
Between HDPMIN and HDPMAX	1	0	0	0
Between D0PMIN and D0PMAX	0	1Note	0	0
Between D1PMIN and D1PMAX	0	0	1Note	0
Between SDPMIN and SDPMAX	0	0	0	1Note
Values not listed above	0	0	0	0

Note When the REMCON1.TYP[1:0] bits are 00B or 01B, the D0FLG, D1FLG, and SDFLG flags remain unchanged until the header pattern is detected.

18.2.6 Interrupt control register (REMINT)

The REMINT register is used to enable or disable generation of each interrupt. Rewriting the REMINT register is prohibited when the REMC is in operation. Rewrite this register while the REMCON1.EN bit and the REMCON0.ENFLG flag are both 0 (REMC operation is stopped). The REMINT register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 18 - 7 Format of Interrupt Control Register (REMINT)

Address: F0643H
After reset: 00H
R/W: R/W

Symbol	<7>	6	<5>	<4>	<3>	<2>	<1>	<0>
REMINT	SDINT	0	DINT **Note**	HDINT	BFULINT	DRINT **Note**	REINT	CPINT

SDINT	Special Data Pattern Match Interrupt Enable
0	Disabled
1	Enabled

DINT	Data 0 Pattern or Data 1 Pattern Match Interrupt Enable
0	Disabled
1	Enabled

HDINT	Header Pattern Match Interrupt Enable
0	Disabled
1	Enabled

BFULINT	Receive Buffer Full Interrupt Enable
0	Disabled
1	Enabled

DRINT**Note**	Data Receiving Interrupt Enable
0	Disabled
1	Enabled

REINT	Receive Error Interrupt Enable
0	Disabled
1	Enabled

CPINT	Compare Match Interrupt Enable
0	Disabled
1	Enabled

Note Rewriting this bit is also allowed when REMC is operating (when REMCON1.EN = 1).

18.2.7 Compare control register (REMCPC)

The REMCPC register is used to specify the number of bits to compare the value of the REMCPD register and the data to be stored in the REMDAT1 and REMDAT0 registers when the compare function is used. Rewriting the REMCPC register is prohibited when the REMC is in operation. This register can be rewritten when the REMCON1.EN bit and the REMCON0.ENFLG flag are both 0 (REMC operation is stopped). The REMCPC register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 18 - 8 Format of Compare Control Register (REMCPC)

Address: F0645H
After reset: 00H
R/W: R/W

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

CPN[3:0]				Compare Bit Count Specification
0	0	0	0	Bit 0 in the REMCPD register is compared with bit 0 in the REMDAT0 register.
0	0	0	1	Bits 1 and 0 in the REMCPD register are compared with bits 1 and 0 in the REMDAT0 register.
•				
•				
0	1	1	1	Bits 7 to 0 in the REMCPD register are compared with bits 7 to 0 in the REMDAT0 register.
•				
•				
1	0	0	1	Bits 9 to 0 in the REMCPD register are compared with bits 1 and 0 in the REMDAT1 register and bits 7 to 0 in the REMDAT0 register.
•				
•				
1	1	1	1	Bits 15 to 0 in the REMCPD register are compared with bits 7 to 0 in the REMDAT1 register and bits 7 to 0 in the REMDAT0 register.
When the setting value of these bits is n, bits n to 0 are compared.				

18.2.8 Compare value setting register (REMCPD)

The REMCPD register is used to set the value to be compared with the data in the REMDAT1 and REMDAT0 registers when the compare function is used. The REMCPC.CPN[3:0] bits can be used to set the number of bits to be compared. Rewriting the REMCPD register is prohibited when the REMC is in operation (when REMCON1.EN = 1). The REMCPD register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 18 - 9 Format of Compare Value Setting Register (REMCPD)

Address: F0646H, F0647H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
REMCPD	CPD[15:8]							

7	6	5	4	3	2	1	0
CPD[7:0]							

18.2.9 Header pattern minimum width setting register (HDPMIN)

The HDPMIN register is used to set the minimum width of the header pattern. The setting range is from 000H to 7FFH. When the interval between the edges of an external input signal measured using the base timer is between HDPMIN and HDPMAX, the pattern of the signal is detected as a header pattern. Rewriting the HDPMIN register is prohibited when the REMC is in operation (when REMCON1.EN = 1). The HDPMIN register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 18 - 10 Format of Header Pattern Minimum Width Setting Register (HDPMIN)

Address: F0648H, F0649H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
HDPMIN	0	0	0	0	0	HDPMIN[10:8]		

7	6	5	4	3	2	1	0
HDPMIN[7:0]							

18.2.10 Header pattern maximum width setting register (HDPMAX)

The HDPMAX register is used to set the maximum width of the header pattern. The setting range is from 000H to 7FFH. When the interval between the edges of an external input signal measured using the base timer is between HDPMIN and HDPMAX, the pattern of the signal is detected as a header pattern. Rewriting the HDPMAX register is prohibited when the REMC is in operation (when REMCON1.EN = 1). The HDPMAX register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 18 - 11 Format of Header Pattern Maximum Width Setting Register (HDPMAX)

Address: F064AH, F064BH
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
HDPMAX	0	0	0	0	0	HDPMAX[10:8]		

7	6	5	4	3	2	1	0
HDPMAX[7:0]							

18.2.11 Data 0 pattern minimum width setting register (D0PMIN)

The D0PMIN register is used to set the minimum width of the data 0 pattern. The setting range is from 00H to FFH. When the interval between the edges of an external input signal measured using the base timer is between D0PMIN and D0PMAX, the pattern of the signal is detected as a data 0 pattern. Rewriting the D0PMIN register is prohibited when the REMC is in operation (when REMCON1.EN = 1). The D0PMIN register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 18 - 12 Format of Data 0 Pattern Minimum Width Setting Register (D0PMIN)

Address: F064CH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
D0PMIN	D0PMIN[7:0]							

18.2.12 Data 0 pattern maximum width setting register (D0PMAX)

The D0PMAX register is used to set the maximum width of the data 0 pattern. The setting range is from 00H to FFH. When the interval between the edges of an external input signal measured using the base timer is between D0PMIN and D0PMAX, the pattern of the signal is detected as a data 0 pattern. Rewriting the D0PMAX register is prohibited when the REMC is in operation (when REMCON1.EN = 1). The D0PMAX register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 18 - 13 Format of Data 0 Pattern Maximum Width Setting Register (D0PMAX)

Address: F064DH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
D0PMAX	D0PMAX[7:0]							

18.2.13 Data 1 pattern minimum width setting register (D1PMIN)

The D1PMIN register is used to set the minimum width of the data 1 pattern. The setting range is from 00H to FFH. When the interval between the edges of an external input signal measured using the base timer is between D1PMIN and D1PMAX, the pattern of the signal is detected as a data 1 pattern. Rewriting the D1PMIN register is prohibited when the REMC is in operation (when REMCON1.EN = 1). The D1PMIN register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 18 - 14 Format of Data 1 Pattern Minimum Width Setting Register (D1PMIN)

Address: F064EH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
D1PMIN	D1PMIN[7:0]							

18.2.14 Data 1 pattern maximum width setting register (D1PMAX)

The D1PMAX register is used to set the maximum width of the data 1 pattern. The setting range is from 00H to FFH. When the interval between the edges of an external input signal measured using the base timer is between D1PMIN and D1PMAX, the pattern of the signal is detected as a data 1 pattern. Rewriting the D1PMAX register is prohibited when the REMC is in operation (when REMCON1.EN = 1). The D1PMAX register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 18 - 15 Format of Data 1 Pattern Maximum Width Setting Register (D1PMAX)

Address: F064FH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
D1PMAX	D1PMAX[7:0]							

18.2.15 Special data pattern minimum width setting register (SDPMIN)

The SDPMIN register is used to set the minimum width of the special data pattern. The setting range is from 000H to 7FFH. When the interval between the edges of an external input signal measured using the base timer is between SDPMIN and SDPMAX, the pattern of the signal is detected as a special data pattern. Rewriting the SDPMIN register is prohibited when the REMC is in operation (when REMCON1.EN = 1). The SDPMIN register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 18 - 16 Format of Special Data Pattern Minimum Width Setting Register (SDPMIN)

Address: F0650H, F0651H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SDPMIN	0	0	0	0	0	SDPMIN[10:8]		

	7	6	5	4	3	2	1	0
	SDPMIN[7:0]							

18.2.16 Special data pattern maximum width setting register (SDPMAX)

The SDPMAX register is used to set the maximum width of the special data pattern. The setting range is from 000H to 7FFH. When the interval between the edges of an external input signal measured using the base timer is between SDPMIN and SDPMAX, the pattern of the signal is detected as a special data pattern. Rewriting the SDPMAX register is prohibited when the REMC is in operation (when REMCON1.EN = 1). The SDPMAX register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 18 - 17 Format of Special Data Pattern Maximum Width Setting Register (SDPMAX)

Address: F0652H, F0653H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SDPMAX	0	0	0	0	0	SDPMAX[10:8]		

7	6	5	4	3	2	1	0
SDPMAX[7:0]							

18.2.17 Pattern end setting register (REMPE)

The REMPE register is used to set the width of the pattern end. The setting range is from 000H to 7FFH. The timing when the REMSTS.DRFLG flag changes from 1 to 0 can be set. Rewriting the REMPE register is prohibited when the REMC is in operation (when REMCON1.EN = 1). The REMPE register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 18 - 18 Format of Pattern End Setting Register (REMPE)

Address: F0654H, F0655H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
REMPE	0	0	0	0	0	PE[10:8]		

7	6	5	4	3	2	1	0
PE[7:0]							

18.2.18 Receiver standby control register (REMSTC)

The REMSTC register is used to control the startup of reception (SNOOZE mode) upon receiving the remote control signal in the STOP mode and select the sampling clock of the digital filter. Rewriting the REMSTC register is prohibited when the REMC is in operation (when REMCON1.EN = 1). The REMSTC register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 18 - 19 Format of Receiver Standby Control Register (REMSTC)

Address: F0656H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	<0>
REMSTC	0	0	0	0	0	0	DNFSL **Note**	SNZON **Note**

DNFSL **Note**	Digital Filter Clock Selection
0	REMC operating clock is selected as a sampling clock
1	f_{SXP} is selected as a sampling clock
This bit is used to select the sampling clock of the digital filter. Set this bit to 1 when setting the SNZON bit to 1 (transition from STOP mode to SNOOZE mode is enabled).	

SNZON **Note**	SNOOZE Mode Operation Control
0	Disables transition from STOP mode to SNOOZE mode.
1	Enables transition from STOP mode to SNOOZE mode.
• When the input level of the RIN0 pin is changed in the STOP mode, the STOP mode is exited, and remote control signal reception is performed without operating the CPU (the SNOOZE mode). • Use the SNOOZE mode only when the timer interrupt (INTTM06) is selected as the REMC operating clock. In this case, select a clock that can operate in SNOOZE mode as the count clock for channel 6 of timer array unit 0. • When using SNOOZE mode, set channel 6 of timer array unit 0 to interval timer mode and enable the count operation before shifting to STOP mode. • Even when SNOOZE mode is to be used, set this bit to 0 after return from SNOOZE mode. Set the SNZON bit to 1 when the transition to STOP mode is to be repeated. • When setting this bit to 1, set the REMCON0.FIL bit to 1 (enables the digital filter) and set the DNFSL bit to 1 (f_{SXP} is selected as a sampling clock). • After returning from STOP mode due to a compare match interrupt or header pattern match interrupt, set this bit to 0.	

Note Set this bit to 1 when the REMCON1.EN bit and the REMCON0.ENFLG flag are both 0 (REMC operation is stopped).

18.2.19 Receive bit count register (REMRBIT)

The REMRBIT register indicates the number of received bits. All the bits of the REMRBIT register are initialized when the setting of the REMCON1.EN bit is 0. The REMRBIT register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 18 - 20 Format of Receive Bit Count Register (REMRBIT)

Address: F0657H
After reset: 00H
R/W: R/W**Notes 1, 2**

Symbol	7	6	5	4	3	2	1	<0>
REMRBIT	0 **Note 1**	RBIT[6:1] **Note 2**						RBIT[0]

RBIT[6:0]	Receive Bit Count Check
These bits indicate the bit position of the buffer to be stored by counting the detected data 0 pattern or data 1 pattern. • When the receive bit count exceeds 64 (40H), the value returns to 1. • The header pattern and special data pattern are not counted. • If an error is detected while the REMCON0.EC bit is 1, the value is not incremented even when the data 0 pattern or data 1 pattern is detected. • The value of this register becomes 0 when the REMSTS.DRFLG flag changes from 0 to 1. • The value of this register becomes 0 when the REMSTS.HDFLG flag changes from 0 to 1. When 0 is written to the REMRBIT.RBIT[0] bit, the value of the REMRBIT register becomes 00H after one to two cycles of the operating clock.	

Note 1. Bit 7 is read as undefined. The write value should be 0.
Note 2. Bits 6 to 1 are read-only.

Caution **If updating and reading data overlap, an undefined value may be read. For details on reading this register, see 18.4.5 Reading registers.**

18.2.20 Receive data 0 register (REMDAT0)

The REMDAT0 register holds received data. All the bits of the REMDAT0 register are initialized when the setting of the REMCON1.EN bit is 0. The REMDAT0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 18 - 21 Format of Receive Data 0 Register (REMDAT0)

Address: F0658H
After reset: 00H
R/W: R/W**Note**

Symbol	7	6	5	4	3	2	1	<0>
REMDAT0	DAT0[7:1] **Note**							DAT0[0]

DAT0[7:0]	Receive Data 0 Store
When data 0 pattern or data 1 pattern is detected, the result is sequentially stored bit by bit starting from the DAT0[0] bit as received data. For details on storing received data, see **18.3.8 Receive data buffer**. When 0 is written to the DAT0[0] bit, the value of this register becomes 00H after one to two cycles of the operating clock.	

Note Bits 7 to 1 are read-only.

Caution **If updating and reading data overlap, an undefined value may be read. For details on reading this register, see 18.4.5 Reading registers.**

18.2.21 Receive data j register (REMDATj)

The REMDATj register holds received data. All the bits of the REMDATj register are initialized when the setting of the REMCON1.EN bit is 0. The REMDATj register can be read by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 18 - 22 Format of Receive Data j Register (REMDATj)

Address: F0659H to F065FH
After reset: 00H
R/W: R

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

DATj[7:0]	Receive Data j Store
When data 0 pattern or data 1 pattern is detected, the result is sequentially stored bit by bit starting from the DATj[0] bit as received data. For details on storing received data, see **18.3.8 Receive data buffer**.	

Caution **If updating and reading data overlap, an undefined value may be read. For details on reading this register, see 18.4.5 Reading registers.**

Remark j = 1 to 7

18.2.22 Measurement result register (REMTIM)

The REMTIM register indicates the measurement result of each pattern width. All the bits of the REMTIM register are initialized when the setting of the REMCON1.EN bit is 0. The REMTIM register can be read by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 18 - 23 Format of Measurement Result Register (REMTIM)

Address: F0660H, F0661H
After reset: 0000H
R/W: R

Symbol	15	14	13	12	11	10	9	8
REMTIM	0	0	0	0	0	TIM[10:8]		

7	6	5	4	3	2	1	0
TIM[7:0]							

TIM[10:0]	Measurement Result
The value of the base timer is captured when any of the following patterns is detected. • Header pattern • Data 0 pattern • Data 1 pattern • Special data pattern • Data pattern other than the above (receive error)	

Caution If updating and reading data overlap, an undefined value may be read. For details on reading this register, see 18.4.5 Reading registers.

18.2.23 Registers for controlling the port functions multiplexed with the remote control signal input

Set the following registers to control the port functions multiplexed with the remote control signal input.

- Port mode registers (PMxx)
- Port mode control T registers (PMCTxx)

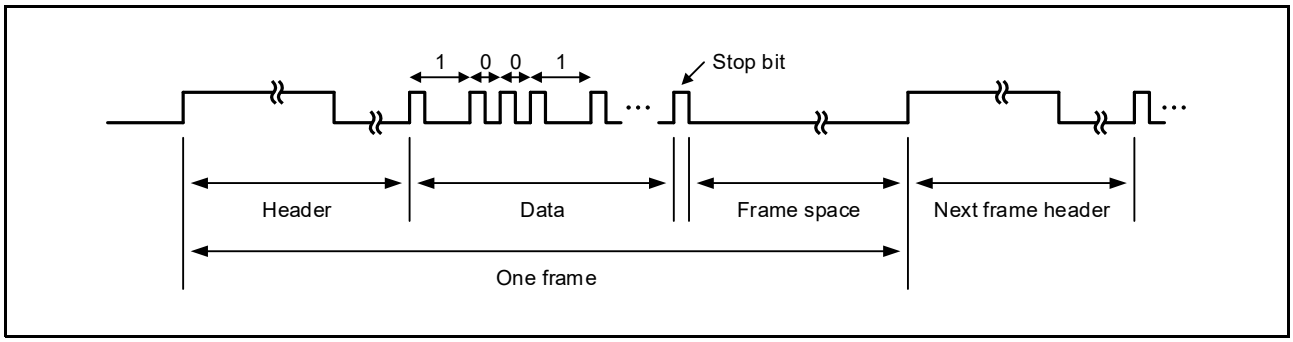
For details, see **4.3.1 Port mode registers (PMxx)** and **4.3.8 Port mode control T registers (PMCTxx)**. For examples of register settings for controlling port functions, see **4.5.4 Examples of register settings for port and multiplexed functions**.

18.3 Operation

18.3.1 Overview of REMC operation

Figure 18 - 24 shows an example of the remote control signal. The signal begins with a header, followed by a sequence of data. This header differs from the subsequent sequence of data in waveform, allowing the header and the data to be distinguished. The sequence of data contains custom code and data code, and 0 or 1 is distinguished depending on the bit length. After a stop bit, there is an interval during which the signal does not change (frame space), thus constituting a frame. The time between the edges of the external input signal is measured using the base timer in the REMC. The patterns of the remote control signal are detected and the data is captured according to the measurement results.

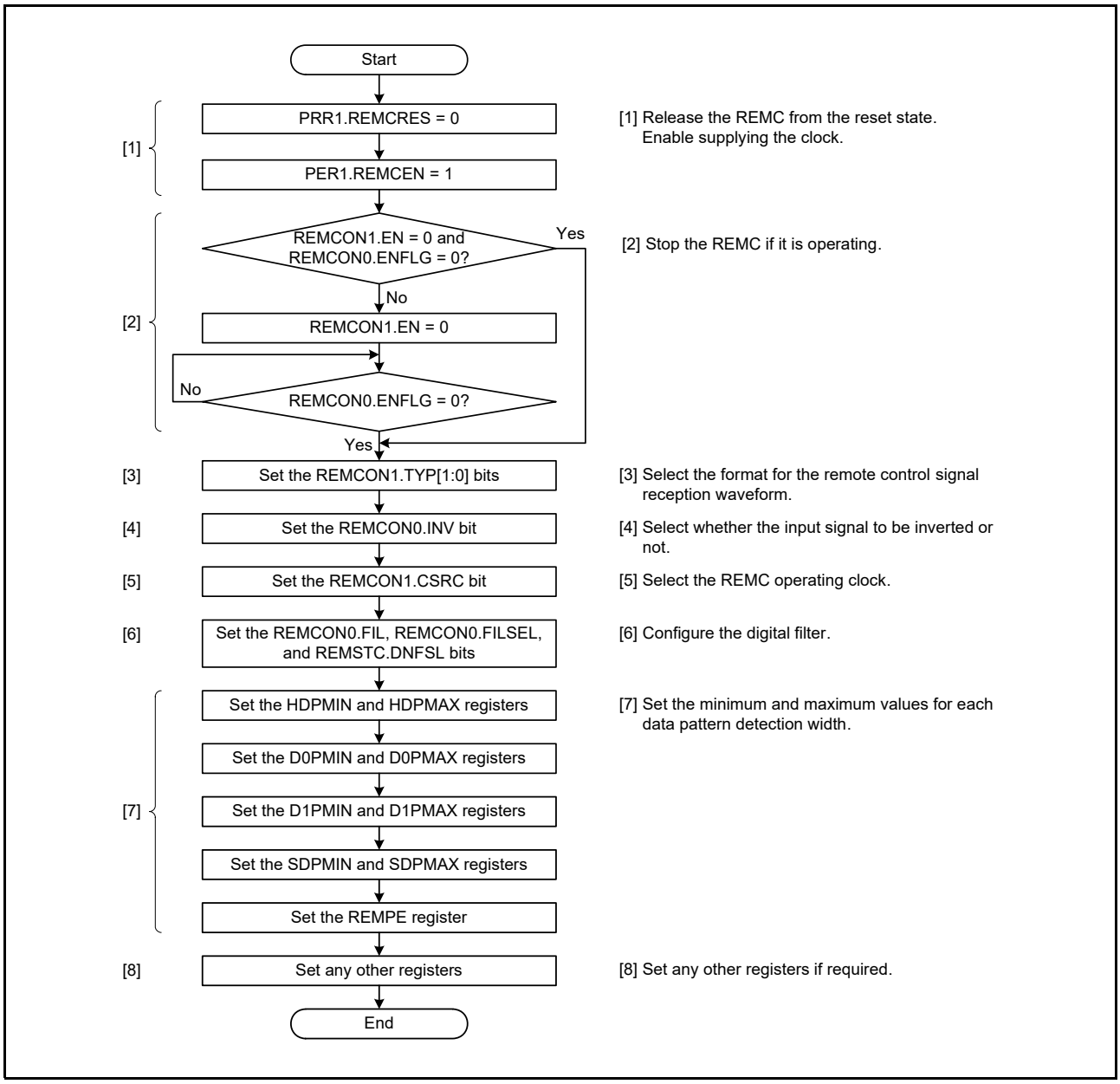
Figure 18 - 24 Example of Remote Control Signal



18.3.2 Initial setting

Set up the REMC in accord with the procedure in **Figure 18 - 25** to receive the remote control signal. Set the REMCON1.EN bit to 0 if the REMC is operating. Then the REMCON0.ENFLG flag becomes 0 and the REMC stops the operation. Set the format for the remote control signal reception waveform by the REMCON1.TYP[1:0] bits; select the signal inversion or non-inversion by the REMCON0.INV bit; select the operating clock by the REMCON1.CSRC bit; and set the digital filter by the REMCON0.FIL, REMCON0.FILSEL, and REMSTC.DNFSL bits, while the REMCON0.ENFLG flag is 0. Set the detecting width for each data pattern into the HDPMIN, HDPMAX, D0PMIN, D0PMAX, D1PMIN, D1PMAX, SDPMIN, SDPMAX, and REMPE registers. Make any other settings such as enabling interrupts by the REMINT register and setting of the compare function by the REMCPC and REMCPD registers if required. After all necessary register settings are completed, set the REMCON1.EN bit to 1 to start REMC operation.

Figure 18 - 25 Example of Flowchart for Initial Settings of REMC



18.3.3 Pattern setting

The format for capturing the remote control signal reception waveform can be set by setting the REMCON1.TYP[1:0] bits. **Figure 18 - 26** and **Figure 18 - 27** show examples of a remote control signal reception waveform captured by setting the REMCON1.TYP[1:0] bits.

When the REMCON1.TYP[1:0] bits are 00B (format A)

The measured result is determined from the setting value of the header pattern at the rising edge of the internal input signal. When the header pattern is received, the measured result is determined from the setting values of the data 0, data 1 and special data patterns at the rising edge of the internal input signal.

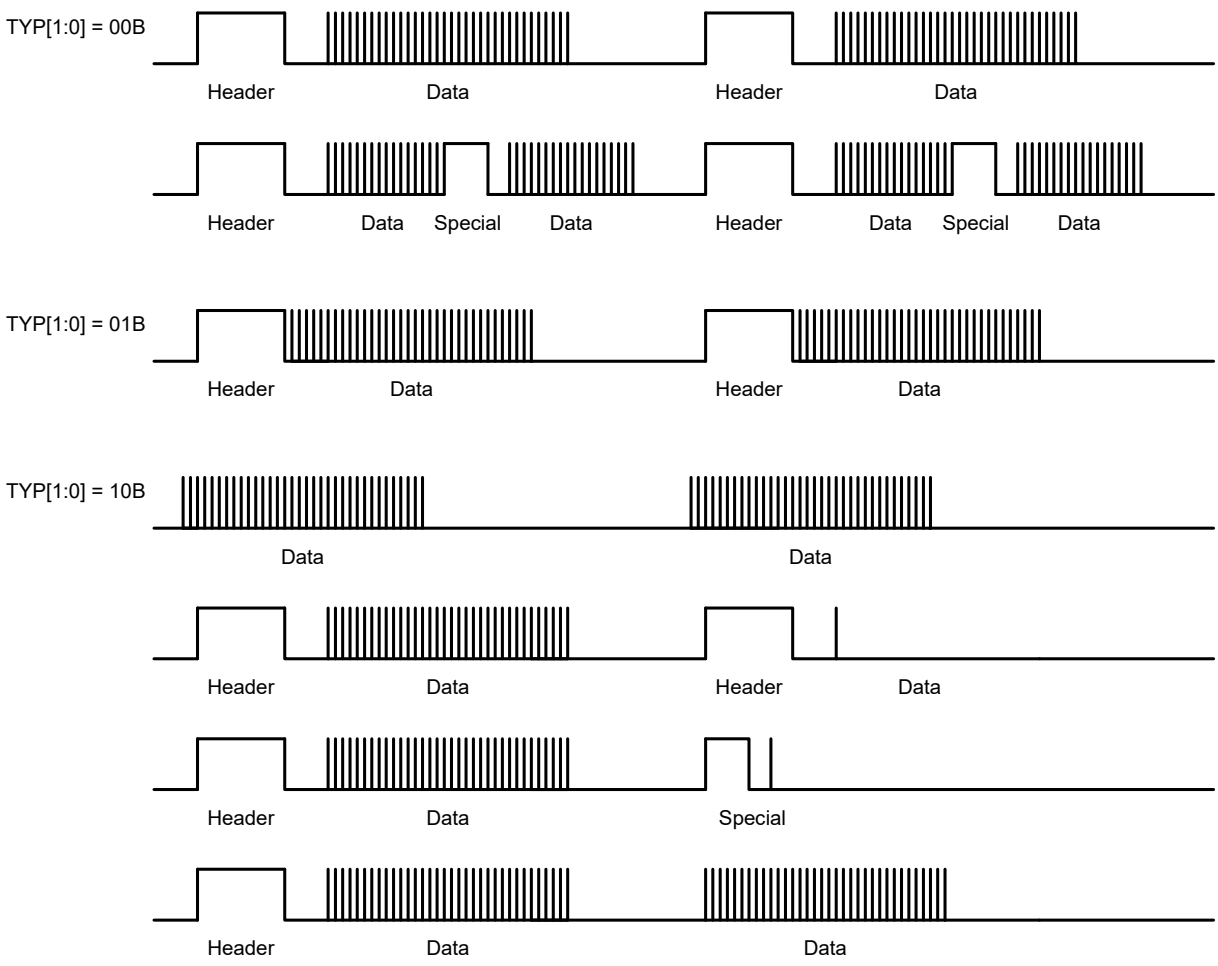
When the REMCON1.TYP[1:0] bits are 01B (format B)

The measured result is determined from the setting value of the header pattern at the falling edge of the internal input signal. When the header pattern is received, the measured result is determined from the setting values of the data 0, data 1 and special data patterns at the falling edge of the internal input signal. The header pattern is detected once within one frame.

When the REMCON1.TYP[1:0] bits are 10B (format C)

The measured result is determined from the setting values of the header, data 0, data 1 and special data patterns at the rising edge of the internal input signal.

Figure 18 - 26 Example of Remote Control Signal Reception Waveform Captured by Setting REMCON1.TYP[1:0] Bits (REMCON0.INV = 0)



TYP[1:0]	Header Pattern	Data 0 Pattern	Data 1 Pattern	Special Data Pattern
TYP[1:0] = 00B				
TYP[1:0] = 01B				
TYP[1:0] = 10B				

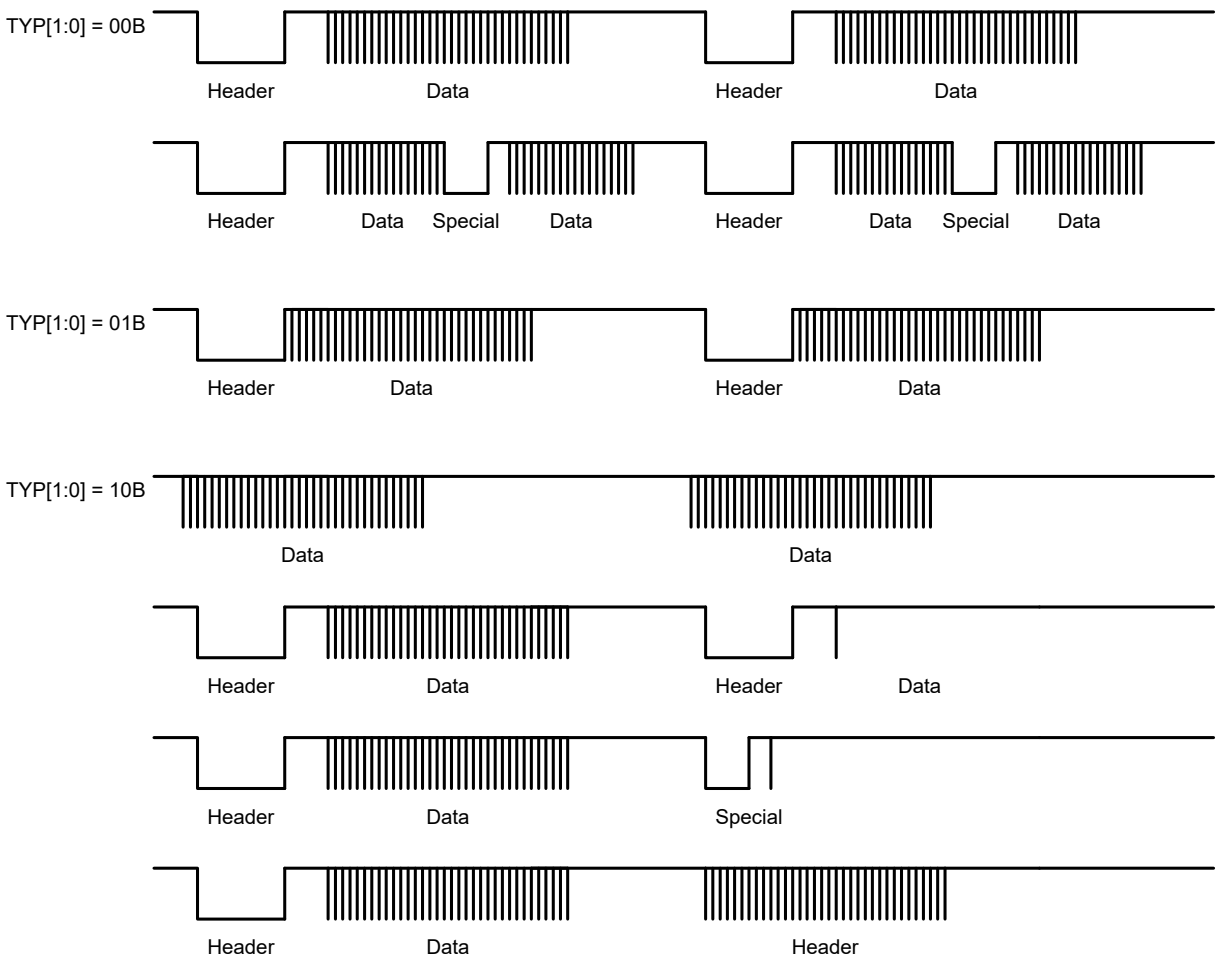
TYP[1:0]: Bits in the REMCON1 register

↑ ↓ • • • Measurement starts

↑ ↓ • • • Pattern determined

When TYP[1:0] = 01B, the number of detecting the header pattern within the receive frame is one.
The above diagram applies when the REMCON0.INV bit is 0.

Figure 18 - 27 Example of Remote Control Signal Reception Waveform Captured by Setting REMCON1.TYP[1:0] Bits (REMCON0.INV = 1)



TYP[1:0]	Header Pattern	Data 0 Pattern	Data 1 Pattern	Special Data Pattern
TYP[1:0] = 00B				
TYP[1:0] = 01B				
TYP[1:0] = 10B				

TYP[1:0]: Bits in the REMCON1 register

↑↓ · · · Measurement starts

↑↓ · · · Pattern determined

When TYP[1:0] = 01B, the number of detecting the header pattern within the receive frame is one.
The above diagram applies when the REMCON0.INV bit is 1.

18.3.4 Operating clock

The REMC can use either of the following clock signals as its operating clock: the low-speed peripheral clock (f_{SXP}) and timer interrupt (INTTM06). When the timer interrupt (INTTM06) is selected as the REMC operating clock, a transition to SNOOZE mode is necessary to receive remote control signals if the current mode is STOP mode. When transition from the STOP mode to the SNOOZE mode is enabled, select the low-speed peripheral clock (f_{SXP}) as the clock of the digital filter. When selecting the timer interrupt (INTTM06) as the REMC operating clock, see **Section 7 Timer Array Unit (TAU)** to confirm the operating clock setting for channel 6 of timer array unit 0. When the low-speed peripheral clock (f_{SXP}) is selected as the REMC operating clock, remote control signals can be received in STOP mode. When the low-speed on-chip oscillator clock (f_{IL}) is selected as the low-speed peripheral clock (f_{SXP}), correct the accuracy of the oscillation frequency of the low-speed on-chip oscillator as required. The following describes how to supply these clock signals.

18.3.4.1 When using f_{SXP} as the REMC operating clock

This section describes the procedure for using the low-speed peripheral clock (f_{SXP}) as the REMC operating clock.

a) When using the low-speed on-chip oscillator clock (f_{IL}) as the low-speed peripheral clock (f_{SXP})
The low-speed on-chip oscillator clock (f_{IL}) is supplied as the REMC operating clock by setting the WUTMMCK0 bit of the subsystem clock supply mode control register (OSMC) to 1 and setting the REMCON1.CSRC bit to 0. For details, see **6.1 Functions of Clock Generator**.

b) When using the subsystem clock X (f_{SX}) as the low-speed peripheral clock (f_{SXP})
The subsystem clock oscillator starts to oscillate by setting the RTCLPC and WUTMMCK0 bits of the subsystem clock supply mode control register (OSMC) to 0, setting the subsystem clock operation mode by the EXCLKS and OSCSELS bits of the clock operation mode control register (CMC), and setting the XTSTOP bit of the clock operation status control register (CSC) to 0. Then, after the oscillation stabilization time for the subsystem clock is secured by using the timer function or another function, by using software, the subsystem clock X (f_{SX}) is supplied as the REMC operating clock by setting the REMCON1.CSRC bit to 0. For the subsystem clock setting, see **6.6.3 Example of setting the XT1 oscillator clock**.

18.3.4.2 When using INTTM06 as the REMC operating clock

The timer interrupt (INTTM06) is supplied as the REMC operating clock by setting the REMCON1.CSRC bit to 1. Operate channel 6 of timer array unit 0 in the interval timer mode. For details about the operation of timer array unit 0, see **Section 7 Timer Array Unit (TAU)**.

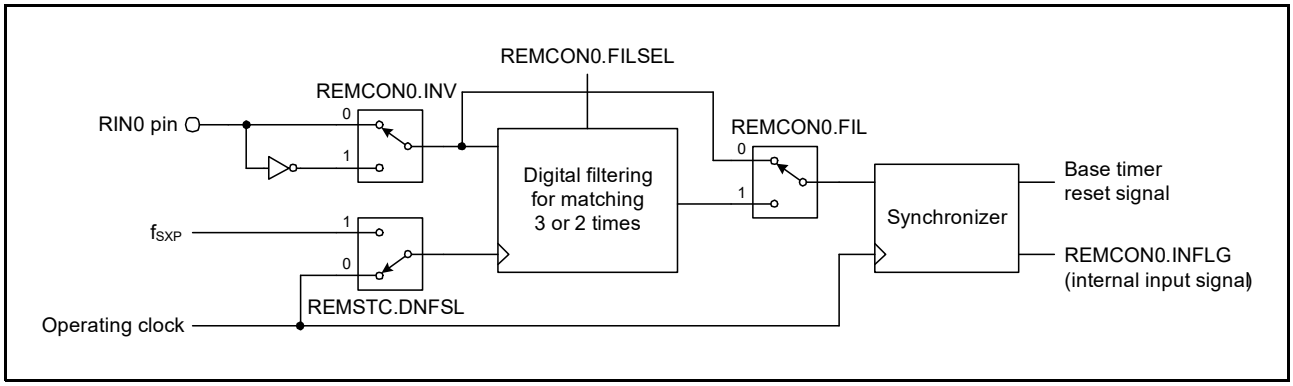
18.3.5 RIN0 input

The options below can be selected in RIN0 input.

- Input polarity
- Digital filter

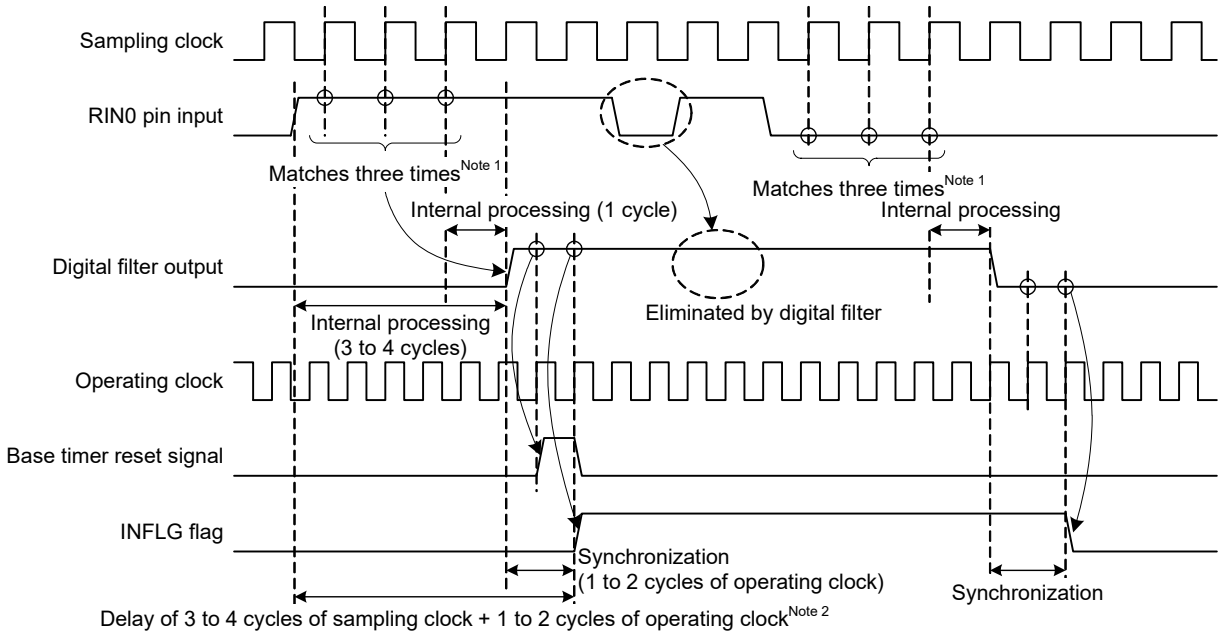
Figure 18 - 28 shows the configuration of RIN0 internal input signal generation.

Figure 18 - 28 RIN0 Internal Input Signal Generation Configuration



The input polarity of the RIN0 pin can be inverted. Whether to invert or not can be selected by the REMCON0.INV bit. When the REMCON0.FIL bit is 1 (digital filter enabled), if the signal input to the RIN0 pin holds the same level for k sequential cycles (k = 3 or 2; value selected by the REMCON0.FILSEL bit), that level is transferred to the internal circuit. This enables noise to be eliminated from k cycles of the sampling clock. The sampling clock of the digital filter is selectable from the REMC operating clock and f_{SXP} by setting the REMSTC.DNFSL bit. When setting the REMSTC.SNZON bit to 1 (transition from STOP mode to SNOOZE mode is enabled), set the REMCON0.FIL bit to 1 (the digital filter is enabled) and the REMSTC.DNFSL bit to 1 (f_{SXP} is selected as a sampling clock). Input to the RIN0 pin is transferred as the REMCON0.INFLG flag (input signal flag) and the base timer reset signal to the internal circuit in synchronization with the operating clock. The base timer reset signal is used to initialize the internal base timer to the pattern detection corresponding to the REMCON1.TYP[1:0] setting. There is a delay caused by internal processing after the input to the RIN0 pin is changed and before these signals are generated. **Figure 18 - 29** shows digital filtering for RIN0 input.

Figure 18 - 29 Digital Filtering for RIN0 Input



Note 1. The above applies when the REMCON0.FIL bit is 1 (digital filter enabled) and the REMCON0.FILSEL bit is 0 (matches three times). When the FILSEL bit is 1 (matches twice), the signal is output earlier by one cycle of sampling clock than that in this figure.

Note 2. When the FIL bit is 0 (digital filter disabled), a delay becomes one to two cycles of the operating clock for synchronization.

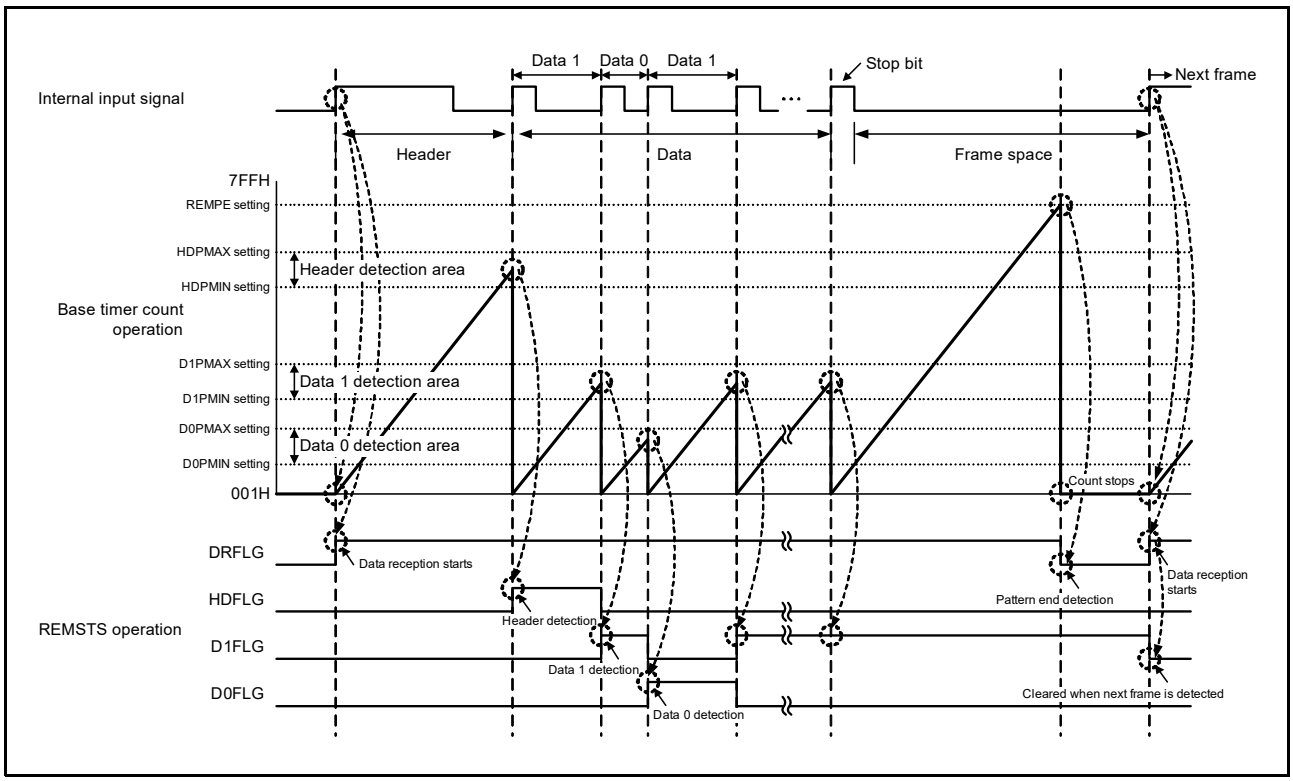
18.3.6 Pattern detection

The REMC has a function that detects the following patterns.

- Header pattern
- Data 0 pattern
- Data 1 pattern
- Special data pattern

Using the base timer included in the REMC, the time between the edges of the external input signal is measured to determine which pattern matches the measurement result. This enables detection of the remote control signal and capturing the data. The width for determining each pattern can be set to any value using each pattern setting register.
Figure 18 - 30 shows the waveform of REMC operation.

Figure 18 - 30 Waveform of REMC Operation



18.3.6.1 Header pattern detection

The header pattern can be detected by setting the minimum width of the header pattern in the HDPMIN register and the maximum width in the HDPMAX register. The minimum and maximum widths of the header pattern must be “1 < HDPMIN register value ≤ HDPMAX register value”.

$$\text{Setting value n} = \frac{\text{Minimum width (maximum width) of header pattern}}{\text{Operating clock cycle time}}$$

When not using the header pattern, set the HDPMIN and HDPMAX registers to 000H. Make sure that the setting value of the header pattern is different from the setting values of data 0, data 1, and special data patterns, and the setting ranges are not overlapped.

When the REMCON1.TYP[1:0] bits are 00B or 01B, if the data 0, data 1, or special data pattern is detected before the header pattern is detected, the following occur:

- The REMSTS.REFLG flag becomes 1 (an error has occurred).
- The REMSTS.D0FLG, REMSTS.D1FLG, and REMSTS.SDFLG flags remain unchanged.
- The REMDAT0 to REMDAT7 registers remain unchanged.

When the REMCON1.TYP[1:0] bits are 01B, the number of detecting the header pattern is one while the DRFLG flag is 1.

18.3.6.2 Data 0 pattern detection

The data 0 pattern can be detected by setting the minimum width of the data 0 pattern in the D0PMIN register and the maximum width in the D0PMAX register. The minimum and maximum widths of the data 0 pattern must be “1 < D0PMIN register value ≤ D0PMAX register value”.

$$\text{Setting value n} = \frac{\text{Minimum width (maximum width) of data 0 pattern}}{\text{Operating clock cycle time}}$$

When not using the data 0 pattern, set the D0PMIN and D0PMAX registers to 00H. Make sure that the setting value of the data 0 pattern is different from the setting values of the header, data 1, and special data patterns, and the setting ranges are not overlapped.

When the REMCON1.TYP[1:0] bits are 00B or 01B, if the data 0 pattern or data 1 pattern is detected before the header pattern is detected, the following occur:

- The REMSTS.REFLG flag becomes 1 (an error has occurred).
- The REMSTS.D0FLG, REMSTS.D1FLG, and REMSTS.SDFLG flags remain unchanged.
- The REMDAT0 to REMDAT7 registers remain unchanged.

18.3.6.3 Data 1 pattern detection

The data 1 pattern can be detected by setting the minimum width of the data 1 pattern in the D1PMIN register and the maximum width in the D1PMAX register. The minimum and maximum widths of the data 1 pattern must be “1 < D1PMIN register value ≤ D1PMAX register value”.

$$\text{Setting value n} = \frac{\text{Minimum width (maximum width) of data 1 pattern}}{\text{Operating clock cycle time}}$$

When not using the data 1 pattern, set the D1PMIN and D1PMAX registers to 00H. Make sure that the setting value of the data 1 pattern is different from the setting values of the header, data 0, and special data patterns, and the setting ranges are not overlapped.

When the REMCON1.TYP[1:0] bits are 00B or 01B, if the data 0 pattern or data 1 pattern is detected before the header pattern is detected, the following occur:

- The REMSTS.REFLG flag becomes 1 (an error has occurred).
- The REMSTS.D0FLG, REMSTS.D1FLG, and REMSTS.SDFLG flags remain unchanged.
- The REMDAT0 to REMDAT7 registers remain unchanged.

18.3.6.4 Special data pattern detection

The special data pattern can be detected by setting the minimum width of the special data pattern in the SDPMIN register and the maximum width in the SDPMAX register. The minimum and maximum widths of the special data pattern must be “1 < SDPMIN register value ≤ SDPMAX register value”.

$$\text{Setting value n} = \frac{\text{Minimum width (maximum width) of special data pattern}}{\text{Operating clock cycle time}}$$

When not using the special data pattern, set the SDPMIN and SDPMAX registers to 000H. Make sure that the setting value of the special data pattern is different from the setting values of the header, data 0, and data 1 patterns, and the setting ranges are not overlapped.

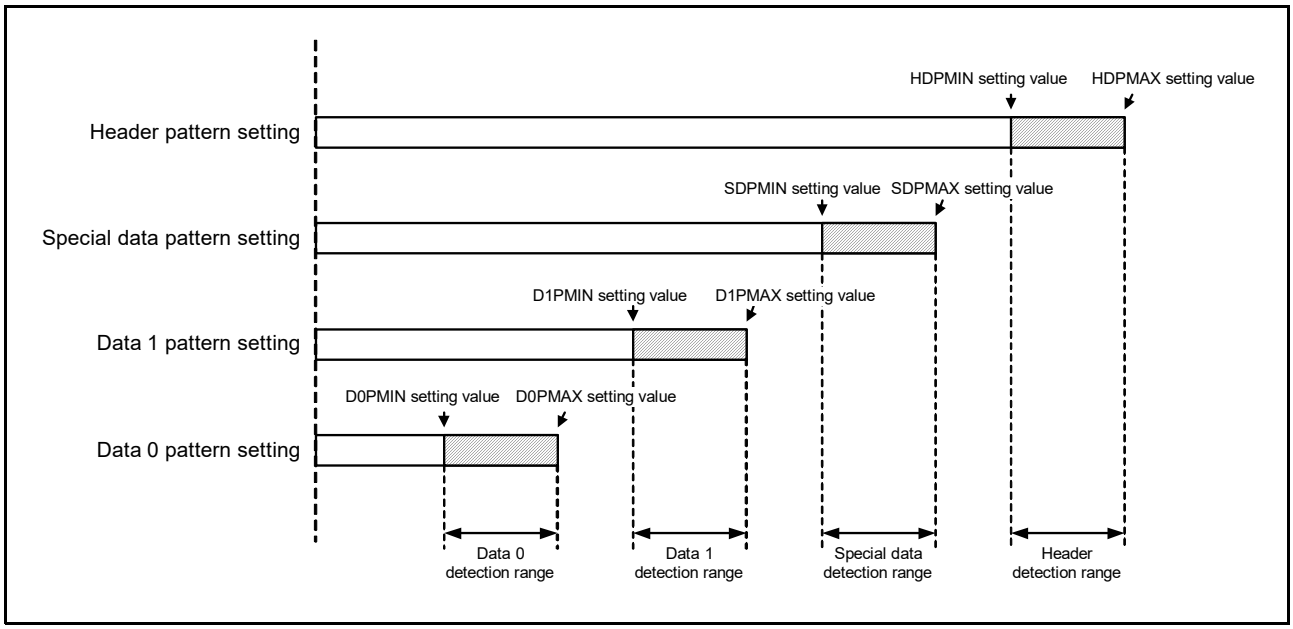
When the REMCON1.TYP[1:0] bits are 00B or 01B, if the special data pattern is detected before the header pattern is detected, the following occur:

- The REMSTS.REFLG flag becomes 1 (an error has occurred).
- The REMSTS.SDFLG flag remains unchanged.
- The REMDAT0 to REMDAT7 registers remain unchanged.

18.3.6.5 Examples of setting pattern setting registers

For the header, data 0, data 1, and special data pattern setting registers, make sure that the minimum to maximum values of each pattern are different, and the setting ranges do not overlap as shown in **Figure 18 - 31**.

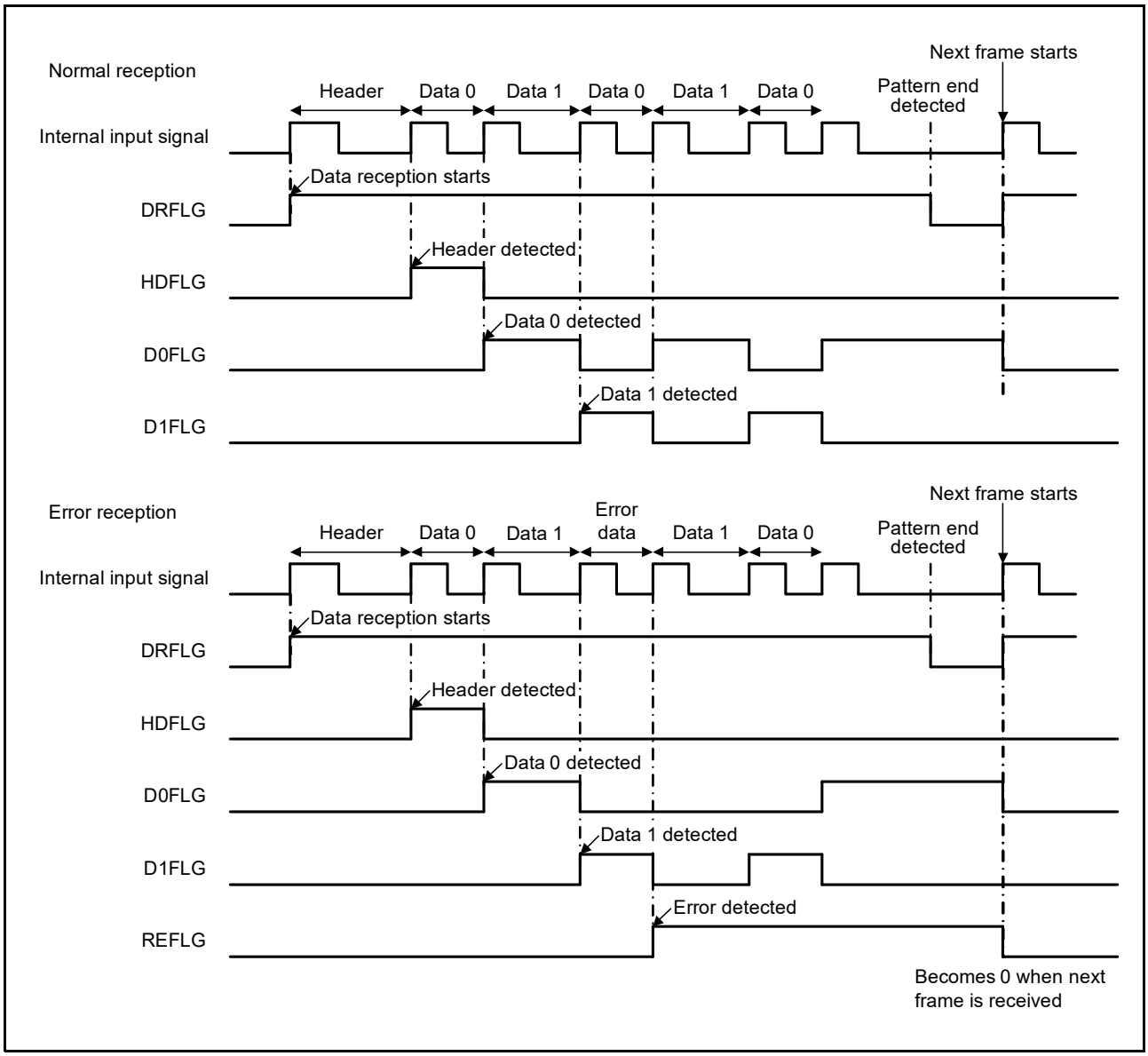
Figure 18 - 31 Examples of Setting Pattern Setting Registers



18.3.6.6 Updating status flags upon pattern detection

The detected patterns can be confirmed by reading the following flags: header pattern match flag (REMSTS.HDFLG), data 0 pattern match flag (REMSTS.D0FLG), data 1 pattern match flag (REMSTS.D1FLG), and special data pattern match flag (REMSTS.SDFLG). These flags are cleared when a different pattern is detected. If a pattern other than the above patterns is detected, it is detected as an error pattern. This can be confirmed by reading the receive error flag (REMSTS.REFLG). This flag is cleared when the next frame is received. **Figure 18 - 32** shows an example of pattern detection and flag operation.

Figure 18 - 32 Example of Flag Operation



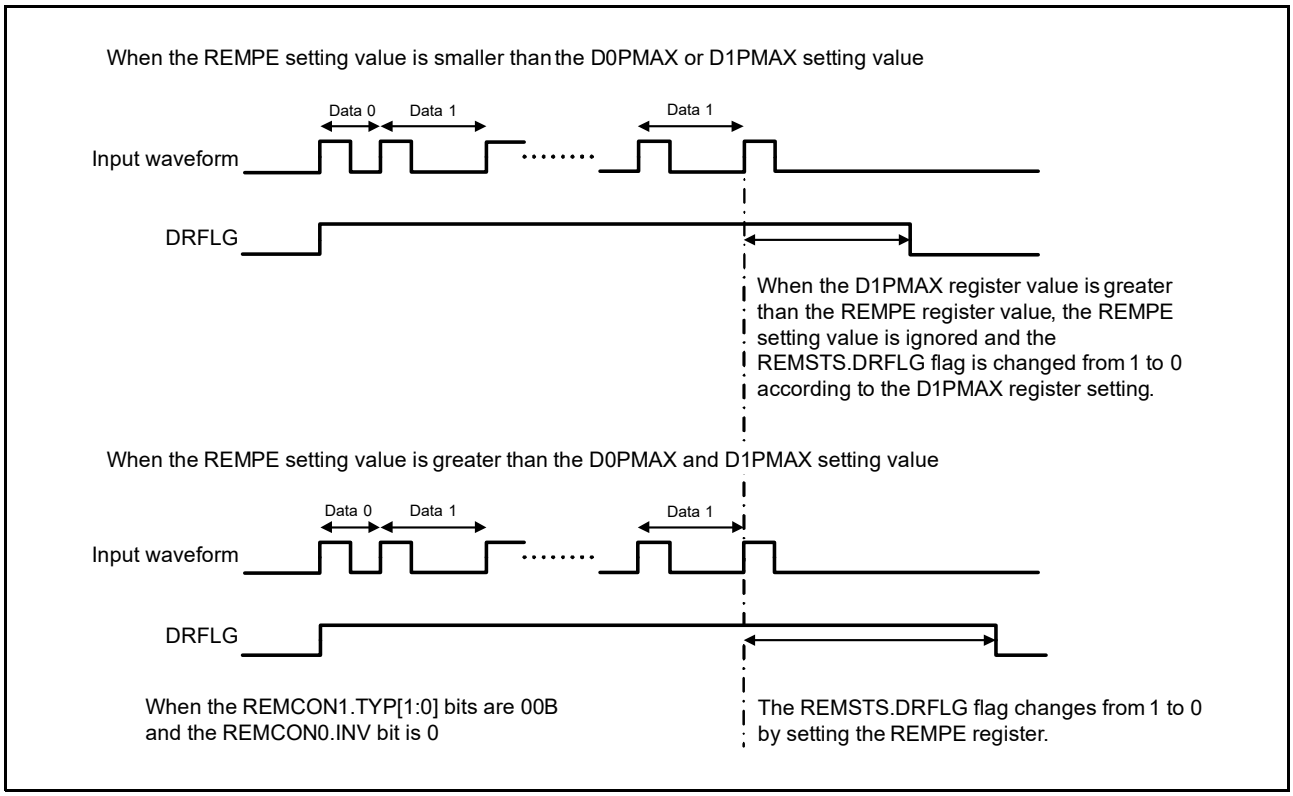
18.3.7 Pattern end

The timing when the REMSTS.DRFLG flag becomes 0 can be set.

When setting the REMPE register, be sure to set that the REMPE value > HDPMAX, D0PMAX, D1PMAX, or SDPMAX value.

When the REMPE value ≤ HDPMAX, D0PMAX, D1PMAX, or SDPMAX value, the value specified in the REMPE register is not used to set the timing when the REMSTS.DRFLG flag becomes 0. In this case, data reception is completed according to the largest value from among the setting values of the HDPMAX, D0PMAX, D1PMAX, and SDPMAX registers. **Figure 18 - 33** shows operation of the data reception complete flag for each pattern end setting.

Figure 18 - 33 Operation of Data Reception Complete Flag for Each Pattern End Setting



18.3.8 Receive data buffer

The receive data j register (REMDATj) (j = 0 to 7) is an 8-byte (64-bit) buffer for storing received data. When data 0 pattern or data 1 pattern is detected, the detection result is sequentially stored starting from the REMDAT0.DAT0[0] bit as shown in **Figure 18 - 34**. The REMRBIT register is counted up at the same time, so the number of the current received bits can be checked by reading the REMRBIT register. See **Table 18 - 4** for the relationship between the number of received bits and the location where data is stored. The values of the REMDATj and REMRBIT registers do not change even when the header pattern or special data pattern is received. If the REMDATj or REMRBIT register is read while the data is being updated, the value read may be undefined.

Figure 18 - 34 Operation of Receive Data Buffer

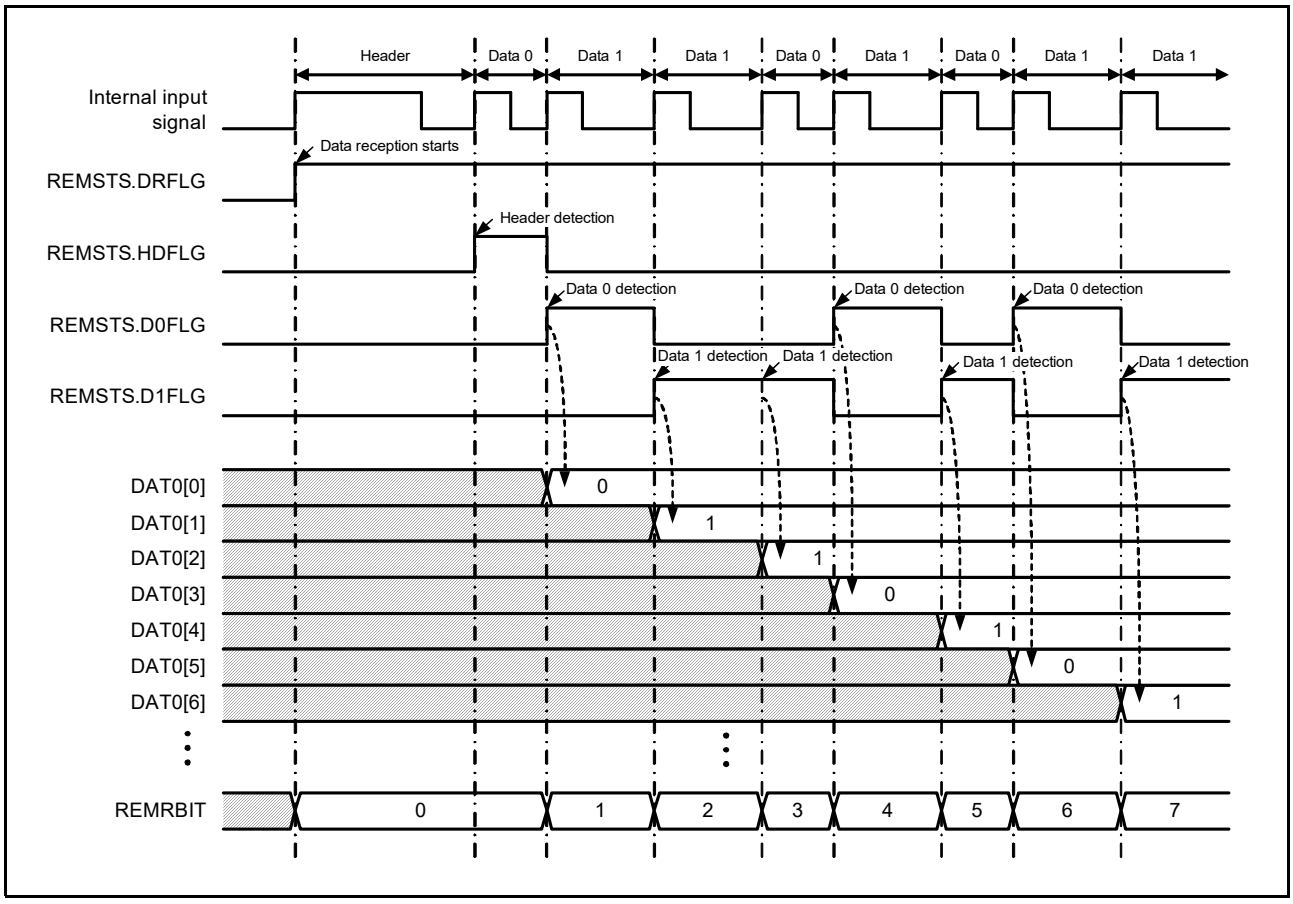


Table 18 - 4 Relationship between Number of Received Bits and Location Where Data is Stored

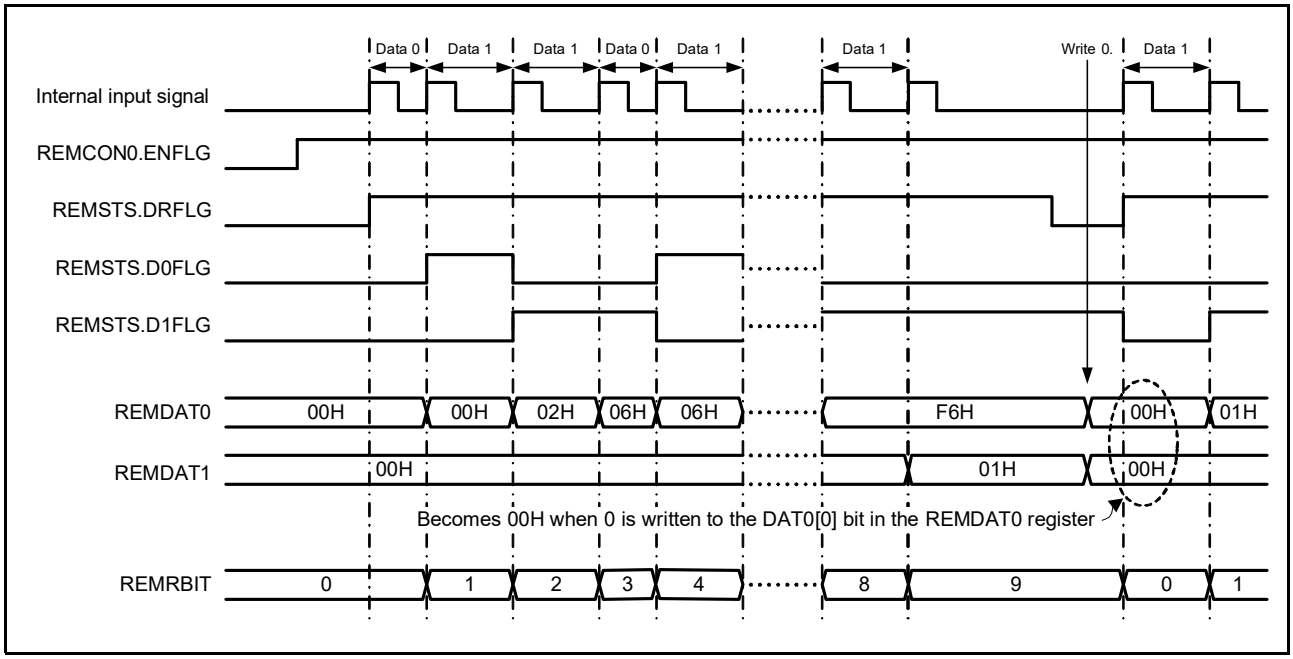
Number of Received Bits	Location Where Data is Stored	
	Register Name	Bit Name
1	REMDAT0	DAT0[0]
2		DAT0[1]
3		DAT0[2]
4		DAT0[3]
5		DAT0[4]
6		DAT0[5]
7		DAT0[6]
8		DAT0[7]
9	REMDAT1	DAT1[0]
10		DAT1[1]
11		DAT1[2]
12		DAT1[3]
13		DAT1[4]
14		DAT1[5]
15		DAT1[6]
16		DAT1[7]
17	REMDAT2	DAT2[0]
18		DAT2[1]
19		DAT2[2]
20		DAT2[3]
21		DAT2[4]
22		DAT2[5]
23		DAT2[6]
24		DAT2[7]
25	REMDAT3	DAT3[0]
26		DAT3[1]
27		DAT3[2]
28		DAT3[3]
29		DAT3[4]
30		DAT3[5]
31		DAT3[6]
32		DAT3[7]

Number of Received Bits	Location Where Data is Stored	
	Register Name	Bit Name
33	REMDAT4	DAT4[0]
34		DAT4[1]
35		DAT4[2]
36		DAT4[3]
37		DAT4[4]
38		DAT4[5]
39		DAT4[6]
40		DAT4[7]
41	REMDAT5	DAT5[0]
42		DAT5[1]
43		DAT5[2]
44		DAT5[3]
45		DAT5[4]
46		DAT5[5]
47		DAT5[6]
48		DAT5[7]
49	REMDAT6	DAT6[0]
50		DAT6[1]
51		DAT6[2]
52		DAT6[3]
53		DAT6[4]
54		DAT6[5]
55		DAT6[6]
56		DAT6[7]
57	REMDAT7	DAT7[0]
58		DAT7[1]
59		DAT7[2]
60		DAT7[3]
61		DAT7[4]
62		DAT7[5]
63		DAT7[6]
64		DAT7[7]

Note When the data exceeds 64 bits, the REMDATj register is sequentially overwritten from the first bit.

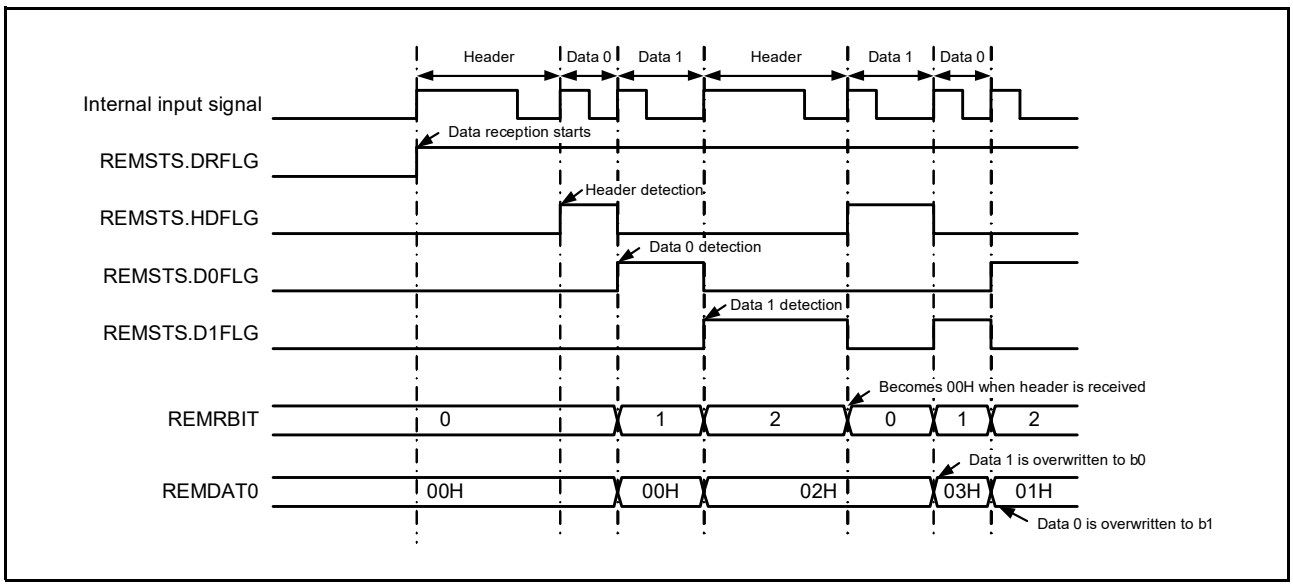
When 0 is written to the REMDAT0.DAT0[0] bit, the values of the REMDAT0 to REMDAT7 registers become 00H after one to two cycles of the operating clock. **Figure 18 - 35** shows the REMDATj/REMRBIT register operation when 00H is written to the REMDAT0 register.

Figure 18 - 35 REMDATj/REMRBIT Register Operation (00H is Written to REMDAT0 Register)



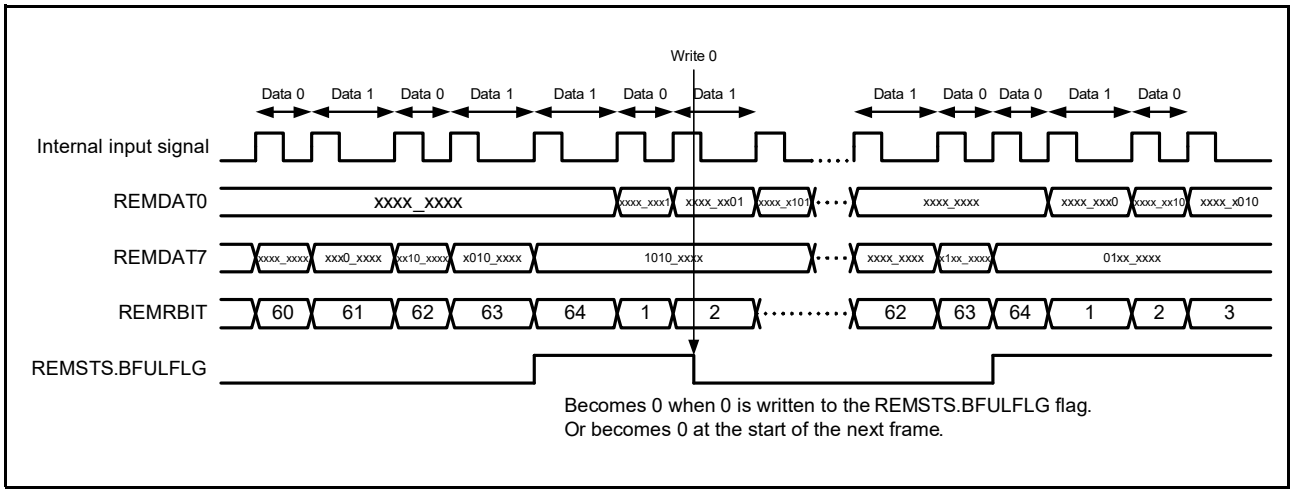
When 0 is written to the REMRBIT.RBIT[0] bit, the value of the REMRBIT register becomes 00H after one to two cycles of the operating clock. When the REMCON1.TYP[1:0] bits are 00B or 01B, if the header pattern is detected during data reception, the value of the REMRBIT register is initialized to 00H and the received data is sequentially overwritten from the REMDAT0.DAT0[0] bit. **Figure 18 - 36** shows operation of header pattern detection during data reception.

Figure 18 - 36 Operation of Header Pattern Detection during Data Reception



When the data exceeds 64 bits, the buffer is sequentially overwritten from the first bit. **Figure 18 - 37** shows the REMRBIT register operation when the REMSTS.BFULFLG flag becomes 1.

Figure 18 - 37 REMRBIT Register Operation (REMSTS.BFULFLG Flag = 1)



18.3.9 Compare function

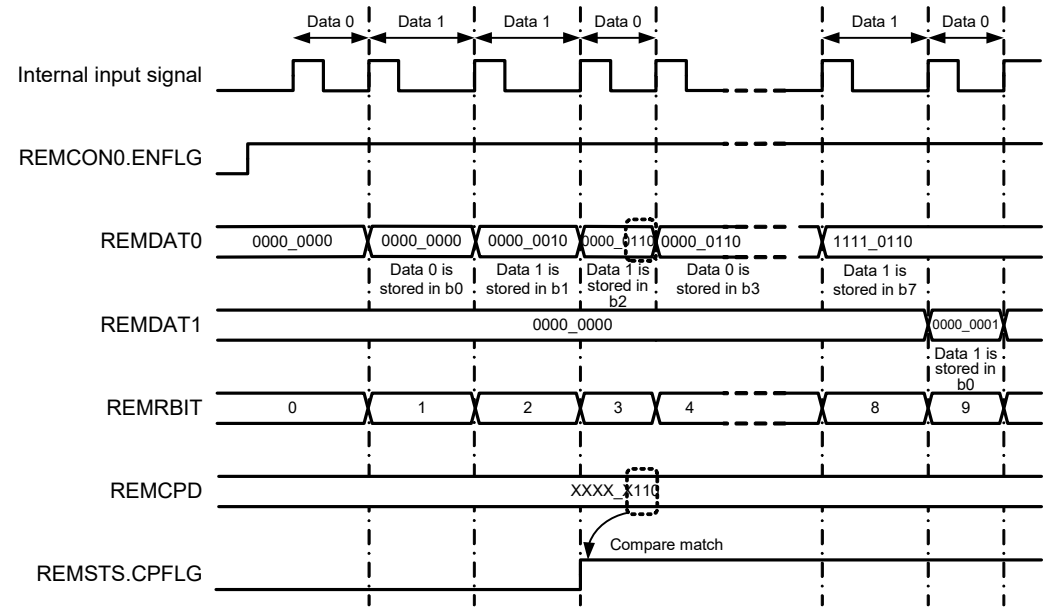
The REMC has a function to compare the value of the REMCPD register with the value of the REMDAT1 and REMDAT0 registers. As a result of comparison, it can be detected that the first 1 to 16 bits of the remote control signal are the specific values. **Figure 18 - 38** shows the operation timing of the receive buffer and the compare function.

When using the compare function, set registers as shown below:

- Select bits to be compared by setting the REMCPC.CPN[3:0] bits (when the setting value is n, bits n to 0 are compared. n: 0 to 15).
- Set the compare data in the REMCPD register.
  When the value of the REMRBIT register becomes the bit count specified by the REMCPC.CPN[3:0] bits, if the value of the REMDAT1 and REMDAT0 registers matches the value of the REMCPD register, the REMSTS.CPFLG flag becomes 1 (compare match).

When the value of the REMRBIT register matches the bit count specified by the REMCPC.CPN[3:0] bits during reception of 64 bits or more, even if the value of the REMDAT1 and REMDAT0 registers matches the value of the REMCPD register, the REMSTS.CPFLG flag does not become 1 (compare match).

Figure 18 - 38 Receive Buffer and Compare Function



The above applies under the following conditions:
• The REMCON1.TYP[1:0] bits are 10B
• The REMCPC.CPN[3:0] bits are set to 2 (bits 2 to 0 are compared)

18.3.10 Error pattern reception

When the error pattern is detected during data reception, subsequent operation differs depending on the setting of the REMCON0.EC bit. **Figure 18 - 39** shows operation of the REMDAT0 and REMRBIT registers when the REMCON0.EC bit is set to 0. If an error is detected while the REMCON0.EC bit is 0, the data when the error is detected is not captured, but data is captured when the data 0 pattern or data 1 pattern is detected later. **Figure 18 - 40** shows operation of the REMDAT0 and REMRBIT registers when the REMCON0.EC bit is set to 1. If an error is detected while the REMCON0.EC bit is 1, the values of the REMRBIT and REMDAT0 to REMDAT7 registers are not updated even when the data 0 pattern or data 1 pattern is detected later. Once the REMSTS.DRFLG flag is cleared and after data reception is completed, if data reception starts again, the REMSTS.REFLG flag is cleared and data is captured.

Figure 18 - 39 REMDAT0 and REMRBIT Registers Operation upon Error Detection when REMCON0.EC = 0

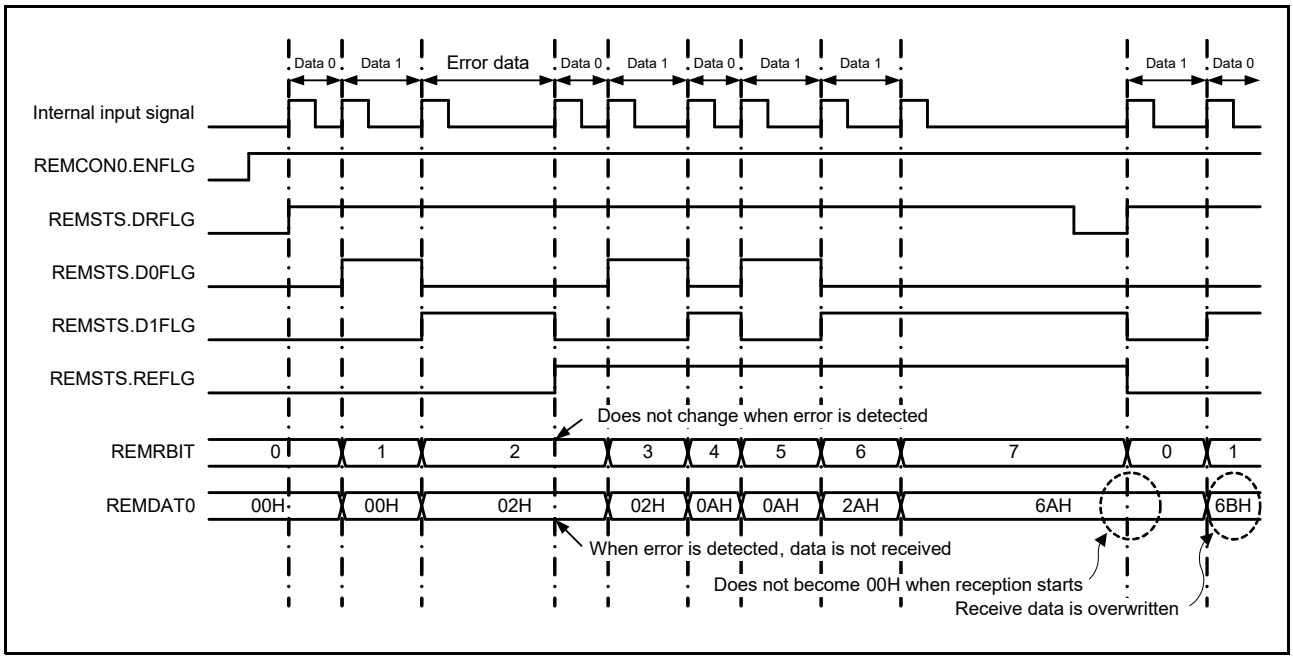
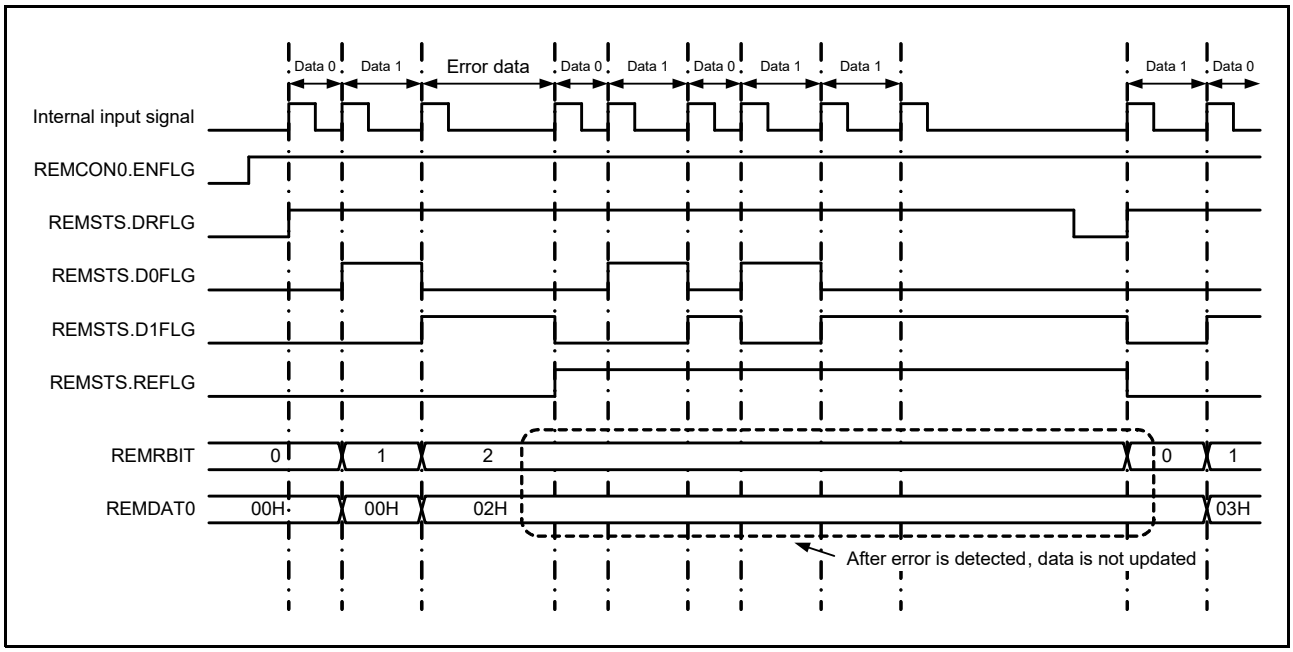


Figure 18 - 40 REMDAT0 and REMRBIT Registers Operation upon Error Detection when REMCON0.EC = 1

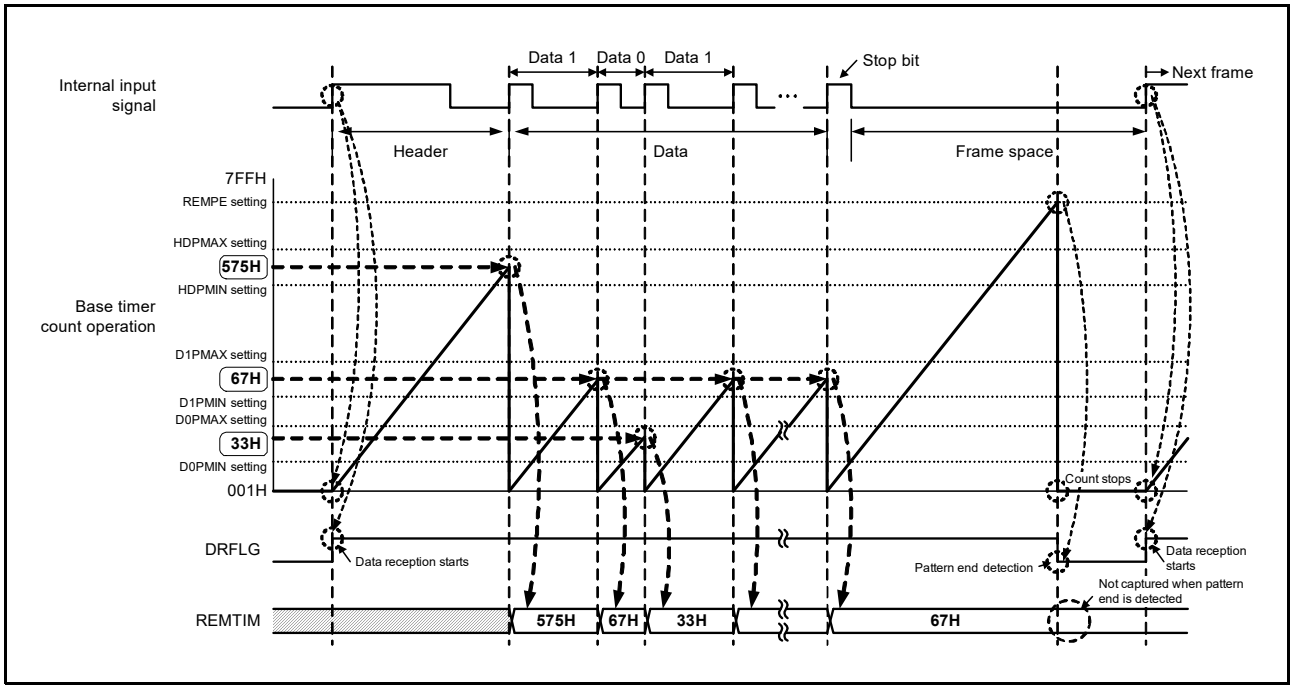


18.3.11 Storing base timer value when pattern is detected

The measurement result register (REMTIM) holds the base timer value when any of the following patterns is detected. This register allows each pattern width to be measured. **Figure 18 - 41** shows an operation example of the measurement function.

- Header pattern
- Data 0 pattern
- Data 1 pattern
- Special data pattern
- Data pattern other than the above (receive error)

Figure 18 - 41 Operation Example of Measurement Function



18.3.12 Interrupts

The REMC generates the following interrupt requests: compare match, receive error, data reception complete, receive buffer full, header pattern match, data 0 pattern or data 1 pattern match, and special data pattern match interrupts. An interrupt request is output when the generation conditions for these interrupt requests and the generation conditions for the selected interrupt mode are met. There are two interrupt modes, normal interrupt mode (OR condition) and sequential interrupt mode (AND condition), which can be selected by using the REMCON1.INTMD bit. In normal interrupt mode, if any of the interrupt generation conditions of sources whose interrupt enable bit of the REMINT register is set to 1 is met, an interrupt request signal (INTREMC) is generated. In sequential interrupt mode, if all the interrupt generation conditions of the sources (compare match, data reception complete, header pattern match, and special data pattern match) whose interrupt enable bit of the REMINT register is set to 1 are met, an interrupt (INTREMC) is generated. **Table 18 - 5** lists the interrupt sources of the REMC. Refer to **Section 21 Interrupt Functions** for details on interrupt control.

Table 18 - 5 REMC Interrupt Sources

Interrupt Source	Interrupt Request Generation Condition	Interrupt Status Flag	Interrupt Enable Bit
Compare match	When the REMSTS.CPFLG flag changes from 0 to 1	REMSTS.CPFLG	REMINT.CPINT
Receive error	When the REMSTS.REFLG flag changes from 0 to 1 (When a receive error is detected)	REMSTS.REFLG	REMINT.REINT
Completion of data reception	When the REMSTS.DRFLG flag changes from 1 to 0	REMSTS.DRFLG	REMINT.DRINT
Receive buffer full	When the REMSTS.BFULFLG flag changes from 0 to 1	REMSTS.BFULFLG	REMINT.BFULINT
Header pattern match	When the REMSTS.HDFLG flag changes from 0 to 1 (When the header pattern is detected)	REMSTS.HDFLG	REMINT.HDINT
Data 0 pattern or data 1 pattern match	• When the REMSTS.D0FLG flag changes from 0 to 1 (When the data 0 pattern is detected) • When the REMSTS.D1FLG flag changes from 0 to 1 (When the data 1 pattern is detected)	REMSTS.D0FLG, REMSTS.D1FLG	REMINT.DINT
Special data pattern match	When the REMSTS.SDFLG flag changes from 0 to 1 (When the special data pattern is detected)	REMSTS.SDFLG	REMINT.SDINT

18.3.13 SNOOZE mode function

In the SNOOZE mode, the REMC performs remote control data reception when detecting a change in the input level of the RIN0 pin. When the timer interrupt (INTTM06) is selected as the REMC operating clock, normally the REMC data reception stops in STOP mode. However, using the SNOOZE mode enables remote control data reception without CPU operation upon detection of a change in the input level of the RIN0 pin. The SNOOZE mode can only be used when the timer interrupt (INTTM06) is selected as the REMC operating clock. In this case, select a CPU/peripheral hardware clock that can operate in SNOOZE mode as the count clock for channel 6 of timer array unit 0.
When using the REMC in SNOOZE mode, make the following setting before switching to the STOP mode.

- Make initial settings of each register immediately before switching to STOP mode. To enable remote control data reception in SNOOZE mode, set the REMCON0.FIL bit to 1 (enables the digital filter) and set the REMSTC.DNFSL bit to 1 (f_{SXP} is selected as a sampling clock).
- After the initial settings have been completed, set the REMSTC.SNZON bit to 1, and then set the REMCON1.EN bit to 1.
- The CPU shifts to the SNOOZE mode on detecting the valid edge of the RIN0 pin input following a transition to the STOP mode. When the timer interrupt (IMTTM06) is supplied as the REMC operating clock, the REMC starts reception.
- After returning to normal operation mode due to generation of an interrupt source specified in the REMINT register such as a compare match interrupt and header pattern match interrupt, set the SNZON bit to 0.

Caution **The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock (f_{IH}) or medium-speed on-chip oscillator clock (f_{IM}) is selected for the count clock for channel 6 of timer array unit 0.**

Remark When the low-speed peripheral clock (f_{SXP}) is selected for the operating clock, remote control data reception is possible even in STOP mode.

Table 18 - 6 shows the relationship between the conditions for returning from SNOOZE mode to normal operation mode and interrupt modes.

Table 18 - 6 Interrupt Modes and Transition from SNOOZE Mode

	Normal interrupt mode	Sequential interrupt mode
Selectable interrupt sources	Header pattern match Compare match Completion of data reception Special data pattern match Data 0 pattern or data 1 pattern match Receive buffer full Receive error	Header pattern match Compare match Special data pattern match Completion of data reception Data 0 pattern or data 1 pattern match**Note** Receive buffer full**Note** Receive error**Note**
SNOOZE → Normal operation mode returning conditions	• Any of the following interrupts that are enabled in REMINT is generated. Header pattern match Compare match Special data pattern match Completion of data reception Data 0 pattern or data 1 pattern match Receive buffer full Receive error	Either of the following conditions: • Among the following interrupt sources, all of those enabled in REMINT are generated. Header pattern match Compare match Special data pattern match Completion of data reception • Any of the following interrupts that are enabled in REMINT is generated. Data 0 pattern or data 1 pattern match Receive buffer full Receive error
SNOOZE → STOP transition conditions	Data reception is completed while data reception complete interrupt is disabled, and none of the enabled interrupts are generated.	Data reception is completed while data reception complete interrupt is disabled, and not all the enabled interrupts are generated.

Note The AND condition does not apply to these sources In the sequential interrupt mode. As in the normal interrupt mode, the condition is the logical OR of the enabled sources.

Figure 18 - 42 SNOOZE Mode Continued Operation in Response to a Compare Mismatch

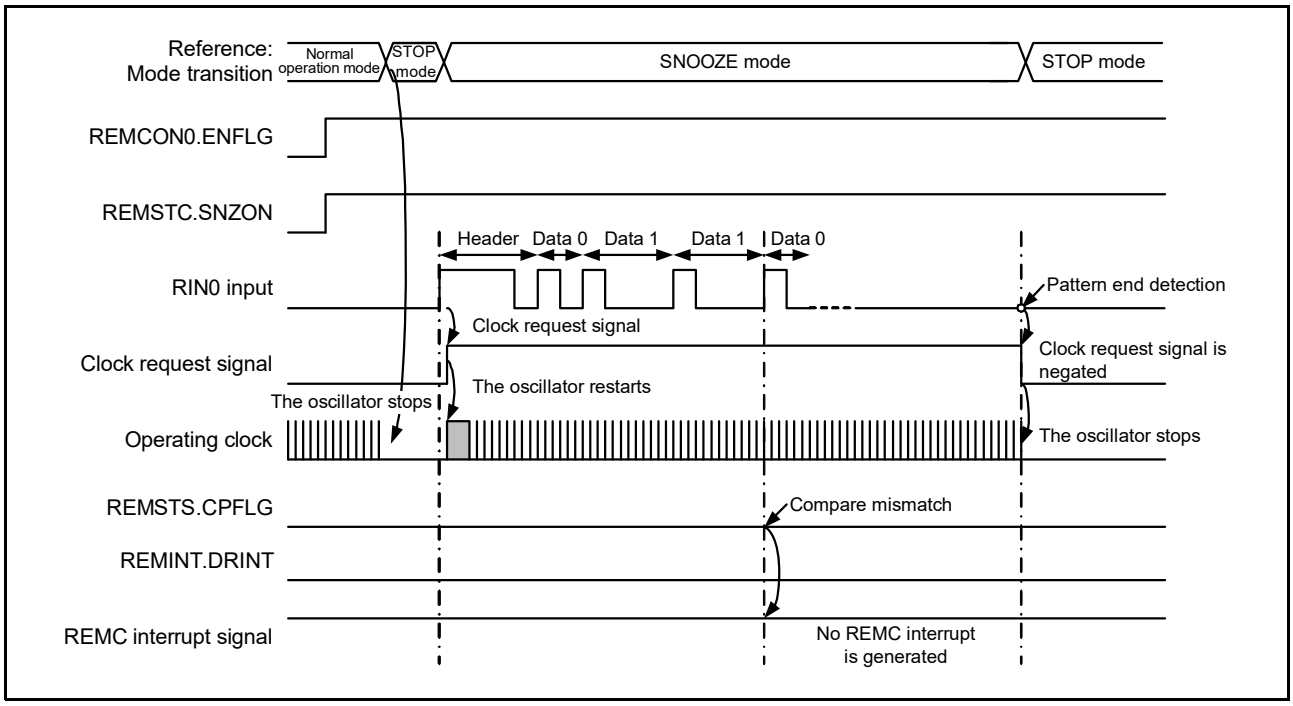


Figure 18 - 43 Returning from SNOOZE Mode in Response to a Compare Match

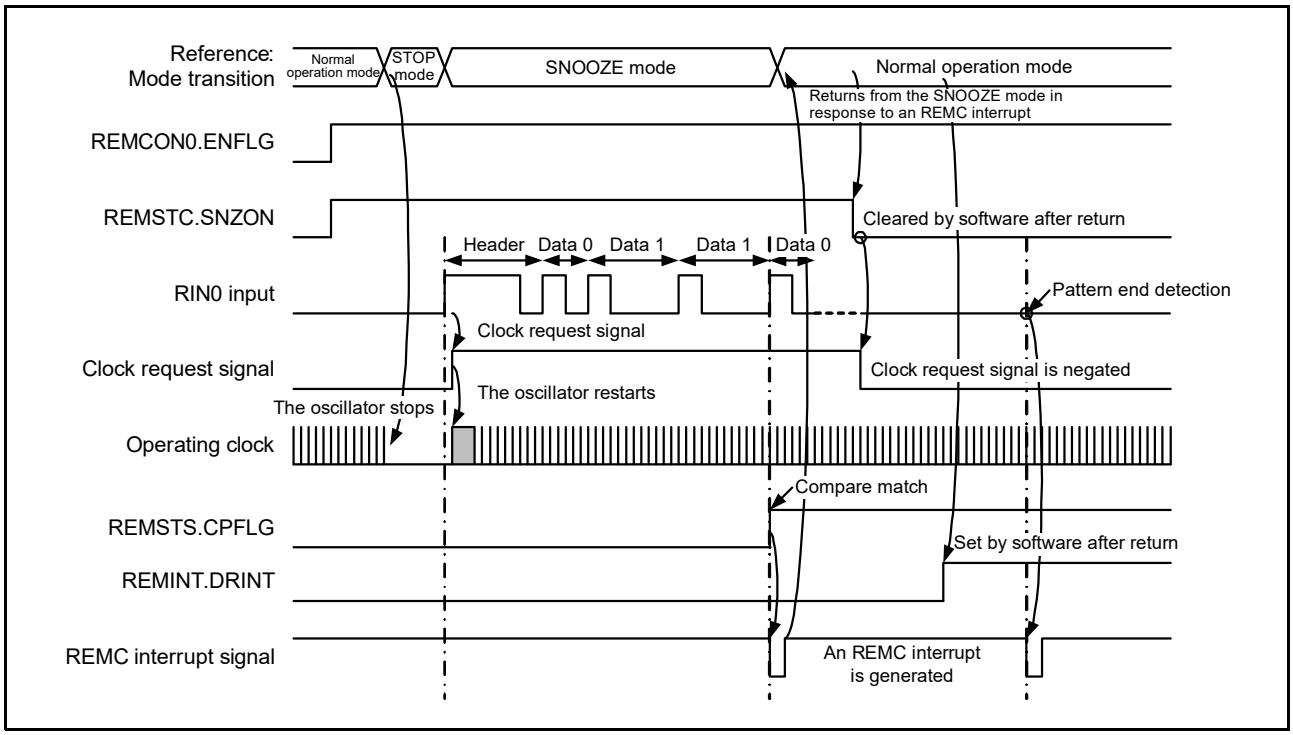
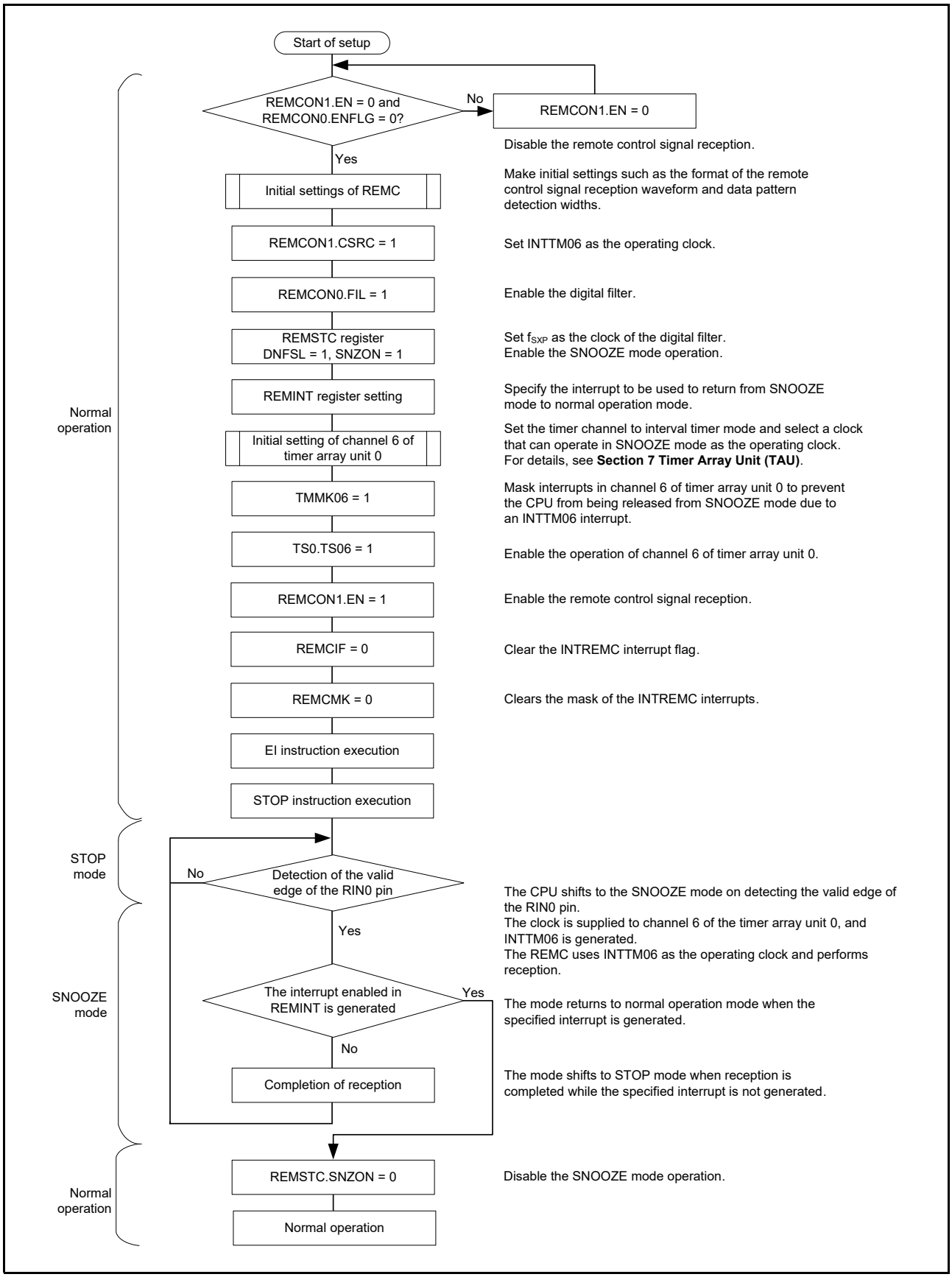


Figure 18 - 44 Example of Flowchart for Setting Up SNOOZE Mode



18.4 Points for Caution when the Remote Control Signal Receiver is to be Used

18.4.1 Access to registers when operation of the remote control signal receiver is enabled

The REMCON1.EN bit enables or disables operation of the remote control signal receiver. The REMCON0.ENFLG flag indicates whether or not the REMC is operating. After the REMCON1.EN bit is set to 1 (operation enabled), it takes up to zero to one cycle of the operating clock before the REMC circuit starts operating and the REMCON0.ENFLG flag becomes 1. During this period, do not access the REMC related registers (listed in **18.2.1** to **18.2.22**) except for the REMCON0.ENFLG flag.

18.4.2 Timing of changing the register values

Change the following registers only when the REMCON1.EN bit and REMCON0.ENFLG flag are both 0 (REMC operation is stopped).

- REMCON0 register
- REMCON1 register (except for bits 2 to 0)
- REMINT register (except for bits 5 and 2)
- REMCPC register
- REMCPD register
- Pattern width setting registers for header, data 0, data 1, and special data patterns
- Pattern end setting register
- REMSTC register

When rewriting the REMCON1.TYP[1:0] bits while the REMCON1.EN bit or REMCON0.ENFLG flag is 1 (REMC is operating), change the values of these bits one bit at a time. If the REMCON1.TYP[1:0] bits are rewritten when the REMCON0.INFLG flag changes, the signal captured into the remote control signal receiver may be undefined. After 0 is written to bit 0 in the REMDAT0 or REMRBIT register or the REMSTS.BFULFLG flag, do not write 0 to the same bit again for two cycles of the operating clock. If 0 is written when the REMCON0.INFLG flag changes, the values of the REMDATj and REMRBIT registers and the REMSTS.BFULFLG flag may be undefined.

18.4.3 RIN0 input control

If the REMCON0.FILSEL, FIL, or INV bit is rewritten, the signal captured into the remote control signal receiver is undefined for three cycles of the digital filter sampling clock.

18.4.4 Changing the operating clock

When the REMCON1.CSRC bit is rewritten, set the following registers again: REMCON0, REMCON1, REMINT, REMCPC, REMCPD, REMPE, and header, data 0, data 1, and special data pattern width setting registers.

18.4.5 Reading registers

Reading the data from the following registers at the timing when their values change may return an undefined value.

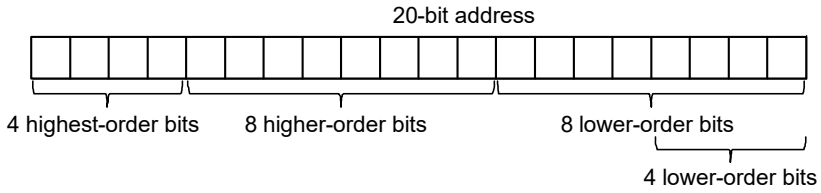
- All flags in the REMCON0 and REMSTS registers other than the REMSTS.DRFLG flag
- Registers REMTIM, REMDAT0 to REMDAT7, and REMRBIT

Follow the procedures below to avoid reading an undefined value.

- Using an interrupt
  Set the REMINT.DRINT bit to 1 (data reception complete interrupt enabled) and read the registers within the REMC interrupt routine.
- Polling by a program 1
  Set the REMINT.DRINT bit to 1 (data reception complete interrupt enabled) and poll the REMCIF flag by a program. Read the registers when the flag becomes 1 (interrupt request generated).
- Polling by a program 2
  1. Poll the REMSTS.DRFLG flag.
  2. When the REMSTS.DRFLG flag becomes 1, poll this flag until it becomes 0.
  3. Read the necessary content of the registers when the REMSTS.DRFLG flag becomes 0.

Section 19 Data Transfer Controller (DTC)

The term “8 higher-order bits of the address” in this section indicates bits 15 to 8 of 20-bit address as shown below.



Unless otherwise specified, the 4 highest-order address bits all become 1 (values are of the form FxxxxH).

19.1 Functions of DTC

The data transfer controller (DTC) transfers data between memory areas without using the CPU. The DTC is activated by a peripheral function interrupt and transfers data. The DTC and CPU use the same bus, and the DTC takes priority over the CPU in using the bus. **Table 19 - 1** lists the DTC specifications.

Table 19 - 1 DTC Specifications (1/2)

Item		Specification
Activation sources of the R7F100G products		30 sources (30- and 32-pin products)/32 sources (36-pin products)/33 sources (40-pin products)/35 sources (44-pin products)/36 sources (48- and 52-pin products)/37 sources (64-pin products)/39 sources (80- to 128-pin products)
Activation sources of the R7F104G products		31 sources (32-, 44-, and 48-pin products)
Allocatable control data		24 sets
Address space available for use with DTC transfer	Address space	64 Kbytes (F0000H to FFFFFH), excluding general-purpose registers
	Source	Special function registers (SFRs), RAM area (excluding general-purpose registers)Note 1, mirror areaNote 2, data flash memory areaNote 2, extended special function registers (2nd SFRs)
	Destination	Special function registers (SFRs), RAM area (excluding general-purpose registers), extended special function registers (2nd SFRs)
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
	Normal mode (16-bit transfer)	512 bytes
	Repeat mode	255 bytes
Unit of transfers		8 bits/16 bits
Transfer mode	Normal mode	Transfers end on completion of the transfer causing the DTCCTj register value to change from 1 to 0.
	Repeat mode	On completion of the transfer causing the DTCCTj register value to change from 1 to 0, the repeat area address is initialized and the DTRLDj register value is reloaded to the DTCCTj register to continue transfers.
Address control	Normal mode	Fixed or incremented
	Repeat mode	Addresses of the area not selected as the repeat area are fixed or incremented.
Priority of activation sources		Refer to **Table 19 - 3 DTC Activation Sources and Vector Addresses**.

Table 19 - 1 DTC Specifications (2/2)

Item		Specification
Interrupt request	Normal mode	When the data transfer causing the DTCCTj register 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 DTCCTj register value to change from 1 to 0 is performed while the RPTINT bit in the DTCCRj register is 1 (interrupt generation enabled), the activation source interrupt request is generated for the CPU, and interrupt handling is performed on completion of the data transfer.
Transfer start		When bits DTCENi7 to DTCENi0 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 DTCENi7 to DTCENi0 are set to 0 (activation disabled). When the data transfer causing the DTCCTj register value to change from 1 to 0 is completed.
	Repeat mode	When bits DTCENi7 to DTCENi0 are set to 0 (activation disabled). When the data transfer causing the DTCCTj register value to change from 1 to 0 is completed while the RPTINT bit is 1 (interrupt generation enabled).

Note 1. If the internal RAM is placed in standby mode or shutdown mode (RAMSDMD = 1) by the memory power reduction control register (PSMCR), the internal RAM cannot be set as a DTC transfer source or destination.

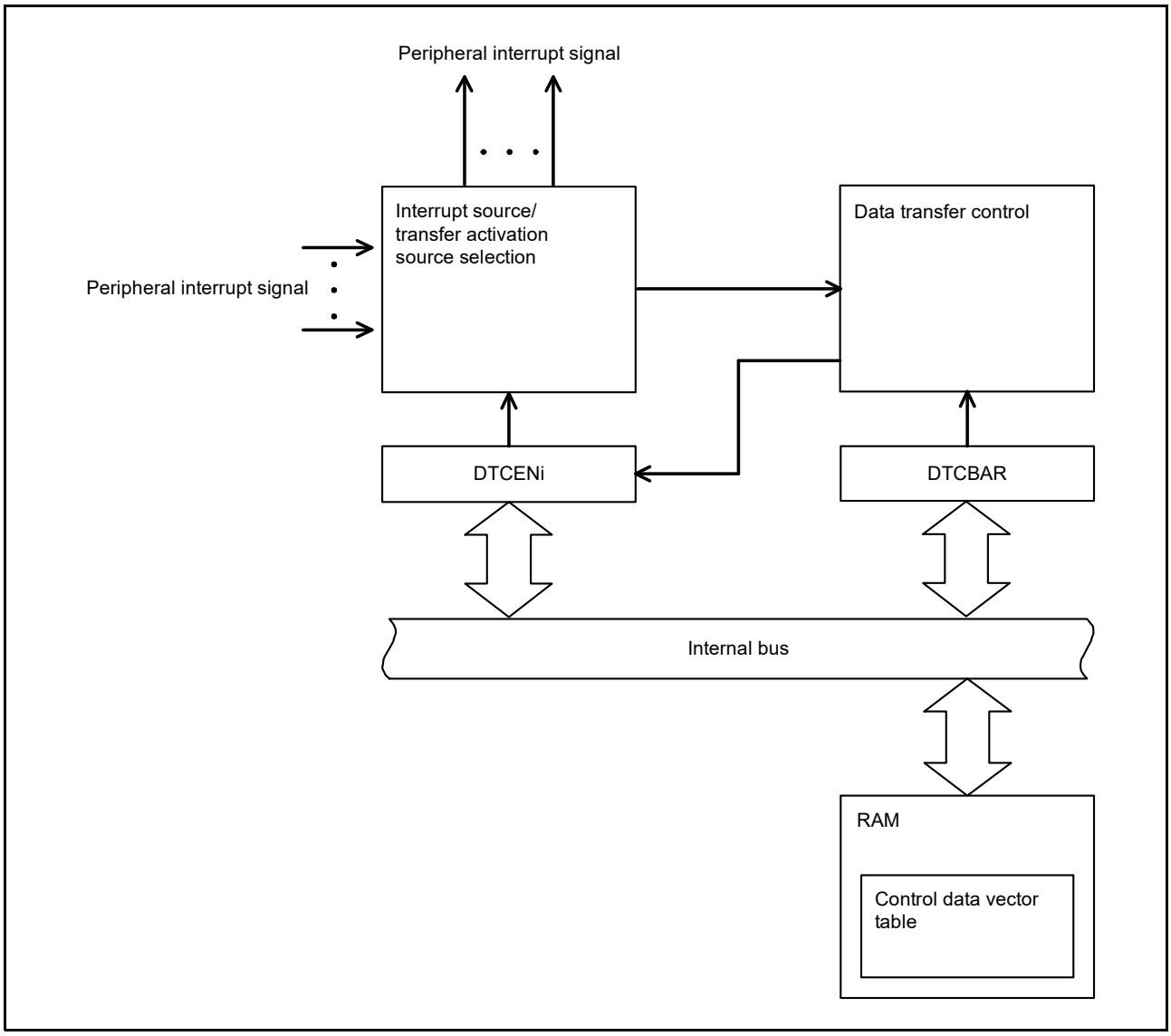
Note 2. In the HALT mode or SNOOZE mode, these areas cannot be set as the sources for DTC transfer since the flash memory is stopped. If the internal RAM is placed in standby mode or shutdown mode (RAMSDMD = 1) by the memory power reduction control register (PSMCR), the internal RAM cannot be set as a DTC transfer source or destination.

Remark i = 0 to 4, j = 0 to 23

19.2 Configuration of DTC

Figure 19 - 1 shows the DTC block diagram.

Figure 19 - 1 DTC Block Diagram



Remark i = 0 to 4

19.3 Registers for Controlling the DTC

The following registers are used to control the DTC.

- Peripheral enable register 1 (PER1)
- DTC activation enable register i (DTCENi)
- DTC base address register (DTCBAR)

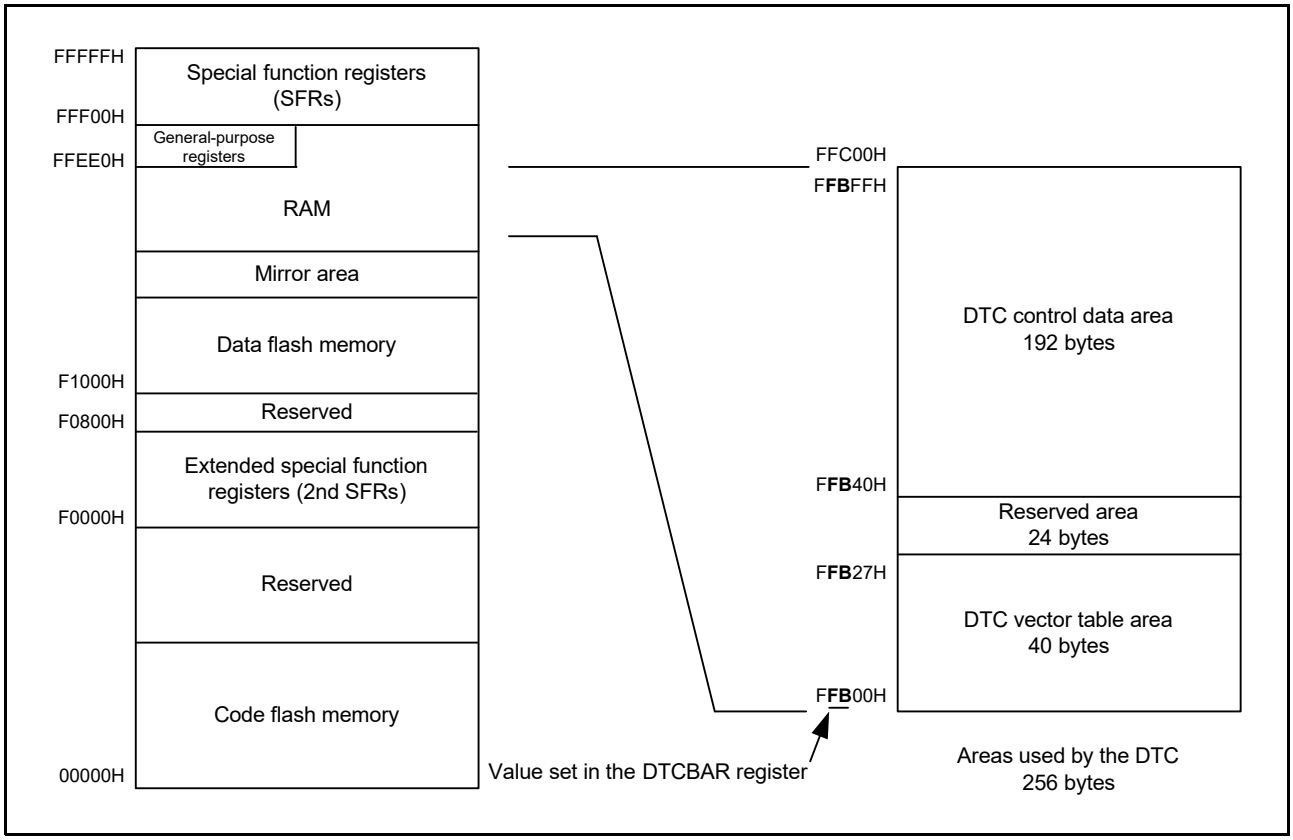
The DTC control data are listed below. DTC control data are allocated in the DTC control data area in RAM. The DTCBAR register is used to set the 256-byte area, including the DTC control data area and the DTC vector table area where the start address for control data is stored.

- 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)

19.3.1 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 are allocated within the RAM area. **Figure 19 - 2** shows the example of the memory map when the DTCBAR register is set to FBH. In the 192-byte DTC control data area, the space not used by the DTC can be used as RAM.

Figure 19 - 2 Example of the Memory Map When the DTCBAR Register is Set to FBH



The areas where the DTC control data and vector table can be allocated differ depending on the product.

Caution 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.

Caution 2. Make sure the stack area, the DTC control data area, and the DTC vector table area do not overlap.

Caution 3. The internal RAM area in the following products cannot be used as the DTC control data area or DTC vector table area when the tracing function of the on-chip debugging emulator is in use.
R7F100GxG (x = A, B, C, E, F, G, J, L, M, P) and R7F104GBG: FC300H to FC6FFH
R7F100GxJ (x = A, B, C, E, F, G, J, L, M, P, S), R7F104GFJ, and R7F104GGJ: FA300H to FA6FFH
R7F100GxL and R7F100GxN (x = F, G, J, L, M, P, S): F4300H to F46FFH

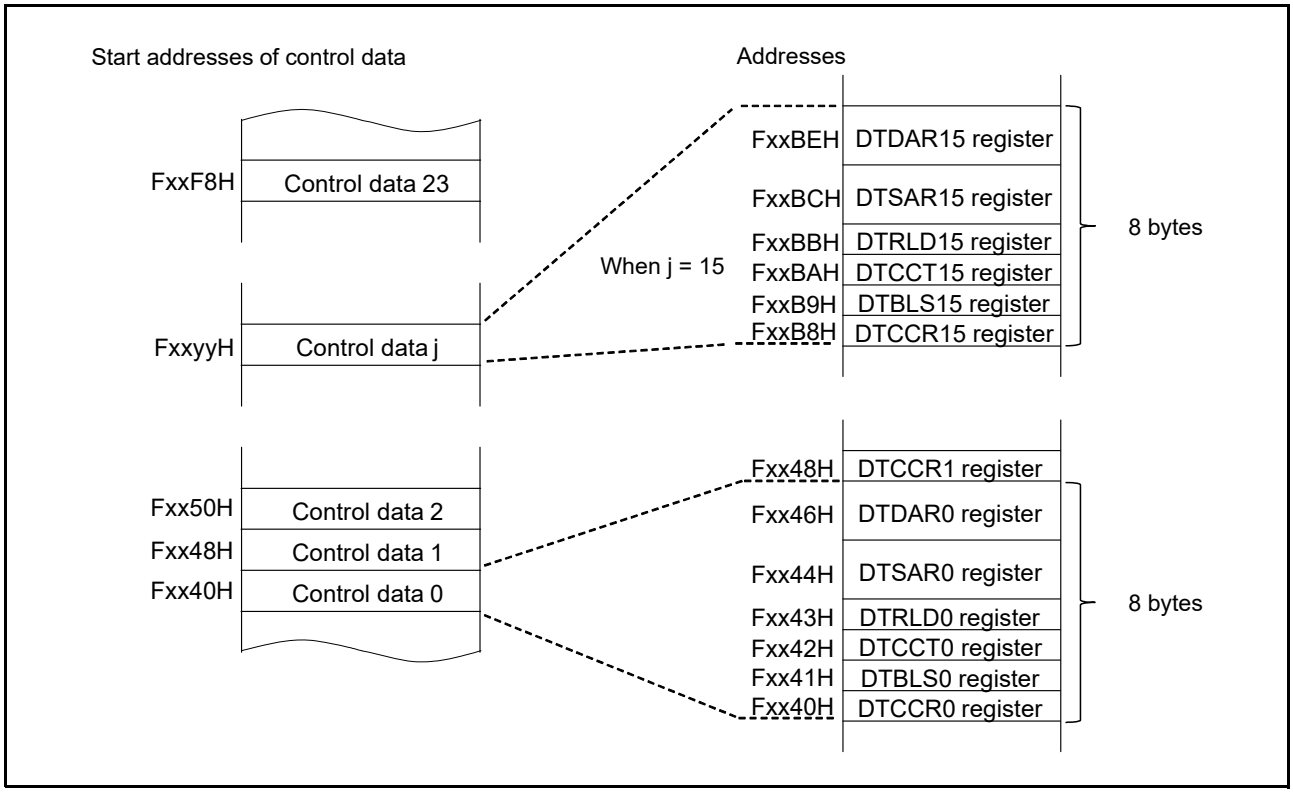
19.3.2 Control data allocation

Control data are allocated beginning with each start address in the order: the DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, and DTDARj registers (j = 0 to 23). The 8 higher-order bits for start addresses 0 to 23 are set by the DTCBAR register, and the 8 lower-order bits are separately set according to the vector table assigned to each activation source. **Figure 19 - 3** shows the control data allocation.

Caution 1. Change the data in registers DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, and DTDARj when the corresponding bit among bits DTCENi7 to DTCENi0 in the DTCENi register (i = 0 to 4) is set to 0 (activation disabled).

Caution 2. Do not access the DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, or DTDARj register using a DTC transfer.

Figure 19 - 3 Control Data Allocation



Remark xx: Value set in the DTCBAR register

Table 19 - 2 Start Addresses of Control Data j

j	Address	j	Address
11	Fxx98H	23	FxxF8H
10	Fxx90H	22	FxxF0H
9	Fxx88H	21	FxxE8H
8	Fxx80H	20	FxxE0H
7	Fxx78H	19	FxxD8H
6	Fxx70H	18	FxxD0H
5	Fxx68H	17	FxxC8H
4	Fxx60H	16	FxxC0H
3	Fxx58H	15	FxxB8H
2	Fxx50H	14	FxxB0H
1	Fxx48H	13	FxxA8H
0	Fxx40H	12	FxxA0H

Remark xx: Value set in the DTCBAR register

19.3.3 Vector table

When the DTC is activated, one control data set is selected according to the data read from the vector table which has been assigned to each activation source, and the selected control data are read from the DTC control data area. **Table 19 - 3** lists the DTC activation sources and vector addresses. One byte of the vector table is assigned to each activation source, and the offset values from 40H to F8H at which each set of control data starts are stored in each of the locations to select the corresponding one from among the 24 control data sets. The 8 higher-order bits for the vector address are set by the DTCBAR register, and 00H to 27H are allocated to the 8 lower-order bits corresponding to the 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 DTCENi7 to DTCENi0 in the DTCENi register (i = 0 to 4) is set to 0 (activation disabled).

Figure 19 - 4 Start Addresses of Control Data and Vector Table

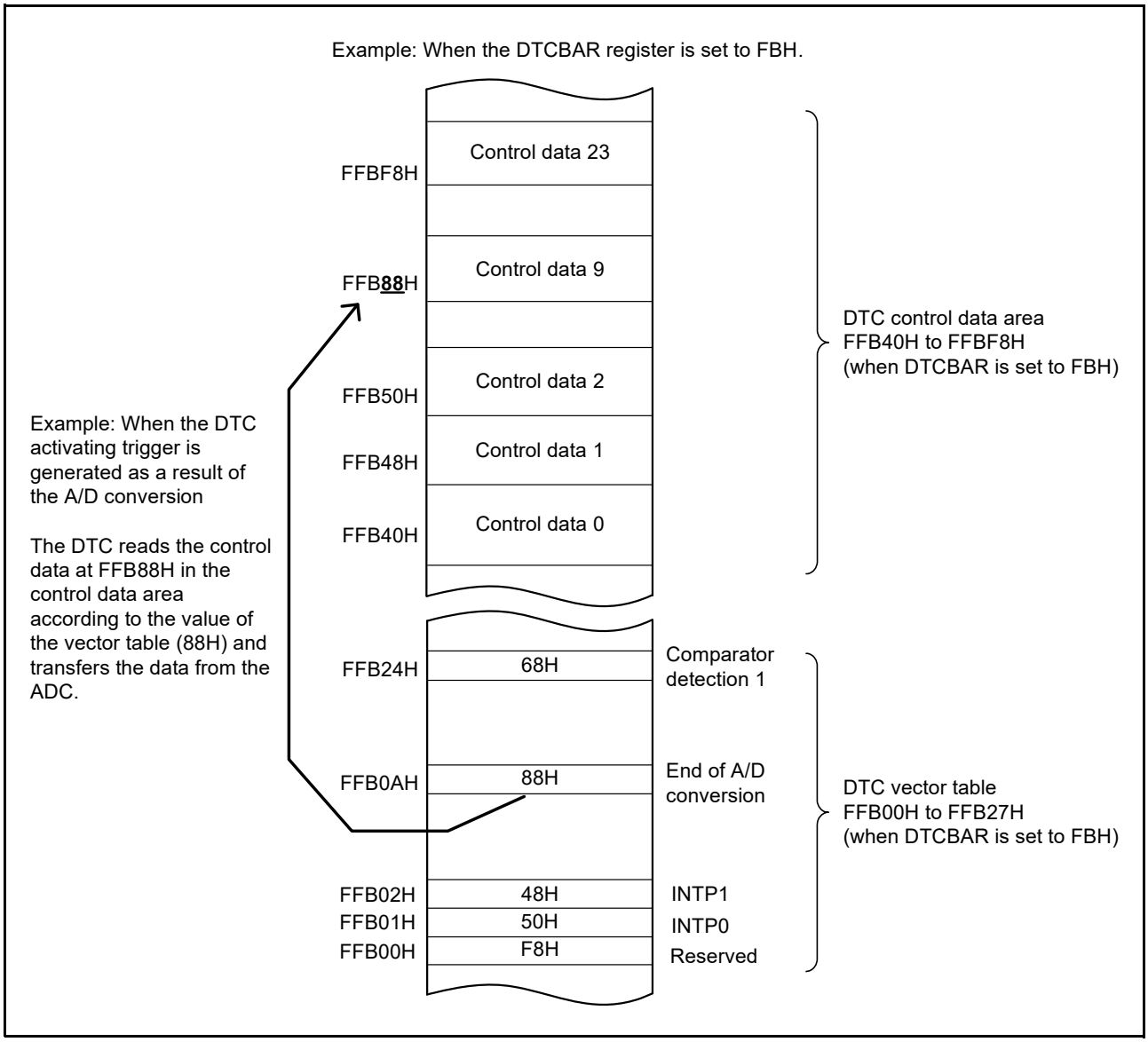


Table 19 - 3 DTC Activation Sources and Vector Addresses

DTC Activation Source (Interrupt Request Source)	Source No.	Vector Address	Priority
Reserved	0	Address set in the DTCBAR register + 00H	Highest
INTP0	1	Address set in the DTCBAR register + 01H	
INTP1	2	Address set in the DTCBAR register + 02H	
INTP2Note 1	3	Address set in the DTCBAR register + 03H	
INTP3	4	Address set in the DTCBAR register + 04H	
INTP4	5	Address set in the DTCBAR register + 05H	
INTP5	6	Address set in the DTCBAR register + 06H	
INTP6Note 2	7	Address set in the DTCBAR register + 07H	
INTP7Note 3	8	Address set in the DTCBAR register + 08H	
Key inputNote 4	9	Address set in the DTCBAR register + 09H	
A/D conversion end	10	Address set in the DTCBAR register + 0AH	
UART0 reception transfer end/CSI01 transfer end or buffer empty/IIC01 transfer end	11	Address set in the DTCBAR register + 0BH	
UART0 transmission transfer end/CSI00 transfer end or buffer empty/IIC00 transfer end	12	Address set in the DTCBAR register + 0CH	
UART1 reception transfer end/CSI11 transfer end or buffer empty/IIC11 transfer end	13	Address set in the DTCBAR register + 0DH	
UART1 transmission transfer end/CSI10 transfer end or buffer empty/IIC10 transfer end	14	Address set in the DTCBAR register + 0EH	
UART2 reception transfer end/CSI21 transfer end or buffer empty/IIC21 transfer end	15	Address set in the DTCBAR register + 0FH	
UART2 transmission transfer end/CSI20 transfer end or buffer empty/IIC20 transfer end	16	Address set in the DTCBAR register + 10H	
UART3 reception transfer end/CSI31 transfer end or buffer empty/IIC31 transfer endNote 5	17	Address set in the DTCBAR register + 11H	
UART3 transmission transfer end/CSI30 transfer end or buffer empty/IIC30 transfer endNote 5	18	Address set in the DTCBAR register + 12H	
UARTA0 reception transfer endNote 6	19	Address set in the DTCBAR register + 13H	
UARTA0 transmission transfer end/buffer emptyNote 6	20	Address set in the DTCBAR register + 14H	
UARTA1 reception transfer endNote 7	21	Address set in the DTCBAR register + 15H	
UARTA1 transmission transfer end/buffer emptyNote 7	22	Address set in the DTCBAR register + 16H	
End of counting or capturing by channel 0 of timer array unit 0	23	Address set in the DTCBAR register + 17H	
End of counting or capturing by channel 1 of timer array unit 0	24	Address set in the DTCBAR register + 18H	
End of counting or capturing by channel 2 of timer array unit 0	25	Address set in the DTCBAR register + 19H	
End of counting or capturing by channel 3 of timer array unit 0	26	Address set in the DTCBAR register + 1AH	
End of counting or capturing by channel 4 of timer array unit 0	27	Address set in the DTCBAR register + 1BH	
End of counting or capturing by channel 5 of timer array unit 0	28	Address set in the DTCBAR register + 1CH	
End of counting or capturing by channel 6 of timer array unit 0	29	Address set in the DTCBAR register + 1DH	
End of counting or capturing by channel 7 of timer array unit 0	30	Address set in the DTCBAR register + 1EH	
Fixed-cycle signal of realtime clock/alarm match detection	31	Address set in the DTCBAR register + 1FH	
Interval signal detection of 32-bit interval timer	32	Address set in the DTCBAR register + 20H	
Request to write to a configuration register of an individual channel of the capacitive sensing unit	33	Address set in the DTCBAR register + 21H	
Request to transfer data measured by the capacitive sensing unit	34	Address set in the DTCBAR register + 22H	
Comparator detection 0	35	Address set in the DTCBAR register + 23H	
Comparator detection 1Note 1	36	Address set in the DTCBAR register + 24H	
Event output from the logic and event link controller	37	Address set in the DTCBAR register + 25H	
Event output from the SNOOZE mode sequencer	38	Address set in the DTCBAR register + 26H	
Voltage detectionNote 8	39	Address set in the DTCBAR register + 27H	Lowest

Note 1. This is applicable to the 30- to 128-pin R7F100G products.
Note 2. This is applicable to the 48- to 128-pin R7F100G products.
Note 3. This is applicable to the 64- to 128-pin R7F100G products.
Note 4. This is applicable to the 40- to 128-pin R7F100G and 32- to 48-pin R7F104G products.
Note 5. This is applicable to the 80- to 128-pin R7F100G products.
Note 6. This is applicable to the 36- to 128-pin R7F100G and 32- to 48-pin R7F104G products.
Note 7. This is applicable to the 44- to 128-pin R7F100G products.
Note 8. When bit 6 (LVD0SEL) of the user option byte (000C1H) is set to 0 or when bit 6 (LVD1SEL) of the voltage detection level register (LVIS) is set to 0

19.3.4 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. If the DTC is to be used, be sure to set the DTCEN bit to 1. The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 19 - 5 Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER1	DACEN	SMSEN	CMPEN	TML32EN	DTCEN	UTAEN **Note 1**	REMCEN **Note 2**	CTSUEN

DTCEN	Control of supply of an input clock to the DTC
0	Stops supply of an input clock. • The SFRs used by the DTC cannot be written.
1	Enables supply of an input clock. • The DTC can operate.

Note 1. This bit is only present in the 36- to 128-pin R7F100G and 32- to 48-pin R7F104G products.
Note 2. This bit is only present in the 32- to 128-pin R7F100G and 32- to 48-pin R7F104G products.

Caution Be sure to leave the following bits set to 0.
Bits 2 and 1 in the 30-pin R7F100G products
Bit 2 in the 32-pin R7F100G products

19.3.5 DTC control register j (DTCCRj)

The DTCCRj register is used to control the DTC operating mode.

Remark j = 0 to 23

Figure 19 - 6 Format of DTC Control Register j (DTCCRj) (1/2)

Address: Refer to **19.3.2 Control data allocation**.
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
DTCCRj	0	SZ	RPTINT	CHNE	DAMOD	SAMOD	RPTSEL	MODE

SZ	Transfer data size selection
0	8 bits
1	16 bits

RPTINT	Enabling/disabling repeat mode interrupts
0	Interrupt generation disabled
1	Interrupt generation enabled
The setting of this bit is invalid when the MODE bit is 0 (normal mode).	

CHNE	Enabling/disabling chain transfers
0	Chain transfers disabled
1	Chain transfers enabled
Set the CHNE bit in the DTCCR23 register to 0 (chain transfers disabled).	

DAMOD	Transfer destination address control
0	Fixed
1	Incremented
The setting of this 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
0	Fixed
1	Incremented
The setting of this 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
0	Transfer destination is the repeat area.
1	Transfer source is the repeat area.
The setting of this bit is invalid when the MODE bit is 0 (normal mode).	

Figure 19 - 6 Format of DTC Control Register j (DTCCRj) (2/2)

MODE	Transfer mode selection
0	Normal mode
1	Repeat mode

Caution **Do not access the DTCCRj register using a DTC transfer.**

Remark j = 0 to 23

19.3.6 DTC block size register j (DTBLSj)

This register is used to set the block size of the data to be transferred by one activation.

Figure 19 - 7 Format of DTC Block Size Register j (DTBLSj)

Address: Refer to **19.3.2 Control data allocation**.
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
DTBLSj	DTBLSj7	DTBLSj6	DTBLSj5	DTBLSj4	DTBLSj3	DTBLSj2	DTBLSj1	DTBLSj0

DTBLSj	Transfer Block Size	
	8-bit Transfer	16-bit Transfer
00H	256 bytes	512 bytes
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.

Remark j = 0 to 23

19.3.7 DTC transfer count register j (DTCCTj)

This register is used to set the number of DTC data transfers. The value is decremented by 1 each time DTC transfer is activated once.

Figure 19 - 8 Format of DTC Transfer Count Register j (DTCCTj)

Address: Refer to **19.3.2 Control data allocation**.
After reset: Undefined
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
DTCCTj	DTCCTj7	DTCCTj6	DTCCTj5	DTCCTj4	DTCCTj3	DTCCTj2	DTCCTj1	DTCCTj0

DTCCTj	Number of Transfers
00H	256 times
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.

Remark j = 0 to 23

19.3.8 DTC transfer count reload register j (DTRLDj)

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 DTCCTj register in repeat mode, set the same value as the initial value of the DTCCTj register.

Figure 19 - 9 Format of DTC Transfer Count Reload Register j (DTRLDj)

Address: Refer to **19.3.2 Control data allocation**.
After reset: Undefined
R/W: R/W

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.

Remark j = 0 to 23

19.3.9 DTC source address register j (DTSARj)

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 (the transfer size is 16 bits), the lowest-order bit is ignored and the address is handled as an even address.

Figure 19 - 10 Format of DTC Source Address Register j (DTSARj)

Address: Refer to **19.3.2 Control data allocation**.
After reset: Undefined
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
DTSARj	DTSARj15	DTSARj14	DTSARj13	DTSARj12	DTSARj11	DTSARj10	DTSARj9	DTSARj8

7	6	5	4	3	2	1	0
DTSARj7	DTSARj6	DTSARj5	DTSARj4	DTSARj3	DTSARj2	DTSARj1	DTSARj0

Caution 1. Do not set the general-purpose register (FFEE0H to FFEFFH) space to the transfer source address.
Caution 2. Do not access the DTSARj register using a DTC transfer.

Remark j = 0 to 23

19.3.10 DTC destination address register j (DTDARj)

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 (the transfer size is 16 bits), the lowest-order bit is ignored and the address is handled as an even address.

Figure 19 - 11 Format of DTC Destination Address Register j (DTDARj)

Address: Refer to **19.3.2 Control data allocation**.
After reset: Undefined
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
DTDARj	DTDARj15	DTDARj14	DTDARj13	DTDARj12	DTDARj11	DTDARj10	DTDARj9	DTDARj8

7	6	5	4	3	2	1	0
DTDARj7	DTDARj6	DTDARj5	DTDARj4	DTDARj3	DTDARj2	DTDARj1	DTDARj0

Caution 1. Do not set the general-purpose register (FFEE0H to FFEFFH) space to the transfer destination address.
Caution 2. Do not access the DTDARj register using a DTC transfer.

Remark j = 0 to 23

19.3.11 DTC activation enable register i (DTCENi)

This is an 8-bit register which enables or disables DTC activation by interrupt sources. **Table 19 - 4** lists the correspondences between interrupt sources and bits DTCENi7 to DTCENi0. The DTCENi register can be set by a 1-bit or 8-bit memory manipulation instruction.

Caution 1. Modify bits DTCENi7 to DTCENi0 if an activation source corresponding to the bit has not been generated.

Caution 2. Do not access the DTCENi register using a DTC transfer.

Caution 3. The assigned functions differ depending on the product. For the bits to which no functions are assigned, be sure to set their values to 0.

Remark i = 0 to 4

Figure 19 - 12 Format of DTC Activation Enable Register i (DTCENi) (1/2)

Address: F02E8H (DTCEN0), F02E9H (DTCEN1), F02EAH (DTCEN2), F02EBH (DTCEN3), F02ECH (DTCEN4)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
DTCENi	DTCENi7	DTCENi6	DTCENi5	DTCENi4	DTCENi3	DTCENi2	DTCENi1	DTCENi0

DTCENi7	DTC activation enable i7
0	Activation disabled
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
0	Activation disabled
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
0	Activation disabled
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
0	Activation disabled
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
0	Activation disabled
1	Activation enabled
The DTCENi3 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

Figure 19 - 12 Format of DTC Activation Enable Register i (DTCENi) (2/2)

DTCENi2	DTC activation enable i2
0	Activation disabled
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
0	Activation disabled
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
0	Activation disabled
1	Activation enabled
The DTCENi0 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

Remark i = 0 to 4

Table 19 - 4 Correspondences between Interrupt Sources and Bits DTCENi7 to DTCENi0

Register	DTCENi7 Bit	DTCENi6 Bit	DTCENi5 Bit	DTCENi4 Bit	DTCENi3 Bit	DTCENi2 Bit	DTCENi1 Bit	DTCENi0 Bit
DTCEN0	Reserved	INTP0	INTP1	INTP2Note 1	INTP3	INTP4	INTP5	INTP6Note 2
DTCEN1	INTP7Note 3	Key inputNote 4	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	UART2 reception transfer end/ CSI21 transfer end or buffer empty/ IIC21 transfer end
DTCEN2	UART2 transmission transfer end/ CSI20 transfer end or buffer empty/ IIC20 transfer end	UART3 reception transfer end/ CSI31 transfer end or buffer empty/ IIC31 transfer endNote 5	UART3 transmission transfer end/ CSI30 transfer end or buffer empty/ IIC30 transfer end Note 5	UARTA0 reception transfer endNote 6	UARTA0 transmission transfer end/ buffer emptyNote 6	UARTA1 reception transfer endNote 7	UARTA1 transmission transfer end/ buffer emptyNote 7	End of counting or capturing by channel 0 of timer array unit 0
DTCEN3	End of counting or capturing by channel 1 of timer array unit 0	End of counting or capturing by channel 2 of timer array unit 0	End of counting or capturing by channel 3 of timer array unit 0	End of counting or capturing by channel 4 of timer array unit 0	End of counting or capturing by channel 5 of timer array unit 0	End of counting or capturing by channel 6 of timer array unit 0	End of counting or capturing by channel 7 of timer array unit 0	Fixed-cycle signal of real-time clock/alarm match detection
DTCEN4	Interval signal detection of 32-bit interval timer	Request to write to a configuration register of an individual channel of the capacitive sensing unit	Request to transfer data measured by the capacitive sensing unit	Comparator detection 0	Comparator detection 1 Note 1	Event output from the logic and event link controller	Event output from the SNOOZE mode sequencer	Voltage detectionNote 8

Note 1. This is applicable to the 30- to 128-pin R7F100G products.

Note 2. This is applicable to the 48- to 128-pin R7F100G products.

Note 3. This is applicable to the 64- to 128-pin R7F100G products.

Note 4. This is applicable to the 40- to 128-pin R7F100G and 32- to 48-pin R7F104G products.

Note 5. This is applicable to the 80- to 128-pin R7F100G products.

Note 6. This is applicable to the 36- to 128-pin R7F100G and 32- to 48-pin R7F104G products.

Note 7. This is applicable to the 44- to 128-pin R7F100G products.

Note 8. When bit 6 (LVD0SEL) of the user option byte (000C1H) is set to 0 or when bit 6 (LVD1SEL) of the voltage detection level register (LVIS) is set to 0

Caution **For the bits to which no functions are assigned, be sure to set their values to 0.**

Remark i = 0 to 4

19.3.12 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 8 higher-order bits to generate a 16-bit address. The DTCBAR register can be set by an 8-bit memory manipulation instruction.

Caution 1. Change the DTCBAR register value with all DTC activation sources set to activation disabled.
Caution 2. Do not rewrite the DTCBAR register more than once.
Caution 3. Do not access the DTCBAR register using a DTC transfer.
Caution 4. For the allocation of the DTC control data area and the DTC vector table area, refer to the cautions in 19.3.1 Allocation of DTC control data area and DTC vector table area.

Figure 19 - 13 Format of DTC Base Address Register (DTCBAR)

Address: F02E0H
After reset: FDH
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
DTCBAR	DTCBAR7	DTCBAR6	DTCBAR5	DTCBAR4	DTCBAR3	DTCBAR2	DTCBAR1	DTCBAR0

19.4 DTC Operation

When the DTC is activated, the DTC reads control data from the DTC control data area, proceeds with data transfer according to the control data, and writes back the control data after data transfer 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. The normal mode and repeat mode are available as transfer mode, with selectable transfer size, 8 bits or 16 bits. When the CHNE bit in the DTCCRj register (j = 0 to 22) is set to 1 (chain transfers enabled), multiple control data sets are 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.

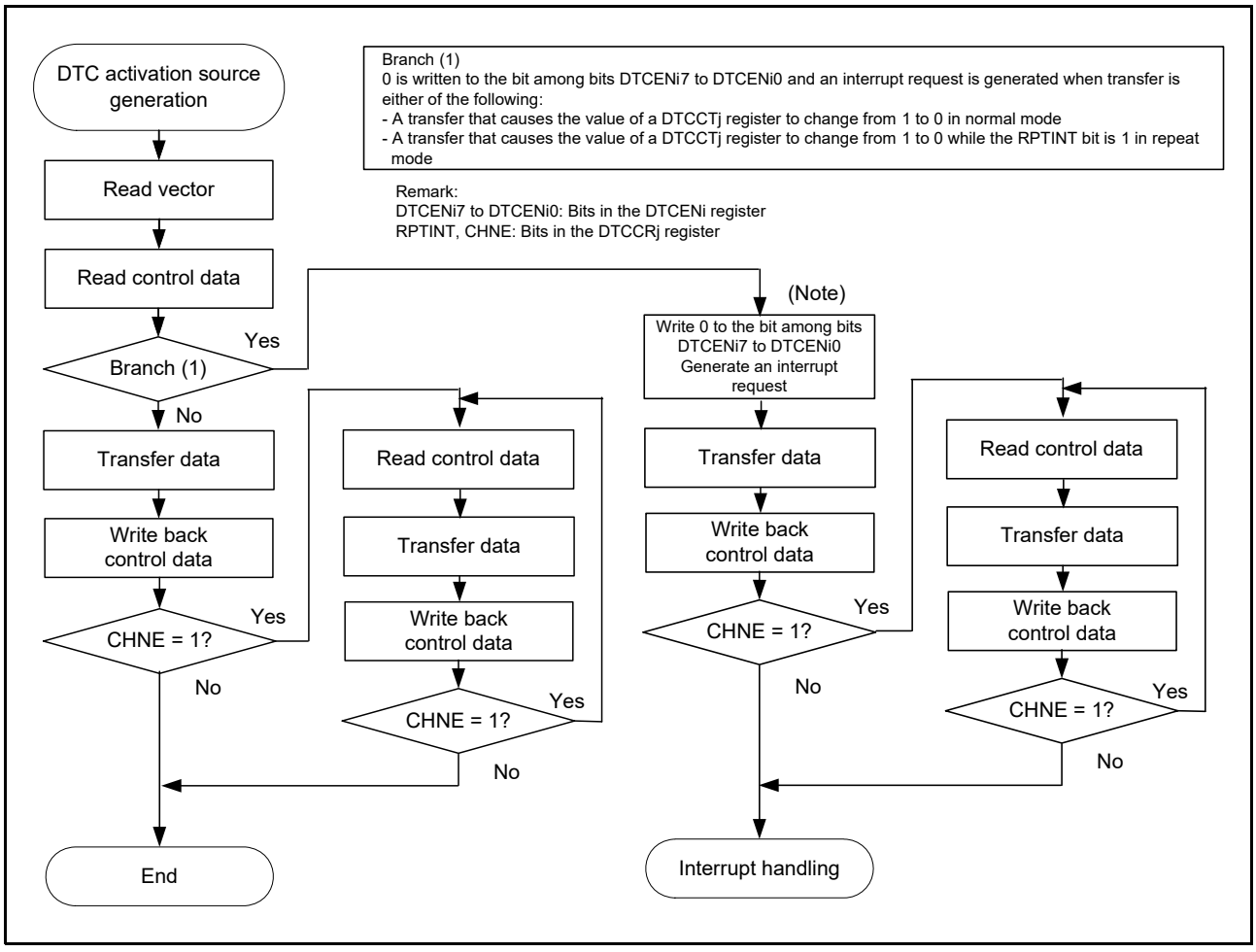
19.4.1 Activation sources

The DTC is activated by an interrupt signal from the peripheral functions. The interrupt signals to activate the DTC are selected with the DTCENi register (i = 0 to 4). The DTC sets the corresponding bit among bits DTCENi7 to DTCENi0 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 value of a DTCCTj register (j = 0 to 23) to change to 0 in normal mode
- A transfer that causes the value of a DTCCTj register to change to 0 while the RPTINT bit in the DTCCRj register is 1 (interrupt generation enabled) in repeat mode

Figure 19 - 14 shows the DTC internal operation flowchart.

Figure 19 - 14 DTC Internal Operation Flowchart



Note 0 is not written to the bit among bits DTCENi7 to DTCENi0 for data transfers activated by the setting to enable chain transfers (the CHNE bit is 1). Also, no interrupt request is generated.

Remark i = 0 to 4, j = 0 to 23

19.4.2 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 value of a DTCCTj register (j = 0 to 23) 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 DTCENi7 to DTCENi0 in the DTCENi register (i = 0 to 4) to 0 (activation disabled). **Table 19 - 5** lists the register functions in normal mode, and **Figure 19 - 15** shows the data transfers in normal mode.

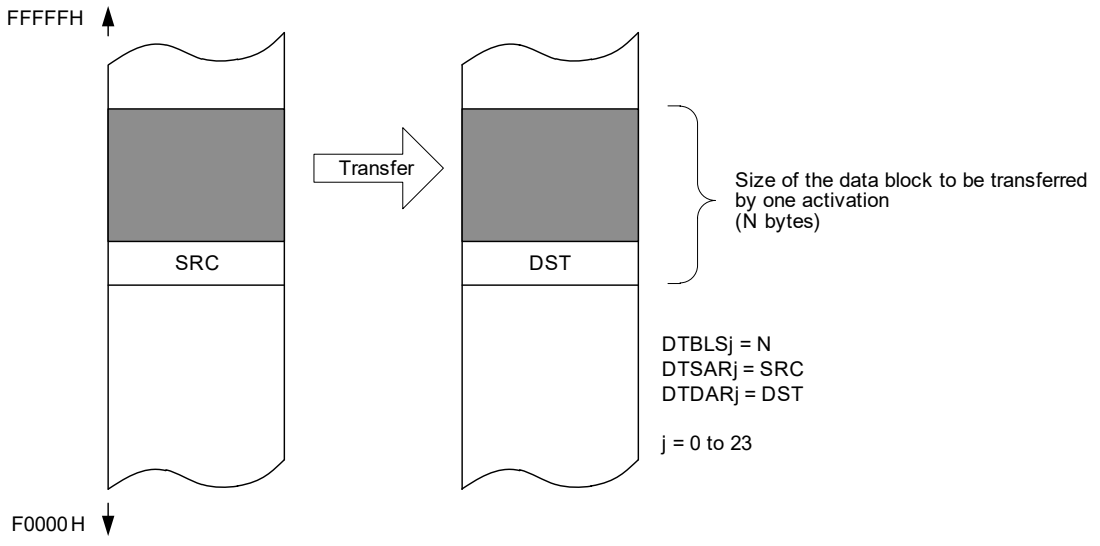
Table 19 - 5 Register Functions in Normal Mode

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 this register to 00H when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.

Remark j = 0 to 23

Figure 19 - 15 Data Transfers in Normal Mode

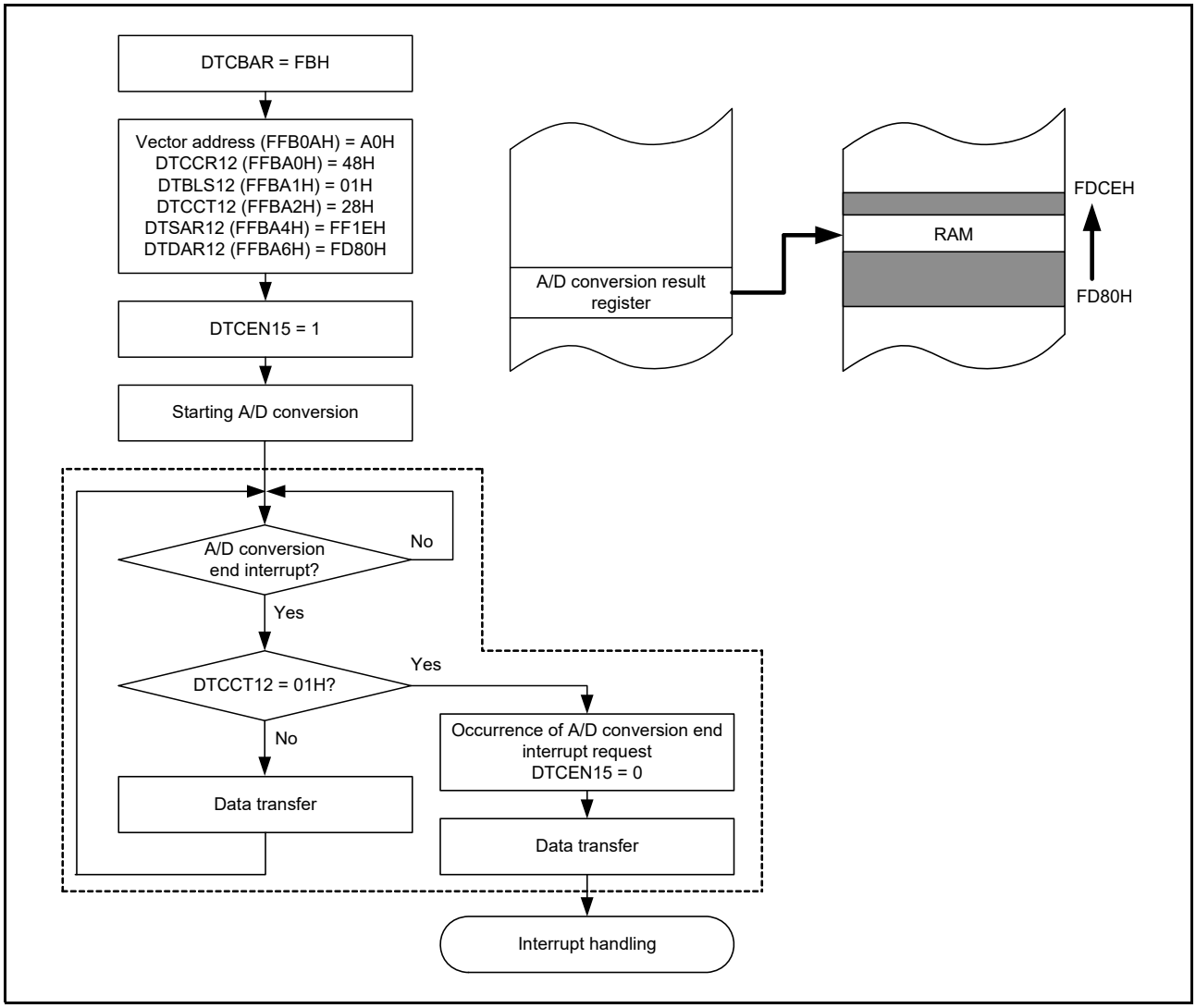


DTCCRj Register Setting				Source Address Control	Destination Address Control	Source Address after Transfer	Destination Address after Transfer
DAMOD	SAMOD	RPTSEL	MODE				
0	0	×	0	Fixed	Fixed	SRC	DST
0	1	×	0	Incremented	Fixed	SRC + N	DST
1	0	×	0	Fixed	Incremented	SRC	DST + N
1	1	×	0	Incremented	Incremented	SRC + N	DST + N

×: 0 or 1

1. Example 1 of using normal mode: Consecutively capturing A/D conversion results
   The DTC is activated by an A/D conversion end interrupt and the value of the A/D conversion result register is transferred to RAM.
   - The vector address is FFB0AH and control data are allocated at FFBA0H to FFBA7H.
   - Transfers 2-byte data of the A/D conversion result register (FFF1EH, FFF1FH) to 80 bytes of FFD80H to FFDCFH of RAM 40 times.

Figure 19 - 16 Example 1 of Using Normal Mode: Consecutively Capturing A/D Conversion Results

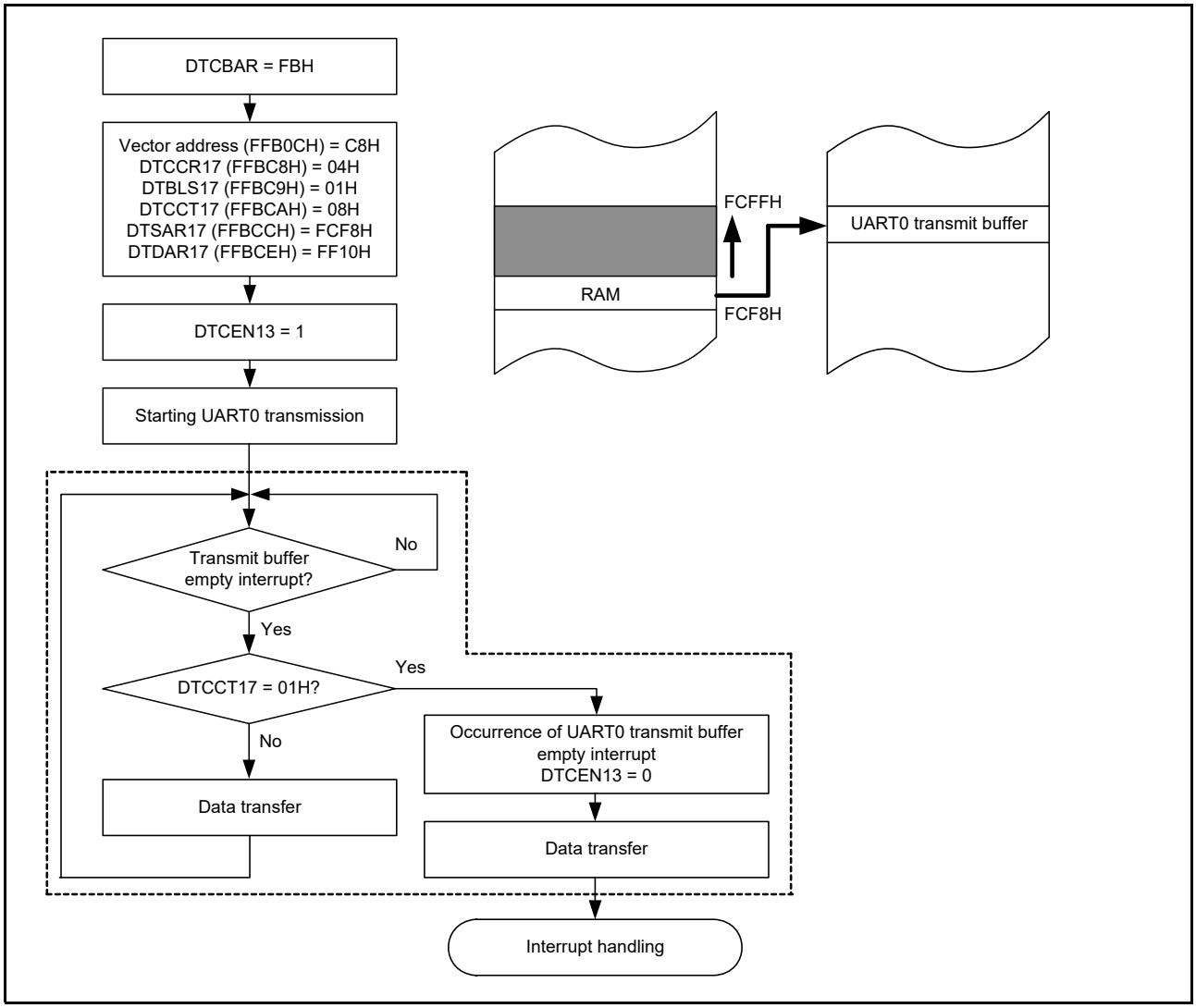


Remark The processing shown inside the dotted lines is automatically executed by the DTC.

The value of the DTRLD12 register is not used because of normal mode, but initialize the register to 00H when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.

2. Example 2 of using normal mode: UART0 consecutive transmission
   The DTC is activated by a UART0 transmit buffer empty interrupt and the value of RAM is transferred to the UART0 transmit buffer.
   - The vector address is FFB0CH and control data are allocated at FFBC8H to FFBCFH.
   - Transfers 8 bytes of data at addresses from FFCF8H to FFCFFH of RAM to the UART0 transmit buffer (FFF10H).

Figure 19 - 17 Example 2 of Using Normal Mode: UART0 Consecutive Transmission



Remark The processing shown inside the dotted lines is automatically executed by the DTC.

The value of the DTRLD17 register is not used because of normal mode, but initialize the register to 00H when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function. Start the first UART0 transmission by software. The second and subsequent transmissions proceeds automatically by using transmit buffer empty interrupts to activate the DTC.

19.4.3 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 (j = 0 to 23) 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 DTCENi7 to DTCENi0 in the DTCENi register (i = 0 to 4) 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 DTCENi7 to DTCENi0 are not set to 0. **Table 19 - 6** lists the register functions in repeat mode, and **Figure 19 - 18** shows the data transfers in repeat mode.

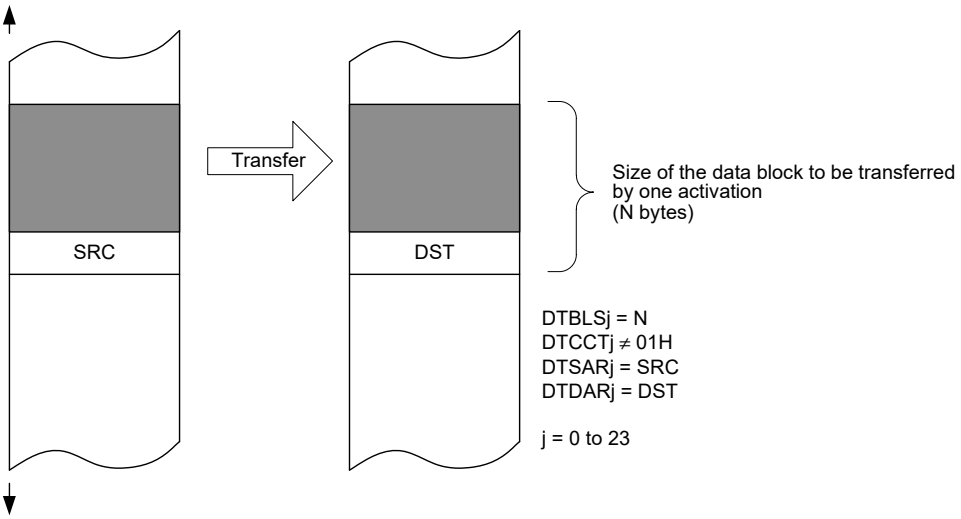
Table 19 - 6 Register Functions in Repeat Mode

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 DTCCTj register (the number of transfers is initialized).
DTC source address register j	DTSARj	Data transfer source address
DTC destination address register j	DTDARj	Data transfer destination address

Remark j = 0 to 23

Figure 19 - 18 Data Transfers in Repeat Mode

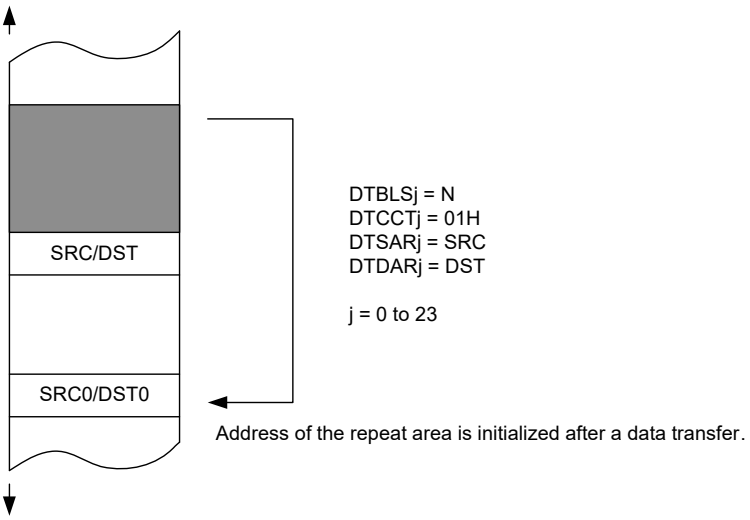
a) DTCCTj ≠ 01H



DTCCR Register Setting				Source Address Control	Destination Address Control	Source Address after Transfer	Destination Address after Transfer
DAMOD	SAMOD	RPTSEL	MODE				
0	×	1	1	Repeat area	Fixed	SRC + N	DST
1	×	1	1	Repeat area	Incremented	SRC + N	DST + N
×	0	0	1	Fixed	Repeat area	SRC	DST + N
×	1	0	1	Incremented	Repeat area	SRC + N	DST + N

×: 0 or 1

b) DTCCTj = 01H



DTCCR Register Setting				Source Address Control	Destination Address Control	Source Address after Transfer	Destination Address after Transfer
DAMOD	SAMOD	RPTSEL	MODE				
0	×	1	1	Repeat area	Fixed	SRC0	DST
1	×	1	1	Repeat area	Incremented	SRC0	DST + N
×	0	0	1	Fixed	Repeat area	SRC	DST0
×	1	0	1	Incremented	Repeat area	SRC + N	DST0

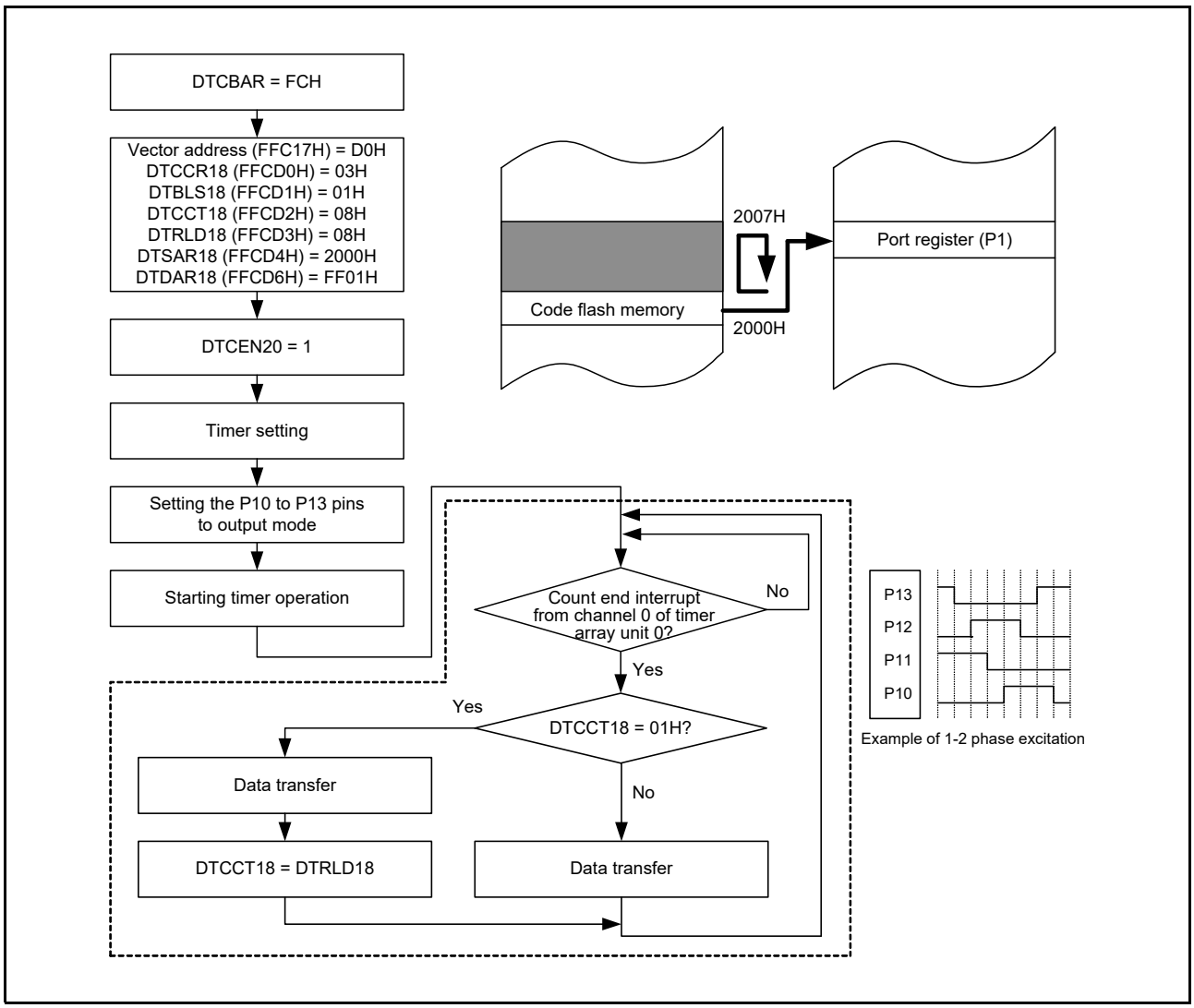
SRC0: Initial source address value
DST0: Initial destination address value
×: 0 or 1

Caution 1. When the repeat mode is used, the 8 lower-order bits of the initial value for the repeat area address must be 00H.

Caution 2. When the 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 stepping motor control pulses using port pins
   The DTC is activated using the interval timer function of channel 0 of timer array unit 0, and the patterns of the motor control pulse stored in the code flash memory are transferred to the general-purpose port pins.
   - The vector address is FFC17H and control data are allocated at FFCD0H to FFCD7H.
   - Transfers 8-byte data at addresses from 02000H to 02007H of the code flash memory from the mirror area (F2000H to F2007H) to the port register (P1, address: FFF01H).
   - A repeat mode interrupt is disabled.

Figure 19 - 19 Example 1 of Using Repeat Mode: Outputting Stepping Motor Control Pulses Using Port Pins

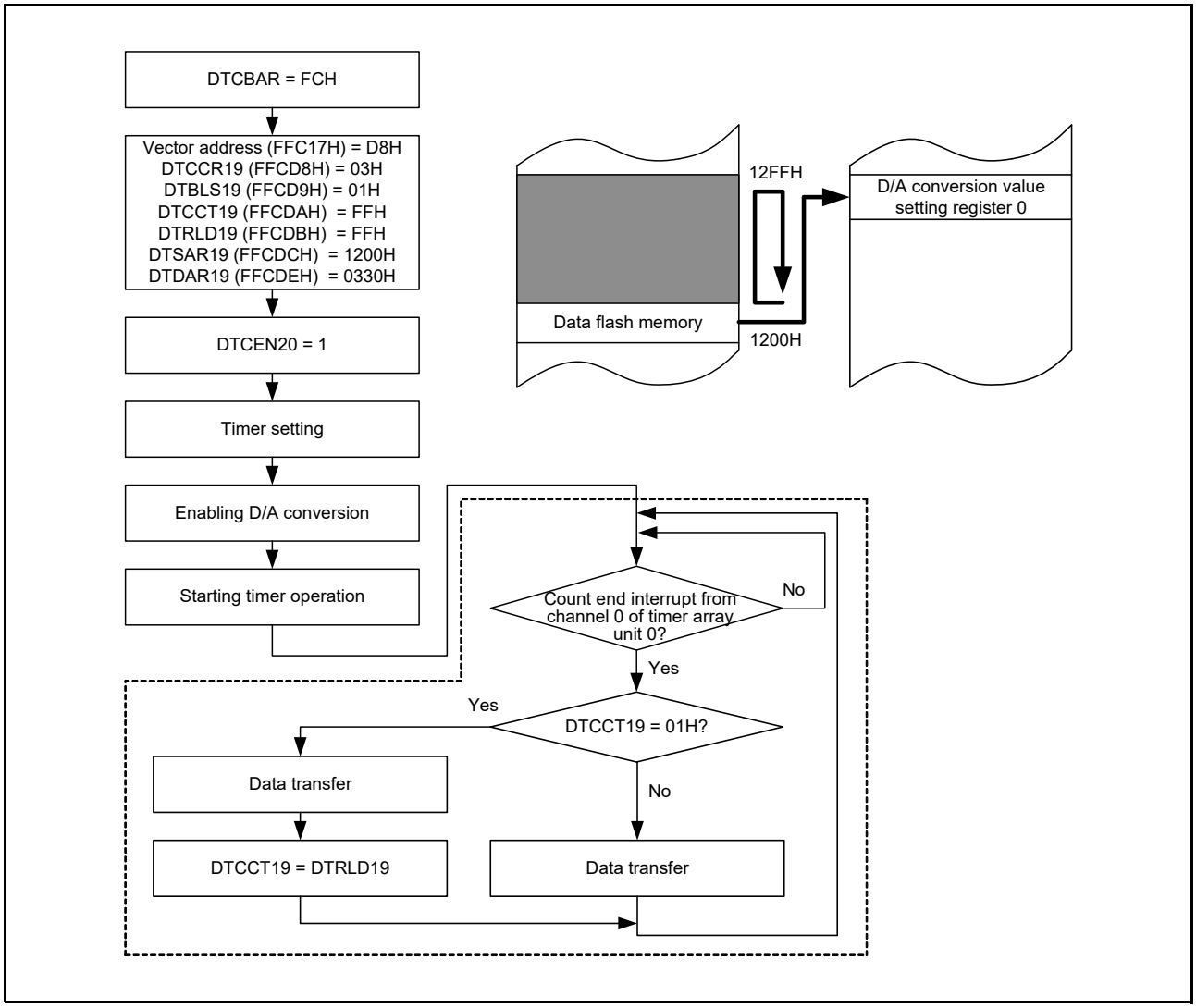


Remark The processing shown inside the dotted lines is automatically executed by the DTC.

Caution **To stop the output, stop the timer first and then clear the DTCEN20 bit.**

2. Example 2 of using repeat mode: Outputting a sine wave using the 8-bit D/A converter
   The DTC is activated using an interrupt of the interval timer function of channel 0 of timer array unit 0, 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 (F0330H). The timer interval time is set to the D/A output setup time.
   - The vector address is FFC17H and control data are allocated at FFCD8H to FFCDFH.
   - Transfers 255-byte data at addresses from F1200H to F12FEH of the data flash memory to the D/A conversion value setting register 0 (F0330H).
   - A repeat mode interrupt is disabled.

Figure 19 - 20 Example 2 of Using Repeat Mode: Outputting a Sine Wave Using the 8-bit D/A Converter



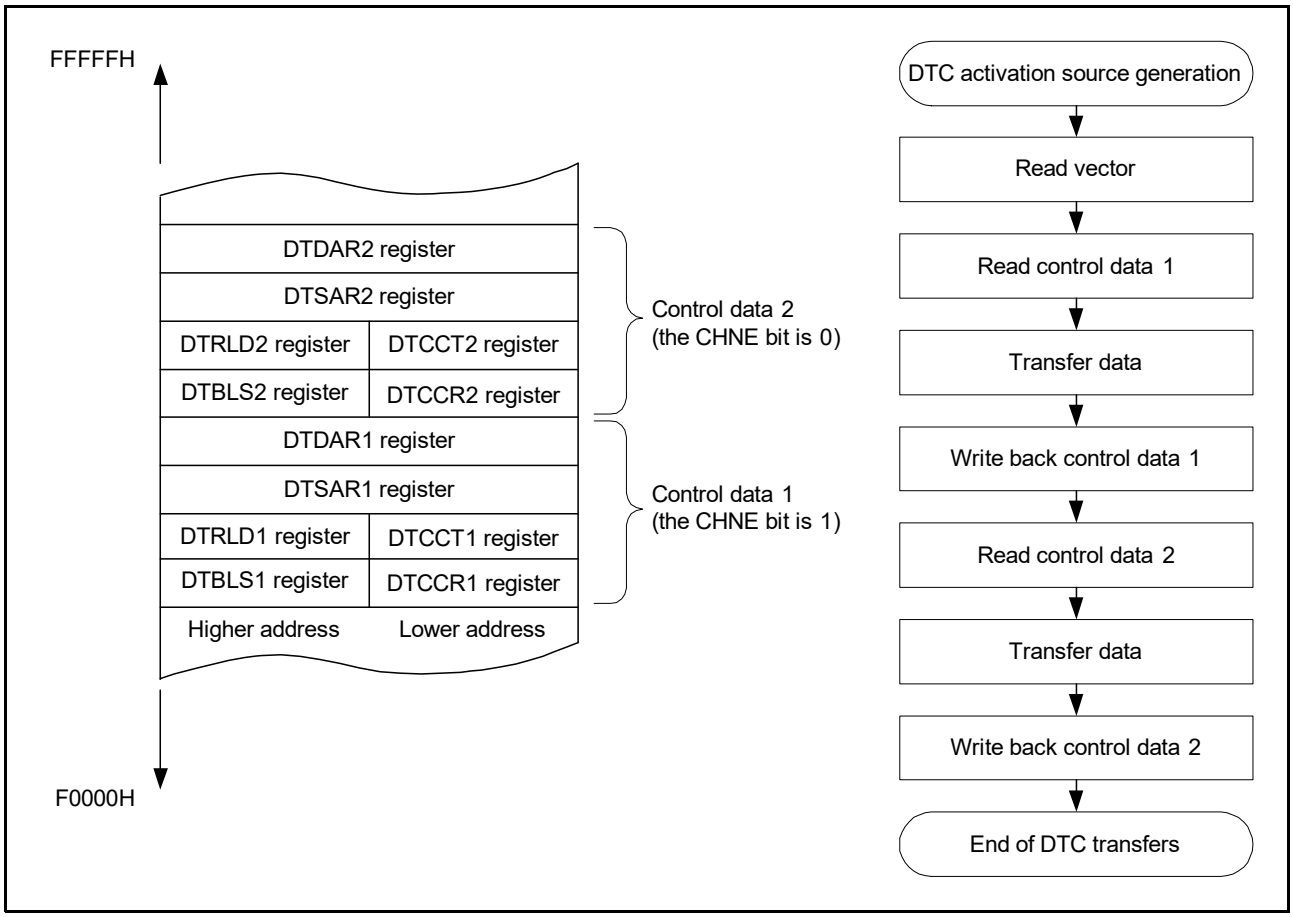
Remark The processing shown inside the dotted lines is automatically executed by the DTC.

Caution To stop the output, stop the timer first and then clear the DTCEN20 bit.

19.4.4 Chain transfers

When the CHNE bit in the DTCCRj register (j = 0 to 22) is 1 (chain transfers enabled), multiple data transfers can be continuously performed by one activation source. When the DTC is activated, one control data set is selected according to the data read from the vector address corresponding to the activation source, and the selected control data are 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 are read to continue a transfer 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. **Figure 19 - 21** shows the data transfers during chain transfers.

Figure 19 - 21 Data Transfers during Chain Transfers

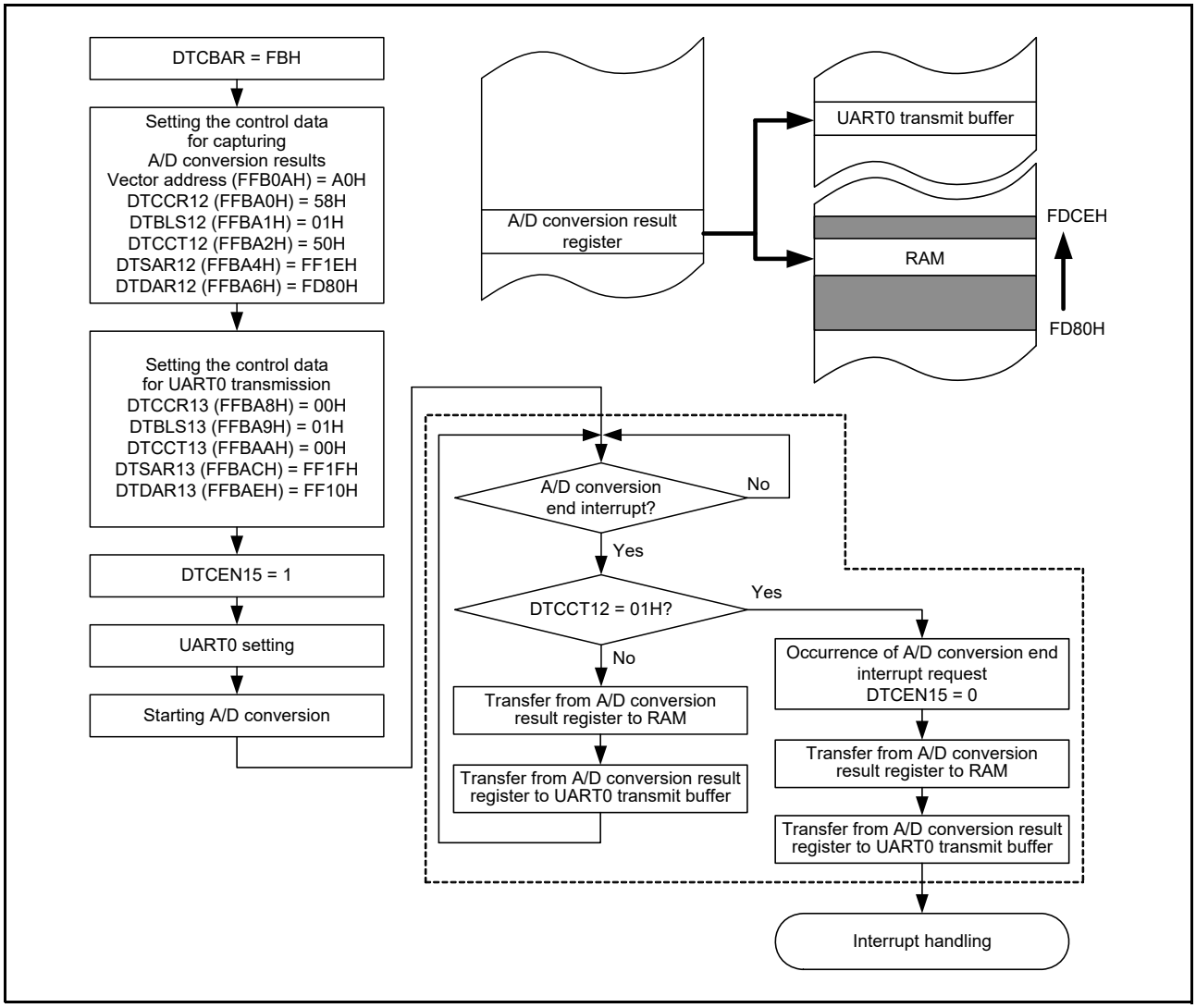


Caution 1. Set the CHNE bit in the DTCCR23 register to 0 (chain transfers disabled).

Caution 2. During chain transfers, bits DTCENi7 to DTCENi0 in the DTCENi register (i = 0 to 4) are not set to 0 (activation disabled) for the second and subsequent transfers. Also, no interrupt request is generated.

1. Example of using chain transfers: Consecutively capturing A/D conversion results and UART0 transmission
   The DTC is activated by an A/D conversion end interrupt and A/D conversion results are transferred to RAM, and then transmitted using the UART0.
   - The vector address is FFB0AH.
   - The control data for capturing A/D conversion results are allocated at FFBA0H to FFBA7H.
   - The control data for UART0 transmission are allocated at FFBA8H to FFBAFH.
   - Transfers 2-byte data of the A/D conversion result register (FFF1FH, FFF1EH) to FFD80H to FFDCFH of RAM, and transfers the 1 higher-order byte (FFF1FH) of the A/D conversion result register to the UART transmit buffer (FFF10H).

Figure 19 - 22 Example of Using Chain Transfers: Consecutively Capturing A/D Conversion Results and UART0 Transmission



Remark The processing shown inside the dotted lines is automatically executed by the DTC.

19.5 Points for Caution When the DTC is to be Used

19.5.1 Setting DTC control data and vector table

- Do not access the DTC extended special function registers (2nd SFRs), the DTC control data area, the DTC vector table area, or the general-purpose 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 DTCENi7 to DTCENi0 in the DTCENi register (i = 0 to 4) is 0 (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 DTCENi7 to DTCENi0 in the DTCENi register (i = 0 to 4) is 0 (activation disabled).

19.5.2 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 product and 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.
- Make sure the stack area, the DTC control data area, and the DTC vector table area do not overlap.
- The internal RAM area in the following products cannot be used as the DTC control data area or DTC vector table area when using the on-chip debugging trace function.
  R7F100GxG (x = A, B, C, E, F, G, J, L, M, P) and R7F104GBG: FC300H to FC6FFH
  R7F100GxJ (x = A, B, C, E, F, G, J, L, M, P, S), R7F104GFJ, and R7F104GGJ: FA300H to FA6FFH
  R7F100GxL and R7F100GxN (x = F, G, J, L, M, P, S): F4300H to F46FFH
- Initialize the DTRLDj register (j = 0 to 23) to 00H even in normal mode when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.
- If the internal RAM is placed in standby mode or shutdown mode (RAMSDMD = 1) by the memory power reduction control register (PSMCR), the internal RAM cannot be used as the DTC control data area or DTC vector table area.
- The area where debug monitor programs are allocated when using the on-chip debugging function cannot be used as the DTC control data area or DTC vector table area. For details, see **34.4 Allocation of Memory Spaces to User Resources**.

19.5.3 DTC pending instruction

Even if a DTC transfer request is generated, DTC transfer is held pending 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 memory manipulation instructions for IFxx, MKxx, PRxx, and PSW, and an 8-bit memory manipulation instruction that has the ES register as operand
- Instruction for accessing the data flash memory
- Multiply/divide/multiply & accumulate instruction (excluding MULU)

Caution 1. When a DTC transfer request is acknowledged, all interrupt requests are held pending until DTC transfer is completed.

Caution 2. While the DTC is held pending by the DTC pending instruction, all interrupt requests are held pending.

19.5.4 Operation when accessing data flash memory area

When accessing the data flash memory area after an instruction execution from the start of DTC data transfer, a wait of three clock cycles 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 memory area

19.5.5 Number of DTC execution clock cycles

Table 19 - 7 lists the operations following DTC activation and required number of cycles for each operation.

Table 19 - 7 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**

Note 1. For the number of clock cycles required for control data write-back, refer to **Table 19 - 8 Number of Clock Cycles Required for Control Data Write-back Operation**.

Note 2. For the number of clock cycles required for data read/write, refer to **Table 19 - 9 Number of Clock Cycles Required for One Data Read/Write Operation**.

Table 19 - 8 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	×	0	Fixed	Fixed	Written back	Written back	Not written back	Not written back	1
0	1	×	0	Incremented	Fixed	Written back	Written back	Written back	Not written back	2
1	0	×	0	Fixed	Incremented	Written back	Written back	Not written back	Written back	2
1	1	×	0	Incremented	Incremented	Written back	Written back	Written back	Written back	3
0	×	1	1	Repeat area	Fixed	Written back	Written back	Written back	Not written back	2
1	×	1	1		Incremented	Written back	Written back	Written back	Written back	3
×	0	0	1	Fixed	Repeat area	Written back	Written back	Not written back	Written back	2
×	1	0	1	Incremented		Written back	Written back	Written back	Written back	3

Remark 1. ×: Don't care

Remark 2. j = 0 to 23

Table 19 - 9 Number of Clock Cycles Required for One Data Read/Write Operation

Operation	RAM	Code Flash Memory	Data Flash Memory	Special Function Register (SFR)	Extended Special Function Register (2nd SFR)	
					No Wait State	Wait States
Data read	1	2	4	1	1	1 + number of wait cycles**Note**
Data write	1	—	—	1	1	1 + number of wait cycles**Note**

Note The number of wait cycles differs depending on the specifications of the register to be accessed in the extended special function register (2nd SFR) area.

19.5.6 DTC response time

Table 19 - 10 lists the DTC response time. The DTC response time is the time from when the DTC activation source is detected until DTC transfer starts. It does not include the number of DTC execution clock cycles.

Table 19 - 10 DTC Response Time

	Minimum Time	Maximum Time
Response Time	3 cycles	19 cycles

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 condition.

- When executing an instruction from the internal RAM
  Maximum response time: 20 cycles
- When executing a DTC pending instruction (refer to **19.5.3 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

Remark 1 cycle: 1/fCLK (fCLK: CPU/peripheral hardware clock)

19.5.7 DTC activation sources

- After inputting a DTC activation source, do not input the same activation source again until DTC transfer is completed.
- While a DTC activation source is generated, do not manipulate the DTC activation enable bit corresponding to the source.
- If DTC activation sources conflict, their priority levels are determined in order to select the source for activation when the CPU acknowledges the DTC transfer. For details on the priority levels of activation sources, refer to **19.3.3 Vector table**.
- When DTC activation is enabled under either of the following conditions, a DTC transfer is started and an interrupt request is generated after completion of the transfer. Therefore, check the monitor flag (CnMON) of the comparator before enabling DTC activation as required (n = 0, 1).
  – The comparator is set to an interrupt request on one-edge detection (CnEDG = 0), an interrupt request at the rising edge for the comparator (CnEPO = 0), and IVCMP > comparator n reference voltage**Note**
  – The comparator is set to an interrupt request on one-edge detection (CnEDG = 0), an interrupt request at the falling edge for the comparator (CnEPO = 1), and IVCMP < comparator n reference voltage**Note**

Note n = 0: IVREF0 or internal reference voltage
n = 1: IVREF1 or D/A converter output 0

19.5.8 Operation in standby mode

State	DTC Operation
HALT mode	Operable[Note 1]
STOP mode	DTC activation sources can be accepted[Note 2]
SNOOZE mode	Operable[Notes 3, 4, 5, 6]

Note 1. When the subsystem clock is selected as f_{CLK}, operation is disabled if the RTCLPC bit of the OSMC register is 1.

Note 2. In the STOP mode, detecting a DTC activation source enables transition to SNOOZE mode and DTC transfer. After completion of transfer, the chip 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.

Note 3. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock is selected as f_{CLK}.

Note 4. When a transfer end interrupt from the CSIp in SNOOZE mode is being used as the DTC activation source, use the transfer end interrupt to release the chip from the SNOOZE mode and start processing by the CPU after completion of DTC transfer, or use a chain transfer to make the settings for reception by the CSIp (writing 1 to the STm.STm0 bit, writing 0 to the SSCm.SWCm bit, setting the SSCm register, and writing 1 to the SSm.SSm0 bit) again.

Note 5. When a transfer end interrupt from the UARTq in SNOOZE mode is being used as the DTC activation source, use the transfer end interrupt to release the chip from the SNOOZE mode and start processing by the CPU after completion of DTC transfer, or use a chain transfer to make the settings for reception by the UARTq (writing 1 to the STm.STm1 bit, writing 0 to the SSCm.SWCm bit, setting the SSCm register, and writing 1 to the SSm.SSm1 bit) again.

Note 6. When an A/D conversion end interrupt from the A/D converter in SNOOZE mode is being used as the DTC activation source, use the A/D conversion end interrupt to release the chip from the SNOOZE mode and start processing by the CPU after completion of DTC transfer, or use a chain transfer to make the settings for the SNOOZE mode function of the A/D converter (writing 1 to the ADM2.AWC bit after having written 0 to it) again.

Caution **If the internal RAM is placed in standby mode or shutdown mode (RAMSDMD = 1) by the memory power reduction control register (PSMCR), the internal RAM cannot be set as a DTC transfer source or destination.**

Remark 30- to 64-pin R7F100G products
p: CSI number (p = 00), q: UART number (q = 0), m: Unit number (m = 0)
80- to 128-pin R7F100G products
p: CSI number (p = 00, 20), q: UART number (q = 0, 2), m: Unit number (m = 0, 1)
32-pin R7F104G product
p: CSI number (p = 00), q: UART number (q = 0), m: Unit number (m = 0)
44- and 48-pin R7F104G products
p: CSI number (p = 00), q: UART number (q = 0, 2), m: Unit number (m = 0, 1)

Section 20 Logic and Event Link Controller (ELCL)

20.1 Functions of the ELCL

The logic and event link controller (ELCL) links signals output by peripheral functions to other specified peripheral functions through internal logic cell blocks, allowing direct communications between peripheral functions without CPU intervention. The ELCL has the following functions.

- Eight event signals selected from among up to 99 signals from peripheral functions can be linked to specified peripheral functions without CPU intervention. The destinations for connection of these eight signals are selected from among up to 29 event input terminals of the specified peripheral functions.
- Event signals from peripheral functions can be input to logic cells (including AND, OR, and EX-OR circuits) to change the link conditions.
- Event signals from peripheral functions can also be input to selectors to specify conditions for activating the selected peripheral functions.
- Event signals from peripheral functions can also be input to flip-flops to link the signals to the selected peripheral functions in synchronization with a selected clock.

20.2 Configuration of ELCL

Figure 20 - 1 shows a block diagram of the ELCL.

Figure 20 - 1 Block Diagram of ELCL

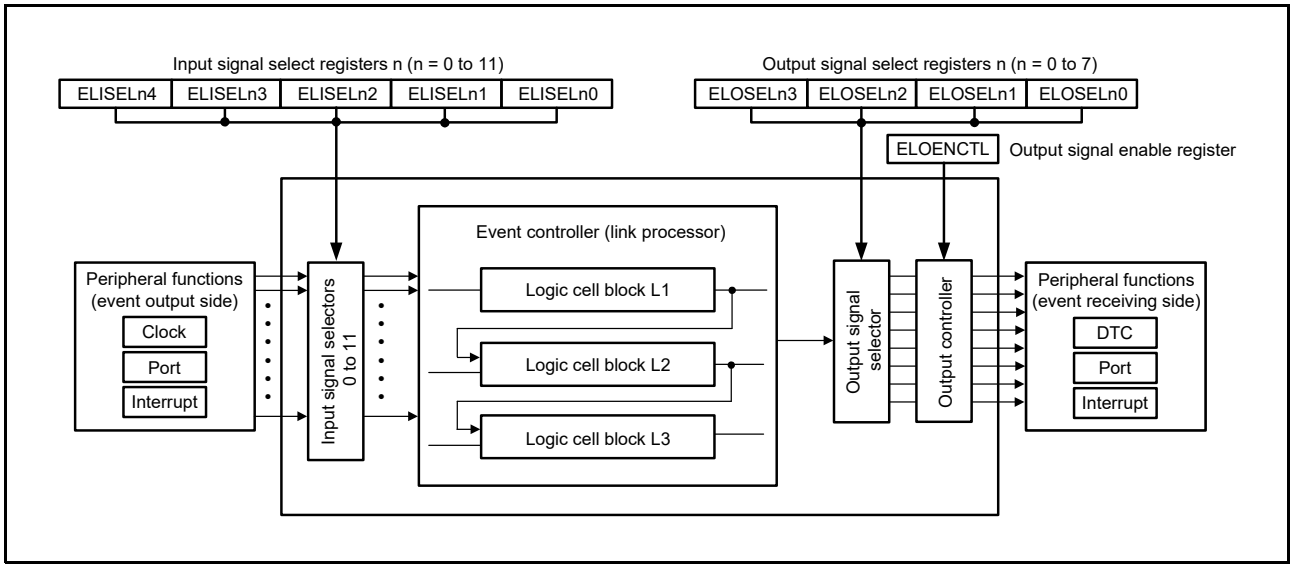


Figure 20 - 2 Block Diagram of Logic Cell Block L1

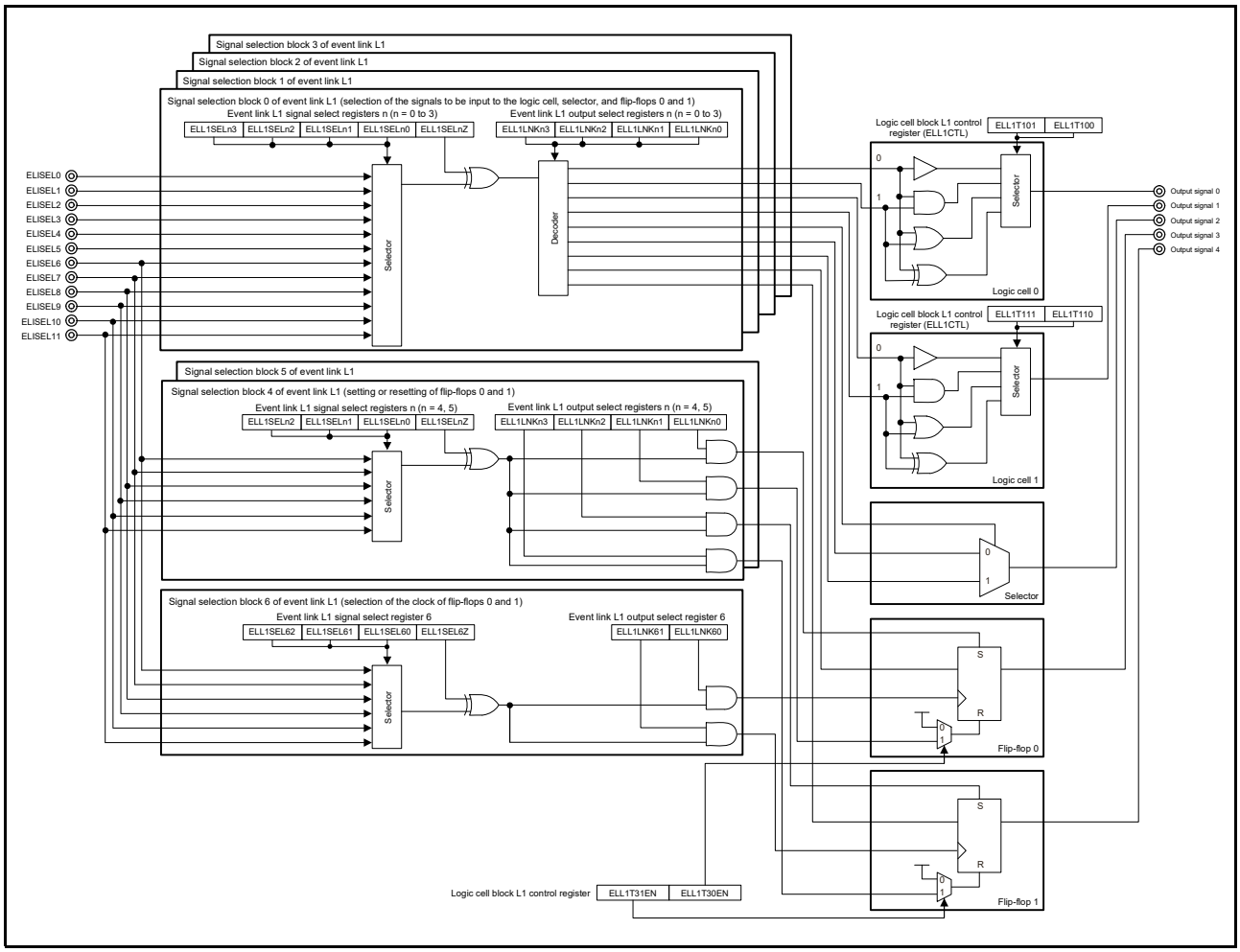


Figure 20 - 3 Block Diagram of Logic Cell Block L2

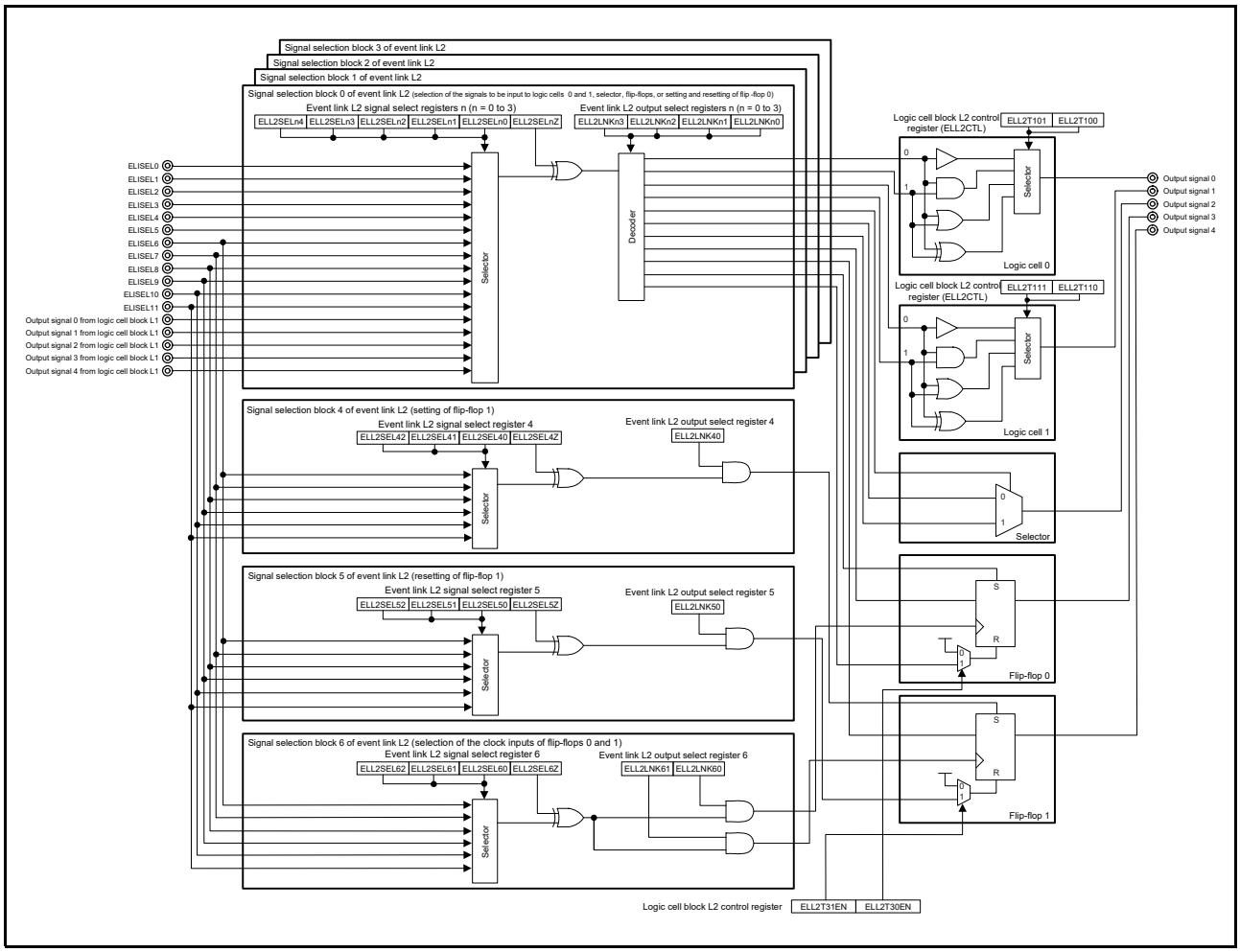


Figure 20 - 4 Block Diagram of Logic Cell Block L3

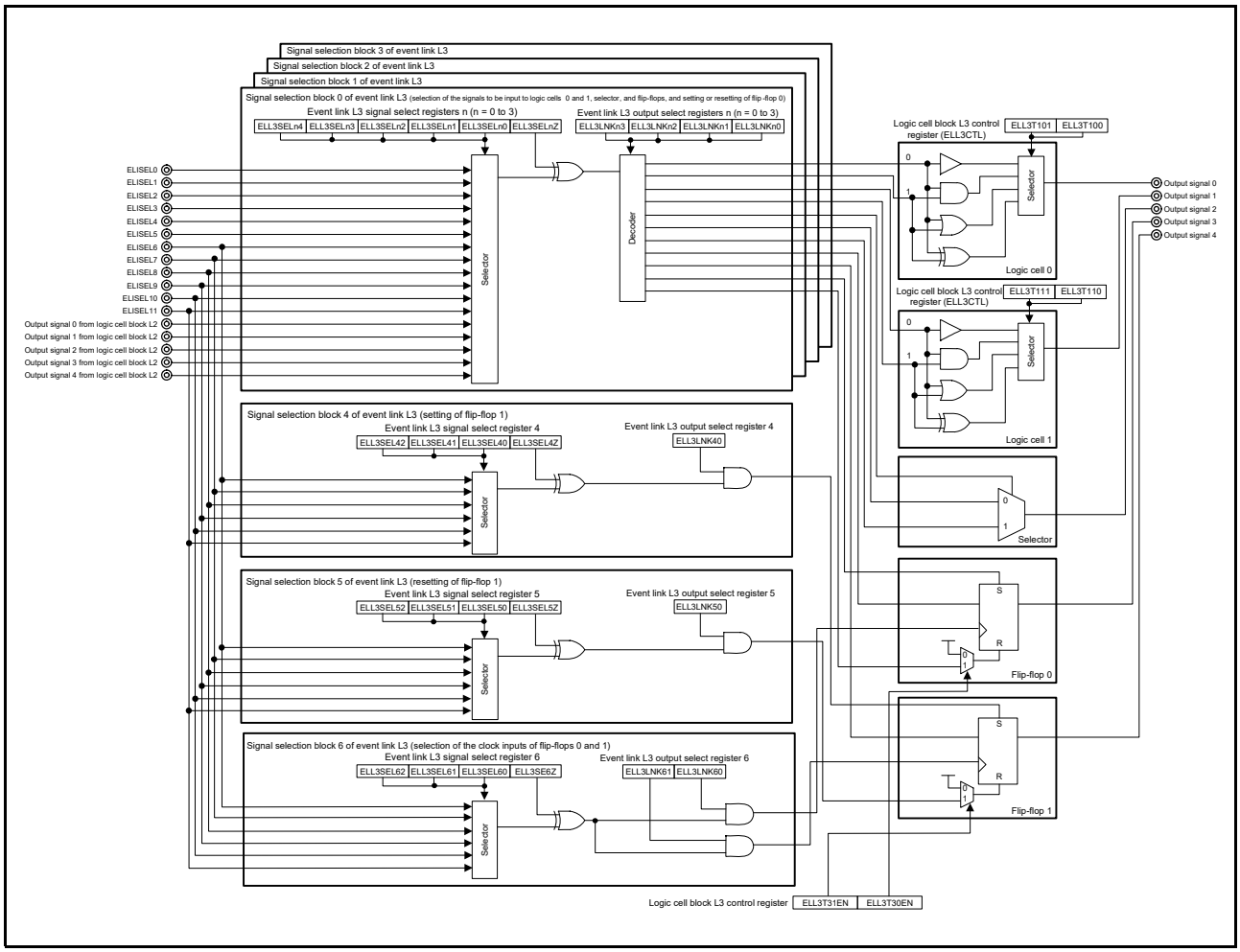


Table 20 - 1 and **Table 20 - 2** list the connections in logic cell block L1. The signals to be input to logic cell block L1 are selectable from among the event-output peripheral functions specified in the input signal select registers n (ELISELn) (n = 0 to 11). The event link L1 signal select registers n (ELL1SELn) (n = 0 to 6) can be used to specify the signal to be input to signal selection block n (n = 0 to 6) of event link L1. The event link L1 output select registers n (ELL1LNKn) (n = 0 to 6) can be used to specify the destinations of the signals to be output from the given block. The ELL1CTL register can be used to select the connections to logic cells and enable or stop flip-flops. The signals output from logic cell block L1 can be used as inputs to event-receiving peripheral functions or as inputs to logic cell block L2.

Table 20 - 1 Connections in Logic Cell Block L1 (1)

Signal selection block n of event link L1	Input Signal	Destination of the signal to be output from signal selection block n of event link L1
Signal selection block 0 of event link L1**Note 1**	• Signals selected by the ELISEL0 to ELISEL11 registers Use the ELL1SEL0 register to select one signal to be input to logic cell block L1 from among the signals selectable by the above registers.	• Logic cell 0 (input 0 or input 1) • Logic cell 1 (input 0 or input 1) • Selector (selection, input 0 or input 1) • Flip-flop 0 (input) • Flip-flop 1 (input) Use the ELL1LNK0 register to select one of the above destinations.
Signal selection block 1 of event link L1**Note 1**	• Signals selected by the ELISEL0 to ELISEL11 registers Use the ELL1SEL1 register to select one signal to be input to logic cell block L1 from among the signals selectable by the above registers.	• Logic cell 0 (input 0 or input 1) • Logic cell 1 (input 0 or input 1) • Selector (selection, input 0 or input 1) • Flip-flop 0 (input) • Flip-flop 1 (input) Use the ELL1LNK1 register to select one of the above destinations.
Signal selection block 2 of event link L1**Note 1**	• Signals selected by the ELISEL0 to ELISEL11 registers Use the ELL1SEL2 register to select one signal to be input to logic cell block L1 from among the signals selectable by the above registers.	• Logic cell 0 (input 0 or input 1) • Logic cell 1 (input 0 or input 1) • Selector (selection, input 0 or input 1) • Flip-flop 0 (input) • Flip-flop 1 (input) Use the ELL1LNK2 register to select one of the above destinations.
Signal selection block 3 of event link L1**Note 1**	• Signals selected by the ELISEL0 to ELISEL11 registers Use the ELL1SEL3 register to select one signal to be input to logic cell block L1 from among the signals selectable by the above registers.	• Logic cell 0 (input 0 or input 1) • Logic cell 1 (input 0 or input 1) • Selector (selection, input 0 or input 1) • Flip-flop 0 (input) • Flip-flop 1 (input) Use the ELL1LNK3 register to select one of the above destinations.
Signal selection block 4 of event link L1**Note 2**	• Signals selected by the ELISEL6 to ELISEL11 registers Use the ELL1SEL4 register to select one signal to be input to logic cell block L1 from among the signals selectable by the above registers.	• Flip-flop 0 (set or reset) • Flip-flop 1 (set or reset) Use the ELL1LNK4 register to select a destination.
Signal selection block 5 of event link L1**Note 2**	• Signals selected by the ELISEL6 to ELISEL11 registers Use the ELL1SEL5 register to select one signal to be input to logic cell block L1 from among the signals selectable by the above registers.	• Flip-flop 0 (set or reset) • Flip-flop 1 (set or reset) Use the ELL1LNK5 register to select a destination.
Signal selection block 6 of event link L1	• Signals selected by the ELISEL6 to ELISEL11 registers Use the ELL1SEL6 register to select one signal to be input to logic cell block L1 from among the signals selectable by the above registers.	• Flip-flop 0 (clock) • Flip-flop 1 (clock) Use the ELL1LNK6 register to select a destination.

Note 1. Select different destination of the signal to be output from signal selection blocks 0 to 6 of event link L1; do not connect two or more signals to a single destination.

Note 2. Do not connect a single signal to both the set and reset control of flip-flop 0 or 1. Make sure that there is no period during which the signals for set and for reset are both high at the same time.

Table 20 - 2 Connections in Logic Cell Block L1 (2)

Signal output from logic cell block L1	Signal source
Signal 0 output from logic cell block L1	Output from logic cell 0 in logic cell block L1
Signal 1 output from logic cell block L1	Output from logic cell 1 in logic cell block L1
Signal 2 output from logic cell block L1	Output from the selector in logic cell block L1
Signal 3 output from logic cell block L1	Output from flip-flop 0 in logic cell block L1
Signal 4 output from logic cell block L1	Output from flip-flop 1 in logic cell block L1

Table 20 - 3 and **Table 20 - 4** list the connections in logic cell block L2. The signals to be input to logic cell block L2 are selectable from among the event-output peripheral functions specified by the input signal select registers n (ELISELn) (n = 0 to 11) and the signals output from logic cell block L1. The event link L2 output select registers n (ELL1LNKn) (n = 0 to 6) can be used to specify the destinations of the signals to be output from the given block. The ELL2CTL register can be used to select the connections to logic cells and enable or stop flip-flops. The signals output from logic cell block L2 can be used as inputs to event-receiving peripheral functions or as inputs to logic cell block L3.

Table 20 - 3 Connections in Logic Cell Block L2 (1)

Signal selection block n of event link L2	Input Signal	Destination of the signal to be output from signal selection block n of event link L2
Signal selection block 0 of event link L2**Notes 1, 2**	• Signals selected by the ELISEL0 to ELISEL11 registers • Signals 0 to 4 output from logic cell block L1 Use the ELL2SEL0 register to select one signal to be input to logic cell block L2 from among the signals selectable by the above registers.	• Logic cell 0 (input 0 or input 1) • Logic cell 1 (input 0 or input 1) • Selector (selection, input 0 or input 1) • Flip-flop 0 (input, set, or reset) • Flip-flop 1 (input) Use the ELL2LNK0 register to select one of the above destinations.
Signal selection block 1 of event link L2**Notes 1, 2**	• Signals selected by the ELISEL0 to ELISEL11 registers • Signals 0 to 4 output from logic cell block L1 Use the ELL2SEL1 register to select one signal to be input to logic cell block L2 from among the signals selectable by the above registers.	• Logic cell 0 (input 0 or input 1) • Logic cell 1 (input 0 or input 1) • Selector (selection, input 0 or input 1) • Flip-flop 0 (input, set, or reset) • Flip-flop 1 (input) Use the ELL2LNK1 register to select one of the above destinations.
Signal selection block 2 of event link L2**Notes 1, 2**	• Signals selected by the ELISEL0 to ELISEL11 registers • Signals 0 to 4 output from logic cell block L1 Use the ELL2SEL2 register to select one signal to be input to logic cell block L2 from among the signals selectable by the above registers.	• Logic cell 0 (input 0 or input 1) • Logic cell 1 (input 0 or input 1) • Selector (selection, input 0 or input 1) • Flip-flop 0 (input, set, or reset) • Flip-flop 1 (input) Use the ELL2LNK2 register to select one of the above destinations.
Signal selection block 3 of event link L2**Notes 1, 2**	• Signals selected by the ELISEL0 to ELISEL11 registers • Signals 0 to 4 output from logic cell block L1 Use the ELL2SEL3 register to select one signal to be input to logic cell block L2 from among the signals selectable by the above registers.	• Logic cell 0 (input 0 or input 1) • Logic cell 1 (input 0 or input 1) • Selector (selection, input 0 or input 1) • Flip-flop 0 (input, set, or reset) • Flip-flop 1 (input) Use the ELL2LNK3 register to select one of the above destinations.
Signal selection block 4 of event link L2**Note 2**	• Signals selected by the ELISEL6 to ELISEL11 registers Use the ELL2SEL4 register to select one signal to be input to logic cell block L2 from among the signals selectable by the above registers.	• Flip-flop 1 (set) Use the ELL2LNK4 register to select one of the above destinations.
Signal selection block 5 of event link L2**Note 2**	• Signals selected by the ELISEL6 to ELISEL11 registers Use the ELL2SEL5 register to select one signal to be input to logic cell block L2 from among the signals selectable by the above registers.	• Flip-flop 1 (reset) Use the ELL2LNK5 register to select one of the above destinations.
Signal selection block 6 of event link L2	• Signals selected by the ELISEL6 to ELISEL11 registers Use the ELL2SEL6 register to select one signal to be input to logic cell block L2 from among the signals selectable by the above registers.	• Flip-flop 0 (clock) • Flip-flop 1 (clock) Use the ELL2LNK6 register to select one of the above destinations.

Note 1. Select different destination of the signal to be output from signal selection blocks 0 to 6 of event link L2; do not connect two or more signals to a single destination.

Note 2. Do not connect a single signal to both the set and reset control of flip-flop 0 or 1. Make sure that there is no period during which the signals for set and for reset are both high at the same time.

Table 20 - 4 Connections in Logic Cell Block L2 (2)

Signal output from logic cell block L2	Signal source
Signal 0 output from logic cell block L2	Output from logic cell 0 in logic cell block L2
Signal 1 output from logic cell block L2	Output from logic cell 1 in logic cell block L2
Signal 2 output from logic cell block L2	Output from the selector in logic cell block L2
Signal 3 output from logic cell block L2	Output from flip-flop 0 in logic cell block L2
Signal 4 output from logic cell block L2	Output from flip-flop 1 in logic cell block L2

Table 20 - 5 and **Table 20 - 6** list the connections in logic cell block L3. The signals to be input to logic cell block L3 are selectable from among the event-output peripheral functions specified in the input signal select registers n (ELISELn) (n = 0 to 11) and the signals output from logic cell block L2. The event link L1 signal select registers n (ELL1SELn) (n = 0 to 6) can be used to specify the signal to be input to signal selection block n (n = 0 to 6) of event link L3. The event link L3 output select registers n (ELL3LNKn) (n = 0 to 6) can be used to specify the destinations of the signals to be output from the given block. The ELL3CTL register can be used to select the connections to logic cells and enable or stop flip-flops. The signals output from logic cell block L3 can be used as inputs to event-receiving peripheral functions or as inputs to logic cell block L3.

Table 20 - 5 Connections in Logic Cell Block L3 (1)

Signal selection block n of event link L3	Input Signal	Destination of the signal to be output from signal selection block n of event link L3
Signal selection block 0 of event link L3**Notes 1, 2**	• Signals selected by the ELISEL0 to ELISEL11 registers • Signals 0 to 4 output from logic cell block L2 Use the ELL3SEL0 register to select one signal to be input to logic cell block L3 from among the signals selectable by the above registers.	• Logic cell 0 (input 0 or input 1) • Logic cell 1 (input 0 or input 1) • Selector (selection, input 0 or input 1) • Flip-flop 0 (input, set, or reset) • Flip-flop 1 (input) Use the ELL3LNK0 register to select one of the above destinations.
Signal selection block 1 of event link L3**Notes 1, 2**	• Signals selected by the ELISEL0 to ELISEL11 registers • Signals 0 to 4 output from logic cell block L2 Use the ELL3SEL1 register to select one signal to be input to logic cell block L3 from among the signals selectable by the above registers.	• Logic cell 0 (input 0 or input 1) • Logic cell 1 (input 0 or input 1) • Selector (selection, input 0 or input 1) • Flip-flop 0 (input, set, or reset) • Flip-flop 1 (input) Use the ELL3LNK1 register to select one of the above destinations.
Signal selection block 2 of event link L3**Notes 1, 2**	• Signals selected by the ELISEL0 to ELISEL11 registers • Signals 0 to 4 output from logic cell block L2 Use the ELL3SEL2 register to select one signal to be input to logic cell block L3 from among the signals selectable by the above registers.	• Logic cell 0 (input 0 or input 1) • Logic cell 1 (input 0 or input 1) • Selector (selection, input 0 or input 1) • Flip-flop 0 (input, set, or reset) • Flip-flop 1 (input) Use the ELL3LNK2 register to select one of the above destinations.
Signal selection block 3 of event link L3**Notes 1, 2**	• Signals selected by the ELISEL0 to ELISEL11 registers • Signals 0 to 4 output from logic cell block L2 Use the ELL3SEL3 register to select one signal to be input to logic cell block 3 from among the signals selectable by the above registers.	• Logic cell 0 (input 0 or input 1) • Logic cell 1 (input 0 or input 1) • Selector (selection, input 0 or input 1) • Flip-flop 0 (input, set, or reset) • Flip-flop 1 (input) Use the ELL3LNK3 register to select one of the above destinations.
Signal selection block 4 of event link L3**Note 2**	• Signals selected by the ELISEL6 to ELISEL11 registers Use the ELL3SEL4 register to select one signal to be input to logic cell block L3 from among the signals selectable by the above registers.	• Flip-flop 1 (set) Use the ELL3LNK4 register to select one of the above destinations.
Signal selection block 5 of event link L3**Note 2**	• Signals selected by the ELISEL6 to ELISEL11 registers Use the ELL3SEL5 register to select one signal to be input to logic cell block L3 from among the signals selectable by the above registers.	• Flip-flop 1 (reset) Use the ELL3LNK5 register to select one of the above destinations.
Signal selection block 6 of event link L3	• Signals selected by the ELISEL6 to ELISEL11 registers Use the ELL3SEL6 register to select one signal to be input to logic cell block L3 from among the signals selectable by the above registers.	• Flip-flop 0 (clock) • Flip-flop 1 (clock) Use the ELL3LNK6 register to select one of the above destinations.

Note 1. Select different destination of the signal to be output from signal selection blocks 0 to 6 of event link L3; do not connect two or more signals to a single destination.

Note 2. Do not connect a single signal to both the set and reset control of flip-flop 0 or 1. Make sure that there is no period during which the signals for set and for reset are both high at the same time.

Table 20 - 6 Connections in Logic Cell Block L3 (2)

Signal output from logic cell block L3	Signal source
Signal 0 output from logic cell block L3	Output from logic cell 0 in logic cell block L3
Signal 1 output from logic cell block L3	Output from logic cell 1 in logic cell block L3
Signal 2 output from logic cell block L3	Output from the selector in logic cell block L3
Signal 3 output from logic cell block L3	Output from flip-flop 0 in logic cell block L3
Signal 4 output from logic cell block L3	Output from flip-flop 1 in logic cell block L3

20.3 Registers for Controlling the ELCL

The following registers are used to control the ELCL.

- Input signal select registers n (ELISELn) (n = 0 to 11)
- Event link L1 signal select registers 0 to 3 (ELL1SEL0 to ELL1SEL3)
- Event link L1 signal select registers 4, 5 (ELL1SEL4, ELL1SEL5)
- Event link L1 signal select register 6 (ELL1SEL6)
- Logic cell block L1 control register (ELL1CTL)
- Event link L1 output select registers 0 to 3 (ELL1LNK0 to ELL1LNK3)
- Event link L1 output select registers 4, 5 (ELL1LNK4, ELL1LNK5)
- Event link L1 output select register 6 (ELL1LNK6)
- Event link L2 signal select registers 0 to 3 (ELL2SEL0 to ELL2SEL3)
- Event link L2 signal select registers 4, 5 (ELL2SEL4, ELL2SEL5)
- Event link L2 signal select register 6 (ELL2SEL6)
- Logic cell block L2 control register (ELL2CTL)
- Event link L2 output select registers 0 to 3 (ELL2LNK0 to ELL2LNK3)
- Event link L2 output select registers 4, 5 (ELL2LNK4, ELL2LNK5)
- Event link L2 output select register 6 (ELL2LNK6)
- Event link L3 signal select registers 0 to 3 (ELL3SEL0 to ELL3SEL3)
- Event link L3 signal select registers 4, 5 (ELL3SEL4, ELL3SEL5)
- Event link L3 signal select register 6 (ELL3SEL6)
- Logic cell block L3 control register (ELL3CTL)
- Event link L3 output select registers 0 to 3 (ELL3LNK0 to ELL3LNK3)
- Event link L3 output select registers 4, 5 (ELL3LNK4, ELL3LNK5)
- Event link L3 output select register 6 (ELL3LNK6)
- Output signal select registers n (ELOSELn) (n = 0 to 7)
- Output signal enable register (ELOENCTL)
- Output signal monitor register (ELOMONI)
- Port mode registers (PMxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)

20.3.1 Input signal select registers n (ELISELn)

These registers are used to select event signals to be input from event-outputting peripheral functions. The event signals selected by these registers are linked to the event-receiving peripheral functions through logic cell blocks L1 to L3. These registers can be set by an 8-bit memory manipulation instruction. The value of each ELISELn register following a reset is 00H.

Remark n = 0 to 11

Figure 20 - 5 Format of Input Signal Select Registers n (ELISELn) (1/3)

Address: F0680H (ELISEL0) to F068BH (ELISEL11)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELISELn	0	0	0	ELISELn4	ELISELn3	ELISELn2	ELISELn1	ELISELn0

Figure 20 - 5 Format of Input Signal Select Registers n (ELISELn) (2/3)

ELISELn (n = 0 to 5)

ELISELn4	ELISELn3	ELISELn2	ELISELn1	ELISELn0	Event Source
0	0	0	0	0	Transmit data output from SAU0 channel 0
0	0	0	0	1	Transmit data output from SAU0 channel 1Note 1
0	0	0	1	0	Serial clock output from SAU0 channel 0
0	0	0	1	1	Serial clock output from SAU0 channel 1Note 1
0	0	1	0	0	Input pin (P11/EI11)
0	0	1	0	1	Input pin (P50/EI50)
0	0	1	1	0	Serial clock output from UARTA0Note 2
0	0	1	1	1	Output from TAU0 channel 0
0	1	0	0	0	Output from TAU0 channel 1
0	1	0	0	1	Output from TAU0 channel 2
0	1	0	1	0	Output from TAU0 channel 3
0	1	0	1	1	Output from TAU0 channel 4
0	1	1	0	0	Output from TAU0 channel 5
0	1	1	0	1	Output from TAU0 channel 6
0	1	1	1	0	Output from TAU0 channel 7
0	1	1	1	1	Input pin (P10/EI10)
1	0	0	0	0	Input pin (P51/EI51)Note 3
1	0	0	0	1	Comparator detection 0
1	0	0	1	0	Comparator detection 1Note 3
1	0	0	1	1	Transmit data output 0 from UARTA0Note 2
1	0	1	0	0	Input pin (P12/EI12)
1	0	1	0	1	Input pin 0Note 4
1	0	1	1	0	Input pin 1Note 4
1	0	1	1	1	INTC0Note 5
1	1	0	0	0	INTC1Note 5
1	1	0	0	1	INTC2Note 5
1	1	0	1	0	INTC3Note 5
1	1	0	1	1	Input pin (P137/EI137)
1	1	1	0	0	Output from flip-flop 0 of logic cell block L1 in the ELCL
1	1	1	0	1	Output from flip-flop 1 of logic cell block L1 in the ELCL
1	1	1	1	0	Output from flip-flop 1 of logic cell block L2 in the ELCL
1	1	1	1	1	Output from flip-flop 1 of logic cell block L3 in the ELCL

Figure 20 - 5 Format of Input Signal Select Registers n (ELISELn) (3/3)

ELISELn (n = 6 to 11)

ELISELn4	ELISELn3	ELISELn2	ELISELn1	ELISELn0	Event Source
0	0	0	0	0	Transmit data output from SAU0 channel 0
0	0	0	0	1	Transmit data output from SAU0 channel 1 **Note 1**
0	0	0	1	0	Serial clock output from SAU0 channel 0
0	0	0	1	1	Serial clock output from SAU0 channel 1 **Note 1**
0	0	1	0	0	Input pin (P11/EI11)
0	0	1	0	1	Input pin (P50/EI50)
0	0	1	1	0	Serial clock output from UARTA0 **Note 2**
0	0	1	1	1	Output from TAU0 channel 0
0	1	0	0	0	Output from TAU0 channel 1
0	1	0	0	1	Output from TAU0 channel 2
0	1	0	1	0	Output from TAU0 channel 3
0	1	0	1	1	Output from TAU0 channel 4
0	1	1	0	0	Output from TAU0 channel 5
0	1	1	0	1	Output from TAU0 channel 6
0	1	1	1	0	Output from TAU0 channel 7
0	1	1	1	1	Input pin (P10EI10)
1	0	0	0	0	Input pin (P51/EI51) **Note 3**
1	0	0	0	1	Comparator detection 0
1	0	0	1	0	Comparator detection 1 **Note 3**
1	0	0	1	1	Transmit data output 0 from UARTA0 **Note 2**
1	0	1	0	0	Input pin (P12/EI12)
1	0	1	0	1	INTC4 **Note 6**
1	0	1	1	0	INTC5 **Note 6**
1	0	1	1	1	INTC6 **Note 6**
1	1	0	0	0	INTC7 **Note 6**
1	1	0	0	1	INTC8 **Note 6**
1	1	0	1	0	INTC9 **Note 6**
1	1	0	1	1	f_{CLK}
1	1	1	0	0	f_{IHP}
1	1	1	0	1	f_{IMP}
1	1	1	1	0	f_{SXP}
1	1	1	1	1	Setting prohibited

Note 1. This is only available in the 48- to 128-pin R7F100G products.

Note 2. This is only available in the 36- to 128-pin R7F100G and 32- to 48-pin R7F104G products.

Note 3. This is only available in the 30- to 128-pin R7F100G products.

Note 4. The event sources selected for input pins 0 and 1 depend on which of the ELISELn (n = 0 to 5) registers is being set. For the presence or absence of input pins in different products, see **2.1 Functions of Port Pins**.

Register	Event Source	
	Input Pin 0	Input Pin 1
ELISEL0	Input pin (P20/EI20)	Input pin (P120/EI120)
ELISEL1	Input pin (P21/EI21)	Input pin (P121/EI121)
ELISEL2	Input pin (P22/EI22)	Input pin (P122/EI122)
ELISEL3	Input pin (P23/EI23)	Input pin (P147/EI147)
ELISEL4	Input pin (P30/EI30)	Input pin (P00/EI00)
ELISEL5	Input pin (P31/EI31)	Input pin (P01/EI01)

Note 5. The interrupt sources that are selectable as event sources for INTC0 to INTC3 depend on which of the ELISELn (n = 0 to 5) registers is being set. Use the interrupt request signals as the hardware triggers for event-receiving peripheral functions. For the presence or absence of interrupt sources in different products, see **21.2 Interrupt Sources and Configuration**.

Register	Event Source			
	INTC0	INTC1	INTC2	INTC3
ELISEL0	INTP6	INTURE0	INTTM16	INTUT1
		INTTM10		
ELISEL1	INTP7	INTURE1	INTTM17	INTUR1
		INTTM11		
ELISEL2	INTP8	INTTM12	INTST3/ INTCSI30/ INTIIC30	INTIICA1
ELISEL3	INTP9	INTSRE3	INTSR3/ INTCSI31/ INTIIC31	No connection
		INTTM13H		
ELISEL4	INTKR	INTTM14	INTUT0	No connection
ELISEL5	INTTM13	INTTM15	INTUR0	No connection

Note 6. The interrupt sources that are selectable as event sources for INTC4 to INTC9 depend on which of the ELISELn (n = 6 to 11) registers is being set. Use the interrupt request signals as the hardware triggers for event-receiving peripheral functions. For the presence or absence of interrupt sources in different products, see **21.2 Interrupt Sources and Configuration**.

Register	Event Source					
	INTC4	INTC5	INTC6	INTC7	INTC8	INTC9
ELISEL6	INTP0	INTTM00	INTTM06	INTST2/ INTCSI20/ INTIIC20	INTSR1/ INTCSI11/ INTIIC11	Setting prohibited
ELISEL7	INTP1	INTTM01	INTITL	INTSR2/ INTCSI21/ INTIIC21	INTSRE1	INTP10
					INTTM03H	INTCMP0
ELISEL8	INTP2	INTTM02	INTWDTI	INTSRE2	INTREMC	INTP11
				INTTM11H		INTCMP1
ELISEL9	INTP3	INTTM03	INTRTC	INTST0/ INTCSI00/ INTIIC00	INTSR0/ INTCSI01/ INTIIC01	INTCTSUWR
ELISEL10	INTP4	INTTM04	INTTM07	INTSRE0	INTLVI	INTCTSURD
				INTTM01H		
ELISEL11	INTP5	INTTM05	INTIICA0	INTST1/ INTCSI10/ INTIIC10	INTAD	INTCTSUFN

Caution 1. Be sure to leave bits 7 to 5 set to 0.

Caution 2. Setting of bits 4 to 0 of the ELISEL6 register to 11010B is prohibited.

Caution 3. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

Remark f_{CLK}: CPU/peripheral hardware clock frequency
f_{IHP}: High-speed on-chip oscillator peripheral clock frequency
f_{IMP}: Middle-speed on-chip oscillator peripheral clock frequency
f_{SXP}: Low-speed peripheral clock frequency

20.3.2 Event link L1 signal select registers 0 to 3 (ELL1SEL0 to ELL1SEL3)

These registers are used to select event signals to be linked to the input of the logic cells (pass-through, AND, OR, and EX-OR circuits), selector, and flip-flops in logic cell block L1. Specify an event signal from among those selected by input signal select registers n (ELISELn). These registers can be set by an 8-bit memory manipulation instruction. The value of each ELL1SELn register following a reset is 00H.

Figure 20 - 6 Format of Event Link L1 Signal Select Registers 0 to 3 (ELL1SEL0 to ELL1SEL3) (1/2)

Address: F0690H (ELL1SEL0) to F0693H (ELL1SEL3)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL1SELn	ELL1SELnZ	0	0	0	ELL1SELn3	ELL1SELn2	ELL1SELn1	ELL1SELn0

ELL1SELnZ	Control of event signal output level
0	Positive logic output
1	Negative logic output (inverted level)

ELL1SELn3	ELL1SELn2	ELL1SELn1	ELL1SELn0	Selection of target signal to be linked
0	0	0	0	No selection (fixed to 0).
0	0	0	1	Signal selected by input signal select register 0 is to be linked.
0	0	1	0	Signal selected by input signal select register 1 is to be linked.
0	0	1	1	Signal selected by input signal select register 2 is to be linked.
0	1	0	0	Signal selected by input signal select register 3 is to be linked.
0	1	0	1	Signal selected by input signal select register 4 is to be linked.
0	1	1	0	Signal selected by input signal select register 5 is to be linked.
0	1	1	1	Signal selected by input signal select register 6 is to be linked.
1	0	0	0	Signal selected by input signal select register 7 is to be linked.
1	0	0	1	Signal selected by input signal select register 8 is to be linked.
1	0	1	0	Signal selected by input signal select register 9 is to be linked.
1	0	1	1	Signal selected by input signal select register 10 is to be linked.

Figure 20 - 6 Format of Event Link L1 Signal Select Registers 0 to 3 (ELL1SEL0 to ELL1SEL3) (2/2)

ELL1SEL n3	ELL1SEL n2	ELL1SEL n1	ELL1SEL n0	Selection of target signal to be linked
1	1	0	0	Signal selected by input signal select register 11 is to be linked.
Others				Setting prohibited

Caution 1. Be sure to leave bits 6 to 4 set to 0.

Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

Remark n = 0 to 3

20.3.3 Event link L1 signal select registers 4, 5 (ELL1SEL4, ELL1SEL5)

These registers are used to select event signals to be linked to the set control and reset control of the flip-flops in logic cell block L1. Specify an event signal from among those selected by input signal select registers n (ELISELn). These registers can be set by an 8-bit memory manipulation instruction. The value of each ELL1SELn register following a reset is 00H.

Figure 20 - 7 Format of Event Link L1 Signal Select Registers 4, 5 (ELL1SEL4, ELL1SEL5)

Address: F0694H (ELL1SEL4), F0695H (ELL1SEL5)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL1SELn	ELL1SEL nZ	0	0	0	0	ELL1SEL n2	ELL1SEL n1	ELL1SEL n0

ELL1SEL nZ	Control of event signal output level
0	Positive logic output
1	Negative logic output (inverted level)

ELL1SEL n2	ELL1SEL n1	ELL1SEL n0	Selection of target signal to be linked
0	0	0	No selection (fixed to 0).
0	0	1	Signal selected by input signal select register 6 is to be linked.
0	1	0	Signal selected by input signal select register 7 is to be linked.
0	1	1	Signal selected by input signal select register 8 is to be linked.
1	0	0	Signal selected by input signal select register 9 is to be linked.
1	0	1	Signal selected by input signal select register 10 is to be linked.
1	1	0	Signal selected by input signal select register 11 is to be linked.
Others			Setting prohibited

Caution 1. Be sure to leave bits 6 to 3 set to 0.
Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

Remark n = 4, 5

20.3.4 Event link L1 signal select register 6 (ELL1SEL6)

This register is used to select an event signal to be linked to the clock of the flip-flops in logic cell block L1. Specify an event signal from among those selected by input signal select registers n (ELISELn). This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 20 - 8 Format of Event Link L1 Signal Select Register 6 (ELL1SEL6)

Address: F0696H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL1SEL6	ELL1SEL 6Z	0	0	0	0	ELL1SEL 62	ELL1SEL 61	ELL1SEL 60

ELL1SEL 6Z	Control of event signal output level
0	Positive logic output
1	Negative logic output (inverted level)

ELL1SEL 62	ELL1SEL 61	ELL1SEL 60	Selection of target signal to be linked
0	0	0	No selection (fixed to 0).
0	0	1	Signal selected by input signal select register 6 is to be linked.
0	1	0	Signal selected by input signal select register 7 is to be linked.
0	1	1	Signal selected by input signal select register 8 is to be linked.
1	0	0	Signal selected by input signal select register 9 is to be linked.
1	0	1	Signal selected by input signal select register 10 is to be linked.
1	1	0	Signal selected by input signal select register11 is to be linked.
Others			Setting prohibited

Caution 1. Be sure to leave bits 6 to 3 set to 0.
Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

20.3.5 Logic cell block L1 control register (ELL1CTL)

This register is used to select logic circuits to be used in the logic cells of logic cell block L1. Pass-through, an AND circuit, an OR circuit, or an EX-OR circuit can be selected for each logic cell. This register is also used to enable or stop flip-flops 0 and 1 in logic cell block L1. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 20 - 9 Format of Logic Cell Block L1 Control Register (ELL1CTL)

Address: F0697H (ELL1CTL)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL1CTL	ELL1T 31EN	ELL1T 30EN	0	0	ELL1T 111	ELL1T 110	ELL1T 101	ELL1T 100

ELL1T 31EN	Selection of enabling or stopping flip-flop 1 in logic cell block L1
0	The flip-flop is stopped (the flip-flop is in the reset state).
1	The flip-flop is enabled.

ELL1T 30EN	Selection of enabling or stopping flip-flop 0 in logic cell block L1
0	The flip-flop is stopped (the flip-flop is in the reset state).
1	The flip-flop is enabled.

ELL1T 111	ELL1T 110	Selection of the logic circuit to be used in logic cell 1 of logic cell block L1
0	0	Pass-through is selected.
0	1	AND circuit is selected.
1	0	OR circuit is selected.
1	1	EX-OR circuit is selected.

ELL1T 101	ELL1T 100	Selection of the logic circuit to be used in logic cell 0 of logic cell block L1
0	0	Pass-through is selected.
0	1	AND circuit is selected.
1	0	OR circuit is selected.
1	1	EX-OR circuit is selected.

Caution 1. Be sure to leave bits 5 and 4 set to 0.
Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

20.3.6 Event link L1 output select registers 0 to 3 (ELL1LNK0 to ELL1LNK3)

These registers are used to specify the destinations where the event signals selected by event link L1 signal select registers n (ELL1SELn) are to be linked. Select an input of a logic cell (pass-through, AND circuit, OR circuit, or EX-OR circuit), the selector, or a flip-flop in logic cell block L1 as a destination. These registers can be set by an 8-bit memory manipulation instruction. The value of each ELL1LNKn register following a reset is 00H.

Figure 20 - 10 Format of Event Link L1 Output Select Registers 0 to 3 (ELL1LNK0 to ELL1LNK3)

Address: F0698H (ELL1LNK0) to F069BH (ELL1LNK3)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL1LNKn	0	0	0	0	ELL1LNK n3	ELL1LNK n2	ELL1LNK n1	ELL1LNK n0

ELL1LNK n3	ELL1LNK n2	ELL1LNK n1	ELL1LNK n0	Selection of link destination
0	0	0	0	No link.
0	0	0	1	Linked to input 0 of logic cell 0 in logic cell block L1.
0	0	1	0	Linked to input 1 of logic cell 0 in logic cell block L1.
0	0	1	1	Linked to input 0 of logic cell 1 in logic cell block L1.
0	1	0	0	Linked to input 1 of logic cell 1 in logic cell block L1.
0	1	0	1	Linked to the selection control of the selector in logic cell block L1.
0	1	1	0	Linked to input 0 of the selector in logic cell block L1.
0	1	1	1	Linked to input 1 of the selector in logic cell block L1.
1	0	0	0	Linked to the input of flip-flop 0 in logic cell block L1.
1	0	0	1	Linked to the input of flip-flop 1 in logic cell block L1.
Others				Setting prohibited

Caution 1. Be sure to leave bits 7 to 4 set to 0.
Caution 2. Select different destinations for event link L1 signal selection blocks 0 to 6; do not connect two or more signals to a single destination.
Caution 3. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

Remark n = 0 to 3

20.3.7 Event link L1 output select registers 4, 5 (ELL1LNK4, ELL1LNK5)

These registers are used to specify the destinations where the event signals selected by event link L1 signal select registers n (ELL1SELn) are to be linked. The signals can be linked to the set control or reset control of the flip-flops in logic cell block L1. These registers can be set by an 8-bit memory manipulation instruction. The value of each ELL1LNKn register following a reset is 00H.

Figure 20 - 11 Format of Event Link L1 Output Select Registers 4, 5 (ELL1LNK4, ELL1LNK5)

Address: F069CH (ELL1LNK4), F069DH (ELL1LNK5)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL1LNKn	0	0	0	0	ELL1LNK n3	ELL1LNK n2	ELL1LNK n1	ELL1LNK n0

ELL1LNK n3	Control of the link to flip-flop in logic cell block L1
0	No link.
1	Linked to the reset control of flip-flop 1 in logic cell block L1.

ELL1LNK n2	Control of the link to flip-flop in logic cell block L1
0	No link.
1	Linked to the set control of flip-flop 1 in logic cell block L1.

ELL1LNK n1	Control of the link to flip-flop in logic cell block L1
0	No link.
1	Linked to the reset control of flip-flop 0 in logic cell block L1.

ELL1LNK n0	Control of the link to flip-flop in logic cell block L1
0	No link.
1	Linked to the set control of flip-flop 0 in logic cell block L1.

Caution 1. Be sure to leave bits 7 to 4 set to 0.
Caution 2. Do not connect a single signal to both the set control and reset control of flip-flop 0 or 1.
Caution 3. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

Remark n = 4, 5

20.3.8 Event link L1 output select register 6 (ELL1LNK6)

This register is used to specify the destinations where the event signal selected by event link L1 signal select register 6 (ELL1SEL6) is to be linked. The signal can be linked to the clock of the flip-flops in logic cell block L1. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 20 - 12 Format of Event Link L1 Output Select Register 6 (ELL1LNK6)

Address: F069EH (ELL1LNK6)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL1LNK6	0	0	0	0	0	0	ELL1LNK 61	ELL1LNK 60

ELL1LNK 61	Control of the link to flip-flop in logic cell block L1
0	No link.
1	Linked to the clock of flip-flop 1 in logic cell block L1.

ELL1LNK 60	Control of the link to flip-flop in logic cell block L1
0	No link.
1	Linked to the clock of flip-flop 0 in logic cell block L1.

Caution 1. Be sure to leave bits 7 to 2 set to 0.

Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

20.3.9 Event link L2 signal select registers 0 to 3 (ELL2SEL0 to ELL2SEL3)

These registers are used to select event signals to be linked to the input of the logic cells (pass-through, AND, OR, and EX-OR circuits) and selector, and the input, set control, and reset control of the flip-flops in logic cell block L2. Choose one of the event signals selected by input signal select registers n (ELISELn). In addition, the signals output from logic cell block L1 can be selected as event signals. These registers can be set by an 8-bit memory manipulation instruction. The value of each ELL2SELn register following a reset is 00H.

Remark n = 0 to 3

Figure 20 - 13 Format of Event Link L2 Signal Select Registers 0 to 3 (ELL2SEL0 to ELL2SEL3) (1/2)

Address: F06A0H (ELL2SEL0) to F06A3H (ELL2SEL3)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL2SELn	ELL2SEL nZ	0	0	ELL2SEL n4	ELL2SEL n3	ELL2SEL n2	ELL2SEL n1	ELL2SEL n0

ELL2SEL nZ	Control of event signal output level
0	Positive logic output
1	Negative logic output (inverted level)

ELL2SEL n4	ELL2SEL n3	ELL2SEL n2	ELL2SEL n1	ELL2SEL n0	Selection of target signal to be linked
0	0	0	0	0	No selection (fixed to 0).
0	0	0	0	1	Signal selected by input signal select register 0 is to be linked.
0	0	0	1	0	Signal selected by input signal select register 1 is to be linked.
0	0	0	1	1	Signal selected by input signal select register 2 is to be linked.
0	0	1	0	0	Signal selected by input signal select register 3 is to be linked.
0	0	1	0	1	Signal selected by input signal select register 4 is to be linked.
0	0	1	1	0	Signal selected by input signal select register 5 is to be linked.
0	0	1	1	1	Signal selected by input signal select register 6 is to be linked.
0	1	0	0	0	Signal selected by input signal select register 7 is to be linked.
0	1	0	0	1	Signal selected by input signal select register 8 is to be linked.
0	1	0	1	0	Signal selected by input signal select register 9 is to be linked.

Figure 20 - 13 Format of Event Link L2 Signal Select Registers 0 to 3 (ELL2SEL0 to ELL2SEL3) (2/2)

ELL2SEL n4	ELL2SEL n3	ELL2SEL n2	ELL2SEL n1	ELL2SEL n0	Selection of target signal to be linked
0	1	0	1	1	Signal selected by input signal select register 10 is to be linked.
0	1	1	0	0	Signal selected by input signal select register 11 is to be linked.
0	1	1	0	1	Output signal 0 from logic cell block L1 is to be linked.
0	1	1	1	0	Output signal 1 from logic cell block L1 is to be linked.
0	1	1	1	1	Output signal 2 from logic cell block L1 is to be linked.
1	0	0	0	0	Output signal 3 from logic cell block L1 is to be linked.
1	0	0	0	1	Output signal 4 from logic cell block L1 is to be linked.
Others					Setting prohibited

Caution 1. Be sure to leave bits 6 and 5 set to 0.

Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

Remark n = 0 to 3

20.3.10 Event link L2 signal select registers 4, 5 (ELL2SEL4, ELL2SEL5)

These registers are used to select event signals to be linked to the set control and reset control of the flip-flops in logic cell block L2. Specify an event signal from among those selected by input signal select registers n (ELISELn). These registers can be set by an 8-bit memory manipulation instruction. The value of each ELL2SELn register following a reset is 00H.

Figure 20 - 14 Format of Event Link L2 Signal Select Registers 4, 5 (ELL2SEL4, ELL2SEL5)

Address: F06A4H (ELL2SEL4), F06A5H (ELL2SEL5)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL2SELn	ELL2SEL nZ	0	0	0	0	ELL2SEL n2	ELL2SEL n1	ELL2SEL n0

ELL2SEL nZ	Control of event signal output level
0	Positive logic output
1	Negative logic output (inverted level)

ELL2SEL n2	ELL2SEL n1	ELL2SEL n0	Selection of target signal to be linked
0	0	0	No selection (fixed to 0).
0	0	1	Signal selected by input signal select register 6 is to be linked.
0	1	0	Signal selected by input signal select register 7 is to be linked.
0	1	1	Signal selected by input signal select register 8 is to be linked.
1	0	0	Signal selected by input signal select register 9 is to be linked.
1	0	1	Signal selected by input signal select register 10 is to be linked.
1	1	0	Signal selected by input signal select register 11 is to be linked.
Others			Setting prohibited

Caution 1. Be sure to leave bits 6 to 3 set to 0.
Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

Remark n = 4, 5

20.3.11 Event link L2 signal select register 6 (ELL2SEL6)

This register is used to select an event signal to be linked to the clock of the flip-flops in logic cell block L2. Specify an event signal from among those selected by input signal select registers n (ELISELn). This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 20 - 15 Format of Event Link L2 Signal Select Register 6 (ELL2SEL6)

Address: F06A6H (ELL2SEL6)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL2SEL6	ELL2SEL 6Z	0	0	0	0	ELL2SEL 62	ELL2SEL 61	ELL2SEL 60

ELL2SEL 6Z	Control of event signal output level
0	Positive logic output
1	Negative logic output (inverted level)

ELL2SEL 62	ELL2SEL 61	ELL2SEL 60	Selection of target signal to be linked
0	0	0	No selection (fixed to 0).
0	0	1	Signal selected by input signal select register 6Note is to be linked.
0	1	0	Signal selected by input signal select register 7Note is to be linked.
0	1	1	Signal selected by input signal select register 8Note is to be linked.
1	0	0	Signal selected by input signal select register 9Note is to be linked.
1	0	1	Signal selected by input signal select register 10Note is to be linked.
1	1	0	Signal selected by input signal select register 11Note is to be linked.
Others			Setting prohibited

Note Select a clock signal in the corresponding one of input signal select registers 6 to 11.

Caution 1. Be sure to leave bits 6 to 3 set to 0.

Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

20.3.12 Logic cell block L2 control register (ELL2CTL)

This register is used to select logic circuits to be used in the logic cells of logic cell block L2. Pass-through, an AND circuit, an OR circuit, or an EX-OR circuit can be selected for each logic cell. This register is also used to enable or stop flip-flops 0 and 1 in logic cell block L2. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 20 - 16 Format of Logic Cell Block L2 Control Register (ELL2CTL)

Address: F06A7H (ELL2CTL)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL2CTL	ELL2T 31EN	ELL2T 30EN	0	0	ELL2T 111	ELL2T 110	ELL2T 101	ELL2T 100

ELL2T 31EN	Selection of enabling or stopping flip-flop 1 in logic cell block L2
0	The flip-flop is stopped (the flip-flop is in the reset state).
1	The flip-flop is enabled.

ELL2T 30EN	Selection of enabling or stopping flip-flop 0 in logic cell block L2
0	The flip-flop is stopped (the flip-flop is in the reset state).
1	The flip-flop is enabled.

ELL2T 111	ELL2T 110	Selection of the logic circuit to be used in logic cell 1 of logic cell block L2
0	0	Pass-through is selected.
0	1	AND circuit is selected.
1	0	OR circuit is selected.
1	1	EX-OR circuit is selected.

ELL2T 101	ELL2T 100	Selection of the logic circuit to be used in logic cell 0 of logic cell block L2
0	0	Pass-through is selected.
0	1	AND circuit is selected.
1	0	OR circuit is selected.
1	1	EX-OR circuit is selected.

Caution 1. Be sure to leave bits 5 and 4 set to 0.
Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

20.3.13 Event link L2 output select registers 0 to 3 (ELL2LNK0 to ELL2LNK3)

These registers are used to specify the destinations where the event signals selected by event link L2 signal select registers n (ELL2SELn) are to be linked. Select an input of a logic cell (pass-through, AND circuit, OR circuit, or EX-OR circuit) or the selector, or the input, set control, or reset control of a flip-flop in logic cell block L2 as a destination. These registers can be set by an 8-bit memory manipulation instruction. The value of each ELL2LNKn register following a reset is 00H.

Figure 20 - 17 Format of Event Link L2 Output Select Registers 0 to 3 (ELL2LNK0 to ELL2LNK3)

Address: F06A8H (ELL2LNK0) to F06ABH (ELL2LNK3)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL2LNKn	0	0	0	0	ELL2LNKn3	ELL2LNKn2	ELL2LNKn1	ELL2LNKn0

ELL2LNKn3	ELL2LNKn2	ELL2LNKn1	ELL2LNKn0	Selection of link destination
0	0	0	0	No link.
0	0	0	1	Linked to input 0 of logic cell 0 in logic cell block L2.
0	0	1	0	Linked to input 1 of logic cell 0 in logic cell block L2.
0	0	1	1	Linked to input 0 of logic cell 1 in logic cell block L2.
0	1	0	0	Linked to input 1 of logic cell 1 in logic cell block L2.
0	1	0	1	Linked to the selection control of the selector in logic cell block L2.
0	1	1	0	Linked to input 0 of the selector in logic cell block L2.
0	1	1	1	Linked to input 1 of the selector in logic cell block L2.
1	0	0	0	Linked to the input of flip-flop 0 in logic cell block L2.
1	0	0	1	Linked to the input of flip-flop 1 in logic cell block L2.
1	0	1	0	Linked to the set control of flip-flop 0 in logic cell block L2.
1	0	1	1	Linked to the reset control of flip-flop 0 in logic cell block L2.
Others				Setting prohibited

Caution 1. Be sure to leave bits 7 to 4 set to 0.
Caution 2. Select different destinations for event link L2 signal selection blocks 0 to 6; do not connect two or more signals to a single destination.
Caution 3. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

Remark n = 0 to 3

20.3.14 Event link L2 output select registers 4, 5 (ELL2LNK4, ELL2LNK5)

These registers are used to specify the destinations where the event signals selected by event link L2 signal select registers n (ELL2SELn) are to be linked. The signals can be linked to the set control or reset control of a flip-flop in logic cell block L2. These registers can be set by an 8-bit memory manipulation instruction. The value of each ELL2LNKn register following a reset is 00H.

Figure 20 - 18 Format of Event Link L2 Output Select Registers 4, 5 (ELL2LNK4, ELL2LNK5)

Address: F06ACH (ELL2LNK4), F06ADH (ELL2LNK5)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL2LNKn	0	0	0	0	0	0	0	ELL2LNK n0

ELL2LNK5 register

ELL2LNK 50	Control of the link to flip-flop in logic cell block L2
0	No link.
1	Linked to the reset control of flip-flop 1 in logic cell block L2.

ELL2LNK4 register

ELL2LNK 40	Control of the link to flip-flop in logic cell block L2
0	No link.
1	Linked to the set control of flip-flop 1 in logic cell block L2.

Caution 1. Be sure to leave bits 7 to 1 set to 0.
Caution 2. Do not connect a single signal to both the set control and reset control of flip-flop 0 or 1.
Caution 3. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

Remark n = 4, 5

20.3.15 Event link L2 output select register 6 (ELL2LNK6)

This register is used to specify the destinations where the event signal selected by event link L2 signal select register 6 (ELL2SEL6) is to be linked. The signal can be linked to the clock of the flip-flops in logic cell block L2. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 20 - 19 Format of Event Link L2 Output Select Register 6 (ELL2LNK6)

Address: F06AEH (ELL2LNK6)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL2LNK6	0	0	0	0	0	0	ELL2LNK61	ELL2LNK60

ELL2LNK61	Control of the link to flip-flop in logic cell block L2
0	No link.
1	Linked to the clock of flip-flop 1 in logic cell block L2.

ELL2LNK60	Control of the link to flip-flop in logic cell block L2
0	No link.
1	Linked to the clock of flip-flop 0 in logic cell block L2.

Caution 1. Be sure to leave bits 7 to 2 set to 0.

Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

20.3.16 Event link L3 signal select registers 0 to 3 (ELL3SEL0 to ELL3SEL3)

These registers are used to select event signals to be linked to the input of the logic cells (pass-through, AND, OR, and EX-OR circuits) and selector, and the input, set control, and reset control of the flip-flops in logic cell block L3. Specify an event signal from among those selected by input signal select registers n (ELISELn). In addition, the signals output from logic cell block L2 can be selected as event signals. These registers can be set by an 8-bit memory manipulation instruction. The value of each ELL3SELn register following a reset is 00H.

Remark n = 0 to 3

Figure 20 - 20 Format of Event Link L3 Signal Select Registers 0 to 3 (ELL3SEL0 to ELL3SEL3) (1/2)

Address: F06B0H (ELL3SEL0) to F06B3H (ELL3SEL3)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL3SELn	ELL3SEL nZ	0	0	ELL3SEL n4	ELL3SEL n3	ELL3SEL n2	ELL3SEL n1	ELL3SEL n0

ELL3SEL nZ	Control of event signal output level
0	Positive logic output
1	Negative logic output (inverted level)

ELL3SEL n4	ELL3SEL n3	ELL3SEL n2	ELL3SEL n1	ELL3SEL n0	Selection of target signal to be linked
0	0	0	0	0	No selection (fixed to 0).
0	0	0	0	1	Signal selected by input signal select register 0 is to be linked.
0	0	0	1	0	Signal selected by input signal select register 1 is to be linked.
0	0	0	1	1	Signal selected by input signal select register 2 is to be linked.
0	0	1	0	0	Signal selected by input signal select register 3 is to be linked.
0	0	1	0	1	Signal selected by input signal select register 4 is to be linked.
0	0	1	1	0	Signal selected by input signal select register 5 is to be linked.
0	0	1	1	1	Signal selected by input signal select register 6 is to be linked.
0	1	0	0	0	Signal selected by input signal select register 7 is to be linked.
0	1	0	0	1	Signal selected by input signal select register 8 is to be linked.
0	1	0	1	0	Signal selected by input signal select register 9 is to be linked.

Figure 20 - 20 Format of Event Link L3 Signal Select Registers 0 to 3 (ELL3SEL0 to ELL3SEL3) (2/2)

ELL3SEL n4	ELL3SEL n3	ELL3SEL n2	ELL3SEL n1	ELL3SEL n0	Selection of target signal to be linked
0	1	0	1	1	Signal selected by input signal select register 10 is to be linked.
0	1	1	0	0	Signal selected by input signal select register 11 is to be linked.
0	1	1	0	1	Output signal 0 from logic cell block L2 is to be linked.
0	1	1	1	0	Output signal 1 from logic cell block L2 is to be linked.
0	1	1	1	1	Output signal 2 from logic cell block L2 is to be linked.
1	0	0	0	0	Output signal 3 from logic cell block L2 is to be linked.
1	0	0	0	1	Output signal 4 from logic cell block L2 is to be linked.
Others					Setting prohibited

Caution 1. Be sure to leave bits 6 and 5 set to 0.

Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

Remark n = 0 to 3

20.3.17 Event link L3 signal select registers 4, 5 (ELL3SEL4, ELL3SEL5)

These registers are used to select event signals to be linked to the set control and reset control of the flip-flops in logic cell block L3. Specify an event signal from among those selected by input signal select registers n (ELISELn). These registers can be set by an 8-bit memory manipulation instruction. The value of each ELL3SELn register following a reset is 00H.

Figure 20 - 21 Format of Event Link L3 Signal Select Registers 4, 5 (ELL3SEL4, ELL3SEL5)

Address: F06B4H (ELL3SEL4), F06B5H (ELL3SEL5)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL3SELn	ELL3SEL nZ	0	0	0	0	ELL3SEL n2	ELL3SEL n1	ELL3SEL n0

ELL3SEL nZ	Control of event signal output level
0	Positive logic output
1	Negative logic output (inverted level)

ELL3SEL n2	ELL3SEL n1	ELL3SEL n0	Selection of target signal to be linked
0	0	0	No selection (fixed to 0).
0	0	1	Signal selected by input signal select register 6 is to be linked.
0	1	0	Signal selected by input signal select register 7 is to be linked.
0	1	1	Signal selected by input signal select register 8 is to be linked.
1	0	0	Signal selected by input signal select register 9 is to be linked.
1	0	1	Signal selected by input signal select register 10 is to be linked.
1	1	0	Signal selected by input signal select register 11 is to be linked.
Others			Setting prohibited

Caution 1. Be sure to leave bits 6 to 3 set to 0.
Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

Remark n = 4, 5

20.3.18 Event link L3 signal select register 6 (ELL3SEL6)

This register is used to select an event signal to be linked to the clock of the flip-flops in logic cell block L3. Specify an event signal from among those selected by input signal select registers n (ELISELn). This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 20 - 22 Format of Event Link L3 Signal Select Register 6 (ELL3SEL6)

Address: F06B6H (ELL3SEL6)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL3SEL6	ELL3SEL 6Z	0	0	0	0	ELL3SEL 62	ELL3SEL 61	ELL3SEL 60

ELL3SEL 6Z	Control of event signal output level
0	Positive logic output
1	Negative logic output (inverted level)

ELL3SEL 62	ELL3SEL 61	ELL3SEL 60	Selection of target signal to be linked
0	0	0	No selection (fixed to 0).
0	0	1	Signal selected by input signal select register 6Note is to be linked.
0	1	0	Signal selected by input signal select register 7Note is to be linked.
0	1	1	Signal selected by input signal select register 8Note is to be linked.
1	0	0	Signal selected by input signal select register 9Note is to be linked.
1	0	1	Signal selected by input signal select register 10Note is to be linked.
1	1	0	Signal selected by input signal select register 11Note is to be linked.
1	1	1	No selection (fixed to 0).

Note Select a clock signal in the corresponding one of input signal select registers 6 to 11.

Caution 1. Be sure to leave bits 6 to 3 set to 0.

Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

20.3.19 Logic cell block L3 control register (ELL3CTL)

This register is used to select logic circuits to be used in the logic cells of logic cell block L3. Pass-through, an AND circuit, an OR circuit, or an EX-OR circuit can be selected for each logic cell. This register is also used to enable or stop flip-flops 0 and 1 in logic cell block L3. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 20 - 23 Format of Logic Cell Block L3 Control Register (ELL3CTL)

Address: F06B7H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL3CTL	ELL3T 31EN	ELL3T 30EN	0	0	ELL3T 111	ELL3T 110	ELL3T 101	ELL3T 100

ELL3T 31EN	Selection of enabling or stopping flip-flop 1 in logic cell block L3
0	The flip-flop is stopped (the flip-flop is in the reset state).
1	The flip-flop is enabled.

ELL3T 30EN	Selection of enabling or stopping flip-flop 0 in logic cell block L3
0	The flip-flop is stopped (the flip-flop is in the reset state).
1	The flip-flop is enabled.

ELL3T 111	ELL3T 110	Selection of the logic circuit to be used in logic cell 1 of logic cell block L3
0	0	Pass-through is selected.
0	1	AND circuit is selected.
1	0	OR circuit is selected.
1	1	EX-OR circuit is selected.

ELL3T 101	ELL3T 100	Selection of the logic circuit to be used in logic cell 0 of logic cell block L3
0	0	Pass-through is selected.
0	1	AND circuit is selected.
1	0	OR circuit is selected.
1	1	EX-OR circuit is selected.

Caution 1. Be sure to leave bits 5 and 4 set to 0.
Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

20.3.20 Event link L3 output select registers 0 to 3 (ELL3LNK0 to ELL3LNK3)

These registers are used to specify the destinations where the event signals selected by event link L3 signal select registers n (ELL3SELn) are to be linked. Select an input of a logic cell (pass-through, AND circuit, OR circuit, or EX-OR circuit) or the selector, or the input, set control, or reset control of a flip-flop in logic cell block L3 as a destination. These registers can be set by an 8-bit memory manipulation instruction. The value of each ELL3LNKn register following a reset is 00H.

Figure 20 - 24 Format of Event Link L3 Output Select Registers 0 to 3 (ELL3LNK0 to ELL3LNK3)

Address: F06B8H (ELL3LNK0) to F06BBH (ELL3LNK3)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL3LNKn	0	0	0	0	ELL3LNK n3	ELL3LNK n2	ELL3LNK n1	ELL3LNK n0

ELL3LNK n3	ELL3LNK n2	ELL3LNK n1	ELL3LNK n0	Selection of link destination
0	0	0	0	No link.
0	0	0	1	Linked to input 0 of logic cell 0 in logic cell block L3.
0	0	1	0	Linked to input 1 of logic cell 0 in logic cell block L3.
0	0	1	1	Linked to input 0 of logic cell 1 in logic cell block L3.
0	1	0	0	Linked to input 1 of logic cell 1 in logic cell block L3.
0	1	0	1	Linked to the selection control of the selector in logic cell block L3.
0	1	1	0	Linked to input 0 of the selector in logic cell block L3.
0	1	1	1	Linked to input 1 of the selector in logic cell block L3.
1	0	0	0	Linked to the input of flip-flop 0 in logic cell block L3.
1	0	0	1	Linked to the input of flip-flop 1 in logic cell block L3.
1	0	1	0	Linked to the set control of flip-flop 0 in logic cell block L3.
1	0	1	1	Linked to the reset control of flip-flop 0 in logic cell block L3.
Others				Setting prohibited

Caution 1. Be sure to leave bits 7 to 4 set to 0.
Caution 2. Select different destinations for event link L3 signal selection blocks 0 to 6; do not connect two or more signals to a single destination.
Caution 3. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

Remark n = 0 to 3

20.3.21 Event link L3 output select registers 4, 5 (ELL3LNK4, ELL3LNK5)

These registers are used to specify the destinations where the event signals selected by event link L3 signal select registers n (ELL3SELn) are to be linked. The signals can be linked to the set control or reset control of a flip-flop in logic cell block L3. These registers can be set by an 8-bit memory manipulation instruction. The value of each ELL3LNKn register following a reset is 00H.

Figure 20 - 25 Format of Event Link L3 Output Select Registers 4, 5 (ELL3LNK4, ELL3LNK5)

Address: F06BCH (ELL3LNK4), F06BDH (ELL3LNK5)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL3LNKn	0	0	0	0	0	0	0	ELL3LNKn0

ELL3LNK5 register

ELL3LNK50	Control of the link to flip-flop in logic cell block L3
0	No link.
1	Linked to the reset control of flip-flop 1 in logic cell block L3.

ELL3LNK4 register

ELL3LNK40	Control of the link to flip-flop in logic cell block L3
0	No link.
1	Linked to the set control of flip-flop 1 in logic cell block L3.

Caution 1. Be sure to leave bits 7 to 1 set to 0.
Caution 2. Do not connect a single signal to both the set control and reset control of flip-flop 0 or 1.
Caution 3. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

Remark n = 4, 5

20.3.22 Event link L3 output select register 6 (ELL3LNK6)

This register is used to specify the destinations where the event signal selected by event link L3 signal select register 6 (ELL3SEL6) is to be linked. The signal can be linked to the clock of the flip-flops in logic cell block L3. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 20 - 26 Format of Event Link L3 Output Select Register 6 (ELL3LNK6)

Address: F06BEH (ELL3LNK6)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELL3LNK6	0	0	0	0	0	0	ELL3LNK61	ELL3LNK60

ELL3LNK61	Control of the link to flip-flop in logic cell block L3
0	No link.
1	Linked to the clock of flip-flop 1 in logic cell block L3.

ELL3LNK60	Control of the link to flip-flop in logic cell block L3
0	No link.
1	Linked to the clock of flip-flop 0 in logic cell block L3.

Caution 1. Be sure to leave bits 7 to 2 set to 0.

Caution 2. Do not change the setting of this register while the output signal enable register (ELOENCTL) is set to enable the output of a signal.

20.3.23 Output signal select registers n (ELOSELn)

These registers are used to select signals to be output to the event-receiving peripheral functions. Select a signal from among those output from logic cell blocks L1 to L3. These registers can be set by an 8-bit memory manipulation instruction. The value of each ELOSELn register following a reset is 00H.

Remark n = 0 to 7

Figure 20 - 27 Format of Output Signal Select Registers n (ELOSELn) (1/2)

Address: F06C0H (ELOSEL0) to F06C7H (ELOSEL7)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELOSELn	ELOSELnZ	0	0	0	ELOSELn3	ELOSELn2	ELOSELn1	ELOSELn0

ELOSELnZ	Control of output signal level
0	Positive logic output
1	Negative logic output (inverted level)

ELOSELn3	ELOSELn2	ELOSELn1	ELOSELn0	Selection of output signal
0	0	0	0	No selection.
0	0	0	1	Output signal 0 from logic cell block L1 (output of logic cell 0) is selected.
0	0	1	0	Output signal 1 from logic cell block L1 (output of logic cell 1) is selected.
0	0	1	1	Output signal 2 from logic cell block L1 (output of selector) is selected.
0	1	0	0	Output signal 3 from logic cell block L1 (output of flip-flop 0) is selected.
0	1	0	1	Output signal 4 from logic cell block L1 (output of flip-flop 1) is selected.
0	1	1	0	Output signal 0 from logic cell block L2 (output of logic cell 0) is selected.
0	1	1	1	Output signal 1 from logic cell block L2 (output of logic cell 1) is selected.
1	0	0	0	Output signal 2 from logic cell block L2 (output of selector) is selected.
1	0	0	1	Output signal 3 from logic cell block L2 (output of flip-flop 0) is selected.
1	0	1	0	Output signal 4 from logic cell block L2 (output of flip-flop 1) is selected.
1	0	1	1	Output signal 0 from logic cell block L3 (output of logic cell 0) is selected.
1	1	0	0	Output signal 1 from logic cell block L3 (output of logic cell 1) is selected.
1	1	0	1	Output signal 2 from logic cell block L3 (output of selector) is selected.

Figure 20 - 27 Format of Output Signal Select Registers n (ELOSELn) (2/2)

ELOSELn3	ELOSELn2	ELOSELn1	ELOSELn0	Selection of output signal
1	1	1	0	Output signal 3 from logic cell block L3 (output of flip-flop 0) is selected.
1	1	1	1	Output signal 4 from logic cell block L3 (output of flip-flop 1) is selected.

Caution **Be sure to leave bits 6 to 4 set to 0.**

Remark n = 0 to 7

20.3.24 Output signal enable register (ELOENCTL)

This register enables or disables the output of the signals selected by the ELOSELn registers. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 20 - 28 Format of Output Signal Enable Register (ELOENCTL)

Address: F06C8H (ELOENCTL)
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ELOENCTL	ELOEN CTL7	ELOEN CTL6	ELOEN CTL5	ELOEN CTL4	ELOEN CTL3	ELOEN CTL2	ELOEN CTL1	ELOEN CTL0

ELOEN CTLn	Enable or disable of the output of signal [n]
0	Output is disabled.
1	Output is enabled.**Note**

Note Only make the setting to enable the output of a signal after confirming that the event source signal is in a state which does not affect the operation of the peripheral function at the event destination.

Table 20 - 7 shows the correspondence between the bits in the output signal enable register and the event-receiving peripheral functions. Each signal output from the ELCL is connected to multiple destinations. When a destination does not use the signal from the ELCL, make settings in the destination so that the signal from the ELCL is not selected as an input signal for the destination. For the selection of input signals, refer to the section of each destination peripheral function.

Table 20 - 7 Bits in Output Signal Enable Register and Event-receiving Peripheral Functions

Register and Bit Name	Event-receiving Peripheral Function					
	Destination 1	Destination 2	Destination 3	Destination 4	Destination 5	Destination 6
ELOENCTL. ELOENCTL0	Output pin (P50/EO50)	Output pin (P01/EO01)**Note 1**	Event link interrupt (INTELCL)**Note 2**	Output pin (P10/EO10)	—	—
ELOENCTL. ELOENCTL1	Output pin (P51/EO51)**Note 3**	Output pin (P60/EO60)**Note 3**	DTC activation trigger**Note 2**	Output pin (P11/EO11)	—	—
ELOENCTL. ELOENCTL2	Serial clock input to SAU0 channel 0	Output pin (P61/EO61)**Note 3**	SMS activation trigger**Note 2**	Output pin (P12/EO12)	—	—
ELOENCTL. ELOENCTL3	Serial clock input to SAU0 channel 1**Note 4**	Timer input to TAU0 channel 0**Note 5**	—	Output pin (P13/EO13)**Note 1**	—	—
ELOENCTL. ELOENCTL4	Receive data input to SAU0 channel 0	Timer input to TAU0 channel 1**Note 5**	—	Output pin (P14/EO14)**Note 6**	—	—
ELOENCTL. ELOENCTL5	Receive data input to SAU0 channel 1**Note 4**	—	—	Output pin (P15/EO15)**Note 3**	Hardware trigger for A/D converter **Note 2**	Clock input to UARTA (UARTA0, UARTA1)**Note 7**
ELOENCTL. ELOENCTL6	—	Timer input to TAU0 channel 5 **Note 5**	—	Output pin (P16/EO16)	Hardware trigger for D/A converter 0**Note 2**	Hardware trigger for CTSU**Note 2**
ELOENCTL. ELOENCTL7	—	—	—	Output pin (P17/EO17)	Hardware trigger for D/A converter 1**Note 2**	TML32 operating clock, capture trigger**Note 2**

Note 1. This is only available in the 30- to 128-pin R7F100G and 44- and 48-pin R7F104G products.

Note 2. In hardware trigger operation, the rising edge is detected as a trigger.

Note 3. This is only available in the 30- to 128-pin R7F100G products.

Note 4. This is only available in the 48- to 128-pin R7F100G products.

Note 5. The widths at high and low level of the timer array unit to be linked should be at least 1/f_{MCK} + 10 ns. Accordingly, make the following settings to select an operating clock with a frequency which matches that requirement: use timer clock select register 0 (TPS0) to select the operating clock, and use noise filter enable register 1 (NFEN1) to enable or disable the noise filter.

Note 6. This is only available in the 30- to 128-pin R7F100G and 48-pin R7F104G products.

Note 7. This is only available in the 36- to 128-pin R7F100G and 32- to 48-pin R7F104G products.

20.3.25 Output signal monitor register (ELOMONI)

This register is used to monitor output signals n selected by the ELOSELn registers. Each bit returns the level of the corresponding output signal when this register is read. This register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Remark n = 0 to 7

Figure 20 - 29 Format of Output Signal Monitor Register (ELOMONI)

Address: F06C9H
After reset: 00H
R/W: R

Symbol	7	6	5	4	3	2	1	0
ELOMONI	ELOMONI7	ELOMONI6	ELOMONI5	ELOMONI4	ELOMONI3	ELOMONI2	ELOMONI1	ELOMONI0

ELOMONIn	Level of output signal n
0	The signal level is 0.
1	The signal level is 1.

Caution Depending on the timing with which this register is read, the value read for output signal n may not be correct. If the value read is incorrect, wait for enough time to confirm the stability of output signal n and then read the register again.

20.3.26 Registers for controlling the port functions multiplexed with the I/O pins of the logic and event link controller

Set the following registers to control the port functions multiplexed with the I/O pins of the logic and event link controller.

- Port mode registers (PMxx)
- Port output mode registers (POMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)

For details, see the following sections.

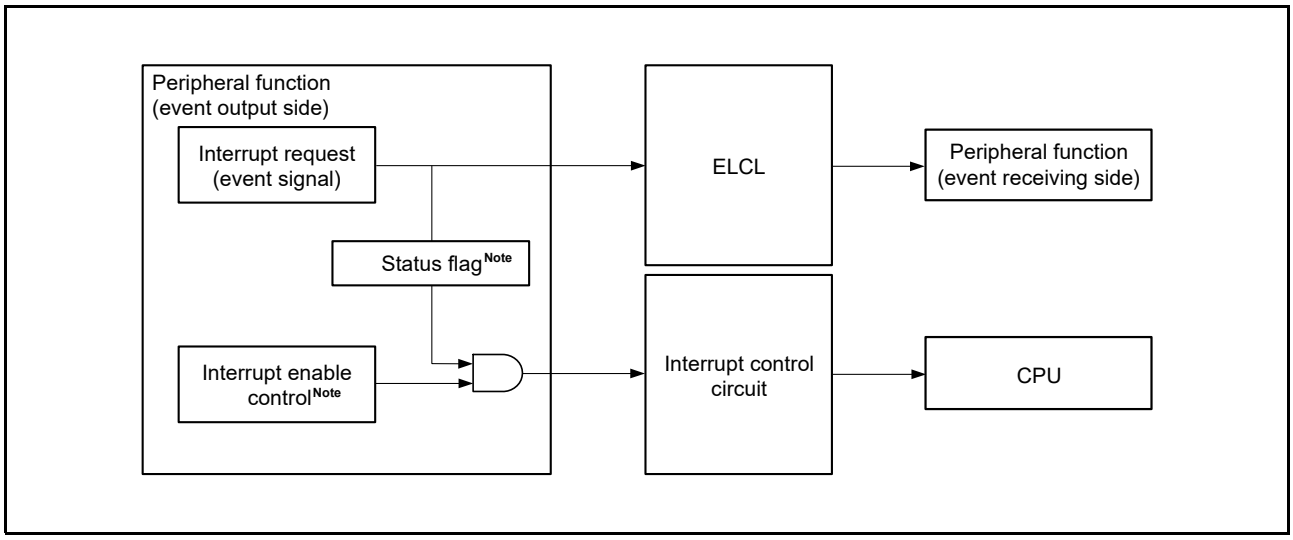
- **4.3.1 Port mode registers (PMxx)**
- **4.3.5 Port output mode registers (POMxx)**
- **4.3.7 Port mode control A registers (PMCAxx)**
- **4.3.8 Port mode control T registers (PMCTxx)**
- **4.3.9 Port mode control E registers (PMCExx)**

For examples of setting the registers for controlling the port functions, see **4.5.4 Examples of register settings for port and multiplexed functions**.

20.4 ELCL Operation

The ELCL inputs event signals from peripheral functions to logic cell blocks to convert them to signals that match the desired purpose before sending them to the event-receiving peripheral functions. A logic cell block has logic cells (pass-through, AND, OR, and EX-OR circuits), a selector, and flip-flops. The path for sending an event signal generated by a peripheral function to the interrupt control circuit as an interrupt request is separated from the path for using the signal as an ELCL event. Therefore, the interrupt function settings do not affect the use of an interrupt request signal as an ELCL event signal. For example, an interrupt request signal for which use as a trigger for an interrupt service routing has been disabled by using the corresponding bit of the interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H), can still be used as an ELCL event signal. **Figure 20 - 30** shows the relationship between interrupt service routing and the ELCL. The figure shows an example of a peripheral function that has an interrupt request status flag and an enable bit for controlling enabling and disabling of the interrupt. A peripheral function that receives an event from the ELCL performs the operation specific to the peripheral function after reception of an event.

Figure 20 - 30 Relationship between Interrupt Service Routing and ELCL

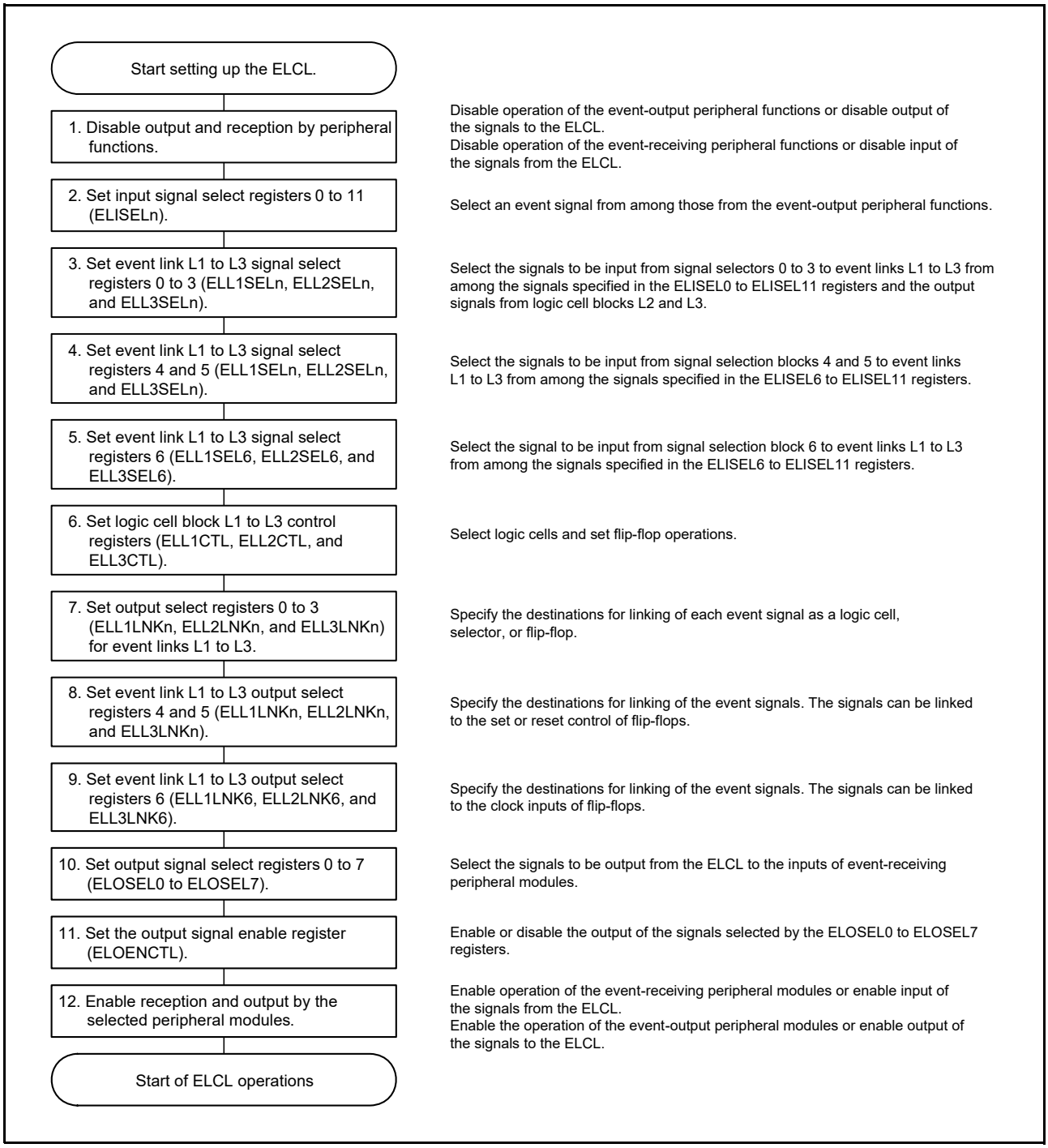


Note Some peripheral functions do not have status flags or interrupt enable control.

20.5 Procedure of Setting the ELCL

Figure 20 - 31 shows a procedure of setting the ELCL. Set up the registers in the ELCL while output by the event-output peripheral functions and reception by the event-receiving peripheral functions are disabled. After enabling output in step 11, enable reception by the event-receiving peripheral functions, and then enable output by the event-output peripheral functions.

Figure 20 - 31 Procedure of Setting the ELCL



1. Example of ELCL settings (OR circuit)
   Figure 20 - 32 shows an OR circuit configuration as an example of the result of ELCL settings. **Table 20 - 8** is a list of the register settings for this example.

Figure 20 - 32 Example of ELCL Settings (OR Circuit)

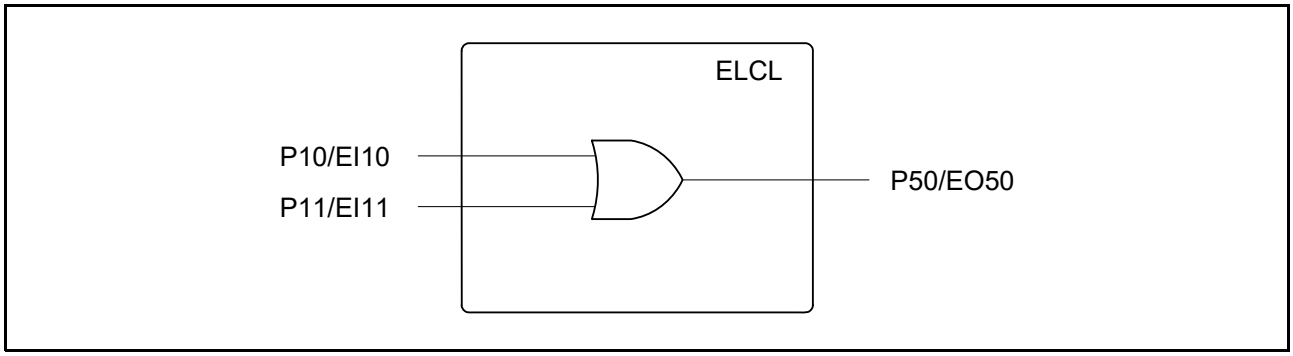


Table 20 - 8 Example of ELCL Register Settings (OR Circuit)

No.	Register Symbol	Register Name	Initial Value	Setting	Description
1	PMCE5	—	00H	00H	A signal other than the ELCL signal is selected for output from the P50 pin.
2	ELISEL0	Input signal select register 0	00H	0FH	P10/EI10 is selected.
2	ELISEL1	Input signal select register 1	00H	04H	P11/EI11 is selected.
3	ELL1SEL0	Event link L1 signal select register 0	00H	01H	Input signal selector 0 is selected.
3	ELL1SEL1	Event link L1 signal select register 1	00H	02H	Input signal selector 1 is selected.
6	ELL1CTL	Logic cell block L1 control register	00H	02H	The OR circuit is selected in logic cell 0.
9	ELL1LNK0	Event link L1 output select register 0	00H	01H	Linked to 0 of logic cell block L1.
9	ELL1LNK1	Event link L1 output select register 1	00H	02H	Linked to 1 of logic cell block L1.
10	ELOSEL0	Output signal select register 0	00H	01H	Output signal [0] from logic cell block L1 is selected.
11	ELOENCTL	Output signal enable register	00H	01H	Output [0] is enabled. The signal is output to P50/EO50.
12	PMCE5	—	00H	01H	The signal from the ELCL is selected for the output from the P50 pin.

Caution 1. The numbers in the No. column correspond to the step numbers (1 to 12) in Figure 20 - 31 Procedure of Setting the ELCL.

Caution 2. Steps 2 to 10 can be done in any order, but modifying the register values specified in steps 2 to 10 is prohibited after the output signal enable register is set up in step 11.

2. Example of ELCL settings (AND circuit)
Figure 20 - 33 shows an AND circuit configuration as an example of the result of ELCL settings. **Table 20 - 9** is a list of the register settings for this example.

Figure 20 - 33 Example of ELCL Settings (AND Circuit)

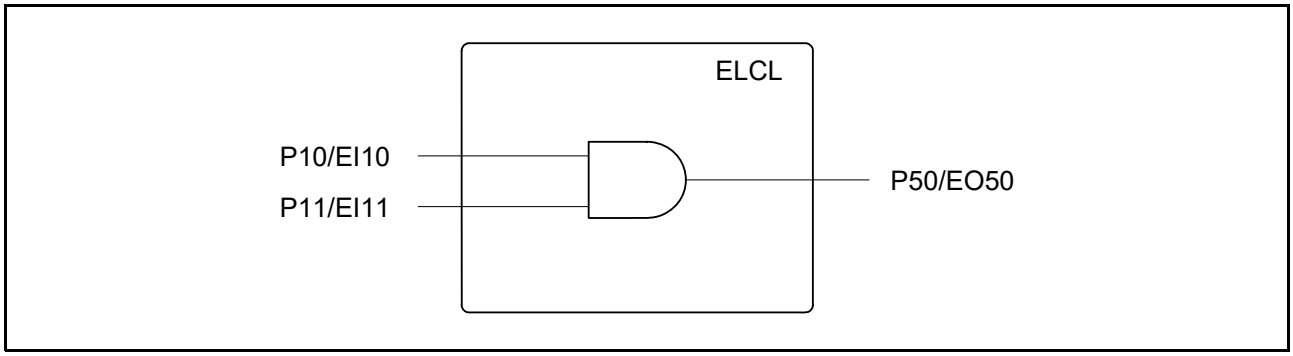


Table 20 - 9 Example of ELCL Register Settings (AND Circuit)

No.	Register Symbol	Register Name	Initial Value	Setting	Description
1	PMCE5	—	00H	00H	A signal other than the ELCL signal is selected for output from the P50 pin.
2	ELISEL0	Input signal select register 0	00H	0FH	P10/EI10 is selected.
2	ELISEL1	Input signal select register 1	00H	04H	P11/EI11 is selected.
3	ELL1SEL0	Event link L1 signal select register 0	00H	01H	Input signal selector 0 is selected.
3	ELL1SEL1	Event link L1 signal select register 1	00H	02H	Input signal selector 1 is selected.
6	ELL1CTL	Logic cell block L1 control register	00H	01H	The AND circuit is selected in logic cell 0.
9	ELL1LNK0	Event link L1 output select register 0	00H	01H	Linked to 0 of logic cell block L1.
9	ELL1LNK1	Event link L1 output select register 1	00H	02H	Linked to 1 of logic cell block L1.
10	ELOSEL0	Output signal select register 0	00H	01H	Output signal [0] from logic cell block L1 is selected.
11	ELOENCTL	Output signal enable register	00H	01H	Output [0] is enabled. The signal is output to P50/EO50.
12	PMCE5	—	00H	01H	The signal from the ELCL is selected for the output from the P50 pin.

Caution 1. The numbers in the No. column correspond to the step numbers (1 to 12) in Figure 20 - 31 Procedure of Setting the ELCL.

Caution 2. Steps 2 to 10 can be done in any order, but modifying the register values specified in steps 2 to 10 is prohibited after the output signal enable register is set up in step 11.

20.6 Points for Caution when the ELCL is to be Used

1. When the SO00, SO01, TxD0, SCK00, SCK01, SCL00, SCL01, TxDA0, CLKA0, or TO00 to TO07 is to be selected as an input to the ELCL but not for output to a pin, set the corresponding PFOEmn bit (m = 0, 1, n = 0 to 7) in the PFOEm register to 0. For details of the setting, see **Section 4 Port Functions**.

2. Each signal output from the ELCL is linked to multiple peripheral functions as shown in **Table 20 - 7**. When a destination peripheral function is not to use a signal output from the ELCL, make settings for the destination peripheral function so that the signal from the ELCL is not selected as an input signal at the destination. For details on how to select the signal from the ELCL, refer to the sections on each of the destination peripheral functions.

3. Do not change the values of the following registers after output of signals is enabled in the output signal enable register (ELOENCTL).
   – Input signal select registers n (ELISELn) (n = 0 to 11)
   – Event link L1 signal select registers 0 to 3 (ELL1SEL0 to ELL1SEL3)
   – Event link L1 signal select registers 4, 5 (ELL1SEL4, ELL1SEL5)
   – Event link L1 signal select register 6 (ELL1SEL6)
   – Logic cell block L1 control register (ELL1CTL)
   – Event link L1 output select registers 0 to 3 (ELL1LNK0 to ELL1LNK3)
   – Event link L1 output select registers 4, 5 (ELL1LNK4, ELL1LNK5)
   – Event link L1 output select register 6 (ELL1LNK6)
   – Event link L2 signal select registers 0 to 3 (ELL2SEL0 to ELL2SEL3)
   – Event link L2 signal select registers 4, 5 (ELL2SEL4, ELL2SEL5)
   – Event link L2 signal select register 6 (ELL2SEL6)
   – Logic cell block L2 control register (ELL2CTL)
   – Event link L2 output select registers 0 to 3 (ELL2LNK0 to ELL2LNK3)
   – Event link L2 output select registers 4, 5 (ELL2LNK4, ELL2LNK5)
   – Event link L2 output select register 6 (ELL2LNK6)
   – Event link L3 signal select registers 0 to 3 (ELL3SEL0 to ELL3SEL3)
   – Event link L3 signal select registers 4, 5 (ELL3SEL4, ELL3SEL5)
   – Event link L3 signal select register 6 (ELL3SEL6)
   – Logic cell block L3 control register (ELL3CTL)
   – Event link L3 output select registers 0 to 3 (ELL3LNK0 to ELL3LNK3)
   – Event link L3 output select registers 4, 5 (ELL3LNK4, ELL3LNK5)
   – Event link L3 output select register 6 (ELL3LNK6)
   – Output signal select registers n (ELOSELn) (n = 0 to 7)

4. The ELCL outputs signals with the use of multiple input signals, logic cell blocks, and output controllers. The logic cell blocks L1, L2, and L3 produce outputs with some delay with respect to inputs. The output signals from the flip-flops are delayed by up to one cycle of the clock selected for synchronization of clock signals by the signal selection blocks 6 of event links L1 to L3. Furthermore, deviations in the timing of input signals may lead to the generation of glitches or expected outputs not being obtained. If an expected output not being obtained may create serious problems for a user system, stop attempting to use the ELCL or employ an external circuit as a workaround.

Section 21 Interrupt Functions

The interrupt function switches the program execution to other processing. When the branch processing is finished, the program returns to the interrupted processing. The number of interrupt sources differs, depending on the product.

		R7F100G										R7F104G		
		30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin	64-pin	80- and 100-pin	128-pin	32-pin	44-pin	48-pin
Maskable interrupts	External	6	6	6	7	7	10	12	13	13	13	8	8	8
	Internal	31	32	35	35	39	39	39	39	44	48	39	41	42

21.1 Interrupt Function Types

The following two types of interrupt functions are used.

1. Maskable interrupts
   These interrupts undergo mask control. Maskable interrupts can be divided into four priority groups by setting the priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, 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 default priority, see **Table 21 - 1** and **Table 21 - 2**. A standby release signal is generated and STOP, HALT, and SNOOZE modes are released. External interrupt requests and internal interrupt requests are provided as maskable interrupts.
2. Software interrupt
   This is a vectored interrupt generated by executing the BRK instruction. It is acknowledged even when interrupts are disabled. The software interrupt does not undergo interrupt priority control.

21.2 Interrupt Sources and Configuration

Interrupt sources include maskable interrupts and a software interrupt. In addition, they also have up to seven reset sources (see **Table 21 - 1** and **Table 21 - 2**). The vector codes that store the program start address when branching due to the generation of a reset or various interrupt requests are two bytes each, so interrupts jump to a 64 K address of 00000H to 0FFFFH.

Table 21 - 1 Interrupt Source List (R7F100G) (1/3)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type[Note 2]	128-pin	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin	40-pin	36-pin	32-pin	30-pin
		Name	Trigger														
Maskable	0	INTWDTI	Watchdog timer interval[Note 3] (75% of overflow time + 1/4fIL)	Internal	00004H	<a>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	1	INTLVI	Voltage detection[Note 4]		00006H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	2	INTP0	Pin input edge detection	External	00008H	<b>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	3	INTP1			0000AH		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	4	INTP2			0000CH		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	5	INTP3			0000EH		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	6	INTP4			00010H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	7	INTP5			00012H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	8	INTST2/ INTCSI20/ INTIIC20	UART2 transmission transfer end or buffer empty interrupt/CSI20 transfer end or buffer empty interrupt/IIC20 transfer end	Internal	00014H	<a>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	9	INTSR2/ INTCSI21/ INTIIC21	UART2 reception transfer end/CSI21 transfer end or buffer empty interrupt/IIC21 transfer end		00016H		✓	✓	✓	✓	✓	✓	✓	✓	✓	Note 5	Note 5
	10	INTSRE2	UART2 reception communication error occurrence		00018H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		INTTM11H	End of counting or capturing by channel 1 of timer array unit 1 (at higher 8-bit timer operation)				✓	✓	✓	—	—	—	—	—	—	—	—
	11	INTELCL	Event link interrupt		0001AH		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	12	INTSMSE	Event output from the SNOOZE mode sequencer		0001CH		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	13	INTST0/ INTCSI00/ INTIIC00	UART0 transmission transfer end or buffer empty interrupt/CSI00 transfer end or buffer empty interrupt/IIC00 transfer end		0001EH		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	14	INTTM00	End of counting or capturing by channel 0 of timer array unit 0		00020H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	15	INTSRE0	UART0 reception communication error occurrence		00022H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		INTTM01H	End of counting or capturing by channel 1 of timer array unit 0 (at higher 8-bit timer operation)				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	16	INTST1/ INTCSI10/ INTIIC10	UART1 transmission transfer end or buffer empty interrupt/CSI10 transfer end or buffer empty interrupt/IIC10 transfer end		00024H		✓	✓	✓	✓	Note 6	Note 6	Note 6	Note 6	Note 6	Note 6	Note 6
	17	INTSR1/ INTCSI11/ INTIIC11	UART1 reception transfer end/CSI11 transfer end or buffer empty interrupt/IIC11 transfer end		00026H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	18	INTSRE1	UART1 reception communication error occurrence		00028H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		INTTM03H	End of counting or capturing by channel 3 of timer array unit 0 (at higher 8-bit timer operation)				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	19	INTIICA0	End of IICA0 communication		0002AH		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	20	INTSR0/ INTCSI01/ INTIIC01	UART0 reception transfer end/CSI01 transfer end or buffer empty interrupt/IIC01 transfer end		0002CH		✓	✓	✓	✓	✓	✓	Note 7	Note 7	Note 7	Note 7	Note 7

Table 21 - 1 Interrupt Source List (R7F100G) (2/3)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type Note 2	128-pin	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin	40-pin	36-pin	32-pin	30-pin
		Name	Trigger														
Maskable	21	INTTM01	End of counting or capturing by channel 1 of timer array unit 0 (at 16-bit/lower 8-bit timer operation)	Internal	0002EH	<a>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	22	INTTM02	End of counting or capturing by channel 2 of timer array unit 0		00030H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	23	INTTM03	End of counting or capturing by channel 3 of timer array unit 0 (at 16-bit/lower 8-bit timer operation)		00032H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	24	INTAD	End of A/D conversion		00034H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	25	INTRTC	Fixed-cycle signal of realtime clock/alarm match detection		00036H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	26	INTITL	Interval signal of 32-bit interval timer detection		00038H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	27	INTKR	Key return signal detection	External	0003AH	<c>	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—
	28	INTST3/ INTCSI30/ INTIIC30	UART3 transmission transfer end or buffer empty interrupt/CSI30 transfer end or buffer empty interrupt/IIC30 transfer end	Internal	0003CH	<a>	✓	✓	✓	—	—	—	—	—	—	—	—
	29	INTSR3/ INTCSI31/ INTIIC31	UART3 reception transfer end/CSI31 transfer end or buffer empty interrupt/IIC31 transfer end		0003EH		✓	✓	✓	—	—	—	—	—	—	—	—
	30	INTTM13	End of counting or capturing by channel 3 of timer array unit 1 (at 16-bit/lower 8-bit timer operation)		00040H		✓	✓	✓	—	—	—	—	—	—	—	—
	31	INTTM04	End of counting or capturing by channel 4 of timer array unit 0		00042H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	32	INTTM05	End of counting or capturing by channel 5 of timer array unit 0		00044H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	33	INTTM06	End of counting or capturing by channel 6 of timer array unit 0		00046H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	34	INTTM07	End of counting or capturing by channel 7 of timer array unit 0		00048H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	35	INTP6	Pin input edge detection	External	0004AH	<b>	✓	✓	✓	✓	✓	✓	—	—	—	—	—
	36	INTP7			0004CH		✓	✓	✓	✓	—	—	—	—	—	—	—
	37	INTP8			0004EH		✓	✓	✓	✓	✓	✓	—	—	—	—	—
	38	INTP9			00050H		✓	✓	✓	✓	✓	✓	—	—	—	—	—
	39	INTFL	Reserved		00052H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	40	INTP10	Pin input edge detection	External	00054H		✓	✓	✓	✓	✓	—	—	—	—	—	—
		INTCMP0	Comparator detection 0	Internal			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	41	INTP11	Pin input edge detection	External	00056H		✓	✓	✓	✓	✓	—	—	—	—	—	—
		INTCMP1	Comparator detection 1	Internal			✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	42	INTURE0	UARTA0 reception communication error occurrence	Internal	00058H	<a>	✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
		INTTM10	End of counting or capturing by channel 0 of timer array unit 1				✓	✓	✓	—	—	—	—	—	—	—	—

Table 21 - 1 Interrupt Source List (R7F100G) (3/3)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type Note 2	128-pin	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin	40-pin	36-pin	32-pin	30-pin
		Name	Trigger														
Maskable	43	INTURE1	UARTA1 reception communication error occurrence	Internal	0005AH	<a>	✓	✓	✓	✓	✓	✓	✓	—	—	—	—
		INTTM11	End of counting or capturing by channel 1 of timer array unit 1 (at 16-bit/lower 8-bit timer operation)				✓	✓	✓	—	—	—	—	—	—	—	—
	44	INTTM12	End of counting or capturing by channel 2 of timer array unit 1		0005CH		✓	✓	✓	—	—	—	—	—	—	—	—
	45	INTSRE3	UART3 reception communication error occurrence		0005EH		✓	✓	✓	—	—	—	—	—	—	—	—
		INTTM13H	End of counting or capturing by channel 3 of timer array unit 1 (at higher 8-bit timer operation)				✓	✓	✓	—	—	—	—	—	—	—	—
	46	INTCTSUWR	Request to write to a configuration register of an individual CTSU channel		00060H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	47	INTIICA1	End of IICA1 communication		00062H		✓	✓	✓	✓	✓	✓	✓	—	—	—	—
	48	INTCTSURD	Request to transfer data measured by the CTSU		00064H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	49	INTCTSUFN	CTSU measurement end		00066H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	50	INTREMC	REMC interrupt request		00068H		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
	51	INTUT0	UARTA0 transmission transfer end or buffer empty interrupt		0006AH		✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
	52	INTUR0	UARTA0 reception transfer end		0006CH		✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
	53	INTUT1	UARTA1 transmission transfer end or buffer empty interrupt		0006EH		✓	✓	✓	✓	✓	✓	✓	—	—	—	—
	54	INTUR1	UARTA1 reception transfer end		00070H		✓	✓	✓	✓	✓	✓	✓	—	—	—	—
	55	INTTM14	End of counting or capturing by channel 4 of timer array unit 1		00072H		✓	—	—	—	—	—	—	—	—	—	—
	56	INTTM15	End of counting or capturing by channel 5 of timer array unit 1		00074H		✓	—	—	—	—	—	—	—	—	—	—
	57	INTTM16	End of counting or capturing by channel 6 of timer array unit 1		00076H		✓	—	—	—	—	—	—	—	—	—	—
	58	INTTM17	End of counting or capturing by channel 7 of timer array unit 1		00078H		✓	—	—	—	—	—	—	—	—	—	—
Software	—	BRK	Execution of BRK instruction	—	0007EH	<d>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Reset	—	RESET	$\overline{\text{RESET}}$ pin input	—	00000H	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		POR	Power-on-reset				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		LVD	Voltage detection				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		WDT	Overflow of watchdog timer				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		TRAP	Execution of illegal instruction				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		IAW	Illegal-memory access				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		RPE	RAM parity error				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

(Notes are listed on the next page.)

Note 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 58 indicates the lowest priority.

Note 2. Basic configuration types <a> to <d> correspond to <a> to <d> in **Figure 21 - 1 Basic Configuration of Interrupt Function**.

Note 3. This is applicable when the value of bit 7 (WDTINT) in the user option byte (000C0H) is 1.

Note 4. This is applicable when the value of bit 6 (LVD1SEL) in the voltage detection level register (LVIS) is 0 or the value of bit 6 (LVD0SEL) in the user option byte (000C1H) is 0.

Note 5. INTSR2 is only present in products of this group.

Note 6. INTST1 is only present in products of this group.

Note 7. INTSR0 is only present in products of this group.

Table 21 - 2 Interrupt Source List (R7F104G) (1/4)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type Note 2	48-pin	44-pin	32-pin
		Name	Trigger						
Maskable	0	INTWDTI	Watchdog timer interval Note 3 (75% of overflow time + 1/4fIL)	Internal	00004H	<a>	✓	✓	✓
	1	INTLVI	Voltage detection Note 4		00006H		✓	✓	✓
	2	INTP0	Pin input edge detection	External	00008H	<b>	✓	✓	✓
	3	INTP1			0000AH		✓	✓	✓
	4	INTP2			0000CH		—	—	—
	5	INTP3			0000EH		✓	✓	✓
	6	INTP4			00010H		✓	✓	✓
	7	INTP5			00012H		✓	✓	✓
	8	INTST2/ INTCSI20/ INTIIC20	UART2 transmission transfer end or buffer empty interrupt/CSI20 transfer end or buffer empty interrupt/IIC20 transfer end	Internal	00014H	<a>	Note 5	Note 5	Note 5
	9	INTSR2/ INTCSI21/ INTIIC21	UART2 reception transfer end/CSI21 transfer end or buffer empty interrupt/ IIC21 transfer end		00016H		✓	✓	✓
	10	INTSRE2	UART2 reception communication error occurrence		00018H		✓	✓	✓
		INTTM11H	End of counting or capturing by channel 1 of timer array unit 1 (at higher 8-bit timer operation)				✓	✓	—
	11	INTELCL	Event link interrupt		0001AH		✓	✓	✓
	12	INTSMSE	Event output from the SNOOZE mode sequencer		0001CH		✓	✓	✓
	13	INTST0/ INTCSI00/ INTIIC00	UART0 transmission transfer end or buffer empty interrupt/CSI00 transfer end or buffer empty interrupt/IIC00 transfer end		0001EH		✓	✓	✓
	14	INTTM00	End of counting or capturing by channel 0 of timer array unit 0		00020H		✓	✓	✓
	15	INTSRE0	UART0 reception communication error occurrence		00022H		✓	✓	✓
		INTTM01H	End of counting or capturing by channel 1 of timer array unit 0 (at higher 8-bit timer operation)				✓	✓	✓
	16	INTST1/ INTCSI10/ INTIIC10	UART1 transmission transfer end or buffer empty interrupt/CSI10 transfer end or buffer empty interrupt/IIC10 transfer end		00024H		Note 6	Note 6	Note 6
	17	INTSR1/ INTCSI11/ INTIIC11	UART1 reception transfer end/CSI11 transfer end or buffer empty interrupt/ IIC11 transfer end		00026H		Note 7	Note 7	Note 7
	18	INTSRE1	UART1 reception communication error occurrence		00028H		✓	✓	✓
		INTTM03H	End of counting or capturing by channel 3 of timer array unit 0 (at higher 8-bit timer operation)				✓	✓	✓
	19	INTIICA0	End of IICA0 communication		0002AH		✓	—	—

Table 21 - 2 Interrupt Source List (R7F104G) (2/4)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type Note 2	48-pin	44-pin	32-pin
		Name	Trigger						
Maskable	20	INTSR0/ INTCSI01/ INTIIC01	UART0 reception transfer end/CSI01 transfer end or buffer empty interrupt/ IIC01 transfer end	Internal	0002CH	<a>	Note 8	Note 8	Note 8
	21	INTTM01	End of counting or capturing by channel 1 of timer array unit 0 (at 16-bit/lower 8-bit timer operation)		0002EH		✓	✓	✓
	22	INTTM02	End of counting or capturing by channel 2 of timer array unit 0		00030H		✓	✓	✓
	23	INTTM03	End of counting or capturing by channel 3 of timer array unit 0 (at 16-bit/lower 8-bit timer operation)		00032H		✓	✓	✓
	24	INTAD	End of A/D conversion		00034H		✓	✓	✓
	25	INTRTC	Fixed-cycle signal of realtime clock/alarm match detection		00036H		✓	✓	✓
	26	INTITL	Interval signal of 32-bit interval timer detection		00038H		✓	✓	✓
	27	INTKR	Key return signal detection	External	0003AH	<c>	✓	✓	✓
	28	INTST3/ INTCSI30/ INTIIC30	UART3 transmission transfer end or buffer empty interrupt/CSI30 transfer end or buffer empty interrupt/IIC30 transfer end	Internal	0003CH	<a>	—	—	—
	29	INTSR3/ INTCSI31/ INTIIC31	UART3 reception transfer end/CSI31 transfer end or buffer empty interrupt/ IIC31 transfer end		0003EH		—	—	—
	30	INTTM13	End of counting or capturing by channel 3 of timer array unit 1 (at 16-bit/lower 8-bit timer operation)		00040H		✓	✓	—
	31	INTTM04	End of counting or capturing by channel 4 of timer array unit 0		00042H		✓	✓	✓
	32	INTTM05	End of counting or capturing by channel 5 of timer array unit 0		00044H		✓	✓	✓
	33	INTTM06	End of counting or capturing by channel 6 of timer array unit 0		00046H		✓	✓	✓
	34	INTTM07	End of counting or capturing by channel 7 of timer array unit 0		00048H		✓	✓	✓
	35	INTP6	Pin input edge detection	External	0004AH	<b>	—	—	—
	36	INTP7			0004CH		—	—	—
	37	INTP8			0004EH		—	—	—
	38	INTP9			00050H		—	—	—
	39	INTFL	Reserved		00052H		✓	✓	✓

Table 21 - 2 Interrupt Source List (R7F104G) (3/4)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type Note 2	48-pin	44-pin	32-pin
		Name	Trigger						
Maskable	40	INTP10	Pin input edge detection	External	00054H	<b>	✓	✓	✓
		INTCMP0	Comparator detection 0	Internal			✓	✓	✓
	41	INTP11	Pin input edge detection	External	00056H		✓	✓	✓
		INTCMP1	Comparator detection 1	Internal			—	—	—
	42	INTURE0	UARTA0 reception communication error occurrence	Internal	00058H		✓	✓	✓
		INTTM10	End of counting or capturing by channel 0 of timer array unit 1				✓	✓	—
	43	INTURE1	UARTA1 reception communication error occurrence		0005AH		—	—	—
		INTTM11	End of counting or capturing by channel 1 of timer array unit 1 (at 16-bit/lower 8-bit timer operation)				✓	✓	—
	44	INTTM12	End of counting or capturing by channel 2 of timer array unit 1		0005CH		✓	✓	—
	45	INTSRE3	UART3 reception communication error occurrence		0005EH		—	—	—
		INTTM13H	End of counting or capturing by channel 3 of timer array unit 1 (at higher 8-bit timer operation)				✓	✓	—
	46	INTCTSUWR	Request to write to a configuration register of an individual CTSU channel		00060H		✓	✓	✓
	47	INTIICA1	End of IICA1 communication		00062H		—	—	—
	48	INTCTSURD	Request to transfer data measured by the CTSU		00064H		✓	✓	✓
	49	INTCTSUFN	CTSU measurement end		00066H	<a>	✓	✓	✓
	50	INTREMC	REMC interrupt request		00068H		✓	✓	✓
	51	INTUT0	UARTA0 transmission transfer end or buffer empty interrupt		0006AH		✓	✓	✓
	52	INTUR0	UARTA0 reception transfer end		0006CH		✓	✓	✓
	53	INTUT1	UARTA1 transmission transfer end or buffer empty interrupt		0006EH		—	—	—
	54	INTUR1	UARTA1 reception transfer end		00070H		—	—	—
	55	INTTM14	End of counting or capturing by channel 4 of timer array unit 1		00072H		—	—	—
	56	INTTM15	End of counting or capturing by channel 5 of timer array unit 1		00074H		—	—	—
	57	INTTM16	End of counting or capturing by channel 6 of timer array unit 1		00076H		—	—	—
	58	INTTM17	End of counting or capturing by channel 7 of timer array unit 1		00078H		—	—	—

Table 21 - 2 Interrupt Source List (R7F104G) (4/4)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type Note 2	48-pin	44-pin	32-pin
		Name	Trigger						
Software	—	BRK	Execution of BRK instruction	—	0007EH	<d>	✓	✓	✓
Reset	—	RESET	$\overline{\text{RESET}}$ pin input	—	00000H	—	✓	✓	✓
		POR	Power-on-reset				✓	✓	✓
		LVD	Voltage detection				✓	✓	✓
		WDT	Overflow of watchdog timer				✓	✓	✓
		TRAP	Execution of illegal instruction				✓	✓	✓
		IAW	Illegal-memory access				✓	✓	✓
		RPE	RAM parity error				✓	✓	✓

Note 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 58 indicates the lowest priority.

Note 2. Basic configuration types <a> to <d> correspond to <a> to <d> in **Figure 21 - 1 Basic Configuration of Interrupt Function**.

Note 3. This is applicable when the value of bit 7 (WDTINT) in the user option byte (000C0H) is 1.

Note 4. This is applicable when the value of bit 6 (LVD1SEL) in the voltage detection level register (LVIS) is 0 or the value of bit 6 (LVD0SEL) in the user option byte (000C1H) is 0.

Note 5. INTSR2 is only present in products of this group.

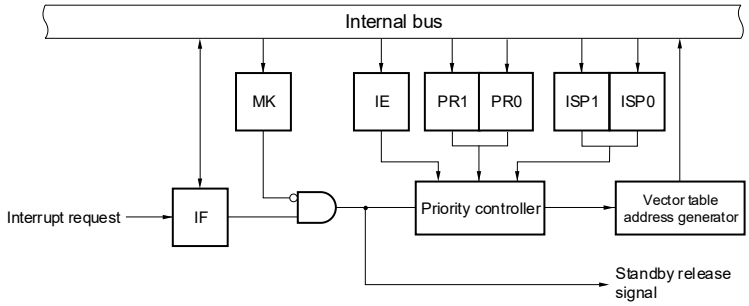
Note 6. INTST1 is only present in products of this group.

Note 7. INTSR1 and INTIIC11 are only present in products of this group.

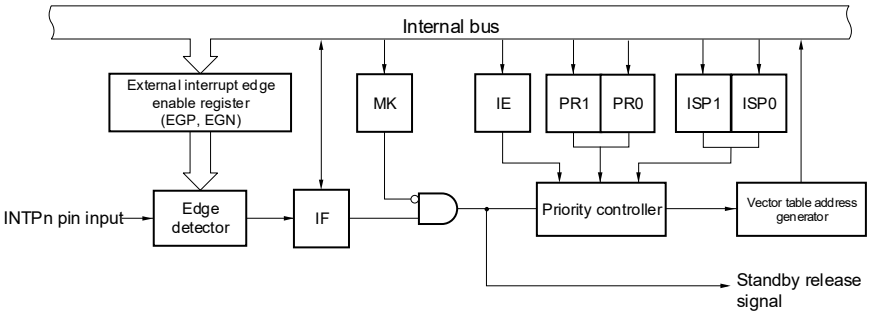
Note 8. INTSR0 is only present in products of this group.

Figure 21 - 1 Basic Configuration of Interrupt Function (1/2)

<a> Internal maskable interrupt



<b> External maskable interrupt (INTPn)

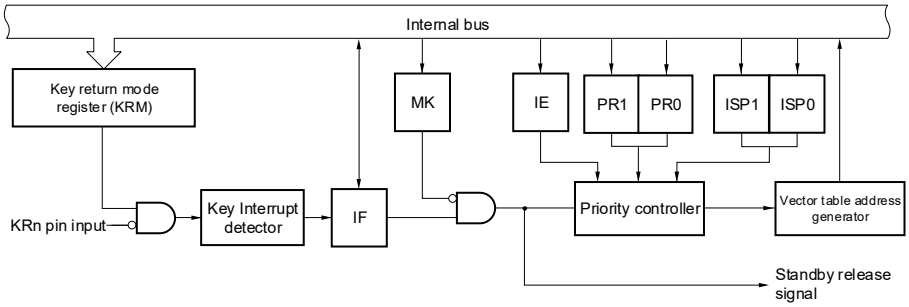


Remark 1. IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

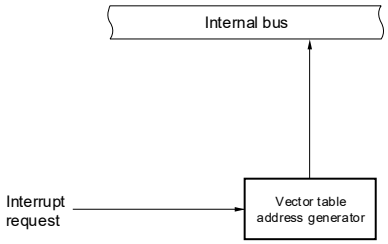
Remark 2. 30- to 44-pin R7F100G products: n = 0 to 5
48-pin R7F100G products: n = 0 to 6, 8, 9
52-pin R7F100G products: n = 0 to 6, 8 to 11
64- to 128-pin R7F100G products: n = 0 to 11
32- to 48-pin R7F104G products: n = 0, 1, 3 to 5, 10, 11

Figure 21 - 1 Basic Configuration of Interrupt Function (2/2)

<c> External maskable interrupt (INTKR)



<d> Software interrupt



Remark 1. 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 2. 40- and 44-pin R7F100G products: n = 0 to 3
48-pin R7F100G products: n = 0 to 5
52- to 128-pin R7F100G products: n = 0 to 7
32- to 48-pin R7F104G products: n = 0 to 3, 6, 7

21.3 Registers for Controlling the Interrupt Functions

The following registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H)
- Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H)
- Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H)
- External interrupt rising edge enable registers (EGP0, EGP1), External interrupt falling edge enable registers (EGN0, EGN1)
- Program status word (PSW)
- Port mode registers (PMxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)

Table 21 - 3 and **Table 21 - 4** show lists of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 21 - 3 Flags Corresponding to Interrupt Request Sources (R7F100G) (1/5)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		128-pin	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin	40-pin	36-pin	32-pin	30-pin
		Register		Register		Register											
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTLVI**Note**	LVIIF**Note**		LVIMK**Note**		LVIPR0, LVIPR1**Note**		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
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		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTP5	PIF5		PMK5		PPR05, PPR15		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Note The DLVD0F and DLVD1F flags of the LVIM register can be used to confirm which voltage detector has issued the given interrupt, LVD0 or LVD1. For details, see **26.3.1 Voltage detection register (LVIM)**.

Table 21 - 3 Flags Corresponding to Interrupt Request Sources (R7F100G) (2/5)

Interrupt Source	Interrupt Request Flag	Register	Interrupt Mask Flag	Register	Priority Specification Flag	Register	128-pin	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin	40-pin	36-pin	32-pin	30-pin
INTST2Note 1	STIF2Note 1	IF0H	STMK2Note 1	MK0H	STPR02, STPR12Note 1	PR00H, PR10H	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTCSI20 Note 1	CSIIF20Note 1		CSIMK20Note 1		CSIPR020, CSIPR120Note 1		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTIIC20Note 1	IICIF20Note 1		IICMK20Note 1		IICPR020, IICPR120 Note 1		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTSR2Note 2	SRIF2Note 2		SRMK2Note 2		SRPR02, SRPR12Note 2		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTCSI21 Note 2	CSIIF21Note 2		CSIMK21Note 2		CSIPR021, CSIPR121Note 2		✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
INTIIC21Note 2	IICIF21Note 2		IICMK21Note 2		IICPR021, IICPR121 Note 2		✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
INTSRE2 Note 3	SREIF2Note 3		SREMK2Note 3		SREPR02, SREPR12 Note 3		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTM11H Note 3	TMIF11HNote 3		TMMK11HNote 3		TMPR011H,TMPR111H Note 3		✓	✓	✓	—	—	—	—	—	—	—	—
INTELCL	ELCLIF		ELCLMK		ELCLPR0, ELCLPR1		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTSMSE	SMSEIF		SMSEMK		SMSEPR0, SMSEPR1		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTST0Note 4	STIF0Note 4		STMK0Note 4		STPR00, STPR10Note 4		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTCSI00 Note 4	CSIIF00Note 4		CSIMK00Note 4		CSIPR000, CSIPR100Note 4		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTIIC00Note 4	IICIF00Note 4		IICMK00Note 4		IICPR000, IICPR100 Note 4		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTSRE0 Note 5	SREIF0Note 5		SREMK0Note 5		SREPR00, SREPR10 Note 5		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTM01H Note 5	TMIF01HNote 5		TMMK01HNote 5		TMPR001H, TMPR101HNote 5		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Note 1. If any of the interrupt sources INTST2, INTCSI20, and INTIIC20 is generated, bit 0 of the IF0H register is set to 1. Bit 0 of the MK0H, PR00H, and PR10H registers supports these three interrupt sources.

Note 2. If any of the interrupt sources INTSR2, INTCSI21, and INTIIC21 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.

Note 3. Do not use a UART2 reception error interrupt and an interrupt of channel 1 of TAU1 (at higher 8-bit timer operation) at the same time because they share flags for the interrupt request sources. If the UART2 reception error interrupt is not used (EOC01 = 0), UART2 and channel 1 of TAU1 (at higher 8-bit timer operation) can be used at the same time. If either of the interrupt sources INTSRE2 or INTTM11H is generated, bit 2 of the IF0H register is set to 1. Bit 2 of the MK0H, PR00H, and PR10H registers supports these two interrupt sources.

Note 4. If any of the interrupt sources INTST0, INTCSI00, and INTIIC00 is generated, bit 5 of the IF0H register is set to 1. Bit 5 of the MK0H, PR00H, and PR10H registers supports these three interrupt sources.

Note 5. Do not use a UART0 reception error interrupt and an interrupt of channel 1 of TAU0 (at higher 8-bit timer operation) at the same time because they share flags for the interrupt request sources. If the UART0 reception error interrupt is not used (EOC01 = 0), UART0 and channel 1 of TAU0 (at higher 8-bit timer operation) can be used at the same time. If either of the interrupt sources INTSRE0 or INTTM01H is generated, bit 7 of the IF0H register is set to 1. Bit 7 of the MK0H, PR00H, and PR10H registers supports these two interrupt sources.

Table 21 - 3 Flags Corresponding to Interrupt Request Sources (R7F100G) (3/5)

Interrupt Source	Interrupt Request Flag	Register	Interrupt Mask Flag	Register	Priority Specification Flag	Register	128-pin	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin	40-pin	36-pin	32-pin	30-pin
INTST1 Note 1	STIF1 Note 1	IF1L	STMK1 Note 1	MK1L	STPR01, STPR11 Note 1	PR01L, PR11L	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTCSI10 Note 1	CSIIF10 Note 1		CSIMK10 Note 1		CSIPR010,CSIPR110 Note 1		✓	✓	✓	✓	—	—	—	—	—	—	—
INTIIC10 Note 1	IICIF10 Note 1		IICMK10 Note 1		IICPR010, IICPR110 Note 1		✓	✓	✓	✓	—	—	—	—	—	—	—
INTSR1 Note 2	SRIF1 Note 2		SRMK1 Note 2		SRPR01, SRPR11 Note 2		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTCSI11 Note 2	CSIIF11 Note 2		CSIMK11 Note 2		CSIPR011, CSIPR111 Note 2		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTIIC11 Note 2	IICIF11 Note 2		IICMK11 Note 2		IICPR011, IICPR111 Note 2		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTSRE1 Note 3	SREIF1 Note 3		SREMK1 Note 3		SREPR01, SREPR11 Note 3		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTM03H Note 3	TMIF03H Note 3		TMMK03H Note 3		TMPR003H,TMPR103H Note 3		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTIICA0	IICAIF0		IICAMK0		IICAPR00, IICAPR10		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTSR0 Note 4	SRIF0 Note 4		SRMK0 Note 4		SRPR00, SRPR10 Note 4		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTCSI01 Note 4	CSIIF01 Note 4		CSIMK01 Note 4		CSIPR001, CSIPR101 Note 4		✓	✓	✓	✓	✓	✓	—	—	—	—	—
INTIIC01 Note 4	IICIF01 Note 4		IICMK01 Note 4		IICPR001, IICPR101 Note 4		✓	✓	✓	✓	✓	✓	—	—	—	—	—
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTM02	TMIF02		TMMK02		TMPR002, TMPR102		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTAD	ADIF	IF1H	ADMK	MK1H	ADPR0, ADPR1	PR01H, PR11H	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTRTC	RTCIF		RTCMK		RTCPR0, RTCPR1		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTITL	ITLIF		ITLMK		ITLPR0, ITLPR1		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTKR	KRIF		KRMK		KRPR0, KRPR1		✓	✓	✓	✓	✓	✓	✓	✓	—	—	—
INTST3 Note 5	STIF3 Note 5		STMK3 Note 5		STPR03, STPR13 Note 5		✓	✓	✓	—	—	—	—	—	—	—	—
INTCSI30 Note 5	CSIIF30 Note 5		CSIMK30 Note 5		CSIPR030, CSIPR130 Note 5		✓	✓	✓	—	—	—	—	—	—	—	—
INTIIC30 Note 5	IICIF30 Note 5		IICMK30 Note 5		IICPR030, IICPR130 Note 5		✓	✓	✓	—	—	—	—	—	—	—	—
INTSR3 Note 6	SRIF3 Note 6		SRMK3 Note 6		SRPR03, SRPR13 Note 6		✓	✓	✓	—	—	—	—	—	—	—	—
INTCSI31 Note 6	CSIIF31 Note 6		CSIMK31 Note 6		CSIPR031, CSIPR131 Note 6		✓	✓	✓	—	—	—	—	—	—	—	—
INTIIC31 Note 6	IICIF31 Note 6		IICMK31 Note 6		IICPR031, IICPR131 Note 6		✓	✓	✓	—	—	—	—	—	—	—	—
INTTM13	TMIF13		TMMK13		TMPR013, TMPR113		✓	✓	✓	—	—	—	—	—	—	—	—
INTTM04	TMIF04		TMMK04		TMPR004, TMPR104		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Note 1. If any of the interrupt sources INTST1, INTCSI10, and INTIIC10 is generated, bit 0 of the IF1L register is set to 1. Bit 0 of the MK1L, PR01L, and PR11L registers supports these three interrupt sources.

Note 2. If any of the interrupt sources INTSR1, INTCSI11, and INTIIC11 is generated, bit 1 of the IF1L register is set to 1. Bit 1 of the MK1L, PR01L, and PR11L registers supports these three interrupt sources.

(Notes are listed on the next page.)

Note 3. Do not use a UART1 reception error interrupt and an interrupt of channel 3 of TAU0 (at higher 8-bit timer operation) at the same time because they share flags for the interrupt request sources. If the UART1 reception error interrupt is not used (EOC03 = 0), UART1 and channel 3 of TAU0 (at higher 8-bit timer operation) can be used at the same time. If either of the interrupt sources INTSRE1 or INTTM03H is generated, bit 2 of the IF1L register is set to 1. Bit 2 of the MK1L, PR01L, and PR11L registers supports these two interrupt sources.

Note 4. If any of the interrupt sources INTSR0, INTCSI01, and INTIIC01 is generated, bit 4 of the IF1L register is set to 1. Bit 4 of the MK1L, PR01L, and PR11L registers supports these three interrupt sources.

Note 5. If any of the interrupt sources INTST3, INTCSI30, and INTIIC30 is generated, bit 4 of the IF1H register is set to 1. Bit 4 of the MK1H, PR01H, and PR11H registers supports these three interrupt sources.

Note 6. If any of the interrupt sources INTSR3, INTCSI31, and INTIIC31 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.

Table 21 - 3 Flags Corresponding to Interrupt Request Sources (R7F100G) (4/5)

Interrupt Source	Interrupt Request Flag	Register	Interrupt Mask Flag	Register	Priority Specification Flag	Register	128-pin	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin	40-pin	36-pin	32-pin	30-pin
INTTM05	TMIF05	IF2L	TMMK05	MK2L	TMPR005, TMPR105	PR02L, PR12L	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTM06	TMIF06		TMMK06		TMPR006, TMPR106		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTTM07	TMIF07		TMMK07		TMPR007, TMPR107		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTP6	PIF6		PMK6		PPR06, PPR16		✓	✓	✓	✓	✓	✓	—	—	—	—	—
INTP7	PIF7		PMK7		PPR07, PPR17		✓	✓	✓	✓	—	—	—	—	—	—	—
INTP8	PIF8		PMK8		PPR08, PPR18		✓	✓	✓	✓	✓	✓	—	—	—	—	—
INTP9	PIF9		PMK9		PPR09, PPR19		✓	✓	✓	✓	✓	✓	—	—	—	—	—
INTFL	FLIF		FLMK		FLPR0, FLPR1		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTP10Note 1	PIF10Note 1	IF2H	PMK10Note 1	MK2H	PPR010, PPR110Note 1	PR02H, PR12H	✓	✓	✓	✓	✓	—	—	—	—	—	—
INTCMP0 Note 1	CMPIF0Note 1		CMPMK0Note 1		CMPPR00, CMPPR10Note 1		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTP11Note 2	PIF11Note 2		PMK11Note 2		PPR011, PPR111Note 2		✓	✓	✓	✓	✓	—	—	—	—	—	—
INTCMP1 Note 2	CMPIF1Note 2		CMPMK1Note 2		CMPPR01, CMPPR11Note 2		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTURE0 Note 3	UREIF0Note 3		UREMK0Note 3		UREPR00, UREPR10Note 3		✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
INTTM10 Note 3	TMIF10Note 3		TMMK10Note 3		TMPR010, TMPR110Note 3		✓	✓	✓	—	—	—	—	—	—	—	—
INTURE1 Note 4	UREIF1Note 4		UREMK1Note 4		UREPR01, UREPR11 Note 4		✓	✓	✓	✓	✓	✓	✓	—	—	—	—
INTTM11 Note 4	TMIF11Note 4		TMMK11Note 4		TMPR011, TMPR111 Note 4		✓	✓	✓	—	—	—	—	—	—	—	—
INTTM12	TMIF12		TMMK12		TMPR012, TMPR112		✓	✓	✓	—	—	—	—	—	—	—	—
INTSRE3 Note 5	SREIF3Note 5		SREMK3Note 5		SREPR03, SREPR13Note 5		✓	✓	✓	—	—	—	—	—	—	—	—
INTTM13H Note 5	TMIF13H Note 5		TMMK13HNote 5		TMPR013H, TMPR113HNote 5		✓	✓	✓	—	—	—	—	—	—	—	—
INTCTSUWR	CTSUWRIF		CTSUWRMK		CTSUWRPR0, CTSUWRPR1		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTIICA1	IICAIF1		IICAMK1		IICAPR01, IICAPR11		✓	✓	✓	✓	✓	✓	✓	—	—	—	—

Note 1. Do not use INTP10 and comparator 0 at the same time because they share the same flag as interrupt request sources. If an interrupt from either the INTP10 or INTCMP0 source is generated, bit 0 of the IF2H register is set to 1. Bit 0 of the MK2H, PR02H, and PR12H registers supports these two interrupt sources.

Note 2. Do not use INTP11 and comparator 1 at the same time because they share the same flag as interrupt request sources. If an interrupt from either the INTP11 or INTCMP1 source is generated, bit 1 of the IF2H register is set to 1. Bit 1 of the MK2H, PR02H, and PR12H registers supports these two interrupt sources.

Note 3. Do not use a UARTA0 reception error interrupt and an interrupt of channel 0 of TAU1 at the same time because they share flags for the interrupt request sources. If the UARTA0 reception error interrupt is not used (ISRMA0 = 1), UARTA0 and channel 0 of TAU1 can be used at the same time. If either of the interrupt sources INTURE0 or INTTM10 is generated, bit 2 of the IF2H register is set to 1. Bit 2 of the MK2H, PR02H, and PR12H registers supports these two interrupt sources.

Note 4. Do not use a UARTA1 reception error interrupt and an interrupt of channel 1 of TAU1 (at 16-bit or higher 8-bit timer operation) at the same time because they share flags for the interrupt request sources. If the UARTA1 reception error interrupt is not used (ISRMA1 = 1), UARTA1 and channel 1 of TAU1 (at 16-bit or higher 8-bit timer operation) can be used at the same time. If either of the interrupt sources INTURE1 or INTTM11 is generated, bit 3 of the IF2H register is set to 1. Bit 3 of the MK2H, PR02H, and PR12H registers supports these two interrupt sources.

(Note is listed on the next page.)

Note 5. Do not use a UART3 reception error interrupt and an interrupt of channel 3 of TAU1 (at higher 8-bit timer operation) at the same time because they share flags for the interrupt request sources. If the UART3 reception error interrupt is not used (EOC03 = 0), UART3 and channel 3 of TAU1 (at higher 8-bit timer operation) can be used at the same time. If either of the interrupt sources INTSRE3 or INTTM13H is generated, bit 4 of the IF2H register is set to 1. Bit 4 of the MK2H, PR02H, and PR12H registers supports these two interrupt sources.

Table 21 - 3 Flags Corresponding to Interrupt Request Sources (R7F100G) (5/5)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		128-pin	100-pin	80-pin	64-pin	52-pin	48-pin	44-pin	40-pin	36-pin	32-pin	30-pin
		Register		Register		Register											
INTCTSURD	CTSURDIF	IF3L	CTSURDMK	MK3L	CTSURDPR0, CTSURDPR1	PR03L, PR13L	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTCTSUFN	CTSUFNIF		CTSUFNMK		CTSUFNPR0, CTSUFNPR1		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
INTREMC	REMCIF		REMCMK		REMCPR0, REMCPR1		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	—
INTUT0	UTIF0		UTMK0		UTPR00, UTPR10		✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
INTUR0	URIF0		URMK0		URPR00, URPR10		✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—
INTUT1	UTIF1		UTMK1		UTPR01, UTPR11		✓	✓	✓	✓	✓	✓	✓	—	—	—	—
INTUR1	URIF1		URMK1		URPR01, URPR11		✓	✓	✓	✓	✓	✓	✓	—	—	—	—
INTTM14	TMIF14		TMMK14		TMPR014, TMPR114		✓	—	—	—	—	—	—	—	—	—	—
INTTM15	TMIF15	IF3H	TMMK15	MK3H	TMPR015, TMPR115	PR03H, PR13H	✓	—	—	—	—	—	—	—	—	—	—
INTTM16	TMIF16		TMMK16		TMPR016, TMPR116		✓	—	—	—	—	—	—	—	—	—	—
INTTM17	TMIF17		TMMK17		TMPR017, TMPR117		✓	—	—	—	—	—	—	—	—	—	—

Table 21 - 4 Flags Corresponding to Interrupt Request Sources (R7F104G) (1/5)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		48-pin	44-pin	32-pin
		Register		Register		Register			
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L	✓	✓	✓
INTLVI**Note**	LVIIF**Note**		LVIMK**Note**		LVIPR0, LVIPR1**Note**		✓	✓	✓
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		✓	✓	✓
INTP5	PIF5		PMK5		PPR05, PPR15		✓	✓	✓

Note The DLVD0F and DLVD1F flags of the LVIM register can be used to confirm which voltage detector has issued the given interrupt, LVD0 or LVD1. For details, see **26.3.1 Voltage detection register (LVIM)**.

Table 21 - 4 Flags Corresponding to Interrupt Request Sources (R7F104G) (2/5)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		48-pin	44-pin	32-pin
		Register		Register		Register			
INTST2	STIF2	IF0H	STMK2	MK0H	STPR02, STPR12	PR00H, PR10H	✓	✓	✓
INTCSI20	CSIIF20		CSIMK20		CSIPR020, CSIPR120		—	—	—
INTIIC20	IICIF20		IICMK20		IICPR020, IICPR120		—	—	—
INTSR2Note 1	SRIF2Note 1		SRMK2Note 1		SRPR02, SRPR12Note 1		✓	✓	✓
INTCSI21Note 1	CSIIF21Note 1		CSIMK21Note 1		CSIPR021, CSIPR121Note 1		✓	✓	✓
INTIIC21Note 1	IICIF21Note 1		IICMK21Note 1		IICPR021, IICPR121Note 1		✓	✓	✓
INTSRE2Note 2	SREIF2Note 2		SREMK2Note 2		SREPR02, SREPR12Note 2		✓	✓	✓
INTTM11HNote 2	TMIF11HNote 2		TMMK11HNote 2		TMPR011H, TMPR111HNote 2		✓	✓	—
INTELCL	ELCLIF		ELCLMK		ELCLPR0, ELCLPR1		✓	✓	✓
INTSMSE	SMSEIF		SMSEMK		SMSEPR0, SMSEPR1		✓	✓	✓
INTST0Note 3	STIF0Note 3		STMK0Note 3		STPR00, STPR10Note 3		✓	✓	✓
INTCSI00Note 3	CSIIF00Note 3		CSIMK00Note 3		CSIPR000, CSIPR100Note 3		✓	✓	✓
INTIIC00Note 3	IICIF00Note 3		IICMK00Note 3		IICPR000, IICPR100Note 3		✓	✓	✓
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100		✓	✓	✓
INTSRE0Note 4	SREIF0Note 4		SREMK0Note 4		SREPR00, SREPR10Note 4		✓	✓	✓
INTTM01HNote 4	TMIF01HNote 4		TMMK01HNote 4		TMPR001H, TMPR101HNote 4		✓	✓	✓

Note 1. If any of the interrupt sources INTSR2, INTCSI21, and INTIIC21 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.

Note 2. Do not use a UART2 reception error interrupt and an interrupt of channel 1 of TAU1 (at higher 8-bit timer operation) at the same time because they share flags for the interrupt request sources. If the UART2 reception error interrupt is not used (EOC01 = 0), UART2 and channel 1 of TAU1 (at higher 8-bit timer operation) can be used at the same time. If either of the interrupt sources INTSRE2 or INTTM11H is generated, bit 2 of the IF0H register is set to 1. Bit 2 of the MK0H, PR00H, and PR10H registers supports these two interrupt sources.

Note 3. If any of the interrupt sources INTST0, INTCSI00, and INTIIC00 is generated, bit 5 of the IF0H register is set to 1. Bit 5 of the MK0H, PR00H, and PR10H registers supports these three interrupt sources.

Note 4. Do not use a UART0 reception error interrupt and an interrupt of channel 1 of TAU0 (at higher 8-bit timer operation) at the same time because they share flags for the interrupt request sources. If the UART0 reception error interrupt is not used (EOC01 = 0), UART0 and channel 1 of TAU0 (at higher 8-bit timer operation) can be used at the same time. If either of the interrupt sources INTSRE0 or INTTM01H is generated, bit 7 of the IF0H register is set to 1. Bit 7 of the MK0H, PR00H, and PR10H registers supports these two interrupt sources.

Table 21 - 4 Flags Corresponding to Interrupt Request Sources (R7F104G) (3/5)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		48-pin	44-pin	32-pin
		Register		Register		Register			
INTST1	STIF1	IF1L	STMK1	MK1L	STPR01, STPR11	PR01L, PR11L	✓	✓	✓
INTCSI10	CSIIF10		CSIMK10		CSIPR010, CSIPR110		—	—	—
INTIIC10	IICIF10		IICMK10		IICPR010, IICPR110		—	—	—
INTSR1Note 1	SRIF1Note 1		SRMK1Note 1		SRPR01, SRPR11Note 1		✓	✓	✓
INTCSI11	CSIIF11		CSIMK11		CSIPR011, CSIPR111		—	—	—
INTIIC11Note 1	IICIF11Note 1		IICMK11Note 1		IICPR011, IICPR111Note 1		✓	✓	✓
INTSRE1Note 2	SREIF1Note 2		SREMK1Note 2		SREPR01, SREPR11Note 2		✓	✓	✓
INTTM03HNote 2	TMIF03HNote 2		TMMK03HNote 2		TMPR003H, TMPR103HNote 2		✓	✓	✓
INTIICA0	IICAIF0		IICAMK0		IICAPR00, IICAPR10		✓	—	—
INTSR0	SRIF0		SRMK0		SRPR00, SRPR10		✓	✓	✓
INTCSI01	CSIIF01		CSIMK01		CSIPR001, CSIPR101		—	—	—
INTIIC01	IICIF01		IICMK01		IICPR001, IICPR101		—	—	—
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101		✓	✓	✓
INTTM02	TMIF02		TMMK02		TMPR002, TMPR102		✓	✓	✓
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103		✓	✓	✓
INTAD	ADIF	IF1H	ADMK	MK1H	ADPR0, ADPR1	PR01H, PR11H	✓	✓	✓
INTRTC	RTCIF		RTCMK		RTCPR0, RTCPR1		✓	✓	✓
INTITL	ITLIF		ITLMK		ITLPR0, ITLPR1		✓	✓	✓
INTKR	KRIF		KRMK		KRPR0, KRPR1		✓	✓	✓
INTST3	STIF3		STMK3		STPR03, STPR13		—	—	—
INTCSI30	CSIIF30		CSIMK30		CSIPR030, CSIPR130		—	—	—
INTIIC30	IICIF30		IICMK30		IICPR030, IICPR130		—	—	—
INTSR3	SRIF3		SRMK3		SRPR03, SRPR13		—	—	—
INTCSI31	CSIIF31		CSIMK31		CSIPR031, CSIPR131		—	—	—
INTIIC31	IICIF31		IICMK31		IICPR031, IICPR131		—	—	—
INTTM13	TMIF13		TMMK13		TMPR013, TMPR113		✓	✓	—
INTTM04	TMIF04		TMMK04		TMPR004, TMPR104		✓	✓	✓

Note 1. If either of the interrupt sources INTSR1 and INTIIC11 is generated, bit 1 of the IF1L register is set to 1. Bit 1 of the MK1L, PR01L, and PR11L registers supports these two interrupt sources.

Note 2. Do not use a UART1 reception error interrupt and an interrupt of channel 3 of TAU0 (at higher 8-bit timer operation) at the same time because they share flags for the interrupt request sources. If the UART1 reception error interrupt is not used (EOC03 = 0), UART1 and channel 3 of TAU0 (at higher 8-bit timer operation) can be used at the same time. If either of the interrupt sources INTSRE1 or INTTM03H is generated, bit 2 of the IF1L register is set to 1. Bit 2 of the MK1L, PR01L, and PR11L registers supports these two interrupt sources.

Table 21 - 4 Flags Corresponding to Interrupt Request Sources (R7F104G) (4/5)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		48-pin	44-pin	32-pin
		Register		Register		Register			
INTTM05	TMIF05	IF2L	TMMK05	MK2L	TMPR005, TMPR105	PR02L, PR12L	✓	✓	✓
INTTM06	TMIF06		TMMK06		TMPR006, TMPR106		✓	✓	✓
INTTM07	TMIF07		TMMK07		TMPR007, TMPR107		✓	✓	✓
INTP6	PIF6		PMK6		PPR06, PPR16		—	—	—
INTP7	PIF7		PMK7		PPR07, PPR17		—	—	—
INTP8	PIF8		PMK8		PPR08, PPR18		—	—	—
INTP9	PIF9		PMK9		PPR09, PPR19		—	—	—
INTFL	FLIF		FLMK		FLPR0, FLPR1		✓	✓	✓
INTP10Note 1	PIF10Note 1	IF2H	PMK10Note 1	MK2H	PPR010, PPR110Note 1	PR02H, PR12H	✓	✓	✓
INTCMP0Note 1	CMPIF0Note 1		CMPMK0Note 1		CMPPR00, CMPPR10Note 1		✓	✓	✓
INTP11	PIF11		PMK11		PPR011, PPR111		✓	✓	✓
INTCMP1	CMPIF1		CMPMK1		CMPPR01, CMPPR11		—	—	—
INTURE0Note 2	UREIF0Note 2		UREMK0Note 2		UREPR00, UREPR10Note 2		✓	✓	✓
INTTM10Note 2	TMIF10Note 2		TMMK10Note 2		TMPR010, TMPR110Note 2		✓	✓	—
INTURE1	UREIF1		UREMK1		UREPR01, UREPR11		—	—	—
INTTM11	TMIF11		TMMK11		TMPR011, TMPR111		✓	✓	—
INTTM12	TMIF12		TMMK12		TMPR012, TMPR112		✓	✓	—
INTSRE3	SREIF3		SREMK3		SREPR03, SREPR13		—	—	—
INTTM13H	TMIF13H		TMMK13H		TMPR013H, TMPR113H		✓	✓	—
INTCTSUWR	CTSUWRIF		CTSUWRMK		CTSUWRPR0, CTSUWRPR1		✓	✓	✓
INTIICA1	IICAIF1		IICAMK1		IICAPR01, IICAPR11		—	—	—

Note 1. Do not use INTP10 and comparator 0 at the same time because they share the same flag as interrupt request sources. If either of the interrupt sources INTP10 or INTCMP0 is generated, bit 0 of the IF2H register is set to 1. Bit 0 of the MK2H, PR02H, and PR12H registers supports these two interrupt sources.

Note 2. Do not use a UARTA0 reception error interrupt and an interrupt of channel 0 of TAU1 at the same time because they share flags for the interrupt request sources. If the UARTA0 reception error interrupt is not used (ISRMA0 = 1), UARTA0 and channel 0 of TAU1 can be used at the same time. If either of the interrupt sources INTURE0 or INTTM10 is generated, bit 2 of the IF2H register is set to 1. Bit 2 of the MK2H, PR02H, and PR12H registers supports these two interrupt sources.

Table 21 - 4 Flags Corresponding to Interrupt Request Sources (R7F104G) (5/5)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		48-pin	44-pin	32-pin
		Register		Register		Register			
INTCTSURD	CTSURDIF	IF3L	CTSURDMK	MK3L	CTSURDPR0, CTSURDPR1	PR03L, PR13L	✓	✓	✓
INTCTSUFN	CTSUFNIF		CTSUFNMK		CTSUFNPR0, CTSUFNPR1		✓	✓	✓
INTREMC	REMCIF		REMCMK		REMCPR0, REMCPR1		✓	✓	✓
INTUT0	UTIF0		UTMK0		UTPR00, UTPR10		✓	✓	✓
INTUR0	URIF0		URMK0		URPR00, URPR10		✓	✓	✓
INTUT1	UTIF1		UTMK1		UTPR01, UTPR11		—	—	—
INTUR1	URIF1		URMK1		URPR01, URPR11		—	—	—
INTTM14	TMIF14		TMMK14		TMPR014, TMPR114		—	—	—
INTTM15	TMIF15	IF3H	TMMK15	MK3H	TMPR015, TMPR115	PR03H, PR13H	—	—	—
INTTM16	TMIF16		TMMK16		TMPR016, TMPR116		—	—	—
INTTM17	TMIF17		TMMK17		TMPR017, TMPR117		—	—	—

21.3.1 Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H)

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 the given interrupt request is acknowledged, a reset signal is generated, or an instruction is executed. 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, 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, 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. The value of each register following a reset is 00H.

Remark If an instruction to write to these registers is executed, the number of clock cycles executing the instruction takes is increased by two.

Figure 21 - 2 Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H) (1/2)

Address: FFFE0H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIF	WDTIIF

Address: FFFE1H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	SREIF0 TMIF01H	TMIF00	STIF0 CSIIF00 IICIF00	SMSEIF	ELCLIF	SREIF2 TMIF11H	SRIF2 CSIIF21 IICIF21	STIF2 CSIIF20 IICIF20

Address: FFFE2H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	TMIF03	TMIF02	TMIF01	SRIF0 CSIIF01 IICIF01	IICAIF0	SREIF1 TMIF03H	SRIF1 CSIIF11 IICIF11	STIF1 CSIIF10 IICIF10

Address: FFFE3H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1H	TMIF04	TMIF13	SRIF3 CSIIF31 IICIF31	STIF3 CSIIF30 IICIF30	KRIF	ITLIF	RTCIF	ADIF

Figure 21 - 2 Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, IF3L, IF3H) (2/2)

Address: FFFD0H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2L	FLIF	PIF9	PIF8	PIF7	PIF6	TMIF07	TMIF06	TMIF05

Address: FFFD1H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2H	IICAIF1	CTSUWRIF	SREIF3 TMIF13H	TMIF12	UREIF1 TMIF11	UREIF0 TMIF10	PIF11 CMPIF1	PIF10 CMPIF0

Address: FFFD2H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF3L	TMIF14	URIF1	UTIF1	URIF0	UTIF0	REMCIF	CTSUFNIF	CTSURDIF

Address: FFFD3H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
IF3H	0	0	0	0	0	TMIF17	TMIF16	TMIF15

xxIFx	Interrupt request flag
0	No interrupt request signal is generated
1	Indicates the generation of the interrupt request signal and the interrupt request currently being in the active state.

Caution 1. The available registers and bits differ depending on the product. For details about the registers and bits available for each product, see Table 21 - 3 Flags Corresponding to Interrupt Request Sources (R7F100G) and Table 21 - 4 Flags Corresponding to Interrupt Request Sources (R7F104G). Be sure to set bits that are not available to the initial value.

Caution 2. 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 memory manipulation instruction such as "IF0L.0 = 0;" or "_asm("clr1 IF0L.0");" because the compiled assembler must be a 1-bit memory manipulation instruction (CLR1). If a program is described in C language using an 8-bit memory manipulation instruction such as "IF0L & = 0xfe;" and compiled, it becomes the assembler of three instructions.

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

In this case, even if the request flag of the another bit of the same interrupt request flag register (IF0L) is set to 1 at the timing between "mov a, IF0L" and "mov IF0L, a", the flag is cleared to 0 at "mov IF0L, a". Therefore, care must be exercised when using an 8-bit memory manipulation instruction in C language.

21.3.2 Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt. 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, and 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. The value of each register following a reset is FFH.

Remark If an instruction to write to these registers is executed, the number of clock cycles executing the instruction takes is increased by two.

Figure 21 - 3 Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H) (1/2)

Address: FFFE4H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK	WDTIMK

Address: FFFE5H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	SREMK0 TMMK01H	TMMK00	STMK0 CSIMK00 IICMK00	SMSEMK	ELCLMK	SREMK2 TMMK11H	SRMK2 CSIMK21 IICMK21	STMK2 CSIMK20 IICMK20

Address: FFFE6H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	TMMK03	TMMK02	TMMK01	SRMK0 CSIMK01 IICMK01	IICAMK0	SREMK1 TMMK03H	SRMK1 CSIMK11 IICMK11	STMK1 CSIMK10 IICMK10

Address: FFFE7H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1H	TMMK04	TMMK13	SRMK3 CSIMK31 IICMK31	STMK3 CSIMK30 IICMK30	KRMK	ITLMK	RTCMK	ADMK

Figure 21 - 3 Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, MK3L, MK3H) (2/2)

Address: FFFD4H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2L	FLMK	PMK9	PMK8	PMK7	PMK6	TMMK07	TMMK06	TMMK05

Address: FFFD5H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2H	IICAMK1	CTSUWRMK	SREMK3 TMMK13H	TMMK12	UREMK1 TMMK11	UREMK0 TMMK10	PMK11 CMPMK1	PMK10 CMPMK0

Address: FFFD6H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK3L	TMMK14	URMK1	UTMK1	URMK0	UTMK0	REMCMK	CTSUFNMK	CTSURDMK

Address: FFFD7H
After reset: FFH
R/W: R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
MK3H	1	1	1	1	1	TMMK17	TMMK16	TMMK15

xxMKx	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Caution The available registers and bits differ depending on the product. For details about the registers and bits available for each product, see Table 21 - 3 Flags Corresponding to Interrupt Request Sources (R7F100G) and Table 21 - 4 Flags Corresponding to Interrupt Request Sources (R7F104G). Be sure to set bits that are not available to the initial value.

21.3.3 Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H)

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 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. The value of each register following a reset is FFH.

Remark If an instruction to write to these registers is executed, the number of clock cycles executing the instruction takes is increased by two.

Figure 21 - 4 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H) (1/4)

Address: FFFE8H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00L	PPR05	PPR04	PPR03	PPR02	PPR01	PPR00	LVIPR0	WDTIPR0

Address: FFFECH
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10L	PPR15	PPR14	PPR13	PPR12	PPR11	PPR10	LVIPR1	WDTIPR1

Address: FFFE9H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00H	SREPR00 TMPR001H	TMPR000	STPR00 CSIPR000 IICPR000	SMSEPR0	ELCLPR0	SREPR02 TMPR011H	SRPR02 CSIPR021 IICPR021	STPR02 CSIPR020 IICPR020

Figure 21 - 4 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H) (2/4)

Address: FFFEDH
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10H	SREPR10 TMPR101H	TMPR100	STPR10 CSIPR100 IICPR100	SMSEPR1	ELCLPR1	SREPR12 TMPR111H	SRPR12 CSIPR121 IICPR121	STPR12 CSIPR120 IICPR120

Address: FFFEAH
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01L	TMPR003	TMPR002	TMPR001	SRPR00 CSIPR001 IICPR001	IICAPR00	SREPR01 TMPR003H	SRPR01 CSIPR011 IICPR011	STPR01 CSIPR010 IICPR010

Address: FFFEEH
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11L	TMPR103	TMPR102	TMPR101	SRPR10 CSIPR101 IICPR101	IICAPR10	SREPR11 TMPR103H	SRPR11 CSIPR111 IICPR111	STPR11 CSIPR110 IICPR110

Address: FFFEBH
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01H	TMPR004	TMPR013	SRPR03 CSIPR031 IICPR031	STPR03 CSIPR030 IICPR030	KRPR0	ITLPR0	RTCPR0	ADPR0

Address: FFFEFH
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11H	TMPR104	TMPR113	SRPR13 CSIPR131 IICPR131	STPR13 CSIPR130 IICPR130	KRPR1	ITLPR1	RTCPR1	ADPR1

Figure 21 - 4 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H) (3/4)

Address: FFFD8H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02L	FLPR0	PPR09	PPR08	PPR07	PPR06	TMPR007	TMPR006	TMPR005

Address: FFFDCH
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12L	FLPR1	PPR19	PPR18	PPR17	PPR16	TMPR107	TMPR106	TMPR105

Address: FFFD9H
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02H	IICAPR01	CTSUWRPR0	SREPR03 TMPR013H	TMPR012	UREPR01 TMPR011	UREPR00 TMPR010	PPR011 CMPPR01	PPR010 CMPPR00

Address: FFFDDH
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12H	IICAPR11	CTSUWRPR1	SREPR13 TMPR113H	TMPR112	UREPR11 TMPR111	UREPR10 TMPR110	PPR111 CMPPR11	PPR110 CMPPR10

Address: FFFDAH
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR03L	TMPR014	URPR01	UTPR01	URPR00	UTPR00	REMCPR0	CTSUFNPR0	CTSURDPR0

Address: FFFDEH
After reset: FFH
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR13L	TMPR114	URPR11	UTPR11	URPR10	UTPR10	REMCPR1	CTSUFNPR1	CTSURDPR1

Figure 21 - 4 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR03L, PR03H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H, PR13L, PR13H) (4/4)

Address: FFFDBH
After reset: FFH
R/W: R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
PR03H	1	1	1	1	1	TMPR017	TMPR016	TMPR015

Address: FFFDFH
After reset: FFH
R/W: R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
PR13H	1	1	1	1	1	TMPR117	TMPR116	TMPR115

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 available registers and bits differ depending on the product. For details about the registers and bits available for each product, see Table 21 - 3 Flags Corresponding to Interrupt Request Sources (R7F100G) and Table 21 - 4 Flags Corresponding to Interrupt Request Sources (R7F104G). Be sure to set bits that are not available to the initial value.

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

These registers specify the valid edge for INTP0 to INTP11. The EGP0, EGP1, EGN0, and EGN1 registers can be set by a 1-bit or 8-bit memory manipulation instruction. The value of each register following a reset is 00H.

Figure 21 - 5 Format of External Interrupt Rising Edge Enable Registers (EGP0, EGP1) and External Interrupt Falling Edge Enable Registers (EGN0, EGN1)

Address: FFF38H
After reset: 00H
R/W: 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: 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: R/W

Symbol	7	6	5	4	3	2	1	0
EGP1	0	0	0	0	EGP11	EGP10	EGP9	EGP8

Address: FFF3BH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
EGN1	0	0	0	0	EGN11	EGN10	EGN9	EGN8

EGPn	EGNn	INTPn pin valid edge selection
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Remark n = 0 to 11

Table 21 - 5 shows the port pins (INTPx) corresponding to the EGPn and EGNn bits.

Table 21 - 5 Port Pins (INTPx) Corresponding to EGPn and EGNn Bits

Detection Enable Bit		Port Pins (INTPx)	R7F100G				R7F104G
			64- to 128-pin	52-pin	48-pin	30- to 44-pin	32- to 48-pin
EGP0	EGN0	INTP0	✓	✓	✓	✓	✓
EGP1	EGN1	INTP1	✓	✓	✓	✓	✓
EGP2	EGN2	INTP2	✓	✓	✓	✓	—
EGP3	EGN3	INTP3	✓	✓	✓	✓	✓
EGP4	EGN4	INTP4	✓	✓	✓	✓	✓
EGP5	EGN5	INTP5	✓	✓	✓	✓	✓
EGP6	EGN6	INTP6	✓	✓	✓	—	—
EGP7	EGN7	INTP7	✓	—	—	—	—
EGP8	EGN8	INTP8	✓	✓	✓	—	—
EGP9	EGN9	INTP9	✓	✓	✓	—	—
EGP10	EGN10	INTP10	✓	✓	—	—	✓
EGP11	EGN11	INTP11	✓	✓	—	—	✓

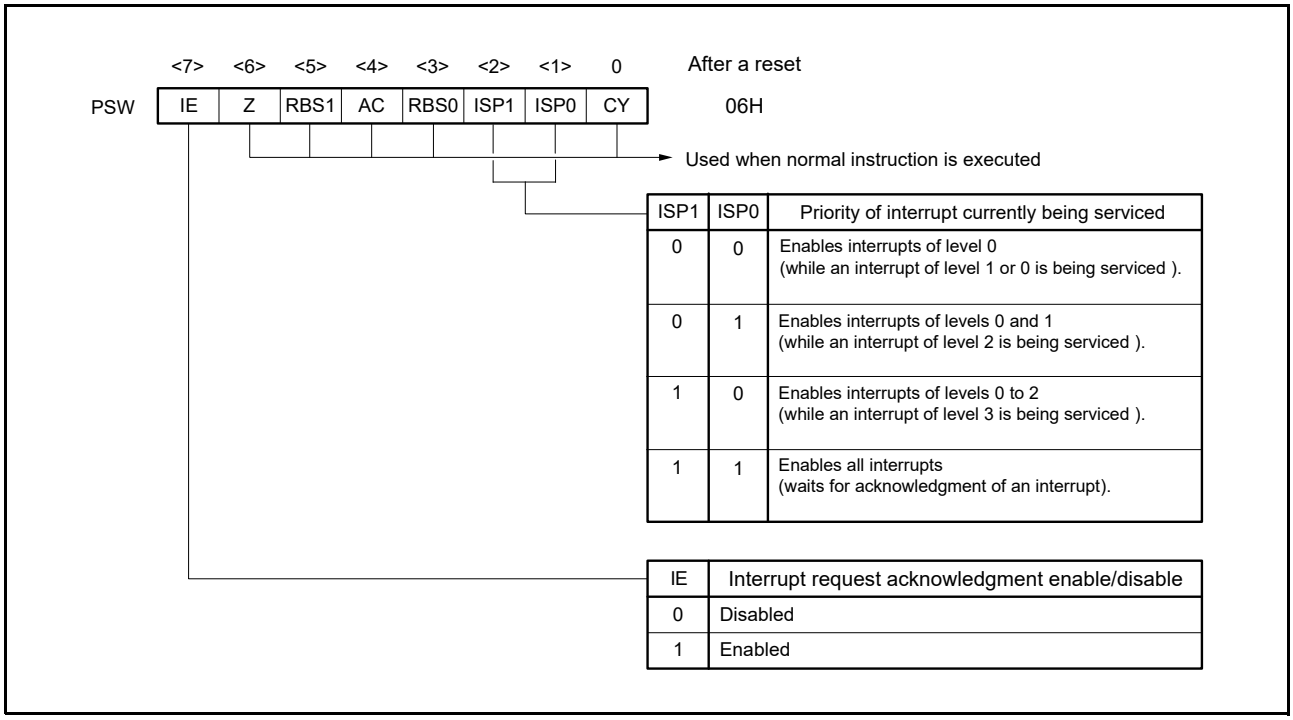
Caution **When the input port pins used for the external interrupt functions are switched to the output mode, the INTPn interrupt might be generated upon detection of a valid edge. When switching the input port pins to the output mode, set the corresponding port mode register (PMxx) to 0 after disabling the edge detection (by setting EGPn and EGNn to 0).**

Remark 1. For edge detection ports, see **2.1 Functions of Port Pins**.
Remark 2. n = 0 to 11

21.3.5 Program status word (PSW)

The program status word (PSW) is a register that holds the results of instruction execution and the current state for interrupts. The IE flag that sets maskable interrupt enable/disable and the ISP1 and ISP0 flags that control multiple interrupt servicing are mapped to the PSW. Besides 8-bit read/write, this register can carry out operations using bit memory 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. Upon acknowledgment of a maskable interrupt request, if the value of the priority specification flag register of the acknowledged interrupt is not 00, its value minus 1 is transferred to the ISP1 and ISP0 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. The value of the PSW following a reset is 06H.

Figure 21 - 6 Configuration of Program Status Word



ISP1	ISP0	Priority of interrupt currently being serviced
0	0	Enables interrupts of level 0 (while an interrupt of level 1 or 0 is being serviced).
0	1	Enables interrupts of levels 0 and 1 (while an interrupt of level 2 is being serviced).
1	0	Enables interrupts of levels 0 to 2 (while an 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

21.3.6 Registers for controlling the port functions multiplexed with the interrupt inputs

Set the following registers to control the port functions multiplexed with the interrupt inputs.

- Port mode registers (PMxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)

For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.8 Port mode control T registers (PMCTxx)**, and **4.3.9 Port mode control E registers (PMCExx)**. For examples of register settings for controlling port functions, see **4.5.4 Examples of register settings for port and multiplexed functions**.

21.4 Interrupt Servicing Operations

21.4.1 Maskable interrupt request acknowledgment

A maskable interrupt request becomes acknowledgeable when the interrupt request flag is set to 1 and the mask (MK) flag corresponding to that interrupt request is cleared to 0. A vectored interrupt request is acknowledged if interrupts are in the interrupt enabled state (when the IE flag is set to 1). However, a low-priority interrupt request is not acknowledged during servicing of a higher priority interrupt request. **Table 21 - 6** shows the times required from acceptance of an interrupt request in the form of the generation of a maskable interrupt request until vectored interrupt servicing proceeds. For the interrupt request acknowledgment timing, see **Figure 21 - 8** and **Figure 21 - 9**.

Table 21 - 6 Times from Generation of Maskable Interrupt Request until Vectored Interrupt Servicing

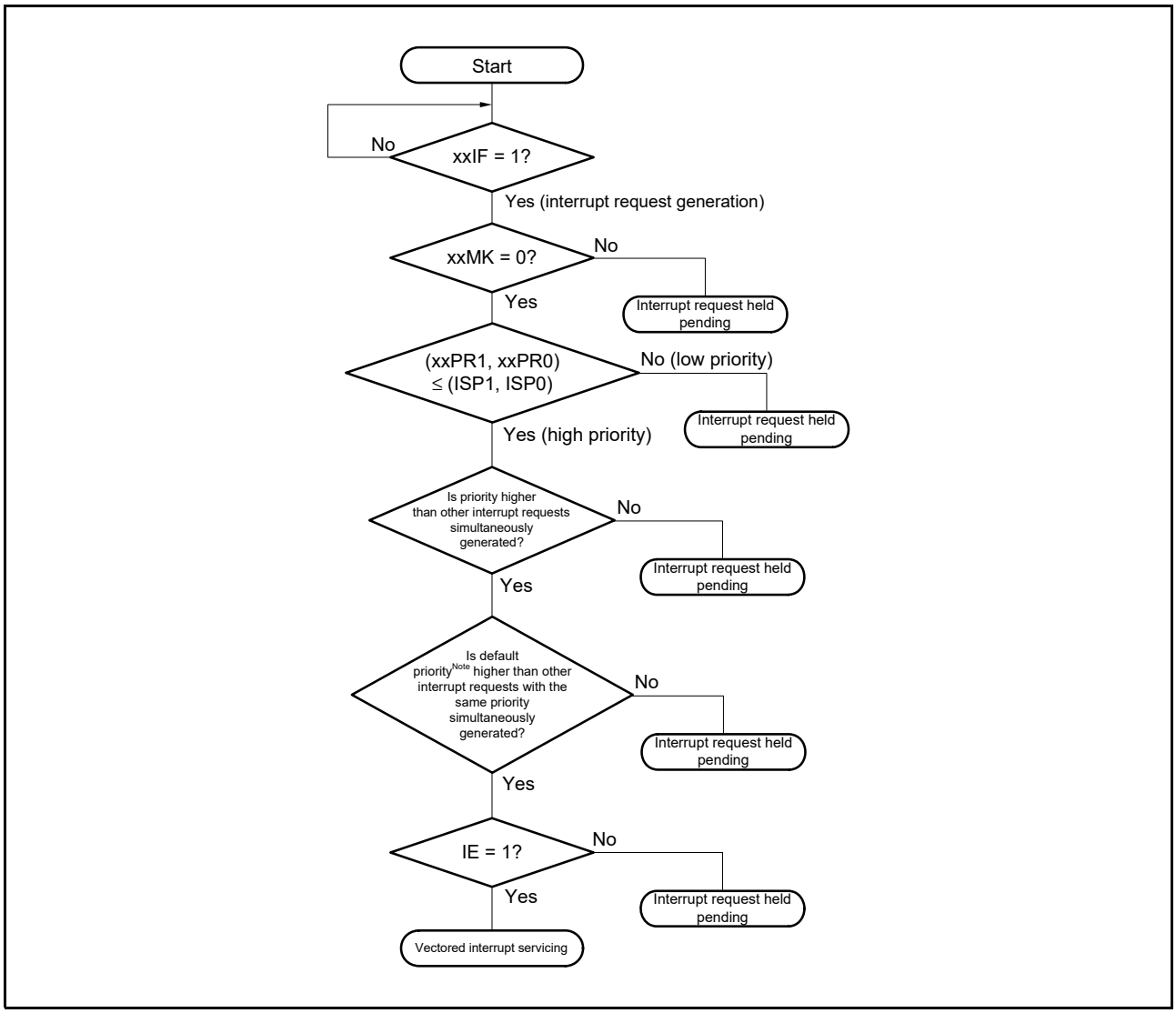
	Minimum Time	Maximum Time[Note]
Times required for interrupt request acceptance	9 clock cycles	16 clock cycles

Note Maximum time does not apply when an instruction from the internal RAM area is executed.

Remark One clock cycle is equivalent to 1/f_{CLK}, where f_{CLK} is the frequency of the 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 interrupt requests have the same priority level, the request with the highest default priority is acknowledged first. An interrupt request that is held pending is acknowledged when it becomes acknowledgeable. **Figure 21 - 7** shows the interrupt request acknowledgment algorithm. If a maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag is reset to 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 program control branches to the specified servicing. Restoring from an interrupt is possible by using the RETI instruction.

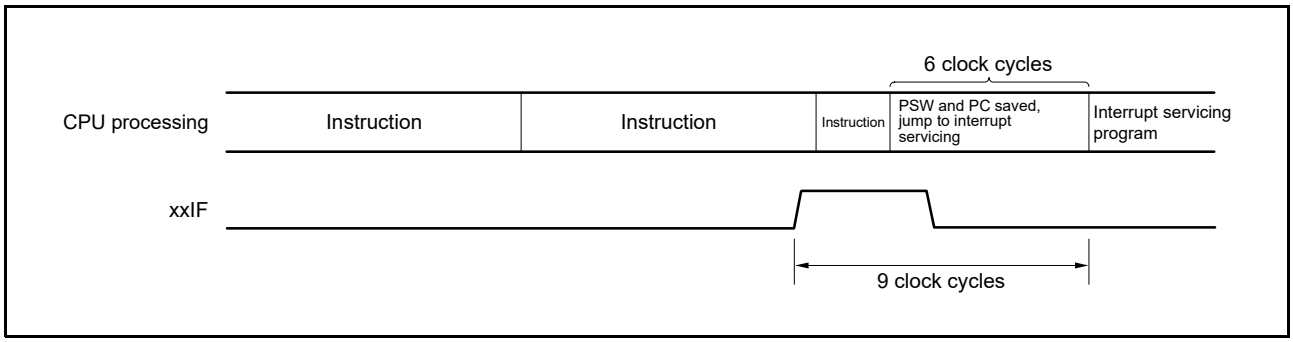
Figure 21 - 7 Interrupt Request Acknowledgment Processing Algorithm



Note For the default priority, refer to **Table 21 - 1 Interrupt Source List (R7F100G)** and **Table 21 - 2 Interrupt Source List (R7F104G)**.

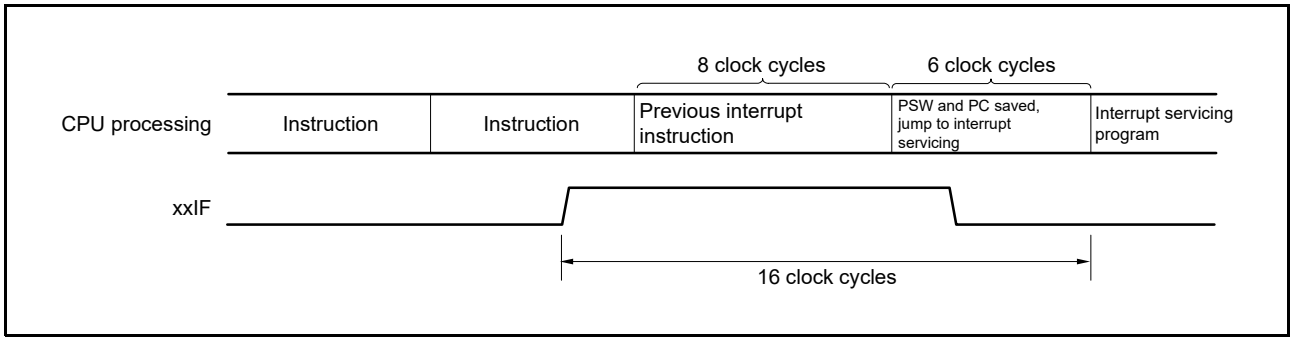
Remark xxIF: Interrupt request flag
xxMK: Interrupt mask flag
xxPR0: Priority specification flag 0
xxPR1: Priority specification flag 1
IE: Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)
ISP1, ISP0: Flags that indicate the priority level of the interrupt currently being serviced (see **Figure 21 - 6 Configuration of Program Status Word**)

Figure 21 - 8 Minimum Time Taken until Acknowledgment of an Interrupt Request



Remark One clock cycle is equivalent to 1/f_{CLK}, where f_{CLK} is the frequency of the CPU clock.

Figure 21 - 9 Maximum Time Taken until Acknowledgment of an Interrupt Request



Remark One clock cycle is equivalent to 1/f_{CLK}, where f_{CLK} is the frequency of the CPU clock.

21.4.2 Software interrupt request acknowledgment

A software interrupt request is acknowledged by BRK instruction execution. Software interrupts cannot be disabled. If a software interrupt request is acknowledged, the contents are saved into the stacks in the order of the program status word (PSW), then program counter (PC), the IE flag is reset to 0, the contents of the vector table (0007EH, 0007FH) are loaded into the PC, and program control branches to the specified servicing. Restoring from a software interrupt is possible by using the RETB instruction.

Caution The RETI instruction cannot be used for restoring from the software interrupt.

21.4.3 Multiple interrupt servicing

Multiple interrupt servicing occurs when another interrupt request is acknowledged during execution of an interrupt. Multiple interrupt servicing does not occur unless the interrupt request acknowledgment enabled state is selected (IE = 1). When an interrupt request is acknowledged, interrupt request acknowledgment becomes disabled (IE = 0). Therefore, to enable multiple interrupt servicing, it is necessary to set the IE flag to 1 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.

Table 21 - 7 shows relationship between interrupt requests enabled for multiple interrupt servicing and **Figure 21 - 10** shows multiple interrupt servicing examples.

Table 21 - 7 Relationship between Interrupt Requests Enabled for Multiple Interrupt Servicing during Interrupt Servicing

Multiple Interrupt Request / Maskable Interrupt		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	
Processing in progress	ISP1 and ISP0 = 00B	✓	—	—	—	—	—	—	—	✓
	ISP1 and ISP0 = 01B	✓	—	✓	—	—	—	—	—	✓
	ISP1 and ISP0 = 10B	✓	—	✓	—	✓	—	—	—	✓
Waiting for reception	ISP1 and ISP0 = 11B	✓	—	✓	—	✓	—	✓	—	✓

Remark 1. ✓: Multiple interrupt servicing enabled

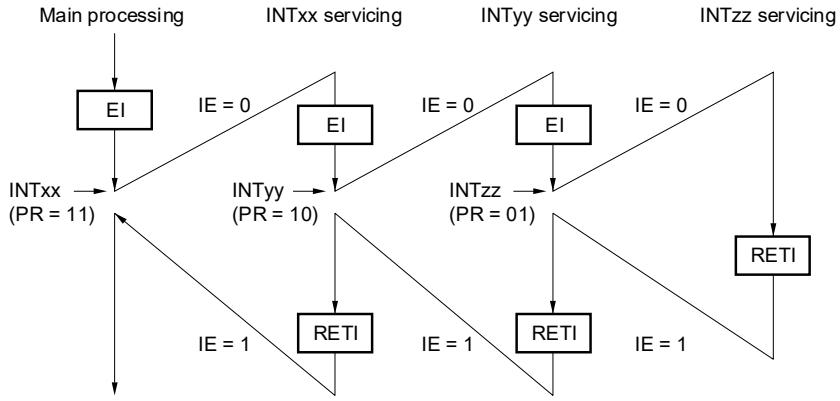
Remark 2. —: Multiple interrupt servicing disabled

Remark 3. ISP1, ISP0, and IE are flags contained in the PSW.
ISP1 and ISP0 = 00B: An interrupt of level 1 or level 0 is being serviced.
ISP1 and ISP0 = 01B: An interrupt of level 2 is being serviced.
ISP1 and ISP0 = 10B: An interrupt of level 3 is being serviced.
ISP1 and ISP0 = 11B: Wait for an interrupt acknowledgment (all interrupts are enabled).
IE = 0: Interrupt request acknowledgment is disabled.
IE = 1: Interrupt request acknowledgment is enabled.

Remark 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 xxPR1x = 0, xxPR0x = 0 (higher priority level)
PR = 01: Specify level 1 with xxPR1x = 0, xxPR0x = 1
PR = 10: Specify level 2 with xxPR1x = 1, xxPR0x = 0
PR = 11: Specify level 3 with xxPR1x = 1, xxPR0x = 1 (lower priority level)

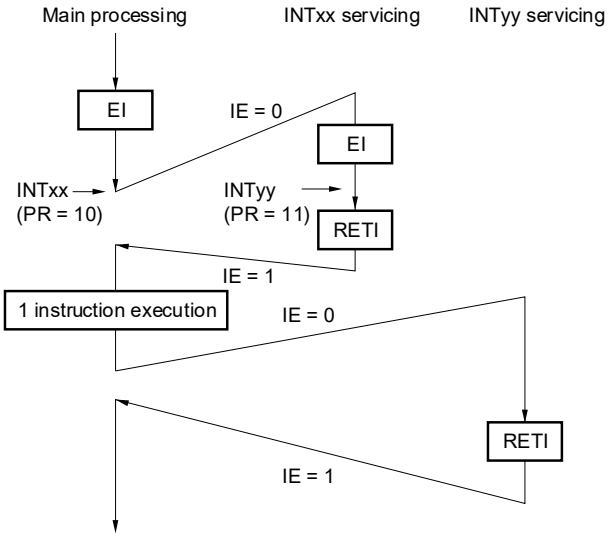
Figure 21 - 10 Examples of Multiple Interrupt Servicing (1/2)

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

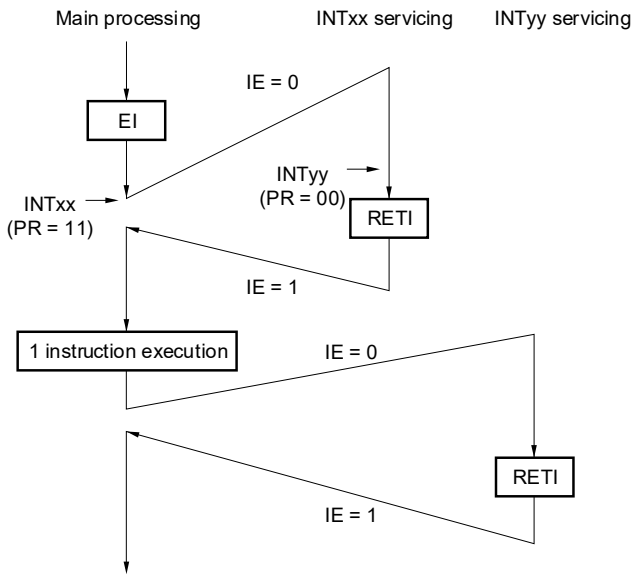


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

Remark PR = 00: Specify level 0 with xxPR1x = 0, xxPR0x = 0 (higher priority level)
PR = 01: Specify level 1 with xxPR1x = 0, xxPR0x = 1
PR = 10: Specify level 2 with xxPR1x = 1, xxPR0x = 0
PR = 11: Specify level 3 with xxPR1x = 1, xxPR0x = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled.
IE = 1: Interrupt request acknowledgment is enabled.

Figure 21 - 10 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.

Remark PR = 00: Specify level 0 with xxPR1x = 0, xxPR0x = 0 (higher priority level)
PR = 01: Specify level 1 with xxPR1x = 0, xxPR0x = 1
PR = 10: Specify level 2 with xxPR1x = 1, xxPR0x = 0
PR = 11: Specify level 3 with xxPR1x = 1, xxPR0x = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

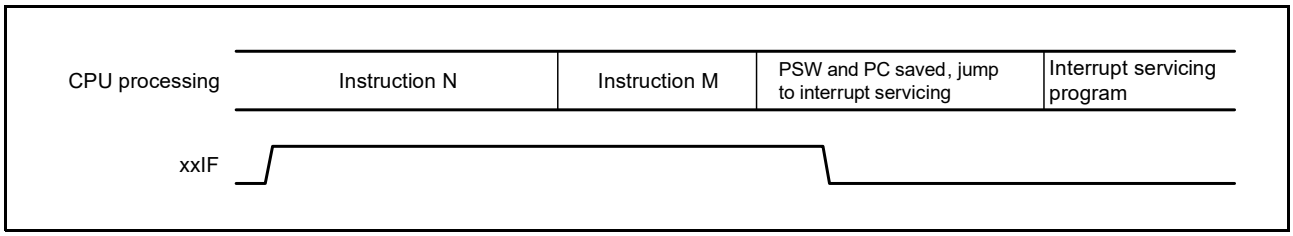
21.4.4 Interrupt request held pending

There are instructions where, even if an interrupt request is issued while the instructions are being executed, interrupt request acknowledgment is held pending until the end of execution of the next instruction. These instructions (instructions that hold interrupt requests pending) 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
- Write 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

Figure 21 - 11 shows the timing at which interrupt requests are held pending.

Figure 21 - 11 Holding Interrupt Requests Pending



Remark 1. Instruction N: Instruction to hold interrupt requests pending
Remark 2. Instruction M: Instruction other than the instruction to hold interrupt requests pending

Section 22 Key Interrupt Function

The number of key interrupt input channels differs, depending on the product.

	R7F100G				R7F104G
	30- to 36-pin products	40- and 44-pin products	48-pin products	52- to 128-pin products	32- to 48-pin products
Number of the key interrupt input channels	—	4	6	8	6

22.1 Functions of the Key Interrupt

A key interrupt (INTKR) can be generated by inputting a rising edge/falling edge to the key interrupt input pins (KR0 to KR7).

Table 22 - 1 Assignment of the Key Interrupt Detection Pins

Key interrupt pins	Key return mode register 0 (KRM0)
KR0	KRM00
KR1	KRM01
KR2	KRM02
KR3	KRM03
KR4	KRM04
KR5	KRM05
KR6	KRM06
KR7	KRM07

Remark Pins KR0 to KR3 are only present in the 40- and 44-pin R7F100G products.
Pins KR0 to KR5 are only present in the 48-pin R7F100G products.
Pins KR0 to KR7 are only present in the 52- to 128-pin R7F100G products.
Pins KR0 to KR3, KR6, and KR7 are only presents in the 32- to 48-pin R7F104G products.

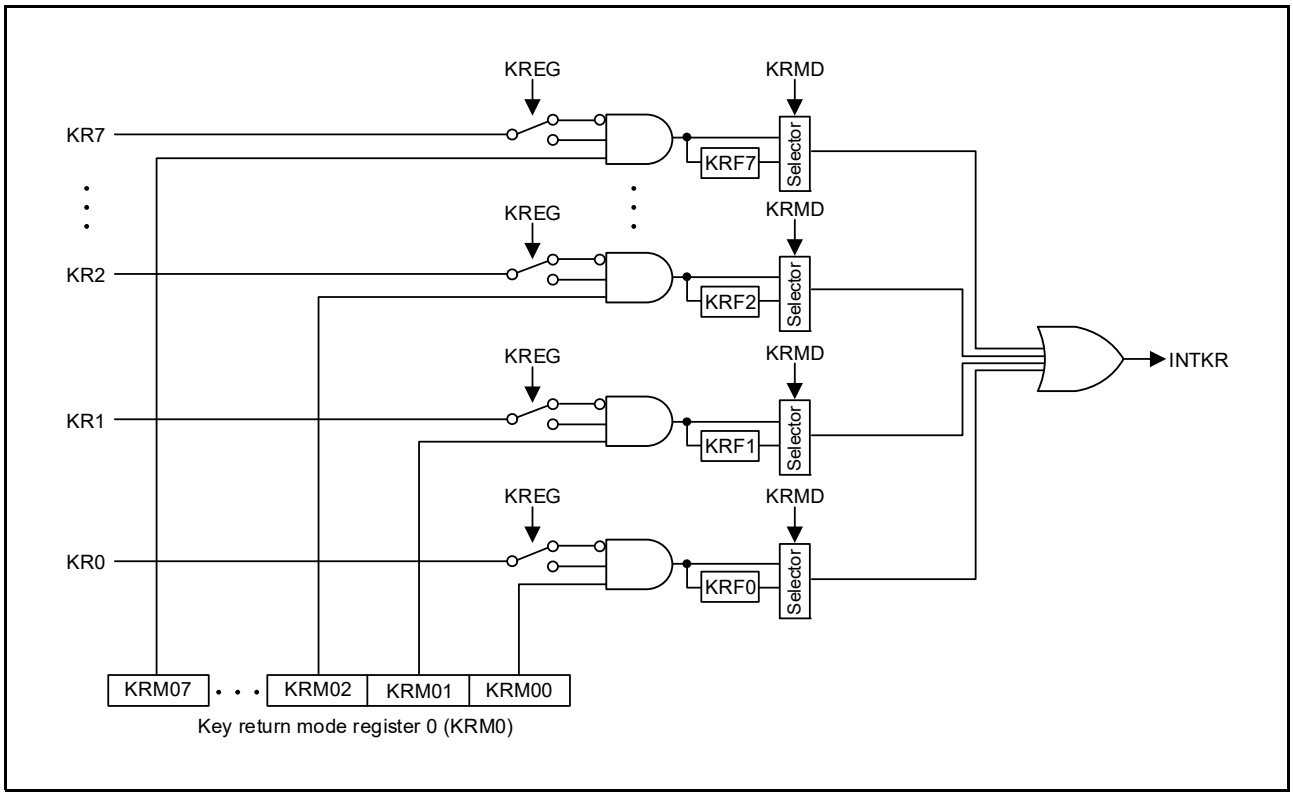
22.2 Configuration of the Key Interrupt

The key interrupt includes the following hardware blocks.

Table 22 - 2 Configuration of the Key Interrupt

Item	Configuration
Control registers	• Key return control register (KRCTL) • Key return mode register 0 (KRM0) • Key return flag register (KRF) • Port mode registers (PMxx)

Figure 22 - 1 Block Diagram of the Key Interrupt



Remark Pins KR0 to KR3 are only present in the 40- and 44-pin R7F100G products.
Pins KR0 to KR5 are only present in the 48-pin R7F100G products.
Pins KR0 to KR7 are only present in the 52- to 128-pin R7F100G products.
Pins KR0 to KR3, KR6, and KR7 are only presents in the 32- to 48-pin R7F104G products.

22.3 Registers for Controlling the Key Interrupt

The following registers are used to control the key interrupt.

- Key return control register (KRCTL)
- Key return mode register 0 (KRM0)
- Key return flag register (KRF)
- Port mode registers (PMxx)
- Port mode control T registers (PMCTxx)

22.3.1 Key return control register (KRCTL)

The KRCTL register controls the usage of the key return flags (KRF0 to KRF7) and sets the detection edge. The KRCTL register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 22 - 2 Format of Key Return Control Register (KRCTL)

Address: FFF34H
After reset: 00H
R/W: R/W

Symbol	<7>	6	5	4	3	2	1	<0>
KRCTL	KRMD	0	0	0	0	0	0	KREG

KRMD	Usage of key return flags (KRF0 to KRF7)
0	Does not use key return flags
1	Uses key return flags

KREG	Selection of detection edge (KR0 to KR7)
0	Falling edge
1	Rising edge

22.3.2 Key return mode register 0 (KRM0)

The KRM0 register controls the KR0 to KR7 signals. The KRM0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 22 - 3 Format of Key Return Mode Register 0 (KRM0)

Address: FFF37H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
KRM0	KRM07	KRM06	KRM05	KRM04	KRM03	KRM02	KRM01	KRM00

KRM0n	Key interrupt mode control
0	Does not detect key interrupt signal
1	Detects key interrupt signal

Caution 1. Be sure to leave the following bits set to 0.
Bits 7 to 0 in the 30- to 36-pin R7F100G products
Bits 7 to 4 in the 40- and 44-pin R7F100G products
Bits 7 and 6 in the 48-pin R7F100G products
Bits 5 and 4 in the 32- to 48-pin R7F104G products

Caution 2. The on-chip pull-up resistor can be applied by setting the corresponding key interrupt input pins (bits) in the pull-up resistor option register (PU7) to 1.

Caution 3. An interrupt will be generated if the target bit of the KRM0 register is set while a low level (the KREG bit is set to 0) or a high level (the KREG bit is set to 1) is being input to the key interrupt input pin. To ignore this interrupt, set the KRM0 register after disabling interrupt servicing by using the interrupt mask flag. Afterward, clear the interrupt request flag and enable interrupt servicing after waiting for the key interrupt input high-level and low-level widths (see 37.4 AC Characteristics).

Caution 4. The pins not used in the key interrupt mode can be used as normal port pins.

Remark n = 0 to 7

22.3.3 Key return flag register (KRF)

The KRF register controls the key interrupt flags (KRF0 to KRF7). The KRF register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 22 - 4 Format of Key Return Flag Register (KRF)

Address: FFF35H
After reset: 00H
R/W: R/W**Note**

Symbol	7	6	5	4	3	2	1	0
KRF	KRF7	KRF6	KRF5	KRF4	KRF3	KRF2	KRF1	KRF0

KRFn	Key interrupt flag
0	No key interrupt signal has been detected.
1	A key interrupt signal has been detected.

Note Writing 1 has no effect. To clear the KRFn bit, write 0 to the corresponding bit and 1 to the other bits using an 8-bit memory manipulation instruction.

Remark n = 0 to 7

22.3.4 Registers for controlling the port functions multiplexed with the key interrupt inputs

Set the following registers to control the port functions multiplexed with the key interrupt inputs.

- Port mode registers (PMxx)
- Port mode control T registers (PMCTxx)

For details, see **4.3.1 Port mode registers (PMxx)** and **4.3.8 Port mode control T registers (PMCTxx)**. For examples of register settings for controlling port functions, see **4.5.4 Examples of register settings for port and multiplexed functions**.

Section 23 Standby Function

23.1 Standby Function

The standby function reduces the operating current of the system, and the following three modes are available.

1. HALT mode
   Executing a HALT instruction places this MCU in the HALT mode. In the HALT mode, the operating clock of the CPU is stopped. If the high-speed system clock oscillator, high-speed on-chip oscillator, middle-speed on-chip oscillator, or subsystem clock 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
   Executing a STOP instruction places this MCU in the STOP mode. In the STOP mode, the high-speed system clock oscillator, high-speed on-chip oscillator, and middle-speed on-chip oscillator stop, stopping the whole system, thereby considerably reducing the operating current of the CPU. An interrupt request can trigger release from this mode, so it enables intermittent operation. However, waiting is required to secure the oscillation stabilization time after release from the STOP mode when the X1 clock is selected, so select the HALT mode instead if an immediate start of processing upon generation of an interrupt request is necessary.

3. SNOOZE mode
   In SNOOZE mode, this MCU is released from STOP mode and the following peripheral modules can operate without CPU intervention. For details, see the sections on the individual modules.
   - **Section 12 A/D Converter (ADC)**
   - **Section 15 Serial Array Unit (SAU)**
   - **Section 18 Remote Control Signal Receiver (REMC)**
   - **Section 19 Data Transfer Controller (DTC)**
   - **Section 29 SNOOZE Mode Sequencer (SMS)**
   - **Section 30 Capacitive Sensing Unit (CTSU2L)**

   This can only be specified when the high-speed on-chip oscillator or middle-speed on-chip oscillator is selected for the CPU/peripheral hardware clock (f_{CLK}).

In any of these modes, registers, flags, and data memory retain all values they had immediately before the transition to the standby mode.**Note** The states of the output latches and the output buffers for I/O port pins are also retained.

Note Memory which has been placed in shutdown mode by the PSMCR register does not retain values it had immediately before the transition to the standby mode.

Caution 1. Shifting to the STOP mode is only enabled when the CPU is operating with the main system clock. Do not execute the STOP instruction while the CPU operates with the subsystem clock. Shifting to the HALT mode is enabled whether the CPU is operating with the main system clock or the subsystem clock.

Caution 2. When shifting to the STOP mode, be sure to stop the peripheral modules operating with the main system clock before executing the STOP instruction. Note that this does not apply to the peripheral modules to be used in the SNOOZE mode and the timer array unit for use with the remote control signal receiver (REMC) in the SNOOZE mode; that is, operations of these modules should not be stopped.

Caution 3. It can be selected by the WDTON bit of the option byte or the WUTMMCK0 bit of the subsystem clock supply mode control register (OSMC) whether the low-speed on-chip oscillator continues oscillating or stops in the HALT or STOP mode. For details, see 6.1 Functions of Clock Generator, 2. Subsystem clock, <2> Low-speed on-chip oscillator.

23.2 Registers for Controlling the Standby Function

The following registers are used to control the standby function.

- Subsystem clock supply mode control register (OSMC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)
- Standby mode release setting register (WKUPMD)
- Memory power reduction control register (PSMCR)

Remark For details of the OSMC, OSTC, and OSTS registers, see **Section 6 Clock Generator**. For registers which control the SNOOZE mode, see the following sections.

- **Section 12 A/D Converter (ADC)**
- **Section 15 Serial Array Unit (SAU)**
- **Section 18 Remote Control Signal Receiver (REMC)**
- **Section 19 Data Transfer Controller (DTC)**
- **Section 29 SNOOZE Mode Sequencer (SMS)**
- **Section 30 Capacitive Sensing Unit (CTSU2L)**

23.2.1 Standby mode release setting register (WKUPMD)

The WKUPMD register is used to set the operation at the time of release from the standby mode. The WKUPMD register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 23 - 1 Format of Standby Mode Release Setting Register (WKUPMD)

Address: F0215H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	<0>
WKUPMD	0	0	0	0	0	0	0	FWKUP

FWKUP	Setting for starting the high-speed on-chip oscillator at the times of release from STOP mode and of transitions to SNOOZE mode**Notes 1, 2**
0	Starting is at normal speed.**Note 3**
1	Starting is at high speed.**Note 3**

Note 1. This setting is only available when the high-speed on-chip oscillator is selected for the CPU clock.

Note 2. This register is initialized on release from STOP mode in response to the generation of a reset signal, so starting of the high-speed on-chip oscillator is at normal speed.

Note 3. For the activation time, see **23.3.2 STOP mode**. The accuracy of the high-speed on-chip oscillator frequency depends on whether starting of the high-speed on-chip oscillator is at normal speed or at high speed. See **Section 37 Electrical Characteristics**.

23.2.2 Memory power reduction control register (PSMCR)

The PSMCR register is used to control the reduction of power consumption by the RAM. The leakage current can be reduced by placing the RAM in shutdown mode. The supply of power to the shut-down part of the RAM stops. Accordingly, that RAM does not retain data. The PSMCR register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 23 - 2 Format of Memory Power Reduction Control Register (PSMCR)

Address: F0216H
After reset: 00H
R/W: R/W

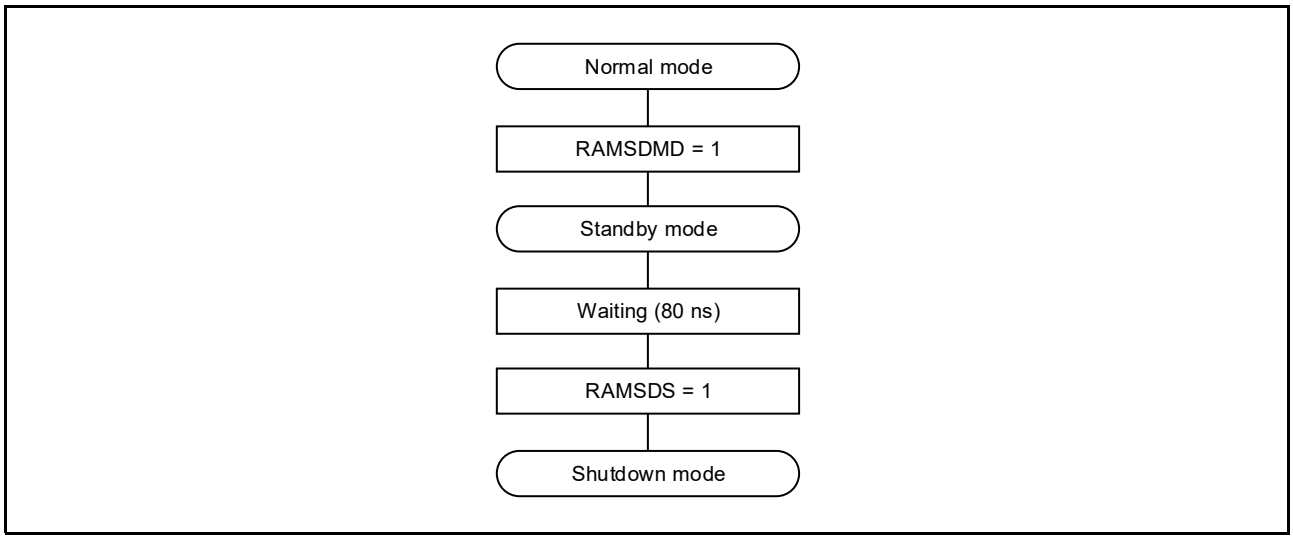
Symbol	7	6	5	4	3	2	<1>	<0>
PSMCR	0	0	0	0	0	0	RAMSDMD	RAMSDS

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 prohibited

Caution 1. Shutdown mode applies to all RAM other than that in the range from FF000H to FFEFFH. The RAM that in the range from FF000H to FFEFFH continues to operate and retains data.
Caution 2. Do not access RAM while it is in the standby mode or shutdown mode.
Caution 3. When the RAM returns to normal mode from shutdown mode, the contents of the RAM other than in the range from FF000H to FFEFFH are undefined.
Caution 4. Access to the address ranges other than FF000H to FFEFFH in the RAM in the shutdown mode is still possible in on-chip debugging. When the RAM returns to normal mode from shutdown mode in on-chip debugging, the contents of the RAM other than in the range from FF000H to FFEFFH are not undefined.

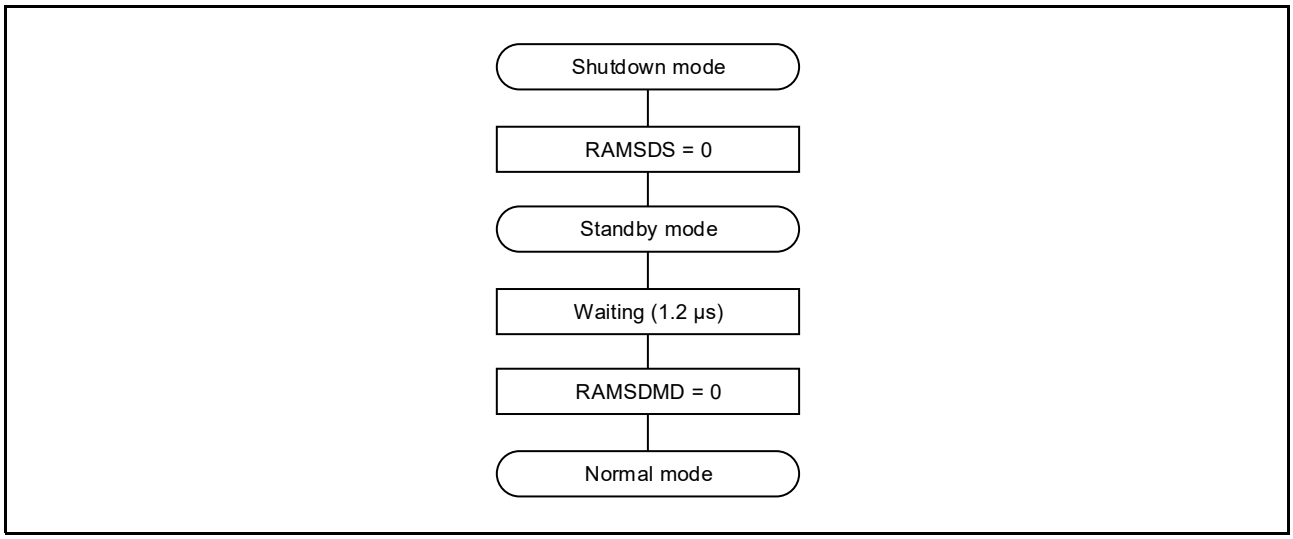
Follow the procedure in **Figure 23 - 3** to switch the operating mode of the RAM from normal mode to shutdown mode.

Figure 23 - 3 Procedure for Settings to Switch from Normal Mode to Shutdown Mode



Follow the procedure in **Figure 23 - 4** to switch the operating mode of the RAM from shutdown mode to normal mode.

Figure 23 - 4 Procedure for Settings to Switch from Shutdown Mode to Normal Mode



Caution When the RAM returns to normal mode from shutdown mode, the contents of the RAM other than in the range from FF000H to FFEFFH are undefined. Initialize the RAM area to be used.

23.3 Standby Operation

23.3.1 HALT mode

1. Setting of and operating states in HALT mode
   The HALT mode is set by executing the HALT instruction. This MCU can be placed in the HALT mode if the CPU clock before entry to the HALT mode is the high-speed system clock, high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock, or subsystem clock.

   Caution The interrupt request signal is used for release from the HALT mode, so if the interrupt mask flag is 0 (enabling the interrupt processing) and the interrupt request flag is 1 (the interrupt request signal is being generated), the HALT mode is not entered if these are the settings even when a HALT instruction is executed.

Table 23 - 1 and **Table 23 - 3** show the operating states in the HALT mode.

Table 23 - 1 Operating States in HALT Mode When the Main System Clock Is Used as the CPU Clock (1/2)

Item \ HALT Mode Setting			When a HALT Instruction is Executed While the CPU is Operating with the Main System Clock			
			When the CPU is Operating with the High-speed On-chip Oscillator Clock (f_{IH})	When the CPU is Operating with the Middle-speed On-chip Oscillator Clock (f_{IM})	When the CPU is Operating with the X1 Clock (f_X)	When the CPU is Operating with the External Main System Clock (f_{EX})
System clock			Clock supply to the CPU is stopped			
	Main system clock	f_{IH}	Operation continues (stopping this clock signal is not possible)	Operation disabled		
		f_{IM}	Operation disabled	Operation continues (stopping this clock signal is not possible)	Operation disabled	
		f_X	Operation disabled		Operation continues (stopping this clock signal is not possible)	Cannot operate
		f_{EX}			Cannot operate	Operation continues (stopping this clock signal is not possible)
	Subsystem clock	f_{XT}	Retains the state before the transition to HALT mode			
		f_{EXS}				
		f_{IL}	Set by the WDSTBYON and WDTON bits of the user option byte (000C0H), and the WUTMMCK0 bit of the subsystem clock supply mode control register (OSMC) WUTMMCK0 = 1 or SELLOSC = 1: Oscillates (Setting of WUTMMCK0 = 1 and SELLOSC = 1 is prohibited when the subsystem clocks X (f_{SX}) and XR (f_{SXR}) are operating.) WUTMMCK0 = 0, SELLOSC = 0, and WDTON = 0: Stopped WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 1: Oscillates WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 0: Stopped			
CPU			Operation stopped			
Code flash memory						
Data flash memory						
RAM			Operation stopped (however, operation in response to access by the DTC or SMS is possible)			
Port (latch)			Retains the state before the transition to HALT mode (however, operation in response to access by the DTC, SMS, or ELCL is possible)			
Timer array unit			Operation enabled			
RTC			Operation enabled			
32-bit interval timer			Operation enabled			
Watchdog timer			See **Section 11 Watchdog Timer (WDT)**.			

Table 23 - 1 Operating States in HALT Mode When the Main System Clock Is Used as the CPU Clock (2/2)

HALT Mode Setting / Item		When a HALT Instruction is Executed While the CPU is Operating with the Main System Clock			
		When the CPU is Operating with the High-speed On-chip Oscillator Clock (f_{IH})	When the CPU is Operating with the Middle-speed On-chip Oscillator Clock (f_{IM})	When the CPU is Operating with the X1 Clock (f_X)	When the CPU is Operating with the External Main System Clock (f_{EX})
Clock output/buzzer output		Operation enabled			
A/D converter					
D/A converter					
Comparator					
Serial array unit					
Serial interface IICA					
Serial interface UARTA					
Remote control signal receiver					
Data transfer controller (DTC)					
SNOOZE mode sequencer (SMS)					
Logic and event link controller (ELCL)		Operation-enabled function blocks can be linked			
Power-on-reset		Operation enabled			
Voltage detection					
External interrupt					
Key interrupt					
Capacitive sensing unit (CTSU2L)**Note**					
CRC operation	High-speed CRC				
	General-purpose CRC	Capable of operation in response to access by the DTC or SMS to obtain data for calculations from the RAM area			
Illicit memory access detection		Capable of operation in response to access by the DTC or SMS			
RAM parity error detection					
RAM guard function					
SFR guard function					
True random number generator		Operation enabled			

Note The accuracy of measurement by the capacitive sensing unit (CTSU2L) depends on the accuracy of the operating clock. When the CTSU2L is to be used, we recommend selecting the high-speed on-chip oscillator clock (f_{IH}) or high-speed system clock (f_{MX}) as the CPU/peripheral hardware clock (f_{CLK}).

Remark 1. Operation stopped: Operation is automatically stopped at the time of switching to the HALT mode.
Operation disabled: Stop operation before switching to the HALT mode.

Remark 2. f_{IH}: High-speed on-chip oscillator clock f_{IL}: Low-speed on-chip oscillator clock
f_{IM}: Middle-speed on-chip oscillator clock f_X: X1 clock
f_{EX}: External main system clock f_{XT}: XT1 clock
f_{EXS}: External subsystem clock

Table 23 - 2 Operating States in HALT Mode When the Subsystem Clock Is Used as the CPU Clock (1/2)

Item \ HALT Mode Setting			When a HALT Instruction is Executed While the CPU is Operating with the Subsystem Clock		
			When the CPU is Operating with the XT1 Clock (f_{XT})	When the CPU is Operating with the External Subsystem Clock (f_{EXS})	When the CPU is Operating with the Low-speed On-chip Oscillator Clock (f_{IL})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{IH}	Operation disabled		
		f_{IM}			
		f_X			
		f_{EX}			
	Subsystem clock	f_{XT}	Operation continues (stopping this clock signal is not possible)	Cannot operate	Operation disabled
		f_{EXS}	Cannot operate	Operation continues (stopping this clock signal is not possible)	Operation disabled
		f_{IL}	Set by the WDSTBYON and WDTON bits of the user option byte (000C0H), and the WUTMMCK0 bit of the subsystem clock supply mode control register (OSMC) WUTMMCK0 = 1: Oscillates (Setting of WUTMMCK0 = 1 and SELLOSC = 1 is prohibited when the subsystem clocks X (f_{SX}) and XR (f_{SXR}) are operating.) WUTMMCK0 = 0, SELLOSC = 0, and WDTON = 0: Stopped WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 1: Oscillates WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 0: Stopped		Operation continues (stopping this clock signal is not possible)
CPU			Operation stopped		
Code flash memory					
Data flash memory					
RAM			Operation stopped (however, operation in response to access by the DTC or SMS is possible)		
Port (latch)			Retains the state before the transition to HALT mode (however, operation in response to access by the DTC, SMS, or ELCL is possible)		
Timer array unit			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operation enabled
RTC			Operation enabled		
32-bit interval timer			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operation enabled
Watchdog timer			See **Section 11 Watchdog Timer (WDT)**.		
Clock output/buzzer output			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operation enabled
A/D converter			Operation disabled		
D/A converter			Retains the state before the transition to HALT mode (however, operation when RTCLPC = 0 and in response to access by the DTC, SMS, or ELCL is possible)		
Comparator			Operation enabled (when the RTCLPC bit is 0, or when the digital filters are not in use)		Operation enabled
Serial array unit			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operation enabled
Serial interface IICA			Operation disabled		
Serial interface UARTA			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operation enabled
Remote control signal receiver			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operation enabled
Data transfer controller (DTC)			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operation enabled

Table 23 - 2 Operating States in HALT Mode When the Subsystem Clock Is Used as the CPU Clock (2/2)

HALT Mode Setting / Item		When a HALT Instruction is Executed While the CPU is Operating with the Subsystem Clock		
		When the CPU is Operating with the XT1 Clock (f_{XT})	When the CPU is Operating with the External Subsystem Clock (f_{EXS})	When the CPU is Operating with the Low-speed On-chip Oscillator Clock (f_{IL})
SNOOZE mode sequencer (SMS)		Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operation enabled
Logic and event link controller (ELCL)		Operation-enabled function blocks can be linked		
Power-on-reset		Operation enabled		
Voltage detection				
External interrupt				
Key interrupt				
Capacitive sensing unit (CTSU2L)		Operation disabled		
CRC operation	High-speed CRC	Operation disabled		
	General-purpose CRC	Capable of operation in response to access by the DTC or SMS to obtain data for calculations from the RAM area		
Illicit memory access detection		Capable of operation in response to access by the DTC or SMS		
RAM parity error detection				
RAM guard function				
SFR guard function				
True random number generator		Operation enabled		

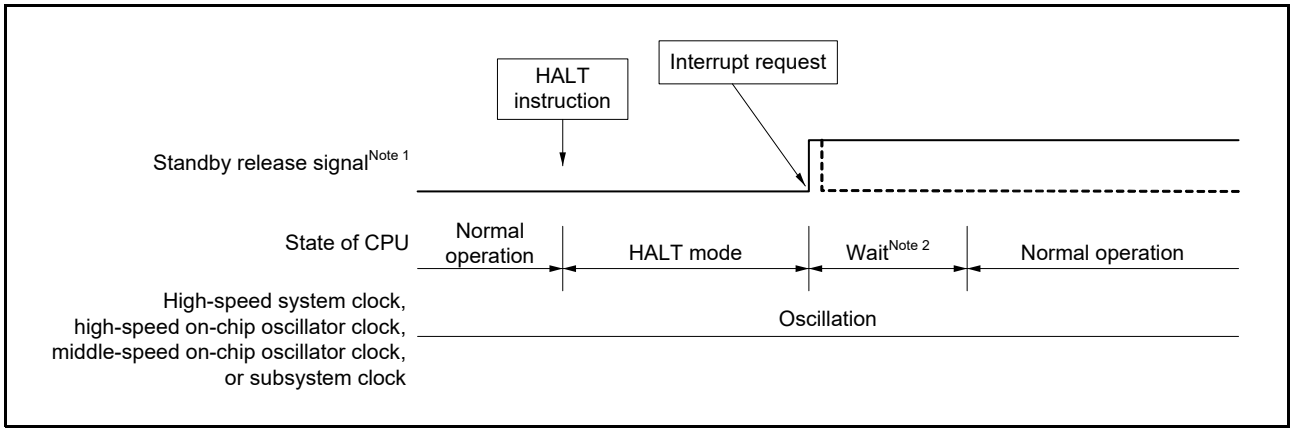
Remark 1. Operation stopped: Operation is automatically stopped at the time of switching to the HALT mode.
Operation disabled: Stop operation before switching to the HALT mode.

Remark 2. f_{IH}: High-speed on-chip oscillator clock f_{IL}: Low-speed on-chip oscillator clock
f_{IM}: Middle-speed on-chip oscillator clock f_X: X1 clock
f_{EX}: External main system clock f_{XT}: XT1 clock
f_{EXS}: External subsystem clock

2. Release from HALT mode
   The following two types of exception release the MCU from the HALT mode.

a) Release by a non-masked interrupt request
   Release from the HALT mode proceeds in response to the generation of a non-masked interrupt request. If acknowledgment of the interrupt is enabled, vectored interrupt servicing proceeds. If acknowledgment of the interrupt is disabled, the instruction at the next address is executed.

Figure 23 - 5 Release from HALT Mode by Generation of an Interrupt Request



Note 1. For details of the standby release signal, see **Figure 21 - 1 Basic Configuration of Interrupt Function**.

Note 2. Wait time for release from the HALT mode

- When vectored interrupt servicing proceeds
  - Main system clock: 15 to 16 clock cycles
  - Subsystem clock (RTCLPC = 0): 10 to 11 clock cycles
  - Subsystem clock (RTCLPC = 1): 11 to 12 clock cycles
- When vectored interrupt servicing does not proceed
  - Main system clock: 9 to 10 clock cycles
  - Subsystem clock (RTCLPC = 0): 4 to 5 clock cycles
  - Subsystem clock (RTCLPC = 1): 5 to 6 clock cycles

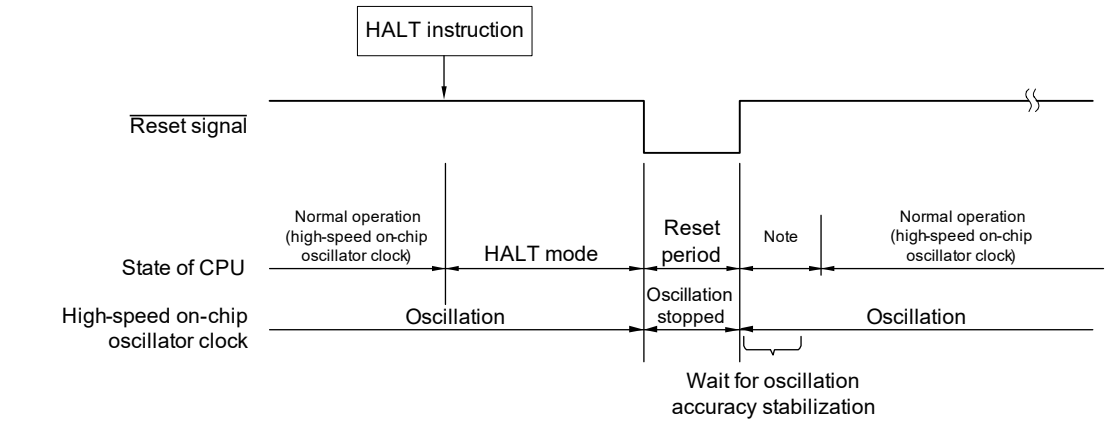
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

b) Release by reset signal generation

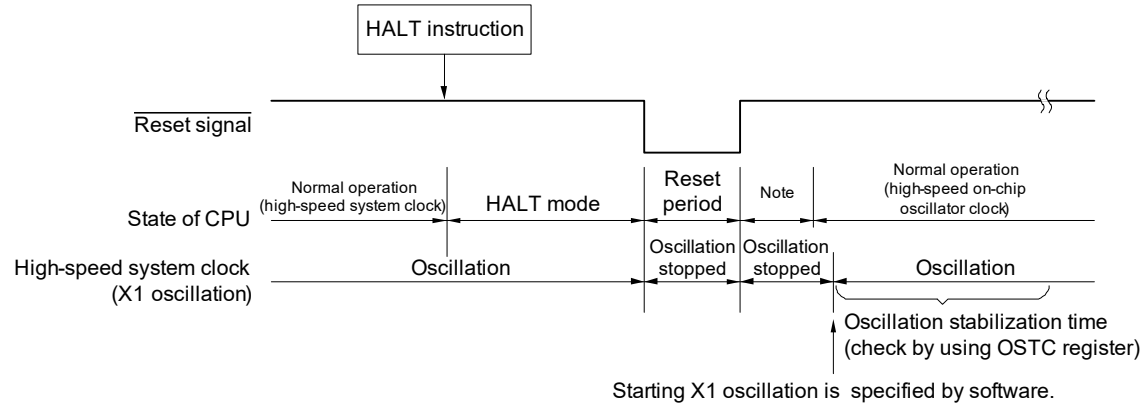
When the reset signal is generated, release from the HALT mode proceeds, and then, as is the case with normal reset operation, the program is executed after branching to the reset vector address.

Figure 23 - 6 Release from HALT Mode by a Reset (1/2)

<1> When high-speed on-chip oscillator clock is used as CPU clock

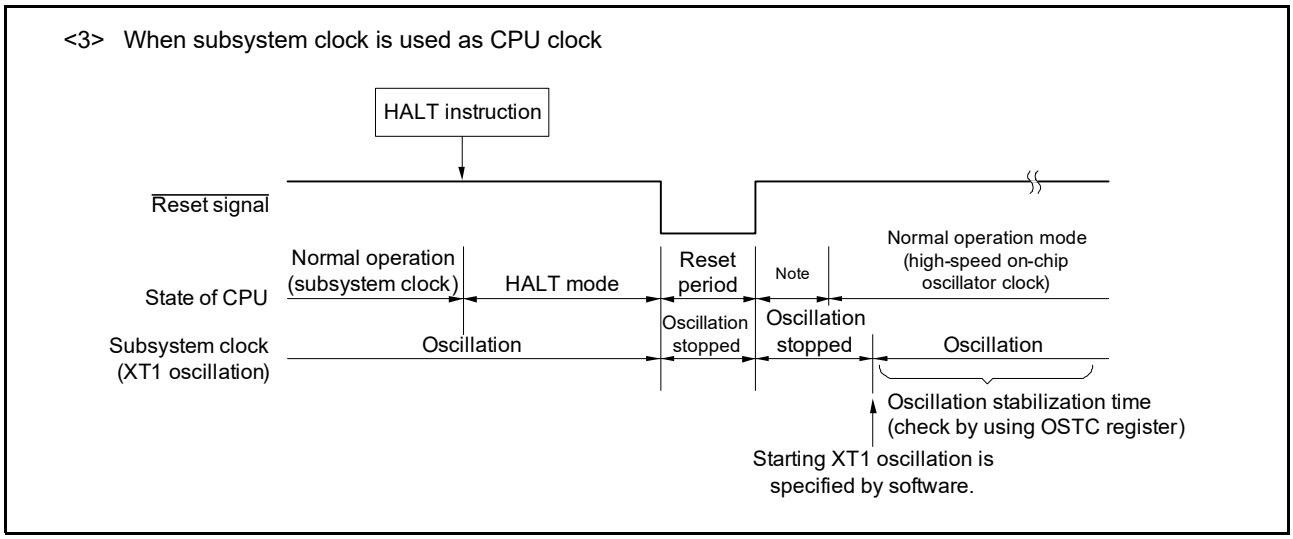


<2> When high-speed system clock is used as CPU clock



Note For the reset processing time, see **Section 24 Reset Function**. For the reset processing time of the power-on-reset circuit (POR) and voltage detectors (LVD0 and LVD1), see **Section 25 Power-on-reset Circuit (POR)**.

Figure 23 - 6 Release from HALT Mode by a Reset (2/2)



Note For the reset processing time, see **Section 24 Reset Function**. For the reset processing time of the power-on-reset circuit (POR) and voltage detectors (LVD0 and LVD1), see **Section 25 Power-on-reset Circuit (POR)**.

23.3.2 STOP mode

1. Setting of and operating states in STOP mode
   The STOP mode is set by executing the STOP instruction. This MCU can be placed in the STOP mode if the CPU clock before entry to the STOP mode is the main system clock.

Caution The interrupt request signal is used for release from the STOP mode, so if the interrupt mask flag is 0 (enabling the interrupt processing) and the interrupt request flag is 1 (the interrupt request signal is being generated), release from the STOP mode immediately proceeds after the transition to the STOP mode if these are the settings when a 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.

Table 23 - 3 shows the operating states in the STOP mode.

Table 23 - 3 Operating States in STOP Mode (1/2)

STOP Mode Setting / Item			When a STOP Instruction is Executed While the CPU is Operating with the Main System Clock			
			When the CPU is Operating with the High-speed On-chip Oscillator Clock (f_{IH})	When the CPU is Operating with the Middle-speed On-chip Oscillator Clock (f_{IM})	When the CPU is Operating with the X1 Clock (f_X)	When the CPU is Operating with the External Main System Clock (f_{EX})
System clock			Clock supply to the CPU is stopped			
	Main system clock	f_{IH}	Stopped			
		f_{IM}	Stopped			
		f_X	Stopped			
		f_{EX}				
	Subsystem clock	f_{XT}	Retains the state before the transition to STOP mode			
		f_{EXS}				
		f_{IL}	Set by the WDSTBYON and WDTON bits of the user option byte (000C0H), and the WUTMMCK0 bit of the subsystem clock supply mode control register (OSMC) WUTMMCK0 = 1 or SELLOSC = 1: Oscillates (Setting of WUTMMCK0 = 1 and SELLOSC = 1 is prohibited when the subsystem clocks X (f_{SX}) and XR (f_{SXR}) are operating.) WUTMMCK0 = 0, SELLOSC = 0, and WDTON = 0: Stopped WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 1: Oscillates WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 0: Stopped			
CPU			Operation stopped			
Code flash memory						
Data flash memory						
RAM						
Port (latch)			Retains the state before the transition to STOP mode			
Timer array unit			Operation stopped			
RTC			Operation enabled			
32-bit interval timer			Capable of operation when f_{SXP} is selected and RTCLPC = 0			
Watchdog timer			See **Section 11 Watchdog Timer (WDT)**.			
Clock output/buzzer output			Capable of operation when f_{SXP} is selected and RTCLPC = 0			
A/D converter			Wakeup operation is enabled (switching to SNOOZE mode)			
D/A converter			Retains the state before the transition to STOP mode			
Comparator			Operation enabled (only when the digital filters are not in use)			
Serial array unit			Wakeup operation is enabled only for CSIp and UARTq (switching to SNOOZE mode) Operation of the types of modules other than CSIp and UARTq is disabled.			
Serial interface IICA			Capable of waking up in response to address matching			
Serial interface UARTA			Capable of operation when f_{SXP} is selected and RTCLPC = 0			

Table 23 - 3 Operating States in STOP Mode (2/2)

Item \ STOP Mode Setting		When a STOP Instruction is Executed While the CPU is Operating with the Main System Clock			
		When the CPU is Operating with the High-speed On-chip Oscillator Clock (f_{IH})	When the CPU is Operating with the Middle-speed On-chip Oscillator Clock (f_{IM})	When the CPU is Operating with the X1 Clock (f_X)	When the CPU is Operating with the External Main System Clock (f_{EX})
Remote control signal receiver		Capable of operation in response to edge detection (switching to SNOOZE mode)			
Data transfer controller (DTC)		DTC activation source receiving operation enabled (switching to SNOOZE mode)			
SNOOZE mode sequencer (SMS)		SMS activation source receiving operation enabled (switching to SNOOZE mode)			
Logic and event link controller (ELCL)		Operation-enabled function blocks can be linked			
Power-on-reset		Operation enabled			
Voltage detection					
External interrupt					
Key interrupt					
Capacitive sensing unit (CTSU2L)Note		CTSU activation source receiving operation enabled (switching to SNOOZE mode)			
CRC operation	High-speed CRC	Operation stopped			
	General-purpose CRC				
Illicit memory access detection					
RAM parity error detection					
RAM guard function					
SFR guard function					
True random number generator					

<R> **Note** The accuracy of measurement by the capacitive sensing unit (CTSU2L) depends on the accuracy of the operating clock. When the CTSU2L is to be used, we recommend selecting the high-speed on-chip oscillator clock (f_{IH}) or high-speed system clock (f_{MX}) as the CPU/peripheral hardware clock (f_{CLK}).

Remark 1. Operation stopped: Operation is automatically stopped at the time of switching to the STOP mode.
Operation disabled: Stop operation before switching to the STOP mode.

Remark 2. f_{IH}: High-speed on-chip oscillator clock f_{IL}: Low-speed on-chip oscillator clock
f_{IM}: Middle-speed on-chip oscillator clock f_X: X1 clock
f_{EX}: External main system clock f_{XT}: XT1 clock
f_{EXS}: External subsystem clock f_{SXP}: Low-speed peripheral clock

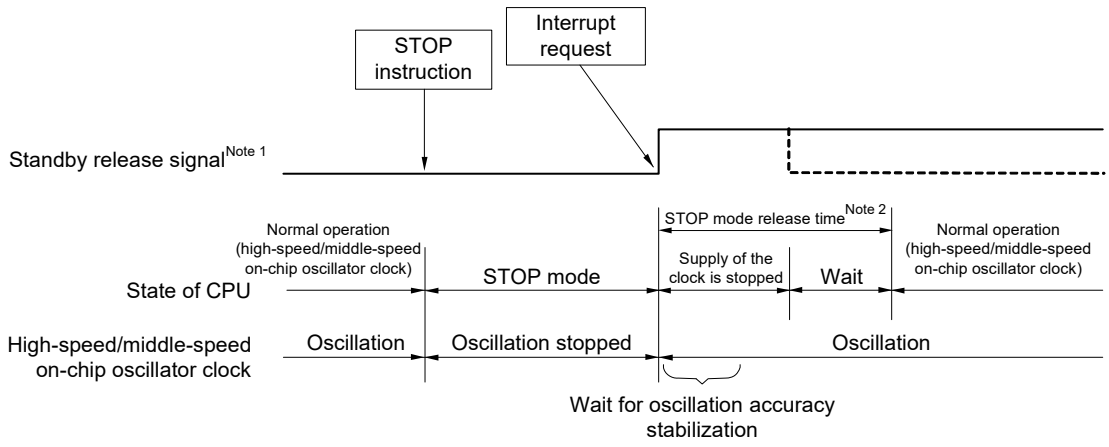
Remark 3. 30- to 64-pin R7F100G products: p = 00, q = 0
80- to 128-pin R7F100G products: p = 00, 20; q = 0, 2
32-pin R7F104G product: p = 00, q = 0
44- and 48-pin R7F104G products: p = 00; q = 0, 2

2. Release from STOP mode
   The following two types of exception release the MCU from the STOP mode.

a) Release by a non-masked interrupt request
   Release from the STOP mode proceeds in response to the generation of a non-masked interrupt request. After the oscillation stabilization time has elapsed, if acknowledgment of the interrupt is enabled, vectored interrupt servicing proceeds. If acknowledgment of the interrupt is disabled, the instruction at the next address is executed.

Figure 23 - 7 Release from STOP Mode by Generation of an Interrupt Request (1/3)

<1> When high-speed/middle-speed on-chip oscillator clock is used as CPU clock



Note 1. For details of the standby release signal, see **Figure 21 - 1 Basic Configuration of Interrupt Function**.

Note 2. STOP mode release time

- Supply of the clock is stopped:
  - For the high-speed on-chip oscillator clock: 3.9 to 5.2 μs + 3 to 4 clock cycles (WKUPMD.FWKUP = 0: Starting of the high-speed on-chip oscillator is at normal speed.)
    0.6 to 0.8 μs + 3 to 4 clock cycles (WKUPMD.FWKUP = 1: Starting of the high-speed on-chip oscillator is at high speed.)
    The accuracy of the high-speed on-chip oscillator frequency depends on whether starting of the high-speed on-chip oscillator is at normal speed or at high speed. See **Section 37 Electrical Characteristics**.
  - For the middle-speed on-chip oscillator clock: 1.5 to 2.5 μs + 3 to 4 clock cycles
- Wait time common to the high-speed and middle-speed on-chip oscillator clocks:
  - When vectored interrupt servicing proceeds: 7 clock cycles
  - When vectored interrupt servicing does not proceed: 1 clock cycle

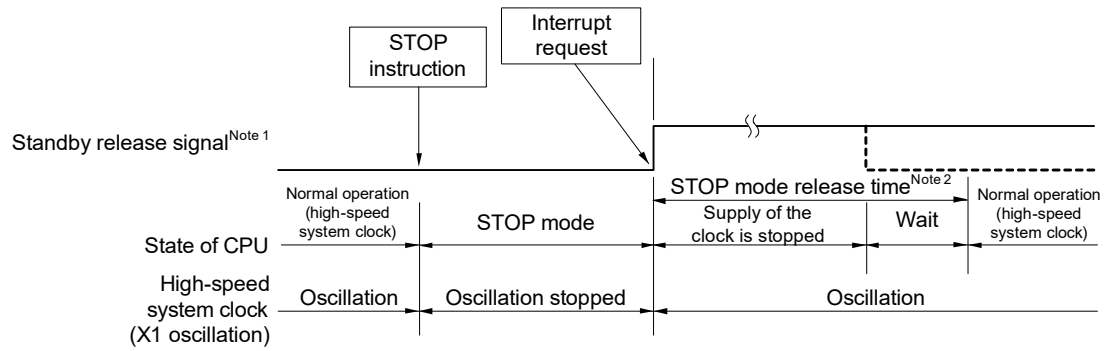
Caution To shorten oscillation stabilization time after release from the STOP mode when the CPU operates with the high-speed system clock (X1 oscillation), temporarily switch the CPU clock to the high-speed on-chip oscillator clock before the execution of the STOP instruction.

Remark 1. The clock supply stop time varies depending on the temperature conditions and STOP mode period.

Remark 2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

Figure 23 - 7 Release from STOP Mode by Generation of an Interrupt Request (2/3)

<2> When high-speed system clock (X1 oscillation) is used as CPU clock



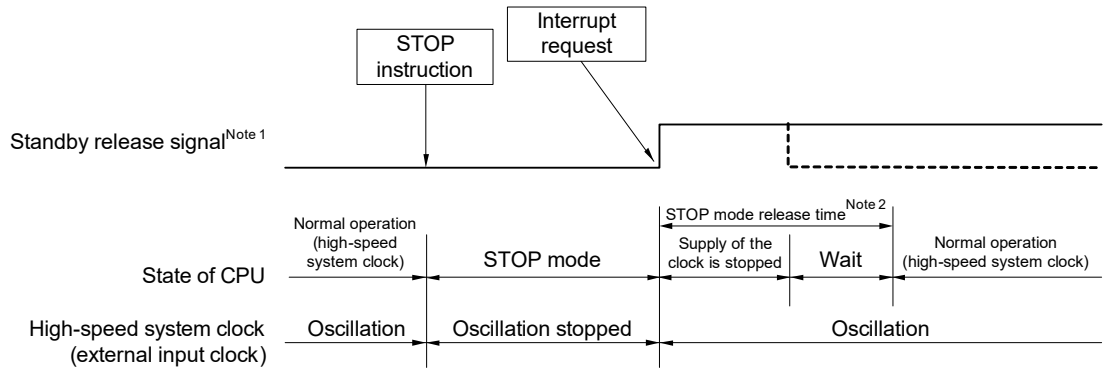
Note 1. For details of the standby release signal, see **Figure 21 - 1 Basic Configuration of Interrupt Function**.

Note 2. STOP mode release time

- Supply of the clock is stopped:
  Oscillation stabilization time (set by the OSTS register) + 3 to 4 clock cycles
- Wait:
  - When vectored interrupt servicing proceeds: 7 clock cycles
  - When vectored interrupt servicing does not proceed: 1 clock cycle

Figure 23 - 7 Release from STOP Mode by Generation of an Interrupt Request (3/3)

<3> When high-speed system clock (external clock input) is used as CPU clock



Note 1. For details of the standby release signal, see **Figure 21 - 1 Basic Configuration of Interrupt Function**.

Note 2. STOP mode release time

- Supply of the clock is stopped:
  50 to 51 cycles of the external clock
- Wait:
  - When vectored interrupt servicing proceeds: 7 clock cycles
  - When vectored interrupt servicing does not proceed: 1 clock cycle

Caution To reduce the oscillation stabilization time after release from the STOP mode while the CPU operates with the high-speed system clock (X1 oscillation), switch the CPU clock to the high-speed on-chip oscillator clock temporarily before executing the STOP instruction.

Remark 1. The clock supply stop time varies depending on the temperature conditions and STOP mode period.

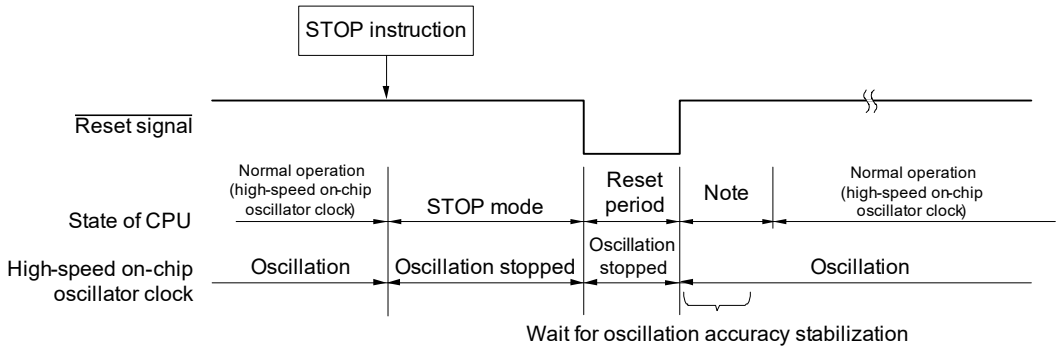
Remark 2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

b) Release by reset signal generation

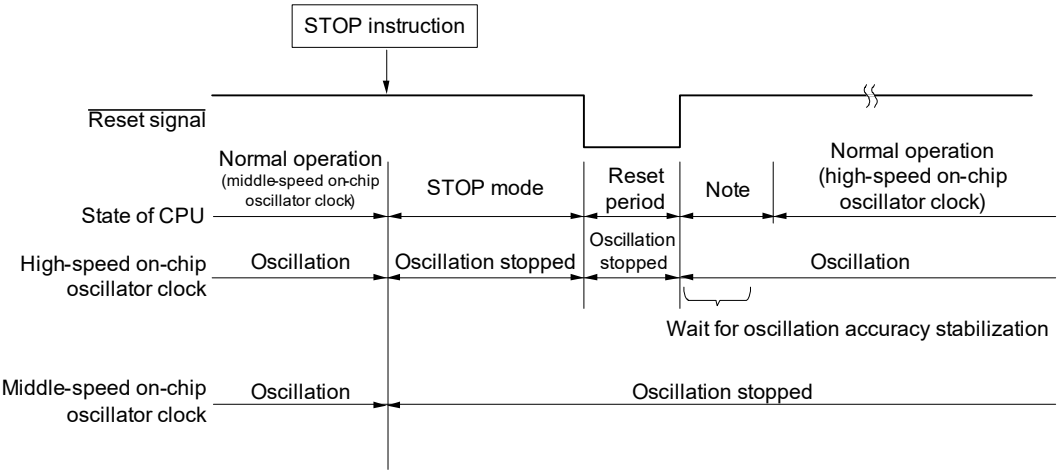
When the reset signal is generated, release from the STOP mode proceeds, and then, as is the case with normal reset operation, the program is executed after branching to the reset vector address.

Figure 23 - 8 Release from STOP Mode by a Reset

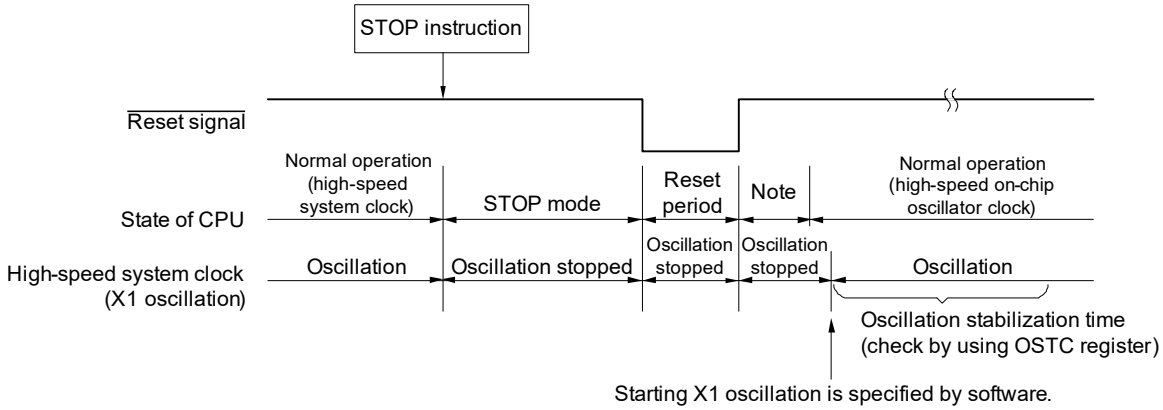
<1> When high-speed on-chip oscillator clock is used as CPU clock



<2> When middle-speed on-chip oscillator clock is used as CPU clock



<3> When high-speed system clock is used as CPU clock



Note For the reset processing time, see **Section 24 Reset Function**. For the reset processing time of the power-on-reset circuit (POR) and voltage detectors (LVD0 and LVD1), see **Section 25 Power-on-reset Circuit (POR)**.

23.3.3 SNOOZE mode

1. Setting of and operating states in SNOOZE mode
   This MCU can be placed in SNOOZE mode, in which operation of the following peripheral modules is selectable. For details, see the sections on the individual modules.
   - **Section 12 A/D Converter (ADC)**
   - **Section 15 Serial Array Unit (SAU)**
   - **Section 18 Remote Control Signal Receiver (REMC)**
   - **Section 19 Data Transfer Controller (DTC)**
   - **Section 29 SNOOZE Mode Sequencer (SMS)**
   - **Section 30 Capacitive Sensing Unit (CTSU2L)**

   This MCU can also be placed in SNOOZE mode if the CPU clock before entry to SNOOZE mode is the high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock.

   For transitions to SNOOZE mode, the following intervals of waiting are inserted.

   a) Transition time from STOP mode to SNOOZE mode:
      - For the high-speed on-chip oscillator clock: 3.9 to 5.2 μs (WKUPMD.FWKUP = 0: Starting of the high-speed on-chip oscillator is at normal speed.)
        0.6 to 0.8 μs (WKUPMD.FWKUP = 1: Starting of the high-speed on-chip oscillator is at high speed.)
        The accuracy of the high-speed on-chip oscillator frequency depends on whether starting of the high-speed on-chip oscillator is at normal speed or at high speed. See **Section 37 Electrical Characteristics**.
      - For the middle-speed on-chip oscillator clock[Note]: 1.3 to 2.5 μs

   b) Transition time from SNOOZE mode to normal operation:
      - For the high-speed on-chip oscillator clock:
        – When vectored interrupt servicing proceeds:
          0.3 to 0.4 μs + 10 to 11 clock cycles
        – When vectored interrupt servicing does not proceed:
          0.3 to 0.4 μs + 4 to 5 clock cycles
      - For the middle-speed on-chip oscillator clock[Note]:
        – When vectored interrupt servicing proceeds:
          0.6 to 1.2 μs + 10 to 11 clock cycles
        – When vectored interrupt servicing does not proceed:
          0.6 to 1.2 μs + 4 to 5 clock cycles

Note This is selected when the setting of the middle-speed on-chip oscillator trimming register (MIOTRM) is its initial value.

Caution The transition time from STOP mode to SNOOZE mode varies depending on the temperature conditions and the STOP mode period.

Table 23 - 4 shows the operating states in the SNOOZE mode.

Table 23 - 4 Operating States in SNOOZE Mode (1/2)

Item \ STOP Mode Setting			Generation of the Source Conditions Which Lead to Transitions to SNOOZE Mode during STOP Mode	
			When the CPU is Operating with the High-speed On-chip Oscillator Clock (f_{IH})	When the CPU is Operating with the Middle-speed On-chip Oscillator Clock (f_{IM})
System clock			Clock supply to the CPU is stopped	
	Main system clock	f_{IH}	Operation started	Stopped
		f_{IM}	Stopped	Operation started
		f_X	Stopped	
		f_{EX}		
	Subsystem clock	f_{XT}	Operation enabled	
		f_{EXS}		
		f_{IL}	Set by the WDSTBYON and WDTON bits of the user option byte (000C0H), and the WUTMMCK0 bit of the subsystem clock supply mode control register (OSMC) WUTMMCK0 = 1 or SELLOSC = 1: Oscillates (Setting of WUTMMCK0 = 1 and SELLOSC = 1 is prohibited when the subsystem clocks X (f_{SX}) and XR (f_{SXR}) are operating.) WUTMMCK0 = 0, SELLOSC = 0, and WDTON = 0: Stopped WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 1: Oscillates WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 0: Stopped	
CPU			Operation stopped	
Code flash memory				
Data flash memory				
RAM			Operation stopped (however, operation in response to access by the DTC or SMS is possible)	
Port (latch)			Retains the state before the transition to SNOOZE mode (however, operation in response to access by the DTC, SMS, or ELCL is possible)	
Timer array unit			Capable of operation in response to access by the DTC, SMS or ELCL	
RTC			Operation enabled	
32-bit interval timer			Capable of operation when f_{SXP} is selected and RTCLPC = 0	
Watchdog timer			See **Section 11 Watchdog Timer (WDT)**.	
Clock output/buzzer output			Capable of operation when f_{SXP} is selected and RTCLPC = 0	
A/D converter			Operation enabled	
D/A converter			Retains the state before the transition to SNOOZE mode (however, operation when RTCLPC = 0 and in response to access by the DTC, SMS, or ELCL is possible)	
Comparator			Operation enabled (only when the digital filters are not in use)	
Serial array unit			Only CSIp and UARTq are capable of operation. Operation of the types of modules other than CSIp and UARTq is disabled.	
Serial interface IICA			Capable of waking up in response to address matching	
Serial interface UARTA			Capable of operation when f_{SXP} is selected and RTCLPC = 0	
Remote control signal receiver			Operation enabled	
Data transfer controller (DTC)			Operation enabled	
SNOOZE mode sequencer (SMS)			Operation enabled	
Logic and event link controller (ELCL)			Operation-enabled function blocks can be linked	
Power-on-reset			Operation enabled	
Voltage detection				
External interrupt				
Key interrupt				
Capacitive sensing unit (CTSU2L)**Note**			Operation enabled	

Table 23 - 4 Operating States in SNOOZE Mode (2/2)

Item \ STOP Mode Setting		Generation of the Source Conditions Which Lead to Transitions to SNOOZE Mode during STOP Mode	
		When the CPU is Operating with the High-speed On-chip Oscillator Clock (f_{IH})	When the CPU is Operating with the Middle-speed On-chip Oscillator Clock (f_{IM})
CRC operation	High-speed CRC	Operation stopped	
	General-purpose CRC	Capable of operation in response to access by the DTC or SMS to obtain data for calculations from the RAM area	
Illicit memory access detection		Capable of operation in response to access by the DTC or SMS	
RAM parity error detection			
RAM guard function			
SFR guard function			
True random number generator			

Note The accuracy of measurement by the capacitive sensing unit (CTSU2L) depends on the accuracy of the operating clock. When the CTSU2L is to be used, we recommend selecting the high-speed on-chip oscillator clock (f_{IH}) or high-speed system clock (f_{MX}) as the CPU/peripheral hardware clock (f_{CLK}).

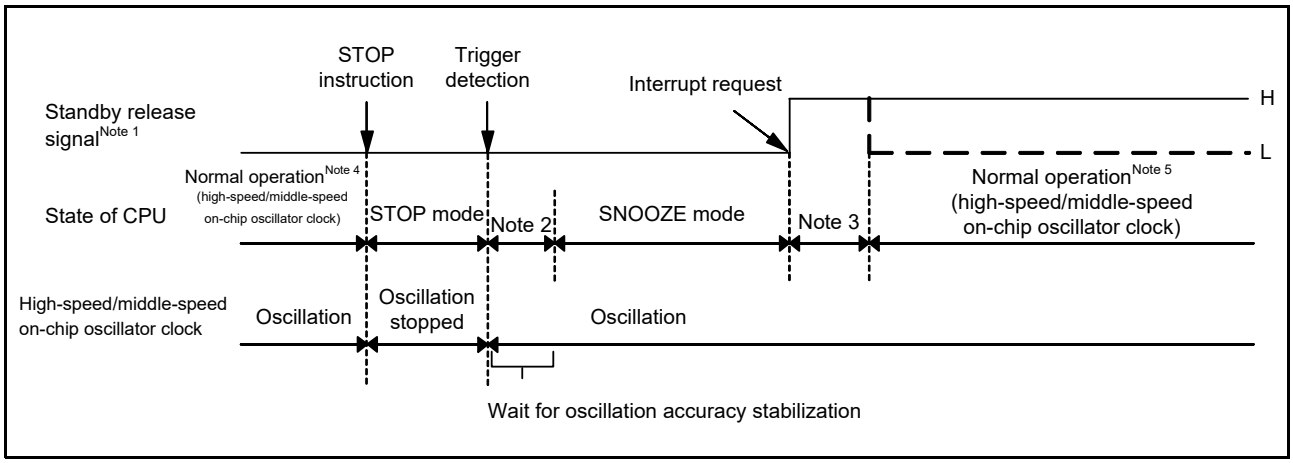
Remark 1. Operation stopped: Operation is automatically stopped at the time of switching to the STOP mode.
Operation disabled: Stop operation before switching to the STOP mode.

Remark 2. f_{IH}: High-speed on-chip oscillator clock
f_{IL}: Low-speed on-chip oscillator clock
f_{IM}: Middle-speed on-chip oscillator clock
f_X: X1 clock
f_{EX}: External main system clock
f_{XT}: XT1 clock
f_{EXS}: External subsystem clock
f_{SX}: Subsystem clock X
f_{SXR}: Subsystem clock XR
f_{SXP}: Low-speed peripheral clock

Remark 3. 30- to 64-pin R7F100G products: p = 00, q = 0
80- to 128-pin R7F100G products: p = 00, 20; q = 0, 2
32-pin R7F104G product: p = 00, q = 0
44- and 48-pin R7F104G products: p = 00; q = 0, 2

2. Timing diagram when the interrupt request signal is generated in the SNOOZE mode

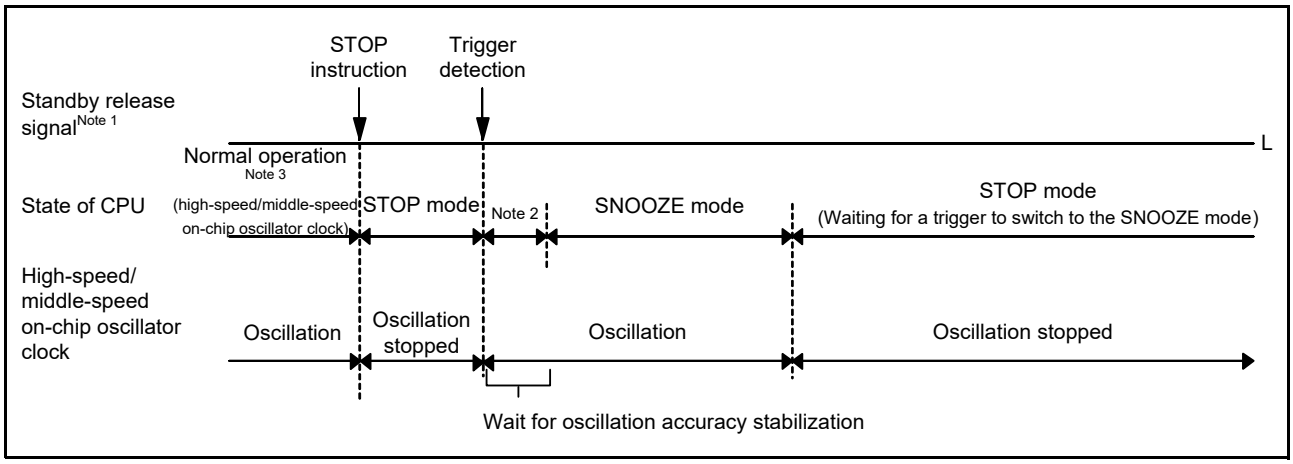
Figure 23 - 9 When the Interrupt Request Signal is Generated in the SNOOZE Mode



Note 1. For details of the standby release signal, see **Figure 21 - 1 Basic Configuration of Interrupt Function**.
Note 2. Transition time from STOP mode to SNOOZE mode
Note 3. Transition time from SNOOZE mode to normal operation
Note 4. Enable the SNOOZE mode immediately before switching to the STOP mode.
Note 5. Be sure to release the SNOOZE mode immediately after return to the normal operation.

3. Timing diagram when the interrupt request signal is not generated in the SNOOZE mode

Figure 23 - 10 When the Interrupt Request Signal is Not Generated in the SNOOZE Mode



Note 1. For details of the standby release signal, see **Figure 21 - 1 Basic Configuration of Interrupt Function**.
Note 2. Transition time from STOP mode to SNOOZE mode
Note 3. Enable the SNOOZE mode immediately before switching to the STOP mode.

Remark For details on the SNOOZE mode, see the following sections.
- **Section 12 A/D Converter (ADC)**
- **Section 15 Serial Array Unit (SAU)**
- **Section 18 Remote Control Signal Receiver (REMC)**
- **Section 19 Data Transfer Controller (DTC)**
- **Section 29 SNOOZE Mode Sequencer (SMS)**
- **Section 30 Capacitive Sensing Unit (CTSU2L)**

Section 24 Reset Function

A reset is triggered by any of the following events.

1. External reset input via the $\overline{\text{RESET}}$ pin
2. Internal reset due to detection of a program malfunction by the watchdog timer
3. Internal reset by comparison of supply voltage and detection voltage of the power-on-reset circuit (POR)
4. Internal reset by comparison of supply voltage and detection voltage of the voltage detectors (LVD0 and LVD1)
5. Internal reset due to execution of an illegal instruction**Note**
6. Internal reset due to a RAM parity error
7. Internal reset due to illegal-memory access

External and internal resets start program execution from the address at 00000H and 00001H when the reset signal is generated. A reset is applied when a low level is input to the $\overline{\text{RESET}}$ pin or the watchdog timer overflows, or on voltage detection by the power-on reset circuit, LVD0, or LVD1, execution of illegal instruction**Note**, generation of a RAM parity error, or illegal-memory access, and each module is set to the state shown in **Table 24 - 1 Operating States during a Reset**.

Note In normal operation, executing the instruction code FFH triggers an internal reset, but this is not the case during emulation by the on-chip debugging emulator.

Caution 1. For an external reset, input a low level for at least 10 µs to the $\overline{\text{RESET}}$ pin. To perform an external reset upon power application, input a low level to the $\overline{\text{RESET}}$ pin, turn power on, continue to input a low level to the pin for at least 10 µs within the operating voltage range shown in 37.4 AC Characteristics, and then input a high level to the pin.

Caution 2. During generation of a reset signal, the X1 clock, high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock, and low-speed on-chip oscillator clock stop oscillating. External main system clock input becomes invalid. The XT1 clock and external subsystem clock only stop oscillating or their inputs become invalid in the POR state.

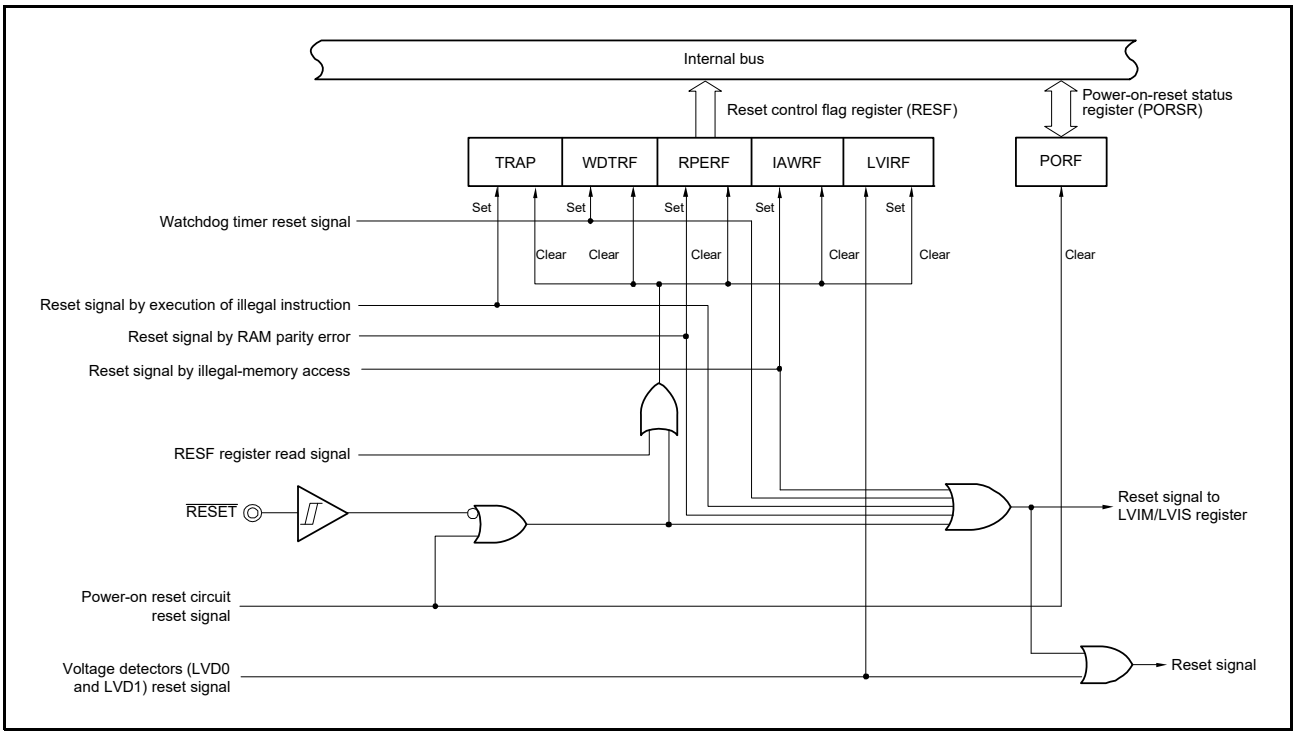
Caution 3. The port pins become the following state because SFRs and 2nd SFRs are initialized after a reset.

- **P40: High-impedance during the external reset period or reset period by the POR. High level during other types of reset and after the reset is released (connected to the on-chip pull-up resistor).**
- **P130: Low-level output during a reset and after the reset is released.**
- **Ports other than P40 and P130: High-impedance during a reset and after the reset is released.**

The following registers are only initialized by a POR reset.

- **RTC-related registers**
- **EXCLKS, OSCSELS, XTSEL, AMPHS1, and AMPHS0 bits of the CMC register**

Figure 24 - 1 Block Diagram of Reset Function

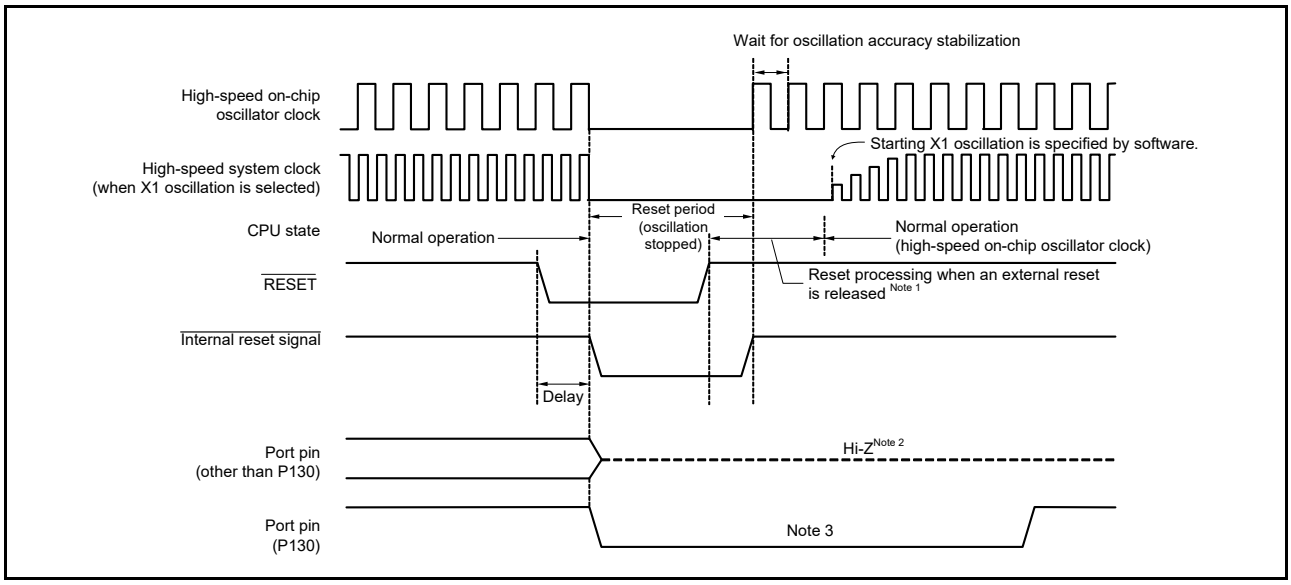


Caution An internal reset triggered by LVD0 does not reset LVD0 itself.

24.1 Timing of Reset Operation

This MCU is reset by input of the low level on the $\overline{\text{RESET}}$ pin and released from the reset state by input of the high level on the $\overline{\text{RESET}}$ pin. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts.

Figure 24 - 2 Timing of Reset by $\overline{\text{RESET}}$ Input



Note 1. Reset times (times for release from the external reset state)
The first external reset following release from the POR state:
- When the LVD is in use: 0.506 ms (typ.), 0.694 ms (max.)
- When the LVD is not in use: 0.201 ms (typ.), 0.335 ms (max.)

The second or subsequent external reset following release from the POR state:
- When the LVD is in use: 0.476 ms (typ.), 0.616 ms (max.)
- When the LVD is not in use: 0.170 ms (typ.), 0.257 ms (max.)

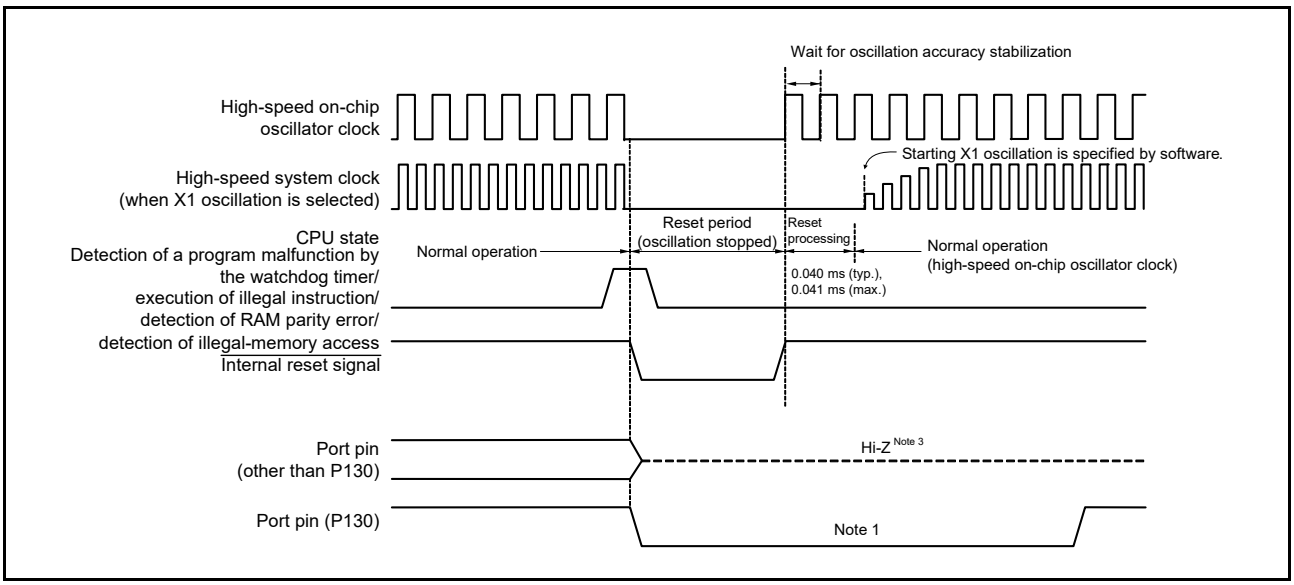
After power is supplied, the following voltage stabilization waiting time is required before reset processing starts after release from the external reset state.
- 4.0 ms (typ.), 9.9 ms (max.)

Note 2. The state of the P40 pin is as follows.
- High-impedance during the external reset period or reset period by the POR.
- High level during other types of reset and after the reset is released (connected to the on-chip pull-up resistor).

Note 3. The P130 pin outputs a low-level signal following the application of a reset. If this pin was in use as a high-level output before the application of a reset, the output signal from the P130 pin can effectively be used as the reset signal for an external device that has an active-low reset signal. To de-assert the reset signal to the external device, set the P130 pin for high-level output by software.

Release from the reset state proceeds automatically in the case of a reset due to detection of a program malfunction by the watchdog timer, execution of an illegal instruction, detection of a RAM parity error, or detection of illegal-memory access. After reset processing, program execution starts with the high-speed on-chip oscillator clock as the operating clock.

Figure 24 - 3 Timing of Reset Due to Detection of a Program Malfunction by the Watchdog Timer, Execution of Illegal Instruction, Detection of RAM Parity Error, or Detection of Illegal-memory Access



Note 1. The state of the P40 pin is as follows.
- High-impedance during the external reset period or reset period by the POR.
- High level during other types of reset and after the reset is released (connected to the on-chip pull-up resistor).

Note 2. The P130 pin outputs a low-level signal following the application of a reset. If this pin was in use as a high-level output before the application of a reset, the output signal from the P130 pin can effectively be used as the reset signal for an external device that has an active-low reset signal. To de-assert the reset signal to the external device, set the P130 pin for high-level output by software.

A reset from the power-on reset circuit or by LVD0 voltage detection is released when $V_{DD} \geq V_{POR}$ or $V_{DD} \geq V_{LVD0}$ after the reset. After reset processing, execution of the program starts with the high-speed on-chip oscillator clock as the operating clock. For details, see **Section 25 Power-on-reset Circuit (POR)** or **Section 26 Voltage Detector (LVD)**.

Remark V_{POR}: POR power supply rise detection voltage
V_{LVD0}: LVD0 detection voltage

Table 24 - 1 Operating States during a Reset (1/2)

Item			Operating State during a Reset
System clock			Clock supply to the CPU is stopped.
	Main system clock	f_{IH}	Operation stopped
		f_{IM}	
		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 enabled (operation stops in the POR reset state, the XT1 and XT2 pins are input port mode)
		f_{EXS}	Operation enabled (operation stops in the POR reset state, the EXCLKS pin is input port mode)
		f_{IL}	Operation stopped
CPU			Operation stopped
Code flash memory			Operation stopped
Data flash memory			Operation stopped
RAM			Operation stopped
Port (latch)			High-impedance**Note**
Timer array unit			Operation stopped
Realtime clock			Resets other than the POR reset: Operation enabled POR reset: Only the values of calendar-related registers are retained.
32-bit interval timer			Operation stopped
Watchdog timer			
Clock output/buzzer output			
A/D converter			
D/A converter			
Comparator			
Serial array unit			
Serial interface IICA			
Serial interface UARTA			
Remote control signal receiver			
Data transfer controller (DTC)			
SNOOZE mode sequencer (SMS)			
Logic and event link controller (ELCL)			
Power-on-reset			Detection possible
Voltage detection			LVD0: Operation is enabled following an LVD0 reset but is stopped following other types of reset. LVD1: Operation stopped

Table 24 - 1 Operating States during a Reset (2/2)

Item		Operating State during a Reset
External interrupt		Operation stopped
Key interrupt		
Capacitive sensing unit (CTSU2L)		
CRC operation	High-speed CRC	
	General-purpose CRC	
Illegal-memory access detection		
RAM parity error detection		
RAM guard function		
SFR guard function		

Note P40 and P130 become the following states.

- P40: High-impedance during the external reset period or reset period by the POR. High level during other types of reset (connected to the on-chip pull-up resistor).
- P130: Low-level output during the reset period

Remark f_{IH}: High-speed on-chip oscillator clock
f_{IM}: Middle-speed on-chip oscillator clock
f_{XT}: XT1 oscillation clock
f_{IL}: Low-speed on-chip oscillator clock
f_X: X1 oscillation clock
f_{EX}: External main system clock
f_{EXS}: External subsystem clock

Table 24 - 2 States of the Hardware Blocks after a Reset is Released

Hardware		State after a Reset is Released
Program counter (PC)		The contents of the reset vector table (00000H, 00001H) are set.**Note**
Stack pointer (SP)		Undefined
Program status word (PSW)		06H
RAM	Data memory	Undefined
	General-purpose registers	Undefined

Note The contents of the PC are undefined during a reset and until the clock oscillation becomes stable after the reset is released.

Remark For the states of the special function registers (SFRs) after a reset is released, see **3.1.4 Special function register (SFR) area** and **3.1.5 Extended special function register (2nd SFR: 2nd special function register) area**.

24.2 Registers for Controlling the Reset Function

The following registers are used to control the reset function.

- Reset control flag register (RESF)
- Power-on-reset status register (PORSR)
- Peripheral reset control register 0 (PRR0)
- Peripheral reset control register 1 (PRR1)

24.2.1 Reset control flag register (RESF)

This MCU has multiple internal reset sources. The RESF register indicates which source has generated the reset request. The RESF register can be read by an 8-bit memory manipulation instruction. The flags TRAP, WDTRF, RPERF, IAWRF, and LVIRF are automatically cleared by any of the following event.

- Reset input via the $\overline{\text{RESET}}$ pin
- Reset by the power-on-reset circuit (POR)
- The RESF register is accessed.

Figure 24 - 4 Format of Reset Control Flag Register (RESF)

Address: FFFA8H
After reset: Undefined**Note 1**
R/W: R

Symbol	7	6	5	4	3	2	1	0
RESF	TRAP	0	0	WDTRF	0	RPERF	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.

WDTRF	Internal reset request by watchdog timer (WDT)
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

RPERF	Internal reset request by RAM parity error
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request 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 or LVD1)
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

Note 1. The value after reset varies depending on the reset source. See **Table 24 - 3**.

(Note and Cautions continue on the next page.)

Note 2. In normal operation, executing the instruction code FFH triggers an internal reset, but this is not the case during emulation by the on-chip debugging emulator.

Caution 1. Do not read data by a 1-bit memory manipulation instruction.

Caution 2. While RAM parity error resets are enabled (RPECTL.RPERDIS = 0), be sure to initialize a RAM area where data access is to proceed, and a RAM area and an additional 10 bytes beyond it when instructions are to be fetched from the RAM area, respectively. Reset generation enables RAM parity error resets (RPERDIS = 0). For details, see 27.3.4 RAM parity error detection.

Table 24 - 3 shows the state of the RESF register when a reset request is generated.

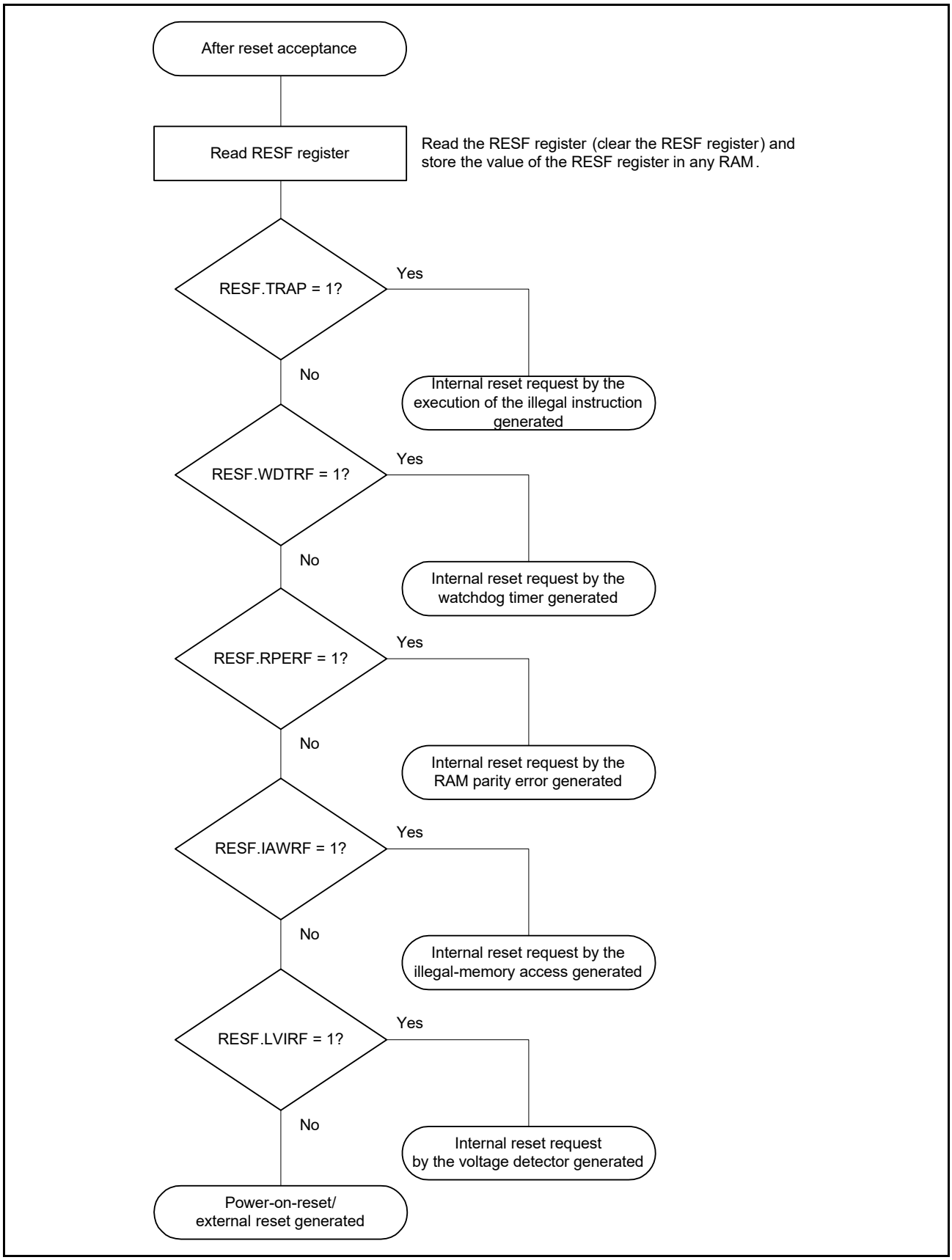
Table 24 - 3 State of the RESF Register when Reset Request is Generated

Reset Source / Flag	$\overline{\text{RESET}}$ Input	Reset by POR	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by RAM Parity Error	Reset by Illegal-memory Access	Reset by LVD0 or LVD1
TRAP	Cleared to 0	Cleared to 0	Set to 1	Retained	Retained	Retained	Retained
WDTRF			Retained	Set to 1			
RPERF				Retained	Set to 1		
IAWRF					Retained	Set to 1	
LVIRF						Retained	Set to 1

The RESF register is automatically cleared after it is read by using an 8-bit memory manipulation instruction.

Figure 24 - 5 shows an example of procedure for checking a reset source.

Figure 24 - 5 Example of Procedure for Checking a Reset Source



The above flow is an example of the procedure for checking a reset source.

24.2.2 Power-on-reset status register (PORSR)

The PORSR register is used to check the occurrence of a power-on reset. Only writing 1 to the PORF bit of this register is effective. Writing 0 to the bit has no effect. Write 1 to the PORF bit in advance to enable checking of the occurrence of a power-on reset. The PORSR register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a power-on reset is 00H.

Caution 1. The PORSR register is reset only by a power-on reset; it retains the value when a reset caused by another source occurs.

Caution 2. The PORF bit being set to 1 guarantees that a power-on reset has not occurred, but does not guarantee that the values in RAM have been retained.

Figure 24 - 6 Format of Power-on-reset Status Register (PORSR)

Address: F00F9H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	<0>
PORSR	0	0	0	0	0	0	0	PORF

PORF	Checking occurrence of power-on reset
0	A value 1 has not been written, or a power-on reset has occurred.
1	No power-on reset has occurred.

24.2.3 Peripheral reset control register 0 (PRR0)

The PRR0 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. The PRR0 register can be set by a 1-bit or 8-bit memory manipulation instruction The value of this register following a reset is 00H.

Figure 24 - 7 Format of Peripheral Reset Control Register 0 (PRR0)

Address: F00F1H
After reset: 00H
R/W: R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PRR0	0	IICA1RES **Note 1**	ADCRES	IICA0RES **Note 2**	SAU1RES	SAU0RES	TAU1RES **Note 3**	TAU0RES

IICAnRES	Control resetting of the serial interface IICAn
0	The serial interface IICAn is released from the reset state.
1	The serial interface IICAn is in the reset state. • The SFRs for use with the serial interface IICAn are initialized.

ADCRES	Control resetting of the A/D converter
0	The A/D converter is released from the reset state.
1	The A/D converter is in the reset state. • The SFRs for use with the A/D converter are initialized.

SAUnRES	Control resetting of serial array unit n
0	Serial array unit n is released from the reset state.
1	Serial array unit n is in the reset state. • The SFRs for use with serial array unit n are initialized.

TAUnRES	Control resetting of timer array unit n
0	Timer array unit n is released from the reset state.
1	Timer array unit n is in the reset state. • The SFRs for use with timer array unit n are initialized.

Note 1. This bit is only present in the 44- to 128-pin R7F100G products.
Note 2. This bit is only present in the 30- to 128-pin R7F100G and 48-pin R7F104G products.
Note 3. This bit is only present in the 80- to 128-pin R7F100G and 44- and 48-pin R7F104G products.

Caution 1. Be sure to leave bit 7 set to 0.
Caution 2. Be sure to leave the following bits set to 0.
Bits 6 and 1 in the 30- to 40-pin R7F100G products
Bit 1 in the 44- to 64-pin R7F100G products
Bits 6, 4, and 1 in the 32-pin R7F104G product
Bits 6 and 4 in the 44-pin R7F104G product
Bit 6 in the 48-pin R7F104G product

Remark n = 0, 1

24.2.4 Peripheral reset control register 1 (PRR1)

The PRR1 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. The PRR1 register can be set by a 1-bit or 8-bit memory manipulation instruction The value of this register following a reset is 00H.

Figure 24 - 8 Format of Peripheral Reset Control Register 1 (PRR1)

Address: F00FBH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	3	2	<1>	<0>
PRR1	DACRES	SMSRES	CMPRES	TML32RES	0	0	REMCRES **Note**	CTSURES

DACRES	Control resetting of the D/A converter
0	The D/A converter is released from the reset state.
1	The D/A converter is in the reset state. • The SFRs for use with the D/A converter are initialized.

SMSRES	Control resetting of the SNOOZE mode sequencer
0	The SNOOZE mode sequencer is released from the reset state.
1	The SNOOZE mode sequencer is in the reset state. • The SFRs for use with the SNOOZE mode sequencer are initialized.

CMPRES	Control resetting of the comparator
0	The comparator is released from the reset state.
1	The comparator is in the reset state. • The SFRs for use with the comparator are initialized.

TML32RES	Control resetting of the 32-bit interval timer
0	The 32-bit interval timer is released from the reset state.
1	The 32-bit interval timer is in the reset state. • The SFRs for use with the 32-bit interval timer are initialized.

REMCRES	Control resetting of the remote control signal receiver
0	The remote control signal receiver is released from the reset state.
1	The remote control signal receiver is in the reset state. • The SFRs for use with the remote control signal receiver are initialized.

CTSURES	Control resetting of the capacitive sensor unit
0	The capacitive sensor unit is released from the reset state.
1	The capacitive sensor unit is in the reset state. • The SFRs for use with the CTSU are initialized.

Note This bit is only present in the 32- to 128-pin R7F100G and 32- to 48-pin R7F104G products.

Caution 1. Be sure to leave bits 3 and 2 set to 0.
Caution 2. Be sure to leave bit 1 in the 30-pin R7F100G products to 0.

Section 25 Power-on-reset Circuit (POR)

25.1 Functions of Power-on-reset Circuit

The power-on-reset circuit (POR) has the following functions.

- Generates internal reset signal at power on.
  The reset signal is released when the supply voltage (V_{DD}) exceeds the detection voltage (V_{POR}). Note that the reset state must be retained until the operating voltage becomes in the range defined in **37.4 AC Characteristics**. This is done by utilizing LVD0 or controlling the externally input reset signal.
- Compares supply voltage (V_{DD}) and detection voltage (V_{PDR}), generates internal reset signal when $V_{DD} < V_{PDR}$. Note that, after power supply is turned off, this MCU should be placed in the STOP mode, or in the reset state by utilizing LVD0 or externally input reset signal, before the operating voltage falls below the range defined in **37.4 AC Characteristics**. When restarting the operation, make sure that the operating voltage has returned within the range of operation.

Caution **If an internal reset signal is generated by the power-on-reset circuit, the reset control flag register (RESF) and power-on-reset status register (PORSR) are cleared to 00H.**

Remark 1. This MCU incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset source is located in the reset control flag register (RESF) for when an internal reset signal is generated by the watchdog timer (WDT), voltage detector (LVD), illegal instruction execution, RAM parity error, or illegal-memory access. The RESF register is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by the watchdog timer (WDT), voltage detector (LVD), illegal instruction execution, RAM parity error, or illegal-memory access. For details of the RESF register, see **Section 24 Reset Function**.

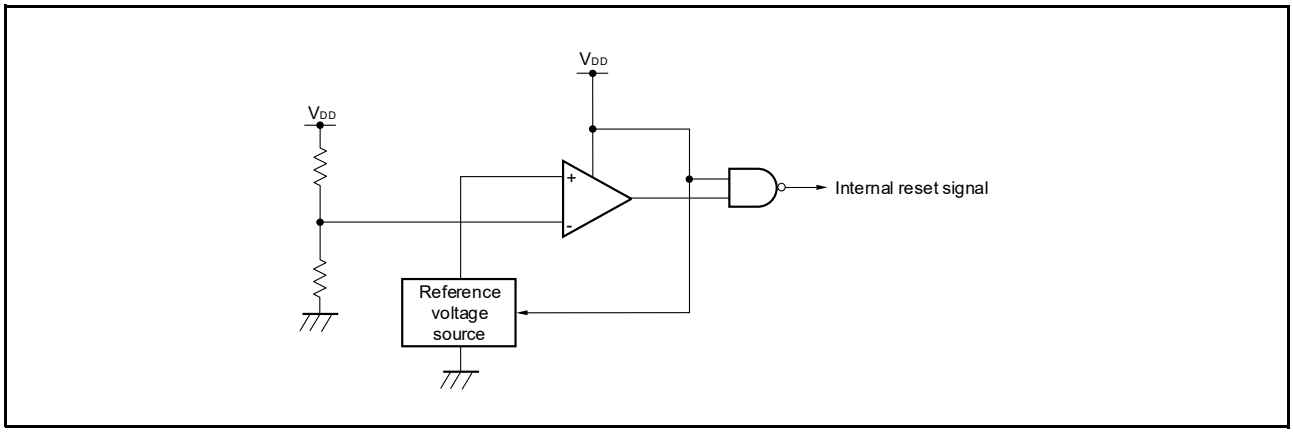
Remark 2. The power-on-reset status register (PORSR) is used to check the occurrence of an internal reset from the power-on-reset circuit. For details of the PORSR register, see **Section 24 Reset Function**.

Remark 3. V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage
For details, see **37.6.5 Characteristics of the POR circuit**.

25.2 Configuration of Power-on-reset Circuit

Figure 25 - 1 shows the block diagram of the power-on-reset circuit.

Figure 25 - 1 Block Diagram of Power-on-reset Circuit

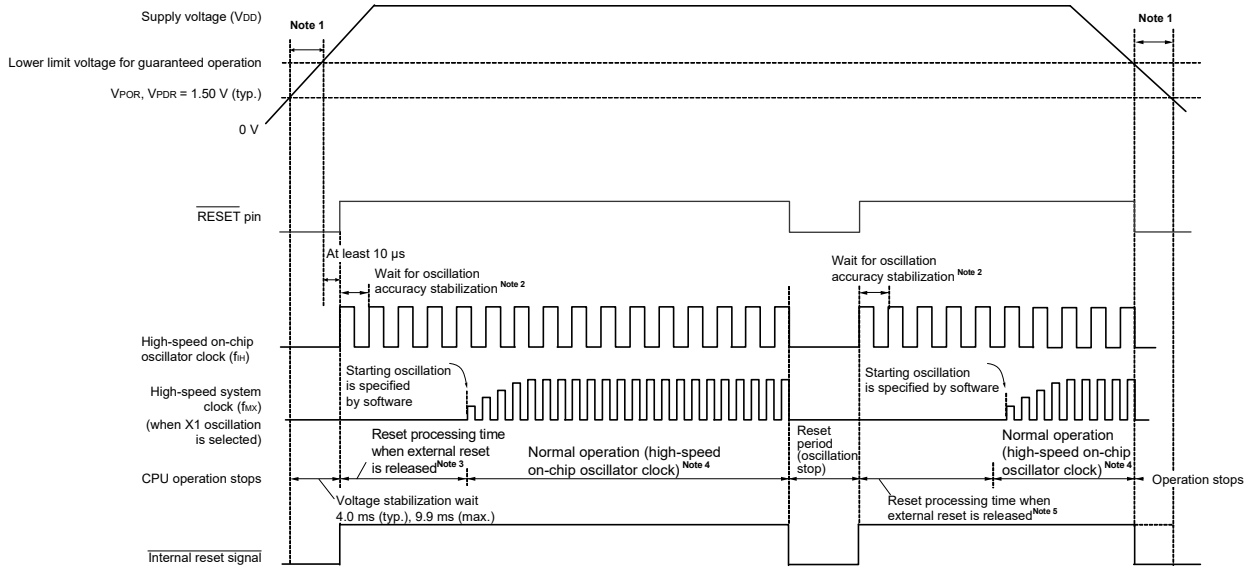


25.3 Operation of Power-on-reset Circuit

Figure 25 - 2 shows the timing of generation of the internal reset signal by the power-on-reset circuit and voltage detector.

Figure 25 - 2 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (1/2)

<1> When the externally input reset signal on the $\overline{\text{RESET}}$ pin is used



Note 1. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **37.4 AC Characteristics**. This is done by controlling the externally input reset signal. After power supply is turned off, this MCU should be placed in the STOP mode, or in the reset state by utilizing the voltage detection circuit or externally input reset signal, before the voltage falls below the range of operation. When restarting the operation, make sure that the operating voltage has returned within the range of operation.

Note 2. The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

Note 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 $\overline{\text{RESET}}$ signal is driven high as well as the voltage stabilization wait time after VPOR (1.50 V, typ.) is reached.
With the LVD circuit in use: 0.506 ms (typ.), 0.694 ms (max.)
With the LVD circuit not in use: 0.201 ms (typ.), 0.335 ms (max.)

<R> **Note 4.** The CPU clock can be switched from the high-speed on-chip oscillator clock to the high-speed system clock or subsystem 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.

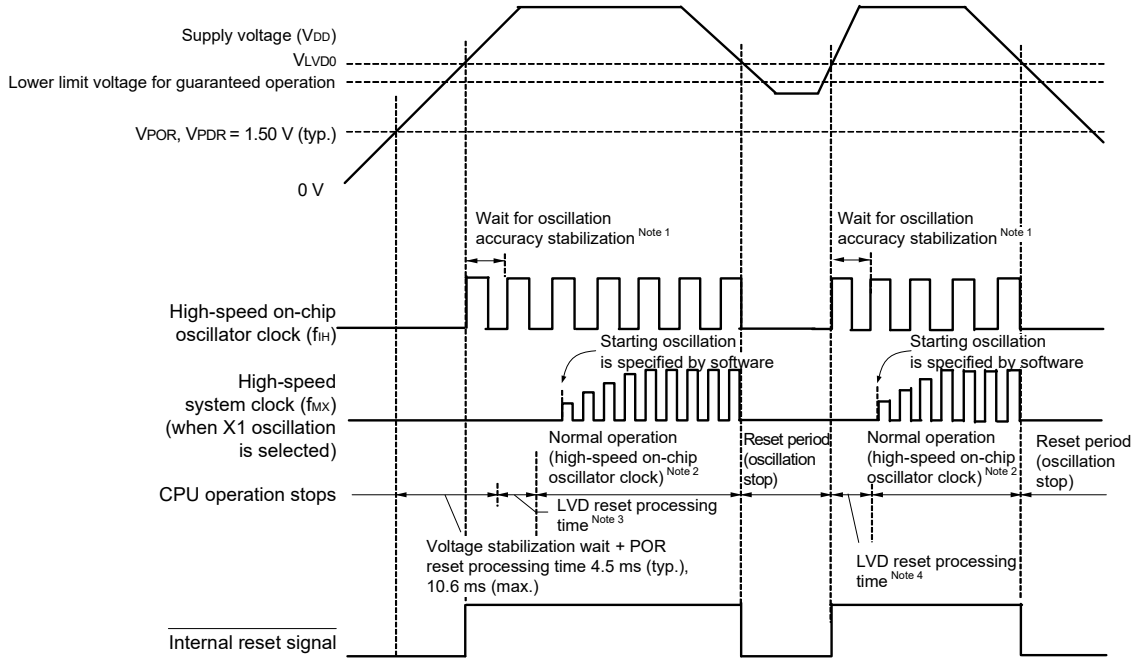
Note 5. The reset processing times in the case of the second or subsequent external reset following release from the POR state are listed below.
With the LVD circuit in use: 0.476 ms (typ.), 0.616 ms (max.)
With the LVD circuit not in use: 0.170 ms (typ.), 0.257 ms (max.)

Caution **For power-on reset, be sure to use the externally input reset signal on the $\overline{\text{RESET}}$ pin when LVD0 is off. For details, see Section 26 Voltage Detector (LVD).**

Remark VPOR: POR power supply rise detection voltage
VPDR: POR power supply fall detection voltage

Figure 25 - 2 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (2/2)

<2> LVD0 reset mode (LVD0SEL of the user option byte (000C1H) = 1)



Note 1. The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

<R> **Note 2.** The CPU clock can be switched from the high-speed on-chip oscillator clock to the high-speed system clock or subsystem 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.

Note 3. The time until normal operation starts includes the following LVD reset processing time after the LVD0 detection level (V_{LVD0}) is reached as well as the voltage stabilization wait + POR reset processing time after the V_{POR} (1.50 V, typ.) is reached.
LVD reset processing time: 0 to 0.041 ms (max.)

Note 4. When the power supply voltage is below the lower limit for operation and the power supply voltage is then restored after an internal reset is generated only by the voltage detector (LVD0), the following LVD reset processing time is required after the LVD0 detection level (V_{LVD0}) is reached.
LVD reset processing time: 0.040 ms (typ.), 0.041 ms (max.)

Remark 1. V_{LVD0}: LVD detection voltage
V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Remark 2. When the LVD0 interrupt mode is selected (LVD0SEL of the user option byte (000C1H) = 0), the time until normal operation starts after power is turned on is the same as the time specified in **Note 3** of **<2> LVD0 reset mode (LVD0SEL of the user option byte (000C1H) = 1)** in **Figure 25 - 2**.

Remark 3. Operation of LVD1 is stopped when power is initially supplied. LVD1 is also stopped by an internal reset.

Section 26 Voltage Detector (LVD)

26.1 Functions of Voltage Detector

Enabling, selecting the operation mode, and setting the detection voltage (V_{LVD0}) for voltage detector 0 (LVD0) are done by using a user option byte (000C1H). On the other hand, enabling, selecting the operation mode, and setting the detection voltage (V_{LVD1}) for voltage detector 1 (LVD1) are done by using the voltage detection level register. The voltage detectors have the following functions.

- LVD0 and LVD1 compare the supply voltage (V_{DD}) with the detection voltage (V_{LVD0}, V_{LVD1}), and generate an internal reset or internal interrupt signal.
- The user option byte is used to select the detection voltage (V_{LVD0}) for LVD0 from among 6 voltages (for details, see **Section 32 Option Bytes**).
- The voltage detection level register is used to select the detection voltage (V_{LVD1}) for LVD1 from among 18 voltages.
- Operable in STOP mode.
- After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **37.4 AC Characteristics**. This is done by utilizing LVD0 or controlling the externally input reset signal. After the power supply is turned off, this MCU should be placed in the STOP mode, or placed in the reset state by utilizing LVD0 or controlling the externally input reset signal before the voltage falls below the operating range.

The internal reset and internal interrupt signals are generated in each mode as follows.

Reset Mode LVD0	Reset Mode LVD1	Interrupt 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 internal reset on detecting $V_{DD} < V_{LVD1}$ after LVD1 operation has been enabled.	Retains the state of an internal reset by the LVD immediately after a reset until $V_{DD} \geq V_{LVD0}$. Releases the LVD internal reset by detecting $V_{DD} \geq V_{LVD0}$. Generates an interrupt request signal (INTLVI) by detecting $V_{DD} < V_{LVD0}$ or $V_{DD} \geq V_{LVD0}$ after the LVD internal reset is released.	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 LVD0 or LVD1 is operating, whether the supply voltage is no less than or less than the detection level can be checked by reading the voltage detection flag (LVDnF: bits 0 and 1 of the voltage detection register (LVIM)). The LVIRF bit of the reset control flag register (RESF) is set to 1 if reset occurs. For details of the RESF register, see **Section 24 Reset Function**.

26.2 Configuration of Voltage Detector

Figures 26 - 1 and **26 - 2** are block diagrams of the voltage detectors.

Figure 26 - 1 Block Diagram of LVD0

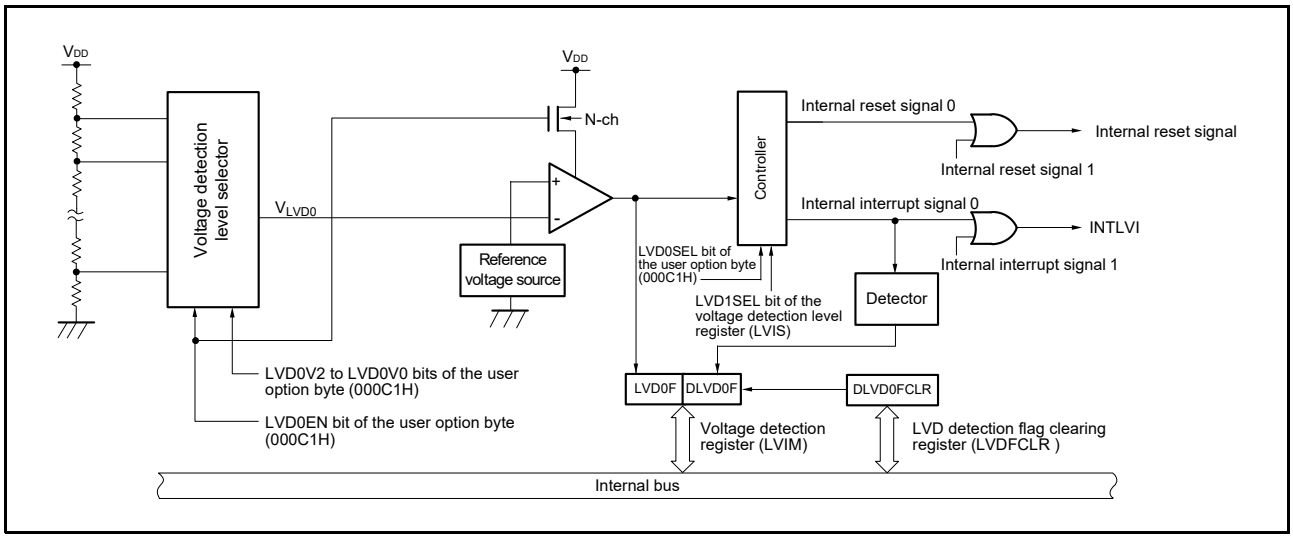
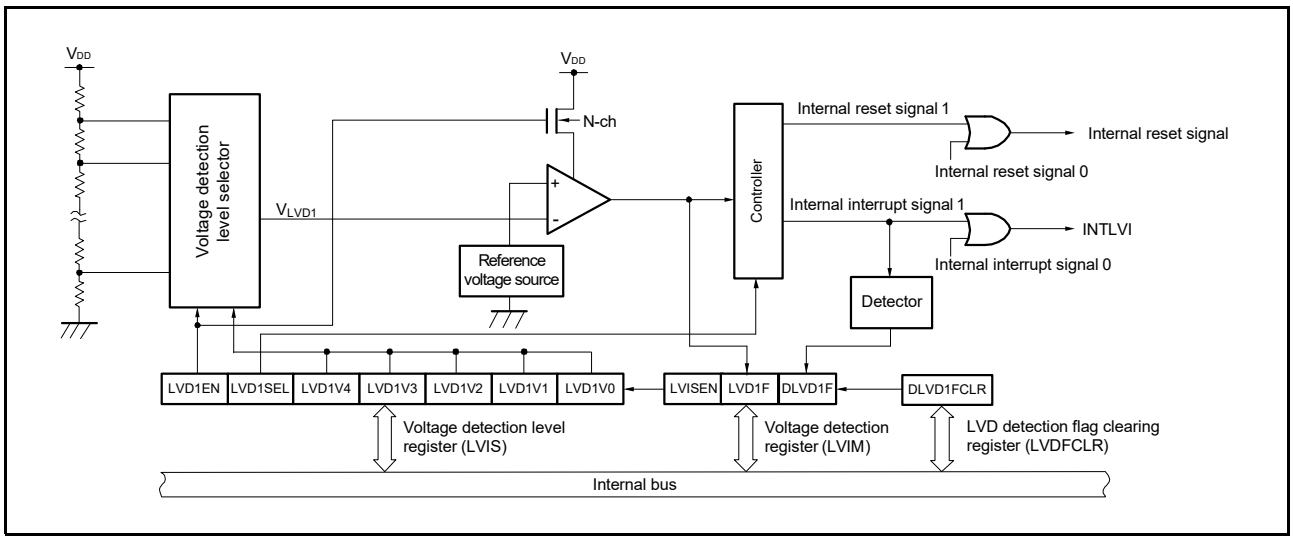


Figure 26 - 2 Block Diagram of LVD1



26.3 Registers for Controlling the Voltage Detector

The following registers are used to control the voltage detector.

- User option byte (000C1H/040C1H): See **Section 32 Option Bytes**.
- Voltage detection register (LVIM)
- LVD detection flag clearing register (LVDFCLR)
- Voltage detection level register (LVIS)

26.3.1 Voltage detection register (LVIM)

This register is used to specify whether to enable or disable rewriting the voltage detection level register (LVIS), as well as to check the states of LVD0 and LVD1. This register can be set by a 1-bit or 8-bit memory manipulation instruction.
The value of this register following a reset is 00H.

Figure 26 - 3 Format of Voltage Detection Register (LVIM)

Address: FFFA9H
After reset: 00H**Note 1**
R/W: R/W**Notes 2, 3**

Symbol	<7>	6	5	4	<3>	<2>	<1>	<0>
LVIM	LVISEN	0	0	0	DLVD1F	DLVD0F	LVD1F	LVD0F

LVISEN	Specification of whether to enable or disable rewriting the voltage detection level register (LVIS)
0	Rewriting of the LVIS register is disabled.
1**Note 4**	Rewriting of the LVIS register is enabled (reset and interrupt generation by LVD1 are masked).

DLVDnF	LVDn detection interrupt flag
0	The given LVDn interrupt has not been detected.
1	The given LVDn interrupt has been detected.

LVDnF	Voltage detection flag
0	Supply voltage (V_{DD}) ≥ detection voltage (V_{LVDn}), or when LVD is off
1	Supply voltage (V_{DD}) < detection voltage (V_{LVDn})

Note 1. 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}).
Note 2. Bits 0 and 1 are read-only.
Note 3. Bits 2 and 3 are read-only. These bits are cleared by using the LVD detection flag clearing register (LVDFCLR).
Note 4. While the LVISEN bit is 1, the reset and interrupt generation by LVD1 are masked. Therefore, set the LVISEN bit to 0 after having written a new value to the LVIS register.

Remark n = 0, 1

26.3.2 LVD detection flag clearing register (LVDFCLR)

This register is used to clear the interrupt detection flags (DLVD0F and DLVD1F) of the voltage detection register (LVIM). The LVDFCLR register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 26 - 4 Format of LVD Detection Flag Clearing Register (LVDFCLR)

Address: F0218H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	<3>	<2>	1	0
LVDFCLR	0	0	0	0	DLVD1FCLR	DLVD0FCLR	0	0

DLVD1FCLR **Note**	Clearing DLVD1F
0	No effect
1	Writing 1 to this bit clears the DLVD1F flag.

DLVD0FCLR **Note**	Clearing DLVD0F
0	No effect
1	Writing 1 to this bit clears the DLVD0F flag.

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.

26.3.3 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. The value of this register following a reset is 19H.

Figure 26 - 5 Format of Voltage Detection Level Register (LVIS)

Address: FFFAAH
After reset: 19H
R/W: R/W

Symbol	<7>	<6>	5	4	3	2	1	0
LVIS	LVD1EN	LVD1SEL	0	LVD1V4	LVD1V3	LVD1V2	LVD1V1	LVD1V0

LVD1EN	Enabling operation of LVD1
0	Operation stopped
1	Operation enabled

LVD1SEL	Operation mode of LVD1
0	Interrupt mode
1	Reset mode

LVD1V4	LVD1V3	LVD1V2	LVD1V1	LVD1V0	Detection voltages for LVD1 Notes 1, 2, 3		
						Rising edge	Falling edge
1	1	1	1	1	V_{LVD117}	1.67 V Note 4	1.63 V Note 4
1	1	1	1	0	V_{LVD116}	1.78 V Note 4	1.74 V Note 4
1	1	1	0	1	V_{LVD115}	1.88 V Note 4	1.84 V Note 4
1	1	1	0	0	V_{LVD114}	1.98 V	1.94 V
1	1	0	1	1	V_{LVD113}	2.09 V	2.04 V
1	1	0	1	0	V_{LVD112}	2.20 V	2.15 V
1	1	0	0	1	V_{LVD111}	2.30 V	2.25 V
1	1	0	0	0	V_{LVD110}	2.40 V	2.35 V
1	0	1	1	1	V_{LVD19}	2.50 V	2.45 V
1	0	1	1	0	V_{LVD18}	2.66 V	2.60 V
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

Note 1. The LVD1V4 to LVD1V0 bits can only be rewritten once after release from the reset state.

Note 2. When setting LVD0 to reset mode, set the detection voltage of LVD1 higher than the detection voltage of LVD0.

(Note and Caution continue on the next page.)

Note 3. If LVD0 is set to interrupt mode and the LVD0 detection voltage is greater than the LVD1 detection voltage, LVD0 becomes undefined after the LVD1 setting following release from the reset state.

Note 4. This setting can only be used when LVD0 is off.

Caution When the values in the LVIS register are to be changed, do so according to the procedure described in Figures 26 - 10 and 26 - 11.

26.4 Operation of Voltage Detector

26.4.1 Setting for use in the reset mode

Enabling, selecting the operation mode (reset mode: LVD0SEL = 1), and setting the detection voltage (V_{LVD0}) for LVD0 are done by using a user option byte (000C1H). On the other hand, enabling, selecting the operation mode (reset mode: LVD1SEL = 1), and setting the detection voltage (V_{LVD1}) for LVD1 are done by using the voltage detection level register (LVIS).

- Operation in LVD reset mode
  When LVD0 is set for the reset mode (the value of the LVD0SEL bit in the user option byte is 1), 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}). Operation of LVD1 is stopped when power is initially supplied. When LVD1 operation is enabled, it generates an internal reset when the power supply voltage (V_{DD}) falls below the voltage detection level (V_{LVD1}). 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 internal reset at the time its operation is enabled. If LVD1 is set for reset mode, LVD0 is placed in interrupt mode. In addition, the generation of an internal reset by LVD1 places LVD0 in reset mode. LVD1 detection voltage can only be set once after release from the reset state.

Figure 26 - 6 and **Figure 26 - 7** show the timing of the internal reset signal generated in LVD0 and LVD1, respectively.

Figure 26 - 6 Timing of the Internal Reset Signal Generated in LVD0

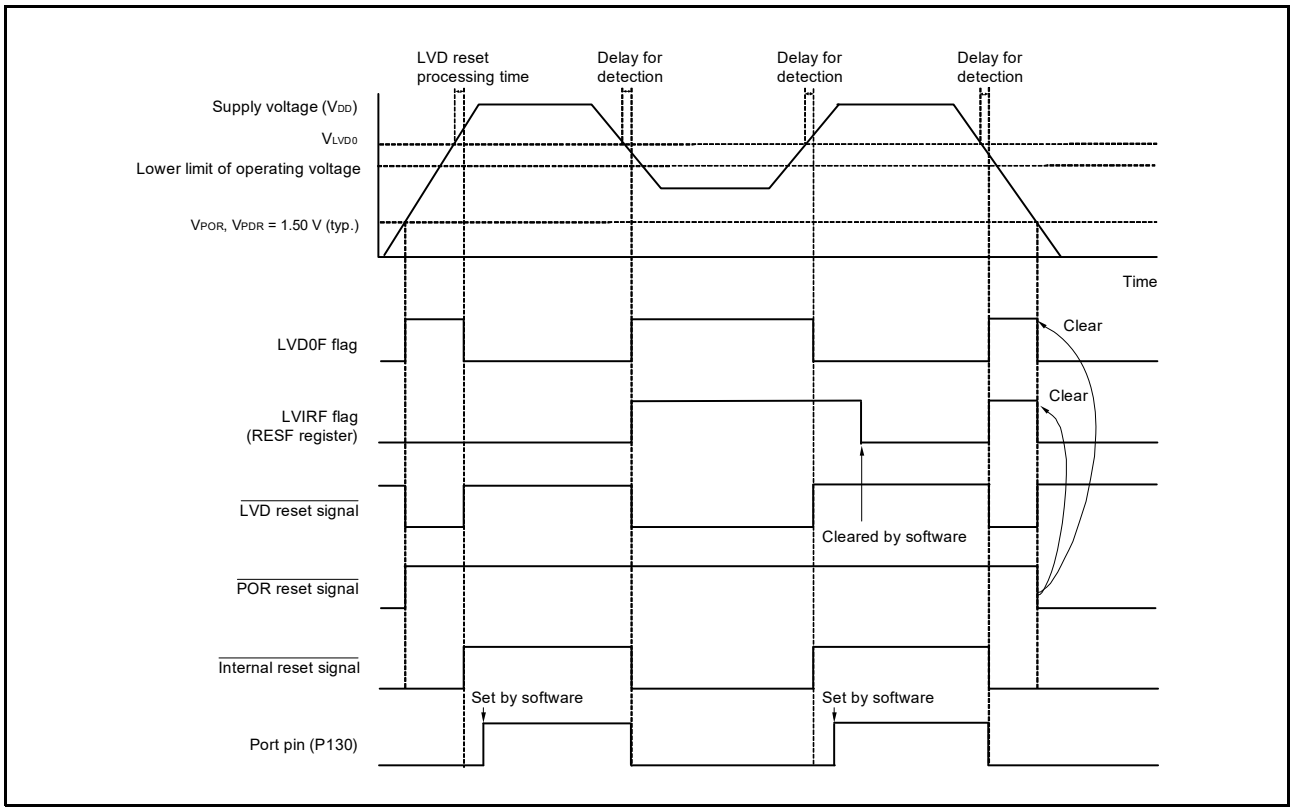
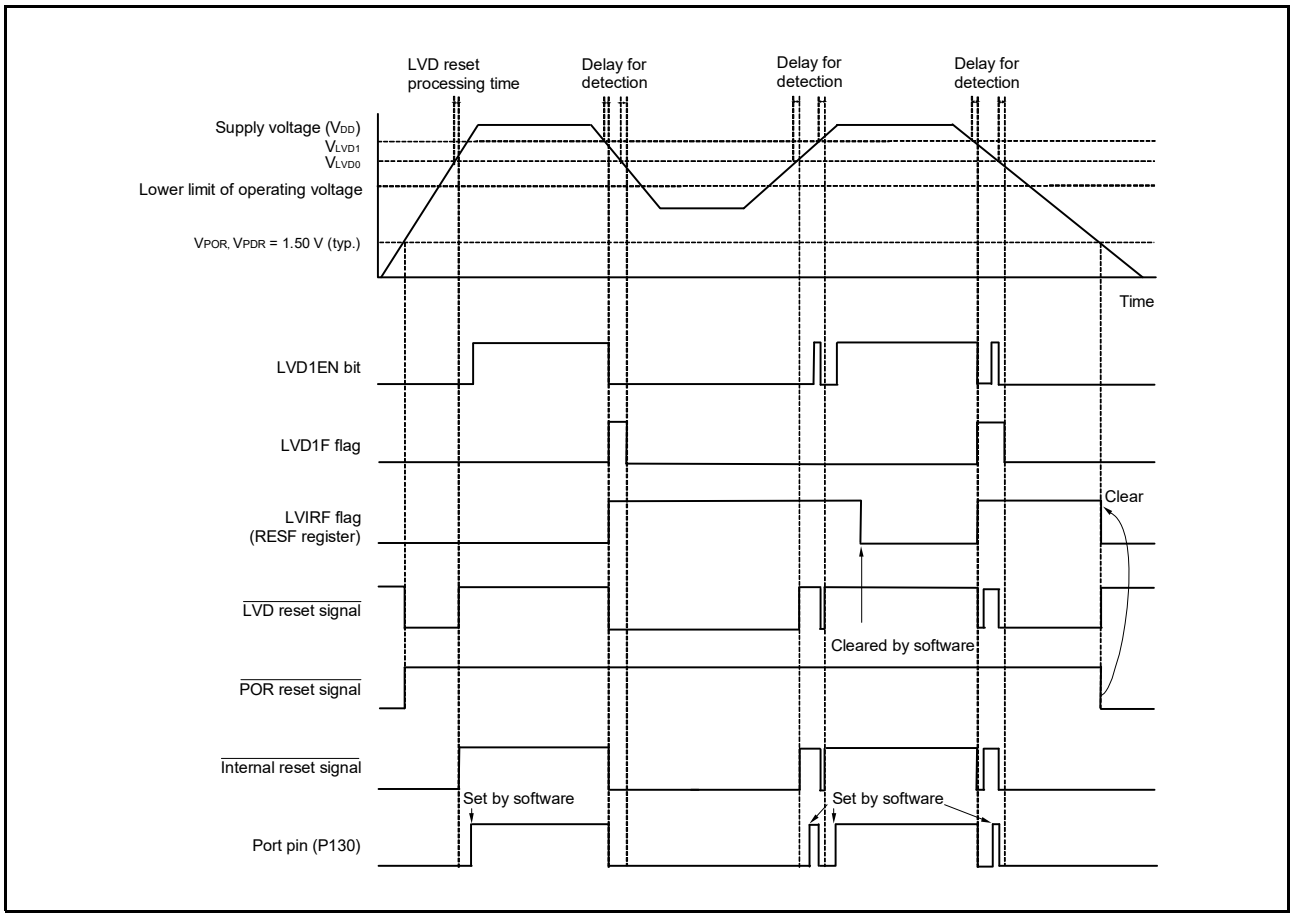


Figure 26 - 7 Timing of the Internal Reset Signal Generated in LVD1



Remark LVD0: Reset mode

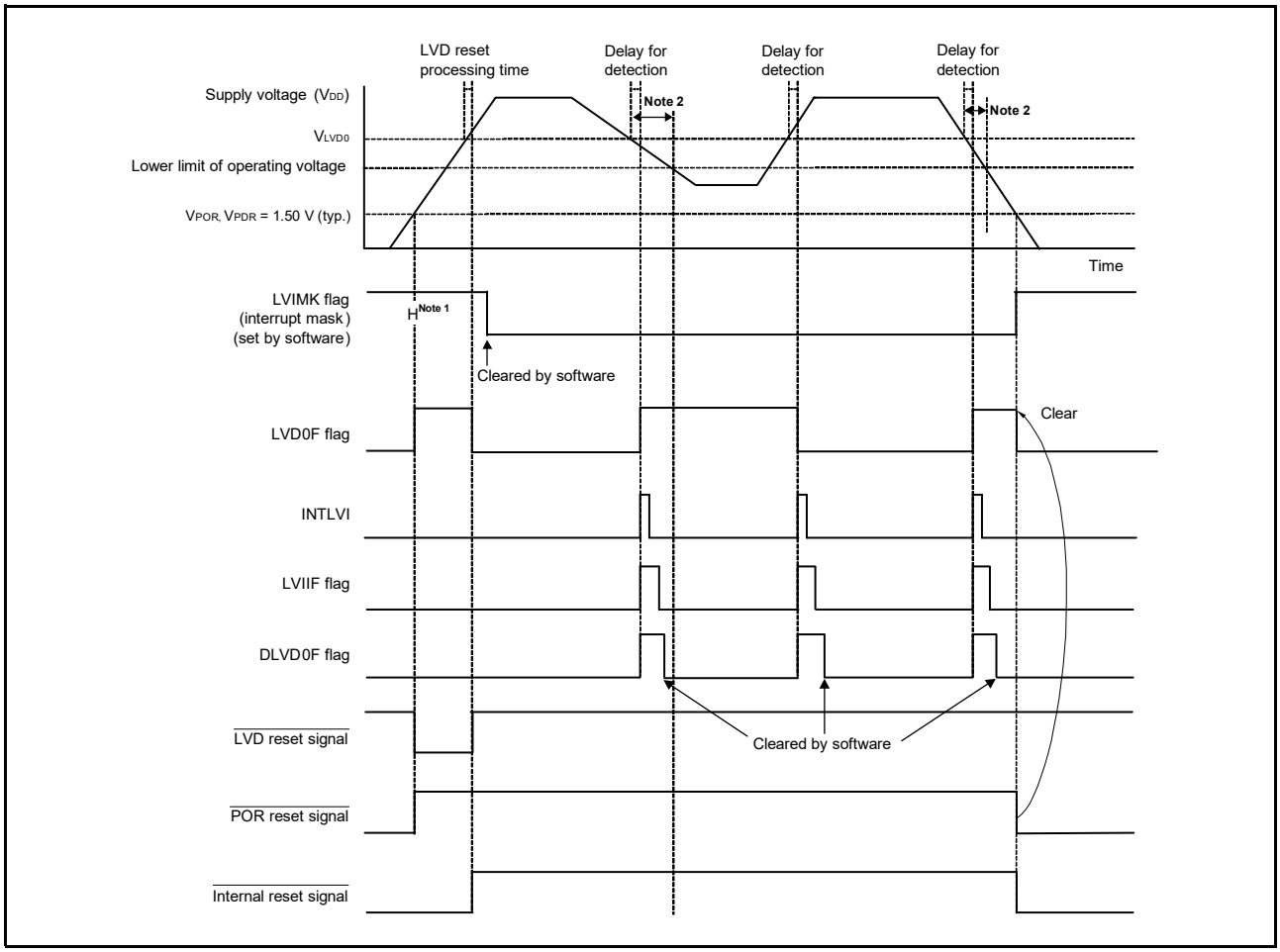
26.4.2 Setting for use in the interrupt mode

Enabling, selecting the operation mode (interrupt mode: LVD0SEL = 0), and setting the detection voltage (V_{LVD0}) for LVD0 are done by using a user option byte (000C1H). On the other hand, enabling, selecting the operation mode (interrupt mode: LVD1SEL = 0), and setting the detection voltage (V_{LVD1}) for LVD1 are done by using the voltage detection level register (LVIS).

- Operation in LVD interrupt mode
  When LVD0 is set for the interrupt mode (the value of the LVD0SEL bit in the user option byte is 0), 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}) immediately after a reset has been generated. The internal reset is released when the power supply voltage (V_{DD}) exceeds the rising voltage detection level (V_{LVD0}). After the internal reset signal has been deasserted, LVD0 generates an interrupt request signal (INTLVI) if the power supply voltage (V_{DD}) falls below the voltage detection level (V_{LVD0}). Similarly, when the power supply voltage (V_{DD}) rises above the voltage detection level (V_{LVD0}), LVD0 also generates an interrupt request signal (INTLVI). When the voltage falls, 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 **37.4 AC Characteristics**. When restarting the operation, make sure that the power supply voltage has returned within the operating voltage range. 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 26 - 8 and **Figure 26 - 9** show the timing of the interrupt request signal generated in LVD0 and LVD1, respectively.

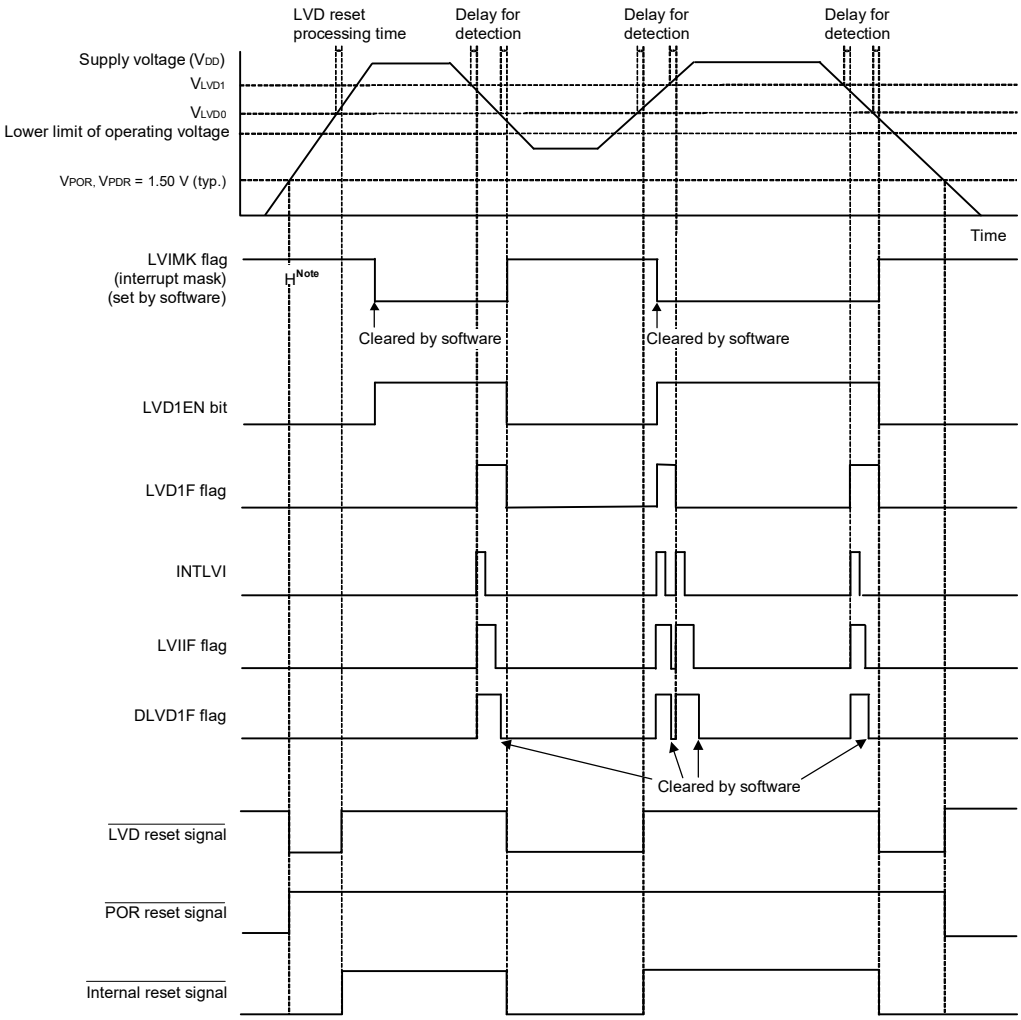
Figure 26 - 8 Timing of the Interrupt Request Signal Generated in LVD0



Note 1. The LVIMK flag is set to 1 by reset signal generation.

Note 2. When the voltage falls, 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 **37.4 AC Characteristics**. When restarting the operation, make sure that the operating voltage has returned within the operating voltage range.

Figure 26 - 9 Timing of the Interrupt Request Signal Generated in LVD1



Note The LVIMK flag is set to 1 by reset signal generation.

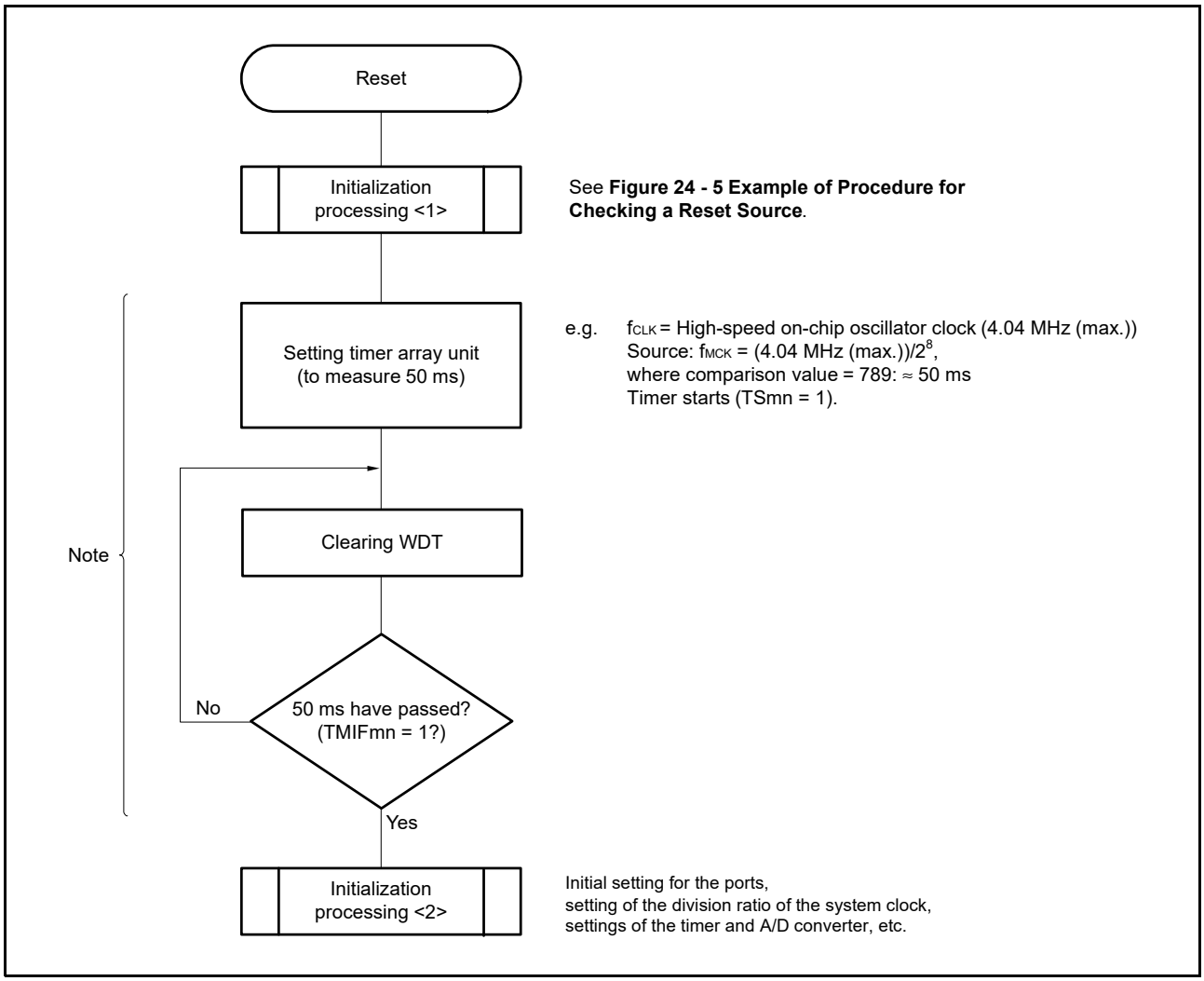
Remark LVD0: Reset mode

26.5 Points for Caution when the Voltage Detector Is to be Used

1. Voltage fluctuation when power is supplied
   In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the detection voltages for LVD0 or LVD1, the system may be repeatedly reset and released from the reset state. In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking the following action.

   <Action>
   After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports.

Figure 26 - 10 Example of Software Processing If Supply Voltage Fluctuation Is 50 ms or Less in Vicinity of the Detection Voltages for LVD0 or LVD1

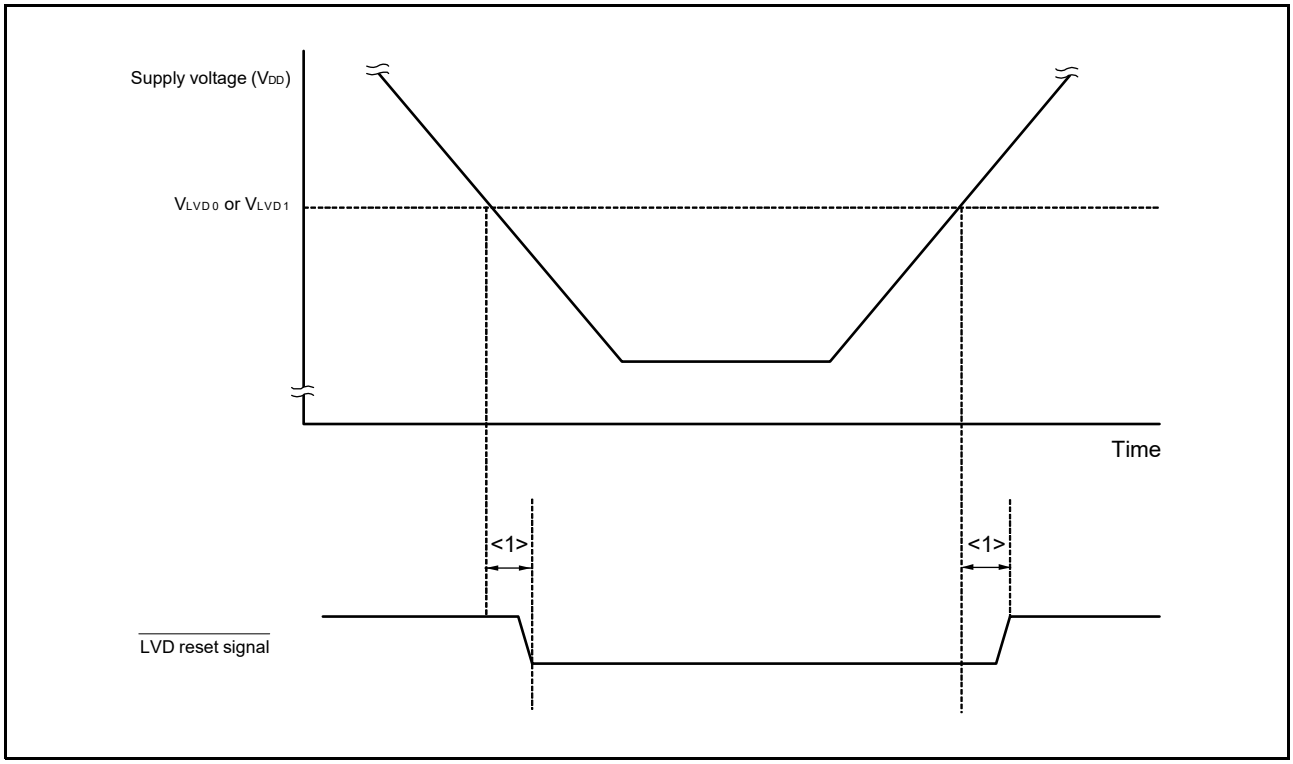


Note If reset is generated again during this period, initialization processing <2> is not started.

Remark m = 0, 1
n = 0 to 7

2. Delays from the time an LVD0 or LVD1 reset source condition is satisfied until an LVD0 or LVD1 reset has been generated and deasserted
   The delay is from the time the power supply voltage (V_{DD}) becomes less than the LVD0 or LVD1 falling detection voltage (V_{LVD0} or V_{LVD1}) until the LVD0 or LVD1 reset is generated. In the same way, the delay is from the time the LVD0 or LVD1 rising detection voltage (V_{LVD0} or V_{LVD1}) becomes no greater than the power supply voltage (V_{DD}) until the LVD0 or LVD1 reset is deasserted. See **Figure 26 - 11**.

Figure 26 - 11 Delays from the Time an LVD0 or LVD1 Reset Source Condition is Satisfied until an LVD0 or LVD1 Reset has been Generated and Deasserted



Remark <1>: Delay for detection (500 μs (max.))

3. Turning power on when LVD0 is off
   Use the external rest input via the $\overline{\text{RESET}}$ pin when LVD0 is off.
   For an external reset, input a low level for 10 µs or more to the $\overline{\text{RESET}}$ pin. To perform an external reset upon power application, input a low level to the $\overline{\text{RESET}}$ pin, turn power on, continue to input a low level to the pin for 10 µs or more within the operating voltage range shown in **37.4 AC Characteristics**, and then input a high level to the pin.

4. Operating voltage fall when LVD0 is off or the interrupt mode is selected
   When the operating voltage falls with LVD0 off or with the interrupt mode selected, 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 **37.4 AC Characteristics**. When restarting the operation, make sure that the operating voltage has returned within the operating voltage range.

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.

Section 27 Safety Functions

27.1 Overview of Safety Functions

This MCU provides the following safety functions to comply with the IEC60730 safety standard. These safety functions enable the microcontroller to self-diagnose abnormalities and safely stop operating if an abnormality is detected.

1. Flash memory CRC operation (high-speed CRC, general-purpose CRC)
   This detects data errors in the flash memory by performing CRC operations. The following two CRC functions are provided in this MCU and they 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. Flash memory guard function
   This prevents rewriting of data in the flash memory due to incorrect CPU operations.
3. RAM parity error detection
   This detects parity errors when reading RAM data.
4. RAM guard function
   This prevents rewriting of data in RAM due to incorrect CPU operations.
5. SFR guard function
   This prevents rewriting of data in the SFRs due to incorrect CPU operations.
6. Illicit memory access detection
   This detects illicit accesses to invalid memory areas (such as areas where no memory is allocated and areas to which access is restricted).
7. Guard function of invalid memory access detection control register (IAWCTL)
   This prevents rewriting of the invalid memory access detection control register due to incorrect CPU operations.
8. Frequency detection
   This function allows a self-check of the CPU/peripheral hardware clock frequencies using the timer array unit.
9. Testing of the A/D converter
   The A/D converter can be self-checked by converting the positive or negative reference voltages, analog input voltages on channels ANIxx, temperature sensor output voltage, or the internal reference voltage.
10. Detection of the digital output signal level of the I/O pins
    This is used to read the output level of an I/O pin when the pin is in the output mode.
11. UART loopback
    This is used to confirm that the transmit data are output normally by shutting off the TxDn and RxDn pins of UARTn and the TxDAm and RxDAm pins of UARTAm from the outside and connecting them within the MCU to loop back the output from the transmission shift register to the reception shift register.

Remark 1. m = 0, 1; n = 0 to 3

Remark 2. For usage examples of the safety functions complying with the IEC60730 safety standard, refer to the **application notes IEC60730/60335 Self Test Library for RL78 MCU**.

27.2 Registers for Controlling the Safety Functions

The following registers are used to control the safety functions.

- Flash memory CRC control register (CRC0CTL)
- Flash memory CRC operation result register (PGCRCL)
- CRC input register (CRCIN)
- CRC data register (CRCD)
- Code flash memory guard register (GFLASH0)
- Data flash memory guard register (GFLASH1)
- Flash security area guard register (GFLASH2)
- RAM parity error control register (RPECTL)
- Invalid memory access detection control register (IAWCTL)
  - RAM guard function
  - SFR guard function
  - Illicit memory access detection
- Guard register of IAWCTL register (GIAWCTL)
- Timer input select register 0 (TIS0)
- A/D test register (ADTES)
- Analog input channel specification register (ADS)
- Port mode select register (PMS)
- UART loopback select register (ULBS)

The content of each register is described in **27.3 Operation of Safety Functions**.

27.3 Operation of Safety Functions

27.3.1 Flash memory CRC operation (high-speed CRC)

The IEC60730 standard mandates the checking of data in the flash memory, and recommends using a CRC to do it. The high-speed CRC provided in this MCU 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. The high-speed CRC performs an operation by reading 32-bit data per clock from the flash memory while stopping the CPU. This function therefore can finish a check in a shorter time (for example, 512 μs@32 MHz with 64-Kbyte flash memory). 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 CRC operation result might differ during on-chip debugging because the monitor program is allocated.**

Remark The operation result is different between the high-speed CRC and the general-purpose CRC, because the general-purpose CRC operates in LSB first order.

27.3.1.1 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. The value of this register following a reset is 00H.

Figure 27 - 1 Format of Flash Memory CRC Control Register (CRC0CTL) (1/2)

Address: F02F0H
After reset: 00H
R/W: R/W

Symbol	<7>	6	5	4	3	2	1	0
CRC0CTL	CRC0EN	0	FEA5	FEA4	FEA3	FEA2	FEA1	FEA0

CRC0EN	Control of CRC ALU operation
0	Stops the operation.
1	Starts the operation according to HALT instruction execution.

FEA5	FEA4	FEA3	FEA2	FEA1	FEA0	High-speed CRC operation range
0	0	0	0	0	0	0000H to 03FFBH (16 Kbytes - 4 bytes)
0	0	0	0	0	1	0000H to 07FFBH (32 Kbytes - 4 bytes)
0	0	0	0	1	0	0000H to 0BFFBH (48 Kbytes - 4 bytes)
0	0	0	0	1	1	0000H to 0FFFBH (64 Kbytes - 4 bytes)
0	0	0	1	0	0	0000H to 13FFBH (80 Kbytes - 4 bytes)
0	0	0	1	0	1	0000H to 17FFBH (96 Kbytes - 4 bytes)
0	0	0	1	1	0	0000H to 1BFFBH (112 Kbytes - 4 bytes)
0	0	0	1	1	1	0000H to 1FFFBH (128 Kbytes - 4 bytes)
0	0	1	0	0	0	0000H to 23FFBH (144 Kbytes - 4 bytes)
0	0	1	0	0	1	0000H to 27FFBH (160 Kbytes - 4 bytes)
0	0	1	0	1	0	0000H to 2BFFBH (176 Kbytes - 4 bytes)
0	0	1	0	1	1	0000H to 2FFFBH (192 Kbytes - 4 bytes)
0	0	1	1	0	0	0000H to 33FFBH (208 Kbytes - 4 bytes)
0	0	1	1	0	1	0000H to 37FFBH (224 Kbytes - 4 bytes)
0	0	1	1	1	0	0000H to 3BFFBH (240 Kbytes - 4 bytes)
0	0	1	1	1	1	0000H to 3FFFBH (256 Kbytes - 4 bytes)
0	1	0	0	0	0	0000H to 43FFBH (272 Kbytes - 4 bytes)
0	1	0	0	0	1	0000H to 47FFBH (288 Kbytes - 4 bytes)
0	1	0	0	1	0	0000H to 4BFFBH (304 Kbytes - 4 bytes)
0	1	0	0	1	1	0000H to 4FFFBH (320 Kbytes - 4 bytes)
0	1	0	1	0	0	0000H to 53FFBH (336 Kbytes - 4 bytes)
0	1	0	1	0	1	0000H to 57FFBH (352 Kbytes - 4 bytes)

Figure 27 - 1 Format of Flash Memory CRC Control Register (CRC0CTL) (2/2)

FEA5	FEA4	FEA3	FEA2	FEA1	FEA0	High-speed CRC operation range
0	1	0	1	1	0	0000H to 5BFFBH (368 Kbytes - 4 bytes)
0	1	0	1	1	1	0000H to 5FFFBH (384 Kbytes - 4 bytes)
0	1	1	0	0	0	0000H to 63FFBH (400 Kbytes - 4 bytes)
0	1	1	0	0	1	0000H to 67FFBH (416 Kbytes - 4 bytes)
0	1	1	0	1	0	0000H to 6BFFBH (432 Kbytes - 4 bytes)
0	1	1	0	1	1	0000H to 6FFFBH (448 Kbytes - 4 bytes)
0	1	1	1	0	0	0000H to 73FFBH (464 Kbytes - 4 bytes)
0	1	1	1	0	1	0000H to 77FFBH (480 Kbytes - 4 bytes)
0	1	1	1	1	0	0000H to 7BFFBH (496 Kbytes - 4 bytes)
0	1	1	1	1	1	0000H to 7FFFBH (512 Kbytes - 4 bytes)
1	0	0	0	0	0	0000H to 83FFBH (528 Kbytes - 4 bytes)
1	0	0	0	0	1	0000H to 87FFBH (544 Kbytes - 4 bytes)
1	0	0	0	1	0	0000H to 8BFFBH (560 Kbytes - 4 bytes)
1	0	0	0	1	1	0000H to 8FFFBH (576 Kbytes - 4 bytes)
1	0	0	1	0	0	0000H to 93FFBH (592 Kbytes - 4 bytes)
1	0	0	1	0	1	0000H to 97FFBH (608 Kbytes - 4 bytes)
1	0	0	1	1	0	0000H to 9BFFBH (624 Kbytes - 4 bytes)
1	0	0	1	1	1	0000H to 9FFFBH (640 Kbytes - 4 bytes)
1	0	1	0	0	0	0000H to A3FFBH (656 Kbytes - 4 bytes)
1	0	1	0	0	1	0000H to A7FFBH (672 Kbytes - 4 bytes)
1	0	1	0	1	0	0000H to ABFFBH (688 Kbytes - 4 bytes)
1	0	1	0	1	1	0000H to AFFFBH (704 Kbytes - 4 bytes)
1	0	1	1	0	0	0000H to B3FFBH (720 Kbytes - 4 bytes)
1	0	1	1	0	1	0000H to B7FFBH (736 Kbytes - 4 bytes)
1	0	1	1	1	0	0000H to BBFFBH (752 Kbytes - 4 bytes)
1	0	1	1	1	1	0000H to BFFFBH (768 Kbytes - 4 bytes)
Other than the above						Setting prohibited

Caution **Be sure to leave bit 6 set to 0.**

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.

27.3.1.2 Flash memory CRC operation result register (PGCRCL)

This register holds the high-speed CRC operation results. The PGCRCL register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 27 - 2 Format of Flash Memory CRC Operation Result Register (PGCRCL)

Address: F02F2H, F02F3H
After reset: 0000H
R/W: 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

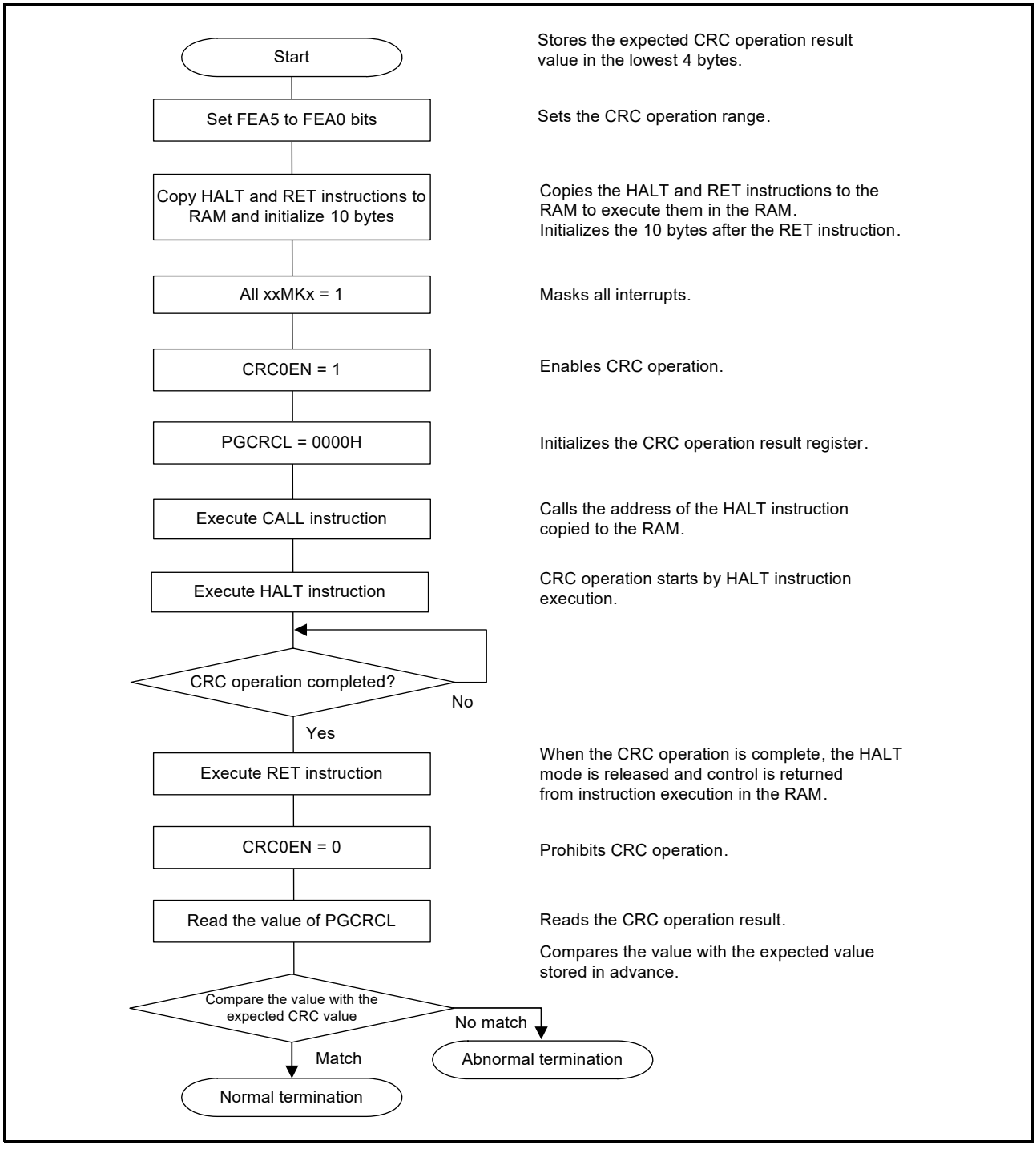
PGCRC15 to PGCRC0	High-speed CRC operation results
0000H to FFFFH	Holds the high-speed CRC operation results.

Caution This register can only be written if the CRC0EN bit of the CRC0CTL register is 1.

Figure 27 - 3 shows the flow of the high-speed CRC operation for the flash memory.

<Operation flow>

Figure 27 - 3 Flow of the High-speed CRC Operation for the Flash Memory

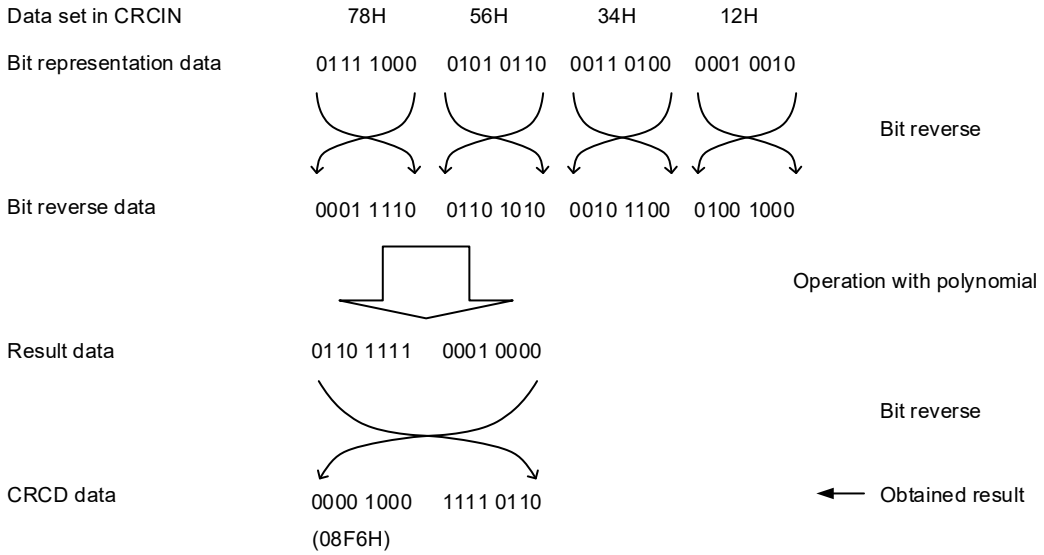


Caution 1. The CRC operation is executed only on the code flash memory.
Caution 2. Store the expected CRC operation value in the area below the operation range in the code flash memory.
Caution 3. The CRC operation is enabled by executing the HALT instruction in the RAM area. Be sure to execute the HALT instruction in the RAM area.

The expected CRC value can be calculated by using the CS+ integrated development environment. See the CS+ Integrated Development Environment user's manual for details.

27.3.2 CRC operation (general-purpose CRC)

The general-purpose CRC handles CRC operation as a peripheral module while the CPU is operating. The general-purpose 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). The CRC calculation function in the HALT mode can be used only during DMA transfer. The general-purpose CRC can operate either in the main system clock operation mode or in 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 are inverted in bit order and then calculated to allow for LSB-first communication. For example, if the data 12345678H are 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 Because the debugger rewrites the software break setting line to a break instruction during program execution, the CRC operation result differs if a software break is set in the CRC operation target area.

27.3.2.1 CRC input register (CRCIN)

This is an 8-bit register to set the CRC operation data of the general-purpose CRC. The possible setting range is 00H to FFH. The CRCIN register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 27 - 4 Format of CRC Input Register (CRCIN)

Address: FFFACH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
CRCIN								

Bits 7 to 0	Function
00H to FFH	Data input

27.3.2.2 CRC data register (CRCD)

This register holds the CRC operation result of the general-purpose CRC. The setting range is 0000H to FFFFH. After one clock cycle of the CPU/peripheral hardware clock (f_{CLK}) has elapsed from the time CRCIN register was written to, the CRC operation result is stored in the CRCD register. The CRCD register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 27 - 5 Format of CRC Data Register (CRCD)

Address: F02FAH, F02FBH
After reset: 0000H
R/W: R/W

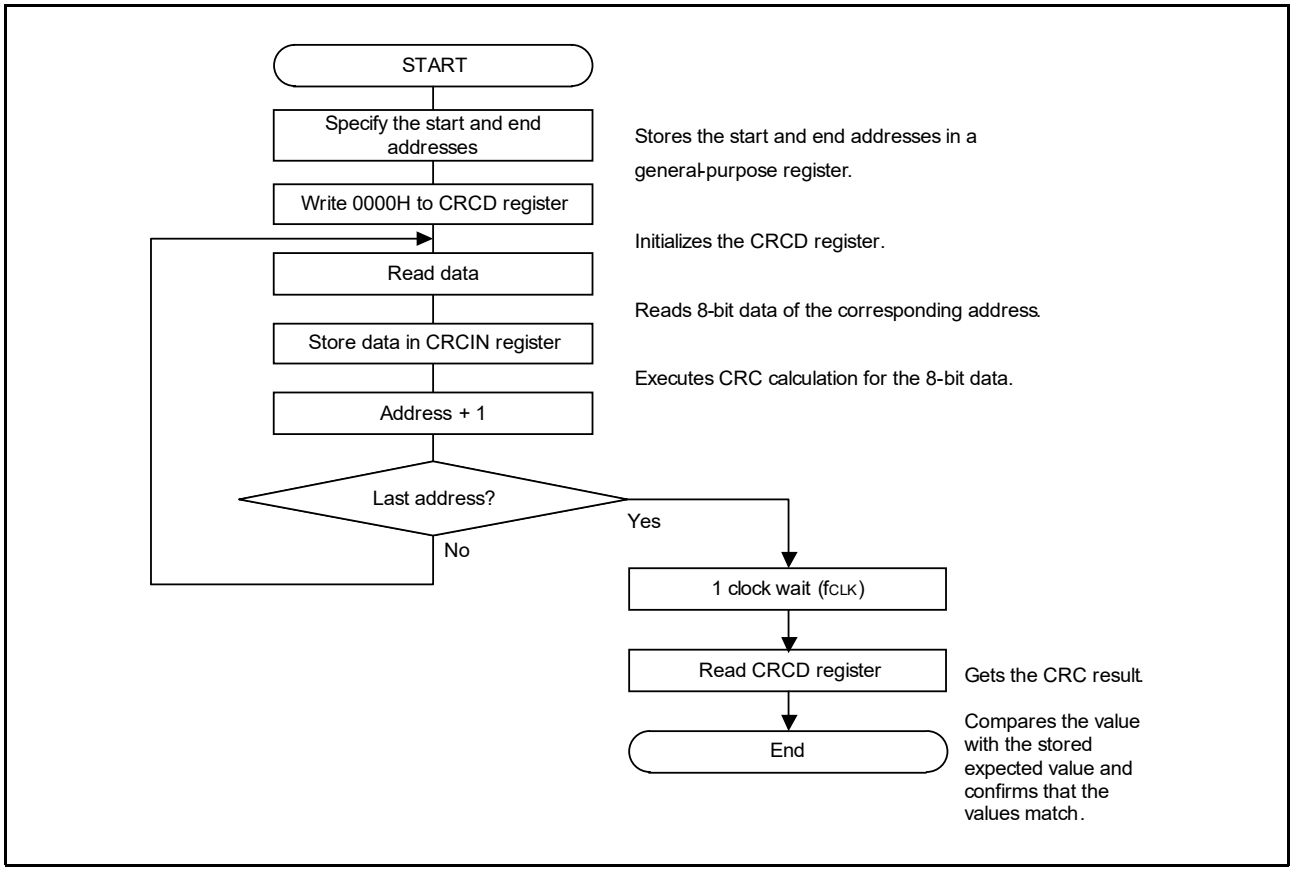
Symbol	15	14	13	12	11	10	9	8
CRCD								

7	6	5	4	3	2	1	0

Caution 1. Read the value written to the CRCD register before writing to the CRCIN register.
Caution 2. If conflict between writing a value and storing the operation result in the CRCD register occurs, the writing is ignored.

<Operation flow>

Figure 27 - 6 CRC Operation Function (General-purpose CRC)



27.3.3 Flash memory guard function

To ensure safe operation, the IEC60730 standard requires protecting flash memory from rewriting of its data due to incorrect CPU operations. This MCU has functionality to protect data in the code flash memory, data flash memory, and the security area in the flash memory. Enabling this function disables writing to the protected area of the flash memory. Reading from the protected area is possible.

27.3.3.1 Code flash memory guard register (GFLASH0)

This register is used to protect the code flash memory against being rewritten. To allow rewriting of the code flash memory, set the GFLASH0.GFL0 bit to 0. To protect the code flash memory, set the GFLASH0.GFL0 bit to 1. Be sure to confirm that the value has been set to 1 before reading from the code flash memory. The GFLASH0 register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 27 - 7 Format of Code Flash Memory Guard Register (GFLASH0)

Address: F0488H, F0489H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
GFLASH0	RKEY[7:0]							

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	GFL0

RKEY[7:0]	Function
Bits 7 to 0	The RKEY[7:0] bits contain the key code to control rewriting of this register. When rewriting the GFL0 bit, set RKEY[7:0] to 30H. These bits return 00H when read.

GFL0	Control of rewriting the code flash memory
0	Disables protection of the code flash memory (rewriting is allowed).
1	Enables protection of the code flash memory (rewriting is not allowed).

27.3.3.2 Data flash memory guard register (GFLASH1)

This register is used to protect the data flash memory against being rewritten. To allow rewriting of the data flash memory, set the GFLASH1.GFL1 bit to 0. To protect the data flash memory, set the GFLASH1.GFL1 bit to 1. Be sure to confirm that the value has been set to 1 before reading from the data flash memory. The GFLASH1 register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 27 - 8 Format of Data Flash Memory Guard Register (GFLASH1)

Address: F048AH, F048BH
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
GFLASH1	RKEY[7:0]							

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	GFL1

RKEY[7:0]	Function
Bits 7 to 0	The RKEY[7:0] bits contain the key code to control rewriting of this register. When rewriting the GFL1 bit, set RKEY[7:0] to C5H. These bits return 00H when read.

GFL1	Control of rewriting the data flash memory
0	Disables protection of the data flash memory (rewriting is allowed).
1	Enables protection of the data flash memory (rewriting is not allowed).

27.3.3.3 Flash security area guard register (GFLASH2)

This register is used to protect the security area in the flash memory, which holds the security settings. To allow rewriting of the security area in the flash memory, set the GFLASH2.GFL2 bit to 0. The GFLASH2 register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 27 - 9 Format of Flash Security Area Guard Register (GFLASH2)

Address: F048CH, F048DH
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
GFLASH2	RKEY[7:0]							

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	GFL2

RKEY[7:0]	Function
Bits 7 to 0	The RKEY[7:0] bits contain the key code to control rewriting of this register. When rewriting the GFL2 bit, set RKEY[7:0] to 9AH. These bits return 00H when read.

GFL2	Control of rewriting the security area in the flash memory
0	Disables protection of the security area in the flash memory (rewriting is allowed).
1	Enables protection of the security area in the flash memory (rewriting is not allowed).

27.3.4 RAM parity error detection

The IEC60730 standard mandates the checking of RAM data. A single-bit parity bit is therefore added to all 8-bit data in the RAM of each product of this group. By using this RAM parity error detection, the parity bit is appended when data are written, and the parity is checked when the data are read. This function can also be used to trigger a reset when a parity error occurs.

27.3.4.1 RAM parity error control register (RPECTL)

This register indicates the presence of a parity error and controls the reset triggered by a parity error. The RPECTL register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 27 - 10 Format of RAM Parity Error Control Register (RPECTL)

Address: F00F5H
After reset: 00H
R/W: R/W

Symbol	<7>	6	5	4	3	2	1	<0>
RPECTL	RPERDIS	0	0	0	0	0	0	RPEF

RPERDIS	Control of the reset triggered by a parity error
0	Enables parity error resets.
1	Disables parity error resets.

RPEF	Parity error status flag
0	No parity error has occurred.
1	A parity error has occurred.

Caution **The parity bit is appended when data are written, and the parity is checked when data are read. Accordingly, while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed before reading data. In pipeline operation, prefetching for the CPU may result in reading from an uninitialized RAM area beyond the RAM areas that are assumed to be in use, which can result in a RAM parity error. While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize a RAM area and an additional 10 bytes beyond it when instructions are to be fetched from a RAM area.**

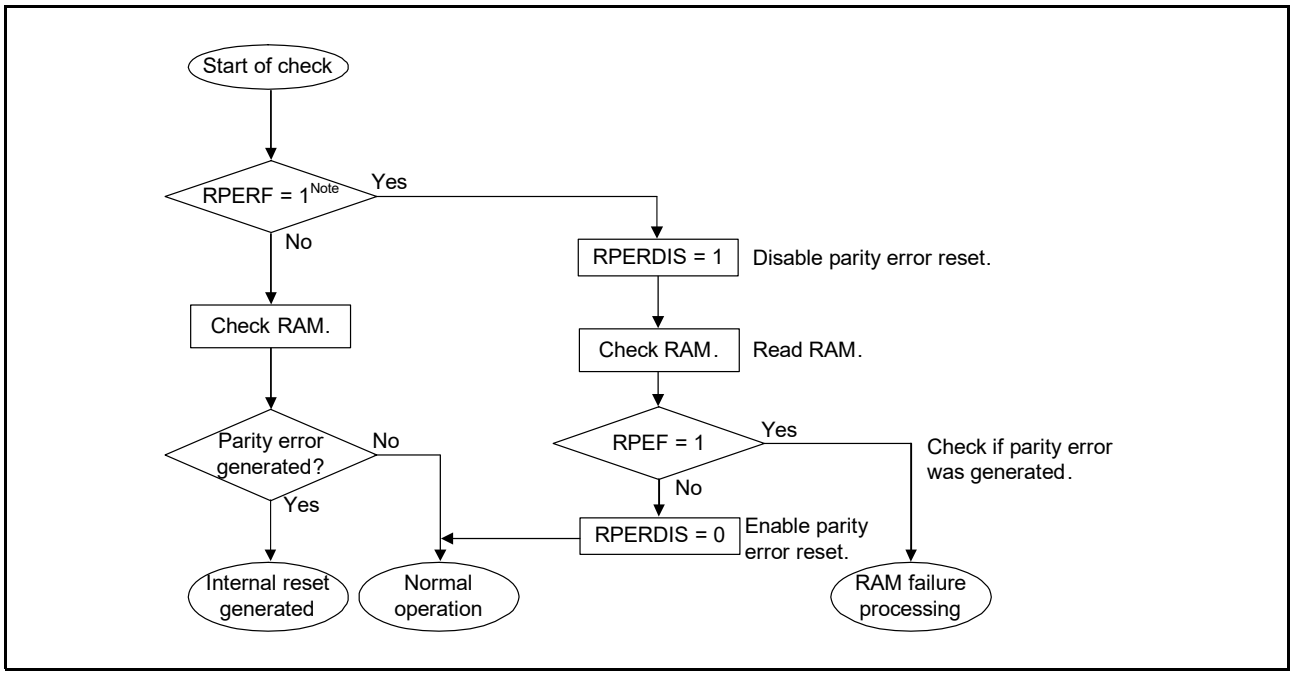
Remark 1. The parity error reset is enabled by default (RPERDIS = 0).

Remark 2. Even if the parity error reset is disabled (RPERDIS = 1), the RPEF flag will be set to 1 if a parity error occurs. If parity error resets are enabled (RPERDIS = 0) with the RPEF flag set to 1, a parity error reset is generated when the RPERDIS bit is set to 0.

Remark 3. The RPEF flag is set to 1 when the RAM parity error occurs and cleared to 0 by writing 0 to it or by any reset source. When the RPEF flag is 1, the value is retained even if RAM for which no parity error has occurred is read.

Remark 4. The general registers are not included for RAM parity error detection.

Figure 27 - 11 Flow for Detecting the RAM Parity Error



Note To check the state following an internal reset triggered by a RAM parity error, see **Section 24 Reset Function**.

27.3.5 RAM guard function

This MCU has functionality to protect data in the specified memory space. Enabling this function disables writing to the specified area of the RAM. Reading from the specified area is possible.

27.3.5.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of illicit memory accesses, and protection of the RAM and SFRs. Use the GRAM1 and GRAM0 bits to protect the RAM. The IAWCTL register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 27 - 12 Format of Invalid Memory Access Detection Control Register (IAWCTL)

Address: F0078H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GRAM1	GRAM0	Protected area in the RAM **Note**
0	0	Disabled. Writing to the RAM is allowed.
0	1	128 bytes from the base address of the RAM
1	0	256 bytes from the base address of the RAM
1	1	512 bytes from the base address of the RAM

Note The base address of the RAM differs depending on the size of the RAM in the product.

Remark The protection of the RAM by this register is only effective against accesses from the CPU, data transfer controller (DTC), and SNOOZE mode sequencer (SMS).

27.3.6 SFR guard function

This MCU provides functionality to protect the data in the control registers for use with the ports, interrupts, clock control, voltage detection, and RAM parity error detection. Enabling this function disables writing to the protected area of the SFRs. Reading from the protected area is possible.

27.3.6.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of illicit memory accesses and the protection of the RAM and SFRs. Use the GPORT, GINT, and GCSC bits to protect the SFRs. The IAWCTL register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 27 - 13 Format of Invalid Memory Access Detection Control Register (IAWCTL)

Address: F0078H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GPORT	Protection of the port control registers
0	Disabled. Reading from and writing to the port control registers are allowed.
1	Enabled. Writing to the port control registers is not allowed. Reading from them is allowed. [Protected SFRs] PMxx, PUxx, PIMxx, POMxx, PMCAxx, PMCTxx, PMCExx, PFOEx, PDIDISxx, CCDE, CCSm, PTDC Note that the port registers (Pxx) are not protected.

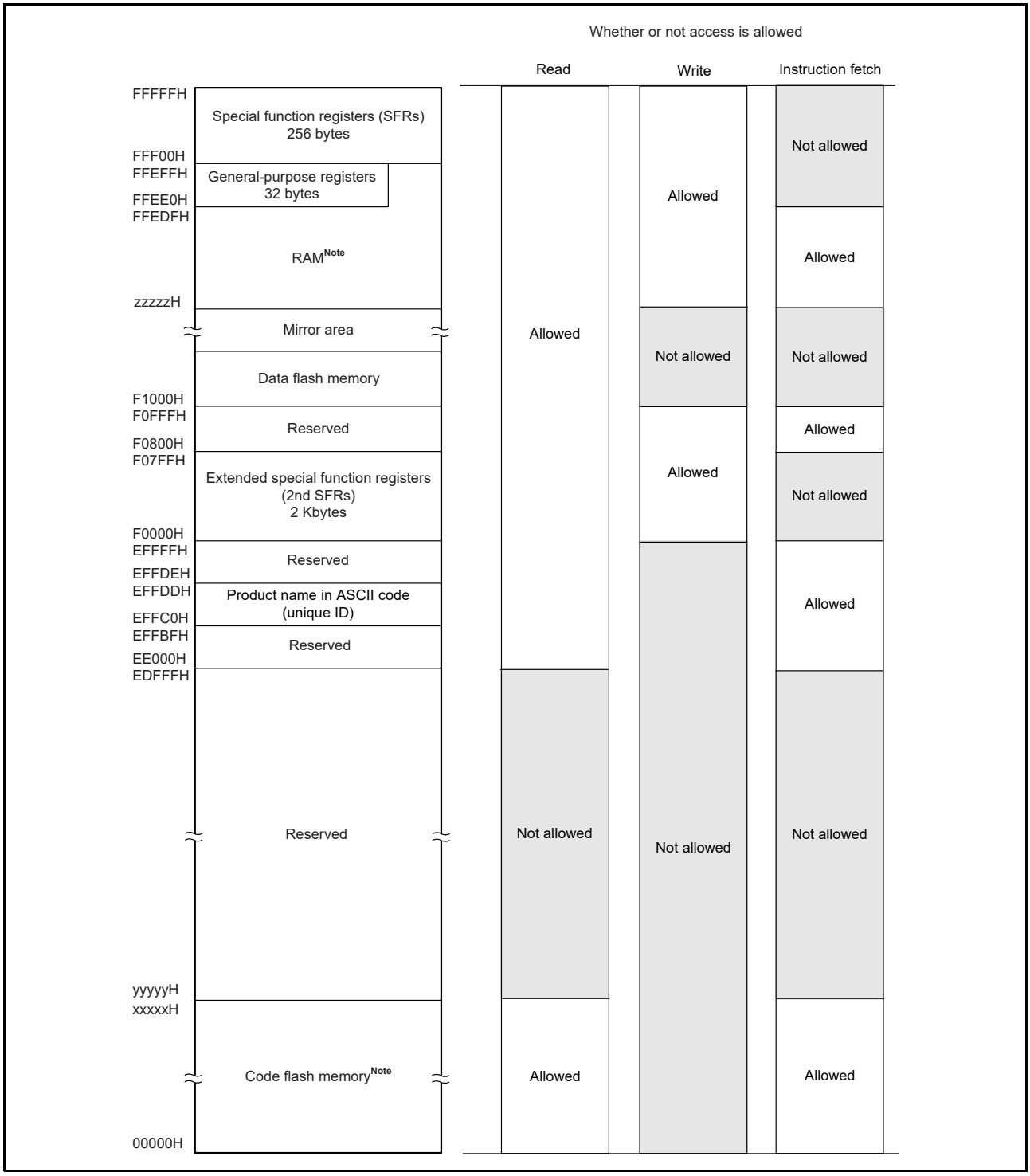
GINT	Protection of the interrupt control registers
0	Disabled. Reading from and writing to the interrupt control registers are allowed.
1	Enabled. Writing to the interrupt control registers is not allowed. Reading from them is allowed. [Protected SFRs] IFxx, MKxx, PRxx, EGPx, EGNx

GCSC	Protection of the clock, voltage detector, and RAM parity error detection control registers
0	Disabled. Reading from and writing to the clock, voltage detector, and RAM parity error detection control registers are allowed.
1	Enabled. Writing to the clock, voltage detector, and RAM parity error detection control registers is not allowed. Reading from them is allowed. [Protected SFRs] CMC, CSC, OSTS, CKC, PERx, OSMC, LVIM, LVIS, RPECTL, CKSEL, PRRx, MOCODIV, WKUPMD, PSMCR

27.3.7 Illicit memory access detection

The IEC60730 standard mandates checking that the CPU and interrupts are operating correctly. This MCU provides functionality to trigger a reset when an invalid memory area is accessed. Access to the areas indicated as “Not allowed” in **Figure 27 - 14** is detected as illicit.

Figure 27 - 14 Illicit Access Areas



(Note is listed on the next page.)

Note The following table lists the capacity and address of the code flash memory and RAM, and the lowest address of the area to be detected as illicit when accessed of each product.

Products	Code flash memory (00000H to xxxxxH)	RAM (zzzzzH to FFEFFH)	Lowest address of the area to be detected as illicit when accessed for reading or instruction fetching (yyyyyH)
R7F100GxF (x = A, B, C, E, F, G, J, L)	98304 × 8 bits (00000H to 17FFFH)	12288 × 8 bits (FCF00H to FFEFFH)	18000H
R7F100GxG (x = A, B, C, E, F, G, J, L, M, P)	131072 × 8 bits (00000H to 1FFFFH)	16384 × 8 bits (FBF00H to FFEFFH)	20000H
R7F104GxG (x = B)			
R7F100GxH (x = A, B, C, E, F, G, J, L, M, P)	196608 × 8 bits (00000H to 2FFFFH)	20480 × 8 bits (FAF00H to FFEFFH)	30000H
R7F100GxJ (x = A, B, C, E, F, G, J, L, M, P, S)	262144 × 8 bits (00000H to 3FFFFH)	24576 × 8 bits (F9F00H to FFEFFH)	40000H
R7F104GxJ (x = F, G)			
R7F100GxK (x = F, G, J, L, M, P, S)	393216 × 8 bits (00000H to 5FFFFH)	32768 × 8 bits (F7F00H to FFEFFH)	60000H
R7F100GxL (x = F, G, J, L, M, P, S)	524288 × 8 bits (00000H to 7FFFFH)	49152 × 8 bits (F3F00H to FFEFFH)	80000H
R7F100GxN (x = F, G, J, L, M, P, S)	786432 × 8 bits (00000H to BFFFFH)	49152 × 8 bits (F3F00H to FFEFFH)	C0000H

27.3.7.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of illicit memory accesses and the protection of the RAM and SFRs. Use the IAWEN bit to protect the SFRs. The IAWCTL register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 27 - 15 Format of Invalid Memory Access Detection Control Register (IAWCTL)

Address: F0078H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

IAWEN[Note]	Control of illicit memory access detection
0	Disables the detection of illicit memory accesses.
1	Enables the detection of illicit memory accesses.

Note Only writing 1 to the IAWEN bit has an effect. Writing 0 to it has no effect after the IAWEN bit has been set to 1.

Remark When the setting of the WDTON bit in the user option byte (000C0H) is 1 (watchdog timer operation enabled), the illicit memory access detection is enabled even if the IAWEN bit is 0.

27.3.8 Guard function of invalid memory access detection control register

To ensure safe operation, the IEC60730 standard requires the setting for enabling or disabling illicit memory access detection to be protected from being rewritten even if a CPU malfunction occurs. This MCU provides functionality to protect the invalid memory access detection control register (IAWCTL) against being rewritten. Enabling the protection of the invalid memory access detection control register disables writing to the given register. Reading from the protected register is possible.

27.3.8.1 Guard register of IAWCTL register (GIAWCTL)

This register is used to protect the setting for enabling or disabling the illicit memory access detection. To allow rewriting of the invalid memory access detection control register (IAWCTL), set the GIAWCTL.GIA bit to 0 to disable protection of the IAWCTL register. The GIAWCTL register can be set by a 16-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 27 - 16 Format of Guard Register of IAWCTL Register (GIAWCTL)

Address: F048EH, F048FH
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
GIAWCTL	RKEY[7:0]							

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	GIA

RKEY[7:0]	Function
Bits 7 to 0	The RKEY[7:0] bits contain the key code to control rewriting of this register. When rewriting the GIA bit, set RKEY[7:0] to A4H. The RKEY[7:0] bits return 00H when read.

GIA	Control of rewriting the IAWCTL register
0	Disables protection of the IAWCTL register (rewriting is allowed).
1	Enables protection of the IAWCTL register (rewriting is not allowed).

27.3.9 Frequency detection

The IEC60730 standard mandates checking that the oscillation frequency is correct. By using the CPU/peripheral hardware clock frequency (f_{CLK}) to measure the pulse width of the input signal to channel 5 of the timer array unit 0 (TAU0), whether the proportional relationship between the two clock frequencies is correct can be determined. Note that, however, if one or both clock operations are completely stopped, the proportional relationship between the clocks cannot be determined.

<Clock frequencies to be compared>

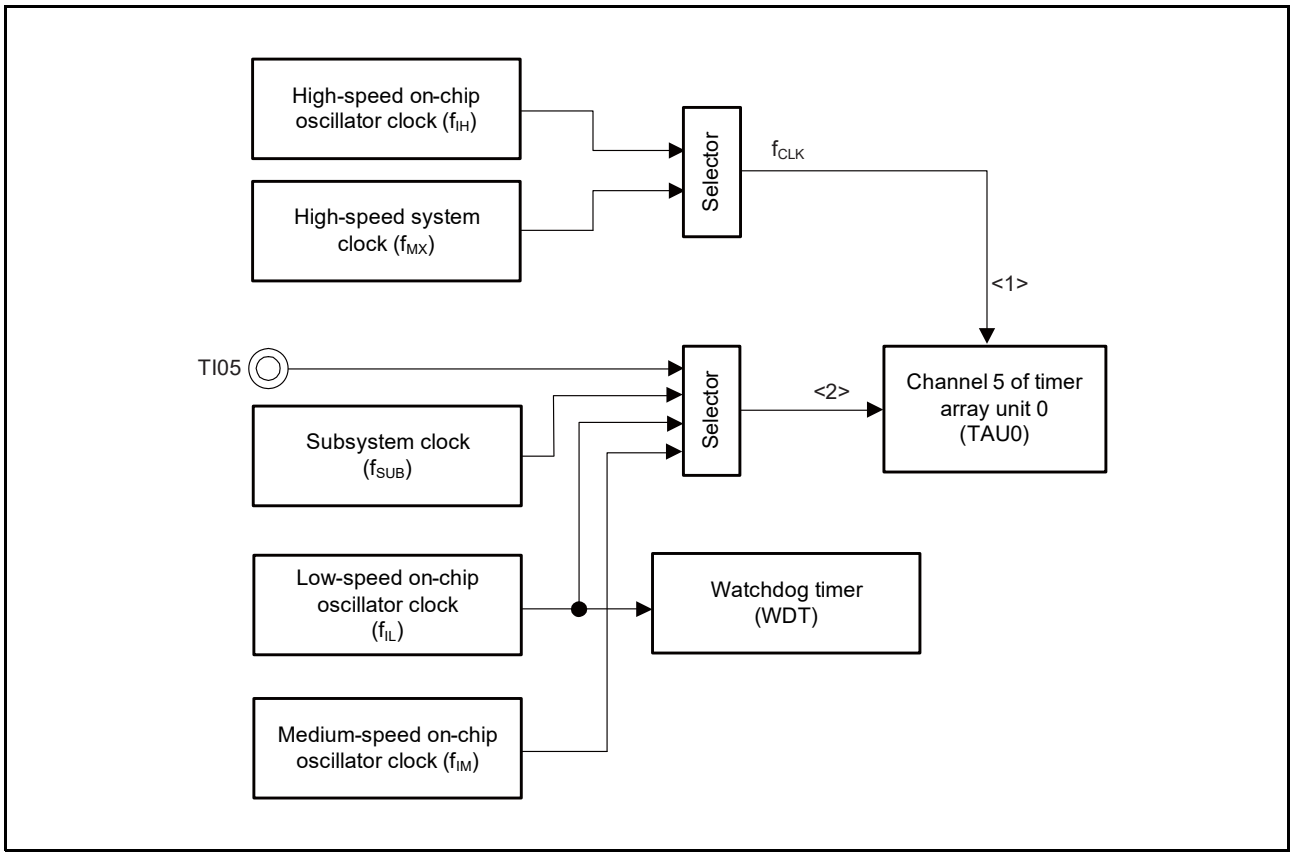
<1> CPU/peripheral hardware clock frequency (f_{CLK}):

- High-speed on-chip oscillator clock (f_{IH})
- High-speed system clock (f_{MX})

<2> Input to channel 5 of the timer array unit 0:

- Timer input to channel 5 (TI05)
- Low-speed on-chip oscillator clock (f_{IL})
- Subsystem clock (f_{SUB})
- Medium-speed on-chip oscillator clock (f_{IM})

Figure 27 - 17 Configuration of Frequency Detection Function



If the results of input pulse interval measurement are abnormal, the clock frequency is considered to be abnormal. For details on the input pulse interval measurement, see **7.8.4 Operation for input pulse interval measurement**.

27.3.9.1 Timer input select register 0 (TIS0)

The TIS0 register is used to select the timer input of channel 5 of the timer array unit 0 (TAU0). The TIS0 register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 27 - 18 Format of Timer Input Select Register 0 (TIS0)

Address: F0074H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	0	0	TIS02	TIS01	TIS00

TIS02	TIS01	TIS00	Selection of timer input used with channel 5
0	0	0	Input signal of timer input pin (TI05)
0	0	1	Event input signal from ELCL
0	1	1	Medium-speed on-chip oscillator clock (f_{IM})
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
1	0	1	Subsystem clock (f_{SUB})
Other than the above			Setting prohibited

27.3.10 Testing of the A/D converter

The IEC60730 standard mandates testing of the A/D converter. This test checks whether or not the A/D converter is operating normally by converting the positive or negative reference voltage for the A/D converter, analog input channels (ANIxx), temperature sensor output voltage, or internal reference voltage. For details on the method of checking, refer to the **application note (R01AN5607) Safety Function (A/D test)**.

Use the following procedure to check the analog multiplexer.

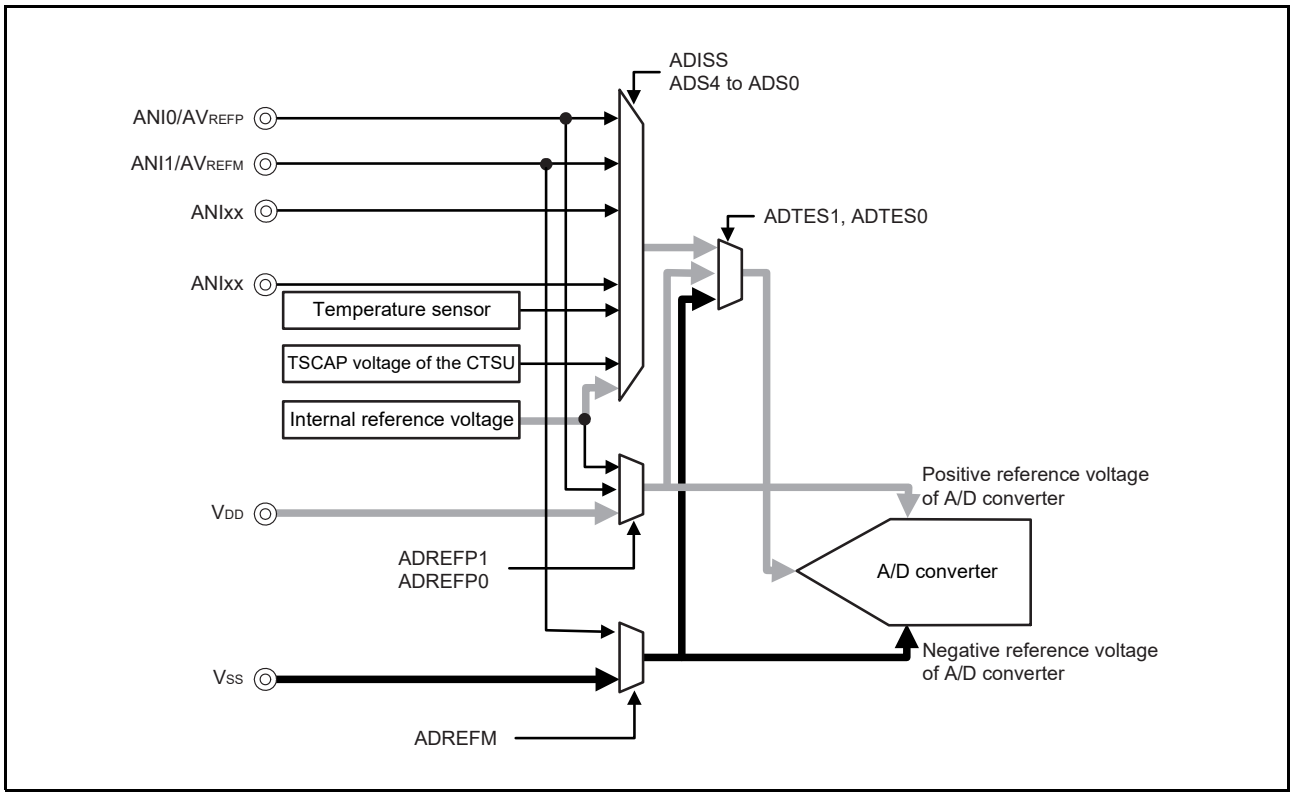
<1> Select the ANIxx pin for A/D conversion using the ADTES register (ADTES1, ADTES0 = 00B).
<2> Perform A/D conversion for the ANIxx pin (conversion result 1-1).
<3> Select the negative reference voltage for A/D conversion using the ADTES register (ADTES1, ADTES0 = 10B)
<4> Perform A/D conversion of the negative reference voltage of the A/D converter (conversion result 2-1).
<5> Select the ANIxx pin for A/D conversion using the ADTES register (ADTES1, ADTES0 = 00B).
<6> Perform A/D conversion for the ANIxx pin (conversion result 1-2).
<7> Select the positive reference voltage for A/D conversion using the ADTES register (ADTES1, ADTES0 = 11B)
<8> Perform A/D conversion of the positive reference voltage of the A/D converter (conversion result 2-2).
<9> Select the ANIxx pin for A/D conversion using the ADTES register (ADTES1, ADTES0 = 00B).
<10> Perform A/D conversion for the ANIxx pin (conversion result 1-3).
<11> Check that the conversion results 1-1, 1-2, and 1-3 are equal.
<12> Check that the A/D conversion result 2-1 is all zero and the A/D conversion result 2-2 is all one.

Using the procedure above can confirm that the analog multiplexer is selected and all wiring is connected.

Remark 1. If the analog input voltage is variable during A/D conversion in steps <1> to <10> above, use another method to check the analog multiplexer.

Remark 2. The results of conversion might include an error. Consider an appropriate level of error in comparison of the results of conversion.

Figure 27 - 19 Configuration of Testing of the A/D Converter



27.3.10.1 A/D test register (ADTES)

This register is used to select the positive or negative reference voltage for the A/D converter, analog input channels (ANIxx), temperature sensor output voltage, internal reference voltage, or TSCAP voltage of the CTSU as the target of A/D conversion. When testing the A/D converter, specify the following settings:

- Select the negative reference voltage as the target of A/D conversion for zero-scale measurement.
- Select the positive reference voltage as the target of A/D conversion for full-scale measurement.

The ADTES register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 27 - 20 Format of A/D Test Register (ADTES)

Address: F0013H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	0	0	ADTES1	ADTES0

ADTES1	ADTES0	A/D conversion target
0	0	ANIxx, temperature sensor output voltage, internal reference voltage[Note], or TSCAP voltage of the CTSU (set by the analog input channel specification register (ADS).)
1	0	Negative reference voltage (selected by the ADREFM bit in the ADM2 register)
1	1	Positive reference voltage (selected by the ADREFP1 or ADREFP0 bit in the ADM2 register)
Other than the above		Setting prohibited

Note For the internal reference voltage, see **Section 37 Electrical Characteristics**.

Caution Be sure to leave bits 7 to 2 set to 0.

27.3.10.2 Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted. Set the A/D test register (ADTES) to 00H when measuring ANIxx, temperature sensor output voltage, or internal reference voltage. The ADS register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 27 - 21 Format of Analog Input Channel Specification Register (ADS)

Address: FFF31H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

Select mode (ADM0.ADMD = 0)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P20/ANI0/AVREFP pin
0	0	0	0	0	1	ANI1	P21/ANI1/AVREFM pin
0	0	0	0	1	0	ANI2	P22/ANI2 pin
0	0	0	0	1	1	ANI3	P23/ANI3 pin
0	0	0	1	0	0	ANI4	P24/ANI4 pin
0	0	0	1	0	1	ANI5	P25/ANI5 pin
0	0	0	1	1	0	ANI6	P26/ANI6 pin
0	0	0	1	1	1	ANI7	P27/ANI7 pin
0	0	1	0	0	0	ANI8	P150/ANI8 pin
0	0	1	0	0	1	ANI9	P151/ANI9 pin
0	0	1	0	1	0	ANI10	P152/ANI10 pin
0	0	1	0	1	1	ANI11	P153/ANI11 pin
0	0	1	1	0	0	ANI12	P154/ANI12 pin
0	0	1	1	0	1	ANI13	P155/ANI13 pin
0	0	1	1	1	0	ANI14	P156/ANI14 pin
0	0	1	1	1	1	Setting prohibited	
0	1	0	0	0	0	ANI16	P03/ANI16 pin**Note 1**
0	1	0	0	0	1	ANI17	P02/ANI17 pin**Note 2**
0	1	0	0	1	0	ANI18	P147/ANI18 pin
0	1	0	0	1	1	ANI19	P120/ANI19 pin
0	1	0	1	0	0	ANI20	P100/ANI20 pin
0	1	0	1	0	1	ANI21	P37/ANI21 pin
0	1	0	1	1	0	ANI22	P36/ANI22 pin
0	1	0	1	1	1	ANI23	P35/ANI23 pin
0	1	1	0	0	0	ANI24	P117/ANI24 pin
0	1	1	0	0	1	ANI25	P116/ANI25 pin
0	1	1	0	1	0	ANI26	P115/ANI26 pin
0	1	1	0	1	1	Setting prohibited	
1	0	0	0	0	0	—	Temperature sensor output voltage
1	0	0	0	0	1	—	Internal reference voltage
Other than the above						Setting prohibited	

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

Note 1. The P01/ANI16 pin is specified in the 30-pin and 32-pin R7F100G products.

Note 2. The P00/ANI17 pin is specified in the 30-pin and 32-pin R7F100G products.

Caution 1. Be sure to leave bits 6 and 5 set to 0.

Caution 2. Select input mode for the ports which are set to analog input with the PMCAxx registers, using the PMxx registers (PM0 to PM3, PM10 to PM12, PM14, and PM15).

Caution 3. Do not use the ADS register to set the pins which should be set as digital I/O with the PMCAxx registers (PMCA0 to PMCA3, PMCA10 to PMCA12, PMCA14, and PMCA15).

Caution 4. Only rewrite the value of the ADISS bit while the conversion operation is stopped (ADCS = 0, ADCE = 0).

Caution 5. If using AV_{REFP} as the positive reference voltage source of the A/D converter, do not select ANI0 as an A/D conversion channel.

Caution 6. If using AV_{REFM} as the negative reference voltage source of the A/D converter, do not select ANI1 as an A/D conversion channel.

Caution 7. When the ADISS bit is 1, the internal reference voltage cannot be used for the positive reference voltage. In addition, the first conversion result obtained after setting the ADISS bit to 1 is not available. For a detailed setting flow, see 12.7.5 Example of settings when temperature sensor output voltage, internal reference voltage, or TSCAP voltage of the CTSU is selected in the software trigger no-wait mode and one-shot conversion mode.

Caution 8. If a transition is made to STOP mode or a transition is made to HALT mode during CPU operation with subsystem clock, do not set the ADISS bit to 1. When the ADISS bit is 1, the A/D converter reference voltage current (I_{ADREF}) shown in 37.3.2 Characteristics of the supply current is added.

27.3.11 Detection of the digital output signal level of the I/O pins

The IEC60730 standard mandates checking that the I/O function is operating correctly. This MCU provides functionality to read the output level of an I/O pin when the pin is in the output mode.

27.3.11.1 Port mode select register (PMS)

This register is used to select whether to read the output latch value of a port or read the output level of a pin when the pin is in the output mode (PMmn bit of port mode register (PMm) is 0). The PMS register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 27 - 22 Format of Port Mode Select Register (PMS)

Address: F007BH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
PMS	0	0	0	0	0	0	0	PMS0

PMS0	Selection of data to be read when pin is in output mode
0	Reads the value of the Pmn register.
1	Reads the digital output level of the pin.

Remark m = 0 to 12, 14, 15
n = 0 to 7

Caution 1. Be sure to leave bits 7 to 1 set to 0.

Caution 2. Do not rewrite a port register (Pxx) with a 1-bit memory manipulation instruction when the setting of the PMS0 bit in the PMS register is 1. Use an 8-bit memory manipulation instruction to rewrite a port register (Pxx).

Caution 3. The setting of this register has no effect on the input port pins (P123, P124, and P137) and output port pin (P130).

Caution 4. The setting of this register has no effect when the settings of the PMCAmn and PMCTmn bits are both 1. To read the digital output level of a pin with the PMCAmn bit and PMCTmn bit both set to 1, first leave the PMCAmn and PMCTmn bits set to 0 and then set the PMS0 bit to 1.

Remark m = 0 to 3, 5 to 7, 10 to 12, 14, 15
n = 0 to 7

Caution 5. The setting of this register has no effect when the setting of the PDIDISmn bit is 1. To read the digital output level of a pin with the PDIDISmn bit set to 1, first leave the PDIDISmn bit set to 0 and then set the PMS0 bit to 1.

Remark m = 0, 1, 3 to 5, 7 to 9, 12 to 14
n = 0 to 7

27.3.12 UART loopback

The IEC60730 standard recommends to diagnose abnormalities of external interfaces (communications). The UART loopback is used to confirm the normal output of the UART transmit data by shutting off the RxD pin from the outside and connecting it within the MCU to loop back the output from the transmission shift register to the reception shift register.

When the UART loopback is selected, the transmit data from the TxD pin is controlled by the port functions so that it does not affect its communication partner.

- Communication with negative logic
  The port pin multiplexed with the TxD pin is set to the input mode (PMxx = 1) and an on-chip pull-up resistor is connected (PUxx = 1) to retain the setting of 1. Retaining of 1 on the TxDA0 pin is possible by setting the PFOE14 bit in the PFOE1 register to 0 while 1 is being output from the TxDA0 pin (PMxx = 0, Pxx = 1). Retaining of 1 on the TxD0 pin is possible by setting the PFOE10 bit in the PFOE1 register to 0 while 1 is being output from the TxD0 pin (PMxx = 0, Pxx = 1).
- Communication with positive logic
  0 (PMxx = 0, Pxx = 0) is output from the port that also uses the TxD pin.

27.3.12.1 UART loopback select register (ULBS)

The ULBS register is used to enable the UART loopback. This register has respective bits for independently controlling each UART channel. Setting the bit corresponding to each channel to 1 will select the UART loopback and loop back the output of the transmission shift register to the reception shift register. The ULBS register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 27 - 23 Format of UART Loopback Select Register (ULBS)

Address: F0079H
After reset: 00H
R/W: R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
ULBS	0	0	ULBS5	ULBS4	ULBS3	ULBS2	ULBS1	ULBS0

ULBS5	UART loopback selection
0	Inputs the state of the RxDA1 pin of serial interface UARTA1 to the reception shift register.
1	Loops back the output of the transmission shift register to the reception shift register.

ULBS4	UART loopback selection
0	Inputs the state of the RxDA0 pin of serial interface UARTA0 to the reception shift register.
1	Loops back the output of the transmission shift register to the reception shift register.

ULBS3	UART loopback selection
0	Inputs the state of the RxD3 pin of serial array unit UART3 to the reception shift register.
1	Loops back the output of the transmission shift register to the reception shift register.

ULBS2	UART loopback selection
0	Inputs the state of the RxD2 pin of serial array unit UART2 to the reception shift register.
1	Loops back the output of the transmission shift register to the reception shift register.

ULBS1	UART loopback selection
0	Inputs the state of the RxD1 pin of serial array unit UART1 to the reception shift register.
1	Loops back the output of the transmission shift register to the reception shift register.

ULBS0	UART loopback selection
0	Inputs the state of the RxD0 pin of serial array unit UART0 to the reception shift register.
1	Loops back the output of the transmission shift register to the reception shift register.

Caution **Be sure to leave bits 7 and 6 set to 0.**

Remark The RxDA0 pin is only available in the 36- to 128-pin R7F100G and 32- to 48-pin R7F104G products.
The RxDA1 is only available in the 44- to 128-pin R7F100G products.

Section 28 Security Functions

28.1 True Random Number Generator (TRNG)

28.1.1 Function of the true random number generator

The true random number generator generates 32-bit random number seeds.

28.1.2 Registers for controlling the true random number generator

The following registers are used to control the true random number generator.

- Random number seed command register 0 (TRNGSCR0)
- Random number seed data register (TRNGSDR)

28.1.2.1 Random number seed command register 0 (TRNGSCR0)

The TRNGSCR0 register controls operation of the true random number generator. Setting the TRNGST bit to 1 after having set the TRNGEN bit to 1 starts the generation of a random number seed. When the true random number generator finishes generating the random number seed, the TRNGRDY flag is set to 1. Since the TRNGST bit serves as the trigger for starting the generation of a random number seed, it is cleared to 0 immediately after 1 having been written to it. The TRNGSCR0 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 28 - 1 Format of Random Number Seed Command Register 0 (TRNGSCR0)

Address: F0542H
After reset: 00H
R/W: R/W

Symbol	<7>	6	5	4	<3>	<2>	1	0
TRNGSCR0	TRNGRDY	0	0	0	TRNGEN	TRNGST	0	0

TRNGRDY	Random number seed generation status flag
0	A random number seed has not been generated or four rounds of reading from the TRNGSDR register have been completed.
1	A random number seed has been generated.

TRNGEN	Control over operation of the true random number generator
0	Stops the true random number generator.
1	Enables the true random number generator.

TRNGST	Trigger to start generating a random number seed
0	The trigger is inactive.
1	Starts generation of a random number seed.

28.1.2.2 Random number seed data register (TRNGSDR)

The TRNGSDR register is an 8-bit register that holds the bytes of random number seeds generated by the true random number generator. The random number seed can be read from this register after the TRNGRDY flag is set to 1. As a random number seed consists of 32 bits, four rounds of access to the register are required for each seed.
The TRNGRDY flag of the TRNGSCR0 register is cleared to 0 following the four rounds of access. The TRNGSDR register can be read by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 28 - 2 Format of Random Number Seed Data Register (TRNGSDR)

Address: F0540H
After reset: 00H
R/W: R

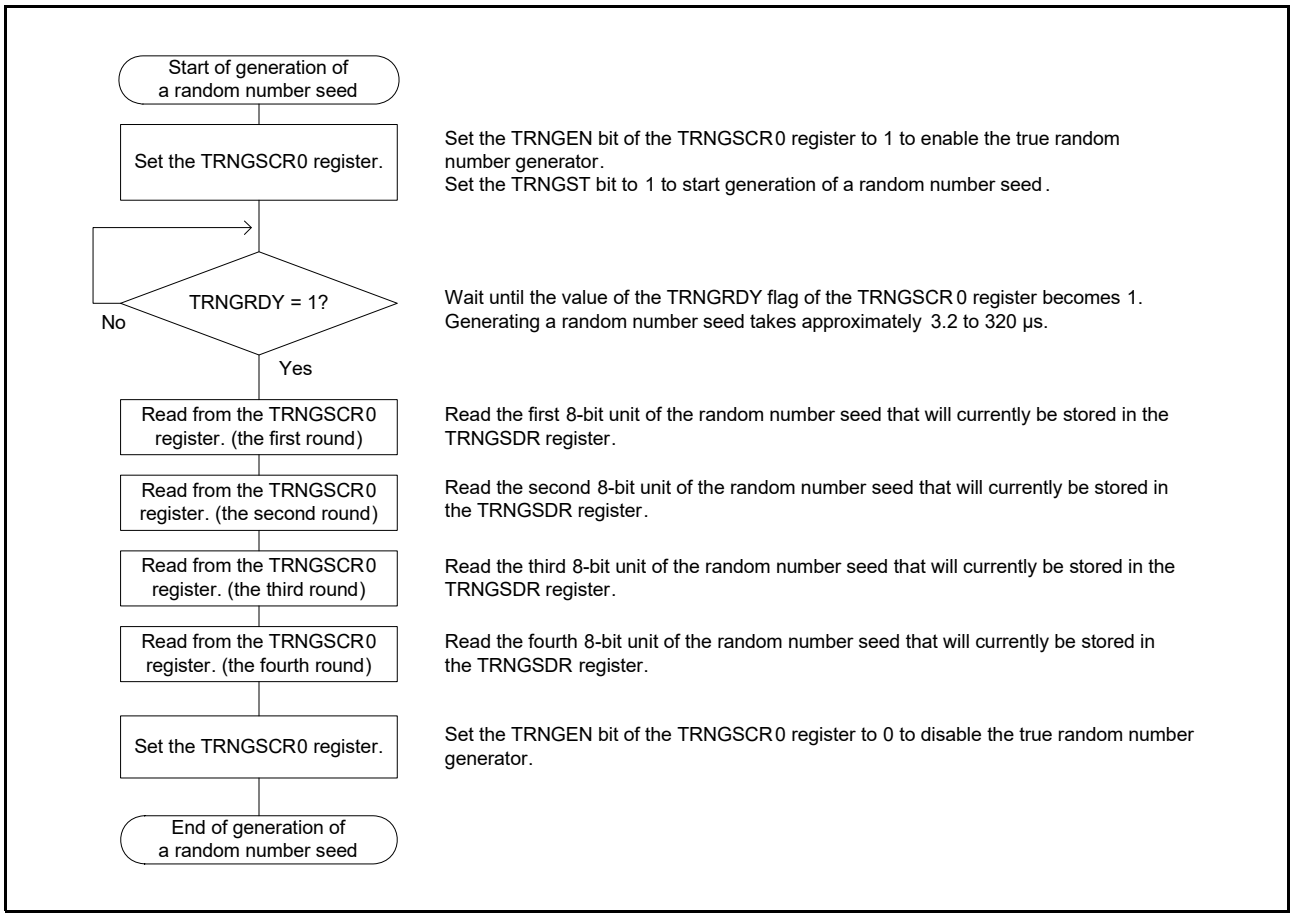
Symbol	7	6	5	4	3	2	1	0
TRNGSDR								

Caution When the value of the TRNGRDY bit is 0, the value in the TRNGSDR register is 00H.

28.1.3 Operations of the true random number generator

Figure 28 - 3 shows the procedure for using the true random number generator to generate a random number seed.

Figure 28 - 3 Procedure for Using the True Random Number Generator to Generate a Random Number Seed



28.2 Flash Read Protection

28.2.1 Function of flash read protection

The flash read protection can be used to protect a specified range of the code flash memory area against read access by the CPU, DTC, or SMS. Note that fetching of instructions in the specified range by the CPU is still possible.
In addition, verifying the code flash memory area protected against read access by the flash read protection is possible in the serial programming mode.

28.2.2 Setting of flash read protection

The settings for flash read protection are made through serial programming by using a flash memory programmer or through self-programming in the extra area. Read access to the whole range of the code flash memory is enabled by the default setting at the time of shipment. Set a range in the code flash memory to be protected against read access by specifying the flash read protection start block and end block, and disable changing of the flash read protection settings. This makes read access to addresses in the specified range of blocks in the code flash memory impossible. In addition, changing of the flash read protection settings can be disabled, which makes the settings for the start and end blocks fixed, thus making changes to the blocks where flash read protection starts and ends impossible. However, disabling changing of the flash read protection settings can be released by using a flash memory programmer. Note that release from disabling of changes to the flash read protection settings requires deletion of the contents of the full range of the code flash memory. Accordingly, reading the data written in the range protected against read access is impossible after release from disabling of changes to the flash read protection settings. **Table 28 - 1** describes the settings for flash read protection and their functions. **Table 28 - 2** describes the method of setting flash read protection.

Table 28 - 1 Settings for Flash Read Protection and their Functions

Item to Be Set	Function
Block where flash read protection starts	Specifies the number of the block where the read-access disabled area starts. The specifiable values are in the range from the block numbered 001H to the number of the block at the highest-order address in the code flash memory. The block for which the number is set as the block where protection starts is part of the read-access disabled area. Setting 000H as the block where protection starts is prohibited. The initial setting is 1FFH.
Block where flash read protection ends	Specifies the number of the block where the read-access disabled area ends. The specifiable values are in the range from the number of the block where the protection starts to the number of the block at the highest-order address in the code flash memory. The block for which the number is set as the block where protection ends is part of the read-access disabled area. The initial setting is 1FFH.
Disabling changing of the flash read protection settings	Fixes the settings of the blocks where flash read protection starts and ends. When the “enabled” setting is made, changes to the blocks where protection starts and ends are not possible.

For details on the relationship between the addresses and block numbers, refer to **Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory**.

Table 28 - 2 Method of Setting Flash Read Protection

Item to Be Set	Method of Setting	Method of Changing
Block where flash read protection starts	Using a flash memory programmer or self-programming.	Using a flash memory programmer or self-programming. Note that the block where protection starts is not adjustable while changing of the flash read protection settings is disabled.
Block where flash read protection ends	Using a flash memory programmer or self-programming.	Using a flash memory programmer or self-programming. Note that the block where protection ends is not adjustable while changing of the flash read protection settings is disabled.
Disabling changing of the flash read protection settings	Using a flash memory programmer or self-programming.	Fixing of the flash read protection settings can be released by using a flash memory programmer.Note If you do so, the values for the start and end blocks are initialized.

Note Release from the fixed setting is only possible when erasure of blocks is not prohibited, rewriting of boot area is not prohibited, and the code and data flash memory areas are blank.

Caution 1. The settings for flash read protection in the extra area are not readable. To confirm that the settings for flash read protection are in place, read from the read-access disabled area and confirm that FFH is returned.

Caution 2. To specify the read-access disabled area for flash read protection, be sure to specify the numbers of both the block where protection starts and the block where it ends.

Caution 3. Reading from the read-access disabled area by using an on-chip debugger is also impossible. This means that program code allocated to the read access-disabled area cannot be debugged by using the on-chip debugger. Therefore, only make the settings for flash read protection after having debugged the program code in the protected areas.

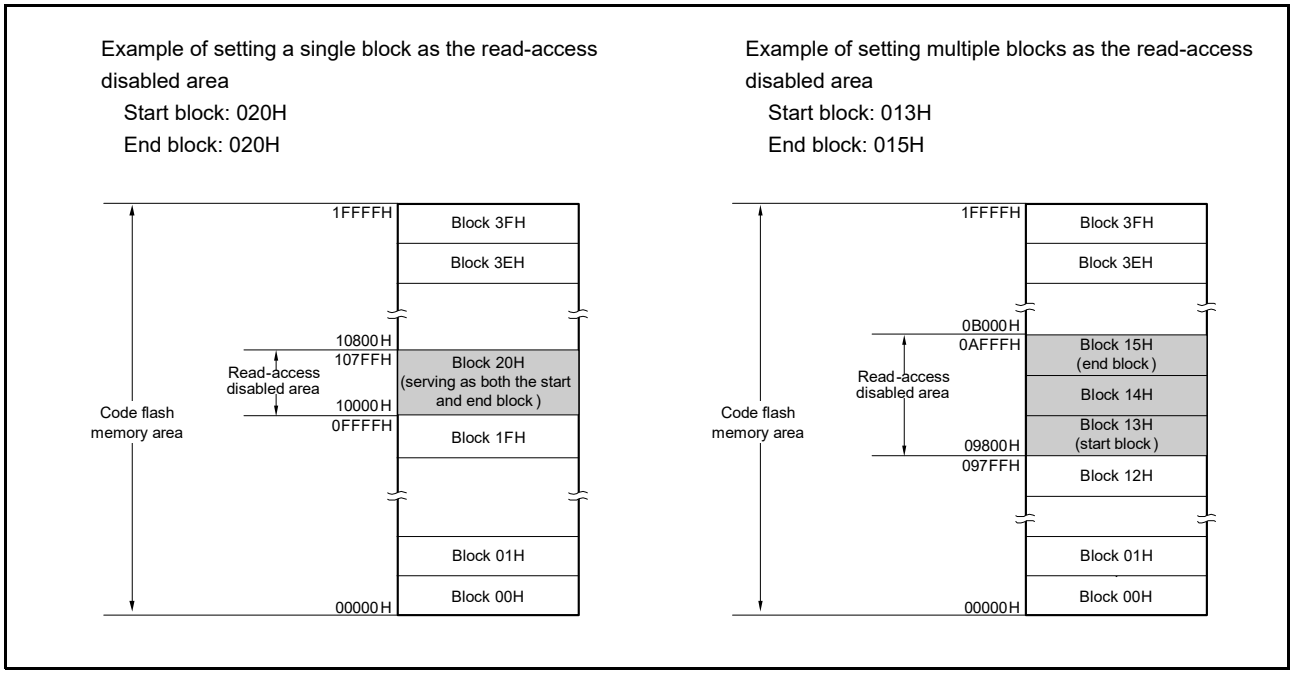
Caution 4. When a part of boot cluster 0 or boot cluster 1 is to be set as a part of the read-access disabled area, boot swapping may cause data in the read-access disabled area to be swapped with data in the read access-enabled area. To prevent this, when setting a part of boot cluster 0 or boot cluster 1 as part of the read-access disabled area, make the setting for prohibiting the rewriting of boot area so as to prohibit boot swapping itself.

Caution 5. When settings for flash read protection have been made through self-programming, the settings become enabled after the MCU is reset and then released from the reset state.

28.2.3 Operation

Reading by the CPU, DTC, or SMS from the area to which read access has been disabled with the use of flash read protection always returns FFH. Fetching of instructions in the read-access disabled area by the CPU is still possible. Note that even if program code is to be executed from the read-access disabled area, it is unable to read data that have been placed in the read-access disabled area. Place data for use with code to be executed from the read-access disabled area in areas that are not protected. **Figure 28 - 4** shows examples of setting read-access disabled areas for the application of flash read protection.

Figure 28 - 4 Examples of Setting Read-access Disabled Areas for the Application of Flash Read Protection



To enable read access to the full range of the code flash memory again after having set a range to be protected against read access, set the flash read protection start block and end block as 1FFH while prohibition of changing the flash read protection settings is disabled.

28.3 Unique ID

28.3.1 Function of a unique ID

A unique ID is a unique value that is allocated to an individual product and stored in the extra area. A unique ID is entered for each product at the time of manufacturing the MCUs. Changing the entered ID is not possible. The data length is 16 bytes (128 bits).

Caution The value of a unique ID is not a random number.

28.3.2 ASCII codes representing the product name

Product names are also stored as strings of ASCII codes in the extra area. Since the product names include indicators of the number of pins and capacity of the flash memory for the given product, conditional branch processing on the basis of the product name is possible. **Table 28 - 3** shows the places of the unique ID, basic product name, and indicators of the number of pins and capacity of the flash memory in the memory map.

Table 28 - 3 Memory Map of the Unique ID and Product Name

Address	Item Name	Value to Be Entered	
		R7F100G	R7F104G
EFFC0H to EFFCFH	Unique ID	The unique value allocated to an individual product	
EFFD5H	ASCII codes representing the product name	52H: “R”	
EFFD6H		37H: “7”	
EFFD7H		46H: “F”	
EFFD8H		31H: “1”	
EFFD9H		30H: “0”	
EFFDAH		30H: “0”	34H: “4”
EFFDBH		47H: “G”	
EFFDCH		41H: “A” (30 pins) 42H: “B” (32 pins) 43H: “C” (36 pins) 45H: “E” (40 pins) 46H: “F” (44 pins) 47H: “G” (48 pins) 4AH: “J” (52 pins) 4CH: “L” (64 pins) 4DH: “M” (80 pins) 50H: “P” (100 pins) 53H: “S” (128 pins)	42H: “B” (30 pins) 46H: “F” (44 pins) 47H: “G” (48 pins)
EFFDDH		46H: “F” (96-Kbyte flash memory) 47H: “G” (128-Kbyte flash memory) 48H: “H” (192-Kbyte flash memory) 4AH: “J” (256-Kbyte flash memory) 4BH: “K” (384-Kbyte flash memory) 4CH: “L” (512-Kbyte flash memory) 4EH: “N” (768-Kbyte flash memory)	47H: “G” (128-Kbyte flash memory) 4AH: “J” (256-Kbyte flash memory)

Section 29 SNOOZE Mode Sequencer (SMS)

The SNOOZE mode sequencer sequentially handles 32 processes that have been set in advance. This function can be started by a signal from another peripheral function as the trigger even in the standby state. This means the sequencer can run through the processes independently of the CPU. Doing so with the CPU placed in the standby state reduces power consumption.

29.1 Functions of the SNOOZE Mode Sequencer

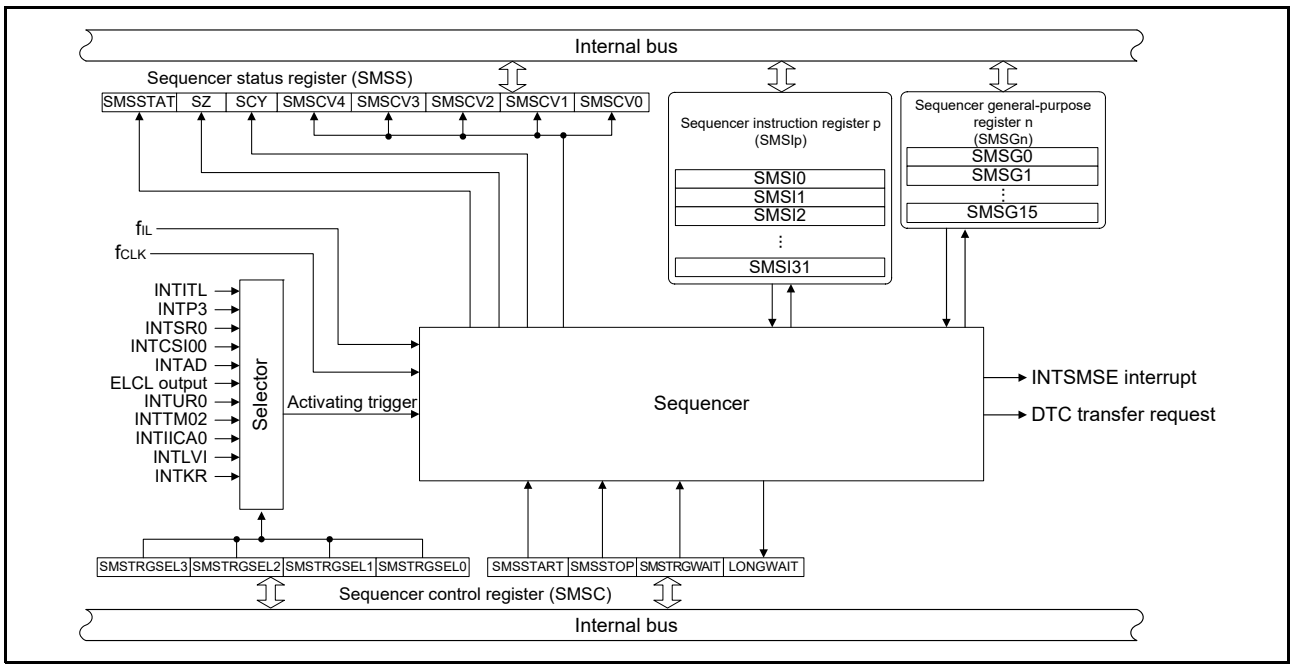
The SNOOZE mode sequencer can handle processing independently of the CPU and has the following functions.

- Sequentially handling a total of 32 processes with the use of desired commands from among 20 different ones
- Able to operate even if the CPU is in the standby state
- Waking the CPU up from the standby state
- Directly starting the data transfer controller (DTC) up
- Access to the RAM and special function register (SFR) area for the peripheral functions
- 16-bit addition and subtraction
- Branch processing
- Automatically switching the source clock for use in waiting to the low-speed on-chip oscillator clock to ensure long enough wait times

29.2 Configuration of the SNOOZE Mode Sequencer

Figure 29 - 1 is a block diagram of the SNOOZE mode sequencer.

Figure 29 - 1 Block Diagram of the SNOOZE Mode Sequencer



Remark p = 0 to 31, n = 0 to 15

29.3 Registers for Controlling the SNOOZE Mode Sequencer

The following registers are used to control the SNOOZE mode sequencer.

- Peripheral enable register 1 (PER1)
- Peripheral reset control register 1 (PRR1)
- Sequencer instruction registers p (SMSIp)
- Sequencer general-purpose registers n (SMSGn)
- Sequencer control register (SMSC)
- Sequencer status register (SMSS)

29.3.1 Peripheral enable register 1 (PER1)

This register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. If the SNOOZE mode sequencer is to be used, be sure to set the SMSEN bit of this register to 1. The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 29 - 2 Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER1	DACEN	SMSEN	CMPEN	TML32EN	DTCEN	UTAEN **Note 1**	REMCEN **Note 2**	CTSUEN

SMSEN	Control of supply of an input clock to the SNOOZE mode sequencer
0	Stops supply of an input clock. • The SFRs used by the SNOOZE mode sequencer cannot be written.**Note 3**
1	Enables supply of an input clock. • The SFRs used by the SNOOZE mode sequencer can be read and written.

Note 1. This bit is only present in the 36- to 128-pin R7F100G and 32- to 48-pin R7F104G products.
Note 2. This bit is only present in the 32- to 128-pin R7F100G and 32- to 48-pin R7F104G products.
Note 3. When an SFR used by the SNOOZE mode sequencer is read, the value returned is 00H or 0000H regardless of the setting value.

Caution 1. Be sure to leave the following bits set to 0.
Bits 2 and 1 in the 30-pin R7F100G products
Bit 2 in the 32-pin R7F100G products
Caution 2. Do not change the value of the SMSEN bit from 1 to 0 while the SNOOZE mode sequencer is operating.

29.3.2 Peripheral reset control register 1 (PRR1)

The PRR1 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. Set the SMSRES bit of this register to 1 to reset the SNOOZE mode sequencer. The PRR1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 29 - 3 Format of Peripheral Reset Control Register 1 (PRR1)

Address: F00FBH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	3	2	<1>	<0>
PRR1	DACRES	SMSRES	CMPRES	TML32RES	0	0	REMCRES **Note**	CTSURES

SMSRES	Control resetting of the SNOOZE mode sequencer
0	The SNOOZE mode sequencer is released from the reset state.
1	The SNOOZE mode sequencer is in the reset state. • The SFRs for use with the SNOOZE mode sequencer are initialized.

Note This bit is only present in the 32- to 128-pin R7F100G and 32- to 48-pin R7F104G products.

Caution 1. Be sure to leave bits 3 and 2 set to 0.
Caution 2. Be sure to leave bit 1 in the 30-pin R7F100G products set to 0.

29.3.3 Sequencer instruction registers p (SMSIp)

Each SMSIp register is a 16-bit register for holding a command to be handled by the SNOOZE mode sequencer. The commands are stored in the total of 32 registers from SMSI0 to SMSI31. The sequencer sequentially handles the commands in order from the SMSI0 register. The contents of each register are referred to as a sequencer code, which consists of an indicator of the processing, a first operand, a second operand, and an additional byte or nybble, the use of which depends on the sequencer code. For details on how the sequencer handles each form of processing, see **29.5 Commands for Use in Processing by the Sequencer**. In addition, after executing processing specified by an SMSIp register, the sequencer executes the processing specified either by the next SMSIp register or by the SMSIp register at a branch destination. The SMSIp register currently in use can be confirmed by reading the SMSCV4 to SMSCV0 bits of the sequencer status register (SMSS). **Table 29 - 1** shows the correspondences between the memory addresses of the SMSIp registers and values of the SMSCV4 to SMSCV0 bits. **Table 29 - 2** lists the types of processing specified by the SMSIp registers. The SMSIp registers can be set by a 16-bit memory manipulation instruction. The value of each SMSIp register following a reset is 0000H.

Figure 29 - 4 Format of Sequencer Instruction Registers p (SMSIp)

Address: F0380H, F0381H (SMSI0) to F03BEH, F03BFH (SMSI31)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SMSIp	Sequencer code				First operand			

7	6	5	4	3	2	1	0
Second operand				Additional byte			

Remark p = 0 to 31

Table 29 - 1 Correspondences between the Memory Addresses of the SMSIp Registers and Values of the SMSCV4 to SMSCV0 Bits

SMSIp	Address	SMSCV4 to SMSCV0
SMSI15	F039EH, F039FH	01111B
SMSI14	F039CH, F039DH	01110B
SMSI13	F039AH, F039BH	01101B
SMSI12	F0398H, F0399H	01100B
SMSI11	F0396H, F0397H	01011B
SMSI10	F0394H, F0395H	01010B
SMSI9	F0392H, F0393H	01001B
SMSI8	F0390H, F0391H	01000B
SMSI7	F038EH, F038FH	00111B
SMSI6	F038CH, F038DH	00110B
SMSI5	F038AH, F038BH	00101B
SMSI4	F0388H, F0389H	00100B
SMSI3	F0386H, F0387H	00011B
SMSI2	F0384H, F0385H	00010B
SMSI1	F0382H, F0383H	00001B
SMSI0	F0380H, F0381H	00000B

SMSIp	Address	SMSCV4 to SMSCV0
SMSI31	F03BEH, F03BFH	11111B
SMSI30	F03BCH, F03BDH	11110B
SMSI29	F03BAH, F03BBH	11101B
SMSI28	F03B8H, F03B9H	11100B
SMSI27	F03B6H, F03B7H	11011B
SMSI26	F03B4H, F03B5H	11010B
SMSI25	F03B2H, F03B3H	11001B
SMSI24	F03B0H, F03B1H	11000B
SMSI23	F03AEH, F03AFH	10111B
SMSI22	F03ACH, F03ADH	10110B
SMSI21	F03AAH, F03ABH	10101B
SMSI20	F03A8H, F03A9H	10100B
SMSI19	F03A6H, F03A7H	10011B
SMSI18	F03A4H, F03A5H	10010B
SMSI17	F03A2H, F03A3H	10001B
SMSI16	F03A0H, F03A1H	10000B

Caution 1. Only set the SMSIp registers while the operation of the sequencer is stopped. Re-writing the SMSIp registers while the sequencer is handling the commands results in an undefined operation of the sequencer.

Caution 2. No register follows the SMSI31 register once the processing it defines has finished. If the command for terminating processing of commands or branch processing is not set in the SMSI31 register, the processing for termination is automatically executed once the processing defined in the SMSI31 register has finished.

Table 29 - 2 Types of Processing Specified by the SMSIp Registers

Name of Processing	Operation[Note 1]	Sequencer Code	First Operand (4 Bits)	Second Operand (4 Bits)	Additional Byte (4 Bits)
8-bit data transfer 1	[SMSGn + byte] ← SMSGm	0000	n of SMSGn Note 2	m of SMSGm Note 2	Byte[Note 3]
8-bit data transfer 2	SMSGm ← [SMSGn + byte]	0001	n of SMSGn Note 2	m of SMSGm Note 2	Byte[Note 3]
16-bit data transfer 1	[SMSGn + byte] ← SMSGm	0010	n of SMSGn Note 2	m of SMSGm Note 2	Byte[Note 3]
16-bit data transfer 2	SMSGm ← [SMSGn + byte]	0011	n of SMSGn Note 2	m of SMSGm Note 2	Byte[Note 3]
1-bit data setting	[SMSGn + byte].bit ← 1	0100	n of SMSGn Note 2	bit[Note 3]	Byte[Note 3]
1-bit data clearing	[SMSGn + byte].bit ← 0	0101	n of SMSGn Note 2	bit[Note 3]	Byte[Note 3]
1-bit data transfer	SCY ← [SMSGn + byte].bit	0110	n of SMSGn Note 2	bit[Note 3]	Byte[Note 3]
Word addition	SMSGn, SCY ← SMSGn + SMSGm	0111	n of SMSGn Note 2	m of SMSGm Note 2	0000
Word subtraction	SMSGn, SCY ← SMSGn - SMSGm	0111	n of SMSGn Note 2	m of SMSGm Note 2	0001
Word comparison	SMSGn - SMSGm	0111	n of SMSGn Note 2	m of SMSGm Note 2	0010
Logical shift right	SCY ← SMSGn[0], SMSGm.15 ← 0, SMSGn.m-1 ← SMSGn.m	0111	n of SMSGn Note 2	0000	0011
Branch 1 (SCY = 1)	SMSS[4:0] ← SMSS[4:0] + jdisp8 if SCY = 1	1000	$addr5[Note 4]		0000
Branch 2 (SCY = 0)	SMSS[4:0] ← SMSS[4:0] + jdisp8 if SCY = 0	1000	$addr5[Note 4]		0001
Branch 3 (SZ = 1)	SMSS[4:0] ← SMSS[4:0] + jdisp8 if SZ = 1	1000	$addr5[Note 4]		0010
Branch 4 (SZ = 0)	SMSS[4:0] ← SMSS[4:0] + jdisp8 if SZ = 0	1000	$addr5[Note 4]		0011
Wait	Holding processing pending for a certain period	1001	IM1		IM2
Conditional wait 1 (bit = 1)	SMSS[4:0] ← SMSS[4:0] if [SMSGn + byte].bit = 1	1010	n of SMSGn Note 2	Bit[Note 3]	Byte[Note 3]
Conditional wait 2 (bit = 0)	SMSS[4:0] ← SMSS[4:0] if [SMSGn + byte].bit = 0	1011	n of SMSGn Note 2	Bit[Note 3]	Byte[Note 3]
Termination	SMSSTAT ← 0, SMSS[4:0] ← 0, Stopping the sequencer	1111	0000	0000	0000
DTC activation	Output of a DTC activating source signal	1111	0000	0000	0010

Note 1. For details on the terms, see **29.5 Commands for Use in Processing by the Sequencer**.

Note 2. Specify values in the range from 0 to 15 (from 0000B to 1111B) for n and m.

Note 3. Specify values in the range from 0 to 7 (from 0000B to 0111B) for the bytes.

Note 4. This is an 8-bit displacement value. Specify a relative address in the ranges from -31 to -1 and 1 to 31 (0000 0001B to 0001 1111B, 1111 1111B to 1110 0001B).

29.3.4 Sequencer general-purpose registers n (SMSGn)

Each SMSGn register is a 16-bit general-purpose register for use with the sequencer and is used for the storage of data, judgement values for use in calculations, source and destination addresses for data manipulation, and so on. Note that the SMSG0 and SMSG15 registers have the fixed values 0000H and FFFFH, respectively. The sequencer has direct access to the SMSGn registers for several types of processing. In such cases, use the suffix n in SMSGn as the internal data addresses to specify the desired SMSGn registers for the sequencer. The SMSG0 to SMSG15 registers can be read by a 16-bit memory manipulation instruction. The SMSG1 to SMSG14 registers can be set by a 16-bit memory manipulation instruction. The value of each of the SMSG1 to SMSG14 registers following a reset is 0000H. The SMSG0 and SMSG15 registers have the fixed values 0000H and FFFFH, respectively.

Remark n = 0 to 15

Figure 29 - 5 Format of Sequencer General-purpose Registers n (SMSGn)

Address: F03C0H, F03C1H
After reset: 0000H
R/W: R

Symbol	15	14	13	12	11	10	9	8
SMSG0	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

Address: F03C2H, F03C3H (SMSG1) to F03DCH, F03DDH (SMSG14)
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
SMSGn								

7	6	5	4	3	2	1	0

Address: F03DEH, F03DFH
After reset: FFFFH
R/W: R

Symbol	15	14	13	12	11	10	9	8
SMSG15	1	1	1	1	1	1	1	1

7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1

Remark n = 1 to 14

Table 29 - 3 Correspondences between the Memory Addresses of the SMSGn Registers and Internal Data Addresses for the Sequencer

SMSGn	Addresses	Internal Data Addresses for the Sequencer
SMSG15	F03DEH, F03DFH	1111B
SMSG14	F03DCH, F03DDH	1110B
SMSG13	F03DAH, F03DBH	1101B
SMSG12	F03D8H, F03D9H	1100B
SMSG11	F03D6H, F03D7H	1011B
SMSG10	F03D4H, F03D5H	1010B
SMSG9	F03D2H, F03D3H	1001B
SMSG8	F03D0H, F03D1H	1000B
SMSG7	F03CEH, F03CFH	0111B
SMSG6	F03CCH, F03CDH	0110B
SMSG5	F03CAH, F03CBH	0101B
SMSG4	F03C8H, F03C9H	0100B
SMSG3	F03C6H, F03C7H	0011B
SMSG2	F03C4H, F03C5H	0010B
SMSG1	F03C2H, F03C3H	0001B
SMSG0	F03C0H, F03C1H	0000B

29.3.5 Sequencer control register (SMSC)

The SMSC register controls operation of the SNOOZE mode sequencer. Specifically, this register is used for control over starting and stopping of operation of the sequencer and holding the activating trigger for the sequencer pending, and for setting the source clock for counting in waiting triggered by the wait commands and the trigger for starting operation of the sequencer. The SMSC register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 29 - 6 Format of Sequencer Control Register (SMSC) (1/2)

Address: F03E0H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
SMSC	SMSSTART	SMSSTOP	SMSTRG WAIT	LONGWAIT	SMSTRGSE L3	SMSTRGSE L2	SMSTRGSE L1	SMSTRGSE L0

SMSSTART **Note 1**	Control over operation of the SNOOZE mode sequencer
0	The sequencer is stopped. Condition for setting to 0: Setting the SMSSTOP bit to 1
1	The sequencer is waiting for the selected activating trigger or operating. Condition for setting to 1: Writing 1 to this bit

SMSSTOP **Notes 1, 2**	Control over operation of the SNOOZE mode sequencer
0	—
1	Trigger for forcibly terminating processing by the sequencer Terminates processing by the sequencer and forcibly places the sequencer in the stopped state. Specifically, the sequencer is stopped on completion of the processing being handled at the time the SMSSTOP bit is set to 1.

SMSTRG WAIT **Note 3**	Control over holding the activating trigger pending
0	Disables holding the activating trigger pending
1	Enables holding the activating trigger pending On reception of the selected activating trigger, the trigger is held pending. Clearing the SMSTRGWAIT bit to 0 enables the trigger being held pending, so the sequencer starts operating.

LONGWAIT	Status flag indicating the source clock for counting for use with the wait command
0	Stops supply of the low-speed on-chip oscillator clock.
1	Enables supply of the low-speed on-chip oscillator clock.

Note 1. Writing 0 to the SMSSTART and SMSSTOP bits has no effect.
Note 2. The SMSSTOP bit is always read as 0.
Note 3. Only set the SMSTRGWAIT bit to 1 while the value of the SMSSTART bit is 0, that is, while the sequencer is stopped.

Figure 29 - 6 Format of Sequencer Control Register (SMSC) (2/2)

SMSTRGSEL3	SMSTRGSEL2	SMSTRGSEL1	SMSTRGSEL0	Selected Activating Trigger for the SNOOZE Mode Sequencer[Note 1]
0	0	0	0	Interval detection interrupt (INTITL) from the 32-bit interval timer
0	0	0	1	Pin input edge detection interrupt (INTP3)
0	0	1	0	UART0 reception transfer end interrupt (INTSR0)
0	0	1	1	CSI00 transfer end interrupt (INTCSI00)
0	1	0	0	A/D conversion end interrupt (INTAD)
0	1	0	1	ELCL output signal
0	1	1	0	UARTA0 reception transfer end interrupt (INTUR0)[Note 2]
0	1	1	1	Timer channel 02 count or capture end interrupt (INTTM02)
1	0	0	0	IICA0 communication end interrupt (INTIICA0)[Note 3]
1	0	0	1	Voltage detection interrupt (INTLVI)
1	0	1	0	Key return signal detection interrupt (INTKR)[Note 4]
Other than above				Setting prohibited

Note 1. Only set the SMSTRGSEL3 to SMSTRGSEL0 bits to 1 while the value of the SMSSTART bit is 0, that is, while the sequencer is stopped.

Note 2. This is only selectable in the R7F100G products with 36 to 128 pins and the R7F104G products with 32 to 48 pins.

Note 3. This is only selectable in the R7F100G products with 30 to 128 pins and the R7F104G products with 48 pins.

Note 4. This is only selectable in the R7F100G products with 40 to 128 pins and the R7F104G products with 32 to 48 pins.

29.3.6 Sequencer status register (SMSS)

The SMSS register indicates the state of the sequencer. This register indicates whether the sequencer is operating or stopped, the values of the zero and carry flags, and the number of the SMSIp register that holds the command currently being executed by the sequencer. The SMSS register can be read by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 29 - 7 Format of Sequencer Status Register (SMSS)

Address: F03E1H
After reset: 00H
R/W: R

Symbol	<7>	6	5	4	3	2	1	0
SMSS	SMSSTAT	SZ	SCY	SMSCV4	SMSCV3	SMSCV2	SMSCV1	SMSCV0

SMSSTAT	State of operation of the SNOOZE mode sequencer
0	The SNOOZE mode sequencer is stopped.
1	The SNOOZE mode sequencer is operating.

SZ	Zero flag of the SNOOZE mode sequencer
0	The result of the most recent operation by the sequencer was not 0.
1	The result of the most recent operation by the sequencer was 0.

SCY	Carry flag of the SNOOZE mode sequencer
0	The operation most recent operation by the sequencer did not produce carrying or borrowing.
1	The operation most recent operation by the sequencer produced carrying or borrowing.

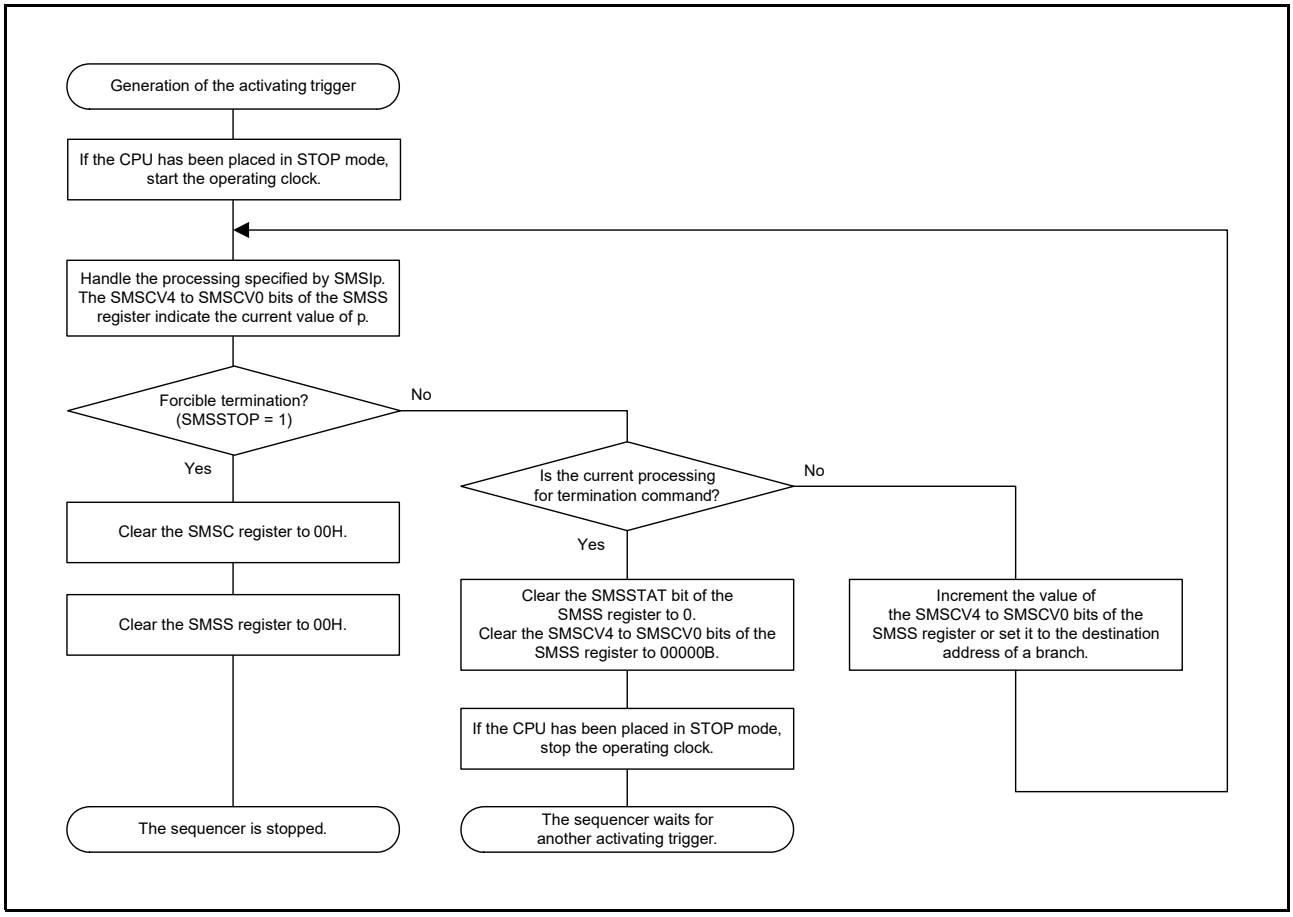
SMSCV4	SMSCV3	SMSCV2	SMSCV1	SMSCV0	State of processing by the SNOOZE mode sequencer
0 to 31					These bits indicate the number of the SMSIp register from which the sequencer is reading the command to be handled or has read the command that is currently being processed. Example: When the value of the SMSCV4 to SMSCV0 bits is 3, the processing set up in the SMSI3 register is currently being read or its execution is in progress.

29.4 Operations of the SNOOZE Mode Sequencer

29.4.1 Internal operations of the SNOOZE mode sequencer

Sequencing by the SNOOZE mode sequencer starts in response to the activating trigger specified by the SMSTRGSEL3 to SMSTRGSEL0 bits of the SMSC register. Following activation, the sequencer handles the processing specified by the SMSI0 register, and then handles the processing specified by the SMSIp register indicated by the SMSCV4 to SMSCV0 bits of the SMSS register. After execution of the processing for termination command, the sequencer has finished one round of processing and waits for another activating trigger. Moreover, setting the SMSSTOP bit of the SMSC register to 1 to generate a trigger for forcible termination leads to the sequencer being stopped. **Figure 29 - 8** shows the flow of internal operations of the SNOOZE mode sequencer.

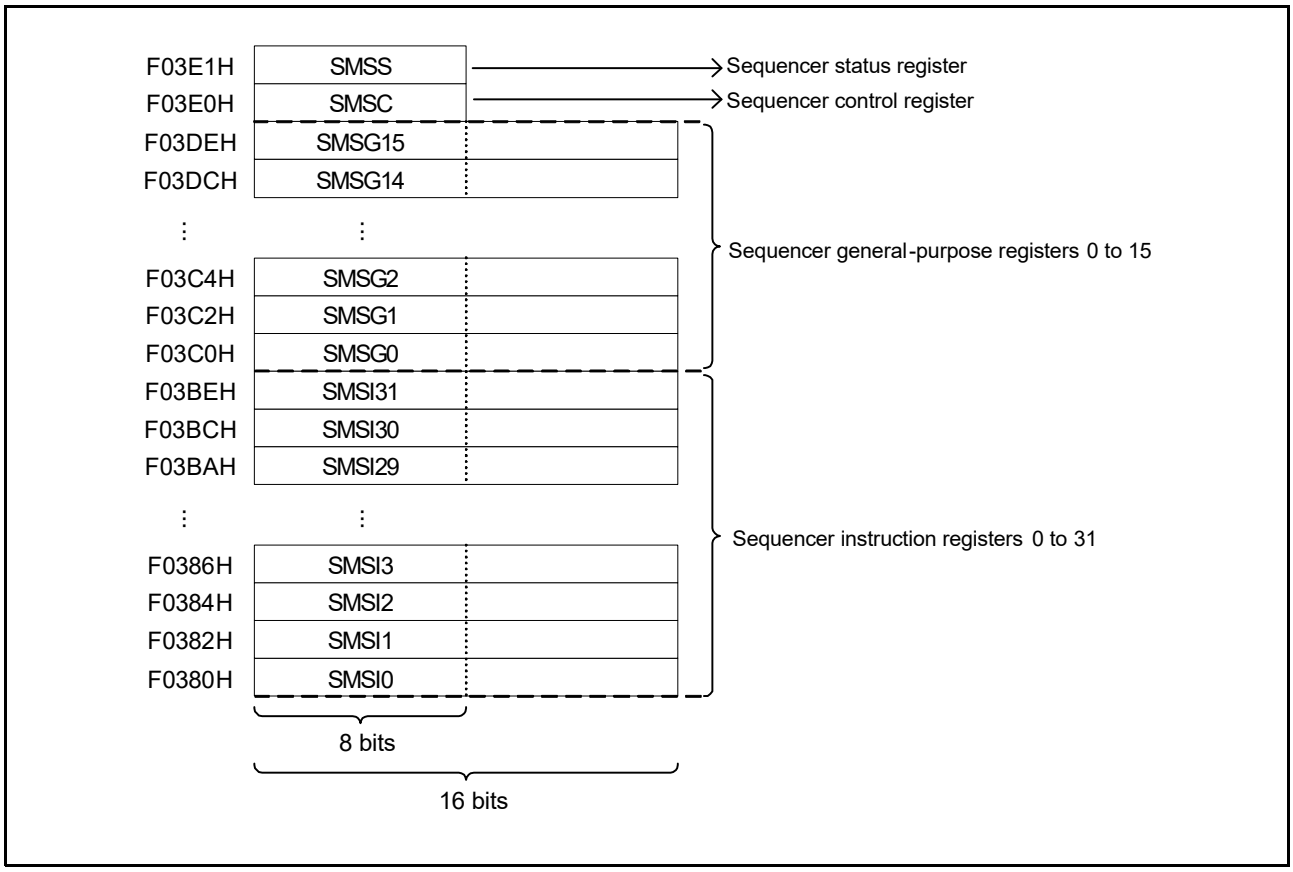
Figure 29 - 8 Flow of Internal Operations of the SNOOZE Mode Sequencer



29.4.2 Memory space allocated to the sequencer

The registers of the SNOOZE mode sequencer are in the second SFR area as shown in **Figure 29 - 9**. The sequencer handles the 16-bit processing codes stored at even addresses in the range from F0380H to F03BFH as being at five-bit addresses from 0 to 31 (00000B to 11111B). The sequencer also has direct access to the 16-bit data for processing stored at even addresses in the range from F03C0H to F03DFH as being at four-bit addresses from 0 to 15 (0000B to 1111B).

Figure 29 - 9 Memory Space Allocated to the SNOOZE Mode Sequencer

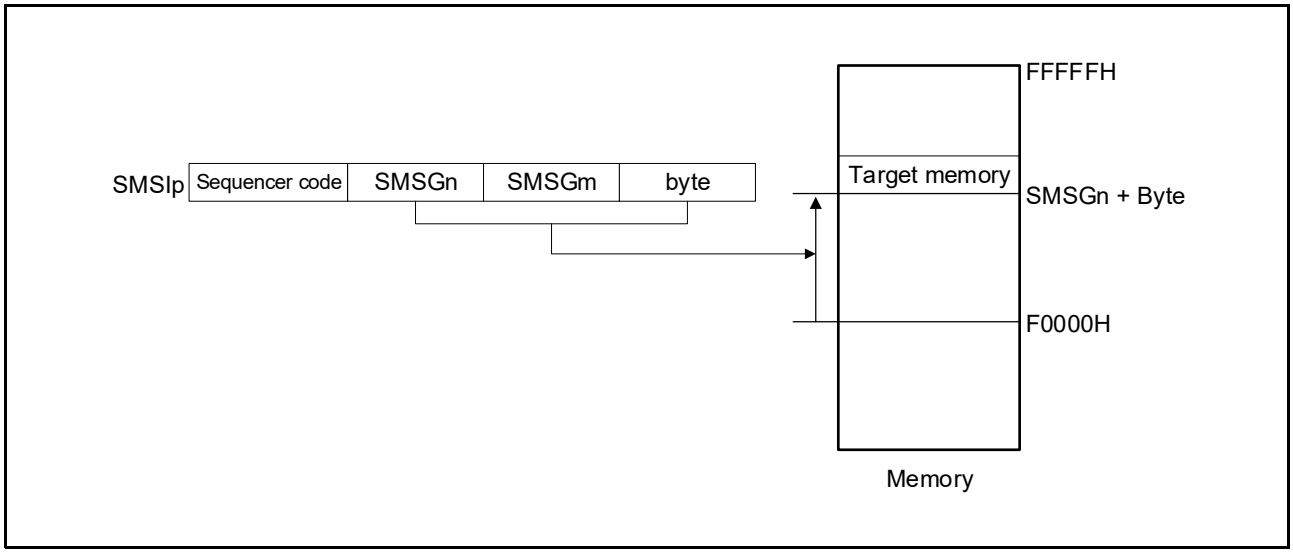


Moreover, the sequencer also has indirect access with some limitations[Note] to the memory space from F0000H to FFFFFH through use of the SMSGn registers.

Note Access by the SNOOZE mode sequencer to the areas listed below is prohibited.

- General-purpose register area
- Mirror area
- Data flash memory area
- An internal RAM area that has been placed in shutdown mode by the setting of the memory power reduction control register (PSMCR)
- The stated internal RAM areas of the products listed below (if the tracing function of on-chip debugging is in use)
  - R7F100GxG (x = A, B, C, E, F, G, J, L, M, and P) and R7F104GBG: FC300H to FC6FFH
  - R7F100GxJ (x = A, B, C, E, F, G, J, L, M, P, and S), R7F104GFJ, and R7F104GGJ: FA300H to FA6FFH
  - R7F100GxL, R7F100GxN (x = F, G, J, L, M, P, and S): F4300H to F46FFH
- Area to which the monitor program for debugging is allocated (when on-chip debugging is in use)
  → For details, see **34.4 Allocation of Memory Spaces to User Resources**.
- Sequencer instruction registers 0 o 31 (SMSI0 to SMSI31)
- Sequencer control register (SMSC)
- Sequencer status register (SMSS)
- SMSEN bit of peripheral enable register 1 (PER1)
- SMSRES bit of peripheral reset control register 1 (PRR1)

Figure 29 - 10 Memory Space to which the Sequencer Has Access



29.4.3 Sequencer flags

The sequencer has flags that are set or reset in response to the results of operations.

a) Sequencer zero flag (SZ)
The SZ flag is an internal flag of the sequencer. The flag is set to 1 when the result of addition, subtraction, or comparison is 0. Otherwise, the flag is cleared to 0. The flag is only for use in the internal processing by the sequencer. For details, see **29.5 Commands for Use in Processing by the Sequencer**.

b) Sequencer carry flag (SCY)
The SCY flag reflects the state of addition or subtraction producing an overflow or underflow, the value of the shifted-out bit in logical shifting processing, or the result of 1-bit data transfer. The flag is only for use in the internal processing by the sequencer. For details, see **29.5 Commands for Use in Processing by the Sequencer**.

The values of the SZ and SCY flags of the sequencer can be read from the corresponding bits of the SMSS register. See **29.3.6 Sequencer status register (SMSS)**.

29.4.4 Interrupt from the SNOOZE Mode Sequencer

The SMSEMK bit controls generation of the INTSMSE interrupt from the SNOOZE mode sequencer.
Before starting the SNOOZE mode sequencer operation (by setting SMSSTART to 1), use the CPU to set the SMSEMK and SMSEIF bits to 1. The SMSEIF and SMSEMK bits are respectively set to 1 and 0 by setting the SMSEMK bit to 0 within the SNOOZE mode sequencer processing, which leads to generation of the INTSMSE interrupt. When the interrupt is disabled (DI), the SMSEMK bit being 0 indicates the end of the sequencer operation.

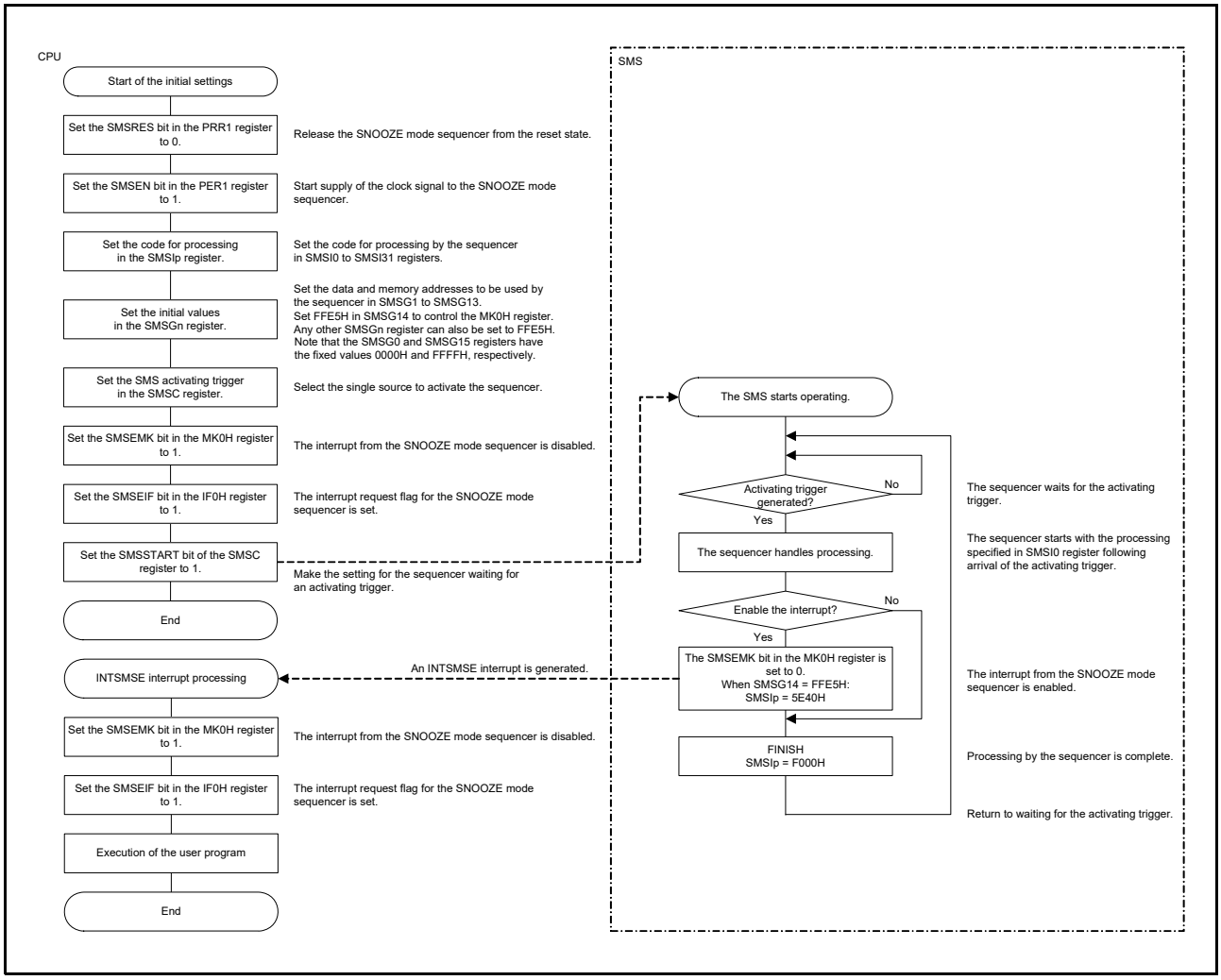
Caution 1. Do not use a CPU instruction to set the SMSEMK bit in the MK0H register or SMSEIF bit in the IF0H register to 0 while the setting of the SMSSTART bit in the SMSC register is 1.

Caution 2. If processing by the SNOOZE mode sequencer and that for the INTSMSE interrupt involve access to the same area in the SFR or RAM, ensure that the two types of processing do not run at the same time.

29.4.5 Procedures for running the SNOOZE mode sequencer

1. Example of the initial settings, activation, and operation of the SNOOZE mode sequencer
   Figure 29 - 11 shows an example of a flow of processing from the initial settings until activation and the completion of processing by the SNOOZE mode sequencer.

Figure 29 - 11 Flow of Activating and Running the SNOOZE Mode Sequencer



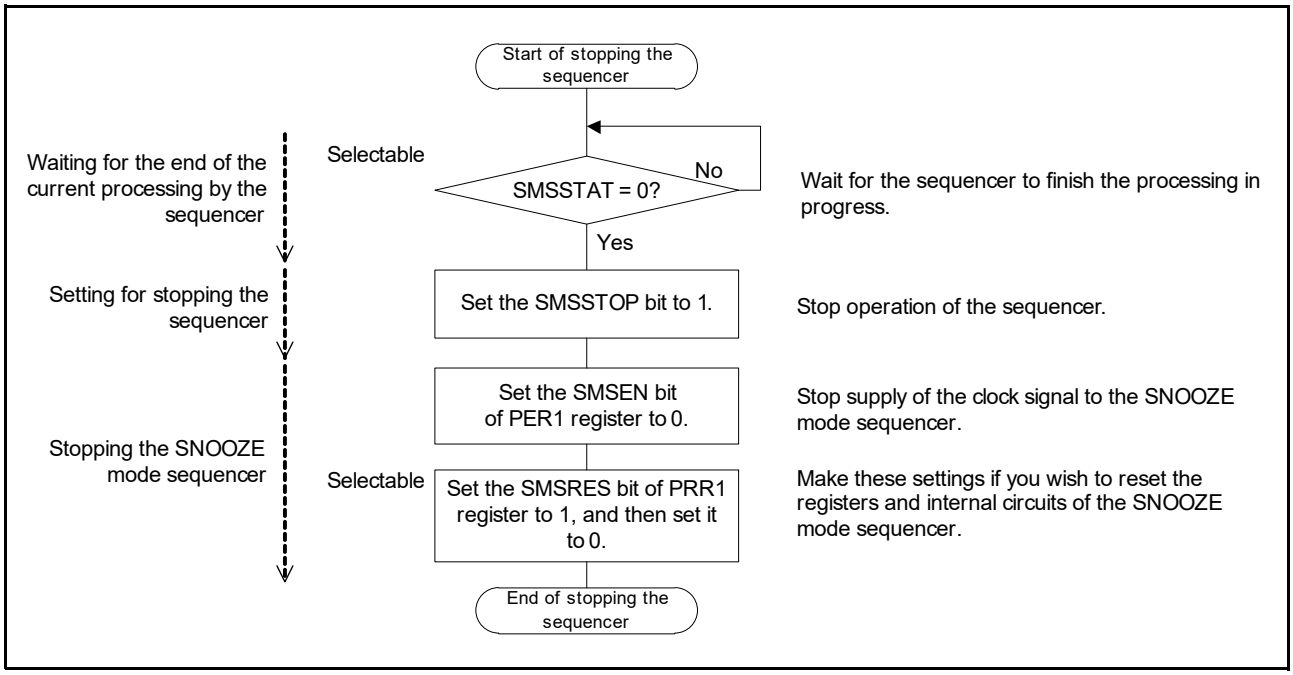
Caution 1. If processing by the sequencer ends for the latter reason (setting of the trigger bit for forcible termination), the SMSC register itself will be initialized. Therefore, to restart processing by the sequencer, make the initial settings of the SMSC register again (the SMSIp and SMSGn registers are not reset).

Caution 2. Do not use a CPU instruction to set the SMSEMK bit in the MK0H register or SMSEIF bit in the IF0H register to 0 while the setting of the SMSSTART bit in the SMSC register is 1.

2. Example of a flow of stopping the SNOOZE mode sequencer

Figure 29 - 12 shows an example of a flow of stopping the SNOOZE mode sequencer while it is operating.

Figure 29 - 12 Flow of Stopping the SNOOZE Mode Sequencer



29.4.6 States of the SNOOZE mode sequencer

The SNOOZE mode sequencer has four states: the sequencer stopped state, the activating trigger waiting state, the activation pending state, and the sequencer operating state. Transitions between these states proceed according to settings of the registers and input of the activating trigger.

Sequencer stopped state

The sequencer enters the stopped state immediately after it is released from the reset state or its operation is forcibly terminated by setting the SMSSTOP bit of the SMSC register to 1. Setting the SMSSTART bit of the SMSC register to 1 places the sequencer in the activating trigger waiting state.

Activating trigger waiting state

The activating trigger waiting state is that in which the detection of the activating trigger for the sequencer is enabled by setting the SMSSTART bit of the SMSC register to 1 but the trigger has not yet arrived. Detection of the activating trigger specified by the SMSTRGSEL3 to SMSTRGSEL0 bits of the SMSC register in this state starts the sequencer operating by placing it in the operating state. Note that if the setting of the SMSTRGWAIT bit of the SMSC register is also 1, the sequencer will not start operating and enters the activation pending state. In addition, if operation of the sequencer is forcibly terminated by setting the SMSSTOP bit of the SMSC register to 1, the sequencer enters the sequencer stopped state.

Activation pending state

The activation pending state is entered following detection of the activating trigger while the value of the SMSTRGWAIT bit of the SMSC register is 1. Clearing the SMSTRGWAIT bit to 0 in this state starts the sequencer operating by placing the sequencer in the sequencer operating state. If operation of the sequencer is forcibly terminated by setting the SMSSTOP bit of the SMSC register to 1, the sequencer enters the sequencer stopped state.

Sequencer operating state

The sequencer operating state is that in which the sequencer is operating and is handling processing specified by the SMSIp registers. Executing the termination command places the sequencer in the activating trigger waiting state. If operation of the sequencer is forcibly terminated by setting the SMSSTOP bit of the SMSC register to 1, the sequencer enters the sequencer stopped state.

The SMSSTART and SMSTRGWAIT bits of the SMSC register and the SMSSTAT bit of the SMSS register can be used to confirm the current state of the SNOOZE mode sequencer.

Table 29 - 4 Correspondence between the Bit Settings and States of the SNOOZE Mode Sequencer

SMSTRGWAIT	SMSSTAT	SMSSTART	State of the SNOOZE Mode Sequencer
0	0	0	Sequencer stopped state
0	0	1	Activating trigger waiting state
0	1	1	Sequencer operating state
1	0	0	Sequencer stopped state
1	0	1	Activating trigger waiting state
1	1	1	Setting prohibited (prohibited state)**Note**

Note Enabling holding a trigger pending while the sequencer is in the operating state is prohibited.

29.5 Commands for Use in Processing by the Sequencer

<R> The sequencer can handle 20 types of processing. For details, see sections **29.5.1** to **29.5.20**. Terms used in the descriptions of the commands (**29.5.1** to **29.5.20**) are defined below.

- Equivalent CPU command: Assembler language instruction that leads to the same processing as the given command.
- Equivalent CPU operation: Processing expressed in symbolic form
- $addr5: 5-bit relative address in 5 of 8 bits (-31 to -1, 1 to 31)
- jdisp8: 8-bit signed displacement value (0000 0001B to 0001 1111B, 1111 1111B to1110 0001B)
- Number of clock cycles for processing: Number of cycles of the f_{CLK} clock from the time the sequencer starts the processing until the result of the processing is actually reflected
- SMSIp: Sequencer instruction registers (p = 0 to 31)
- SMSGn: Sequencer internal data address (n = 0 to 15) for sequencer general-purpose registers, which is stored as the first operand of most commands
- SMSGm: Sequencer internal data address (m = 0 to 15) for sequencer general-purpose registers, which is stored as the second operand of most commands
- Example of a statement: Example of a binary value to be entered in the SMSIp register
- Sequencer code: Code that is to be entered in bits 15 to 12 of the SMSIp register and indicates the type of processing
- First operand: Code to be entered in bits 11 to 8 of the SMSIp register
- Second operand: Code to be entered in bits 7 to 4 of the SMSIp register
- Additional byte: Code to be entered in bits 3 to 0 of the SMSIp register

29.5.1 8-bit data transfer 1

8-bit data transfer 1 is for transferring 8 bits of data. The value of the SMSGn register specified as the first operand and that of the additional byte are used to specify the address of the destination for transfer in memory. The sequencer transfers the 8 bits of data specified as the second operand in the SMSGm register to the above destination in memory. Set a value in the range from 0 to 7 (from 0000B to 0111B) as the additional byte. To specify an SFR as the destination for transfer, set the address of the SFR that can be manipulated in 8-bit units in the first operand and additional byte. The bit units with which each SFR can be manipulated are the same as those for the CPU. Note that the sequencer general-purpose registers n (n = 0 to 15) can be specified as destinations for transfer. Executing 8-bit data transfer 1 results in a value being stored in the 8 lower-order bits of the specified sequencer general-purpose register n (n = 0 to 15).

Sequencer code: 0000B
Number of clock cycles for processing: 3 cycles of f_{CLK}
Flags: The states of the SZ and SCY flags are retained.
Equivalent CPU command: MOV [SMSGn+byte], SMSGm
Equivalent CPU operation: [SMSGn+byte] ← SMSGm

Symbol	15	14	13	12	11	10	9	8
SMSIp	0	0	0	0	First operand (SMSGn)			

7	6	5	4	3	2	1	0
Second operand (SMSGm)				Additional byte (byte)			

Example of a statement: 0000 0001 0111 0010 B
The equivalent CPU command in this case is MOV [SMSG1+2H], SMSG7. In the case where SMSG1 = FE00H and SMSG7 = xx12H, the value 12H stored in the SMSG7 register will be stored at the address FFE02H.

29.5.2 8-bit data transfer 2

8-bit data transfer 2 is for transferring 8 bits of data. The value of the SMSGn register specified as the first operand and that of the additional byte are used to specify the address of the source for transfer in memory. The sequencer stores the 8 bits of data at the source address for transfer in the SMSGm register specified as the second operand. Set a value in the range from 0 to 7 (from 0000B to 0111B) as the additional byte. To specify an SFR as the source for transfer, set the address of the SFR that can be manipulated in 8-bit units in the second operand and additional byte. The bit units with which each SFR can be manipulated are the same as those for the CPU. Note that the sequencer general-purpose registers n (n = 0 to 15) can be specified as sources for transfer. Executing 8-bit data transfer 2 results in reading of the value of the 8 lower-order bits of the specified sequencer general-purpose register n (n = 0 to 15).

Sequencer code: 0001B
Number of clock cycles for processing: 3 cycles of f_{CLK}
Flags: The states of the SZ and SCY flags are retained.
Equivalent CPU command: MOV SMSGm, [SMSGn+byte]
Equivalent CPU operation: SMSGm ← [SMSGn+byte]

Symbol

15	14	13	12	11	10	9	8
SMSIp 0	0	0	1	First operand (SMSGn)			

7	6	5	4	3	2	1	0
Second operand (SMSGm)				Additional byte (byte)			

Example of a statement: 0001 0001 0111 0010 B
The equivalent CPU command in this case is MOV SMSG7, [SMSG1+2H]. In the case where SMSG1 = FE00H and FFE02H = xx12H, the value 12H stored at the address FFE02H will be stored in the 8 lower-order bits of the SMSG7 register.

29.5.3 16-bit data transfer 1

16-bit data transfer 1 is for transferring 16 bits of data. The value of the SMSGn register specified as the first operand and that of the additional byte are used to specify the address of the destination for transfer in memory. The sequencer transfers the 16 bits of data specified as the second operand in the SMSGm register to the above destination in memory. Specify an even address as the destination for transfer in memory. If an odd address is specified, the LSB of the address is automatically changed to 0 before the data are transferred. Set a value in the range from 0 to 7 (from 0000B to 0111B) as the additional byte. To specify an SFR as the destination for transfer, set the address of the SFR that can be manipulated in 16-bit units in the first operand and additional byte. The bit units with which each SFR can be manipulated are the same as those for the CPU.

Sequencer code: 0010B
Number of clock cycles for processing: 3 cycles of f_{CLK}
Flags: The states of the SZ and SCY flags are retained.
Equivalent CPU command: MOVW [SMSGn+byte], SMSGm
Equivalent CPU operation: [SMSGn+byte] ← SMSGm

Symbol	15	14	13	12	11	10	9	8
SMSIp	0	0	1	0	First operand (SMSGn)			

7	6	5	4	3	2	1	0
Second operand (SMSGm)				Additional byte (byte)			

Example of a statement: 0010 0001 0111 0010 B
The equivalent CPU command in this case is MOVW [SMSG1+2H], SMSG7. In the case where SMSG1 = FE00H and SMSG7 = 1234H, the value 1234H stored in the SMSG7 register will be stored at the addresses FFE02H and FFE03H.

29.5.4 16-bit data transfer 2

16-bit data transfer 2 is for transferring 16 bits of data. The value of the SMSGn register specified as the first operand and that of the additional byte are used to specify the address of the source for transfer in memory. The sequencer stores the 16 bits of data at the source address for transfer in the SMSGm register specified as the second operand. Specify an even address as the source for transfer in memory. If an odd address is specified, the LSB of the address is automatically changed to 0 before the data are transferred. Set a value in the range from 0 to 7 (from 0000B to 0111B) as the additional byte. To specify an SFR as the source for transfer, set the address of the SFR that can be manipulated in 16-bit units in the second operand and additional byte. The bit units with which each SFR can be manipulated are the same as those for the CPU.

Sequencer code: 0011B
Number of clock cycles for processing: 3 cycles of f_{CLK}
Flags: The states of the SZ and SCY flags are retained.
Equivalent CPU command: MOV SMSGm, [SMSGn+byte]
Equivalent CPU operation: SMSGm ← [SMSGn+byte]

Symbol	15	14	13	12	11	10	9	8
SMSIp	0	0	1	1	First operand (SMSGn)			

7	6	5	4	3	2	1	0
Second operand (SMSGm)				Additional byte (byte)			

Example of a statement: 0011 0001 0111 0010 B
The equivalent CPU command in this case is MOVW SMSG7, [SMSG1+2H]. In the case where SMSG1 = FE00H, and FFE02H and FFE03H = 1234H, the value 1234H stored at the addresses FFE02H and FFE03H will be stored in the SMSG7 register.

29.5.5 1-bit data setting

1-bit data setting is for setting a specified bit among eight bits of data at a specified address to 1. The value of the SMSGn register specified as the first operand and that of the additional byte are used to specify the address in memory. The second operand is used to specify the number of the bit to be set to 1. Set values in the range from 0 to 7 (from 0000B to 0111B) in the second operand and additional byte. When an SFR is to be specified, set the address of the SFR that can be manipulated in 1-bit units and the bit number as the first operand, second operand, and additional byte. The bit units with which each SFR can be manipulated are the same as those for the CPU.

Sequencer code: 0100B
Number of clock cycles for processing: 4 cycles of f_{CLK}
Flags: The states of the SZ and SCY flags are retained.
Equivalent CPU command: SET1 [SMSGn+byte].bit
Equivalent CPU operation: [SMSGn+byte].bit ← 1

Symbol	15	14	13	12	11	10	9	8
SMSIp	0	1	0	0	First operand (SMSGn)			

7	6	5	4	3	2	1	0
Second operand (bit number)				Additional byte (byte)			

Example of a statement: 0100 0001 0011 0010 B
The equivalent CPU command in this case is SET1 [SMSG1+2H].3. In the case where SMSG1 = FE00H and FFE02H = 0000 0000B, bit 3 at the address FFE02H will be set to 1, making the value at the address FFE02H 0000 1000B.

29.5.6 1-bit data clearing

1-bit data clearing is for clearing a specified bit among eight bits of data at a specified address to 0. The value of the SMSGn register specified as the first operand and that of the additional byte are used to specify the address in memory. The second operand is used to specify the number of the bit to be cleared to 0. Set values in the range from 0 to 7 (from 0000B to 0111B) in the second operand and additional byte. When an SFR is to be specified, set the address of the SFR that can be manipulated in 1-bit units and the bit number as the first operand, second operand, and additional byte. The bit units with which each SFR can be manipulated are the same as those for the CPU.

Sequencer code: 0101B
Number of clock cycles for processing: 4 cycles of f_{CLK}
Flags: The states of the SZ and SCY flags are retained.
Equivalent CPU command: CLR1 [SMSGn+byte].bit
Equivalent CPU operation: [SMSGn+byte].bit ← 0

Symbol	15	14	13	12	11	10	9	8
SMSIp	0	1	0	1	First operand (SMSGn)			

7	6	5	4	3	2	1	0
Second operand (bit number)				Additional byte (byte)			

Example of a statement: 0101 0001 0011 0010 B
The equivalent CPU command in this case is CLR1 [SMSG1+2H].3. In the case where SMSG1 = FE00H and FFE02H = 0000 1000B, bit 3 at the address FFE02H will be cleared to 0, making the value at the address FFE02H 0000 0000B.

29.5.7 1-bit data transfer

1-bit data transfer is for transferring the value of a specified bit among eight bits of data at a specified address to the SCY flag. The value of the SMSGn register specified as the first operand and that of the additional byte are used to specify the address of the source for transfer in memory. The second operand is used to specify the bit whose value is to be transferred. Set values in the range from 0 to 7 (from 0000B to 0111B) in the second operand and additional byte. When an SFR is to be specified, set the address of the SFR that can be manipulated in 1-bit units and the bit number as the first operand, second operand, and additional byte. The bit units with which each SFR can be manipulated are the same as those for the CPU.

Sequencer code: 0110B
Number of clock cycles for processing: 3 cycles of f_{CLK}
Flags: The state of the SZ flag is retained. The SCY flag reflects the result of the transfer.
Equivalent CPU command: MOV1 SCY, [SMSGn+byte].bit
Equivalent CPU operation: SCY ← [SMSGn+byte].bit

Symbol	15	14	13	12	11	10	9	8
SMSIp	0	1	1	0	First operand (SMSGn)			

7	6	5	4	3	2	1	0
Second operand (bit number)				Additional byte (byte)			

Example of a statement: 0110 0001 0011 0010 B
The equivalent CPU command in this case is MOV1 SCY, [SMSG1+2H].3. In the case where SMSG1 = FE00H and FFE02H = 0000 1000B, the value of bit 3 at the address FFE02H will be transferred to the SCY flag, so the SCY flag will be set to 1.

29.5.8 Word addition

Word addition is for adding two 16-bit values. The values of the SMSGn and SMSGm registers that are specified as the first and second operands are added, and the result is stored in the SMSGn register specified as the first operand. If the result of addition that is stored in the SMSGn register is 0, the SZ flag is set to 1. Otherwise, the flag is cleared to 0. Also, if an arithmetic carry has been generated from bit 15, the SCY flag is set to 1. Otherwise, the flag is cleared to 0. Set the additional byte to 0000B.

Sequencer code: 0111B (additional byte: 0000B)
Number of clock cycles for processing: 1 cycle of f_{CLK}
Flags: The SZ and SCY flags reflect the result of the calculation.
Equivalent CPU command: ADDW SMSGn, SMSGm
Equivalent CPU operation: SMSGn, SCY ← SMSGn + SMSGm

Symbol	15	14	13	12	11	10	9	8
SMSIp	0	1	1	1	First operand (SMSGn)			

7	6	5	4	3	2	1	0
Second operand (SMSGm)				0	0	0	0

Example of a statement: 0111 0001 0011 0000 B
The equivalent CPU command in this case is ADDW SMSG1, SMSG3. In the case where SMSG1 = FFF0H and SMSG3 = 0010H, the settings of the register and flags as the results of the addition will be as follows: SMSG1 = 0000H, SCY = 1, and SZ = 1.

29.5.9 Word subtraction

Word subtraction is for subtracting two 16-bit values. The value of the SMSGm register specified as the second operand is subtracted from the value of the SMSGn register specified as the first operand, and then the resulting value is stored in the SMSGn register specified in the first operand. If the result of subtraction that is stored in the SMSGn register is 0, the SZ flag is set to 1. Otherwise, the flag is cleared to 0. Also, if an arithmetic borrow has been generated from bit 0, the SCY flag is set to 1. Otherwise, the flag is cleared to 0. Set the additional byte to 0001B.

Sequencer code: 0111B (additional byte: 0001B)
Number of clock cycles for processing: 1 cycle of f_{CLK}
Flags: The SZ and SCY flags reflect the result of the calculation.
Equivalent CPU command: SUBW SMSGn, SMSGm
Equivalent CPU operation: SMSGn, SCY ← SMSGn - SMSGm

Symbol	15	14	13	12	11	10	9	8
SMSIp	0	1	1	1	First operand (SMSGn)			

7	6	5	4	3	2	1	0
Second operand (SMSGm)				0	0	0	1

Example of a statement: 0111 0001 0011 0001 B
The equivalent CPU command in this case is SUBW SMSG1, SMSG3. In the case where SMSG1 = 1234H and SMSG3 = 1200H, the settings of the register and flags as the results of the subtraction will be as follows: SMSG1 = 0034H, SCY = 0, and SZ = 0.

29.5.10 Word comparison

Word comparison is for comparing two 16-bit values. The value of the SMSGm register specified as the second operand is subtracted from the value of the SMSGn register specified as the first operand. If the setting of the SMSGn register as the result of the subtraction is 0, the SZ flag is set to 1. Otherwise, the flag is cleared to 0. Also, if an arithmetic borrow has been generated from bit 0, the SCY flag is set to 1. Otherwise, the flag is cleared to 0. Set the additional byte to 0010B.

Sequencer code: 0111B (additional byte: 0010B)
Number of clock cycles for processing: 1 cycle of f_{CLK}
Flags: The SZ and SCY flags reflect the result of the calculation.
Equivalent CPU command: CMPW SMSGn, SMSGm
Equivalent CPU operation: SMSGn - SMSGm

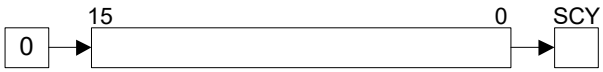
Symbol	15	14	13	12	11	10	9	8
SMSIp	0	1	1	1	First operand (SMSGn)			

7	6	5	4	3	2	1	0
Second operand (SMSGm)				0	0	1	0

Example of a statement: 0111 0001 0011 0010 B
The equivalent CPU command in this case is CMPW SMSG1, SMSG3. In the case where SMSG1 = 1234H and SMSG3 = 1200H, the settings of the flags as the results of the comparison will be as follows: SCY = 0 and SZ = 0.

29.5.11 Logical shift right

Logical shift right is for shifting a 16-bit value by 1 bit to the right. The value of the SMSGn register specified as the first operand is shifted by 1 bit to the right. At this time, the MSB (bit 15) of the SMSGn register is set to 0 and the value of the LSB (bit 0) that is shifted out is stored in the SCY flag.



Set the additional byte to 0011B. Since the second operand is not used, set it to 0000B.

Sequencer code: 0111B (additional byte: 0011B)
Number of clock cycles for processing: 1 cycle of f_{CLK}
Flags: The state of the SZ flag is retained. The SCY flag reflects the result of the calculation.
Equivalent CPU command: SHRW SMSGn
Equivalent CPU operation: SCY ← SMSGn.0, SMSGn.m-1 ← SMSGn.m, SMSGn.15 ← 0

Symbol	15	14	13	12	11	10	9	8
SMSIp	0	1	1	1	First operand (SMSGn)			

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	1

Example of a statement: 0111 0001 0000 0011 B
The equivalent CPU command in this case is SHRW SMSG1. In the case where SMSG1 = AAF5H, the settings of the register and flag as the results of the logical shift will be as follows: SMSG1 = 557AH and SCY = 1.

29.5.12 Branch 1 (SCY = 1)

Branch 1 is for processing to branch if SCY = 1. With SCY = 1, program control branches to the processing stored in the SMSIp register at the relative address specified as the first and second operands. If SCY = 0, processing proceeds to that set in the next SMSIp register. Set the additional byte to 0000B. Set values in the range from -31 to -1 and 1 to 31 as the first and second operands, respectively.

Sequencer code: 1000B (additional byte: 0000B)
Number of clock cycles for processing: 1 cycle of f_{CLK}
Flags: The states of the SZ and SCY flags are retained.
Equivalent CPU command: BC $addr5
Equivalent CPU operation: SMSS[4:0] ← SMSS[4:0] + jdisp8 if SCY = 1

Symbol	15	14	13	12	11	10	9	8
SMSIp	1	0	0	0	First and second operands ($addr5)			

7	6	5	4	3	2	1	0
First and second operands ($addr5)				0	0	0	0

Example of a statement: 1000 0000 0101 0000 B
The equivalent CPU command in this case is BC $05H. In the case where the value of SMSS[4:0] is 03H, and SCY = 1, the value of SMSS[4:0] will be 08H, so processing will proceed to that stored in the SMSI8 register.

29.5.13 Branch 2 (SCY = 0)

Branch 2 is for processing to branch if SCY = 0. With SCY = 0, program control branches to the processing stored in the SMSIp register at the relative address specified as the first and second operands. If SCY = 1, processing proceeds to that set in the next SMSIp register. Set the additional byte to 0001B. Set values in the range from -31 to -1 and 1 to 31 as the first and second operands, respectively.

Sequencer code: 1000B (additional byte: 0001B)
Number of clock cycles for processing: 1 cycle of f_{CLK}
Flags: The states of the SZ and SCY flags are retained.
Equivalent CPU command: BNC $addr5
Equivalent CPU operation: SMSS[4:0] ← SMSS[4:0] + jdisp8 if SCY = 0

Symbol	15	14	13	12	11	10	9	8
SMSIp	1	0	0	0	First and second operands ($addr5)			

7	6	5	4	3	2	1	0
First and second operands ($addr5)				0	0	0	1

Example of a statement: 1000 1111 1110 0001 B
The equivalent CPU command in this case is BNC $0FEH. In the case where the value of SMSS[4:0] is 0EH, and SCY = 0, the value of SMSS[4:0] will be 0CH, so processing will proceed to that stored in the SMSI12 register.

29.5.14 Branch 3 (SZ = 1)

Branch 3 is for processing to branch if SZ = 1. With SZ = 1, program control branches to the processing stored in the SMSIp register at the relative address specified as the first and second operands. If SZ = 0, processing proceeds to that set in the next SMSIp register. Set the additional byte to 0010B. Set values in the range from -31 to -1 and 1 to 31 as the first and second operands, respectively.

Sequencer code: 1000B (additional byte: 0010B)
Number of clock cycles for processing: 1 cycle of f_{CLK}
Flags: The states of the SZ and SCY flags are retained.
Equivalent CPU command: BZ $addr5
Equivalent CPU operation: SMSS[4:0] ← SMSS[4:0] + jdisp8 if SZ = 1

Symbol	15	14	13	12	11	10	9	8
SMSIp	1	0	0	0	First and second operands ($addr5)			

7	6	5	4	3	2	1	0
First and second operands ($addr5)				0	0	1	0

Example of a statement: 1000 0000 0101 0010 B
The equivalent CPU command in this case is BZ $05H. In the case where the value of SMSS[4:0] is 03H, and SZ = 1, the value of SMSS[4:0] will be 08H, so processing will proceed to that stored in the SMSI8 register.

29.5.15 Branch 4 (SZ = 0)

Branch 4 is for processing to branch if SZ = 0. With SZ = 0, program control branches to the processing stored in the SMSIp register at the relative address specified as the first and second operands. If SZ = 1, processing proceeds to that set in the next SMSIp register. Set the additional byte to 0011B. Set values in the range from -31 to -1 and 1 to 31 as the first and second operands, respectively.

Sequencer code: 1000B (additional byte: 0011B)
Number of clock cycles for processing: 1 cycle of f_{CLK}
Flags: The states of the SZ and SCY flags are retained.
Equivalent CPU command: BNZ $addr5
Equivalent CPU operation: SMSS[4:0] ← SMSS[4:0] + jdisp8 if SZ = 0

Symbol	15	14	13	12	11	10	9	8
SMSIp	1	0	0	0	First and second operands ($addr5)			

7	6	5	4	3	2	1	0
First and second operands ($addr5)				0	0	1	1

Example of a statement: 1000 1111 1110 0011 B
The equivalent CPU command in this case is BNZ $0FEH. In the case where the value of SMSS[4:0] is 0EH, and SZ = 0, the value of SMSS[4:0] will be 0CH, so processing will proceed to that stored in the SMSI12 register.

29.5.16 Wait

The wait command holds processing pending for a certain period. Execution of processing is held pending for the number of cycles of f_{WAIT} specified in the second operand by using the clock and division ratio specified as the first operand and additional byte. Use the first and second operands to select the source clock for counting of the wait time and the value to be counted as the wait period, and use the additional byte to select the clock for use in waiting, specifically, select the divisor for the frequency of the selected source clock. If the low-speed on-chip oscillator clock is to be used as the source clock for counting of the wait time, set the LONGWAIT bit of the SMSC register to 1 in advance.

Sequencer code: 1001B
Number of clock cycles for processing: See the description titled "**1. Number of clock cycles for use in waiting**".
Flags: The states of the SZ and SCY flags are retained.
Equivalent CPU command: WAIT IM1, IM2
Equivalent CPU operation: Holding processing pending for a certain period

Symbol	15	14	13	12	11	10	9	8
SMSIp	1	0	0	1		First and second operands (IM1)		

7	6	5	4	3	2	1	0
First and second operands (IM1)				Additional byte (IM2)			

IM1[7]	Selection of the Source Clock f_{SMS} for Use in Waiting
0	f_{CLK}
1	f_{IL} **Notes 1, 2**

IM1[6:0]	Selection of the Wait Time
0000000	128/f_{WAIT}
0000001	1/f_{WAIT}
0000010	2/f_{WAIT}
• • •	• • •
1111110	126/f_{WAIT}
1111111	127/f_{WAIT}

IM2	Selection of the Clock (f_{WAIT}) for Use in Waiting
0000	f_{SMS}
0001	f_{SMS}/2
0010	f_{SMS}/4
0011	f_{SMS}/8
0100	f_{SMS}/16
0101	f_{SMS}/32
0110	f_{SMS}/64
0111	f_{SMS}/128
Other than the above	Settings prohibited

Example of a statement:

Example of the Setting of the SMSIp Register	Source Clock for Use in Waiting	Clock for Use in Waiting	Number to Count for Waiting	Wait Time in μs (typ.)
1001 0 0000001 0000	f_{CLK}	f_{CLK}	1	0.03125
1001 0 0000000 0111	f_{CLK}	$f_{CLK}/128$	128	512
1001 0 1000000 0110	f_{CLK}	$f_{CLK}/64$	64	128
1001 1 0000001 0000	f_{IL}	f_{IL}	1	30.51757813
1001 1 0000000 0111	f_{IL}	$f_{IL}/128$	128	500000
1001 1 1100000 0101	f_{IL}	$f_{IL}/32$	96	93750

Note 1. When the low-speed on-chip oscillator clock (f_{IL}) is in use for the CPU/peripheral hardware clock frequency (f_{CLK}), select f_{CLK} (IM1[7] = 0) as the source clock f_{SMS} for counting the wait time.

Note 2. When the low-speed on-chip oscillator clock (f_{IL}) is to be used as the source clock for the timing of waiting by the SNOOZE mode sequencer, set peripheral enable register 0 (PER0) and peripheral enable register 1 (PER1) to 00H and 40H, respectively.

Remark f_{CLK}: CPU/peripheral hardware clock frequency
f_{IL}: Low-speed on-chip oscillator clock frequency (32.768 kHz)

1. Number of clock cycles for use in waiting
   The number of clock cycles for use in waiting depends on the setting of the wait command. Refer to values set for IM1 and IM2 to calculate the required processing time. Note that the wait processing requires time for synchronization before the counter for use in timing the wait starts operating and after the counter finishes operating. Refer to an example described below to calculate the time for synchronization.
   Before execution of the wait command:
   (1 cycle of f_{CLK}) + (2^{Note} cycles of the selected clock for use in waiting)

   After execution of the wait command:
   2^{Note} cycles of f_{CLK}

Note This is the number of clock cycles for synchronization and will be no greater than 2. This time may also be extremely close to 1 cycle with some types of timing.

29.5.17 Conditional wait 1 (bit = 1)

In conditional wait 1, processing is repeatedly held pending as long as the value of a specified bit among eight bits of data at a specified address is 1. The value of the SMSGn register specified as the first operand and that of the additional byte are used to specify the address in memory. The second operand is used to specify the target bit for the condition. The sequencer reads the specified bit. If the setting is 1, the sequencer does not handle the succeeding processing but goes into wait processing. The sequencer repeats this as long as the value of the specified bit is 1. When reading from the specified bit returns 0, the sequencer handles the succeeding processing. Set a value in the range from 0 to 7 (from 0000B to 0111B) as the second operand. When an SFR is to be specified, set the address of the SFR that can be manipulated in 1-bit units and the bit number as the first operand, second operand, and additional byte. The bit units with which each SFR can be manipulated are the same as those for the CPU.

Sequencer code: 1010B
Number of clock cycles for processing: See the description titled "**1. Number of clock cycles required for conditional wait 1**".
Flags: The states of the SZ and SCY flags are retained.
Equivalent CPU command: WHILE1 [SMSGn+byte].bit
Equivalent CPU operation: SMSS[4:0] ← SMSS[4:0] if [SMSGn+byte].bit = 1

Symbol	15	14	13	12	11	10	9	8
SMSIp	1	0	1	0	First operand (SMSGn)			

7	6	5	4	3	2	1	0
Second operand (bit number)				Additional byte (byte)			

Example of a statement: 1010 0011 0101 0010 B
The equivalent CPU command in this case is WHILE1 [SMSG3+2].5. In the case where SMSG3 = 02E8H, the sequencer holds processing pending as long as the value of bit 5 at address F02EAH is 1. When the value of the bit becomes 0, the sequencer handles the succeeding processing.

1. Number of clock cycles required for conditional wait 1
   In conditional wait 1 processing, access to the specified bit is repeated until its value becomes 0. Starting the processing requires one cycle of f_{CLK} and each round of reference to the register bit and judgement requires two cycles of f_{CLK}. For example, the time set up in the case where the fifth access to the bit for reference of the destination register returns 0 is calculated from the following formula.
   $f_{CLK} \times 1 + (f_{CLK} \times 2) \times 5 = f_{CLK} \times 11$
   Note that if attempted access to memory by the SNOOZE mode sequencer is in contention with access by the CPU or DTC, the SNOOZE mode sequencer may be placed in the pending state. Such cases also increase the number of clock cycles.

29.5.18 Conditional wait 2 (bit = 0)

In conditional wait 2, processing is repeatedly held pending as long as the value of a specified bit among eight bits of data at a specified address is 0. The value of the SMSGn register specified as the first operand and that of the additional byte are used to specify the address in memory. The second operand is used to specify the target bit for the condition. The sequencer reads the specified bit. If the setting is 0, the sequencer does not handle the succeeding processing but goes into wait processing. The sequencer repeats this as long as the value of the specified bit is 0. When reading from the specified bit returns 1, the sequencer handles the succeeding processing. Set a value in the range from 0 to 7 (from 0000B to 0111B) as the second operand. When an SFR is to be specified, set the address of the SFR that can be manipulated in 1-bit units and the bit number as the first operand, second operand, and additional byte. The bit units with which each SFR can be manipulated are the same as those for the CPU.

Sequencer code: 1011B
Number of clock cycles for processing: See the description titled "**1. Number of clock cycles required for conditional wait 2**".
Flags: The states of the SZ and SCY flags are retained.
Equivalent CPU command: WHILE0 [SMSGn+byte].bit
Equivalent CPU operation: SMSS[4:0] ← SMSS[4:0] if [SMSGn+byte].bit = 0

Symbol	15	14	13	12	11	10	9	8
SMSIp	1	0	1	1	First operand (SMSGn)			

7	6	5	4	3	2	1	0
Second operand (bit number)				Additional byte (byte)			

Example of a statement: 1011 0011 0101 0010 B
The equivalent CPU command in this case is WHILE0 [SMSG3+2].5. In the case where SMSG3 = FFE0H, the sequencer holds processing pending as long as the value of bit 5 at address FFFE2H is 0. When the value of the bit becomes 1, the sequencer handles the succeeding processing.

1. Number of clock cycles required for conditional wait 2
   Since the number of clock cycles of waiting set up by conditional wait 2 is counted in the same way as described for conditional wait 1, refer to **29.5.17 Conditional wait 1 (bit = 1)**, **1. Number of clock cycles required for conditional wait 1**.

29.5.19 Termination

The termination command stops the SNOOZE mode sequencer. Specifically, execution of the command stops the SNOOZE mode sequencer, clears the SMSSTAT and SMSCV4 to SMSCV0 bits in the SMSS register to 0, and places the sequencer in the activating trigger waiting state. Set the additional byte to 0000B. Set all bits of the first and second operands to 0.

Sequencer code: 1111B (additional byte: 0000B)
Number of clock cycles for processing: 1 cycle of f_{CLK}
Flags: The states of the SZ and SCY flags are retained.
Equivalent CPU command: FINISH
Equivalent CPU operation: SMSSTAT ← 0, SMSS[4:0] ← 0, stopping the sequencer

Symbol	15	14	13	12	11	10	9	8
SMSIp	1	1	1	1	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0

Example of a statement: 1111 0000 0000 0000 B
The equivalent CPU command in this case is FINISH. The termination command stops the sequencer, clears the SMSSTAT and SMSCV4 to SMSCV0 bits of the SMSS register to 0, and places the sequencer in the activating trigger waiting state.

29.5.20 DTC activation

The DTC activation command issues a DTC activating source signal so that DTC transfer proceeds. That is, execution of the command issues an activating source signal for the DTC. If the corresponding control settings for the DTC have been made in advance, the specified DTC transfer will proceed. Set the additional byte to 0010B. Set all bits of the first and second operands to 0.

Sequencer code: 1111B (additional byte: 0010B)
Number of clock cycles for processing: 1 cycle of f_{CLK}
Flags: The states of the SZ and SCY flags are retained.
Equivalent CPU command: DTCTRG
Equivalent CPU operation: Output of a DTC activating source signal

Symbol	15	14	13	12	11	10	9	8
SMSIp	1	1	1	1	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	0

Example of a statement: 1111 0000 0000 0010 B
The equivalent CPU command in this case is DTCTRG. The DTC activation command issues a DTC activating source signal. DTC transfer proceeds according to the DTC control settings. To hold processing by the sequencer pending until completion of DTC transfer, use the wait command to hold processing pending until the DTC has been started, and then use conditional wait 1 to hold the processing pending until the corresponding bit of DTC activation enable register i (DTCENi) (i = 0 to 4) has been set to 0.

29.6 Operation in Standby Modes

State	Operation of the SNOOZE Mode Sequencer
HALT mode	Operation continues.**Note 1**
STOP mode	The activating trigger for the SNOOZE mode sequencer can be accepted.**Note 2**
SNOOZE mode	Operation continues.**Notes 3, 4, 5, 6**

Note 1. When the subsystem clock is selected as f_{CLK}, operation is disabled if the RTCLPC bit of the OSMC register is 1.

Note 2. Detection of an SMS activating trigger in STOP mode places the chip in SNOOZE mode, making the SNOOZE mode sequencer capable of operation. The state of the chip returns to the STOP mode after the operations of the SMS are completed. Note that the sequencer does not have access to certain memory areas in SNOOZE mode. For details, see **29.4.2 Memory space allocated to the sequencer**.

Note 3. The SNOOZE mode can only be set when the high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock is selected as f_{CLK}.

Note 4. When a transfer end interrupt from CSI00 is being used as the activating trigger for the SNOOZE mode sequencer but the transfer end interrupt is disabled (CSIMK = 1), proceed with the following steps before the processing for termination of the SNOOZE mode sequencer. Write 0 to the SMSEMK bit in the MK0H register, and release the chip from the SNOOZE mode to start processing by the CPU, or make the settings for reception by CSI00 (writing 1 to the ST00 bit, writing 0 to the SWC0 bit, setting the SSC0 register, and writing 1 to the SS00 bit) again.

Note 5. When a transfer end interrupt from UART0 is being used as the activating trigger for the SNOOZE mode sequencer, but the transfer end interrupt is disabled (SRMK0 = 1), proceed with the following steps before the processing for termination of the SNOOZE mode sequencer. Write 0 to the SMSEMK bit in the MK0H register, and release the chip from the SNOOZE mode to start processing by the CPU, or make the settings for reception by UART0 (writing 1 to the ST01 bit, writing 0 to the SWC0 bit, setting the SSC0 register, and writing 1 to the SS01 bit) again.

Note 6. When an A/D conversion end interrupt from the A/D converter is being used as the activating trigger for the SNOOZE mode sequencer, but the A/D conversion end interrupt is disabled (ADMK = 1), proceed with the following steps before the processing for termination of the SNOOZE mode sequencer. Write 0 to the SMSEMK bit in the MK0H register, and release the chip from the SNOOZE mode to start processing by the CPU, or make the settings for the SNOOZE mode of the A/D converter (writing 1 to the AWC bit after having written 0 to it) again.

Caution Access to the following realtime clock registers through the SNOOZE mode sequencer is not possible in the standby mode.

- **RTCC0, RTCC1, SEC, MIN, HOUR, DAY, WEEK, MONTH, YEAR, SUBCUD, ALARMWM, ALARMWH, and ALARMWW**

Section 30 Capacitive Sensing Unit (CTSU2L)

The number of measurement channels of the capacitive sensing unit depends on the product.

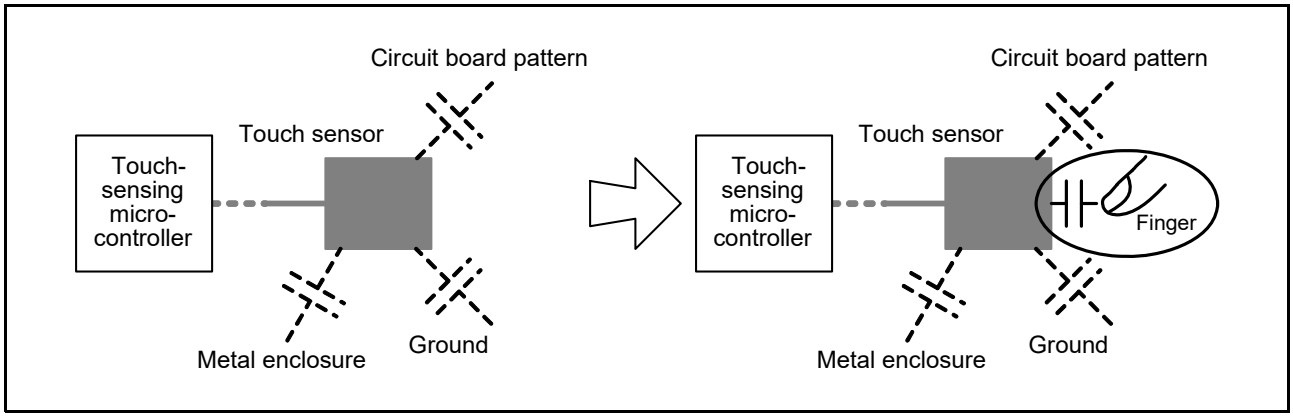
	R7F100G								
ROM size	96 to 128 Kbytes							128 Kbytes	
Pin count	30	32	36	40, 44	48	52	64	80	100
Number of the CTSU2L output channels	2 TS00, TS01	3 TS00 to TS02	5 TS00 to TS04	6 TS00 to TS05	8 TS00 to TS07	10 TS00 to TS09	12 TS00 to TS11	30 TS00 to TS15, TS20 to TS33	32 TS00 to TS15, TS20 to TS35

	R7F100G										
ROM size	192 to 256 Kbytes				192 to 768 Kbytes						256 to 768 Kbytes
Pin count	30	32	36	40	44	48	52	64	80	100	128
Number of the CTSU2L output channels	6 TS00, TS01, TS20, TS21, TS26, TS27	7 TS00 to TS02, TS20, TS21, TS26, TS27	11 TS00 to TS04, TS20 to TS23, TS26, TS27	13 TS00 to TS05, TS20 to TS24, TS26, TS27	14 TS00 to TS05, TS20 to TS27	16 TS00 to TS07, TS20 to TS27	20 TS00 to TS09, TS20 to TS29	22 TS00 to TS11, TS20 to TS29	30 TS00 to TS15, TS20 to TS33	32 TS00 to TS15, TS20 to TS35	

	R7F104G		
ROM size	128 Kbytes	256 Kbytes	
Pin count	32	44	48
Number of the CTSU2L output channels	10 TS00 to TS05, TS08 to TS11	24 TS00 to TS05, TS08 to TS13, TS20 to TS30, TS33	25 TS00 to TS05, TS08 to TS14, TS20 to TS30, TS33

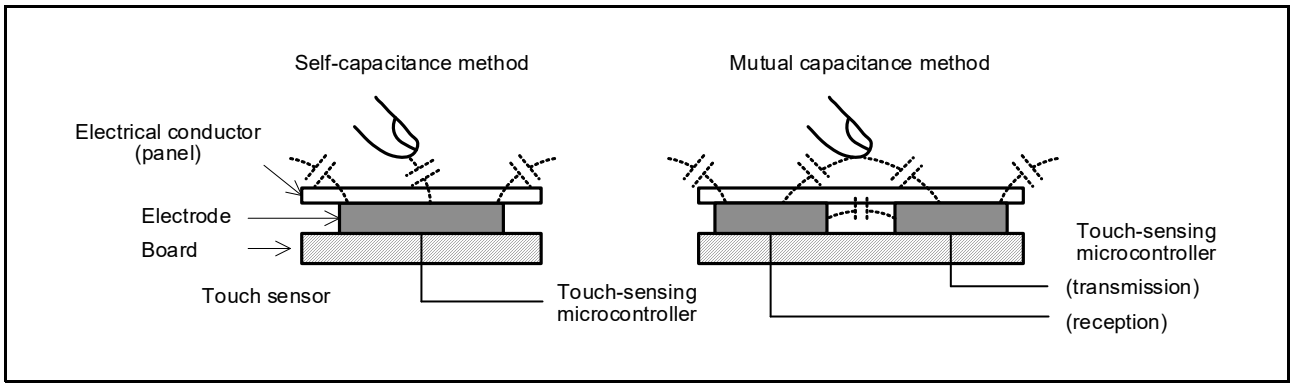
The capacitive sensing unit (CTSU2L) measures the electrostatic capacitance of the capacitive sensor. Changes in the electrostatic capacitance are determined by software, which enables the CTSU2L to detect whether a finger is in contact with the capacitive sensor. The electrode surface of the capacitive 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 30 - 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 30 - 1 Increased Electrostatic Capacitance Because of the Presence of a Finger



The CTSU supports measurement 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 as a transmission electrode and the other as a reception electrode. The CTSU detects changes in the electrostatic capacitance generated between these electrodes when a finger is placed close to them.

Figure 30 - 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, see **30.3.1 Principles of measurement**. For details on the development of touch applications, see the application note, **Using QE and SIS to Develop Capacitive Touch Applications (R01AN5512)**.

30.1 Overview

30.1.1 Functional overview of the CTSU

Table 30 - 1 lists the CTSU functions and **Figure 30 - 3** shows a block diagram of the CTSU. **Figure 30 - 4** shows the configuration of clock signals for sensor drive pulse output.

Table 30 - 1 Functional Overview of the CTSU

Item		Configuration
Module type name		CTSU2L
Operating voltage		V_{DD} = 1.8 to 5.5 V
Operating clock**Note**		f_{CLK} (at least 1 MHz), $f_{CLK}/2$, $f_{CLK}/4$, or $f_{CLK}/8$
Pins	Electrostatic capacitance measurement	TSm (m = 00 to 15, 20 to 35) up to 32 channels
	Connection pin to capacitor for measurement secondary power	TSCAP (10 nF) We recommend connecting a 10-nF 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	Current on a measurement pin is measured.
Calibration mode		Characteristic correction of the current control oscillator for measurement
Noise prevention		• Spread spectrum for sensor drive pulses • Random phase-shifting for sensor drive pulses • Majority decision from multi-frequency measurement
Adjustment for each pin		• Offset current adjustment • Sensor drive pulse frequency specification • Measurement time specification
Measurement start conditions		• Software trigger • External trigger (event input from the ELCL)
Low-power function		SNOOZE mode supported
Request outputs	DTC activation sources	• Request to write to a configuration register of an individual CTSU channel • Request to transfer data measured by the CTSU
	Interrupt source	• Write request interrupt for a configuration register of an individual CTSU channel (INTCTSUWR) • Request to transfer data measured by the CTSU (INTCTSURD) • CTSU measurement end (INTCTSUFN)
Transmission power switching of the mutual capacitance method		The source of power to the pins can be switched at the time of transmission.

Note The accuracy of measurement by the capacitive sensing unit (CTSU2L) depends on the accuracy of the operating clock. When the CTSU2L is to be used, we recommend selecting the high-speed on-chip oscillator clock (f_{IH}) or high-speed system clock (f_{MX}) as the CPU/peripheral hardware clock (f_{CLK}). Do not select the subsystem clock (f_{SUB}) as the CPU/peripheral hardware clock (f_{CLK}).

As shown in **Figure 30 - 3**, the CTSU consists of a status control block, a trigger control block, a clock control block, a channel control block, a port control block, a sensor drive pulse generator, a measurement block, an interrupt control block, a SNOOZE control block, and control registers.

Figure 30 - 3 Block Diagram of the CTSU

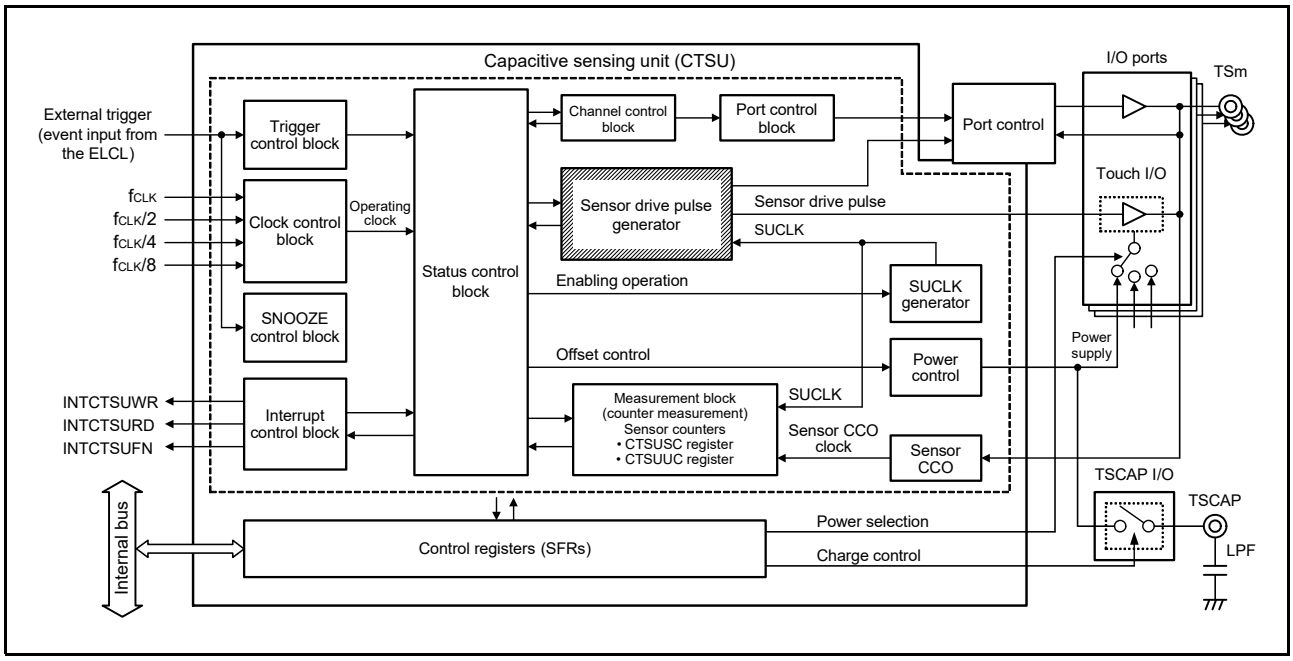


Figure 30 - 4 Configuration of Clock Signals for Sensor Drive Pulse Output

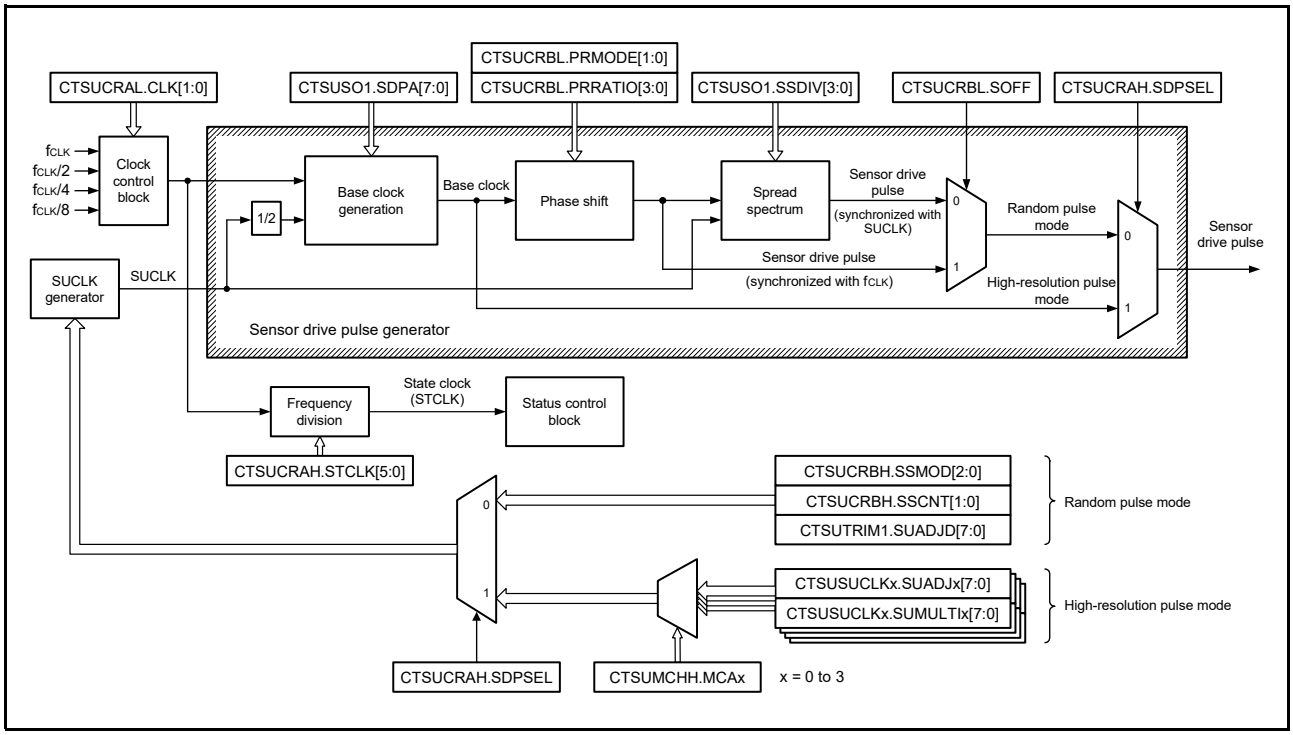


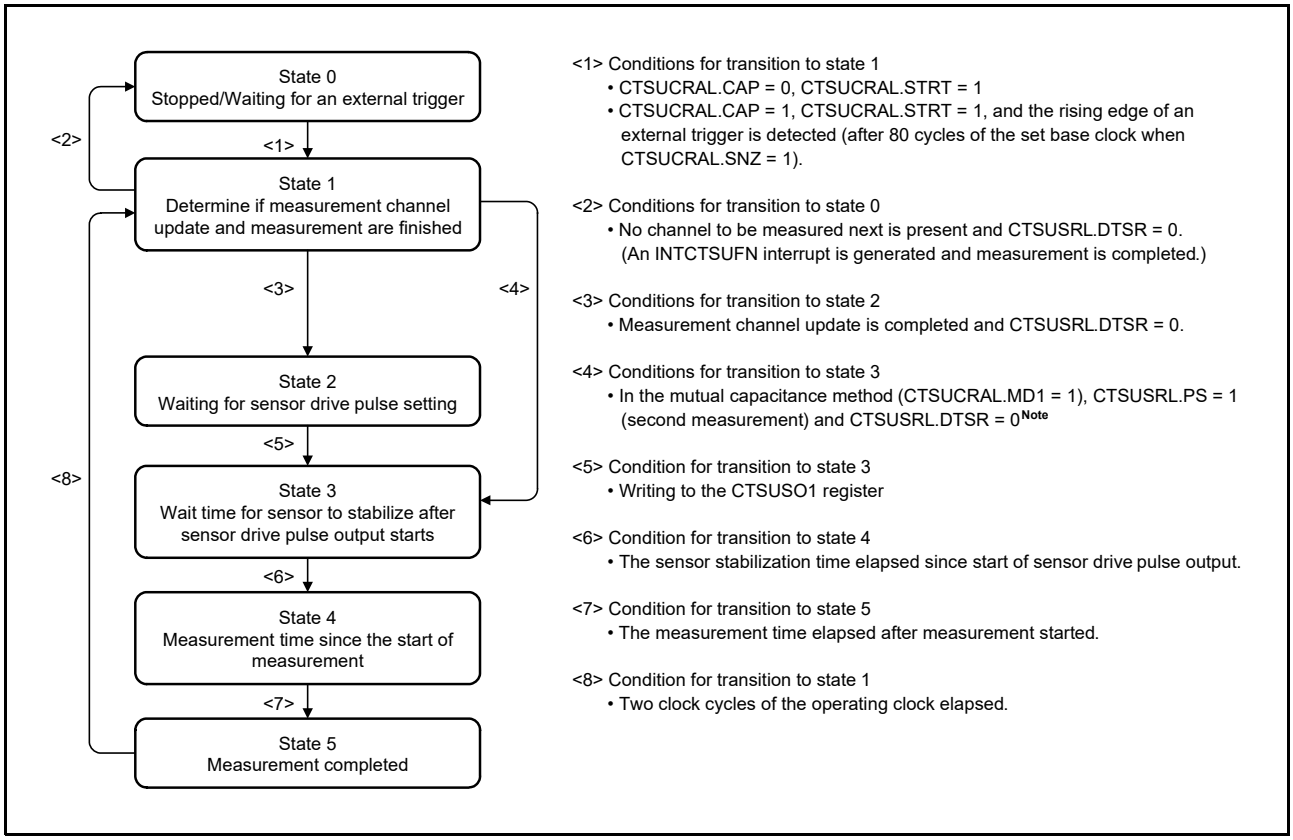
Table 30 - 2 Input and Output Pins of the CTSU

Pin name	Input/output	Function
TSm (m = 00 to 15, 20 to 35)	Input/output	Electrostatic capacitance measurement pins (touch pins)
TSCAP	—	Connection pin to capacitor for measurement secondary power

30.1.2 Measurement states

The CTSU has a state controller. The measurement states are common to both modes of measurement. The measurement status counter indicates the current state of measurement, and its value can be read from the CTSUSRL.STC[2:0] bits. **Figure 30 - 5** shows the transitions between the measurement states.

Figure 30 - 5 Transitions between the Measurement States



Note When CTSUSRL.DTSR = 1, waiting until the reading of the previous measurement result is required.

The status counter makes the transition to state 0 when measurement has been completed on all of the specified measurement channels. The CTSUCRAL.STRT bit is set to 0 by hardware when the software trigger is to be used. When an external trigger is to be used, the STRT bit retains the value 1, and the CTSU waits for the next trigger. When operation is forcibly stopped (by writing 0 to the STRT bit and 1 to the CTSUCRAL.INIT bit at the same time) during measurement or is in the state of waiting for a trigger, the state makes the transition to state 0 and measurement is stopped. If no channels for measurement are set in the CTSUMCHL, CTSUMCHH, CTSUCHACAL, CTSUCHACAH, CTSUCHACBL, CTSUCHTRCAL, CTSUCHTRCAH, and CTSUCHTRCBL registers, a measurement end interrupt (INTCTSUFN) is generated immediately after the transition to state 1, and the state then makes the transition to state 0.

The following are the cases when there are no channels for measurement.

- No target measurement channels are specified in the CTSUCHACAL, CTSUCHACAH, and CTSUCHACBL registers.
- In single scan mode, no channels specified in the CTSUMCHL and CTSUMCHH registers are set as targets for measurement in the CTSUCHACAL, CTSUCHACAH, and CTSUCHACBL registers.
- In the self-capacitance method, there is no reception channel for measurement in the following combinations of registers: CTSUCHACAL/CTSUCHACAH/CTSUCHACBL and CTSUCHTRCAL/CTSUCHTRCAH/CTSUCHTRCBL.
- In the mutual capacitance method, there is no transmission channel or reception channel for measurement in the following combinations of registers: CTSUCHACAL/CTSUCHACAH/CTSUCHACBL and CTSUCHTRCAL/CTSUCHTRCAH/CTSUCHTRCBL.

30.2 Registers for Controlling the Capacitive Sensing Unit

The following registers are used to control the capacitive sensing unit.

- Peripheral enable register 1 (PER1)
- Peripheral reset control register 1 (PRR1)
- CTSU control registers AL and AH (CTSUCRAL, CTSUCRAH)
- CTSU control registers BL and BH (CTSUCRBL, CTSUCRBH)
- CTSU measurement channel registers L and H (CTSUMCHL, CTSUMCHH)
- CTSU channel enable control registers AL, AH, BL, and BH (CTSUCHACAL, CTSUCHACAH, CTSUCHACBL, CTSUCHACBH)
- CTSU channel transmit/receive control registers AL, AH, BL, and BH (CTSUCHTRCAL, CTSUCHTRCAH, CTSUCHTRCBL, CTSUCHTRCBH)
- CTSU status register L (CTSUSRL)
- CTSU sensor offset registers 0 and 1 (CTSUSO0, CTSUSO1)
- CTSU sensor counter registers (CTSUSC, CTSUUC)
- CTSU calibration registers L and H (CTSUDBGR0, CTSUDBGR1)
- CTSU sensor unit clock control registers AL, AH, BL, and BH (CTSUSUCLK0, CTSUSUCLK1, CTSUSUCLK2, CTSUSUCLK3)
- CTSU trimming registers AL and AH (CTSUTRIM0, CTSUTRIM1)
- CTSU trimming registers BL and BH (CTSUTRIM2, CTSUTRIM3)
- Port mode registers (PMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)

30.2.1 Peripheral enable register 1 (PER1)

This register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise. Be sure to set the CTSUEN bit of this register to 1 to use the capacitive sensing unit. The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 30 - 6 Format of Peripheral Enable Register 1 (PER1)

Address: F00FAH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER1	DACEN	SMSEN	CMPEN	TML32EN	DTCEN	UTAEN **Note 1**	REMCEN **Note 2**	CTSUEN

CTSUEN	Control of supply of an input clock to the capacitive sensing unit
0	Stops supply of an input clock. • The SFRs used by the capacitive sensing unit cannot be written.**Note 3**
1	Enables supply of an input clock. • The SFRs used by the capacitive sensing unit can be read and written.

Note 1. This bit is only present in the 36- to 128-pin R7F100G and 32- to 48-pin R7F104G products.
Note 2. This bit is only present in the 32- to 128-pin R7F100G and 32- to 48-pin R7F104G products.
Note 3. When an SFR used by the CTSU2L is read, the value returned is 00H or 0000H regardless of the setting value.

Caution Be sure to leave the following bits set to 0.
Bits 2 and 1 in the 30-pin R7F100G products
Bit 2 in the 32-pin R7F100G products

30.2.2 Peripheral reset control register 1 (PRR1)

The PRR1 register is used to control resetting of the on-chip peripheral modules. Each bit in this register controls resetting and release from the reset state of the corresponding on-chip peripheral module. The PRR1 register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 30 - 7 Format of Peripheral Reset Control Register 1 (PRR1)

Address: F00FBH
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	<5>	<4>	3	2	<1>	<0>
PRR1	DACRES	SMSRES	CMPRES	TML32RES	0	0	REMCRES **Note**	CTSURES

CTSURES	Control resetting of the capacitive sensing unit
0	The capacitive sensing unit is released from the reset state.
1	The capacitive sensing unit is in the reset state. • The SFRs used by the capacitive sensing unit are initialized.

Note This bit is only present in the 32- to 128-pin R7F100G and 32- to 48-pin R7F104G products.

Caution 1. Be sure to leave bits 3 and 2 set to 0.
Caution 2. Be sure to leave bit 1 in the 30-pin R7F100G products set to 0.

30.2.3 CTSU control registers AL and AH (CTSUCRAL, CTSUCRAH)

These registers can be set by a 16-bit memory manipulation instruction. Each of these registers can also be set by the following way.

- The higher- and lower-order bits of the CTSUCRAL register can respectively be set as the CTSUCR1 and CTSUCR0 registers by a 1-bit or 8-bit memory manipulation instruction.
- The higher- and lower-order bits of the CTSUCRAH register can respectively be set as the CTSUCR3 and CTSUCR2 registers by a 1-bit or 8-bit memory manipulation instruction.

The value of each register following a reset is 0000H.

Figure 30 - 8 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) (1/8)

Address: F0500H, F0501H (CTSUCRAL), F0502H, F0503H (CTSUCRAH)
After reset: 0000H
R/W: 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]		LOAD1	LOAD0	ATUNE2	0

Symbol	15	14	13	12	11	10	9	8
CTSUCRAL	MD1	MD0	CLK[1:0]		ATUNE1	ATUNE0	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 the TSCAP pin.
1	Selects measurement pin.

This bit is valid only when DCMODE = 1 to switch the pin for feedback to measurement power.

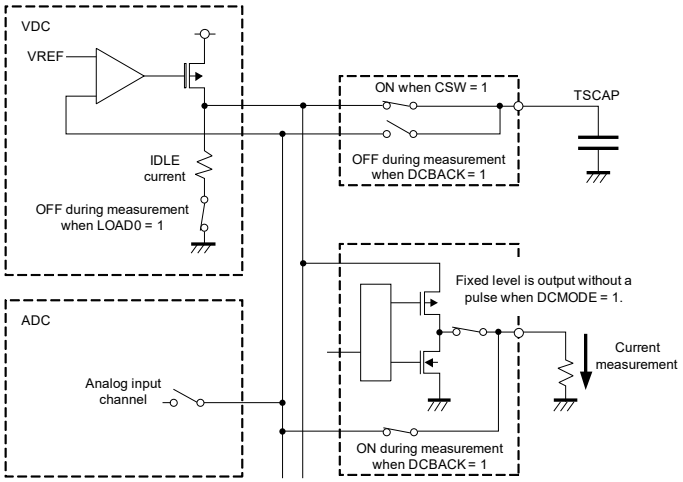


Figure 30 - 8 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) (2/8)

DCMODE	Current Measurement Mode Select
0	Normal mode
1	Current measurement mode
Setting this bit to 1 enables measurement of the current without using sensor drive pulses. A fixed level is output through the reception pin for measuring the current in accord with the value in the CTSUDBGR0.IOC bit, and the sensor drive pulses are stopped. CTSUDBGR0.IOC = 0: High-level output CTSUDBGR0.IOC = 1: Low-level output	

STCLK[5:0]						STCLK Select
0	0	0	0	0	0	Operating clock divided by 2
0	0	0	0	0	1	Operating clock divided by 4
0	0	0	0	1	0	Operating clock divided by 6
0	0	0	0	1	1	Operating clock divided by 8
:	:	:	:	:	:	:
1	1	1	1	1	0	Operating clock divided by 126
1	1	1	1	1	1	Operating clock divided by 128
These bits set the division value for the operating clock to obtain the state clock (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 these bits so that STCLK will be 0.5 MHz (2 µs) is recommended.						

PCSEL	Booster Clock Select
0	Sensor drive pulse
1	STCLK
This bit selects the clock for the voltage boost circuit.	

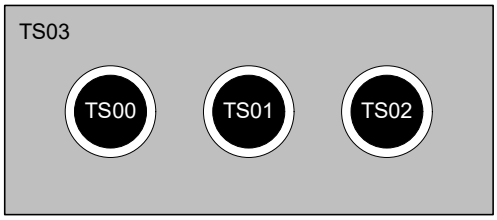
Figure 30 - 8 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) (3/8)

SDPSEL	Sensor Drive Pulse and SUCLK Select
0	Random pulse mode • Sensor drive pulse Sensor drive pulses synchronized with f_{CLK} are generated with randomly extended widths at low level by using pseudo-random numbers based on a base clock which is the operating clock divided by the value of the CTSUSO1.SDPA[7:0] bits. • SUCLK SUCLK is generated in accord with the settings of the CTSUCRBH.SSCNT[1:0], CTSUCRBH.SSMOD[2:0], and CTSUTRIM1.SUADJD[7:0] bits. SUCLK is used for spreading the spectrum of the sensor drive pulses.
1	High-resolution pulse mode • Sensor drive pulse The clock (base clock) obtained by dividing SUCLK by the CTSUSO1.SDPA[7:0] bits is used to produce the sensor drive pulses. • SUCLK Setting the CTSUSUCLKx register causes clock recovery control to proceed and the CTSUSUCLKx.SUADJx[7:0] bits are updated. SUCLK is generated in accord with this setting. Set the CTSUSUCLKx register before setting this bit to 1. Do not change the CTSUSUCLKx register setting while this bit is 1.
The SDPSEL bit is used to switch between the random pulse mode and high-resolution pulse mode. Setting this bit determines how sensor drive pulses and SUCLK are generated.	

Figure 30 - 8 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) (4/8)

POSEL[1:0]		Non-measured Pin Output Select
0	0	Low-level output (from the I/O port regardless of the settings of the TXVSEL and TXVSEL2 bits)
0	1	Hi-Z
1	0	Low-level output (by the settings of the TXVSEL and TXVSEL2 bits)
1	1	Outputs a pulse in phase with the transmit channel (by the settings of the TXVSEL and TXVSEL2 bits)

The POSEL[1:0] bits select the non-measured pin output.
[Example] When measurement proceeds by the self-capacitance method using the electrode shown below to enable the active shield function:
- Shield pin: TS03
  Set the transmit/receive control register to 1 to output a transmit pulse to use TS03 as a shield pin.
- Measurement pins: TS00 to TS02
  Set the POSEL[1:0] bits to 11B. A pulse in phase with the shield pin is output from the non-measured pin.



LOAD1	LOAD0	Measurement Load Control
0	0	2.5-μA constant current load mode (for normal measurement)
0	1	No-load mode
1	0	20-μA constant current load mode (for calibration)
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, the amount of resistive load changes depending on the settings of the ATUNE2 and ATUNE1 bits.

ATUNE2	Current Range Switching 2
0	When the ATUNE1 bit is 0: 80 μA When the ATUNE1 bit is 1: 40 μA
1	When the ATUNE1 bit is 0: 20 μA When the ATUNE1 bit is 1: 160 μA

Setting this bit in combination with the ATUNE1 bit specifies the maximum supply current (measurement range) by changing the amount of the resistive load on the measurement power supply.

CTSUCRAH.ATUNE2	CTSUCRAL.ATUNE1	Measurement range
0	0	80 μA
0	1	40 μA
1	0	20 μA
1	1	160 μA

Figure 30 - 8 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) (5/8)

MD1	Measurement Mode Select 1
0	Self-capacitance method Set the pin to be used for reception in measurement (CTSUCHTRCAL.CHTRCm = 0, CTSUCHTRCAH.CHTRCm = 0, CTSUCHTRCBL.CHTRCm = 0). Set the pin to be used as an active shield for transmission (by setting CHTRCm to 1). If a pin set for transmission is present, the transmission pin outputs a pulse in the same phase as that for the channel being measured. When all TSm pins are set for reception, the measurement of reception pins only proceeds.
1	Mutual capacitance method The measurement proceeds with the TSm pin set for transmission (CHTRCm = 1) and the TSm pin set for reception (CHTRCm = 0). When all CHTRCm bits are set to 0, no measurement proceeds.
The MD1 bit selects the self-capacitance method or the mutual capacity method.	

MD0	Measurement Mode Select 0
0	Single scan mode
1	Multi-scan mode
The MD0 bit selects single scan mode or multi-scan mode.	

CLK[1:0]		Operating Clock Select
0	0	f_{CLK}
0	1	$f_{CLK}/2$
1	0	$f_{CLK}/4$
1	1	$f_{CLK}/8$
The CLK[1:0] bits are used to divide the CPU/peripheral hardware clock (f_{CLK}) by using the prescaler.		

ATUNE1	Current Range Switching 1
0	When the ATUNE2 bit is 0: 80 μA When the ATUNE2 bit is 1: 40 μA
1	When the ATUNE2 bit is 0: 20 μA When the ATUNE2 bit is 1: 160 μA
Setting this bit in combination with the ATUNE2 bit specifies the maximum supply current (measurement range) by changing the amount of the resistive load on the measurement power supply. For details, see the description for the ATUNE2 bit.	

ATUNE0	Power Supply Voltage Setting
0	Measurement power supply voltage = 1.5 V
1	Measurement power supply voltage = 1.2 V**Note 1**
This bit is used to control the power supply voltage of the CTSU. Set this bit to 1 when V_{DD} < 2.4 V. This bit can be set to either 0 or 1 when 2.4 V ≤ V_{DD}.	

Figure 30 - 8 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) (6/8)

CSW	Secondary Power Supply Capacitor for Measurement Connection
0	Turns off the external capacitor connection switch.
1	Turns on the external capacitor connection switch.
This bit controls charging of the secondary power supply capacitor for measurement connected to the TSCAP pin by turning the capacitor connection switch on or off. After turning on the capacitor connection switch, wait for a certain amount of time until the capacitor connected to the TSCAP pin has been charged, and then start the measurement (STRT = 1). Before setting this bit to 1, use the I/O port pin assigned to TSCAP (P30) to output a low-level signal, and fully discharge the charge from the secondary measurement power supply capacitor which is connected to the TSCAP pin. Set this 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 this bit to 1.	

PON	Measurement Power On
0	Power off
1	Power on
The PON bit controls the supply of the power (1.5 V) for measurement. Before setting this bit to 1, set the PUMPON bit as required and set the CSW bit to 1. Do not set this bit to 1 while the CSW bit is set to 0.	

TXVSEL	TXVSEL2	Power for the Transmission Pin Select
0	0	Not recommended
1	0	Power for the internal logic circuits Use this setting when the active shield is to be used.
0	1	Dedicated power for the CTSU Use this setting when the mutual capacitance method is to be used.
1	1	

PUMPON	Boost Circuit Control
0	Disables power boosting.
1	Enables power boosting.
The PUMPON bit controls power boosting. Set this bit to 1 when V_{DD} < 4.5 V.	

INIT	Initialization of the Control Block
0	—
1	Initializes the internal control registers.
Setting this bit to 1 initializes the following registers. • CTSUSC, CTSUUC • CTSUMCHL, CTSUMCHH • All bits in CTSUSRL except bits ICOMP0 and ICOMP1 To forcibly stop operation, be sure to set the STRT bit to 0 and the INIT bit to 1 at the same time. Doing so stops the measurement and initializes the internal control registers. Do not set the STRT bit to 1 (starting measurement by the CTSU) and the INIT bit to 1 at the same time. This bit is read as 0.	

Figure 30 - 8 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) (7/8)

SNZ	SNOOZE Enable
0	Disables the SNOOZE mode.
1	Enables the SNOOZE mode.

Setting this bit to 1 places the CTSU in a suspended state, which reduces power consumption while on standby from measurement. The SNZ bit enables measurement in SNOOZE mode when an external trigger is selected (CAP = 1) as the condition for starting measurement.

- When an external trigger is selected as the condition for starting the measurement (CAP = 1)
  When the SNOOZE mode is to be used, set this bit to 1, then set STRT to 1 (starting measurement by the CTSU) to place the CPU in the STOP mode. If an external trigger is detected while in STOP mode, the CTSU is placed in the SNOOZE mode and starts measurement. After a measurement end interrupt, use software to clear this bit to 0.
- When a software trigger is selected as the condition for starting the measurement (CAP = 0)
  The CTSU is placed in the suspended state (low-power mode) under the conditions listed in the table below. An interrupt is used to cause the CPU to resume from the STOP mode while the CTSU is in the suspended state (low-power mode). To start measurement following resumption by the CPU, set this bit to 0 and then set the STRT bit to 1 to start measurement by the CTSU.

PON	SNZ	CAP	STRT	External trigger	State of the CTSU hardware
0	0	0	0	—	Stopped
1	0	0	0	—	State of waiting for the start of measurement
1	0	0	1	—	Measurement is proceeding in the normal mode
1	1	1	0	—	Preparing for an external trigger to start measurement
1	1	1	1	Not present (waiting)	Suspended (waiting for an external trigger)
1	1	1	1	Present (operating)	Measurement in the SNOOZE mode
1	1	0	0	—	Suspended (low-power mode)
Other than above					Setting prohibited

When using the SNOOZE mode function with the high-resolution pulse mode selected for the sensor drive pulses (SDPSEL = 1), the clock of the voltage boosting circuit cannot be selected for the sensor drive pulses (PCSEL = 0).

CAP	Measurement Start Trigger Select
0	Software trigger
1	External trigger (event input from the ELCL)

The CAP bit specifies the measurement start condition. For details, see the description for the STRT bit.

Figure 30 - 8 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) (8/8)

STRT	Measurement Start
0	Stops measurement
1	Starts measurement (measurement in progress or waiting for an external trigger)

The STRT bit specifies whether to start or stop CTSU measurement.

- When a software trigger is selected as the condition for starting the measurement (CAP = 0)
  Writing 1 to the STRT bit starts measurement. At the end of measurement, this bit is cleared to 0 by hardware.
- When an external trigger is selected as the condition for starting the measurement (CAP = 1)
  Writing 1 to this bit drives the CTSU into the state of waiting for an external trigger. Measurement starts on detection of an external trigger. After the measurement is completed, the CTSU waits for the next external trigger. To release the CTSU from the state of waiting for an external trigger, set this bit to 0 and the INIT bit to 1 at the same time.

The following table shows the CTSU states.

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 2

If 1 is written to the STRT bit by software when the STRT bit is already set to 1, the writing is ignored and operation continues. To have software forcibly stop operation while the STRT bit is set to 1, be sure to set the STRT bit to 0 and the INIT bit to 1 at the same time.

Note 1. The active shield cannot be used when ATUNE0 = 1.

Note 2. The state in terms of the two alternatives can be read from the CTSUSRL.STC[2:0] bits.

- CTSUSRL.STC[2:0] ≠ 000B: Measurement in progress
- CTSUSRL.STC[2:0] = 000B: Waiting for an external trigger

Caution Set the bits of this register other than the STRT and INIT bits while STRT = 0 (measurement stopped).

Remark 1. x = 0 to 3

Remark 2. m = 00 to 15, 20 to 35

30.2.4 CTSU control registers BL and BH (CTSUCRBL, CTSUCRBH)

These registers can be set by a 16-bit memory manipulation instruction. Each of these registers can also be set by the following way.

- The higher-order bits of the CTSUCRBL register can be set as the CTSUSST register by an 8-bit memory manipulation instruction, whereas the lower-order bits can be set as the CTSUSDPRS register by a 1-bit or 8-bit memory manipulation instruction.
- The higher-order bits of the CTSUCRBH register can be set as the CTSUDCLKC register by an 8-bit memory manipulation instruction.

The value of each register following a reset is 0000H.

Figure 30 - 9 Format of CTSU Control Registers BL and BH (CTSUCRBL, CTSUCRBH) (1/3)

Address: F0504H, F0505H (CTSUCRBL), F0506H, F0507H (CTSUCRBH)
After reset: 0000H
R/W: 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

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 Spread Spectrum Control
0	0	CTSUCRAH.SDPSEL = 0: CTSUTRIMx.SUADJD[7:0] + 00H
		CTSUCRAH.SDPSEL = 1: CTSUSUCLKx.SUADJx[7:0] + 00H
0	1	CTSUCRAH.SDPSEL = 0: CTSUTRIMx.SUADJD[7:0] + 10H
		CTSUCRAH.SDPSEL = 1: CTSUSUCLKx.SUADJx[7:0] + 20H
1	0	CTSUCRAH.SDPSEL = 0: CTSUTRIMx.SUADJD[7:0] + 20H
		CTSUCRAH.SDPSEL = 1: CTSUSUCLKx.SUADJx[7:0] + 40H
1	1	CTSUCRAH.SDPSEL = 0: CTSUTRIMx.SUADJD[7:0] + 30H
		CTSUCRAH.SDPSEL = 1: CTSUSUCLKx.SUADJx[7:0] + 60H
These bits adjust the frequency of SUCLK.		

Figure 30 - 9 Format of CTSU Control Registers BL and BH (CTSUCRBL, CTSUCRBH) (2/3)

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

The SSMOD[2:0] bits set the number of spread spectrum cycles of SUCLK. The oscillation frequency of SUCLK is determined by the frequency specified by the CTSUTRIM1.SUADJD[7:0] bits and SSCNT[1:0] bits, as well as the setting of this bit.

SST[7:0]	Sensor Stabilization Wait Time Control

The SST[7:0] bits set the period from the start of sensor drive pulse output until the TSCAP pin voltage becomes stable. The TSCAP pin voltage is stabilized by supplying sensor drive pulses. The following shows the correspondence between the setting of the SST[7:0] bits and the number of cycles.

- In random pulse mode (CTSUCRAH.SDPSEL = 0)

  The stabilization wait time is specified in units of the number of cycles of the sensor drive pulse.

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 so that the following relation is satisfied.

  Number of cycles set by CTSUCRBL.SST[7:0] ≥ (PRRATIO[3:0] + 1)

- In high-resolution pulse mode (CTSUCRAH.SDPSEL = 1)

  The stabilization wait time is specified in units of the number of the STCLK cycles.

0	0	0	0	0	0	0	0	Setting prohibited Note 1
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)

PROFF	Pseudo-random Number Disable Control
0	Pseudo-random number control is enabled.
1	Pseudo-random number control is disabled.

The PROFF bit disables the pseudo-random number control. A pseudo-random number is generated per cycle to output a value of 1 or 0 (1-bit random number generation).

Figure 30 - 9 Format of CTSU Control Registers BL and BH (CTSUCRBL, CTSUCRBH) (3/3)

SOFF[Note 2]	Spread Spectrum Disable Control
0	Enables the spread spectrum.
1	Disables the spread spectrum.
The SOFF bit enables or disables the spread spectrum to prevent synchronous noise. Set this bit to 1 when sensor drive pulses (synchronized with fCLK) are to be used. This bit is valid when CTSUCRAH.SDPSEL = 0 (random pulse mode).	

PRMODE[1:0][Note 2]		Pseudo-random Number Generation Cycle
0	0	255 cycles
0	1	63 cycles
1	0	31 cycles
1	1	3 cycles
The PRMODE[1:0] bits set the number of cycles for updating the pseudo-random number generation. The number of basic pulses is twice the number of cycles selected by these bits. These bits are valid when CTSUCRAH.SDPSEL = 0 (random pulse mode).		

PRRATIO[3:0][Note 2]	Phase Shift Frequency
The PRRATIO[3:0] bits set the phase shift cycle for pseudo-random number generation. The number of measurement pulses and measurement time are specified as follows depending on the setting of these bits. Number of measurement pulses = number of basic pulses (set by the PRMODE[1:0] bits) × (value of the PRRATIO[3:0] bits + 1) Measurement time = (number of measurement pulses + (number of basic pulses – 2) × 0.25) × base clock cycle These bits are valid when CTSUCRAH.SDPSEL = 0 (random pulse mode).	

Note 1. Do not set the SST[7:0] bits to 00H when CTSUCRAH.SDPSEL = 1.

Note 2. The SOFF, PRMODE[1:0], and PRRATIO[3:0] bits are only valid when the CTSUCRAH.SDPSEL bit is 0 (random pulse mode).

Caution **Set the CTSUCRBL and CTSUCRBH registers when the CTSUCRAL.STRT bit is 0.**

Remark x = 0 to 3

30.2.5 CTSU measurement channel registers L and H (CTSUMCHL, CTSUMCHH)

These registers can be set by a 16-bit memory manipulation instruction. Each of these registers can also be set by the following way.

- The higher- and lower-order bits of the CTSUMCHL register can respectively be set as the CTSUMCH1 and CTSUMCH0 registers by an 8-bit memory manipulation instruction.
- The lower-order bits of the CTSUMCHH register can be set as the CTSUMFAF register by a 1-bit or 8-bit memory manipulation instruction.

The value of the CTSUMCHL register following a reset is 3F3FH, and that of the CTSUMCHH register is 0000H.

Figure 30 - 10 Format of CTSU Measurement Channel Registers L and H (CTSUMCHL, CTSUMCHH) (1/2)

Address: F0508H, F0509H (CTSUMCHL), F050AH, F050BH (CTSUMCHH)
After reset: 3F3FH (CTSUMCHL), 0000H (CTSUMCHH)
R/W: 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

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
0	Disabled
1	Enabled

The MCAx bit sets the clocks to be used when the measurement is in progress by using SUCLKs with two or more frequencies (multi-clock measurement). For measurement using three clocks, set three bits out of the four bits (MCA3 to MCA0) to 1. When measurement starts with two or more MCAx bits enabled, valid channels are scanned in ascending order from the MCA0 bit (SUCLK0). After a channel has been measured, if the next valid clock (SUCLKx) is present, the same channel is measured using that clock. If no valid clock is present, the next channel is measured. The SUCLK frequency is set by the CTSUSUCLKx register. The following shows the relationship between the MCAx bits and the CTSUSUCLKx register.

- MCA0: SUCLK0 set by the CTSUSUCLK0 register
- MCA1: SUCLK1 set by the CTSUSUCLK1 register
- MCA2: SUCLK2 set by the CTSUSUCLK2 register
- MCA3: SUCLK3 set by the CTSUSUCLK3 register

Figure 30 - 10 Format of CTSU Measurement Channel Registers L and H (CTSUMCHL, CTSUMCHH) (2/2)

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 the channels for which the measurement is disabled by the CTSUCHACAH, CTSUCHACAL, or CTSUCHACBL register. If such channels are set, measurement ends immediately after it starts.
- In multi-scan mode (CTSUCRAL.MD0 = 1)
  The MCH0[5:0] bits indicate the value of the transmit channel that is being measured, and writing to these bits has no effect.

0	0	0	0	0	0	TS00
0	0	0	0	0	1	TS01
0	0	0	0	1	0	TS02
0	0	0	0	1	1	TS03
:	:	:	:	:	:	:
0	0	0	1	1	1	TS15
0	0	1	0	0	0	Setting prohibited
:	:	:	:	:	:	
0	0	1	0	1	1	
0	0	1	1	0	0	TS20
:	:	:	:	:	:	:
1	0	0	0	1	0	TS34
1	0	0	0	1	1	TS35
1	0	0	1	0	0	Setting prohibited
:	:	:	:	:	:	
1	1	1	1	1	1	

MCH0[5:0]	Measurement Channel 0

- In single scan mode (CTSUCRAL.MD0 = 0)
  The MCH0[5:0] bits set the reception channel to be measured. Do not set the channels for which the measurement is disabled by the CTSUCHACAH, CTSUCHACAL, or CTSUCHACBL register. If such channels are set, measurement ends immediately after it starts.
- In multi-scan mode (CTSUCRAL.MD0 = 1)
  The MCH1[5:0] bits indicate the value of the reception channel that is being measured, and writing to these bits has no effect.

0	0	0	0	0	0	TS00
0	0	0	0	0	1	TS01
0	0	0	0	1	0	TS02
0	0	0	0	1	1	TS03
:	:	:	:	:	:	:
0	0	0	1	1	1	TS15
0	0	1	0	0	0	Setting prohibited
:	:	:	:	:	:	
0	0	1	0	1	1	
0	0	1	1	0	0	TS20
:	:	:	:	:	:	:
1	0	0	0	1	0	TS34
1	0	0	0	1	1	TS35
1	0	0	1	0	0	Setting prohibited
:	:	:	:	:	:	
1	1	1	1	1	1	

Caution **Do not modify the CH1[5:0] and MCH0[5:0] bits during measurement (CTSUCRAL.STRT = 1). Otherwise, operation is not guaranteed. The value of the CH1[5:0] and MCH0[5:0] bits is 111111B when the measurement is stopped.**

Remark x = 0 to 3

30.2.6 CTSU channel enable control registers AL, AH, BL, and BH (CTSUCHACAL, CTSUCHACAH, CTSUCHACBL, CTSUCHACBH)

These registers can be set by a 16-bit memory manipulation instruction. Each of these registers can also be set by the following way.

- The higher- and lower-order bits of the CTSUCHACAL register can respectively be set as the CTSUCHAC1 and CTSUCHAC0 registers by a 1-bit or 8-bit memory manipulation instruction.
- The higher- and lower-order bits of the CTSUCHACAH register can respectively be set as the CTSUCHAC3 and CTSUCHAC2 registers by a 1-bit or 8-bit memory manipulation instruction.
- The higher- and lower-order bits of the CTSUCHACBL register can respectively be set as the CTSUCHAC5 and CTSUCHAC4 registers by a 1-bit or 8-bit memory manipulation instruction.
- The higher- and lower-order bits of the CTSUCHACBH register can respectively be set as the CTSUCHAC7 and CTSUCHAC6 registers by a 1-bit or 8-bit memory manipulation instruction.

The value of each register following a reset is 0000H.

Figure 30 - 11 Format of CTSU Channel Enable Control Registers AL, AH, BL, and BH (CTSUCHACAL, CTSUCHACAH, CTSUCHACBL, CTSUCHACBH)

Address: F050CH, F050DH (CTSUCHACAL), F050EH, F050FH (CTSUCHACAH), F0510H, F0511H (CTSUCHACBL), F0512H, F0513H (CTSUCHACBH)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
CTSUCHACAH	CHAC31	CHAC30	CHAC29	CHAC28	CHAC27	CHAC26	CHAC25	CHAC24
	7	6	5	4	3	2	1	0
	CHAC23	CHAC22	CHAC21	CHAC20	CHAC19	CHAC18	CHAC17	CHAC16

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

Symbol	15	14	13	12	11	10	9	8
CTSUCHACBH	CHAC63	CHAC62	CHAC61	CHAC60	CHAC59	CHAC58	CHAC57	CHAC56
	7	6	5	4	3	2	1	0
	CHAC55	CHAC54	CHAC53	CHAC52	CHAC51	CHAC50	CHAC49	CHAC48

Symbol	15	14	13	12	11	10	9	8
CTSUCHACBL	CHAC47	CHAC46	CHAC45	CHAC44	CHAC43	CHAC42	CHAC41	CHAC40
	7	6	5	4	3	2	1	0
	CHAC39	CHAC38	CHAC37	CHAC36	CHAC35	CHAC34	CHAC33	CHAC32

CHACm	Channel Enable Control
0	Disables measurement.
1	Enables measurement.
The CHACm bit enables or disables the measurement of the TSm pin. Setting the CHACm bit to 1 enables the capacitance measurement of the TSm pin (both transmission and reception).	

Caution 1. Make sure that the bits to which no measurement pins have been assigned are set to 0.

Caution 2. Set the CTSUCHACAH, CTSUCHACAL, CTSUCHACBH, and CTSUCHACBL registers when CTSUCRAL.STRT is 0.

Remark m = 00 to 15, 20 to 35

30.2.7 CTSU channel transmit/receive control registers AL, AH, BL, and BH (CTSUCHTRCAL, CTSUCHTRCAH, CTSUCHTRCBL, CTSUCHTRCBH)

These registers can be set by a 16-bit memory manipulation instruction. Each of these registers can also be set by the following way.

- The higher- and lower-order bits of the CTSUCHTRCAL register can respectively be set as the CTSUCHTRC1 and CTSUCHTRC0 registers by a 1-bit or 8-bit memory manipulation instruction.
- The higher- and lower-order bits of the CTSUCHTRCAH register can respectively be set as the CTSUCHTRC3 and CTSUCHTRC2 registers by a 1-bit or 8-bit memory manipulation instruction.
- The higher- and lower-order bits of the CTSUCHTRCBL register can respectively be set as the CTSUCHTRC5 and CTSUCHTRC4 registers by a 1-bit or 8-bit memory manipulation instruction.
- The higher- and lower-order bits of the CTSUCHTRCBH register can respectively be set as the CTSUCHTRC7 and CTSUCHTRC6 registers by a 1-bit or 8-bit memory manipulation instruction.

The value of each register following a reset is 0000H.

Figure 30 - 12 Format of CTSU Channel Transmit/Receive Control Registers AL, AH, BL, and BH (CTSUCHTRCAL, CTSUCHTRCAH, CTSUCHTRCBL, CTSUCHTRCBH)

Address: F0514H, F0515H (CTSUCHTRCAL), F0516H, F0517H (CTSUCHTRCAH), F0518H, F0519H (CTSUCHTRCBL), F051AH, F051BH (CTSUCHTRCBH)

After reset: 0000H

R/W: R/W

Symbol	15	14	13	12	11	10	9	8
CTSUCHTRCAH	CHTRC31	CHTRC30	CHTRC29	CHTRC28	CHTRC27	CHTRC26	CHTRC25	CHTRC24
	7	6	5	4	3	2	1	0
	CHTRC23	CHTRC22	CHTRC21	CHTRC20	CHTRC19	CHTRC18	CHTRC17	CHTRC16

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

Symbol	15	14	13	12	11	10	9	8
CTSUCHTRCBH	CHTRC63	CHTRC62	CHTRC61	CHTRC60	CHTRC59	CHTRC58	CHTRC57	CHTRC56
	7	6	5	4	3	2	1	0
	CHTRC55	CHTRC54	CHTRC53	CHTRC52	CHTRC51	CHTRC50	CHTRC49	CHTRC48

Symbol	15	14	13	12	11	10	9	8
CTSUCHTRCBL	CHTRC47	CHTRC46	CHTRC45	CHTRC44	CHTRC43	CHTRC42	CHTRC41	CHTRC40
	7	6	5	4	3	2	1	0
	CHTRC39	CHTRC38	CHTRC37	CHTRC36	CHTRC35	CHTRC34	CHTRC33	CHTRC32

CHTRCm	Channel Transmit/Receive Control
0	Reception
1	Transmission
The CHTRCm bit assigns the TSm pin for use in transmission or reception. When the self-capacitance method is in use (CTSUCRAL.MD1 = 0), setting this bit to 1 (transmission) allows the corresponding pin to be used as an active shield signal output.**Note**	

Note When the active shield output is to be used, do not set multiple bits to 1 at the same time.

Caution 1. Make sure that the bits to which no measurement pins have been assigned are set to 0.

Caution 2. Set the CTSUCHTRCAH, CTSUCHTRCAL, CTSUCHTRCBH, and CTSUCHTRCBL registers when CTSUCRAL.STRT is 0.

Remark m = 00 to 15, 20 to 35

30.2.8 CTSU status register L (CTSUSRL)

This register can be set by a 16-bit memory manipulation instruction. The higher- and lower-order bits of this register can respectively be set as the CTSUST and CTSUST1 registers by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 0000H.

Figure 30 - 13 Format of CTSU Status Register L (CTSUSRL) (1/3)

Address: F051CH, F051DH
After reset: 0000H
R/W: 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[Note 1]	Mutual Capacitance Measurement State
0	First measurement
1	Second measurement
In the mutual capacitance method, which handles two measurements (CTSUCRAL.MD1 = 1), the PS bit indicates whether the measurement for each channel is the first or second measurement. The value of this bit is 0 when measurement is stopped or in self-capacitance method (CTSUCRAL.MD1 = 0).	

SUCKOVF	SUCLK Counter Overflow
0	No overflow occurred.
1	An overflow occurred.
The SUCKOVF bit indicates whether the SUCLK counter has overflowed. When an overflow occurs, the result of measurement is read from the CTSUUC register 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. [Clearing conditions] • 0 has been written to SUCKOVF after reading SUCKOVF = 1 • The CTSUCRAL.INIT bit is set to 1[Note 2] [Setting condition] • The CTSUUC register overflows during measurement	

Figure 30 - 13 Format of CTSU Status Register L (CTSUSRL) (2/3)

SENSOVF	Sensor Counter Overflow
0	No overflow occurred.
1	An overflow occurred.
The SENSOVF bit indicates whether the sensor counter has overflowed. When an overflow occurs, the result of measurement is read from the CTSUSC register 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. [Clearing conditions] • 0 has been written to SUCKOVF after reading SUCKOVF = 1 • The CTSUCRAL.INIT bit is set to 1**Note 2** [Setting condition] • The CTSUUC register overflows during measurement	

DTSR**Note 1**	Data Transfer State
0	The measurement result has been read.
1	The measurement result has not been read.
The DTSR bit indicates whether the measurement result stored in the sensor counter has been read. [Clearing conditions] • Reading data from the CTSUSC or CTSUUC register by using software or by using the DTC • The CTSUCRAL.INIT bit is set to 1 [Setting condition] • The measurement result is stored in the CTSUSC and CTSUUC registers following the completion of measurement.	

STC[2:0]**Note 1**			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
The STC[2:0] bits indicate the current state of measurement. For details, see **30.1.2 Measurement states**.			

ICOMP0 **Note 1**	Abnormal Voltage Flag
0	Normal
1	Abnormal TSCAP voltage
This flag indicates that the output of an abnormal voltage by the power supply used for the measurement. This bit is cleared to 0 when the CTSUCRAL.PON bit is set to 0 or the ICOMPRST bit is set to 1. When this bit is set to 1 (abnormal voltage detection), the CTSUSU register indicates 0000H.	

Figure 30 - 13 Format of CTSU Status Register L (CTSUSRL) (3/3)

ICOMP1 **Note 1**	Abnormal Current Flag
0	Normal
1	Abnormal current
This flag indicates the output of an abnormal current by the power supply used for the measurement. This bit is cleared to 0 when the CTSUCRAL.PON bit is set to 0 or the ICOMPRST bit is set to 1.	

ICOMPRST	ICOMP0 and ICOMP1 Reset
Writing 1 to this bit clears the ICOMP1 and ICOMP0 bits to 0. This bit is read as 0.	

MFC[1:0]		Multi-clock Counter
0	0	MCA0 (SUCLK0)
0	1	MCA1 (SUCLK1)
1	0	MCA2 (SUCLK2)
1	1	MCA3 (SUCLK3)
The MFC[1:0] bits indicate the clock being used for the multi-clock measurement, i.e., when CTSUCRAH.SDPSEL = 1, CTSUCRAH.PCSEL = 1, and two or more of the CTSUMCH.MCAn bits (n = 0 to 3) are set to 1. Only writing 00B is allowed.		

Note 1. This bit is read-only.

Note 2. To clear the SUCKOVF and SENSOVF bits while CTSUCRAL.INIT is set to 1, do so when the setting of the CTSUCRAL.STRT bit is 0.

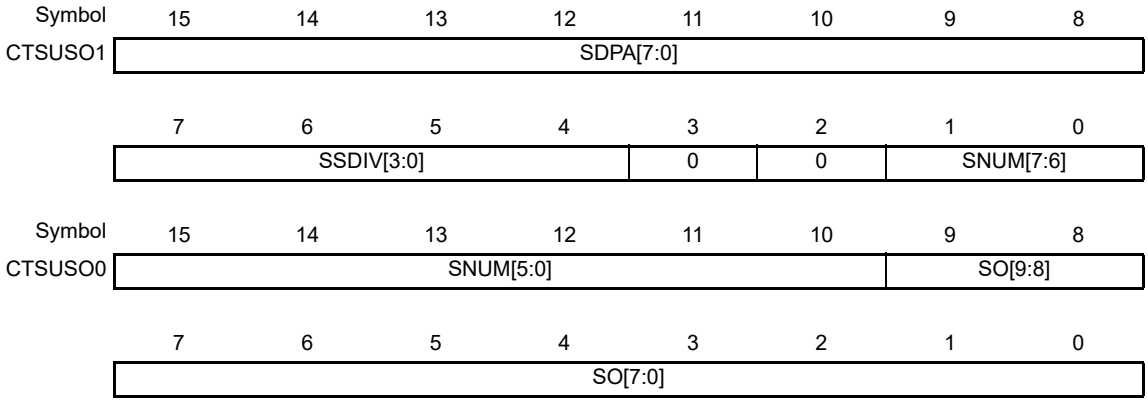
Caution Only write to the CTSUSRL register while the setting the CTSUCRAL.STRT bit is 0.

30.2.9 CTSU sensor offset registers 0 and 1 (CTSUSO0, CTSUSO1)

These registers can be set by a 16-bit memory manipulation instruction. The value of each register following a reset is 0000H.

Figure 30 - 14 Format of CTSU Sensor Offset Registers 0 and 1 (CTSUSO0, CTSUSO1) (1/3)

Address: F0520H, F0521H (CTSUSO0), F0522H, F0523H (CTSUSO1)
After reset: 0000H
R/W: R/W



Write to the CTSUSO0 and CTSUSO1 registers after an INTCTSUWR interrupt is generated. Writing to the CTSUSO1 register causes a transition to state 3. Be sure to write to the CTSUSO0 register before writing to the CTSUSO1 register. Also, set all 16 bits of the CTSUSO1 register at once.

Figure 30 - 14 Format of CTSU Sensor Offset Registers 0 and 1 (CTSUSO0, CTSUSO1) (2/3)

SDPA[7:0]	Base Clock (Sensor Drive Pulse Frequency Divisor) Setting

• In random pulse mode (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 available for setting the voltage stabilization time of the CTSU. When the setting of these bits is n, the base clock is obtained by frequency-dividing the operating clock by 2 (n+1).

0	0	0	0	0	0	0	0	Operating clock divided by 2 Note 1
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

• In high-resolution pulse mode (CTSUCRAH.SDPSEL = 1)

The SUCLK clock is divided to generate a sensor drive pulse. The frequency of SUCLK is determined by the following equation.

SUCLK = STCLK × SUCLK multiplication rate setting made by the CTSUSUCLKx.SUMULTIx[7:0] bits

When the setting of these bits is n, the sensor drive pulses are obtained by frequency-dividing the SUCLK clock by 2 (n+1).

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

Figure 30 - 14 Format of CTSU Sensor Offset Registers 0 and 1 (CTSUSO0, CTSUSO1) (3/3)

SSDIV[3:0]Note 2				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
The SSDIV[3:0] bits set the sampling cycle of the spread spectrum. Set the clock division ratio for resampling the sensor drive pulse (synchronized with f_{CLK}) with SUCLK. Set these bits so that the sampling cycle is less than 1/4 of the sensor drive pulse cycle. Set these bits for the pin to be measured next after an INTCTSUWR interrupt is generated.				

SNUM[7:0]	Measurement Count Setting
• In random pulse mode (CTSUCRAH.SDPSEL = 0) These bits set the measurement count. The SNUM[7:0] bits specify how many times the measurement pulse (basic measurement unit) determined by the CTSUCRBL.PRRATIO[3:0] and CTSUCRBL.PRMODE[1:0] bits is repeated during measurement. Measurement count = (SNUM[5:0] + 1) × 2 Set the SNUM[7:6] bits to 00B. • In high-resolution pulse mode (CTSUCRAH.SDPSEL = 1) These bits set the measurement time based on the STCLK cycle. Measurement time = 8 × (SNUM[7:0] + 1) × STCLK cycle Set these bits for the pin to be measured next after an INTCTSUWR interrupt is generated.	

SO[9:0]	Sensor CCO Offset Adjustment
The SO[9:0] bits adjust the offset of the input current to the sensor CCO. The input current to the sensor CCO, which is generated due to the capacitance in the non-touch state, is set as an offset to prevent the sensor counter from overflowing during measurement. Set these bits to specify the input current offset for the pin to be measured next after the INTCTSUWR interrupt is generated.	

Note 1. When the mutual capacitance method is selected (CTSUCRAL.MD1 = 1) or the spread spectrum is disabled (CTSUCRBL.SOFF = 1), setting SDPA[7:0] = 00000000B is prohibited.

Note 2. The SSDIV[3:0] bits are only valid when the CTSUCRAH.SDPSEL bit is 0 (random pulse mode).

30.2.10 CTSU sensor counter registers (CTSUSC, CTSUUC)

These registers can be set by a 16-bit memory manipulation instruction. The value of each register following a reset is 0000H.

Figure 30 - 15 Format of CTSU Sensor Counter Registers (CTSUSC, CTSUUC)

Address: F0524H, F0525H (CTSUSC), F0526H, F0527H (CTSUUC)
After reset: 0000H
R/W: R

Symbol	15	14	13	12	11	10	9	8
CTSUUC	UC[15:8]							

7	6	5	4	3	2	1	0
UC[7:0]							

Symbol	15	14	13	12	11	10	9	8
CTSUSC	SC[15:8]							

7	6	5	4	3	2	1	0
SC[7:0]							

UC[15:0]	SUCLK Counter
This register indicates the count value of the clock (twice the frequency of SUCLK). This counter indicates FFFFH when an overflow occurs.	

SC[15:0]	Sensor Counter
This register indicates the measurement result. This counter indicates FFFFH when an overflow occurs.	

Read these registers when an INTCTSURD interrupt is generated. Reading these registers the number of times specified by the CTSUDBGR0.CNTRDSEL bit causes measurement to transition to state 0 or 2. The register value is cleared immediately before transition to state 4 (CTSUSRL.STC[2:0] = 100B) in the next measurement. This register is also cleared by writing 1 to the CTSUCRAL.INIT bit.

- When CTSUDBGR0.CNTRDSEL = 0
  Reading the CTSUSC register or CTSUUC register once causes measurement to transition to the next state and both registers are cleared before transition to state 4. If the user require the measurement result, be sure to read the CTSUSC register.
- When CTSUDBGR0.CNTRDSEL = 1
  Reading the CTSUSC or CTSUUC register twice causes measurement to transition to the next state. Use this setting if the user require the results of both the CTSUSC and CTSUUC registers by 16-bit access.

30.2.11 CTSU calibration registers L and H (CTSUDBGR0, CTSUDBGR1)

These registers can be set by a 16-bit memory manipulation instruction. The value of each register following a reset is 0000H.

Figure 30 - 16 Format of CTSU Calibration Registers L and H (CTSUDBGR0, CTSUDBGR1) (1/3)

Address: F0528H, F0529H (CTSUDBGR0), F052AH, F052BH (CTSUDBGR1)
After reset: 0000H
R/W: 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

Symbol	15	14	13	12	11	10	9	8
CTSUDBGR0	0	0	0	0	DCOFF	0	IOC	CNTRDSEL
	7	6	5	4	3	2	1	0
<R>	TSOC	SUCLKEN	0	0	DRV	TSOD	0	0

TXREV	Transmission Pin Inverted Output
0	The pulse output from the transmission pin is not inverted.
1	The pulse output from the transmission pin is inverted.

The TXREV bit is used to invert the output from the transmission pin.
- In the self-capacitance method (CTSUCRAL.MD1 = 0)
  0: A signal of the same phase as the reception pin is output.
  1: A signal in reverse phase to the reception pin is output
- In the mutual capacitance method (CTSUCRAL.MD1 = 1)
  0: First time: A signal of the same phase as the reception pin is output; second time: A signal in reverse phase to the reception pin is output
  1: First time: A signal in reverse phase to the reception pin is output; second time: A signal of the same phase as the reception pin is output

Set this bit to 0 for normal measurement (the setting of 1 is intended for use in debugging).

CCOCALIB	CCO Calibration Mode Select
0	Normal mode
1	Oscillator calibration mode The oscillator calibration mode is used when the external current is compared with the output of the current offset DAC to correct the oscillator characteristics by software.

CCOCLK	CCO Modulation Circuit Clock Select
0	Operating clock (specified by the CTSUCRAL.CLK[1:0] bits)
1	SUCLK

DACCLK	DAC Modulation Circuit Clock Select
0	Operating clock (specified by the CTSUCRAL.CLK[1:0] bits)
1	SUCLK

Figure 30 - 16 Format of CTSU Calibration Registers L and H (CTSUDBGR0, CTSUDBGR1) (2/3)

SUCARRY	CCO Carry Input
0	Normal operation
1	Adds 32 to input data.
The SUCARRY bit is used in the current oscillator input current test mode. Set this bit to 0 when the electrostatic capacitance is to be measured.	

SUMSEL	Current Oscillator Input Current Matrix Test
0	Normal operation
1	Current oscillator input current test mode
The SUMSEL bit is used in the current oscillator input current test mode. Set this bit to 0 when the electrostatic capacitance is to be measured.	

DACCARRY	DAC Upper Current Source Carry Input
0	Normal operation
1	Adds 64 to input data.
The DACCARRY bit is used in the current offset DAC input current test mode. Set this bit to 0 when the electrostatic capacitance is to be measured.	

DACMSEL	Current Offset DAC Current Matrix Test
0	Normal operation
1	Current offset DAC current test mode
The DACMSEL bit controls the current for the current offset DAC. Set this bit to 0 when the electrostatic capacitance is to be measured.	

DCOFF	Down-conversion Disable
0	Normal operation mode
1	Down-conversion disabled
The DCOFF bit disables the output from the LDO. Set this bit to 0 when the electrostatic capacitance is to be measured.	

IOC	Transmission Pin Control
The IOC bit selects the signal level output from the TSm pin when the TSOD bit or the CTSUCRAL.DCMODE bit is set to 1. • When TSOD = 1 0: Low level is output from the TSm pin. 1: High level is output from the TSm pin. • When CTSUCRAL.DCMODE = 1 0: High level is output from the TSm pin. 1: Low level is output from the TSm pin. This bit is ignored when the TSOD and SUCRAL.DCMODE bits are 0.	

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.
Set this bit to 1 to read both the CTSUSC and the CTSUUC registers.	

Figure 30 - 16 Format of CTSU Calibration Registers L and H (CTSUDBGR0, CTSUDBGR1) (3/3)

TSOC	Calibration Setting 2
0	Electrostatic capacitance measurement mode
1	Calibration setting 2
The TSOC bit is used to set calibration of the CTSU. Set this bit to 0 when the electrostatic capacitance is to be measured.	

SUCLKEN	SUCLK Enable
0	SUCLK is disabled.
1	SUCLK is enabled.

<R>

DRV	Calibration Setting 1
0	Electrostatic capacitance measurement mode
1	Calibration setting 1
The DRV bit is used to set calibration of the CTSU. Set this bit to 0 when the electrostatic capacitance is to be measured. Setting this bit to 1 causes measurement to transition to pseudo state 3 or 4, in which operations differ in the following points from those in states 3 and 4 in normal measurement. • The sensor drive pulses are not output. • The measurement counter is not enabled, and the reset signal is not output.	

TSOD	TSm Pin Fixed Output
0	Electrostatic capacitance measurement mode
1	TSm pin fixed output (low- or high-level output)
The TSOD bit is used for calibration of the CTSU. When this bit is set to 1, the level specified by the IOC bit is output from the TSm pin.	

Caution Be sure to leave bits 5 and 4 of the CTSUDBGR0 register set to 0.

30.2.12 CTSU sensor unit clock control registers AL, AH, BL, and BH (CTSUSUCLK0, CTSUSUCLK1, CTSUSUCLK2, CTSUSUCLK3)

These registers can be set by a 16-bit memory manipulation instruction. The value of each register following a reset is 0000H.

Figure 30 - 17 Format of CTSU Sensor Unit Clock Control Registers AL, AH, BL, and BH (CTSUSUCLK0, CTSUSUCLK1, CTSUSUCLK2, CTSUSUCLK3) (1/2)

Address: F052CH, F052DH (CTSUSUCLK0), F052EH, F052FH (CTSUSUCLK1), F0530H, F0531H (CTSUSUCLK2), F0532H, F0533H (CTSUSUCLK3)

After reset: 0000H

R/W: 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]							

Symbol	15	14	13	12	11	10	9	8
CTSUSUCLK0	SUMULTI0[7:0]							

7	6	5	4	3	2	1	0
SUADJ0[7:0]							

Symbol	15	14	13	12	11	10	9	8
CTSUSUCLK3	SUMULTI3[7:0]							

7	6	5	4	3	2	1	0
SUADJ3[7:0]							

Symbol	15	14	13	12	11	10	9	8
CTSUSUCLK2	SUMULTI2[7:0]							

7	6	5	4	3	2	1	0
SUADJ2[7:0]							

Figure 30 - 17 Format of CTSU Sensor Unit Clock Control Registers AL, AH, BL, and BH (CTSUSUCLK0, CTSUSUCLK1, CTSUSUCLK2, CTSUSUCLK3) (2/2)

SUMULTIx[7:0]	STCLK Multiplication Rate Setting

The SUMULTIx[7:0] bits set the multiplication rate for generating SUCLK from STCLK (which is taken to be at 0.5 MHz, obtained by frequency-dividing f_{CLK}). The SUADJx[7:0] bits are updated in accord with SUCLK generated by this setting. The frequency of SUCLK is determined by the following equation. The frequency of the SUCLK must be set in the range between 16 MHz and 32 MHz.

SUCLK = STCLK × SUCLK multiplication rate set by the SUMULTIx[7:0] bits

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

The SUADJx[7:0] bits set the initial value of the SUCLK frequency. The clock recovery function corrects SUCLK for clock drift, resulting in updating of this register value. Do not modify the updated value.

30.2.13 CTSU trimming registers AL and AH (CTSUTRIM0, CTSUTRIM1)

These registers can be set by a 16-bit memory manipulation instruction. Each of these registers can also be set by the following way.

- The higher- and lower-order bits of the CTSUTRIM0 register can respectively be set as the DACTRIM and RTRIM registers by an 8-bit memory manipulation instruction.
- The higher- and lower-order bits of the CTSUTRIM1 register can respectively be set as the TRESULT4 and SUADJD registers by an 8-bit memory manipulation instruction.

Figure 30 - 18 Format of CTSU Trimming Registers AL and AH (CTSUTRIM0, CTSUTRIM1)

Address: F0600H, F0601H (CTSUTRIM0), F0602H, F0603H (CTSUTRIM1)
After reset: **Note**
R/W: 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]							

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
The TRESULT4[7:0] bits hold the coefficient of variation for the 120-kΩ reference load resistance. These bits hold the initial value set as shipped. Do not modify these bits.	

SUADJD[7:0]	SUCLK Frequency Adjustment
The SUADJD[7:0] bits are used to adjust the SUCLK frequency in the random pulse mode (selected by CTSUCRAH.SDPSEL = 0). These bits hold the initial value set as shipped. Do not modify these bits.	

DACTRIM[7:0]	Offset Current Adjustment
The DACTRIM[7:0] bits are used to adjust the DAC coefficient for offset current. These bits hold the initial value set as shipped. Do not modify this value.	

RTRIM[7:0]	Reference Resistance Adjustment
The RTRIM[7:0] bits are used to adjust the value of the internal reference resistor. These bits hold the initial value set as shipped. Do not modify this value.	

Note The value following a reset is the value adjusted as shipped.

30.2.14 CTSU trimming registers BL and BH (CTSUTRIM2, CTSUTRIM3)

These registers can be set by a 16-bit memory manipulation instruction. Each of these registers can also be set by the following way.

- The higher- and lower-order bits of the CTSUTRIM2 register can respectively be set as the TRESULT1 and TRESULT0 registers by an 8-bit memory manipulation instruction.
- The higher- and lower-order bits of the CTSUTRIM3 register can respectively be set as the TRESULT3 and TRESULT2 registers by an 8-bit memory manipulation instruction.

Figure 30 - 19 Format of CTSU Trimming Registers BL and BH (CTSUTRIM2, CTSUTRIM3)

Address: F0604H, F0605H (CTSUTRIM2), F0606H, F0607H (CTSUTRIM3)
After reset: **Note**
R/W: 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]							

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 hold the initial value set as shipped. 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 hold the initial value set as shipped. 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 hold the initial value set as shipped. 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 hold the initial value set as shipped. Do not modify these bits.	

Note The value following a reset is the value adjusted as shipped.

30.2.15 Registers for controlling the port functions multiplexed with the inputs and outputs of the capacitive sensing unit

Set the following registers for controlling the port functions multiplexed with the inputs and outputs of the capacitive sensing unit.

- Port mode registers (PMxx)
- Port mode control A registers (PMCAxx)
- Port mode control T registers (PMCTxx)
- Port mode control E registers (PMCExx)

For details, see the following sections.

- **4.3.1 Port mode registers (PMxx)**
- **4.3.7 Port mode control A registers (PMCAxx)**
- **4.3.8 Port mode control T registers (PMCTxx)**
- **4.3.9 Port mode control E registers (PMCExx)**

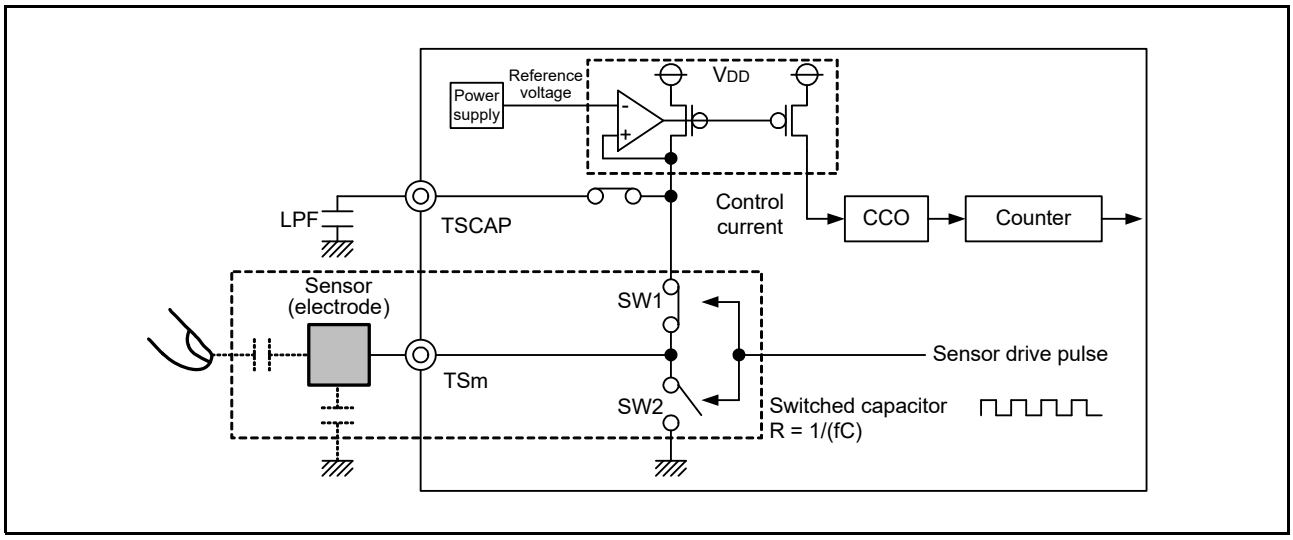
For examples of register settings for controlling port functions, see **4.5.4 Examples of register settings for port and multiplexed functions**.

30.3 Operation

30.3.1 Principles of measurement

Figure 30 - 20 shows the measurement circuit.

Figure 30 - 20 Measurement Circuit



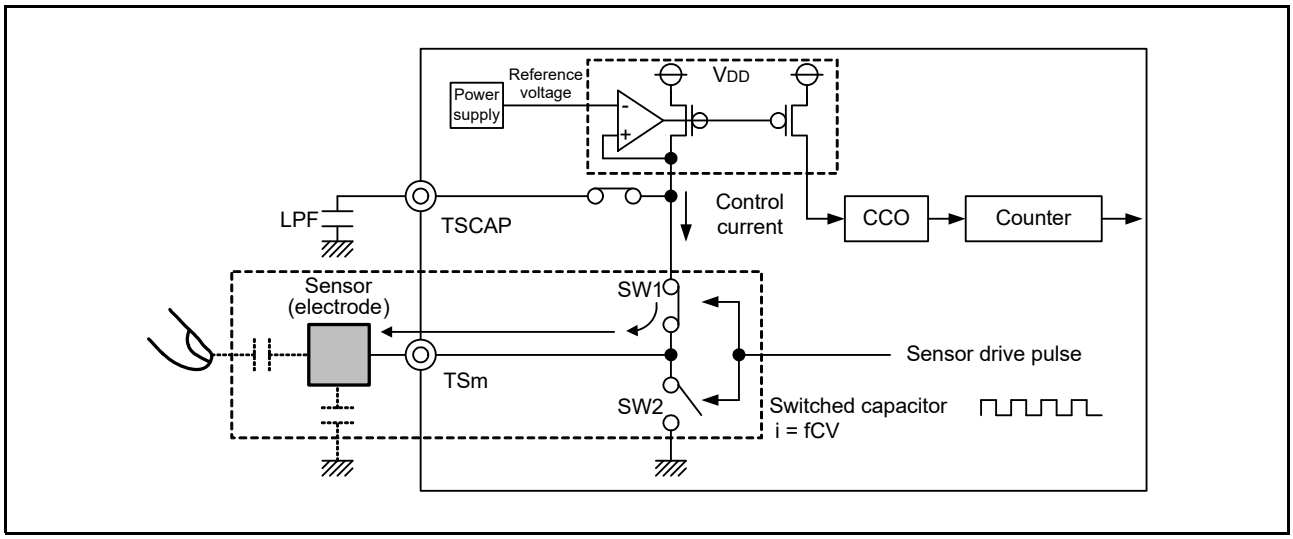
Remark m = 00 to 15, 20 to 35

Figure 30 - 21 to **Figure 30 - 23** explain the principle of the electrostatic capacitance measurement of the current-to-frequency conversion method for the CTSU.

1. The electrostatic capacitance of the electrode is charged by turning SW1 on and SW2 off (**Figure 30 - 21**).
2. The charged capacitance is discharged by turning SW1 off and SW2 on (**Figure 30 - 22**).

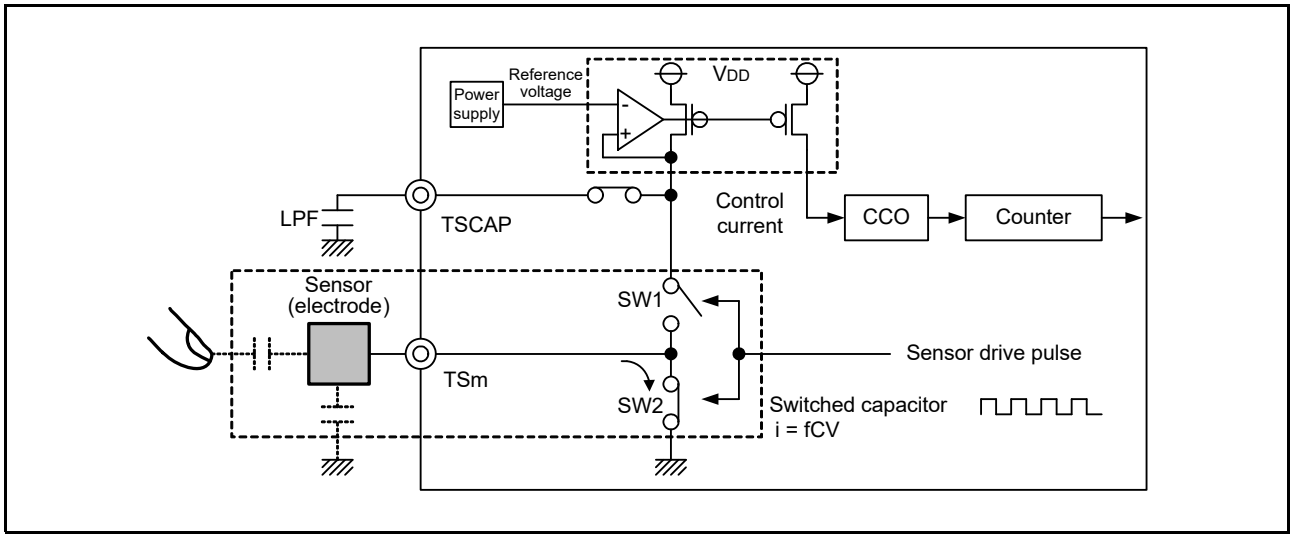
Current flows through the switched capacitor by rapidly switching between charging and discharging in steps 1 and 2. At this time, the value of electrostatic capacitance varies depending on whether the human body (e.g., a finger) is in close proximity, so the flowing current changes. A clock is generated by supplying a control current, which is proportional to the amount of the current flowing through the switched capacitor, from the power supply (LDO) that generates the TSCAP power supply to the current controlled oscillator (CCO). The counter is used to measure the clock frequency which changes depending on whether the human body is in close proximity, and the value read from the counter is used by software to determine whether the human body is in close proximity (**Figure 30 - 23**).

Figure 30 - 21 Charging



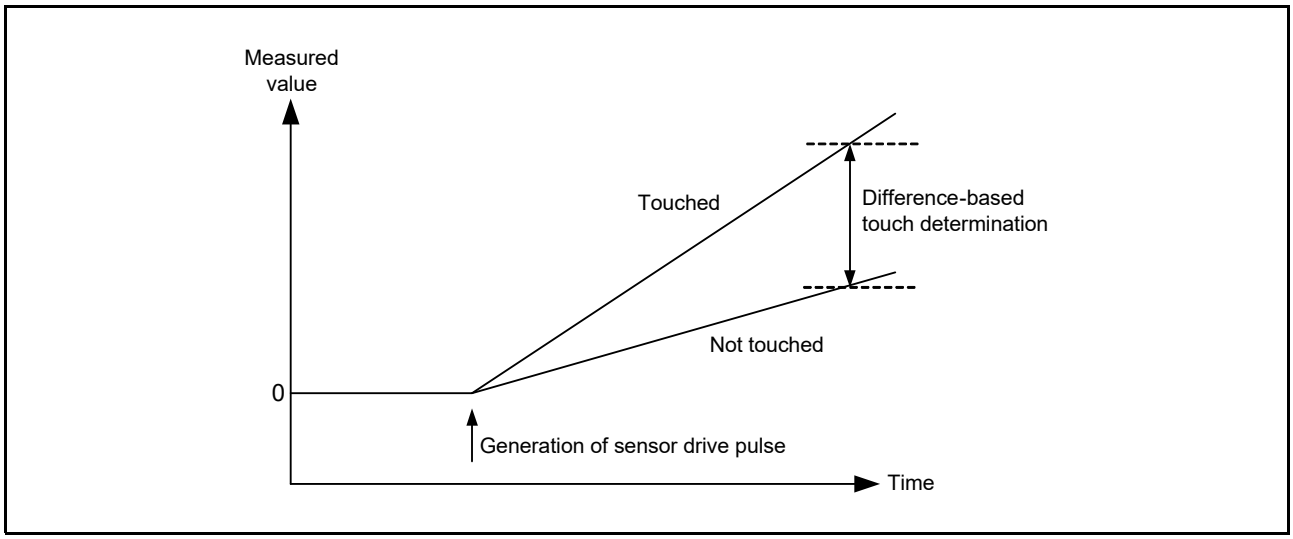
Remark m = 00 to 15, 20 to 35

Figure 30 - 22 Discharging



Remark m = 00 to 15, 20 to 35

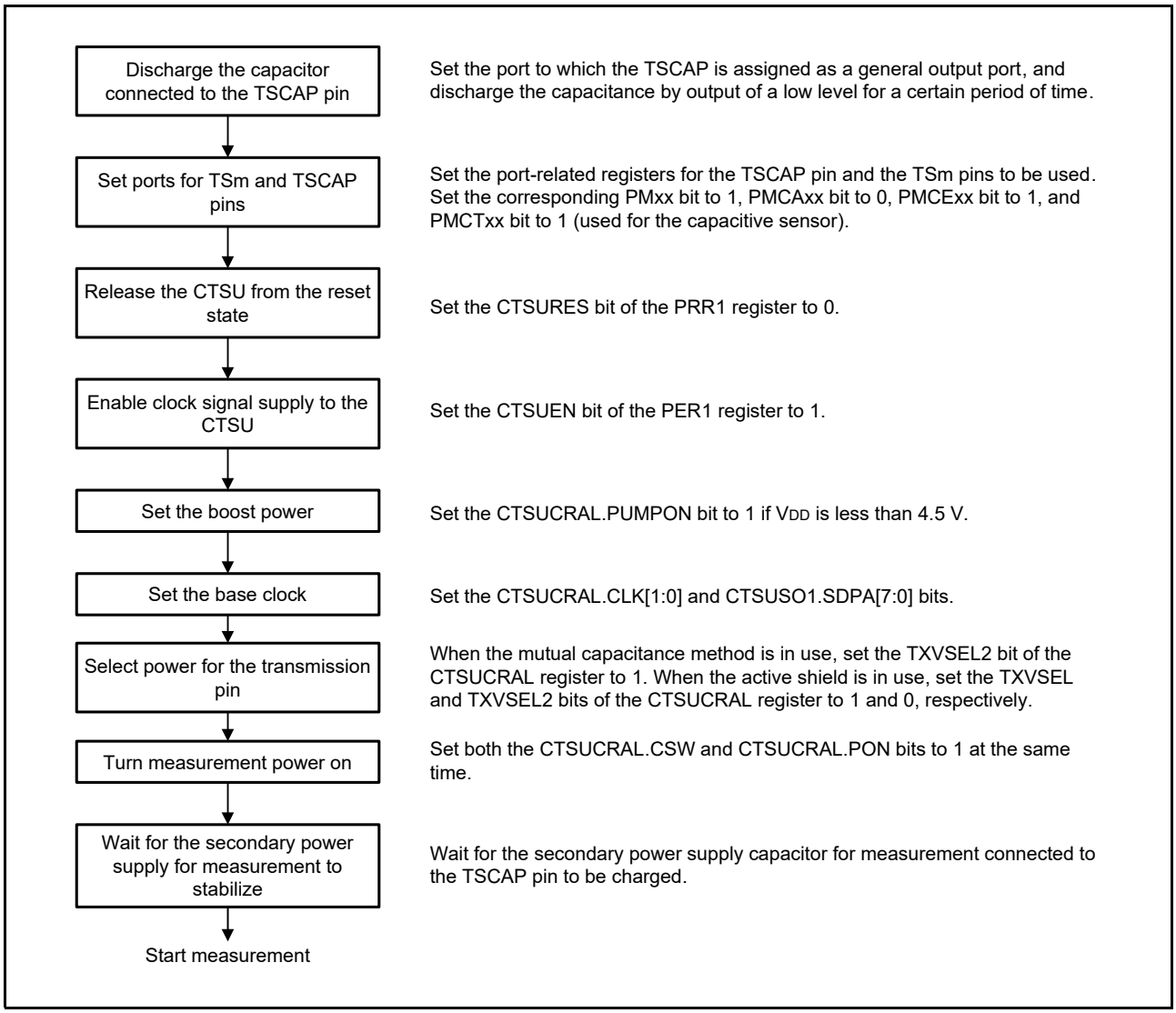
Figure 30 - 23 Change in Measured Value When the Electrode Is Touched or Not Touched



30.3.2 Initial setting procedure

Figure 30 - 24 shows the initial setting procedure for the CTSU.

Figure 30 - 24 CTSU Initial Setting Procedure



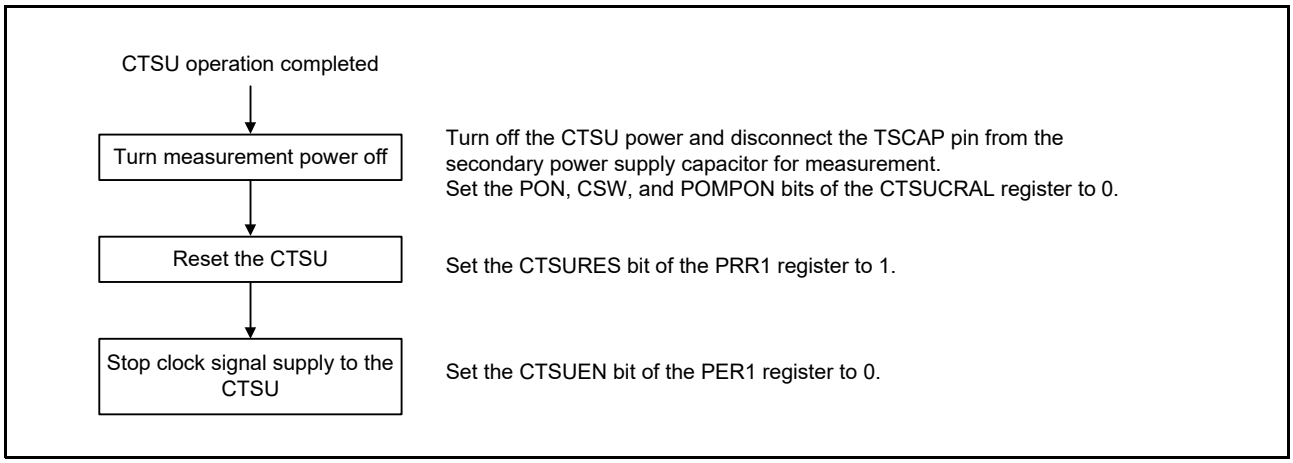
Remark 1. m = 00 to 15, 20 to 35

Remark 2. xx = 0, 2, 3, 5 to 7, 15

Note that PMCA5 to PMCA7, PMCA13, PMCE2, PMCE3, PMCE7, and PMCE15 registers are not present in products of this group.

Figure 30 - 25 shows the flow for stopping CTSU operation and placing it in the standby state.

Figure 30 - 25 CTSU Stopping Flowchart

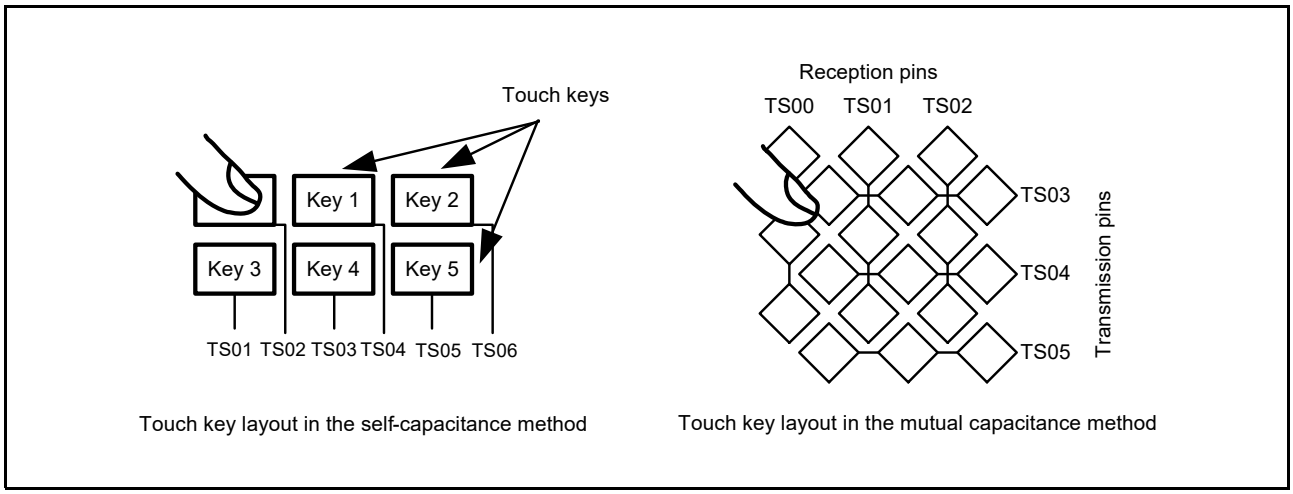


To restart operation after it has been stopped, follow the procedure of the initial settings for the CTSU shown in **Figure 30 - 24**.

30.3.3 Measurement methods

The CTSU supports self-capacitance and mutual capacitance methods. **Figure 30 - 26** illustrates these methods.

Figure 30 - 26 Overview of the Self-capacitance Method and the Mutual Capacitance Method

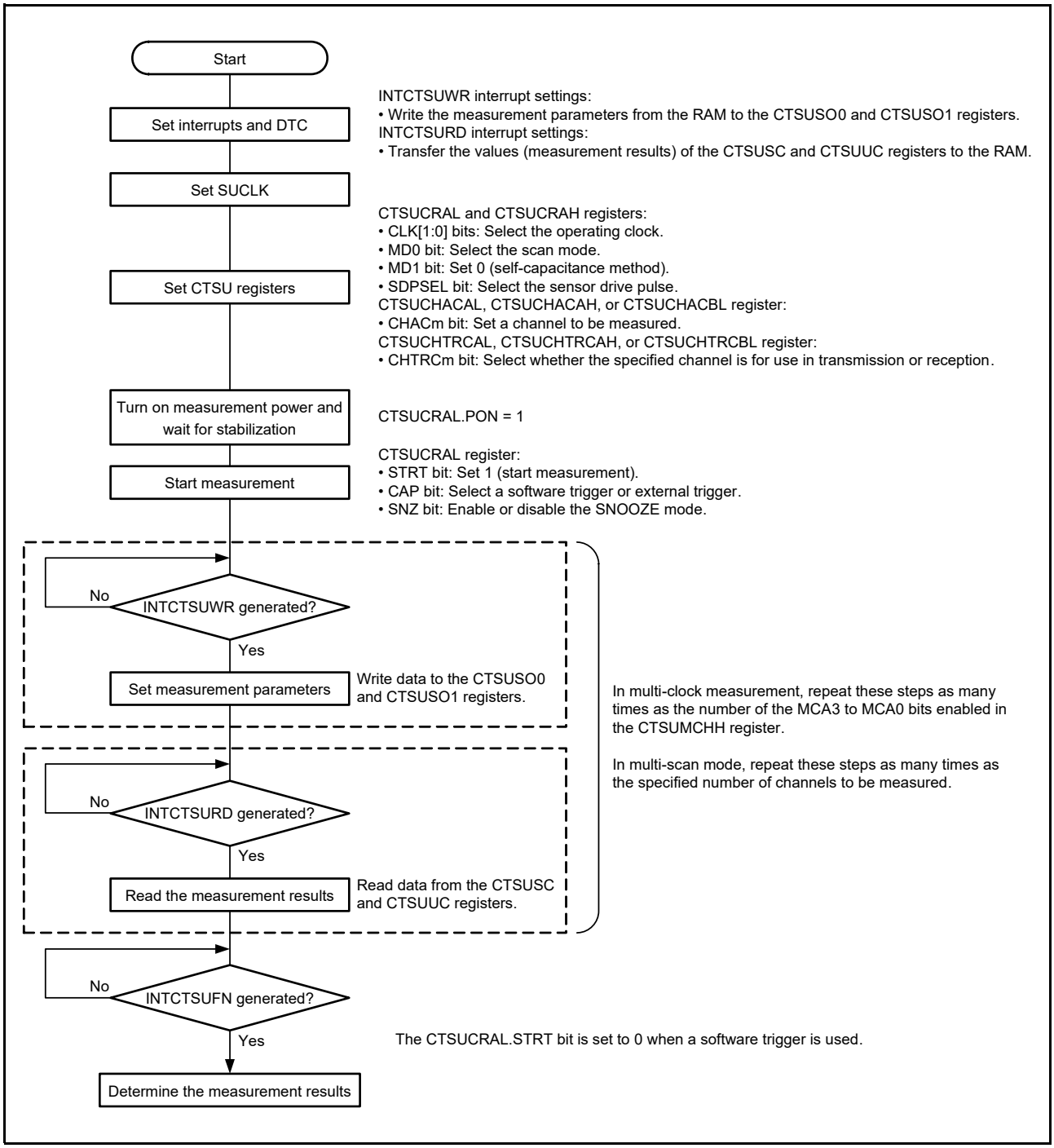


In the self-capacitance method, a single touch pin (TSm) is allocated to a single touch key for the measurement of the individual electrostatic capacitance when the human body (e.g., a finger) is in close proximity. In the mutual capacitance method, the capacitance between two facing electrodes (transmission and reception pins) is measured.

1. Operation in the self-capacitance method

   Figure 30 - 27 shows the control procedure in the self-capacitance method.

Figure 30 - 27 Control Procedure in the Self-capacitance Method

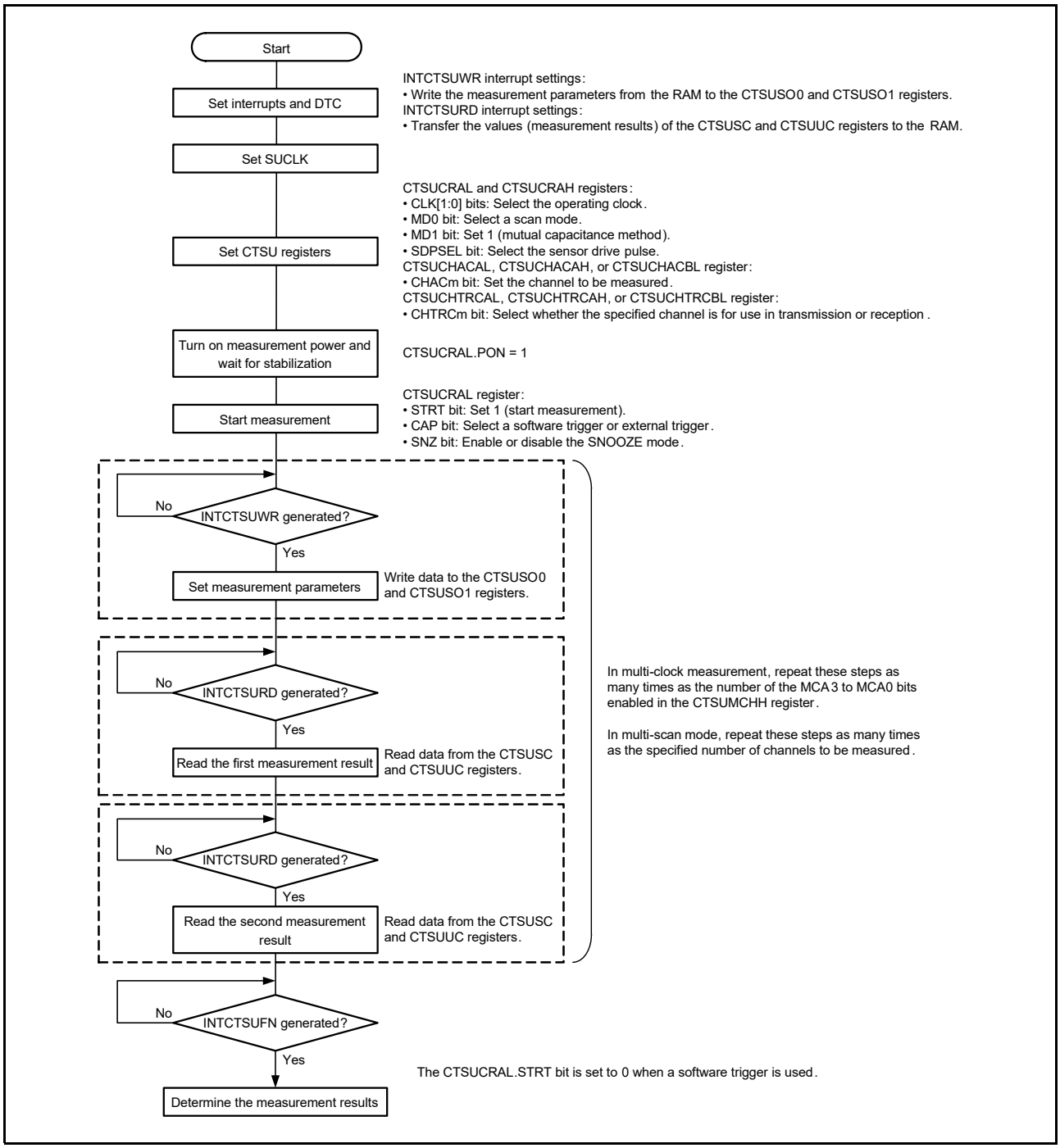


Remark m = 00 to 15, 20 to 35

2. Operation in the mutual capacitance method

In the mutual capacitance method, measurement proceeds by applying edges to the target transmit channel to be measured during the high-level period of the sensor drive pulse on the reception channel. A single measurement target is measured twice, on the rising and falling edges. The difference between the data of these two measurements determines whether or not the electrode was touched. The electrostatic capacitance is measured for all combinations of the set transmission and reception pins. **Figure 30 - 28** shows the control procedure in the mutual capacitance method.

Figure 30 - 28 Control Procedure in the Mutual Capacitance Method



Remark m = 00 to 15, 20 to 35

30.3.4 Scan modes

The CTSU has two scan modes: single scan mode and multi-scan mode. Use the CTSUCRAL.MD0 bits to select a scan mode.

1. Single scan mode
   A single measurement proceeds by using the combination of the reception and transmission channels respectively specified by the MCH0[5:0] and MCH1[5:0] bits of the CTSUMCHL register.
2. Multi-scan mode
   Measurement proceeds once for each combination of the reception and transmission channels, respectively selected by the CTSUCHTRCAL, CTSUCHTRCAH, and CTSUCHTRCBL registers, among the channels to be measured specified by the CTSUCHACAL, CTSUCHACAH, and CTSUCHACBL registers. For both the reception and transmission channels, the channel with the lowest channel number is selected first. When the measurement for all combinations of the reception channels for one transmission channel is completed, the transmit channel will switch.

30.3.5 Multi-clock measurement

Measurements sequentially proceeds switching multiple clock signals with different frequencies. This is effective in the high-resolution pulse mode (CTSUCRAH.SDPSEL = 1). The clock used for measurement (SUCLKx) should be specified using the MCAx bits (x = 3 to 0) of the CTSUMCHH register. Also, set the frequency of each clock in the CTSUSUCLKx register. Measurement proceeds in ascending order, starting from SUCLK0. On completion of the measurement using all the specified clock signals, the measurement in the next channel proceeds.

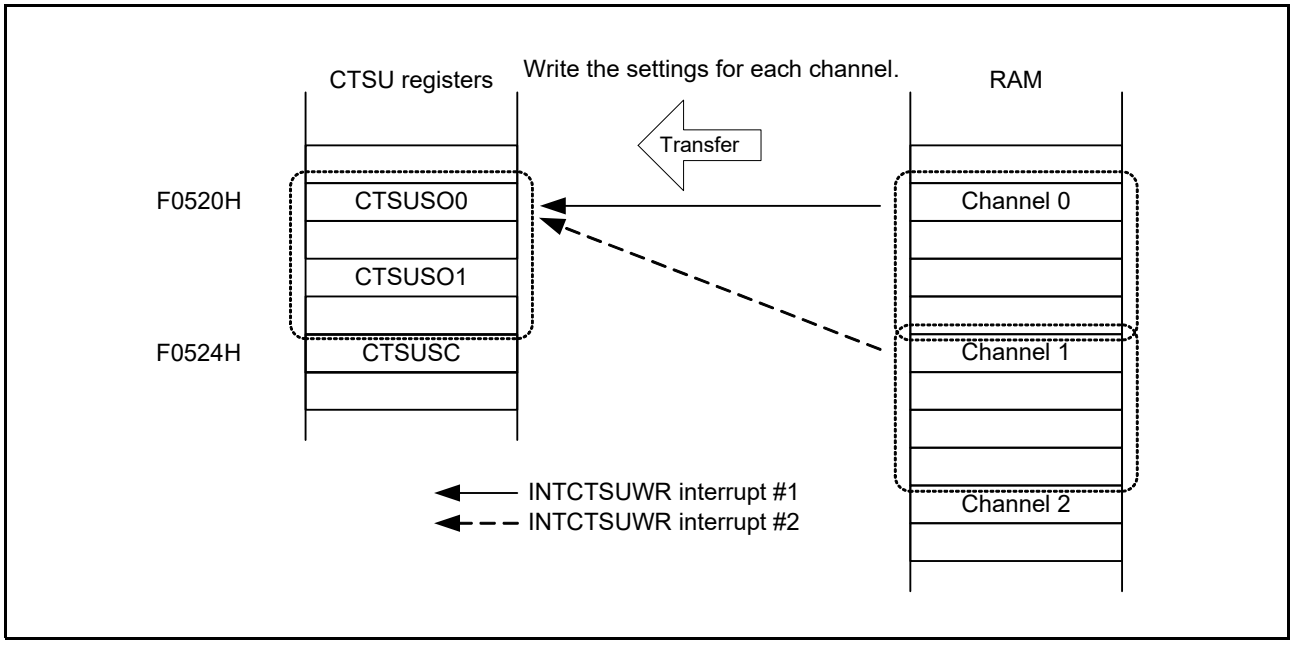
30.3.6 Interrupts

There are three types of interrupts for the CTSU:

- Request to write to a configuration register of an individual CTSU channel (INTCTSUWR)
- Request to transfer data measured by the CTSU (INTCTSURD)
- CTSU measurement end (INTCTSUFN)

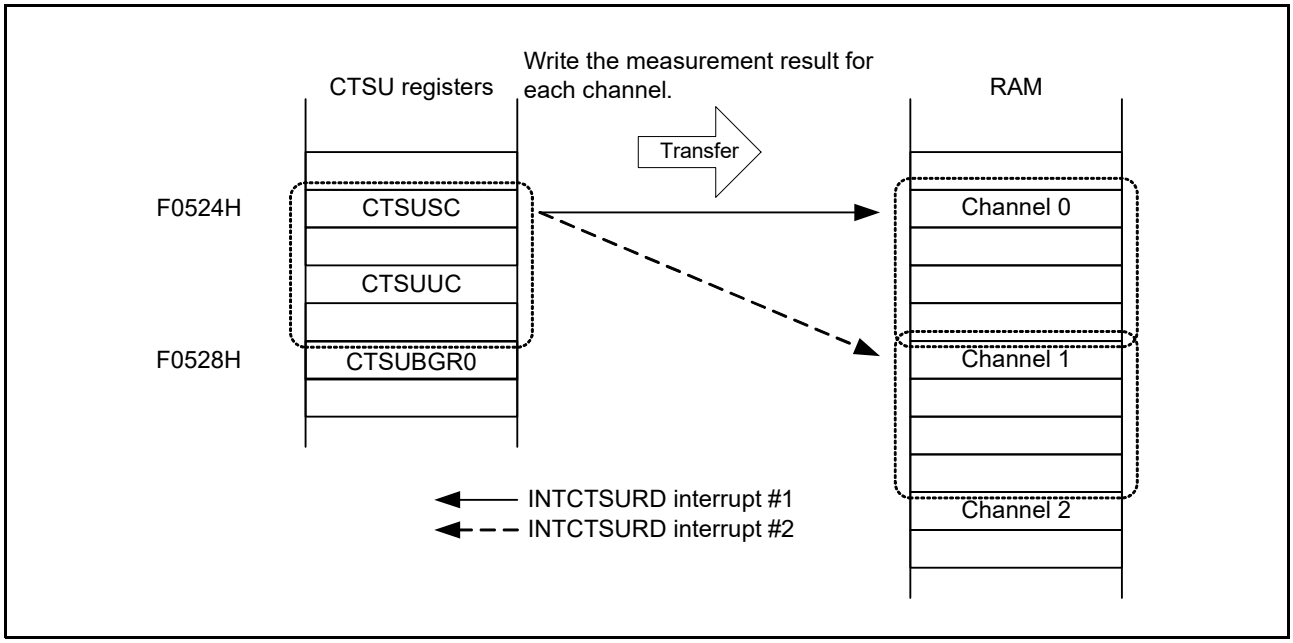
1. INTCTSUWR interrupt
   Store the settings for each measurement channel in the RAM, and set up the transfer associated with the INTCTSUWR interrupt in advance. An INTCTSUWR interrupt is generated when the measurement transitions from state 1 to 2. Write the setting data to the CTSUSO0 and CTSUSO1 registers in response to an INTCTSUWR interrupt. Writing data to the CTSUSO1 register causes measurement to transition to state 3.

Figure 30 - 29 Data Transfer on an INTCTSUWR Interrupt



2. INTCTSURD interrupt
   An INTCTSURD interrupt is generated when the measurement of one channel is finished and the measurement transitions from state 5 to 1. Read the measurement result (sensor counter value) from the CTSUSC register in response to this interrupt. If reading the CTSUUC register is required, set the CTSUDBGR0.CNTRDSEL bit to 1 (transitions to the next state after reading twice). Reading the CTSUSC and CTSUUC registers the number of times specified by the CNTRDSEL bit causes measurement to transition to state 0 or 2.

Figure 30 - 30 Data Transfer Operation on an INTCTSURD Interrupt



3. INTCTSUFN interrupt
   An INTCTSUFN interrupt is generated when the measurement of all specified channels is completed and the measurement transitions from state 1 to 0.

30.4 Usage Notes on the Capacitive Sensing Unit

30.4.1 Evaluation of detection by the capacitive sensing unit

The final stages of product development require operating the system under conditions close to those at the time of product shipment to judge the validity of the results of detection by the touch sensor. “QE for Capacitive Touch” (a development assistance tool for capacitive touch sensors) is provided for use in evaluation. Monitor the states in the measurement of capacitance and evaluate them thoroughly. If the expected results are not obtained, use the “QE for Capacitive Touch” tool to adjust and re-evaluate the CapTouch parameters (mainly the touch thresholds). Note that, if the CTSU is to be used with the mutual capacitance method, the measured values for electrostatic capacitance (counted values) might fluctuate according to the state of the port output of the microcomputer due to fluctuations of the output voltage on the transmission pin. If such a phenomenon is observed, use QE for Capacitive Touch to set the touch threshold in consideration of the fluctuations in the counted values. Such fluctuations will not be seen with the self-capacitance method.

30.4.2 TSCAP pin

A capacitor must be connected to the TSCAP pin to stabilize the internal voltage of the CTSU. Make sure the wiring between the TSCAP pin and the capacitor, and between the capacitor and GND, is as thick and as short as possible. **Table 30 - 3** states the condition for the capacitor to be connected to the TSCAP pin.

Table 30 - 3 Condition for the Capacitor to Be Connected to the TSCAP Pin

Item	Symbol	Condition
Capacitance of the smoothing capacitor to be externally connected to the TSCAP pin	C_{TSCAP}	10 nF ±10%**Note**

Note Use a multilayer ceramic capacitor with a stated capacitance of 10 nF and a capacitance tolerance no greater than ±10%. Select from among those with a capacitance change limited to ±15% in accord with the usage environment, such as X7R as specified by the EIA standard.

Before setting the CTSUCRAL.CSW bit to 1 (the external capacitor connection switch turned on), the capacitor connected to the TSCAP pin must output a low level from the port to which the TSCAP function is assigned and be fully discharged.

30.4.3 Changing the clock and port settings during measurement

Do not change or stop the CPU/peripheral hardware clock (f_{CLK}) while measurement is in progress (CTSUCRAL.STRT = 1). Also, do not change the port settings for the measurement pins (TSm and TSCAP). When the settings are to be changed, forcibly stop the CTSU by respectively setting the INIT and STRT bits in the CTSUCRAL register to 1 and 0 at the same time. For restarting measurement by the CTSU, make the initial settings of the CTSU by using either of the following procedures.

- After writing PRR1.CTSURES = 1 and PER1.CTSUEN = 0, make the initial settings of the CTSU according to **30.3.2 Initial setting procedure**.
- After writing CTSUCRAL.PON, CTSUCRAL.CSW = 00B, and CTSUCRAL.SNZ = 0, make the initial settings of the CTSU according to **30.3.2 Initial setting procedure**.

30.4.4 Software trigger

When 10B (f_{CLK}/4) or 11B (f_{CLK}/8) is set by the CTSUCRAL.CLK[1:0] bits, for restarting measurement immediately after the measurement is completed, write 1 to the CTSUCRAL.STRT bit after the following time has elapsed since the interrupt is generated.

- When CLK[1:0] = 10B (f_{CLK}/4): At least 3 cycles
- When CLK[1:0] = 11B (f_{CLK}/8): At least 7 cycles

30.4.5 External trigger

If an external trigger (an event from the ELCL) is input during a measurement period, the trigger is ignored. The external trigger input is enabled when one cycle of the operating clock has elapsed after an INTCTSUFN interrupt is generated. To release the CTSU from the external trigger mode (forced stop), respectively set the CTSUCRAL.STRT and CTSUCRAL.INIT bits to 0 and 1 at the same time. Follow the procedure below to use external triggers.

1. Make the initial settings of the CTSU.
2. Select the capacitive sensing unit as the destination of the event link controller (ELCL) while the CTSUCRAL.STRT bit is 0 (measurement stopped).
3. After setting the CTSUCRAL.CAP bit to 1 (external trigger), set the CTSUCRAL.STRT bit to 1 (starting measurement by the CTSU).
4. Start the operation of the peripheral function that generates external trigger sources.

30.4.6 Reading the CTSU sensor counter registers (CTSUSC and CTSUUC)

Do not attempt to read the CTSU sensor counter registers (CTSUSC and CTSUUC) during measurement (states 3 to 5). The value cannot be guaranteed if reading of these registers is attempted.

30.4.7 Forced stop

To forcibly stop measurement, be sure to set the CTSUCRAL.STRT bit to 0 and the CTSUCRAL.INIT bit to 1 at the same time. Setting the INIT bit to 1 initializes the internal measurement state and also initializes the following registers.

- CTSUSC and CTSUUC registers
- CTSUMCHL and CTSUMCHH registers
- All bits in CTSUSRL except bits ICOMP0 and ICOMP1

If operation is forcibly stopped, an interrupt request might be generated depending on the internal state. Disable the interrupts and stop the DTC before forcibly stopping the operation. If stoppage of the DTC transfer is required when the DTC is in use for controlling the CTSU, also forcibly stop the CTSU and proceed with the initialization.

30.4.8 Sampling cycle when the spread spectrum is in use

When the spread spectrum is enabled (CTSUCRBL.SOFF = 0), set the CTSUCRAL.CLK[1:0] and CTSUSO1.SDPA[7:0] bits so that the sampling cycle of the sensor drive pulse is less than 1/4 of the sensor drive pulse cycle.

30.4.9 Settings of the transmission pin when the self-capacitance method is to be used

The transmission pin cannot be used for measurement when the self-capacitance method is in use. A pulse that is in phase with the measurement pulse is output from the transmission pin. Use the transmission pin as an active shield on the board. However, the active shield cannot be used when CTSUCRAL.ATUNE0 = 1 (measurement power supply voltage = 1.2 V). Also, when the self-capacitance method is to be used, do not assign multiple TSm pins to transmission at the same time.

30.4.10 Points to note on noise countermeasures

Fluctuations in the V_{DD} power supply due to noise or other factors may lead to variation in the results of measurement. In addition, the application of ripple noise to the V_{DD} power supply may change (decrease) the control current of the capacitive sensing unit and in turn decrease the measured values of the electrostatic capacitance, depending on the frequency band of the ripple noise. For details, refer to the application note, **Capacitive Touch Ripple Noise Prevention Guide (R30AN0453)**. Take note of these points when designing the power supply circuit. **Table 30 - 4** shows the characteristics of the amount of decrease in measured capacitance due to ripple noise to the power source (reference values). **Figure 30 - 31** shows the amount of decrease in measured capacitance (reference value).

Table 30 - 4 Characteristics of the Amount of Decrease in Measured Capacitance Due to Ripple Noise to the Power Source (Reference Values)

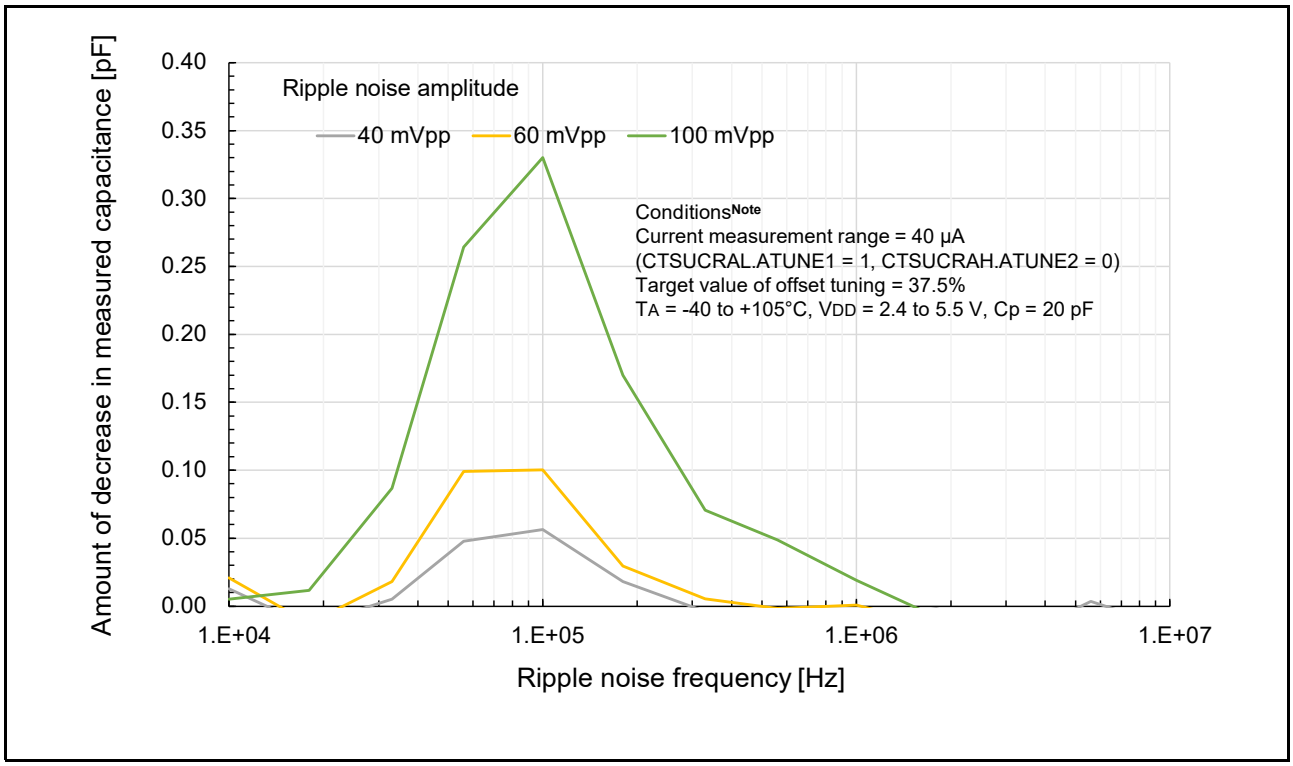
(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Cp = 20 pF)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Characteristics of the amount of decrease in measured capacitance **Note**	Cdown	Ripple noise frequency < 20 kHz	Ripple noise amplitude 100 mVpp	—	—	0.02	pF
		20 kHz ≤ Ripple noise frequency ≤ 2 MHz	Ripple noise amplitude 40 mVpp	—	—	0.06	
			Ripple noise amplitude 60 mVpp	—	—	0.10	
			Ripple noise amplitude 100 mVpp	—	—	0.33	
		2 MHz < Ripple noise frequency	Ripple noise amplitude 100 mVpp	—	—	0.01	

Note These are the values under the following conditions.
- The self-capacitance method is used (CTSUCRAL.MD1 = 0).
- The current measurement range is 40 µA (CTSUCRAL.ATUNE1 = 1, CTSUCRAH.ATUNE2 = 0).
- The target value of offset tuning is 37.5%.

Remark Cp: Parasitic capacitance

Figure 30 - 31 Amount of Decrease in Measured Capacitance (Reference Value)



Note For details of the conditions, refer to the application note, **Capacitive Sensor MCU QE for Capacitive Touch Advanced Mode Parameter Guide (R30AN0428)**.

30.4.11 Memory occupancy

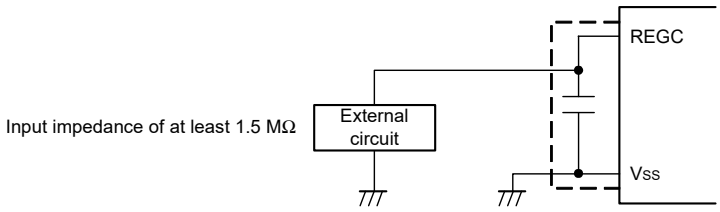
The amount of memory required to run the capacitive sensing unit depends on the configuration of the buttons. For details, refer to the following application notes.

- **RL78 Family CTSU Module Software Integration System (R11AN0484)**
- **RL78 Family TOUCH Module Software Integration System (R11AN0485)**

Section 31 Regulator

31.1 Overview

Each product of this group incorporates a circuit for constant voltage operation. To stabilize the output voltage from the regulator, connect the REGC pin to Vss via a capacitor (0.47 to 1 μF). Use a capacitor with good characteristics, since it is for stabilizing the internal voltage. The REGC pin can be used as a reference voltage for an external circuit. An external circuit for connection to the REGC pin for this purpose must have an input impedance of at least 1.5 MΩ. The voltage on the REGC pin is in the range from 1.38 to 1.60 V, and the typical value is 1.5 V.



Caution The length of the wiring within the broken lines in the above figure should be as short as possible.

The regulator outputs a 1.5-V voltage.

Section 32 Option Bytes

32.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory in each product of this group form an option byte area. Option bytes consist of user option bytes (000C0H to 000C2H) and on-chip debug option byte (000C3H). Upon power application or resetting and starting, an option byte is automatically referenced and a specified function is set. When using the product, be sure to set the following functions by using the option bytes. For the bits to which no function is allocated, do not change their initial values. To use the boot swap operation during self-programming, 000C0H to 000C3H are replaced by 040C0H to 040C3H. Therefore, set the same values as 000C0H to 000C3H to 040C0H to 040C3H.

Caution The option bytes should always be set regardless of whether each function is used.

32.1.1 User option bytes (000C0H to 000C2H or 040C0H to 040C2H)

1. 000C0H or 040C0H
   - Setting of watchdog timer operation
     - Enabling or disabling of counter operation
     - Enabling or stopping of counter operation in the HALT or STOP mode
   - Setting of overflow time of watchdog timer
   - Setting of window open period of watchdog timer
   - Setting of interval interrupt of watchdog timer
     - Interval interrupt is used or not used

Caution When boot swapping is to be used, 000C0H is replaced with 040C0H. Therefore, set the same value as the setting in 000C0H in 040C0H.

2. 000C1H or 040C1H
   - Setting of the operating mode of LVD0
     - Reset mode
     - Interrupt mode
     - LVD0 off (by controlling the externally input reset signal on the $\overline{\text{RESET}}$ pin)
   - Setting of LVD0 detection level (V_{LVD0})

Caution 1. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 37.4 AC Characteristics. This is done by utilizing the voltage detection circuit or controlling the externally input reset signal. After the power supply is turned off, this MCU should be placed in the STOP mode, or placed in the reset state by utilizing the voltage detection circuit or controlling the externally input reset signal, before the voltage falls below the operating range. The range of operating voltage varies with the setting of the user option byte (000C2H or 040C2H).

Caution 2. When boot swapping is to be used, 000C1H is replaced with 040C1H. Therefore, set the same value as the setting in 000C1H in 040C1H.

3. 000C2H or 040C2H
   - Setting of the operating mode of the flash memory
     Make the setting depending on the main system clock frequency (f_{MAIN}) and power supply voltage (V_{DD}) to be used.
     – LS (low-speed main) mode
     – HS (high-speed main) mode
     – LP (low-power main) mode
   - Setting of the frequency of the high-speed on-chip oscillator
     – Select from 1 to 32 MHz.

Caution When boot swapping is to be used, 000C2H is replaced with 040C2H. Therefore, set the same value as the setting in 000C2H in 040C2H.

32.1.2 On-chip debug option byte (000C3H or 040C3H)

- Control of on-chip debug operation
  – On-chip debug 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 When boot swapping is to be used, 000C3H is replaced with 040C3H. Therefore, set the same value as the setting in 000C3H in 040C3H.

32.2 Format of User Option Bytes

Figure 32 - 1 Format of User Option Byte (000C0H or 040C0H)

Address: 000C0H or 040C0H[Note 1]

Symbol	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 + 1/4 f_{IL} is reached.

WINDOW1	WINDOW0	Watchdog timer window open period[Note 2]
0	0	Setting prohibited
0	1	50%
1	0	Setting prohibited
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_{IL} = 37.683 kHz (max.))
0	0	0	$2^7/f_{IL}$ (3.39 ms)
0	0	1	$2^8/f_{IL}$ (6.79 ms)
0	1	0	$2^9/f_{IL}$ (13.58 ms)
0	1	1	$2^{10}/f_{IL}$ (27.17 ms)
1	0	0	$2^{12}/f_{IL}$ (108.69 ms)
1	0	1	$2^{14}/f_{IL}$ (434.78 ms)
1	1	0	$2^{15}/f_{IL}$ (869.56 ms)
1	1	1	$2^{17}/f_{IL}$ (3478.26 ms)

WDSTBYON	Operation control of watchdog timer counter (HALT/STOP mode)
0	Counter operation stopped in HALT/STOP mode[Note 2]
1	Counter operation enabled in HALT/STOP mode

Note 1. When boot swapping is to be used, 000C0H is replaced with 040C0H. Therefore, set the same value as the setting in 000C0H in 040C0H.

Note 2. The window open period is 100% when WDSTBYON = 0, regardless of the value of the WINDOW1 and WINDOW0 bits.

Remark When the setting of the WDTON bit is 1, the illicit memory access detection remains enabled regardless of the setting of the IAWEN bit in the IAWCTL register.

Figure 32 - 2 Format of User Option Byte (000C1H or 040C1H)

Address: 000C1H or 040C1H[Note]

Symbol	7	6	5	4	3	2	1	0
	LVD0EN	LVD0SEL	1	1	1	LVD0V2	LVD0V1	LVD0V0

LVD0 setting (reset mode)

Detection Voltage		Option Byte Setting Value				
V_{LVD0}		LVD0EN	Mode setting	LVD0V2	LVD0V1	LVD0V0
Rising edge	Falling edge		LVD0SEL			
1.69 V	1.65 V	1	1	1	1	1
1.90 V	1.86 V			1	1	0
2.38 V	2.33 V			1	0	1
2.67 V	2.62 V			1	0	0
2.97 V	2.91 V			0	1	1
3.96 V	3.88 V			0	1	0
—		Settings other than the above are prohibited.				

LVD0 setting (interrupt mode)

Detection Voltage		Option Byte Setting Value				
V_{LVD0}		LVD0EN	Mode setting	LVD0V2	LVD0V1	LVD0V0
Rising edge	Falling edge		LVD0SEL			
1.69 V	1.65 V	1	0	1	1	1
1.90 V	1.86 V			1	1	0
2.38 V	2.33 V			1	0	1
2.67 V	2.62 V			1	0	0
2.97 V	2.91 V			0	1	1
3.96 V	3.88 V			0	1	0
—		Settings other than the above are prohibited.				

LVD0 off setting (external reset input from the $\overline{\text{RESET}}$ pin is used)

Detection Voltage		Option Byte Setting Value				
V_{LVD0}		LVD0EN	Mode setting	LVD0V2	LVD0V1	LVD0V0
Rising edge	Falling edge		LVD0SEL			
—	—	0	×	0	1	0
—		Settings other than the above are prohibited.				

(Note, Cautions, and Remarks are listed on the next page.)

Note When boot swapping is to be used, 000C1H is replaced with 040C1H. Therefore, set the same value as the setting in 000C1H in 040C1H.

Caution 1. Be sure to set bits 5 to 3 to 1.

Caution 2. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 37.4 AC Characteristics. This is done by utilizing the voltage detection circuit or controlling the externally input reset signal. After the power supply is turned off, this MCU should be placed in the STOP mode, or placed in the reset state by utilizing the voltage detection circuit or controlling the externally input reset signal, before the voltage falls below the operating range. The range of operating voltage varies with the setting of the user option byte (000C2H or 040C2H).

Remark 1. ×: Don't care

Remark 2. For details on LVD0, see **Section 26 Voltage Detector (LVD)**.

Remark 3. The detection voltage is a typical value. For details, see **37.6.6 Characteristics of the LVD circuit**.

Figure 32 - 3 Format of User Option Byte (000C2H or 040C2H)

Address: 000C2H or 040C2H[Note]

Symbol	7	6	5	4	3	2	1	0
	CMODE1	CMODE0	1	0	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0

CMODE1	CMODE0	Operating Mode of the Flash Memory	Operating Frequency Range	Operating Voltage Range
0	1	LP (low-power main) mode	1 to 2 MHz (Rewriting of flash memory is not possible.)	1.6 to 5.5 V
1	0	LS (low-speed main) mode	1 to 2 MHz	1.6 to 5.5 V
			1 to 4 MHz (Rewriting of flash memory is not possible.)	
			1 to 24 MHz	1.8 to 5.5 V
1	1	HS (high-speed main) mode	1 to 2 MHz	1.6 to 5.5 V
			1 to 4 MHz (Rewriting of flash memory is not possible.)	
			1 to 32 MHz	1.8 to 5.5 V
Other than above		Setting prohibited		

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the High-speed On-chip Oscillator Clock
1	0	0	0	32 MHz
0	0	0	0	24 MHz
1	0	0	1	16 MHz
0	0	0	1	12 MHz
1	0	1	0	8 MHz
0	0	1	0	6 MHz
1	0	1	1	4 MHz
0	0	1	1	3 MHz
1	1	0	0	2 MHz
1	1	0	1	1 MHz
Other than above				Setting prohibited

Note When boot swapping is to be used, 000C2H is replaced with 040C2H. Therefore, set the same value as the setting in 000C2H in 040C2H.

Caution 1. Be sure to set bits 5 and 4 to 10B.

Caution 2. The operating frequency range and operating voltage range vary depending on the operating mode of the flash memory. For details, see 37.4 AC Characteristics.

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 or 040C3H)

Address: 000C3H or 040C3H[Note]

Symbol	7	6	5	4	3	2	1	0
	OCDENSET	0	0	0	0	1	0	OCDERSD

OCDENSET	OCDERSD	Control of on-chip debug operation
0	0	Disables on-chip debugging.
0	1	Setting prohibited
1	0	Enables on-chip debugging. Erases data of flash memory in case of failures in authenticating on-chip debug security ID.
1	1	Enables on-chip debugging. Does not erase data of flash memory in case of failures in authenticating on-chip debug security ID.

Note When boot swapping is to be used, 000C3H is replaced with 040C3H. Therefore, set the same value as the setting in 000C3H in 040C3H.

Caution **Only bits 7 and 0 (OCDENSET and OCDERSD) are specifiable.**
Be sure to set bits 6 to 1 to 000010B.

Remark The use of on-chip debugging changes the values of bits 3 to 1. Accordingly, the values after such settings are made become undefined. However, note that when setting the option byte, be sure to set bits 3 to 1 to their default values (010B).

32.4 Setting of Option Bytes

The user option bytes and on-chip debug option byte can also be set by using linker options instead of statements in the source code. In such cases, the settings made by using linker options are given priority over the statements in the source code as shown below. An example of the statements in relation to the option byte settings in software is shown below.

```
        .CSEG    OPT_BYTE
        .DB      0x36     ; Does not use the interval interrupt of the watchdog timer.
                          ; Enables watchdog timer operation.
                          ; The window open period of the watchdog timer is 50%.
                          ; The overflow time of the watchdog timer is 2^10/fIL.
                          ; Stops watchdog timer operation during HALT/STOP mode.
        .DB      0xBF     ; Selects VLVD0 as 1.69 V on rising edges and 1.65 V on falling edges.
                          ; Select the interrupt mode as the operating mode of LVD0.
        .DB      0x6D     ; Select the LP (low-power main) mode as the operating mode of the flash memory
                          ; and 1 MHz as the frequency of the high-speed on-chip oscillator clock.
        .DB      0x85     ; Enables on-chip debug operation, does not erase flash memory data when
                          ; security ID authentication fails.
```

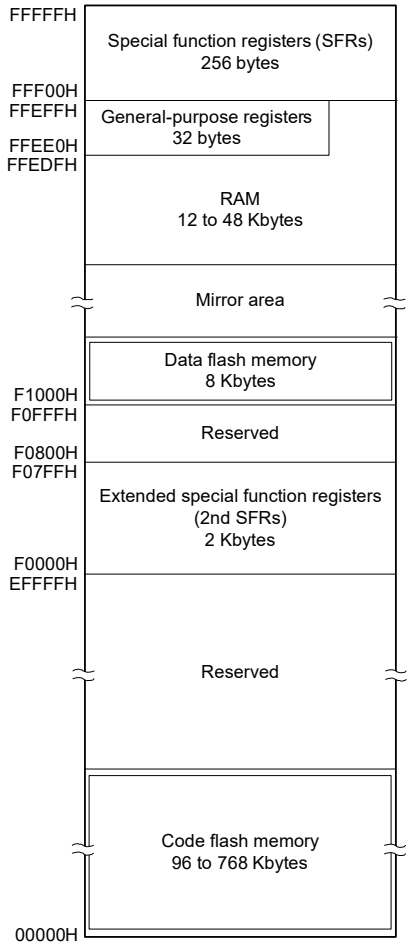
When boot swapping is to be used during self-programming, 000C0H to 000C3H are replaced with 040C0H to 040C3H. Therefore, set the same values as the settings in 000C0H to 000C3H in 040C0H to 040C3H as shown below.

```
OPT2    .CSEG    AT       0x040C0
        .DB              0x36     ; Does not use the interval interrupt of the watchdog timer.
                                  ; Enables watchdog timer operation.
                                  ; The window open period of the watchdog timer is 50%.
                                  ; The overflow time of the watchdog timer is 2^10/fIL.
                                  ; Stops watchdog timer operation during HALT/STOP mode.
        .DB              0xBF     ; Selects VLVD0 as 1.69 V on rising edges and 1.65 V on falling edges.
                                  ; Select the interrupt mode as the operating mode of LVD0.
        .DB              0x6D     ; Select the LP (low-power main) mode as the operating mode of the flash memory
                                  ; and 1 MHz as the frequency of the high-speed on-chip oscillator clock.
        .DB              0x85     ; Enables on-chip debug operation, does not erase flash memory data when
                                  ; security ID authentication fails.
```

Caution To specify the option byte by using assembly language, use OPT_BYTE as the relocation attribute name of the CSEG pseudo instruction. To specify 040C0H to 040C3H for use as the option bytes in the case of boot swapping, use the relocation attribute AT to specify absolute addresses.

Section 33 Flash Memory

This MCU incorporates flash memory that allows the writing, erasure, and overwriting of programs. The flash memory consists of a code flash memory area, from which programs can be executed, a data flash memory area, which is suitable for storing data, and an extra area, which is for storing flash memory operation and security settings.



The following methods for programming the flash memory are available. The contents of code flash memory can be rewritten through serial programming by using a flash memory programmer or other external device (via UART communications), or through self-programming.

- Serial Programming Using Flash Memory Programmer (see **33.1**.)
  Data can be written to the flash memory on-board or off-board by using a dedicated flash memory programmer.
- Serial Programming Using External Device (that Incorporates UART) (see **33.2**.)
  Data can be written to the flash memory on-board through UART communications with an external device (a microcontroller or ASIC).
- Self-programming (see **33.6**.)
  The user application can execute self-programming of the code flash memory.

Caution When rewriting the flash memory, stop the middle-speed on-chip oscillator (MIOEN = 0) and select the high-speed on-chip oscillator (MCM1 = 0) as the main on-chip oscillator clock (fOCO). Do not change the flash operating mode select register (FLMODE).

The data flash memory can be rewritten to by using self-programming during user program execution (background operation). For access and writing to the data flash memory, see **33.6 Self-programming** and **33.10 Data Flash Memory**. The code flash memory and the data flash memory have a function to protect them from rewriting. For details, see **27.3.3 Flash memory guard function**.

33.1 Serial Programming Using Flash Memory Programmer

The following dedicated flash memory programmers can be used to write data to the internal flash memory of this MCU.

- 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 this MCU has been mounted on the target system. Mount a connector for the dedicated flash memory programmer on the target system.
2. Off-board programming
   Data can be written to the flash memory with a dedicated program adapter before this MCU is mounted on the target system.

Table 33 - 1 Wiring between This MCU and Dedicated Flash Memory Programmer (1/2)

Pin Configuration of Dedicated Flash Memory Programmer				Pin Name	Pin No. for R7F100G						
					30-pin	32-pin	36-pin	40-pin	44-pin	48-pin	52-pin
Signal Name		I/O	Pin Function		LSSOP (300 mil)	HWQFN (5 × 5) LQFP (7 × 7)	WFLGA (4 × 4)	HWQFN (6 × 6)	LQFP (10 × 10)	LFQFP (7 × 7) HWQFN (7 × 7)	LQFP (10 × 10)
PG-FP6	E2 or E2 Lite On-chip Debugging Emulator										
—	TOOL0	I/O	Transmit/ receive signal	TOOL0/ P40	5	1	F6	1	2	39	4
SI/RxD	—	I/O									
—	RESET_OUT	Output	Reset signal	$\overline{\text{RESET}}$	6	2	E5	2	3	40	5
$\overline{\text{RESET}}$	—	Output									
VCC	VDD	I/O	VDD voltage generation/ power monitoring	VDD	12	8	B6	10	11	48	13
GND		—	Ground	VSS	11	7	C5	9	10	47	12
				REGC **Note**	10	6	D5	8	9	46	11
FLMD1	EMVDD	—	Driving power for TOOL0 pin	VDD	12	8	B6	10	11	48	13

Pin Configuration of Dedicated Flash Memory Programmer				Pin Name	Pin No. for R7F100G					
					64-pin	64-pin	80-pin	100-pin	100-pin	128-pin
Signal Name		I/O	Pin Function		LQFP (12 × 12) LFQFP (10 × 10)	WFLGA (5 × 5)	LQFP (14 × 14) LFQFP (12 × 12)	LFQFP (14 × 14)	LQFP (14 × 20)	LFQFP (14 × 20)
PG-FP6	E2 or E2 Lite On-chip Debugging Emulator									
—	TOOL0	I/O	Transmit/ receive signal	TOOL0/ P40	5	D6	9	12	89	22
SI/RxD	—	I/O								
—	RESET_OUT	Output	Reset signal	$\overline{\text{RESET}}$	6	E7	10	13	90	26
$\overline{\text{RESET}}$	—	Output								
VCC	VDD	I/O	VDD voltage generation/ power monitoring	VDD	15	B7	19	22	99	35
GND		—	Ground	VSS	13	C7	17	20	97	33
				EVSS0, EVSS1	14	B8	18	21, 43	98, 20	34, 56
				REGC **Note**	12	D7	16	19	96	32
FLMD1	EMVDD	—	Driving power for the TOOL0 pin	EVDD0, EVDD1	16	A8	20	23, 53	100, 30	36, 57

Table 33 - 1 Wiring between This MCU and Dedicated Flash Memory Programmer (2/2)

Pin Configuration of Dedicated Flash Memory Programmer				Pin Name	Pin No. for R7F104G		
					32-pin	44-pin	48-pin
Signal Name		I/O	Pin Function		LQFP (7 × 7)	LQFP (10 × 10)	LFQFP (7 × 7)
PG-FP6	E2 or E2 Lite On-chip Debugging Emulator						
—	TOOL0	I/O	Transmit/ receive signal	TOOL0/ P40	1	2	39
SI/RxD	—	I/O					
—	RESET_OUT	Output	Reset signal	$\overline{\text{RESET}}$	2	3	40
$\overline{\text{RESET}}$	—	Output					
V_{CC}	V_{DD}	I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	8	11	48
GND		—	Ground	V_{SS}	7	10	47
				REGC **Note**	6	9	46
FLMD1	EMV_{DD}	—	Driving power for the TOOL0 pin	V_{DD}	8	11	48

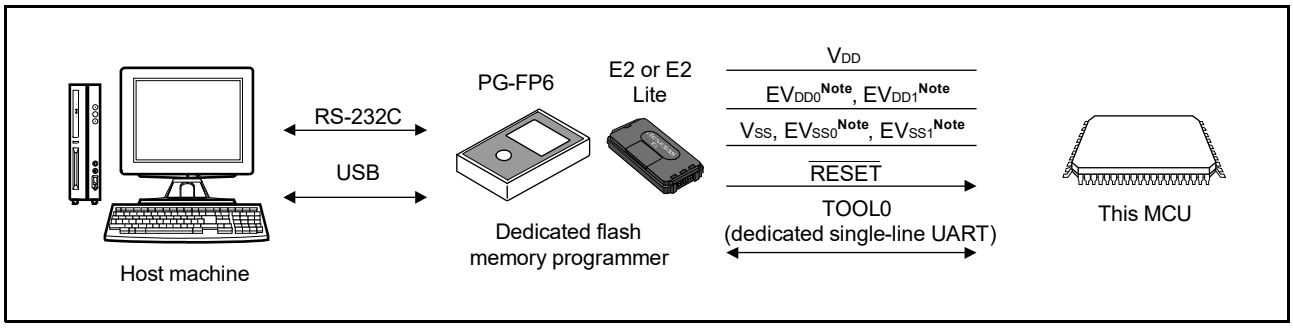
Note Connect the REGC pin to the ground via a capacitor (0.47 to 1 μF).

Remark Pins that are not indicated in the above table can be left open-circuit when using the flash memory programmer for flash memory programming.

33.1.1 Programming environment

Figure 33 - 1 shows the environment required for writing a program to the flash memory of this MCU.

Figure 33 - 1 Environment for Writing Program to Flash Memory



Note This pin is only present in the 64- to 128-pin R7F100G products.

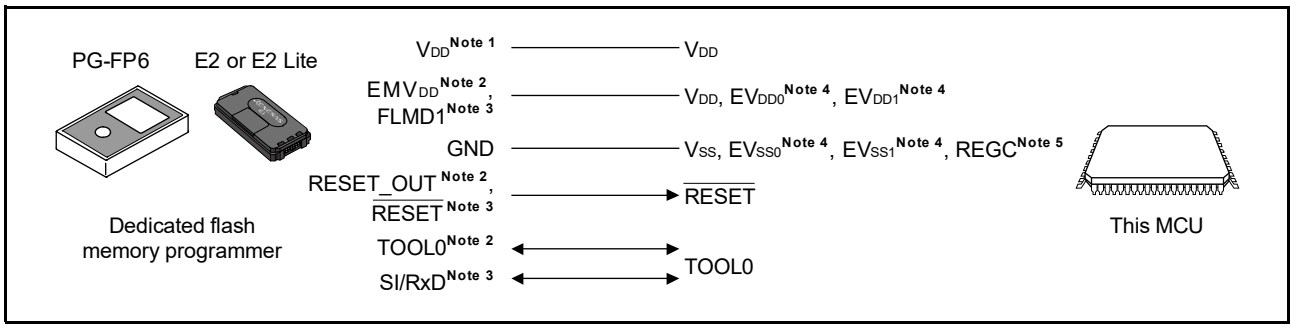
A host machine that controls the dedicated flash memory programmer is required. For the interface between the dedicated flash memory programmer and this MCU, the TOOL0 pin of this MCU is used for writing and erasure via the dedicated single-line UART.

33.1.2 Communications mode

The TOOL0 pin of this MCU is used for communications between the dedicated flash memory programmer and this MCU through serial communications by using the dedicated single-line UART.

Transfer rate: 1 Mbps, 500 kbps, 250 kbps, 115.2 kbps

Figure 33 - 2 Communications with Dedicated Flash Memory Programmer



Note 1. The signal name for the PG-FP6 is VCC.
Note 2. This connection applies when an E2 or E2 Lite on-chip debugging emulator is in use.
Note 3. This connection applies when a PG-FP6 is in use.
Note 4. This pin is only present in the 64- to 128-pin R7F100G products.
Note 5. Connect the REGC pin to the ground via a capacitor (0.47 to 1 µF).

The dedicated flash memory programmer generates the following signals for this MCU. See the manual for the PG-FP6 or, E2 or E2 Lite on-chip debugging emulator for details.

Table 33 - 2 Pin Connection

Dedicated Flash Memory Programmer				This MCU
Signal Name		I/O	Pin Function	Pin Name[Note 1]
PG-FP6	E2 or E2 Lite On-chip Debugging Emulator			
V_{CC}	V_{DD}	I/O	V_{DD} voltage generation/power monitoring	V_{DD}
GND		—	Ground	V_{SS}, EV_{SS0}, EV_{SS1}, REGC[Note 2]
FLMD1	EMV_{DD}	—	Driving power for the TOOL0 pin	V_{DD}, EV_{DD0}, EV_{DD1}
$\overline{RESET}$	—	Output	Reset signal	$\overline{RESET}$
—	RESET_OUT	Output		
—	TOOL0	I/O	Transmit/receive signal	TOOL0
SI/RxD	—	I/O	Transmit/receive signal	

Note 1. Pins to be connected differ with the product. For details, see **Table 33 - 1**.
Note 2. Connect the REGC pin to the ground via a capacitor (0.47 to 1 μF).

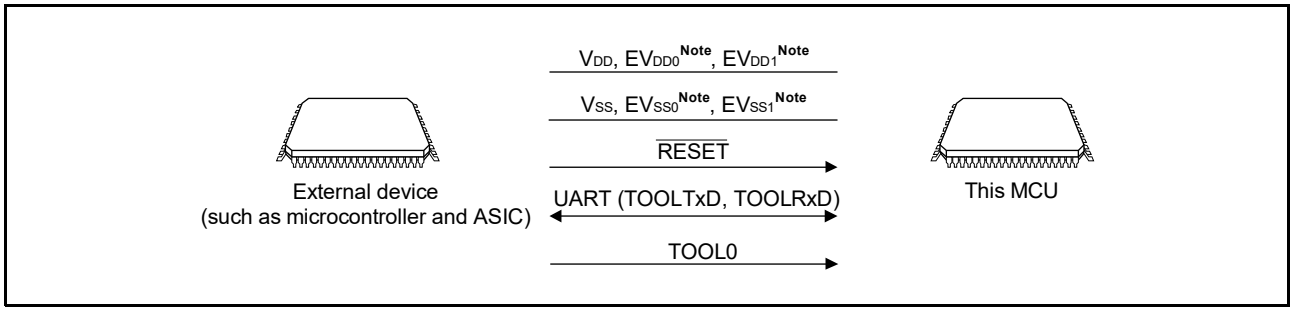
33.2 Serial Programming Using External Device (that Incorporates UART)

On-board data writing to the flash memory of this MCU is possible by using an external device (a microcontroller or ASIC) that is connected to the microcontroller through a UART. On the development of flash memory programmer by user, refer to the application note, **RL78 Microcontroller (RL78 Protocol C) Serial Programming Guide (R01AN5756)**.

33.2.1 Programming environment

Figures 33 - 3 shows the environment required for writing a program to the flash memory of this MCU.

Figure 33 - 3 Environment for Writing Program to Flash Memory



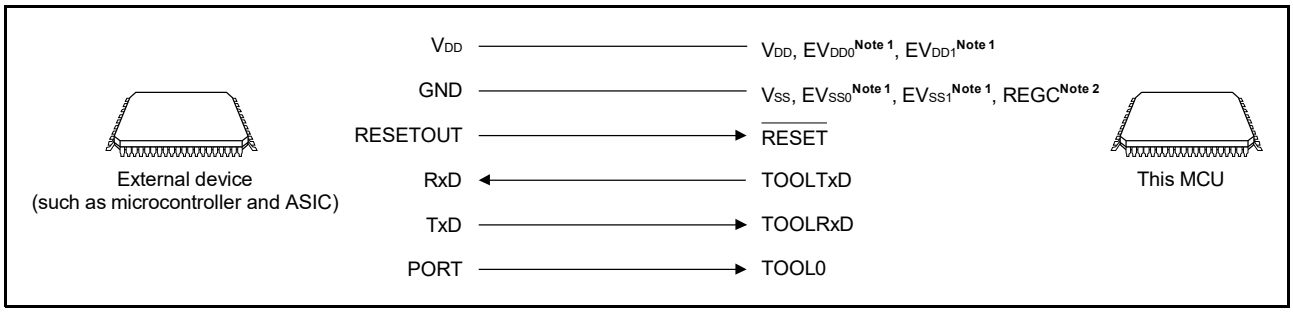
Note This pin is only present in the 64- to 128-pin R7F100G products.

33.2.2 Communications mode

The TOOLTxD and TOOLRxD pins of this MCU are used for communications between the external device and this MCU through serial communications by using the dedicated UART.

Transfer rate: 1 Mbps, 500 kbps, 250 kbps, 115.2 kbps

Figure 33 - 4 Communications with External Device



Note 1. This pin is only present in the 64- to 128-pin R7F100G products.

Note 2. Connect the REGC pin to the ground via a capacitor (0.47 to 1 µF).

The external device generates the following signals for this MCU.

Table 33 - 3 Pin Connection

External Device			This MCU
Signal Name	I/O	Pin Function	Pin Name
VDD	I/O	VDD voltage generation/power monitoring	VDD, EVDD0Note 1, EVDD1Note 1
GND	—	Ground	VSS, EVSS0Note 1, EVSS1Note 1, REGCNote 2
RESETOUT	Output	Reset signal output	$\overline{\text{RESET}}$
RxD	Input	Receive signal	TOOLTxD
TxD	Output	Transmit signal	TOOLRxD
PORT	Output	Mode signal	TOOL0

Note 1. This pin is only present in the 64- to 128-pin R7F100G products.

Note 2. Connect the REGC pin to the ground via a capacitor (0.47 to 1 µF).

33.3 Handling of Pins on the Board

To write to the flash memory on board by using a flash memory programmer, mount a connector for the dedicated flash memory programmer on the target system. In addition, include a function for mode switching from normal operation to flash memory programming in the board design. Entry to flash memory programming mode places all pins that are not to be used in programming the flash memory in the same states as those immediately after a reset. If the given states do not suit the operation of external devices, apply appropriate handling to the pins.

Remark For details on flash memory programming mode, refer to **33.4.2 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 the low level is prohibited for t_{HD} period after external reset release. However, when this pin is used via a pull-down resistor, use a resistor with a value of 500 kΩ or more.
When used as an output pin: When this pin is used via a pull-down resistor, use a resistor with a value of 500 kΩ or more.

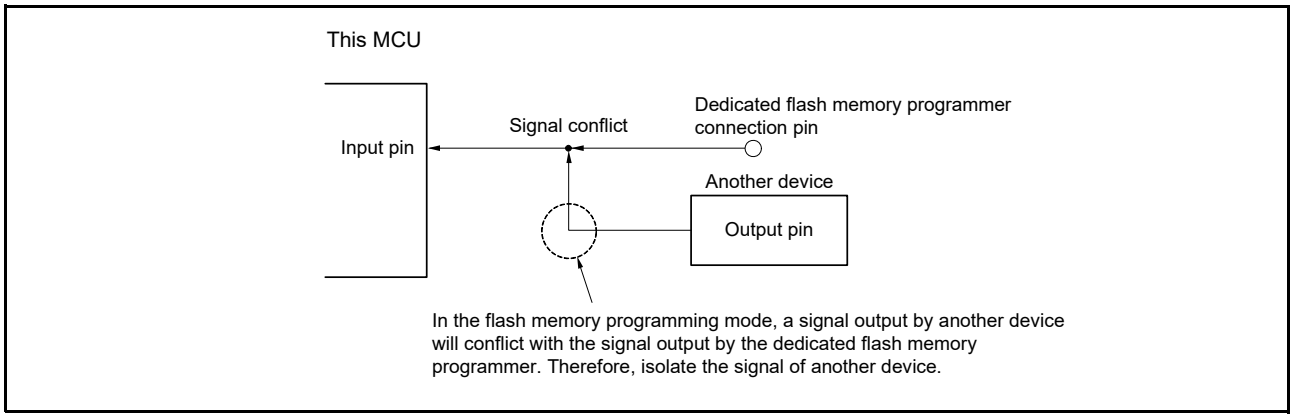
Remark 1. t_{HD}: This is the time over which the TOOL0 pin must be kept at the low level following the end of the external and internal resets for setting of the flash memory programming mode. For details, see **37.10 Timing of Entry to Flash Memory Programming Modes**.

Remark 2. The SAU and IICA pins are not used for communications between this MCU and dedicated flash memory programmer, because the single-line UART (TOOL0 pin) is used.

33.3.2 $\overline{\text{RESET}}$ pin

Signal conflict will occur if the reset signal of the dedicated flash memory programmer and external device is connected to the $\overline{\text{RESET}}$ pin that is connected to the reset signal generator on the board. To prevent this conflict, isolate the connection with the reset signal generator. The flash memory will not be correctly programmed if the reset signal is input from the user system while the flash memory programming mode is set. Do not input any signal other than the reset signal of the dedicated flash memory programmer and external device.

Figure 33 - 5 Signal Conflict ($\overline{\text{RESET}}$ Pin)



33.3.3 Port pins

Entry to flash memory programming mode places all pins that are not to be used in programming the flash memory in the same states as those immediately after a reset. If the given states of the port pins do not suit the operation of external devices that are connected to individual port pins, apply appropriate handling to the pins such as connection to VDD or VSS via resistors.

33.3.4 REGC pin

Connect the REGC pin to GND via a capacitor having excellent characteristics (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 Power supply

To use the supply voltage output of the flash memory programmer, connect the VDD pin to VDD[Note] of the flash memory programmer, and the VSS pin to GND of the flash memory programmer. To use the on-board supply voltage, connect the power supply in the same manner as during normal operation. 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 VDD and VSS pins to VDD[Note] and GND of the flash memory programmer to use the power monitor function by the flash memory programmer.

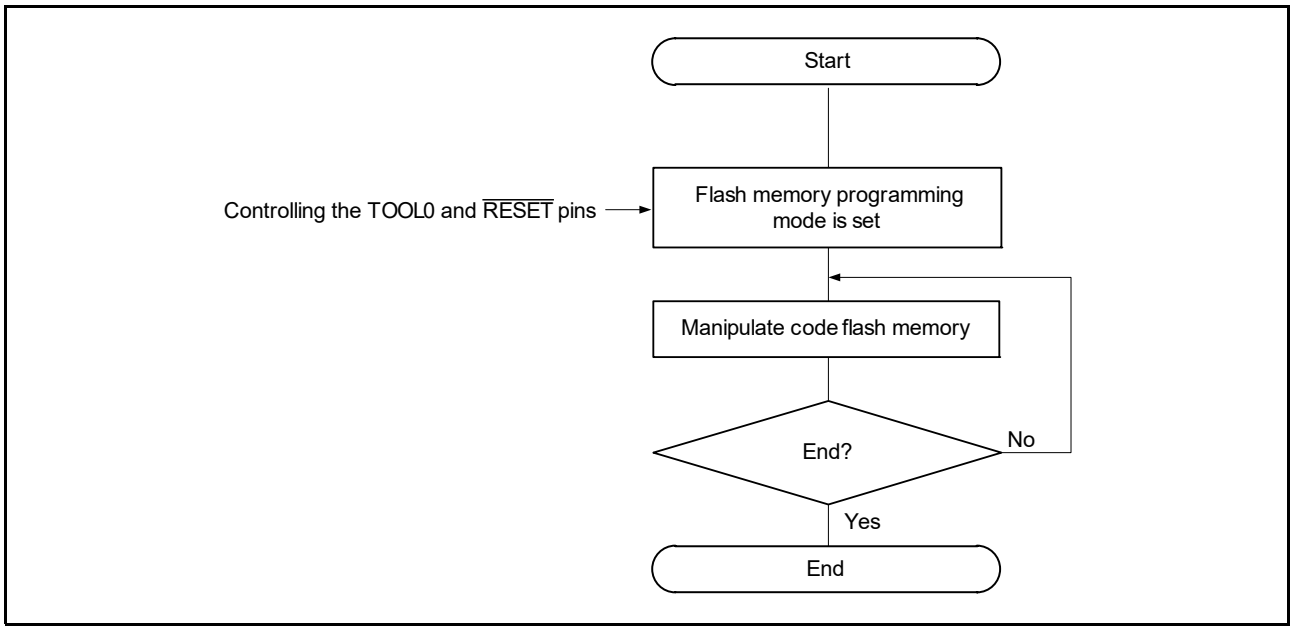
Note The signal name for the PG-FP6 is VCC.

33.4 Programming Method

33.4.1 Serial programming procedure

The following figure illustrates a flow for rewriting the code flash memory through serial programming.

Figure 33 - 6 Code Flash Memory Manipulation Procedure



33.4.2 Flash memory programming mode

To rewrite the contents of the code flash memory through serial programming, specify the flash memory programming mode. The following describes how to enter the flash memory programming mode.

<For serial programming by using the dedicated flash memory programmer>

Connect this MCU to a dedicated flash memory programmer. Starting communications with the dedicated flash memory programmer automatically places this MCU in the flash memory programming mode.

<For serial programming by using an external device (UART communications)>

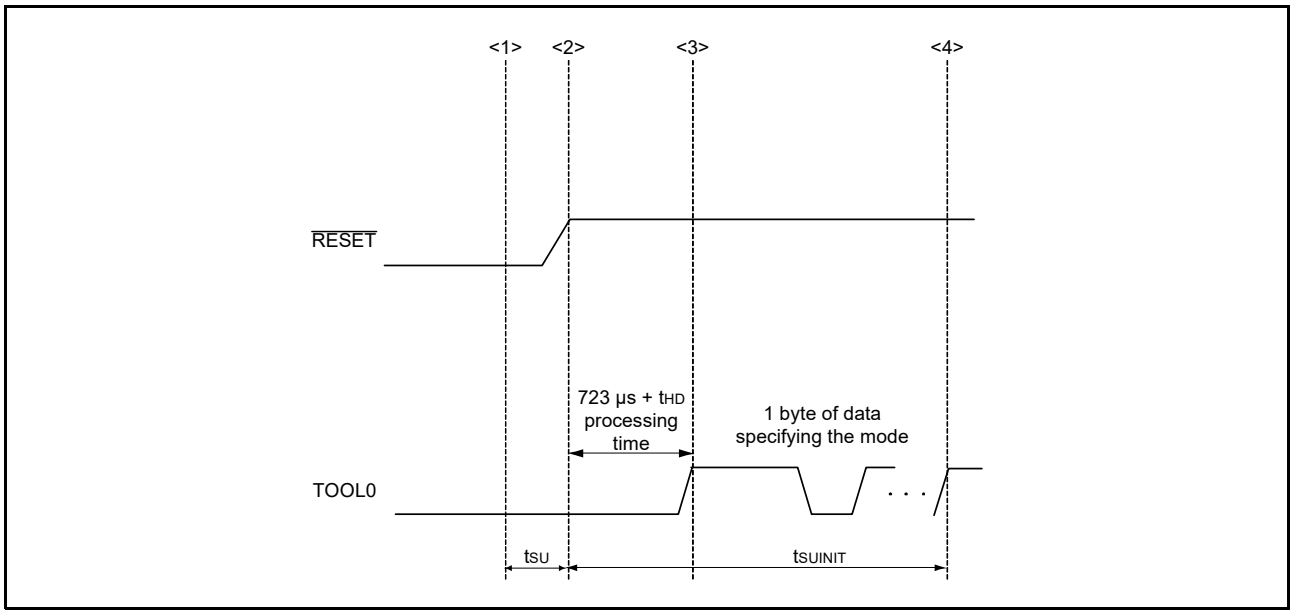
Set the TOOL0 pin to the low level, and then release the MCU from the reset state (see **Table 33 - 4**). After that, enter flash memory programming mode according to steps <1> to <4> shown in **Figure 33 - 7**. For details, refer to the application note, **RL78 Microcontroller (RL78 Protocol C) Serial Programming Guide (R01AN5756)**.

Table 33 - 4 Relationship between the Voltage Applied to the TOOL0 Pin and the Operating Mode after Release from the Reset State

TOOL0	Operating Mode
EV_{DD}	Normal operation mode
0 V	Flash memory programming mode

Remark Pins EV_{DD0} and EV_{DD1} are not present in some products. In such cases, use the V_{DD} pins as the alternative.

Figure 33 - 7 Entry to Flash Memory Programming Mode



<1> The low level is input to the TOOL0 pin.

<2> The external reset is de-asserted (release from a POR or LVD reset must precede this).

<3> The TOOL0 pin is set to the high level.

<4> The bit rate setting is based on the UART reception.

Remark t_{SUINIT} : The section is the up to 100 ms from the end of the external reset within which specifying the initial communications settings must be finished.

t_{SU} : This is the time from the TOOL0 pin being placed at the low level until the end of the external reset.

t_{HD} : This is the time over which the TOOL0 pin must be kept at the low level following the end of an external reset (excluding the processing time of the firmware to control the flash memory).

For details, see **37.10 Timing of Entry to Flash Memory Programming Modes**.

33.4.3 Selecting communications mode

Table 33 - 5 shows the communications modes of this MCU.

Table 33 - 5 Communications Modes

Communications Mode	Standard Setting[Note 1]				Pins Used
	Port	Speed[Note 2]	Frequency	Multiply Rate	
Single-line UART (when a flash memory programmer is used, or when an external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	—	—	TOOL0
Dedicated UART (when an external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	—	—	TOOLTxD, TOOLRxD

Note 1. Selection items for standard settings on GUI of the flash memory programmer.

Note 2. As factors other than bit rate errors, such as dullness of the signal waveform, may also affect UART communications, conduct extensive evaluation with a selected bit rate before attempting to use it.

33.4.4 Communications commands

This MCU executes serial programming through the commands listed in **Table 33 - 6**. The signals sent from the dedicated flash memory programmer or external device to this MCU are called commands, and programming functions corresponding to the commands are executed. For details, refer to the application note, **RL78 Microcontroller (RL78 Protocol C) Serial Programming Guide (R01AN5756)**.

Table 33 - 6 Flash Memory Control Commands

Classification	Command Name	Function
Verification	Verify	Compares the contents of a specified area of the flash memory with data transmitted from the programmer.
Erasure	Block Erase	Erases a specified area in the flash memory.
Blank checking	Block Blank Check	Checks if a specified block in the flash memory has been correctly erased.
Writing	Programming	Writes data to a specified area in the flash memory**Note**.
Getting information	Silicon Signature	Gets the RL78 microcontroller information (such as the device name, 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	Releases the setting to prohibit writing.
Others	Reset	Used to detect the state of synchronization during communications.
	Baud Rate Set	Sets the bit rate when UART communications mode is selected.

Note Confirm that no data have been written to the write area. If data in the area has not been erased, do not attempt further writing of data because data cannot be erased after the setting to prohibit block erasure has been made.

The product information (such as device name and firmware version) can be obtained by executing the Silicon Signature command. **Tables 33 - 7** and **33 - 8** show the signature data list and an example of signature data, respectively.

Table 33 - 7 Signature Data List

Field Name	Description	Number of Bytes 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 end address	End address of the code flash memory area (Sent from the lower-order byte of the address. Example: 00000H to 1FFFFH (128 Kbytes) → FFH, FFH, 01H)	3 bytes
Data flash memory area end address	End address of the data flash memory area (Sent from the lower-order byte of the address. Example: F1000H to F2FFFH (8 Kbytes) → FFH, 2FH, 0FH)	3 bytes
Firmware version	Version information of firmware for programming (Sent from the byte equivalent to the highest-order digit of the version number. Example: Ver. 1.02 → 01H, 00H, 02H)	3 bytes

Table 33 - 8 Example of Signature Data

Field Name	Description	Number of Bytes of Transmit Data	Data (Hexadecimal)
Device code	RL78 protocol C	3 bytes	10 00 0A
Device name	R7F100GLG	10 bytes	52 = “R” 37 = “7” 46 = “F” 31 = “1” 30 = “0” 30 = “0” 47 = “G” 4C = “L” 47 = “G” 20 = “ ”
Code flash memory area end address	Code flash memory area 00000H to 1FFFFH (128 Kbytes)	3 bytes	FF FF 01
Data flash memory area end address	Data flash memory area F1000H to F2FFFH (8 Kbytes)	3 bytes	FF 2F 0F
Firmware version	Ver. 1.02	3 bytes	01 00 02

33.5 Processing Times for Commands When the Dedicated Flash Memory Programmer Is in Use (Reference Values)

The following shows the processing times for each command (reference value) when the PG-FP6 is used as a dedicated flash memory programmer.

Table 33 - 9 Processing Times for Commands When the PG-FP6 Is in Use (Reference Values)

PG-FP6 Command	Code Flash Memory						
	96 Kbytes	128 Kbytes	192 Kbytes	256 Kbytes	384 Kbytes	512 Kbytes	768 Kbytes
Erasure	1.2 s	1.3 s	1.7 s	2.1 s	2.8 s	3.5 s	5 s
Writing	2.4 s	2.9 s	4 s	5.2 s	7.3 s	9.5 s	14.1 s
Verification	1.7 s	2.2 s	3 s	3.9 s	5.6 s	7.3 s	10.9 s
Writing after erasure	3.2 s	4 s	5.4 s	6.9 s	9.8 s	12.7 s	18.8 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

33.6 Self-programming

This MCU supports a self-programming function that can be used to rewrite the flash memory via a user program. Because this function allows a user application to rewrite the flash memory, it can be used to update the program in the field.

For details, refer to the **RL78 Family Renesas Flash Driver RL78 Type01 User's Manual (R20UT4830EJ)**.

Caution 1. The self-programming function cannot be used when the CPU operates with the subsystem clock (f_{SUB}).

Caution 2. To prohibit an interrupt during self-programming, in the same way as in the normal operation mode, execute self-programming in the state where the IE flag is cleared to 0 by the DI instruction. To enable an interrupt, clear the interrupt mask flag for an interrupt to be accepted to 0 in the state where the IE flag is set to 1 by the EI instruction, and then execute self-programming.

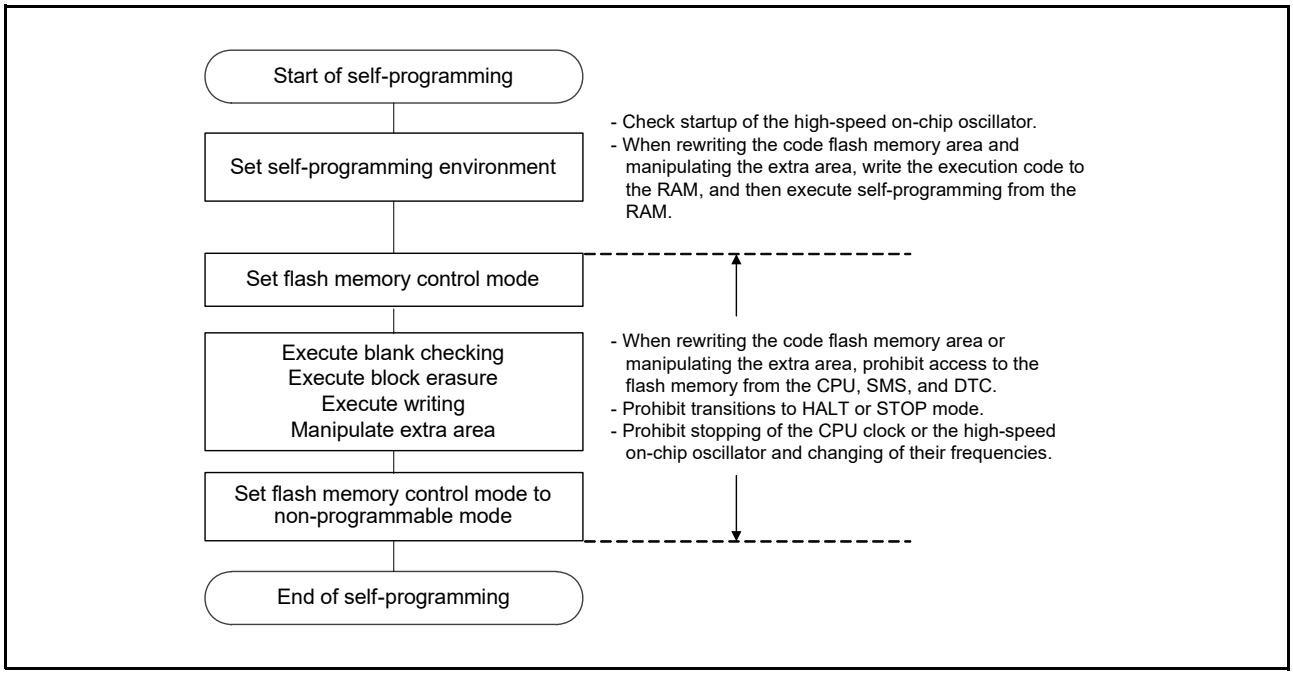
Caution 3. The high-speed on-chip oscillator should be kept operating during self-programming. If it is stopped, it should be made to operate again (HIOSTOP = 0), and self-programming should be re-executed after 5 μs have elapsed. Stop the middle-speed on-chip oscillator (MIOEN = 0) and select the high-speed on-chip oscillator (MCM1 = 0) as the main on-chip oscillator clock (f_{OCO}).

Caution 4. Do not change the flash operating mode select register (FLMODE) while the flash memory is being rewritten.

33.6.1 Self-programming procedure

The following figure illustrates a flow for rewriting the flash memory by using self-programming. For details on registers for use in self-programming, see **33.6.2 Registers for controlling the flash memory**.

Figure 33 - 8 Flow of Self-programming (Rewriting the Flash Memory)



33.6.2 Registers for controlling the flash memory

The following registers are used to control the flash memory.

- Flash address pointer registers H and L (FLAPH, FLAPL)
- Flash end address pointer registers H and L (FLSEDH, FLSEDL)
- Flash write buffer registers H and L (FLWH, FLWL)
- Flash protect command register (PFCMD)
- Flash status register (PFS)
- Flash programming mode control register (FLPMC)
- Flash area selection register (FLARS)
- Flash memory sequencer initial setting register (FSSET)
- Flash memory sequencer control register (FSSQ)
- Flash extra area sequencer control register (FSSE)
- Flash registers initialization register (FLRST)
- Flash memory sequencer status registers H and L (FSASTH, FSASTL)
- Flash security flag monitoring register (FLSEC)
- Flash FSW monitoring register E (FLFSWE)
- Flash FSW monitoring register S (FLFSWS)
- Data flash control register (DFLCTL)
- Interrupt vector jump enable register (VECTCTRL)
- Interrupt vector change registers 0 and 1 (FLSIVC0, FLSIVC1)

33.6.2.1 Flash address pointer registers H and L (FLAPH, FLAPL)

The FLAPH and FLAPL registers specify the address where programming the flash memory is to start. The FLAPH and FLAPL registers can be set by 8-bit and 16-bit memory manipulation instructions, respectively. 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.

Figure 33 - 9 Format of Flash Address Pointer Registers H and L (FLAPH, FLAPL)

Address: F02C4H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
FLAPH	0	0	0	0	FLAP19	FLAP18	FLAP17	FLAP16

Address: F02C2H, F02C3H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
FLAPL	FLAP15	FLAP14	FLAP13	FLAP12	FLAP11	FLAP10	FLAP9	FLAP8

7	6	5	4	3	2	1	0
FLAP7	FLAP6	FLAP5	FLAP4	FLAP3	FLAP2	FLAP1	FLAP0

Caution 1. The FLAPH and FLAPL registers can be rewritten under either of the following conditions.
• The FLSPM bit in the FLPMC register is 1 (code flash memory area: programming mode).
• The EEEMD bit in the FLPMC register is 1 (data flash memory area: programming mode).

Caution 2. Rewrite or read these registers when the extra area sequencer and the code/data flash memory area sequencer are stopped (SQEND = 0, ESQEND = 0 in the FSASTH register, SQST = 0 in the FSSQ register, and ESQST = 0 in the FSSE register).

33.6.2.2 Flash end address pointer registers H and L (FLSEDH, FLSEDL)

The FLSEDH and FLSEDL registers specify the address where programming of the flash memory is to end. The FLSEDH and FLSEDL registers can be set by 8-bit and 16-bit memory manipulation instructions, respectively. 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.

Figure 33 - 10 Format of Flash End Address Pointer Registers H and L (FLSEDH, FLSEDL)

Address: F02C8H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
FLSEDH	0	0	0	0	EWA19	EWA18	EWA17	EWA16

Address: F02C6H, F02C7H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
FLSEDL	EWA15	EWA14	EWA13	EWA12	EWA11	EWA10	EWA9	EWA8

7	6	5	4	3	2	1	0
EWA7	EWA6	EWA5	EWA4	EWA3	EWA2	EWA1	EWA0

Caution 1. The FLSEDH and FLSEDL registers can be rewritten under either of the following conditions.
• The FLSPM bit in the FLPMC register is 1 (code flash memory area: programming mode).
• The EEEMD bit in the FLPMC register is 1 (data flash memory area: programming mode).

Caution 2. Rewrite or read these registers when the extra area sequencer and the code/data flash memory area sequencer are stopped (SQEND = 0, ESQEND = 0 in the FSASTH register, SQST = 0 in the FSSQ register, and ESQST = 0 in the FSSE register).

Caution 3. The settings of the EWA1 and EWA0 bits are meaningless during programming of the code flash memory.

Table 33 - 10 Method of Setting the FLAPH, FLAPL, FLSEDH, and FLSEDL Registers

Commands Exclusively for Use with the Code/Data Flash Memory Area Sequencer		Settings of the FLAP and FLSED Registers	
FSSQ	Writing	FLAPH, FLAPL: Address from which writing is to proceed FLSEDH, FLSEDL: All 0s	
	Blank checking	For one word: Settings of FLAPH and FLAPL = settings of FLSEDH and FLSEDL For two or more words: Settings of FLAPH and FLAPL < settings of FLSEDH and FLSEDL	
	Block erasure**Note**	Code flash memory	FLAPH, FLAPL: Set the start address and 0s in the FLAP19 to FLAP11 bits and the FLAP10 to FLAP2 bits, respectively.
			FLSEDH, FLSEDL: Set the end address and 1s in the EWA19 to EWA11 bits and the EWA10 to EWA2 bits, respectively.
		Data flash memory	FLAPH, FLAPL: Set the start address and 0s in the FLAP19 to FLAP8 bits and the FLAP7 to FLAP0 bits, respectively.
			FLSEDH, FLSEDL: Set the end address and 1s in the EWA19 to EWA8 bits and the EWA7 to EWA0 bits, respectively.
FSSE	All commands	Settings of FLAPH and FLAPL: All 0s Settings of FLSEDH and FLSEDL: All 0s	

Note Set the FLAPH, FLAPL, FLSEDH, and FLSEDL registers so that the following condition is met.
Combined settings of FLAPH and FLAPL ≤ combined settings of FLSEDH and FLSEDL

33.6.2.3 Flash write buffer registers H and 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 value of each of the FLWH and FLWL registers 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.

Set data to be written to the data flash memory in the 8 lower-order bits of the FLWL register.

Figure 33 - 11 Format of Flash Write Buffer Registers H and L (FLWH, FLWL)

Address: F02CEH, F02CFH
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
FLWH	FLW31	FLW30	FLW29	FLW28	FLW27	FLW26	FLW25	FLW24

7	6	5	4	3	2	1	0
FLW23	FLW22	FLW21	FLW20	FLW19	FLW18	FLW17	FLW16

Address: F02CCH, F02CDH
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
FLWL	FLW15	FLW14	FLW13	FLW12	FLW11	FLW10	FLW9	FLW8

7	6	5	4	3	2	1	0
FLW7	FLW6	FLW5	FLW4	FLW3	FLW2	FLW1	FLW0

Caution 1. The FLWH and FLWL registers can be rewritten under either of the following conditions.
• The FLSPM bit in the FLPMC register is 1 (code flash memory area: programming mode).
• The EEEMD bit in the FLPMC register is 1 (data flash memory area: programming mode).

Caution 2. Rewrite or read these registers when the extra area sequencer and the code/data flash memory area sequencer are stopped (SQEND = 0, ESQEND = 0 in the FSASTH register, SQST = 0 in the FSSQ register, and ESQST = 0 in the FSSE register).

Caution 3. When writing to the data flash memory, set write data in the 8 lower-order bits of the FLWL register. Set other bits to 0.

33.6.2.4 Flash protect command register (PFCMD)

The PFCMD register protects the flash programming mode control register (FLPMC) against write access. To enable write access to the FLPMC register, write A5H according to the specific sequence. For details on the procedure for handling the specific sequence, see **33.6.3 Setting the flash memory control mode**. The PFCMD register can be set by an 8-bit memory manipulation instruction.

Figure 33 - 12 Format of Flash Protect Command Register (PFCMD)

Address: F00C0H
After reset: Undefined
R/W: W

Symbol	7	6	5	4	3	2	1	0
PFCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0

33.6.2.5 Flash status register (PFS)

The PFS register indicates whether or not a protection error has occurred during write access to the flash programming mode control register (FLPMC). For details on the conditions for setting and clearing the FPRERR bit, see **33.6.3.1 Procedure for executing the specific sequence**. The PFS register can be read by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 33 - 13 Format of Flash Status Register (PFS)

Address: F00C1H
After reset: 00H
R/W: 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.

33.6.2.6 Flash programming mode control register (FLPMC)

The FLPMC register disables or enables writing to the flash memory and selects the programming mode. Enabling write access to the FLPMC register requires the execution of a specific sequence. For details on the procedure for handling the specific sequence, see **33.6.3 Setting the flash memory control mode**. The FLPMC register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 08H.

Figure 33 - 14 Format of Flash Programming Mode Control Register (FLPMC)

Address: F02C0H
After reset: 08H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
FLPMC	0	0	0	EEEMD	FWEDIS	0	FLSPM	0

EEEMD	Selection of the programming mode for the data flash memory
0	Non-programmable mode
1	Programming mode

FWEDIS	Software control over enabling or disabling erasure and programming of the code flash memory[Note]
0	Enables erasure and programming.
1	Disables erasure and programming.

FLSPM	Selection of the programming mode for the code flash memory
0	Non-programmable mode
1	Programming mode

Note Be sure to keep the value of this bit at 0 until erasure or programming of the code flash memory is completed.

Caution **When the extra area sequencer and the code/data flash memory area sequencer are stopped (SQEND = 0, ESQEND = 0 in the FSASTH register, SQST = 0 in the FSSQ register, and ESQST = 0 in the FSSE register), rewriting to the FLPMC register is enabled.**

33.6.2.7 Flash area selection register (FLARS)

The FLARS register selects the area of the flash memory for self-programming. The FLARS 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 - 15 Format of Flash Area Selection Register (FLARS)

Address: F02C1H
After reset: 00H
R/W: 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 the flash memory for self-programming
0	Code/data flash memory areas
1	Extra area

Caution The FLARS register can be rewritten under either of the following conditions.
• The FLSPM bit in the FLPMC register is 1 (code flash memory area: programming mode).
• The EEEMD bit in the FLPMC register is 1 (data flash memory area: programming mode).

33.6.2.8 Flash memory sequencer initial setting register (FSSET)

The FSSET register sets the operating frequency of the flash memory sequencer and makes the initial settings for boot swapping. The FSSET register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 33 - 16 Format of Flash Memory Sequencer Initial Setting Register (FSSET)

Address: F00B6H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
FSSET	TMSPMD	TMBTSEL	0	FSET4	FSET3	FSET2	FSET1	FSET0

TMSPMD	Selection of boot area setting **Note**
0	Specifies the boot area according to the setting of EX bit 8 (BTFLG) in the security flag and boot swap function setting area of the extra area. BTFLG = 0: Boot cluster 1 as the boot area BTFLG = 1: Boot cluster 0 as the boot area
1	Specifies the boot area according to the setting of the TMBTSEL bit.

TMBTSEL	Specification of the boot area when TMSPMD = 1
0	Specifies boot cluster 0 as the boot area.
1	Specifies boot cluster 1 as the boot area.

FSET4 to FSET0	Setting of the operating frequency of the flash memory sequencer
—	Sets the operating frequency of the flash memory sequencer. For the correspondence between the operating frequency of the flash memory sequencer and the setting of the FSET4 to FSET0 bits, see **Table 33 - 11**.

Note Setting the TMSPMD and TMBTSEL bits is not possible while the BTPR bit in FLSEC is 0 (rewriting of the boot area is disabled).

Caution 1. The FSSET register can be rewritten under either of the following conditions.
- **The FLSPM bit in the FLPMC register is 1 (code flash memory area: programming mode).**
- **The EEEMD bit in the FLPMC register is 1 (data flash memory area: programming mode).**

Caution 2. Setting of the boot area by the BTFLG bit is immediately reflected. To change the boot area after a reset is released, read the MBTSEL bit in FSASTL while TMSPMD = 0 and set the same value to the TMBTSEL bit. After that, set the TMSPMD bit to 1, and then specify the boot cluster to be activated as the boot area at release from the reset state in the BTFLG bit using the extra area sequencer. The boot cluster set by the BTFLG bit is activated as the boot area at the next reset release.

Table 33 - 11 Correspondence between the Operating Frequency of the Flash Memory Sequencer and the Setting of the FSET4 to FSET0 Bits

Operating Frequency (MHz)	Setting of the FSET4 to FSET0 Bits	Operating Frequency (MHz)	Setting of the FSET4 to FSET0 Bits	Operating Frequency (MHz)	Setting of the FSET4 to FSET0 Bits
32	11111B	31	11110B	30	11101B
29	11100B	28	11011B	27	11010B
26	11001B	25	11000B	24	10111B
23	10110B	22	10101B	21	10100B
20	10011B	19	10010B	18	10001B
17	10000B	16	01111B	15	01110B
14	01101B	13	01100B	12	01011B
11	01010B	10	01001B	9	01000B
8	00111B	7	00110B	6	00101B
5	00100B	4	00011B	3	00010B
2	00001B	1	00000B	—	—

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 4.5 MHz, set the bits to 00100B for 5 MHz. Note that frequencies that are not whole numbers, such as 1.5 MHz, are not available as CPU operating frequencies below 4 MHz.**

33.6.2.9 Flash memory sequencer control register (FSSQ)

The FSSQ register specifies operation control and commands for use with the code/data flash memory area sequencer. When the SQST bit in this register is set to 1, the code/data flash memory area sequencer executes the command set in the MDCH, SQMD2, SQMD1, and SQMD0 bits. The FSSQ 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 - 17 Format of Flash Memory Sequencer Control Register (FSSQ)

Address: F02C5H
After reset: 00H
R/W: R/W

Symbol	<7>	<6>	5	4	3	2	1	0
FSSQ	SQST	FSSTP	0	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.Note 1
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.

MDCH	SQMD2	SQMD1	SQMD0	Commands for use with the code/data flash memory area sequencer
0	0	0	1	• Writing Writes data stored in the FLWH and FLWL registers to the address specified in the FLAPH and FLAPL registers. Note 2 Writes 4-byte data when the code flash memory area address is specified. Writes the 1-byte data stored in the eight lower-order bits (FLW7 to FLW0) in FLWL to the specified address when the data flash memory area address is specified.
0	0	1	1	• Blank checking of the code flash memory area Checks whether the value of the code flash memory area from the address specified in the FLAPH and FLAPL registers to the address specified in the FLSEDH and FLSEDL registers is 1. Note 3
1	0	1	1	• Blank checking of the data flash memory area Checks whether the value of the data flash memory area from the address specified in the FLAPH and FLAPL registers to the address specified in the FLSEDH and FLSEDL registers is 1.
0	1	0	0	• Block erasure Erases blocks in the range from the block start address specified in the FLAPH and FLAPL registers to the block end address specified in the FLSEDH and FLSEDL registers.Note 4
Other than above				Setting prohibited

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

Note 1. Check that the SQEND bit in the FSASTH register is 1 (the code/data flash memory area sequencer being stopped), and then set the SQST bit to 0 to stop the code/data flash memory area sequencer.

<R> **Note 2.** Four-byte data can be written to the code flash memory area. Set the two lower-order bits of the FLAPL register to 00B to be a multiple of 4. For details, see **33.6.6.4 Operations for rewriting the code flash memory area**.

<R> **Note 3.** Specify a start address (at intervals of four bytes) for blank checking of the code flash memory area. Set the two lower-order bits of the FLAPL register to 00B to be a multiple of 4. For details, see **33.6.6.4 Operations for rewriting the code flash memory area**.

Note 4. The code flash memory area blocks can be erased in units of 2 Kbytes. The data flash memory blocks can be erased in units of 256 bytes. Specify the erase addresses (start address and end address) so that all blocks to be erased are included. For details, see **33.6.6.4 Operations for rewriting the code flash memory area** and **33.6.6.5 Operations for rewriting the data flash memory area**. For the relationship between the address and block number, see **Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory**.

Caution The FSSQ register can be rewritten under either of the following conditions.

- **The FLSPM bit in the FLPMC register is 1 (code flash memory area: programming mode) and the FWEDIS bit is 0 (enabling erasure and programming of the code flash memory).**
- **The EEEMD bit in the FLPMC register is 1 (data flash memory area: programming mode).**

33.6.2.10 Flash extra area sequencer control register (FSSE)

The FSSE register specifies operation control and commands for use with the extra area sequencer. When the ESQST bit of the FSSE register is set to 1, the extra area sequencer executes the command set by the ESQMD3 to ESQMD0 bits. The FSSE 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 - 18 Format of Flash Extra Area Sequencer Control Register (FSSE)

Address: F00B7H
After reset: 00H
R/W: R/W

Symbol	<7>	6	5	4	3	2	1	0
FSSE	ESQST	0	0	0	ESQMD3	ESQMD2	ESQMD1	ESQMD0

ESQST	Operation control of the extra area sequencer
0	The extra area sequencer is stopped.**Note**
1	The extra area sequencer is started.

ESQMD3	ESQMD2	ESQMD1	ESQMD0	Commands for use with the extra area sequencer
0	0	0	1	• Write to the flash shield window setting area Writes the 4-byte data specified in the FLWH and FLWL registers to the flash shield window setting area of the extra area to control flash shield window mode and set the start block and end block. Also, if the setting of EX bit 15 (FSPR) in the flash shield window setting area is 0, no value can be written to the area. Attempted writing leads to setting of the extra area sequencer error flag (ESEQER) to 1.
0	1	1	0	• Write to the flash read protection setting area Writes the 4-byte data specified in the FLWH and FLWL registers to the flash read protection setting area of the extra area to disable changing of the flash read protection setting and set the start block and end block. Also, if the setting of EX bit 31 (SWPR) in the flash read protection setting area is 0, no value can be written to the area. Attempted writing leads to setting of the extra area sequencer error flag (ESEQER) to 1.
0	1	1	1	• Write to the security flag and boot swap function setting area Writes the 4-byte data specified in the FLWH and FLWL registers to the flash memory security flag and boot swap function setting area of the extra area to disable block erasure, writing, and rewriting of the boot area and set selection of the boot area. Also, if the setting of EX bit 9 (BTPR) in the security flag and boot swap function setting area is 0, no value can be written to the area. Attempted writing leads to setting of the extra area sequencer error flag (ESEQER) to 1.
Other than above				Setting prohibited

(Note and Cautions are listed on the next page.)

Note Check that the ESQEND bit in the FSASTH register is 1 (the extra area sequencer being stopped), and then set the ESQST bit to 0 to stop the extra area sequencer.

Caution 1. The FSSE register can be rewritten when the following conditions are both satisfied.
- **The FLSPM bit in the FLPMC register is 1 (code flash memory area: programming mode)**
- **The FWEDIS bit is 0 (enabling erasure and programming of the code flash memory)**

Caution 2. To write to 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.

Caution 3. Rewrite the ESQMD3 to ESQMD0 bits while the extra area sequencer and the code/data flash memory area sequencer are stopped (SQEND = 0, ESQEND = 0 in the FSASTH register, SQST = 0 in the FSSQ register, and ESQST = 0 in the FSSE register).

33.6.2.11 Flash registers initialization register (FLRST)

The FLRST register initializes all registers related to the sequencer whether it is to be used with the extra area or code/data flash memory areas. The FLRST register can be set by a 1-bit or 8-bit memory manipulation instruction. For details about how to manipulate the FLRST register, see **33.6.4 Initializing the registers for use with the flash memory sequencer**. The value of this register following a reset is 00H.

Figure 33 - 19 Format of Flash Registers Initialization Register (FLRST)

Address: F02C9H
After reset: 00H
R/W: 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
0	The registers are not reset.
1	The FLAPH, FLAPL, FLSEDH, FLSEDL, FLWH, FLWL, FLARS, FSSQ, and FSSE registers are reset.

Caution 1. Registers can be initialized by setting the FLRST bit to 1 only when the extra area sequencer and the code/data flash memory area sequencer are stopped (SQEND = 0, ESQEND = 0 in the FSASTH register, SQST = 0 in the FSSQ register, and ESQST = 0 in the FSSE register).

Caution 2. When using the sequencer, be sure to set the FLRST bit to 0 before setting the FLAPH, FLAPL, FLSEDH, FLSEDL, FLWH, FLWL, FLARS, FSSQ, and FSSE registers. Do not set the FLRST bit to 1 during operation of the sequencer.

33.6.2.12 Flash memory sequencer status registers H and L (FSASTH, FSASTL)

The FSASTH and FSASTL registers indicate the results of the respective operations of the flash memory sequencer when it has been used with the extra area or code/data flash memory areas. The FSASTH and FSASTL registers can be read by a 1-bit or 8-bit memory manipulation instruction.

Figure 33 - 20 Format of Flash Memory Sequencer Status Registers H and L (FSASTH, FSASTL) (1/2)

Address: F02CBH
After reset: 00H/04H
R/W: R

Symbol	7	6	5	4	3	2	1	0
FSASTH	ESQEND	SQEND	0	0	0	×	0	0

Address: F02CAH
After reset: 00H/80H Note 1
R/W: R

Symbol	7	6	5	4	3	2	1	0
FSASTL	MBTSEL	MOPEN	ESEQER	SEQER	BLER	0	WRER	ERER

ESQEND	Extra area sequencer operation end status flag
0	The extra area sequencer is operating or stopped by setting the ESQST bit to 0.
1	The extra area sequencer is stopped.

SQEND	Code/data flash memory area sequencer operation end status flag
0	The code/data flash memory area sequencer is operating or stopped by setting the SQST bit to 0.
1	The code/data flash memory area sequencer is stopped.

MBTSEL	Boot flag monitoring bit Note 2
0	BTFLG = 1 (boot area: boot cluster 0)
1	BTFLG = 0 (boot area: boot cluster 1)

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.
This flag is cleared to 0 when the extra area sequencer is activated.	

Figure 33 - 20 Format of Flash Memory Sequencer Status Registers H and L (FSASTH, FSASTL) (2/2)

SEQER	Error flag of the flash memory sequencer
0	No error has occurred.
1	An error has occurred.
This flag is cleared to 0 when the extra area sequencer or the code/data flash memory area sequencer is activated.	

BLER	Error flag for the blank check command
0	No error has occurred.
1	An error has occurred.
This flag is cleared to 0 when the extra area sequencer or the code/data flash memory area sequencer is activated.	

WRER	Error flag for the write command
0	No error has occurred.
1	An error has occurred.
This flag is cleared to 0 when the extra area sequencer or the code/data flash memory area sequencer is activated. The read value of this bit becomes undefined if the command is forcibly terminated during writing.	

ERER	Error flag for the block erase command
0	No error has occurred.
1	An error has occurred.
This flag is cleared to 0 when the extra area sequencer or the code/data flash memory area sequencer is activated. The read value of this bit becomes undefined if the command is forcibly terminated during block erasure.	

Note 1. The initial value of the MBTSEL bit is undefined since it depends on the value of the BTFLG bit (boot area switching flag) stored in the extra area.

Note 2. This bit indicates the inverse of the value of the BTFLG bit (boot area switching flag) stored in the extra area.

33.6.2.13 Flash security flag monitoring register (FLSEC)

The FLSEC register monitors the information on the security flag and boot swap function settings in the extra area. The FLSEC register can be read by a 16-bit memory manipulation instruction.

Figure 33 - 21 Format of Flash Security Flag Monitoring Register (FLSEC)

Address: F00B0H, F00B1H
After reset: Undefined
R/W: R

Symbol	15	14	13	12	11	10	9	8
FLSEC	0	0	0	WRPR	0	SEPR	BTPR	BTFLG

7	6	5	4	3	2	1	0
0	0	0	0	SWPR	0	IFPR	IDEN

WRPR	Write-disabled flag
0	Writing is disabled.
1	Writing is enabled.

SEPR	Block erase-disabled flag
0	Block erasure is disabled.
1	Block erasure is enabled.

BTPR	Boot area rewrite-disabled flag
0	Rewriting of the boot area is disabled.
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.

SWPR	Changing of the flash memory read protection setting disabled flag
0	Changing of the flash memory read protection setting is disabled.
1	Changing of the flash memory read protection setting is enabled.

IFPR	Connection to the programmer and on-chip debugger disabled flag
0	Connection to the programmer and on-chip debugger is disabled.
1	Connection to the programmer and on-chip debugger is enabled.

IDEN	Programmer connection ID authentication enabled flag
0	ID authentication is enabled.
1	ID authentication is disabled.

33.6.2.14 Flash FSW monitoring register E (FLFSWE)

The FLFSWE register monitors the end block number specified for the flash memory shield area and whether the shield area is inside or outside the window range. After a reset or writing to the extra area, the values in the extra area are reflected in the FLFSWE register. For details on the flash shield window function, see **33.8 Flash Shield Window Function**. The FLFSWE register can be read by a 16-bit memory manipulation instruction.

Figure 33 - 22 Format of Flash FSW Monitoring Register E (FLFSWE)

Address: F00B4H, F00B5H
After reset: Undefined
R/W: R

Symbol	15	14	13	12	11	10	9	8
FLFSWE	FSWC	0	0	0	0	0	0	FSWE8

7	6	5	4	3	2	1	0
FSWE7	FSWE6	FSWE5	FSWE4	FSWE3	FSWE2	FSWE1	FSWE0

FSWC	Setting of the flash memory shield area
0	Inside shield mode The flash memory shield area is set inside the window range.
1	Outside shield mode The flash memory shield area is set outside the window range.

FSWE8 to FSWE0	End block number of the flash memory shield area
—	End block number + 1 **Note**

Note These bits show the value set in the extra area. The actual end block number is (the value of the FSWE8 to FSWE0 bits - 1). Though the end block number is specified for serial programming, (end block number + 1) is set in the extra area. For details, see **Table 33 - 12**.

33.6.2.15 Flash FSW monitoring register S (FLFSWS)

The FLFSWS register monitors the start block number of the flash memory shield area and whether rewriting of the flash shield window setting is disabled or enabled. After a reset or writing to the extra area, the values in the extra area are reflected in the FLFSWS register. For details on the flash shield window function, see **33.8 Flash Shield Window Function**. The FLFSWS register can be read by a 16-bit memory manipulation instruction.

Figure 33 - 23 Format of Flash FSW Monitoring Register S (FLFSWS)

Address: F00B2H, F00B3H
After reset: Undefined
R/W: R

Symbol	15	14	13	12	11	10	9	8
FLFSWS	FSPR	0	0	0	0	0	0	FSWS8

7	6	5	4	3	2	1	0
FSWS7	FSWS6	FSWS5	FSWS4	FSWS3	FSWS2	FSWS1	FSWS0

FSPR	Changing of the flash shield window setting disabled flag
0	Changing of the flash shield window setting is disabled.
1	Changing of the flash shield window setting is enabled.

FSWS8 to FSWS0	Start block number of the flash memory shield area
—	Start block number

33.6.2.16 Data flash control register (DFLCTL)

The DFLCTL register enables or disables access to the data flash memory area. The DFLCTL register can be set by a 1-bit or 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 33 - 24 Format of Data Flash Control Register (DFLCTL)

Address: F0090H
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	<0>
DFLCTL	0	0	0	0	0	0	0	DFLEN

DFLEN	Data flash memory area access control
0	Access to the data flash memory area is disabled.
1	Access to the data flash memory area is enabled.

33.6.2.17 Interrupt vector jump enable register (VECTCTRL)

The VECTCTRL register specifies the destination address of the jump in response to any interrupt which occurs during self-programming. The VECTCTRL register can be set by an 8-bit memory manipulation instruction. The value of this register following a reset is 00H.

Figure 33 - 25 Format of Interrupt Vector Jump Enable Register (VECTCTRL)

Address: F00FFH
After reset: 00H
R/W: R/W

Symbol	7	6	5	4	3	2	1	0
VECTCTRL	0	0	0	0	0	0	0	VECTCTRL

VECTCTRL	Setting the interrupt branch destinations
0	Interrupt vector addresses in the ROM
1	Specified addresses in the RAM[Note]

Note A destination address in the RAM is specified by the FLSIVC1 and FLSIVC0 registers. For details, see **33.6.2.18 Interrupt vector change registers 0 and 1 (FLSIVC0, FLSIVC1)**.

33.6.2.18 Interrupt vector change registers 0 and 1 (FLSIVC0, FLSIVC1)

The FLSIVC0 and FLSIVC1 registers specify the destination address of the jump in response to any interrupt which occurs during self-programming. For details on how to handle an interrupt during self-programming, see **33.6.7 Interrupts in code flash memory programming mode**. The FLSIVC0 and FLSIVC1 registers can be set by a 16-bit memory manipulation instruction. The values of the FLSIVC0 and FLSIVC1 registers following a reset are 0000H and 000FH, respectively.

Figure 33 - 26 Format of Interrupt Vector Change Registers 0 and 1 (FLSIVC0, FLSIVC1)

Address: F0480H, F0481H
After reset: 0000H
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
FLSIVC0	FLSIV15	FLSIV14	FLSIV13	FLSIV12	FLSIV11	FLSIV10	FLSIV9	FLSIV8

7	6	5	4	3	2	1	0
FLSIV7	FLSIV6	FLSIV5	FLSIV4	FLSIV3	FLSIV2	FLSIV1	FLSIV0

Address: F0482H, F0483H
After reset: 000FH
R/W: R/W

Symbol	15	14	13	12	11	10	9	8
FLSIVC1	0	0	0	0	0	0	0	0

7	6	5	4	3	2	1	0
0	0	0	0	FLSIV19	FLSIV18	FLSIV17	FLSIV16

Caution Set the values of the 4 higher-order bits and 16 lower-order bits of the branch destination address in the FLSIVC1 and FLSIVC0 registers, respectively.

33.6.3 Setting the flash memory control mode

The flash memory has the following flash memory control modes.

- Code flash memory programming mode
  The code flash memory area and the extra area can be rewritten.
- Data flash memory programming mode
  The data flash memory area can be rewritten.
- Non-programmable mode
  The flash memory (code flash memory area, data flash memory area, and extra area) cannot be rewritten.

To rewrite the flash memory, set the flash memory control mode to code flash memory programming mode or data flash memory programming mode. Setting each of the flash memory control modes requires executing the specific sequence for setting the flash protect command register (PFCMD) and flash programming mode control register (FLPMC).

Caution 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).

33.6.3.1 Procedure for executing the specific sequence

Writing the required values to the flash protect command register (PFCMD) and the flash programming mode control register (FLPMC) 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.

33.6.3.2 Procedure for entry to the code flash memory programming mode

The procedure for entry to the code flash memory programming mode is given below.

<1> Write A5H to the PFCMD register.
<2> Write 02H to the FLPMC register (EEEMD = 0, FWEDIS = 0, FLSPM = 1).
<3> Write FDH to the FLPMC register (inverse of 02H).
<4> Write 02H to the FLPMC register (EEEMD = 0, FWEDIS = 0, FLSPM = 1).

33.6.3.3 Procedure for entry to the data flash memory programming mode

The procedure for entry to the data flash memory programming mode is given below.

<1> Write A5H to the PFCMD register.
<2> Write 10H to the FLPMC register (EEEMD = 1, FWEDIS = 0, FLSPM = 0).
<3> Write EFH to the FLPMC register (inverse of 10H).
<4> Write 10H to the FLPMC register (EEEMD = 1, FWEDIS = 0, FLSPM = 0).

33.6.3.4 Procedure for entry to the non-programmable mode

After executing the procedure for entry to the non-programmable mode from the code flash memory programming mode or data flash memory programming mode and waiting**Note**, reading from the target flash memory for the programming mode before the mode transition becomes possible.

Note Wait time: 10 μs

<When the interrupt vector has not been changed to addresses in the RAM>

The procedure for the transition in cases where branch destinations in response to interrupts are interrupt vector addresses in the ROM is given below.

<1> Write A5H to the PFCMD register.
<2> Write 08H to the FLPMC register (EEEMD = 0, FWEDIS = 1, FLSPM = 0).
<3> Write F7H to the FLPMC register (inverse of 08H).
<4> Write 08H to the FLPMC register (EEEMD = 0, FWEDIS = 1, FLSPM = 0).
<5> Reading from the target flash memory area becomes possible after waiting for 10 μs.

<When the interrupt vectors have been changed to addresses in the RAM>

The procedure for the transition in cases where branch destinations in response to interrupts are changed to specified addresses in the RAM is given below.

<1> Write A5H to the PFCMD register.
<2> Write 00H to the FLPMC register (EEEMD = 0, FWEDIS = 0, FLSPM = 0).
<3> Write FFH to the FLPMC register (inverse of 00H).
<4> Write 00H to the FLPMC register (EEEMD = 0, FWEDIS = 0, FLSPM = 0).
<5> Reading from the target flash memory area becomes possible after waiting for 10 μs.

33.6.4 Initializing the registers for use with the flash memory sequencer

The following registers can be initialized 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 given below.

<1> Set the FLRST bit.
<2> Use software code to wait for at least one cycle of the CPU clock.
<3> Clear the FLRST bit.

33.6.5 Setting the operating frequency of the flash memory sequencer

Set the value corresponding to the operating frequency of the CPU (1 MHz to 32 MHz) in the FSET4 to FSET0 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. For example, when the CPU operating frequency is 4.5 MHz, set the bits to 00100B for 5 MHz.

How to set the operating frequency of the flash memory sequencer is described below.

<1> Enter the code flash memory programming mode or data flash memory programming mode. For the procedures for entry to each of these flash memory programming modes, see **33.6.3.1 Procedure for executing the specific sequence**, **33.6.3.2 Procedure for entry to the code flash memory programming mode**, and **33.6.3.3 Procedure for entry to the data flash memory 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 FSET4 to FSET0 bits to the value corresponding to the CPU operating frequency, respectively.

Caution When using the code/data flash memory area sequencer and the extra area sequencer to rewrite the code or data flash memory or the extra area, set the value corresponding to the CPU operating frequency in the FSET4 to FSET0 bits of the FSSET register before doing so. 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.

33.6.6 Rewriting the flash memory

33.6.6.1 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 sequencer.

33.6.6.2 Selecting the area to be rewritten

Select the code flash memory area, data flash memory area, or extra area for rewriting by using the flash area selection register (FLARS).

33.6.6.3 Commands for use with the code/data flash memory area sequencer

Use the commands exclusively used for the code/data flash memory area sequencer to rewrite the code or data flash memory area. To execute a command, set the target command in the SQMD2 to SQMD0 bits of the flash memory sequencer control register (FSSQ) and set the SQST bit to 1. The SQMD2 to SQMD0 bits and the SQST bit can be set simultaneously. For the commands exclusively for use with the code/data flash memory area sequencer, see **33.6.2.9 Flash memory sequencer control register (FSSQ)**.

33.6.6.4 Operations for rewriting the code flash memory area

To rewrite the code flash memory area, 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. Allocate the processing software to rewrite the code flash memory area in the RAM and execute it from the RAM.

Units for block erasure and for writing in rewriting of the code flash memory area

- Unit for blocks to be erased: 2 Kbytes
- Unit for writing: 4 bytes

<Handling the commands>

<1> Enter the code flash memory programming mode. For the procedure for entry to the code flash memory programming mode, see **33.6.3.1 Procedure for executing the specific sequence** and **33.6.3.2 Procedure for entry to the code flash memory programming mode**.

<2> Set the EXA bit of the FLARS register to 0 (code/data flash memory areas).

<3> Before executing each command, set the address data, write data, and command in the corresponding registers.

- Block erasure

  Set the block start address[Note 1] (example: 002000H) of the code flash memory to be erased in the FLAPH and FLAPL registers. Set the block end address[Note 1] (example: 0027FFH) of the code flash memory to be erased in the FLSEDH and FLSEDL registers.
- Writing

  Set the start address[Note 2] (example: 002000H) of the flash memory to be written in the FLAPH and FLAPL registers. Set 4-byte write data in the FLWH and FLWL registers.
- Blank checking

  Set the start address[Note 2] (example: 002000H) of the flash memory to be blank-checked in the FLAPH and FLAPL registers. Set the end address (example: 0027FFH) of the flash memory to be blank-checked in the FLSEDH and FLSEDL registers. 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.

<4> When the value of the command to be executed is set in the MDCH and SQMD2 to SQMD0 bits of the FSSQ register and the SQST bit is set to 1, the code/data flash memory area sequencer executes the specified command. The MDCH, SQST, and SQMD2 to SQMD0 bits can be set simultaneously. If these bits are set simultaneously, the FSSQ register is set to the following values.

- Block erasure: 84H
- Writing: 81H
- Blank checking of the code flash memory area: 83H

<5> Wait until the command for use with the code/data flash memory area sequencer is complete. 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 **33.6.6.9 Procedures for checking completion of the commands for use with the flash memory sequencer in the respective areas**.

<6> Processing after executing a command

<Continuing command processing>

The same command or another command can be executed by continuously updating the address data and write data in step 3 with the state remaining in code flash memory programming mode.

<Completing command processing>

Switch to the non-programmable mode. For the procedure for switching to the non-programmable mode, see **33.6.3.1 Procedure for executing the specific sequence** and **33.6.3.4 Procedure for entry to the non-programmable mode**.

(Notes are listed on the next page.)

Note 1. The code flash memory area blocks can be erased in units of 2 Kbytes. Specify the erase addresses (start address and end address) so that all blocks to be erased are included. For the relationship between addresses and block numbers, see **Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory**.

Note 2. The code flash memory area can be written and blank-checked in units of 4 bytes. Therefore, set the two lower-order bits of the FLAPL register for specifying the address to 00B (an integer of 4).

33.6.6.5 Operations for rewriting the data flash memory area

To rewrite the data flash memory area, 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 blocks to be erased: 256 bytes
- Unit for writing: 1 byte

<Handling the commands>

<1> Enter the data flash memory programming mode. For the procedure for entry to the data flash memory programming mode, see **33.6.3.1 Procedure for executing the specific sequence** and **33.6.3.3 Procedure for entry to the data flash memory programming mode**.

<2> Set the EXA bit of the FLARS register to 0 (code/data flash memory areas).

<3> Before executing each command, set the address data, write data, and command in the corresponding registers.

- Block erasure
  Set the block start address[Note] (example: 0F1100H) of the data flash memory to be erased in the FLAPH and FLAPL registers. Set the block end address[Note] (example: 0F11FFH) of the data flash memory to be erased in the FLSEDH and FLSEDL registers.
- Writing
  Set the start address (example: 0F1101H) of the flash memory to be written in the FLAPH and FLAPL registers. Set the write data in the 8 lower-order bits of the FLWL register.
- Blank checking:
  Set the start address (example: 0F1100H) of the flash memory to be blank-checked in the FLAPH and FLAPL registers. Set the end address (example: 0F11FFH) of the flash memory to be blank-checked in the FLSEDH and FLSEDL registers. 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.

<4> When the value of the command to be executed is set in the MDCH and SQMD2 to SQMD0 bits of the FSSQ register and the SQST bit is set to 1, the code/data flash memory area sequencer executes the specified command. The MDCH, SQST, and SQMD2 to SQMD0 bits can be set simultaneously. If these bits are set simultaneously, the FSSQ register is set to the following values.

- Block erasure: 84H
- Writing: 81H
- Blank checking of the data flash memory area: 8BH

<5> Wait until the command for use with the code/data flash memory area sequencer is complete. 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 **33.6.6.9 Procedures for checking completion of the commands for use with the flash memory sequencer in the respective areas**.

<6> Processing after executing a command

<Continuing command processing>

The same command or another command can be executed by continuously updating the address data and write data in step 3 with the state remaining in data flash memory programming mode.

<Completing command processing>

Switch to the non-programmable mode. For the procedure for switching to the non-programmable mode, see **33.6.3.1 Procedure for executing the specific sequence** and **33.6.3.4 Procedure for entry to the non-programmable mode**.

(Note is listed on the next page.)

Note The data flash memory area blocks can be erased in units of 256 bytes. Therefore, set the 8 lower-order bits of the FLAPL register for specifying the start address to 0000 0000B (an integer of 256). Also set the 8 lower-order bits of the FLSEDL register for specifying the end address to 1111 1111B.

33.6.6.6 Commands for use with the extra area sequencer

Use the commands for use with the extra area sequencer to rewrite the settings of the flash shield window, flash read protection, flash memory security, and boot swap function in the extra area. To execute a command, set the command to be executed in the ESQMD3 to ESQMD0 bits of the flash extra area sequencer control register (FSSE), and then set the ESQST bit to 1. The ESQMD3 to ESQMD0 bits and the ESQST bit can be set simultaneously. Allocate the extra area sequencer command processing software in the RAM and execute it from the RAM.

33.6.6.7 Operations for rewriting the extra area

To rewrite the extra area, enter the code flash memory programming mode and then execute commands for use with the extra area sequencer. Before starting to execute a command, set the data required for executing each command in the corresponding registers.

<Handling the commands>

<1> Enter the code flash memory programming mode. For the procedure for entry to the code flash memory programming mode, see **33.6.3.1 Procedure for executing the specific sequence** and **33.6.3.2 Procedure for entry to the code flash memory programming mode**.

<2> Set the EXA bit of the FLARS register to 1 (extra area).

<3> Before executing a command, set 4-byte data in the FLWH and FLWL registers. Each bit of the combined FLW31 to FLW0 bits of the FLWH and FLWL registers corresponds to EX bits 31 to 0 of the target extra area data. For details on data to be set for each command, see **33.6.6.8 Data to be set for the commands for use with the extra area sequencer**.

<4> When the value of the command to be executed is set in the ESQMD3 to ESQMD0 bits of the FSSE register and the ESQST bit is set to 1, the extra area sequencer executes the specified command. The ESQMD3 to ESQMD0 bits and the ESQST bit can be set simultaneously. If these bits are set simultaneously, the FSSE register is set to the following values.

- Write data to the flash shield window setting area: 81H
- Write data to the flash read protection setting area: 86H
- Write data to the security flag and boot swap function setting area: 87H

<5> Wait until the command for use with the extra area sequencer is complete. 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 extra area sequencer” in **33.6.6.9 Procedures for checking completion of the commands for use with the flash memory sequencer in the respective areas**.

<6> Processing after executing a command

<Continuing command processing>

The same command or another command can be executed by continuously updating the FLWH and FLWL register data to be set in the extra area in step 3 with the state remaining in code flash memory programming mode.

<Completing command processing>

Switch to the non-programmable mode. For the procedure for switching to the non-programmable mode, see **33.6.3.1 Procedure for executing the specific sequence** and **33.6.3.4 Procedure for entry to the non-programmable mode**.

33.6.6.8 Data to be set for the commands for use with the extra area sequencer

Writing to the extra area proceeds per 4 bytes. Each command for use with the extra area sequencer writes the data set in the combined FLW31 to FLW0 bits of the FLWH and FLWL registers to EX bits 31 to 0 in the extra area corresponding to the given command.

1. Write to the flash shield window setting area
   Set the data in the FLWH and FLWL registers to the flash shield window setting area.

EX bit 31	EX bit 30	EX bit 29	EX bit 28	EX bit 27	EX bit 26	EX bit 25	EX bit 24
FSWC	1	1	1	1	1	1	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
FSPR	1	1	1	1	1	1	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

Bit Name	Setting
FSWC	Specifies the range of the flash memory shield area. 0: Flash memory shield area: Inside the window range 1: Flash memory shield area: Outside the window range (default)
FSPR	Specifies whether to enable or disable changing of the flash shield window setting area. 0: Changing of the flash shield window setting area is disabled. 1: Changing of the flash shield window setting area is enabled (default).
FSWE8 to FSWE0	Flash shield window end block setting area Specify the block number (end block number + 1).**Note**
FSWS8 to FSWS0	Flash shield window start block setting area Specify the start block number.**Note**

Note For the relationship between addresses and block numbers, see **Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory**.

Caution **The value of the FSPR bit can be changed from 0 (disabling) to 1 (enabling) by executing the chip erase command of the dedicated flash memory programmer or the Security Release command for use in serial programming mode with the entire flash memory erased. Note that if either of the disabling settings listed below is made, executing the chip erase command of the dedicated flash memory programmer or the Security Release command for use in serial programming mode is not possible.**

- **SEPR = 0 (block erasure is disabled)**
- **BTPR = 0 (rewriting of the boot area is disabled)**

Moreover, setting the FSPR bit to 1 (enabling) is not possible because a command cannot be transmitted when connection in serial programming mode is not available due to the setting for disabling connection to the programmer and on-chip debugger or for enabling programmer connection ID authentication having been made.

2. Write to the flash read protection setting area
   Set the data in the FLWH and FLWL registers to the flash read protection setting area.

EX bit 31	EX bit 30	EX bit 29	EX bit 28	EX bit 27	EX bit 26	EX bit 25	EX bit 24
SWPR	1	1	1	1	1	1	UPAddr8

EX bit 23	EX bit 22	EX bit 21	EX bit 20	EX bit 19	EX bit 18	EX bit 17	EX bit 16
UPAddr7	UPAddr6	UPAddr5	UPAddr4	UPAddr3	UPAddr2	UPAddr1	UPAddr0

EX bit 15	EX bit 14	EX bit 13	EX bit 12	EX bit 11	EX bit 10	EX bit 9	EX bit 8
1	1	1	1	1	1	1	LOWAddr8

EX bit 7	EX bit 6	EX bit 5	EX bit 4	EX bit 3	EX bit 2	EX bit 1	EX bit 0
LOWAddr7	LOWAddr6	LOWAddr5	LOWAddr4	LOWAddr3	LOWAddr2	LOWAddr1	LOWAddr0

Bit Name	Setting
SWPR	Specifies whether to enable or disable changing of the flash read protection setting area. 0: Changing of the flash read protection setting area is disabled. 1: Changing of the flash read protection setting area is enabled (default).
UPAddr8 to UPAddr0	Flash read protection end block setting area Specify the end block number.**Note**
LOWAddr8 to LOWAddr0	Flash read protection start block setting area Specify the start block number.**Note**

Note For the relationship between addresses and block numbers, see **Table 3 - 1 Correspondence between Addresses and Block Numbers in Flash Memory**. The flash read protection setting area cannot be read after a reset is released.

Caution **The value of the SWPR bit can be changed from 0 (disabling) to 1 (enabling) by executing the chip erase command of the dedicated flash memory programmer or the Security Release command for use in serial programming mode with the entire flash memory erased. Note that if either of the disabling settings listed below is made, executing the chip erase command of the dedicated flash memory programmer or the Security Release command for use in serial programming mode is not possible.**
- **SEPR = 0 (block erasure is disabled)**
- **BTPR = 0 (rewriting of the boot area is disabled)**

Moreover, setting the SWPR bit to 1 (enabling) is not possible because a command cannot be transmitted when connection in serial programming mode is not available due to the setting for disabling connection to the programmer and on-chip debugger or for enabling programmer connection ID authentication having been made.

3. Write to the security flag and boot swap function setting area
   Set the data in the FLWH and FLWL registers to the security flag and boot swap function setting area. For details of the security settings, see **33.9 Security Settings**.

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
1	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	IFPR	1	IDEN

Bit Name	Setting
WRPR	Specifies whether to enable or disable writing in serial programming mode. Writing in serial programming mode is disabled. 0: Writing in serial programming mode is disabled. 1: Writing in serial programming mode is enabled (default).
SEPR	Specifies whether to enable or disable block erasure. Block erasure in serial programming mode is disabled. 0: Block erasure in serial programming mode is disabled. 1: Block erasure in serial programming mode is enabled (default).
BTPR	Specifies whether to enable or disable rewriting of the boot area. Rewriting of the boot area and boot swapping are disabled. 0: Rewriting of the boot area and boot swapping are disabled. 1: Rewriting of the boot area and boot swapping are enabled (default).
BTFLG	Specifies the boot area when the TMSPMD bit in the FSSET register is 0. 0: Boot area: Boot cluster 1 1: Boot area: Boot cluster 0 (default)
IFPR	Specifies whether to enable or disable connection to the programmer and on-chip debugger. The serial programming mode and connection to the on-chip debugger are disabled. 0: Serial programming mode and connection to the on-chip debugger are disabled. 1: Serial programming mode and connection to the on-chip debugger are enabled (default).
IDEN	Specifies whether to enable or disable programmer connection ID authentication. ID authentication for connection in serial programming mode proceeds. 0: ID authentication for connection in serial programming mode is enabled. 1: ID authentication for connection in serial programming mode is disabled (default).

(Cautions are listed on the next page.)

Caution 1. When changing the value of the BTFLG bit, set all other bits to 1.

Caution 2. When changing the values of security flags other than BTFLG to 0 (disabling), read the register first and set the BTFLG bit to the same value as was read, and set the other bits to 1.

Caution 3. The value of the WRPR bit can be changed from 0 (disabling) to 1 (enabling) by executing the chip erase command of the dedicated flash memory programmer or the Security Release command for use in serial programming mode with the entire flash memory erased. Note that if either of the disabling settings listed below is made, executing the chip erase command of the dedicated flash memory programmer or the Security Release command for use in serial programming mode is not possible.

- **SEPR = 0 (block erasure is disabled)**
- **BTPR = 0 (rewriting of the boot area is disabled)**

Moreover, setting the WRPR bit to 1 (enabling) is not possible because a command cannot be transmitted when connection in serial programming mode is not available due to the setting for disabling connection to the programmer and on-chip debugger or for enabling programmer connection ID authentication having been made.

Caution 4. Restoring the value of any among the SEPR, BTPR, IFPR, and IDEN bits to 1 after having set it to 0 is not possible.

33.6.6.9 Procedures for checking completion of the commands for use with the flash memory sequencer in the respective areas

Perform the following procedure for checking the completion of commands when terminating an activated command for use with the code/data flash memory area sequencer and the extra area sequencer.

- Procedure for checking the completion of commands for use with the code/data flash memory area sequencer
  - <1> Activate a command for use with the code/data flash memory area sequencer, and then wait until the SQEND flag of the FSASTH register is set to 1.
  - <2> Confirm that the SQEND flag of the FSASTH register is set to 1, and then clear the SQST bit of the FSSQ register.
  - <3> Wait until the SQEND flag of the FSASTH register is cleared. When the SQEND flag is cleared, the command is completed and the sequencer stops.

- Procedure for checking the completion of commands for use with the extra area sequencer
  - <1> Activate a command for use with the extra area sequencer, and then wait until the ESQEND flag of the FSASTH register is set to 1.
  - <2> Confirm that the ESQEND flag of the FSASTH register is set to 1, and then clear the ESQST bit of the FSSE register.
  - <3> Wait until the ESQEND flag of the FSASTH register is cleared. When the ESQEND flag is cleared, the command is completed and the sequencer stops.

33.6.6.10 Procedure for forcibly terminating a command for use with the code/data flash memory area sequencer

A command for use with the code/data flash memory area sequencer can be forcibly terminated while it is in progress. Note that no command for use with the extra area sequencer can be forcibly terminated while it is in progress.

<Procedure for forcible termination>

<1> Set the FSSTP bit of the FSSQ register to 1 after a command for use with the code/data flash memory area sequencer is activated until the SQST bit of the FSSQ register in <2> is cleared.

<2> Confirm that the SQEND flag of the FSASTH register is set to 1, and then clear the SQST and FSSTP bits of the FSSQ register.

<3> Wait until the SQEND flag of the FSASTH register is automatically cleared. When the SQEND flag is cleared, the forcible termination is completed.

33.6.7 Interrupts in code flash memory programming mode

33.6.7.1 Overview

When an interrupt occurs, reference to the interrupt vector in the ROM proceeds, and execution branches to the interrupt processing that is allocated in the up to 64-Kbyte ROM space in accord with the address in the interrupt vector (16 bits), from which the interrupt processing is executed. However, in code flash memory programming mode, where the code flash memory and extra area can be rewritten, referring to the ROM is not possible. As a result, the interrupt processing cannot be executed in this way. Note that even when referring to the ROM is not possible, the interrupt processing can be executed by changing the branch destinations of the interrupts. Specifically, changing the branch destinations of all interrupts to specified addresses in the RAM enables the execution of interrupt processing from the RAM instead of by using the interrupt vector and interrupt processing in the ROM.

33.6.7.2 Operation to change the branch destinations of the interrupts

To change the branch destinations of the interrupts, set the interrupt vector change registers 1 and 0 (FLSIVC1 and FLSIVC0) and interrupt vector jump enable register (VECTCTRL) so that execution branches to an address in the RAM following an interrupt. This enables execution of the interrupt processing from the RAM without reference to the interrupt vector in the ROM, even when an interrupt occurs in code flash memory programming mode. The FLSIVC1 and FLSIVC0 registers are used to specify the branch destination address in response to interrupts which occur during rewriting of the code flash memory or extra area. Set the values of the 16 lower-order bits and 4 higher-order bits of the branch destination address in the FLSIVC0 and FLSIVC1 registers, respectively.

Settings for control of the branch destination of interrupts that occur during self-programming are as follows.

- For branching to vector addresses in the ROM: VECTCTRL = 0 or FWEDIS of FLPMC = 1
- For branching to a RAM address: VECTCTRL = 1 (with FWEDIS of FLPMC = 0)

Caution 1. The user must determine the type of interrupt by checking the interrupt flags. Therefore, the interrupt flags are not automatically cleared after the VECTCTRL register has been set.

Caution 2. A changed interrupt branch destination in the ROM is not specifiable.

Caution 3. The change to the interrupt branch destination made by the VECTCTRL register is only effective during self-programming.

Caution 4. Disable interrupts while changing the interrupt branch destination to an address in the RAM.

33.6.7.3 Operation to change the interrupt branch destination

To specify interrupt processing from the RAM, update the FLSIVC1, FLSIVC0, and VECTCTRL registers while the value of the FWEDIS bit of the flash programming mode control register (FLPMC) is 0. Execute the specific sequence to handle the FWEDIS bit of the FLPMC register, and set the FLSIVC1, FLSIVC0, and VECTCTRL registers. The interrupt branch destination is thus changed to an address in RAM.

<When changing the interrupt branch destinations to an address in RAM >

The following describes operation for changing all interrupt branch destinations to a specified address in the RAM.

<1> Save the setting for enabling or disabling interrupts that was in place before starting this procedure and make the setting to disable interrupts.

<2> Execute the specific sequence and set the FWEDIS bit of the FLPMC register to 0.

1. Write A5H to the PFCMD register.
2. Write 00H to the FLPMC register (EEEMD = 0, FWEDIS = 0, FLSPM = 0).
3. Write FFH to the FLPMC register (inverse of 00H).
4. Write 00H to the FLPMC register (EEEMD = 0, FWEDIS = 0, FLSPM = 0).

<3> Specify an address in RAM in the FLSIVC1 and FLSIVC0 registers.

<4> Set the VECTCTRL register to 01H to set the RAM address as the branch destination for interrupts.

<5> Restore the saved setting for enabling or disabling interrupts.

Caution 1. Retain the value 0 in the FWEDIS bit as long as interrupt processing from the RAM remains specified.

Caution 2. Do not set the interrupt branch destination in the saddr space at addresses from FFE20H to FFEFFH.

Caution 3. When instructions are to be executed from a RAM area and RAM parity error resets are enabled (RPERDIS = 0), initialize the RAM area where such access is to proceed and an additional 10 bytes beyond the area.

<When returning the interrupt branch destinations from the RAM address to the vectors in the ROM>

The following describes operation for returning the interrupt branch destinations to the addresses indicated by the interrupt vectors in the ROM (default state).

<1> Save the setting for enabling or disabling interrupts that was in place before starting this procedure and make the setting to disable interrupts.

<2> Execute the specific sequence and set the FWEDIS bit of the FLPMC register to 1.

1. Write A5H to the PFCMD register.
2. Write 08H to the FLPMC register (EEEMD = 0, FWEDIS = 1, FLSPM = 0).
3. Write F7H to the FLPMC register (inverse of 08H).
4. Write 08H to the FLPMC register (EEEMD = 0, FWEDIS = 1, FLSPM = 0).

<3> Set the VECTCTRL register to 00H to set the vector addresses in the ROM as the interrupt branch destinations.

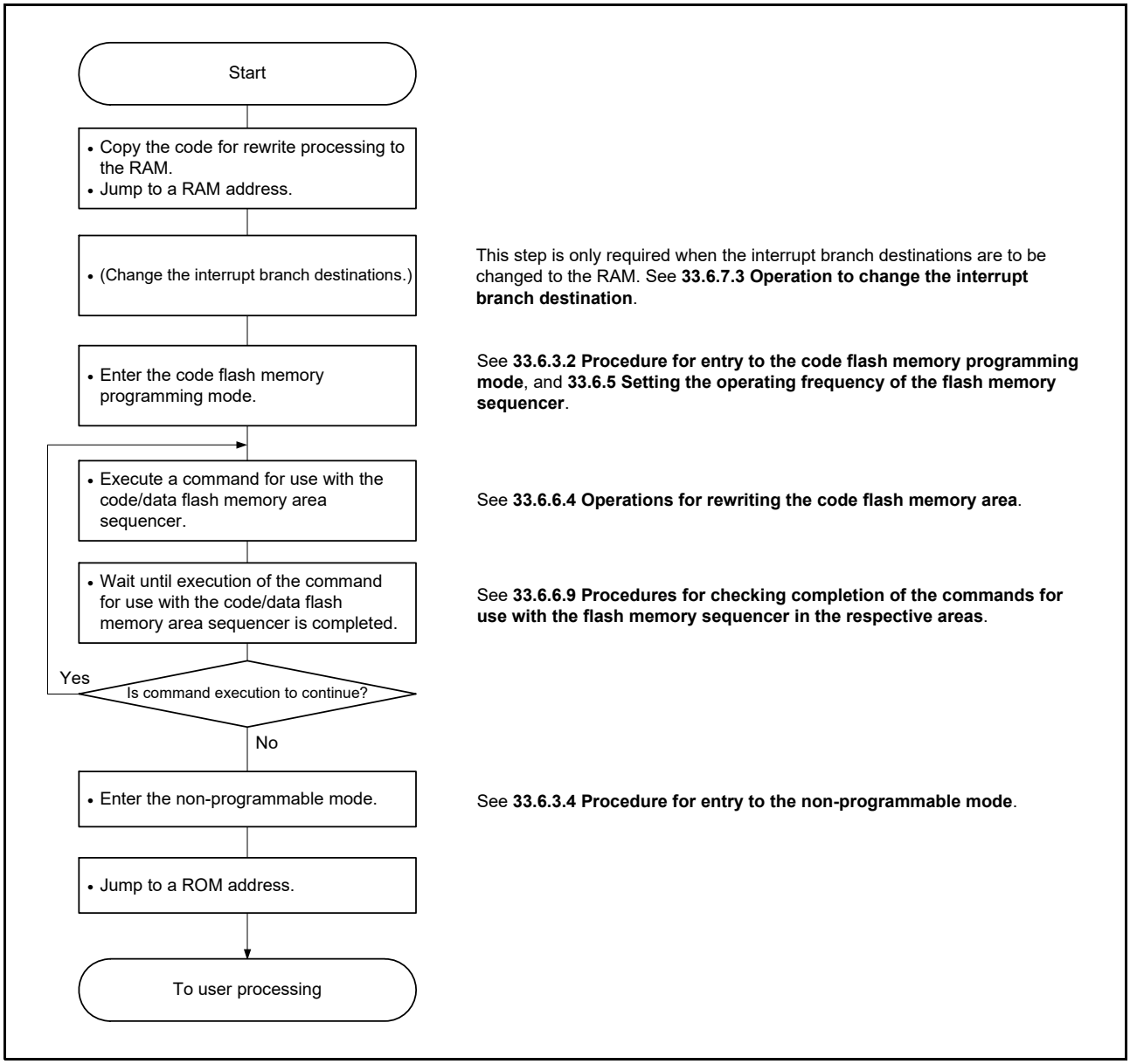
<4> Restore the saved setting for enabling or disabling interrupts.

33.6.8 Example of executing the commands to rewrite the flash memory areas

33.6.8.1 Example of executing the commands to rewrite the code flash memory area

Figure 33 - 27 shows the flow of executing the commands to rewrite the code flash memory area.

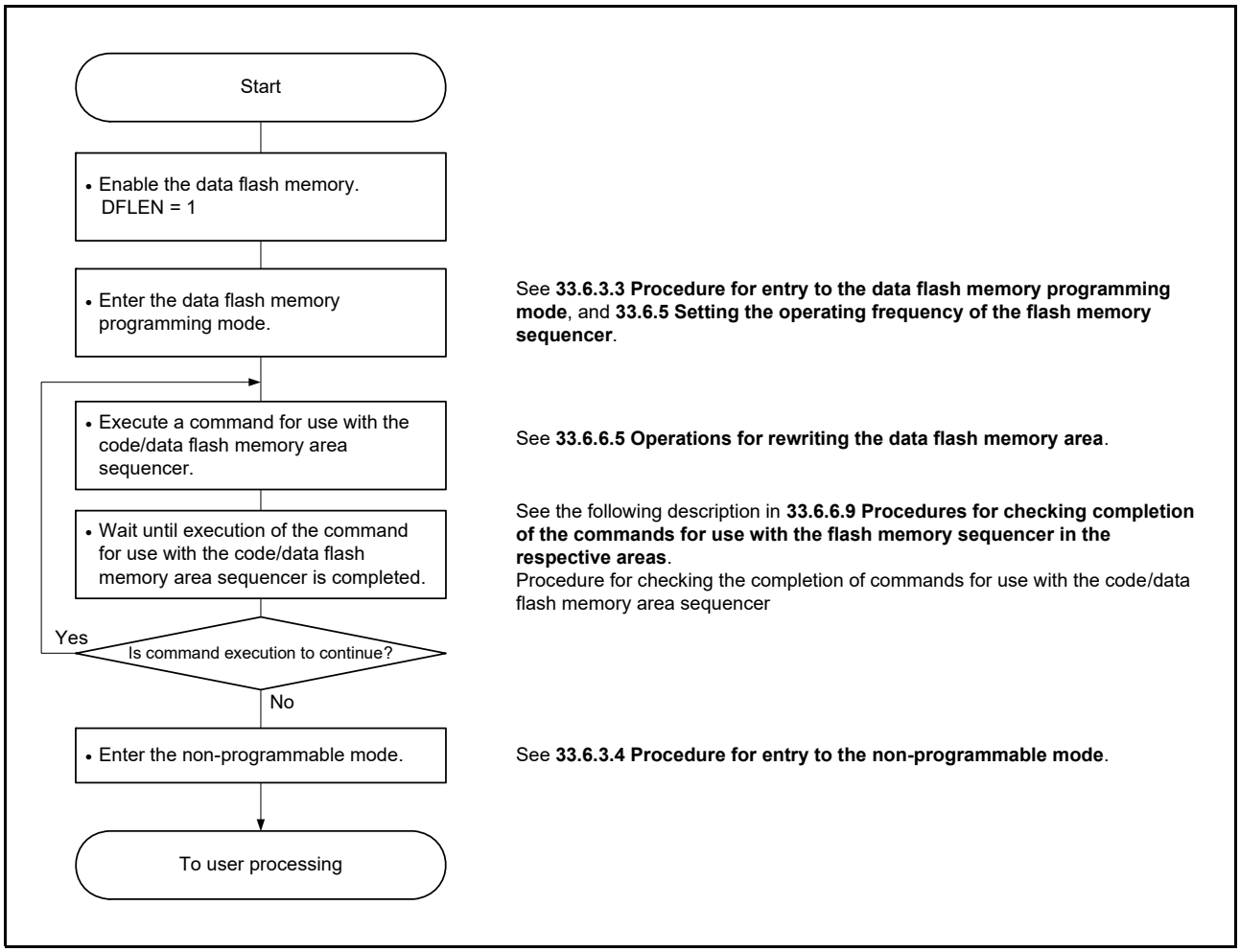
Figure 33 - 27 Flow of Executing the Commands to Rewrite the Code Flash Memory Area



33.6.8.2 Example of executing the commands to rewrite the data flash memory area

Figure 33 - 28 shows the flow of executing the commands to rewrite the data flash memory area.

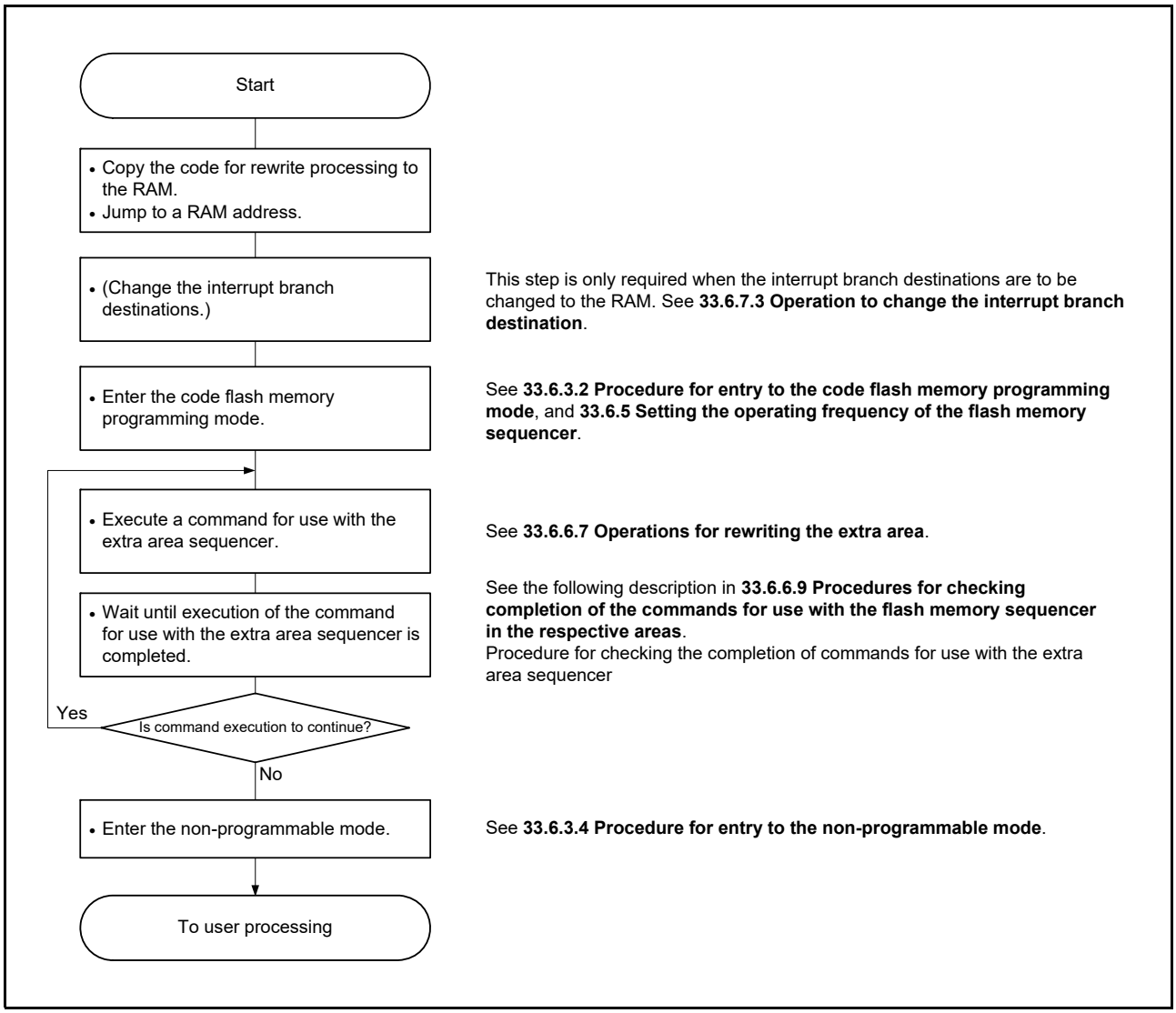
Figure 33 - 28 Flow of Executing the Commands to Rewrite the Data Flash Memory Area



33.6.8.3 Example of executing the commands to rewrite the extra area

Figure 33 - 29 shows the flow of executing the commands to rewrite the extra area.

Figure 33 - 29 Flow of Executing the Commands to Rewrite the Extra Area



33.6.9 Notes on self-programming

1. Rewriting the code flash memory or extra area
   To rewrite the code flash memory or extra area, place the code or values in the RAM.

2. 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).

3. 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 memory programming mode, reference to the code flash memory is not possible. Accordingly, in code flash memory 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 memory programming mode, reference to the data flash memory is not possible. Accordingly, in data flash memory programming mode, copy data that are for reference to the RAM in advance so that reference to the data in the RAM is possible.

4. Specifying the range of unavailable area
   Specify the range of blank checking and block erasure within the range of code flash memory area or data flash memory area. Do not specify any unavailable area or both the code flash memory area and data flash memory area including an unavailable area.

5. Specifying the range of code flash memory area
   For products that have a code flash memory area larger than 512 Kbytes, specifying a range that spans addresses or blocks across the 512-Kbyte boundary for blank checking or block erasure is not possible. Specify an address range from 00000H to 7FFFFH or 80000H to BFFFFH for blank checking. Specify a block range from 000H to 099H or 100H to 17FH for block erasure.

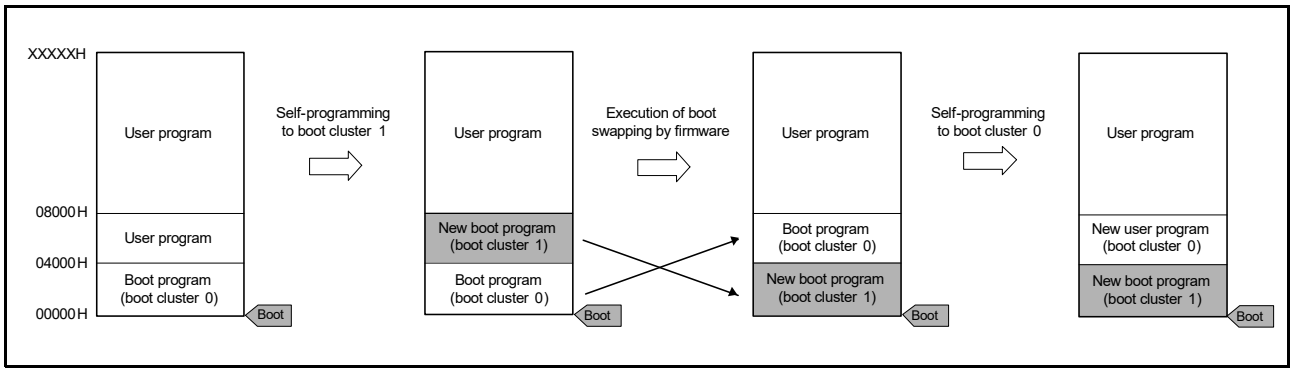
33.7 Boot Swap Function

The boot area consists of the vector table area, CALLT table area, option bytes area, setting areas for the on-chip debug security ID and programmer connection ID, and program area. The required settings and information respectively for starting the program and for connecting an on-chip debugger and programmer are stored in these areas. Thus, if rewriting in the boot area through self-programming 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. When the boot area is set to boot cluster 0, write a new boot program to boot cluster 1 through self-programming before erasing boot cluster 0**Note**. After the boot program has been successfully written to boot cluster 1, change the boot area from boot cluster 0 to boot cluster 1 by self-programming to use boot cluster 1 as the boot area. After that, erase the data in boot cluster 0, and then write data to 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 when the program is reset and started next. A boot cluster is a 16-Kbyte area.

Note Allocate the addresses in the new boot program as follows.

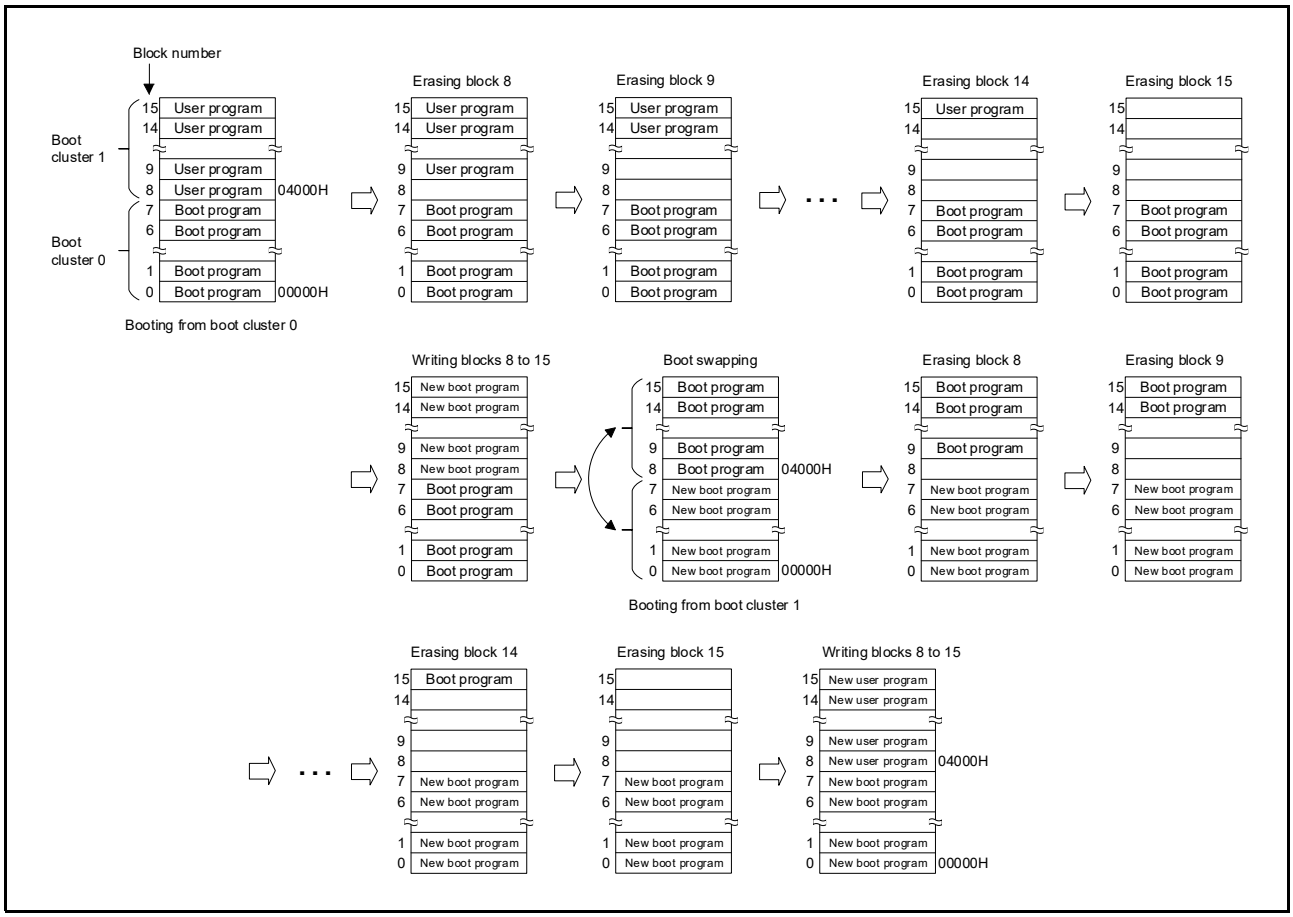
- 04000H to 0407FH (128 bytes): Vector table area
- 04080H to 040BFH (64 bytes): CALLT table area
- 040C0H to 040C3H (4 bytes): Option bytes area
- 040C4H to 040CDH (10 bytes): On-chip debug security ID setting area

Figure 33 - 30 Boot Swap Function



In the example of the above figure, the boot areas before and after boot swapping are as follows.
Boot cluster 0: Boot area before boot swapping
Boot cluster 1: Boot area after boot swapping

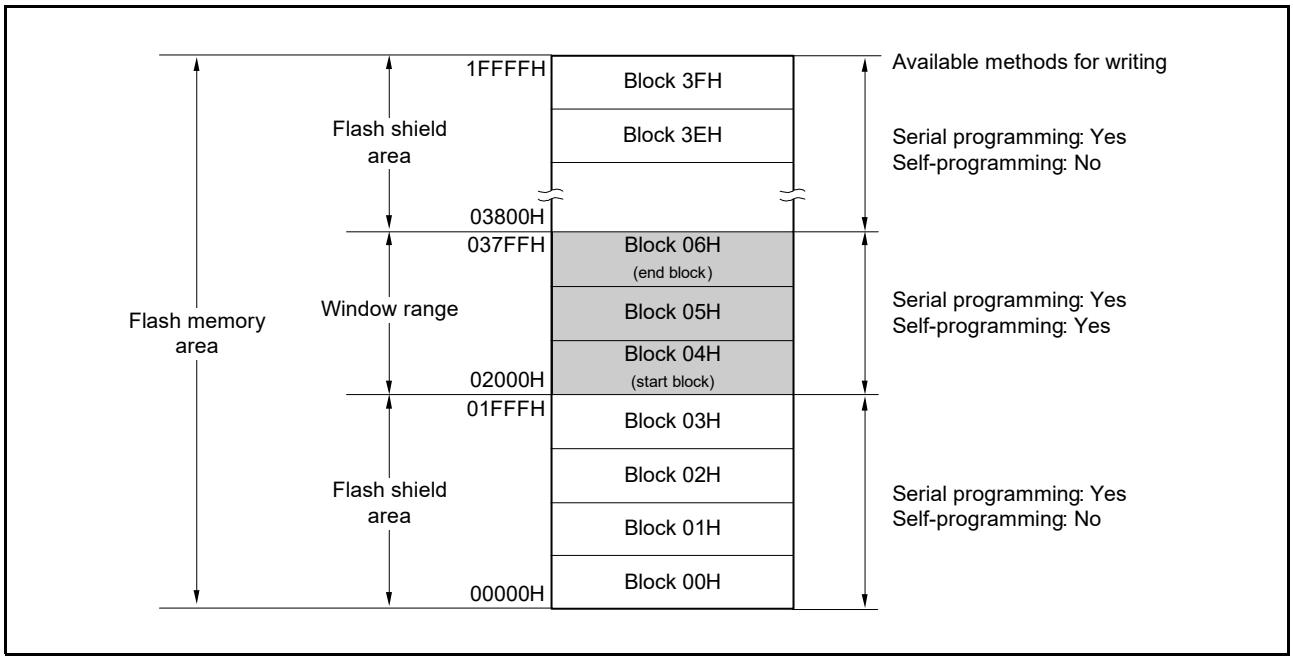
Figure 33 - 31 Example of Executing Boot Swapping



33.8 Flash Shield Window Function

The flash shield window function is provided as a security function which disables writing to and erasing of the selected flash memory shield area. This function is only effective for self-programming. The flash memory shield area is selectable as either the area inside or areas outside the range specified as the window. The window range is set by specifying the blocks where it starts and ends. The flash memory shield area can be set or changed during both serial programming and self-programming. Writing to and erasing of the flash memory shield area are disabled during self-programming. During serial programming, however, the flash memory shield area can also be written and erased.

Figure 33 - 32 Flash Shield Window Setting Example
(Target Devices: R7F100GLG, Start Block Number: 04H, End Block Number: 06H, FSWC: 1)



Caution 1. If the non-programmable area of the boot area overlaps with the flash shield window range, disabling of rewriting the boot area takes priority.

Caution 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 - 12 Relationship between Flash Shield Window Function Setting/Change Methods and Commands

Programming Condition	Window Range Setting/Change Method	Command to be Executed	
		Block erase	Write
Self-programming	Specify the window start block number and the (end block number + 1) block number (following the end block) in the flash shield window setting area using the self-programming.	Block erasure is not possible inside the flash memory shield area.	Writing is not possible inside the flash memory shield area.
Serial programming	Specify the start block number in the window range and the end block number in the window range on GUI of dedicated flash memory programmer, etc.	Block erasure is also possible inside the flash memory shield area.	Writing is also possible inside the flash memory shield area.

33.9 Security Settings

This MCU supports a security function that prohibits 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 erasure
  Execution of the block erase command for a specific block in the code flash memory and data flash memory is prohibited during serial programming. However, blocks can be erased by means of self-programming.
- Disabling writing
  Execution of the write command for entire blocks in the code flash memory and data flash memory is prohibited during serial programming. However, blocks can be written by means of self-programming. After the setting to prohibit writing has been made, releasing the setting by the Security Release command is enabled by a reset.
- Disabling rewriting the boot area
  Execution of the block erase command and write command for the boot area (00000H to 03FFFH) in the code flash memory is prohibited.
- Disabling connection to the programmer and on-chip debugger
  Connection to a dedicated flash memory programmer and on-chip debugger is prohibited. A dedicated flash memory programmer and on-chip debugger cannot be used to manipulate the flash memory.
- Enabling programmer connection ID authentication
  Authentication for an arbitrary 10-byte ID code is enabled when connecting to a dedicated flash memory programmer. The 10-byte ID area is 000C4H to 000CDH[Note]. If the ID does not match when using serial programming, the dedicated flash memory programmer cannot be used to manipulate the flash memory.

Block erasure, writing, and rewriting the boot area are enabled by the default setting at the time of shipment. Security can be set by serial programming and self-programming. Each security setting can be used in combination. **Table 33 - 13** shows the relationship between the erase and write commands when the security function of this MCU is enabled.

Note The 10-byte ID code area for the programmer connection ID is shared with the security ID code for on-chip debugging.

Caution **The security function of the dedicated flash programmer does not support self-programming.**

Remark To prohibit writing and erasure during self-programming, use the flash shield window function (see **33.8 Flash Shield Window Function** for detail).

Table 33 - 13 Relationship between Enabling the Security Function and Commands

1. During serial programming

Valid Security	Command to be Executed	
	Block Erase	Write
Prohibition of block erasure	Blocks cannot be erased.	Data can be written.Note
Prohibition of writing	Blocks can be erased.	Data cannot be written.
Prohibition of rewriting the boot area	The boot area cannot be erased.	The boot area cannot be written.
Prohibition of connection to the programmer and on-chip debugger	Blocks cannot be erased.	Data cannot be written.
Success in authentication with programmer connection ID authentication enabled	Blocks can be erased.	Data can be written.
Failure in authentication with programmer connection ID authentication enabled	Blocks cannot be erased.	Data cannot be written.

Note Confirm that no data have been written to the write area. If data in the area has not been erased, do not attempt further writing of data because data cannot be erased after the setting to prohibit block erasure has been made.

2. During self-programming

Valid Security	Command to be Executed	
	Block Erase	Write
Prohibition of block erasure	Blocks can be erased.	Data can be written.
Prohibition of writing		
Prohibition of rewriting the boot area	The boot area cannot be erased.	The boot area cannot be written.
Prohibition of connection to the programmer and on-chip debugger	Blocks can be erased.	Data can be written.
Programmer connection ID authentication enabled	Blocks can be erased.	Data can be written.

Remark To prohibit writing and erasure during self-programming, use the flash shield window function. For details, see **33.8 Flash Shield Window Function**.

Table 33 - 14 Setting Security in Each Programming Mode

1. During serial programming

Security	Security Setting	Disabling the Security Setting
Prohibition of block erasure	Set via GUI of dedicated flash memory programmer, etc.	Disabling the setting is not possible.
Prohibition of writing		Set via GUI of dedicated flash memory programmer, etc.
Prohibition of rewriting the boot area		Disabling the setting is not possible.
Prohibition of connection to the programmer and on-chip debugger		
Enabling programmer connection ID authentication		

Caution **The setting to prohibit writing can only be released when the settings to prohibit erasing blocks and rewriting the boot area are not made and the code and data flash memory areas are blank. However, if connection for serial programming is prohibited due to the setting to prohibit connection to the programmer and on-chip debugger or to enable programmer connection ID authentication, releasing the setting to prohibit writing is not possible because serial programming cannot be executed.**

2. During self-programming

Security	Security Setting	Disabling the Security Setting
Prohibition of block erasure	Set by using self-programming.	Disabling the setting is not possible.
Prohibition of writing		Disabling the setting is not possible during self-programming. Set via GUI of dedicated flash memory programmer, etc.
Prohibition of rewriting the boot area		Disabling the setting is not possible.
Prohibition of connection to the programmer and on-chip debugger		
Enabling programmer connection ID authentication		

33.10 Data Flash Memory

33.10.1 Overview of the data flash memory

An overview of the data flash memory is provided below.

- The user program can rewrite the data flash memory by self-programming. For details, refer to the **RL78 Family Renesas Flash Driver RL78 Type 01 User's Manual (R20UT4830EJ)**.
- The data flash memory can also be rewritten to through serial programming using a dedicated flash memory programmer or an external device.
- The data flash memory can be erased in 1-block (256-byte) 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 (that is, background operation (BGO) is supported).
- The data flash memory area is exclusively to be used for data, so 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 or STOP mode is prohibited while rewriting the data flash memory.

Caution 1. The data flash memory is stopped after release from a reset. The data flash control register (DFLCTL) must be set up in order to use the data flash memory.

Caution 2. The high-speed on-chip oscillator should be kept operating during rewriting of the data flash memory area. If it is stopped, it should be made to operate again (HIOSTOP = 0), and self-programming should be executed after 5 μs have elapsed.

Remark For rewriting the code flash memory via a user program, see **33.6 Self-programming**.

33.10.2 Procedure for accessing the data flash memory

The data flash memory is stopped after release from a reset. For access to this memory, follow the procedure below to make the required initial setting.

<1> Set the DFLEN bit of the data flash control register (DFLCTL) to 1.
<2> Wait for the setup time by using a software timer, etc.
The setup time depends on the operating mode of the flash memory.
<Setup time for each operating mode of the flash memory>
- HS (high-speed main) mode: 250 ns
- LS (low-speed main) mode: 250 ns
- LP (low-power main) mode: 0 ns

<3> After the wait, the data flash memory can be accessed.

Caution 1. Accessing the data flash memory is prohibited during the setup time.
Caution 2. Transition to the STOP mode is prohibited during the setup time. To enter the STOP mode during the setup time, set DFLEN to 0 and then execute the STOP instruction.
Caution 3. The high-speed on-chip oscillator should be kept operating during rewriting of the data flash memory area. If it is stopped, it should be made to operate again (CSC.HIOSTOP = 0), and self-programming should be executed after 5 μs have elapsed.
Caution 4. When switching the CPU clock between the main system clock and sub system clock, accessing the data flash memory is prohibited until the CLS bit changes after a clock is selected by the CSS bit.

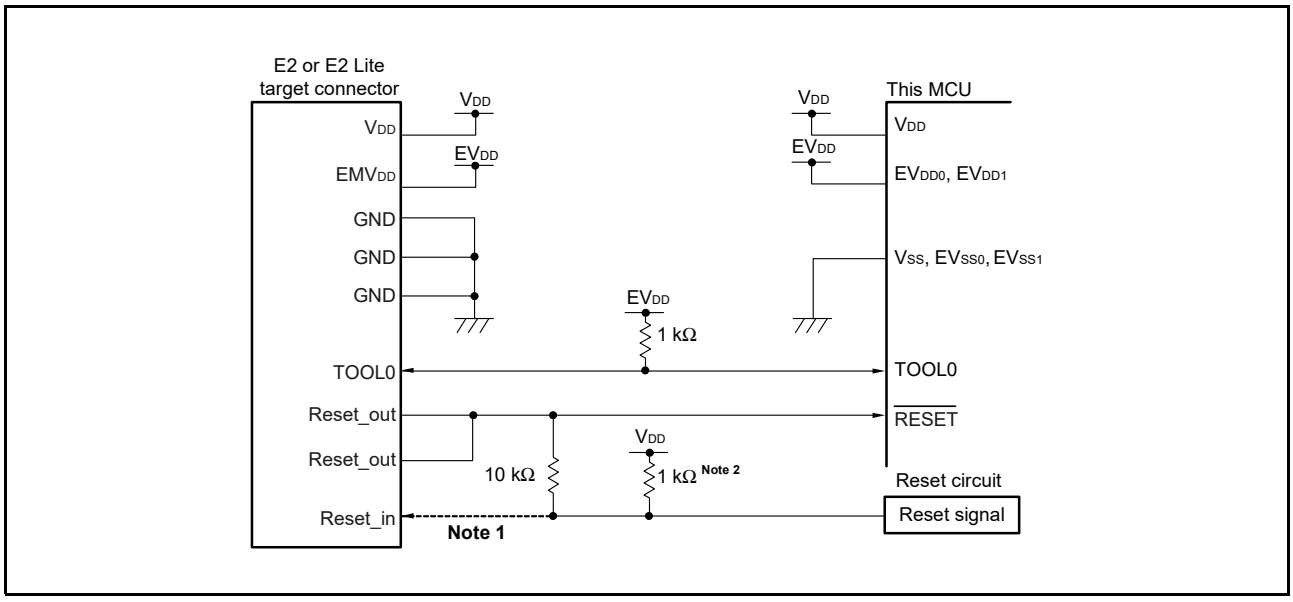
Section 34 On-chip Debugging

34.1 Connection between the E2 or E2 Lite On-chip Debugging Emulator and This MCU

On-chip debugging is handled by connecting this MCU and the host machine through the E2 or E2 Lite on-chip debugging emulator. Pins V_{DD}, EV_{DD0}, EV_{DD1}, V_{SS}, EV_{SS0}, EV_{SS1}, $\overline{RESET}$, and TOOL0 are used for connection to the emulator. Serial communications are handled through the TOOL0 pin as a single-line UART connection. For details and usage notes on the circuit to make the connection, refer to the **E1/E20/E2 Emulator, E2 Emulator Lite Additional Document for User's Manual (Notes on Connection of RL78) (R20UT1994)**.

Caution This MCU has on-chip debugging functionality for use in the development and evaluation of user systems. Do not use on-chip debugging with products designated as part of mass production, because using this function may cause the guaranteed number of times the flash memory is rewritten to be exceeded, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when on-chip debugging is used with products designated as part of mass production.

Figure 34 - 1 Example of Connection between the E2 or E2 Lite On-chip Debugging Emulator and This MCU



Note 1. The connection shown as a broken line is not required for serial programming.

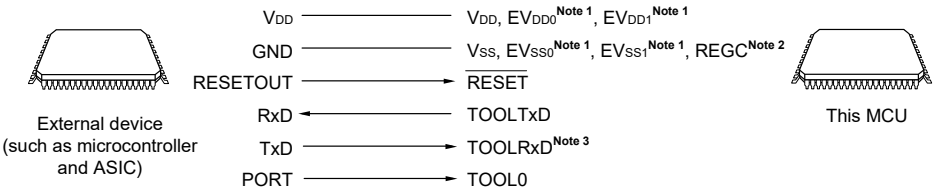
Note 2. If the reset circuit on the target system does not have a buffer so the reset signal is only generated through resistors and capacitors, this pull-up resistor is not required.

Caution This circuit diagram is an example where the reset signal is output from an N-channel open-drain buffer with an output resistance no greater than 100 Ω.

Remark Pins EV_{DD0}, EV_{DD1}, EV_{SS0}, and EV_{SS1} are not present in some products. In such cases, use the V_{DD} and V_{SS} pins as the alternative.

34.2 Connection between the External Device that Incorporates UART and This MCU

On-board communications between an external device (a microcontroller or ASIC) that is connected to this MCU via a UART and the host machine is possible. Pins V_{DD}, EV_{DD0}, EV_{DD1}, V_{SS}, EV_{SS0}, EV_{SS1}, $\overline{RESET}$, TOOLTxD, TOOLRxD, and TOOL0 are used for the communications. Communications between the external device and this MCU is established by serial communications using the TOOLTxD and TOOLRxD pins via the dedicated UART of this MCU. For details and usage notes on the circuit to make the connection, refer to the **RL78 Debugging Functions Using the Serial Port (R20AN0632EJ)**.



Note 1. This pin is only present in the 64- to 128-pin R7F100G products.

Note 2. Connect the REGC pin to the ground via a capacitor (0.47 to 1 µF).

Note 3. Set the port pin with which TOOLRxD is multiplexed as an input. The input to the input buffer must also be enabled by using the PDIDISxx register.

34.3 Security Settings for On-chip Debugging

To protect against third parties reading the contents of memory, on-chip debugging includes the following functionality.

- Disabling of connection between this MCU and the programmer or on-chip debugger (see **33.9 Security Settings** in **Section 33 Flash Memory**).
- On-chip debugging control bits in the flash memory at 000C3H (see **Section 32 Option Bytes**)
- An area in the range from 000C4H to 000CDH**Note** to hold the security ID code for on-chip debugging.

Note The area to hold the security ID code for use in on-chip debugging is also used to hold the ID code for the programmer connection ID authentication when a programmer is to be used.

Table 34 - 1 On-chip Debug Security ID

Address	Security ID Code for On-chip Debugging
000C4H to 000CDH	Any 10-byte ID code**Notes 1, 2**
040C4H to 040CDH	

Note 1. The setting FFFFFFFFFFFFFFFFFFFFH is not allowed.

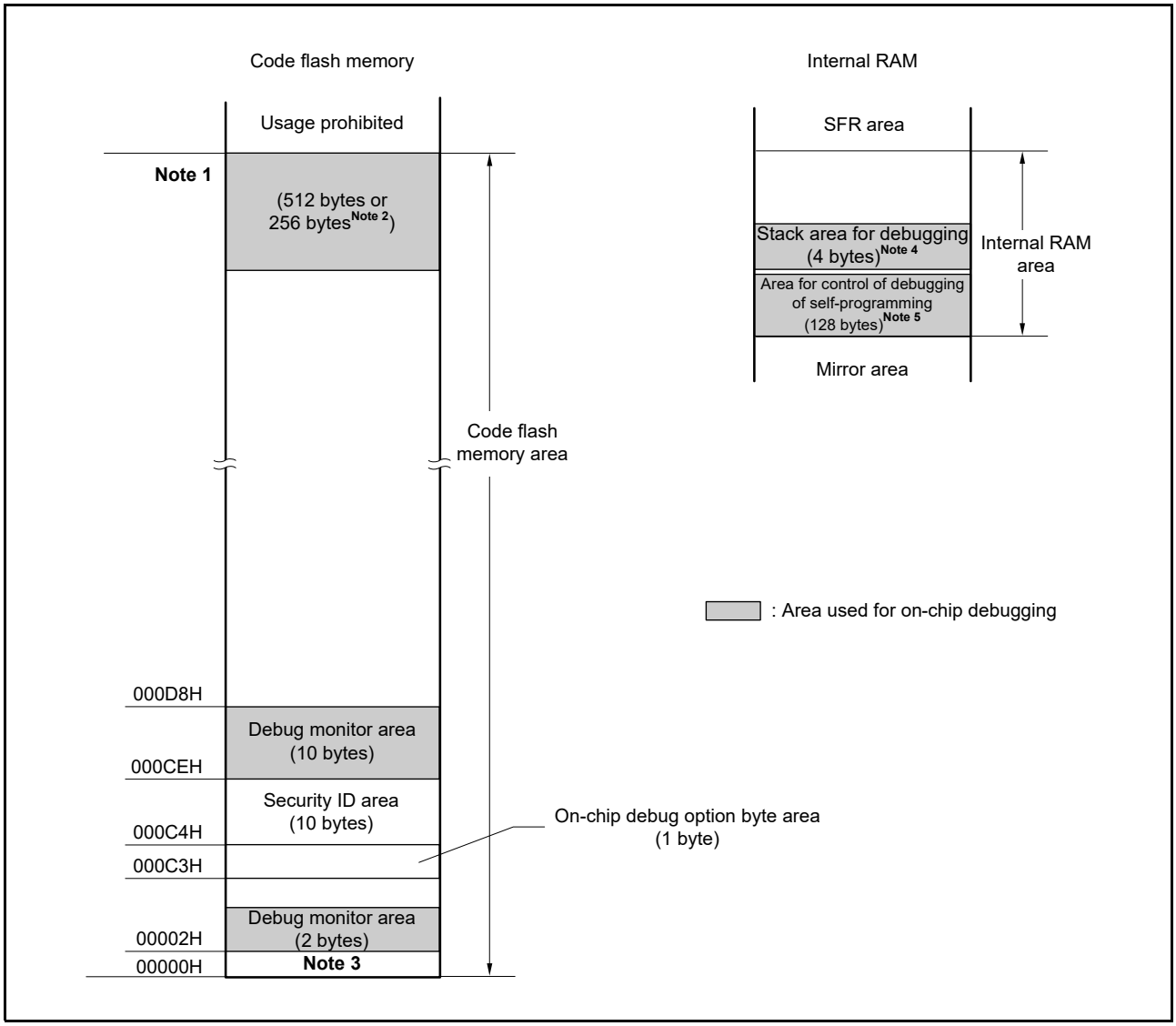
Note 2. Set the 10-byte security ID for on-chip debugging at both 000C4H to 000CDH and 040C4H to 040CDH when boot swapping is to be used or the value of the BTFLG bit in the FLSEC register is 0.

34.4 Allocation of Memory Spaces to User Resources

Allocation of memory spaces to user resources is required before communications between this MCU and E2 or E2 Lite on-chip debugging emulator, and on-chip debugging, can proceed. If you are using an assembler or compiler from Renesas Electronics, you can use linker options to allocate the memory spaces.

1. Allocation of memory spaces to the user program
   The shaded areas in **Figure 34 - 2** are reserved for the monitor program for debugging, and user programs and data cannot be allocated to these areas. When using on-chip debugging, ensure that nothing is allocated to these areas so that they can be secured for on-chip debugging. Also ensure that the contents of these areas are not modified by the user program.

Figure 34 - 2 Memory Spaces Allocated for Use by the Monitor Program for Debugging



Note 1. The address depends on the products as shown below.

Products	Address of **Note 1**
R7F100GxF (x = A to C, E to G, J, L)	17FFFH
R7F100GxG (x = A to C, E to G, J, L, M, P)	1FFFFH
R7F104GxG (x = B)	
R7F100GxH (x = A to C, E to G, J, L, M, P)	2FFFFH
R7F100GxJ (x = A to C, E to G, J, L, M, P, S)	3FFFFH
R7F104GxJ (x = F, G)	
R7F100GxK (x = F, G, J, L, M, P, S)	5FFFFH
R7F100GxL (x = F, G, J, L, M, P, S)	7FFFFH
R7F100GxN (x = F, G, J, L, M, P, S)	BFFFFH

Note 2. When the realtime RAM monitor (RRM) and dynamic memory modification (DMM) are not to be used, the size of this area is 256 bytes.

Note 3. During debugging, the reset vector is relocated to the address of the monitor program.

Note 4. Since this area is allocated immediately below the portion of the main stack area that is currently in use, the address range of this area depends on the amount of the stack in use other than for debugging. Accordingly, four additional bytes are required for the entire stack area. In the case of self-programming, this is a 12-byte area, so 12 additional bytes are required.

Note 5. The on-chip debugger uses the 128-byte RAM areas of the products listed in the table below for breaks in self-programming. The correspondences between target products and RAM areas are given in the table.

Products	RAM Area to be Used (128 bytes)
R7F100GxG (x = A, B, C, E, F, G, J, L)	FBF00H to FBF7FH
R7F104GxG (x = B)	
R7F100GxJ (x = A, B, C, E, F, G, J, L, M)	F9F00H to F9F7FH
R7F104GxJ (x = F, G)	
R7F100GxL (x = F, G, J, L, M, P, S)	F3F00H to F3F7FH
R7F100GxN (x = F, G, J, L, M, P, S)	F3F00H to F3F7FH

The given address range will not be in use unless the setting to enable debugging during self-programming are made by the on-chip debugger. For the setting of debugging of self-programming, refer to the user's manual for the given integrated development environment.

Section 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 Register for Controlling the BCD Correction Circuit

The following register is used to control the BCD correction circuit.

- BCD correction result register (BCDADJ)

35.2.1 BCD correction result register (BCDADJ)

The BCDADJ register holds 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. The value of this register following a reset is undefined.

Figure 35 - 1 Format of BCD Correction Result Register (BCDADJ)

Address: F00FEH
After reset: Undefined
R/W: 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: Obtaining the result of the binary addition of BCD values and BCD values as a BCD 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 values 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 and AC flags) is restored by the RETI instruction.

Examples are shown below.

Example 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	—

Example 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	—

Example 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: Obtaining the result of the binary subtraction of BCD values and BCD values as a BCD 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 and AC flags) 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	—

Section 36 Instruction Set

This section 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 (R01US0015)**.

36.1 Conventions Used in Operation List

36.1.1 Operand identifiers and description 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 Description Methods

Identifier	Description Method
r rp sfr sfrp	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7) AX (RP0), BC (RP1), DE (RP2), HL (RP3) Special-function register symbol (SFR symbol) FFF00H to FFFFFH Special-function register symbols (16-bit manipulatable SFR symbol. Even addresses only**Note**) FFF00H to FFFFFH
saddr saddrp	FFE20H to FFF1FH Immediate data or labels FFE20H to FF1FH Immediate data or labels (even addresses only**Note**)
addr20 addr16 addr5	00000H to FFFFFH Immediate data or labels 0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions**Note**) 0080H to 00BFH Immediate data or labels (even addresses only)
word byte bit	16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label
RBn	RB0 to RB3

Note Bit 0 = 0 when an odd address is specified.

Remark The special function registers can be described to operand sfr as symbols. See **Table 3 - 5 List of Special Function Registers (SFRs)** for the symbols of the special function registers. The extended special function registers can be described to operand !addr16 as symbols. See **Table 3 - 6 List of Extended Special Function Registers (2nd SFRs)** for the symbols of the extended special function registers.

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 or 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. An 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.

36.2 Operation List

Table 36 - 5 Operation List (1/18)

Instruction Group	Mnemonic	Operands	Bytes	Clock Cycles		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**Note 3**, A	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			

Note 1. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the internal RAM area, SFR area, or 2nd SFR area, or when the instruction does not involve data access.

Note 2. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the code flash memory area, or of access to the data flash memory area by an 8-bit instruction.

Note 3. Except when r = A.

Remark The listed numbers of clock cycles are for when the program is present in the internal ROM (flash memory) area. If the instruction is fetched from the internal RAM area, the number becomes up to double the listed number plus 3 clock cycles.

Table 36 - 5 Operation List (2/18)

Instruction Group	Mnemonic	Operands	Bytes	Clock Cycles		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			

Note 1. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the internal RAM area, SFR area, or 2nd SFR area, or when the instruction does not involve data access.

Note 2. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the code flash memory area, or of access to the data flash memory area by an 8-bit instruction.

Remark The listed numbers of clock cycles are for when the program is present in the internal ROM (flash memory) area. If the instruction is fetched from the internal RAM area, the number becomes up to double the listed number plus 3 clock cycles.

Table 36 - 5 Operation List (3/18)

Instruction Group	Mnemonic	Operands	Bytes	Clock Cycles		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)			

Note 1. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the internal RAM area, SFR area, or 2nd SFR area, or when the instruction does not involve data access.

Note 2. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the code flash memory area, or of access to the data flash memory area by an 8-bit instruction.

Note 3. Except when r = A.

Remark The listed numbers of clock cycles are for when the program is present in the internal ROM (flash memory) area. If the instruction is fetched from the internal RAM area, the number becomes up to double the listed number plus 3 clock cycles.

Table 36 - 5 Operation List (4/18)

Instruction Group	Mnemonic	Operands	Bytes	Clock Cycles		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**Note 3**, AX	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			

Note 1. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the internal RAM area, SFR area, or 2nd SFR area, or when the instruction does not involve data access.

Note 2. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the code flash memory area, or of access to the data flash memory area by an 8-bit instruction.

Note 3. Except when rp = AX.

Remark The listed numbers of clock cycles are for when the program is present in the internal ROM (flash memory) area. If the instruction is fetched from the internal RAM area, the number becomes up to double the listed number plus 3 clock cycles.

Table 36 - 5 Operation List (5/18)

Instruction Group	Mnemonic	Operands	Bytes	Clock Cycles		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			

Note 1. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the internal RAM area, SFR area, or 2nd SFR area, or when the instruction does not involve data access.

Note 2. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the code flash memory area, or of access to the data flash memory area by an 8-bit instruction.

Remark The listed numbers of clock cycles are for when the program is present in the internal ROM (flash memory) area. If the instruction is fetched from the internal RAM area, the number becomes up to double the listed number plus 3 clock cycles.

Table 36 - 5 Operation List (6/18)

Instruction Group	Mnemonic	Operands	Bytes	Clock Cycles		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)	×	×	×

Note 1. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the internal RAM area, SFR area, or 2nd SFR area, or when the instruction does not involve data access.

Note 2. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the code flash memory area, or of access to the data flash memory area by an 8-bit instruction.

Note 3. Except when rp = AX.

Note 4. Except when r = A.

Remark The listed numbers of clock cycles are for when the program is present in the internal ROM (flash memory) area. If the instruction is fetched from the internal RAM area, the number becomes up to double the listed number plus 3 clock cycles.

Table 36 - 5 Operation List (7/18)

Instruction Group	Mnemonic	Operands	Bytes	Clock Cycles		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, 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	×	×	×
	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)	×	×	×

Note 1. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the internal RAM area, SFR area, or 2nd SFR area, or when the instruction does not involve data access.

Note 2. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the code flash memory area, or of access to the data flash memory area by an 8-bit instruction.

Note 3. Except when r = A.

Remark The listed numbers of clock cycles are for when the program is present in the internal ROM (flash memory) area. If the instruction is fetched from the internal RAM area, the number becomes up to double the listed number plus 3 clock cycles.

Table 36 - 5 Operation List (8/18)

Instruction Group	Mnemonic	Operands	Bytes	Clock Cycles		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)	×		

Note 1. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the internal RAM area, SFR area, or 2nd SFR area, or when the instruction does not involve data access.

Note 2. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the code flash memory area, or of access to the data flash memory area by an 8-bit instruction.

Note 3. Except when r = A.

Remark The listed numbers of clock cycles are for when the program is present in the internal ROM (flash memory) area. If the instruction is fetched from the internal RAM area, the number becomes up to double the listed number plus 3 clock cycles.

Table 36 - 5 Operation List (9/18)

Instruction Group	Mnemonic	Operands	Bytes	Clock Cycles		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 ∨ (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)	×		
	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)	×		

Note 1. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the internal RAM area, SFR area, or 2nd SFR area, or when the instruction does not involve data access.

Note 2. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the code flash memory area, or of access to the data flash memory area by an 8-bit instruction.

Note 3. Except when r = A.

Remark The listed numbers of clock cycles are for when the program is present in the internal ROM (flash memory) area. If the instruction is fetched from the internal RAM area, the number becomes up to double the listed number plus 3 clock cycles.

Table 36 - 5 Operation List (10/18)

Instruction Group	Mnemonic	Operands	Bytes	Clock Cycles		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 Note 3	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)	×	×	×

Note 1. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the internal RAM area, SFR area, or 2nd SFR area, or when the instruction does not involve data access.

Note 2. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the code flash memory area, or of access to the data flash memory area by an 8-bit instruction.

Note 3. Except when r = A.

Remark The listed numbers of clock cycles are for when the program is present in the internal ROM (flash memory) area. If the instruction is fetched from the internal RAM area, the number becomes up to double the listed number plus 3 clock cycles.

Table 36 - 5 Operation List (11/18)

Instruction Group	Mnemonic	Operands	Bytes	Clock Cycles		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)	×	×	×

Note 1. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the internal RAM area, SFR area, or 2nd SFR area, or when the instruction does not involve data access.

Note 2. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the code flash memory area, or of access to the data flash memory area by an 8-bit instruction.

Remark The listed numbers of clock cycles are for when the program is present in the internal ROM (flash memory) area. If the instruction is fetched from the internal RAM area, the number becomes up to double the listed number plus 3 clock cycles.

Table 36 - 5 Operation List (12/18)

Instruction Group	Mnemonic	Operands	Bytes	Clock Cycles		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Multiply, Divide, Multiply & accumulate	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)		×	×

Note 1. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the internal RAM area, SFR area, or 2nd SFR area, or when the instruction does not involve data access.

Note 2. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the code flash memory area, or of access to the data flash memory area 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. The following compilers automatically add a NOP instruction immediately after any DIVHU or DIVWU instruction output during the build process.

- **V. 1.01.00 and later versions of CC-RL (Renesas Electronics compiler), for both C and assembly language source code**
- **Service pack 1.40.3 and later versions of the EWRL78 (IAR compiler), for C language source code**
- **LLVM RL78 (CyberTHOR compiler), for both C and C++ language source code**
- **GNURL78 (CyberTHOR compiler), for both C and C++ language source code**

Remark 1. The listed numbers of clock cycles are for when the program is present in the internal ROM (flash memory) area. If the instruction is fetched from the internal RAM area, the number becomes up to double the listed number plus 3 clock cycles.

Remark 2. MACR indicates the multiplication and accumulation register (MACRH, MACRL).

Table 36 - 5 Operation List (13/18)

Instruction Group	Mnemonic	Operands	Bytes	Clock Cycles		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			×

Note 1. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the internal RAM area, SFR area, or 2nd SFR area, or when the instruction does not involve data access.

Note 2. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the code flash memory area, or of access to the data flash memory area by an 8-bit instruction.

Remark 1. The listed numbers of clock cycles are for when the program is present in the internal ROM (flash memory) area. If the instruction is fetched from the internal RAM area, the number becomes up to double the listed number plus 3 clock cycles.

Remark 2. cnt indicates the bit shift count.

Table 36 - 5 Operation List (14/18)

Instruction Group	Mnemonic	Operands	Bytes	Clock Cycles		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			×

Note 1. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the internal RAM area, SFR area, or 2nd SFR area, or when the instruction does not involve data access.

Note 2. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the code flash memory area, or of access to the data flash memory area by an 8-bit instruction.

Remark The listed numbers of clock cycles are for when the program is present in the internal ROM (flash memory) area. If the instruction is fetched from the internal RAM area, the number becomes up to double the listed number plus 3 clock cycles.

Table 36 - 5 Operation List (15/18)

Instruction Group	Mnemonic	Operands	Bytes	Clock Cycles		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 ← $\overline{CY}$			×

Note 1. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the internal RAM area, SFR area, or 2nd SFR area, or when the instruction does not involve data access.

Note 2. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the code flash memory area, or of access to the data flash memory area by an 8-bit instruction.

Remark The listed numbers of clock cycles are for when the program is present in the internal ROM (flash memory) area. If the instruction is fetched from the internal RAM area, the number becomes up to double the listed number plus 3 clock cycles.

Table 36 - 5 Operation List (16/18)

Instruction Group	Mnemonic	Operands	Bytes	Clock Cycles		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

Note 1. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the internal RAM area, SFR area, or 2nd SFR area, or when the instruction does not involve data access.

Note 2. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the code flash memory area, or of access to the data flash memory area by an 8-bit instruction.

Remark The listed numbers of clock cycles are for when the program is present in the internal ROM (flash memory) area. If the instruction is fetched from the internal RAM area, the number becomes up to double the listed number plus 3 clock cycles.

Table 36 - 5 Operation List (17/18)

Instruction Group	Mnemonic	Operands	Bytes	Clock Cycles		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	—	rpL ← (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	2/4Note 3	—	PC ← PC + 2 + jdisp8 if CY = 1			
	BNC	$addr20	2	2/4Note 3	—	PC ← PC + 2 + jdisp8 if CY = 0			
	BZ	$addr20	2	2/4Note 3	—	PC ← PC + 2 + jdisp8 if Z = 1			
	BNZ	$addr20	2	2/4Note 3	—	PC ← PC + 2 + jdisp8 if Z = 0			
	BH	$addr20	3	2/4Note 3	—	PC ← PC + 3 + jdisp8 if (Z ∨ CY) = 0			
	BNH	$addr20	3	2/4Note 3	—	PC ← PC + 3 + jdisp8 if (Z ∨ CY) = 1			
	BT	saddr.bit, $addr20	4	3/5Note 3	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 1			
		sfr.bit, $addr20	4	3/5Note 3	—	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
		A.bit, $addr20	3	3/5Note 3	—	PC ← PC + 3 + jdisp8 if A.bit = 1			
		PSW.bit, $addr20	4	3/5Note 3	—	PC ← PC + 4 + jdisp8 if PSW.bit = 1			
		[HL].bit, $addr20	3	3/5Note 3	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 1			
		ES:[HL].bit, $addr20	4	4/6Note 3	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1			

Note 1. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the internal RAM area, SFR area, or 2nd SFR area, or when the instruction does not involve data access.

Note 2. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the code flash memory area, or of access to the data flash memory area by an 8-bit instruction.

Note 3. These indicate the numbers of clock cycles when the condition is and is not met.

Remark The listed numbers of clock cycles are for when the program is present in the internal ROM (flash memory) area. If the instruction is fetched from the internal RAM area, the number becomes up to double the listed number plus 3 clock cycles.

Table 36 - 5 Operation List (18/18)

Instruction Group	Mnemonic	Operands	Bytes	Clock Cycles		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Conditional branch	BF	saddr.bit, $addr20	4	3/5Note 3	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 0			
		sfr.bit, $addr20	4	3/5Note 3	—	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
		A.bit, $addr20	3	3/5Note 3	—	PC ← PC + 3 + jdisp8 if A.bit = 0			
		PSW.bit, $addr20	4	3/5Note 3	—	PC ← PC + 4 + jdisp8 if PSW.bit = 0			
		[HL].bit, $addr20	3	3/5Note 3	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 0			
		ES:[HL].bit, $addr20	4	4/6Note 3	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 0			
	BTCLR	saddr.bit, $addr20	4	3/5Note 3	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 1 then reset (saddr).bit			
		sfr.bit, $addr20	4	3/5Note 3	—	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit			
		A.bit, $addr20	3	3/5Note 3	—	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit			
		PSW.bit, $addr20	4	3/5Note 3	—	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	×	×	×
		[HL].bit, $addr20	3	3/5Note 3	—	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit			
		ES:[HL].bit, $addr20	4	4/6Note 3	—	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	RBnNote 4	2	1	—	RBS1, RBS0 ← 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			

Note 1. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the internal RAM area, SFR area, or 2nd SFR area, or when the instruction does not involve data access.

Note 2. These are the numbers of CPU clock cycles (f_{CLK}) in the case of access to the code flash memory area, or of access to the data flash memory area by an 8-bit instruction.

Note 3. These indicate the numbers of clock cycles when the condition is and is not met.

Note 4. n indicates the number of register banks (n = 0 to 3).

Remark The listed numbers of clock cycles are for when the program is present in the internal ROM (flash memory) area. If the instruction is fetched from the internal RAM area, the number becomes up to double the listed number plus 3 clock cycles.

Section 37 Electrical Characteristics

This section describes the electrical characteristics of the following products.

- 2D : Consumer applications, T_A = -40 to +85°C
  R7F100Gxx2Dxx
- 3C : Industrial applications, T_A = -40 to +105°C
  R7F100Gxx3Cxx and R7F104Gxx3Cxx

Caution 1. This MCU has on-chip debugging functionality for use in the development and evaluation of user systems. Do not use on-chip debugging with products designated as part of mass production, because using this function may cause the guaranteed number of times the flash memory is rewritten to be exceeded, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when on-chip debugging is used with products designated as part of mass production.

Caution 2. For the consumer application products, the ambient operating temperature of T_A = -40°C to +85°C applies.

Caution 3. For products that do not have an EV_{DD0}, EV_{DD1}, EV_{SS0}, or EV_{SS1} pin, read EV_{DD0} and EV_{DD1} as V_{DD}, and EV_{SS0} and EV_{SS1} as V_{SS}.

Caution 4. The present pins differ depending on the products. For details, see section 2.1 Functions of Port Pins through section 2.2.1 Non-GPIO pin functions of the R7F100G products per number of pins.

37.1 Absolute Maximum Ratings

(1/2)

Item	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}		-0.5 to +6.5	V
	EV_{DD0}, EV_{DD1}	EV_{DD0} = EV_{DD1}	-0.5 to +6.5	V
	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 P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P100 to P106, P110 to P117, P120, P125 to P127, P140 to P147	-0.3 to EV_{DD0} + 0.3 and -0.3 to V_{DD} + 0.3 **Note 2**	V
	V_{I2}	P60 to P63 (N-ch open-drain)	-0.3 to +6.5	V
	V_{I3}	P20 to P27, P121 to P124, P137, P150 to P156, EXCLK, EXCLKS, $\overline{\text{RESET}}$	-0.3 to V_{DD} + 0.3 **Note 2**	V
Output voltage	V_{O1}	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P97, P100 to P106, P110 to P117, P120, P125 to P127, P130, P140 to P147	-0.3 to EV_{DD0} + 0.3 and -0.3 to V_{DD} + 0.3 **Note 2**	V
	V_{O2}	P20 to P27, P121, P122, P150 to P156	-0.3 to V_{DD} + 0.3 **Note 2**	V
Analog input voltage	V_{AI1}	ANI16 to ANI26	-0.3 to EV_{DD0} + 0.3 and -0.3 to AV_{REFP} + 0.3 **Notes 2, 3**	V
	V_{AI2}	ANI0 to ANI14	-0.3 to V_{DD} + 0.3 and -0.3 to AV_{REFP} + 0.3 **Notes 2, 3**	V

Note 1. Connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF). The listed value is the absolute maximum rating of the REGC pin. Only use the capacitor connection. Do not apply a specific voltage to this pin.

Note 2. This voltage must be no higher than 6.5 V.

Note 3. The voltage on a pin in use for A/D conversion must not exceed AV_{REFP} + 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 1. The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

Remark 2. AV_{REFP} refers to the positive reference voltage of the A/D converter.

Remark 3. The reference voltage is V_{SS}.

(2/2)

Item	Symbols	Conditions		Ratings	Unit
High-level output current	I_{OH1}	Per pin	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P100 to P106, P110 to P117, P120, P125 to P127, P130, P140 to P147	-40	mA
		Total of all pins -170 mA	P00 to P04, P07, P32 to P37, P40 to P47, P102 to P106, P120, P125 to P127, P130, P140 to P145	-70	mA
			P05, P06, P10 to P17, P30, P31, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P100, P101, P110 to P117, P146, P147	-100	mA
	I_{OH2}	Per pin	P20 to P27, P121, P122, P150 to P156	-5	mA
		Total of all pins		-20	mA
Low-level output current	I_{OL1}	Per pin	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P97, P100 to P106, P110 to P117, P120, P125 to P127, P130, P140 to P147	40 Note	mA
		Total of all pins 170 mA	P00 to P04, P07, P32 to P37, P40 to P47, P102 to P106, P120, P125 to P127, P130, P140 to P145	70	mA
			P05, P06, P10 to P17, P30, P31, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P97, P100, P101, P110 to P117, P146, P147	100	mA
	I_{OL2}	Per pin	P20 to P27, P121, P122, P150 to P156	10	mA
		Total of all pins		20	mA
Ambient operating temperature	T_A	In normal operation mode	3C: Industrial applications	-40 to +105	°C
			2D: Consumer applications	-40 to +85	
		In flash memory programming mode	3C: Industrial applications	-40 to +105	
			2D: Consumer applications	-40 to +85	
Storage temperature	T_{stg}			-65 to +150	°C

Note The rating for the following port pins is 80 mA when I_{OL1} = 40.0 mA is specified by the 40-mA port output control register (PTDC).

- Pins P04, P10, and P120 of the 64- to 100-pin R7F100G products with 384- to 768-Kbyte code flash memory
- Pin P101 of the 100-pin R7F100G products with 384- to 768-Kbyte code flash memory
- Pins P17 and P51 of the 30- to 52-pin R7F100G products
- Pin P70 of the 32- to 52-pin R7F100G products

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 The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

37.2 Characteristics of the Oscillators

37.2.1 Characteristics of the X1 oscillator

(T_A = -40 to +105°C, 1.6 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Item	Resonator	Conditions	Min.	Typ.	Max.	Unit
X1 clock oscillation allowable input cycle time**Note**	Ceramic resonator/ crystal resonator		0.05		1	μs

Note The listed time and frequency indicate permissible ranges of the oscillator. For actual applications, request evaluation by the manufacturer of the oscillator circuit mounted on a board so you can use appropriate values. Refer to **37.4 AC Characteristics** for instruction execution time.

Caution **Since the CPU is started by the high-speed on-chip oscillator clock after release from the reset state, the user should use the oscillation stabilization time counter status register (OSTC) to check the X1 clock oscillation stabilization time. Specify the values for the oscillation stabilization time in the OSTC register and the oscillation stabilization time select register (OSTS) after having sufficiently evaluated the oscillation stabilization time with the resonator to be used.**

37.2.2 Characteristics of the XT1 oscillator

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V for the 30- to 36-pin R7F100G products, 1.6 V ≤ V_{DD} ≤ 5.5 V for the 40- to 128-pin R7F100G and 44- and 48-pin R7F104G products, V_{SS} = 0 V)

Item	Resonator	Conditions	Min.	Typ.	Max.	Unit
XT1 clock oscillation frequency (f_{XT})**Note**	Crystal resonator			32.768		kHz

Note The listed time and frequency indicate permissible ranges of the oscillator. For actual applications, request evaluation by the manufacturer of the oscillator circuit mounted on a board so you can use appropriate values. Refer to **37.4 AC Characteristics** for instruction execution time.

37.2.3 Characteristics of the on-chip oscillators

(T_A = -40 to +105°C, 1.6 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Item	Symbol	Conditions			Min.	Typ.	Max.	Unit
High-speed on-chip oscillator clock frequency	f_{IH}				1		32	MHz
High-speed on-chip oscillator clock frequency accuracy **Note 1**		HIPREC = 1	+85 to +105°C	1.8 V ≤ V_{DD} ≤ 5.5 V	-2.0		+2.0	%
				1.6 V ≤ V_{DD} ≤ 5.5 V	-6.0		+6.0	%
			-20 to +85°C	1.8 V ≤ V_{DD} ≤ 5.5 V	-1.0		+1.0	%
				1.6 V ≤ V_{DD} ≤ 5.5 V	-5.0		+5.0	%
			-40 to -20°C	1.8 V ≤ V_{DD} ≤ 5.5 V	-1.5		+1.5	%
				1.6 V ≤ V_{DD} ≤ 5.5 V	-5.5		+5.5	%
		HIPREC = 0 **Note 2**			-15		0	%
High-speed on-chip oscillator clock correction resolution						0.05		%
Middle-speed on-chip oscillator clock frequency **Note 3**	f_{IM}				1		4	MHz
Middle-speed on-chip oscillator clock frequency accuracy **Note 1**					-12		+12	%
Middle-speed on-chip oscillator clock correction resolution						0.15		%
Middle-speed on-chip oscillator frequency temperature coefficient							±0.17 **Note 4**	%/°C
Low-speed on-chip oscillator clock frequency **Note 3**	f_{IL}					32.768		kHz
Low-speed on-chip oscillator clock frequency accuracy **Note 1**					-15		+15	%
Low-speed on-chip oscillator clock correction resolution						0.3		%
Low-speed on-chip oscillator frequency temperature coefficient							±0.21 **Note 4**	%/°C

Note 1. The accuracy values were obtained in testing of this product.

Note 2. The listed condition applies when the setting of the FRQSEL3 bit of the user option byte is 1.

Note 3. The listed values only indicate the characteristics of the oscillators. Refer to **37.4 AC Characteristics** for instruction execution time.

Note 4. These values are the results of characteristic evaluation and are not checked for shipment.

37.3 DC Characteristics

37.3.1 Characteristics of pins

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (1/7)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Allowable high-level output current**Note 1**	I_{OH1}	Per pin for P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P100 to P106, P110 to P117, P120, P125 to P127, P130, P140 to P147	1.6 V ≤ EV_{DD0} ≤ 5.5 V			-10.0 **Note 2**	mA
		Total of P00 to P04, P07, P32 to P37, P40 to P47, P102 to P106, P120, P125 to P127, P130, P140 to P145 (when duty ≤ 70%**Note 3**)	4.0 V ≤ EV_{DD0} ≤ 5.5 V			-55.0 **Note 4**	mA
			2.7 V ≤ EV_{DD0} < 4.0 V			-10.0	mA
			1.8 V ≤ EV_{DD0} < 2.7 V			-5.0	mA
			1.6 V ≤ EV_{DD0} < 1.8 V			-2.5	mA
		Total of P05, P06, P10 to P17, P30, P31, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P100, P101, P110 to P117, P146, P147 (when duty ≤ 70%**Note 3**)	4.0 V ≤ EV_{DD0} ≤ 5.5 V			-80.0 **Note 5**	mA
			2.7 V ≤ EV_{DD0} < 4.0 V			-19.0	mA
			1.8 V ≤ EV_{DD0} < 2.7 V			-10.0	mA
			1.6 V ≤ EV_{DD0} < 1.8 V			-5.0	mA
		Total of all pins (when duty ≤ 70%**Note 3**)	1.6 V ≤ EV_{DD0} ≤ 5.5 V			-135.0 **Note 6**	mA
	I_{OH2}	Per pin for P20 to P27, P121, P122, P150 to P156	4.0 V ≤ V_{DD} ≤ 5.5 V			-3.0 **Note 2**	mA
			2.7 V ≤ V_{DD} < 4.0 V			-1.0 **Note 2**	mA
			1.8 V ≤ V_{DD} < 2.7 V			-1.0 **Note 2**	mA
			1.6 V ≤ V_{DD} < 1.8 V			-0.5 **Note 2**	mA
		Total of all pins (when duty ≤ 70%**Note 3**)	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
			1.6 V ≤ V_{DD} < 1.8 V			-5.0	mA

Note 1. Device operation is guaranteed at the listed currents even if current is flowing from the EV_{DD0}, EV_{DD1}, or V_{DD} pin to an output pin.

Note 2. The combination of these and other pins must also not exceed the value for maximum total current.

Note 3. The listed currents apply when the duty cycle is no greater than 70%. Use the following formula to calculate the output current when the duty cycle is greater than 70%, where n is the duty cycle.

- Total output current from the listed pins = (I_{OH} × 0.7)/(n × 0.01)

  Example when n = 80% and I_{OH} = -10.0 mA

  Total output current from the listed pins = (-10.0 × 0.7)/(80 × 0.01) ≈ -8.7 mA

Note that the duty cycle has no effect on the current that is allowed to flow into a single pin. A current higher than the absolute maximum rating must not flow into a single pin.

(Notes, Caution, and Remark continue on the next page.)

Note 4. The maximum value is -30 mA in the products for industrial applications (R7F100Gxx3Cxx and R7F104Gxx3Cxx) with an ambient operating temperature range of +85°C to +105°C.

Note 5. The maximum value is -50 mA in the products for industrial applications (R7F100Gxx3Cxx and R7F104Gxx3Cxx) with an ambient operating temperature range of +85°C to +105°C.

Note 6. The maximum values are respectively -100 mA and -60 mA in the products for industrial applications (R7F100Gxx3Cxx and R7F104Gxx3Cxx) with an ambient operating temperature range of -40°C to +85°C and of +85°C to +105°C.

Caution **The following pins are not capable of the output of high-level signals in the N-ch open-drain output mode. P00, P02 to P04, P10 to P15, P17, P34, P42 to P45, P50, P52 to P55, P71, P72, P74, P80 to P83, P96, P120, and P142 to P144**

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (2/7)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Allowable low-level output current**Note 1**	I_{OL1}	Per pin for P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P100 to P106, P110 to P117, P120, P125 to P127, P130, P140 to P147				20.0 **Notes 2, 3**	mA
		Per pin for P60 to P63				15.0 **Note 2**	mA
		Total of P00 to P04, P07, P32 to P37, P40 to P47, P102 to P106, P120, P125 to P127, P130, P140 to P145 (when duty ≤ 70%**Note 4**)	4.0 V ≤ EV_{DD0} ≤ 5.5 V			70.0 **Note 5**	mA
			2.7 V ≤ EV_{DD0} < 4.0 V			15.0	mA
			1.8 V ≤ EV_{DD0} < 2.7 V			9.0	mA
			1.6 V ≤ EV_{DD0} < 1.8 V			4.5	mA
		Total of P05, P06, P10 to P17, P30, P31, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P97, P100, P101, P110 to P117, P146, P147 (when duty ≤ 70%**Note 4**)	4.0 V ≤ EV_{DD0} ≤ 5.5 V			80.0 **Note 5**	mA
			2.7 V ≤ EV_{DD0} < 4.0 V			35.0	mA
			1.8 V ≤ EV_{DD0} < 2.7 V			20.0	mA
			1.6 V ≤ EV_{DD0} < 1.8 V			10.0	mA
		Total of all pins (when duty ≤ 70%**Note 4**)				150.0 **Note 6**	mA
	I_{OL2}	Per pin for P20 to P27, P121, P122, P150 to P156	4.0 V ≤ V_{DD} ≤ 5.5 V			8.5**Note 2**	mA
			2.7 V ≤ V_{DD} < 4.0 V			1.5**Note 2**	mA
			1.8 V ≤ V_{DD} < 2.7 V			0.6**Note 2**	mA
			1.6 V ≤ V_{DD} < 1.8 V			0.4**Note 2**	mA
		Total of all pins (when duty ≤ 70%**Note 4**)	4.0 V ≤ V_{DD} ≤ 5.5 V			20	mA
			2.7 V ≤ V_{DD} < 4.0 V			20	mA
			1.8 V ≤ V_{DD} < 2.7 V			15	mA
			1.6 V ≤ V_{DD} < 1.8 V			10	mA

Note 1. Device operation is guaranteed at the listed currents even if current is flowing from an output pin to the EV_{SS0}, EV_{SS1}, or V_{SS} pin.

Note 2. The combination of these and other pins must also not exceed the value for maximum total current.

Note 3. The maximum rating for the following port pins is 40 mA when I_{OL1} = 40.0 mA is specified by the 40-mA port output control register (PTDC).

- Pins P04, P10, and P120 of the 64- to 100-pin R7F100G products with 384- to 768-Kbyte code flash memory
- Pin P101 of the 100-pin R7F100G products with 384- to 768-Kbyte code flash memory
- Pins P17 and P51 of the 30- to 52-pin R7F100G products
- Pin P70 of the 32- to 52-pin R7F100G products

Note 4. The listed currents apply when the duty cycle is no greater than 70%. Use the following formula to calculate the output current when the duty cycle is greater than 70%, where n is the duty cycle.

- Total output current from the listed pins = (I_{OL} × 0.7)/(n × 0.01)

  Example when n = 80% and I_{OL} = 10.0 mA

  Total output current from the listed pins = (10.0 × 0.7)/(80 × 0.01) ≈ 8.7 mA

Note that the duty cycle has no effect on the current that is allowed to flow into a single pin. A current higher than the absolute maximum rating must not flow into a single pin.

(Notes and Remark continue on the next page.)

Note 5. The maximum value is 40 mA in the products for industrial applications (R7F100Gxx3Cxx and R7F104Gxx3Cxx) with an ambient operating temperature range of +85°C to +105°C.

Note 6. The maximum value is 80 mA in the products for industrial applications (R7F100Gxx3Cxx and R7F104Gxx3Cxx) with an ambient operating temperature range of +85°C to +105°C.

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (3/7)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Input voltage, high	V_{IH1}	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P100 to P106, P110 to P117, P120, P125 to P127, P140 to P147	Normal input buffer	0.8 EV_{DD0}		EV_{DD0}	V
	V_{IH2}	P01, P03, P04, P10, P11, P13 to P17, P41, P43, P44, P53 to P55, P71, P80, P81, P142, P143	TTL input buffer 4.0 V ≤ EV_{DD0} ≤ 5.5 V	2.2		EV_{DD0}	V
			TTL input buffer 3.3 V ≤ EV_{DD0} < 4.0 V	2.0		EV_{DD0}	V
			TTL input buffer 1.6 V ≤ EV_{DD0} < 3.3 V	1.5		EV_{DD0}	V
	V_{IH3}	P20 to P27, P150 to P156		0.7 V_{DD}		V_{DD}	V
	V_{IH4}	P60 to P63		0.7 EV_{DD0}		6.0	V
	V_{IH5}	P121 to P124, P137, EXCLK, EXCLKS, $\overline{RESET}$		0.8 V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P100 to P106, P110 to P117, P120, P125 to P127, P140 to P147	Normal input buffer	0		0.2 EV_{DD0}	V
	V_{IL2}	P01, P03, P04, P10, P11, P13 to P17, P41, P43, P44, P53 to P55, P71, P80, P81, P142, P143	TTL input buffer 4.0 V ≤ EV_{DD0} ≤ 5.5 V	0		0.8	V
			TTL input buffer 3.3 V ≤ EV_{DD0} < 4.0 V	0		0.5	V
			TTL input buffer 1.6 V ≤ EV_{DD0} < 3.3 V	0		0.32	V
	V_{IL3}	P20 to P27, P150 to P156		0		0.3 V_{DD}	V
	V_{IL4}	P60 to P63		0		0.3 EV_{DD0}	V
	V_{IL5}	P121 to P124, P137, EXCLK, EXCLKS, $\overline{RESET}$		0		0.2 V_{DD}	V

Caution The maximum value of V_{IH} of pins P00, P02 to P04, P10 to P15, P17, P34, P42 to P45, P50, P52 to P55, P71, P72, P74, P80 to P83, P96, P120, and P142 to P144 is EV_{DD0}, even in the N-ch open-drain output mode.

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (4/7)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Output voltage, high	V_{OH1}	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P100 to P106, P110 to P117, P120, P125 to P127, P130, P140 to P147	4.0 V ≤ EV_{DD0} ≤ 5.5 V, I_{OH1} = -10.0 mA	EV_{DD0} - 1.5			V
			4.0 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} = -2.0 mA	EV_{DD0} - 0.6			V
			1.8 V ≤ EV_{DD0} ≤ 5.5 V, I_{OH1} = -1.5 mA	EV_{DD0} - 0.5			V
			1.6 V ≤ EV_{DD0} < 5.5 V, I_{OH1} = -1.0 mA	EV_{DD0} - 0.5			V
	V_{OH2}	P20 to P27, P121, P122, P150 to P156	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH2} = -3.0 mA	V_{DD} - 0.7			V
			2.7 V ≤ V_{DD} < 4.0 V, I_{OH2} = -1.0 mA	V_{DD} - 0.5			V
			1.8 V ≤ V_{DD} < 2.7 V, I_{OH2} = -1.0 mA	V_{DD} - 0.5			V
			1.6 V ≤ V_{DD} < 1.8 V, I_{OH2} = -0.5 mA	V_{DD} - 0.5			V

Caution **P00, P02 to P04, P10 to P15, P17, P34, P42 to P45, P50, P52 to P55, P71, P72, P74, P80 to P83, P96, P120, and P142 to P144 do not output high-level signals in the N-ch open-drain output mode.**

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (5/7)

Item	Symbol	Conditions			Min.	Typ.	Max.	Unit
Output voltage, low	V_{OL1}	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P100 to P106, P110 to P117, P120, P125 to P127, P130, P140 to P147	4.0 V ≤ EV_{DD0} ≤ 5.5 V	I_{OL1} = 20.0 mA			1.3	V
				I_{OL1} = 40.0 mA Note			1.3	V
			4.0 V ≤ EV_{DD0} ≤ 5.5 V	I_{OL1} = 8.5 mA			0.7	V
				I_{OL1} = 17.0 mA Note			0.7	V
			2.7 V ≤ EV_{DD0} ≤ 5.5 V	I_{OL1} = 3.0 mA			0.6	V
				I_{OL1} = 6.0 mA Note			0.6	V
			2.7 V ≤ EV_{DD0} ≤ 5.5 V	I_{OL1} = 1.5 mA			0.4	V
				I_{OL1} = 3.0 mA Note			0.4	V
			1.8 V ≤ EV_{DD0} ≤ 5.5 V	I_{OL1} = 0.6 mA			0.4	V
				I_{OL1} = 1.2 mA Note			0.4	V
			1.6 V ≤ EV_{DD0} ≤ 5.5 V	I_{OL1} = 0.3 mA			0.4	V
				I_{OL1} = 0.6 mA Note			0.4	V
	V_{OL2}	P20 to P27, P121, P122, P150 to P156	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL2} = 8.5 mA				0.7	V
			2.7 V ≤ V_{DD} < 4.0 V, I_{OL2} = 1.5 mA				0.5	V
			1.8 V ≤ V_{DD} < 2.7 V, I_{OL2} = 0.6 mA				0.4	V
			1.6 V ≤ V_{DD} < 1.8 V, I_{OL2} = 0.4 mA				0.4	V
	V_{OL3}	P60 to P63	4.0 V ≤ EV_{DD0} ≤ 5.5 V, I_{OL3} = 15.0 mA				2.0	V
			4.0 V ≤ EV_{DD0} ≤ 5.5 V, I_{OL3} = 5.0 mA				0.4	V
			2.7 V ≤ EV_{DD0} ≤ 5.5 V, I_{OL3} = 3.0 mA				0.4	V
			1.8 V ≤ EV_{DD0} ≤ 5.5 V, I_{OL3} = 2.0 mA				0.4	V
			1.6 V ≤ EV_{DD0} ≤ 5.5 V, I_{OL3} = 1.0 mA				0.4	V

Note The listed value applies when I_{OL1} = 40.0 mA is specified for the following port pins by the 40-mA port output control register (PTDC).

- Pins P04, P10, and P120 of the 64- to 100-pin R7F100G products with 384- to 768-Kbyte code flash memory
- Pin P101 of the 100-pin R7F100G products with 384- to 768-Kbyte code flash memory
- Pins P17 and P51 of the 30- to 52-pin R7F100G products
- Pin P70 of the 32- to 52-pin R7F100G products

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (6/7)

Item	Symbol	Conditions			Min.	Typ.	Max.	Unit
Output current[Note]	CCDIOL	P16, P17, P50, P51, P60 to P63	CCSm = 01H	4.0 V ≤ EV_{DD0} ≤ 5.5 V	1.0	1.8	2.6	mA
				2.7 V ≤ EV_{DD0} < 4.0 V	0.8	1.5	2.3	mA
			CCSm = 02H	4.0 V ≤ EV_{DD0} ≤ 5.5 V	3.0	4.9	6.5	mA
				3.0 V ≤ EV_{DD0} < 4.0 V	2.7	4.3	5.9	mA
			CCSm = 03H	4.0 V ≤ EV_{DD0} ≤ 5.5 V	6.6	10.0	13.2	mA
				3.3 V ≤ EV_{DD0} < 4.0 V	6.0	9.1	12.1	mA
		P60 to P63	CCSm = 04H	4.0 V ≤ EV_{DD0} ≤ 5.5 V	10.2	15.0	19.8	mA
				3.3 V ≤ EV_{DD0} < 4.0 V	9.4	13.8	18.2	mA

Note The listed currents apply when the output current control function is enabled.

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (7/7)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Input leakage current, high	I_{LIH1}	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P97, P100 to P106, P110 to P117, P120, P125 to P127, P140 to P147	V_I = EV_{DD0}			0.5	μA
	I_{LIH2}	P20 to P27, P137, P150 to P156, $\overline{\text{RESET}}$	V_I = V_{DD}			0.5	μA
	I_{LIH3}	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	V_I = V_{DD}			0.5	μA
Input leakage current, low	I_{LIL1}	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P97, P100 to P106, P110 to P117, P120, P125 to P127, P140 to P147	V_I = EV_{SS0}			-0.5	μA
	I_{LIL2}	P20 to P27, P137, P150 to P156, $\overline{\text{RESET}}$	V_I = V_{SS}			-0.5	μA
	I_{LIL3}	P121 to P124 (X1, X2, XT1, XT2, EXCLK, EXCLKS)	V_I = V_{SS}			-0.5	μA
On-chip pull-up resistance	R_U	P00 to P07, P10 to P17, P30 to P37, P40 to P47, P50 to P57, P64 to P67, P70 to P77, P80 to P87, P90 to P97, P100 to P106, P110 to P117, P120 to P122, P125 to P127, P140 to P147	V_I = EV_{SS0}, in input port	10	20	100	kΩ

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

37.3.2 Characteristics of the supply current

1. 30- to 64-pin R7F100G and 32-pin R7F104G products with 96- to 128-Kbyte code flash memory

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (1/4)

Item	Symbol	Conditions					Min.	Typ.	Max.	Unit
Supply current **Note 1**	I_{DD1}	Operating mode	HS (high-speed main) mode	f_{IH} = 32 MHz **Note 2**	Basic operation	V_{DD} = 5.0 V		1.3	—	mA
						V_{DD} = 1.8 V		1.3	—	
					Normal operation	V_{DD} = 5.0 V		3.0	5.0	mA
						V_{DD} = 1.8 V		3.0	5.0	
			LS (low-speed main) mode	f_{IH} = 24 MHz **Note 2**	Normal operation	V_{DD} = 5.0 V		2.3	3.8	mA
						V_{DD} = 1.8 V		2.3	3.8	
				f_{IH} = 16 MHz **Note 2**	Normal operation	V_{DD} = 5.0 V		1.7	2.7	mA
						V_{DD} = 1.8 V		1.7	2.7	
				f_{IM} = 4 MHz **Note 3**	Normal operation	V_{DD} = 5.0 V		0.4	0.7	mA
						V_{DD} = 1.6 V		0.4	0.7	
			LP (low-power main) mode	f_{IM} = 2 MHz **Note 3**	Normal operation	V_{DD} = 5.0 V		200	325	μA
						V_{DD} = 1.6 V		200	325	
				f_{IM} = 1 MHz **Note 3**	Normal operation	V_{DD} = 5.0 V		112	178	μA
						V_{DD} = 1.6 V		111	176	
			HS (high-speed main) mode	f_{MX} = 20 MHz **Note 4**, square wave input	Normal operation	V_{DD} = 5.0 V		1.9	3.2	mA
						V_{DD} = 1.8 V		1.9	3.2	
			LS (low-speed main) mode	f_{MX} = 20 MHz **Note 4**, square wave input	Normal operation	V_{DD} = 5.0 V		1.8	3.0	mA
						V_{DD} = 1.8 V		1.7	3.0	
				f_{MX} = 20 MHz **Note 4**, resonator connection	Normal operation	V_{DD} = 5.0 V		1.9	3.2	mA
						V_{DD} = 1.8 V		1.9	3.2	
				f_{MX} = 10 MHz **Note 4**, square wave input	Normal operation	V_{DD} = 5.0 V		0.9	1.6	mA
						V_{DD} = 1.8 V		0.9	1.6	
				f_{MX} = 10 MHz **Note 4**, resonator connection	Normal operation	V_{DD} = 5.0 V		1.0	1.7	mA
						V_{DD} = 1.8 V		1.0	1.7	
				f_{MX} = 8 MHz **Note 4**, square wave input	Normal operation	V_{DD} = 5.0 V		0.8	1.3	mA
						V_{DD} = 1.8 V		0.7	1.3	
				f_{MX} = 8 MHz **Note 4**, resonator connection	Normal operation	V_{DD} = 5.0 V		0.9	1.4	mA
						V_{DD} = 1.8 V		0.8	1.4	

Note 1. The listed currents are the total currents flowing into V_{DD} and EV_{DD0}, including the input leakage currents flowing when the level of the input pin is fixed to V_{DD}, EV_{DD0} or V_{SS}, EV_{SS0}. The following points apply in the HS (high-speed main), LS (low-speed main), and LP (low-power main) modes.
- The currents in the “Typ.” column do not include the operating currents of the peripheral modules.
- The currents in the “Max.” column include the operating currents of the PCLBUZ, TAU, SAU, and IICA modules.

<R> **Note 2.** The listed currents apply when the high-speed system clock, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock X are stopped.

<R> **Note 3.** The listed currents apply when the high-speed on-chip oscillator, high-speed system clock, low-speed on-chip oscillator, and subsystem clock X are stopped.

<R> **Note 4.** The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock X are stopped.

(Remarks are listed on the next page.)

Remark 1. f_{IH}: High-speed on-chip oscillator clock frequency
Remark 2. f_{IM}: Middle-speed on-chip oscillator clock frequency
Remark 3. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)
Remark 4. The typical value for the ambient operating temperature (T_A) is +25°C unless otherwise specified.

1. 30- to 64-pin R7F100G and 32-pin R7F104G products with 96- to 128-Kbyte code flash memory

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (2/4)

Item	Symbol	Conditions					Min.	Typ.	Max.	Unit
Supply current **Note 1**	I_{DD1}	Operating mode	Subsystem clock operation mode	f_{SUB} = 32.768 kHz **Note 2**, low-speed on-chip oscillator operation	Normal operation	T_A = -40°C		3.2	5.5	μA
						T_A = +25°C		3.5	5.8	
						T_A = +50°C		3.8	8.5	
						T_A = +70°C		4.4	13.8	
						T_A = +85°C		5.3	22.1	
						T_A = +105°C		7.7	40.9	
				f_{SUB} = 32.768 kHz **Note 3**, square wave input	Normal operation	T_A = -40°C		3.2	5.6	μA
						T_A = +25°C		3.4	5.7	
						T_A = +50°C		3.7	8.5	
						T_A = +70°C		4.3	13.7	
						T_A = +85°C		5.2	21.4	
						T_A = +105°C		7.6	39.0	
				f_{SUB} = 32.768 kHz **Note 3**, resonator connection	Normal operation	T_A = -40°C		3.2	5.2	μA
						T_A = +25°C		3.4	5.4	
						T_A = +50°C		3.7	7.7	
						T_A = +70°C		4.3	13.4	
						T_A = +85°C		5.2	20.9	
						T_A = +105°C		7.7	38.5	

Note 1. The listed currents are the total currents flowing into V_{DD} and EV_{DD0}, including the input leakage currents flowing when the level of the input pin is fixed to V_{DD}, EV_{DD0} or V_{SS}, EV_{SS0}. In the subsystem clock operation mode, the currents in both the "Typ." and "Max." columns do not include the operating currents of the peripheral modules.

<R> **Note 2.** The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, high-speed system clock, and subsystem clock X are stopped.

Note 3. The listed currents apply when the high-speed on-chip oscillator, high-speed system clock, middle-speed on-chip oscillator, and low-speed on-chip oscillator are stopped, and the low power consumption oscillation 3 is specified (AMPHS1, AMPHS0 = 11B).

Remark 1. f_{IL}: Low-speed on-chip oscillator clock frequency

<R> **Remark 2.** f_{SUB}: Subsystem clock frequency

1. 30- to 64-pin R7F100G and 32-pin R7F104G products with 96- to 128-Kbyte code flash memory

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (3/4)

Item	Symbol	Conditions				Min.	Typ.	Max.	Unit
Supply current Note 1	I_{DD2} Note 2	HALT mode	HS (high-speed main) mode	f_{IH} = 32 MHz Note 3	V_{DD} = 5.0 V		0.54	1.93	mA
					V_{DD} = 1.8 V		0.53	1.92	
			LS (low-speed main) mode	f_{IH} = 24 MHz Note 3	V_{DD} = 5.0 V		0.45	1.50	mA
					V_{DD} = 1.8 V		0.44	1.49	
				f_{IH} = 16 MHz Note 3	V_{DD} = 5.0 V		0.45	1.19	mA
					V_{DD} = 1.8 V		0.44	1.18	
				f_{IM} = 4 MHz Note 4	V_{DD} = 5.0 V		0.08	0.26	mA
					V_{DD} = 1.6 V		0.08	0.26	
			LP (low-power main) mode	f_{IM} = 2 MHz Note 4	V_{DD} = 5.0 V		33	120	μA
					V_{DD} = 1.6 V		33	120	
				f_{IM} = 1 MHz Note 4	V_{DD} = 5.0 V		29	76	μA
					V_{DD} = 1.6 V		28	74	
			HS (high-speed main) mode	f_{MX} = 20 MHz Note 5, square wave input	V_{DD} = 5.0 V		0.22	1.07	mA
					V_{DD} = 1.8 V		0.19	1.03	
			LS (low-speed main) mode	f_{MX} = 20 MHz Note 5, square wave input	V_{DD} = 5.0 V		0.22	1.07	mA
					V_{DD} = 1.8 V		0.19	1.03	
				f_{MX} = 20 MHz Note 5, resonator connection	V_{DD} = 5.0 V		0.40	1.28	mA
					V_{DD} = 1.8 V		0.39	1.27	
				f_{MX} = 10 MHz Note 5, square wave input	V_{DD} = 5.0 V		0.14	0.57	mA
					V_{DD} = 1.8 V		0.12	0.54	
				f_{MX} = 10 MHz Note 5, resonator connection	V_{DD} = 5.0 V		0.24	0.69	mA
					V_{DD} = 1.8 V		0.23	0.68	
				f_{MX} = 8 MHz Note 5, square wave input	V_{DD} = 5.0 V		0.12	0.47	mA
					V_{DD} = 1.8 V		0.10	0.44	
				f_{MX} = 8 MHz Note 5, resonator connection	V_{DD} = 5.0 V		0.21	0.58	mA
					V_{DD} = 1.8 V		0.20	0.57	

Note 1. The listed currents are the total currents flowing into V_{DD} and EV_{DD0}, including the input leakage currents flowing when the level of the input pin is fixed to V_{DD}, EV_{DD0} or V_{SS}, EV_{SS0}. The following points apply in the HS (high-speed main), LS (low-speed main), and LP (low-power main) modes.
- The currents in the “Typ.” column do not include the operating currents of the peripheral modules.
- The currents in the “Max.” column include the operating currents of the PCLBUZ, TAU, SAU, and IICA modules.

Note 2. The listed currents apply when the HALT instruction has been fetched from the flash memory for execution.

<R> **Note 3.** The listed currents apply when the high-speed system clock, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock X are stopped.

<R> **Note 4.** The listed currents apply when the high-speed on-chip oscillator, high-speed system clock, low-speed on-chip oscillator, and subsystem clock X are stopped.

<R> **Note 5.** The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock X are stopped.

(Remarks are listed on the next page.)

Remark 1. f_{IH}: High-speed on-chip oscillator clock frequency
Remark 2. f_{IM}: Middle-speed on-chip oscillator clock frequency
Remark 3. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)
Remark 4. The typical value for the ambient operating temperature (T_A) is +25°C unless otherwise specified.

1. 30- to 64-pin R7F100G and 32-pin R7F104G products with 96- to 128-Kbyte code flash memory

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (4/4)

Item	Symbol	Conditions				Min.	Typ.	Max.	Unit
Supply current **Note 1**	I_{DD2} **Note 2**	HALT mode	Subsystem clock operation mode	f_{SUB} = 32.768 kHz **Note 3**, low-speed on-chip oscillator operation	T_A = -40°C		0.53	2.31	μA
					T_A = +25°C		0.65	2.38	
					T_A = +50°C		0.80	4.95	
					T_A = +70°C		1.17	9.97	
					T_A = +85°C		1.78	17.96	
					T_A = +105°C		4.41	37.71	
				f_{SUB} = 32.768 kHz **Note 4**, square wave input	T_A = -40°C		0.20	1.97	μA
					T_A = +25°C		0.29	2.00	
					T_A = +50°C		0.54	5.33	
					T_A = +70°C		0.99	10.94	
					T_A = +85°C		1.70	19.62	
					T_A = +105°C		4.10	41.82	
				f_{SUB} = 32.768 kHz **Note 5**, resonator connection	T_A = -40°C		0.21	2.04	μA
					T_A = +25°C		0.33	2.28	
					T_A = +50°C		0.49	4.98	
					T_A = +70°C		1.05	11.36	
					T_A = +85°C		1.76	20.04	
					T_A = +105°C		4.20	42.52	
	I_{DD3}	STOP mode	RAMSDS = 0 **Note 6**		T_A = -40°C		0.15	1.45	μA
					T_A = +25°C		0.23	1.45	
					T_A = +50°C		0.45	4	
					T_A = +70°C		0.9	9	
					T_A = +85°C		1.6	17	
					T_A = +105°C		4	35	
			RAMSDS = 1 **Note 7**		T_A = -40°C		0.14	1.45	μA
					T_A = +25°C		0.21	1.45	
					T_A = +50°C		0.4	3.5	
					T_A = +70°C		0.8	8.5	
					T_A = +85°C		1.4	15	
					T_A = +105°C		3.2	30	
			RAMSDS = 1, 128-Hz realtime clock operation **Note 8**		T_A = -40°C		0.22	1.53	μA
					T_A = +25°C		0.32	1.56	
					T_A = +50°C		0.53	3.62	
					T_A = +70°C		0.94	8.64	
					T_A = +85°C		1.55	15.15	
					T_A = +105°C		3.40	30.20	

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

Note 1. The listed currents are the total currents flowing into V_{DD} and EV_{DD0}, including the input leakage currents flowing when the level of the input pin is fixed to V_{DD}, EV_{DD0} or V_{SS}, EV_{SS0}. In the subsystem clock operation mode or the STOP mode, the currents in both the “Typ.” and “Max.” columns do not include the operating currents of the peripheral modules.

Note 2. The listed currents apply when the HALT instruction has been fetched from the flash memory for execution.

<R> **Note 3.** The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, high-speed system clock, and subsystem clock X are stopped.

Note 4. The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, high-speed system clock, and low-speed on-chip oscillator are stopped.

Note 5. The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, high-speed system clock, and low-speed on-chip oscillator are stopped, and the setting of RTCLPC is 1, and the low power consumption oscillation 3 is specified (AMPHS1, AMPHS0 = 11B).

<R> **Note 6.** The listed currents with this setting allow retention of the contents of the entire RAM area. The listed currents apply when the low-speed on-chip oscillator and subsystem clock X oscillation are stopped. For the currents when operating the low-speed on-chip oscillator or subsystem clock X in the STOP mode, refer to the currents when operating the subsystem clock in the HALT mode.

<R> **Note 7.** The listed currents with this setting allow retention of the contents of a specified 4-Kbyte area of the RAM. The listed currents apply when the low-speed on-chip oscillator and subsystem clock X oscillation are stopped.

Note 8. The listed currents with this setting allow retention of the contents of a specified 4-Kbyte area of the RAM. The listed currents apply when the low-speed on-chip oscillator is stopped, the setting of RTCLPC is 1, and the low power consumption oscillation 3 is specified (AMPHS1, AMPHS0 = 11B). The current flowing into the RTC is included.

Remark 1. f_{IL}: Low-speed on-chip oscillator clock frequency

<R> **Remark 2.** f_{SUB}: Subsystem clock frequency

2. 30- to 64-pin R7F100G and 44- and 48-pin R7F104G products with 192- to 256-Kbyte code flash memory and 80-pin R7F100G products with 128- to 256-Kbyte code flash memory

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (1/4)

Item	Symbol	Conditions					Min.	Typ.	Max.	Unit
Supply current **Note 1**	I_{DD1}	Operating mode	HS (high-speed main) mode	f_{IH} = 32 MHz **Note 2**	Basic operation	V_{DD} = 5.0 V		1.4	—	mA
						V_{DD} = 1.8 V		1.4	—	
					Normal operation	V_{DD} = 5.0 V		3.0	5.0	mA
						V_{DD} = 1.8 V		3.0	5.0	
			LS (low-speed main) mode	f_{IH} = 24 MHz **Note 2**	Normal operation	V_{DD} = 5.0 V		2.3	3.8	mA
						V_{DD} = 1.8 V		2.3	3.8	
				f_{IH} = 16 MHz **Note 2**	Normal operation	V_{DD} = 5.0 V		1.7	2.8	mA
						V_{DD} = 1.8 V		1.7	2.7	
				f_{IM} = 4 MHz **Note 3**	Normal operation	V_{DD} = 5.0 V		0.4	0.7	mA
						V_{DD} = 1.6 V		0.4	0.7	
			LP (low-power main) mode	f_{IM} = 2 MHz **Note 3**	Normal operation	V_{DD} = 5.0 V		203	329	µA
						V_{DD} = 1.6 V		202	328	
				f_{IM} = 1 MHz **Note 3**	Normal operation	V_{DD} = 5.0 V		115	181	µA
						V_{DD} = 1.6 V		114	180	
			HS (high-speed main) mode	f_{MX} = 20 MHz **Note 4**, square wave input	Normal operation	V_{DD} = 5.0 V		1.9	3.2	mA
						V_{DD} = 1.8 V		1.9	3.2	
			LS (low-speed main) mode	f_{MX} = 20 MHz **Note 4**, square wave input	Normal operation	V_{DD} = 5.0 V		1.8	3.0	mA
						V_{DD} = 1.8 V		1.7	3.0	
				f_{MX} = 20 MHz **Note 4**, resonator connection	Normal operation	V_{DD} = 5.0 V		1.9	3.2	mA
						V_{DD} = 1.8 V		1.9	3.2	
				f_{MX} = 10 MHz **Note 4**, square wave input	Normal operation	V_{DD} = 5.0 V		0.9	1.6	mA
						V_{DD} = 1.8 V		0.9	1.6	
				f_{MX} = 10 MHz **Note 4**, resonator connection	Normal operation	V_{DD} = 5.0 V		1.0	1.7	mA
						V_{DD} = 1.8 V		1.0	1.7	
				f_{MX} = 8 MHz **Note 4**, square wave input	Normal operation	V_{DD} = 5.0 V		0.8	1.3	mA
						V_{DD} = 1.8 V		0.7	1.3	
				f_{MX} = 8 MHz **Note 4**, resonator connection	Normal operation	V_{DD} = 5.0 V		0.9	1.4	mA
						V_{DD} = 1.8 V		0.8	1.4	

Note 1. The listed currents are the total currents flowing into V_{DD} and EV_{DD0}, including the input leakage currents flowing when the level of the input pin is fixed to V_{DD}, EV_{DD0} or V_{SS}, EV_{SS0}. The following points apply in the HS (high-speed main), LS (low-speed main), and LP (low-power main) modes.
- The currents in the “Typ.” column do not include the operating currents of the peripheral modules.
- The currents in the “Max.” column include the operating currents of the PCLBUZ, TAU, SAU, and IICA modules.

<R> **Note 2.** The listed currents apply when the high-speed system clock, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock X are stopped.

<R> **Note 3.** The listed currents apply when the high-speed on-chip oscillator, high-speed system clock, low-speed on-chip oscillator, and subsystem clock X are stopped.

<R> **Note 4.** The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock X are stopped.

(Remarks are listed on the next page.)

Remark 1. f_{IH}: High-speed on-chip oscillator clock frequency
Remark 2. f_{IM}: Middle-speed on-chip oscillator clock frequency
Remark 3. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)
Remark 4. The typical value for the ambient operating temperature (T_A) is +25°C unless otherwise specified.

2. 30- to 64-pin R7F100G and 44- and 48-pin R7F104G products with 192- to 256-Kbyte code flash memory and 80-pin R7F100G products with 128- to 256-Kbyte code flash memory

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (2/4)

Item	Symbol	Conditions					Min.	Typ.	Max.	Unit
Supply current **Note 1**	I_{DD1}	Operating mode	Subsystem clock operation mode	f_{SUB} = 32.768 kHz **Note 2**, low-speed on-chip oscillator operation	Normal operation	T_A = -40°C		3.3	6.1	μA
						T_A = +25°C		3.6	6.3	
						T_A = +50°C		3.9	9.6	
						T_A = +70°C		4.5	15.9	
						T_A = +85°C		5.4	25.3	
						T_A = +105°C		7.8	56.3	
				f_{SUB} = 32.768 kHz **Note 3**, square wave input	Normal operation	T_A = -40°C		3.3	6.1	μA
						T_A = +25°C		3.5	6.4	
						T_A = +50°C		3.8	9.6	
						T_A = +70°C		4.4	16.1	
						T_A = +85°C		5.3	26.4	
						T_A = +105°C		7.8	57.0	
				f_{SUB} = 32.768 kHz **Note 3**, resonator connection	Normal operation	T_A = -40°C		3.3	6.0	μA
						T_A = +25°C		3.5	6.0	
						T_A = +50°C		3.8	8.9	
						T_A = +70°C		4.4	15.3	
						T_A = +85°C		5.3	25.6	
						T_A = +105°C		7.9	55.3	

Note 1. The listed currents are the total currents flowing into V_{DD} and EV_{DD0}, including the input leakage currents flowing when the level of the input pin is fixed to V_{DD}, EV_{DD0} or V_{SS}, EV_{SS0}. In the subsystem clock operation mode, the currents in both the "Typ." and "Max." columns do not include the operating currents of the peripheral modules.

<R> **Note 2.** The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, high-speed system clock, and subsystem clock X are stopped.

Note 3. The listed currents apply when the high-speed on-chip oscillator, high-speed system clock, middle-speed on-chip oscillator, and low-speed on-chip oscillator are stopped, and the low power consumption oscillation 3 is specified (AMPHS1, AMPHS0 = 11B).

Remark 1. f_{IL}: Low-speed on-chip oscillator clock frequency

<R> **Remark 2.** f_{SUB}: Subsystem clock frequency

2. 30- to 64-pin R7F100G and 44- and 48-pin R7F104G products with 192- to 256-Kbyte code flash memory and 80-pin R7F100G products with 128- to 256-Kbyte code flash memory

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (3/4)

Item	Symbol	Conditions				Min.	Typ.	Max.	Unit
Supply current**Note 1**	I_{DD2} **Note 2**	HALT mode	HS (high-speed main) mode	f_{IH} = 32 MHz**Note 3**	V_{DD} = 5.0 V		0.57	1.97	mA
					V_{DD} = 1.8 V		0.56	1.96	
			LS (low-speed main) mode	f_{IH} = 24 MHz**Note 3**	V_{DD} = 5.0 V		0.47	1.53	mA
					V_{DD} = 1.8 V		0.47	1.52	
				f_{IH} = 16 MHz**Note 3**	V_{DD} = 5.0 V		0.48	1.22	mA
					V_{DD} = 1.8 V		0.47	1.21	
				f_{IM} = 4 MHz**Note 4**	V_{DD} = 5.0 V		0.08	0.27	mA
					V_{DD} = 1.6 V		0.08	0.26	
			LP (low-power main) mode	f_{IM} = 2 MHz**Note 4**	V_{DD} = 5.0 V		38	126	μA
					V_{DD} = 1.6 V		37	125	
				f_{IM} = 1 MHz**Note 4**	V_{DD} = 5.0 V		32	79	μA
					V_{DD} = 1.6 V		32	79	
			HS (high-speed main) mode	f_{MX} = 20 MHz**Note 5**, square wave input	V_{DD} = 5.0 V		0.23	1.07	mA
					V_{DD} = 1.8 V		0.19	1.03	
			LS (low-speed main) mode	f_{MX} = 20 MHz**Note 5**, square wave input	V_{DD} = 5.0 V		0.23	1.07	mA
					V_{DD} = 1.8 V		0.19	1.03	
				f_{MX} = 20 MHz**Note 5**, resonator connection	V_{DD} = 5.0 V		0.41	1.30	mA
					V_{DD} = 1.8 V		0.40	1.28	
				f_{MX} = 10 MHz**Note 5**, square wave input	V_{DD} = 5.0 V		0.14	0.57	mA
					V_{DD} = 1.8 V		0.12	0.54	
				f_{MX} = 10 MHz**Note 5**, resonator connection	V_{DD} = 5.0 V		0.24	0.69	mA
					V_{DD} = 1.8 V		0.23	0.68	
				f_{MX} = 8 MHz**Note 5**, square wave input	V_{DD} = 5.0 V		0.12	0.47	mA
					V_{DD} = 1.8 V		0.10	0.44	
				f_{MX} = 8 MHz**Note 5**, resonator connection	V_{DD} = 5.0 V		0.21	0.58	mA
					V_{DD} = 1.8 V		0.20	0.57	

Note 1. The listed currents are the total currents flowing into V_{DD} and EV_{DD0}, including the input leakage currents flowing when the level of the input pin is fixed to V_{DD}, EV_{DD0} or V_{SS}, EV_{SS0}. The following points apply in the HS (high-speed main), LS (low-speed main), and LP (low-power main) modes.
- The currents in the “Typ.” column do not include the operating currents of the peripheral modules.
- The currents in the “Max.” column include the operating currents of the PCLBUZ, TAU, SAU, and IICA modules.

Note 2. The listed currents apply when the HALT instruction has been fetched from the flash memory for execution.

<R> **Note 3.** The listed currents apply when the high-speed system clock, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock X are stopped.

<R> **Note 4.** The listed currents apply when the high-speed on-chip oscillator, high-speed system clock, low-speed on-chip oscillator, and subsystem clock X are stopped.

<R> **Note 5.** The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock X are stopped.

(Remarks are listed on the next page.)

Remark 1. f_{IH}: High-speed on-chip oscillator clock frequency
Remark 2. f_{IM}: Middle-speed on-chip oscillator clock frequency
Remark 3. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)
Remark 4. The typical value for the ambient operating temperature (T_A) is +25°C unless otherwise specified.

2. 30- to 64-pin R7F100G and 44- and 48-pin R7F104G products with 192- to 256-Kbyte code flash memory and 80-pin R7F100G products with 128- to 256-Kbyte code flash memory

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = 0 V) (4/4)

Item	Symbol	Conditions				Min.	Typ.	Max.	Unit
Supply current **Note 1**	I_{DD2} **Note 2**	HALT mode	Subsystem clock operation mode	f_{SUB} = 32.768 kHz **Note 3**, low-speed on-chip oscillator operation	T_A = -40°C		0.62	2.94	μA
					T_A = +25°C		0.74	3.00	
					T_A = +50°C		0.88	6.00	
					T_A = +70°C		1.22	12.01	
					T_A = +85°C		2.69	22.92	
					T_A = +105°C		5.08	54.47	
				f_{SUB} = 32.768 kHz, square wave input **Note 4**	T_A = -40°C		0.25	2.54	μA
					T_A = +25°C		0.37	2.73	
					T_A = +50°C		0.74	7.35	
					T_A = +70°C		1.33	15.13	
					T_A = +85°C		2.35	27.33	
					T_A = +105°C		4.81	62.95	
				f_{SUB} = 32.768 kHz, resonator connection **Note 5**	T_A = -40°C		0.27	2.68	μA
					T_A = +25°C		0.39	2.87	
					T_A = +50°C		0.78	7.63	
					T_A = +70°C		1.34	15.20	
					T_A = +85°C		2.35	27.33	
					T_A = +105°C		4.67	61.97	
	I_{DD3}	STOP mode	RAMSDS = 0 **Note 6**		T_A = -40°C		0.19	2.00	μA
					T_A = +25°C		0.30	2.00	
					T_A = +50°C		0.65	5.00	
					T_A = +70°C		1.20	11.00	
					T_A = +85°C		2.20	20.00	
					T_A = +105°C		4.50	50.00	
			RAMSDS = 1 **Note 7**		T_A = -40°C		0.18	2.00	μA
					T_A = +25°C		0.29	2.00	
					T_A = +50°C		0.60	4.50	
					T_A = +70°C		1.10	10.00	
					T_A = +85°C		2.00	19.00	
					T_A = +105°C		4.00	45.00	
			RAMSDS = 1, 128-Hz realtime clock operation **Note 8**		T_A = -40°C		0.23	2.05	μA
					T_A = +25°C		0.40	2.11	
					T_A = +50°C		0.72	4.62	
					T_A = +70°C		1.23	10.13	
					T_A = +85°C		2.14	19.14	
					T_A = +105°C		4.16	45.16	

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

Note 1. The listed currents are the total currents flowing into V_{DD} and EV_{DD0}, including the input leakage currents flowing when the level of the input pin is fixed to V_{DD}, EV_{DD0} or V_{SS}, EV_{SS0}. In the subsystem clock operation mode or the STOP mode, the currents in both the “Typ.” and “Max.” columns do not include the operating currents of the peripheral modules.

Note 2. The listed currents apply when the HALT instruction has been fetched from the flash memory for execution.

<R> **Note 3.** The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, high-speed system clock, and subsystem clock X are stopped.

Note 4. The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, high-speed system clock, and low-speed on-chip oscillator are stopped.

Note 5. The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, high-speed system clock, and low-speed on-chip oscillator are stopped, and the setting of RTCLPC is 1, and the low power consumption oscillation 3 is specified (AMPHS1, AMPHS0 = 11B).

<R> **Note 6.** The listed currents with this setting allow retention of the contents of the entire RAM area. The listed currents apply when the low-speed on-chip oscillator and subsystem clock X oscillation are stopped. For the currents when operating the low-speed on-chip oscillator or subsystem clock X in the STOP mode, refer to the currents when operating the subsystem clock in the HALT mode.

<R> **Note 7.** The listed currents with this setting allow retention of the contents of a specified 4-Kbyte area of the RAM. The listed currents apply when the low-speed on-chip oscillator and subsystem clock X oscillation are stopped.

Note 8. The listed currents with this setting allow retention of the contents of a specified 4-Kbyte area of the RAM. The listed currents apply when the low-speed on-chip oscillator is stopped, the setting of RTCLPC is 1, and the low power consumption oscillation 3 is specified (AMPHS1, AMPHS0 = 11B). The current flowing into the RTC is included.

Remark 1. f_{IL}: Low-speed on-chip oscillator clock frequency

<R> **Remark 2.** f_{SUB}: Subsystem clock frequency

3. 44- to 80-pin R7F100G products with 384- to 768-Kbyte code flash memory and 100- and 128-pin R7F100G products

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (1/4)

Item	Symbol	Conditions					Min.	Typ.	Max.	Unit
Supply current **Note 1**	I_{DD1}	Operating mode	HS (high-speed main) mode	f_{IH} = 32 MHz **Note 2**	Basic operation	V_{DD} = 5.0 V		1.6	—	mA
						V_{DD} = 1.8 V		1.5	—	
					Normal operation	V_{DD} = 5.0 V		3.5	5.6	mA
						V_{DD} = 1.8 V		3.5	5.6	
			LS (low-speed main) mode	f_{IH} = 24 MHz **Note 2**	Normal operation	V_{DD} = 5.0 V		2.6	4.2	mA
						V_{DD} = 1.8 V		2.6	4.2	
				f_{IH} = 16 MHz **Note 2**	Normal operation	V_{DD} = 5.0 V		2.0	3.1	mA
						V_{DD} = 1.8 V		1.9	3.1	
				f_{IM} = 4 MHz **Note 3**	Normal operation	V_{DD} = 5.0 V		0.5	0.9	mA
						V_{DD} = 1.6 V		0.5	0.8	
			LP (low-power main) mode	f_{IM} = 2 MHz **Note 3**	Normal operation	V_{DD} = 5.0 V		229	361	µA
						V_{DD} = 1.6 V		227	358	
				f_{IM} = 1 MHz **Note 3**	Normal operation	V_{DD} = 5.0 V		128	197	µA
						V_{DD} = 1.6 V		125	193	
			HS (high-speed main) mode	f_{MX} = 20 MHz **Note 4**, square wave input	Normal operation	V_{DD} = 5.0 V		2.2	3.5	mA
						V_{DD} = 1.8 V		2.2	3.5	
			LS (low-speed main) mode	f_{MX} = 20 MHz **Note 4**, square wave input	Normal operation	V_{DD} = 5.0 V		2.1	3.4	mA
						V_{DD} = 1.8 V		2.0	3.3	
				f_{MX} = 20 MHz **Note 4**, resonator connection	Normal operation	V_{DD} = 5.0 V		2.2	3.6	mA
						V_{DD} = 1.8 V		2.2	3.5	
				f_{MX} = 10 MHz **Note 4**, square wave input	Normal operation	V_{DD} = 5.0 V		1.1	1.8	mA
						V_{DD} = 1.8 V		1.1	1.8	
				f_{MX} = 10 MHz **Note 4**, resonator connection	Normal operation	V_{DD} = 5.0 V		1.2	1.9	mA
						V_{DD} = 1.8 V		1.2	1.9	
				f_{MX} = 8 MHz **Note 4**, square wave input	Normal operation	V_{DD} = 5.0 V		0.9	1.5	mA
						V_{DD} = 1.8 V		0.9	1.5	
				f_{MX} = 8 MHz **Note 4**, resonator connection	Normal operation	V_{DD} = 5.0 V		1.0	1.6	mA
						V_{DD} = 1.8 V		1.0	1.6	

Note 1. The listed currents are the total currents flowing into V_{DD}, EV_{DD0} and EV_{DD1}, including the input leakage currents flowing when the level of the input pin is fixed to V_{DD}, EV_{DD0}, EV_{DD1} or V_{SS}, EV_{SS0}, EV_{SS1}. The following points apply in the HS (high-speed main), LS (low-speed main), and LP (low-power main) modes.
- The currents in the “Typ.” column do not include the operating currents of the peripheral modules.
- The currents in the “Max.” column include the operating currents of the PCLBUZ, TAU, SAU, and IICA modules.

<R> **Note 2.** The listed currents apply when the high-speed system clock, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock X are stopped.

<R> **Note 3.** The listed currents apply when the high-speed on-chip oscillator, high-speed system clock, low-speed on-chip oscillator, and subsystem clock X are stopped.

<R> **Note 4.** The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock X are stopped.

(Remarks are listed on the next page.)

Remark 1. f_{IH}: High-speed on-chip oscillator clock frequency
Remark 2. f_{IM}: Middle-speed on-chip oscillator clock frequency
Remark 3. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)
Remark 4. The typical value for the ambient operating temperature (T_A) is +25°C unless otherwise specified.

3. 44- to 80-pin R7F100G products with 384- to 768-Kbyte code flash memory and 100- and 128-pin R7F100G products

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (2/4)

Item	Symbol	Conditions					Min.	Typ.	Max.	Unit
Supply current **Note 1**	I_{DD1}	Operating mode	Subsystem clock operation mode	f_{SUB} = 32.768 kHz **Note 2**, low-speed on-chip oscillator operation	Normal operation	T_A = -40°C		3.8	7.7	μA
						T_A = +25°C		4.1	8.0	
						T_A = +50°C		4.6	13.5	
						T_A = +70°C		5.6	24.0	
						T_A = +85°C		7.1	40.8	
						T_A = +105°C		11.1	88.8	
				f_{SUB} = 32.768 kHz **Note 3**, square wave input	Normal operation	T_A = -40°C		3.8	7.7	μA
						T_A = +25°C		4.0	8.0	
						T_A = +50°C		4.5	13.6	
						T_A = +70°C		5.3	24.1	
						T_A = +85°C		6.7	40.3	
						T_A = +105°C		10.7	88.1	
				f_{SUB} = 32.768 kHz **Note 3**, resonator connection	Normal operation	T_A = -40°C		3.8	7.4	μA
						T_A = +25°C		4.1	7.8	
						T_A = +50°C		4.5	12.6	
						T_A = +70°C		5.4	24.2	
						T_A = +85°C		6.8	39.8	
						T_A = +105°C		10.8	87.4	

Note 1. The listed currents are the total currents flowing into V_{DD}, EV_{DD0} and EV_{DD1}, including the input leakage currents flowing when the level of the input pin is fixed to V_{DD}, EV_{DD0}, EV_{DD1} or V_{SS}, EV_{SS0}, EV_{SS1}. In the subsystem clock operation mode, the currents in both the “Typ.” and “Max.” columns do not include the operating currents of the peripheral modules.

<R> **Note 2.** The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, high-speed system clock, and subsystem clock X are stopped.

Note 3. The listed currents apply when the high-speed on-chip oscillator, high-speed system clock, middle-speed on-chip oscillator, and low-speed on-chip oscillator are stopped, and the low power consumption oscillation 3 is specified (AMPHS1, AMPHS0 = 11B).

Remark 1. f_{IL}: Low-speed on-chip oscillator clock frequency

<R> **Remark 2.** f_{SUB}: Subsystem clock frequency

3. 44- to 80-pin R7F100G products with 384- to 768-Kbyte code flash memory and 100- and 128-pin R7F100G products

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (3/4)

Item	Symbol	Conditions				Min.	Typ.	Max.	Unit
Supply current**Note 1**	I_{DD2} **Note 2**	HALT mode	HS (high-speed main) mode	f_{IH} = 32 MHz**Note 3**	V_{DD} = 5.0 V		0.60	2.00	mA
					V_{DD} = 1.8 V		0.59	1.99	
			LS (low-speed main) mode	f_{IH} = 24 MHz**Note 3**	V_{DD} = 5.0 V		0.49	1.56	mA
					V_{DD} = 1.8 V		0.48	1.55	
				f_{IH} = 16 MHz**Note 3**	V_{DD} = 5.0 V		0.49	1.24	mA
					V_{DD} = 1.8 V		0.48	1.23	
				f_{IM} = 4 MHz**Note 4**	V_{DD} = 5.0 V		0.09	0.28	mA
					V_{DD} = 1.6 V		0.09	0.27	
			LP (low-power main) mode	f_{IM} = 2 MHz**Note 4**	V_{DD} = 5.0 V		40	129	μA
					V_{DD} = 1.6 V		37	125	
				f_{IM} = 1 MHz**Note 4**	V_{DD} = 5.0 V		33	80	μA
					V_{DD} = 1.6 V		32	79	
			HS (high-speed main) mode	f_{MX} = 20 MHz**Note 5**, square wave input	V_{DD} = 5.0 V		0.25	1.10	mA
					V_{DD} = 1.8 V		0.21	1.05	
			LS (low-speed main) mode	f_{MX} = 20 MHz**Note 5**, square wave input	V_{DD} = 5.0 V		0.25	1.10	mA
					V_{DD} = 1.8 V		0.21	1.05	
				f_{MX} = 20 MHz**Note 5**, resonator connection	V_{DD} = 5.0 V		0.41	1.30	mA
					V_{DD} = 1.8 V		0.40	1.28	
				f_{MX} = 10 MHz**Note 5**, square wave input	V_{DD} = 5.0 V		0.15	0.59	mA
					V_{DD} = 1.8 V		0.13	0.55	
				f_{MX} = 10 MHz**Note 5**, resonator connection	V_{DD} = 5.0 V		0.25	0.70	mA
					V_{DD} = 1.8 V		0.24	0.69	
				f_{MX} = 8 MHz**Note 5**, square wave input	V_{DD} = 5.0 V		0.13	0.48	mA
					V_{DD} = 1.8 V		0.11	0.45	
				f_{MX} = 8 MHz**Note 5**, resonator connection	V_{DD} = 5.0 V		0.22	0.59	mA
					V_{DD} = 1.8 V		0.21	0.58	

Note 1. The listed currents are the total currents flowing into V_{DD}, EV_{DD0} and EV_{DD1}, including the input leakage currents flowing when the level of the input pin is fixed to V_{DD}, EV_{DD0}, EV_{DD1} or V_{SS}, EV_{SS0}, EV_{SS1}. The following points apply in the HS (high-speed main), LS (low-speed main), and LP (low-power main) modes.
- The currents in the “Typ.” column do not include the operating currents of the peripheral modules.
- The currents in the “Max.” column include the operating currents of the PCLBUZ, TAU, SAU, and IICA modules.

Note 2. The listed currents apply when the HALT instruction has been fetched from the flash memory for execution.

<R> **Note 3.** The listed currents apply when the high-speed system clock, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock X are stopped.

<R> **Note 4.** The listed currents apply when the high-speed on-chip oscillator, high-speed system clock, low-speed on-chip oscillator, and subsystem clock X are stopped.

<R> **Note 5.** The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, low-speed on-chip oscillator, and subsystem clock X are stopped.

(Remarks are listed on the next page.)

Remark 1. f_{IH}: High-speed on-chip oscillator clock frequency
Remark 2. f_{IM}: Middle-speed on-chip oscillator clock frequency
Remark 3. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)
Remark 4. The typical value for the ambient operating temperature (T_A) is +25°C unless otherwise specified.

3. 44- to 80-pin R7F100G products with 384- to 768-Kbyte code flash memory and 100- and 128-pin R7F100G products

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (4/4)

Item	Symbol	Conditions				Min.	Typ.	Max.	Unit
Supply current **Note 1**	I_{DD2} **Note 2**	HALT mode	Subsystem clock operation mode	f_{SUB} = 32.768 kHz **Note 3**, low-speed on-chip oscillator operation	T_A = -40°C		0.62	3.95	μA
					T_A = +25°C		0.78	4.00	
					T_A = +50°C		1.03	9.16	
					T_A = +70°C		1.62	19.34	
					T_A = +85°C		3.50	37.35	
					T_A = +105°C		6.77	85.36	
				f_{SUB} = 32.768 kHz, square wave input **Note 4**	T_A = -40°C		0.25	3.55	μA
					T_A = +25°C		0.41	3.73	
					T_A = +50°C		0.90	10.93	
					T_A = +70°C		1.76	23.42	
					T_A = +85°C		2.92	41.07	
					T_A = +105°C		6.27	94.30	
				f_{SUB} = 32.768 kHz, resonator connection **Note 5**	T_A = -40°C		0.27	3.62	μA
					T_A = +25°C		0.43	3.87	
					T_A = +50°C		0.92	11.07	
					T_A = +70°C		1.79	23.63	
					T_A = +85°C		2.94	41.21	
					T_A = +105°C		6.28	94.37	
	I_{DD3}	STOP mode	RAMSDS = 0 **Note 6**		T_A = -40°C		0.21	3.00	μA
					T_A = +25°C		0.35	3.00	
					T_A = +50°C		0.75	8.00	
					T_A = +70°C		1.60	18.00	
					T_A = +85°C		2.80	34.00	
					T_A = +105°C		6.00	80.00	
			RAMSDS = 1 **Note 7**		T_A = -40°C		0.19	3.00	μA
					T_A = +25°C		0.32	3.00	
					T_A = +50°C		0.65	7.00	
					T_A = +70°C		1.25	17.00	
					T_A = +85°C		2.10	30.00	
					T_A = +105°C		4.50	70.00	
			RAMSDS = 1, 128-Hz realtime clock operation **Note 8**		T_A = -40°C		0.27	3.08	μA
					T_A = +25°C		0.42	3.10	
					T_A = +50°C		0.76	7.11	
					T_A = +70°C		1.38	17.13	
					T_A = +85°C		2.23	30.13	
					T_A = +105°C		4.64	70.14	

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

Note 1. The listed currents are the total currents flowing into V_{DD}, EV_{DD0} and EV_{DD1}, including the input leakage currents flowing when the level of the input pin is fixed to V_{DD}, EV_{DD0}, EV_{DD1} or V_{SS}, EV_{SS0}, EV_{SS1}. In the subsystem clock operation mode or the STOP mode, the currents in both the “Typ.” and “Max.” columns do not include the operating currents of the peripheral modules.

Note 2. The listed currents apply when the HALT instruction has been fetched from the flash memory for execution.

<R> **Note 3.** The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, high-speed system clock, and subsystem clock X are stopped.

Note 4. The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, high-speed system clock, and low-speed on-chip oscillator are stopped.

Note 5. The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, high-speed system clock, and low-speed on-chip oscillator are stopped, and the setting of RTCLPC is 1, and the low power consumption oscillation 3 is specified (AMPHS1, AMPHS0 = 11B).

<R> **Note 6.** The listed currents with this setting allow retention of the contents of the entire RAM area. The listed currents apply when the low-speed on-chip oscillator and subsystem clock X oscillation are stopped. For the currents when operating the low-speed on-chip oscillator or subsystem clock X in the STOP mode, refer to the currents when operating the subsystem clock in the HALT mode.

<R> **Note 7.** The listed currents with this setting allow retention of the contents of a specified 4-Kbyte area of the RAM. The listed currents apply when the low-speed on-chip oscillator and subsystem clock X oscillation are stopped.

Note 8. The listed currents with this setting allow retention of the contents of a specified 4-Kbyte area of the RAM. The listed currents apply when the low-speed on-chip oscillator is stopped, the setting of RTCLPC is 1, and the low power consumption oscillation 3 is specified (AMPHS1, AMPHS0 = 11B). The current flowing into the RTC is included.

Remark 1. f_{IL}: Low-speed on-chip oscillator clock frequency

<R> **Remark 2.** f_{SUB}: Subsystem clock frequency

4. Peripheral Functions (Common to all products)

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (1/2)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
High-speed on-chip oscillator operating current	I_{FIH} **Note 1**				380		μA
Middle-speed on-chip oscillator operating current	I_{FIM} **Note 1**				20		μA
Low-speed on-chip oscillator operating current	I_{FIL} **Note 1**				0.3		μA
RTC operating current	I_{RTC} **Notes 1, 2, 3**	f_{RTCCLK} = 32.768 kHz			0.005		μA
		f_{RTCCLK} = 128 Hz			0.002		μA
32-bit interval timer operating current	I_{IT} **Notes 1, 2, 4**				0.04		μA
Watchdog timer operating current	I_{WDT} **Notes 1, 2, 5**	f_{IL} = 32.768 kHz (typ.)			0.32		μA
A/D converter operating current	I_{ADC} **Notes 1, 6**	When conversion at maximum speed	Normal mode, AV_{REFP} = V_{DD} = 5.0 V		0.95	1.6	mA
			Low voltage mode, AV_{REFP} = V_{DD} = 3.0 V		0.5	0.75	mA
AV_{REFP} current	I_{ADREF} **Note 7**	AV_{REFP} = 5.0 V			52		μA
A/D converter internal reference voltage current	I_{ADREF} **Note 1**				114		μA
Temperature sensor operating current	I_{TMPS} **Note 1**				110		μA
D/A converter operating current	I_{DAC} **Notes 1, 8**	Per channel			150		μA
Comparator operating current	I_{CMP} **Notes 1, 9**				6		μA
<R> Serial interface UARTA operating current	I_{UARTA} **Notes 1, 2, 10**	f_{SXP} = 32.768 kHz	1 channel: Reception at 2400 bps		0.14		μA
LVD operating current	I_{LVD0} **Notes 1, 11**				0.02		μA
	I_{LVD1} **Notes 1, 11**				0.02		μA
Self-programming operating current	I_{FSP} **Notes 1, 12**				2.5	12.2	mA
Data flash memory rewrite operating current	I_{BGO} **Notes 1, 13**				2.5	12.2	mA

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (2/2)

Item	Symbol	Conditions			Min.	Typ.	Max.	Unit
SNOOZE mode sequencer operating current	I_{SMS} **Notes 1, 14**	f_{IH} = 32 MHz		30- to 64-pin R7F100G and 32-pin R7F104G products with 96- to 128-Kbyte code flash memory		1.1		mA
				30- to 64-pin R7F100G and 44- and 48-pin R7F104G products with 192- to 256-Kbyte code flash memory and 80-pin R7F100G products with 128- to 256-Kbyte code flash memory		1.1		
				44- to 80-pin R7F100G products with 384- to 768-Kbyte code flash memory and 100- and 128-pin R7F100G products		1.4		
		f_{IL} = 32.768 kHz		30- to 64-pin R7F100G products with 96- to 128-Kbyte code flash memory		1.2		μA
				30- to 64-pin R7F100G and 44- and 48-pin R7F104G products with 192- to 256-Kbyte code flash memory and 80-pin R7F100G products with 128- to 256-Kbyte code flash memory		1.2		
				44- to 80-pin R7F100G products with 384- to 768-Kbyte code flash memory and 100- and 128-pin R7F100G products		1.6		
SNOOZE operating current	I_{SNOZ} **Note 1**	f_{IH} = 32 MHz	ADC to be in use	The ADC is shifting from the STOP mode to the SNOOZE mode. **Note 15**		0.6	0.81	mA
				The ADC is operating in the low-voltage mode. AV_{REFP} = V_{DD} = 3.0 V		1.2	1.56	
			Simplified SPI (CSI)/UART to be in use			0.7	0.92	mA
			SMS **Note 16**	30- to 64-pin R7F100G and 32-pin R7F104G products with 96- to 128-Kbyte code flash memory		1.6		mA
				30- to 64-pin R7F100G and 44- and 48-pin R7F104G products with 192- to 256-Kbyte code flash memory and 80-pin R7F100G products with 128- to 256-Kbyte code flash memory		1.7		
				44- to 80-pin R7F100G products with 384- to 768-Kbyte code flash memory, and 100- and 128-pin R7F100G products		2.0		
Remote control signal receiver operating current	I_{REM} **Notes 1, 17**					0.03		μA
Low-speed peripheral clock supply current	I_{SXP} **Notes 1, 18**	RTCLPC = 0				0.22		μA
Output current control operating current	I_{CCDA} **Notes 1, 19**	The setting of the CCDE register is not 00H.				100		μA
	I_{CCDP} **Notes 20, 21**	Per single output current control port		Setting of the low-level output current: Hi-Z		30		μA
				Setting of the low-level output current: 2 to 15 mA		200		μA
Operating current of the true random number generator	I_{TRNG} **Note 1**					1.1		mA

(Notes and Remarks continue on the next page.)

Note 1. This current flows into V_{DD}.

Note 2. The listed currents apply when the high-speed on-chip oscillator, middle-speed on-chip oscillator, and high-speed system clock are stopped.

Note 3. This current flows into the realtime clock (RTC). It does not include the operating current of the low-speed on-chip oscillator or the XT1 oscillator. The supply current of this MCU is the sum of either I_{DD1} or I_{DD2}, and I_{RTC}, when the realtime clock is operating or in the HALT mode. When the low-speed on-chip oscillator is selected, I_{FIL} should be included in the supply current. I_{DD2} in the subsystem clock operation mode includes the operating current of the realtime clock.

Note 4. This current only flows to the 32-bit interval timer. It does not include the operating current of the low-speed on-chip oscillator or the XT1 oscillator. The supply current of this MCU is the sum of either I_{DD1} or I_{DD2}, and I_{IT}, when the 32-bit interval timer is operating or in the HALT mode. When the low-speed on-chip oscillator is selected, I_{FIL} should be included in the supply current.

Note 5. This current only flows to the watchdog timer. It includes the operating current of the low-speed on-chip oscillator. The supply current of this MCU is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{WDT} when the watchdog timer is operating.

Note 6. This current only flows to the A/D converter. The supply current of this MCU is the sum of I_{DD1} or I_{DD2} and I_{ADC} when the A/D converter is operating or in the HALT mode.

Note 7. This current flows into AV_{REFP}.

Note 8. This current only flows to the D/A converter. The supply current of this MCU is the sum of the values of either I_{DD1} or I_{DD2}, and I_{DAC}, when the D/A converter is operating or in the HALT mode.

Note 9. This current only flows to the comparator. The supply current of this MCU is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{CMP} when the comparator is in operation.

Note 10. This current only flows to the serial interface UARTA. It does not include the operating current of the low-speed on-chip oscillator or the XT1 oscillator. The supply current of this MCU is the sum of I_{DD1} or I_{DD2} and I_{UARTA} when the serial interface UARTA is operating or in the HALT mode. Note that when the UARTA is in the HALT mode with the subsystem clock X (f_{SX}) selected as the CPU/peripheral hardware clock (f_{CLK}), the low-speed peripheral clock supply current (I_{SXP}) is also added to the supply current.

Note 11. This current only flows to the LVD. The supply current of this MCU is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{LVD} when the LVD is in operation.

Note 12. This current only flows during self programming.

Note 13. This current only flows while the data flash memory is being rewritten.

Note 14. This current only flows into the SNOOZE mode sequencer. Note that the operating current of the low-speed on-chip oscillator and the XT1 oscillator are not included. The supply current of this MCU is the sum of either I_{DD1} or I_{DD2}, and I_{SMS}, when the SNOOZE mode sequencer is operating or in the HALT mode.

Note 15. For shift time to the SNOOZE mode, see **23.3.3 SNOOZE mode**.

Note 16. The listed values apply when the SNOOZE mode sequencer is in normal operation equivalent to I_{DD1}. They do not include the current flowing into the peripheral functions other than the SNOOZE mode sequencer.

Note 17. This current flows into the remote control signal receiver. It does not include the operating current of the low-speed on-chip oscillator or the XT1 oscillator. The supply current of this MCU is the sum of either I_{DD1} or I_{DD2}, and I_{IT}, when the remote control signal receiver is operating or in the HALT mode. When the low-speed on-chip oscillator is selected, I_{FIL} should be included in the supply current.

Note 18. This current is added to the supply current in the HALT mode when the setting of RTCLPC is 0 in the STOP mode, or when the setting of RTCLPC is 0 with the subsystem clock X (f_{SX}) selected as the CPU clock, while the subsystem clock X (f_{SX}) is oscillating.

Note 19. This current is added to the supply current when the output voltage control port is set.

Note 20. This current does not include the current flowing into the I/O port pins.

Note 21. This current only flows to EV_{DD0} and EV_{DD1}.

Remark 1. f_{IL}: Low-speed on-chip oscillator clock frequency

Remark 2. f_{SX}: Subsystem clock X frequency

Remark 3. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 4. The typical value for the ambient operating temperature (T_A) is +25°C unless otherwise specified.

37.4 AC Characteristics

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V)

Item	Symbol	Conditions			Min.	Typ.	Max.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation	HS (high-speed main) mode	1.8 V ≤ V_{DD} ≤ 5.5 V	0.03125		1	μs
				1.6 V ≤ V_{DD} ≤ 1.8 V	0.25		1	μs
			LS (low-speed main) mode	1.8 V ≤ V_{DD} ≤ 5.5 V	0.04167		1	μs
				1.6 V ≤ V_{DD} ≤ 1.8 V	0.25		1	μs
			LP (low-power main) mode	1.6 V ≤ V_{DD} ≤ 5.5 V	0.5		1	μs
		Subsystem clock (f_{SUB}) operation		1.6 V ≤ V_{DD} ≤ 5.5 V	26.041	30.5	31.3	μs
		In the self-programming mode	HS (high-speed main) mode	1.8 V ≤ V_{DD} ≤ 5.5 V	0.03125		1	μs
				1.6 V ≤ V_{DD} ≤ 1.8 V	0.5		1	μs
			LS (low-speed main) mode	1.8 V ≤ V_{DD} ≤ 5.5 V	0.04167		1	μs
				1.6 V ≤ V_{DD} ≤ 1.8 V	0.5		1	μs
External system clock frequency	f_{EX}	1.8 V ≤ V_{DD} ≤ 5.5 V			1.0		20.0	MHz
		1.6 V ≤ V_{DD} < 1.8 V			1.0		4.0	MHz
	f_{EXS}				32		38.4	kHz
External system clock input high-level width, low-level width	t_{EXH}, t_{EXL}	1.8 V ≤ V_{DD} ≤ 5.5 V			24			ns
		1.6 V ≤ V_{DD} < 1.8 V			120			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[Note]
TO00 to TO07, TO10 to TO17 output frequency	f_{TO}	HS (high-speed main) mode LS (low-speed main) mode		4.0 V ≤ EV_{DD0} ≤ 5.5 V			16	MHz
				2.7 V ≤ EV_{DD0} < 4.0 V			8	MHz
				1.8 V ≤ EV_{DD0} < 2.7 V			4	MHz
				1.6 V ≤ EV_{DD0} < 1.8 V			2	MHz
		LP (low-power main) mode		1.6 V ≤ EV_{DD0} ≤ 5.5 V			2	MHz
PCLBUZ0, PCLBUZ1 output frequency	f_{PCL}	HS (high-speed main) mode LS (low-speed main) mode		4.0 V ≤ EV_{DD0} ≤ 5.5 V			16	MHz
				2.7 V ≤ EV_{DD0} < 4.0 V			8	MHz
				1.8 V ≤ EV_{DD0} < 2.7 V			4	MHz
				1.6 V ≤ EV_{DD0} < 1.8 V			2	MHz
		LP (low-power main) mode		1.6 V ≤ EV_{DD0} < 1.8 V			2	MHz
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}	INTP0		1.6 V ≤ V_{DD} ≤ 5.5 V	1			μs
		INTP1 to INTP11		1.6 V ≤ EV_{DD0} ≤ 5.5 V	1			μs
Key interrupt input high-level width, low-level width	t_{KRH}, t_{KRL}	KR0 to KR7		1.8 V ≤ EV_{DD0} ≤ 5.5 V	250			ns
				1.6 V ≤ EV_{DD0} < 1.8 V	1			μs
$\overline{RESET}$ low-level width	t_{RSL}				10			μs

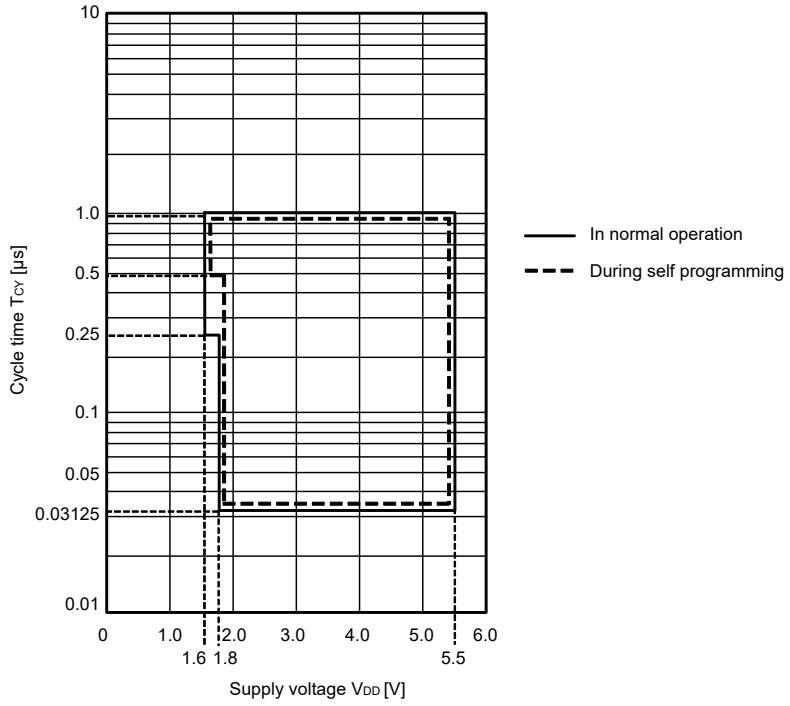
(Note and Remark are listed on the next page.)

Note The following conditions are required for low voltage interface when $EV_{DD0} < V_{DD}$.
1.8 V ≤ EV_{DD0} < 2.7 V: 125 ns min.
1.6 V ≤ EV_{DD0} < 1.8 V: 250 ns min.

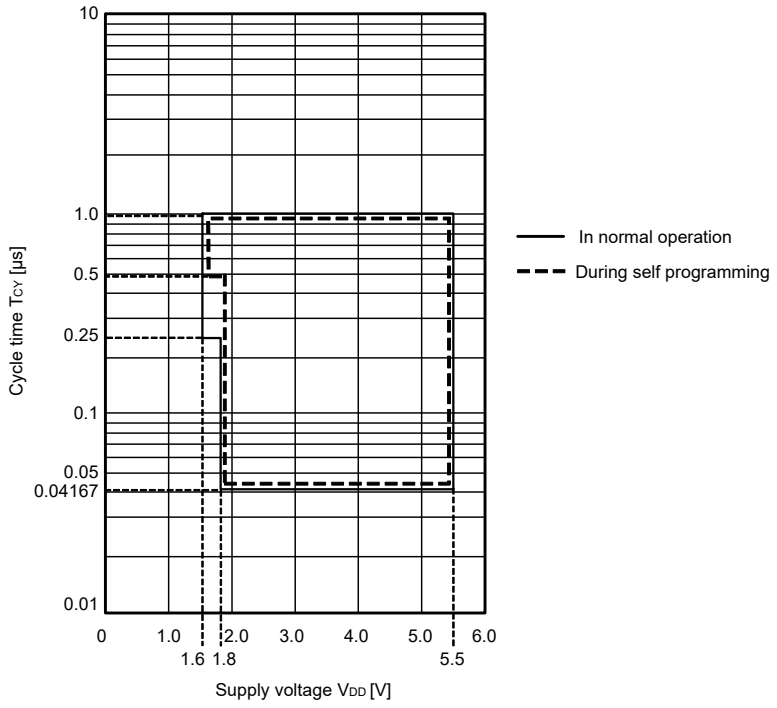
Remark f_{MCK}: Frequency of the operating clock for the timer array unit
To set this operating clock, set the CKSmn0 and CKSmn1 bits in timer mode register mn (TMRmn).
m: unit number (m = 0, 1), n: channel number (n = 0 to 7)

Minimum Instruction Execution Time during Main System Clock Operation

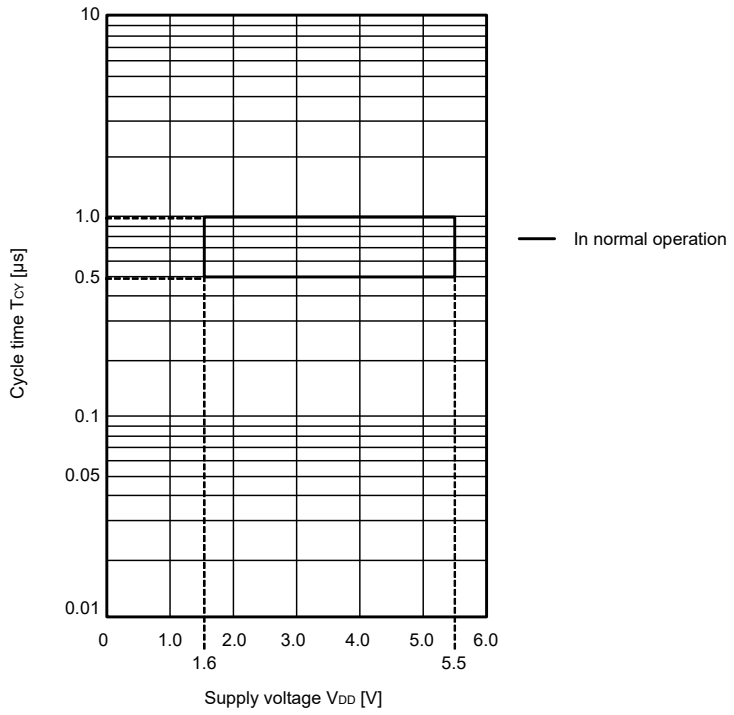
TCY vs VDD (HS (high-speed main) mode)



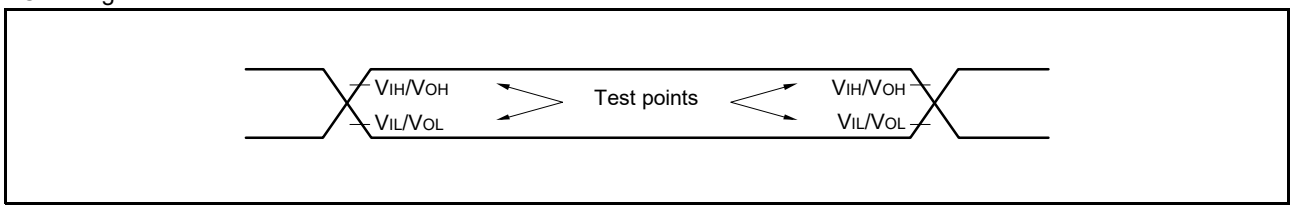
TCY vs VDD (LS (low-speed main) mode)



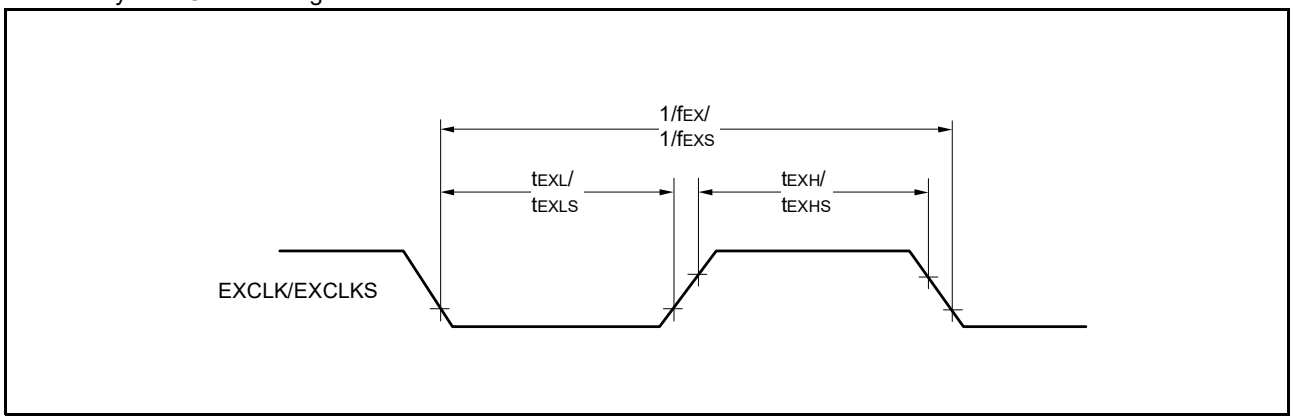
TCY vs VDD (LP (low-power main) mode)



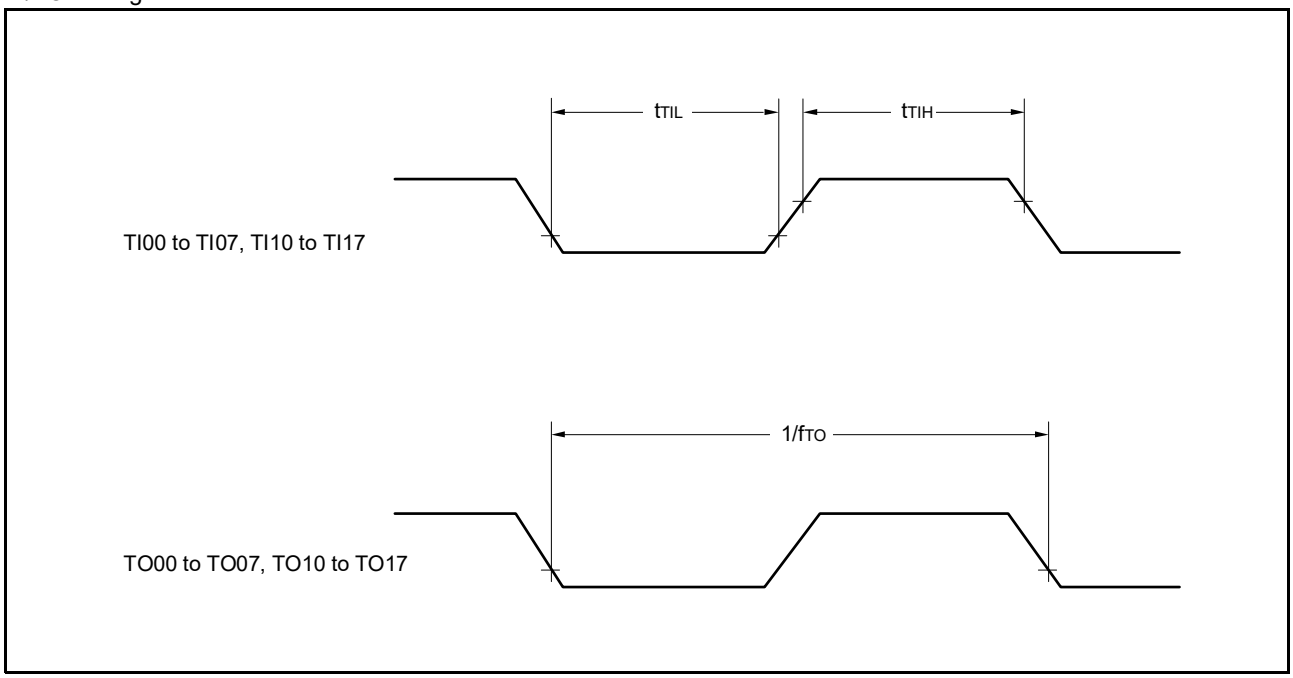
AC Timing Test Points



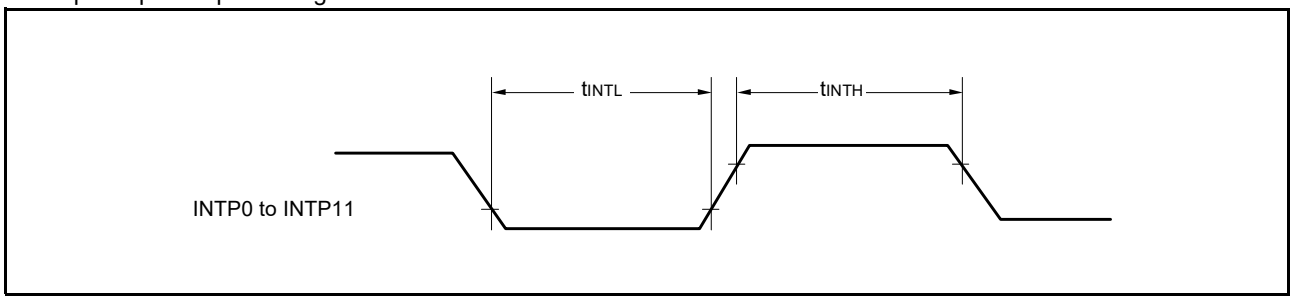
External System Clock Timing



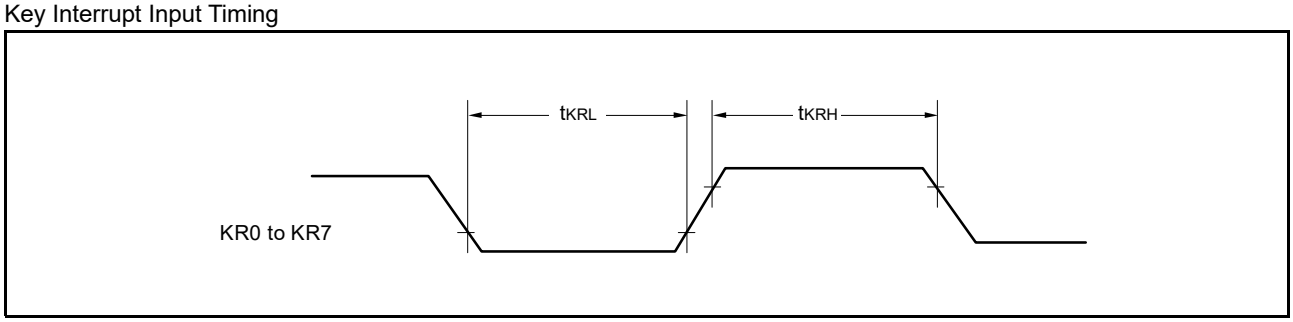
TI/TO Timing



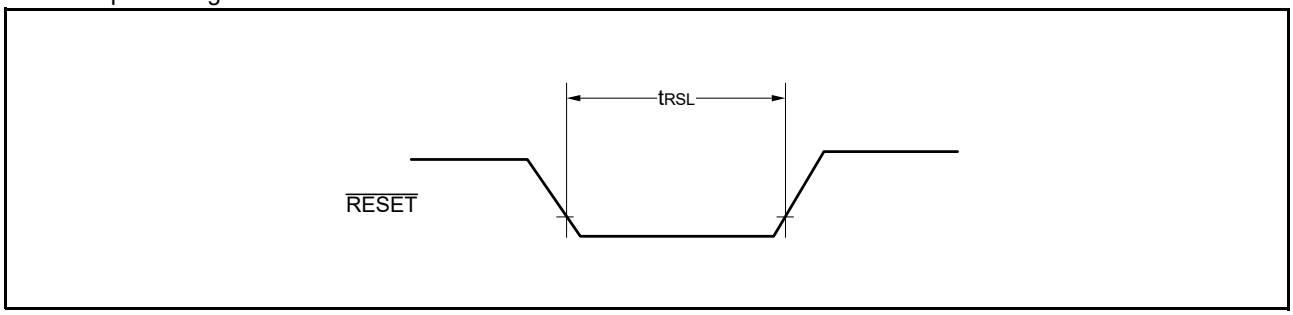
Interrupt Request Input Timing



Key Interrupt Input Timing

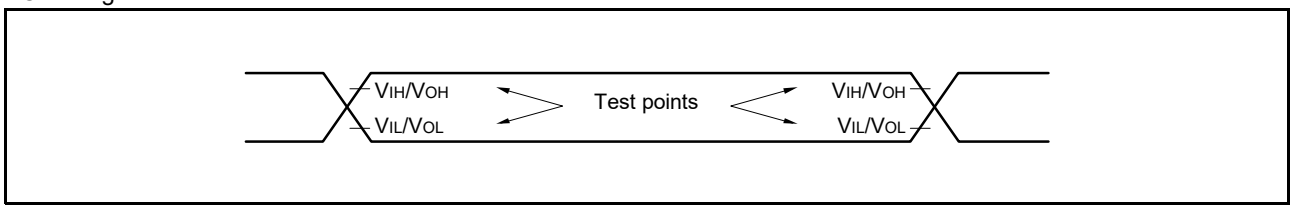


$\overline{\text{RESET}}$ Input Timing



37.5 Characteristics of the Peripheral Functions

AC Timing Test Points



37.5.1 Serial array unit

1. In UART communications with devices operating at same voltage levels

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V)

Item	Symbol	Conditions	HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
Transfer rate **Note 1**		1.6 V ≤ EV_{DD0} ≤ 5.5 V		f_{MCK}/6 **Note 2**		f_{MCK}/6 **Note 2**		f_{MCK}/6	bps
		Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} **Note 3**		5.3		4		0.33	Mbps

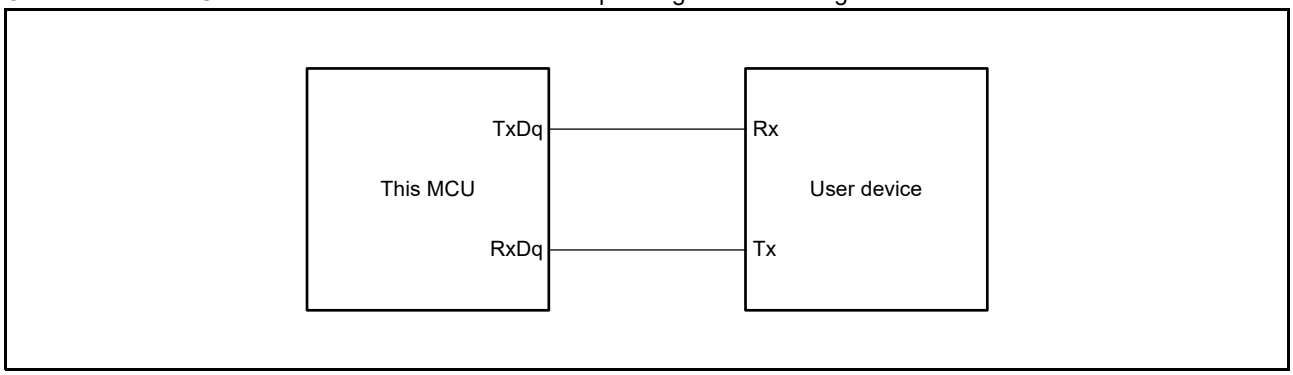
Note 1. The transfer rate in the SNOOZE mode is within the range from 4800 to 9600 bps.

Note 2. The following conditions are required for low voltage interface when EV_{DD0} < V_{DD}.
2.4 V ≤ EV_{DD0} < 2.7 V: 2.6 Mbps max.
1.8 V ≤ EV_{DD0} < 2.4 V: 1.3 Mbps max.
1.6 V ≤ EV_{DD0} < 1.8 V: 0.6 Mbps max.

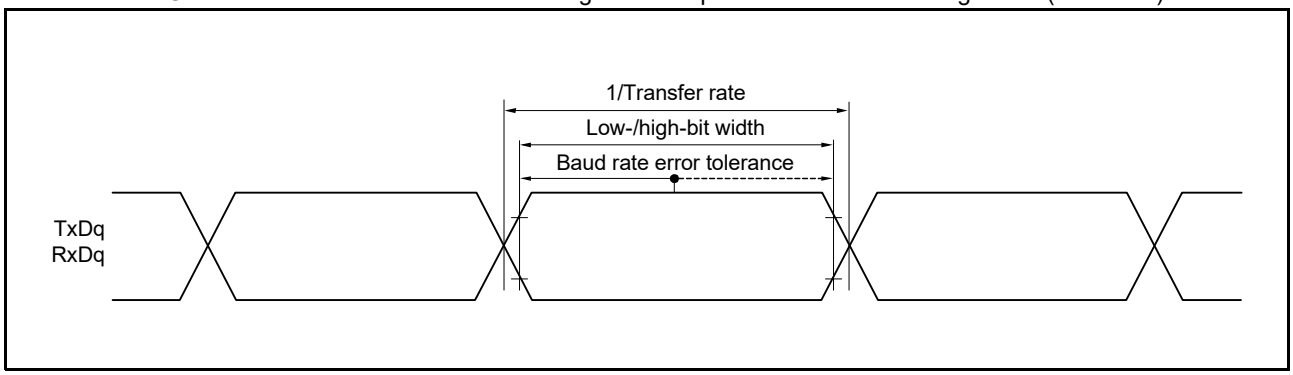
Note 3. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are as follows.
HS (high-speed main) mode: 32 MHz (1.8 V ≤ V_{DD} ≤ 5.5 V)
4 MHz (1.6 V ≤ V_{DD} ≤ 5.5 V)
LS (low-speed main) mode: 24 MHz (1.8 V ≤ V_{DD} ≤ 5.5 V)
4 MHz (1.6 V ≤ V_{DD} ≤ 5.5 V)
LP (low-power main) mode: 2 MHz (1.6 V ≤ V_{DD} ≤ 5.5 V)

Caution Select the normal input buffer for the RxDq pin and the normal output mode for the TxDq pin by using a port input mode register (PIMg) and port output mode register (POMg).

Connection in the UART communications with devices operating at same voltage levels



Bit width in the UART communications when interfacing devices operate at the same voltage level (reference)



Remark 1. q: UART number (q = 0 to 3), g: PIM and POM number (g = 0, 1, 8, 14)

Remark 2. f_{MCK}: Frequency of the operating clock for the serial array unit
To set this operating clock, set the CKSmn bit in serial mode register mn (SMRmn).
m: unit number, n: channel number (mn = 00 to 03, 10 to 13)

2. In simplified SPI (CSI) communications in the master mode with devices operating at same voltage levels with the internal SCKp clock (the ratings below are only applicable to CSI00)

(T_A = -40 to +85°C, 2.7 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V)

Item	Symbol	Conditions		HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ 2/f_{CLK}	4.0 V ≤ EV_{DD0} ≤ 5.5 V	62.5		83.3		1000		ns
			2.7 V ≤ EV_{DD0} ≤ 5.5 V	83.3		125		1000		ns
SCKp high-level width, low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY1}/2 - 7		t_{KCY1}/2 - 10		t_{KCY1}/2 - 50		ns
		2.7 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY1}/2 - 10		t_{KCY1}/2 - 15		t_{KCY1}/2 - 50		ns
SIp setup time (to SCKp↑)**Note 1**	t_{SIK1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V		23		33		110		ns
		2.7 V ≤ EV_{DD0} ≤ 5.5 V		33		50		110		ns
SIp hold time (from SCKp↑) **Note 1**	t_{KSI1}	2.7 V ≤ EV_{DD0} ≤ 5.5 V		10		10		10		ns
Delay time from SCKp↓ to SOp output**Note 2**	t_{KSO1}	C = 20 pF**Note 3**			10		10		10	ns

Note 1. The setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The setting for the SIp setup time becomes “to SCKp↓” and that for the SIp hold time becomes “from SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. This setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The setting for the delay time to SOp output becomes “from SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. C is the load capacitance of the SCKp and SOp output lines.

Caution Select the normal input buffer for the SIp pin and the normal output mode for the SOp pin and SCKp pin by using a port input mode register (PIMg) and port output mode register (POMg).

Remark 1. The listed times are only valid when the peripheral I/O redirect function of CSI00 is not in use.

Remark 2. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM numbers (g = 1)

Remark 3. f_{MCK}: Frequency of the operating clock for the serial array unit
To set this operating clock, set the CKSmn bit in serial mode register mn (SMRmn).
m: unit number, n: channel number (mn = 00)

3. In simplified SPI (CSI) communications in the master mode with devices operating at same voltage levels with the internal SCKp clock

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V)

Item	Symbol	Conditions		HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ 4/f_{CLK}	2.7 V ≤ EV_{DD0} ≤ 5.5 V	125		166		2000		ns
			2.4 V ≤ EV_{DD0} ≤ 5.5 V	250		250		2000		ns
			1.8 V ≤ EV_{DD0} ≤ 5.5 V	500		500		2000		ns
			1.6 V ≤ EV_{DD0} ≤ 5.5 V	1000		1000		2000		ns
SCKp high-level width, low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY1}/2 - 12		t_{KCY1}/2 - 21		t_{KCY1}/2 - 50		ns
		2.7 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY1}/2 - 18		t_{KCY1}/2 - 25		t_{KCY1}/2 - 50		ns
		2.4 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY1}/2 - 38		t_{KCY1}/2 - 38		t_{KCY1}/2 - 50		ns
		1.8 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY1}/2 - 50		t_{KCY1}/2 - 50		t_{KCY1}/2 - 50		ns
		1.6 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY1}/2 - 100		t_{KCY1}/2 - 100		t_{KCY1}/2 - 100		ns
SIp setup time (to SCKp↑)**Note 1**	t_{SIK1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V		44		54		110		ns
		2.7 V ≤ EV_{DD0} ≤ 5.5 V		44		54		110		ns
		2.4 V ≤ EV_{DD0} ≤ 5.5 V		75		75		110		ns
		1.8 V ≤ EV_{DD0} ≤ 5.5 V		110		110		110		ns
		1.6 V ≤ EV_{DD0} ≤ 5.5 V		220		220		220		ns
SIp hold time (from SCKp↑) **Note 1**	t_{KSI1}	1.6 V ≤ EV_{DD0} ≤ 5.5 V		19		19		19		ns
Delay time from SCKp↓ to SOp output**Note 2**	t_{KSO1}	1.6 V ≤ EV_{DD0} ≤ 5.5 V C = 30 pF**Note 3**			25		25		25	ns

Note 1. This setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The setting for the SIp setup time becomes “to SCKp↓” and that for the SIp hold time becomes “from SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. This setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The setting for the delay time to SOp output becomes “from SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. C is the load capacitance of the SCKp and SOp output lines.

Caution Select the normal input buffer for the SIp pin and the normal output mode for the SOp pin and SCKp pin by using a port input mode register (PIMg) and port output mode register (POMg).

Remark 1. p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), g: PIM and POM numbers (g = 0, 1, 4, 5, 8, 14)

Remark 2. f_{MCK}: Frequency of the operating clock for the serial array unit
To set this operating clock, set the CKSmn bit in serial mode register mn (SMRmn).
m: unit number, n: channel number (mn = 00 to 03, 10 to 13)

4. In simplified SPI (CSI) communications in the slave mode with devices operating at same voltage levels with the SCKp external clock

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (1/2)

Item	Symbol	Conditions		HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
SCKp cycle time **Note 1**	t_{KCY2}	4.0 V ≤ EV_{DD0} ≤ 5.5 V	20 MHz < f_{MCK}	8/f_{MCK}		8/f_{MCK}		—		ns
			f_{MCK} ≤ 20 MHz	6/f_{MCK}		6/f_{MCK}		6/f_{MCK}		ns
		2.7 V ≤ EV_{DD0} ≤ 5.5 V	16 MHz < f_{MCK}	8/f_{MCK}		8/f_{MCK}		—		ns
			f_{MCK} ≤ 16 MHz	6/f_{MCK}		6/f_{MCK}		6/f_{MCK}		ns
		2.4 V ≤ EV_{DD0} ≤ 5.5 V		6/f_{MCK} and 500		6/f_{MCK} and 500		6/f_{MCK} and 500		ns
		1.8 V ≤ EV_{DD0} ≤ 5.5 V		6/f_{MCK} and 750		6/f_{MCK} and 750		6/f_{MCK} and 750		ns
		1.6 V ≤ EV_{DD0} ≤ 5.5 V		6/f_{MCK} and 1500		6/f_{MCK} and 1500		6/f_{MCK} and 1500		ns
SCKp high-level width, low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY2}/2 - 7		t_{KCY2}/2 - 7		t_{KCY2}/2 - 7		ns
		2.7 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY2}/2 - 8		t_{KCY2}/2 - 8		t_{KCY2}/2 - 8		ns
		1.8 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY2}/2 - 18		t_{KCY2}/2 - 18		t_{KCY2}/2 - 18		ns
		1.6 V ≤ EV_{DD0} ≤ 5.5 V		t_{KCY2}/2 - 66		t_{KCY2}/2 - 66		t_{KCY2}/2 - 66		ns

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

4. In simplified SPI (CSI) communications in the slave mode with devices operating at same voltage levels with the SCKp external clock

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (2/2)

Item	Symbol	Conditions		HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
SIp setup time (to SCKp↑)**Note 2**	tSIK2	2.7 V ≤ EV_{DD0} ≤ 5.5 V		1/f_{MCK} + 20		1/f_{MCK} + 30		1/f_{MCK} + 30		ns
		1.8 V ≤ EV_{DD0} ≤ 5.5 V		1/f_{MCK} + 30		1/f_{MCK} + 30		1/f_{MCK} + 30		ns
		1.6 V ≤ EV_{DD0} ≤ 5.5 V		1/f_{MCK} + 40		1/f_{MCK} + 40		1/f_{MCK} + 40		ns
SIp hold time (from SCKp↑)**Note 2**	tKSI2	1.8 V ≤ EV_{DD0} ≤ 5.5 V		1/f_{MCK} + 31		1/f_{MCK} + 31		1/f_{MCK} + 31		ns
		1.6 V ≤ EV_{DD0} ≤ 5.5 V		1/f_{MCK} + 250		1/f_{MCK} + 250		1/f_{MCK} + 250		ns
Delay time from SCKp↓ to SOp output **Note 3**	tKSO2	C = 30 pF **Note 4**	2.7 V ≤ EV_{DD0} ≤ 5.5 V		2/f_{MCK} + 44		2/f_{MCK} + 110		2/f_{MCK} + 110	ns
			2.4 V ≤ EV_{DD0} ≤ 5.5 V		2/f_{MCK} + 75		2/f_{MCK} + 110		2/f_{MCK} + 110	ns
			1.8 V ≤ EV_{DD0} ≤ 5.5 V		2/f_{MCK} + 110		2/f_{MCK} + 110		2/f_{MCK} + 110	ns
			1.6 V ≤ EV_{DD0} ≤ 5.5 V		2/f_{MCK} + 220		2/f_{MCK} + 220		2/f_{MCK} + 220	ns

Note 1. Transfer rate in the SNOOZE mode is 1 Mbps at the maximum.

Note 2. This setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The setting for the SIp setup time becomes “to SCKp↓” and that for the SIp hold time becomes “from SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. This setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The setting for the delay time to SOp output becomes “from SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

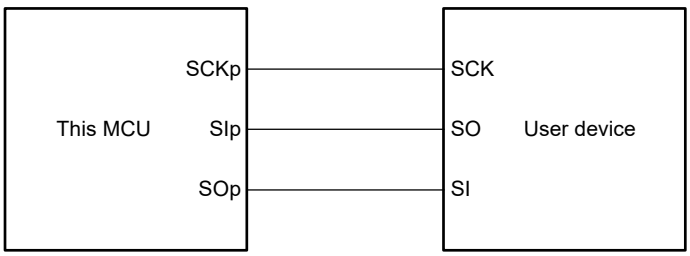
Note 4. C is the load capacitance of the SOp output line.

Caution **Select the normal input buffer for the SIp pin and SCKp pin and the normal output mode for the SOp pin by using a port input mode register (PIMg) and port output mode register (POMg).**

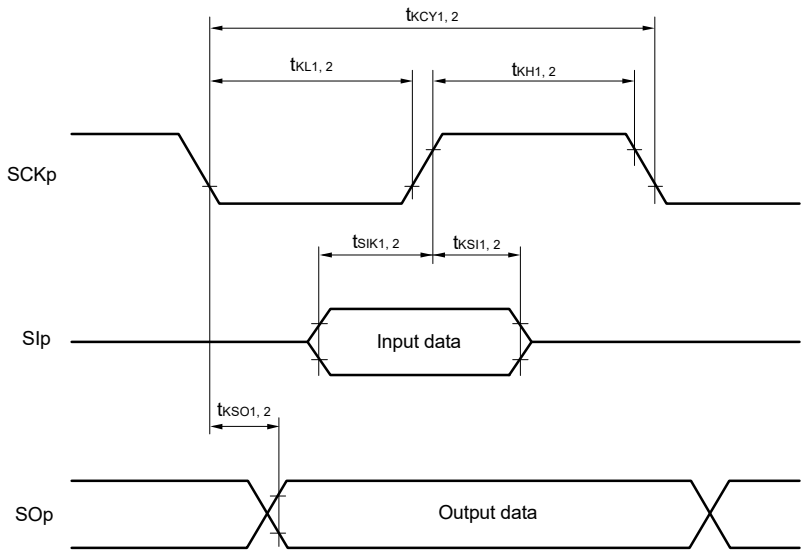
Remark 1. p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31), m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), g: PIM and POM numbers (g = 0, 1, 4, 5, 8, 14)

Remark 2. f_{MCK}: Frequency of the operating clock for the serial array unit
To set this operating clock, set the CKSmn bit in serial mode register mn (SMRmn).
m: unit number, n: channel number (mn = 00 to 03, 10 to 13)

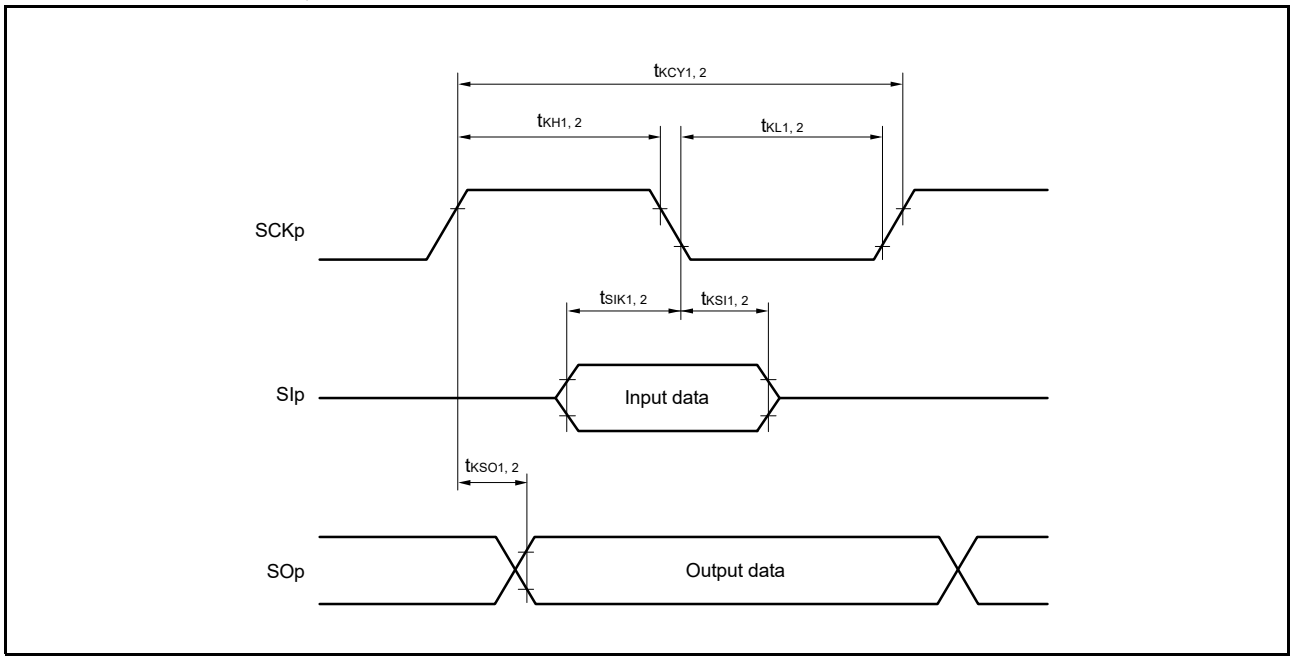
Connection in the simplified SPI (CSI) communications with devices operating at same voltage levels



Timing of serial transfer in the simplified SPI (CSI) communications with devices operating at same voltage levels when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1



Timing of serial transfer in the simplified SPI (CSI) communications with devices operating at same voltage levels when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0



Remark 1. p: CSI number (p = 00, 01, 10, 11, 20, 21, 30, 31)

Remark 2. m: Unit number, n: Channel number (mn = 00 to 03, 10 to 13)

5. In simplified I²C communications with devices operating at same voltage levels

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (1/2)

Item	Symbol	Conditions	HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
SCLr clock frequency	f_{SCL}	2.7 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ		1000 Note 1		1000 Note 1		400 Note 1	kHz
		1.8 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ		400 Note 1		400 Note 1		400 Note 1	kHz
		1.8 V ≤ EV_{DD0} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ		300 Note 1		300 Note 1		300 Note 1	kHz
		1.6 V ≤ EV_{DD0} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ		250 Note 1		250 Note 1		250 Note 1	kHz
Hold time when SCLr is low	t_{LOW}	2.7 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	475		475		1150		ns
		1.8 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1150		1150		1150		ns
		1.8 V ≤ EV_{DD0} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ	1550		1550		1550		ns
		1.6 V ≤ EV_{DD0} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	1850		1850		1850		ns
Hold time when SCLr is high	t_{HIGH}	2.7 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	475		475		1150		ns
		1.8 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1150		1150		1150		ns
		1.8 V ≤ EV_{DD0} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ	1550		1550		1550		ns
		1.6 V ≤ EV_{DD0} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	1850		1850		1850		ns

(Notes and Caution are listed on the next page, and Remarks are listed on page 1527.)

5. In simplified I²C communications with devices operating at same voltage levels

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (2/2)

Item	Symbol	Conditions	HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	1/f_{MCK} + 85 **Note 2**		1/f_{MCK} + 85 **Note 2**		1/f_{MCK} + 145 **Note 2**		ns
		1.8 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1/f_{MCK} + 145 **Note 2**		1/f_{MCK} + 145 **Note 2**		1/f_{MCK} + 145 **Note 2**		ns
		1.8 V ≤ EV_{DD0} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ	1/f_{MCK} + 230 **Note 2**		1/f_{MCK} + 230 **Note 2**		1/f_{MCK} + 230 **Note 2**		ns
		1.6 V ≤ EV_{DD0} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	1/f_{MCK} + 290 **Note 2**		1/f_{MCK} + 290 **Note 2**		1/f_{MCK} + 290 **Note 2**		ns
Data hold time (transmission)	$t_{HD:DAT}$	2.7 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	0	305	0	305	0	305	ns
		1.8 V ≤ EV_{DD0} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	0	355	0	355	0	355	ns
		1.8 V ≤ EV_{DD0} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ	0	405	0	405	0	405	ns
		1.6 V ≤ EV_{DD0} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	0	405	0	405	0	405	ns

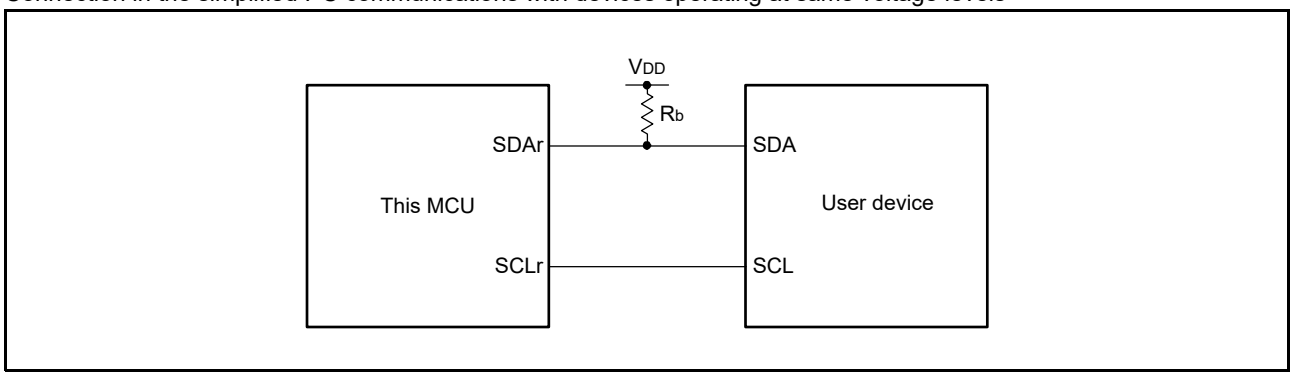
Note 1. The listed times must be no greater than f_{MCK}/4.

Note 2. Set f_{MCK} so that it will not exceed the hold time when SCLr is low or high.

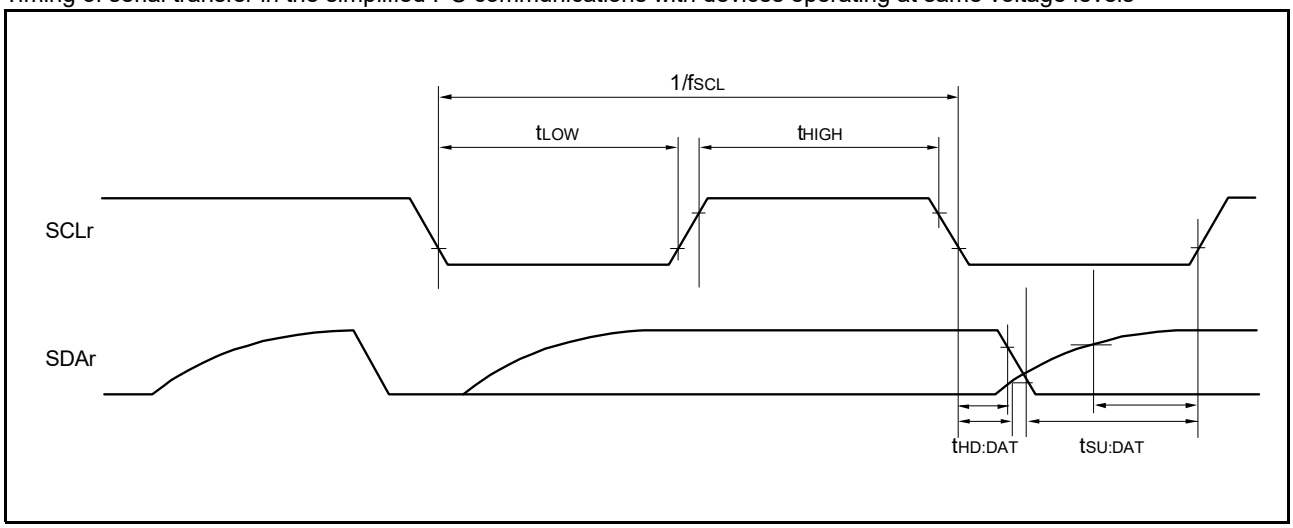
Caution Select the normal input buffer and the N-ch open drain output (withstand voltage of V_{DD} (when 30- to 52-pin R7F100G and 32- to 48-pin R7F104G products)/withstand voltage of EV_{DD} (when 64- to 128-pin R7F100G products)) mode for the SDAr pin and the normal output mode for the SCLr pin by using a port input mode register (PIMg) and port output mode register (POMh).

(Remarks are listed on the next page.)

Connection in the simplified I2C communications with devices operating at same voltage levels



Timing of serial transfer in the simplified I2C communications with devices operating at same voltage levels



Remark 1. R_b[Ω]: Communication line (SDAr) pull-up resistance, C_b[F]: Communication line (SDAr, SCLr) load capacitance

Remark 2. r: IIC number (r = 00, 01, 10, 11, 20, 21, 30, 31), g: PIM number (g = 0, 1, 4, 5, 8, 14), h: POM number (g = 0, 1, 4, 5, 7 to 9, 14)

Remark 3. f_{MCK}: Frequency of the operating clock for the serial array unit
To set this operating clock, set the CKSmn bit in serial mode register mn (SMRmn).
m: unit number, n: channel number (mn = 00 to 03, 10 to 13)

6. In UART communications with devices operating at different voltage levels (1.8 V, 2.5 V, 3 V)

(T_A = -40 to +105°C, 1.8 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (1/2)

Item	Symbol	Conditions		HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
Transfer rate		Reception	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		f_{MCK}/6 **Note 1**		f_{MCK}/6 **Note 1**		f_{MCK}/6 **Note 1**	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} **Note 2**		5.3		4		0.33	Mbps
			2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		f_{MCK}/6 **Note 1**		f_{MCK}/6 **Note 1**		f_{MCK}/6 **Note 1**	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} **Note 2**		5.3		4		0.33	Mbps
			1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V		f_{MCK}/6 **Notes 1, 3, 4**		f_{MCK}/6 **Notes 1, 3**		f_{MCK}/6 **Notes 1, 3**	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} **Note 2**		5.3		4		0.33	Mbps

Note 1. Transfer rate in the SNOOZE mode is within the range from 4800 to 9600 bps.

Note 2. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:
HS (high-speed main) mode: 32 MHz (1.8 V ≤ V_{DD} ≤ 5.5 V)
4 MHz (1.6 V ≤ V_{DD} ≤ 5.5 V)
LS (low-speed main) mode: 24 MHz (1.8 V ≤ V_{DD} ≤ 5.5 V)
4 MHz (1.6 V ≤ V_{DD} ≤ 5.5 V)
LP (low-power main) mode: 2 MHz (1.6 V ≤ V_{DD} ≤ 5.5 V)

Note 3. Use this rate with EV_{DD0} ≥ V_b.

Note 4. The following conditions are required for low voltage interface when EV_{DD0} < V_{DD}.
2.4 V ≤ EV_{DD0} < 2.7 V: 2.6 Mbps (max.)
1.8 V ≤ EV_{DD0} < 2.4 V: 1.3 Mbps (max.)

Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (withstand voltage of V_{DD} (when 30- to 52-pin R7F100G and 32- to 48-pin R7F104G products)/withstand voltage of EV_{DD} (when 64- to 128-pin R7F100G products)) mode for the TxDq pin by using a port input mode register (PIMg) and port output mode register (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Remark 1. V_b[V]: Communication line voltage

Remark 2. q: UART number (q = 0 to 3), g: PIM and POM number (g = 0, 1, 8, 14)

Remark 3. f_{MCK}: Frequency of the operating clock for the serial array unit
To set this operating clock, set the CKSmn bit in serial mode register mn (SMRmn).
m: unit number, n: channel number (mn = 00 to 03, 10 to 13)

Remark 4. Communications by using UART2 with devices operating at different voltage levels are not possible when the setting of the PIOR1 bit of the peripheral I/O redirection register (PIOR) is 1.

6. In UART communications with devices operating at different voltage levels (1.8 V, 2.5 V, 3 V)

(T_A = -40 to +105°C, 1.8 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (2/2)

Item	Symbol	Conditions		HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
Transfer rate		Transmission	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		**Note 1**		**Note 1**		**Note 1**	bps
			Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 1.4 kΩ, V_b = 2.7 V		2.8**Note 2**		2.8**Note 2**		2.8**Note 2**	Mbps
			2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		**Note 3**		**Note 3**		**Note 3**	bps
			Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 2.7 kΩ, V_b = 2.3 V		1.2**Note 4**		1.2**Note 4**		1.2**Note 4**	Mbps
			1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V		**Notes 5, 6**		**Notes 5, 6**		**Notes 5, 6**	bps
			Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 5.5 kΩ, V_b = 1.6 V		0.43 **Note 7**		0.43 **Note 7**		0.43 **Note 7**	Mbps

Note 1. The smaller maximum transfer rate derived by using f_{MCK}/6 or the following expression is the valid maximum transfer rate.
Expression for calculating the transfer rate when 4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

Note 2. This rate is calculated as an example when the conditions described in the “Conditions” column are met. See **Note 1** above to calculate the maximum transfer rate under conditions of the customer.

(Notes and Caution continue in the next page.)

Note 3. The smaller maximum transfer rate derived by using $f_{MCK}/6$ or the following expression is the valid maximum transfer rate.
Expression for calculating the transfer rate when 2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

Note 4. This rate is calculated as an example when the conditions described in the “Conditions” column are met. See **Note 3** above to calculate the maximum transfer rate under conditions of the customer.

Note 5. Use this rate with $EV_{DD0} \geq V_b$.

Note 6. The smaller maximum transfer rate derived by using $f_{MCK}/6$ or the following expression is the valid maximum transfer rate.
Expression for calculating the transfer rate when 1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\} \times 3} \text{ [bps]}$$

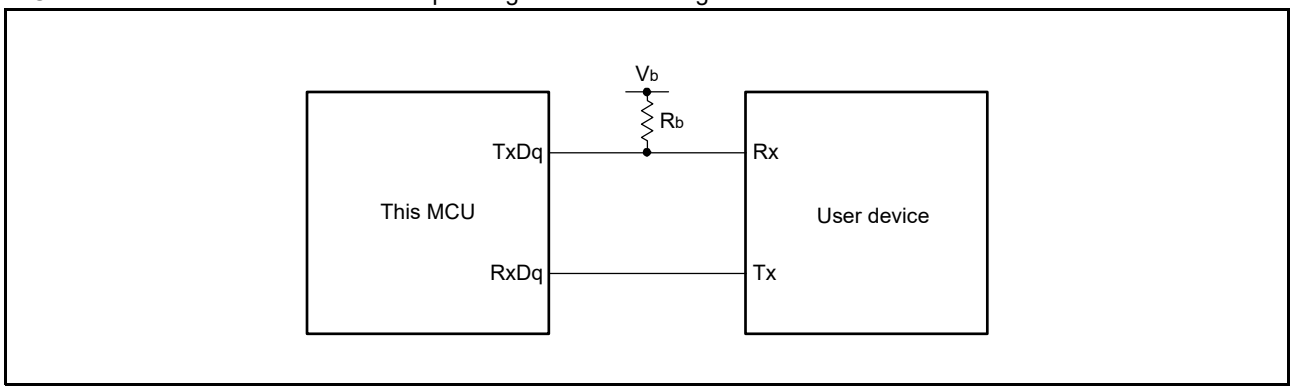
$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{1.5}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

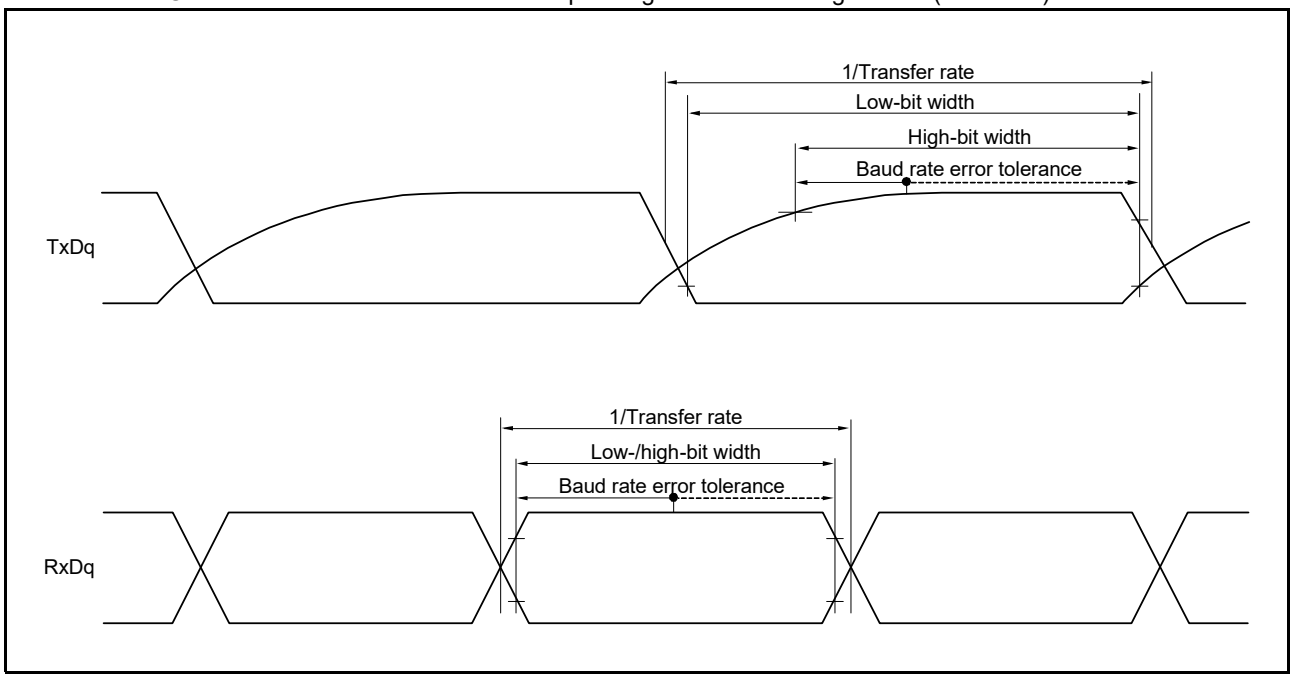
Note 7. This rate is calculated as an example when the conditions described in the “Conditions” column are met. Refer to **Note 6** above to calculate the maximum transfer rate under conditions of the customer.

Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (withstand voltage of V_{DD} (when 30- to 52-pin R7F100G and 32- to 48-pin R7F104G products)/withstand voltage of EV_{DD} (when 64- to 128-pin R7F100G products)) mode for the TxDq pin by using a port input mode register (PIMg) and port output mode register (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

In UART communications with devices operating at different voltage levels



Bit width in the UART communications with devices operating at different voltage levels (reference)



Remark 1. R_b[Ω]: Communication line (TxDq) pull-up resistance, C_b[F]: Communication line (TxDq) load capacitance, V_b[V]: Communication line voltage

Remark 2. q: UART number (q = 0 to 3), g: PIM and POM number (g = 0, 1, 8, 14)

Remark 3. f_{MCK}: Frequency of the operating clock for the serial array unit
To set this operating clock, set the CKSmn bit in serial mode register mn (SMRmn).
m: unit number, n: channel number (mn = 00 to 03, 10 to 13)

Remark 4. Communications by using UART2 with devices operating at different voltage levels are not possible when the setting of the PIOR1 bit of the peripheral I/O redirection register (PIOR) is 1.

7. In simplified SPI (CSI) communications in the master mode with devices operating at different voltage levels (2.5 V or 3 V) with the internal SCKp clock (the ratings below are only applicable to CSI00)

(T_A = -40 to +105°C, 2.7 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (1/2)

Item	Symbol	Conditions		HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ 2/f_{CLK}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ	200		200		2300		ns
			2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ	300		300		2300		ns
SCKp high-level width	t_{KH1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		t_{KCY1}/2 - 50		t_{KCY1}/2 - 50		t_{KCY1}/2 - 50		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		t_{KCY1}/2 - 120		t_{KCY1}/2 - 120		t_{KCY1}/2 - 120		ns
SCKp low-level width	t_{KL1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		t_{KCY1}/2 - 7		t_{KCY1}/2 - 7		t_{KCY1}/2 - 50		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		t_{KCY1}/2 - 10		t_{KCY1}/2 - 10		t_{KCY1}/2 - 50		ns
SIp setup time (to SCKp↑)**Note 1**	t_{SIK1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		58		58		479		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		121		121		479		ns
SIp hold time (from SCKp↑)**Note 1**	t_{KSI1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		10		10		10		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		10		10		10		ns
Delay time from SCKp↓ to SOp output**Note 1**	t_{KSO1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ			60		60		60	ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ			130		130		130	ns

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

7. In simplified SPI (CSI) communications in the master mode with devices operating at different voltage levels (2.5 V or 3 V) with the internal SCKp clock (the ratings below are only applicable to CSI00)

(T_A = -40 to +105°C, 2.7 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (2/2)

Item	Symbol	Conditions	HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
SIp setup time (to SCKp↓)**Note 2**	t_{SIK1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ	23		23		110		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ	33		33		110		ns
SIp hold time (from SCKp↓)**Note 2**	t_{KSI1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ	10		10		10		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ	10		10		10		ns
Delay time from SCKp↑ to SOp output**Note 2**	t_{KSO1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		10		10		10	ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		10		10		10	ns

Note 1. This setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

Note 2. This setting applies when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (withstand voltage of V_{DD} (when 30- to 52-pin R7F100G and 32- to 48-pin R7F104G products)/withstand voltage of EV_{DD} (when 64- to 128-pin R7F100G products)) mode for the SOp pin and SCKp pin by using a port input mode register (PIMg) and port output mode register (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Remark 1. R_b[Ω]: Communication line (SCKp, SOp) pull-up resistance, C_b[F]: Communication line (SCKp, SOp) load capacitance, V_b[V]: Communication line voltage

Remark 2. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM numbers (g = 1)

Remark 3. f_{MCK}: Frequency of the operating clock for the serial array unit
To set this operating clock, set the CKSmn bit in serial mode register mn (SMRmn).
m: unit number, n: channel number (mn = 00)

Remark 4. The listed times are only valid when the peripheral I/O redirect function of CSI00 is not in use.

8. In simplified SPI (CSI) communications in the master mode with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V) with the internal SCKp clock

(T_A = -40 to +105°C, 1.8 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (1/3)

Item	Symbol	Conditions		HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ 4/f_{CLK}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	300		300		2300		ns
			2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	500		500		2300		ns
			1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V**Note**, C_b = 30 pF, R_b = 5.5 kΩ	1150		1150		2300		ns
SCKp high-level width	t_{KH1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KCY1}/2 - 75		t_{KCY1}/2 - 75		t_{KCY1}/2 - 75		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 - 170		t_{KCY1}/2 - 170		t_{KCY1}/2 - 170		ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V**Note**, C_b = 30 pF, R_b = 5.5 kΩ		t_{KCY1}/2 - 458		t_{KCY1}/2 - 458		t_{KCY1}/2 - 458		ns
SCKp low-level width	t_{KL1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KCY1}/2 - 12		t_{KCY1}/2 - 12		t_{KCY1}/2 - 50		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 - 18		t_{KCY1}/2 - 18		t_{KCY1}/2 - 50		ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V**Note**, C_b = 30 pF, R_b = 5.5 kΩ		t_{KCY1}/2 - 50		t_{KCY1}/2 - 50		t_{KCY1}/2 - 50		ns

Note Use this setting with EV_{DD0} ≥ V_b.

Caution **Select the TTL input buffer for the SIp pin and the N-ch open drain output (withstand voltage of V_{DD} (when 30- to 52-pin R7F100G and 32- to 48-pin R7F104G products)/withstand voltage of EV_{DD} (when 64- to 128-pin R7F100G products)) mode for the SOp pin and SCKp pin by using a port input mode register (PIMg) and port output mode register (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.**

(Remarks are listed on page 1537.)

8. In simplified SPI (CSI) communications in the master mode with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V) with the internal SCKp clock

(T_A = -40 to +105°C, 1.8 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (2/3)

Item	Symbol	Conditions	HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
SIp setup time (to SCKp↑)Note 1	t_{SIK1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	81		81		479		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	177		177		479		ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 VNote 2, C_b = 30 pF, R_b = 5.5 kΩ	479		479		479		ns
SIp hold time (from SCKp↑)Note 1	t_{KSI1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	19		19		19		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	19		19		19		ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 VNote 2, C_b = 30 pF, R_b = 5.5 kΩ	19		19		19		ns
Delay time from SCKp↓ to SOp outputNote 1	t_{KSO1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		100		100		100	ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		195		195		195	ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 VNote 2, C_b = 30 pF, R_b = 5.5 kΩ		483		483		483	ns

Note 1. This setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

Note 2. Use this setting with EV_{DD0} ≥ V_b.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (withstand voltage of V_{DD} (when 30- to 52-pin R7F100G and 32- to 48-pin R7F104G products)/withstand voltage of EV_{DD} (when 64- to 128-pin R7F100G products)) mode for the SOp pin and SCKp pin by using a port input mode register (PIMg) and port output mode register (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

(Remarks are listed on page 1537.)

8. In simplified SPI (CSI) communications in the master mode with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V) with the internal SCKp clock

(T_A = -40 to +105°C, 1.8 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (3/3)

Item	Symbol	Conditions	HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
SIp setup time (to SCKp↓)Note 1	t_{SIK1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	44		44		110		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	44		44		110		ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 VNote 2, C_b = 30 pF, R_b = 5.5 kΩ	110		110		110		ns
SIp hold time (from SCKp↓)Note 1	t_{KSI1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	19		19		19		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	19		19		19		ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 VNote 2, C_b = 30 pF, R_b = 5.5 kΩ	19		19		19		ns
Delay time from SCKp↑ to SOp outputNote 1	t_{KSO1}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		25		25		25	ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		25		25		25	ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 VNote 2, C_b = 30 pF, R_b = 5.5 kΩ		25		25		25	ns

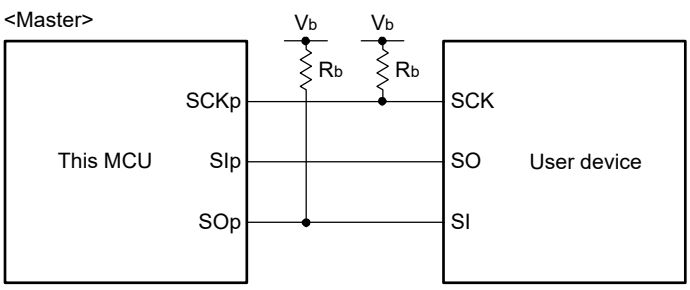
Note 1. This setting applies when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. Use this setting with EV_{DD0} ≥ V_b.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (withstand voltage of V_{DD} (when 30- to 52-pin R7F100G and 32- to 48-pin R7F104G products)/withstand voltage of EV_{DD} (when 64- to 128-pin R7F100G products)) mode for the SOp pin and SCKp pin by using a port input mode register (PIMg) and port output mode register (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

(Remarks are listed on the next page.)

Connection in the simplified SPI (CSI) communications with devices operating at different voltage levels



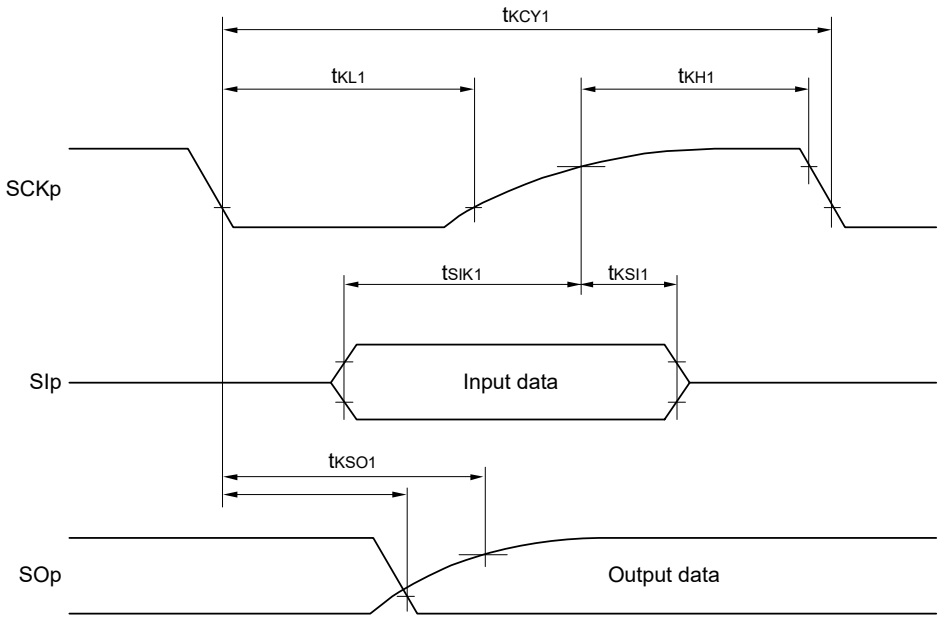
Remark 1. R_b[Ω]: Communication line (SCKp, SOp) pull-up resistance, C_b[F]: Communication line (SCKp, SOp) load capacitance, V_b[V]: Communication line voltage

Remark 2. p: CSI number (p = 00, 01, 10, 20, 30, 31), m: Unit number, n: Channel number (mn = 00, 01, 02, 10, 12, 13), g: PIM and POM number (g = 0, 1, 4, 5, 8, 14)

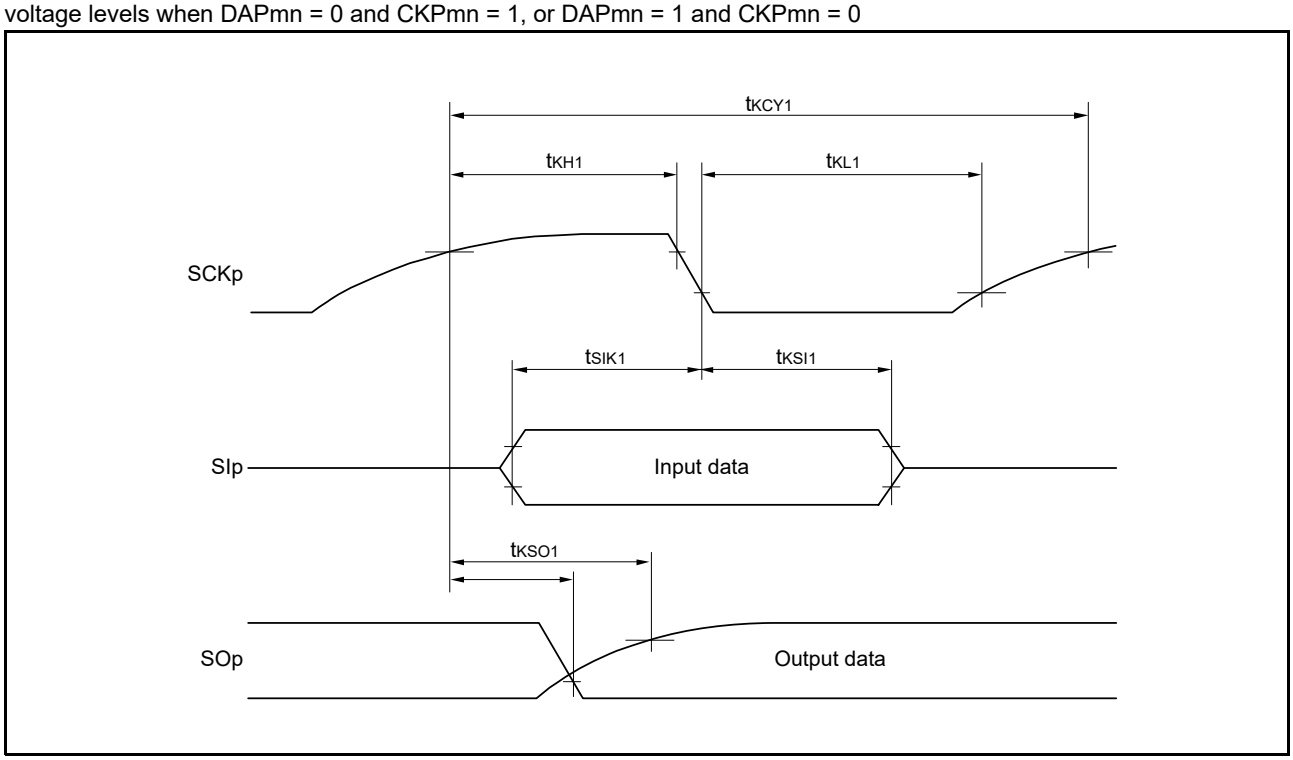
Remark 3. f_{MCK}: Frequency of the operating clock for the serial array unit
To set this operating clock, set the CKSmn bit in serial mode register mn (SMRmn).
m: unit number, n: channel number (mn = 00)

Remark 4. Communications by using CSI01 of 48- to 64-pin R7F100G products, and CSI11 and CSI21 with devices operating at different voltage levels are not possible. Use other CSI channels to handle such communications.

Timing of serial transfer in the simplified SPI (CSI) communications in the master mode with devices operating at different voltage levels when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1



Timing of serial transfer in the simplified SPI (CSI) communications in the master mode with devices operating at different voltage levels when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0



Remark 1. p: CSI number (p = 00, 01, 10, 20, 30, 31), m: Unit number, n: Channel number (mn = 00, 01, 02, 10, 12, 13), g: PIM and POM number (g = 0, 1, 4, 5, 8, 14)

Remark 2. Communications by using CSI01 of 48- to 64-pin R7F100G products, and CSI11 and CSI21 with devices operating at different voltage levels are not possible. Use other CSI channels to handle such communications.

9. In simplified SPI (CSI) communications in the slave mode with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V) with the external SCKp clock

(T_A = -40 to +105°C, 1.8 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (1/2)

Item	Symbol	Conditions		HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
				Min.	Max.	Min.	Max.	Min.	Max.	
SCKp cycle time **Note 1**	t_{KCY2}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V	24 MHz < f_{MCK}	14/f_{MCK}		—		—		ns
			20 MHz < f_{MCK} ≤ 24 MHz	12/f_{MCK}		12/f_{MCK}		—		ns
			8 MHz < f_{MCK} ≤ 20 MHz	10/f_{MCK}		10/f_{MCK}		—		ns
			4 MHz < f_{MCK} ≤ 8 MHz	8/f_{MCK}		8/f_{MCK}		—		ns
			f_{MCK} ≤ 4 MHz	6/f_{MCK}		6/f_{MCK}		10/f_{MCK}		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V,	24 MHz < f_{MCK}	20/f_{MCK}		—		—		ns
			20 MHz < f_{MCK} ≤ 24 MHz	16/f_{MCK}		16/f_{MCK}		—		ns
			16 MHz < f_{MCK} ≤ 20 MHz	14/f_{MCK}		14/f_{MCK}		—		ns
			8 MHz < f_{MCK} ≤ 16 MHz	12/f_{MCK}		12/f_{MCK}		—		ns
			4 MHz < f_{MCK} ≤ 8 MHz	8/f_{MCK}		8/f_{MCK}		—		ns
			f_{MCK} ≤ 4 MHz	6/f_{MCK}		6/f_{MCK}		10/f_{MCK}		ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V **Note 2**	24 MHz < f_{MCK}	48/f_{MCK}		—		—		ns
			20 MHz < f_{MCK} ≤ 24 MHz	36/f_{MCK}		36/f_{MCK}		—		ns
			16 MHz < f_{MCK} ≤ 20 MHz	32/f_{MCK}		32/f_{MCK}		—		ns
			8 MHz < f_{MCK} ≤ 16 MHz	26/f_{MCK}		26/f_{MCK}		—		ns
			4 MHz < f_{MCK} ≤ 8 MHz	16/f_{MCK}		16/f_{MCK}		—		ns
			f_{MCK} ≤ 4 MHz	10/f_{MCK}		10/f_{MCK}		10/f_{MCK}		ns

(Notes and Caution are listed on the next page, and Remarks are listed on page 1541.)

9. In simplified SPI (CSI) communications in the slave mode with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V) with the external SCKp clock

(T_A = -40 to +105°C, 1.8 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (2/2)

Item	Symbol	Conditions	HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
SCKp high-level width, low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V	t_{KCY2}/2 - 12		t_{KCY2}/2 - 12		t_{KCY2}/2 - 50		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V	t_{KCY2}/2 - 18		t_{KCY2}/2 - 18		t_{KCY2}/2 - 50		ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V**Note 2**	t_{KCY2}/2 - 50		t_{KCY2}/2 - 50		t_{KCY2}/2 - 50		ns
SIp setup time (to SCKp↑)**Note 3**	t_{SIK2}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V	1/f_{MCK} + 20		1/f_{MCK} + 20		1/f_{MCK} + 30		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V	1/f_{MCK} + 20		1/f_{MCK} + 20		1/f_{MCK} + 30		ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V**Note 2**	1/f_{MCK} + 30		1/f_{MCK} + 30		1/f_{MCK} + 30		ns
SIp hold time (from SCKp↑)**Note 3**	t_{KSI2}		1/f_{MCK} + 31		1/f_{MCK} + 31		1/f_{MCK} + 31		ns
Delay time from SCKp↓ to SOp output**Note 4**	t_{KSO2}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		2/f_{MCK} + 120		2/f_{MCK} + 120		2/f_{MCK} + 573	ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		2/f_{MCK} + 214		2/f_{MCK} + 214		2/f_{MCK} + 573	ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V**Note 2**, C_b = 30 pF, R_b = 5.5 kΩ		2/f_{MCK} + 573		2/f_{MCK} + 573		2/f_{MCK} + 573	ns

Note 1. Transfer rate in the SNOOZE mode is 1 Mbps at the maximum.

Note 2. Use this setting with EV_{DD0} ≥ V_b.

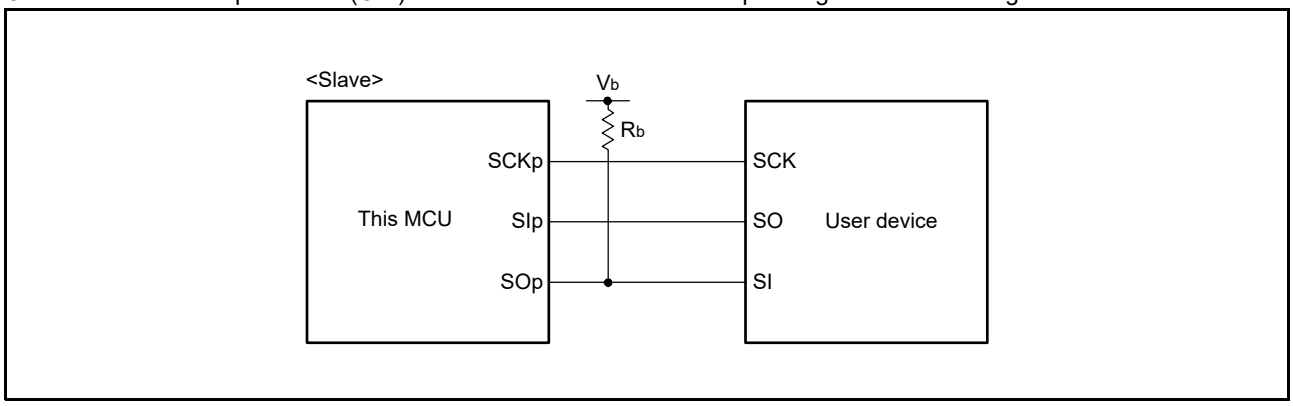
Note 3. This setting applies when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" and SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 4. This setting applies 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.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (withstand voltage of V_{DD} (for the 30- to 52-pin R7F100G and 32- to 48-pin R7F104G products)/withstand voltage of EV_{DD} (for the 64- to 128-pin R7F100G products)) mode for the SOp pin and SCKp pin by using a port input mode register (PIMg) and port output mode register (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

(Remarks are listed on the next page.)

Connection in the simplified SPI (CSI) communications with devices operating at different voltage levels



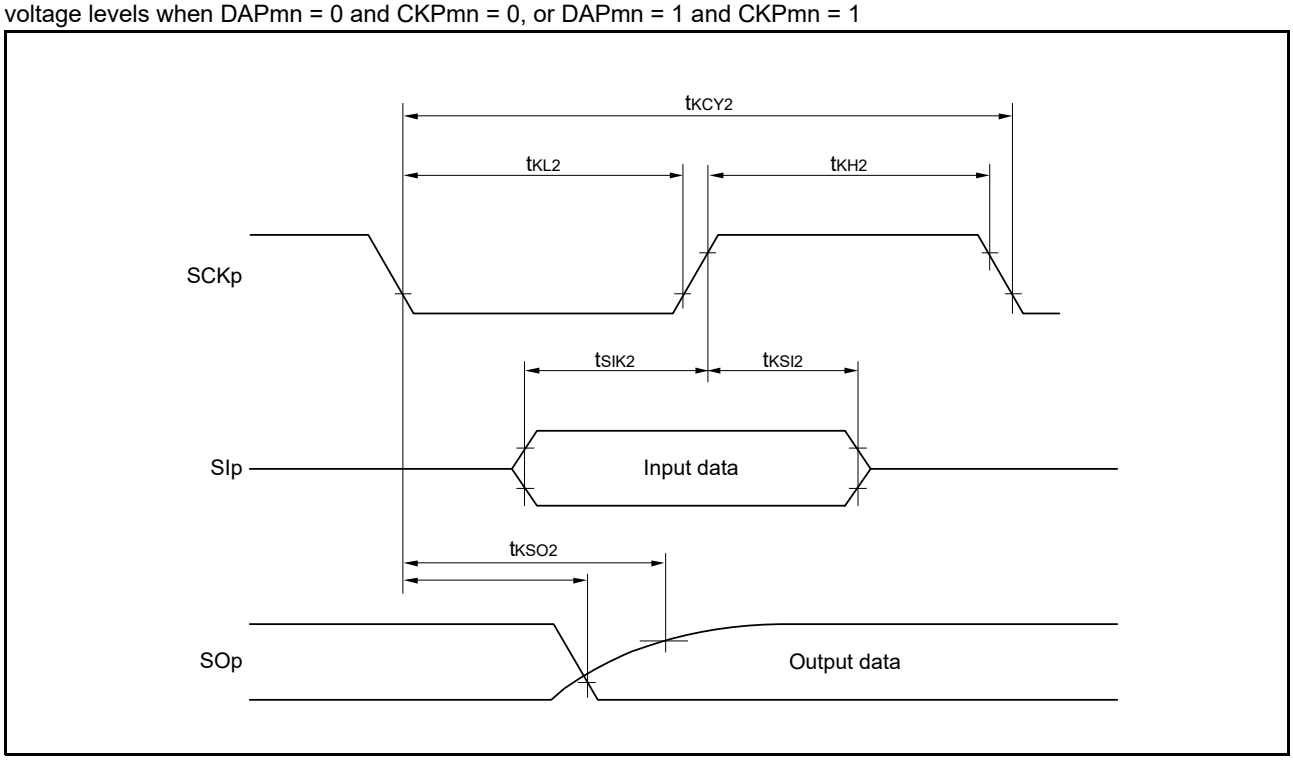
Remark 1. R_b[Ω]: Communication line (SOp) pull-up resistance, C_b[F]: Communication line (SOp) load capacitance, V_b[V]: Communication line voltage

Remark 2. p: CSI number (p = 00, 01, 10, 20, 30, 31), m: Unit number, n: Channel number (mn = 00, 01, 02, 10, 12, 13), g: PIM and POM number (g = 0, 1, 4, 5, 8, 14)

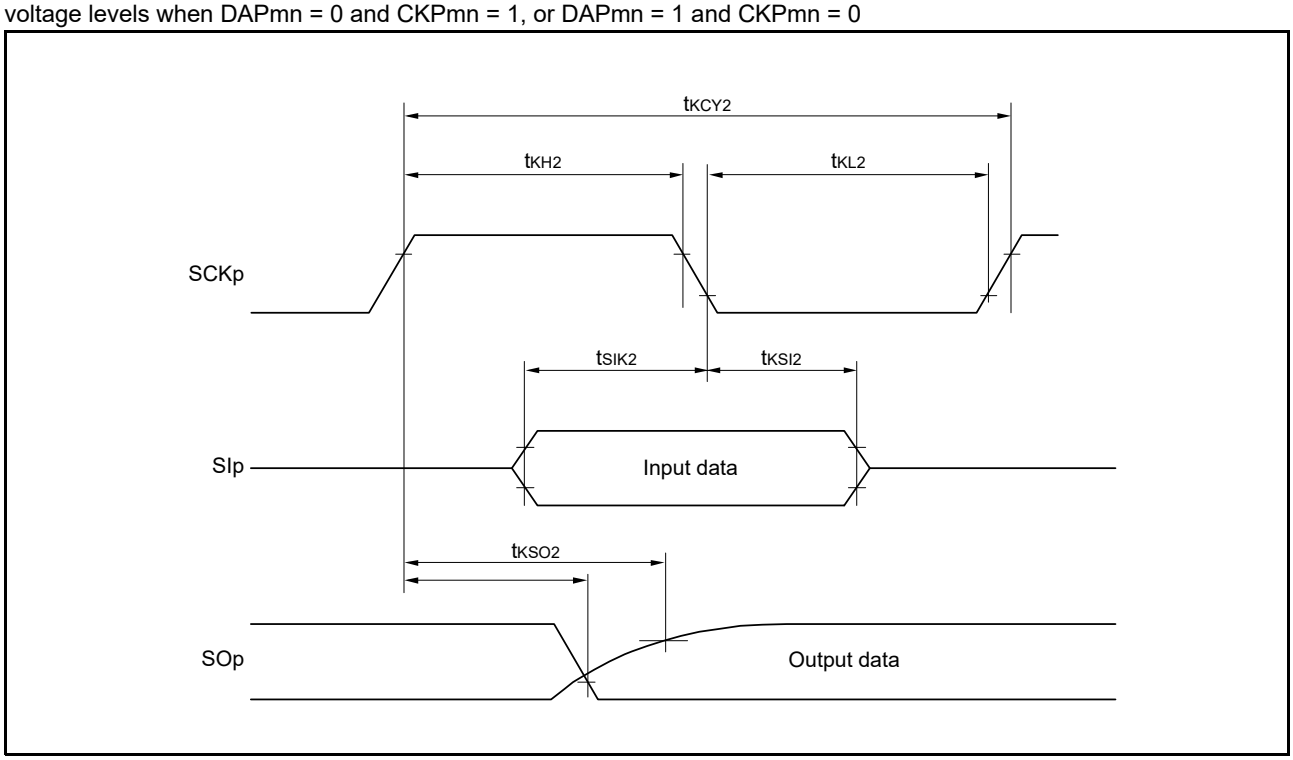
Remark 3. f_{MCK}: Frequency of the operating clock for the serial array unit
To set this operating clock, set the CKSmn bit in serial mode register mn (SMRmn).
m: unit number, n: channel number (mn = 00 to 02, 10, 12, 13)

Remark 4. Communications by using CSI01 of 48- to 64-pin R7F100G products, and CSI11 and CSI21 with devices operating at different voltage levels are not possible. Use other CSI channels to handle such communications.

Timing of serial transfer in the simplified SPI (CSI) communications in the slave mode with devices operating at different voltage levels when DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1



Timing of serial transfer in the simplified SPI (CSI) communications in the slave mode with devices operating at different voltage levels when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0



Remark 1. p: CSI number (p = 00, 01, 10, 20, 30, 31), m: Unit number, n: Channel number (mn = 00, 01, 02, 10, 12, 13), g: PIM and POM number (g = 0, 1, 4, 5, 8, 14)

Remark 2. Communications by using CSI01 of 48- to 64-pin R7F100G products, and CSI11 and CSI21 with devices operating at different voltage levels are not possible. Use other CSI channels to handle such communications.

10. Simplified I²C communications with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V)

(T_A = -40 to +105°C, 1.8 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (1/2)

Item	Symbol	Conditions	HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
SCLr clock frequency	f_{SCL}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ		1000 **Note 1**		1000 **Note 1**		300 **Note 1**	kHz
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ		1000 **Note 1**		1000 **Note 1**		300 **Note 1**	kHz
		4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ		400 **Note 1**		400 **Note 1**		300 **Note 1**	kHz
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ		400 **Note 1**		400 **Note 1**		300 **Note 1**	kHz
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V **Note 2**, C_b = 100 pF, R_b = 5.5 kΩ		300 **Note 1**		300 **Note 1**		300 **Note 1**	kHz
Hold time when SCLr is low	t_{LOW}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	475		475		1550		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	475		475		1550		ns
		4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	1150		1550		1550		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	1150		1550		1550		ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V **Note 2**, C_b = 100 pF, R_b = 5.5 kΩ	1550		1550		1550		ns
Hold time when SCLr is high	t_{HIGH}	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	245		245		610		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	200		200		610		ns
		4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	675		675		610		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	600		600		610		ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V **Note 2**, C_b = 100 pF, R_b = 5.5 kΩ	610		610		610		ns

10. Simplified I2C communications with devices operating at different voltage levels (1.8 V, 2.5 V, and 3 V)

(T_A = -40 to +105°C, 1.8 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V) (2/2)

Item	Symbol	Conditions	HS (High-speed Main) Mode		LS (Low-speed Main) Mode		LP (Low-power Main) Mode		Unit
			Min.	Max.	Min.	Max.	Min.	Max.	
Data setup time (reception)	$t_{SU:DAT}$	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	1/f_{MCK} + 135 **Note 3**		1/f_{MCK} + 135 **Note 3**		1/f_{MCK} + 190 **Note 3**		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	1/f_{MCK} + 135 **Note 3**		1/f_{MCK} + 135 **Note 3**		1/f_{MCK} + 190 **Note 3**		ns
		4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	1/f_{MCK} + 190 **Note 3**		1/f_{MCK} + 190 **Note 3**		1/f_{MCK} + 190 **Note 3**		ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	1/f_{MCK} + 190 **Note 3**		1/f_{MCK} + 190 **Note 3**		1/f_{MCK} + 190 **Note 3**		ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V **Note 2**, C_b = 100 pF, R_b = 5.5 kΩ	1/f_{MCK} + 190 **Note 3**		1/f_{MCK} + 190 **Note 3**		1/f_{MCK} + 190 **Note 3**		ns
Data hold time (transmission)	$t_{HD:DAT}$	4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	0	305	0	305	0	305	ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	0	305	0	305	0	305	ns
		4.0 V ≤ EV_{DD0} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	0	355	0	355	0	355	ns
		2.7 V ≤ EV_{DD0} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	0	355	0	355	0	355	ns
		1.8 V ≤ EV_{DD0} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V **Note 2**, C_b = 100 pF, R_b = 5.5 kΩ	0	405	0	405	0	405	ns

Note 1. The listed times must be no greater than f_{MCK}/4.

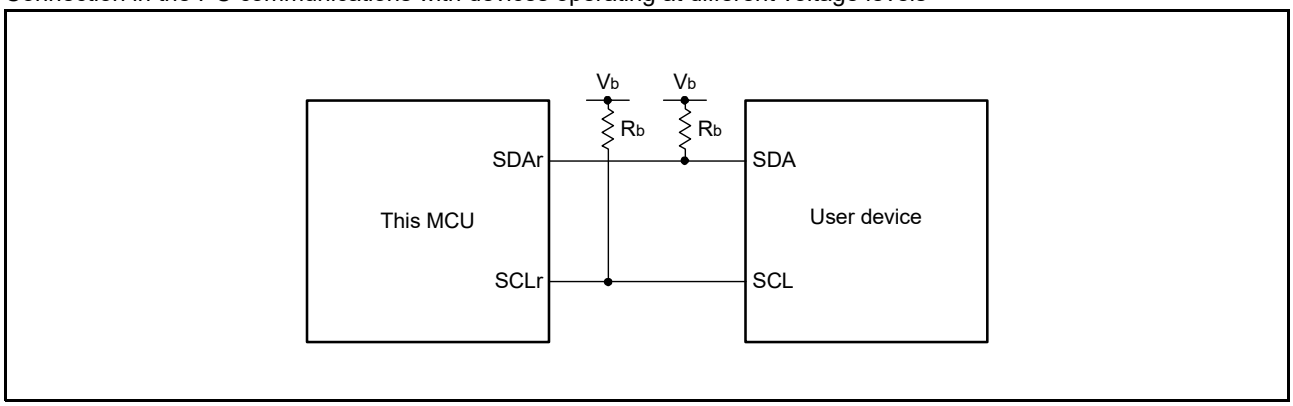
Note 2. Use this setting with EV_{DD0} ≥ V_b.

Note 3. Set f_{MCK} so that it will not exceed the hold time when SCLr is low or high.

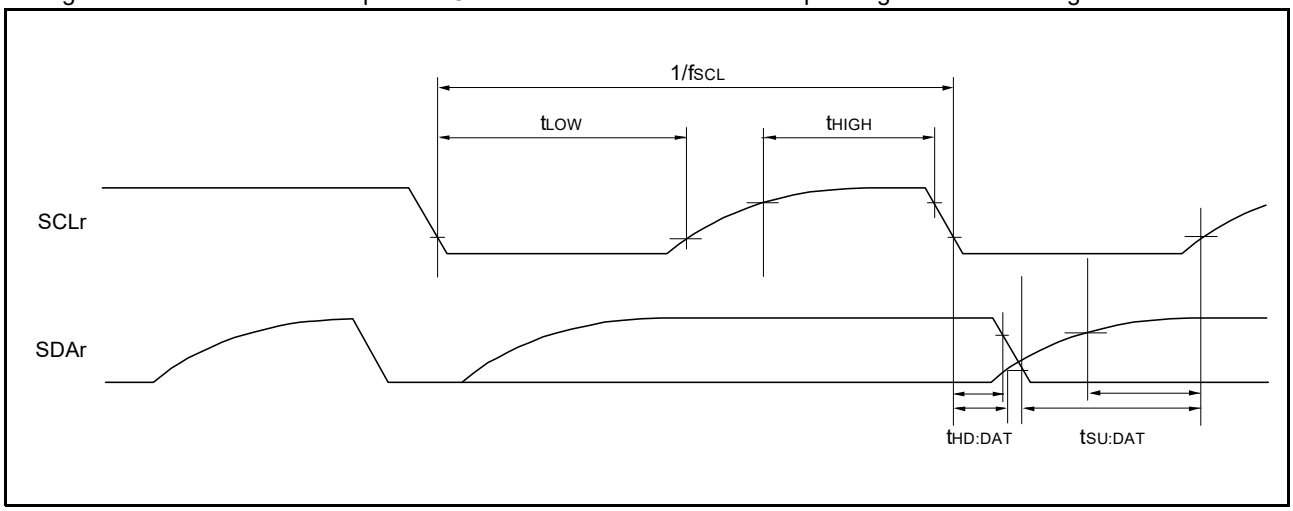
Caution Select the TTL input buffer and the N-ch open drain output (withstand voltage of V_{DD} (for the 30- to 52-pin R7F100G and 32- to 48-pin R7F104G products)/withstand voltage of EV_{DD} (for the 64- to 128-pin R7F100G products)) mode for the SDAr pin and the N-ch open drain output (withstand voltage of V_{DD} (for the 30- to 52-pin R7F100G and 32- to 48-pin R7F104G products)/withstand voltage of EV_{DD} (for the 64- to 128-pin R7F100G products)) mode for the SCLr pin by using a port input mode register (PIMg) and port output mode register (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

(Remarks are listed on the next page.)

Connection in the I²C communications with devices operating at different voltage levels



Timing of serial transfer in the simplified I²C communications with devices operating at different voltage levels



Remark 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

Remark 2. r: IIC number (r = 00, 01, 10, 20, 30, 31), g: PIM and POM number (g = 0, 1, 4, 5, 8, 14)

Remark 3. f_{MCK}: Frequency of the operating clock for the serial array unit
To set this operating clock, set the CKSmn bit in serial mode register mn (SMRmn).
m: unit number, n: channel number (mn = 00 to 02, 10, 12, 13)

37.5.2 Serial interface UARTA

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Transfer rate			200	0	153600	bps

Caution Select the normal input buffer for the RxDAn pin and the normal output mode for the TxDAn pin by using a port input mode register (PIMg) and port output mode register (POMg).

Remark 1. g: PIM number (g = 3, 4, 7, 8), h: POM number (h = 3, 4, 7, 8, 12)
Remark 2. n: Unit number (n = 0, 1)

37.5.3 Serial interface IICA

1. I^2C standard mode

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
SCLA0 clock frequency	f_{SCL}	Standard mode: f_{CLK} ≥ 1 MHz	0		100	kHz
Setup time of restart condition	$t_{SU:STA}$		4.7			μs
Hold time[Note 1]	$t_{HD:STA}$		4.0			μs
Hold time when SCLA0 is low	t_{LOW}		4.7			μs
Hold time when SCLA0 is high	t_{HIGH}		4.0			μs
Data setup time (reception)	$t_{SU:DAT}$		250			ns
Data hold time (transmission)[Note 2]	$t_{HD:DAT}$		0		3.45	μs
Setup time of stop condition	$t_{SU:STO}$		4.0			μs
Bus-free time	t_{BUF}		4.7			μs

Note 1. The first clock pulse is generated after this period when the start or restart condition is detected.

Note 2. The maximum value of $t_{HD:DAT}$ applies to normal transfer. The clock stretching will be inserted on reception of an acknowledgment (ACK) signal.

Caution **The listed frequency and times apply even when the PIOR2 bit in the peripheral I/O redirection register (PIOR) is 1. In such cases, the pin characteristics (I_{OH1}, I_{OL1}, V_{OH1}, V_{OL1}) must satisfy the values in the redirect destination.**

Remark The maximum value of communication line capacitance (C_b) and communication line pull-up resistor (R_b) are as follows.
C_b = 400 pF, R_b = 2.7 kΩ

2. I^2C fast mode

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
SCLA0 clock frequency	f_{SCL}	Fast mode: f_{CLK} ≥ 3.5 MHz 1.8 V ≤ EV_{DD0} ≤ 5.5 V	0		400	kHz
Setup time of restart condition	$t_{SU:STA}$	1.8 V ≤ EV_{DD0} ≤ 5.5 V	0.6			μs
Hold time[Note 1]	$t_{HD:STA}$	1.8 V ≤ EV_{DD0} ≤ 5.5 V	0.6			μs
Hold time when SCLA0 is low	t_{LOW}	1.8 V ≤ EV_{DD0} ≤ 5.5 V	1.3			μs
Hold time when SCLA0 is high	t_{HIGH}	1.8 V ≤ EV_{DD0} ≤ 5.5 V	0.6			μs
Data setup time (reception)	$t_{SU:DAT}$	1.8 V ≤ EV_{DD0} ≤ 5.5 V	100			ns
Data hold time (transmission)[Note 2]	$t_{HD:DAT}$	1.8 V ≤ EV_{DD0} ≤ 5.5 V	0		0.9	μs
Setup time of stop condition	$t_{SU:STO}$	1.8 V ≤ EV_{DD0} ≤ 5.5 V	0.6			μs
Bus-free time	t_{BUF}	1.8 V ≤ EV_{DD0} ≤ 5.5 V	1.3			μs

Note 1. The first clock pulse is generated after this period when the start or restart condition is detected.

Note 2. The maximum value of $t_{HD:DAT}$ applies to normal transfer. The clock stretching will be inserted on reception of an acknowledgment (ACK) signal.

Caution **The values in the above table apply even when the PIOR2 bit in the peripheral I/O redirection register (PIOR) is 1. In such cases, the pin characteristics (I_{OH1}, I_{OL1}, V_{OH1}, V_{OL1}) must satisfy the values in the redirect destination.**

Remark The maximum value of communication line capacitance (C_b) and communication line pull-up resistor (R_b) are as follows.
C_b = 320 pF, R_b = 1.1 kΩ

3. I^2C fast mode plus

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
SCLA0 clock frequency	f_{SCL}	Fast mode plus: f_{CLK} ≥ 10 MHz 2.7 V ≤ EV_{DD0} ≤ 5.5 V	0		1000	kHz
Setup time of restart condition	$t_{SU:STA}$	2.7 V ≤ EV_{DD0} ≤ 5.5 V	0.26			µs
Hold time[Note 1]	$t_{HD:STA}$	2.7 V ≤ EV_{DD0} ≤ 5.5 V	0.26			µs
Hold time when SCLA0 is low	t_{LOW}	2.7 V ≤ EV_{DD0} ≤ 5.5 V	0.5			µs
Hold time when SCLA0 is high	t_{HIGH}	2.7 V ≤ EV_{DD0} ≤ 5.5 V	0.26			µs
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ EV_{DD0} ≤ 5.5 V	50			ns
Data hold time (transmission)[Note 2]	$t_{HD:DAT}$	2.7 V ≤ EV_{DD0} ≤ 5.5 V	0		0.45	µs
Setup time of stop condition	$t_{SU:STO}$	2.7 V ≤ EV_{DD0} ≤ 5.5 V	0.26			µs
Bus-free time	t_{BUF}	2.7 V ≤ EV_{DD0} ≤ 5.5 V	0.5			µs

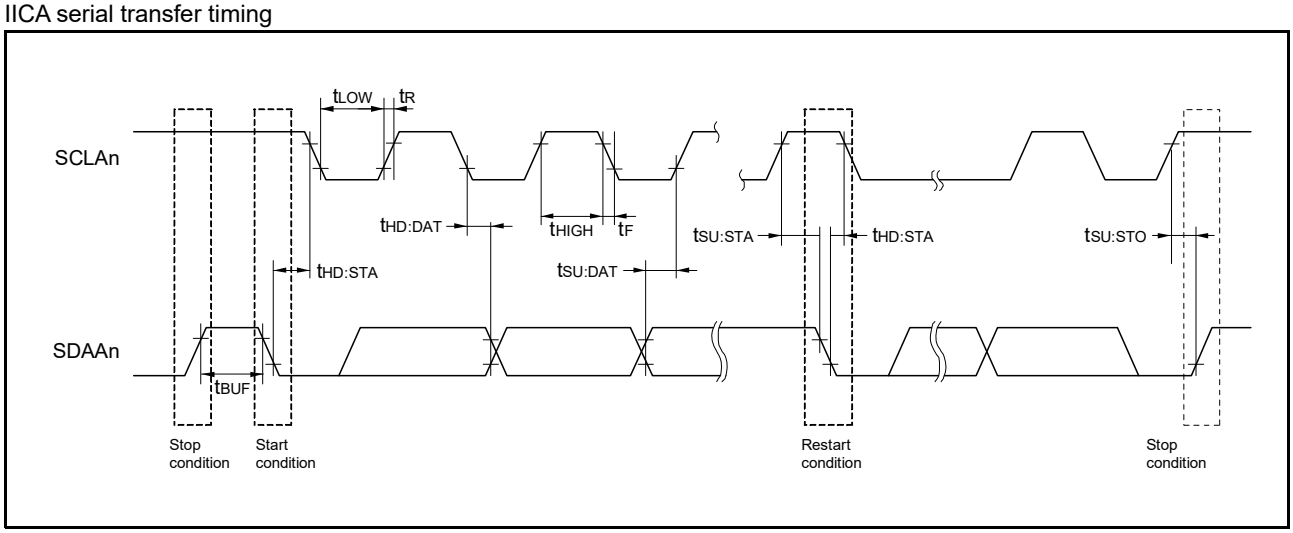
Note 1. The first clock pulse is generated after this period when the start or restart condition is detected.

Note 2. The maximum value of $t_{HD:DAT}$ applies to normal transfer. The clock stretching will be inserted on reception of an acknowledgment (ACK) signal.

Caution The values in the above table apply even when the PIOR2 bit in the peripheral I/O redirection register (PIOR) is 1. In such cases, the pin characteristics (I_{OH1}, I_{OL1}, V_{OH1}, V_{OL1}) must satisfy the values in the redirect destination.

Remark The maximum value of communication line capacitance (C_b) and communication line pull-up resistor (R_b) are as follows.
C_b = 120 pF, R_b = 1.1 kΩ

IICA serial transfer timing



Remark n = 0, 1

37.6 Characteristics of the Analog Circuits

37.6.1 Characteristics of the A/D converter

1. Normal modes 1 and 2

(T_A = -40 to +105°C, 2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V,
reference voltage (+) = AV_{REFP} (ADREFP1 = 0, ADREFP0 = 1), reference voltage (-) = AV_{REFM} (ADREFM = 1),
target pins: ANI2 to ANI14, internal reference voltage, and temperature sensor output voltage)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Resolution	RES				12	bit
Conversion clock	f_{AD}		1		32	MHz
Overall error **Notes 1, 2, 3, 4, 5**	AINL	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±2.4	±4.5	LSB
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±2.9	±5.7	LSB
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±3.0	±5.8	LSB
Conversion time **Notes 5, 6**	t_{CONV}	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	2.0			µs
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	2.0			µs
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	2.0			µs
Zero-scale error **Notes 1, 2, 3, 4, 5, 7**	E_{ZS}	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±0.01	±0.08	%FSR
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±0.01	±0.09	%FSR
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±0.03	±0.13	%FSR
Full-scale error **Notes 1, 2, 3, 4, 5, 7**	E_{FS}	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±0.03	±0.09	%FSR
		2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±0.05	±0.13	%FSR
		2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±0.05	±0.13	%FSR
Analog input voltage	V_{AIN}		0		AV_{REFP}	V

Note 1. This value does not include the quantization error (±1/2 LSB).

Note 2. The values in the column Max. only apply in the case of a normal distribution with ±3σ variation from the mean.

Note 3. We do not inspect the characteristics of the A/D converter before shipment. The listed values are only results of evaluation.

Note 4. When AV_{REFP} < V_{DD}, the maximum values are as follows.

Overall error/zero-scale error/full-scale error: Add (±0.75 LSB × (V_{DD} voltage (V) - AV_{REFP} voltage (V)) to the maximum value.

Integral linearity error: Add (±0.2 LSB × (V_{DD} voltage (V) - AV_{REFP} voltage (V)) to the maximum value.

Note 5. The listed values apply when the conversion resolution is set to 12 bits.

Note 6. When the internal reference voltage or the temperature sensor output voltage is selected as the target for conversion, the sampling time must be at least 5 µs. Accordingly, use standard mode 2 with the longer sampling time.

Note 7. This value is indicated as a ratio (%FSR) to the full-scale value.

(T_A = -40 to +105°C, 2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V,
reference voltage (+) = AV_{REFP} (ADREFP1 = 0, ADREFP0 = 1), reference voltage (-) = AV_{REFM} (ADREFM = 1),
target pins: ANI2 to ANI14, internal reference voltage, and temperature sensor output voltage)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Resolution	RES			8		12	Bit
Conversion clock	f_{AD}			1		32	MHz
Overall error Notes 1, 2, 3, 4	AINL	12-bit resolution	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±7.5	LSB
			2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±9.0	LSB
			2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±9.0	LSB
Conversion time Note 5	t_{CONV}	12-bit resolution	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	2.0			µs
			2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	2.0			µs
			2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	2.0			µs
Zero-scale error Notes 1, 2, 3, 4, 6	E_{ZS}	12-bit resolution	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.17	%FSR
			2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
			2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
Full-scale error Notes 1, 2, 3, 4, 6	E_{FS}	12-bit resolution	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.17	%FSR
			2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
			2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
Integral linearity error Notes 1, 3, 4	ILE	12-bit resolution	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±3.0	LSB
			2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±3.0	LSB
			2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±3.0	LSB
Differential linearity error Note 1	DLE	12-bit resolution	4.5 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±1.0		LSB
			2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±1.0		LSB
			2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±1.0		LSB
Analog input voltage	V_{AIN}			0		AV_{REFP}	V

Note 1. This value does not include the quantization error (±1/2 LSB).

Note 2. When pins ANI16 to ANI26 are selected as the target pins for conversion or the TSCAP voltage of the CTSU is to be A/D converted, the maximum values are as follows.
Overall error: Add ±3 LSB to the maximum value.
Zero-scale/full-scale error: Add ±0.04%FSR to the maximum value.

Note 3. When reference voltage (+) = V_{DD} and reference voltage (-) = V_{SS}, the maximum values are as follows.
Overall error: Add ±10 LSB to the maximum value.
Zero-scale/full-scale error: Add ±0.25%FSR to the maximum value.
Integral linearity error: Add ±4 LSB to the maximum value.

Note 4. When AV_{REFP} < V_{DD}, the maximum values are as follows.
Overall error/zero-scale error/full-scale error: Add (±0.75 LSB × (V_{DD} voltage (V) - AV_{REFP} voltage (V)) to the maximum value.
Integral linearity error: Add (±0.2 LSB × (V_{DD} voltage (V) - AV_{REFP} voltage (V)) to the maximum value.

Note 5. When the internal reference voltage or the temperature sensor output voltage is selected as the target for conversion, the sampling time must be at least 5 µs. Accordingly, use standard mode 2 with the longer sampling time.

Note 6. This value is indicated as a ratio (%FSR) to the full-scale value.

2. Low-voltage modes 1 and 2

(T_A = -40 to +105°C, 1.6 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V,
reference voltage (+) = AV_{REFP} (ADREFP1 = 0, ADREFP0 = 1), reference voltage (-) = AV_{REFM} (ADREFM = 1),
target pins ANI2 to ANI14, internal reference voltage**Note 1**, and temperature sensor output voltage**Note 1**)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Resolution	RES			8		12	Bit
Conversion clock	f_{AD}			1		24	MHz
Overall error**Notes 2, 3, 4, 5**	AINL	12-bit resolution	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±9	LSB
			2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±9	LSB
			1.8 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±11.5	LSB
			1.6 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±12.0	LSB
Conversion time**Note 6**	t_{CONV}	12-bit resolution	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	3.33			µs
			2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	5.0			µs
			1.8 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	10.0			µs
			1.6 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V	20.0			µs
Zero-scale error**Notes 2, 3, 4, 5, 7**	E_{ZS}	12-bit resolution	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
			2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
			1.8 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.27	%FSR
			1.6 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.28	%FSR
Full-scale error**Notes 2, 3, 4, 5, 7**	E_{FS}	12-bit resolution	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
			2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.21	%FSR
			1.8 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.27	%FSR
			1.6 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±0.28	%FSR
Integral linearity error**Notes 2, 4, 5**	ILE	12-bit resolution	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±4.0	LSB
			2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±4.0	LSB
			1.8 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±4.5	LSB
			1.6 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V			±4.5	LSB
Differential linearity error**Note 2**	DLE	12-bit resolution	2.7 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±1.5		LSB
			2.4 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±1.5		LSB
			1.8 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±2.0		LSB
			1.6 V ≤ AV_{REFP} = V_{DD} ≤ 5.5 V		±2.0		LSB
Analog input voltage	V_{AIN}			0		AV_{REFP}	V

Note 1. If the internal reference voltage, temperature sensor output voltage, or TSCAP voltage of the CTSU is to be A/D converted, V_{DD} must be at least 1.8 V.

Note 2. This value does not include the quantization error (±1/2 LSB).

Note 3. When pins ANI16 to ANI26 are selected as the target pins for conversion or the TSCAP voltage of the CTSU is to be A/D converted, the maximum values are as follows.
Overall error: Add ±3 LSB to the maximum value.
Zero-scale/full-scale error: Add ±0.04%FSR to the maximum value.

Note 4. When reference voltage (+) = V_{DD} and reference voltage (-) = V_{SS}, the maximum values are as follows.
Overall error: Add ±10 LSB to the maximum value.
Zero-scale/full-scale error: Add ±0.25%FSR to the maximum value.
Integral linearity error: Add ±4 LSB to the maximum value.

Note 5. When AV_{REFP} < V_{DD}, the maximum values are as follows.
Overall error/zero-scale error/full-scale error: Add (±0.75 LSB × (V_{DD} voltage (V) - AV_{REFP} voltage (V)) to the maximum value.
Integral linearity error: Add (±0.2 LSB × (V_{DD} voltage (V) - AV_{REFP} voltage (V)) to the maximum value.

Note 6. When the internal reference voltage or the temperature sensor output voltage is selected as the target for conversion, the sampling time must be at least 5 μs. Accordingly, use standard mode 2 with the longer sampling time, and use the conversion clock (f_{AD}) of no more than 16 MHz.

Note 7. This value is indicated as a ratio (%FSR) to the full-scale value.

3. When the internal reference voltage is selected as reference voltage (+)

(T_A = -40 to +105°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, low-voltage modes 1 and 2,
reference voltage (+) = internal reference voltage (ADREFP1 = 1, ADREFP0 = 0),
reference voltage (-) = AV_{REFM} (ADREFM = 1)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Resolution	RES		8			Bit
Conversion clock	f_{AD}	1.6 V ≤ V_{DD} ≤ 5.5 V	1		2	MHz
Zero-scale error **Notes 1, 2, 3**	E_{ZS}	1.6 V ≤ V_{DD} ≤ 5.5 V			±0.6	%FSR
Integral linearity error **Notes 1, 3**	ILE	1.6 V ≤ V_{DD} ≤ 5.5 V			±2.0	LSB
Differential linearity error **Note 1**	DLE	1.6 V ≤ V_{DD} ≤ 5.5 V		±1.0		LSB
Analog input voltage	V_{AIN}		0		V_{BGR} **Note 4**	V

Note 1. This value does not include the quantization error (±1/2 LSB).

Note 2. This value is indicated as a ratio (%FSR) to the full-scale value.

Note 3. When reference voltage (-) is selected as V_{SS}, the maximum values are as follows.
Zero-scale error: Add ±0.35%FSR to the maximum value.
Integral linearity error: Add ±0.5 LSB to the maximum value.

Note 4. Refer to **37.6.2 Characteristics of the temperature sensor and internal reference voltage**.

37.6.2 Characteristics of the temperature sensor and internal reference voltage

(T_A = -40 to +105°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Temperature sensor output voltage	V_{TMPS25}	Setting ADS register = 80H, T_A = +25°C		1.05		V
Internal reference voltage	V_{BGR}	Setting ADS register = 81H	1.42	1.48	1.54	V
Temperature coefficient	F_{VTMPS}	Temperature dependency of the temperature sensor voltage		-3.3		mV/°C
Operation stabilization wait time	t_{AMP}		5			μs

37.6.3 Characteristics of the D/A converter

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Resolution	RES					8	Bit
Overall error	AINL	Rload = 8 MΩ	1.8 V ≤ V_{DD} ≤ 5.5 V			±2.5	LSB
		Rload = 4 MΩ	1.8 V ≤ V_{DD} ≤ 5.5 V			±2.5	LSB
Settling time	t_{SET}	Cload = 20 pF	2.7 V ≤ V_{DD} ≤ 5.5 V			3	μs
			1.6 V ≤ V_{DD} ≤ 5.5 V			6	μs

37.6.4 Characteristics of the comparator

(T_A = -40 to +105°C, 1.6 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V)

Item	Symbol	Conditions		Min.	Typ.	Max.	Unit
Input voltage range	I_{VREF}	Input to the IVREF0 and IVREF1 pins C0LVL = 0, C1LVL = 0		0		V_{DD} - 1.4 and EV_{DD0}	V
		Input to the IVREF0 and IVREF1 pins C0LVL = 1, C1LVL = 1		1.4		EV_{DD0}	V
	I_{VCMP}	Input to the IVCMP0 and IVCMP1 pins		-0.3		EV_{DD0} + 0.3	V
Output delay	td	V_{DD} = 3.0 V, input slew rate > 1 V/μs	High-speed mode			1.5	μs
			Low-speed mode		3.0		μs
Offset voltage	—	High-speed mode				50	mV
		Low-speed mode				40	mV
Operation stabilization wait time	t_{CMP}			30			μs
Internal reference voltage[Note]	V_{BGR2}			1.4		1.6	V

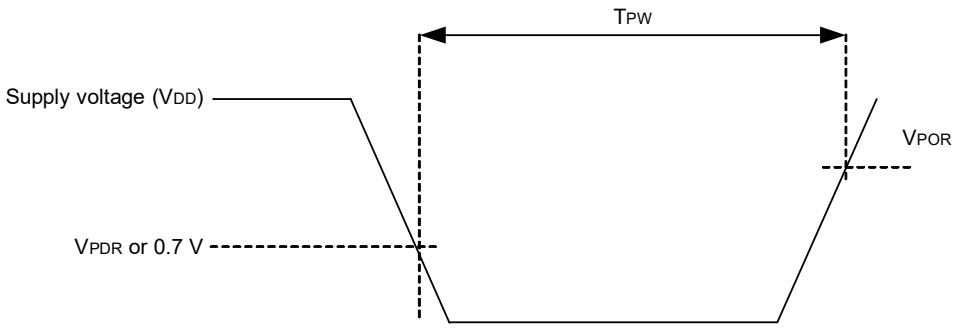
Note The internal reference voltage can be selected as comparator reference voltage only when 1.8 V ≤ V_{DD} ≤ 5.5 V.

37.6.5 Characteristics of the POR circuit

(T_A = -40 to +105°C, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Detection voltage	V_{POR}, V_{PDR}		1.43	1.50	1.57	V
Minimum pulse width**Note**	T_{PW}		300			μs

Note This width is the minimum time required for a POR reset when V_{DD} falls below V_{PDR}. This width is also the minimum time required for a POR reset from when V_{DD} falls below 0.7 V to when V_{DD} exceeds V_{POR} in the STOP mode or while the main system clock is stopped through setting the HIOSTOP and MSTOP bits in the clock operation status control register (CSC).



37.6.6 Characteristics of the LVD circuit

1. LVD0 Detection Voltage in the Reset Mode and Interrupt Mode

(T_A = -40 to +105°C, $V_{PDR} \leq V_{DD} \leq 5.5$ V, V_{SS} = 0 V)

Item		Symbol	Conditions	Min.	Typ.	Max.	Unit
Detection voltage	Supply voltage level	V_{LVD00}	The power supply voltage is rising.	3.84	3.96	4.08	V
			The power supply voltage is falling.	3.76	3.88	4.00	V
		V_{LVD01}	The power supply voltage is rising.	2.88	2.97	3.06	V
			The power supply voltage is falling.	2.82	2.91	3.00	V
		V_{LVD02}	The power supply voltage is rising.	2.59	2.67	2.75	V
			The power supply voltage is falling.	2.54	2.62	2.70	V
		V_{LVD03}	The power supply voltage is rising.	2.31	2.38	2.45	V
			The power supply voltage is falling.	2.26	2.33	2.40	V
		V_{LVD04}	The power supply voltage is rising.	1.84	1.90	1.95	V
			The power supply voltage is falling.	1.80	1.86	1.91	V
		V_{LVD05}	The power supply voltage is rising.	1.64	1.69	1.74	V
			The power supply voltage is falling.	1.60	1.65	1.70	V
Minimum pulse width		t_{LW}		500			μs
Detection delay time						500	μs

2. LVD1 Detection Voltage of Reset Mode and Interrupt Mode

(T_A = -40 to +105°C, $V_{PDR} \leq V_{DD} \leq 5.5$ V, V_{SS} = 0 V)

Item		Symbol	Conditions	Min.	Typ.	Max.	Unit
Detection voltage	Supply voltage level	V_{LVD10}	The power supply voltage is rising.	4.08	4.16	4.24	V
			The power supply voltage is falling.	4.00	4.08	4.16	V
		V_{LVD11}	The power supply voltage is rising.	3.88	3.96	4.04	V
			The power supply voltage is falling.	3.80	3.88	3.96	V
		V_{LVD12}	The power supply voltage is rising.	3.68	3.75	3.82	V
			The power supply voltage is falling.	3.60	3.67	3.74	V
		V_{LVD13}	The power supply voltage is rising.	3.48	3.55	3.62	V
			The power supply voltage is falling.	3.40	3.47	3.54	V
		V_{LVD14}	The power supply voltage is rising.	3.28	3.35	3.42	V
			The power supply voltage is falling.	3.20	3.27	3.34	V
		V_{LVD15}	The power supply voltage is rising.	3.07	3.13	3.19	V
			The power supply voltage is falling.	3.00	3.06	3.12	V
		V_{LVD16}	The power supply voltage is rising.	2.91	2.97	3.03	V
			The power supply voltage is falling.	2.85	2.91	2.97	V
		V_{LVD17}	The power supply voltage is rising.	2.76	2.82	2.87	V
			The power supply voltage is falling.	2.70	2.76	2.81	V
		V_{LVD18}	The power supply voltage is rising.	2.61	2.66	2.71	V
			The power supply voltage is falling.	2.55	2.60	2.65	V
		V_{LVD19}	The power supply voltage is rising.	2.45	2.50	2.55	V
			The power supply voltage is falling.	2.40	2.45	2.50	V
		V_{LVD110}	The power supply voltage is rising.	2.35	2.40	2.45	V
			The power supply voltage is falling.	2.30	2.35	2.40	V
		V_{LVD111}	The power supply voltage is rising.	2.25	2.30	2.34	V
			The power supply voltage is falling.	2.20	2.25	2.29	V
		V_{LVD112}	The power supply voltage is rising.	2.15	2.20	2.24	V
			The power supply voltage is falling.	2.10	2.15	2.19	V
		V_{LVD113}	The power supply voltage is rising.	2.05	2.09	2.13	V
			The power supply voltage is falling.	2.00	2.04	2.08	V
		V_{LVD114}	The power supply voltage is rising.	1.94	1.98	2.02	V
			The power supply voltage is falling.	1.90	1.94	1.98	V
		V_{LVD115} **Note**	The power supply voltage is rising.	1.84	1.88	1.91	V
			The power supply voltage is falling.	1.80	1.84	1.87	V
		V_{LVD116} **Note**	The power supply voltage is rising.	1.74	1.78	1.81	V
			The power supply voltage is falling.	1.70	1.74	1.77	V
		V_{LVD117} **Note**	The power supply voltage is rising.	1.64	1.67	1.70	V
			The power supply voltage is falling.	1.60	1.63	1.66	V
Minimum pulse width		t_{LW}		500			μs
Detection delay time						500	μs

Note This setting can only be used when LVD0 is disabled.

37.6.7 Characteristics of the rising slope of the power supply voltage

(T_A = -40 to +105°C, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Power supply voltage rising slope	S_{VDD}				54	V/ms

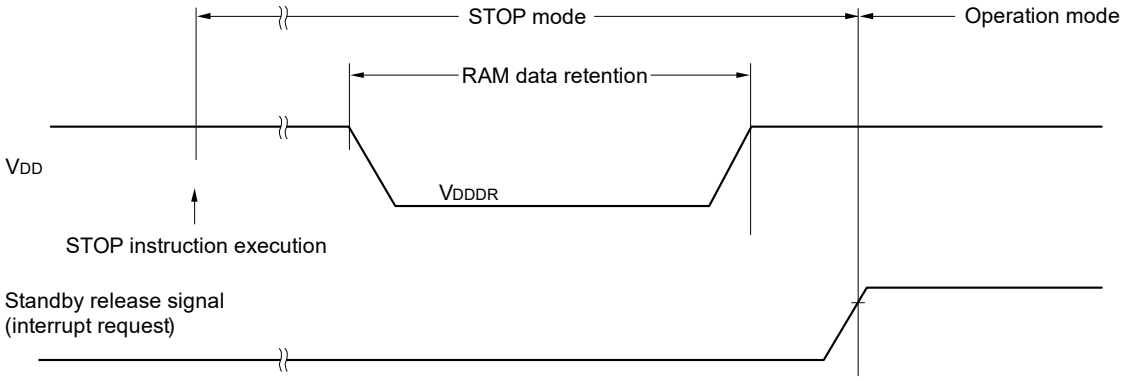
Caution Make sure to keep the internal reset state by LVD0 or an external reset until V_{DD} reaches the operating voltage range shown in AC characteristics.

37.7 Characteristics of Retention of RAM Data

(T_A = -40 to +105°C, V_{SS} = 0V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Data retention supply voltage	V_{DDDR}		1.43Note		5.5	V

Note This voltage depends on the POR detection voltage. When the voltage drops, the data in RAM are retained until a POR is applied, but are not retained following a POR.



37.8 Characteristics of Flash Memory Programming

(T_A = -40 to +105°C, 1.6 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
CPU/peripheral hardware clock frequency	f_{CLK}		1		32	MHz
Number of code flash memory rewrites Notes 1, 2, 3	C_{erwr}	Retained for 20 years T_A = +85°C	1,000			Times
Number of data flash memory rewrites Notes 1, 2, 3		Retained for 1 year T_A = +25°C		1,000,000		
		Retained for 5 years T_A = +85°C	100,000			
		Retained for 20 years T_A = +85°C	10,000			

Note 1. 1 erase + 1 write after the erase is regarded as 1 rewrite. The retaining years are until next rewrite after the rewrite.

Note 2. The listed numbers of times apply when using flash memory programmer and Renesas Electronics self programming library.

Note 3. These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas Electronics Corporation.

1. Code flash memory

(T_A = -40 to +105°C, 1.6 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Item		Symbol	f_{CLK} = 1 MHz			f_{CLK} = 2 MHz, 3 MHz			4 MHz ≤ f_{CLK} < 8 MHz			8 MHz ≤ f_{CLK} < 32 MHz			f_{CLK} = 32 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time	4 bytes	t_{P4}	—	74.7	656.5	—	51.0	464.6	—	41.7	384.8	—	37.1	346.2	—	34.2	321.9	µs
Erasure time	2 Kbytes	t_{E2K}	—	10.4	312.2	—	7.7	258.5	—	6.4	231.8	—	5.8	218.4	—	5.6	214.4	ms
Blank checking time	4 bytes	t_{BC4}	—	—	38.4	—	—	19.2	—	—	13.1	—	—	10.2	—	—	8.3	µs
	2 Kbytes	t_{BC2K}	—	—	2618.9	—	—	1309.5	—	—	658.3	—	—	332.8	—	—	234.1	µs
Time taken to forcibly stop the erasure		t_{SED}	—	—	18.0	—	—	14.0	—	—	12.0	—	—	11.0	—	—	10.3	µs
Security setting time		t_{AWSSAS}	—	18.2	526.2	—	14.4	469.2	—	12.5	441.1	—	11.6	427.1	—	11.3	422.6	ms
Time until programming starts following cancellation of the STOP instruction		—	20	—	—	20	—	—	20	—	—	20	—	—	20	—	—	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

(T_A = -40 to +105°C, 1.6 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Item		Symbol	f_{CLK} = 1 MHz			f_{CLK} = 2 MHz, 3 MHz			4 MHz ≤ f_{CLK} < 8 MHz			8 MHz ≤ f_{CLK} < 32 MHz			f_{CLK} = 32 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time	1 byte	t_{P4}	—	74.7	656.5	—	51.0	464.6	—	41.7	384.8	—	37.1	346.2	—	34.2	321.9	µs
Erasure time	256 bytes	t_{E2K}	—	7.8	259.2	—	6.4	232.0	—	5.8	218.5	—	5.5	211.8	—	5.4	209.7	ms
Blank checking time	1 byte	t_{BC4}	—	—	38.4	—	—	19.2	—	—	13.1	—	—	10.2	—	—	8.3	µs
	256 bytes	t_{BC2K}	—	—	1326.1	—	—	663.1	—	—	335.1	—	—	171.2	—	—	121.0	µs
Time taken to forcibly stop the erasure		t_{SED}	—	—	18.0	—	—	14.0	—	—	12.0	—	—	11.0	—	—	10.3	µs
Time until programming starts following cancellation of the STOP instruction		—	20	—	—	20	—	—	20	—	—	20	—	—	20	—	—	µs
Time until reading starts following setting DFLCTL.DFLEN to 1		—	0.25	—	—	0.25	—	—	0.25	—	—	0.25	—	—	0.25	—	—	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.

37.9 Dedicated Flash Memory Programmer Communication (UART)

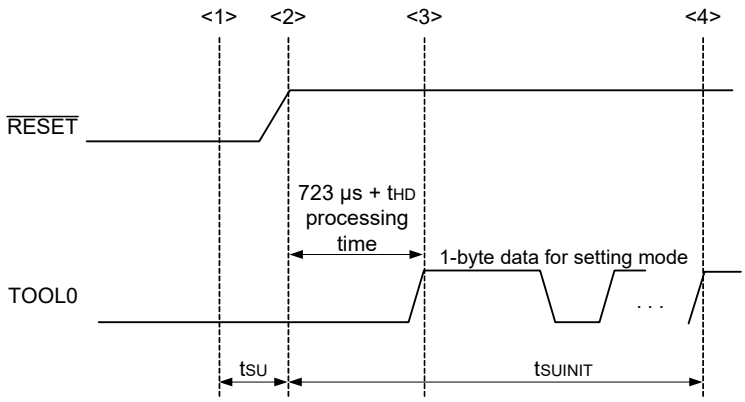
(T_A = -40 to +105°C, 1.8 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Transfer rate		During serial programming	115,200		1,000,000	bps

37.10 Timing of Entry to Flash Memory Programming Modes

(T_A = -40 to +105°C, 1.8 V ≤ EV_{DD0} = EV_{DD1} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS0} = EV_{SS1} = 0 V)

Item	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 (the processing time of the firmware to control the flash memory is not included)	t_{HD}	POR and LVD reset must be released before the external reset is released.	1			ms



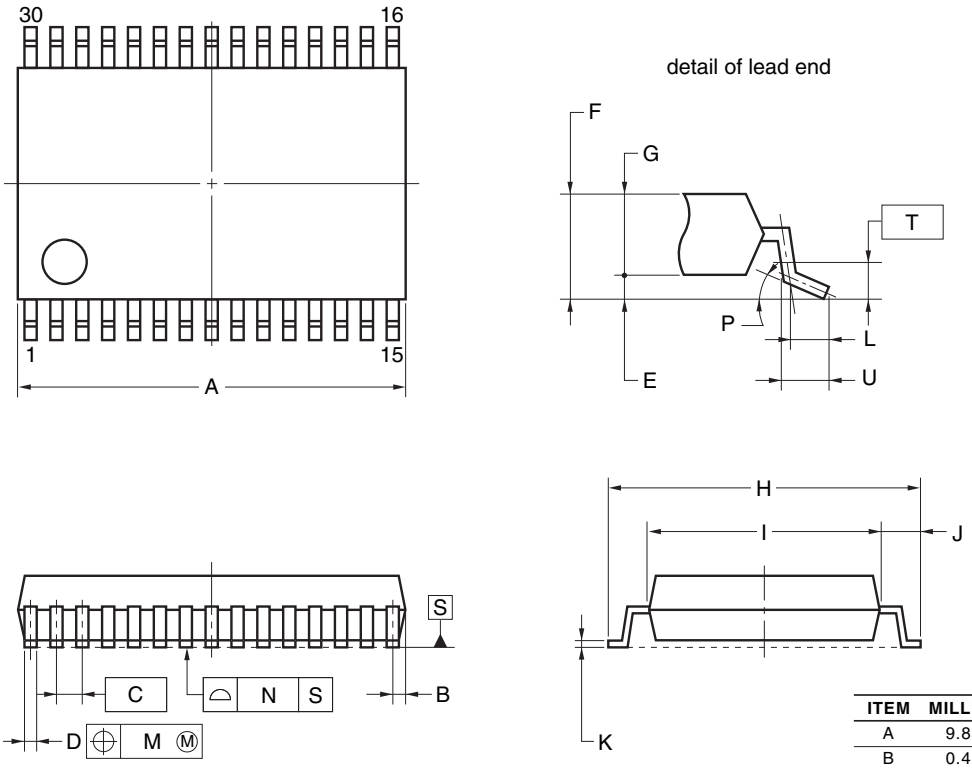
<1> The low level is input to the TOOL0 pin.
<2> The external reset is released. Note that the POR and LVD reset must be released before the external reset is released.
<3> The TOOL0 pin is set to the high level.
<4> Setting of the flash memory programming mode by UART reception and complete the baud rate setting.

Remark t_{SUINIT}: The time during which the communications for the initial setting must be completed within 100 ms after the external reset is released.
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. It does not include the processing time of the firmware to control the flash memory.

Section 38 Package Drawings

38.1 30-pin Products

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LSSOP30-0300-0.65	PLSP0030JB-B	S30MC-65-5A4-3	0.18



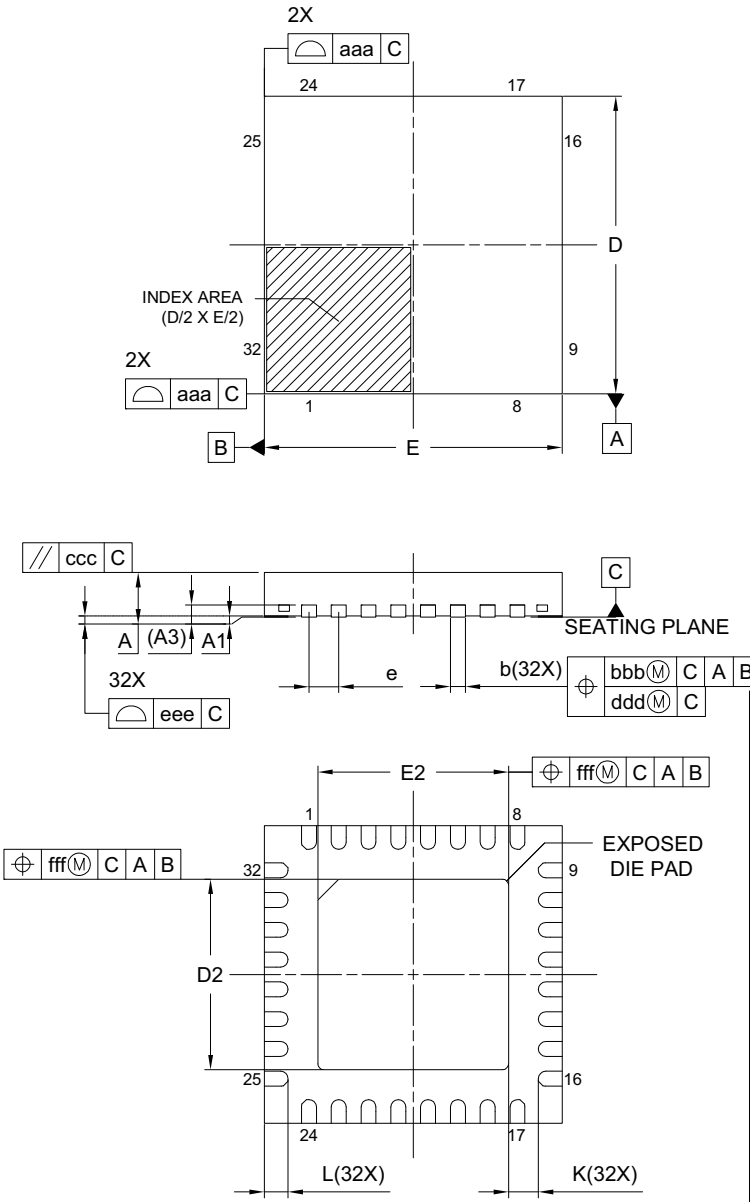
NOTE
Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	9.85±0.15
B	0.45 MAX.
C	0.65 (T.P.)
D	$0.24^{+0.08}_{-0.07}$
E	0.1±0.05
F	1.3±0.1
G	1.2
H	8.1±0.2
I	6.1±0.2
J	1.0±0.2
K	0.17±0.03
L	0.5
M	0.13
N	0.10
P	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
T	0.25
U	0.6±0.15

©2012 Renesas Electronics Corporation. All rights reserved.

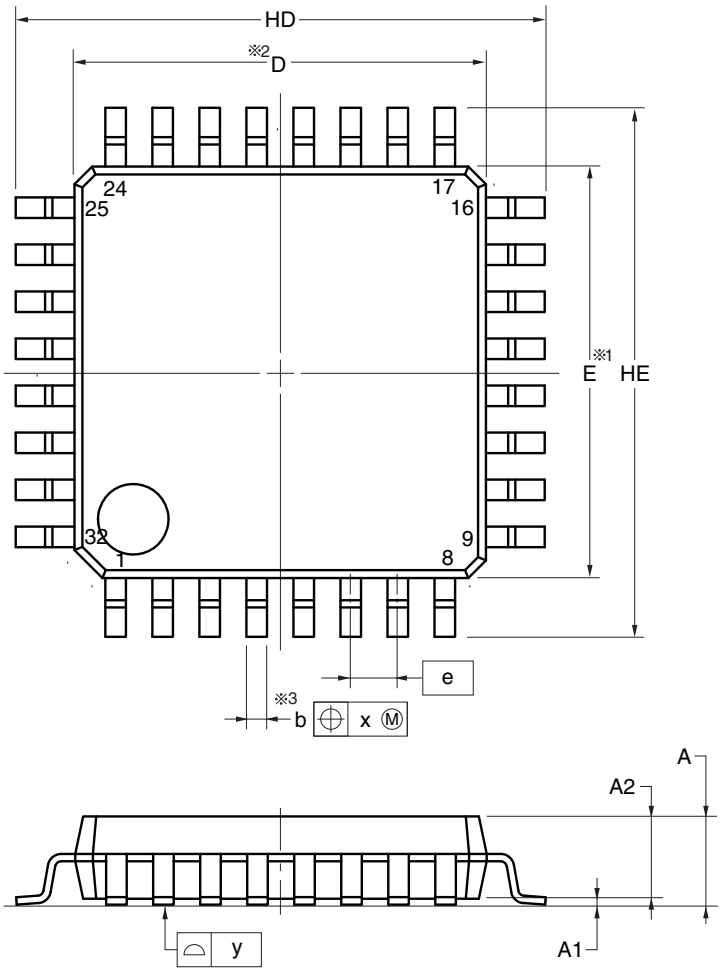
38.2 32-pin Products

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN032-5x5-0.50	PWQN0032KE-A	0.06

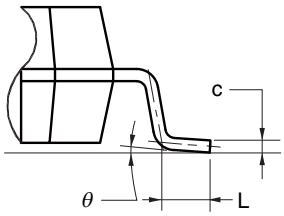


Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A_1	0.00	0.02	0.05
A_3		0.203 REF.	
b	0.18	0.25	0.30
D		5.00 BSC	
E		5.00 BSC	
e		0.50 BSC	
L	0.35	0.40	0.45
K	0.20	—	—
D_2	3.15	3.20	3.25
E_2	3.15	3.20	3.25
aaa		0.15	
bbb		0.10	
ccc		0.10	
ddd		0.05	
eee		0.08	
fff		0.10	

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LQFP32-7x7-0.80	PLQP0032GB-A	P32GA-80-GBT-1	0.2



detail of lead end



(UNIT:mm)

ITEM	DIMENSIONS
D	7.00±0.10
E	7.00±0.10
HD	9.00±0.20
HE	9.00±0.20
A	1.70 MAX.
A1	0.10±0.10
A2	1.40
b	0.37±0.05
c	0.145±0.055
L	0.50±0.20
θ	0° to 8°
e	0.80
x	0.20
y	0.10

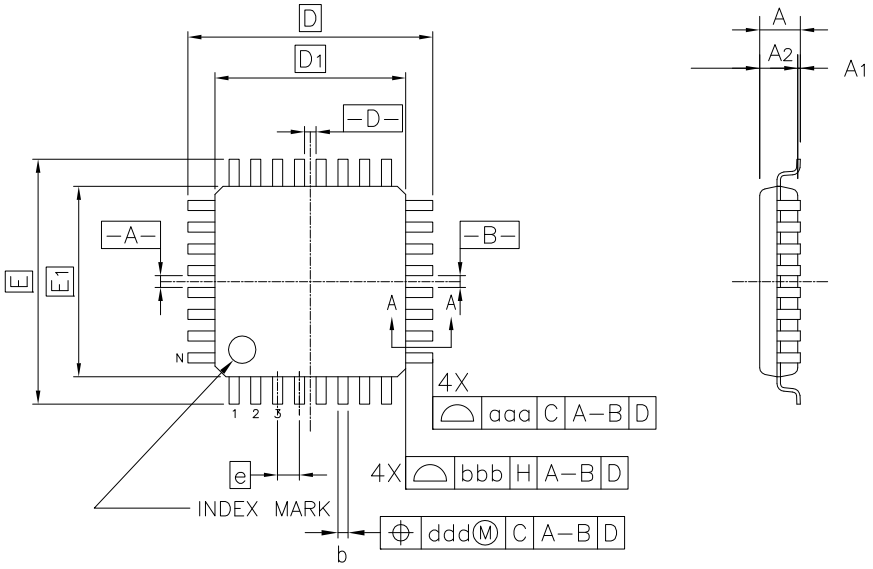
NOTE

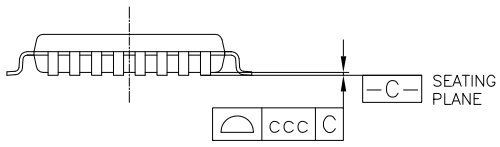
1.Dimensions "※1" and "※2" do not include mold flash.

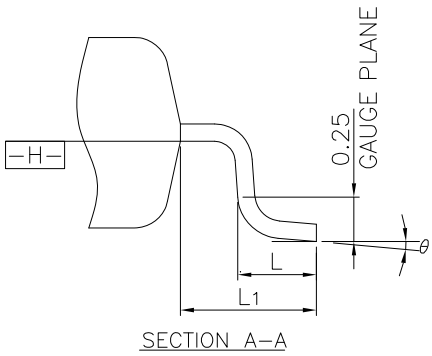
2.Dimension "※3" does not include trim offset.

© 2012 Renesas Electronics Corporation. All rights reserved.

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-LQFP32-7x7-0.80	PLQP0032GE-A	0.18



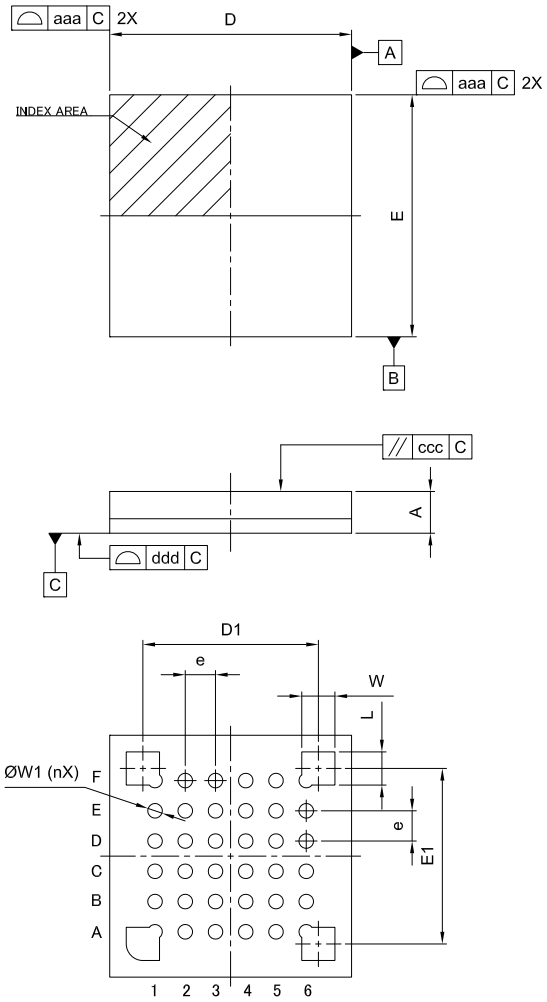




Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	–	–	1.60
A_1	0.05	–	0.15
A_2	1.35	1.40	1.45
D	–	9.00	–
D_1	–	7.00	–
E	–	9.00	–
E_1	–	7.00	–
N	–	32	–
e	–	0.80	–
b	0.30	0.37	0.45
c	0.09	–	0.20
θ	0°	3.5°	7°
L	0.45	0.60	0.75
L_1	–	1.00	–
aaa	–	–	0.20
bbb	–	–	0.20
ccc	–	–	0.10
ddd	–	–	0.20

38.3 36-pin Products

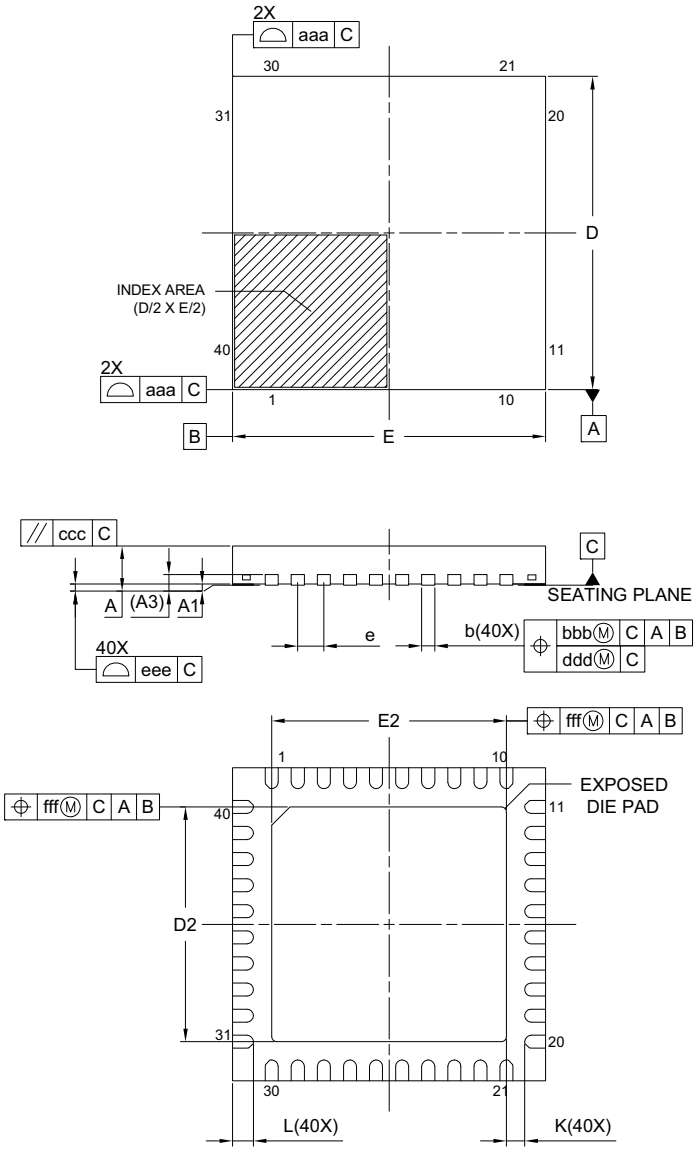
JEITA Package Code	RENESAS Code	MASS(Typ.)[g]
P-WFLGA36-4×4-0.50	PWLG0036KB-A	0.02



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	—	4.00	—
E	—	4.00	—
D1		2.90 BSC	
E1		2.90 BSC	
A	—	—	0.76
W1	0.19	0.24	0.29
W	—	0.55	—
L	—	0.55	—
e		0.50 BSC	
aaa		0.10	
ccc		0.20	
ddd		0.08	
n	—	36	—

38.4 40-pin Products

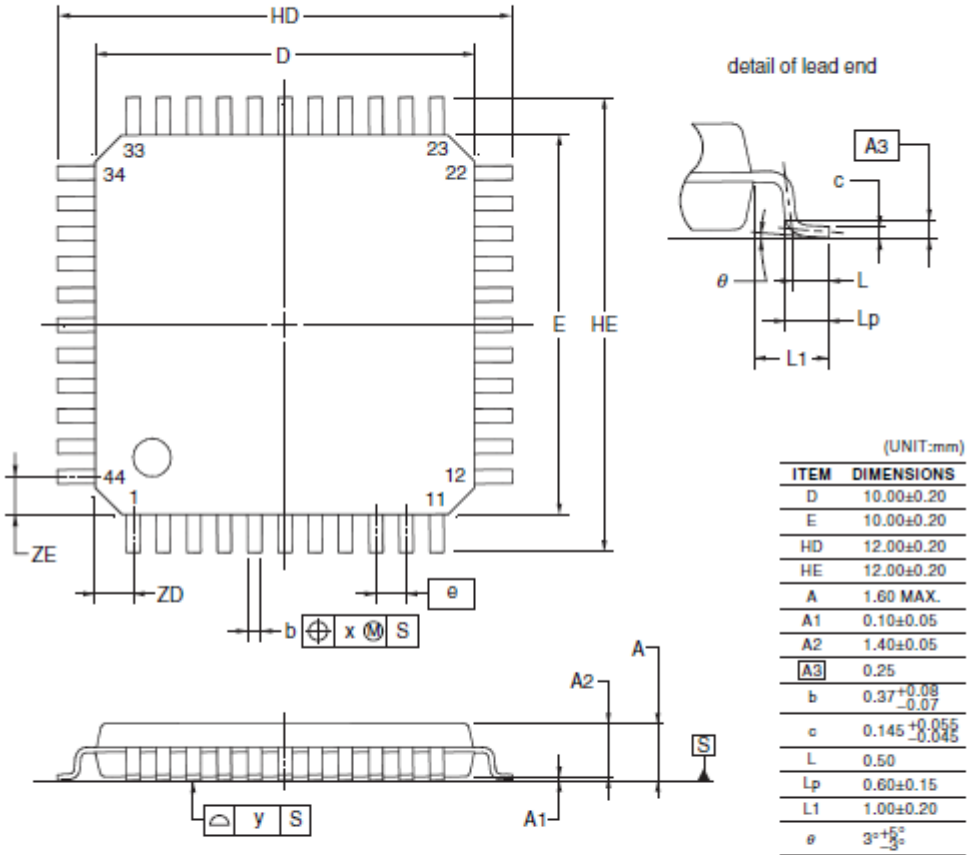
JEITA Package code	RENESAS code	MASS (TYP.)[g]
P-HWQFN040-6x6-0.50	PWQN0040KD-A	0.08



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A_1	0.00	0.02	0.05
A_3		0.203 REF.	
b	0.18	0.25	0.30
D		6.00 BSC	
E		6.00 BSC	
e		0.50 BSC	
L	0.30	0.40	0.50
K	0.20	—	—
D_2	4.45	4.50	4.55
E_2	4.45	4.50	4.55
aaa		0.15	
bbb		0.10	
ccc		0.10	
ddd		0.05	
eee		0.08	
fff		0.10	

38.5 44-pin Products

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LQFP44-10x10-0.80	PLQP0044GC-A	P44GB-80-UES-2	0.36



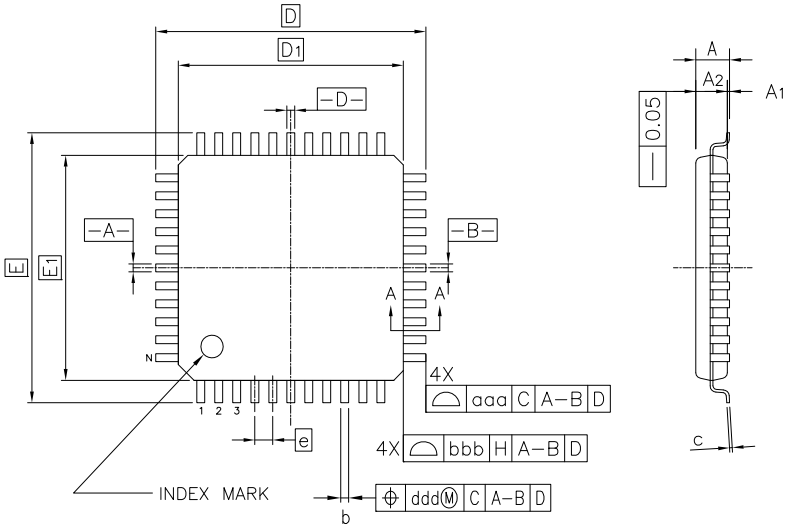
(UNIT:mm)

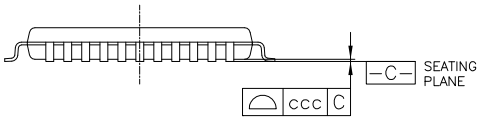
ITEM	DIMENSIONS
D	10.00±0.20
E	10.00±0.20
HD	12.00±0.20
HE	12.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	$0.37^{+0.08}_{-0.07}$
c	$0.145^{+0.055}_{-0.045}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ+5^{\circ}}_{-3^{\circ}}$
e	0.80
x	0.20
y	0.10
ZD	1.00
ZE	1.00

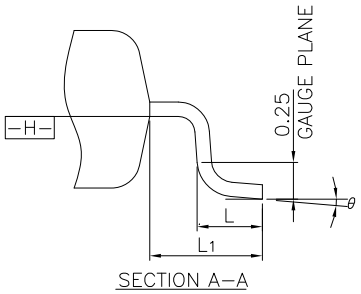
NOTE
Each lead centerline is located within 0.20 mm of its true position at maximum material condition.

©2012 Renesas Electronics Corporation. All rights reserved.

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-LQFP044-10x10-0.80	PLQP0044GE-A	0.34



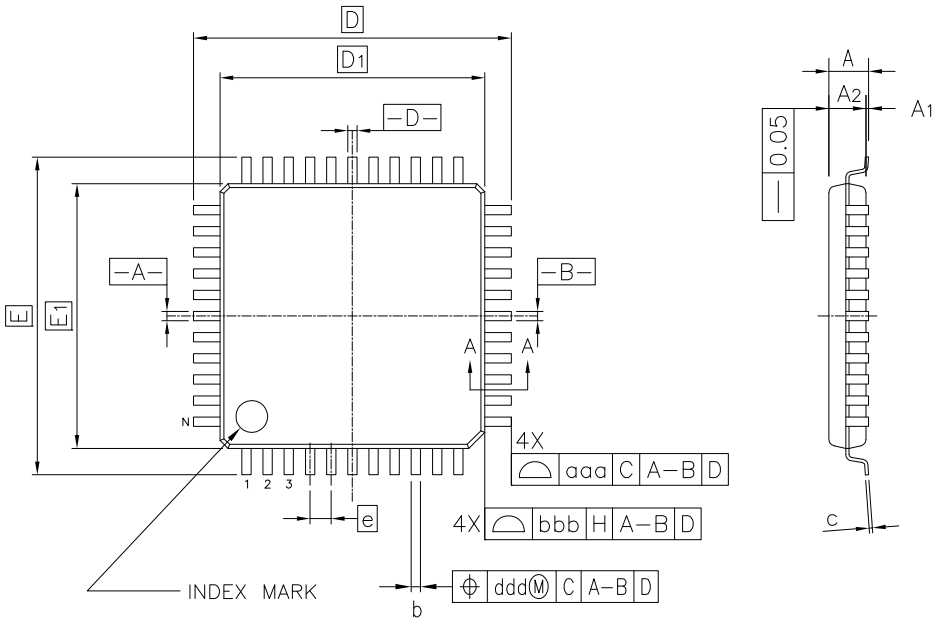


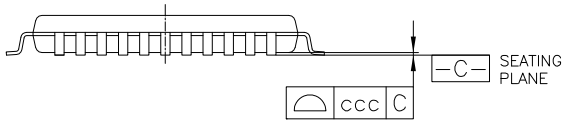


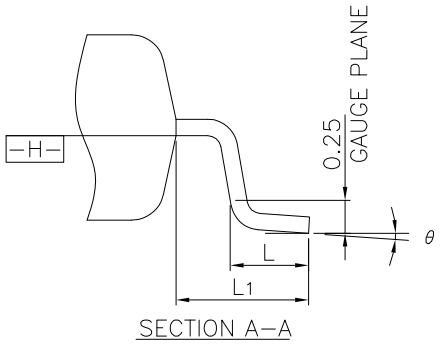
Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	–	–	1.60
A_1	0.05	–	0.15
A_2	1.35	1.40	1.45
D	–	12.00	–
D_1	–	10.00	–
E	–	12.00	–
E_1	–	10.00	–
N	–	44	–
e	–	0.80	–
b	0.30	0.37	0.45
c	0.09	–	0.20
θ	0°	3.5°	7°
L	0.45	0.60	0.75
L_1	–	1.00	–
aaa	–	–	0.20
bbb	–	–	0.20
ccc	–	–	0.10
ddd	–	–	0.20

<R>

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-LQFP44-10x10-0.80	PLQP0044GF-A	0.3



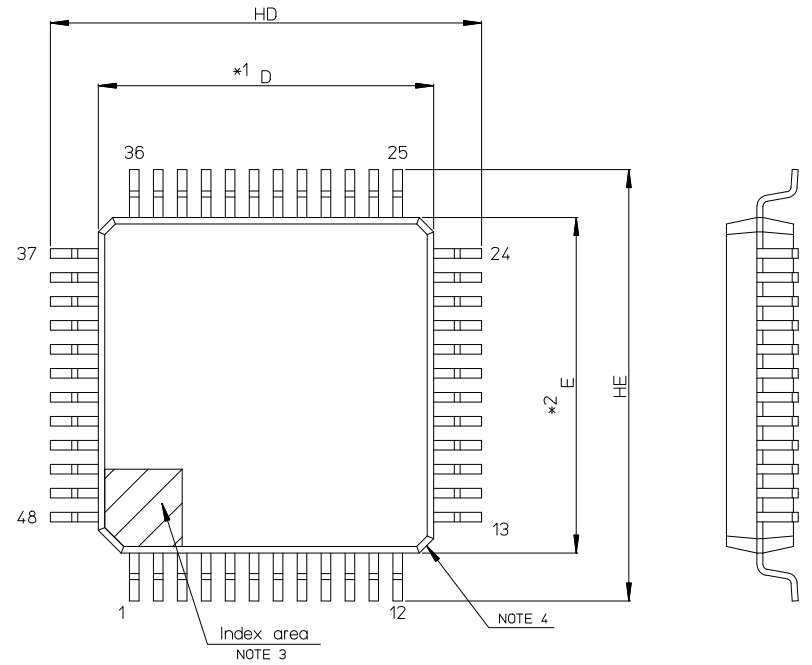


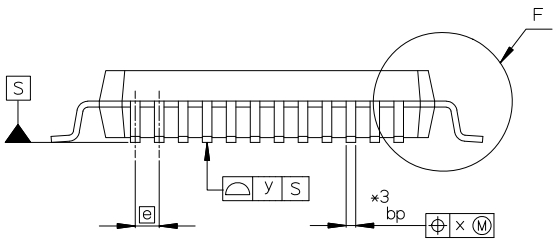


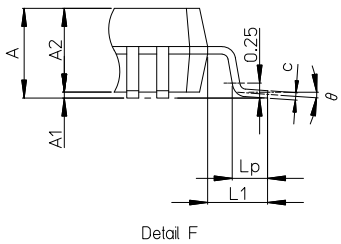
Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	–	–	1.70
A_1	0.05	–	0.15
A_2	1.35	1.40	1.45
D	12.00 BSC.		
D_1	10.00 BSC.		
E	12.00 BSC.		
E_1	10.00 BSC.		
N	–	44	–
e	0.80 BSC.		
b	0.30	0.37	0.45
c	0.09	–	0.20
θ	0°	3.5°	8°
L	0.45	0.60	0.75
L_1	1.00 REF.		
aaa	–	–	0.20
bbb	–	–	0.20
ccc	–	–	0.10
ddd	–	–	0.20

38.6 48-pin Products

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP48-7x7-0.50	PLQP0048KB-B	——	0.2g





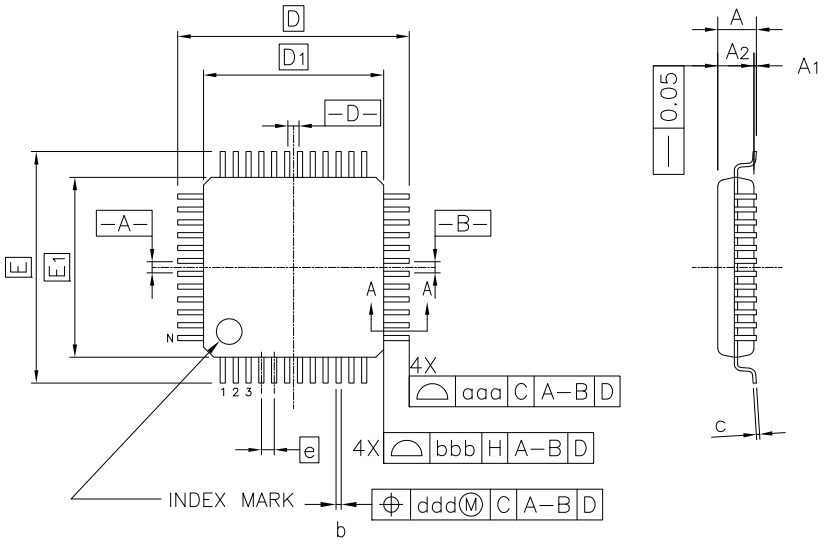


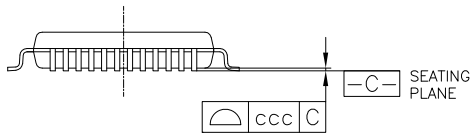
Detail F

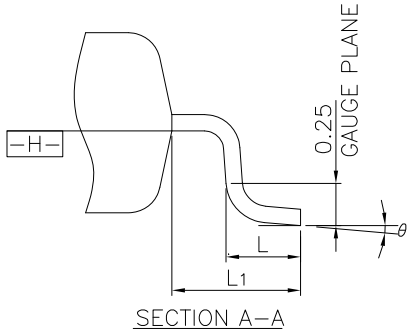
NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL; SIZE MAY VARY.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	6.9	7.0	7.1
E	6.9	7.0	7.1
A2	——	1.4	——
HD	8.8	9.0	9.2
HE	8.8	9.0	9.2
A	——	——	1.7
A1	0.05	——	0.15
bp	0.17	0.20	0.27
c	0.09	——	0.20
θ	0°	3.5°	8°
e	——	0.5	——
x	——	——	0.08
y	——	——	0.08
Lp	0.45	0.6	0.75
L1	——	1.0	——

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-LFQFP48-7x7-0.50	PLQP0048KL-A	0.18

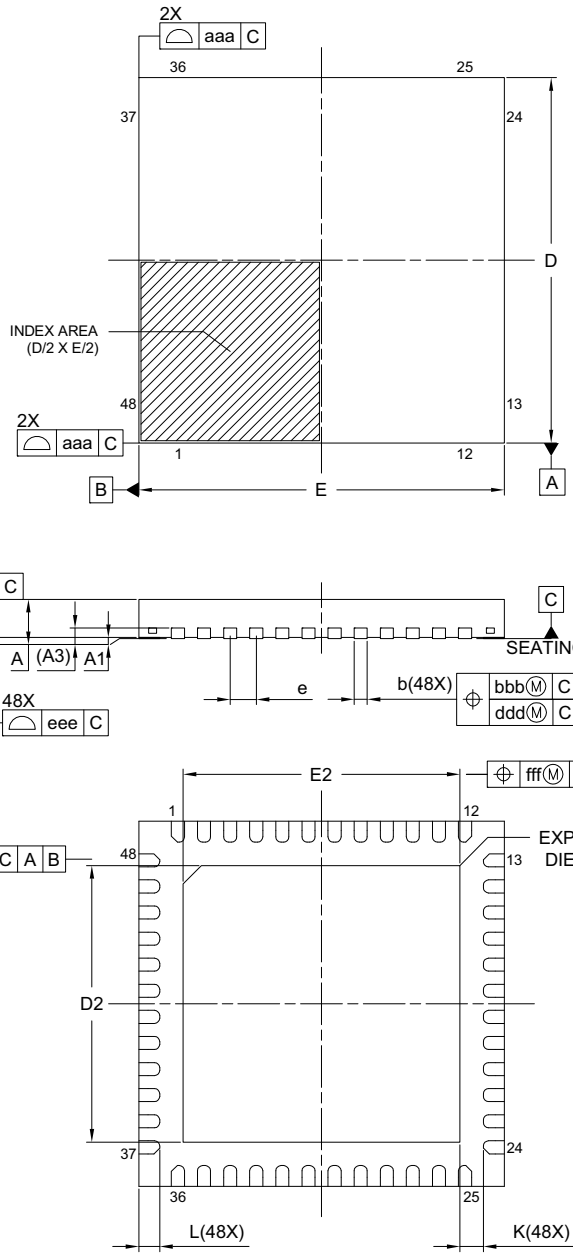






Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	–	–	1.60
A_1	0.05	–	0.15
A_2	1.35	1.40	1.45
D	–	9.00	–
D_1	–	7.00	–
E	–	9.00	–
E_1	–	7.00	–
N	–	48	–
e	–	0.50	–
b	0.17	0.22	0.27
c	0.09	–	0.20
θ	0°	3.5°	7°
L	0.45	0.60	0.75
L_1	–	1.00	–
aaa	–	–	0.20
bbb	–	–	0.20
ccc	–	–	0.08
ddd	–	–	0.08

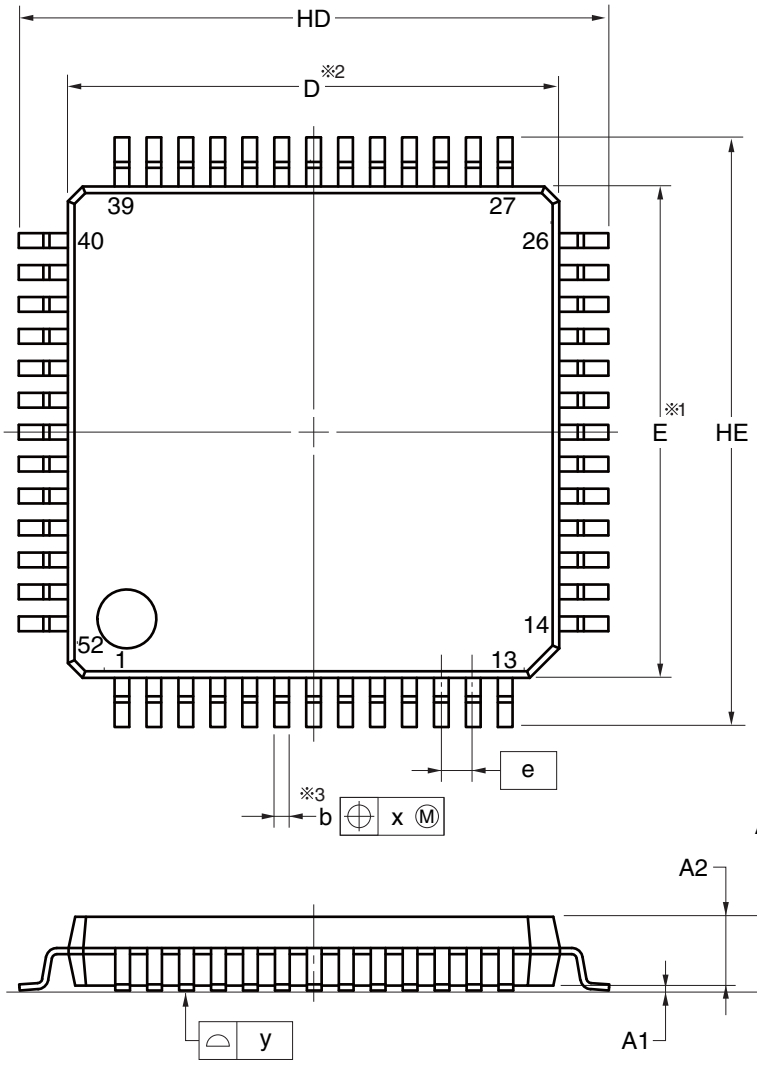
JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN048-7x7-0.50	PWQN0048KC-A	0.13 g



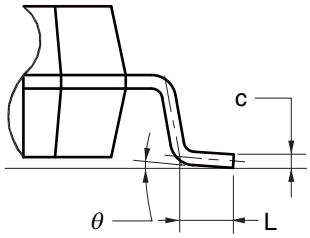
Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A_1	0.00	0.02	0.05
A_3		0.203 REF.	
b	0.20	0.25	0.30
D		7.00 BSC	
E		7.00 BSC	
e		0.50 BSC	
L	0.30	0.40	0.50
K	0.20	—	—
D_2	5.25	5.30	5.35
E_2	5.25	5.30	5.35
aaa		0.15	
bbb		0.10	
ccc		0.10	
ddd		0.05	
eee		0.08	
fff		0.10	

38.7 52-pin Products

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LQFP52-10x10-0.65	PLQP0052JA-A	P52GB-65-GBS-1	0.3



detail of lead end



(UNIT:mm)

ITEM	DIMENSIONS
D	10.00±0.10
E	10.00±0.10
HD	12.00±0.20
HE	12.00±0.20
A	1.70 MAX.
A1	0.10±0.05
A2	1.40
b	0.32±0.05
c	0.145±0.055
L	0.50±0.15
θ	0° to 8°
e	0.65
x	0.13
y	0.10

NOTE

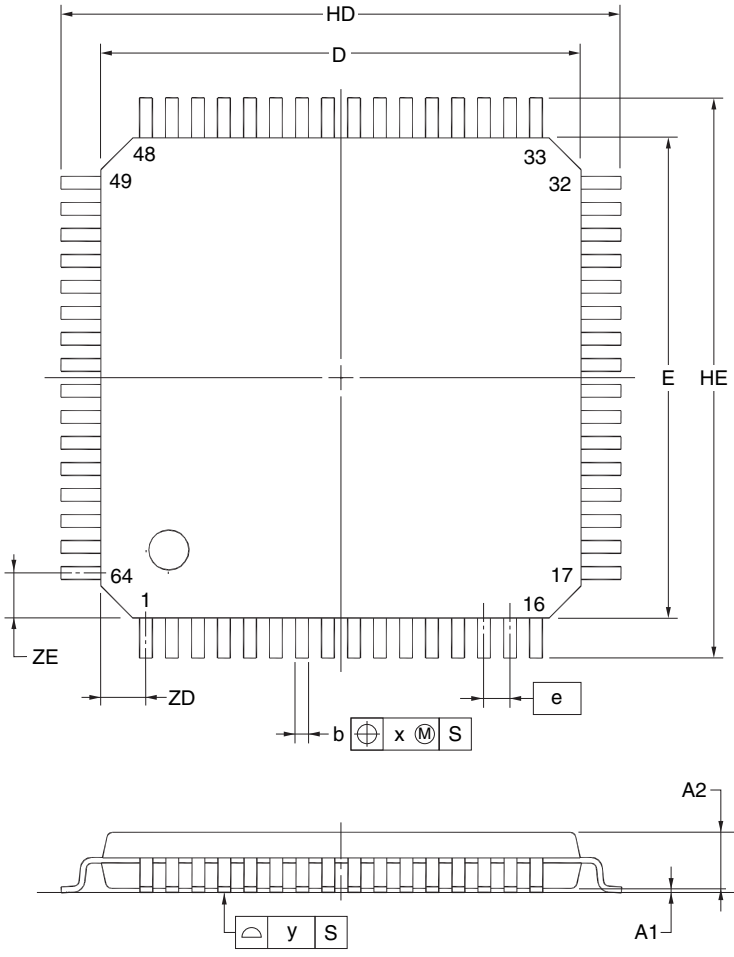
1.Dimensions "※1" and "※2" do not include mold flash.

2.Dimension "※3" does not include trim offset.

© 2012 Renesas Electronics Corporation. All rights reserved.

38.8 64-pin Products

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LQFP64-12x12-0.65	PLQP0064JA-A	P64GK-65-UET-2	0.51



(UNIT:mm)

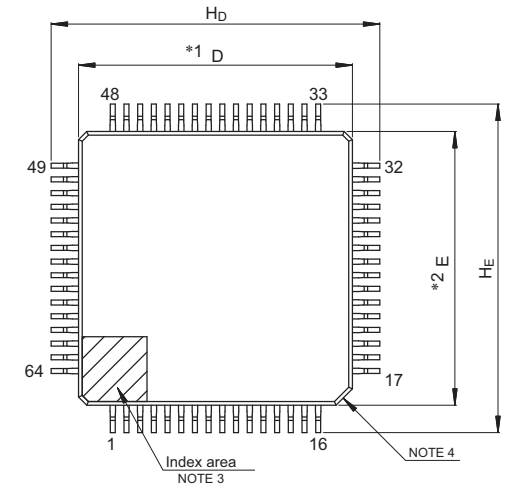
ITEM	DIMENSIONS
D	12.00±0.20
E	12.00±0.20
HD	14.00±0.20
HE	14.00±0.20
A	1.60 MAX.
A1	0.10±0.05
A2	1.40±0.05
A3	0.25
b	$0.32^{+0.08}_{-0.07}$
c	$0.145^{+0.055}_{-0.045}$
L	0.50
Lp	0.60±0.15
L1	1.00±0.20
θ	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
e	0.65
x	0.13
y	0.10
ZD	1.125
ZE	1.125

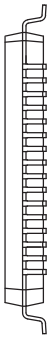
NOTE
Each lead centerline is located within 0.13 mm of its true position at maximum material condition.

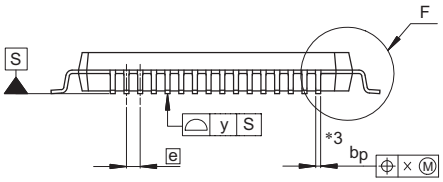
©2012 Renesas Electronics Corporation. All rights reserved.

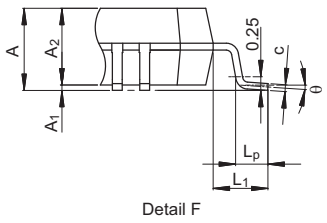
JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP64-10x10-0.50	PLQP0064KB-C	—	0.3

Unit: mm









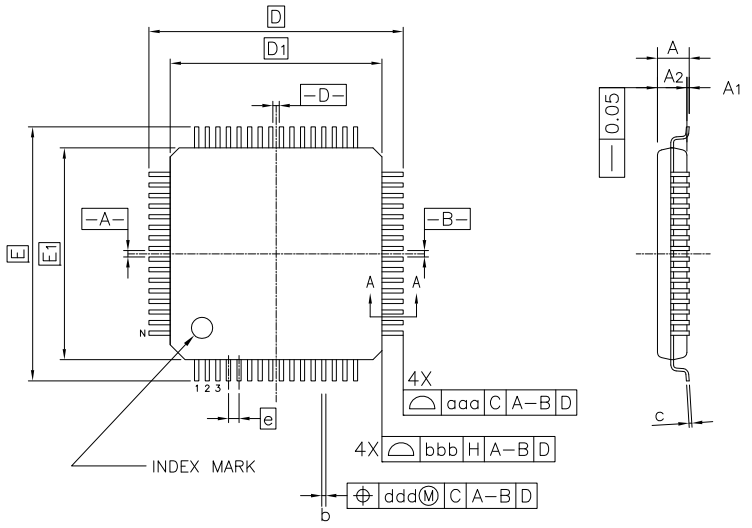
Detail F

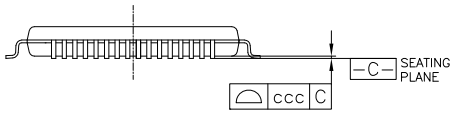
NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

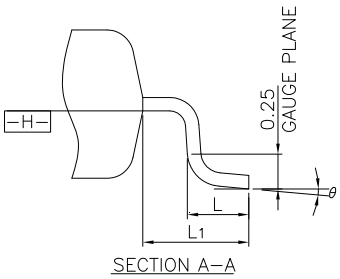
Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	9.9	10.0	10.1
E	9.9	10.0	10.1
A_2	—	1.4	—
H_D	11.8	12.0	12.2
H_E	11.8	12.0	12.2
A	—	—	1.7
A_1	0.05	—	0.15
b_p	0.15	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
L_p	0.45	0.6	0.75
L_1	—	1.0	—

© 2015 Renesas Electronics Corporation. All rights reserved.

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-LFQFP064-10x10-0.50	PLQP0064KL-A	0.36

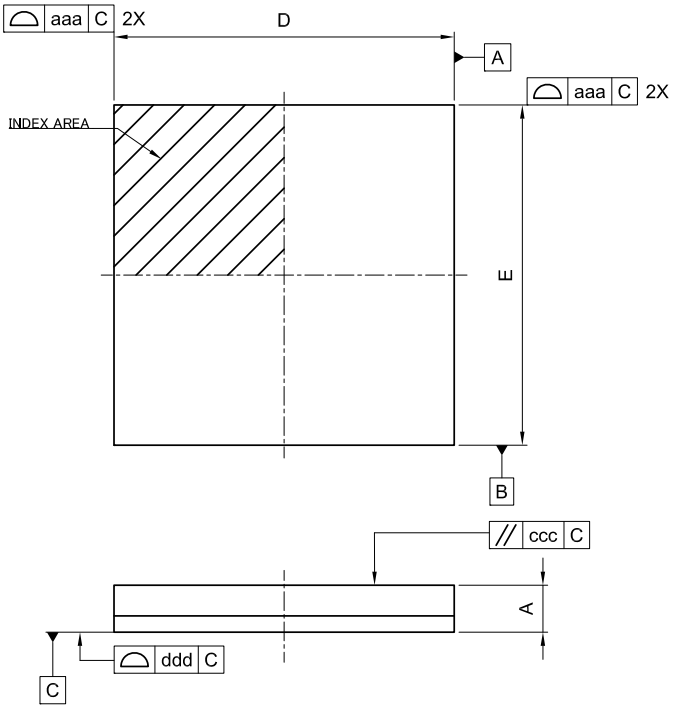


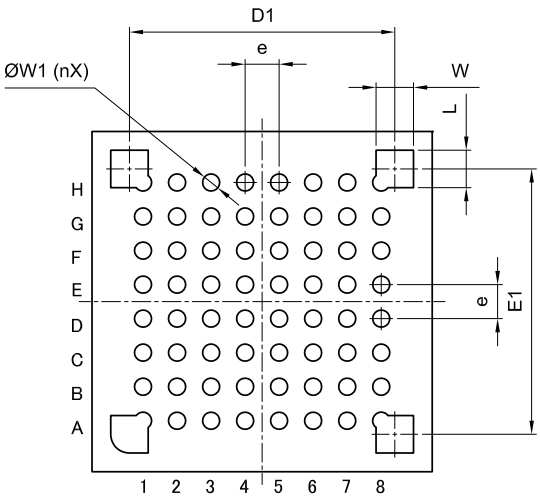




Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	–	–	1.60
A_1	0.05	–	0.15
A_2	1.35	1.40	1.45
D	–	12.00	–
D_1	–	10.00	–
E	–	12.00	–
E_1	–	10.00	–
N	–	64	–
e	–	0.50	–
b	0.17	0.22	0.27
c	0.09	–	0.20
θ	0°	3.5°	7°
L	0.45	0.60	0.75
L_1	–	1.00	–
aaa	–	–	0.20
bbb	–	–	0.20
ccc	–	–	0.08
ddd	–	–	0.08

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-WFLGA64-5x5-0.50	PWLG0064KB-A	0.035



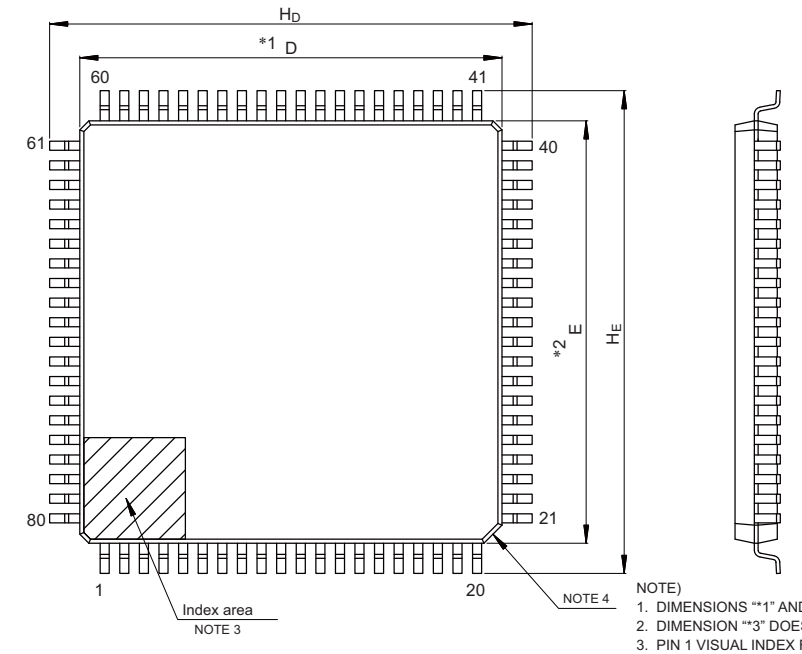


Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	—	5.00	—
E	—	5.00	—
D1		3.90 BSC	
E1		3.90 BSC	
A	—	—	0.76
W1	0.21	0.25	0.29
W	—	0.55	—
L	—	0.55	—
e		0.50 BSC	
aaa	—	—	0.10
ccc	—	—	0.20
ddd	—	—	0.08
n	—	64	—

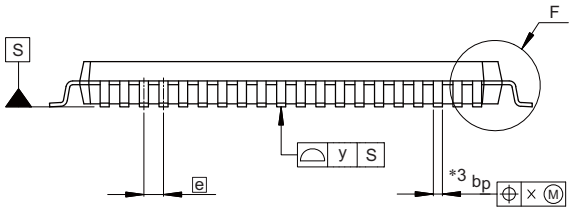
38.9 80-pin Products

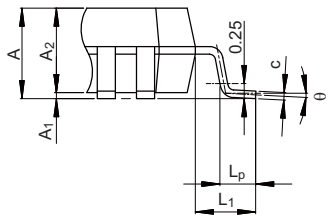
JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LQFP80-14x14-0.65	PLQP0080JA-B	—	0.6

Unit: mm



NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.





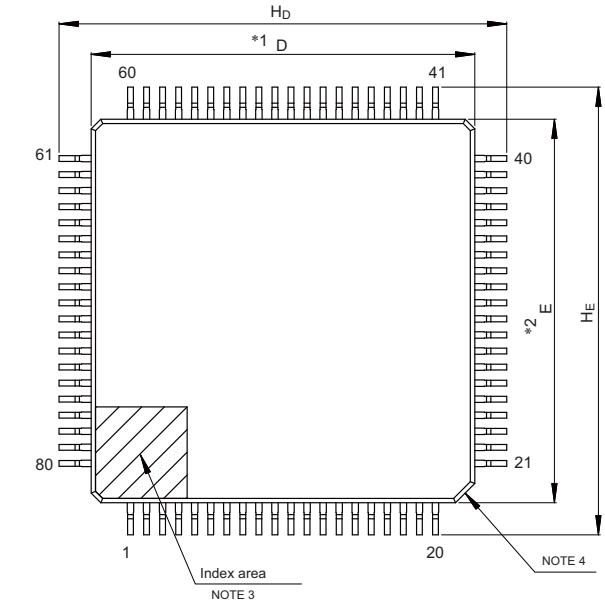
Detail F

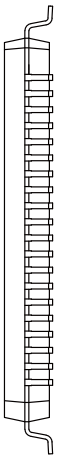
Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	13.9	14.0	14.1
E	13.9	14.0	14.1
A_2	—	1.4	—
H_D	15.8	16.0	16.2
H_E	15.8	16.0	16.2
A	—	—	1.7
A_1	0.05	—	0.15
b_p	0.22	0.30	0.38
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.65	—
x	—	—	0.13
y	—	—	0.10
L_p	0.45	0.6	0.75
L_1	—	1.0	—

© 2016 Renesas Electronics Corporation. All rights reserved.

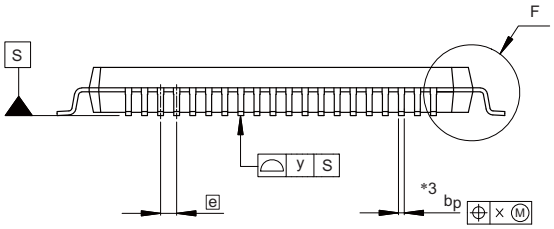
JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP80-12x12-0.50	PLQP0080KB-B	—	0.5

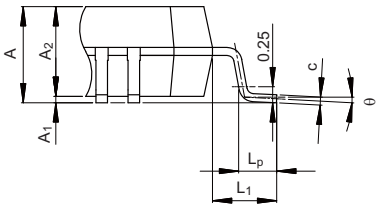
Unit: mm





NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.



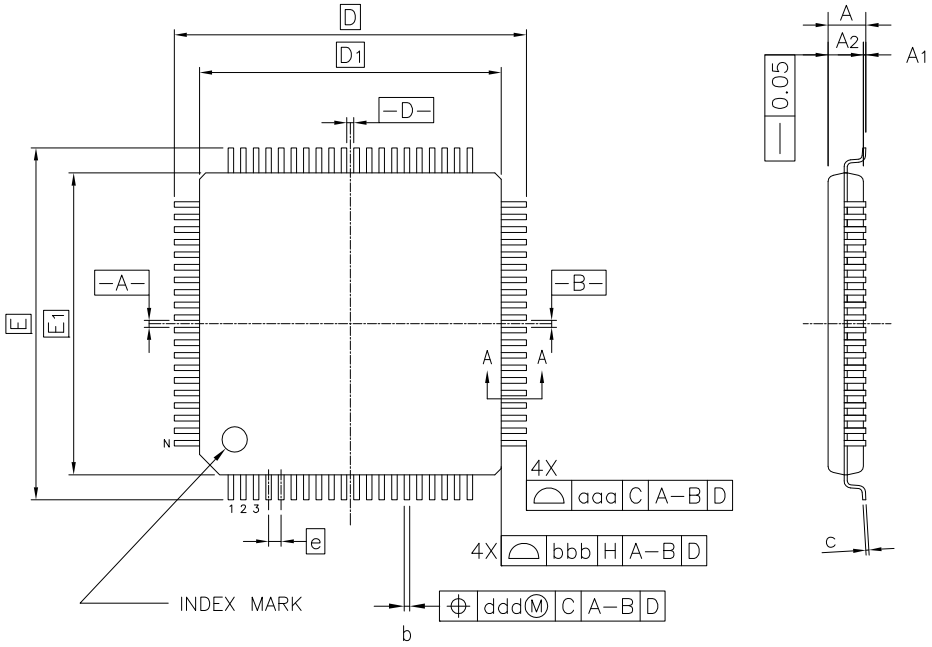


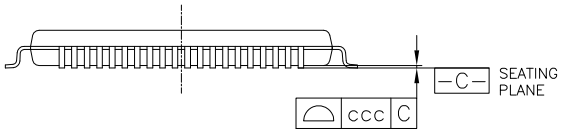
Detail F

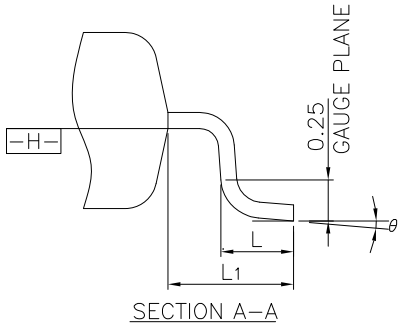
Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	11.9	12.0	12.1
E	11.9	12.0	12.1
A_2	—	1.4	—
H_D	13.8	14.0	14.2
H_E	13.8	14.0	14.2
A	—	—	1.7
A_1	0.05	—	0.15
b_p	0.15	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
L_p	0.45	0.6	0.75
L_1	—	1.0	—

© 2017 Renesas Electronics Corporation. All rights reserved.

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-LFQFP80-12x12-0.50	PLQP0080KJ-A	0.49





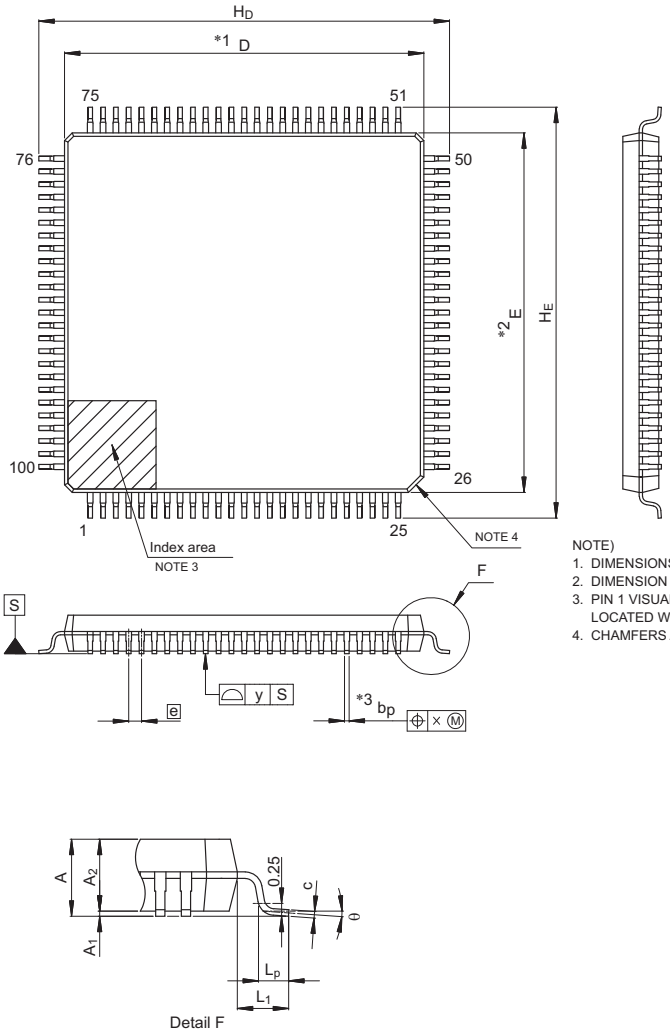


Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	–	–	1.60
A_1	0.05	–	0.15
A_2	1.35	1.40	1.45
D	–	14.00	–
D_1	–	12.00	–
E	–	14.00	–
E_1	–	12.00	–
N	–	80	–
e	–	0.50	–
b	0.17	0.22	0.27
c	0.09	–	0.20
θ	0°	3.5°	7°
L	0.45	0.60	0.75
L_1	–	1.00	–
aaa	–	–	0.20
bbb	–	–	0.20
ccc	–	–	0.08
ddd	–	–	0.08

38.10 100-pin Products

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-LFQFP100-14x14-0.50	PLQP0100KB-B	—	0.6

Unit: mm

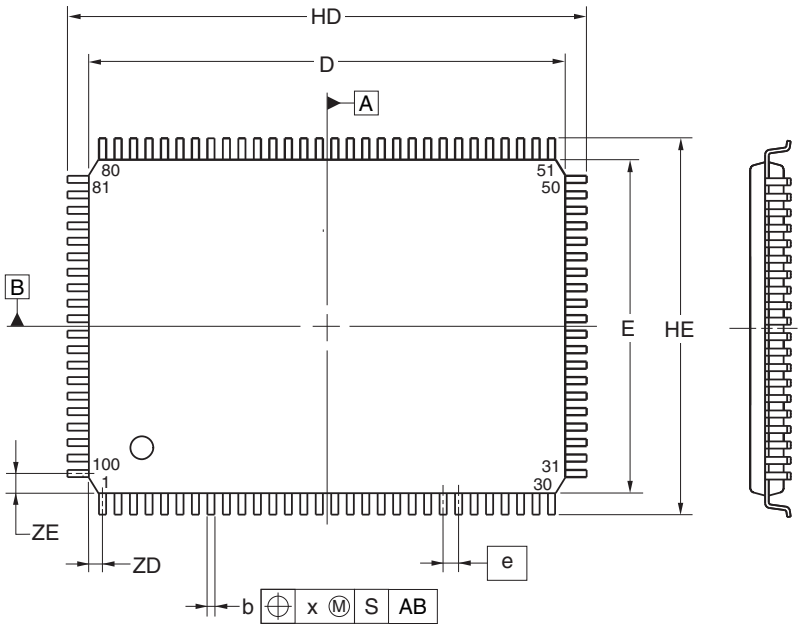


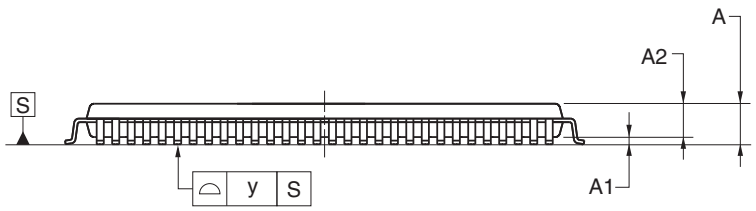
NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.

Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	13.9	14.0	14.1
E	13.9	14.0	14.1
A_2	—	1.4	—
H_D	15.8	16.0	16.2
H_E	15.8	16.0	16.2
A	—	—	1.7
A_1	0.05	—	0.15
b_p	0.15	0.20	0.27
c	0.09	—	0.20
θ	0°	3.5°	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.08
L_p	0.45	0.6	0.75
L_1	—	1.0	—

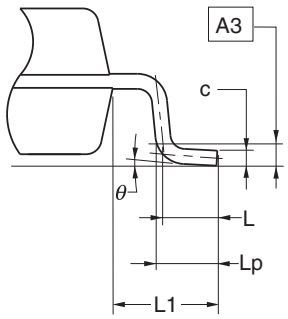
© 2015 Renesas Electronics Corporation. All rights reserved.

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LQFP100-14x20-0.65	PLQP0100JC-A	P100GF-65-GBN-1	0.92





detail of lead end



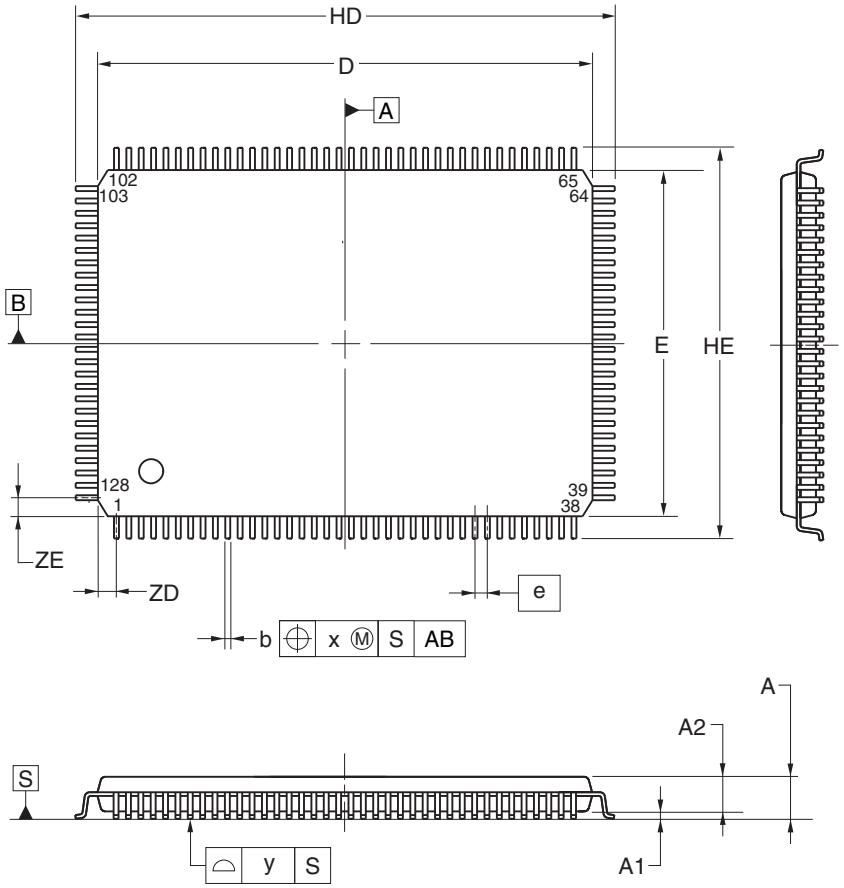
(UNIT:mm)

ITEM	DIMENSIONS
D	20.00±0.20
E	14.00±0.20
HD	22.00±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	$0.32^{+0.08}_{-0.07}$
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	0.65
x	0.13
y	0.10
ZD	0.575
ZE	0.825

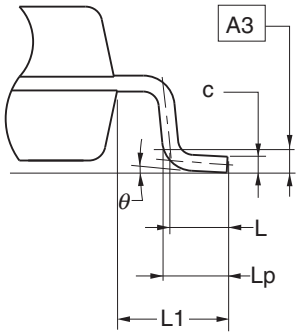
©2012 Renesas Electronics Corporation. All rights reserved.

38.11 128-pin Products

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LFQFP128-14x20-0.50	PLQP0128KD-A	P128GF-50-GBP-1	0.92



detail of lead end



(UNIT:mm)

ITEM	DIMENSIONS
D	20.00±0.20
E	14.00±0.20
HD	22.00±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	0.75
ZE	0.75

©2012 Renesas Electronics Corporation. All rights reserved.

Appendix A Revision History

A.1 Major Revisions in This Edition

(1/3)

Page	Description	Classification
Overall		
—	The specifications of the R7F104G products of the RL78/G23 Group were added.	(c)
Section 1 Outline		
p.1	1.1 Differences between R7F100G and R7F104G Products was added.	(c)
p.4	1.2 Features was modified.	(a)
Section 3 CPU Architecture		
p.189	3.1.6 Data memory addressing: The description was modified.	(c)
Section 7 Timer Array Unit (TAU)		
p.494	Figure 7 - 74 Procedure for Operations When the One-shot Pulse Output Function Is to be Used (2/2) was modified.	(a)
Section 8 Realtime Clock (RTC)		
p.539	8.4.6 Example of time error correction by the realtime clock: Correction example 1 was modified.	(c)
Section 12 A/D Converter (ADC)		
p.599	Figure 12 - 5 Timing of Operations When the A/D Voltage Comparator Is Used: Note 2 was modified.	(c)
p.600 to p.612	Table 12 - 3 A/D Conversion Time Selection: The conditions for selection of conversion times were modified.	(c)
p.603, p.605, p.610, p.612	Table 12 - 3 A/D Conversion Time Selection: Note 1 and Caution 1 were modified.	(c)
p.617	Figure 12 - 8 Format of A/D Converter Mode Register 2 (ADM2) was modified.	(a)
p.618	Figure 12 - 9 Range of Interrupt Signals Generated under Control by the ADRCK Bit: Remark was modified.	(a)
p.627	12.4 A/D Converter Operations: The description was modified.	(a)
p.632	12.6.3 Software trigger no-wait mode: Scan mode and sequential conversion mode: The description was modified.	(a)
p.633	12.6.4 Software trigger no-wait mode: Scan mode and one-shot conversion mode: The description was modified.	(a)
p.636	12.6.7 Software trigger wait mode: Scan mode and sequential conversion mode: The description was modified.	(a)
p.637	12.6.8 Software trigger wait mode: Scan mode and one-shot conversion mode: The description was modified.	(a)
p.640	12.6.11 Hardware trigger no-wait mode: Scan mode and sequential conversion mode: The description was modified.	(a)
p.641	12.6.12 Hardware trigger no-wait mode: Scan mode and one-shot conversion mode: The description was modified.	(a)
p.645	12.6.15 Hardware trigger wait mode: Scan mode and sequential conversion mode: The description was modified.	(a)

Remark “Classification” in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note,
(d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

(2/3)

Page	Description	Classification
p.646	12.6.16 Hardware trigger wait mode: Scan mode and one-shot conversion mode: The description was modified.	(a)
p.647	Figure 12 - 34 Settings in Software Trigger No-wait Mode and Note were modified.	(a)
p.648	Figure 12 - 35 Settings in Software Trigger Wait Mode and Note were modified.	(a)
p.649	Figure 12 - 36 Settings in Hardware Trigger No-wait Mode and Note were modified.	(a)
p.650	Figure 12 - 37 Settings in Hardware Trigger Wait Mode and Note were modified.	(a)
p.651	Figure 12 - 38 Settings When Selecting the Temperature Sensor Output Voltage, Internal Reference Voltage, or TSCAP Voltage of the CTSU and Note were modified.	(a)
p.652	Figure 12 - 39 Settings in Test Mode and Note were modified.	(a)
p.654	Figure 12 - 41 Flow of Setting up the SNOOZE Mode for Use with Software-triggered Conversion was modified.	(a)
p.655, p.656	12.8.2 A/D conversion by inputting a hardware trigger: Note and the description were modified.	(a)
p.658	Figure 12 - 45 Flow of Setting up the SNOOZE Mode for Use with Hardware-triggered Conversion and Note 1 were modified.	(a)
p.663	12.10 Points for Caution When the A/D Converter is to be Used: The description was modified.	(c)
p.664, p.665	12.10 Points for Caution When the A/D Converter is to be Used: The descriptions were modified.	(a)
Section 13 D/A Converter (DAC)		
p.669	Figure 13 - 2 Format of Peripheral Enable Register 1 (PER1) was modified.	(a)
Section 15 Serial Array Unit (SAU)		
p.861	15.8.5 Calculating transfer rate: Caution was modified.	(a)
Section 16 Serial Interface IICA (IICA)		
p.892	16.5.1 Start conditions: The description was modified.	(a)
p.903	Table 16 - 4 State during Arbitration and Interrupt Request Generation Timing: Note 1 was modified.	(c)
p.951	Figure 16 - 33 Example of Slave-to-Master Communications with Clock Stretching in the 8th-Cycle for the Master and 9th-Cycle for the Slave (2/3) was modified.	(a)
p.953	Figure 16 - 33 Example of Slave-to-Master Communications with Shifted Clock Stretching (8th- to 9th-Cycle) for the Master and 9th-Cycle for the Slave (3/3) was modified.	(a)
Section 23 Standby Function		
p.1195	Table 23 - 3 Operating States in STOP Mode: Note was added.	(c)
Section 25 Power-on-reset Circuit (POR)		
p.1217	Figure 25 - 2 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (1/2): Note 4 was modified.	(a)
p.1218	Figure 25 - 2 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (2/2): Note 2 was modified.	(a)
Section 29 SNOOZE Mode Sequencer (SMS)		
p.1288	29.5 Commands for Use in Processing by the Sequencer: The description was modified.	(a)
Section 30 Capacitive Sensing Unit (CTSU2L)		
p.1343, p.1345	Figure 30 - 16 Format of CTSU Calibration Registers L and H (CTSUDBGR0, CTSUDBGR1) was modified.	(a)
Section 33 Flash Memory		
p.1404	Figure 33 - 17 Format of Flash Memory Sequencer Control Register (FSSQ): Notes 2 and 3 were modified.	(a)

Remark “Classification” in the above table classifies revisions as follows.

(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note,

(d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

(3/3)

Page	Description	Classification
Section 37 Electrical Characteristics		
p.1487	37.3.2 Characteristics of the supply current, 1. 30- to 64-pin R7F100G and 32-pin R7F104G products with 96- to 128-Kbyte code flash memory: Notes 2 to 4 were modified.	(a)
p.1489	37.3.2 Characteristics of the supply current, 1. 30- to 64-pin R7F100G and 32-pin R7F104G products with 96- to 128-Kbyte code flash memory: Note 2 and Remark 2 were modified.	(a)
p.1490	37.3.2 Characteristics of the supply current, 1. 30- to 64-pin R7F100G and 32-pin R7F104G products with 96- to 128-Kbyte code flash memory: Notes 3 to 5 were modified.	(a)
p.1493	37.3.2 Characteristics of the supply current, 1. 30- to 64-pin R7F100G and 32-pin R7F104G products with 96- to 128-Kbyte code flash memory: Notes 3, 6, and 7 and Remark 2 were modified.	(a)
p.1494	37.3.2 Characteristics of the supply current, 2. 30- to 64-pin R7F100G and 44- and 48-pin R7F104G products with 192- to 256-Kbyte code flash memory and 80-pin R7F100G products with 128- to 256-Kbyte code flash memory: Notes 2 to 4 were modified.	(a)
p.1496	37.3.2 Characteristics of the supply current, 2. 30- to 64-pin R7F100G and 44- and 48-pin R7F104G products with 192- to 256-Kbyte code flash memory and 80-pin R7F100G products with 128- to 256-Kbyte code flash memory: Note 2 and Remark 2 were modified.	(a)
p.1497	37.3.2 Characteristics of the supply current, 2. 30- to 64-pin R7F100G and 44- and 48-pin R7F104G products with 192- to 256-Kbyte code flash memory and 80-pin R7F100G products with 128- to 256-Kbyte code flash memory: Notes 3 to 5 were modified.	(a)
p.1500	37.3.2 Characteristics of the supply current, 2. 30- to 64-pin R7F100G and 44- and 48-pin R7F104G products with 192- to 256-Kbyte code flash memory and 80-pin R7F100G products with 128- to 256-Kbyte code flash memory: Notes 3, 6, and 7 and Remark 2 were modified.	(a)
p.1501	37.3.2 Characteristics of the supply current, 3. 44- to 80-pin R7F100G products with 384- to 768-Kbyte code flash memory and 100- and 128-pin R7F100G products: Notes 2 to 4 were modified.	(a)
p.1503	37.3.2 Characteristics of the supply current, 3. 44- to 80-pin R7F100G products with 384- to 768-Kbyte code flash memory and 100- and 128-pin R7F100G products: Note 2 and Remark 2 were modified.	(a)
p.1504	37.3.2 Characteristics of the supply current, 3. 44- to 80-pin R7F100G products with 384- to 768-Kbyte code flash memory and 100- and 128-pin R7F100G products: Notes 3 to 5 were modified.	(a)
p.1507	37.3.2 Characteristics of the supply current, 3. 44- to 80-pin R7F100G products with 384- to 768-Kbyte code flash memory and 100- and 128-pin R7F100G products: Notes 3, 6, and 7 and Remark 2 were modified.	(a)
p.1508	37.3.2 Supply current characteristics: 4. Peripheral Functions (Common to all products) was modified.	(b)
Section 38 Package Drawings		
p.1571	38.5 44-pin Products: The figure was added.	(d)

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.

(1/36)

Edition	Description	Section
Rev.0.50	First Edition issued.	Throughout
Rev.0.80	The name of a connection pin to a capacitor TCAP was modified to TSCAP.	Overall
	“E20 emulator” was deleted.	
	1.1 Features: The descriptions were modified.	Section 1 Outline
	Figure 1 - 1 Part Number, Memory Size, and Package of RL78/G23 was modified.	
	Table 1 - 1 List of Ordering Part Numbers was modified.	
	1.3.3 36-pin products: The figure was modified.	
	1.3.6 48-pin products: The figure was modified.	
	1.3.9 80-pin products: Cautions 1 to 3 were modified and Remark 3 was added.	
	1.3.10 100-pin products: Cautions 1 to 3 were modified and Remarks 1 to 3 were added.	
	1.3.11 128-pin products: Cautions 1 to 3 were modified and Remarks 1 to 3 were added.	
	1.6 Outline of Functions: The descriptions were modified and Notes 6 and 7 were added.	
	Figure 2 - 5 Pin Block Diagram for Pin Type 4-3-5 was modified.	Section 2 Pin Functions
	Figure 2 - 6 Pin Block Diagram for Pin Type 4-33-1 was modified.	
	Figure 2 - 7 Pin Block Diagram for Pin Type 4-35-1 was modified.	
	Figure 2 - 8 Pin Block Diagram for Pin Type 4-37-1 was modified.	
	Figure 3 - 1 Memory Map (R7F100GxF (x = A, B, C, E, F, G, J, L)): Note 2 was modified.	Section 3 CPU Architecture
	Figure 3 - 2 Memory Map (R7F100GxG (x = A, B, C, E, F, G, J, L, M, P)): Note 2 was modified.	
	Figure 3 - 3 Memory Map (R7F100GxH (x = A, B, C, E, F, G, J, L, M, P)): Note 2 was modified.	
	Figure 3 - 4 Memory Map (R7F100GxJ (x = A, B, C, E, F, G, J, L, M, P, S)): Note 2 was modified.	
	Figure 3 - 5 Memory Map (R7F100GxK (x = F, G, J, L, M, P, S)): Note 2 was modified.	
	Figure 3 - 6 Memory Map (R7F100GxL (x = F, G, J, L, M, P, S)): Note 2 was modified.	
	Figure 3 - 7 Memory Map (R7F100GxN (x = F, G, J, L, M, P, S)): Note 2 was modified.	
	Table 3 - 5 Vector Table was modified.	
	3.1.1 Internal program memory space: (2), (3), and (4) were modified.	
	Figure 3 - 8 Format of Configuration of Processor Mode Control Register (PMC) was modified.	
	Figure 3 - 9 Correspondence Between Data Memory and Addressing was modified.	
	Table 3 - 7 List of Special Function Registers (SFRs): Note 2 was added.	
	Table 3 - 8 List of Extended Special Function Registers (2nd SFRs): The register name was modified.	
	Table 4 - 1 Port Configuration was modified.	Section 4 Port Function
	4.3 Registers Controlling Port Function: The description was modified.	
	4.3.12 Output current control enable register (CCDE): The descriptions were modified.	
	4.3.13 Output current select registers (CCSx): The descriptions were modified.	
	4.3.14 40-mA port output control register (PTDC): The title and descriptions were modified.	

(2/36)

Edition	Description	Section
Rev.0.80	Figure 4 - 14 Format of 40-mA Port Output Control Register (PTDC): The title was modified.	Section 4 Port Function
	4.3.15 Port function output enable registers (PFOEx): The descriptions were modified.	
	4.4.5 Handling different potential (1.8 V, 2.5 V, or 3 V) by using I/O buffers: The descriptions in (1) were modified.	
	4.4.5 Handling different potential (1.8 V, 2.5 V, or 3 V) by using I/O buffers: The descriptions in (2) were modified.	
	4.5.4 Examples of register settings for port and alternate functions was added.	
	5.4 Transition between Flash Operation Modes: The descriptions were modified.	Section 5 Operation State Control
	Figure 5 - 4 State Transitions between Flash Operation Modes was modified.	
	5.5.2 Details of LS (low-speed main) mode: The descriptions and Note were modified.	
	Figure 5 - 6 Operating Range in LS Mode is modified.	
	5.5.3 Details of LP (low-power main) mode: The descriptions were modified.	
	The description in (1) Main system clock in 6.1 Functions of Clock Generator was modified.	Section 6 Clock Generator
	Figure 6 - 1 Block Diagram of Clock Generator was modified.	
	The description, note 3, caution 6, and caution 7 in Figure 6 - 2 Format of Clock Operation Mode Control Register (CMC) were modified.	
	Caution 5 in Figure 6 - 4 Format of Clock Operation Status Control Register (CSC) was modified.	
	The description and note in Figure 6 - 7 Format of Peripheral Enable Register 0 (PER0) were modified.	
	The description and note in Figure 6 - 8 Format of Peripheral Enable Register 1 (PER1) were modified.	
	The description and note 7 in Figure 6 - 9 Format of Subsystem Clock Supply Mode Control Register (OSMC) were modified.	
	The description in 6.3.12 High-speed on-chip oscillator trimming register (HIOTRM) was modified.	
	The description in 6.3.13 Middle-speed on-chip oscillator trimming register (MIOTRM) was modified.	
	The description in 6.3.14 Low-speed on-chip oscillator trimming register (LIOTRM) was modified.	
	The description in 6.4.1 X1 oscillator was modified.	
	The caution in Figure 6 - 19 Examples of External Circuits for the XT1 Oscillator was modified.	
	Table 6 - 2 Examples of Transitions of the CPU Clock and SFR Settings was modified.	
	A note was added to Figure 7 - 11 Format of Peripheral Enable Register 0 (PER0).	Section 7 Timer Array Unit (TAU)
	Figure 8 - 1 Block Diagram of the Realtime Clock was modified.	Section 8 Realtime Clock (RTC)
	Figure 8 - 3 Format of Subsystem Clock Supply Mode Control Register (OSMC) was modified and Notes 1 and 2 were added.	
	Figure 8 - 4 Format of Realtime Clock Control Register 0 (RTCC0): Note was modified, and Caution 3 was added.	
	Figure 9 - 2 Format of Peripheral Enable Register 1 (PER1) was modified.	Section 9 32-bit Interval Timer (TML32)
	The description in 9.2.6 Interval timer capture register 000 (ITLCAP000) was modified.	
	The description in 9.2.7 Interval timer capture register 00 (ITLCAP00) was modified.	

(3/36)

Edition	Description	Section
Rev.0.80	The description in 9.2.10 Interval timer frequency division register 0 (ITLFDIV00) was modified.	Section 9 32-bit Interval Timer (TML32)
	A caution and remark were added to 9.2.13 Interval timer status register (ITLS0).	
	The description in 9.3.2 Capture mode settings was modified.	
	The description in 9.3.3 Timer operation was modified.	
	Figure 10 - 2 Format of Clock Output Select Register n (CKSn) was modified.	Section 10 Clock Output/Buzzer Output Controller (PCLBUZ)
	Figure 12 - 2 Format of Peripheral Enable Register 0 (PER0): Note 3 was added.	Section 12 A/D Converter (ADC)
	Figure 12 - 5 Timing Chart When A/D Voltage Comparator Is Used: Note 1 was modified.	
	Figure 12 - 7 Format of A/D Converter Mode Register 1 (ADM1) was modified.	
	Figure 12 - 37 Setting up Software Trigger Wait Mode was modified.	
	Figure 12 - 41 Flowchart for Setting up SNOOZE Mode (Software Trigger) was modified.	
	Figure 13 - 2 Format of Peripheral Enable Register 1 (PER1): Note was added.	Section 13 D/A Converter (DAC)
	Figure 14 - 2 Format of Peripheral Enable Register 1 (PER1): Note was added.	Section 14 Comparator (CMP)
	Figure 15 - 5 Format of Peripheral Enable Register 0 (PER0): Note 2 was added.	Section 15 Serial Array Unit (SAU)
	16.1 Functions of Serial Interface IICA (3): The descriptions were modified.	Section 16 Serial Interface IICA (IICA)
	Figure 16 - 5 Format of Peripheral Enable Register 0 (PER0) was modified.	
	Figure 17 - 1 Block Diagram of UARTAn was modified.	Section 17 Serial Interface UARTA (UARTA)
	17.2 Register Descriptions: Remark was modified.	
	Figure 17 - 2 Format of Peripheral Enable Register 1 (PER1): The descriptions and Note were modified.	
	17.2.8 Status clear trigger register (ASCTAn) (n = 0, 1): The descriptions were modified.	
	Figure 17 - 13 Flowchart of Communication Procedure: Caution was modified.	
	Figure 18 - 2 Format of Peripheral Enable Register 1 (PER1): Note was added.	Section 18 Remote Control Signal Receiver (REMC)
	Figure 18 - 5 Format of Function Select Register 1 (REMCON1) was modified.	
	Figure 18 - 7 Format of Interrupt Control Register (REMINT) was modified.	
	Figure 18 - 8 Format of Compare Control Register (REMCPC) was modified.	
	Figure 18 - 20 Format of Receive Bit Count Register (REMRBIT) was modified.	
	Figure 18 - 25 Example of Flowchart for Initial Settings of REMC was modified.	
	18.3.4.1 When using f_{SXL} as the REMC operating clock: The descriptions were modified.	
	18.3.5 RIN0 Input: The descriptions were modified.	
	18.3.9 Compare Function: The descriptions were modified.	
	Table 18 - 6 Interrupt Modes and Transition from SNOOZE Mode was modified.	
	Figure 18 - 44 Example of Flowchart for Setting up SNOOZE Mode was modified.	
	18.4.5 Reading Registers: The descriptions were modified.	
	Figure 19 - 5 Format of Peripheral Enable Register 1 (PER1): Note was added.	Section 19 Data Transfer Controller (DTC)

(4/36)

Edition	Description	Section
Rev.0.80	Figure 20 - 2 Block Diagram of Logic Cell Block L1 was modified.	Section 20 Logic and Event Link Controller (ELCL)
	Figure 20 - 3 Block Diagram of Logic Cell Block L2 was modified.	
	Figure 20 - 4 Block Diagram of Logic Cell Block L3 was modified.	
	The description and caution 2 in Figure 20 - 5 Format of Input Signal Select Registers n (ELISELn) (n = 0 to 11) were modified.	
	Caution 2 in Figure 20 - 6 Format of Event Link L1 Signal Select Registers n (ELL1SELn) (n = 0 to 3) was modified.	
	Caution 2 in Figure 20 - 7 Format of Event Link L1 Signal Select Registers n (ELL1SELn) (n = 4, 5) was modified.	
	Caution 2 in Figure 20 - 8 Format of Event Link L1 Signal Select Register 6 (ELL1SEL6) was modified.	
	The description and caution 2 in Figure 20 - 9 Format of Logic Cell Block L1 Control Register (ELL1CTL) were modified.	
	Caution 3 in Figure 20 - 10 Format of Event Link L1 Output Select Registers n (ELL1LNKn) (n = 0 to 3) was modified.	
	Caution 3 in Figure 20 - 11 Format of Event Link L1 Output Select Registers n (ELL1LNKn) (n = 4, 5) was modified.	
	Caution 2 in Figure 20 - 12 Format of Event Link L1 Output Select Register 6 (ELL1LNK6) was modified.	
	Caution 2 in Figure 20 - 13 Format of Event Link L2 Signal Select Registers n (ELL2SELn) (n = 0 to 3) was modified.	
	Caution 2 in Figure 20 - 14 Format of Event Link L2 Signal Select Registers n (ELL2SELn) (n = 4, 5) was modified.	
	Caution 2 in Figure 20 - 15 Format of Event Link L2 Signal Select Register 6 (ELL2SEL6) was modified.	
	The description and caution 2 in Figure 20 - 16 Format of Logic Cell Block L2 Control Register (ELL2CTL) were modified.	
	Caution 3 in Figure 20 - 17 Format of Event Link L2 Output Select Registers n (ELL2LNKn) (n = 0 to 3) was modified.	
	Caution 3 in Figure 20 - 18 Format of Event Link L2 Output Select Registers n (ELL2LNKn) (n = 4, 5) was modified.	
	Caution 2 in Figure 20 - 19 Format of Event Link L2 Output Select Register 6 (ELL2LNK6) was modified.	
	Caution 2 in Figure 20 - 20 Format of Event Link L3 Signal Select Registers n (ELL3SELn) (n = 0 to 3) was modified.	
	Caution 2 in Figure 20 - 21 Format of Event Link L3 Signal Select Registers n (ELL3SELn) (n = 4, 5) was modified.	
	Caution 2 in Figure 20 - 22 Format of Event Link L3 Signal Select Register 6 (ELL3SEL6) was modified.	
	The description and caution 2 in Figure 20 - 23 Format of Logic Cell Block L3 Control Register (ELL3CTL) were modified.	
	Caution 3 in Figure 20 - 24 Format of Event Link L3 Output Select Registers n (ELL3LNKn) (n = 0 to 3) was modified.	
	Caution 3 in Figure 20 - 25 Format of Event Link L3 Output Select Registers n (ELL3LNKn) (n = 4, 5) was modified.	
	Caution 2 in Figure 20 - 26 Format of Event Link L3 Output Select Register 6 (ELL3LNK6) was modified.	
	The description in 20.4 ELCL Operation was modified.	
	Point to note (4) was added to 20.6 Usage Notes.	

(5/36)

Edition	Description	Section
Rev.0.80	Table 21 - 2 Flags Corresponding to Interrupt Request Sources (4/5): Notes 1 and 2 were added.	Section 21 Interrupt Functions
	Table 23 - 1 Operating Statuses in HALT Mode (1) was modified.	Section 23 Standby Function
	Table 23 - 2 Operating Statuses in HALT Mode (2) was modified.	
	Table 23 - 3 Operating Statuses in STOP Mode was modified.	
	Table 23 - 4 Operating Statuses in SNOOZE Mode was modified.	
	Figure 27 - 16 Format of Guard Register of IAWCTL Register (GIAWCTL) was modified.	Section 27 Safety Functions
	Figure 29 - 8 Flow of Internal Operations of the SNOOZE Mode Sequencer was modified.	Section 29 SNOOZE Mode Sequencer (SMS)
	The description in 29.5.13 Branch 2 (SCY = 0) was modified.	
	The description in 29.5.14 Branch 3 (SZ = 1) was modified.	
	The description in 29.5.15 Branch 4 (SZ = 0) was modified.	
	The description in 29.5.16 Wait was modified.	
	Section 30 CAPACITIVE SENSING UNIT (CTSU2L): The descriptions were modified.	Section 30 Capacitive Sensing Unit (CTSU2L)
	Table 30 - 1 CTSU Functions was modified.	
	Figure 30 - 3 CTSU Block Diagram was modified.	
	Table 30 - 2 External Pins Used in CTSU was modified.	
	30.3 Operation was deleted.	
	30.4 Usage Notes was deleted.	
	Figure 33 - 9 Boot Swap Function was modified.	Section 33 Flash Memory
	37.1 Absolute Maximum Ratings was modified.	Section 37 Electrical Characteristics TA = -40 to +105°C
	37.2.1 X1, XT1 oscillator characteristics was modified.	
	37.3.1 Pin characteristics (2/7): Note 3 was modified.	
	37.3.2 Supply current characteristics (1) 30- to 64-pin package products with 96- to 128-KB ROM (2/4): Note 3 was modified.	
	37.3.2 Supply current characteristics (1) 30- to 64-pin package products with 96- to 128-KB ROM (4/4): Notes 4, 5, and 8 were modified.	
	37.3.2 Supply current characteristics (2) 30- to 80-pin package products with 192- to 256-KB flash ROM (2/4): Note 3 was modified.	
	37.3.2 Supply current characteristics (2) 30- to 80-pin package products with 192- to 256-KB flash ROM (4/4): Notes 5 and 8 were modified.	
	37.3.2 Supply current characteristics (3) 44- to 80-pin package products with 384- to 768-KB flash ROM, and 100- and 128-pin products (2/4): Note 3 was modified.	
	37.3.2 Supply current characteristics (3) 44- to 80-pin package products with 384- to 768-KB flash ROM, and 100- and 128-pin products (4/4): Notes 5 and 8 were modified.	
	37.4 AC Characteristics was modified.	
	37.5.3 Serial interface IICA (1) was modified.	
	37.5.3 Serial interface IICA (2) was modified.	
	37.5.3 Serial interface IICA (3) was modified.	
	37.6.3 D/A converter characteristics was modified.	

(6/36)

Edition	Description	Section
Rev.0.80	37.6.4 Comparator characteristics was modified.	Section 37 Electrical Characteristics TA = -40 to +105°C
Rev.0.90	Table 1 - 1 List of Ordering Part Numbers was modified.	Section 1 Outline
	1.6 Outline of Functions: Tables were modified.	
	Table 3 - 7 List of Special Function Registers (SFRs) was modified.	Section 3 CPU Architecture
	Table 3 - 8 List of Extended Special Function Registers (2nd SFRs) was modified.	
	Figure 4 - 12 Format of Output Current Control Enable Register (CCDE): Figure was modified and caution was added.	Section 4 Port Functions
	4.3.14 40-mA port output control register (PTDC): The descriptions were modified.	
	Figure 4 - 14 Format of 40-mA Port Output Control Register (PTDC) was modified.	
	Figure 4 - 15 Format of Port Function Output Enable Register 0 (PFOE0) was modified.	
	Figure 4 - 16 Format of Port Function Output Enable Register 1 (PFOE1) was modified.	
	Table 4 - 5 Concept of Basic Settings: Table and note 1 were modified.	
	4.5.2 Register settings for alternate function whose output function is not used: The descriptions were modified.	
	Table 4 - 7 Examples of Register and Output Latch Settings for Alternate Functions (30-Pin to 64-Pin Products with 96-Kbyte or 128-Kbyte Flash Memory) was modified.	
	Table 4 - 8 Examples of Register and Output Latch Settings for Alternate Functions (Products with 192-Kbyte to 768-Kbyte Flash Memory and 80-Pin and 100-Pin Products with 128-Kbyte Flash Memory) was modified.	
	Table 5 - 2 Features of Each Flash Operation Mode: Note was modified.	Section 5 Operation State Control
	Figure 6 - 1 Block Diagram of Clock Generator: Figure and remark were modified.	Section 6 Clock Generator
	Figure 6 - 7 Format of Peripheral Enable Register 0 (PER0): Figure was modified and note was deleted.	
	Figure 6 - 8 Format of Peripheral Enable Register 1 (PER1): Figure was modified and note was deleted.	
	Figure 6 - 9 Format of Subsystem Clock Supply Mode Control Register (OSMC): Figure and notes 2 to 4 were modified.	
	6.3.8 Subsystem clock select register (CKSEL): The descriptions were modified.	
	Figure 6 - 10 Format of Subsystem Clock Select Register (CKSEL): Figure and note were modified.	
	6.5 Operations of the Clock Generator: The descriptions were modified.	
	Table 6 - 2 Examples of Transitions of the CPU Clock and SFR Settings was modified.	
	Figure 7 - 11 Format of Peripheral Enable Register 0 (PER0) was modified.	Section 7 Timer Array Unit (TAU)
	Figure 7 - 12 Format of Peripheral Reset Control Register 0 (PRR0) was modified.	
	Figure 7 - 19 Format of Timer Input Select Register 0 (TIS0) was modified.	
	7.10.2 Point for caution when a timer output is to be used as an event input for the ELCL: The descriptions were modified.	
	Figure 8 - 3 Format of Subsystem Clock Supply Mode Control Register (OSMC): Figure, note 1, and caution were modified.	Section 8 Realtime Clock (RTC)
	9.1 Overview: The descriptions were modified.	Section 9 32-bit Interval Timer (TML32)
	Table 9 - 1 Specifications of 32-Bit Interval Timer Operations: Table and remark were modified.	
	Figure 9 - 1 Block Diagram of 32-Bit Interval Timer: Figure and note were modified.	

(7/36)

Edition	Description	Section
Rev.0.90	Table 9 - 2 List of Registers was modified.	Section 9 32-bit Interval Timer (TML32)
	Figure 9 - 2 Format of Peripheral Enable Register 1 (PER1): Figure was modified and note was deleted.	
	Figure 9 - 3 Format of Peripheral Reset Control Register 1 (PRR1) was modified.	
	9.2.4 Interval timer compare registers 0mn (ITLCMP0mn) (mn = 00, 01, 12, 13): Title and descriptions were modified.	
	Figure 9 - 4 Format of Interval Timer Compare Registers 0mn (ITLCMP0mn) was modified.	
	9.2.5 Interval timer compare registers 0n (ITLCMP0n) (n = 0, 1): The descriptions were modified.	
	Figure 9 - 5 Format of Interval Timer Compare Registers 0n (ITLCMP0n) was modified.	
	9.2.6 Interval timer capture register 00 (ITLCAP00): The descriptions were modified.	
	Figure 9 - 6 Format of Interval Timer Capture Register 00 (ITLCAP00) was modified.	
	Figure 9 - 7 Format of Interval Timer Control Register (ITLCTL0): Figure and notes were modified.	
	Figure 9 - 8 Format of Interval Timer Clock Select Register 0 (ITLCSEL0): Figure and remark were modified.	
	9.2.9 Interval timer frequency division register 0 (ITLFDIV00): The descriptions were modified.	
	Figure 9 - 9 Format of Interval Timer Frequency Division Register 0 (ITLFDIV00): Figure and notes 1 and 2 were modified.	
	9.2.10 Interval timer frequency division register 1 (ITLFDIV01): The descriptions were modified.	
	Figure 9 - 10 Format of Interval Timer Frequency Division Register 1 (ITLFDIV01): Figure and notes 1 and 2 were modified.	
	Figure 9 - 11 Format of Interval Timer Capture Control Register 0 (ITLCC0): Figure was modified and note 7 was added.	
	9.2.12 Interval timer status register (ITLS0): The descriptions were modified.	
	Table 9 - 3 Conditions for Setting the Interrupt Status Flags in Each Timer Mode was modified.	
	9.2.13 Interval timer match detection mask register (ITLMKF0): The descriptions were modified.	
	Table 9 - 4 Registers and Settings Used in 8-Bit Counter Mode: Table and remark were modified.	
	Table 9 - 5 Registers and Settings Used in 16-Bit Counter Mode was modified.	
	Table 9 - 6 Registers and Settings Used in 32-Bit Counter Mode was modified.	
	Table 9 - 7 Registers and Settings Used in 16-Bit Capture Mode: Caution was added.	
	9.3.3 Timer operation: The descriptions were modified.	
	Figure 9 - 14 Example of Timer Operation: Figure and remark were modified.	
	9.3.4 Capture operation: The descriptions were modified.	
	Figure 9 - 15 Example of Capture Operation: Figure and remark were modified.	
	Table 9 - 8 Interrupt Sources in 8-Bit, 16-Bit, and 32-Bit Counter Modes was modified.	
	Figure 9 - 16 Procedure for Starting the 32-Bit Interval Timer was modified.	
	Figure 9 - 17 Procedure for Stopping the 32-Bit Interval Timer was modified.	
	Figure 9 - 18 Procedure for Changing the Operating Mode of the 32-Bit Interval Timer was modified.	
	Figure 9 - 19 Procedure for Resetting the 32-Bit Interval Timer was modified.	

(8/36)

Edition	Description	Section
Rev.0.90	Figure 9 - 20 Procedure for Starting Event Input from the ELCL was modified.	Section 9 32-bit Interval Timer (TML32)
	Figure 9 - 21 Procedure for Stopping Event Input from the ELCL was modified.	
	10.1 Functions of Clock Output/Buzzer Output Controller: Caution was modified.	Section 10 Clock Output/Buzzer Output Controller (PCLBUZ)
	Figure 10 - 1 Block Diagram of Clock Output/Buzzer Output Controller was modified.	
	Figure 10 - 2 Format of Clock Output Select Register n (CKSn): Figure, caution 3, and remark 2 were modified.	
	12.3 Registers Controlling A/D Converter: The descriptions were modified.	Section 12 A/D Converter (ADC)
	Figure 12 - 2 Format of Peripheral Enable Register 0 (PER0) was modified.	
	Figure 12 - 3 Format of Peripheral Reset Control Register 0 (PRR0) was modified.	
	Table 12 - 3 A/D Conversion Time Selection was modified.	
	Figure 12 - 8 Format of A/D Converter Mode Register 2 (ADM2) was modified.	
	12.8.2 A/D conversion by inputting a hardware trigger: The descriptions were modified and caution was added.	
	12.10 Points for Caution when the A/D Converter is to be Used: The descriptions were modified.	
	Figure 12 - 52 Analog Input Pin Connection was modified.	
	Section 13 D/A CONVERTER (DAC): The descriptions were modified.	Section 13 D/A Converter (DAC)
	Figure 13 - 1 Block Diagram of D/A Converter was modified.	
	Figure 13 - 2 Format of Peripheral Enable Register 1 (PER1): Figure was modified and note was deleted.	
	Figure 13 - 3 Format of Peripheral Reset Control Register 1 (PRR1) was modified.	
	13.4.2 Operation in realtime output mode: The descriptions were modified.	
	Figure 13 - 6 Timing for Outputting D/A Conversion Value was modified.	
	13.5 Points for Caution when the D/A Converter is to be Used: (8) was deleted.	
	Figure 14 - 1 Comparator Block Diagram was modified.	Section 14 Comparator (CMP)
	Figure 14 - 2 Format of Peripheral Enable Register 1 (PER1): Figure was modified and note was deleted.	
	Figure 14 - 3 Format of Peripheral Reset Control Register 1 (PRR1): Figure was modified and caution 2 was added.	
	Figure 14 - 4 Format of Comparator Mode Setting Register (COMPMDR): Note 3 was added.	
	Figure 15 - 1 Block Diagram of Serial Array Unit 0 was modified.	Section 15 Serial Array Unit (SAU)
	Figure 15 - 2 Block Diagram of Serial Array Unit 1 was modified.	
	Figure 15 - 5 Format of Peripheral Enable Register 0 (PER0) was modified.	
	Figure 15 - 6 Format of Peripheral Reset Control Register 0 (PRR0) was modified.	
	Figure 15 - 22 Format of Input Switch Control Register (ISC): Figure was modified and notes 1 to 4 were added.	
	15.5.9 Procedure for Processing Errors that Occurred During 3-wire Serial I/O (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31) Communication: The descriptions were modified.	
	Figure 16 - 5 Format of Peripheral Enable Register 0 (PER0) was modified.	Section 16 Serial Interface IICA (IICA)
	Figure 16 - 6 Format of Peripheral Reset Control Register 0 (PRR0) was modified.	
	Figure 16 - 7 Format of IICA Control Register n0 (IICCTLn0): Note 2 was modified.	

(9/36)

Edition	Description	Section
Rev.0.90	Figure 17 - 1 Block Diagram of UARTAn: Figure and remark were modified.	Section 17 Serial Interface UARTA (UARTA)
	Figure 17 - 2 Format of Peripheral Enable Register 1 (PER1): Figure was modified and note was deleted.	
	17.2.9 UARTA clock select register 0 (UTA0CK): The descriptions were modified.	
	Figure 17 - 10 Format of UARTA0 Clock Select Register (UTA0CK): Figure and remark were modified.	
	17.2.10 UARTA clock select register 1 (UTA1CK): The descriptions were modified.	
	Figure 17 - 11 Format of UARTA1 Clock Select Register (UTA1CK): Figure and remark were modified.	
	17.3.2 UART Mode, (4) Normal transmission: The descriptions were modified.	
	Table 17 - 8 Set Data of Baud Rate Generator was modified.	
	17.4.3 Point for Caution when Selecting the UARTAn Operation Clock (f_{UTAn}): The descriptions were modified.	
	Table 18 - 1 Specifications of Remote Control Signal Receiver: Table and note 2 were modified.	Section 18 Remote Control Signal Receiver (REMC)
	Figure 18 - 1 Block Diagram of Remote Control Signal Receiver was modified.	
	Figure 18 - 2 Format of Peripheral Enable Register 1 (PER1): Figure was modified and note was deleted.	
	Figure 18 - 3 Format of Peripheral Reset Control Register 1 (PRR1) was modified.	
	Figure 18 - 5 Format of Function Select Register 1 (REMCON1) was modified.	
	Figure 18 - 19 Format of Receiver Standby Control Register (REMSTC) was modified.	
	18.3.4 Operating Clock: The descriptions were modified.	
	Figure 18 - 28 RIN0 Internal Input Signal Generation Configuration was modified.	
	18.3.5 RIN0 Input: The descriptions were modified.	
	18.3.12 Interrupts: The descriptions were modified.	
	18.3.13 SNOOZE Mode Function: The descriptions and remark were modified.	
	Figure 18 - 44 Example of Flowchart for Setting up SNOOZE Mode was modified.	
	18.4.1 Register Access when Starting Operation of the Remote Control Signal Receiver: Title was modified.	
	18.4.2 Timing of Changing the Register Values: Title was modified.	
	18.4.4 Changing the Operating Clock: Title was modified.	
	Figure 19 - 5 Format of Peripheral Enable Register 1 (PER1): Note was deleted.	Section 19 Data Transfer Controller (DTC)
	19.5.8 Operation in standby mode: The descriptions were modified and note 1 was added.	
	Table 20 - 1 Connections in Logic Cell Block L1: Note 2 was modified.	Section 20 Logic and Event Link Controller (ELCL)
	Table 20 - 2 Connections in Logic Cell Block L2: Note 2 was modified.	
	Table 20 - 3 Connections in Logic Cell Block L3: Note 2 was modified.	
	Figure 20 - 5 Format of Input Signal Select Registers n (ELISELn) (n = 0 to 11): Notes 3 and 4 and caution 2 were modified.	
	Figure 20 - 6 Format of Event Link L1 Signal Select Registers n (ELL1SELn) (n = 0 to 3): Caution 2 was modified.	
	Figure 20 - 7 Format of Event Link L1 Signal Select Registers n (ELL1SELn) (n = 4, 5): Caution 2 was modified.	
	Figure 20 - 8 Format of Event Link L1 Signal Select Register 6 (ELL1SEL6): Caution 2 was modified.	

(10/36)

Edition	Description	Section
Rev.0.90	Figure 20 - 9 Format of Logic Cell Block L1 Control Register (ELL1CTL): Caution 2 was modified.	Section 20 Logic and Event Link Controller (ELCL)
	Figure 20 - 10 Format of Event Link L1 Output Select Registers n (ELL1LNKn) (n = 0 to 3): Caution 3 was modified.	
	Figure 20 - 11 Format of Event Link L1 Output Select Registers n (ELL1LNKn) (n = 4, 5): Caution 3 was modified.	
	Figure 20 - 12 Format of Event Link L1 Output Select Register 6 (ELL1LNK6): Caution 2 was modified.	
	Figure 20 - 13 Format of Event Link L2 Signal Select Registers n (ELL2SELn) (n = 0 to 3): Caution 2 was modified.	
	Figure 20 - 14 Format of Event Link L2 Signal Select Registers n (ELL2SELn) (n = 4, 5): Caution 2 was modified.	
	Figure 20 - 15 Format of Event Link L2 Signal Select Register 6 (ELL2SEL6): Caution 2 was modified.	
	Figure 20 - 16 Format of Logic Cell Block L2 Control Register (ELL2CTL): Caution 2 was modified.	
	Figure 20 - 17 Format of Event Link L2 Output Select Registers n (ELL2LNKn) (n = 0 to 3): Caution 3 was modified.	
	Figure 20 - 18 Format of Event Link L2 Output Select Registers n (ELL2LNKn) (n = 4, 5): Caution 3 was modified.	
	Figure 20 - 19 Format of Event Link L2 Output Select Register 6 (ELL2LNK6): Caution 2 was modified.	
	Figure 20 - 20 Format of Event Link L3 Signal Select Registers n (ELL3SELn) (n = 0 to 3): Caution 2 was modified.	
	Figure 20 - 21 Format of Event Link L3 Signal Select Registers n (ELL3SELn) (n = 4, 5): Caution 2 was modified.	
	Figure 20 - 22 Format of Event Link L3 Signal Select Register 6 (ELL3SEL6): Caution 2 was modified.	
	Figure 20 - 23 Format of Logic Cell Block L3 Control Register (ELL3CTL): Caution 2 was modified.	
	Figure 20 - 24 Format of Event Link L3 Output Select Registers n (ELL3LNKn) (n = 0 to 3): Caution 3 was modified.	
	Figure 20 - 25 Format of Event Link L3 Output Select Registers n (ELL3LNKn) (n = 4, 5): Caution 3 was modified.	
	Figure 20 - 26 Format of Event Link L3 Output Select Register 6 (ELL3LNK6): Caution 2 was modified.	
	Figure 20 - 27 Format of Output Signal Select Registers n (ELOSELn) (n = 0 to 7) was modified.	
	Figure 20 - 28 Format of Output Signal Enable Register (ELOENCTL): Note was modified.	
	Table 20 - 5 Bits in Output Signal Enable Register and Event-Receiving Peripheral Functions: Table was modified and note 4 was added.	
	20.6 Points for Caution when the ELCL is to be Used: (1) and (3) were modified.	
	23.1 Standby Function: Caution 2 was modified.	Section 23 Standby Function
	23.2.2 Memory power reduction control register (PSMCR): The descriptions were added.	
	Table 23 - 1 Operating Statuses in HALT Mode (1) was modified.	
	Table 23 - 2 Operating Statuses in HALT Mode (2) was modified.	
	Table 23 - 3 Operating Statuses in STOP Mode: Table and remark 1 were modified.	
	Table 23 - 4 Operating Statuses in SNOOZE Mode: Table and remark 1 were modified.	

(11/36)

Edition	Description	Section
Rev.0.90	Figure 24 - 7 Format of Peripheral Reset Control Register 0 (PRR0) was modified.	Section 24 Reset Function
	Figure 24 - 8 Format of Peripheral Reset Control Register 1 (PRR1) was modified.	
	25.1 Functions of Power-on-reset Circuit: The descriptions were modified.	Section 25 Power-on-Reset Circuit (POR)
	Figure 25 - 2 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector: Caution was modified.	
	Figure 25 - 2 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector: Remark 2 was modified.	
	26.1 Functions of Voltage Detector: The descriptions were modified.	Section 26 Voltage Detector (LVD)
	Figure 26 - 3 Format of Voltage Detection Register (LVIM): Figure was modified and note 4 was added.	
	Figure 26 - 5 Format of Voltage Detection Level Register (LVIS): Figure and notes 1 and 2 were modified.	
	26.4.1 When used as reset mode: The descriptions were modified.	
	Figure 26 - 7 Timing of LVD1 Internal Reset Signal Generation was added.	
	26.4.2 When used as interrupt mode: The descriptions were modified.	
	Figure 26 - 9 Timing of LVD1 Interrupt Signal Generation was added.	
	Figure 27 - 23 Format of UART Loopback Select Register (ULBS) was modified.	Section 27 Safety Functions
	Figure 29 - 2 Format of Peripheral Enable Register 1 (PER1): Figure was modified and note was deleted.	Section 29 SNOOZE Mode Sequencer (SMS)
	Figure 29 - 3 Format of Peripheral Reset Control Register 1 (PRR1) was modified.	
	Table 29 - 3 Types of Processing Specified by the SMSIp Registers: Note 3 was modified.	
	29.5 Commands for Use in Processing by the Sequencer: The descriptions were modified.	
	29.5.1 8-bit data transfer 1: The descriptions were modified.	
	29.5.2 8-bit data transfer 2: The descriptions were modified.	
	29.5.3 16-bit data transfer 1: The descriptions were modified.	
	29.5.4 16-bit data transfer 2: The descriptions were modified.	
	29.5.5 1-bit data setting: The descriptions were modified.	
	29.5.6 1-bit data clearing: The descriptions were modified.	
	29.5.7 1-bit data transfer: The descriptions were modified.	
	29.5.12 Branch 1 (SCY = 1): The descriptions were modified.	
	29.5.13 Branch 2 (SCY = 0): The descriptions were modified.	
	29.5.14 Branch 3 (SZ = 1): The descriptions were modified.	
	29.5.15 Branch 4 (SZ = 0): The descriptions were modified.	
	29.5.16 Wait: The descriptions and note 1 were modified.	
	29.5.17 Conditional wait 1 (bit = 1): The descriptions were modified.	
	29.5.18 Conditional wait 2 (bit = 0): The descriptions were modified.	
	29.5.19 Termination: The descriptions were modified.	
	29.5.20 Interrupt plus termination: The descriptions were modified.	
	29.6 Operation in Standby Modes: The descriptions were modified and note 1 was added.	

(12/36)

Edition	Description	Section
Rev.0.90	30.2 Registers Controlling CTSU: The descriptions were modified.	Section 30 Capacitive Sensing Unit (CTSU2L)
	Figure 30 - 5 Format of Peripheral Enable Register 1 (PER1): Figure was modified and note was deleted.	
	Figure 30 - 6 Format of Peripheral Reset Control Register 1 (PRR1) was modified.	
	30.2.10 CTSU sensor counter register L/H (CTSUSC/CTSUUC): Title was modified.	
	Figure 30 - 14 Format of CTSU Sensor Counter Register L/H (CTSUSC/CTSUUC): Title was modified.	
	30.2.13 CTSU trimming register AL/AH (CTSUTRIM0/CTSUTRIM1): Title and the descriptions were modified.	
	Figure 30 - 17 Format of CTSU Trimming Register AL/AH (CTSUTRIM0/CTSUTRIM1): Title was modified.	
	30.2.14 CTSU trimming register BL/BH (CTSUTRIM2/CTSUTRIM3) was added.	
	Figure 32 - 2 Format of User Option Byte (000C1H or 040C1H) was modified.	Section 32 Option Bytes
	Figure 32 - 3 Format of User Option Byte (000C1H or 040C1H) was modified.	
	34.2 Connection between the External Device that Incorporates UART and RL78/G23: Note 3 was added.	Section 34 On-chip Debugging
	Table 36 - 16 Operation List: Caution was modified.	Section 36 Instruction Set
	37.2.1 Characteristics of the X1 and XT1 oscillators was modified.	Section 37 Electrical Characteristics TA = -40 to +105°C
	37.2.2 Characteristics of the on-chip oscillators was modified.	
	37.3.2 Supply current characteristics, (1) 30- to 64-pin package products with 96- to 128-Kbyte flash ROM (1/4): Note 2 was modified.	
	37.3.2 Supply current characteristics, (1) 30- to 64-pin package products with 96- to 128-Kbyte flash ROM (2/4): Note 3 was modified.	
	37.3.2 Supply current characteristics, (2) Peripheral Functions (Common to all products) was modified.	
	37.4 AC Characteristics was modified.	
	37.6.4 Comparator characteristics was modified.	
	37.6.6 LVD circuit characteristics, (1) LVD0 Detection Voltage in the Reset Mode and Interrupt Mode was modified.	
	37.6.6 LVD circuit characteristics, (2) LVD1 Detection Voltage of Reset Mode and Interrupt Mode was modified and Note was added.	
	37.6.7 Power supply voltage rising slope characteristics: Note was modified.	
Rev.1.00	“CSI” was modified to “SPI (CSI)”.	Overall
	“Wait” was modified to “clock stretch”.	
	1.1 Features: Note was added.	Section 1 Outline
	1.1 Features: The description was added.	
	Figure 1 - 1 Part Number, Memory Size, and Package of RL78/G23 was modified.	
	Table 1 - 1 List of Ordering Part Numbers: Tables were modified and notes 2 and 3 were added.	
	Figure 4 - 12 Format of Output Current Control Enable Register (CCDE): Caution 2 was modified.	Section 4 Port Functions
	4.4.5 Handling different potential (1.8 V, 2.5 V, or 3 V) by using I/O buffers, (1) Setting procedure when using input pins of UART0 to UART3, UARTA0, UARTA1, CSI00, CSI01, CSI10, CSI20, CSI30, and CSI31 functions in the TTL input buffer mode: Note was added.	

(13/36)

Edition	Description	Section
Rev.1.00	Table 4 - 7 Examples of Register and Output Latch Settings for Alternate Functions (30-Pin to 64-Pin Products with 96-Kbyte or 128-Kbyte Flash Memory) was modified.	Section 4 Port Functions
	Table 4 - 8 Examples of Register and Output Latch Settings for Alternate Functions (Products with 192-Kbyte to 768-Kbyte Flash Memory and 80-Pin and 100-Pin Products with 128-Kbyte Flash Memory) was modified.	
	Figure 6 - 1 Block Diagram of Clock Generator: Figure and remark were modified.	Section 6 Clock Generator
	Figure 8 - 19 Procedure for Starting the Realtime Clock Operation was modified.	Section 8 Realtime Clock (RTC)
	Table 12 - 4 Resistance and Capacitance Values of Equivalent Circuit (Reference Values) was modified.	Section 12 A/D Converter (ADC)
	Section 15 SERIAL ARRAY UNIT (SAU): Note was added.	Section 15 Serial Array Unit (SAU)
	Figure 15 - 1 Block Diagram of Serial Array Unit 0 was modified.	
	15.3.15 Serial standby control register m (SSCm): Caution was modified.	
	15.6.3 SNOOZE Mode Function: Caution 2 was modified.	
	Figure 17 - 1 Block Diagram of UARTAn was modified.	Section 17 Serial Interface UARTA (UARTA)
	Figure 17 - 5 Format of Operation Mode Setting Register 0 (ASIMAn0): Caution 4 was modified.	
	Figure 25 - 2 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector: Note 3 was modified.	Section 25 Power-on-Reset Circuit (POR)
	26.1 Functions of Voltage Detector: The description was modified.	Section 26 Voltage Detector (LVD)
	Figure 26 - 1 Block Diagram of LVD0 was modified.	
	Figure 26 - 2 Block Diagram of LVD1 was modified.	
	26.3.1 Voltage detection register (LVIM): The description was modified.	
	Figure 26 - 3 Format of Voltage Detection Register (LVIM): Note 1 was modified.	
	Figure 26 - 5 Format of Voltage Detection Level Register (LVIS): Figure and note 2 were modified and notes 3 and 4 were added.	
	26.4.1 When used as reset mode: The descriptions were modified.	
	Figure 26 - 6 Timing of LVD0 Internal Reset Signal Generation was modified.	
	Figure 26 - 7 Timing of LVD1 Internal Reset Signal Generation: Figure and remark were modified.	
	Figure 26 - 8 Timing of LVD0 Interrupt Signal Generation was modified.	
	Figure 26 - 9 Timing of LVD1 Interrupt Signal Generation: Figure and remark were modified.	
	26.5 Points for Caution when the Voltage Detector is to be Used: (5) was added.	
	Figure 27 - 13 Format of Invalid Memory Access Detection Control Register (IAWCTL) was modified.	Section 27 Safety Functions
	Table 30 - 1 CTSU Functions was modified.	Section 30 Capacitive Sensing Unit (CTSU2L)
	Figure 30 - 7 Format of CTSU Control Register AL/AH (CTSUCRAL/CTSUCRAH) was modified.	
	Figure 30 - 8 Format of CTSU Control Register BL/BH (CTSUCRBL/CTSUCRBH) was modified.	
	Figure 30 - 9 Format of CTSU Measurement Channel Register L/H (CTSUMCHL/CTSUMCHH) was modified.	

(14/36)

Edition	Description	Section
Rev.1.00	Figure 30 - 11 Format of CTSU Channel Transmit/Receive Control Register AL/AH/BL/BH (CTSUCHTRCAL/CTSUCHTRCAH/CTSUCHTRCBL/CTSUCHTRCBH): Caution 4 was modified.	Section 30 Capacitive Sensing Unit (CTSU2L)
	Figure 30 - 12 Format of CTSU Status Register L (CTSUSRL) was modified.	
	Figure 30 - 13 Format of CTSU Sensor Offset Register 0/1 (CTSUSO0/CTSUSO1) was modified.	
	Figure 30 - 15 Format of CTSU Calibration Register L/H (CTSUDBGR0/CTSUDBGR1) was modified.	
	Section 33 FLASH MEMORY: The descriptions were modified.	Section 33 Flash Memory
	33.2 Serial Programming Using External Device (that Incorporates UART): The descriptions were modified.	
	33.4.2 Flash memory programming mode: The descriptions were modified.	
	33.4.4 Communication commands: The descriptions were modified.	
	33.6 Self-Programming: Remark was deleted.	
	33.6.1 Self-programming procedure: The descriptions were modified.	
	33.6.2 Registers to control the flash memory was added.	
	33.6.3 Setting the flash memory control mode was added.	
	33.6.4 Clearing the registers for use with the flash memory sequencer was added.	
	33.6.5 Setting the operating frequency of the flash memory sequencer was added.	
	33.6.6 Commands for use with the flash memory sequencer in the respective areas was added.	
	33.6.8 Flash shield window function was modified.	
	Figure 33 - 29 Flash Shield Window Setting Example (Target Devices: R7F100GLG, Start Block Number in the Window Range: 04H, End Block Number in the Window Range: 06H, FSWC: 1) was modified.	
	Table 33 - 13 Relationship between Flash Shield Window Function Setting/Change Methods and Commands was modified.	
	33.6.9 Interrupts in code flash programming mode was added.	
	33.6.10 Example of executing the commands to rewrite the flash memory areas was added.	
	33.6.11 Notes on self-programming was added.	
	37.2.1 Characteristics of the X1 and XT1 oscillators: Note 1 was modified.	Section 37 Electrical Characteristics TA = -40 to +105°C
	37.2.2 Characteristics of the On-chip Oscillators and Note 1 were modified.	
	37.3.2 Supply current characteristics (1) 30- to 64-pin package products with 96- to 128-Kbyte flash ROM was modified.	
	37.3.2 Supply current characteristics (2) Peripheral Functions (Common to all products) was modified.	
	37.3.2 Supply current characteristics (2) Peripheral Functions (Common to all products): Notes 15, 16, 17, and 18 were added.	
	37.4 AC Characteristics was modified.	
	37.5.1 Serial array unit (1) In UART communications with devices operating at same voltage levels: Note 1 was modified.	
	37.5.1 Serial array unit (2) In SPI (CSI) communications in the master mode with devices operating at same voltage levels with the internal SCKp clock (the ratings below are only applicable to CSI00) was modified.	

(15/36)

Edition	Description	Section
Rev.1.00	37.5.1 Serial array unit (3) In SPI (CSI) communications in the master mode with devices operating at same voltage levels with the internal SCKp clock was modified.	Section 37 Electrical Characteristics TA = -40 to +105°C
	37.5.1 Serial array unit (5) In simplified I2C communications with devices operating at same voltage levels was modified.	
	37.5.1 Serial array unit (6) In UART communications with devices operating at different voltage levels (1.8 V, 2.5 V, 3 V): Note 1 was modified.	
	37.5.1 Serial array unit (7) In SPI (CSI) communications in the master mode with devices operating at different voltage levels (2.5 V or 3 V) with the internal SCKp clock (the ratings below are only applicable to CSI00) was modified.	
	37.5.1 Serial array unit (8) In SPI (CSI) communications in the master mode with devices operating at different voltage levels (2.5 V or 3 V) with the internal SCKp clock (the ratings below are only applicable to CSI00) was modified.	
	37.5.1 Serial array unit (9) In SPI (CSI) communications in the slave mode with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V) with the external SCKp clock was modified.	
	37.5.1 Serial array unit (10) Simplified I2C communications with devices operating at different voltage levels (1.8 V, 2.5 V, or 3 V) was modified.	
	37.6.1 A/D converter characteristics (1) Normal modes 1 and 2 were modified.	
	37.6.1 A/D converter characteristics (2) Low-voltage modes 1 and 2 were modified.	
	37.6.1 A/D converter characteristics (3) When the internal reference voltage is selected as reference voltage (+) was modified.	
	37.6.1 A/D converter characteristics (4) When reference voltage (+) = Internal reference voltage (ADREFP1 = 1, ADREFP0 = 0), reference voltage (-) = AVREFM/ANI1 (ADREFM = 1), target pin: ANI0, ANI2 to ANI14, ANI16 to ANI26 was deleted.	
	37.6.6 LVD circuit characteristics (1) LVD0 Detection Voltage in the Reset Mode and Interrupt Mode was modified.	
	37.6.6 LVD circuit characteristics (2) LVD1 Detection Voltage of Reset Mode and Interrupt Mode and Note were modified.	
	37.8 Flash Memory Programming Characteristics (1) Code flash memory was added.	
	37.8 Flash Memory Programming Characteristics (2) Data flash memory was added.	
Rev.1.10	The module name for 3-wire SPI was changed to simplified SPI.	Overall
	The module name for SPI was changed to simplified SPI.	
	1.1 Features: The descriptions of Middle-speed on-chip oscillator were modified.	Section 1 Outline
	1.1 Features: The descriptions of Timers were modified.	
	Figure 1 - 1 Part Number, Memory Size, and Package of RL78/G23 was modified.	
	1.3.4 40-pin products: Figure was modified.	
	1.3.5 44-pin products: Figure was modified.	
	1.3.6 48-pin products: Note 2 was modified.	
	1.3.6 48-pin products: Remark 3 was added.	
	1.5 Block Diagram was modified.	
	1.6 Outline of Functions [30-, 32-, 36-, 40-, 44-, and 48-pin products]: The descriptions were modified.	
	1.6 Outline of Functions [52-, 64-, 80-, 100-, and 128-pin products]: The descriptions were modified.	
	2.1.1 30-pin products to 2.2.1 Functions for each product: Remarks were modified.	Section 2 Pin Functions
	2.1.5 44-pin products: The descriptions were modified.	
	2.2.1 Functions for each product: The descriptions were modified.	

(16/36)

Edition	Description	Section
Rev.1.10	2.2.2 Description of pin functions: The descriptions were modified.	Section 2 Pin Functions
	Table 2 - 3 Connections of Unused Pins: The descriptions were modified. Note was added.	
	Figure 2 - 25 Pin Block Diagram for Pin Type 8-31-1 was modified.	
	Figure 2 - 33 Pin Block Diagram for Pin Type 12-38-3: Caution was modified.	
	Table 3 - 8 List of Extended Special Function Registers (2nd SFRs) was modified.	Section 3 CPU Architecture
	Table 4 - 1 Port Configuration: The descriptions were modified.	Section 4 Port Functions
	4.2.6 Port 5: Note 3 was added.	
	4.3.1 Port mode registers (PMxx): The descriptions were modified.	
	Figure 4 - 2 Format of Port Registers (Pxx): The title was modified.	
	Figure 4 - 3 Format of Pull-up Resistor Option Registers (PUxx): The title was modified.	
	Figure 4 - 4 Format of Port Input Mode Registers (PIMxx): The title was modified.	
	Figure 4 - 5 Format of Port Output Mode Registers (POMxx): The title was modified.	
	Figure 4 - 6 Format of Port Digital Input Disable Registers (PDIDISxx): The title was modified.	
	Figure 4 - 7 Format of Port Mode Control A Registers (PMCAxx): The title was modified. Caution 4 was added.	
	Figure 4 - 8 Format of Port Mode Control T Registers (PMCTxx): The title was modified.	
	Figure 4 - 9 Format of Port Mode Control E Registers (PMCEx): The title was modified.	
	Figure 4 - 12 Format of Output Current Control Enable Register (CCDE) was modified. Note 1 was added.	
	Figure 4 - 15 Format of Port Function Output Enable Register 0 (PFOE0): The descriptions were modified.	
	Figure 4 - 16 Format of Port Function Output Enable Register 1 (PFOE1): Figure 4 - 16 Format of Port Function Output Enable Register 1 (PFOE1)	
	4.4.5 Handling different potential (1.8 V, 2.5 V, or 3 V) by using I/O buffers: The descriptions were modified.	
	4.5.2 Register settings for alternate function whose output function is not used (6) TxDAp = 1, CLKAp = 0 (settings when the UARTA is not used): The title and descriptions were changed.	
	4.5.4 Examples of register settings for port and alternate functions: Caution was added.	
	Table 4 - 7 Examples of Register and Output Latch Settings for Alternate Functions (30-Pin to 64-Pin Products with 96-Kbyte or 128-Kbyte Flash Memory): The descriptions were modified.	
	Table 4 - 8 Examples of Register and Output Latch Settings for Alternate Functions (Products with 192-Kbyte to 768-Kbyte Flash Memory and 80-Pin and 100-Pin Products with 128-Kbyte Flash Memory): The descriptions were modified.	
	Table 5 - 2 Features of Each Flash Operation Mode: The descriptions were modified.	Section 5 Operation State Control
	Figure 5 - 2 Format of Flash operating mode select register (FLMODE): Caution 7 was modified.	
	5.5.3 Details of LP (low-power main) mode: The descriptions were modified.	
	Figure 6 - 3 Format of System Clock Control Register (CKC): Cautions 4 and 5 were added.	Section 6 Clock Generator
	Figure 6 - 9 Format of Subsystem Clock Supply Mode Control Register (OSMC): The descriptions were modified.	
	7.3.2 Peripheral reset control register 0 (PRR0): The descriptions were modified.	Section 7 Timer Array Unit (TAU)

(17/36)

Edition	Description	Section
Rev.1.10	Figure 8 - 2 Format of Peripheral Enable Register 0 (PER0): Notes 1 and 2 were deleted, and Caution 3 was modified.	Section 8 Realtime Clock (RTC)
	9.1 Overview: The descriptions and figures are added.	Section 9 32-bit Interval Timer (TML32)
	Table 9 - 1 Specifications of 32-Bit Interval Timer Operations: The descriptions and remark were modified.	
	9.2.2 Peripheral reset control register 1 (PRR1): The descriptions were modified.	
	9.2.4 Interval timer compare registers 0n (ITLCMP0n) (n = 0, 1): The descriptions were modified.	
	9.2.5 Interval timer capture register 00 (ITLCAP00): The descriptions were modified.	
	Figure 9 - 7 Format of Interval Timer Control Register (ITLCTL0): The descriptions were modified.	
	Figure 9 - 8 Format of Interval Timer Clock Select Register 0 (ITLCSEL0): Remark was modified.	
	Figure 9 - 9 Format of Interval Timer Frequency Division Register 0 (ITLFDIV00): The descriptions were modified.	
	Figure 9 - 10 Format of Interval Timer Frequency Division Register 1 (ITLFDIV01): The descriptions were modified.	
	Figure 9 - 11 Format of Interval Timer Capture Control Register 0 (ITLCC0): The descriptions were modified.	
	9.2.11 Interval timer status register (ITLS0): The descriptions were modified.	
	Figure 9 - 12 Format of Interval Timer Status Register (ITLS0): Note was modified. Remark was deleted.	
	Table 9 - 2 Conditions for Setting the Status Flags in Each Timer Mode: The title and descriptions were modified.	
	9.2.12 Interval timer match detection mask register (ITLMKF0): The descriptions were modified.	
	Figure 9 - 13 Format of Interval Timer Match Detection Mask Register (ITLMKF0): Note was modified.	
	9.3.1 Counter mode settings: The descriptions were modified.	
	Table 9 - 3 Registers and Settings Used in 8-Bit Counter Mode: The descriptions were modified.	
	Table 9 - 5 Registers and Settings Used in 32-Bit Counter Mode: The descriptions were modified.	
	9.3.2 Capture mode settings: The descriptions were modified.	
	Table 9 - 6 Registers and Settings Used in 16-Bit Capture Mode: The title, descriptions, and Note were modified.	
	9.3.4 Capture operation: The descriptions were modified.	
	9.3.5 Interrupt: The title was added and the descriptions were modified.	
	Figure 9 - 16 Example of Clearing the Detected Flags was modified.	
	Figure 9 - 17 Procedure for Starting the 32-Bit Interval Timer was modified.	
	Figure 12 - 2 Format of Peripheral Enable Register 0 (PER0): Notes 1 and 2 were deleted.	Section 12 A/D Converter (ADC)
	12.3.2 Peripheral reset control register 0 (PRR0): The descriptions were modified.	
	Table 12 - 3 A/D Conversion Time Selection (3) When there is A/D power supply stabilization wait time Normal mode 1 and 2: The descriptions were modified.	
	Figure 12 - 10 Format of 12-bit/10-bit A/D Conversion Result Register (ADCRn):Caution 3 was added.	

(18/36)

Edition	Description	Section
Rev.1.10	Figure 12 - 12 Format of Analog Input Channel Specification Register (ADS): The descriptions were modified. Caution 9 was added.	Section 12 A/D Converter (ADC)
	Figure 12 - 19 Example of Software Select No-Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing: Caution was modified.	
	Figure 12 - 21 Example of Software Trigger No-Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing: Caution was modified.	
	Figure 12 - 23 Example of Software Trigger Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing: Caution was modified.	
	Figure 12 - 25 Example of Software Trigger Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing: Caution was modified.	
	Figure 12 - 26 Example of Hardware Trigger No-Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing: Caution was modified.	
	Figure 12 - 27 Example of Hardware Trigger No-Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing: Caution was modified.	
	Figure 12 - 28 Example of Hardware Trigger No-Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing: Caution was modified.	
	Figure 12 - 29 Example of Hardware Trigger No-Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing: Caution was modified.	
	Figure 12 - 31 Example of Hardware Trigger Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing: Cautions were modified.	
	Figure 12 - 33 Example of Hardware Trigger Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing: Caution was modified.	
	12.10 Points for Caution when the A/D Converter is to be Used: The descriptions were modified.	
	13.3.2 Peripheral reset control register 1 (PRR1): The descriptions were modified.	Section 13 D/A Converter (DAC)
	14.3.2 Peripheral reset control register 1 (PRR1): The descriptions were modified.	Section 14 Comparator (CMP)
	15.1.1 Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31): Caution was added.	Section 15 Serial Array Unit (SAU)
	Figure 15 - 5 Format of Peripheral Enable Register 0 (PER0): Note was deleted and Caution 2 was modified.	
	15.3.2 Peripheral reset control register 0 (PRR0): The descriptions were modified.	
	Figure 15 - 12 Format of Serial Status Register mn (SSRmn) (1/2): Caution was modified.	
	Figure 15 - 12 Format of Serial Status Register mn (SSRmn) (2/2): Caution was modified.	
	Figure 15 - 25 Peripheral Enable Register 0 (PER0) Setting When Stopping the Operation by Units: The description was modified, Note was deleted, and Caution 2 was modified.	
	15.5 Operation of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31) Communication: Caution was added.	
	Figure 15 - 35 Example of Contents of Registers for Master Reception of simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31) (b) Serial communication operation setting register mn (SCRmn): The descriptions were modified.	
	Table 15 - 3 Baud Rate Setting for UART Reception in SNOOZE Mode when Starting of the High-Speed On-Chip Oscillator is at Normal Speed (FWKUP = 0) was added.	
	Table 15 - 4 Baud Rate Setting for UART Reception in SNOOZE Mode when Starting of the High-Speed On-Chip Oscillator is at High Speed (FWKUP = 1) was added.	
	Figure 17 - 3 Format of Transmit Buffer Register (TXBAn): Remark was modified.	Section 17 Serial Interface UARTA (UARTA)
	Figure 17 - 6 Format of Operation Mode Setting Register 1 (ASIMAn1): Note was modified.	
	Figure 17 - 8 Format of Status Register (ASISAn): Cautions 2 and 3 were modified.	

(19/36)

Edition	Description	Section
Rev.1.10	Figure 17 - 13 Flowchart of Communication Procedure: Caution was modified.	Section 17 Serial Interface UARTA (UARTA)
	17.3.2 UART Mode, (4) Normal transmission (4): The descriptions were modified.	
	Figure 17 - 16 Interrupt Output Timing was modified.	
	17.3.2 UART Mode, (5) Continuous transmission: The descriptions and Caution 1 were modified.	
	Figure 17 - 18 Flow Example of Continuous Transmission Using Interrupt was modified.	
	17.3.2 UART Mode, (6) Normal reception: The descriptions were modified.	
	17.3.2 UART Mode, (7) Reception error: The descriptions were modified.	
	Figure 17 - 22 Various Interrupt Output Waveforms Depending on ISRMAn Setting was modified.	
	18.2.2 Peripheral reset control register 1 (PRR1): The descriptions were modified.	Section 18 Remote Control Signal Receiver (REMC)
	18.3.4.1 When using f_{SXP} as the REMC operating clock: The descriptions were modified.	
	Table 18 - 6 Interrupt Modes and Transition from SNOOZE Mode: The descriptions were modified, and Note was added.	
	Figure 23 - 2 Format of Memory Power Reduction Control Register (PSMCR): Caution 3 was added.	Section 23 Standby Function
	Figure 23 - 4 Procedure for Settings to Switch from Shutdown Mode to Normal Mode: Caution was added.	
	Table 23 - 1 Operating Statuses in HALT Mode (1): The descriptions were added.	
	Table 23 - 2 Operating Statuses in HALT Mode (2): The descriptions were added.	
	Table 23 - 3 Operating Statuses in STOP Mode (2/2): The descriptions were added.	
	Table 23 - 4 Operating Statuses in SNOOZE Mode (2/2): The descriptions were added.	
	24.2.1 Reset control flag register (RESF): The descriptions were modified.	Section 24 Reset Function
	Figure 26 - 11Delays from the Time an LVD0 or LVD1 Reset Source Condition is Satisfied until an LVD0 or LVD1 Reset has been Generated and Deasserted: The descriptions were modified.	Section 26 Voltage Detector (LVD)
	Figure 28 - 3 Procedure for Using the True Random Number Generator to Generate a Random Number Seed was modified.	Section 28 Security Functions
	28.2.2 Setting of flash read protection: The descriptions were modified.	
	Table 28 - 1 Settings for Flash Read Protection and their Functions: The descriptions were modified.	
	Table 28 - 2 Method of Setting Flash Read Protection: The descriptions, Note, and Caution 4 were modified.	
	29.3.2 Peripheral reset control register 1 (PRR1): The descriptions were modified.	Section 29 SNOOZE Mode Sequencer (SMS)
	Table 29 - 1 Correspondences between the Memory Addresses of the SMSIp Registers and Values of the SMSCV0 to SMSCV4 Bits: Caution 1 was modified.	
	Section 30 Capacitive Sensing Unit (CTSU2L): The descriptions were modified.	Section 30 Capacitive Sensing Unit (CTSU2L)
	Table 30 - 1 CTSU Function: The descriptions were modified.	
	Figure 30 - 3 CTSU Block Diagram was modified.	
	Figure 30 - 4 Sensor Drive Pulse Output Clock Configuration was modified.	
	Table 30 - 2 External Pins Used in CTSU: The descriptions were modified.	
	Figure 30 - 6 Format of Peripheral Reset Control Register 1 (PRR1): The descriptions were modified.	
	Figure 30 - 7 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH): Address and descriptions were modified.	

(20/36)

Edition	Description	Section
Rev.1.10	Figure 30 - 8 Format of CTSU Control Registers BL and BH (CTSUCRBL, CTSUCRBH): Address and descriptions were modified.	Section 30 Capacitive Sensing Unit (CTSU2L)
	Figure 30 - 9 Format of CTSU Measurement Channel Registers L and H (CTSUMCHL, CTSUMCHH): Address and descriptions were modified.	
	Figure 30 - 10 Format of CTSU Channel Enable Control Registers AL, AH, BL, and BH (CTSUCHACAL, CTSUCHACAH, CTSUCHACBL, CTSUCHACBH): Address was modified.	
	Figure 30 - 11 Format of CTSU Channel Transmit/Receive Control Registers AL, AH, BL, and BH (CTSUCHTRCAL, CTSUCHTRCAH, CTSUCHTRCBL, CTSUCHTRCBH): Address and Caution 3 were modified.	
	Figure 30 - 12 Format of CTSU Status Register L (CTSUSRL): The descriptions were modified.	
	Figure 30 - 13 Format of CTSU Sensor Offset Registers 0 and 1 (CTSUSO0, CTSUSO1): Address and descriptions were modified.	
	Figure 30 - 14 Format of CTSU Sensor Counter Registers L and H (CTSUSC, CTSUUC): Address was modified.	
	Figure 30 - 15 Format of CTSU Calibration Registers L and H (CTSUDBGR0, CTSUDBGR1): Address and descriptions were modified.	
	Figure 30 - 16 Format of CTSU Sensor Unit Clock Control Registers AL, AH, BL, and BH (CTSUSUCLK0, CTSUSUCLK1, CTSUSUCLK2, CTSUSUCLK3): Address and descriptions were modified.	
	Figure 30 - 17 Format of CTSU Trimming Registers AL and AH (CTSUTRIM0, CTSUTRIM1): Address was modified.	
	Figure 30 - 18 Format of CTSU Trimming Registers BL and BH (CTSUTRIM2, CTSUTRIM3): Address was modified.	
	30.3 Usage Notes of the Capacitive Sensing Unit was added.	
	31.1 Overview: The descriptions were modified.	Section 31 Regulator
	Figure 32 - 2 Format of User Option Byte (000C1H or 040C1H): The descriptions were modified.	Section 32 Option Bytes
	Figure 32 - 3 Format of User Option Byte (000C2H or 040C2H): The descriptions were modified.	
	Section 33 Flash Memory: The description was modified.	Section 33 Flash Memory
	Figure 33 - 2 Communications with Dedicated Flash Memory Programmer was modified.	
	Table 33 - 2 Pin Connection was modified.	
	33.2.1 Programming environment: The description was deleted.	
	33.3.5 X1 and X2 pins was deleted.	
	33.6 Self-Programming: The description and cautions 2 to 4 were modified.	
	33.6.1 Self-programming procedure: The description was modified.	
	Figure 33 - 8 Flow of Self-Programming (Rewriting the Flash Memory) was modified.	
	33.6.2 Registers to control the flash memory: The description was modified.	
	Table 33 - 10 Conditions That Enable Write Access to the Flash Memory was deleted.	
	Figure 33 - 9 Format of Flash Address Pointer Registers H and L (FLAPH, FLAPL): Cautions 1 and 2 were modified.	
	Figure 33 - 10 Format of Flash End Address Pointer Registers H and L (FLSEDH, FLSEDL): Cautions 1 to 3 were modified.	
	Table 33 - 10 Method of Setting the FLAPH, FLAPL, FLSEDH, and FLSEDL Registers: The description and note were modified, and cautions 1 to 3 were deleted.	
	33.6.2.3 Flash write buffer registers H and L (FLWH, FLWL): The description was modified.	

(21/36)

Edition	Description	Section
Rev.1.10	Figure 33 - 11 Format of Flash Write Buffer Registers H and L (FLWH, FLWL): Cautions 1 to 3 were modified.	Section 33 Flash Memory
	33.6.2.4 Flash protect command register (PFCMD): The description was modified.	
	33.6.2.5 Flash status register (PFS): The description was modified.	
	33.6.2.6 Flash programming mode control register (FLPMC): The description was modified.	
	Figure 33 - 14 Format of Flash Programming Mode Control Register (FLPMC): Caution was added.	
	33.6.2.7 Flash area selection register (FLARS): The description was modified.	
	Figure 33 - 15 Format of Flash Area Selection Register (FLARS): The description was modified and caution was added.	
	Figure 33 - 16 Format of Flash Memory Sequencer Initial Setting Register (FSSET): The description, note, and cautions were modified.	
	Table 33 - 11 Correspondence between the Operating Frequency of the Flash Memory Sequencer and the Setting of the FSET4 to FSET0 Bits: The title and description were modified, and caution was added.	
	33.6.2.9 Flash memory sequencer control register (FSSQ): The description was modified.	
	Figure 33 - 17 Format of Flash Memory Sequencer Control Register (FSSQ): The description and caution were modified, notes 1 to 4 were added, and remark was deleted.	
	33.6.2.10 Flash extra area sequencer control register (FSSE): The title and description were modified.	
	Figure 33 - 18 Format of Flash Extra Area Sequencer Control Register (FSSE): The title, description, and cautions 1 and 3 were modified, and note was added.	
	33.6.2.11 Flash registers initialization register (FLRST): The description was modified.	
	Figure 33 - 19 Format of Flash Registers Initialization Register (FLRST): The description and cautions 1 and 2 were modified, and note was deleted.	
	Figure 33 - 20 Format of Flash Memory Sequencer Status Registers H and L (FSASTH, FSASTL): The description and value after reset were modified.	
	33.6.2.13 Flash security flag monitoring register (FLSEC): The description was modified.	
	Figure 33 - 21 Format of Flash Security Flag Monitoring Register (FLSEC): The description was modified.	
	Figure 33 - 22 Format of Flash FSW Monitoring Register E (FLFSWE): The description and note were modified.	
	Figure 33 - 23 Format of Flash FSW Monitoring Register S (FLFSWS): The description was modified.	
	33.6.2.16 Data flash control register (DFLCTL): The description was modified.	
	Figure 33 - 24 Format of Data Flash Control Register (DFLCTL): The description was modified.	
	33.6.3 Setting the flash memory control mode: The description was modified.	
	33.6.3.1 Procedure for executing the specific sequence: The description was modified.	
	33.6.3.4 Procedure for entry to the non-programmable mode: The description and note were modified	
	33.6.4 Initializing the registers for use with the flash memory sequencer: The title and description were modified.	
	33.6.5 Setting the operating frequency of the flash memory sequencer: Caution was modified.	
	33.6.6 Rewriting the flash memory: The title was modified.	
	33.6.6.2 Selecting the area to be rewritten: The description was modified.	

(22/36)

Edition	Description	Section
Rev.1.10	33.6.6.3 Commands for use with the code/data flash memory area sequencer: The description was modified.	Section 33 Flash Memory
	33.6.6.4 Operations for rewriting the code flash memory area: The description was modified and notes were added.	
	33.6.6.5 Operations for rewriting the data flash memory area: The description was modified and note was added.	
	33.6.6.6 Commands for use with the extra area sequencer: The description was modified.	
	33.6.6.7 Operations for rewriting the extra area: The description was modified.	
	33.6.6.8 Data to be set for the commands for use with the extra area sequencer: Note was added and the description and cautions were added and modified.	
	33.6.6.9 Procedures for checking completion of the commands for use with the flash memory sequencer in the respective areas: The description was modified.	
	33.6.6.10 Procedure for forcibly terminating a command for use with the code/data flash memory area sequencer: The description was modified.	
	33.6.7.3 Operation to change the interrupt branch destination: The description was modified.	
	Figure 33 - 29 Flow of Executing the Commands to Rewrite the Extra Area was modified.	
	33.6.9 Notes on self-programming: The description was modified.	
	33.7 Boot Swap Function: The heading level, description, and note were modified.	
	33.8 Flash Shield Window Function: The heading level was modified.	
	Figure 33 - 32 Flash Shield Window Setting Example (Target Devices: R7F100GLG, Start Block Number: 04H, End Block Number: 06H, FSWC: 1) was modified. Caution 1 was modified.	
	Table 33 - 12 Relationship between Flash Shield Window Function Setting/Change Methods and Commands: The description was modified.	
	33.9 Security Settings: The description was modified and note was added.	
	Table 33 - 13 Relationship between Enabling the Security Function and Commands was modified.	
	Table 33 - 14 Setting Security in Each Programming Mode was modified.	
	33.10.2 Procedure for accessing the data flash memory: Caution 4 was added.	
	34.1 Connection between the E2 or E2 Lite On-chip Debugging Emulator and RL78/G23: The descriptions were added.	Section 34 On-chip Debugging
	34.2 Connection between the External Device that Incorporates UART and RL78/G23: The descriptions and Note 3 were modified.	
	34.3 Security Settings for On-Chip Debugging: The title and descriptions were modified.	
	Table 34 - 1 On-Chip Debug Security ID: The descriptions and Note 1 were added. Note 2 was modified.	
	37.1 Absolute Maximum Ratings: Note was modified.	Section 37 Electrical Characteristics T_A = -40 to +105°C
	37.2.1 Characteristics of the X1 and XT1 oscillators: Condition was modified.	
	37.3.1 Pin characteristics: Notes 4 to 6 were modified.	
	37.3.1 Pin characteristics: Notes 3, 5, and 6 were modified.	
	37.3.2 Supply current characteristics, (1) 30- to 64-pin package products with 96- to 128-Kbyte flash ROM: The descriptions in the tables were modified.	
	37.3.2 Supply current characteristics, (2) 30- to 64-pin package products with 192- to 256-Kbyte flash ROM and 80-pin package product with 128- to 256-Kbyte flash ROM was added.	

(23/36)

Edition	Description	Section
Rev.1.10	37.3.2 Supply current characteristics, (3) 44- to 80-pin package products with 384- to 768-Kbyte flash ROM and 100- to 128-pin package products was added.	Section 37 Electrical Characteristics T_A = -40 to +105°C
	37.3.2 Supply current characteristics, (4) Peripheral Functions (Common to all products): The descriptions in the tables were added. Notes 13, 14, and 16 were modified. Note 19 was modified.	
	37.5.2 Serial interface UARTA: The table was modified.	
	37.6.1 A/D converter characteristics, (1) Normal modes 1 and 2: The descriptions in the table were modified.	
	37.6.1 A/D converter characteristics, (2) Low-voltage modes 1 and 2: The descriptions in the table were modified.	
	37.6.1 A/D converter characteristics, (3) When the internal reference voltage is selected as reference voltage (+): The descriptions in the table were modified.	
	37.6.4 Comparator characteristics: The descriptions in the table were modified.	
	37.6.6 LVD circuit characteristics, (2) LVD1 Detection Voltage of Reset Mode and Interrupt Mode : The table was modified.	
	37.8 Flash Memory Programming Characteristics, (2) Data flash memory: The descriptions in the table were modified.	
	38.4 40-Pin Products: The figure was added.	Section 38 Package Drawings
	38.6 48-Pin Products: The figure was added.	
Rev.1.20	Table 1 - 1 List of Ordering Part Numbers was modified.	Section 1 Outline
	Table 1 - 2 Multiplexed Pin Functions of the 30-pin Products was added.	
	Table 1 - 3 Multiplexed Pin Functions of the 32-pin Products was added.	
	Table 1 - 4 Multiplexed Pin Functions of the 36-pin Products was added.	
	Table 1 - 5 Multiplexed Pin Functions of the 40-pin Products was added.	
	Table 1 - 6 Multiplexed Pin Functions of the 44-pin Products was added.	
	Table 1 - 7 Multiplexed Pin Functions of the 48-pin Products was added.	
	Table 1 - 8 Multiplexed Pin Functions of the 52-pin Products was added.	
	Table 1 - 9 Multiplexed Pin Functions of the 64-pin Products was added.	
	Table 1 - 10 Multiplexed Pin Functions 2 of the 64-pin Products was added.	
	Table 1 - 11 Multiplexed Pin Functions of the 80-pin Products was added.	
	Table 1 - 12 Multiplexed Pin Functions of the 100-pin Products was added.	
	Table 1 - 13 Multiplexed Pin Functions 2 of the 100-pin Products was added.	
	Table 1 - 14 Multiplexed Pin Functions of the 128-pin Products was added.	
	1.6 Outline of Functions [30-, 32-, 36-, 40-, 44-, and 48-pin products] was modified.	
	1.6 Outline of Functions [52-, 64-, 80-, 100-, and 128-pin products] was modified.	
	2.2.2 Description of pin functions was modified.	Section 2 Pin Functions
	2.2.3 VBAT Pin was added.	
	Figure 2 - 33 Pin Block Diagram for Pin Type 12-38-1: Caution was deleted.	
	Figure 2 - 34 Pin Block Diagram for Pin Type 12-38-2: Caution was modified.	
	Figure 3 - 1 Memory Map (R7F100GxF (x = A, B, C, E, F, G, J, L)) was modified.	Section 3 CPU Architecture
	Figure 3 - 1 Memory Map (R7F100GxF (x = A, B, C, E, F, G, J, L)): Note 3 was modified. Note 4 was added.	
	Figure 3 - 2 Memory Map (R7F100GxG (x = A, B, C, E, F, G, J, L, M, P)) was modified.	

(24/36)

Edition	Description	Section
Rev.1.20	Figure 3 - 2 Memory Map (R7F100GxG (x = A, B, C, E, F, G, J, L, M, P)): Note 3 was modified. Note 4 was added.	Section 3 CPU Architecture
	Figure 3 - 3 Memory Map (R7F100GxH (x = A, B, C, E, F, G, J, L, M, P)) was modified.	
	Figure 3 - 3 Memory Map (R7F100GxH (x = A, B, C, E, F, G, J, L, M, P)): Note 3 was modified. Note 4 was added.	
	Figure 3 - 4 Memory Map (R7F100GxJ (x = A, B, C, E, F, G, J, L, M, P, S)) was modified.	
	Figure 3 - 4 Memory Map (R7F100GxJ (x = A, B, C, E, F, G, J, L, M, P, S)): Note 3 was modified. Note 4 was added.	
	Figure 3 - 5 Memory Map (R7F100GxK (x = F, G, J, L, M, P, S)) was modified.	
	Figure 3 - 5 Memory Map (R7F100GxK (x = F, G, J, L, M, P, S)): Note 3 was modified. Note 4 was added.	
	Figure 3 - 6 Memory Map (R7F100GxL (x = F, G, J, L, M, P, S)) was modified.	
	Figure 3 - 6 Memory Map (R7F100GxL (x = F, G, J, L, M, P, S)): Note 3 was modified. Note 4 was added.	
	Figure 3 - 7 Memory Map (R7F100GxN (x = F, G, J, L, M, P, S)) was modified.	
	Figure 3 - 7 Memory Map (R7F100GxN (x = F, G, J, L, M, P, S)): Note 3 was modified. Note 4 was added.	
	3.1.5 Extended special function register (2nd SFR: 2nd special function register) area: Caution 2 was modified.	
	Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) was modified.	
	4.4.5 Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching I/O buffers: The description was added.	Section 4 Port Functions
	4.4.5 Communications with devices operating at a different voltage (1.8 V, 2.5 V, or 3 V) by switching I/O buffers, 2.: The description was added.	
	Table 4 - 7 Examples of Register and Output Latch Settings for Alternate Functions (30-Pin to 64-Pin Products with 96-Kbyte or 128-Kbyte Flash Memory) was modified.	
	Table 4 - 8 Examples of Register and Output Latch Settings for Alternate Functions (Products with 192-Kbyte to 768-Kbyte Flash Memory and 80-Pin and 100-Pin Products with 128-Kbyte Flash Memory) was modified.	
	Figure 6 - 7 Format of Peripheral Enable Register 0 (PER0): Notes 1 and 2 were added.	Section 6 Clock Generator
	Figure 6 - 8 Format of Peripheral Enable Register 1 (PER1): Notes 1 and 2 were added.	
	6.3.7 Subsystem clock supply mode control register (OSMC): The description was modified.	
	6.6.2 Example of setting the X1 oscillator clock: Caution was modified.	
	Figure 7 - 11 Format of Peripheral Enable Register 0 (PER0): Notes 1 and 2 were added and Caution 2 was modified.	Section 7 Timer Array Unit (TAU)
	Figure 7 - 12 Format of Peripheral Reset Control Register 0 (PRR0): Caution was deleted. Notes 1 and 2 and Cautions 1 and 2 were added.	
	7.3.17 Registers for controlling the port functions multiplexed with the inputs and outputs of timers was modified.	
	Figure 8 - 2 Format of Peripheral Enable Register 0 (PER0): Notes 1 and 2 were added.	Section 8 Realtime Clock (RTC)
	8.3.2 Subsystem clock supply mode control register (OSMC): The descriptions were modified.	
	Figure 8 - 4 Format of Realtime Clock Control Register 0 (RTCC0): Caution 4 was added.	
	Figure 8 - 5 Format of Realtime Clock Control Register 1 (RTCC1): Caution 2 was added.	
	Figure 8 - 6 Format of Second Count Register (SEC): Caution 2 was added.	
	Figure 8 - 7 Format of Minute Count Register (MIN): Caution 2 was added.	

(25/36)

Edition	Description	Section
Rev.1.20	Figure 8 - 8 Format of Hour Count Register (HOUR): Caution 3 was added.	Section 8 Realtime Clock (RTC)
	Figure 8 - 9 Format of Day Count Register (DAY): Caution 2 was added.	
	Figure 8 - 10 Format of Day-of-Week Count Register (WEEK): Caution 3 was added.	
	Figure 8 - 11 Format of Month Count Register (MONTH): Caution 2 was added.	
	Figure 8 - 12 Format of Year Count Register (YEAR): Caution 2 was added.	
	Figure 8 - 13 Format of Time Error Correction Register (SUBCUD): Caution 2 was added.	
	Figure 8 - 14 Format of Alarm Minute Register (ALARMWM): Caution was added.	
	Figure 8 - 15 Format of Alarm Hour Register (ALARMWH): Caution 2 was added.	
	Figure 8 - 16 Format of Alarm Day-of-Week Register (ALARMWW): Caution was added.	
	8.3.16 Registers for controlling the port functions multiplexed with the realtime clock output was modified.	
	9.1 Overview: 3. 16-bit capture mode was modified.	Section 9 32-bit Interval Timer (TML32)
	Figure 9 - 2 Format of Peripheral Enable Register 1 (PER1): Notes 1 and 2, and Caution were added.	
	Figure 9 - 3 Format of Peripheral Reset Control Register 1 (PRR1): Note and Cautions 1 and 2 were added.	
	Figure 9 - 11 Format of Interval Timer Capture Control Register 0 (ITLCC0) was modified and Note 8 was added.	
	Figure 9 - 15 Example of Capture Operation: The description was modified.	
	10.3.2 Registers for controlling the port functions multiplexed with the clock or buzzer outputs: The descriptions were modified.	Section 10 Clock Output/Buzzer Output Controller (PCLBUZ)
	Table 11 - 4 Setting Window Open Period of Watchdog Timer: Remark was modified.	Section 11 Watchdog Timer (WDT)
	Figure 12 - 3 Format of Peripheral Reset Control Register 0 (PRR0): Notes 1 and 2, and Cautions 1 and 2 were added.	Section 12 A/D Converter (ADC)
	Table 12 - 3 A/D Conversion Time Selection (1/8) to (8/8): Caution 1 was modified.	
	Figure 12 - 10 Format of 12-bit/10-bit A/D Conversion Result Register (ADCRn): Caution 3 was modified.	
	12.3.12 Registers for controlling the port functions multiplexed with the analog inputs of the A/D converter was modified.	
	Figure 12 - 42 Block Diagram When Using SNOOZE Mode in Hardware Trigger Wait Mode: Note was modified.	
	The description at the beginning of the section was deleted.	Section 13 D/A Converter (DAC)
	Figure 13 - 2 Format of Peripheral Enable Register 1 (PER1): Notes 1 and 2 were added, and Caution 2 was modified.	
	Figure 13 - 3 Format of Peripheral Reset Control Register 1 (PRR1): Note was added, and Caution 2 was modified.	
	13.3.6 Registers for controlling the port functions multiplexed with the analog outputs of the D/A converter was modified.	
	Figure 14 - 2 Format of Peripheral Enable Register 1 (PER1): Notes 1 and 2, and Caution 2 were added.	Section 14 Comparator (CMP)
	Figure 14 - 3 Format of Peripheral Reset Control Register 1 (PRR1): Note was added, and Caution 2 was modified.	
	14.3.6 Registers for controlling the port functions multiplexed with the analog inputs and outputs of the comparator was modified.	

(26/36)

Edition	Description	Section
Rev.1.20	Figure 15 - 5 Format of Peripheral Enable Register 0 (PER0): Notes 1 and 2 were added, and Caution 2 was modified.	Section 15 Serial Array Unit (SAU)
	Figure 15 - 6 Format of Peripheral Reset Control Register 0 (PRR0): Notes 1 and 2, and Caution 2 were added.	
	15.3.18 Registers for controlling the port functions multiplexed with the inputs and outputs of the serial array unit was modified.	
	Figure 15 - 24 Peripheral Enable Register 0 (PER0) Setting When Stopping the Operation by Units: Note 1 was added.	
	Figure 15 - 71 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) was modified.	
	15.6.3 SNOOZE Mode Function: The description and Caution 4 were modified.	
	Figure 16 - 5 Format of Peripheral Enable Register 0 (PER0): Notes 1 and 2 were added, and Cautions 1 and 2 were modified.	Section 16 Serial Interface IICA (IICA)
	Figure 16 - 6 Format of Peripheral Reset Control Register 0 (PRR0): Notes 1 and 2, and Cautions 1 and 2 were added.	
	16.3.9 Registers for controlling the port functions multiplexed with the IICA inputs and outputs was modified.	
	Figure 17 - 2 Format of Peripheral Enable Register 1 (PER1): Notes 1 and 2 were added, and Caution 2 was modified.	Section 17 Serial Interface UARTA (UARTA)
	Figure 17 - 10 Format of UARTA0 Clock Select Register (UTA0CK): Note was added.	
	Figure 17 - 11 Format of UARTA1 Clock Select Register (UTA1CK): Note was added.	
	17.2.12 Registers for controlling the port functions multiplexed with the inputs and outputs of the UARTA serial interface was added.	
	Figure 17 - 21 Timing of UART Receive Operation was modified.	
	Figure 18 - 2 Format of Peripheral Enable Register 1 (PER1): Notes 1 and 2 were added. Caution was modified.	Section 18 Remote Control Signal Receiver (REMC)
	Figure 18 - 3 Format of Peripheral Reset Control Register 1 (PRR1): Note was added. Caution 2 was modified.	
	18.2.23 Registers for controlling the port functions multiplexed with the remote control signal input was added.	
	18.3.4 Operating Clock: The descriptions were modified.	
	18.3.13 SNOOZE Mode Function: Remark was modified.	
	Figure 19 - 5 Format of Peripheral Enable Register 1 (PER1): Notes 1 and 2, and Caution were modified.	Section 19 Data Transfer Controller (DTC)
	Table 20 - 1 Connections in Logic Cell Block L1 (1) and Note 1 were modified.	Section 20 Logic and Event Link Controller (ELCL)
	Table 20 - 3 Connections in Logic Cell Block L2 (1): Note 1 was modified.	
	Table 20 - 5 Connections in Logic Cell Block L3 (1): Note 1 was modified.	
	Figure 20 - 5 Format of Input Signal Select Registers n (ELISELn) (n = 0 to 11) (2/3) and (3/3), and Note 2 were modified.	
	Table 20 - 7 Bits in Output Signal Enable Register and Event-Receiving Peripheral Functions was modified.	
	20.3.26 Registers for controlling the port functions multiplexed with the I/O pins of the logic and event link controller was added.	
	Figure 20 - 32 Example of ELCL Settings (OR Circuit) was modified.	
	Table 20 - 8 Example of ELCL Register Settings (OR Circuit) was modified.	
	Figure 20 - 33 Example of ELCL Settings (AND Circuit) was modified.	
	Table 20 - 9 Example of ELCL Register Settings (AND Circuit) was modified.	

(27/36)

Edition	Description	Section
Rev.1.20	21.3.6 Registers for controlling the port functions multiplexed with the interrupt inputs was added.	Section 21 Interrupt Functions
	Figure 22 - 3 Format of Key Return Mode Register 0 (KRM0): Caution 4 was added.	Section 22 Key Interrupt Function
	22.3.4 Registers for controlling the port functions multiplexed with the key interrupt inputs was modified.	
	Figure 23 - 2 Format of Memory Power Reduction Control Register (PSMCR): Note 4 was added.	Section 23 Standby Function
	Figure 23 - 7 STOP Mode Release by Interrupt Request Generation (1/3): Note 2 was modified.	
	Figure 23 - 7 STOP Mode Release by Interrupt Request Generation (2/3): Note 2 was modified.	
	23.3.3 SNOOZE mode: 1. SNOOZE mode setting and operating states was modified.	
	Figure 24 - 7 Format of Peripheral Reset Control Register 0 (PRR0) was modified, and Notes 1 and 2, and Caution 1 were added.	Section 24 Reset Function
	Figure 24 - 8 Format of Peripheral Reset Control Register 1 (PRR1) was modified, and Note and Caution 1 were added.	
	Figure 27 - 1 Format of Flash Memory CRC Control Register (CRC0CTL): Caution was added.	Section 27 Safety Functions
	Figure 27 - 14 Illicit Access Areas was modified.	
	Figure 27 - 16 Format of Guard Register of IAWCTL Register (GIAWCTL) was modified.	
	27.3.10 Testing of the A/D converter: The description was modified.	
	Figure 27 - 22 Format of Port Mode Select Register (PMS): Caution 3 was added.	
	27.3.12 UART loopback: The description was added.	
	Figure 27 - 23 Format of UART Loopback Select Register (ULBS) was modified, and Notes 1 and 2 were deleted.	
	28.2.1 Function of flash read protection: The description was modified.	Section 28 Security Functions
	28.2.2 Setting of flash read protection: The description was modified.	
	Table 28 - 1 Settings for Flash Read Protection and their Functions was modified.	
	Table 28 - 2 Method of Setting Flash Read Protection: Caution 5 was added.	
	Figure 28 - 4 Examples of Setting Read-Access Disabled Areas for the Application of Flash Read Protection: The description was added.	
	Figure 29 - 1 Block Diagram of the SNOOZE Mode Sequencer was modified.	Section 29 SNOOZE Mode Sequencer (SMS)
	Figure 29 - 2 Format of Peripheral Enable Register 1 (PER1): Notes 1 and 2, and Caution 2 were added.	
	Figure 29 - 3 Format of Peripheral Reset Control Register 1 (PRR1): Note was modified, and Cautions 1 and 2 were added.	
	Table 29 - 2 Types of Processing Specified by the SMSIp Registers: Note 3 was modified.	
	29.5 Commands for Use in Processing by the Sequencer: The description was modified.	
	29.5.2 8-bit data transfer 2: The description was modified.	
	29.5.12 Branch 1 (SCY = 1): The description was modified.	
	29.5.13 Branch 2 (SCY = 0): The description was modified.	
	29.5.14 Branch 3 (SCY = 1): The description was modified.	
	29.5.15 Branch 4 (SZ = 0): The description was modified.	
	29.6 Operation in Standby Modes: Caution was added.	

(28/36)

Edition	Description	Section
Rev.1.20	Table 30 - 2 External Pins Used in CTSU was modified.	Section 30 Capacitive Sensing Unit (CTSU2L)
	Figure 30 - 5 Format of Peripheral Enable Register 1 (PER1): Notes 1 and 2 and Caution were added.	
	Figure 30 - 6 Format of Peripheral Reset Control Register 1 (PRR1): Note and Cautions 1 and 2 were added.	
	30.2.3 CTSU control registers AL and AH (CTSUCRAL, CTSUCRAH): The description was modified.	
	Figure 30 - 7 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) was modified.	
	30.2.4 CTSU control registers BL and BH (CTSUCRBL, CTSUCRBH): The description was modified.	
	30.2.5 CTSU measurement channel registers L and H (CTSUMCHL, CTSUMCHH): The description was modified.	
	30.2.6 CTSU channel enable control registers AL, AH, BL, and BH (CTSUCHACAL, CTSUCHACAH, CTSUCHACBL, CTSUCHACBH): The description was modified.	
	30.2.7 CTSU channel transmit/receive control registers AL, AH, BL, and BH (CTSUCHTRCAL, CTSUCHTRCAH, CTSUCHTRCBL, CTSUCHTRCBH): The description was modified.	
	30.2.8 CTSU status register L (CTSUSRL): The description was modified.	
	30.2.9 CTSU sensor offset registers 0 and 1 (CTSUSO0, CTSUSO1): The description was added.	
	30.2.10 CTSU sensor counter registers L and H (CTSUSC, CTSUUC): The description was added.	
	30.2.11 CTSU calibration registers L and H (CTSUDBGR0, CTSUDBGR1): The description was added.	
	30.2.12 CTSU sensor unit clock control registers AL, AH, BL, and BH (CTSUSUCLK0, CTSUSUCLK1, CTSUSUCLK2, CTSUSUCLK3): The description was added.	
	30.2.13 CTSU trimming registers AL and AH (CTSUTRIM0, CTSUTRIM1): The description was modified.	
	30.2.14 CTSU trimming registers BL and BH (CTSUTRIM2, CTSUTRIM3): The description was modified.	
	30.2.15 Registers for controlling the port functions multiplexed with the inputs and outputs of the capacitive sensor was added.	
	33.6 Self-Programming was modified.	Section 33 Flash Memory
	Figure 33 - 9 Format of Flash Address Pointer Registers H and L (FLAPH, FLAPL): Caution 2 was modified.	
	Figure 33 - 10 Format of Flash End Address Pointer Registers H and L (FLSEDH, FLSEDL): Caution 2 was modified.	
	Figure 33 - 11 Format of Flash Write Buffer Registers H and L (FLWH, FLWL): Caution 2 was modified.	
	Figure 33 - 14 Format of Flash Programming Mode Control Register (FLPMC): Caution was modified.	
	Figure 33 - 17 Format of Flash Memory Sequencer Control Register (FSSQ) was modified.	
	Figure 33 - 18 Format of Flash Extra Area Sequencer Control Register (FSSE) and Caution 3 were modified.	
	Figure 33 - 19 Format of Flash Registers Initialization Register (FLRST), and Caution 1 was modified.	
	Figure 33 - 20 Format of Flash Memory Sequencer Status Registers H and L (FSASTH, FSASTL) was modified.	

(29/36)

Edition	Description	Section
Rev.1.20	33.6.2.16 Data flash control register (DFLCTL): The description was modified.	Section 33 Flash Memory
	Figure 33 - 24 Format of Data Flash Control Register (DFLCTL) was modified.	
	33.6.3 Setting the flash memory control mode: The description and Caution were modified.	
	33.6.4 Initializing the registers for use with the flash memory sequencer: The description was modified.	
	Figure 33 - 29 Flow of Executing the Commands to Rewrite the Extra Area was modified.	
	33.6.9 Notes on self-programming: The description of 2. was modified.	
	33.7 Boot Swap Function: The description and Note were modified.	
	33.9 Security Settings: The description was modified.	
	33.10.1 Overview of the data flash memory: The description and Caution 2 were modified.	
	33.10.2 Procedure for accessing the data flash memory: Caution 3 was modified.	
	Figure 34 - 2 Memory Spaces Allocated for Use by the Monitor Program for Debugging: Figure was modified and note 5 was added.	Section 34 On-chip Debugging
	37 The section title was modified, and the description and Cautions 1 to 4 were added to the beginning of the section.	Section 37 Electrical Characteristics
	37.1 Absolute Maximum Ratings was modified.	
	37.2.1 Characteristics of the X1 oscillator was modified.	
	37.2.2 Characteristics of the XT1 oscillator was added.	
	37.3.1 Pin characteristics, Note, and Cautions were modified.	
	37.3.2 Supply current characteristics: 1. Notes 3 to 5 were modified.	
	37.3.2 Supply current characteristics: 2. Notes 3 to 5 were modified.	
	37.3.2 Supply current characteristics: 3. Notes 3 to 5 were modified.	
	37.3.2 Supply current characteristics: 4. Peripheral Functions (Common to all products) was modified, and Note 20 was added.	
	37.4 AC Characteristics was modified.	
	37.6.1 A/D converter characteristics: 2. Low-voltage modes 1 and 2 was modified, and Note 7 was added.	
	38.3 36-pin Products was modified.	Section 38 Package Drawings
	38.8 64-pin Products was modified.	
Rev.1.21	The bookmarks were modified.	Overall
	Table 1 - 1 List of Ordering Part Numbers was modified.	Section 1 Outline
Rev.1.30	Figure 1 - 1 Part Number, Memory Size, and Package of RL78/G23 was modified.	Section 1 Outline
	Table 1 - 1 List of Ordering Part Numbers was modified.	
	Figure 2 - 14 Pin Block Diagram for Pin Type 7-2-1 was modified.	Section 2 Pin Functions
	4.5.2 Register settings for alternate function whose output function is not used: The description of 6. was modified.	Section 4 Port Functions
	Figure 8 - 5 Format of Realtime Clock Control Register 1 (RTCC1) was modified.	Section 8 Realtime Clock (RTC)
	Figure 8 - 19 Procedure for Reading Realtime Clock: Caution was modified.	
	Figure 8 - 20 Procedure for Writing Realtime Clock: Caution 1 was modified.	
	Figure 12 - 4 Format of A/D Converter Mode Register 0 (ADM0): Caution 4 was added.	Section 12 A/D Converter (ADC)
	Figure 12 - 5 Timing Chart When A/D Voltage Comparator Is Used: Caution 4 was modified.	
	Table 12 - 3 A/D Conversion Time Selection: Caution 2 was modified.	

(30/36)

Edition	Description	Section
Rev.1.30	Figure 12 - 7 Format of A/D Converter Mode Register 1 (ADM1): Caution 2 was modified.	Section 12 A/D Converter (ADC)
	Figure 12 - 8 Format of A/D Converter Mode Register 2 (ADM2) was modified.	
	Figure 16 - 30 Slave Operation Flowchart (1) was modified.	Section 16 Serial Interface IICA (IICA)
	Table 20 - 1 Connections in Logic Cell Block L1 (1) was modified.	Section 20 Logic and Event Link Controller (ELCL)
	Table 20 - 3 Connections in Logic Cell Block L2 (1) was modified.	
	Table 20 - 5 Connections in Logic Cell Block L3 (1) was modified.	
	Figure 20 - 5 Format of Input Signal Select Registers n (ELISELn) (n = 0 to 11): Note 4 was modified and Caution 1 was added.	
	20.6 Points for Caution when the ELCL is to be Used: The description of 4. was modified.	
	27.1 Overview of Safety Functions: The description and Remark 2 were modified.	Section 27 Safety Functions
	27.3.2 CRC operation (general-purpose CRC): The description was modified.	
	27.3.5 RAM guard function: The description was modified.	
	27.3.6 SFR guard function: The description was modified.	
	Table 29 - 1 Correspondences between the Memory Addresses of the SMSIp Registers and Values of the SMSCV0 to SMSCV4 Bits: Caution 2 was modified.	Section 29 SNOOZE Mode Sequencer (SMS)
	Table 29 - 2 Types of Processing Specified by the SMSIp Registers was modified.	
	29.4.1 Internal operations of the SNOOZE mode sequencer: The description was modified.	
	Figure 29 - 8 Flow of Internal Operations of the SNOOZE Mode Sequencer was modified.	
	29.4.4 Interrupt from the SNOOZE Mode Sequencer was added.	
	Figure 29 - 11 Flow of Activating and Running the SNOOZE Mode Sequencer was modified, Note was deleted, and Cautions 1 and 2 were added.	
	29.4.6 States of the SNOOZE mode sequencer: The description was modified.	
	29.5.20 Interrupt plus termination was deleted.	
	29.6 Operation in Standby Modes: Notes 4 to 6 were modified.	
	37.3.2 Supply current characteristics, 1. 30- to 64-pin package products with 96- to 128-Kbyte flash ROM (1/4): Note 1 was modified.	Section 37 Electrical Characteristics
	37.3.2 Supply current characteristics, 1. 30- to 64-pin package products with 96- to 128-Kbyte flash ROM (2/4): Notes 1 to 3 were modified.	
	37.3.2 Supply current characteristics, 1. 30- to 64-pin package products with 96- to 128-Kbyte flash ROM (3/4): Note 1 was modified.	
	37.3.2 Supply current characteristics, 1. 30- to 64-pin package products with 96- to 128-Kbyte flash ROM (4/4): Notes 1 and 3 to 8 were modified.	
	37.3.2 Supply current characteristics, 2. 30- to 64-pin package products with 192- to 256-Kbyte flash ROM and 80-pin package product with 128- to 256-Kbyte flash ROM (1/4): Note 1 was modified.	
	37.3.2 Supply current characteristics, 2. 30- to 64-pin package products with 192- to 256-Kbyte flash ROM and 80-pin package product with 128- to 256-Kbyte flash ROM (2/4): Notes 1 to 3 were modified.	
	37.3.2 Supply current characteristics, 2. 30- to 64-pin package products with 192- to 256-Kbyte flash ROM and 80-pin package product with 128- to 256-Kbyte flash ROM (3/4): Note 1 was modified.	
	37.3.2 Supply current characteristics, 2. 30- to 64-pin package products with 192- to 256-Kbyte flash ROM and 80-pin package product with 128- to 256-Kbyte flash ROM (4/4): Notes 1 and 3 to 8 were modified.	
	37.3.2 Supply current characteristics, 3. 44- to 80-pin package products with 384- to 768-Kbyte flash ROM and 100- to 128-pin package products (1/4) and Note 1 was modified.	

(31/36)

Edition	Description	Section
Rev.1.30	37.3.2 Supply current characteristics, 3. 44- to 80-pin package products with 384- to 768-Kbyte flash ROM and 100- to 128-pin package products (2/4) and Notes 1 to 3 were modified.	Section 37 Electrical Characteristics
	37.3.2 Supply current characteristics, 3. 44- to 80-pin package products with 384- to 768-Kbyte flash ROM and 100- to 128-pin package products (3/4) and Note 1 was modified.	
	37.3.2 Supply current characteristics, 3. 44- to 80-pin package products with 384- to 768-Kbyte flash ROM and 100- to 128-pin package products (4/4), Notes 1, and 3 to 8 were modified.	
	37.4 AC Characteristics was modified.	
	37.6.4 Comparator characteristics: Note was added.	
	38.2 32-pin Products: The figure was added.	Section 38 Package Drawings
	38.5 44-pin Products: The figure was added.	
	38.6 48-pin Products: The figure was added.	
	38.8 64-pin Products: The figure was added.	
Rev.1.40	Table 1 - 1 List of Ordering Part Numbers (2/3) was modified.	Section 1 Outline
	1.6 Outline of Functions: Table was modified.	
	2.2.2 Description of pin functions: Table was modified.	Section 2 Pin Functions
	Figure 2 - 1 Example of the Connection of the VBAT Pin was modified.	
	2.2.3.3 Using the VBAT pin: The description was modified.	
	Figure 2 - 21 Pin Block Diagram for Pin Type 7-38-1 was modified.	
	Figure 2 - 22 Pin Block Diagram for Pin Type 7-39-1 was modified.	
	Figure 2 - 23 Pin Block Diagram for Pin Type 8-1-3 was modified.	
	Figure 2 - 24 Pin Block Diagram for Pin Type 8-1-10 was modified.	
	Figure 2 - 25 Pin Block Diagram for Pin Type 8-1-11 was modified.	
	Figure 2 - 26 Pin Block Diagram for Pin Type 8-6-9 was modified.	
	Figure 2 - 27 Pin Block Diagram for Pin Type 8-31-1 was modified.	
	Figure 2 - 28 Pin Block Diagram for Pin Type 8-31-2 was modified.	
	Figure 2 - 29 Pin Block Diagram for Pin Type 8-33-2 was modified.	
	Figure 2 - 30 Pin Block Diagram for Pin Type 8-33-3 was modified.	
	Figure 2 - 31 Pin Block Diagram for Pin Type 8-38-1 was modified.	
	Figure 2 - 32 Pin Block Diagram for Pin Type 8-38-2 was modified.	
	Figure 2 - 33 Pin Block Diagram for Pin Type 12-38-1 was modified.	
	Figure 2 - 34 Pin Block Diagram for Pin Type 12-38-2 was modified.	
	Figure 2 - 35 Pin Block Diagram for Pin Type 12-38-3 was modified.	
	3.1.3 Internal data memory space: Caution 2 was modified.	Section 3 CPU Architecture
	Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (2/15) was modified.	
	Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (13/15) was modified.	
	Table 3 - 6 List of Extended Special Function Registers (2nd SFRs) (15/15): Note 4 was added.	
	Figure 3 - 40 Example of CALL, CALLT was modified.	

(32/36)

Edition	Description	Section
Rev.1.40	4.3.7 Port mode control A registers (PMCAxx): The description was modified.	Section 4 Port Functions
	Figure 4 - 7 Format of Port Mode Control A Registers (PMCAxx) was modified.	
	Table 4 - 7 Examples of Register and Output Latch Settings for Alternate Functions (30- to 64-pin Products with 96- or 128-Kbyte Flash Memory) (16/18) was modified.	
	Table 4 - 8 Examples of Register and Output Latch Settings for Alternate Functions (Products with 192- to 768-Kbyte Flash Memory and 80- and 100-pin Products with 128-Kbyte Flash Memory) (19/22) was modified.	
	Figure 5 - 1 Block Diagram of Operation State Control was modified.	Section 5 Operation State Control
	Figure 6 - 1 Block Diagram of Clock Generator was modified.	Section 6 Clock Generator
	6.3.7 Subsystem clock supply mode control register (OSMC): Note was added.	
	Figure 6 - 9 Format of Subsystem Clock Supply Mode Control Register (OSMC): Note 1 was added.	
	Figure 6 - 11 Format of High-speed On-chip Oscillator Frequency Select Register (HOCODIV): Caution 3 was modified.	
	6.6.3 Example of setting the XT1 oscillator clock: The description was modified.	
	Table 6 - 2 Examples of Transitions of the CPU Clock and SFR Settings (2/3) was modified.	
	7.3.3 Timer clock select register m (TPSm): The description was modified.	Section 7 Timer Array Unit (TAU)
	8.1 Functions of Realtime Clock: The description was modified.	Section 8 Realtime Clock (RTC)
	Figure 8 - 1 Block Diagram of the Realtime Clock was modified.	
	8.3.2 Subsystem clock supply mode control register (OSMC): Note was added.	
	Figure 8 - 3 Format of Subsystem Clock Supply Mode Control Register (OSMC): Notes 1 and 2 were added.	
	Figure 8 - 4 Format of Realtime Clock Control Register 0 (RTCC0) (2/2): Note was modified.	
	Figure 8 - 19 Procedure for Reading the Realtime Clock: Note was modified.	
	Figure 8 - 20 Procedure for Reading the Realtime Clock (When the Alarm Interrupt is in Use) was added.	
	Figure 8 - 21 Procedure for Writing to the Realtime Clock: Note was modified.	
	Figure 8 - 22 Procedure for Writing to the Realtime Clock (When the Alarm Interrupt is in Use) was added.	
	Figure 9 - 14 Example of Timer Operation in 8-bit Counter Mode was modified and Remark 2 was added.	Section 9 32-bit Interval Timer (TML32)
	Figure 9 - 15 Example of Capture Operation was modified and Remark was deleted.	
	Figure 12 - 12 Format of Analog Input Channel Specification Register (ADS): Note 10 was added.	Section 12 A/D Converter (ADC)
	12.3.11 A/D test register (ADTES): The description was modified.	
	Figure 12 - 15 Format of A/D Test Register (ADTES) was modified, and Note was added.	
	Figure 12 - 23 Example of Software Trigger Wait Mode (Select Mode, One-shot Conversion Mode) Operation Timing: Caution 2 was added.	
	12.6.11 Hardware trigger no-wait mode (scan mode, sequential conversion mode): <9> was modified.	
	12.6.13 Hardware trigger wait mode (select mode, sequential conversion mode): <7> and <8> were added.	

(33/36)

Edition	Description	Section
Rev.1.40	12.6.14 Hardware trigger wait mode (select mode, one-shot conversion mode): <8> and <9> were added.	Section 12 A/D Converter (ADC)
	12.6.15 Hardware trigger wait mode (scan mode, sequential conversion mode): <1> and <7> were modified, and <8> was added.	
	12.6.16 Hardware trigger wait mode (scan mode, one-shot conversion mode): <1> and <8> were modified, and <9> was added.	
	Figure 12 - 35 Settings in Software Trigger Wait Mode was modified.	
	12.7.5 Settings when selecting the temperature sensor output voltage, internal reference voltage, or TSCAP voltage of the CTSU, and software trigger no-wait mode and one-shot conversion mode: The title was modified.	
	Figure 12 - 38 Settings When Selecting the Temperature Sensor Output Voltage, Internal Reference Voltage, or TSCAP voltage of the CTSU was modified.	
	Figure 12 - 45 Flowchart for Setting up SNOOZE Mode (Hardware Trigger) was modified.	
	Figure 15 - 1 Block Diagram of Serial Array Unit 0 was modified.	Section 15 Serial Array Unit (SAU)
	Figure 15 - 2 Block Diagram of Serial Array Unit 1 was modified.	
	Figure 15 - 9 Format of Serial Communication Operation Setting Register mn (SCRmn) (2/2): Note 2 was modified.	
	Figure 15 - 12 Format of Serial Status Register mn (SSRmn) (2/2): Caution 2 was modified.	
	Figure 15 - 17 Format of Serial Output Register m (SOm): Note and Caution were modified.	
	15.3.16 Input switch control register (ISC): The description was modified.	
	Figure 15 - 21 Format of Input Switch Control Register (ISC) and Note 4 were modified.	
	15.5.1 Master transmission: The table was modified.	
	Figure 15 - 58 Example of Contents of Registers for Slave Reception of Simplified SPI (CSI00, CSI01, CSI10, CSI11, CSI20, CSI21, CSI30, CSI31): Remark 2 was modified.	
	Figure 16 - 11 Format of IICA Low-level Width Setting Register n (IICWLn): Remarks 1 and 2 were added.	Section 16 Serial Interface IICA (IICA)
	Figure 16 - 12 Format of IICA High-level Width Setting Register n (IICWHn): Remark 1 was modified.	
	16.4.2 Setting transfer clock by using IICWLn and IICWHn registers: Caution 3 was added.	
	Table 16 - 2 INTIICAn Generation Timing and Control of Clock Stretching: Note 3 and the description of 1. were modified.	
	17.2.12 Registers for controlling the port functions multiplexed with the inputs and outputs of the UARTA serial interface: The description was modified.	Section 17 Serial Interface UARTA (UARTA)
	Figure 17 - 19 Timing Chart When Continuous Transmission Is Started was modified.	
	Figure 17 - 20 Timing Chart When Continuous Transmission Is Completed was modified.	
	17.3.4 Baud rate generator 3. Baud rate calculation d) Permissible baud rate range during reception: The description was modified.	
	Table 17 - 5 Maximum/Minimum Permissible Baud Rate Error was modified.	
	Figure 19 - 4 Start Addresses of Control Data and Vector Table was modified.	Section 19 Data Transfer Controller (DTC)
	19.4 DTC Operation: The description was modified.	
	19.4.3 Repeat mode 1. Example 1 of using repeat mode: Outputting stepping motor control pulses using port pins: The description was modified.	
	Figure 19 - 19 Example 1 of Using Repeat Mode: Outputting Stepping Motor Control Pulses Using Port Pins was modified.	

(34/36)

Edition	Description	Section
Rev.1.40	19.4.3 Repeat mode 2. Example 2 of using repeat mode: Outputting a sine wave using the 8-bit D/A converter: The description was modified.	Section 19 Data Transfer Controller (DTC)
	Figure 19 - 20 Example 2 of Using Repeat Mode: Outputting a Sine Wave Using the 8-bit D/A Converter was modified.	
	19.4.4 Chain transfers: The description was modified.	
	Figure 19 - 22 Example of Using Chain Transfers: Consecutively Capturing A/D Conversion Results and UART0 Transmission was modified.	
	19.5.6 DTC response time: The description was modified.	
	20.6 Points for Caution when the ELCL is to be Used: The description was modified.	Section 20 Logic and Event Link Controller (ELCL)
	Table 21 - 2 Flags Corresponding to Interrupt Request Sources (4/5): Notes 3 and 4 were modified.	Section 21 Interrupt Functions
	Table 23 - 1 Operating States in HALT Mode When the Main System Clock Is Used as the CPU Clock: Note was added.	Section 23 Standby Function
	Table 23 - 2 Operating States in HALT Mode When the Subsystem Clock Is Used as the CPU Clock (2/2) was modified.	
	Table 23 - 4 Operating States in SNOOZE Mode: Note was added.	
	Figure 25 - 2 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (2/2): Remark 1 was modified.	Section 25 Power-on-reset Circuit (POR)
	Figure 27 - 14 Illicit Access Areas was modified.	Section 27 Safety Functions
	27.3.10 Testing of the A/D converter was modified.	
	Figure 27 - 19 Configuration of Testing of the A/D Converter was modified.	
	27.3.10.1 A/D test register (ADTES): The description was modified.	
	Figure 27 - 20 Format of A/D Test Register (ADTES) was modified.	
	28.2.2 Setting of flash read protection: The description was modified.	Section 28 Security Functions
	Section 30 Capacitive Sensing Unit (CTSU2L): The description was modified.	Section 30 Capacitive Sensing Unit (CTSU2L)
	Table 30 - 1 Functional Overview of the CTSU was modified and Note was added.	
	30.1.1 Functional overview of the CTSU: The description was modified.	
	Figure 30 - 3 Block Diagram of the CTSU was modified.	
	Figure 30 - 4 Configuration of Clock Signals for Sensor Driving Pulse Output was modified.	
	Table 30 - 2 Input and Output Pins of the CTSU was modified.	
	30.1.2 Measurement states was added.	
	30.2.3 CTSU control registers AL and AH (CTSUCRAL, CTSUCRAH): The description was modified.	
	Figure 30 - 8 Format of CTSU Control Registers AL and AH (CTSUCRAL, CTSUCRAH) was modified.	
	30.2.4 CTSU control registers BL and BH (CTSUCRBL, CTSUCRBH): The description was modified.	
	Figure 30 - 9 Format of CTSU Control Registers BL and BH (CTSUCRBL, CTSUCRBH) was modified.	
	30.2.5 CTSU measurement channel registers L and H (CTSUMCHL, CTSUMCHH): The description was modified.	
	Figure 30 - 10 Format of CTSU Measurement Channel Registers L and H (CTSUMCHL, CTSUMCHH) was modified.	

(35/36)

Edition	Description	Section
Rev.1.40	30.2.6 CTSU channel enable control registers AL, AH, BL, and BH (CTSUCHACAL, CTSUCHACAH, CTSUCHACBL, CTSUCHACBH): The description was modified.	Section 30 Capacitive Sensing Unit (CTSU2L)
	Figure 30 - 11 Format of CTSU Channel Enable Control Registers AL, AH, BL, and BH (CTSUCHACAL, CTSUCHACAH, CTSUCHACBL, CTSUCHACBH) was modified.	
	30.2.7 CTSU channel transmit/receive control registers AL, AH, BL, and BH (CTSUCHTRCAL, CTSUCHTRCAH, CTSUCHTRCBL, CTSUCHTRCBH): The description was modified.	
	Figure 30 - 12 Format of CTSU Channel Transmit/Receive Control Registers AL, AH, BL, and BH (CTSUCHTRCAL, CTSUCHTRCAH, CTSUCHTRCBL, CTSUCHTRCBH) was modified.	
	30.2.8 CTSU status register L (CTSUSRL): The description was modified.	
	Figure 30 - 13 Format of CTSU Status Register L (CTSUSRL) was modified.	
	30.2.9 CTSU sensor offset registers 0 and 1 (CTSUSO0, CTSUSO1): The description was modified.	
	Figure 30 - 14 Format of CTSU Sensor Offset Registers 0 and 1 (CTSUSO0, CTSUSO1) was modified.	
	30.2.10 CTSU sensor counter registers (CTSUSC, CTSUUC): The description was modified.	
	30.2.11 CTSU calibration registers L and H (CTSUDBGR0, CTSUDBGR1): The description was modified.	
	Figure 30 - 16 Format of CTSU Calibration Registers L and H (CTSUDBGR0, CTSUDBGR1) was modified.	
	30.2.12 CTSU sensor unit clock control registers AL, AH, BL, and BH (CTSUSUCLK0, CTSUSUCLK1, CTSUSUCLK2, CTSUSUCLK3): The description was modified.	
	Figure 30 - 17 Format of CTSU Sensor Unit Clock Control Registers AL, AH, BL, and BH (CTSUSUCLK0, CTSUSUCLK1, CTSUSUCLK2, CTSUSUCLK3) was modified.	
	Figure 30 - 18 Format of CTSU Trimming Registers AL and AH (CTSUTRIM0, CTSUTRIM1) was modified and Note was added.	
	Figure 30 - 19 Format of CTSU Trimming Registers BL and BH (CTSUTRIM2, CTSUTRIM3) was modified and Note was added.	
	30.3 Operation was added.	
	30.4 Usage Notes on the Capacitive Sensing Unit was modified.	
	33.9 Security Settings: The description was modified.	Section 33 Flash Memory
	Table 36 - 5 Operation List (12/18): Caution was modified.	Section 36 Instruction Set
	37.3.1 Pin characteristics: Table (3/7) was modified.	Section 37 Electrical Characteristics
	37.3.2 Supply current characteristics 1. 30- to 64-pin package products with 96- to 128-Kbyte code flash memory Table (1/4): Note 1 was modified.	
	37.3.2 Supply current characteristics 1. 30- to 64-pin package products with 96- to 128-Kbyte code flash memory Table (3/4): Note 1 was modified.	
	37.3.2 Supply current characteristics 2. 30- to 64-pin package products with 192- to 256-Kbyte code flash memory and 80-pin package product with 128- to 256-Kbyte code flash memory Table (1/4): Note 1 was modified.	
	37.3.2 Supply current characteristics 2. 30- to 64-pin package products with 192- to 256-Kbyte code flash memory and 80-pin package product with 128- to 256-Kbyte code flash memory Table (3/4): Note 1 was modified.	

(36/36)

Edition	Description	Section
Rev.1.40	37.3.2 Supply current characteristics 3. 44- to 80-pin package products with 384- to 768-Kbyte code flash memory and 100- to 128-pin package products Table (1/4): Note 1 was modified.	Section 37 Electrical Characteristics
	37.3.2 Supply current characteristics 3. 44- to 80-pin package products with 384- to 768-Kbyte code flash memory and 100- to 128-pin package products Table (3/4): Note 1 was modified.	
	37.4 AC Characteristics: Table was modified.	
	37.5.2 Serial interface UARTA: Caution was modified.	
	37.6.1 A/D converter characteristics 1. Normal modes 1 and 2: Note 2 was modified.	
	37.6.1 A/D converter characteristics 2. Low-voltage modes 1 and 2: Notes 1 and 3 were modified.	
	38.9 80-pin Products: Figure was added.	Section 38 Package Drawings

RL78/G23 User's Manual: Hardware

Publication Date: Rev.0.50 May 28, 2020
Rev.2.00 Jun 30, 2026

Published by: Renesas Electronics Corporation

RL78/G23

Renesas Electronics Corporation

R01UH0896EJ0200