User's Manual

16

RL78/I1D

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.2.40 Mar 2023

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.

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/I1D and design and develop application systems and programs for these devices.
The target products are as follows.

- 20-pin: R5F1176x (x = 8, A)
- 24-pin: R5F1177x (x = 8, A)
- 30-pin: R5F117Ax (x = 8, A, C)
- 32-pin: R5F117Bx (x = A, C)
- 48-pin: R5F117Gx (x = A, C)

Purpose

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

Organization

The RL78/I1D manual is separated into two parts: this manual and the software edition (common to the RL78 family).

RL78/I1D User's Manual Hardware (This Manual)	**RL78 Family User's Manual Software**
• Pin functions • Internal block functions • Interrupts • Other on-chip peripheral functions • Electrical specifications	• CPU functions • Instruction set • Explanation of each instruction

How to Read This Manual

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

- To gain a general understanding of functions:
  → Read this manual in the order of the **CONTENTS**. 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 RL78/I1D Microcontroller instructions:
  → Refer to the separate document **RL78 Family User's Manual Software (R01US0015E)**.

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

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

Documents Related to Devices

Document Name	Document No.
RL78/I1D User's Manual Hardware	This manual
RL78 Family User's Manual Software	R01US0015E

Documents Related to Flash Memory Programming (User's Manual)

Document Name		Document No.
PG-FP5 Flash Memory Programmer User's Manual		—
	RL78, 78K, V850, RX100, RX200, RX600 (Except RX64x), R8C, SH	R20UT2923E
	Common	R20UT2922E
	Setup Manual	R20UT0930E
PG-FP6 Flash Memory Programmer User's Manual		R20UT4025E
E1, E20 Emulator User's Manual		R20UT0398E
E2 Emulator User's Manual		R20UT3538E
E2 Lite Emulator User's Manual		R20UT3240E
Renesas Flash Programmer Flash Memory Programming Software User's Manual		R20UT4066E
Renesas Flash Development Toolkit User's Manual		R20UT0508E

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.

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

CONTENTS

CHAPTER 1 OUTLINE

1.1 Features

Ultra-low power consumption technology

- V_{DD} = 1.6 V to 3.6 V
- HALT mode
- STOP mode
- SNOOZE mode

RL78 CPU core

- CISC architecture with 3-stage pipeline
- Minimum instruction execution time: Can be changed from high speed (0.04167 μs: @ 24 MHz operation with high-speed on-chip oscillator) to ultra-low speed (66.6 μs: @ 15 kHz operation with low-speed on-chip oscillator clock)
- Multiply/divide/multiply & accumulate instructions are supported.
- Address space: 1 MB
- General-purpose registers: (8-bit register × 8) × 4 banks
- On-chip RAM: 0.7 to 3 KB

Code flash memory

- Code flash memory: 8 to 32 KB
- Block size: 1 KB
- Prohibition of block erase and rewriting (security function)
- On-chip debug function
- Self-programming (with boot swap function/flash shield window function)

Data flash memory

- Data flash memory: 2 KB
- Back ground operation (BGO): Instructions can be executed from the program memory while rewriting the data flash memory.
- Number of rewrites: 1,000,000 times (TYP.)
- Voltage of rewrites: V_{DD} = 1.8 to 3.6 V

High-speed on-chip oscillator

- Select from 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 3.6 V, T_A = -20 to +85°C)

Middle-speed on-chip oscillator

- Selectable from 4 MHz, 2 MHz, and 1 MHz.

Operating ambient temperature

- T_A = -40 to +105°C (G: Industrial applications)

Power management and reset function

- On-chip power-on-reset (POR) circuit
- On-chip voltage detector (LVD) (Select interrupt and reset from 12 levels)

Data transfer controller (DTC)

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

Event link controller (ELC)

- Event signals of 20 types can be linked to the specified peripheral function.

Serial interfaces

- Simplified SPI (CSI[Note]): 1 or 2 channels
- UART: 1 channel
- I^2C/simplified I^2C: 1 or 2 channels

Timers

- 16-bit timer: 4 channels
- 12-bit interval timer: 1 channel
- 8-bit interval timer: 4 channels
- Real-time clock: 1 channel (calendar for 99 years, alarm function, and clock correction function)
- Watchdog timer: 1 channel

A/D converter

- 8/12-bit resolution A/D converter (V_{DD} = 1.6 to 3.6 V)
- Analog input: 6 to 17 channels
- Internal reference voltage (1.45 V) and temperature sensor

Comparator

- 2 channels
- Operating modes: Comparator high-speed mode, comparator low-speed mode, window mode

Operational amplifier

- 4 channels

I/O ports

- I/O port: 14 to 42 (N-ch open drain I/O [withstand voltage of 6 V]: 4, N-ch open drain I/O [V_{DD} withstand voltage]: 3 to 7)
- Can be set to N-ch open drain, TTL input buffer, and on-chip pull-up resistor
- Different potential interface: Can connect to a 1.8/2.5 V device
- On-chip key interrupt function
- On-chip clock output/buzzer output controller

Others

- On-chip BCD (binary-coded decimal) correction circuit
- On-chip data operation circuit

<R> **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.

Remark The functions mounted depend on the product. See **1.6 Outline of Functions**.

○ ROM, RAM capacities

Flash ROM	Data flash	RAM	RL78/I1D				
			20 pins	24 pins	30 pins	32 pins	48 pins
32 KB	2 KB	3 KB [Note]	—	—	R5F117AC	R5F117BC	R5F117GC
16 KB	2 KB	2 KB	R5F1176A	R5F1177A	R5F117AA	R5F117BA	R5F117GA
8 KB	2 KB	0.7 KB	R5F11768	R5F11778	R5F117A8	—	—

Note The flash library uses RAM in self-programming and rewriting of the data flash memory.
The target products and start address of the RAM areas used by the flash library are shown below.

R5F117xC (x = A, B, G): Start address FF300H

For the RAM areas used by the flash library, see **Self RAM list of Flash Self-Programming Library for RL78 Family (R20UT2944)**.

1.2 Ordering Information

Figure 1 - 1 Part Number, Memory Size, and Package of RL78/I1D

Part No. R 5 F 1 1 7 G C G x x x F B # U 0

Packaging specification
- #10: Tray (LFQFP, LQFP, LSSOP, TSSOP)
- #30: Tray (LFQFP, LQFP, LSSOP, TSSOP), Tube (LSSOP) Note 1
- #U0, #00, #20: Tray (HWQFN, HVQFN)
- #50: Embossed Tape (LFQFP, LQFP, LSSOP, TSSOP)
- #W0, #40: Embossed Tape (HWQFN, HVQFN)

Package type:
- SM: TSSOP, 0.65 mm pitch
- SP: LSSOP, 0.65 mm pitch
- FP: LQFP, 0.80 mm pitch
- FB: LFQFP, 0.50 mm pitch
- NA: HWQFN, 0.50 mm pitch Note 2
- NA: HVQFN, 0.50 mm pitch Note 3

ROM number (Omitted for blank products)

Fields of application:
- G: Industrial applications, T_A = -40 to +105 °C

ROM capacity:
- 8: 8 KB
- A: 16 KB
- C: 32 KB

Pin count:
- 6: 20-pin
- 7: 24-pin
- A: 30-pin
- B: 32-pin
- G: 48-pin

RL78/I1D

Memory type:
- F : Flash memory

Renesas MCU

Renesas semiconductor product

Note 1. The packaging specification is only “Tube” for products in the 20-pin LSSOP.

Note 2. 24-pin products

Note 3. 32-pin products

Pin count	Package	Ordering Part Number	RENESAS Code
20 pins	20-pin plastic LSSOP (4.4 × 6.5 mm, 0.65 mm pitch)	R5F11768GSP#30, R5F1176AGSP#30, R5F11768GSP#50, R5F1176AGSP#50	PLSP0020JB-A
	20-pin plastic TSSOP (4.4 × 6.5 mm, 0.65 mm pitch)	R5F11768GSM#10, R5F1176AGSM#10, R5F11768GSM#30, R5F1176AGSM#30, R5F11768GSM#50, R5F1176AGSM#50	PTSP0020JI-A
24 pins	24-pin plastic HWQFN (4 × 4 mm, 0.5 mm pitch)	R5F11778GNA#U0, R5F1177AGNA#U0, R5F11778GNA#W0, R5F1177AGNA#W0	PWQN0024KE-A
		R5F11778GNA#00, R5F1177AGNA#00, R5F11778GNA#20, R5F1177AGNA#20, R5F11778GNA#40, R5F1177AGNA#40	PWQN0024KF-A PWQN0024KH-A
30 pins	30-pin plastic LSSOP (7.62 mm (300), 0.65 mm pitch)	R5F117A8GSP#10, R5F117AAGSP#10, R5F117ACGSP#10, R5F117A8GSP#30, R5F117AAGSP#30, R5F117ACGSP#30, R5F117A8GSP#50, R5F117AAGSP#50, R5F117ACGSP#50	PLSP0030JB-B
32 pins	32-pin plastic HVQFN (5 × 5 mm, 0.5 mm pitch)	R5F117BAGNA#00, R5F117BCGNA#00, R5F117BAGNA#20, R5F117BCGNA#20, R5F117BAGNA#40, R5F117BCGNA#40	PVQN0032KE-A
	32-pin plastic LQFP (7 × 7 mm, 0.8 mm pitch)	R5F117BAGFP#10, R5F117BCGFP#10, R5F117BAGFP#30, R5F117BCGFP#30, R5F117BAGFP#50, R5F117BCGFP#50	PLQP0032GB-A
48 pins	48-pin plastic LFQFP (7 × 7 mm, 0.5 mm pitch)	R5F117GAGFB#10, R5F117GCGFB#10, R5F117GAGFB#30, R5F117GCGFB#30, R5F117GAGFB#50, R5F117GCGFB#50	PLQP0048KB-A

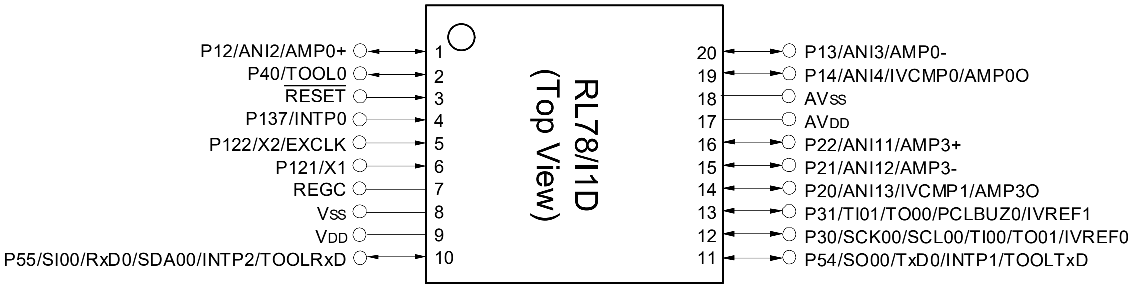
<R>

Caution The ordering part numbers represent the numbers at the time of publication. For the latest ordering part numbers, refer to the target product page of the Renesas Electronics website.

1.3 Pin Configuration (Top View)

1.3.1 20-pin products

- 20-pin plastic LSSOP (4.4 × 6.5 mm, 0.65 mm pitch)
- 20-pin plastic TSSOP (4.4 × 6.5 mm, 0.65 mm pitch)



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

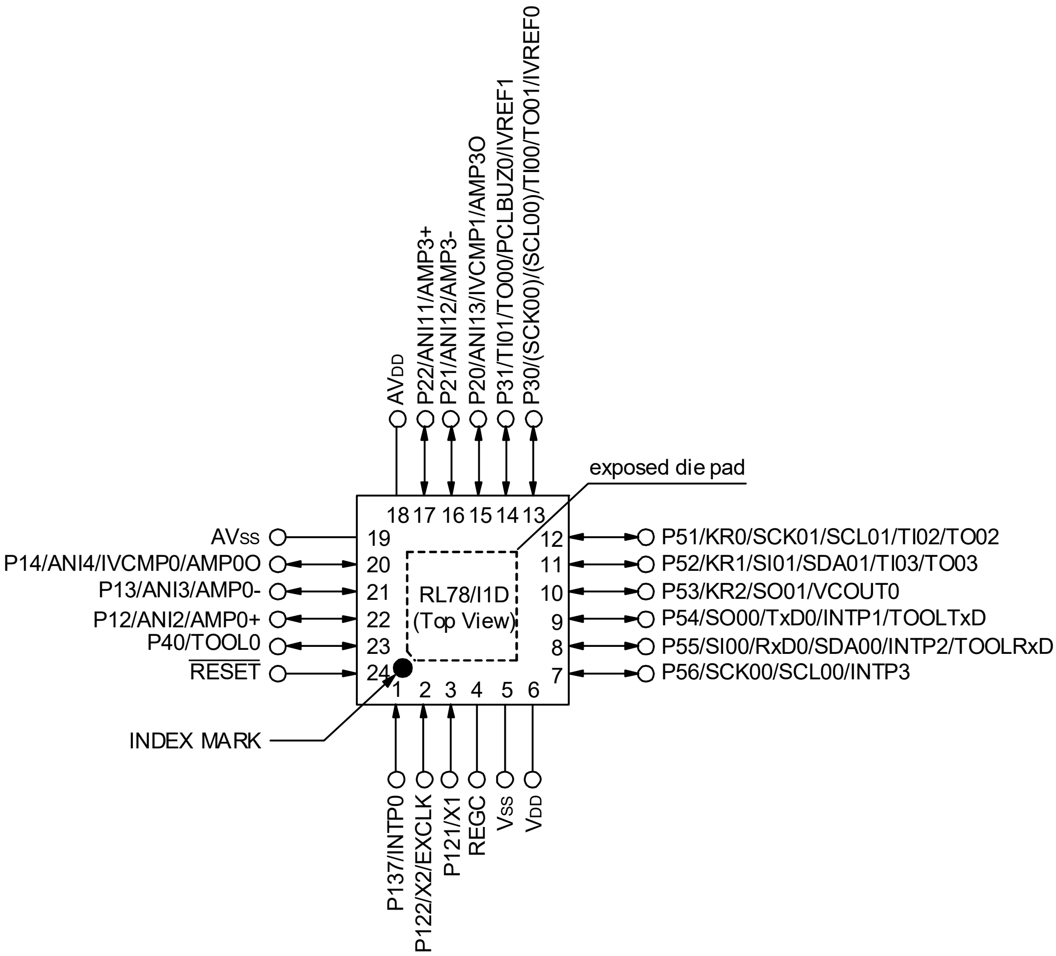
Caution 2. Make AVSS pin the same potential as VSS pin.

Caution 3. Make AVDD pin the same potential as VDD pin.

Remark For pin identification, see **1.4 Pin Identification**.

1.3.2 24-pin products

- 24-pin plastic HWQFN (4 × 4 mm, 0.5 mm pitch)



Caution 1. Connect the REGC pin to V_{SS} pin via a capacitor (0.47 to 1 μF).

Caution 2. Make AV_{SS} pin the same potential as V_{SS} pin.

Caution 3. Make AV_{DD} pin the same potential as V_{DD} pin.

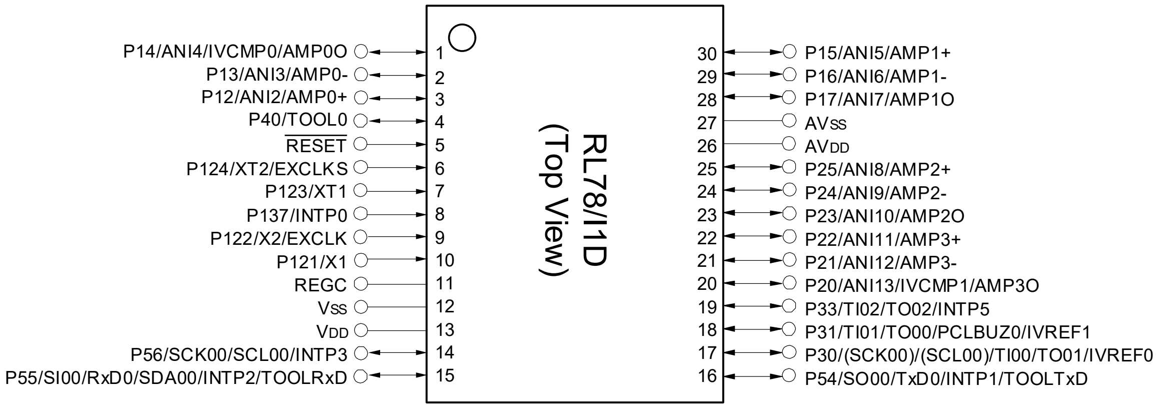
Remark 1. For pin identification, see **1.4 Pin Identification**.

Remark 2. It is recommended to connect an exposed die pad to V_{SS}.

Remark 3. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register 0 (PIOR0).

1.3.3 30-pin products

• 30-pin plastic LSSOP (7.62 mm (300), 0.65 mm pitch)



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

Caution 2. Make AVSS pin the same potential as VSS pin.

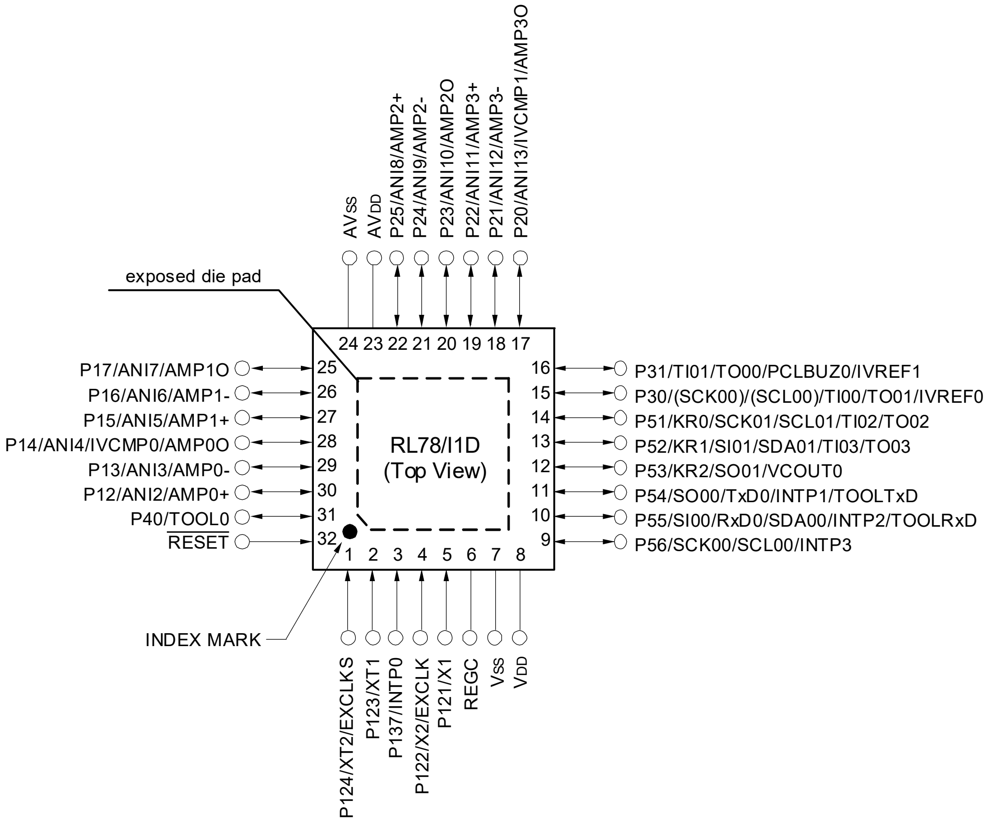
Caution 3. Make AVDD pin the same potential as VDD pin.

Remark 1. For pin identification, see **1.4 Pin Identification**.

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register 0 (PIOR0).

1.3.4 32-pin products

• 32-pin plastic HVQFN (5 × 5 mm, 0.5 mm pitch)



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

Caution 2. Make AVss pin the same potential as Vss pin.

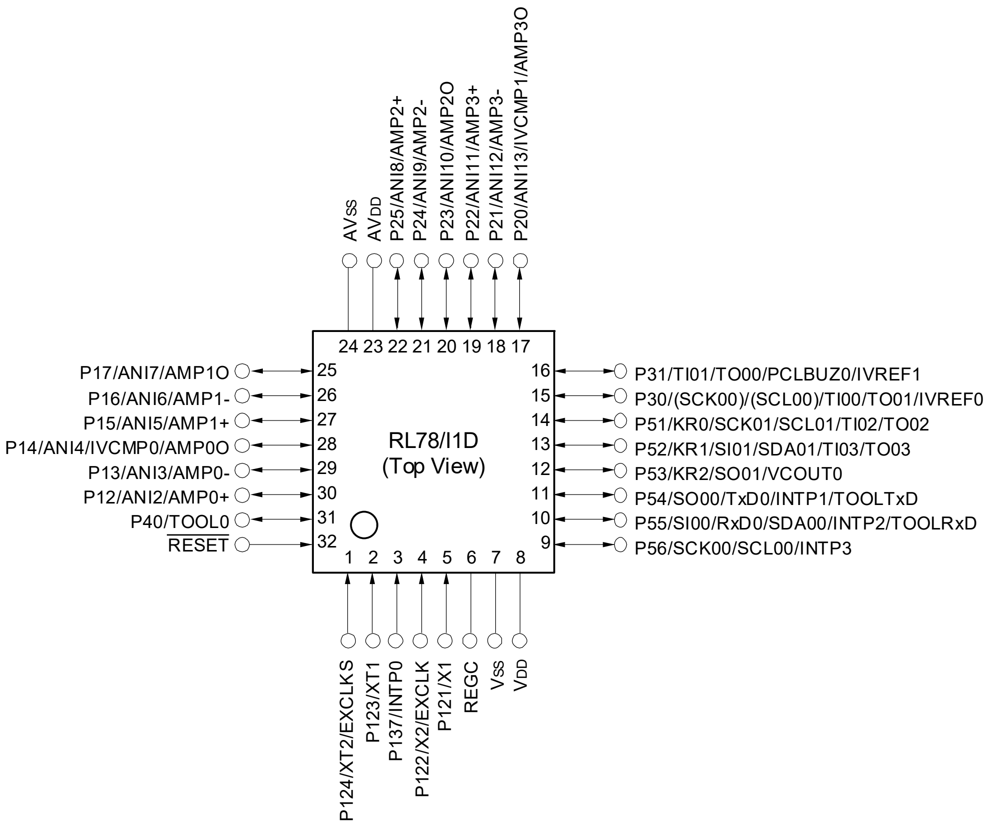
Caution 3. Make AVDD pin the same potential as VDD pin.

Remark 1. For pin identification, see **1.4 Pin Identification**.

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register 0 (PIOR0).

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

• 32-pin plastic LQFP (7 × 7 mm, 0.8 mm pitch)



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

Caution 2. Make AVSS pin the same potential as VSS pin.

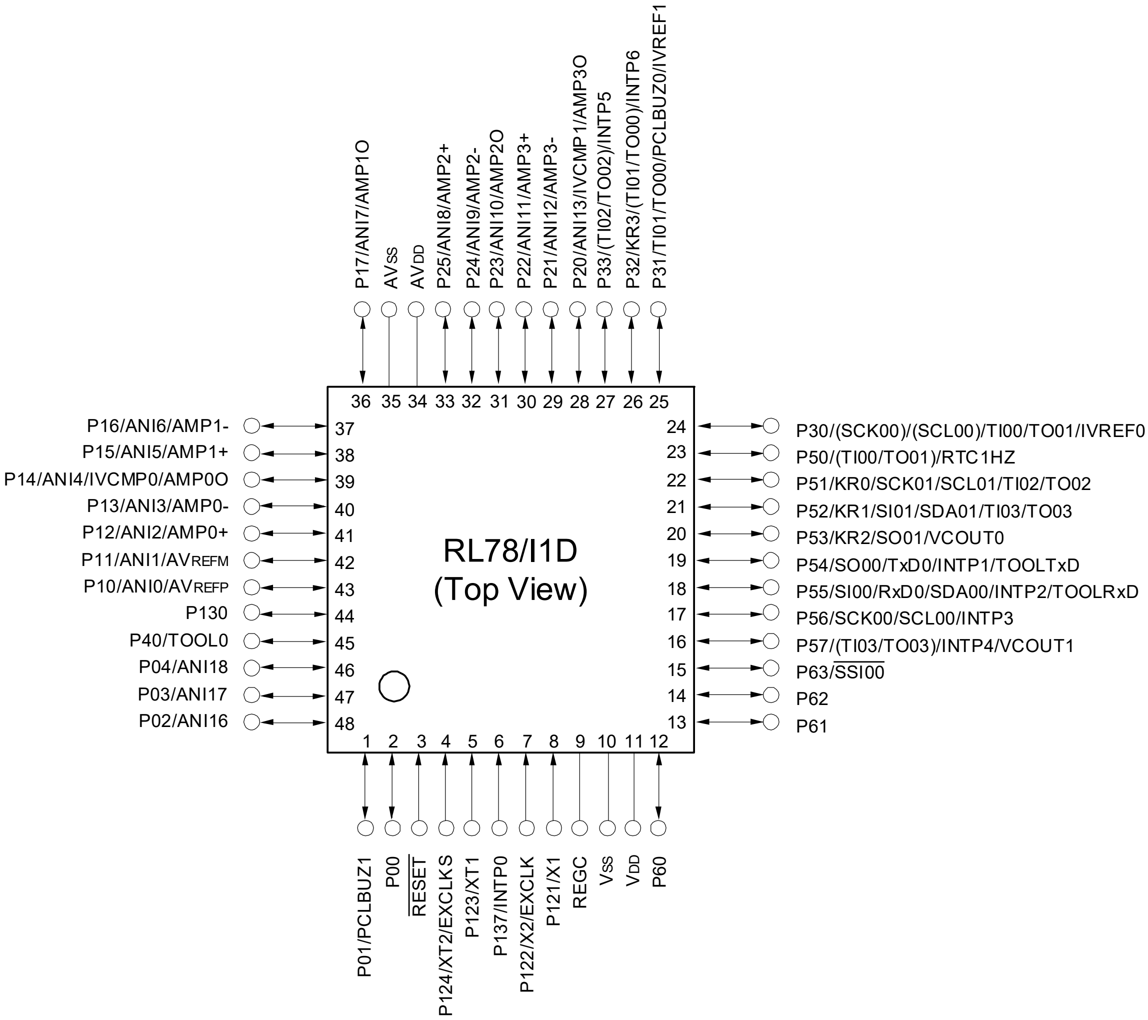
Caution 3. Make AVDD pin the same potential as VDD pin.

Remark 1. For pin identification, see **1.4 Pin Identification**.

Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register 0 (PIOR0).

1.3.5 48-pin products

• 48-pin plastic LFQFP (7 × 7 mm, 0.5 mm pitch)



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

Caution 2. Make AVss pin the same potential as Vss pin.

Caution 3. Make AVDD pin the same potential as VDD pin.

Remark 1. For pin identification, see **1.4 Pin Identification**.

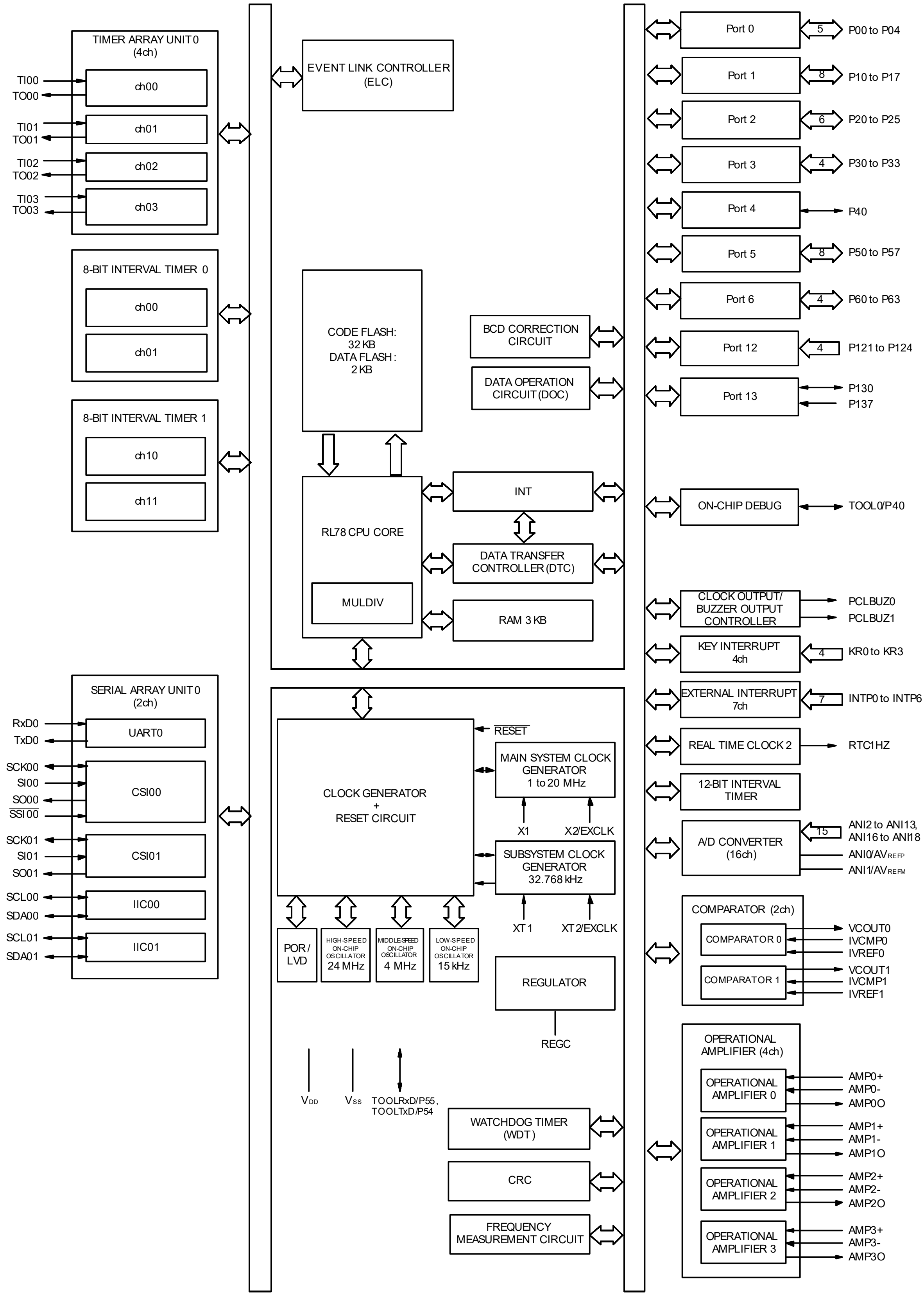
Remark 2. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register 0 (PIOR0).

1.4 Pin Identification

ANI0 to ANI13, ANI16 to ANI18	: Analog input
AV_{DD}	: Analog power supply
AV_{REFM}	: A/D converter reference potential (- side) input
AV_{REFP}	: A/D converter reference potential (+ side) input
AV_{SS}	: Analog ground
EXCLK	: External clock input (main system clock)
EXCLKS	: External clock input (subsystem clock)
INTP0 to INTP6	: External interrupt input
IVCMP0, IVCMP1	: Comparator input
IVREF0, IVREF1	: Comparator reference input
KR0 to KR3	: Key return
P00 to P04	: Port 0
P10 to P17	: Port 1
P20 to P25	: Port 2
P30 to P33	: Port 3
P40	: Port 4
P50 to P57	: Port 5
P60 to P63	: Port 6
P121 to P124	: Port 12
P130, P137	: Port 13
PCLBUZ0, PCLBUZ1	: Programmable clock output/buzzer output
REGC	: Regulator capacitance
$\overline{\text{RESET}}$	: Reset
RTC1HZ	: Real-time clock correction clock (1 Hz) output
RxD0	: Receive data
SCK00, SCK01	: Serial clock input/output
SCL00, SCL01	: Serial clock input/output
SDA00, SDA01	: Serial data input/output
SI00, SI01	: Serial data input
SO00, SO01	: Serial data output
$\overline{\text{SSI00}}$	: Serial interface chip select input
TI00 to TI03	: Timer input
TO00 to TO03	: Timer output
TOOL0	: Data input/output for tool
TOOLRxD, TOOLTxD	: Data input/output for external device
TxD0	: Transmit data
VCOUT0, VCOUT1	: Comparator output
AMP0+, AMP1+, AMP2+, AMP3+	: Operational amplifier (+side) input
AMP0-, AMP1-, AMP2-, AMP3-	: Operational amplifier (-side) input
AMP0O, AMP1O, AMP2O, AMP3O	: Operational amplifier output
V_{DD}	: Power supply
V_{SS}	: Ground
X1, X2	: Crystal oscillator (main system clock)
XT1, XT2	: Crystal oscillator (subsystem clock)

1.5 Block Diagram

1.5.1 48-pin products



1.6 Outline of Functions

Remark This outline describes the functions at the time when Peripheral I/O redirection register 0 (PIOR0) are set to 00H.

(1/2)

Item		20-pin	24-pin	30-pin	32-pin	48-pin
		R5F1176x (x = 8, A)	R5F1177x (x = 8, A)	R5F117Ax (x = 8, A, C)	R5F117Bx (x = A, C)	R5F117Gx (x = A, C)
Code flash memory (KB)		8 to 16 KB	8 to 16 KB	8 to 32 KB	16 to 32 KB	16 to 32 KB
Data flash memory (KB)		2 KB	2 KB	2 KB	2 KB	2 KB
RAM		0.7 to 2.0 KB	0.7 to 2.0 KB	0.7 to 3.0 KB Note	2.0 to 3.0 KB Note	2.0 to 3.0 KB Note
Address space		1 MB				
Main system clock	High-speed system clock (f_{MX})	X1 (crystal/ceramic) oscillation, external main system clock input (EXCLK) HS (High-speed main) mode:1 to 20 MHz (V_{DD} = 2.7 to 3.6 V), HS (High-speed main) mode:1 to 16 MHz (V_{DD} = 2.4 to 3.6 V), LS (Low-speed main) mode:1 to 8 MHz (V_{DD} = 1.8 to 3.6 V), LV (Low-voltage main) mode:1 to 4 MHz (V_{DD} = 1.6 to 3.6 V), LP (Low-power main) mode:1 MHz (V_{DD} = 1.8 to 3.6 V)				
	High-speed on-chip oscillator clock (f_{IH}) Max: 24 MHz Middle-speed on-chip oscillator clock (f_{IM}) Max: 4 MHz	HS (High-speed main) mode: 1 to 24 MHz (V_{DD} = 2.7 to 3.6 V), HS (High-speed main) mode: 1 to 16 MHz (V_{DD} = 2.4 to 3.6 V), LS (Low-speed main) mode: 1 to 8 MHz (V_{DD} = 1.8 to 3.6 V), LV (Low-voltage main) mode: 1 to 4 MHz (V_{DD} = 1.6 to 3.6 V), LP (Low-power main) mode: 1 MHz (V_{DD} = 1.8 to 3.6 V)				
Subsystem clock	Subsystem clock oscillator (f_{SX}, f_{SXR})	—		XT1 (crystal) oscillation 32.768 kHz (TYP.): V_{DD} = 1.6 to 3.6 V		
	Low-speed on-chip oscillator clock (f_{IL})	15 kHz (TYP.): V_{DD} = 1.6 to 3.6 V				
General-purpose register		8 bits × 32 registers (8 bits × 8 registers × 4 banks)				
Minimum instruction execution time		0.04167 μs (High-speed on-chip oscillator clock: f_{IH} = 24 MHz operation)				
		0.05 μs (High-speed system clock: f_{MX} = 20 MHz operation)				
		—		30.5 μs (Subsystem clock oscillator clock: f_{SX} = 32.768 kHz operation)		
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	14	18	24	26	42
	CMOS I/O	11	15	19	21	33
	CMOS input	3	3	5	5	5
	N-ch open-drain I/O (6 V tolerance)	—	—	—	—	4
Timer	16-bit timer	4 channels				
	Watchdog timer	1 channel				
	Real-time clock	1 channel				
	12-bit interval timer	1 channel				
	8/16-bit interval timer	4 channels (8 bit) / 2 channels (16 bit)				
	Timer output	2	4	3	4	4
	RTC output	—		1 channel • 1 Hz (subsystem clock generator and RTC2/other clock: f_{SX} = 32.768 kHz)		

Note The flash library uses RAM in self-programming and rewriting of the data flash memory.
The target products and start address of the RAM areas used by the flash library are shown below.

R5F117xC (x = A, B, G): Start address FF300H

For the RAM areas used by the flash library, see **Self RAM list of Flash Self-Programming Library for RL78 Family (R20UT2944)**.

(2/2)

Item		20-pin R5F1176x (x = 8, A)	24-pin R5F1177x (x = 8, A)	30-pin R5F117Ax (x = 8, A, C)	32-pin R5F117Bx (x = A, C)	48-pin R5F117Gx (x = A, C)
Clock output/buzzer output		1	1	1	1	2
		[20-pin, 24-pin products] • 2.44 kHz, 4.88 kHz, 9.76 kHz, 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz (Main system clock: f_{MAIN} = 20 MHz operation) [30-pin, 32-pin, 48-pin products] • 2.44 kHz, 4.88 kHz, 9.76 kHz, 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz (Main system clock: f_{MAIN} = 20 MHz operation) • 256 Hz, 512 Hz, 1.024 kHz, 2.048 kHz, 4.096 kHz, 8.192 kHz, 16.384 kHz, 32.768 kHz (subsystem clock generator and RTC/other clock: f_{SXR} = 32.768 kHz operation)				
12-bit resolution A/D converter		6 channels	6 channels	12 channels	12 channels	17 channels
Comparator (Window Comparator)		2 channels				
Operational amplifier		2 channels		4 channels		
Data Operation Circuit (DOC)		Comparison, addition, and subtraction of 16-bit data				
Serial interface		[20-pin, 30-pin products] • Simplified SPI (CSI): 1 channel/UART: 1 channel/simplified I^{2}C: 1 channel [24-pin, 32-pin, 48-pin products] • Simplified SPI (CSI): 2 channels/UART: 1 channel/simplified I^{2}C: 2 channels				
Data transfer controller (DTC)		16 sources	20 sources	19 sources	20 sources	22 sources
Event link controller (ELC)		Event input: 15 Event trigger output: 5	Event input: 17 Event trigger output: 5	Event input: 17 Event trigger output: 7	Event input: 17 Event trigger output: 7	Event input: 20 Event trigger output: 7
Vectored interrupt sources	Internal	22	22	24	24	24
	External	3	5	5	5	8
Key interrupt		—	3	—	3	4
Reset		• Reset by $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on-reset • Internal reset by voltage detector • Internal reset by illegal instruction execution Note • Internal reset by RAM parity error • Internal reset by illegal-memory access				
Power-on-reset circuit		• Power-on-reset: 1.51 ± 0.04V (T_A = -40 to +85°C) • Power-down-reset: 1.50 ± 0.04 V (T_A = -40 to +85°C)				
Voltage detector	Power on	1.67 V to 3.13 V (12 stages)				
	Power down	1.63 V to 3.06 V (12 stages)				
On-chip debug function		Provided (Enable to tracing)				
Power supply voltage		V_{DD} = 1.6 to 3.6 V				
Operating ambient temperature		T_A = -40 to +105°C				

The illegal instruction is generated when instruction code FFH is executed.
Reset by the illegal instruction execution is not issued by emulation with the in-circuit emulator or on-chip debug emulator.

CHAPTER 2 PIN FUNCTIONS

2.1 Port Functions

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) 20-pin products

Power Supply	Corresponding Pins
V_{DD}	P30, P31, P40, P54, P55, P121, P122, and P137
AV_{DD}	P12 to P14 and P20 to P22

(2) 24-pin products

Power Supply	Corresponding Pins
V_{DD}	P30, P31, P40, P51 to P56, P121, P122, and P137
AV_{DD}	P12 to P14 and P20 to P22

(3) 30-pin products

Power Supply	Corresponding Pins
V_{DD}	P30, P31, P33, P40, P54 to P56, P121 to P124, and P137
AV_{DD}	P12 to P17 and P20 to P25

(4) 32-pin products

Power Supply	Corresponding Pins
V_{DD}	P30, P31, P40, P51 to P56, P121 to P124, and P137
AV_{DD}	P12 to P17 and P20 to P25

(5) 48-pin products

Power Supply	Corresponding Pins
V_{DD}	P00 to P04, P30 to P33, P40, P50 to P57, P60 to P63, P121 to P124, P130, and P137
AV_{DD}	P10 to P17 and P20 to P25

Set in each port I/O, buffer, pull-up resistor is also valid for alternate functions.

2.1.1 20-pin Products

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P12	4-15-3	I/O	Analog input port	ANI2/AMP0+	Port 1. 3-bit I/O port. Input/output can be specified in 1-bit units.
P13				ANI3/AMP0-	
P14	4-17-3			ANI4/IVCMP0/AMP0O	
P20	4-17-3	I/O	Analog input port	ANI13/IVCMP1/AMP3O	Port 2. 3-bit I/O port. Input/output can be specified in 1-bit units.
P21	4-15-3			ANI12/AMP3-	
P22				ANI11/AMP3+	
P30	8-6-4	I/O	Input port	SCK00/SCL00/TI00/TO01/IVREF0	Port 3. 2-bit I/O port. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Input of P30 can be set to TTL input buffer. Output of P30 can be set to N-ch open-drain output. Input/output can be specified in 1-bit units.
P31	7-6-2			TI01/TO00/PCLBUZ0/IVREF1	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 1-bit I/O port. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P54	8-1-4	I/O	Input port	SO00/TxD0/INTP1/TOOLTxD	Port 5. 2-bit I/O port. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Input of P54 and P55 can be set to TTL input buffer. Output of P54 and P55 can be set to N-ch open-drain output. Input/output can be specified in 1-bit units.
P55				SI00/RxD0/SDA00/INTP2/TOOLRxD	
P121	2-2-1	Input	Input port	X1	Port 12. 2-bit input-only port.
P122				X2/EXCLK	
P137	2-1-2	Input	Input port	INTP0	Port 13. 1-bit input-only port.

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register 0 (PIOR0).

2.1.2 24-pin Products

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P12	4-15-3	I/O	Analog input port	ANI2/AMP0+	Port 1. 3-bit I/O port. Input/output can be specified in 1-bit units.
P13				ANI3/AMP0-	
P14	4-17-3			ANI4/IVCMP0/AMP0O	
P20	4-17-3	I/O	Analog input port	ANI13/IVCMP1/AMP3O	Port 2. 3-bit I/O port. Input/output can be specified in 1-bit units.
P21	4-15-3			ANI12/AMP3-	
P22				ANI11/AMP3+	
P30	8-6-4	I/O	Input port	(SCK00)/(SCL00)/TI00/TO01/IVREF0	Port 3. 2-bit I/O port. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Input of P30 can be set to TTL input buffer. Output of P30 can be set to N-ch open-drain output. Input/output can be specified in 1-bit units.
P31	7-6-2			TI01/TO00/PCLBUZ0/IVREF1	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 1-bit I/O port. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P51	8-1-4	I/O	Input port	KR0/SCK01/SCL01/TI02/TO02	Port 5. 6-bit I/O port. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Input of P51, P52, and P54 to P56 can be set to TTL input buffer. Output of P51 to P56 can be set to N-ch open-drain output. Input/output can be specified in 1-bit units.
P52				KR1/SI01/SDA01/TI03/TO03	
P53	7-1-4			KR2/SO01/VCOUT0	
P54	8-1-4			SO00/TxD0/INTP1/TOOLTxD	
P55				SI00/RxD0/SDA00/INTP2/TOOLRxD	
P56				SCK00/SCL00/INTP3	
P121	2-2-1	Input	Input port	X1	Port 12. 2-bit input-only port.
P122				X2/EXCLK	
P137	2-1-2	Input	Input port	INTP0	Port 13. 1-bit input-only port.

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register 0 (PIOR0).

2.1.3 30-pin Products

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P12	4-15-3	I/O	Analog input port	ANI2/AMP0+	Port 1. 6-bit I/O port. Input/output can be specified in 1-bit units.
P13				ANI3/AMP0-	
P14	4-17-3			ANI4/IVCMP0/AMP0O	
P15	4-15-3			ANI5/AMP1+	
P16				ANI6/AMP1-	
P17	4-16-3			ANI7/AMP1O	
P20	4-17-3	I/O	Analog input port	ANI13/IVCMP1/AMP3O	Port 2. 6-bit I/O port. Input/output can be specified in 1-bit units.
P21	4-15-3			ANI12/AMP3-	
P22				ANI11/AMP3+	
P23	4-16-3			ANI10/AMP2O	
P24	4-15-3			ANI9/AMP2-	
P25				ANI8/AMP2+	
P30	8-6-4	I/O	Input port	(SCK00)/(SCL00)/TI00/TO01/IVREF0	Port 3. 3-bit I/O port. Input/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. Input of P30 and P33 can be set to TTL input buffer. Output of P30 can be set to N-ch open-drain output.
P31	7-6-2			TI01/TO00/PCLBUZ0/IVREF1	
P33	8-1-3			TI02/TO02/INTP5	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 1-bit I/O port. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P54	8-1-4	I/O	Input port	SO00/TxD0/INTP1/TOOLTxD	Port 5. 3-bit I/O port. Input/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. Input of P54 to P56 can be set to TTL input buffer. Output of P54 to P56 can be set to N-ch open-drain output.
P55				SI00/RxD0/SDA00/INTP2/TOOLRxD	
P56				SCK00/SCL00/INTP3	
P121	2-2-1	Input	Input port	X1	Port 12. 4-bit input-only port.
P122				X2/EXCLK	
P123				XT1	
P124				XT2/EXCLKS	
P137	2-1-2	Input	Input port	INTP0	Port 13. 1-bit input-only port.

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register 0 (PIOR0).

2.1.4 32-pin Products

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P12	4-15-3	I/O	Analog input port	ANI2/AMP0+	Port 1. 6-bit I/O port. Input/output can be specified in 1-bit units.
P13				ANI3/AMP0-	
P14	4-17-3			ANI4/IVCMP0/AMP0O	
P15	4-15-3			ANI5/AMP1+	
P16				ANI6/AMP1-	
P17	4-16-3			ANI7/AMP1O	
P20	4-17-3	I/O	Analog input port	ANI13/IVCMP1/AMP3O	Port 2. 6-bit I/O port. Input/output can be specified in 1-bit units.
P21	4-15-3			ANI12/AMP3-	
P22				ANI11/AMP3+	
P23	4-16-3			ANI10/AMP2O	
P24	4-15-3			ANI9/AMP2-	
P25				ANI8/AMP2+	
P30	8-6-4	I/O	Input port	(SCK00)/(SCL00)/TI00/TO01/IVREF0	Port 3. 2-bit I/O port. Input/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. Input of P30 can be set to TTL input buffer. Output of P30 can be set to N-ch open-drain output.
P31	7-6-2			TI01/TO00/PCLBUZ0/IVREF1	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 1-bit I/O port. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P51	8-1-4	I/O	Input port	KR0/SCK01/SCL01/TI02/TO02	Port 5. 6-bit I/O port. Input/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. Input of P51, P52, and P54 to P56 can be set to TTL input buffer. Output of P51 to P56 can be set to N-ch open-drain output.
P52				KR1/SI01/SDA01/TI03/TO03	
P53	7-1-4			KR2/SO01/VCOUT0	
P54	8-1-4			SO00/TxD0/INTP1/TOOLTxD	
P55				SI00/RxD0/SDA00/INTP2/TOOLRxD	
P56				SCK00/SCL00/INTP3	
P121	2-2-1	Input	Input port	X1	Port 12. 4-bit input-only port.
P122				X2/EXCLK	
P123				XT1	
P124				XT2/EXCLKS	
P137	2-1-2	Input	Input port	INTP0	Port 13. 1-bit input-only port.

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register 0 (PIOR0).

2.1.5 48-pin Products

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-1-3	I/O	Analog input port	—	Port 0. 5-bit I/O port. Input/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.
P01				PCLBUZ1	
P02	7-3-3			ANI16	
P03				ANI17	
P04				ANI18	
P10	4-3-5	I/O	Analog input port	ANI0/AV_{REFP}	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units.
P11				ANI1/AV_{REFM}	
P12	4-15-3			ANI2/AMP0+	
P13				ANI3/AMP0-	
P14	4-17-3			ANI4/IVCMP0/AMP0O	
P15	4-15-3			ANI5/AMP1+	
P16				ANI6/AMP1-	
P17	4-16-3			ANI7/AMP1O	
P20	4-17-3	I/O	Analog input port	ANI13/IVCMP1/AMP3O	Port 2. 6-bit I/O port. Input/output can be specified in 1-bit units.
P21	4-15-3			ANI12/AMP3-	
P22				ANI11/AMP3+	
P23	4-16-3			ANI10/AMP2O	
P24	4-15-3			ANI9/AMP2-	
P25				ANI8/AMP2+	
P30	8-6-4	I/O	Input port	(SCK00)/(SCL00)/TI00/TO01/IVREF0	Port 3. 4-bit I/O port. Input/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. Input of P30, P32, and P33 can be set to TTL input buffer. Output of P30 can be set to N-ch open-drain output.
P31	7-6-2			TI01/TO00/PCLBUZ0/IVREF1	
P32	8-1-3			KR3/(TI01/TO00)/INTP6	
P33				(TI02/TO02)/INTP5	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 1-bit I/O port. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting at input port.

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register 0 (PIOR0).

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P50	7-1-3	I/O	Input port	(TI00/TO01)/RTC1HZ	Port 5. 8-bit I/O port. Input/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. Input of P51, P52, and P54 to P57 can be set to TTL input buffer. Output of P51 to P56 can be set to N-ch open-drain output.
P51	8-1-4			KR0/SCK01/SCL01/TI02/TO02	
P52				KR1/SI01/SDA01/TI03/TO03	
P53	7-1-4			KR2/SO01/VCOUT0	
P54	8-1-4			SO00/TxD0/INTP1/TOOLTxD	
P55				SI00/RxD0/SDA00/INTP2/TOOLRxD	
P56				SCK00/SCL00/INTP3	
P57	8-1-3			(TI03/TO03)/INTP4/VCOUT1	
P60	12-1-1	I/O	Input port	—	Port 6. 4-bit I/O port. Input/output can be specified in 1-bit units. Output of P60 to P63 is N-ch open-drain output (6 V tolerance).
P61				—	
P62				—	
P63				$\overline{\text{SSI00}}$	
P121	2-2-1	Input	Input port	X1	Port 12. 4-bit input-only port.
P122				X2/EXCLK	
P123				XT1	
P124				XT2/EXCLKS	
P130	7-1-3	I/O	Input port	—	Port 13. 1-bit I/O port. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P137	2-1-2	Input	Input port	INTP0	Port 13. 1-bit input-only port.

Remark Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register 0 (PIOR0).

2.2 Functions other than port pins

2.2.1 Functions for each product

(1/3)

Function Name	48-pin	32-pin	30-pin	24-pin	20-pin
ANI0	√	—	—	—	—
ANI1	√	—	—	—	—
ANI2	√	√	√	√	√
ANI3	√	√	√	√	√
ANI4	√	√	√	√	√
ANI5	√	√	√	—	—
ANI6	√	√	√	—	—
ANI7	√	√	√	—	—
ANI8	√	√	√	—	—
ANI9	√	√	√	—	—
ANI10	√	√	√	—	—
ANI11	√	√	√	√	√
ANI12	√	√	√	√	√
ANI13	√	√	√	√	√
ANI16	√	—	—	—	—
ANI17	√	—	—	—	—
ANI18	√	—	—	—	—
INTP0	√	√	√	√	√
INTP1	√	√	√	√	√
INTP2	√	√	√	√	√
INTP3	√	√	√	√	—
INTP4	√	—	—	—	—
INTP5	√	—	√	—	—
INTP6	√	—	—	—	—
IVCMP0	√	√	√	√	√
IVCMP1	√	√	√	√	√
IVREF0	√	√	√	√	√
IVREF1	√	√	√	√	√
KR0	√	√	—	√	—
KR1	√	√	—	√	—
KR2	√	√	—	√	—
KR3	√	—	—	—	—
PCLBUZ0	√	√	√	√	√
PCLBUZ1	√	—	—	—	—
REGC	√	√	√	√	√
RTC1HZ	√	—	—	—	—
$\overline{\text{RESET}}$	√	√	√	√	√
RxD0	√	√	√	√	√
SCK00	√	√	√	√	√

(2/3)

Function Name	48-pin	32-pin	30-pin	24-pin	20-pin
SCK01	√	√	—	√	—
SCL00	√	√	√	√	√
SCL01	√	√	—	√	—
SDA00	√	√	√	√	√
SDA01	√	√	—	√	—
SI00	√	√	√	√	√
SI01	√	√	—	√	—
SO00	√	√	√	√	√
SO01	√	√	—	√	—
$\overline{SSI00}$	√	—	—	—	—
TI00	√	√	√	√	√
TI01	√	√	√	√	√
TI02	√	√	√	√	—
TI03	√	√	—	√	—
TO00	√	√	√	√	√
TO01	√	√	√	√	√
TO02	√	√	√	√	—
TO03	√	√	—	√	—
TxD0	√	√	√	√	√
VCOUT0	√	√	—	√	—
VCOUT1	√	—	—	—	—
X1	√	√	√	√	√
X2	√	√	√	√	√
EXCLK	√	√	√	√	√
EXCLKS	√	√	√	—	—
XT1	√	√	√	—	—
XT2	√	√	√	—	—
V_{DD}	√	√	√	√	√
AV_{DD}	√	√	√	√	√
AV_{REFP}	√	—	—	—	—
AV_{REFM}	√	—	—	—	—
V_{SS}	√	√	√	√	√
TOOLRxD	√	√	√	√	√
TOOLTxD	√	√	√	√	√
TOOL0	√	√	√	√	√
AV_{SS}	√	√	√	√	√
AMP0+	√	√	√	√	√
AMP1+	√	√	√	—	—
AMP2+	√	√	√	—	—

(3/3)

Function Name	48-pin	32-pin	30-pin	24-pin	20-pin
AMP3+	√	√	√	√	√
AMP0-	√	√	√	√	√
AMP1-	√	√	√	—	—
AMP2-	√	√	√	—	—
AMP3-	√	√	√	√	√
AMP0O	√	√	√	√	√
AMP1O	√	√	√	—	—
AMP2O	√	√	√	—	—
AMP3O	√	√	√	√	√

2.2.2 Pins for each product (pins other than port pins)

(1/2)

Function Name	I/O	Function
ANI0 to ANI13, ANI16 to ANI18	Input	A/D converter analog input (see **Figure 14 - 46 Analog Input Pin Connection**)
INTP0 to INTP6	Input	External interrupt request input pin for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.
IVCMP0, IVCMP1	Input	Comparator analog voltage input
IVREF0, IVREF1	Input	Comparator reference voltage input
VCOUT0, VCOUT1	Output	Comparator output
KR0 to KR3	Input	Key interrupt input
PCLBUZ0, PCLBUZ1	Output	Clock output/buzzer output
REGC	—	Pin for connecting regulator output stabilization capacitance for internal operation. Connect this pin to V_{SS} 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	Real-time clock correction clock (1 Hz) output
$\overline{\text{RESET}}$	Input	This is the active-low system reset input pin. When the external reset pin is not used, connect this pin directly or via a resistor to V_{DD}.
RxD0	Input	Serial data input pins of serial interface UART0
TxD0	Output	Serial data output pins of serial interface UART0
SCK00, SCK01	I/O	Serial clock I/O pins of serial interface CSI00 and CSI01
SCL00, SCL01	Output	Serial clock output pins of serial interface IIC00 and IIC01
SDA00, SDA01	I/O	Serial data I/O pins of serial interface IIC00 and IIC01
SI00, SI01	Input	Serial data input pins of serial interface CSI00 and CSI01
$\overline{\text{SSI00}}$	Input	Chip select input pin of serial interface CSI00
SO00, SO01	Output	Serial data output pins of serial interface CSI00 and CSI01
TI00 to TI03	Input	The pins for inputting an external count clock/capture trigger to 16-bit timers 00 to 03
TO00 to TO03	Output	Timer output pins of 16-bit timers 00 to 03
X1, X2	—	Resonator connection for main system clock
EXCLK	Input	External clock input for main system clock
XT1, XT2	—	Resonator connection for subsystem clock
EXCLKS	Input	External clock input for subsystem clock
V_{DD}	—	Positive power supply for pins other than analog pins
AV_{DD}	—	Positive power supply for analog pins
AV_{REFP}	Input	A/D converter reference potential (+ side) input
AV_{REFM}	Input	A/D converter reference potential (- side) input
V_{SS}	—	Ground potential for pins other than analog pins
AV_{SS}	—	Ground potential for analog pins
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

(2/2)

Function Name	I/O	Function
TOOL0	I/O	Data I/O for flash memory programmer/debugger
AMP0+	Input	Operational amplifier positive side input
AMP1+		
AMP2+		
AMP3+		
AMP0-	Input	Operational amplifier negative side input
AMP1-		
AMP2-		
AMP3-		
AMP0O	Output	Operational amplifier output
AMP1O		
AMP2O		
AMP3O		

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

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

P40/TOOL0	Operating mode
V_{DD}	Normal operation mode
0 V	Flash memory programming mode

For details, see 30.4 Programming Method.

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} lines.

2.3 Connection of Unused Pins

Table 2 - 3 shows the Connection of Unused Pins.

Remark The mounted pins depend on the product. Refer to **1.3 Pin Configuration (Top View)** and **2.1 Port Functions**.

Table 2 - 3 Connection of Unused Pins

Pin Name	I/O	Recommended Connection of Unused Pins
P00 to P04	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P10 to P17 P20 to P25		Input: Independently connect to AV_{DD} or AV_{SS} via a resistor. Output: Leave open.
P30 to P33		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P40/TOOL0		Input: Independently connect to V_{DD} via a resistor, or leave open. Output: Leave open.
P50 to P57		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P60 to P63		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Set the port's output latch to 0 and leave the pins open, or set the port's output latch to 1 and independently connect the pins to V_{DD} or V_{SS} via a resistor.
P121 to P124	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
P130	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P137	Input	Independently connect to V_{DD} or V_{SS} via a resistor.
$\overline{\text{RESET}}$	Input	Connect to V_{DD} directly or via a resistor.
REGC	—	Connect to V_{SS} via a capacitor (0.47 to 1 μF).

2.4 Pin Block Diagrams

For the pin types listed in 2.1.1 20-pin Products to 2.1.5 48-pin Products, pin block diagrams are shown in Figures 2 - 3 to 2 - 15.

Figure 2 - 1 Pin Block Diagram of Pin Type 2-1-1

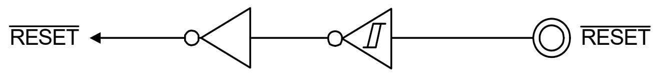
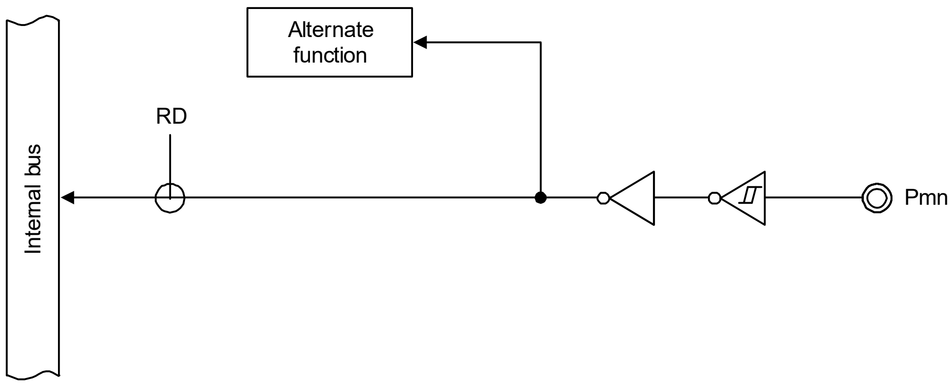
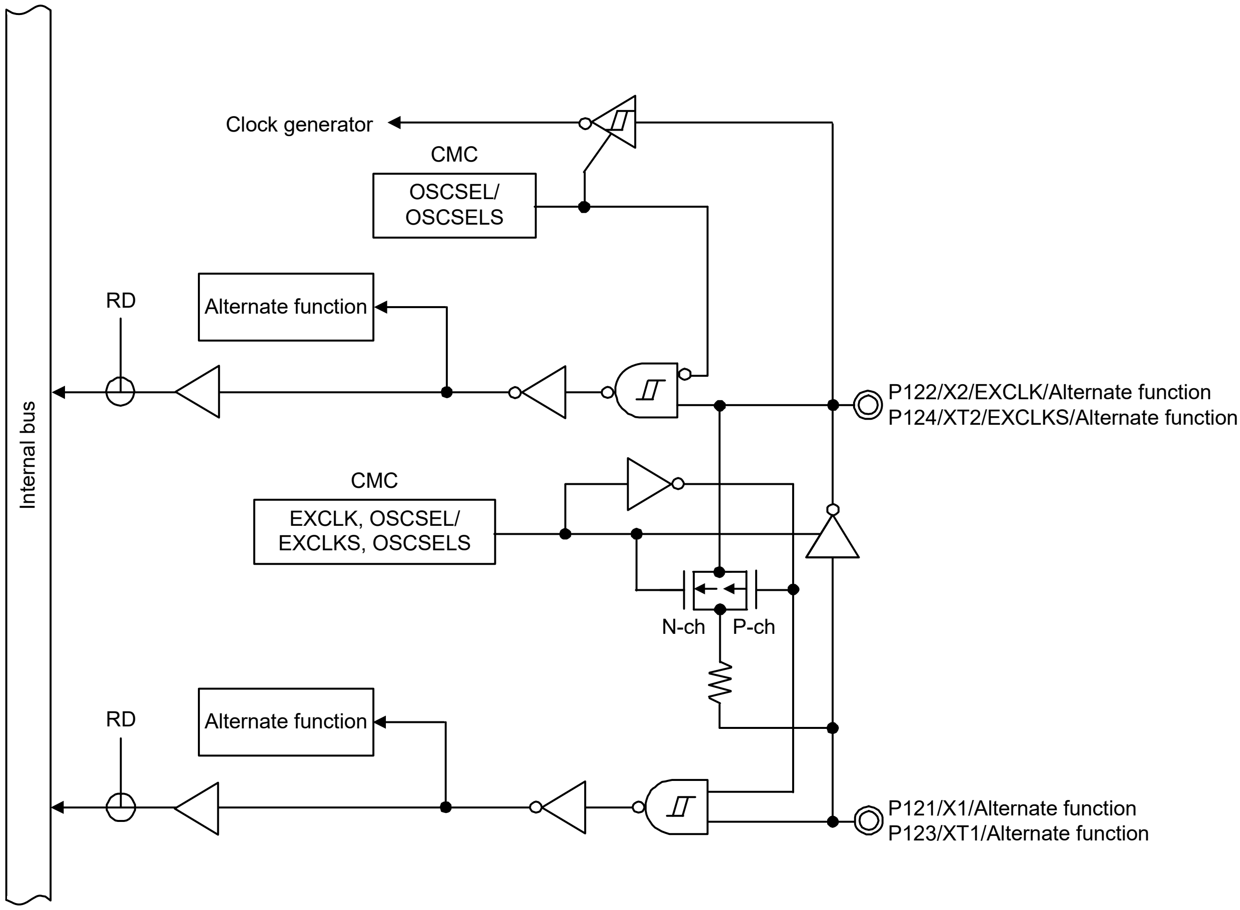


Figure 2 - 2 Pin Block Diagram of Pin Type 2-1-2



Remark Refer to **2.1 Port Functions** for alternate functions.

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



Remark Refer to **2.1 Port Functions** for alternate functions.

Figure 2 - 4 Pin Block Diagram of Pin Type 4-3-5

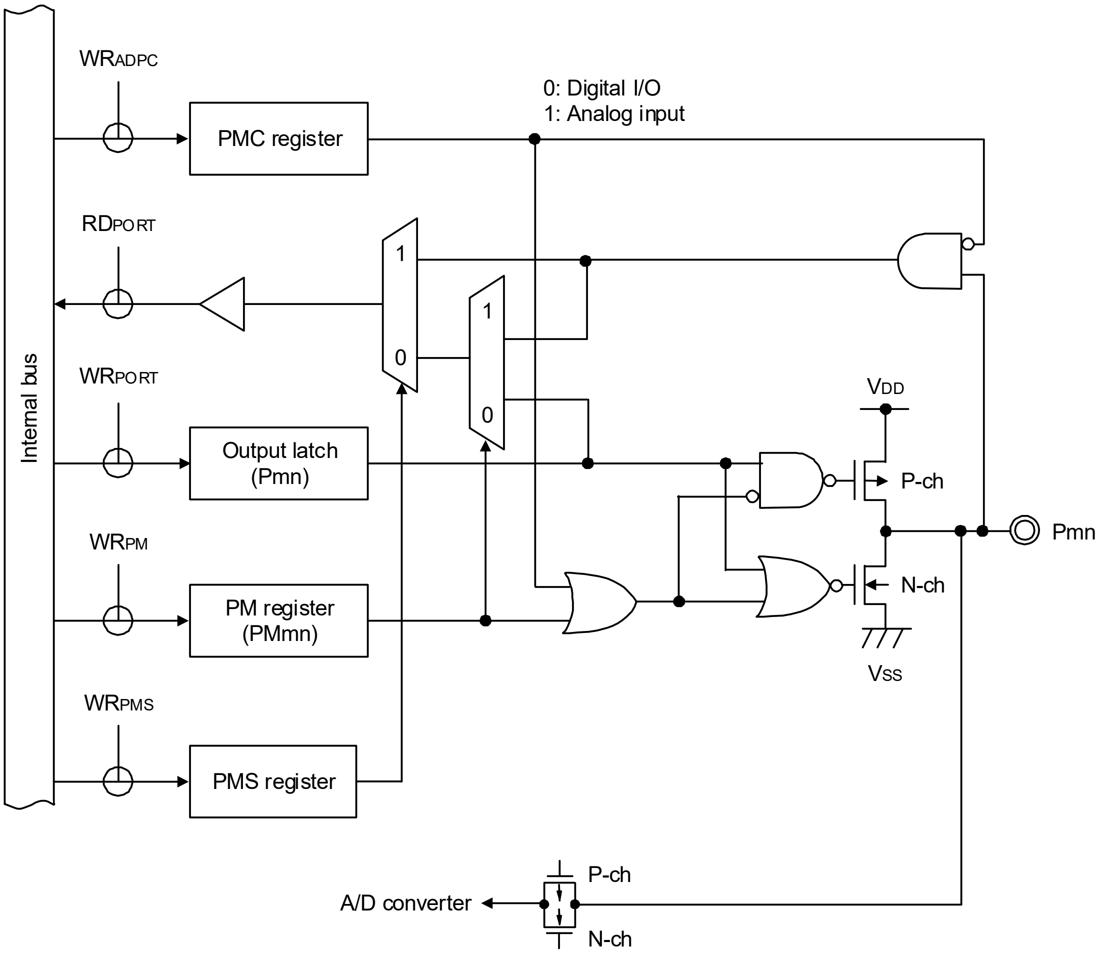


Figure 2 - 5 Pin Block Diagram of Pin Type 4-15-3

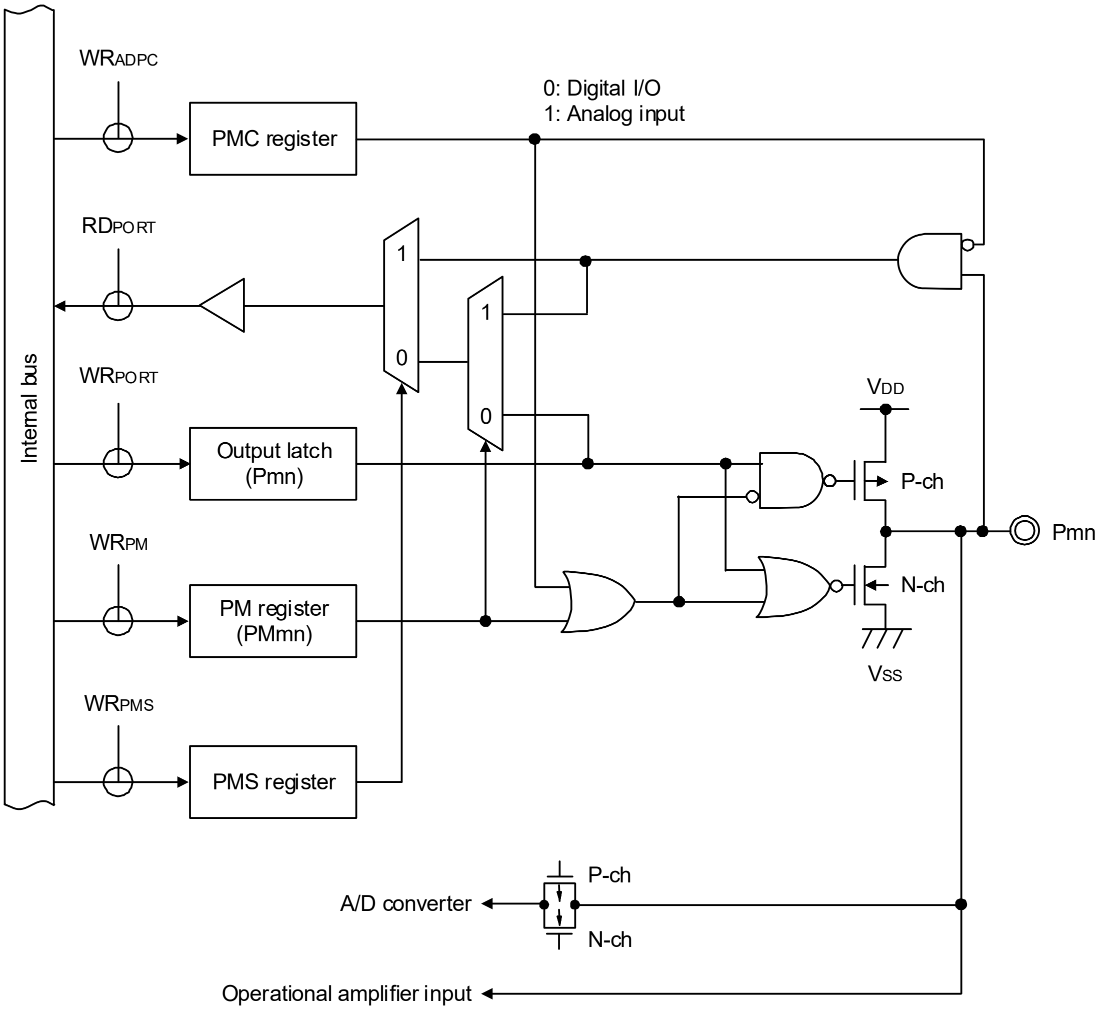


Figure 2 - 6 Pin Block Diagram of Pin Type 4-16-3

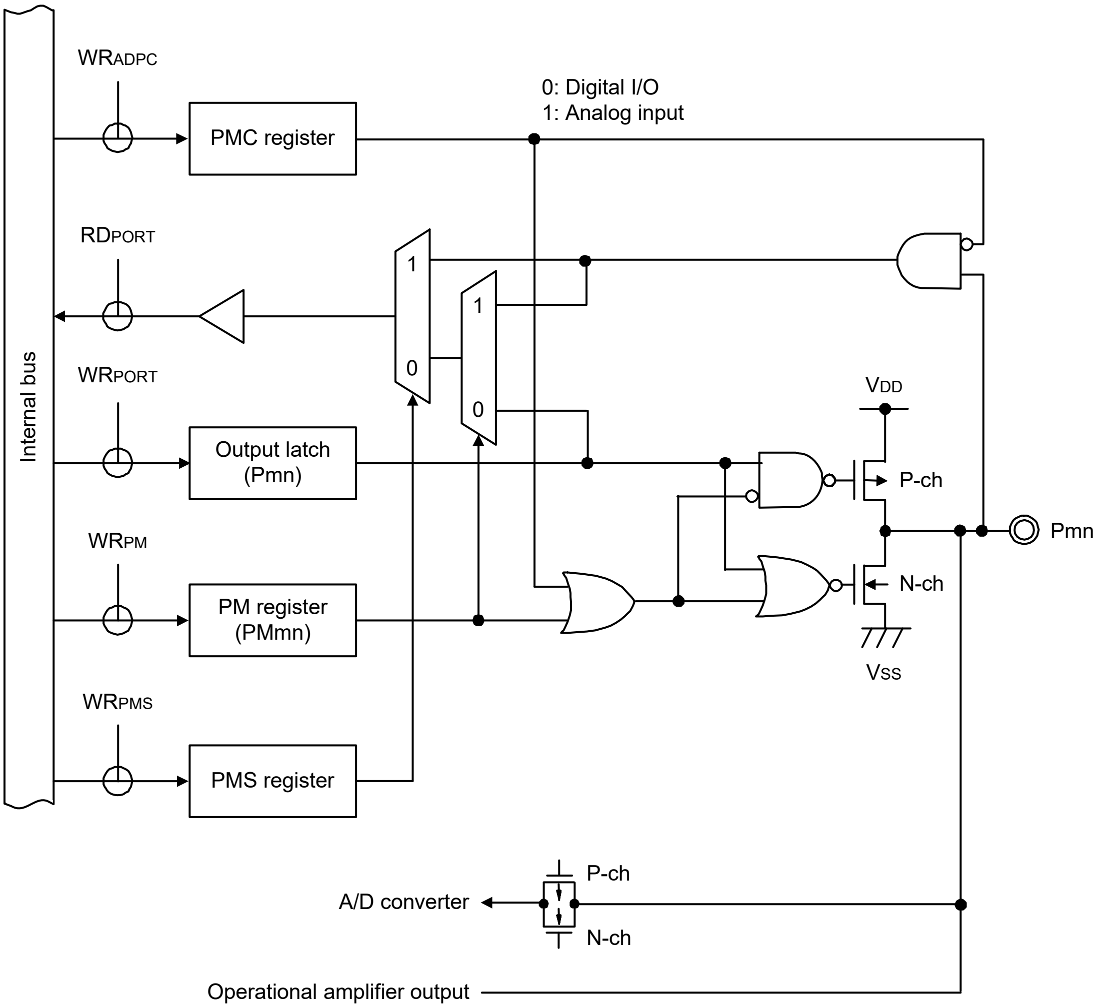


Figure 2 - 7 Pin Block Diagram of Pin Type 4-17-3

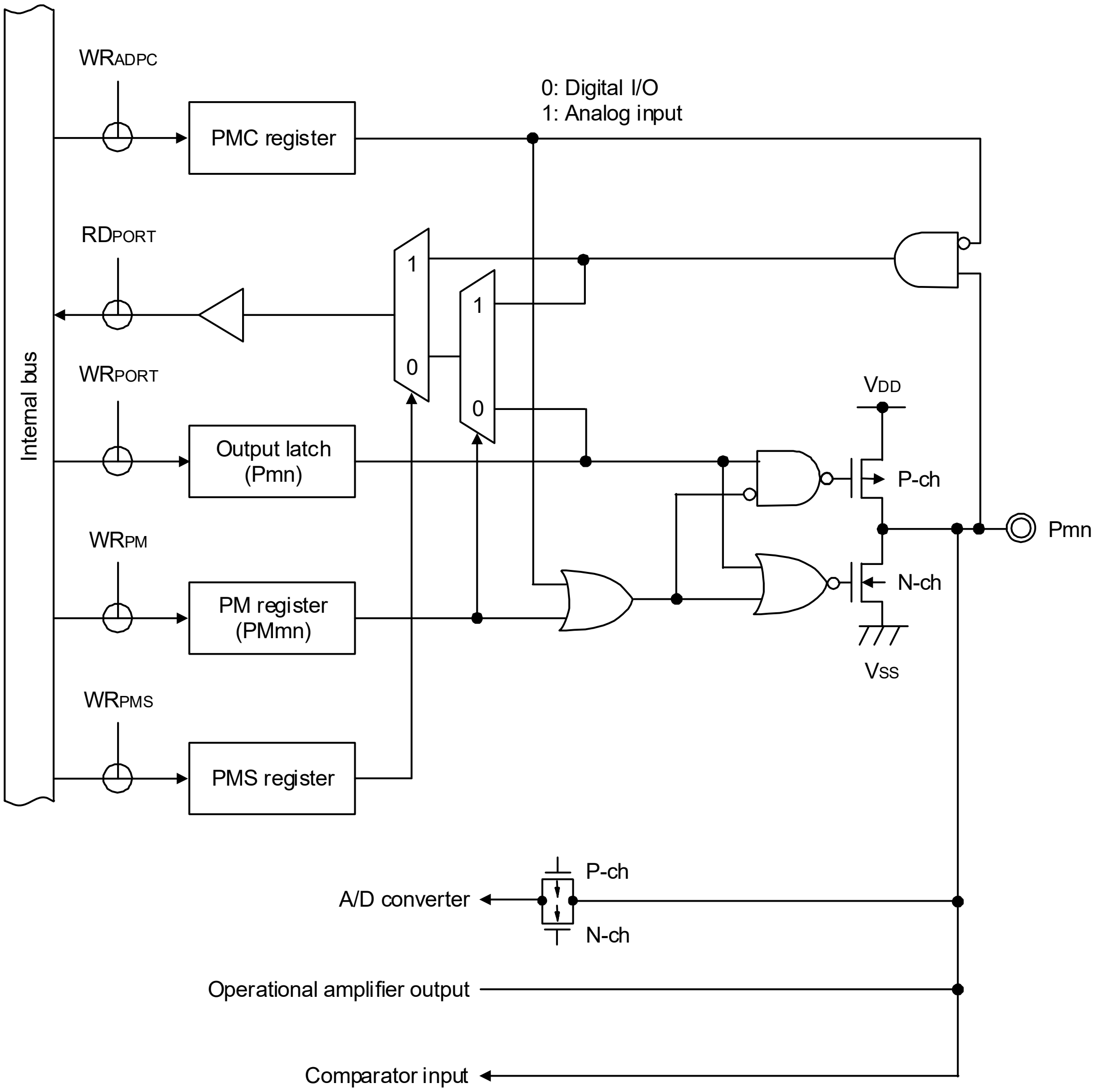
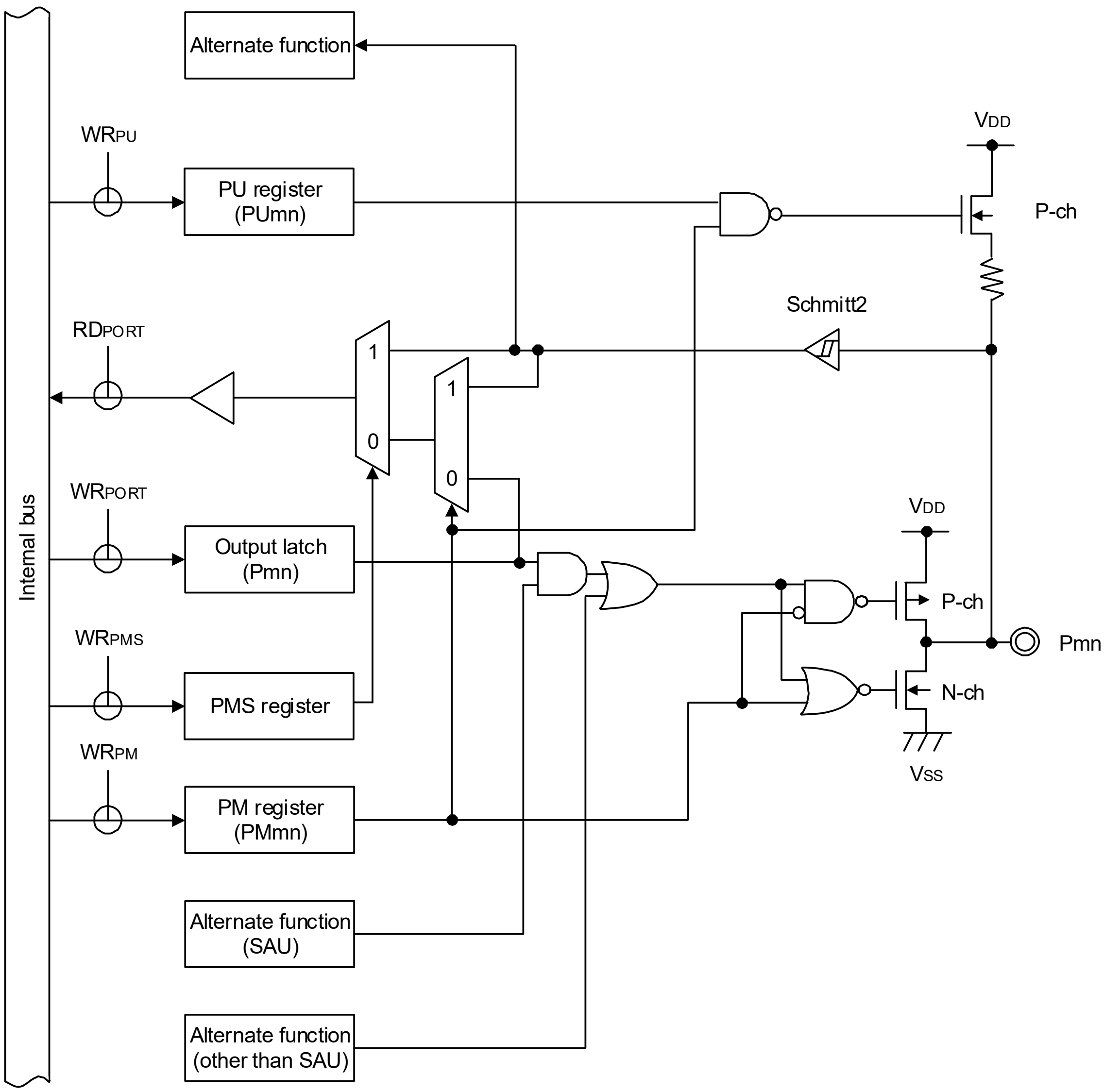
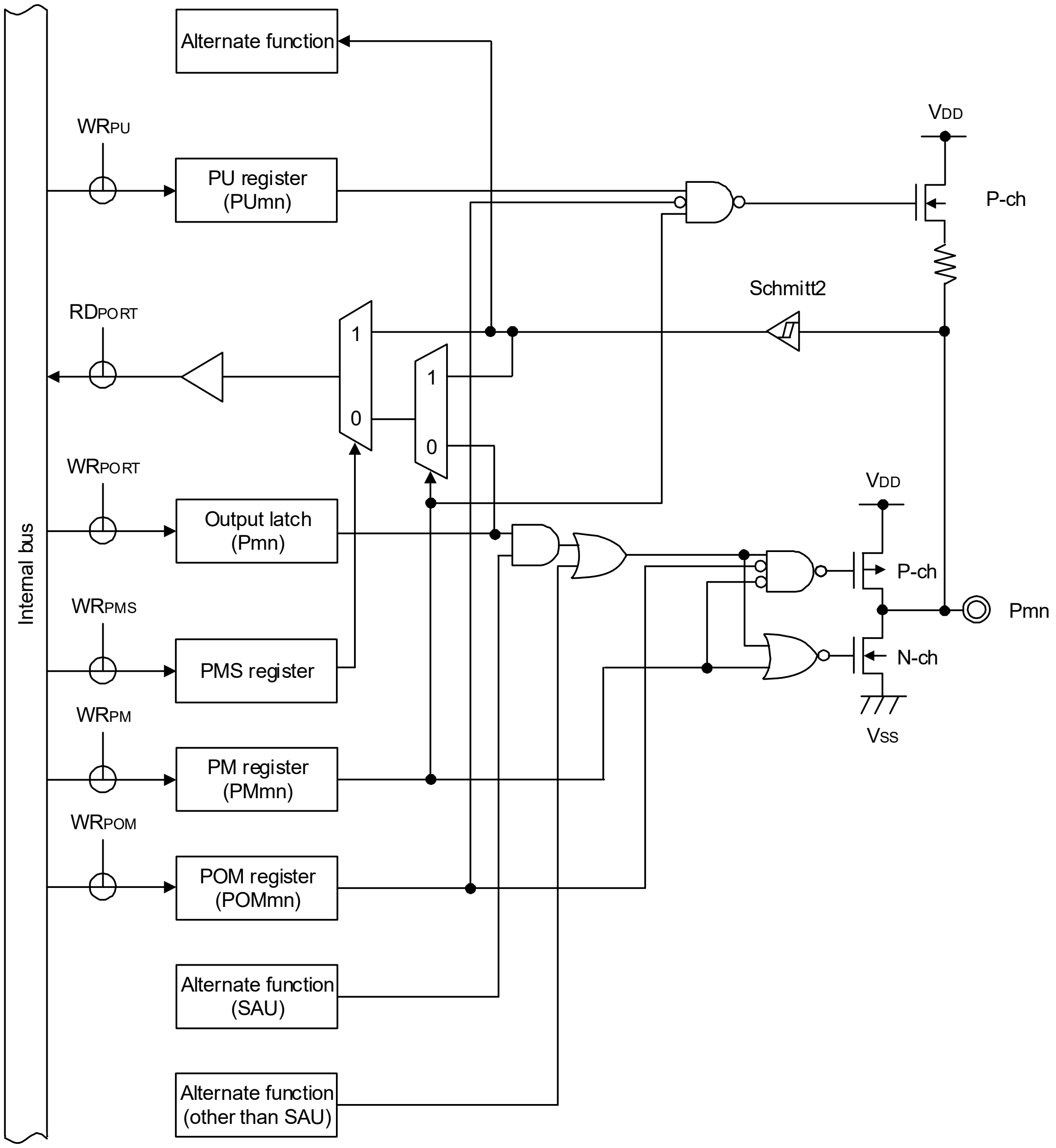


Figure 2 - 8 Pin Block Diagram of Pin Type 7-1-3



Remark 1. Refer to **2.1 Port Functions** for alternate functions.
Remark 2. SAU: Serial array unit

Figure 2 - 9 Pin Block Diagram of Pin Type 7-1-4

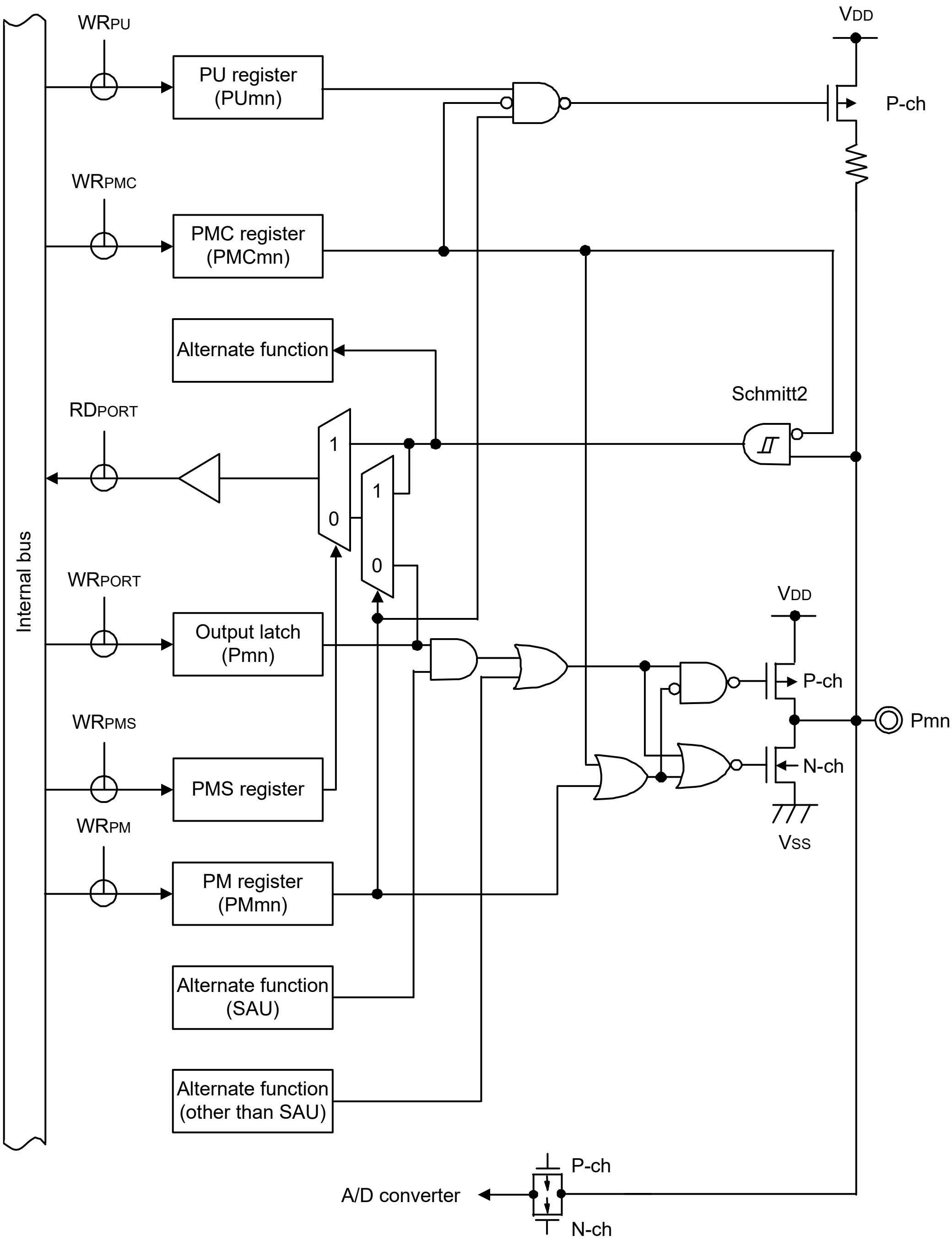


Caution **The input buffer is enabled even if the type 7-1-4 pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMx). This may lead to a through current flowing through the type 7-1-4 pin when the voltage level on this pin is intermediate. Changing the output level when the N-ch open drain output mode is selected may cause a glitch (VDD level).**

Remark 1. Refer to **2.1 Port Functions** for alternate functions.

Remark 2. SAU: Serial array unit

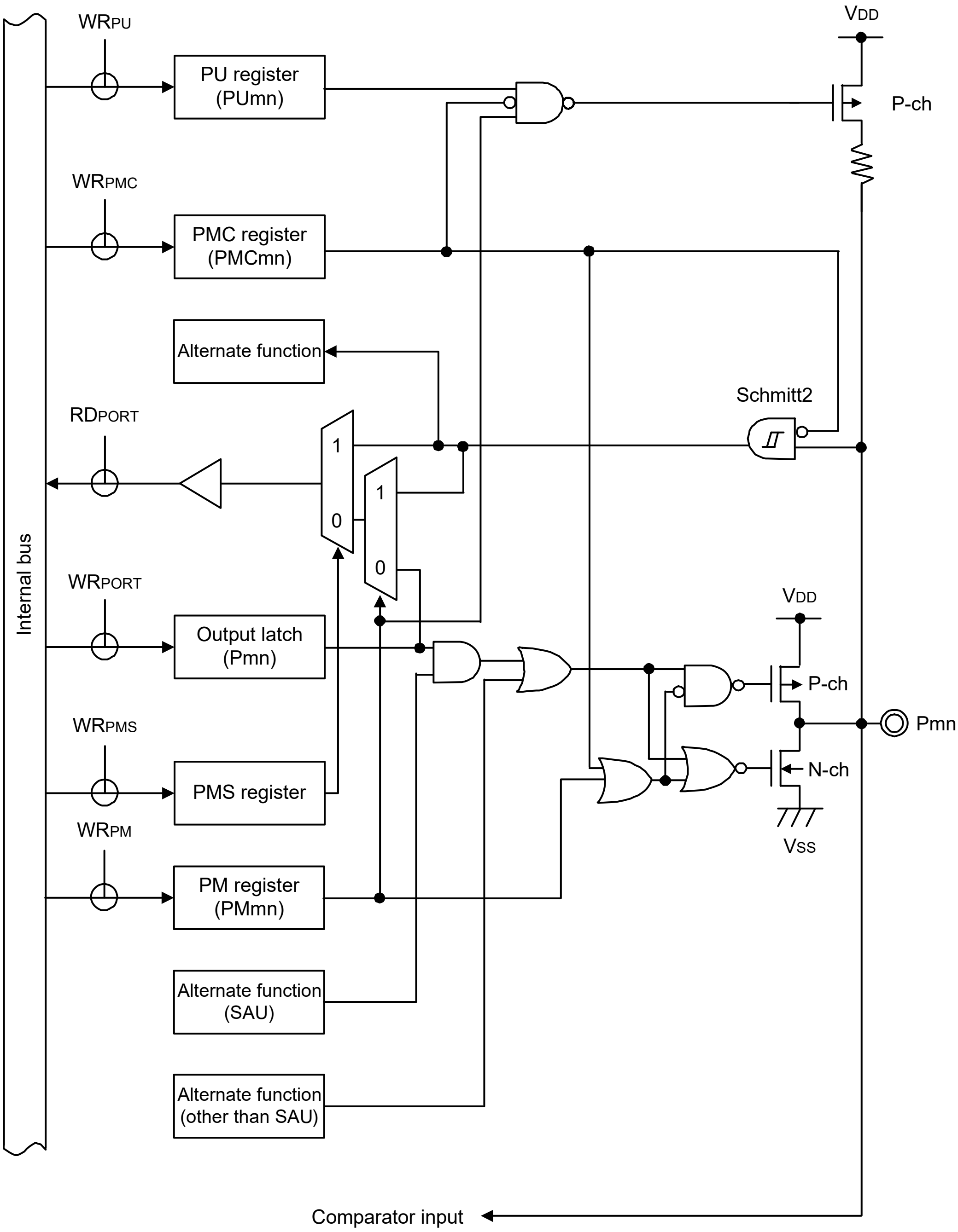
Figure 2 - 10 Pin Block Diagram of Pin Type 7-3-3



Remark 1. Refer to **2.1 Port Functions** for alternate functions.

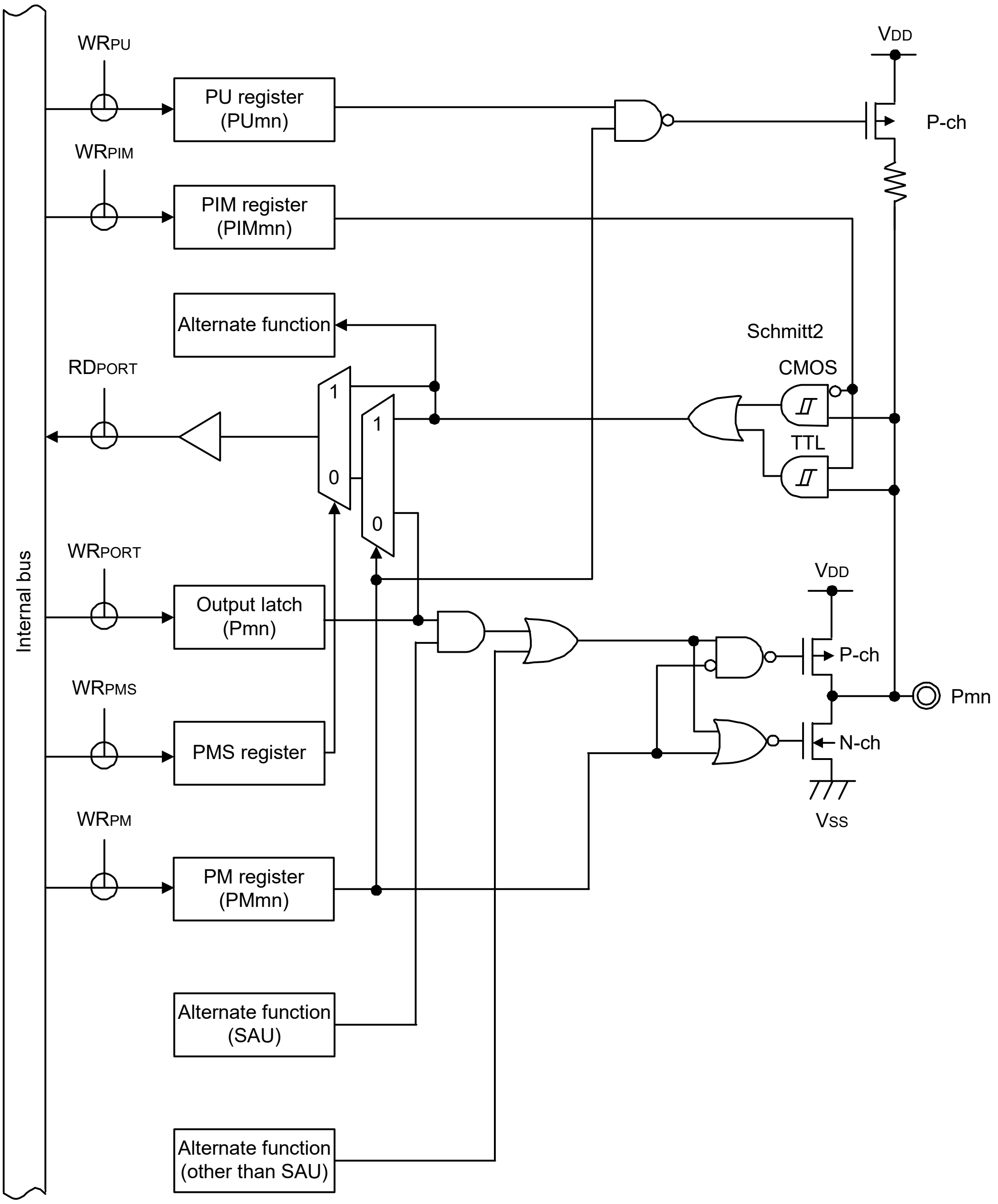
Remark 2. SAU: Serial array unit

Figure 2 - 11 Pin Block Diagram of Pin Type 7-6-2



Remark 1. Refer to **2.1 Port Functions** for alternate functions.
Remark 2. SAU: Serial array unit

Figure 2 - 12 Pin Block Diagram of Pin Type 8-1-3

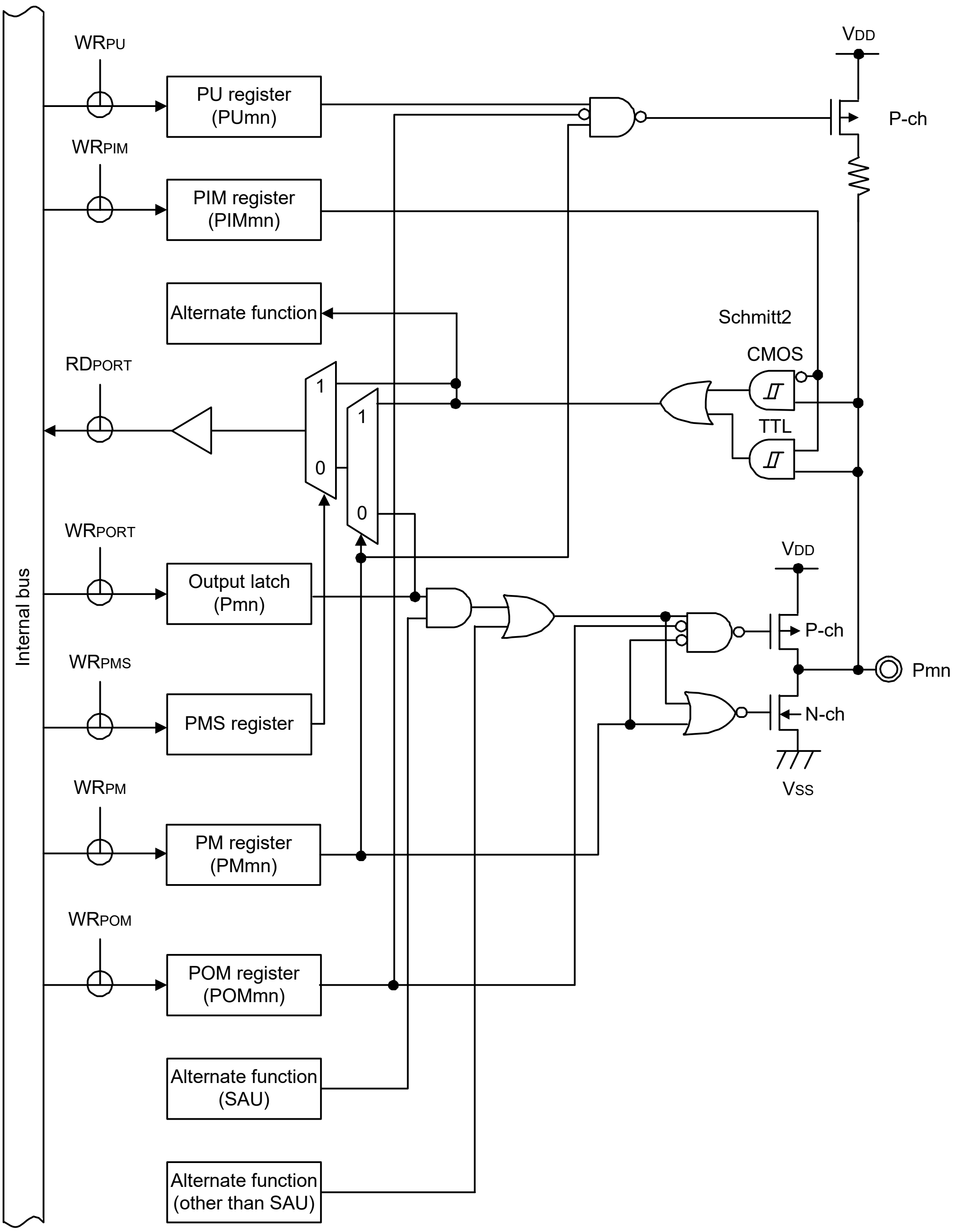


Caution **When the type 8-1-3 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMx) and is driven high, a through current may flow through the type 8-1-3 pin due to the configuration of the TTL input buffer. Drive the type 8-1-3 pin low to prevent the through current.**

Remark 1. Refer to **2.1 Port Functions** for alternate functions.

Remark 2. SAU: Serial array unit

Figure 2 - 13 Pin Block Diagram of Pin Type 8-1-4



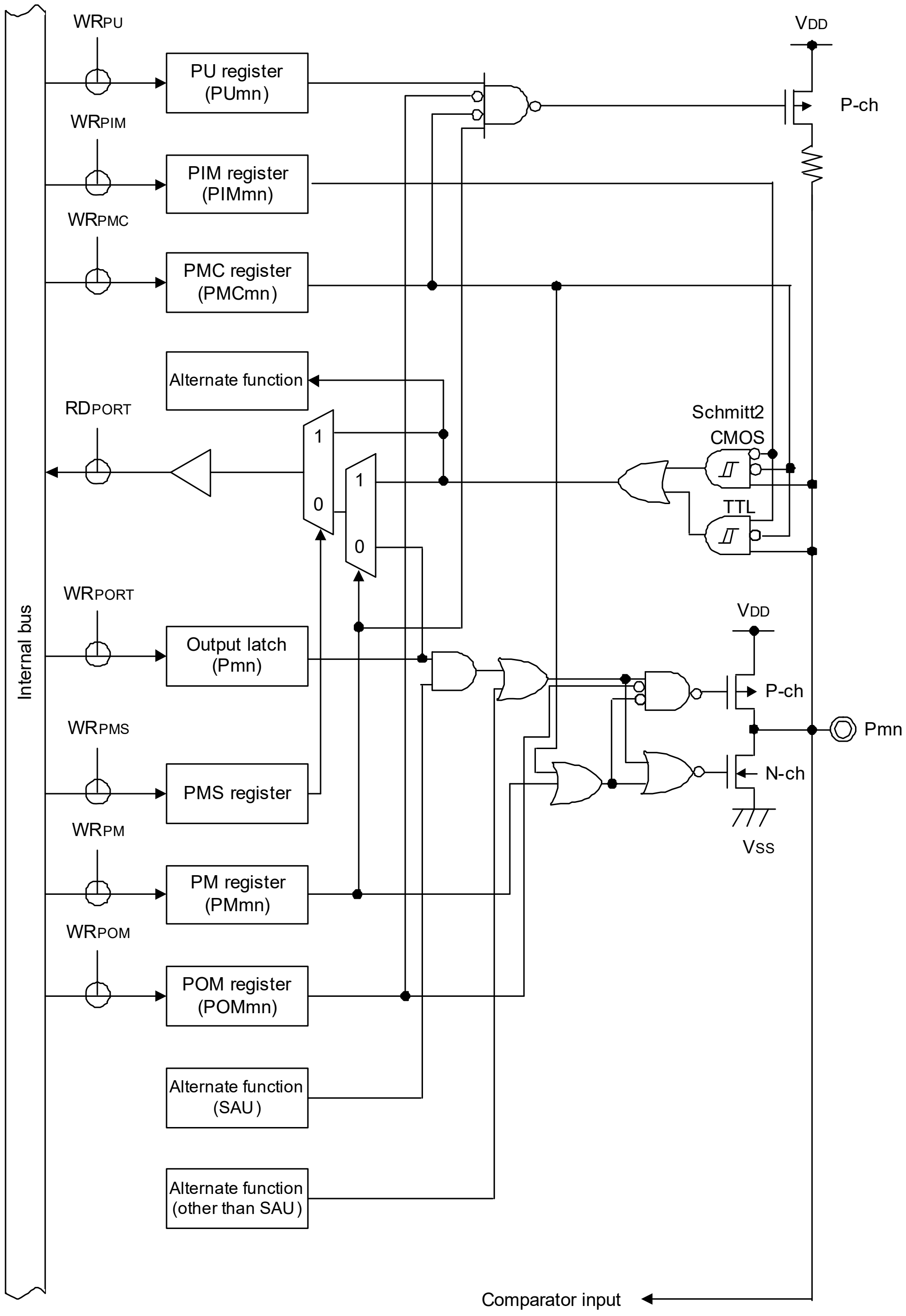
Caution 1. **The input buffer is enabled even if the type 8-1-4 pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMx). This may lead to a through current flowing through the type 8-1-4 pin when the voltage level on this pin is intermediate. Changing the output level when the N-ch open drain output mode is selected may cause a glitch (V_{DD} level).**

Caution 2. **When the type 8-1-4 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMx) and is driven high, a through current may flow through the type 8-1-4 pin due to the configuration of the TTL input buffer. Drive the type 8-1-4 pin low to prevent the through current.**

Remark 1. Refer to **2.1 Port Functions** for alternate functions.

Remark 2. SAU: Serial array unit

Figure 2 - 14 Pin Block Diagram of Pin Type 8-6-4



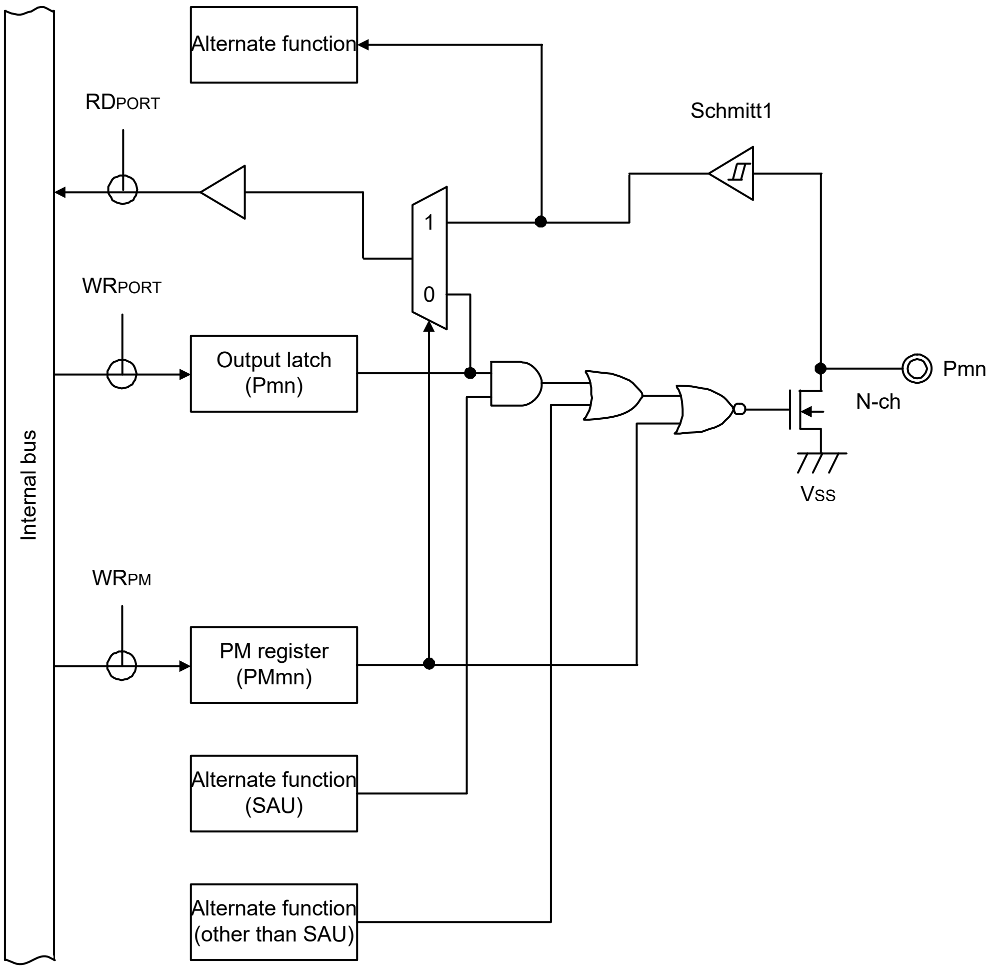
Caution 1. **The input buffer is enabled even if the type 8-6-4 pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMx). This may lead to a through current flowing through the type 8-6-4 pin when the voltage level on this pin is intermediate. Changing the output level when the N-ch open drain output mode is selected may cause a glitch (VDD level).**

Caution 2. **When the type 8-6-4 pin is set to TTL input buffer by the corresponding bit in the port input mode register (PIMx) and is driven high, a through current may flow through the type 8-6-4 pin due to the configuration of the TTL input buffer. Drive the type 8-6-4 pin low to prevent the through current.**

Remark 1. Refer to **2.1 Port Functions** for alternate functions.

Remark 2. SAU: Serial array unit

Figure 2 - 15 Pin Block Diagram of Pin Type 12-1-1



Caution **The input buffer is enabled even if the type 12-1-1 pin is operating as an output. This may lead to a through current flowing through the type 12-1-1 pin when the voltage level on this pin is intermediate.**

Remark 1. Refer to **2.1 Port Functions** for alternate functions.

Remark 2. SAU: Serial array unit

CHAPTER 3 CPU ARCHITECTURE

The RL78/I1D is a microcontroller that has the RL78-S3 CPU core.

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

- 3-stage pipeline CISC architecture
- Address space: 1 Mbyte
- Minimum instruction execution time: One instruction per clock cycle
- General-purpose registers: Eight 8-bit registers
- Type of instruction: 81

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)

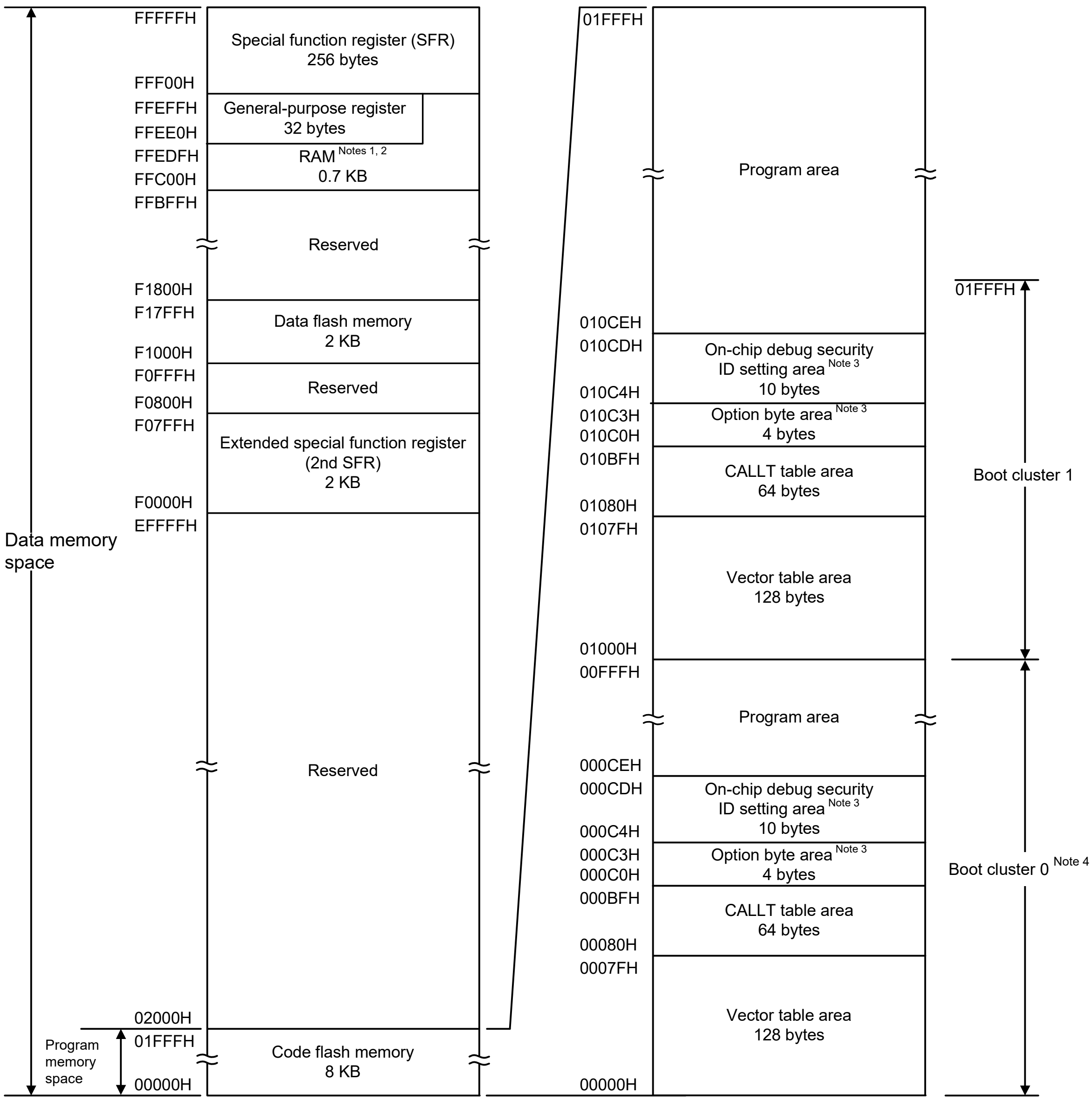
- Data allocation: Little endian

The RL78/I1D supports an OCD trace function.

3.1 Memory Space

Products in the RL78/I1D can access a 1 MB address space. Figures 3 - 1 to 3 - 3 show the memory maps.

Figure 3 - 1 Memory Map (R5F117x8 (x = 6, 7, A))



Note 1. Do not allocate the stack area, data buffers, branch destinations in the processing of vectored interrupts, or destinations or sources for DTC transfer to the area from FFE20H to FFEDFH when performing self-programming or rewriting of the data flash memory.

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

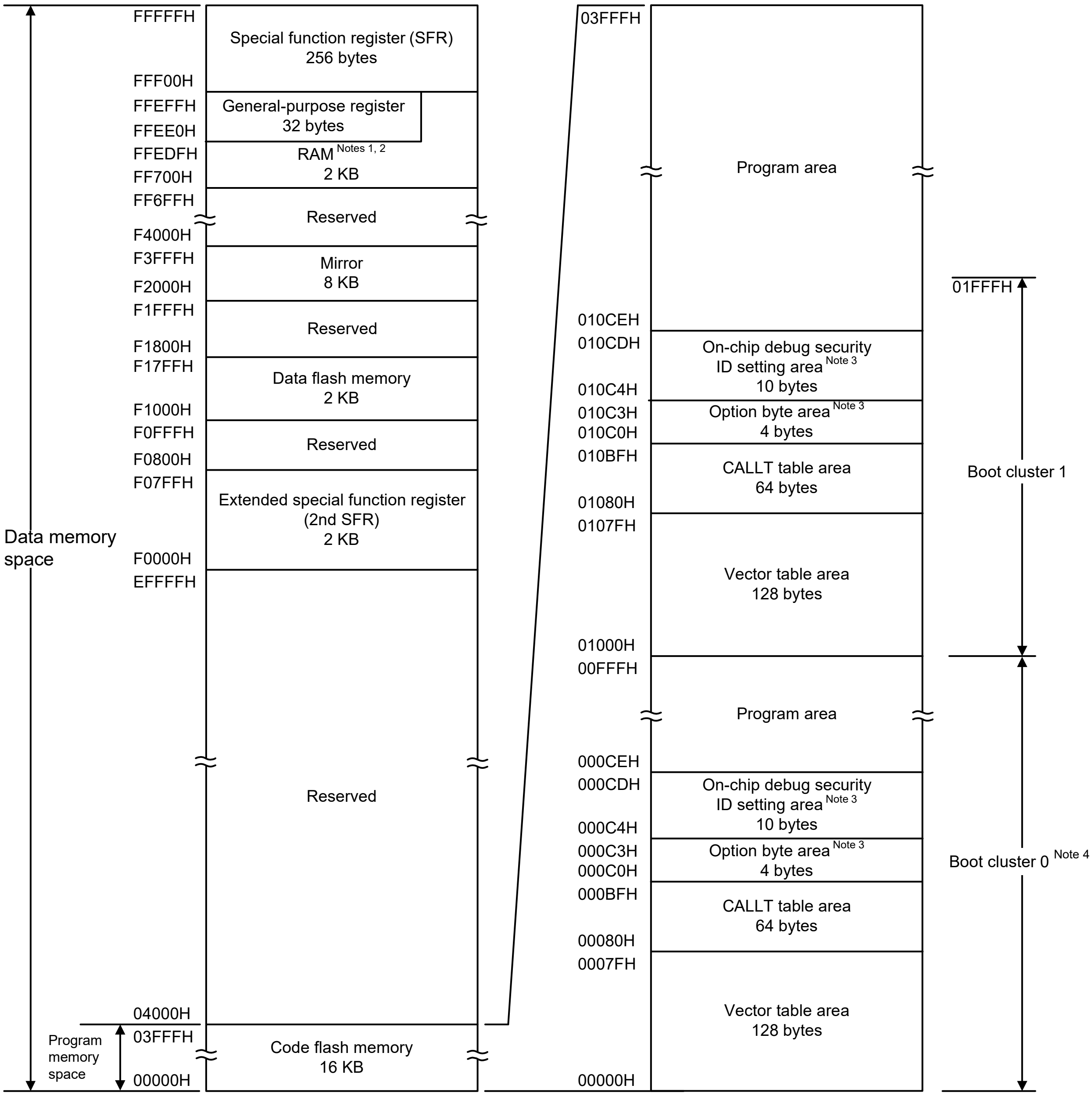
Note 3. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

When boot swap is used: Set the option bytes to 000C0H to 000C3H and 010C0H to 010C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 010C4H to 010CDH.

Note 4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **30.7 Security Settings**).

Caution **While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed and the RAM area + 10 bytes when instructions are fetched from RAM areas, respectively.**
Reset signal generation sets RAM parity error resets to enabled (RPERDIS = 0). For details, see 27.3.3 RAM parity error detection function.

Figure 3 - 2 Memory Map (R5F117xA (x = 6, 7, A, B, G))



Note 1. Do not allocate the stack area, data buffers for use by the flash library, arguments of library functions, branch destinations in the processing of vectored interrupts, or destinations or sources for DTC transfer to the area from FFE20H to FFEDFH when performing self-programming or rewriting of the data flash memory.The RAM area used by the flash library starts at FF700H. For the RAM areas used by the flash library, see **Self RAM list of Flash Self-Programming Library for RL78 Family (R20UT2944)**.

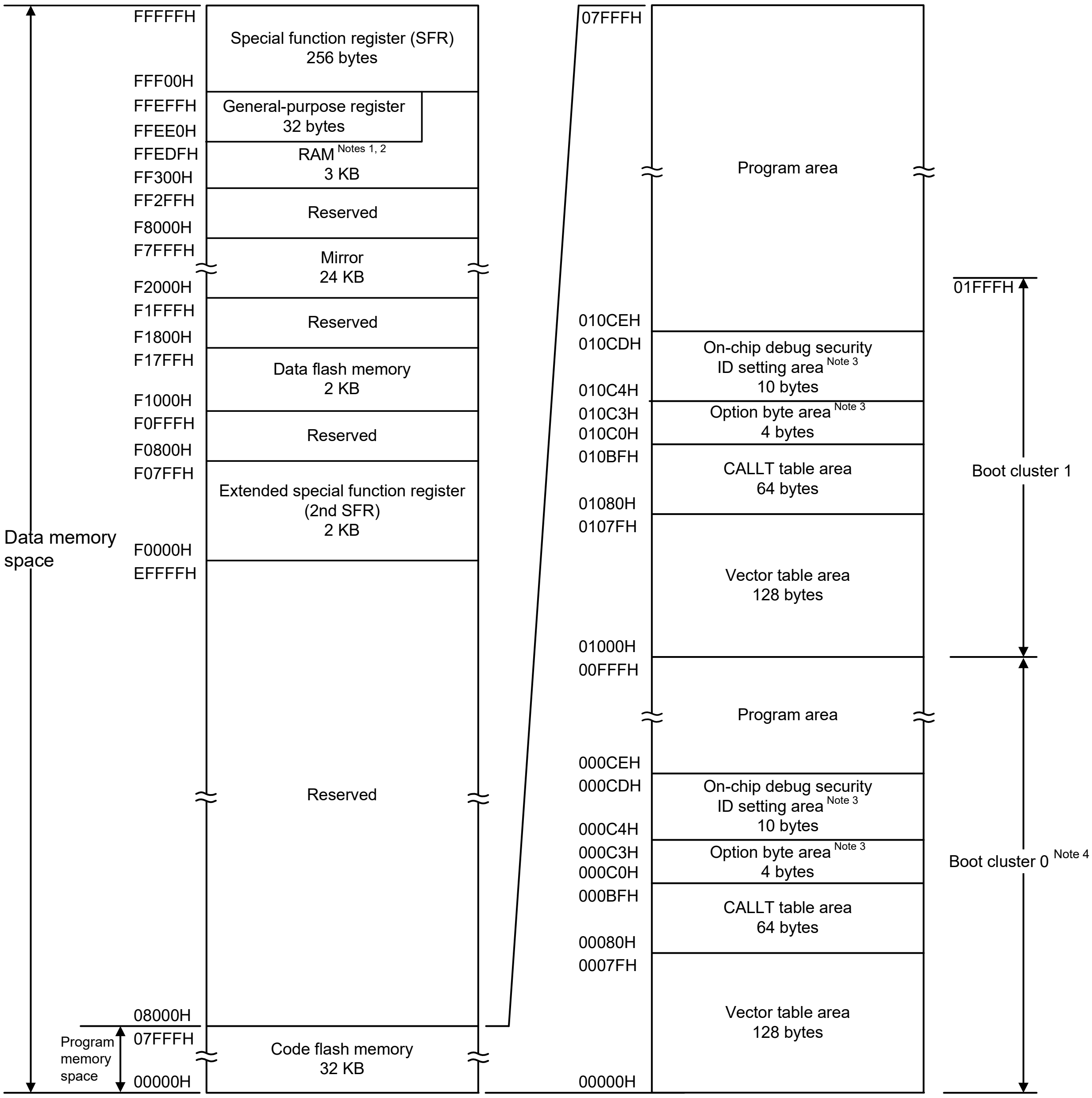
Note 2. Instructions can be executed from the RAM area excluding the general-purpose register area.

Note 3. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.
When boot swap is used: Set the option bytes to 000C0H to 000C3H and 010C0H to 010C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 010C4H to 010CDH.

Note 4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **30.7 Security Settings**).

Caution **While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed and the RAM area + 10 bytes when instructions are fetched from RAM areas, respectively.**
Reset signal generation sets RAM parity error resets to enabled (RPERDIS = 0). For details, see 27.3.3 RAM parity error detection function.

Figure 3 - 3 Memory Map (R5F117xC (x = A, B, G))



Note 1. Do not allocate the stack area, data buffers, branch destinations in the processing of vectored interrupts, or destinations or sources for DTC transfer to the area from FFE20H to FFEDFH when performing self-programming or rewriting of the data flash memory.
The RAM area used by the flash library starts at FF300H. For the RAM areas used by the flash library, see **Self RAM list of Flash Self-Programming Library for RL78 Family (R20UT2944)**.

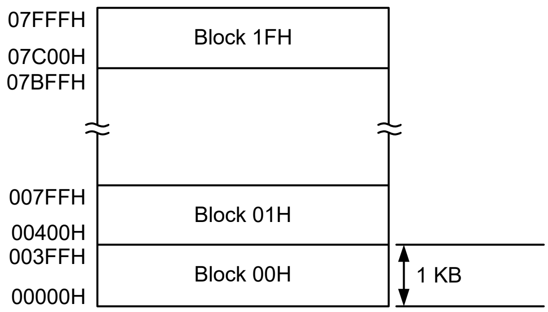
Note 2. Instructions can be executed from the RAM area excluding the general-purpose register area.

Note 3. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.
When boot swap is used: Set the option bytes to 000C0H to 000C3H and 010C0H to 010C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 010C4H to 010CDH.

Note 4. Writing boot cluster 0 can be prohibited depending on the setting of security (see **30.7 Security Settings**).

Caution **While RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed and the RAM area + 10 bytes when instructions are fetched from RAM areas, respectively.**
Reset signal generation sets RAM parity error resets to enabled (RPERDIS = 0). For details, see 27.3.3 RAM parity error detection function.

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Table 3 - 1 Correspondence Between Address Values and Block Numbers in Flash Memory**.



(R5F117xC (x = A, B, G))

Correspondence between the address values and block numbers in the flash memory are shown below.

Table 3 - 1 Correspondence Between Address Values and Block Numbers in Flash Memory

Address Value	Block Number	Address Value	Block Number	Address Value	Block Number	Address Value	Block Number
00000H to 003FFH	00H	02000H to 023FFH	08H	04000H to 043FFH	10H	06000H to 063FFH	18H
00400H to 007FFH	01H	02400H to 027FFH	09H	04400H to 047FFH	11H	06400H to 067FFH	19H
00800H to 00BFFH	02H	02800H to 02BFFH	0AH	04800H to 04BFFH	12H	06800H to 06BFFH	1AH
00C00H to 00FFFH	03H	02C00H to 02FFFH	0BH	04C00H to 04FFFH	13H	06C00H to 06FFFH	1BH
01000H to 013FFH	04H	03000H to 033FFH	0CH	05000H to 053FFH	14H	07000H to 073FFH	1CH
01400H to 017FFH	05H	03400H to 037FFH	0DH	05400H to 057FFH	15H	07400H to 077FFH	1DH
01800H to 01BFFH	06H	03800H to 03BFFH	0EH	05800H to 05BFFH	16H	07800H to 07BFFH	1EH
01C00H to 01FFFH	07H	03C00H to 03FFFH	0FH	05C00H to 05FFFH	17H	07C00H to 07FFFH	1FH

Remark R5F117x8 (x = 6, 7, A): Block numbers 00H to 07H
R5F117xA (x = 6, 7, A, B, G): Block numbers 00H to 0FH
R5F117xC (x = A, B, G): Block numbers 00H to 1FH

3.1.1 Internal program memory space

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

The RL78/I1D products incorporate internal ROM (flash memory), as shown below.

Table 3 - 2 Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
R5F117x8 (x = 6, 7, A)	Flash memory	8192 × 8 bits (00000H to 01FFFH)
R5F117xA (x = 6, 7, A, B, G)		16384 × 8 bits (00000H to 03FFFH)
R5F117xC (x = A, B, G)		32768 × 8 bits (00000H to 07FFFH)

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 01000H to 0107FH.

Table 3 - 3 lists the vector table. “√” indicates an interrupt source which is supported. “—” indicates an interrupt source which is not supported.

Table 3 - 3 Vector Table

Vector Table Address	Interrupt Source	48-pin	32-pin	30-pin	24-pin	20-pin
0000H	RESET, POR, LVD, WDT, TRAP, IAW, RPE	√	√	√	√	√
0004H	INTWDTI	√	√	√	√	√
0006H	INTLVI	√	√	√	√	√
0008H	INTP0	√	√	√	√	√
000AH	INTP1	√	√	√	√	√
000CH	INTP2	√	√	√	√	√
000EH	INTP3	√	√	√	√	—
0010H	INTP4	√	—	—	—	—
0012H	INTP5	√	—	√	—	—
0014H	INTP6	√	—	—	—	—
0016H	INTST0/INTCSI00/INTIIC00	√	√	√	√	√
0018H	INTSR0	√	√	√	√	√
	INTCSI01	√	√	—	√	—
	INTIIC01	√	√	—	√	—
001EH	INTSRE0	√	√	√	√	√
0020H	INTTM00	√	√	√	√	√
0022H	INTRTIT	√	√	√	—	—
0024H	INTFM	√	√	√	—	—
0026H	INTTM01H	√	√	√	√	√
0028H	INTTM03H	√	√	√	√	√
002AH	INTTM01	√	√	√	√	√
002CH	INTTM02	√	√	√	√	√
002EH	INTTM03	√	√	√	√	√
0034H	INTAD	√	√	√	√	√
0036H	INTRTC	√	√	√	√	√
0038H	INTIT	√	√	√	√	√
003AH	INTKR	√	√	—	√	—
003CH	INTCMP0	√	√	√	√	√
003EH	INTCMP1	√	√	√	√	√
0040H	INTDOC	√	√	√	√	√
0044H	INTTM00	√	√	√	√	√
0046H	INTTM01	√	√	√	√	√
0048H	INTTM10	√	√	√	√	√
004AH	INTIT11	√	√	√	√	√
0052H	INTFL	√	√	√	√	√
007EH	BRK	√	√	√	√	√

(2) CALLT instruction table area

The 64-byte area 00080H to 000BFH can store the subroutine entry address of a 2-byte call instruction (CALLT). Set the subroutine entry address to a value in a range of 00000H to 0FFFFH (because an address code is 2 bytes).

To use the boot swap function, set a CALLT instruction table also at 01080H to 010BFH.

(3) Option byte area

A 4-byte area of 000C0H to 000C3H can be used as an option byte area. Set the option byte at 010C0H to 010C3H when the boot swap is used. For details, see **CHAPTER 29 OPTION BYTE**.

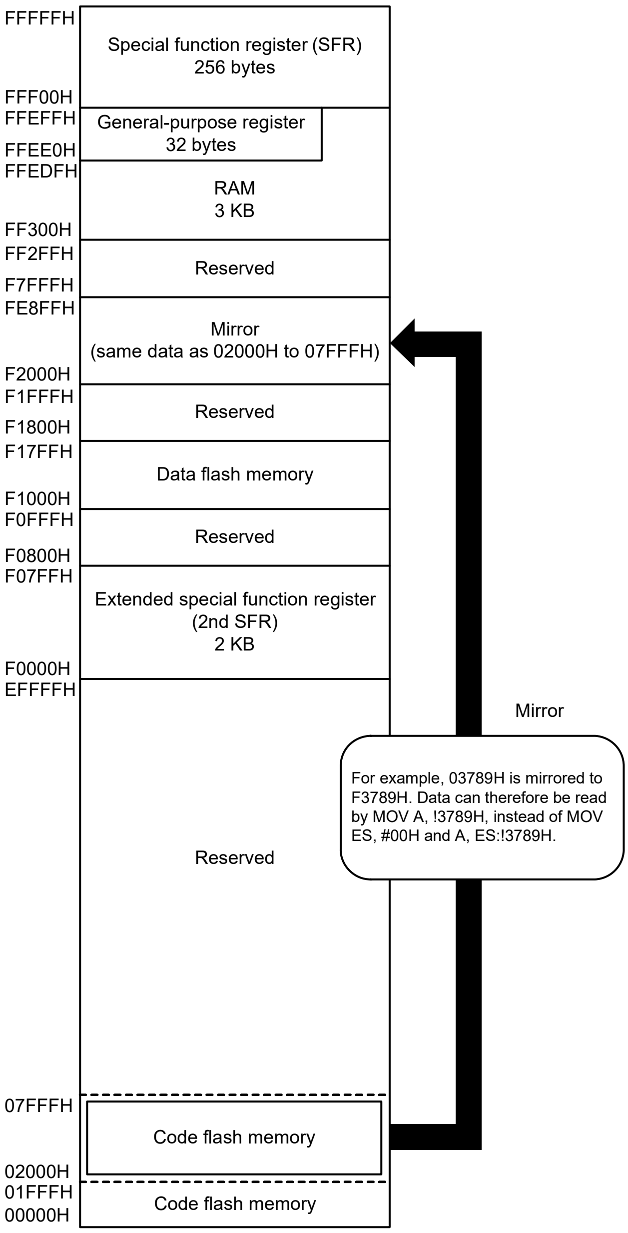
(4) On-chip debug security ID setting area

A 10-byte area of 000C4H to 000CDH and 010C4H to 010CDH can be used as an on-chip debug security ID setting area. Set the on-chip debug security ID of 10 bytes at 000C4H to 000CDH when the boot swap is not used and at 000C4H to 000CDH and at 010C4H to 010CDH when the boot swap is used. For details, see **CHAPTER 31 ON-CHIP DEBUG FUNCTION**.

3.1.2 Mirror area

The RL78/I1D mirrors the code flash area of 00000H to 0FFFFH, to F0000H to FFFFFH (the code flash area to be mirrored is set by the processor mode control register (PMC)).
By reading data from F0000H to FFFFFH, an instruction that does not have the ES register as an operand can be used, and thus the contents of the code flash can be read with the shorter code. However, the code flash area is not mirrored to the special function register (SFR), extended special function register (2nd SFR), RAM, data flash memory, and use prohibited areas.
See **3.1 Memory Space** for the mirror area of each product.
The mirror area can only be read and no instruction can be fetched from this area.
The following show examples.

Example R5F117xC (x = A, B, G) (Flash memory: 32 KB, RAM: 3 KB)



The PMC register is described below.

• Processor mode control register (PMC)

This register sets the flash memory space for mirroring to area from F0000H to FFFFFH.

The PMC register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 00H.

Figure 3 - 4 Format of Configuration of Processor mode control register (PMC)

Address: FFFFEH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
PMC	0	0	0	0	0	0	0	MAA

MAA	Selection of flash memory space for mirroring to area from F0000H to FFFFFH
0	00000H to 0FFFFH is mirrored to F0000H to FFFFFH
1	Setting prohibited

Caution 1. Be sure to clear bit 0 (MAA) of this register to 0 (default value).

Caution 2. After setting the PMC register, wait for at least one instruction and access the mirror area.

3.1.3 Internal data memory space

The RL78/I1D products incorporate the following RAMs.

Table 3 - 4 Internal RAM Capacity

Part Number	Internal RAM
R5F117x8 (x = 6, 7, A)	768 × 8 bits (FFC00H to FFEFFH)
R5F117xA (x = 6, 7, A, B, G)	2048 × 8 bits (FF700H to FFEFFH)
R5F117xC (x = A, B, G)	3096 × 8 bits (FF300H to FFEFFH)

The internal RAM can be used as a data area and a program area where instructions are fetched (it is prohibited to use the general-purpose register area for fetching instructions). 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.

Caution 1. It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or as a stack area.

Caution 2. Do not allocate the stack area, data buffers, branch destinations in the processing of vectored interrupts, or destinations or sources for DTC transfer to the area from FFE20H to FFEDFH when performing self-programming or rewriting of the data flash memory.

Caution 3. The flash library uses RAM in self-programming and rewriting of the data flash memory.
The target products and start address of the RAM areas used by the flash library are shown below.
R5F117xC (x = A, B, G): Start address FF300H
For the RAM areas used by the flash library, see Self RAM list of Flash Self-Programming Library for RL78 Family (R20UT2944).

Caution 4. The internal RAM area in the following products cannot be used as stack memory when using the on-chip debugging trace function.
R5F117xC (x = A, B, G): FF700H to FF8FFH

3.1.4 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area FFF00H to FFFFFH (see **Tables 3 - 5** to **3 - 7** in **3.2.4 Special function registers (SFRs)**).

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

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

On-chip peripheral hardware special function registers (2nd SFRs) are allocated in the area F0000H to F07FFH (see **Tables 3 - 8** to **3 - 12** in **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**).

Caution 1. Do not access addresses to which extended SFRs are not assigned.

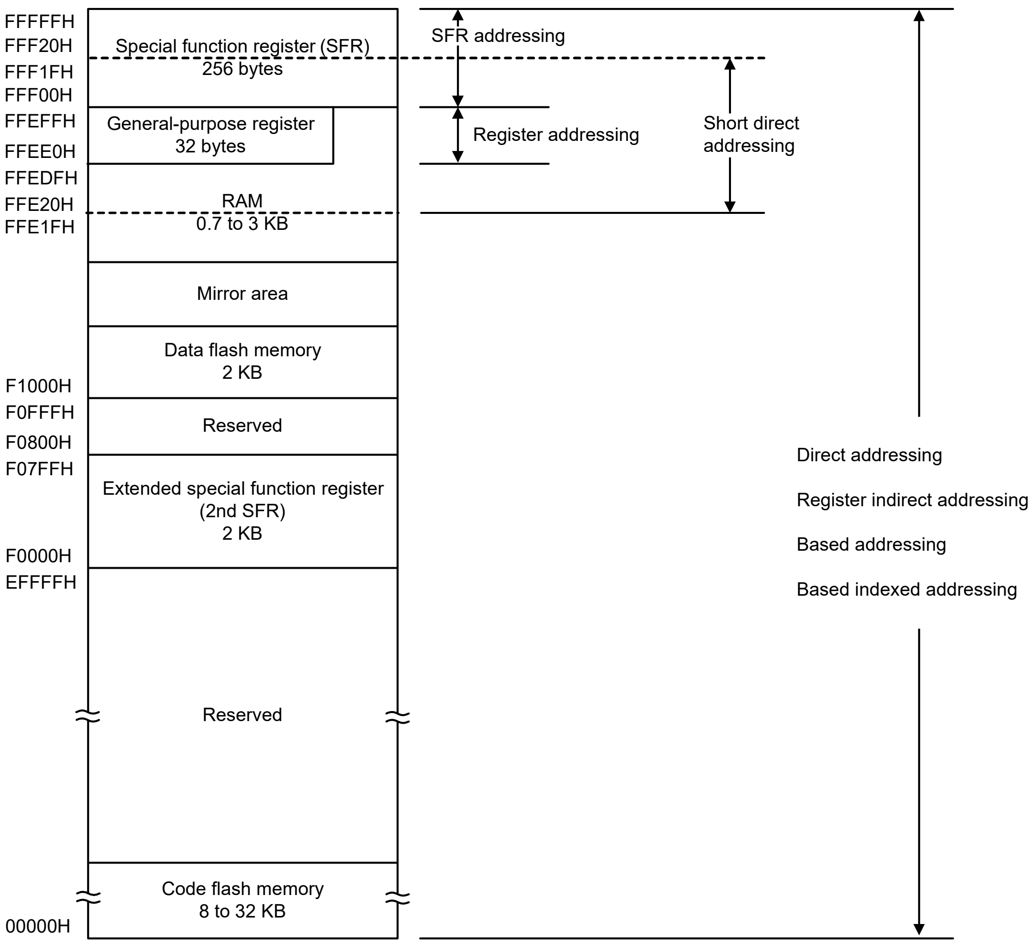
Caution 2. When accessing 8-bit interval timer counter register 0 (TRT0), 8-bit interval timer counter register 1 (TRT1), DOC control register (DOCR), DOC data input register (DODIR), and DOC data setting register (DODSR) allocated in F0500H to F0515H of the extended SFR (2nd SFR), the CPU does not proceed to the next instruction processing but enters the wait state for CPU processing. For this reason, if this wait state occurs, the number of instruction execution clocks is increased by the number of wait clocks. The number of wait clocks for access to 8-bit interval timer counter register 0 (TRT0), 8-bit interval timer counter register 1 (TRT1), DOC control register (DOCR), DOC data input register (DODIR), and DOC data setting register (DODSR) is one clock for both writing and reading.

3.1.6 Data memory addressing

Addressing refers to the method of specifying the address of the instruction to be executed next or the address of the register or memory relevant to the execution of instructions.

Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the RL78/I1D, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of the special function registers (SFR) and general-purpose registers are available for use. Figure 3 - 5 shows correspondence between data memory and addressing. For details of each addressing, see **3.4 Addressing for Processing Data Addresses**.

Figure 3 - 5 Correspondence Between Data Memory and Addressing



3.2 Processor Registers

The RL78/I1D products incorporate the following processor registers.

3.2.1 Control registers

The control registers control the program sequence, statuses and stack memory. The control registers consist of a program counter (PC), a program status word (PSW) and a stack pointer (SP).

(1) Program counter (PC)

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

In normal operation, PC is automatically incremented according to the number of bytes of the instruction to be fetched. When a branch instruction is executed, immediate data and register contents are set.

Reset signal generation sets the reset vector table values at addresses 0000H and 0001H to the program counter.

Figure 3 - 6 Format of Program Counter

	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. Reset signal generation sets the PSW register to 06H.

Figure 3 - 7 Format of Program Status Word

	7							0
PSW	IE	Z	RBS1	AC	RBS0	ISP1	ISP0	CY

(a) Interrupt enable flag (IE)

This flag controls the interrupt request acknowledge operations of the CPU.

When 0, the IE flag is set to the interrupt disabled (DI) state, and all maskable interrupt requests are disabled.

When 1, the IE flag is set to the interrupt enabled (EI) state and interrupt request acknowledgment is controlled with an in-service priority flag (ISP1, ISP0), an interrupt mask flag for various interrupt sources, and a priority specification flag.

The IE flag is reset (0) upon DI instruction execution or interrupt acknowledgment and is set (1) upon EI instruction execution.

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

(c) Register bank select flags (RBS0, RBS1)
These are 2-bit flags to select one of the four register banks.
In these flags, the 2-bit information that indicates the register bank selected by SEL RBn instruction execution is stored.

(d) Auxiliary carry flag (AC)
If the operation result has a carry from bit 3 or a borrow at bit 3, this flag is set (1). It is reset (0) in all other cases.

(e) In-service priority flags (ISP1, ISP0)
This flag manages the priority of acknowledgeable maskable vectored interrupts. Vectored interrupt requests specified lower than the value of ISP0 and ISP1 flags by the priority specification flag registers (PRn0L, PRn0H, PRn1L, PRn1H, PRn2L, PRn2H) (see **21.3.3**) can not be acknowledged. Actual request acknowledgment is controlled by the interrupt enable flag (IE).

Remark n = 0, 1

(f) Carry flag (CY)
This flag stores overflow and underflow upon add/subtract instruction execution. It stores the shift-out value upon rotate instruction execution and functions as a bit accumulator during bit operation instruction execution.

(3) Stack pointer (SP)
This is a 16-bit register to hold the start address of the memory stack area. Only the internal RAM area can be set as the stack area.

Figure 3 - 8 Format of Stack Pointer

	15															0
SP	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SP7	SP6	SP5	SP4	SP3	SP2	SP1	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 reset signal generation makes the SP contents undefined, be sure to initialize the SP before using the stack.

Caution 2. It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or as a stack area.

Caution 3. Do not allocate RAM addresses which are used as a stack area, a data buffer, a branch destination of vector interrupt processing, and a DTC transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

Caution 4. Use of the RAM areas of the following products is prohibited when performing self-programming and rewriting the data flash memory, because these areas are used for each library.
R5F117xC (x = A, B, G): FF300H to FF709H

Caution 5. The internal RAM area in the following products cannot be used as stack memory when using the on-chip debugging trace function.
R5F117xC (x = A, B, G): FF700H to FF8FFH

3.2.2 General-purpose registers

General-purpose registers are mapped at particular addresses (FFEE0H to FFEFFH) of the data memory. The general-purpose registers consists of 4 banks, each bank consisting of eight 8-bit registers (X, A, C, B, E, D, L, and H).

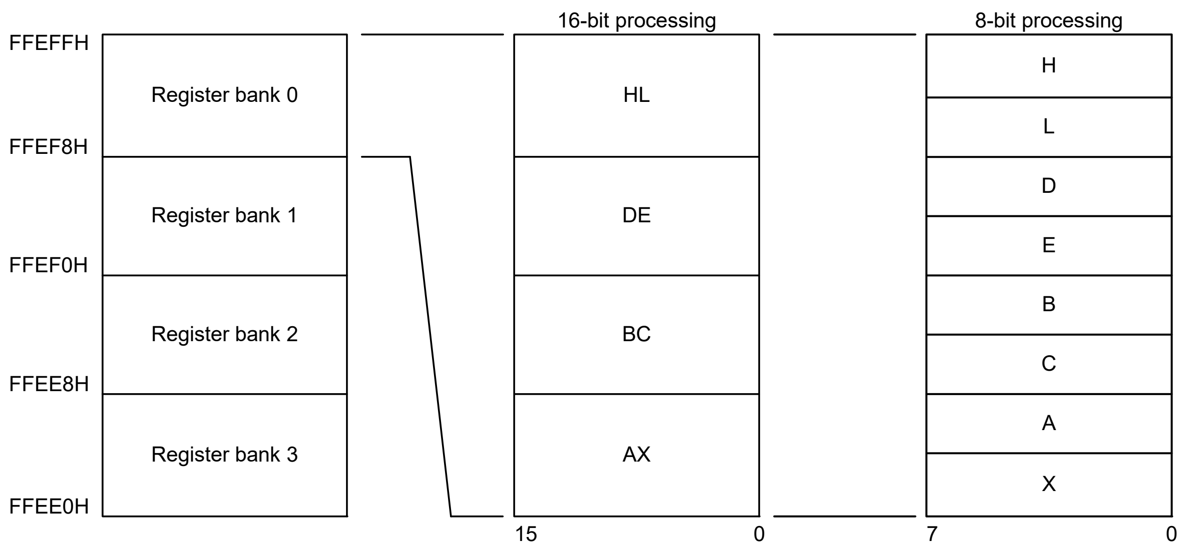
Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

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

Figure 3 - 9 Configuration of General-Purpose Registers

(a) 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 direct addressing), respectively.

The default value of the ES register after reset is 0FH, and that of the CS register is 00H.

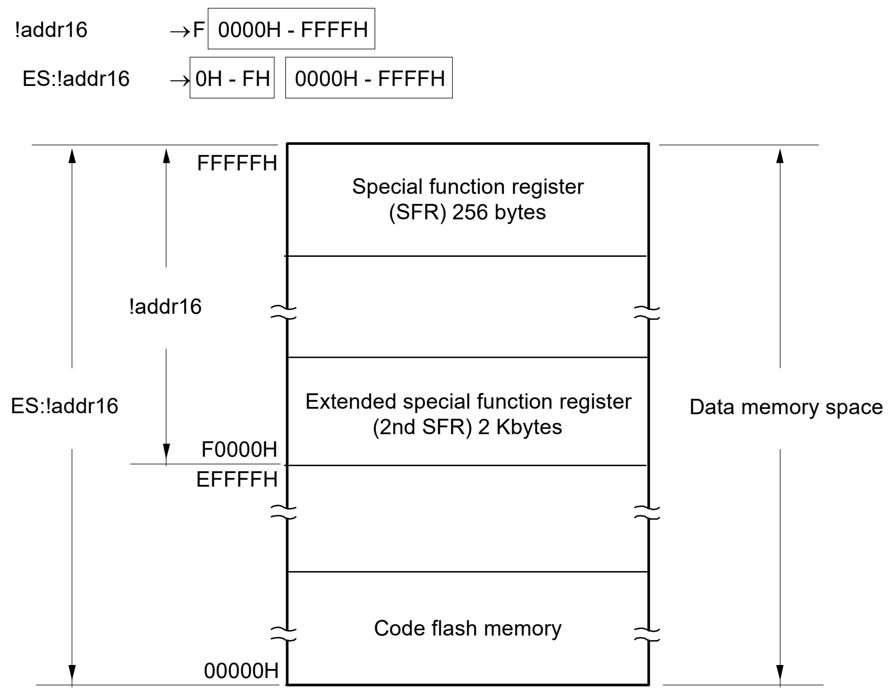
Figure 3 - 10 Configuration of ES and CS Registers

	7	6	5	4	3	2	1	0
ES	0	0	0	0	ES3	ES2	ES1	ES0

	7	6	5	4	3	2	1	0
CS	0	0	0	0	CS3	CS2	CS1	CS0

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 - 11 Extension of Data Area Which Can Be Accessed



3.2.4 Special function registers (SFRs)

Unlike a general-purpose register, each SFR has a special function.
SFRs are allocated to the FFF00H to FFFFFH area.
SFRs can be manipulated like general-purpose registers, using operation, transfer, and bit manipulation instructions. The manipulable bit units, 1, 8, and 16, depend on the SFR type.
Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describe as follows for the 1-bit 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 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 manipulation instruction operand (sfrp). When specifying an address, describe an even address.

Tables 3 - 5 to 3 - 7 give 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
  This item indicates each register status upon reset signal generation.

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

Remark For extended SFRs (2nd SFRs), see **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

Table 3 - 5 Special Function Register (SFR) List (1/3)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF00H	Port register 0	P0		R/W	√	√	—	00H
FFF01H	Port register 1	P1		R/W	√	√	—	00H
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
FFF0CH	Port register 12	P12		R/W	√	√	—	Undefined
FFF0DH	Port register 13	P13		R/W	√	√	—	Undefined
FFF10H	Serial data register 00	TXD0/ SIO00	SDR00	R/W	—	√	√	0000H
FFF11H		—			—	—		
FFF12H	Serial data register 01	RXD0/ SIO01	SDR01	R/W	—	√	√	0000H
FFF13H		—			—	—		
FFF18H	Timer data register 00	TDR00		R/W	—	—	√	0000H
FFF19H								
FFF1AH	Timer data register 01	TDR01L	TDR01	R/W	—	√	√	00H
FFF1BH		TDR01H			—	√		00H
FFF1EH	12-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
FFF2DH	Port mode register 13	PM13		R/W	√	√	—	FFH
FFF30H	A/D converter mode register 0	ADM0		R/W	√	√	—	00H
FFF31H	Analog input channel specification register	ADS		R/W	√	√	—	00H
FFF32H	A/D converter mode register 1	ADM1		R/W	√	√	—	00H
FFF34H	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

Table 3 - 6 Special Function Register (SFR) List (2/3)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF64H	Timer data register 02	TDR02		R/W	—	—	√	0000H
FFF65H								
FFF66H	Timer data register 03	TDR03L	TDR03	R/W	—	√	√	00H
FFF67H		TDR03H			—	√		00H
FFF90H	12-bit interval timer control register	ITMC		R/W	—	—	√	0FFFH
FFF91H								
FFF92H	Second count register	SEC		R/W	—	√	—	00H
FFF93H	Minute count register	MIN		R/W	—	√	—	00H
FFF94H	Hour count register	HOUR		R/W	—	√	—	12H Note
FFF95H	Week count register	WEEK		R/W	—	√	—	00H
FFF96H	Day count register	DAY		R/W	—	√	—	01H
FFF97H	Month count register	MONTH		R/W	—	√	—	01H
FFF98H	Year count register	YEAR		R/W	—	√	—	00H
FFF9AH	Alarm minute register	ALARMWM		R/W	—	√	—	00H
FFF9BH	Alarm hour register	ALARMWH		R/W	—	√	—	12H
FFF9CH	Alarm week register	ALARMWW		R/W	—	√	—	00H
FFF9DH	Real-time clock control register 0	RTCC0		R/W	√	√	—	00H
FFF9EH	Real-time clock control register 1	RTCC1		R/W	√	√	—	00H
FFFA0H	Clock operation mode control register	CMC		R/W	—	√	—	00H
FFFA1H	Clock operation status control register	CSC		R/W	√	√	—	C0H
FFFA2H	Oscillation stabilization time counter status register	OSTC		R	√	√	—	00H
FFFA3H	Oscillation stabilization time select register	OSTS		R/W	—	√	—	07H
FFFA4H	System clock control register	CKC		R/W	√	√	—	00H
FFFA5H	Clock output select register 0	CKS0		R/W	√	√	—	00H
FFFA6H	Clock output select register 1	CKS1		R/W	√	√	—	00H
FFFA7H	Subsystem clock select register	CKSEL		R/W	√	√	—	00H

Note The value of this register is 00H if the AMPM bit (bit 3 of real-time clock control register 0 (RTCC0)) is set to 1 after reset.

Table 3 - 7 Special Function Register (SFR) List (3/3)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFFA8H	Reset control flag register	RESF		R	—	√	—	Undefined Note 1
FFFA9H	Voltage detection register	LVIM		R/W	√	√	—	00H Note 1
FFFAAH	Voltage detection level register	LVIS		R/W	√	√	—	Note 3
FFFABH	Watchdog timer enable register	WDTE		R/W	—	√	—	1AH/9AH Note 2
FFFACH	CRC input register	CRCIN		R/W	—	√	—	00H
FFFD0H	Interrupt request flag register 2L	IF2L		R/W	√	√	—	00H
FFFD4H	Interrupt mask flag register 2L	MK2L		R/W	√	√	—	FFH
FFFD8H	Priority specification flag register 02L	PR02L		R/W	√	√	—	FFH
FFFDCH	Priority specification flag register 12L	PR12L		R/W	√	√	—	FFH
FFFE0H	Interrupt request flag register 0L	IF0L	IF0	R/W	√	√	√	00H
FFFE1H	Interrupt request flag register 0H	IF0H		R/W	√	√		00H
FFFE2H	Interrupt request flag register 1L	IF1L	IF1	R/W	√	√	√	00H
FFFE3H	Interrupt request flag register 1H	IF1H		R/W	√	√		00H
FFFE4H	Interrupt mask flag register 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
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	Multiply and accumulation register (L)	MACRL		R/W	—	—	√	0000H
FFFF1H								
FFFF2H	Multiply and accumulation register (H)	MACRH		R/W	—	—	√	0000H
FFFF3H								
FFFFEH	Processor mode control register	PMC		R/W	√	√	—	00H

Note 1. These values vary depending on the reset source.

<table>
<tr><th colspan="2">Reset Source / Register</th><th>$\overline{\text{RESET}}$ Input</th><th>Reset by POR</th><th>Reset by Execution of Illegal Instruction</th><th>Reset by WDT</th><th>Reset by RAM parity error</th><th>Reset by illegal-memory access</th><th>Reset by LVD</th></tr>
<tr><td rowspan="5">RESF</td><td>TRAP</td><td colspan="2" rowspan="5">Cleared (0)</td><td>Set (1)</td><td colspan="3">Held</td><td rowspan="4">Held</td></tr>
<tr><td>WDTRF</td><td>Held</td><td>Set (1)</td><td colspan="2">Held</td></tr>
<tr><td>RPERF</td><td colspan="2">Held</td><td>Set (1)</td><td>Held</td></tr>
<tr><td>IAWRF</td><td colspan="3">Held</td><td>Set (1)</td></tr>
<tr><td>LVIRF</td><td colspan="4">Held</td><td>Set (1)</td></tr>
<tr><td rowspan="3">LVIM</td><td>LVISEN</td><td colspan="6">Cleared (0)</td><td rowspan="3">Held</td></tr>
<tr><td>LVIOMSK</td><td colspan="6" rowspan="2">Held</td></tr>
<tr><td>LVIF</td></tr>
</table>

Note 2. The reset value of the WDTE register is determined by the setting of the option byte.
Note 3. The reset value of the LVIS register is determined by the setting of the option byte.

Remark For extended SFRs (2nd SFRs), see **Tables 3 - 8** to **3 - 12 Extended Special Function Register (2nd SFR) List**.

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

Unlike a general-purpose register, each extended SFR (2nd SFR) has a special function.
Extended SFRs are allocated to the F0000H to F07FFH area. SFRs other than those in the SFR area (FFF00H to FFFFFH) are allocated to this area. An instruction that accesses the extended SFR area, however, is 1 byte longer than an instruction that accesses the SFR area.
Extended SFRs can be manipulated like general-purpose registers, using operation, transfer, and bit manipulation instructions. The manipulable bit units, 1, 8, and 16, depend on the SFR type.
Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describe as follows for the 1-bit 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 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 manipulation instruction operand (!addr16). When specifying an address, describe an even address.

Tables 3 - 8 to 3 - 12 give lists of the extended SFRs. The meanings of items in the table are as follows.

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

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

Remark For SFRs in the SFR area, see **3.2.4 Special function registers (SFRs)**.

Table 3 - 8 Extended Special Function Register (2nd SFR) List (1/5)

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
F0030H	Pull-up resistor option register 0	PU0	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
F003DH	Pull-up resistor option register 13	PU13	R/W	√	√	—	00H
F0043H	Port input mode register 3	PIM3	R/W	√	√	—	00H
F0045H	Port input mode register 5	PIM5	R/W	√	√	—	00H
F0053H	Port output mode register 3	POM3	R/W	√	√	—	00H
F0055H	Port output mode register 5	POM5	R/W	√	√	—	00H
F0060H	Port mode control register 0	PMC0	R/W	√	√	—	FFH
F0061H	Port mode control register 1	PMC1	R/W	√	√	—	FFH
F0062H	Port mode control register 2	PMC2	R/W	√	√	—	FFH
F0063H	Port mode control register 3	PMC3	R/W	√	√	—	FFH
F0070H	Noise filter enable register 0	NFEN0	R/W	√	√	—	00H
F0071H	Noise filter enable register 1	NFEN1	R/W	√	√	—	00H
F0073H	Input switch control register	ISC	R/W	√	√	—	00H
F0074H	Timer input select register 0	TIS0	R/W	—	√	—	00H
F0077H	Peripheral I/O redirection register 0	PIOR0	R/W	—	√	—	00H
F0078H	Invalid memory access detection control register	IAWCTL	R/W	—	√	—	00H
F007AH	Frequency measurement clock select register	FMCKS	R/W	√	√	—	00H
F007BH	Port mode select register	PMS	R/W	√	√	—	00H
F0090H	Data flash control register	DFLCTL	R/W	√	√	—	00H
F00A0H	High-speed on-chip oscillator trimming register	HIOTRM	R/W	—	√	—	Undefined Note 1
F00A8H	High-speed on-chip oscillator frequency select register	HOCODIV	R/W	—	√	—	Undefined Note 2
F00AAH	Flash operating mode select register	FLMODE	R/W	√	√	—	Undefined Note 3
F00ABH	Flash operating mode protect register	FLMWRP	R/W	√	√	—	00H
F00F0H	Peripheral enable register 0	PER0	R/W	√	√	—	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 FRQSEL2 to FRQSEL0 of the option byte (000C2H).
Note 3. The reset value of the FLMODE register is determined by the setting of the option byte.

Table 3 - 9 Extended Special Function Register (2nd SFR) List (2/5)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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
F00F5H	RAM parity error control register	RPECTL		R/W	√	√	—	00H
F00F8H	Regulator mode control register	PMMC		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
F00FCH	Peripheral enable register 2	PER2		R/W	√	√	—	00H
F00FDH	Peripheral reset control register 2	PRR2		R/W	√	√	—	00H
F00FEH	BCD correction result register	BCDADJ		R	—	√	—	Undefined
F0100H	Serial status register 00	SSR00L	SSR00	R	—	√	√	0000H
F0101H		—			—	—		
F0102H	Serial status register 01	SSR01L	SSR01	R	—	√	√	0000H
F0103H		—			—	—		
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		—			—	—		
F0110H	Serial mode register 00	SMR00		R/W	—	—	√	0020H
F0111H								
F0112H	Serial mode register 01	SMR01		R/W	—	—	√	0020H
F0113H								
F0118H	Serial communication operation setting register 00	SCR00		R/W	—	—	√	0087H
F0119H								
F011AH	Serial communication operation setting register 01	SCR01		R/W	—	—	√	0087H
F011BH								
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	—	—	√	0303H
F0129H								

Table 3 - 10 Extended Special Function Register (2nd SFR) List (3/5)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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		—			—	—		
F0180H	Timer counter register 00	TCR00		R	—	—	√	FFFFH
F0181H								
F0182H	Timer counter register 01	TCR01		R	—	—	√	FFFFH
F0183H								
F0184H	Timer counter register 02	TCR02		R	—	—	√	FFFFH
F0185H								
F0186H	Timer counter register 03	TCR03		R	—	—	√	FFFFH
F0187H								
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								
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		—			—	—		
F01B0H	Timer channel enable status register 0	TE0L	TE0	R	√	√	√	0000H
F01B1H		—			—	—		
F01B2H	Timer channel start register 0	TS0L	TS0	R/W	√	√	√	0000H
F01B3H		—			—	—		
F01B4H	Timer channel stop register 0	TT0L	TT0	R/W	√	√	√	0000H
F01B5H		—			—	—		
F01B6H	Timer clock select register 0	TPS0		R/W	—	—	√	0000H
F01B7H								
F01B8H	Timer output register 0	TO0L	TO0	R/W	—	√	√	0000H
F01B9H		—			—	—		
F01BAH	Timer output enable register 0	TOE0L	TOE0	R/W	√	√	√	0000H
F01BBH		—			—	—		

Table 3 - 11 Extended Special Function Register (2nd SFR) List (4/5)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F01BCH	Timer output level register 0	TOL0L	TOL0	R/W	—	√	√	0000H
F01BDH		—			—	—		
F01BEH	Timer output mode register 0	TOM0L	TOM0	R/W	—	√	√	0000H
F01BFH		—			—	—		
F0240H	Event output destination select register 00	ELSELR00		R/W	—	√	—	00H
F0241H	Event output destination select register 01	ELSELR01		R/W	—	√	—	00H
F0242H	Event output destination select register 02	ELSELR02		R/W	—	√	—	00H
F0243H	Event output destination select register 03	ELSELR03		R/W	—	√	—	00H
F0244H	Event output destination select register 04	ELSELR04		R/W	—	√	—	00H
F0245H	Event output destination select register 05	ELSELR05		R/W	—	√	—	00H
F0246H	Event output destination select register 06	ELSELR06		R/W	—	√	—	00H
F0247H	Event output destination select register 07	ELSELR07		R/W	—	√	—	00H
F0248H	Event output destination select register 08	ELSELR08		R/W	—	√	—	00H
F0249H	Event output destination select register 09	ELSELR09		R/W	—	√	—	00H
F024AH	Event output destination select register 10	ELSELR10		R/W	—	√	—	00H
F024BH	Event output destination select register 11	ELSELR11		R/W	—	√	—	00H
F024CH	Event output destination select register 12	ELSELR12		R/W	—	√	—	00H
F024DH	Event output destination select register 13	ELSELR13		R/W	—	√	—	00H
F024EH	Event output destination select register 14	ELSELR14		R/W	—	√	—	00H
F024FH	Event output destination select register 15	ELSELR15		R/W	—	√	—	00H
F0250H	Event output destination select register 16	ELSELR16		R/W	—	√	—	00H
F0251H	Event output destination select register 17	ELSELR17		R/W	—	√	—	00H
F0252H	Event output destination select register 18	ELSELR18		R/W	—	√	—	00H
F0253H	Event output destination select register 19	ELSELR19		R/W	—	√	—	00H
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
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
F0310H	Watch error correction register	SUBCUD		R/W	—	—	√	0020H Note
F0312H	Frequency measurement count register L	FMCRL		R	—	—	√	0000H
F0314H	Frequency measurement count register H	FMCRH		R	—	—	√	0000H
F0316H	Frequency measurement control register	FMCTL		R/W	√	√	—	00H
F0340H	Comparator mode setting register	COMPMDR		R/W	√	√	—	00H

Note Initialized only after a reset by a power-on reset.

Table 3 - 12 Extended Special Function Register (2nd SFR) List (5/5)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0341H	Comparator filter control register	COMPFIR		R/W	√	√	—	00H
F0342H	Comparator output control register	COMPOCR		R/W	√	√	—	00H
F0348H	Operational amplifier mode control register	AMPMC		R/W	√	√	—	00H
F0349H	Operational amplifier trigger mode control register	AMPTRM		R/W	√	√	—	00H
F034AH	Operational amplifier ELC trigger select register	AMPTRS		R/W	√	√	—	00H
F034BH	Operational amplifier control register	AMPC		R/W	√	√	—	00H
F034CH	Operational amplifier monitor register	AMPMON		R	√	√	—	00H
F0350H	8-bit interval timer compare register 00	TRTCMP00	TRTCMP0	R/W	—	√	√	FFH
F0351H	8-bit interval timer compare register 01	TRTCMP01		R/W	—	√		FFH
F0352H	8-bit interval timer control register 0	TRTCR0		R/W	√	√	—	00H
F0353H	8-bit interval timer division register 0	TRTMD0		R/W	—	√	—	00H
F0358H	8-bit interval timer compare register 10	TRTCMP10	TRTCMP1	R/W	—	√	√	FFH
F0359H	8-bit interval timer compare register 11	TRTCMP11		R/W	—	√		FFH
F035AH	8-bit interval timer control register 1	TRTCR1		R/W	√	√	—	00H
F035BH	8-bit interval timer division register 1	TRTMD1		R/W	—	√	—	00H
F0500H	8-bit interval timer counter register 00	TRT00	TRT0	R	—	√	√	00H
F0501H	8-bit interval timer counter register 01	TRT01		R	—	√		00H
F0508H	8-bit interval timer counter register 10	TRT10	TRT1	R	—	√	√	00H
F0509H	8-bit interval timer counter register 11	TRT11		R	—	√		00H
F0511H	DOC control register	DOCR		R/W	√	√	—	00H
F0512H	DOC data input register	DODIR		R/W	—	—	√	0000H
F0514H	DOC data setting register	DODSR		R/W	—	—	√	0000H

Remark For SFRs in the SFR area, see **Tables 3 - 5** to **3 - 7 Special Function Register (SFR) List**.

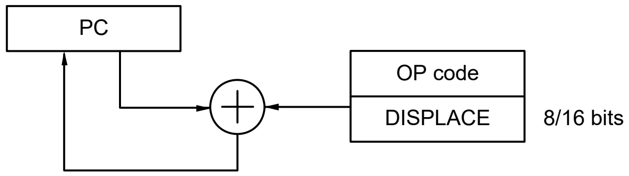
3.3 Instruction Address Addressing

3.3.1 Relative addressing

[Function]

Relative addressing stores in the program counter (PC) the result of adding a displacement value included in the instruction word (signed complement data: -128 to +127 or -32768 to +32767) to the program counter (PC)'s value (the start address of the next instruction), and specifies the program address to be used as the branch destination. Relative addressing is applied only to branch instructions.

Figure 3 - 12 Outline of Relative Addressing



3.3.2 Immediate addressing

[Function]

Immediate addressing stores immediate data of the instruction word in the program counter, and specifies the program address to be used as the branch destination.

For immediate addressing, CALL !!addr20 or BR !!addr20 is used to specify 20-bit addresses and CALL !addr16 or BR !addr16 is used to specify 16-bit addresses. 0000 is set to the higher 4 bits when specifying 16-bit addresses.

Figure 3 - 13 Example of CALL !!addr20/BR !!addr20

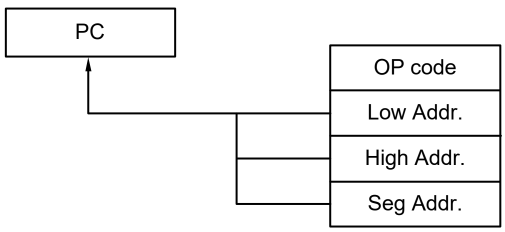
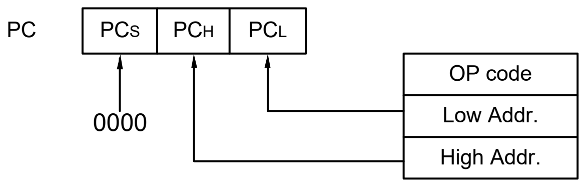


Figure 3 - 14 Example of CALL !addr16/BR !addr16



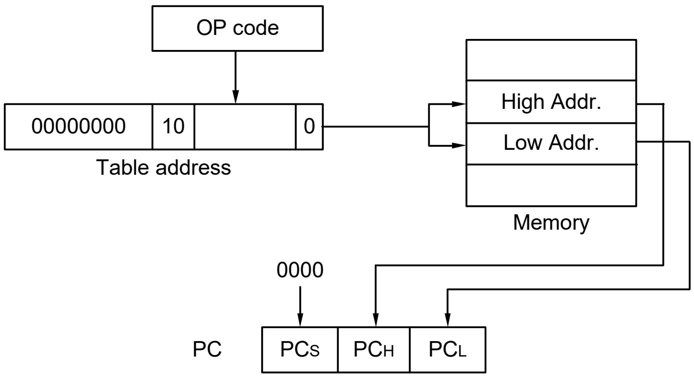
3.3.3 Table indirect addressing

[Function]

Table indirect addressing specifies a table address in the CALLT table area (0080H to 00BFH) with the 5-bit immediate data in the instruction word, stores the contents at that table address and the next address in the program counter (PC) as 16-bit data, and specifies the program address. Table indirect addressing is applied only for CALLT instructions.

In the RL78 microcontrollers, branching is enabled only to the 64 KB space from 00000H to 0FFFFH.

Figure 3 - 15 Outline of Table Indirect Addressing

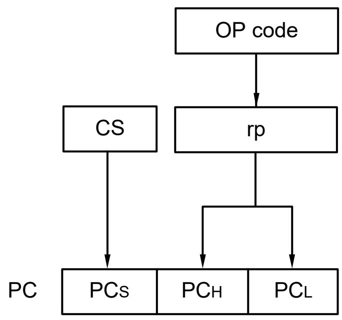


3.3.4 Register indirect addressing

[Function]

Register direct addressing stores in the program counter (PC) the contents of a general-purpose register pair (AX/BC/DE/HL) and CS register of the current register bank specified with the instruction word as 20-bit data, and specifies the program address. Register indirect addressing can be applied only to the CALL AX, BC, DE, HL, and BR AX instructions.

Figure 3 - 16 Outline of Register Indirect Addressing



3.4 Addressing for Processing Data Addresses

3.4.1 Implied addressing

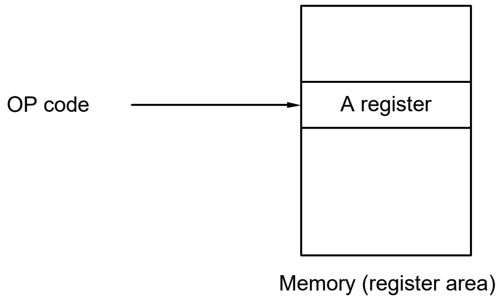
[Function]

Instructions for accessing registers (such as accumulators) that have special functions are directly specified with the instruction word, without using any register specification field in the instruction word.

[Operand format]

Implied addressing can be applied only to MULU X.

Figure 3 - 17 Outline of Implied Addressing



3.4.2 Register addressing

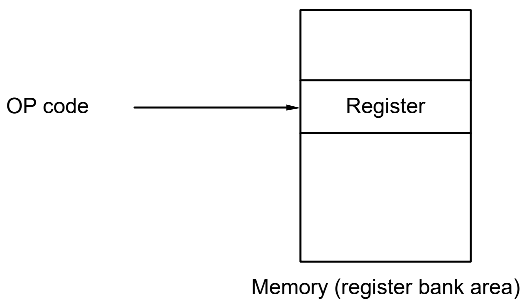
[Function]

Register addressing accesses a general-purpose register as an operand. The instruction word of 3-bit long is used to select an 8-bit register and the instruction word of 2-bit long is used to select a 16-bit register.

[Operand format]

Identifier	Description
r	X, A, C, B, E, D, L, H
rp	AX, BC, DE, HL

Figure 3 - 18 Outline of Register Addressing



3.4.3 Direct addressing

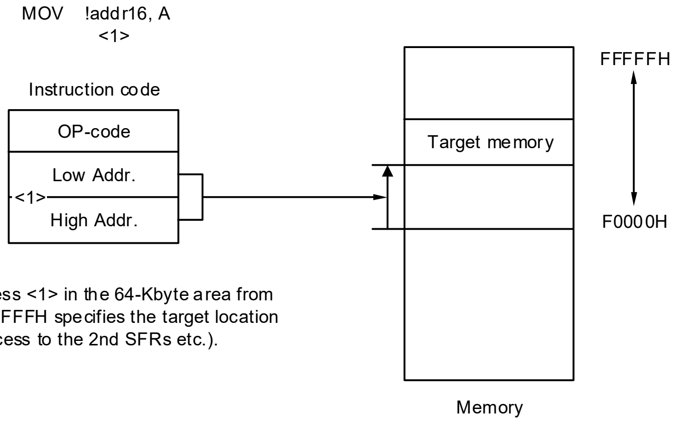
[Function]

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

[Operand format]

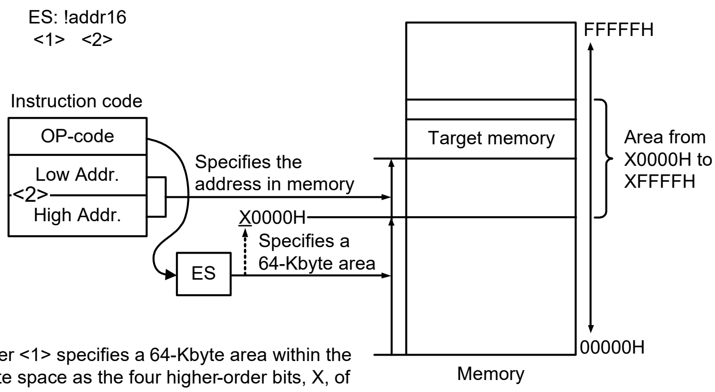
Identifier	Description
!addr16	Label or 16-bit immediate data (only the space from F0000H to FFFFFH is specifiable)
ES:!addr16	Label or 16-bit immediate data (higher 4-bit addresses are specified by the ES register)

Figure 3 - 19 Example of !addr16



• A 16-bit address <1> in the 64-Kbyte area from F0000H to FFFFFH specifies the target location (for use in access to the 2nd SFRs etc.).

Figure 3 - 20 Example of ES:!addr16



• The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.

• A 16-bit address <2> in the area from X0000H to XFFFFH and the ES register <1> specify the target location; this is used for access to fixed data other than that in mirrored areas.

3.4.4 Short direct addressing

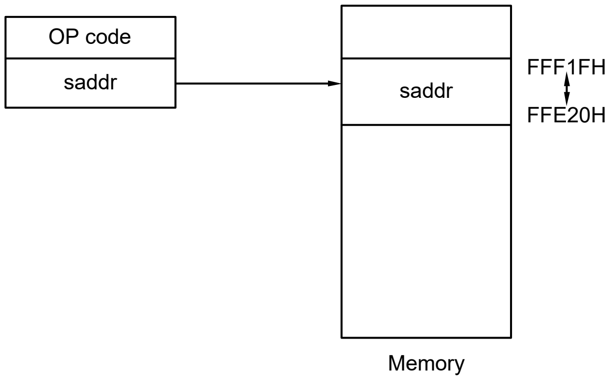
[Function]

Short direct addressing directly specifies the target addresses using 8-bit data in the instruction word. This type of addressing is applied only to the space from FFE20H to FFF1FH.

[Operand format]

Identifier	Description
SADDR	Label, FFE20H to FFF1FH immediate data, or 0FE20H to 0FF1FH immediate data (only the space from FFE20H to FFF1FH is specifiable)
SADDRP	Label, FFE20H to FFF1FH immediate data, or 0FE20H to 0FF1FH immediate data (even address only) (only the space from FFE20H to FFF1FH is specifiable)

Figure 3 - 21 Outline of Short Direct Addressing



Remark SADDR and SADDRP are used to describe the values of addresses FE20H to FF1FH with 16-bit immediate data (higher 4 bits of actual address are omitted), and the values of addresses FFE20H to FFF1FH with 20-bit immediate data.
Regardless of whether SADDR or SADDRP is used, addresses within the space from FFE20H to FFF1FH are specified for the memory.

3.4.5 SFR addressing

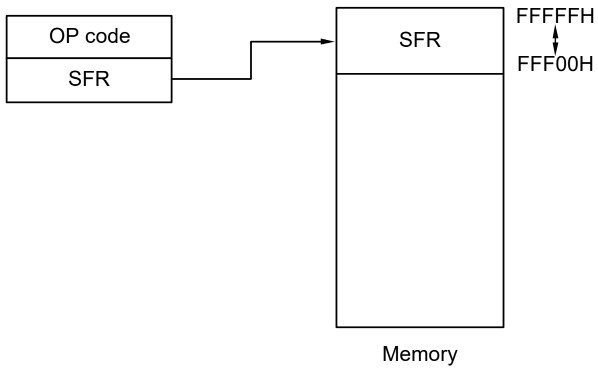
[Function]

SFR addressing directly specifies the target SFR addresses using 8-bit data in the instruction word. This type of addressing is applied only to the space from FFF00H to FFFFFH.

[Operand format]

Identifier	Description
SFR	SFR name
SFRP	16-bit-manipulatable SFR name (even address)

Figure 3 - 22 Outline of SFR Addressing



3.4.6 Register indirect addressing

[Function]

Register indirect addressing directly specifies the target addresses using the contents of the register pair specified with the instruction word as an operand address.

[Operand format]

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

Figure 3 - 23 Example of [DE], [HL]

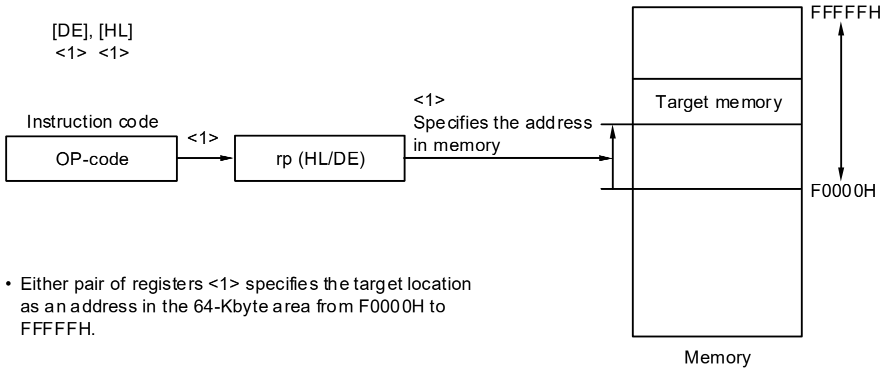
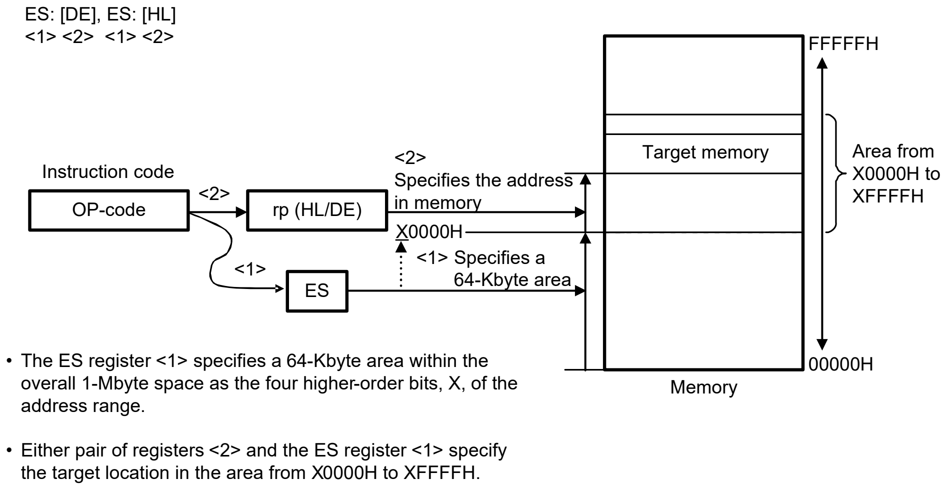


Figure 3 - 24 Example of ES:[DE], ES:[HL]



3.4.7 Based addressing

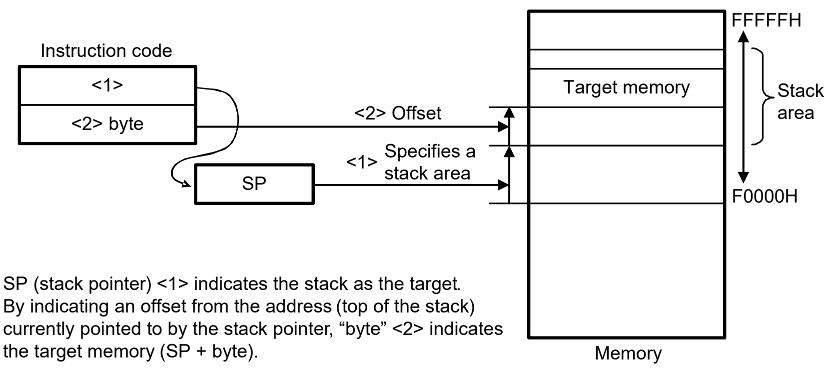
[Function]

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]

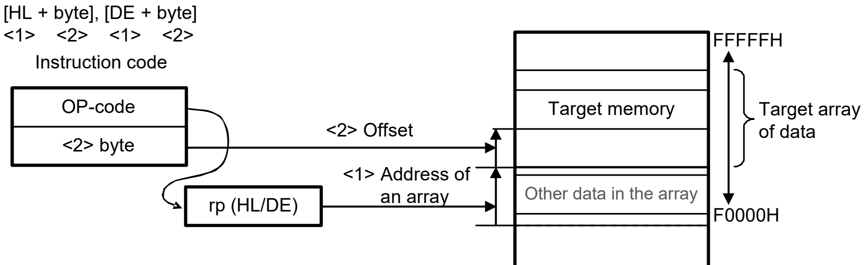
Identifier	Description
—	[HL + byte], [DE + byte], [SP + byte] (only the space from F0000H to FFFFFH is specifiable)
—	word[B], word[C] (only the space from F0000H to FFFFFH is specifiable)
—	word[BC] (only the space from F0000H to FFFFFH is specifiable)
—	ES:[HL + byte], ES:[DE + byte] (higher 4-bit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



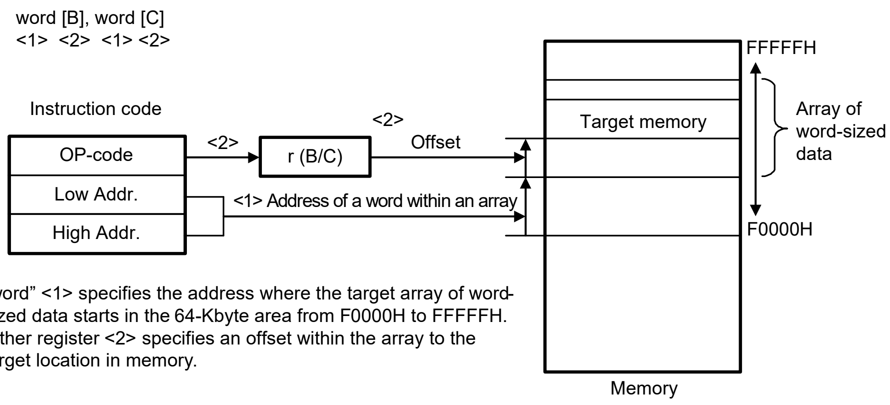
• SP (stack pointer) <1> indicates the stack as the target.
• By indicating an offset from the address (top of the stack) currently pointed to by the stack pointer, “byte” <2> indicates the target memory (SP + byte).

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



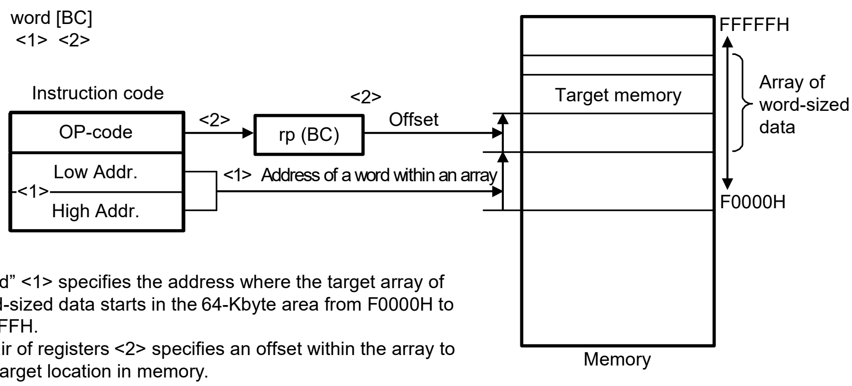
- Either pair of registers <1> specifies the address where the target array of data starts in the 64-Kbyte area from F0000H to FFFFFH.
- “byte” <2> specifies an offset within the array to the target location in memory.

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



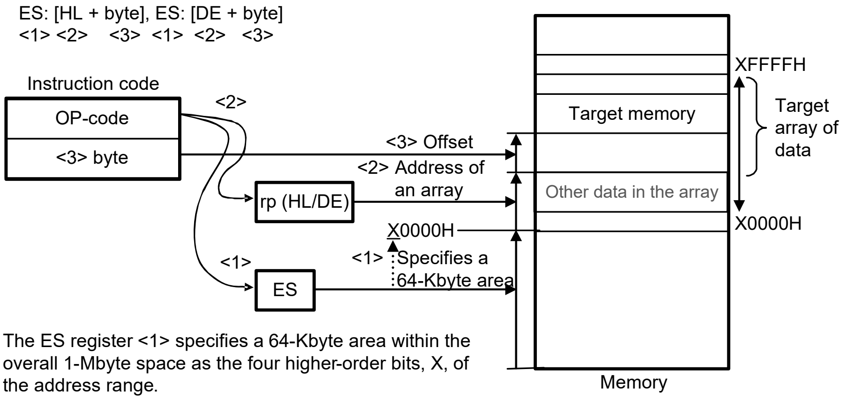
- “word” <1> specifies the address where the target array of word-sized data starts in the 64-Kbyte area from F0000H to FFFFFH.
- Either register <2> specifies an offset within the array to the target location in memory.

Figure 3 - 28 Example of word [BC]



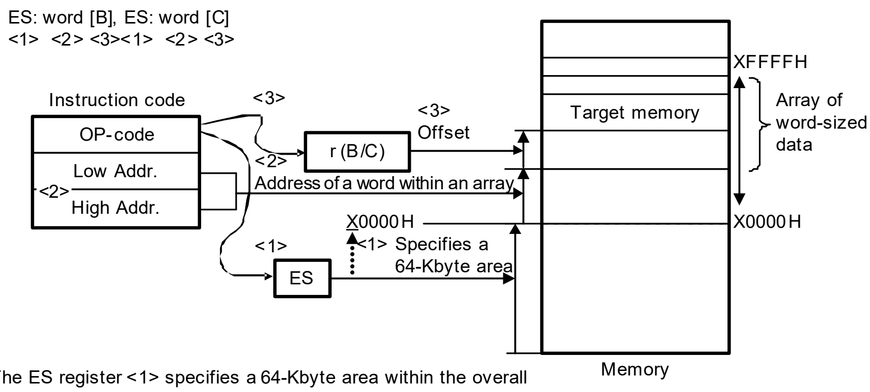
- “word” <1> specifies the address where the target array of word-sized data starts in the 64-Kbyte area from F0000H to FFFFFH.
- A pair of registers <2> specifies an offset within the array to the target location in memory.

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



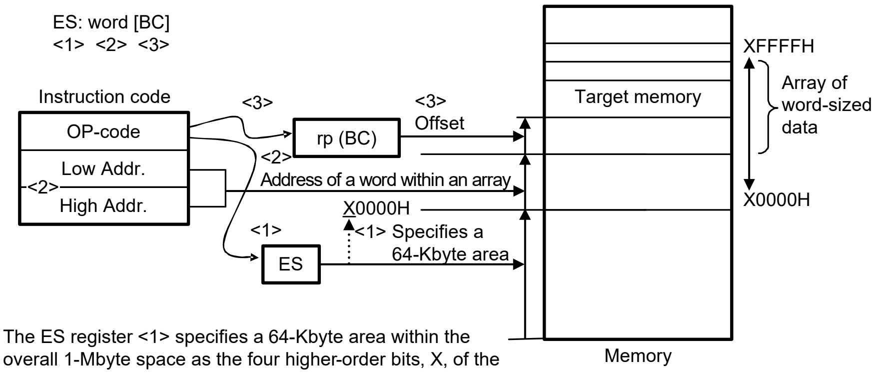
- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- Either pair of registers <2> specifies the address where the target array of data starts in the 64-Kbyte area specified in the ES register <1>.
- “byte” <3> specifies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

Figure 3 - 30 Example of word [B], 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 the64-Kbyte area specified in the ES register <1>.
- Either register <3> specifies an offset within the array to the target location in memory.

Figure 3 - 31 Example of word [BC]



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

3.4.8 Based indexed addressing

[Function]

Based indexed addressing uses the contents of a register pair specified with the instruction word as the base address, and the content of the B register or C register similarly specified with the instruction word as offset address. The sum of these values is used to specify the target address.

[Operand format]

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

Figure 3 - 32 Example of [HL + B], [HL + C]

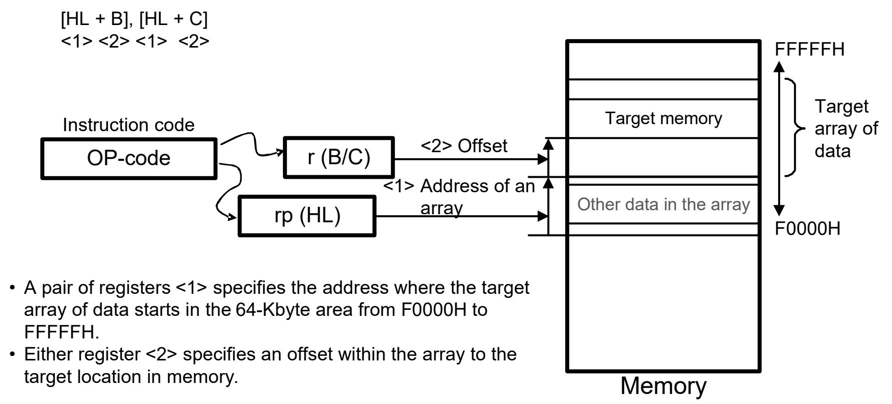
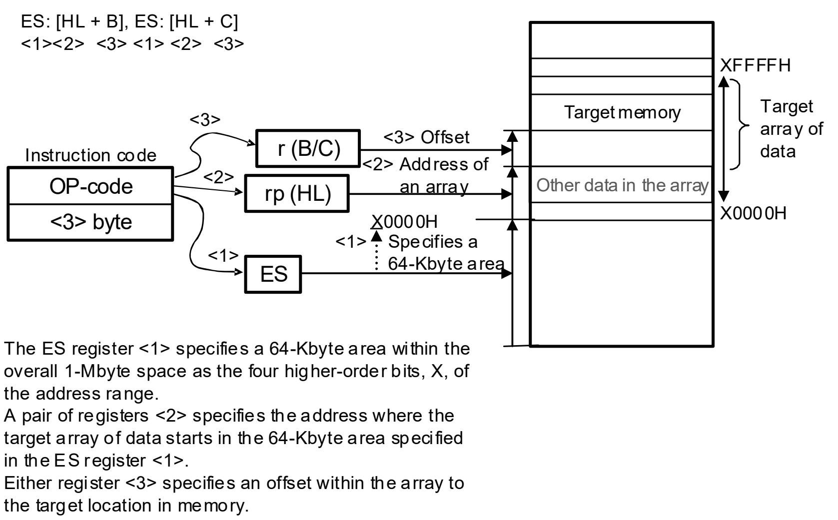


Figure 3 - 33 Example of ES:[HL + B], ES:[HL + C]



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.

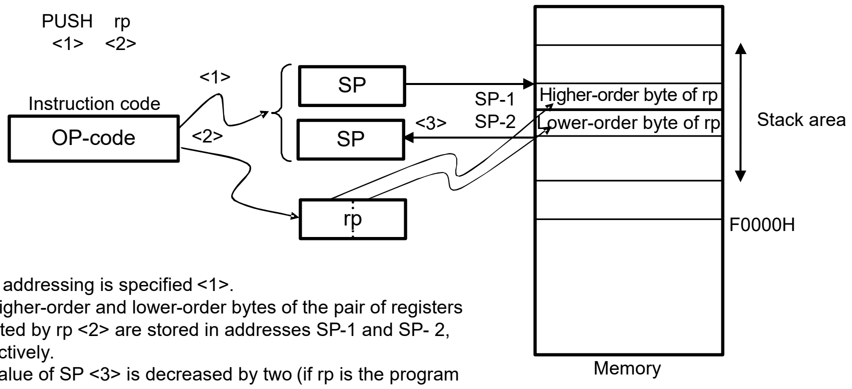
Only the internal RAM area can be set as the stack area.

[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

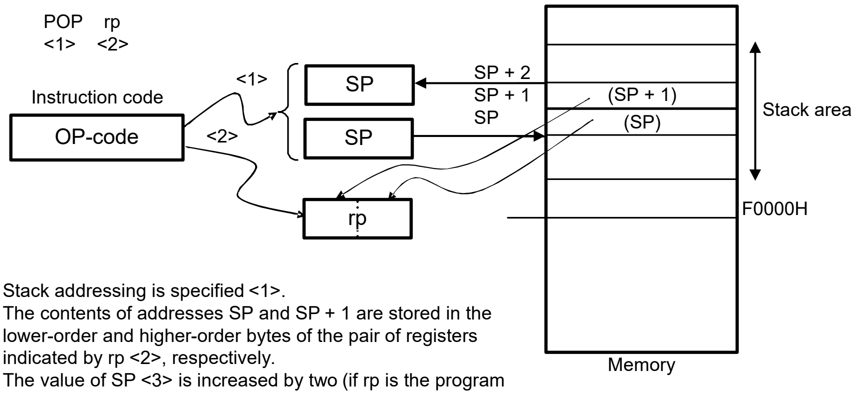
The data to be saved/restored by each stack operation is shown in Figures 3 - 34 to 3 - 39.

Figure 3 - 34 Example of PUSH rp



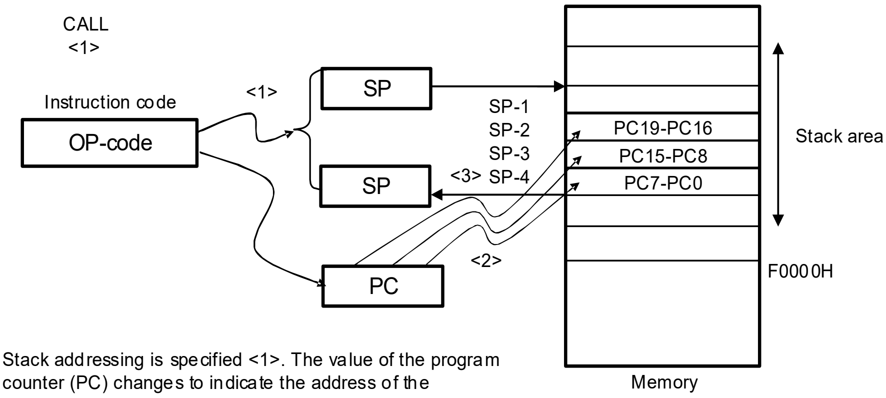
- Stack addressing is specified <1>.
- The higher-order and lower-order bytes of the pair of registers indicated by rp <2> are stored in addresses SP-1 and SP- 2, respectively.
- The value of SP <3> is decreased by two (if rp is the program status word (PSW), the value of the PSW is stored in SP-1 and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

Figure 3 - 35 Example of POP



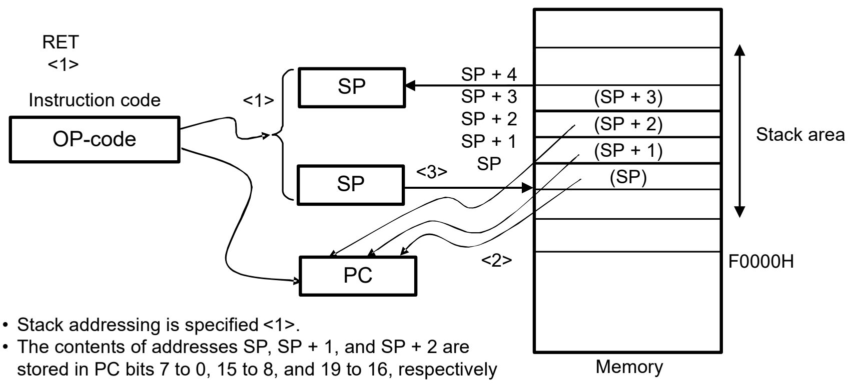
- Stack addressing is specified <1>.
- The contents of addresses SP and SP + 1 are stored in the lower-order and higher-order bytes of the pair of registers indicated by rp <2>, respectively.
- The value of SP <3> is increased by two (if rp is the program status word (PSW), the content of address SP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

Figure 3 - 36 Example of CALL, CALLT



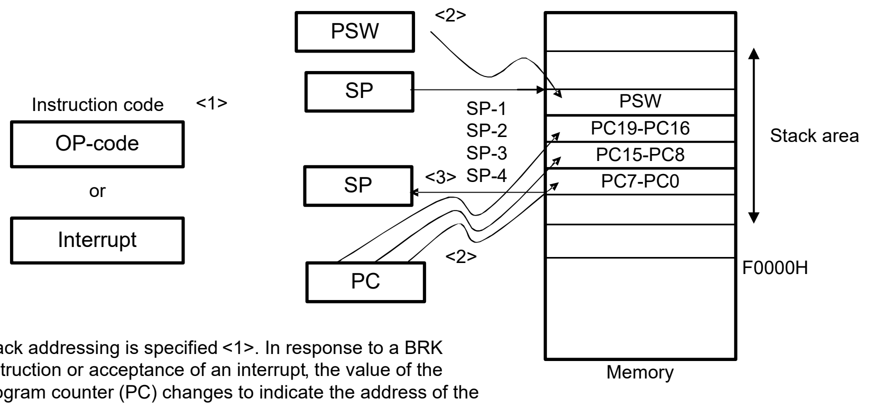
- Stack addressing is specified <1>. The value of the program counter (PC) changes to indicate the address of the instruction following the CALL instruction.
- The values of PC bits 19 to 16, 15 to 8, and 7 to 0 are stored in addresses SP-2, SP-3, and SP-4, respectively <2>.
- The value of the SP <3>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

Figure 3 - 37 Example of RET



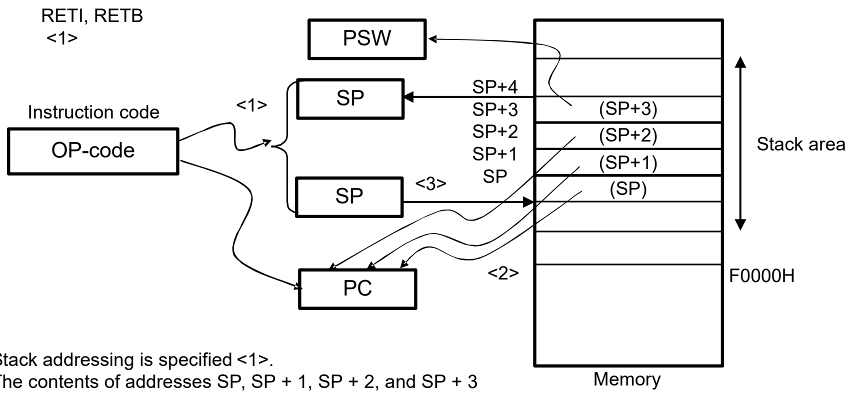
- Stack addressing is specified <1>.
- The contents of addresses SP, SP + 1, and SP + 2 are stored in PC bits 7 to 0, 15 to 8, and 19 to 16, respectively <2>.
- The value of SP <3> is increased by four.

Figure 3 - 38 Example of Interrupt, BRK



- Stack addressing is specified <1>. In response to a BRK instruction or acceptance of an interrupt, the value of the program counter (PC) changes to indicate the address of the next instruction.
- The values of the PSW, PC bits 19 to 16, 15 to 8, and 7 to 0 are stored in addresses SP-1, SP-2, SP-3, and SP-4, respectively <2>.
- The value of the SP <3> is decreased by 4.

Figure 3 - 39 Example of RETI, RETB



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

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The RL78/I1D microcontrollers are provided with digital I/O ports, which enable variety of control operations.

In addition to the function as digital I/O ports, these ports have several alternate functions. For details of the alternate functions, see **CHAPTER 2 PIN FUNCTIONS**.

4.2 Port Configuration

Ports include the following hardware.

Table 4 - 1 Port Configuration

Item	Configuration
Control registers	Port mode registers (PM0 to PM6, PM13) Port registers (P0 to P6, P12, P13) Pull-up resistor option registers (PU0, PU3 to PU5, PU13) Port input mode registers (PIM3, PIM5) Port output mode registers (POM3, POM5) Port mode control registers (PMC0 to PMC3) Peripheral I/O redirection registers (PIOR0)
Port	• 20-pin products Total: 14 (CMOS I/O: 11, CMOS input: 3) • 24-pin products Total: 18 (CMOS I/O: 15, CMOS input: 3) • 30-pin products Total: 24 (CMOS I/O: 19, CMOS input: 5) • 32-pin products Total: 26 (CMOS I/O: 21, CMOS input: 5) • 48-pin products Total: 42 (CMOS I/O: 33, CMOS input: 5, N-ch open drain I/O: 4)
Pull-up resistor	• 20-pin products Total: 5 • 24-pin products Total: 9 • 30-pin products Total: 7 • 32-pin products Total: 9 • 48-pin products Total: 19

4.2.1 Port 0

Port 0 is an I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00 to P04 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 0 (PU0).
To use P02 to P04 as digital input/output pins, set them in the digital I/O mode by using port mode control register 0 (PMC0) (can be specified in 1-bit units).
This port can also be used for A/D converter analog input and buzzer output.
Reset signal generation sets port 0 to analog input port.

4.2.2 Port 1

Port 1 is an I/O port with an output latch. Port 1 can be set to the input mode or output mode in 1-bit units using port mode register 1 (PM1).
This port can also be used for A/D converter analog input, operational amplifier I/O, and comparator input.
To use the P10 to P17 pins as analog input pins, set them to analog input using port mode control register 1 (PMC1) (can be specified in 1-bit units).
Reset signal generation sets port 1 to analog input port.

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 using port mode register 2 (PM2).
This port can also be used for A/D converter analog input, operational amplifier I/O, and comparator input.
To use P20 to P25 as analog input pins, set them to analog input using port mode control register 2 (PMC2) (can be specified in 1-bit units).
Reset signal generation sets port 2 to analog input port.

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 using port mode register 3 (PM3). When the P30 to P33 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 3 (PU3).
Input to the P30, P32, and P33 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 3 (PIM3).
Output from the P30 pin can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 3 (POM3).
This port can also be used for external interrupt request input, real-time clock 2 correction clock output, serial interface clock I/O, timer I/O, and comparator input.
Reset signal generation sets port 3 to input port.

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 using port mode register 4 (PM4). When the P40 pin is used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 4 (PU4).
This port can also be used for data I/O for a flash memory programmer/debugger.
Reset signal generation sets port 4 to input port.

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 using port mode register 5 (PM5). When the P50 to P57 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 5 (PU5).
Input to the P51, P52, and P54 to P57 pin can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 5 (PIM5).
Output from the P51 to P56 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 5 (POM5).
This port can also be used for external interrupt request input, serial interface data I/O, clock I/O, programming UART transmission/reception, timer I/O, and comparator output.
Reset signal generation sets port 5 to input port.

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 using port mode register 6 (PM6).
The output of the P60 to P63 pins is N-ch open-drain output (6 V tolerance).
This port can also be used for serial interface, chip select input, and timer I/O.
Reset signal generation sets port 6 to input port.

4.2.8 Port 12

P121 to P124 are 4-bit input ports.
This port can also be used for connecting a resonator for the main system clock, connecting a resonator for the subsystem clock, external clock input for the main system clock, and external clock input for the subsystem clock.
Reset signal generation sets P121 to P124 to input port.

4.2.9 Port 13

P130 is a 1-bit I/O port with an output latch. Port 13 can be set to input mode or output mode in 1-bit units using port mode register 13 (PM13). P137 is a 1-bit input-only port.
P137 is fixed an input ports.
This port can also be used for external interrupt request input.

4.3 Registers Controlling Port Function

Port functions are controlled by the following registers.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Pull-up resistor option registers (PUxx)
- Port input mode registers (PIMxx)
- Port output mode registers (POMxx)
- Port mode control registers (PMCxx)
- Peripheral I/O redirection register 0 (PIOR0)

Caution Which registers and bits are included depends on the product. For registers and bits mounted on each product, see Tables 4 - 2 to 4 - 4. Be sure to set bits that are not mounted to their initial values.

Table 4 - 2 PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx registers and the bits mounted on each product (1/3)

Port		Bit Name						48-pin	32-pin	30-pin	24-pin	20-pin
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register					
Port 0	0	PM00	P00	PU00	—	—	—	√	—	—	—	—
	1	PM01	P01	PU01	—	—	—	√	—	—	—	—
	2	PM02	P02	PU02	—	—	PMC02	√	—	—	—	—
	3	PM03	P03	PU03	—	—	PMC03	√	—	—	—	—
	4	PM04	P04	PU04	—	—	PMC04	√	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—
Port 1	0	PM10	P10	—	—	—	PMC10	√	—	—	—	—
	1	PM11	P11	—	—	—	PMC11	√	—	—	—	—
	2	PM12	P12	—	—	—	PMC12	√	√	√	√	√
	3	PM13	P13	—	—	—	PMC13	√	√	√	√	√
	4	PM14	P14	—	—	—	PMC14	√	√	√	√	√
	5	PM15	P15	—	—	—	PMC15	√	√	√	—	—
	6	PM16	P16	—	—	—	PMC16	√	√	√	—	—
	7	PM17	P17	—	—	—	PMC17	√	√	√	—	—
Port 2	0	PM20	P20	—	—	—	PMC20	√	√	√	√	√
	1	PM21	P21	—	—	—	PMC21	√	√	√	√	√
	2	PM22	P22	—	—	—	PMC22	√	√	√	√	√
	3	PM23	P23	—	—	—	PMC23	√	√	√	—	—
	4	PM24	P24	—	—	—	PMC24	√	√	√	—	—
	5	PM25	P25	—	—	—	PMC25	√	√	√	—	—
	6	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—

Table 4 - 3 PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx registers and the bits mounted on each product (2/3)

Port		Bit Name						48-pin	32-pin	30-pin	24-pin	20-pin
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register					
Port 3	0	PM30	P30	PU30	PIM30	POM30	PMC30	√	√	√	√	√
	1	PM31	P31	PU31	—	—	PMC31	√	√	√	√	√
	2	PM32	P32	PU32	PIM32	—	—	√	—	—	—	—
	3	PM33	P33	PU33	PIM33	—	—	√	—	√	—	—
	4	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—
Port 4	0	PM40	P40	PU40	—	—	—	√	√	√	√	√
	1	—	—	—	—	—	—	—	—	—	—	—
	2	—	—	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—
Port 5	0	PM50	P50	PU50	—	—	—	√	—	—	—	—
	1	PM51	P51	PU51	PIM51	POM51	—	√	√	—	√	—
	2	PM52	P52	PU52	PIM52	POM52	—	√	√	—	√	—
	3	PM53	P53	PU53	—	POM53	—	√	√	—	√	—
	4	PM54	P54	PU54	PIM54	POM54	—	√	√	√	√	√
	5	PM55	P55	PU55	PIM55	POM55	—	√	√	√	√	√
	6	PM56	P56	PU56	PIM56	POM56	—	√	√	√	√	—
	7	PM57	P57	PU57	PIM57	—	—	√	—	—	—	—
Port 6	0	PM60	P60	—	—	—	—	√	—	—	—	—
	1	PM61	P61	—	—	—	—	√	—	—	—	—
	2	PM62	P62	—	—	—	—	√	—	—	—	—
	3	PM63	P63	—	—	—	—	√	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—

Table 4 - 4 PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx registers and the bits mounted on each product (3/3)

Port		Bit Name						48-pin	32-pin	30-pin	24-pin	20-pin
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register					
Port 12	0	—	—	—	—	—	—	—	—	—	—	—
	1	—	P121	—	—	—	—	√	√	√	√	√
	2	—	P122	—	—	—	—	√	√	√	√	√
	3	—	P123	—	—	—	—	√	√	√	—	—
	4	—	P124	—	—	—	—	√	√	√	—	—
	5	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—
Port 13	0	PM130	P130	PU130	—	—	—	√	—	—	—	—
	1	—	—	—	—	—	—	—	—	—	—	—
	2	—	—	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—
	7	—	P137	—	—	—	—	√	√	√	√	√

4.3.1 Port mode registers (PMxx)

These registers specify input or output mode for the port in 1-bit units.
These registers can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets these registers to FFH.
When port pins are used as alternate-function pins, set the port mode register by referencing **4.5 Register Settings When Using Alternate Function**.

Figure 4 - 1 Format of Port mode register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM2	1	1	PM25	PM24	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W
PM3	1	1	1	1	PM33	PM32	PM31	PM30	FFF23H	FFH	R/W
PM4	1	1	1	1	1	1	1	PM40	FFF24H	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FFF25H	FFH	R/W
PM6	1	1	1	1	PM63	PM62	PM61	PM60	FFF26H	FFH	R/W
PM13	1	1	1	1	1	1	1	PM130	FFF2DH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 6, 12, 13; n = 0 to 7)
0	Output mode (the pin functions as an output port (output buffer on))
1	Input mode (the pin functions as an input port (output buffer off))

Caution Be sure to set bits that are not mounted to their initial values.

4.3.2 Port registers (Pxx)

These registers set the output latch value of a port.
If the data is read in the input mode, the pin level is read. If it is read in the output mode, the output latch value is read [Note].
These registers can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears these registers to 00H.

Note When P02, P03, P04, P10 to P17, and P20 to P25 are set to the analog function, if a port is read in input mode, the read value is always 0, not the pin level.

Figure 4 - 2 Format of Port register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	0	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	0	0	P25	P24	P23	P22	P21	P20	FFF02H	00H (output latch)	R/W
P3	0	0	0	0	P33	P32	P31	P30	FFF03H	00H (output latch)	R/W
P4	0	0	0	0	0	0	0	P40	FFF04H	00H (output latch)	R/W
P5	P57	P56	P55	P54	P53	P52	P51	P50	FFF05H	00H (output latch)	R/W
P6	0	0	0	0	P63	P62	P61	P60	FFF06H	00H (output latch)	R/W
P12	0	0	0	P124	P123	P122	P121	0	FFF0CH	Undefined	R/W [Note 1]
P13	P137	0	0	0	0	0	0	P130	FFF0DH	Note 2	R/W [Note 1]

Pmn	m = 0 to 6, 12, 13; n = 0 to 7	
	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

Note 1. P121 to P124, and P137 are read-only.
Note 2. P137: Undefined
P130: 0 (output latch)

Caution Be sure to set bits that are not mounted to their initial values.

4.3.3 Pull-up resistor option registers (PUxx)

These registers specify whether the on-chip pull-up resistors are to be used or not. On-chip pull-up resistors can be used in 1-bit units only for the bits set to input mode (PMmn = 1 and POMmn = 0) for the pins to which the use of an on-chip pull-up resistor has been specified in these registers. On-chip pull-up resistors cannot be connected to bits set to output mode and bits used as alternate-function output pins, regardless of the settings of these registers. Similarly, on-chip pull-up resistors cannot be connected to the pins used as alternate-function output pins and the pins set to the analog function.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H (Only PU4 is set to 01H).

Figure 4 - 3 Format of Pull-up resistor option register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	PU04	PU03	PU02	PU01	PU00	F0030H	00H	R/W
PU3	0	0	0	0	PU33	PU32	PU31	PU30	F0033H	00H	R/W
PU4	0	0	0	0	0	0	0	PU40	F0034H	01H	R/W
PU5	PU57	PU56	PU55	PU54	PU53	PU52	PU51	PU50	F0035H	00H	R/W
PU13	0	0	0	0	0	0	0	PU130	F003DH	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0, 3 to 5, 13; 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 mounted to their initial values.

4.3.4 Port input mode registers (PIMxx)

These registers set the input buffer in 1-bit units.
TTL input buffer can be selected during serial communication with an external device of the different potential.
Port input mode registers can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears these registers to 00H.

Figure 4 - 4 Format of Port input mode register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM3	0	0	0	0	PIM33	PIM32	0	PIM30	F0043H	00H	R/W
PIM5	PIM57	PIM56	PIM55	PIM54	0	PIM52	PIM51	0	F0045H	00H	R/W

PIMmn	Pmn pin input buffer selection (m = 3, 5; n = 0 to 7)
0	Normal input buffer
1	TTL input buffer

Caution Be sure to set bits that are not mounted to their initial values.

4.3.5 Port output mode registers (POMxx)

These registers set the output mode in 1-bit units.
N-ch open-drain output (V_{DD} tolerance) mode can be selected during serial communication with an external device of the different potential, and for the SDA00 and SDA01 pins during simplified I^2C communication with an external device of the same potential.
In addition, POMxx register is set with PUxx register, whether or not to use the on-chip pull-up resistor.
These registers can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears these registers to 00H.

Caution An on-chip pull-up resistor is not connected to a bit for which N-ch open drain output (V_{DD} tolerance) mode is set.

Figure 4 - 5 Format of Port output mode register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM3	0	0	0	0	0	0	0	POM30	F0053H	00H	R/W
POM5	0	POM56	POM55	POM54	POM53	POM52	POM51	0	F0055H	00H	R/W

POMmn	Pmn pin output mode selection (m = 3, 5; n = 0 to 6)
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance) mode

Caution Be sure to set bits that are not mounted to their initial values.

4.3.6 Port mode control registers (PMCxx)

These registers set the P02 to P04, P10 to P17, P20 to P25, P30, and P31 digital I/O/analog input in 1-bit units.

These registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to FFH.

Figure 4 - 6 Format of Port mode control register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMC0	1	1	1	PMC04	PMC03	PMC02	1	1	F0060H	FFH	R/W
PMC1	PMC17	PMC16	PMC15	PMC14	PMC13	PMC12	PMC11	PMC10	F0061H	FFH	R/W
PMC2	1	1	PMC25	PMC24	PMC23	PMC22	PMC21	PMC20	F0062H	FFH	R/W
PMC3	1	1	1	1	1	1	PMC31	PMC30	F0063H	FFH	R/W

PMCmn	Pmn pin digital I/O/analog input selection (m = 0 to 3; n = 0 to 7)
0	Digital I/O (alternate function other than analog input)
1	Analog input

Caution Be sure to set bits that are not mounted to their initial values.

4.3.7 Peripheral I/O redirection register 0 (PIOR0)

This register is used to specify whether to enable or disable the peripheral I/O redirect function.
This function is used to switch ports to which alternate functions are assigned.
In addition, the settings for redirection can be changed only until operation of the function is enabled.
The PIOR0 register can be set by an 8-bit memory manipulation instruction.
Reset signal generation sets this register to 00H.

Figure 4 - 7 Format of Peripheral I/O redirection register 0 (PIOR0)

Address: F0077H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PIOR0	0	0	0	PIOR04	PIOR03	PIOR02	PIOR01	PIOR00

Bit	Function	48-pin		32-pin		30-pin		24-pin		20-pin	
		Setting value		Setting value		Setting value		Setting value		Setting value	
		0	1	0	1	0	1	0	1	0	1
PIOR04	TI03/TO03	P52	P57	P52	Note	Note		P52	Note	Note	
PIOR03	TI02/TO02	P51	P33	P51	Note	P33	Note	P51	Note	Note	
PIOR02	TI01/TO00	P31	P32	P31	Note	P31	Note	P31	Note	P31	Note
PIOR01	TI00/TO01	P30	P50	P30	Note	P30	Note	P30	Note	P30	Note
PIOR00	SCK00	P56	P30	P56	P30	P56	P30	P56	P30	P30	Note
	SCL00	P56	P30	P56	P30	P56	P30	P56	P30	Note	P30

Note This area cannot be used. Be set to 0 (default value).

4.4 Port Function Operations

Port operations differ depending on whether the input or output mode is set, as shown below.

4.4.1 Writing to I/O port

(1) Output mode

A value is written to the output latch by a transfer instruction, and the output latch contents are output from the pin.

Once data is written to the output latch, it is retained until data is written to the output latch again.

The data of the output latch is cleared when a reset signal is generated.

(2) Input mode

A value is written to the output latch by a transfer instruction, but since the output buffer is off, the pin status does not change.

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 status is read by a transfer instruction. The output latch contents do not change.

4.4.3 Operations on I/O port

(1) Output mode

An operation is performed on the output latch contents, and the result is written to the output latch. The output latch contents are output from the pins.

Once data is written to the output latch, it is retained until data is written to the output latch again.

The data of the output latch is cleared when a reset signal is generated.

(2) Input mode

The pin level is read and an operation is performed on its contents. The result of the operation is written to the output latch, but since the output buffer is off, the pin status does not change. 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 Handling different potential (1.8 V, 2.5 V) by using I/O buffers

It is possible to connect an external device operating on a different potential (1.8 V, 2.5 V) by switching I/O buffers with the port input mode register (PIMxx) and port output mode register (POMxx).
When receiving input from an external device with a different potential (1.8 V, 2.5 V), set the port input mode registers 3 and 5 (PIM3 and PIM5) on a bit-by-bit basis to enable normal input (CMOS)/TTL input buffer switching.
When outputting data to an external device with a different potential (1.8 V, 2.5 V), set the port output mode registers 3 and 5 (POM3 and POM5) on a bit-by-bit basis to enable normal output (CMOS)/N-ch open drain (V_{DD} tolerance) switching.

The connection of a serial interface is described in the following.

(1) Setting procedure when using input pins of UART0, CSI00, and CSI01 functions for the TTL input buffer

In case of UART0:	P55
In case of CSI00:	P55, P56 (P30)
In case of CSI01:	P51, P52

Remark Functions in parentheses can be assigned via settings in the peripheral I/O redirection register 0 (PIOR0).

<1> Using an external resistor, pull up externally the pin to be used to the power supply of the target device (on-chip pull-up resistor cannot be used).
<2> Set the corresponding bit of the PIM3 and PIM5 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 or simplified SPI (CSI[Note]) mode.

<R> **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) Setting procedure when using output pins of UART0, CSI00, and CSI01 functions in N-ch open-drain output mode

In case of UART0:	P54
In case of CSI00:	P54, P56 (P30)
In case of CSI01:	P51, P53

Remark Functions in parentheses can be assigned via settings in the peripheral I/O redirection register 0 (PIOR0).

<1> Using an external resistor, pull up externally the pin to be used to the power supply of the target device (on-chip pull-up resistor cannot be used).
<2> After reset release, the port mode is the input mode (Hi-Z).
<3> Set the output latch of the corresponding port to 1.
<4> Set the corresponding bit of the POM3 and POM5 registers to 1 to set the N-ch open drain output (V_{DD} tolerance) mode.
<5> Enable the operation of the serial array unit and set the mode to the UART or simplified SPI (CSI) mode.
<6> Set the corresponding bit of the PM3 and PM5 registers to the output mode. At this time, the output data is high level, so the pin is in the Hi-Z state.

(3) Setting procedure when using I/O pins of IIC00 and IIC01 functions with a different potential (1.8 V, 2.5 V)

In case of simplified IIC00: P55, P56 (P30)
In case of simplified IIC01: P51, P52

Remark Functions in parentheses can be assigned via settings in the peripheral I/O redirection register 0 (PIOR0).

<1> Using an external resistor, pull up externally the pin to be used to the power supply of the target device (on-chip pull-up resistor cannot be used).
<2> After reset release, the port mode is the input mode (Hi-Z).
<3> Set the output latch of the corresponding port to 1.
<4> Set the corresponding bit of the POM3 and POM5 registers to 1 to set the N-ch open drain output (V_{DD} tolerance) mode.
<5> Set the corresponding bit of the PIM3 and PIM5 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 PM3 and PM5 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.

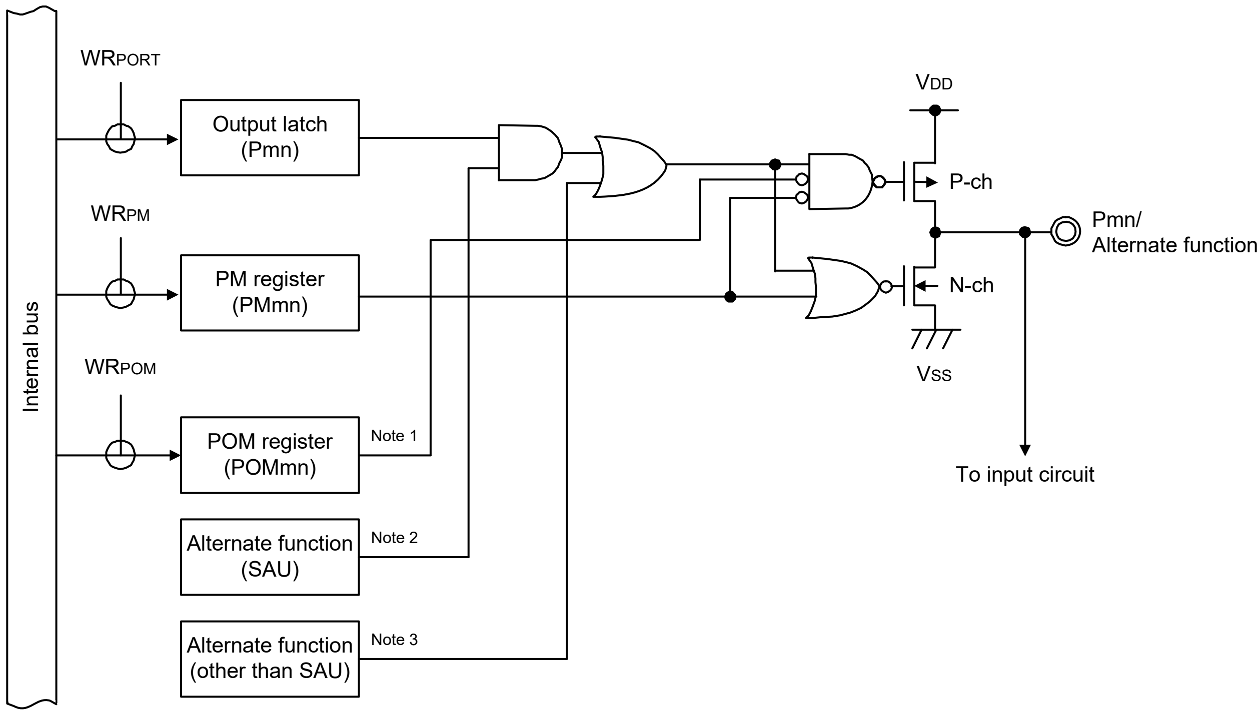
4.5 Register Settings When Using Alternate Function

4.5.1 Basic concept when using alternate function

In the beginning, for a pin also assigned to be used for analog function, use the port mode control register (PMCxx) to specify whether to use the pin for analog function or digital input/output.

Figure 4 - 8 shows the basic configuration of an output circuit for pins used for digital input/output. The output of the output latch for the port and the output of the alternate SAU function are input to an AND gate. The output of the AND gate is input to an OR gate. The output of an alternate function other than SAU (Timer, RTC2, clock/buzzer output, etc.) is connected to the other input pin of the OR gate. When such kind of pins are used by the port function or an alternate function, the unused alternate function must not hinder the output of the function to be used. An idea of basic settings for this kind of case is shown in Table 4 - 5.

Figure 4 - 8 Basic Configuration of Output Circuit for Pins



Note 1. When there is no POM register, this signal should be considered to be low level (0).

Note 2. When there is no alternate function, this signal should be considered to be high level (1).

Note 3. When there is no alternate function, this signal should be considered to be low level (0).

Table 4 - 5 Concept of Basic Settings

Output Function of Used Pin	Output Settings of Unused Alternate Function		
	Output Function for Port	Output Function for SAU	Output Function for other than SAU
Output function for port	—	Output is high (1)	Output is low (0)
Output function for SAU	High (1)	—	Output is low (0)
Output function for other than SAU	Low (0)	Output is high (1)	Output is low (0) Note

Note Since more than one output function other than SAU may be assigned to a single pin, the output of an unused alternate function must be set to low level (0). For details on the setting method, see **4.5.2 Register settings for alternate function whose output function is not used**.

4.5.2 Register settings for alternate function whose output function is not used

When the output of an alternate function of the pin is not used, the following settings should be made. Note that when the peripheral I/O redirection function is the target, the output can be switched to another pin by setting peripheral I/O redirection register 0 (PIOR0). This allows usage of the port function or other alternate 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). 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). 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 bit in timer output register 0 (TO0) to 0 (low). These are the same settings as the initial state.

(4) 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.

4.5.3 Register setting examples for used port and alternate functions

Register setting examples for used port and alternate functions are shown in Tables 4 - 6 to 4 - 14. The registers used to control the port functions should be set as shown in Tables 4 - 6 to 4 - 14. See the following remark for legends used in Tables 4 - 6 to 4 - 14.

Remark —: Not supported
×: Don’t care
PIORx: Peripheral I/O redirection register
POMxx: Port output mode register
PMCxx: Port mode control register
PMxx: Port mode register
Pxx: Port output latch
Functions in parentheses can be assigned via settings in peripheral I/O redirection register 0 (PIOR0).

Table 4 - 6 Setting Examples of Registers When Using P00 to P04 Pin Function

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		48 -pin	32 -pin	30 -pin	24 -pin	20 -pin
	Function Name	I/O						SAU Output Function	Other than SAU					
P00	P00	Input	—	—	—	1	×	—	—	√	×	×	×	×
		Output	—	—	—	0	0/1	—	—					
P01	P01	Input	—	—	—	1	×	—	×	√	×	×	×	×
		Output	—	—	—	0	0/1	—	PCLBUZ1 = 0					
	PCLBUZ1	Output	—	—	—	0	0/1	—	×	√	×	×	×	×
P02	P02	Input	—	—	0	1	×	—	—	√	×	×	×	×
		Output	—	—	0	0	0/1	—	—					
	ANI16	Analog input	—	—	1	1	×	—	—	√	×	×	×	×
P03	P03	Input	—	—	0	1	×	—	—	√	×	×	×	×
		Output	—	—	0	0	0/1	—	—					
	ANI17	Analog input	—	—	1	1	×	—	—	√	×	×	×	×
P04	P04	Input	—	—	0	1	×	—	—	√	×	×	×	×
		Output	—	—	0	0	0/1	—	—					
	ANI18	Analog input	—	—	1	1	×	—	—	√	×	×	×	×

Table 4 - 7 Setting Examples of Registers When Using P10 to P17 Pin Function

Pin Name	Used Function		ADM2	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		48-pin	32-pin	30-pin	24-pin	20-pin
	Function Name	I/O						SAU Output Function	Other than SAU					
P10	P10	Input	×	—	0	1	×	—	—	√	×	×	×	×
		Output	×	—	0	0	0/1	—	—					
	ANI0	Analog input	00x0xx0x, 10x0xx0x	—	1	1	×	—	—	√	×	×	×	×
	AVREFP	Reference power supply	01x0xx0x	—	1	1	×	—	—	√	×	×	×	×
P11	P11	Input	×	—	0	1	×	—	—	√	×	×	×	×
		Output	×	—	0	0	0/1	—	—					
	ANI1	Analog input	xx00xx0x	—	1	1	×	—	—	√	×	×	×	×
	AVREFM	Reference power supply	xx10xx0x	—	1	1	×	—	—	√	×	×	×	×
P12	P12	Input	—	—	0	1	×	—	—	√	√	√	√	√
		Output	—	—	0	0	0/1	—	—					
	ANI2	Analog input	—	—	1	1	×	—	—	√	√	√	√	√
	AMP0+	Analog input	—	—	1	1	×	—	—	√	√	√	√	√
P13	P13	Input	—	—	0	1	×	—	—	√	√	√	√	√
		Output	—	—	0	0	0/1	—	—					
	ANI3	Analog input	—	—	1	1	×	—	—	√	√	√	√	√
	AMP0-	Analog input	—	—	1	1	×	—	—	√	√	√	√	√
P14	P14	Input	—	—	0	1	×	—	×	√	√	√	√	√
		Output	—	—	0	0	0/1	—	AMPC. AMPE0 = 0					
	ANI4	Analog input	—	—	1	1	×	—	×	√	√	√	√	√
	IVCMP0	Analog input	—	—	1	1	×	—	×	√	√	√	√	√
	AMP0O	Analog output	—	—	1	1	×	—	×	√	√	√	√	√
P15	P15	Input	—	—	0	1	×	—	—	√	√	×	×	×
		Output	—	—	0	0	0/1	—	—					
	ANI5	Analog input	—	—	1	1	×	—	—	√	√	×	×	×
	AMP1+	Analog input	—	—	1	1	×	—	—	√	√	×	×	×
P16	P16	Input	—	—	0	1	×	—	—	√	√	×	×	×
		Output	—	—	0	0	0/1	—	—					
	ANI6	Analog input	—	—	1	1	×	—	—	√	√	×	×	×
	AMP1-	Analog input	—	—	1	1	×	—	—	√	√	×	×	×
P17	P17	Input	—	—	0	1	×	—	AMPC. AMPE1 = 0	√	√	×	×	×
		Output	—	—	0	0	0/1	—						
	ANI7	Analog input	—	—	1	1	×	—	×	√	√	×	×	×
	AMP1O	Analog output	—	—	1	1	×	—	×	√	√	×	×	×

Table 4 - 8 Setting Examples of Registers When Using P20 to P25 Pin Function

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		48 -pin	32 -pin	30 -pin	24 -pin	20 -pin
	Function Name	I/O						SAU Output Function	Other than SAU					
P20	P20	Input	—	—	—	1	×	—	AMPC. AMPE3 = 0	√	√	√	√	√
		Output	—	—	—	0	0/1	—						
	ANI13	Analog input	—	—	1	1	×	—	×	√	√	√	√	√
	IVCMP1	Analog input	—	—	1	1	×	—	×	√	√	√	√	√
	AMP3O	Analog output	—	—	1	1	×	—	×	√	√	√	√	√
P21	P21	Input	—	—	0	1	×	—	—	√	√	√	√	√
		Output	—	—	0	0	0/1	—	—					
	ANI12	Analog input	—	—	1	1	×	—	—	√	√	√	√	√
	AMP3-	Analog input	—	—	1	1	×	—	—	√	√	√	√	√
P22	P22	Input	—	—	0	1	×	—	—	√	√	√	√	√
		Output	—	—	0	0	0/1	—	—					
	ANI11	Analog input	—	—	1	1	×	—	—	√	√	√	√	√
	AMP3+	Analog input	—	—	1	1	×	—	—	√	√	√	√	√
P23	P23	Input	—	—	0	1	×	—	AMPC. AMPE2 = 0	√	√	√	×	×
		Output	—	—	0	0	0/1	—						
	ANI10	Analog input	—	—	1	1	×	—	×	√	√	√	×	×
	AMP2O	Analog output	—	—	1	1	×	—	×	√	√	√	×	×
P24	P24	Input	—	—	0	1	×	—	—	√	√	√	×	×
		Output	—	—	0	0	0/1	—	—					
	ANI9	Analog input	—	—	1	1	×	—	—	√	√	√	×	×
	AMP2-	Analog input	—	—	1	1	×	—	—	√	√	√	×	×
P25	P25	Input	—	—	0	1	×	—	—	√	√	√	×	×
		Output	—	—	0	0	0/1	—	—					
	ANI8	Analog input	—	—	1	1	×	—	—	√	√	√	×	×
	AMP2+	Analog input	—	—	1	1	×	—	—	√	√	√	×	×

Table 4 - 9 Setting Examples of Registers When Using P30 to P33 Pin Function

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		48 -pin	32 -pin	30 -pin	24 -pin	20 -pin
	Function Name	I/O						SAU Output Function	Other than SAU					
P30	P30	Input	×	×	0	1	×	×	×	√	√	√	√	√
		Output	×	0	0	0	0/1	(SCK00)/ (SCL00) = 1 Note 3	TO01 = 0					
		N ch-OD output	×	1	0	0	0/1							
	TI00	Input	PIOR01 = 0	×	0	1	×	×	×	√	√	√	√	√
	TO01	Output		0	0	0	0	(SCK00)/ (SCL00) = 1 Note 3	×					
	IVREF0	Analog input	×	×	1	1	×	×	×	√	√	√	√	√
	(SCK00)	Input	PIOR00 = 1 Note 1	×	0	1	×	×	×	√	√	√	√	√
		Output		0/1	0	0	1	×	TO01 = 0	√	√	√	√	√
	(SCL00)	Output		0/1	0	0	1	×		√	√	√	√	√
P31	P31	Input	×	—	0	1	×	—	×	√	√	√	√	√
		Output	×	—	0	0	0/1	—	TO00 = 0 PCLBUZ0 = 0					
	TI01	Input	PIOR02 = 0	—	0	1	×	—	×	√	√	√	√	√
	TO00	Output		—	0	0	0	—	PCLBUZ0 = 0	√	√	√	√	√
	PCLBUZ0	Output	×	—	0	0	0	—	TO00 = 0	√	√	√	√	√
	IVREF1	Analog input	×	—	1	1	×	—	×	√	√	√	√	√
P32	P32	Input	×	—	—	1	×	—	×	√	×	×	×	×
		Output	×	—	—	0	0/1	—	(TO00) = 0 Note 4					
	KR3	Input	×	—	—	1	×	—	×	√	×	×	×	×
	INTP6	Input	×	—	—	1	×	—	×	√	×	×	×	×
	(TI01)	Input	PIOR02 = 1	—	—	1	×	—	×	√	×	×	×	×
	(TO00)	Output		—	—	0	0	—	×	√	×	×	×	×
P33	P33	Input	×	—	—	1	×	—	×	√	×	√	×	×
		Output	×	—	—	0	0/1	—	(TO02) = 0 Note 5					
	INTP5	Input	×	—	—	1	×	—	×	√	×	√	×	×
	(TI02)	Input	PIOR03 = 1 Note 2	—	—	1	×	—	×	√	×	√	×	×
	(TO02)	Output		—	—	0	0	—	×	√	×	√	×	×

Note 1. Don't care when the 20-pin product is used.
Note 2. Don't care when the 30-pin product is used.
Note 3. Don't care when PIOR00 = 0.
Note 4. Don't care when PIOR02 = 0.
Note 5. Don't care when PIOR03 = 0.

Table 4 - 10 Setting Examples of Registers When Using P40 Pin Function

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		48 -pin	32 -pin	30 -pin	24 -pin	20 -pin
	Function Name	I/O						SAU Output Function	Other than SAU					
P40	P40	Input	—	—	—	1	×	—	—	√	√	√	√	√
		Output	—	—	—	0	0/1	—	—					

Table 4 - 11 Setting Examples of Registers When Using P50 to P57 Pin Function

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		48 -pin	32 -pin	30 -pin	24 -pin	20 -pin
	Function Name	I/O						SAU Output Function	Other than SAU					
P50	P50	Input	×	—	—	1	×	—	×	√	×	×	×	×
		Output	×	—	—	0	0/1	—	(TO01) = 0 Note 1					
	RTC1HZ	Output	×	—	—	0	0	—	×	√	×	×	×	×
	(TI00)	Input	PIOR01 = 1	—	—	1	×	—	×	√	×	×	×	×
	(TO01)	Output		—	—	0	0	—	×	√	×	×	×	×
P51	P51	Input	×	0	—	1	×	×	×	√	√	×	√	×
		Output	×	0	—	0	0/1	SCK01/SCL01 = 1	TO02 = 0					
		N ch-OD output	×	1	—	0	0/1							
	KR0	Input	×	×	—	1	×	×	×	√	√	×	√	×
	TI02	Input	PIOR03 = 0	×	—	1	×	×	×	√	√	×	√	×
	TO02	Output		0	—	0	0	SCK01/SCL01 = 1	×	√	√	×	√	×
	SCK01	Input	×	×	—	1	×	×	×	√	√	×	√	×
		Output	×	0/1	—	0	1	×	TO02 = 0	√	√	×	√	×
	SCL01	Output	×	0/1	—	0	1	×		√	√	×	√	×
P52	P52	Input	—	0	—	1	×	×	×	√	√	×	√	×
		Output	—	0	—	0	0/1	SDA00 = 1	TO03 = 0					
		N ch-OD output	—	1	—	0	0/1							
	KR1	Input	—	×	—	1	×	×	×	√	√	×	√	×
	TI03	Input	—	×	—	1	×	×	×	√	√	×	√	×
	TO03	Output	—	0	—	0	0	SDA00 = 1	×	√	√	×	√	×
	SI01	Input	—	×	—	1	×	×	×	√	√	×	√	×
	SDA01	I/O	—	1	—	0	1	×	TO03 = 0	√	√	×	√	×
P53	P53	Input	—	0	—	1	×	×	×	√	√	×	√	×
		Output	—	0	—	0	0/1	SO01 = 1	VCOUT0 = 0					
		N ch-OD output	—	1	—	0	0/1							
	KR2	Input	—	×	—	1	×	×	×	√	√	×	√	×
	SO01	Output	—	0/1	—	0	1	SO01 = 1	×	√	√	×	√	×
	VCOUT0	Output	—	0	—	0	0	×	×	√	√	×	√	×
P54	P54	Input	—	0	—	1	×	×	—	√	√	√	√	√
		Output	—	0	—	0	0/1	SO00 = 1 TxD0 = 1	—					
		N ch-OD output	—	1	—	0	0/1							
	INTP1	Input	—	×	—	1	×	×	—	√	√	√	√	√
	SO00	Output	—	0/1	—	0	1	×	—	√	√	√	√	√
	TxD0	Output	—	0/1	—	0	1	×	—	√	√	√	√	√
P55	P55	Input	—	0	—	1	×	×	—	√	√	√	√	√
		Output	—	0	—	0	0/1	SDA00 = 1	—					
		N ch-OD output	—	1	—	0	0/1							
	INTP2	Input	—	×	—	1	×	×	—	√	√	√	√	√
	SI00	Input	—	×	—	1	×	×	—	√	√	√	√	√
	RxD0	Input	—	×	—	1	×	×	—	√	√	√	√	√
	SDA00	I/O	—	1	—	0	1	×	—	√	√	√	√	√

Table 4 - 11 Setting Examples of Registers When Using P50 to P57 Pin Function

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		48 -pin	32 -pin	30 -pin	24 -pin	20 -pin
	Function Name	I/O						SAU Output Function	Other than SAU					
P56	P56	Input	—	0	—	1	×	×	—	√	√	√	√	×
		Output	—	0	—	0	0/1	SCK00 = 1 SCL00 = 1	—					
		N ch-OD output	—	1	—	0	0/1							
	INTP3	Input	—	×	—	1	×	×	—	√	√	√	√	×
	SCK00	Input	PIOR00 = 0	×	—	1	×	×	—	√	√	√	√	×
		Output		0/1	—	0	1	×	—	√	√	√	√	×
	SCL00	Output		0/1	—	0	1	×	—	√	√	√	√	×
P57	P57	Input	—	—	—	1	×	—	—	√	×	×	×	×
		Output	—	—	—	0	0/1	—	(TO03) = 0 Note 2 VCOUT1 = 0					
	INTP4	Input	—	—	—	1	×	—	×	√	×	×	×	×
	VCOUT1	Output	—	—	—	0	0	—	(TO03) = 0 Note 2	√	×	×	×	×
	(TI03)	Input	PIOR04 = 1	—	—	1	×	—	×	√	×	×	×	×
	(TO03)	Output		—	—	0	0	—	VCOUT1 = 0	√	×	×	×	×

Note 1. Don't care when PIOR01 = 0.

Note 2. Don't care when PIOR04 = 0.

Table 4 - 12 Setting Examples of Registers When Using P60 to P63 Pin Function

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		48 -pin	32 -pin	30 -pin	24 -pin	20 -pin
	Function Name	I/O						SAU Output Function	Other than SAU					
P60	P60	Input	—	—	—	1	×	—	—	√	×	×	×	×
		Output	—	—	—	0	0/1	—	—					
P61	P61	Input	—	—	—	1	×	—	—	√	×	×	×	×
		Output	—	—	—	0	0/1	—	—					
P62	P62	Input	—	—	—	1	×	—	—	√	×	×	×	×
		Output	—	—	—	0	0/1	—	—					
P63	P63	Input	—	—	—	1	×	—	—	√	×	×	×	×
		Output	—	—	—	0	0/1	—	—					
	$\overline{SSI00}$	Input	—	—	—	1	×	—	—	√	×	×	×	×

Table 4 - 13 Setting Examples of Registers When Using P121 to P125 Pin Function

Pin Name	Used Function		CMC (EXCLK, OSCSEL, EXCLKS, OSCSELS)	Pxx	48-pin	32-pin	30-pin	24-pin	20-pin
	Function Name	I/O							
P121	P121	Input	00xx/10xx/11xx	×	√	√	√	√	√
	X1	—	01xx	—	√	√	√	√	√
P122	P122	Input	00xx/10xx	×	√	√	√	√	√
	X2	—	01xx	—	√	√	√	√	√
	EXCLK	Input	11xx	×	√	√	√	√	√
P123	P123	Input	xx00/xx10/xx11	×	√	√	√	×	×
	XT1	—	xx01	—	√	√	√	×	×
P124	P124	Input	xx00/xx10	×	√	√	√	×	×
	XT2	—	xx01	—	√	√	√	×	×
	EXCLKS	Input	xx11	—	√	√	√	×	×
P125	P125	Input	xx00/xx10/xx11	×	√	√	√	×	×
	XT1	—	xx01	—	√	√	√	×	×

Table 4 - 14 Setting Examples of Registers When Using P130 and P137 Pin Function

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		48-pin	32-pin	30-pin	24-pin	20-pin
	Function Name	I/O						SAU Output Function	Other than SAU					
P130	P130	Input	—	—	—	1	×	—	—	√	×	×	×	×
		Output	—	—	—	0	0/1	—	—	√	×	×	×	×
P137	P137	Input	—	—	—	—	×	—	—	√	√	√	√	√
	INTP0	Input	—	—	—	—	×	—	—					

4.6 Cautions When Using Port Function

4.6.1 Cautions on 1-Bit Manipulation Instruction for Port Register n (Pn)

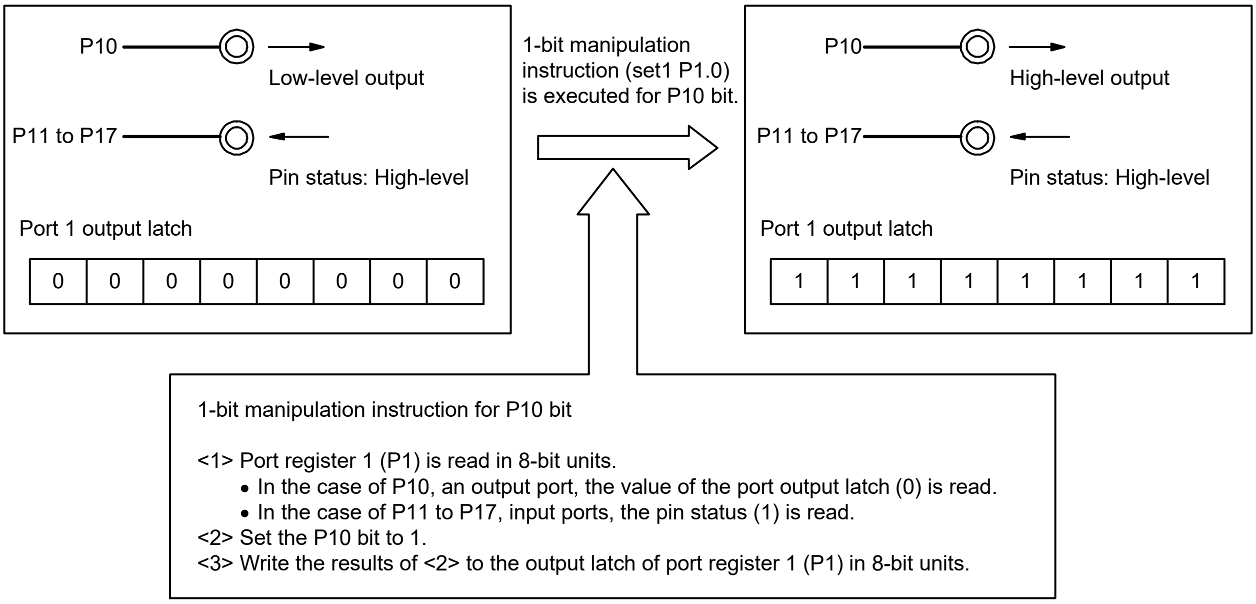
When a 1-bit manipulation instruction is executed on a port that provides both input and output functions, the output latch value of an input port that is not subject to manipulation may be written in addition to the targeted bit.
Therefore, it is recommended to rewrite the output latch when switching a port from input mode to output mode.

<Example> When P10 is an output port, P11 to P17 are input ports (all pin statuses are high level), and the port latch value of port 1 is 00H, if the output of output port P10 is changed from low level to high level via a 1-bit manipulation instruction, the output latch value of port 1 is FFH.

Explanation: The targets of writing to and reading from the Pn register of a port whose PMnm bit is 1 are the output latch and pin status, respectively.
A 1-bit manipulation instruction is executed in the following order in the RL78/I1D.

<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.
In step <1>, the output latch value (0) of P10, which is an output port, is read, while the pin statuses of P11 to P17, which are input ports, are read. If the pin statuses of P11 to P17 are high level at this time, the read value is FEH.
The value is changed to FFH by the manipulation in <2>.
FFH is written to the output latch by the manipulation in <3>.

Figure 4 - 9 Bit Manipulation Instruction (P10)



4.6.2 Notes on specifying the pin settings

For an output pin to which multiple functions are assigned, the output of the unused alternate functions must be set to its initial state so as to prevent conflicting outputs. This also applies to the functions assigned by using the peripheral I/O redirection register 0 (PIOR0). For details about the alternate function output, see **4.5 Register Settings When Using Alternate Function**.

No specific setting is required for input pins because the output of their alternate functions is disabled (the buffer output is Hi-Z).

Disabling the unused functions, including blocks that are only used for input or do not have I/O, is recommended for lower power consumption.

CHAPTER 5 OPERATION STATE CONTROL

The operating voltage, operating timing, and operating current of the internal circuit are optimized using flash operation modes. Select an appropriate flash operation mode according to the supply voltage range and clock frequencies used to operate the MCU.
The flash operation mode set by the option byte is selected for operation immediately after a reset release. Then, each mode is selected according to register settings.

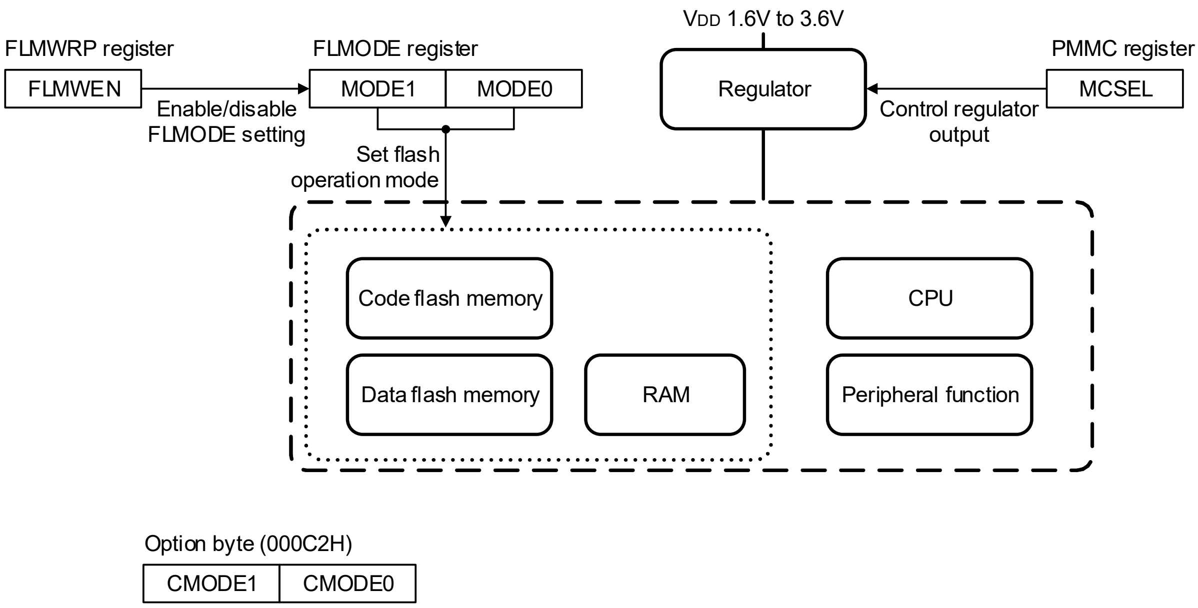
5.1 Configuration of Operation State Control

Operation state control is supported by the following hardware.

Table 5 - 1 Configuration of Operation State Control

Item	Configuration
Option byte	• User option byte address: 000C2H
Control registers	• Flash operating mode select register (FLMODE) • Flash operating mode protect register (FLMWRP) • Regulator mode control register (PMMC)

Figure 5 - 1 Block Diagram of Operation State Control



There are the following four flash operation modes.

- HS (high-speed main) mode
- LS (low-speed main) mode
- LV (low-voltage main) mode
- LP (low-power main) mode

The MCU can be operated efficiently by setting these flash operation modes according to MCU operating conditions.
Table 5 - 2 lists the Features of Each Flash Operation Mode.

Table 5 - 2 Features of Each Flash Operation Mode

Flash Operation Mode	Regulator Mode	Recommended Operating Range		Operating Current (typ.)	Description
HS (high-speed main) mode	Normal setting only (MCSEL = 0)	2.4 V to 3.6 V	1 MHz to 24 MHz	3.2 mA (during operation at 24 MHz Note 1)	High-speed CPU operation (at 24 MHz (max.)) is possible in this mode. Suitable when CPU processing capacity is required.
LS (low-speed main) mode	Normal setting (MCSEL = 0)	1.8 V to 3.6 V	1 MHz to 8 MHz	1.1 mA (during operation at 8 MHz Note 1)	The operating current and CPU operation processing (at 8 MHz (max.)) are well-balanced in this mode. To operate the CPU at 4 to 8 MHz, set regulator mode to the normal setting. When operating the CPU at 1 to 4 MHz, the operating current can be reduced by setting regulator mode to the low-power consumption setting.
	Low-power consumption setting (MCSEL = 1)	1.8 V to 3.6 V	1 MHz to 4 MHz	0. 58 mA (during operation at 4 MHz Note 2)	
LP (low-power main) mode	Low-power consumption setting only (MCSEL = 1)	1.8 V to 3.6 V	1 MHz	0.124 mA (during operation at 1 MHz Note 2)	The CPU operates at 1 MHz in this mode. Low operating current is realized at 1 MHz.
LV (low-voltage main) mode Note 1	Normal setting only (MCSEL = 0)	1.6 V to 3.6 V	1 MHz to 4 MHz	1.2 mA (during operation at 4 MHz)	Low-voltage operation up to 1.6 V is possible in this mode. To operate the CPU at the supply voltage range of 1.6 to 1.8 V, select this mode.

Note 1. Operable only with the high-speed on-chip oscillator.
Note 2. When the middle-speed on-chip oscillator operates.

5.2 Registers Controlling Operation State Control

Operation state control is controlled by the following registers.

- Flash operating mode select register (FLMODE)
- Flash operating mode protect register (FLMWRP)
- Regulator mode control register (PMMC)

5.2.1 Flash operating mode select register (FLMODE)

The FLMODE register is an 8-bit register used to control flash operation modes and operation of the code 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 FLMWEN in the flash operation mode protect register (FLMWRP) is 0.

Reset generation updates MODE1 and MODE0 with the set value of CMODE1 and CMODE0 in the option byte (address: 000C2H).

Figure 5 - 2 Format of Flash operating mode select register (FLMODE)

Address: F00AAH After reset: 00H/80H/C0H Note 1 R/W

Symbol	<7>	<6>	5	4	3	2	1	0
FLMODE	MODE1	MODE0	0	0	0	0	0	0

MODE1	MODE0	Selection of flash operation mode
0	0	LV (low-voltage main) mode (Selectable when 1 MHz ≤ f_{CLK} ≤ 4 MHz in LS mode.)
0	1	LP (low-power main) mode (Selectable when 1.8 V ≤ V_{DD} ≤ 3.6 V and f_{CLK} = 1 MHz in LS mode. Note 2)
1	0	LS (low-speed main) mode (Selectable when 1.8 V ≤ V_{DD} ≤ 3.6 V and 1 MHz ≤ f_{CLK} ≤ 8 MHz in HS mode, LP mode, or LV mode.)
1	1	HS (high-speed main) mode (Selectable when 2.4 V ≤ V_{DD} ≤ 3.6 V in LS mode.)

Note 1. 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 (address: 000C2H).

Note 2. After LP (low-power main) mode is selected, set the MCSEL bit in the regulator mode control register (PMMC) to 1.

(Cautions are on the next page.)

Caution 1. The value of the FLMODE register cannot be changed when the FLMWEN bit in the flash operation mode protect register (FLMWRP) is 0. Also, do not change the value of the FLMODE register when the MCSEL bit in the regulator mode control register is 1.
When changing the value of the FLMODE register, first set the FLMWEN bit in the FLMERP register to 1 while MCSEL is 0. After the value of the FLMODE register is changed, set the FLMWEN bit to 0.

Caution 2. The MODE1 and MODE0 bits cannot be set when the CSS bit in the system clock control register (CKC) is 1 (CPU/peripheral function operates on subsystem clock).

Caution 3. Do not change the value of the MODE1 and MODE0 bits using the DTC.

Caution 4. When changing the flash operation mode, make sure that operation is possible within the voltage range and operating frequency range in the changed flash operation mode before changing the mode.

Caution 5. The middle-speed on-chip oscillator cannot be used in LV (low-voltage main) mode. When entering LV mode, first switch the operating clock to an oscillator other than the middle-speed on-chip oscillator, and then enter LV mode.

Caution 6. When the flash operation mode is changed by the MODE1 and MODE0 bits, the CPU enters a wait state for the following time until the mode changes. Interrupt requests are held pending during this wait period.

Table 5 - 3 Flash Operation Mode Change Time

Flash Operation Mode Change	Change Time
LS (low-speed main) mode → HS (high-speed main) mode	225 cycles Note 1
LS (low-speed main) mode → LV (low-voltage main) mode	99 cycles Note 1, 2
LP (low-power main) mode → LS (low-speed main) mode	10 cycles Note 1
LS (low-speed main) mode → LP (low-power main) mode	10 cycles Note 1
LV (low-voltage main) mode → LS (low-speed main) mode	20 cycles Note 1
HS (high-speed main) mode → LS (low-speed main) mode	30 cycles Note 1

Note 1. The cycle of the CPU/peripheral hardware clock (f_{CLK})

Note 2. Switching of the mode from LS (low-speed main) mode to LV (low-voltage main) mode must proceed while oscillation of the high-speed on-chip oscillator is stable.

Caution 7. When rewriting the FLMODE register, insert one or more clock cycles after rewriting the FLMODE register and before writing to this register. Do not write to the FLMODE register successively.

Caution 8. Do not change the FLMODE register when rewriting the flash memory.

5.2.2 Flash operating mode protect register (FLMWRP)

The FLMWRP register is an 8-bit register used to control access to the flash operation mode select register.
This register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset generation sets this register to 00H.

Figure 5 - 3 Format of Flash operating mode protect register (FLMWRP)

Address: F00ABH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
FLMWRP	0	0	0	0	0	0	0	FLMWEN

FLMWEN	Control of flash operation mode select register (FLMODE)
0	Rewriting the FLMODE register is disabled
1	Rewriting the FLMODE register is enabled

5.2.3 Regulator mode control register (PMMC)

The PMMC register is an 8-bit register used to control the mode of the on-chip regulator.
This register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset generation sets this register to 00H.

Figure 5 - 4 Format of Regulator mode control register (PMMC)

Address: F00F8H After reset: 00H R/W

Symbol	7	<6>	5	4	3	2	1	0
PMMC	0	MCSEL	0	0	0	0	0	0

MCSEL	Control of regulator mode
0	Normal setting
1	Low-power consumption setting

Caution 1. Do not change the flash operation mode select register (FLMODE) when MCSEL is 1.
Caution 2. Do not set MCSEL to 1 in HS (high-speed main) mode and LV (low-voltage main) mode.
Caution 3. In LS (low-speed main) mode, transition to the STOP mode is prohibited when MCSEL is 1.

5.3 Initial Setting of Flash Operation Modes

The option byte (000C2H) is used to set the initial state of flash operation mode and the high-speed on-chip oscillator after a reset is released.

Set an appropriate flash operation mode according to the V_{DD} voltage and the high-speed on-chip oscillator frequency at a reset release.

When a reset is released, the value of CMODE1 and CMODE0 is updated in MODE1 and MODE0 in the flash operation mode select register (FLMODE) and the value of FRQSEL3 to FRQSEL0 is updated in the high-speed on-chip oscillator frequency select register (HIODIV).

Figure 5 - 5 Format of User option byte (000C2H)

Address: 000C2H

Symbol	7	6	5	4	3	2	1	0
	CMODE1	CMODE0	1	0	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0

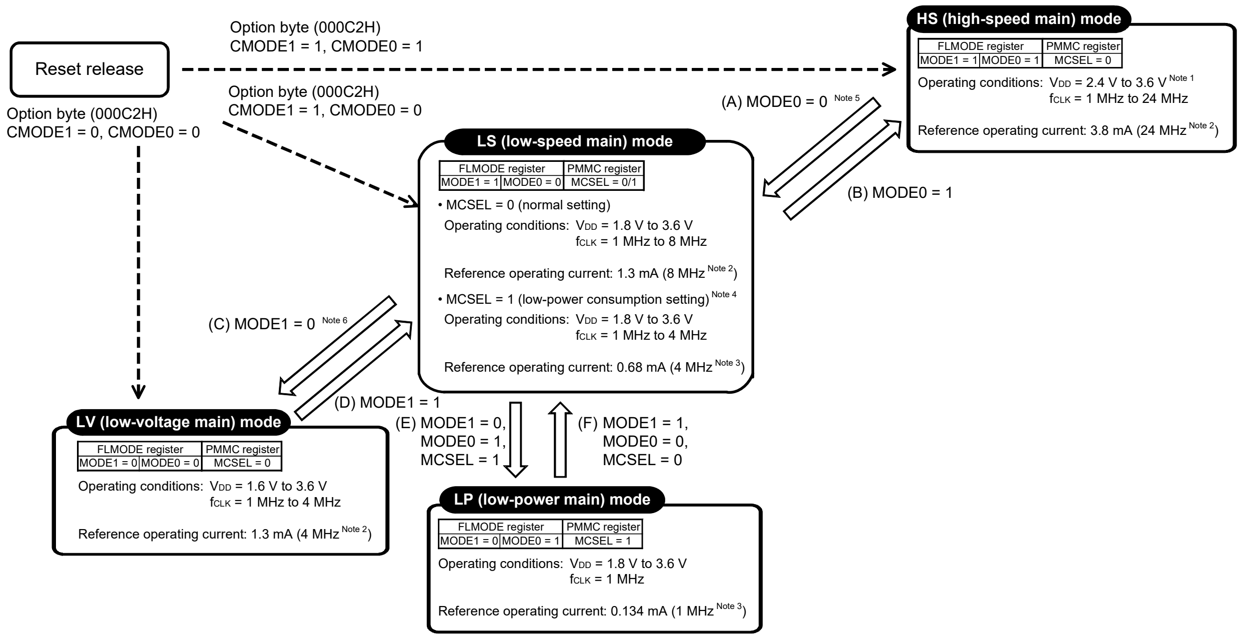
CMODE1	CMODE0	Selection of flash operation mode after reset release
0	0	LV (low-voltage main) mode
1	0	LS (low-speed main) mode
1	1	HS (high-speed main) mode
Other than above		Setting prohibited

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	High-speed on-chip oscillator frequency
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

5.4 Transitions between Flash Operation Modes

HS (high-speed main) mode, LS (low-speed main) mode, or LV (low-voltage main) mode can be selected as the flash operation mode immediately after a reset release, by setting CMODE1 and CMODE0 in the option byte (000C2H). The value of CMODE1 and CMODE0 is updated in the MODE1 and MODE0 bits in the flash operation mode select register (FLMODE). After that, the flash operation mode can be changed by changing the value of the FLMODE register during CPU operation.

Figure 5 - 6 State Transitions between Flash Operation Modes



Note 1. The operating voltage range is 2.7 V < V_{DD} ≤ 3.6 V when the operating frequency is 16 MHz < f_{CLK} ≤ 24 MHz.

Note 2. Current when the high-speed on-chip oscillator operates.

Note 3. Current when the middle-speed on-chip oscillator operates.

Note 4. Transitions between flash operation modes or transition to the STOP mode cannot be made when MCSEL = 1 (low-power consumption setting). When changing the flash operation mode or making transition to the STOP mode, be sure to set MCSEL = 0 (normal setting) before changing the mode.

Note 5. When CMODE1 and CMODE0 of the option byte (000C2H) are set to 1, operation is not guaranteed if a reset is generated while the operating voltage is 2.4 V or lower after entry to the LS (low-speed main) mode.

Note 6. When CMODE1 and CMODE0 of the option byte (000C2H) are set to 1 and 0 respectively, operation is not guaranteed if a reset is generated while the operating voltage is 1.8 V or lower after entry to the LV (low-voltage main) mode.

Caution **When a reset is applied while the MCU operates, operation always starts in the flash operation mode 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 flash operation mode set in the option byte.**

5.5 Details of Flash Operation Modes

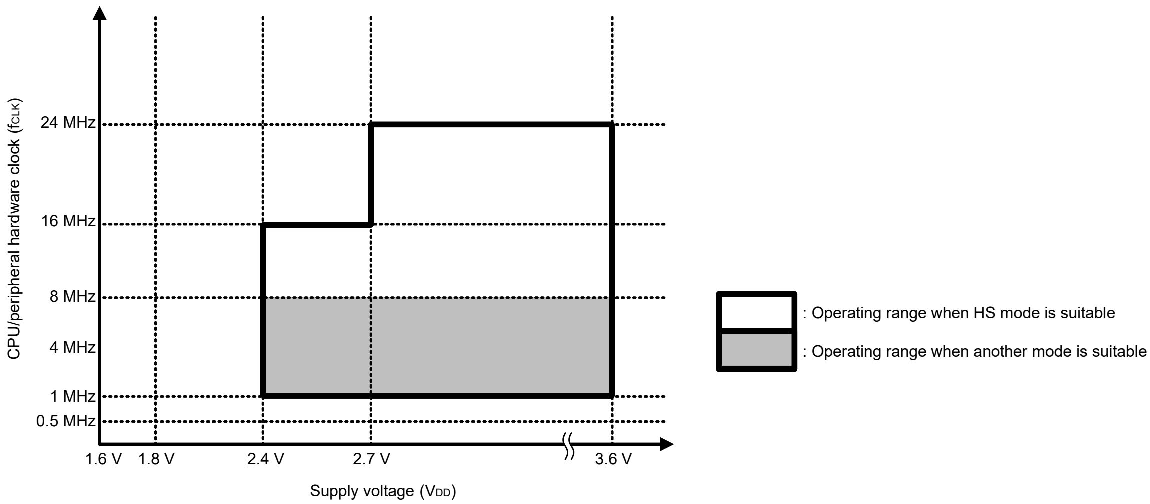
5.5.1 Details of HS (high-speed main) mode

HS (high-speed main) mode is suitable for applications that require CPU high-speed processing.

HS mode can be operated immediately after a reset release. Also, this mode can be entered from LS (low-speed main) mode. When entering HS mode, make sure that the supply voltage is 2.4 V ≤ V_{DD} ≤ 3.6 V in LS mode and the operating frequency is 1 MHz ≤ f_{CLK} ≤ 8 MHz.

The suitable operating range in HS mode is when the supply voltage is 2.4 V ≤ V_{DD} ≤ 3.6 V and the operating frequency is 8 MHz < f_{CLK} ≤ 16 MHz or when the supply voltage is 2.7 V ≤ V_{DD} ≤ 3.6 V and the operating frequency is 8 MHz < f_{CLK} ≤ 24 MHz. When 8 MHz or lower is used for operation, another mode can be used as the suitable flash operation mode.

Figure 5 - 7 Operating Range in HS Mode



5.5.2 Details of LS (low-speed main) mode

LS (low-speed main) mode supports both CPU processing capacity and operating voltage performance, suitable for applications that require low-power consumption at 1 to 8 MHz.

LS mode can be operated immediately after a reset release. Also, this mode can be entered from HS (high-speed main) mode, LV (low-voltage main) mode, or LP (low-power main) mode. When entering from HS mode to LS mode, make sure that the operating frequency is 1 MHz ≤ f_{CLK} ≤ 8 MHz.

In LS mode, low-power consumption can be set by the MCSEL bit in the regulator mode control register (PMMC). When setting low-power consumption, set the MCSEL bit to 1 while the operating frequency is 1 MHz ≤ f_{CLK} ≤ 4 MHz.

The suitable operating range in LS mode is when the supply voltage is 1.8 V ≤ V_{DD} ≤ 3.6 V and the operating frequency is 4 MHz < f_{CLK} ≤ 8 MHz if MCSEL = 0, and when the supply voltage is 1.8 V ≤ V_{DD} ≤ 3.6 V and the operating frequency is 1 MHz < f_{CLK} ≤ 4 MHz if MCSEL = 1.

Figure 5 - 8 Operating Range in LS Mode (MCSEL = 0)

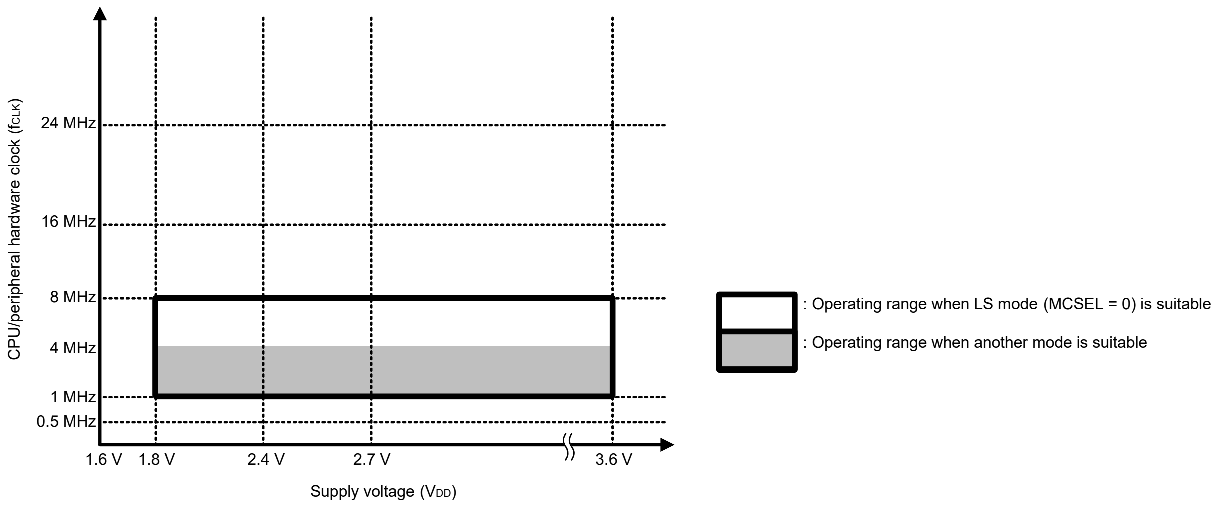
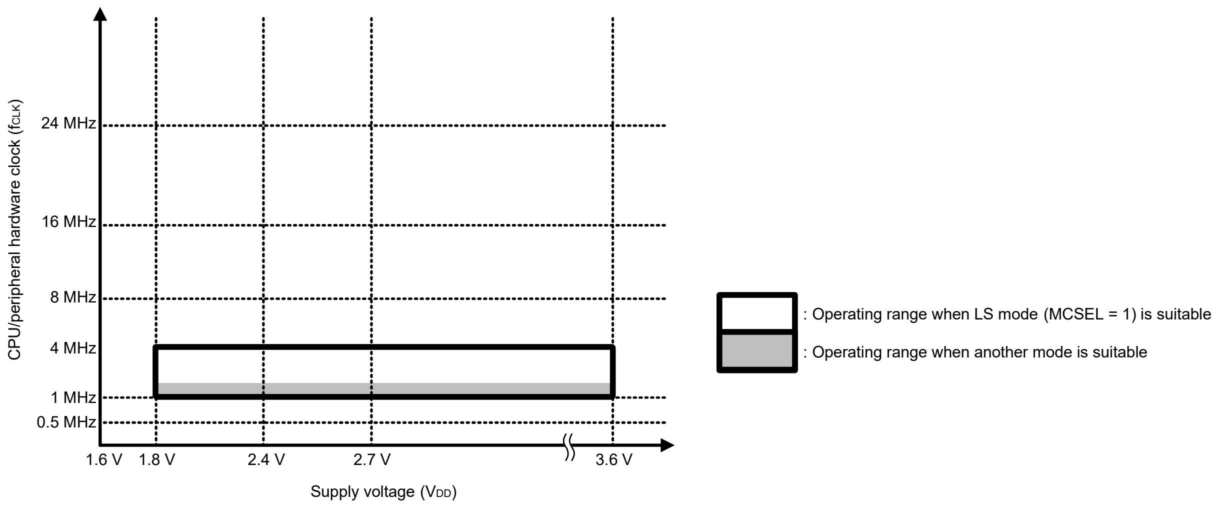


Figure 5 - 9 Operating Range in LS Mode (MCSEL = 1)



Caution When entering another flash operation mode, make sure that MCSEL = 0.

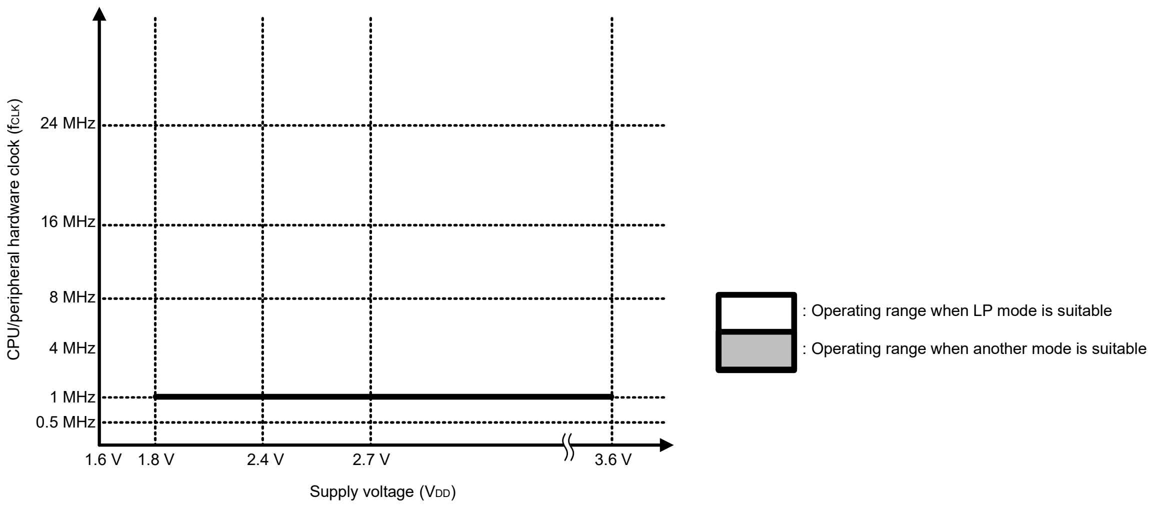
5.5.3 Details of LP (low-power main) mode

LP (low-power main) mode can be use to operate the CPU on low power at a 1-MHz frequency.

LP mode can be entered from LS (low-speed main) mode. When entering from LS mode to LP mode, make sure the operating frequency is f_{CLK} = 1 MHz. After the mode is entered, set the MCSEL bit in the regulator mode control register to 1.

The suitable operating range in LP mode is when the supply voltage is 1.8 V ≤ V_{DD} ≤ 3.6 V and the operating frequency is 1 MHz.

Figure 5 - 10 Operating Range in LP Mode



Caution When entering LS (low-speed main) mode, make sure that MCSEL = 0.

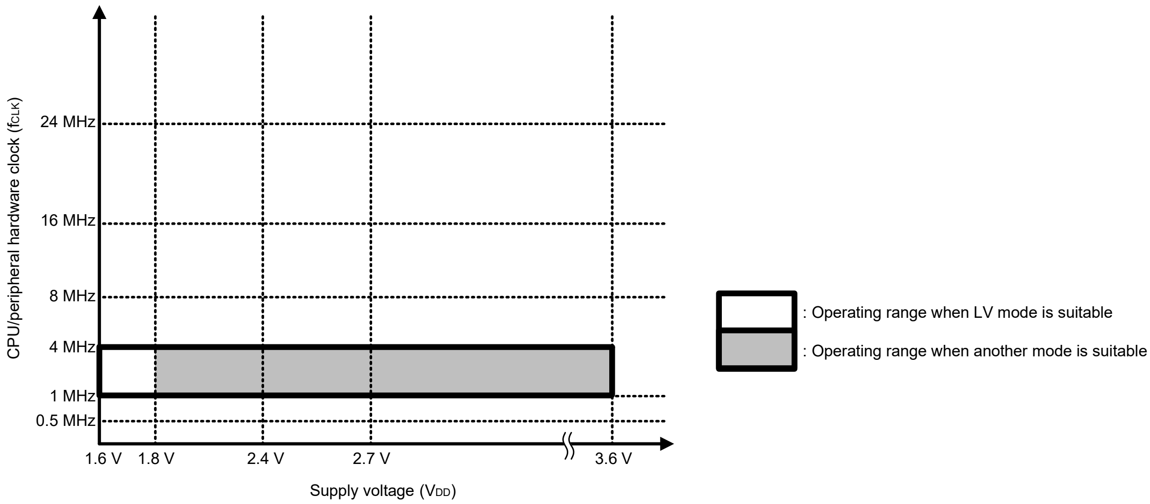
5.5.4 Details on LV (low-voltage main) mode

LV (low-voltage main) mode is suitable for applications that require operation at 1.8 V or lower.

LV mode can be operated immediately after a reset release. Also, this mode can be entered from LS (low-speed main) mode. When entering from LS mode to LV mode, make sure that the operating frequency is 1 MHz ≤ f_{CLK} ≤ 4 MHz.

The suitable operating range in LV mode is when the supply voltage is 1.6 V ≤ V_{DD} < 1.8 V. When a supply voltage of 1.8 V ≤ V_{DD} ≤ 3.6 V is used for operation, another mode can be used as the suitable flash operation mode.

Figure 5 - 11 Operating Range in LV Mode



Caution **The middle-speed on-chip oscillator cannot be used in LV (low-voltage main) mode. When entering LV mode, first switch the operating clock to an oscillator other than the middle-speed on-chip oscillator, and then enter LV mode.**

CHAPTER 6 CLOCK GENERATOR

The presence or absence of connecting resonator pin for main system clock, connecting resonator pin for subsystem clock, external clock input pin for main system clock, and external clock input pin for subsystem clock, depends on the product.

	20, 24-pin products	30, 32, 48-pin products
X1, X2 pins	√	√
EXCLK pin	√	√
XT1, XT2 pins	—	√
EXCLKS pin	—	√

6.1 Functions of Clock Generator

The clock generator generates the clock to be supplied to the CPU and peripheral hardware.
The following three kinds of system clocks and clock oscillators are selectable.

(1) Main system clock

<1> X1 oscillator

This circuit oscillates a clock of f_X = 1 to 20 MHz by connecting a resonator to X1 pin and X2 pin. Oscillation can be stopped by executing the STOP instruction or setting of the MSTOP bit (bit 7 of the clock operation status control register (CSC)).

<2> High-speed on-chip oscillator

The frequency at which to oscillate can be selected from among f_{IH} = 24, 16, 12, 8, 6, 4, 3, 2, or 1 MHz (TYP.) by using the option byte (000C2H). After a reset release, the CPU always starts operating with this high-speed on-chip oscillator clock. Oscillation can be stopped by executing the STOP instruction or setting of the HIOSTOP bit (bit 0 of the CSC register).

The frequency specified by using an 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 - 12 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 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
2.7 V ≤ V_{DD} ≤ 3.6 V	√	√	√	√	√	√	√	√	√
2.4 V ≤ V_{DD} ≤ 3.6 V	√	√	√	√	√	√	√	√	—
1.8 V ≤ V_{DD} ≤ 3.6 V	√	√	√	√	√	√	—	—	—
1.6 V ≤ V_{DD} ≤ 3.6 V	√	√	√	√	—	—	—	—	—

<3> Middle-speed on-chip oscillator

The frequency at which to oscillate can be selected from among f_{IM} = 4, 2, 1 MHz (TYP.) by setting of the MOCODIV bit (bits 0, 1 of the MOCODIV register). Oscillation can be stopped by executing the STOP instruction or clearing of the MIOEN bit (bit 1 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. An external main system clock input can be disabled by executing the 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 (bits 4 and 0 of the system clock control register (CKC)).
However, note that the usable frequency range of the main system clock differs depending on the V_{DD} power supply voltage setting. The operating voltage of the flash memory must be set by using the CMODE0 and CMODE1 bits of the option byte (000C2H) (see **CHAPTER 29 OPTION BYTE**).

(2) Subsystem clock

<1> XT1 clock oscillator

This circuit oscillates a clock of f_{XT} = 32.768 kHz by connecting a 32.768 kHz resonator to XT1 pin and XT2 pin. Oscillation can be stopped by setting the XTSTOP bit (bit 6 of the clock operation status control register (CSC)).

An external subsystem clock (f_{EXS} = 32.768 kHz) can also be supplied from the EXCLKS/XT2/P124 pin. An external subsystem clock input can be disabled by the setting of the XTSTOP bit.

<2> Low-speed on-chip oscillator

This circuit oscillates a clock of f_{IL} = 15 kHz (TYP.).

Low-speed on-chip oscillator operates when following one or more bits are 1, bit 4 (WDTON) of option byte (000C0H), bit 4 (WUTMMCK0) of subsystem clock supply mode control register (OSMC), bit 0 (SELLOSC) of subsystem clock select register (CKSEL).
However, if HALT or STOP instruction is executed when WDTON = 1, WUTMMCK0 = 0, SELLOSC = 0 and bit 0 (WDSTBYON) of option byte (000C0H) is 0, this oscillator stops the oscillation.

Remark f_X: X1 clock oscillation frequency
f_{IH}: High-speed on-chip oscillator clock frequency (24 MHz max.)
f_{IM}: Middle-speed on-chip oscillator clock frequency
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 frequency

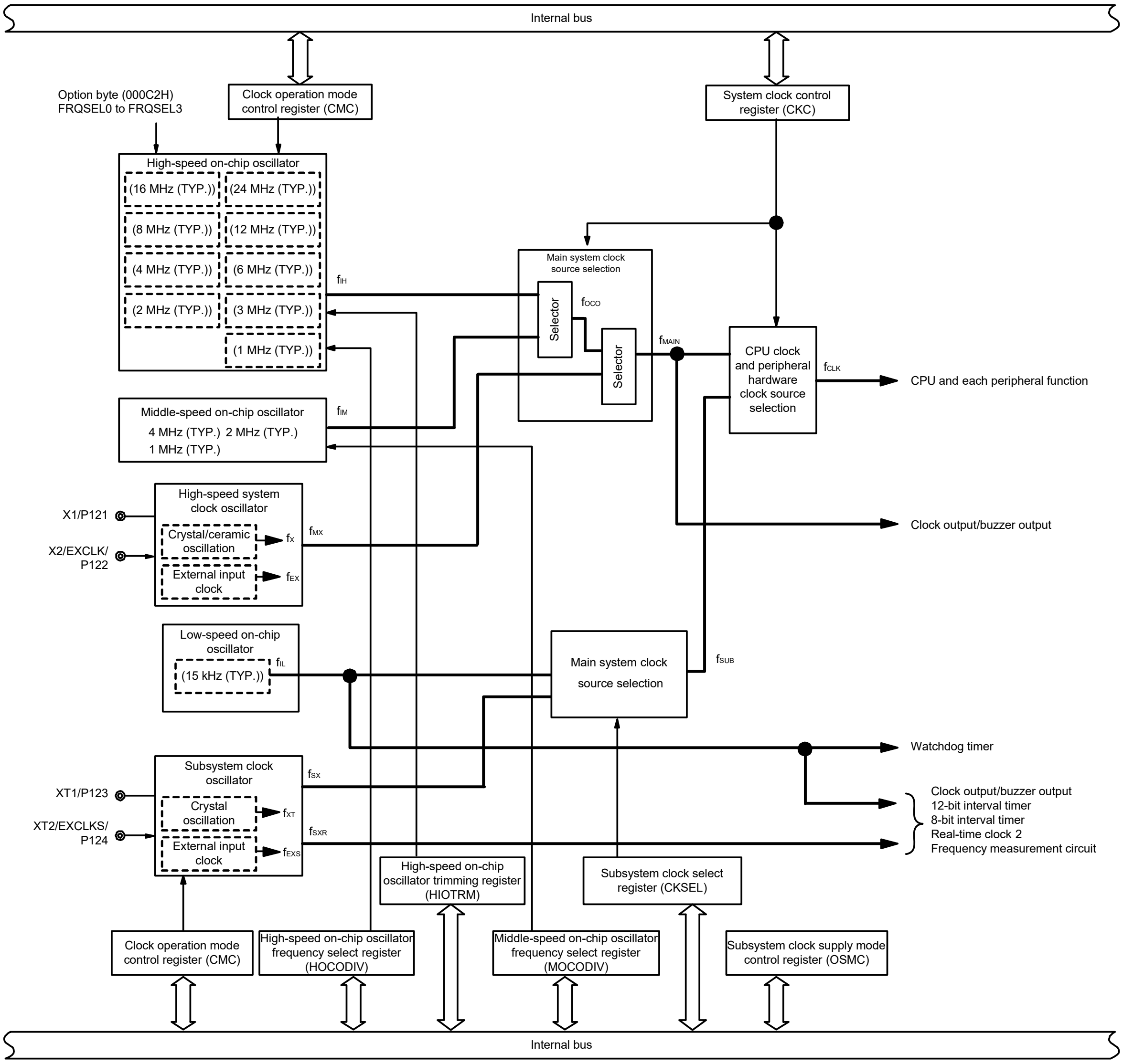
6.2 Configuration of Clock Generator

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, 2 (PER0, PER1, PER2) Subsystem clock supply mode control register (OSMC) High-speed on-chip oscillator frequency select register (HOCODIV) High-speed on-chip oscillator trimming register (HIOTRM) Subsystem clock select register (CKSEL) Middle-speed on-chip oscillator frequency select register (MOCODIV) Frequency measurement clock select register (FMCKS)
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 f_X: X1 clock oscillation frequency

f_{IH}: High-speed on-chip oscillator clock frequency (24 MHz max.)

f_{IM}: Middle-speed on-chip oscillator clock frequency (4 MHz max.)

f_{EX}: External main system clock frequency

f_{MX}: High-speed system clock frequency

f_{MAIN}: Main system clock frequency

f_{XT}: XT1 clock oscillation frequency

f_{EXS}: External subsystem clock frequency

f_{SX}: Subsystem clock oscillator clock frequency

f_{SXR} Subsystem clock oscillation circuit and RTC2/other clock frequency

f_{SUB}: Subsystem clock frequency

f_{CLK}: CPU/peripheral hardware clock frequency

f_{IL}: Low-speed on-chip oscillator clock frequency

f_{OCO}: Main on-chip oscillator clock frequency (f_{IH} or f_{IM})

6.3 Registers Controlling Clock Generator

The following registers are used to control the clock generator.

- Clock operation mode control register (CMC)
- System clock control register (CKC)
- Clock operation status control register (CSC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)
- Peripheral enable registers 0, 1, 2 (PER0, PER1, PER2)
- Subsystem clock supply mode control register (OSMC)
- High-speed on-chip oscillator frequency select register (HOCODIV)
- High-speed on-chip oscillator trimming register (HIOTRM)
- Subsystem clock select register (CKSEL)
- Middle-speed on-chip oscillator frequency select register (MOCODIV)

Caution Which registers and bits are included depends on the product. Be sure to set registers and bits that are not mounted in a product to their initial values.

6.3.1 Clock operation mode control register (CMC)

This register is used to set the operation mode of the X1/P121, X2/EXCLK/P122, XT1/P123, and XT2/EXCLKS/P124 pins, and to select a gain of the oscillator.

The CMC register can be written only once by an 8-bit memory manipulation instruction after reset release. This register can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6 - 2 Format of Clock operation mode control register (CMC)

Address: FFFA0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CMC	EXCLK	OSCSEL	EXCLKS Note	OSCSELS Note	0	AMPHS1 Note	AMPHS0 Note	AMPH

EXCLK	OSCSEL	High-speed system clock pin operation mode	X1/P121 pin	X2/EXCLK/P122 pin
0	0	Input port mode	Input port	
0	1	X1 oscillation mode	Crystal/ceramic resonator connection	
1	0	Input port mode	Input port	
1	1	External clock input mode	Input port	External clock input

EXCLKS Note	OSCSELS Note	Subsystem clock pin operation mode	XT1/P123 pin	XT2/EXCLKS/P124 pin
0	0	Input port mode	Input port	
0	1	XT1 oscillation mode	Crystal resonator connection	
1	0	Input port mode	Input port	
1	1	External clock input mode	Input port	External clock input

AMPHS1 Note	AMPHS0 Note	XT1 oscillator oscillation mode selection
0	0	Low power consumption oscillation (default)
0	1	Normal oscillation
1	0	Ultra-low power consumption oscillation
1	1	Setting prohibited

AMPH	Control of X1 clock oscillation frequency
0	1 MHz ≤ fx ≤ 10 MHz
1	10 MHz < fx ≤ 20 MHz

Note The EXCLKS, OSCSELS, AMPHS1, and AMPHS0 bits are reset only by a power-on reset; they retain the values when a reset caused by another factor occurs.

Caution 1. The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction. When using the CMC register with its initial value (00H), be sure to set the register to 00H after a reset ends in order to prevent malfunction due to a program loop. Such a malfunction becomes unrecoverable when a value other than 00H is mistakenly written.

Caution 2. After reset release, set the CMC register before X1 or XT1 oscillation is started as set by the clock operation status control register (CSC).

Caution 3. Be sure to set the AMPH bit to 1 if the X1 clock oscillation frequency exceeds 10 MHz.

Caution 4. Specify the settings for the AMPH, AMPHS1, and AMPHS0 bits while f_{IH} is selected as f_{CLK} after a reset ends (before f_{CLK} is switched to f_{MX} or f_{SUB}).

Caution 5. Oscillation stabilization time of f_{XT}, counting on the software.

Caution 6. Although the maximum system clock frequency is 24 MHz, the maximum frequency of the X1 oscillator is 20 MHz.

(Cautions and Remark are given on the next page.)

Caution 7. 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 the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) as the mode of the XT1 oscillator, evaluate the resonators described in 6.7 Resonator and Oscillator Constants.**
- **Make the wiring between the XT1 and XT2 pins and the resonators as short as possible, and minimize the parasitic capacitance and wiring resistance. Note this particularly when the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) is selected.**
- **Configure the circuit of the circuit board, using material with little parasitic capacitance and wiring resistance.**
- **Place a ground pattern that has the same potential as V_{SS} as much as possible near the XT1 oscillator.**
- **Be sure that the signal lines between the XT1 and XT2 pins, and the resonators do not cross with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **The impedance between the XT1 and XT2 pins may drop and oscillation may be disturbed due to moisture absorption of the circuit board in a high-humidity environment or dew condensation on the board. When using the circuit board in such an environment, take measures to damp-proof the circuit board, such as by coating.**
- **When coating the circuit board, use material that does not cause capacitance or leakage between the XT1 and XT2 pins.**

Remark f_X: X1 clock frequency

6.3.2 System clock control register (CKC)

This 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.

Reset signal generation sets this register to 00H.

Figure 6 - 3 Format of System clock control register (CKC)

Address: FFFA4H After reset: 00H R/W Note 1

Symbol	<7>	<6>	<5>	<4>	3	2	<1>	<0>
CKC	CLS	CSS Note 2	MCS	MCM0 Note 2	0	0	MCS1	MCM1 Note 2

CLS	Status of CPU/peripheral hardware clock (f_{CLK})
0	Main system clock (f_{MAIN})
1	Subsystem clock (f_{SUB})

CSS Note 2	Selection of CPU/peripheral hardware clock (f_{CLK})
0	Main system clock (f_{MAIN})
1	Subsystem clock (f_{SUB})

MCS	Status of 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	Status of 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 read-only.

Note 2. Changing the value of the MCM0 and MCM1 bits is prohibited while the CSS bit is set to 1.

Remark f_{IH}: High-speed on-chip oscillator clock frequency (24 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})

Caution 1. Be sure to set bits 2 and 3 of the CKC register 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 real-time clock 2, 12-bit interval timer, clock output/buzzer output, 8-bit interval timer, frequency measurement circuit, 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 are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 34 ELECTRICAL SPECIFICATIONS.

6.3.3 Clock operation status control register (CSC)

This register is used to control the operations of the high-speed system clock, high-speed on-chip oscillator clock, 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.

Reset signal generation sets this register to C0H.

Figure 6 - 4 Format of Clock operation status control register (CSC)

Address: FFFA1H After reset: C0H R/W

Symbol	<7>	<6>	5	4	3	2	<1>	<0>
CSC	MSTOP	XTSTOP Note	0	0	0	0	MIOEN	HIOSTOP

MSTOP	High-speed system clock operation control		
	X1 oscillation mode	External clock input mode	Input port mode
0	X1 oscillator operating	External clock from EXCLK pin is valid	Input port
1	X1 oscillator stopped	External clock from EXCLK pin is invalid	

XTSTOP Note	Subsystem clock operation control		
	XT1 oscillation mode	External clock input mode	Input port mode
0	XT1 oscillator operating	External clock from EXCLKS pin is valid	Input port
1	XT1 oscillator stopped	External clock from EXCLKS pin is invalid	

MIOEN	Middle-speed on-chip oscillator clock operation control
0	Middle-speed on-chip oscillator stopped
1	Middle-speed on-chip oscillator operating

HIOSTOP	High-speed on-chip oscillator clock operation control
0	High-speed on-chip oscillator operating
1	High-speed on-chip oscillator stopped

Note The XTSTOP bit is initialized after a reset by a power-on reset; it retains its value when a reset caused by another source occurs.

Caution 1. After reset release, 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 releasing reset. Note that if the OSTS register is being used with its default settings, the OSTS register is not required to be set here.

Caution 3. To start X1 oscillation as set by the MSTOP bit, check the oscillation stabilization time of the X1 clock by using the oscillation stabilization time counter status register (OSTC).

Caution 4. When starting XT1 oscillation by setting the XTSTOP bit, wait for oscillation of the subsystem clock to stabilize by setting a wait time using software.

Caution 5. Do not stop the clock selected for the CPU peripheral hardware clock (f_{CLK}) with the OSC register.

Caution 6. The setting of the flags of the register to stop clock oscillation (invalidate the external clock input) and the condition before clock oscillation is to be stopped are as Table 6 - 2.
When stopping the clock, confirm the condition before stopping clock.

Table 6 - 2 Stopping Clock Method

Clock	Condition Before Stopping Clock (Invalidating External Clock Input)	Setting of CSC Register Flags
X1 clock	CPU and peripheral hardware clocks operate with a clock other than the high-speed system clock. (CLS = 0 and MCS = 0, or CLS = 1)	MSTOP = 1
External main system clock		
XT1 clock	CPU and peripheral hardware clocks operate with a clock other than the subsystem clock. (CLS = 0)	XTSTOP = 1
External subsystem clock		
High-speed on-chip oscillator clock	CPU and peripheral hardware clocks operate with a clock other than the high-speed on-chip oscillator clock (CLS = 0 and MCS = 1, or CLS = 1) or CLS = 0, MCS = 0, and MCS1 = 1	HIOSTOP = 1

6.3.4 Oscillation stabilization time counter status register (OSTC)

This is the register that indicates the count status of the X1 clock oscillation stabilization time counter.
The X1 clock oscillation stabilization time can be checked in the following case,

- If the X1 clock starts oscillation while the high-speed on-chip oscillator clock or subsystem clock is being used as the CPU clock.
- If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock with the X1 clock oscillating.

The OSTC register can be read by a 1-bit or 8-bit memory manipulation instruction.
The generation of reset signal, the STOP instruction and MSTOP (bit 7 of clock operation status control register (CSC)) = 1 clear the OSTC register to 00H.

Remark The oscillation stabilization time counter starts counting in the following cases.

- When oscillation of the X1 clock starts (EXCLK, OSCSEL = 0, 1 → 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

Symbol	7	6	5	4	3	2	1	0
OSTC	MOST8	MOST9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18

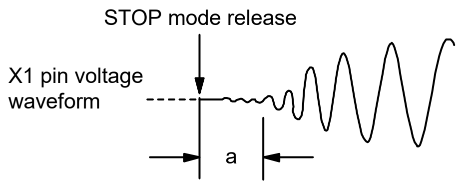
MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	Oscillation stabilization time status		
									fx = 10 MHz	fx = 20 MHz
0	0	0	0	0	0	0	0	2^8/fx max.	25.6 μs max.	12.8 μs max.
1	0	0	0	0	0	0	0	2^8/fx min.	25.6 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	2^9/fx min.	51.2 μs min.	25.6 μs min.
1	1	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 oscillation stabilization time counter counts up to the oscillation stabilization time set by the oscillation stabilization time select register (OSTS).
In the following cases, set the oscillation stabilization time of the OSTS register to the value greater than the count value which is to be checked by the OSTC register.

- **If the X1 clock starts oscillation while the high-speed on-chip oscillator clock or subsystem clock is being used as the CPU clock.**
- **If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock with the X1 clock oscillating.**
  (Note, therefore, that only the status up to the oscillation stabilization time set by the OSTS register is set to the OSTC register after the STOP mode is released.)

Caution 3. The X1 clock oscillation stabilization wait 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)

This register is used to select the X1 clock oscillation stabilization wait time.

When the X1 clock is made to oscillate by clearing the MSTOP bit to start the X1 oscillation circuit operating, actual operation is automatically delayed for the time set in the OSTS register.

When switching the CPU clock from the high-speed on-chip oscillator clock or the subsystem clock to the X1 clock, and when using the high-speed on-chip oscillator clock for switching the X1 clock from the oscillating state to STOP mode, use the oscillation stabilization time counter status register (OSTC) to confirm that the desired oscillation stabilization time has elapsed after release from the STOP mode. That is, use the OSTC register to check that the oscillation stabilization time corresponding to its setting has been reached.

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

Reset signal generation sets the OSTS register to 07H.

Figure 6 - 6 Format of Oscillation stabilization time select register (OSTS)

Address: FFFA3H After reset: 07H R/W

Symbol	7	6	5	4	3	2	1	0
OSTS	0	0	0	0	0	OSTS2	OSTS1	OSTS0

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection		
				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

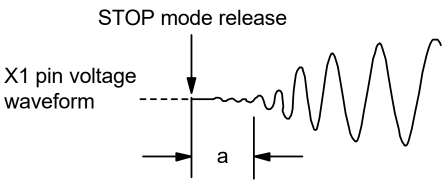
Caution 1. Change the setting of the OSTS register before setting the MSTOP bit of the clock operation status control register (CSC) to 0.

Caution 2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by the OSTS register.

In the following cases, set the oscillation stabilization time of the OSTS register to the value greater than the count value which is to be checked by the OSTC register after the oscillation starts.

- **If the X1 clock starts oscillation while the high-speed on-chip oscillator clock or subsystem clock is being used as the CPU clock.**
- **If the STOP mode is entered and then released while the high-speed on-chip oscillator clock is being used as the CPU clock with the X1 clock oscillating. (Note, therefore, that only the status up to the oscillation stabilization time set by the OSTS register is set to the OSTC register after the STOP mode is released.)**

Caution 3. The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts ("a" below).



Remark fx: X1 clock oscillation frequency

6.3.6 Subsystem clock select register (CKSEL)

The CKSEL register is used to select the sub clock 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.
Reset signal generation sets this register to 00H.

Figure 6 - 7 Format of Subsystem clock select register (CKSEL)

Address: FFFA7H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
CKSEL	0	0	0	0	0	0	0	SELLOSC

SELLOSC	Selection of sub clock/low-speed on-chip oscillator clock
0	Sub clock
1	Low-speed on-chip oscillator clock Note

Note Do not set SELLOSC to 1 when the sub clock (f_{SX}, f_{SXR}) operates.

Caution 1. When changing SELLOSC, be sure to set CSS to 0 (f_{MAIN} selected) and change the value of SELLOSC while CLS is 0.

6.3.7 Peripheral enable registers 0, 1, 2 (PER0, PER1, PER2)

These registers are used to enable or disable supplying the clock to the peripheral hardware. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise.
To use the peripheral functions below, which are controlled by these registers, set (1) the bit corresponding to each function before specifying the initial settings of the peripheral functions.

- Real-time clock 2
- A/D converter
- Serial array unit 0
- Timer array unit 0
- Comparator
- DTC
- 12-bit interval timer
- Frequency measurement circuit
- Data operation circuit (DOC)

The PER0, PER1, and PER2 registers can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears these registers to 00H.

Figure 6 - 8 Format of Peripheral enable register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	4	3	<2>	1	<0>
PER0	RTCWEN	0	ADCEN	0	0	SAU0EN	0	TAU0EN

RTCWEN	Control of access to real-time clock 2
0	• SFR used by the real-time clock 2 cannot be written.
1	• SFR used by the real-time clock 2 can be read and written.

ADCEN	Control of A/D converter input clock supply
0	Stops input clock supply. • SFR used by the A/D converter cannot be written. The read value is 0H. However, the SFR is not initialized. Note 1
1	Enables input clock supply. • SFR used by the A/D converter can be read and written.

SAU0EN	Control of serial array unit 0 input clock supply
0	Stops input clock supply. • SFR used by the serial array unit 0 cannot be written. The read value is 0H. However, the SFR is not initialized. Note 2
1	Enables input clock supply. • SFR used by the serial array unit 0 can be read and written.

TAU0EN	Control of timer array unit 0 input clock supply
0	Stops input clock supply. • SFR used by the timer array unit 0 cannot be written. The read value is 0H. However, the SFR is not initialized. Note 3
1	Enables input clock supply. • SFR used by timer array unit 0 can be read and written.

Note 1. To initialize the A/D converter and the SFR used by the A/D converter, use bit 5 (ADCRES) of PRR0.

Note 2. To initialize serial array unit 0 and the SFR used by serial array unit 0, use bit 2 (SAU0RES) of PRR0.

Note 3. To initialize timer array unit 0 and the SFR used by timer array unit 0, use bit 0 (TAU0RES) of PRR0.

Caution 1. Be sure to clear the following bits to 0.
Bits 1, 3, 4, and 6

Caution 2. Do not change the target bit in the PER0 register while operation of each peripheral function is enabled. Change the setting specified by PER0 while operation of each peripheral function assigned to PER0 is stopped (except for RTCWEN).

Figure 6 - 9 Format of Peripheral enable register 1 (PER1)

Address: F00FAH After reset: 00H R/W

Symbol	7	6	<5>	4	<3>	2	1	0
PER1	0	0	CMPEN	0	DTCEN	0	0	0

CMPEN	Control of comparator input clock supply
0	Stops input clock supply. • SFR used by comparator cannot be written. • SFR used by the comparator cannot be written. The read value is 0H. However, the SFR is not initialized. Note
1	Enables input clock supply. • SFR used by comparator can be read and written.

DTCEN	Control of DTC input clock supply
0	Stops input clock supply. • DTC cannot run.
1	Enables input clock supply. • DTC can run.

Note To initialize the comparator and the SFR used by the comparator, use bit 5 (CMPRES) of PRR1.

Caution 1. Be sure to clear the following bits to 0.
Bits 0 to 2, 4, 6 and 7

Caution 2. Do not change the target bit in the PER1 register while operation of each peripheral function is enabled. Change the setting specified by PER1 while operation of each peripheral function assigned to PER1 is stopped.

Figure 6 - 10 Format of Peripheral enable register 2 (PER2)

Address: F00FCH After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
PER2	TMKAEN	FMCEN	DOCEN	0	0	0	0	0

TMKAEN	Control of 12-bit interval timer input clock supply
0	Stops input clock supply. • SFR used by the 12-bit interval timer cannot be written. The read value is 0H. However, the SFR is not initialized. Note 1
1	Enables input clock supply. • SFR used by the 12-bit interval timer can be read and written.

FMCEN	Control of frequency measurement circuit input clock supply
0	Stops input clock supply. • SFR used by frequency measurement circuit cannot be written.
1	Enables input clock supply. • SFR used by frequency measurement circuit can be read and written.

DOCEN	Control of data operation circuit input clock supply
0	Stops input clock supply. • SFR used by the data operation circuit cannot be written. The read value is 0H. However, the SFR is not initialized. Note 2
1	Enables input clock supply. • SFR used by data operation circuit can be read and written.

Note 1. To initialize the 12-bit interval timer and the SFR used by the 12-bit interval timer, use bit 7 (TMKARES) of PRR2.

Note 2. To initialize the data operation circuit and the SFR used by the data operation circuit, use bit 5 (DOCRES) of PRR2.

Caution 1. Be sure to clear the following bits to 0.
Bits 0 to 4

Caution 2. Do not change the target bit in the PER2 register while operation of each peripheral function is enabled. Change the setting specified by PER2 while operation of each peripheral function assigned to PER2 is stopped (except for FMCEN).

6.3.8 Subsystem clock supply mode control register (OSMC)

This register is used to reduce power consumption by stopping unnecessary clock functions.
If the RTCLPC bit is set to 1, power consumption can be reduced, because clock supply to the peripheral functions, except the real-time clock 2, 12-bit interval timer, 8-bit interval timer, clock output/buzzer output controller is stopped in STOP mode or HALT mode while subsystem clock is selected as CPU clock.
In addition, the OSMC register can be used to select the operation clock of the real-time clock 2, frequency measurement circuit, 12-bit interval timer, 8-bit interval timer, and clock output/buzzer output controller.
The OSMC register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 6 - 11 Format of Subsystem clock supply mode control register (OSMC)

Address: F00F3H After reset: Undefined R/W Note 1

Symbol	<7>	6	5	<4>	3	2	1	0
OSMC	RTCLPC	0	0	WUTMMCK0	x	x	0	0

RTCLPC Note 7	Setting in STOP mode or HALT mode while subsystem clock is selected as CPU clock
0	Enables supply of subsystem clock to peripheral functions (See **Tables 23 - 1** to **23 - 4** for peripheral functions whose operations are enabled.)
1	Stops supply of subsystem clock to peripheral functions other than the real-time clock 2, 12-bit interval timer, 8-bit interval timer, and clock output/buzzer output controller.

WUTMMCK0	Selection of operation clock for real-time clock 2, frequency measurement circuit, 12-bit interval timer, 8-bit interval timer, and clock output/buzzer output controller
0	Subsystem clock Note 2
1	Low-speed on-chip oscillator clock Notes 3, 4, 5, 6

FMTRGSEL Note 10	WUTMMCK0	Operating clock selection for frequency measurement circuit count operation/stop trigger clock and real-time clock 2
0	0	f_{SX} selected for the frequency measurement circuit/real-time clock 2
0	1	f_{IL} selected for the real-time clock 2 (constant-period interrupt function) Note 8
1	0	Setting prohibited
1	1	f_{IL} selected for the frequency measurement circuit Note 9

Note 1. Be sure to set bits 0, 1, 5, and 6 to 0. Bits 2 and 3 are read-only, write is ignored.

Note 2. Do not set the WUTMMCK0 bit to 0 and the FMTRGSEL bit in the FMCKS register to 1.

Note 3. Do not set the WUTMMCK0 bit to 1 while the sub clock is oscillating.

Note 4. Switching between the subsystem clock and the low-speed on-chip oscillator clock can be enabled by the WUTMMCK0 bit only when all of the real-time clock 2, frequency measurement circuit, 12-bit interval timer, 8-bit interval timer, and clock output/buzzer output functions are stopped.

Note 5. f_{IL} can be selected as the operating clock of the real-time clock 2 when the WUTMMCK0 bit is 1 and the FMTRGSEL bit in the FMCKS register is 0. Also, only the constant-period interrupt function of the real-time clock 2 can be used at this time; the clock count function cannot be used.

Note 6. $f_{IL}/2$ can be selected as the operating clock of the frequency measurement circuit when the WUTMMCK0 bit is 1 and the FMTRGSEL bit in the FMCKS register is 1.

Note 7. When the sub clock is selected (SELLOSC = 0) by bit 0 (SELLOSC) of the CKSEL register and RTCLPC is set to 1, the subsystem clock is stopped. However, when the low-speed on-chip oscillator clock is selected (SELLOSC = 1) and RTCLPC is set to 1, the subsystem clock is not stopped.

Note 8. The frequency measurement function cannot be used.

Note 9. The real-time clock 2 cannot be used.

Note 10. Bit 4 of the frequency measurement circuit clock select register (FMCKS)

Remark x: Undefined

6.3.9 High-speed on-chip oscillator frequency select register (HOCODIV)

The frequency of the high-speed on-chip oscillator which is set by an option byte (000C2H) can be changed by using high-speed on-chip oscillator frequency select register (HOCODIV). However, the selectable frequency depends on the FRQSEL3 bit of the option byte (000C2H).

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

Reset signal generation clears this register to the value set by FRQSEL2 to FRQSEL0 of the option byte (000C2H).

Figure 6 - 12 Format of High-speed on-chip oscillator frequency select register (HOCODIV)

Address: F00A8H After reset: the value set by FRQSEL2 to FRQSEL0 of the option byte (000C2H) R/W

Symbol	7	6	5	4	3	2	1	0
HOCODIV	0	0	0	0	0	HOCODIV2	HOCODIV1	HOCODIV0

HOCODIV2	HOCODIV1	HOCODIV0	Selection of high-speed on-chip oscillator clock frequency	
			FRQSEL3 = 0	FRQSEL3 = 1
0	0	0	f_{IH} = 24 MHz	Setting prohibited
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 within the operable voltage range of the flash operation mode set in the option byte (000C2H) before and after the frequency change.

Option Byte (000C2H) Value		Flash Operation Mode	Operating Frequency Range	Operating Voltage Range
CMODE1	CMODE0			
0	0	LV (low-voltage main) mode	1 to 4 MHz	1.6 to 3.6 V
1	0	LS (low-speed main) mode	1 to 8 MHz	1.8 to 3.6 V
1	1	HS (high-speed main) mode	1 to 16 MHz	2.4 to 3.6 V
			1 to 24 MHz	2.7 to 3.6 V
Setting prohibited		Other than above		

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 the frequency is changed with the HOCODIV register, the frequency is switched after the following transition time has elapsed.

- **Operation for up to three clocks at the pre-change frequency**
- **CPU/peripheral hardware clock wait at the post-change frequency for up to three clocks**

6.3.10 High-speed on-chip oscillator trimming register (HIOTRM)

This register is used to adjust the accuracy of the high-speed on-chip oscillator.

With self-measurement of the high-speed on-chip oscillator frequency via a timer using high-accuracy external clock input, and so on, the accuracy can be adjusted.

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 - 13 Format of High-speed on-chip oscillator trimming register (HIOTRM)

Address: F00A0H After reset: Note R/W

Symbol	7	6	5	4	3	2	1	0
HIOTRM	0	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 after reset is the value adjusted at shipment.

Remark 1. The HIOTRM register can be used to adjust the high-speed on-chip oscillator clock to an accuracy within about 0.05%.

Remark 2. For the usage example of the HIOTRM register, see the application note for RL78 MCU Series High-speed On-chip Oscillator (HOCO) Clock Frequency Correction (R01AN0464).

6.3.11 Middle-speed on-chip oscillator frequency select register (MOCODIV)

The MOCODIV register is used to select the division ratio of the middle-speed on-chip oscillator.

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

Reset signal generation clears this register to 00H.

Figure 6 - 14 Format of Middle-speed on-chip oscillator frequency select register (MOCODIV)

Address: F00F2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MOCODIV	0	0	0	0	0	0	MOCODIV1	MOCODIV0

MOCODIV1	MOCODIV0	Selection of middle-speed on-chip oscillator clock frequency
0	0	4 MHz
0	1	2 MHz
1	0	1 MHz
Other than above		Setting prohibited

6.3.12 Frequency measurement clock select register (FMCKS)

The FMCKS register is used to select the operating clock and frequency count clock to be input to the frequency measurement circuit.

The FMCKS register can be used by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the FMCKS register to 00H.

Figure 6 - 15 Format of Frequency measurement clock select register (FMCKS)

Address: F007AH After reset: 00H R/W

Symbol	7	6	5	<4>	3	2	1	0
FMCKS	0	0	0	FMTRGSEL	0	0	FMCKSEL1	FMCKSEL0

FMTRGSEL	WUTMMCK0 Note 3	Selection of frequency measurement circuit count operation/ stop trigger clock/real-time clock 2 operating clock
0	0	f_{SXR} selected for the frequency measurement circuit/real-time clock 2
0	1	f_{IL} selected for the real-time clock 2 (constant-period interrupt function) Note 1
1	0	Setting prohibited
1	1	f_{IL} selected for the frequency measurement circuit Note 2

FMCKSEL1	FMCKSEL0	Selection of frequency count clock
0	0	f_{MX} selected
0	1	f_{IM} selected
1	×	f_{IH} selected

Note 1. The frequency measurement function cannot be used.

Note 2. The real-time clock 2 cannot be used.

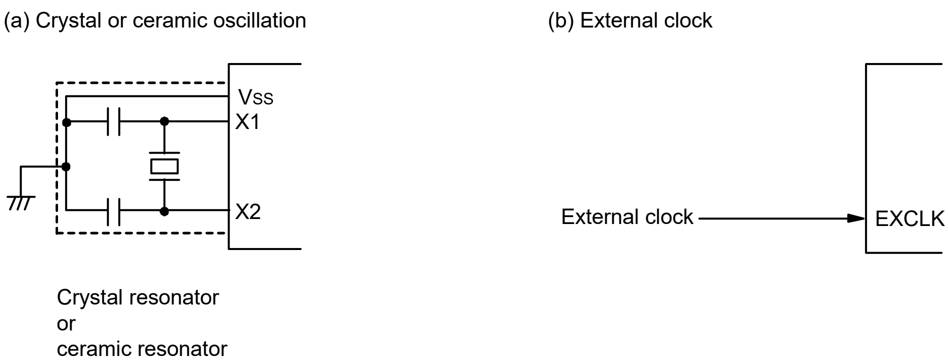
Note 3. Refer to the description on the WUTMMCK0 bit in the OSMC register.

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 bits 7 and 6 (EXCLK, OSCSEL) of the clock operation mode control register (CMC) as follows.
• Crystal or ceramic oscillation: EXCLK, OSCSEL = 0, 1
• External clock input: EXCLK, OSCSEL = 1, 1
When the X1 oscillator is not used, set the input port mode (EXCLK, OSCSEL = 0, 0).
When the pins are not used as input port pins, either, see **Table 2 - 3 Connection of Unused Pins**.
Figure 6 - 16 shows an example of the external circuit of the X1 oscillator.

Figure 6 - 16 Example of External Circuit of X1 Oscillator

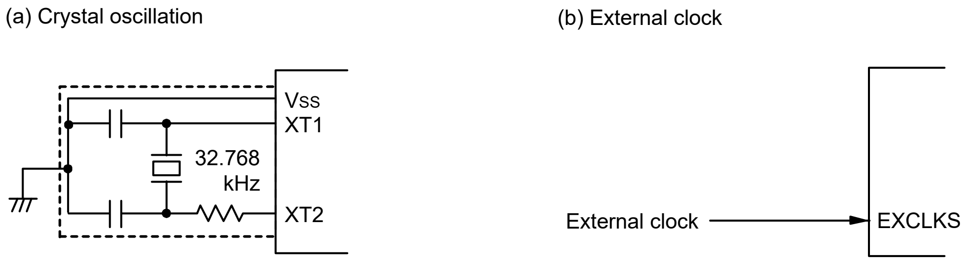


Caution is listed on the next page.

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 bit 4 (OSCSELS) of the clock operation mode control register (CMC) to 1.
An external clock can also be input. In this case, input the clock signal to the EXCLKS pin.
To use the XT1 oscillator, set bits 5 and 4 (EXCLKS, OSCSELS) of the clock operation mode control register (CMC) as follows.
• Crystal oscillation:EXCLKS, OSCSELS = 0, 1
• External clock input:EXCLKS, OSCSELS = 1, 1
When the XT1 oscillator is not used, set the input port mode (EXCLKS, OSCSELS = 0, 0).
When the pins are not used as input port pins, either, see **Table 2 - 3 Connection of Unused Pins**.
Figure 6 - 17 shows an example of the external circuit of the XT1 oscillator.

Figure 6 - 17 Example of External Circuit of XT1 Oscillator



Caution When using the X1 oscillator and XT1 oscillator, wire as follows in the area enclosed by the broken lines in the Figures 6 - 16 and 6 - 17 to avoid an adverse effect from wiring capacitance.

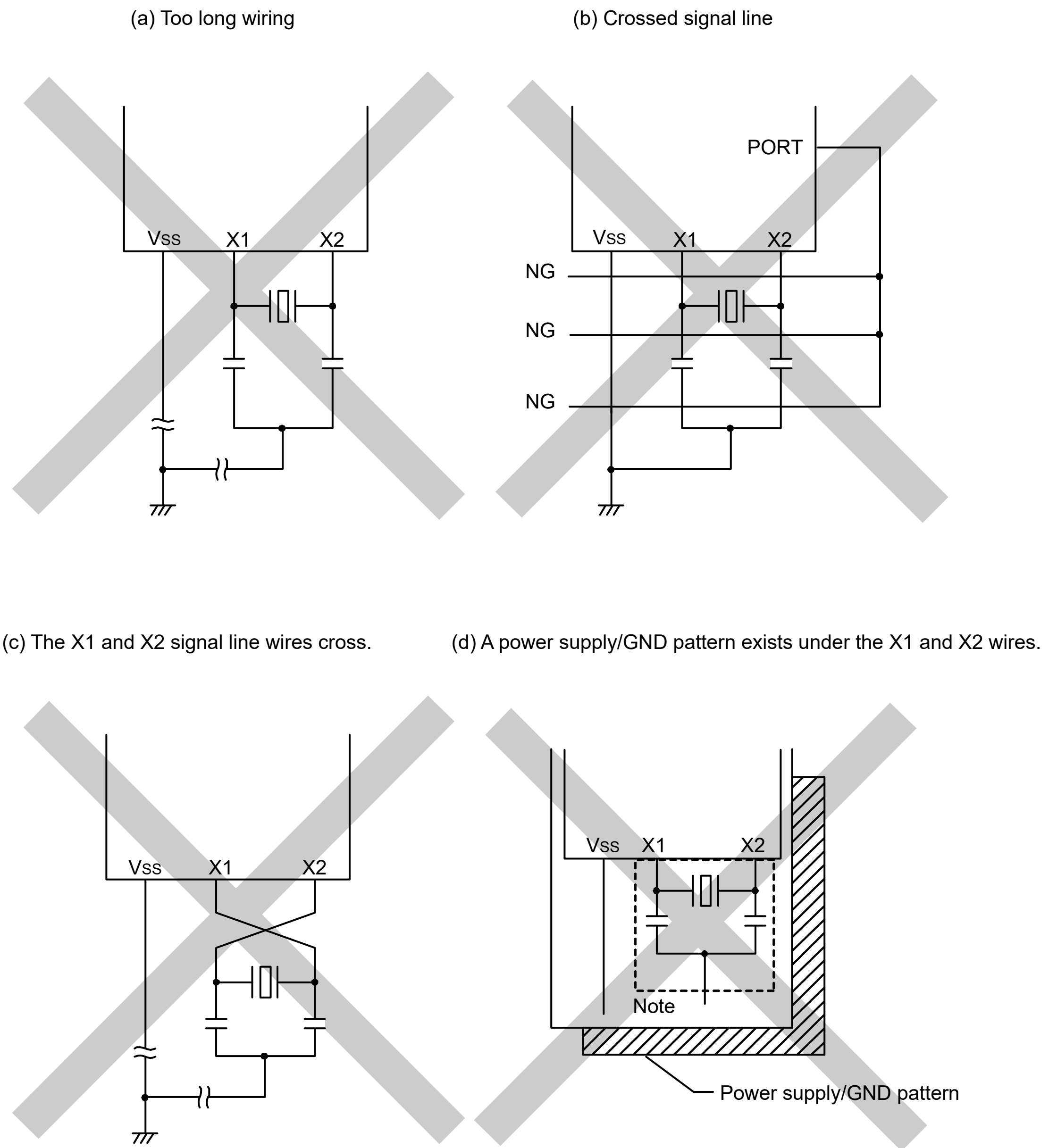
- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **Always make the ground point of the oscillator capacitor the same potential as Vss. Do not ground the capacitor to a ground pattern through which a high current flows.**
- **Do not fetch signals from the oscillator.**

The XT1 oscillator is a circuit with low amplification in order to achieve low-power consumption. Note the following points when designing the circuit.

- **Pins and circuit boards include parasitic capacitance. Therefore, perform oscillation evaluation using a circuit board to be actually used and confirm that there are no problems.**
- **When using the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) as the mode of the XT1 oscillator, evaluate the resonators described in 6.7 Resonator and Oscillator Constants.**
- **Make the wiring between the XT1 and XT2 pins and the resonators as short as possible, and minimize the parasitic capacitance and wiring resistance. Note this particularly when the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) is selected.**
- **Configure the circuit of the circuit board, using material with little wiring resistance.**
- **Place a ground pattern that has the same potential as Vss as much as possible near the XT1 oscillator.**
- **Be sure that the signal lines between the XT1 and XT2 pins, and the resonators do not cross with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **The impedance between the XT1 and XT2 pins may drop and oscillation may be disturbed due to moisture absorption of the circuit board in a high-humidity environment or dew condensation on the board. When using the circuit board in such an environment, take measures to damp-proof the circuit board, such as by coating.**
- **When coating the circuit board, use material that does not cause capacitance or leakage between the XT1 and XT2 pins.**

Figure 6 - 18 shows examples of incorrect resonator connection.

Figure 6 - 18 Examples of Incorrect Resonator Connection (1/2)

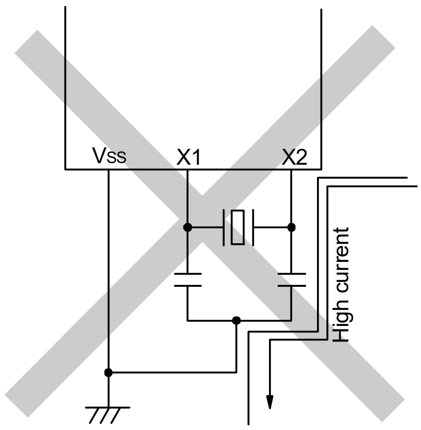


Note Do not place a power supply/GND pattern under the wiring section (section indicated by a broken line in the figure) of the X1 and X2 pins and the resonators in a multi-layer board or double-sided board.
Do not configure a layout that will cause capacitance elements and affect the oscillation characteristics.

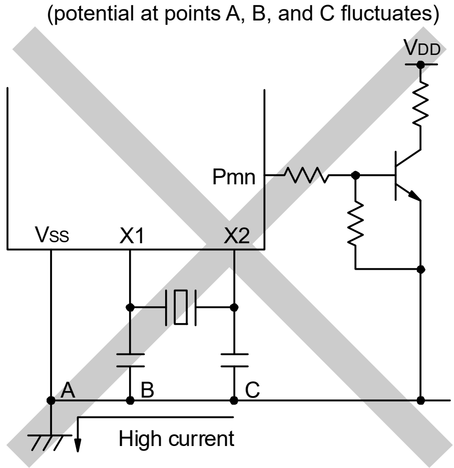
Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

Figure 6 - 19 Examples of Incorrect Resonator Connection (2/2)

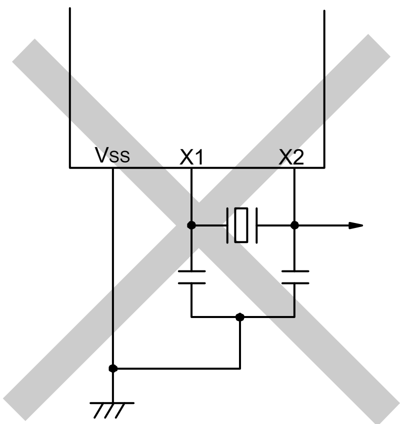
(e) Wiring near high alternating current



(f) Current flowing through ground line of oscillator (potential at points A, B, and C fluctuates)



(g) Signals are fetched



Caution **When X2 and XT1 are wired in parallel, the crosstalk noise of X2 may increase with XT1, resulting in malfunctioning.**

Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

6.4.3 High-speed on-chip oscillator

The high-speed on-chip oscillator is incorporated in the RL78/I1D. The frequency can be selected from among 24, 16, 12, 8, 6, 4, 3, 2, or 1 MHz by using the option byte (000C2H). Oscillation can be controlled by bit 0 (HIOSTOP) of the clock operation status control register (CSC).

The high-speed on-chip oscillator automatically starts oscillating after reset release.

6.4.4 Middle-speed on-chip oscillator

The middle-speed on-chip oscillator is incorporated in the RL78/I1D. Oscillation can be controlled by bit 1 (MIOEN) of the clock operation status control register (CSC).

6.4.5 Low-speed on-chip oscillator

The low-speed on-chip oscillator is incorporated in the RL78/I1D.

The low-speed on-chip oscillator runs while the watchdog timer is operating or when the setting of either or both of the following bits is 1: bit 4 (WUTMMCK0) in the subsystem clock supply mode control register (OSMC) or bit 0 (SELLOSC) in the subsystem clock select register (CKSEL).

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 Clock Generator Operation

The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode (see **Figure 6 - 1**).

- Main system clock f_{MAIN}
  - High-speed system clock f_{MX}
    X1 clock f_X
    External main system clock f_{EX}
  - High-speed on-chip oscillator clock f_{IH}
  - Middle-speed on-chip oscillator clock f_{IM}

- Subsystem clock f_{SUB}
  - XT1 clock f_{XT}
  - External subsystem clock f_{EXS}
  - Low-speed on-chip oscillator clock f_{IL}

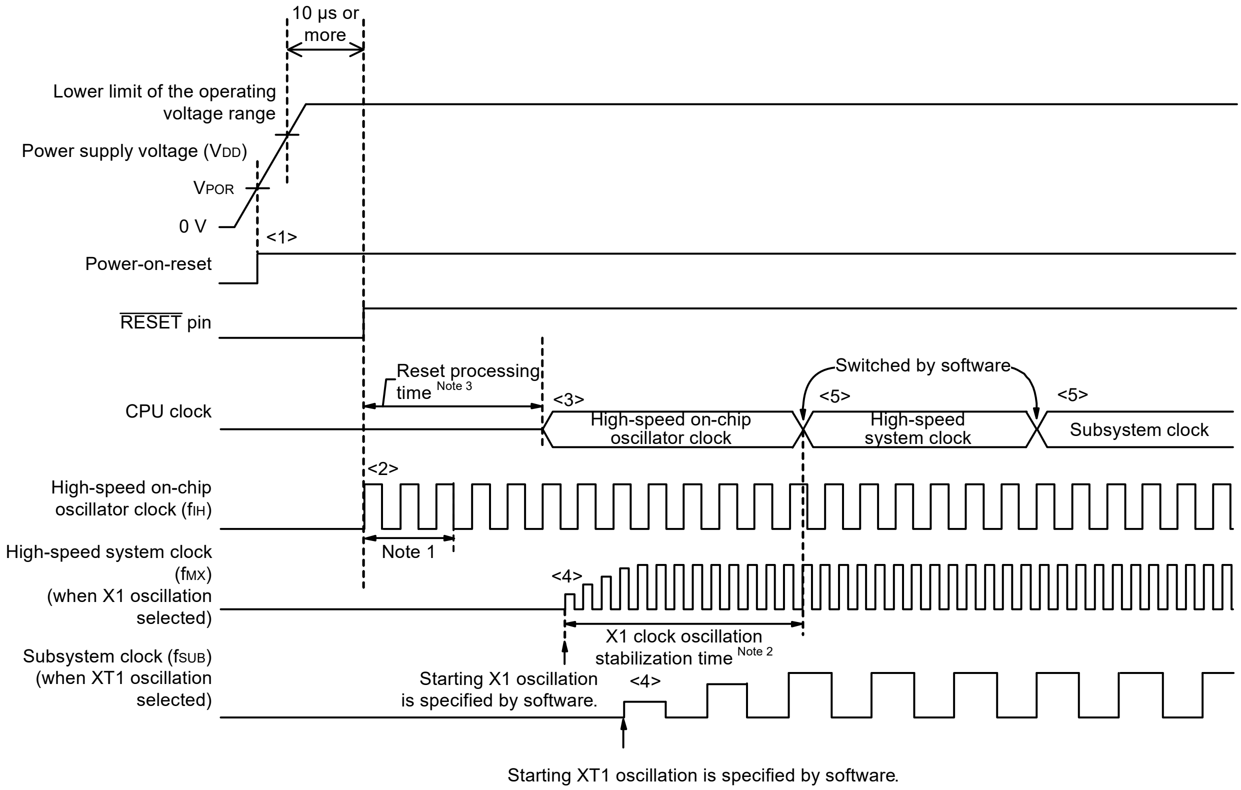
- CPU/peripheral hardware clock f_{CLK}

- Subsystem clock generator clock f_{SX}

- Subsystem clock generator and RTC2/other clock f_{SXR}

The CPU starts operation when the high-speed on-chip oscillator starts outputting after a reset release in the RL78/I1D.
When the power supply voltage is turned on, the clock generator operation is shown in Figure 6 - 20.

Figure 6 - 20 Clock Generator Operation When Power Supply Voltage Is Turned On



<1> When the power is turned on, an internal reset signal is generated by the power-on-reset (POR) circuit.
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 **34.4 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 on the high-speed on-chip oscillator clock after waiting for the voltage to stabilize and a reset processing have been performed after reset release.

<4> Set the start of oscillation of the X1 or XT1 clock via software (see **6.6.2 Example of setting X1 oscillation clock** and **6.6.3 Example of setting XT1 oscillation clock**).

<5> When switching the CPU clock to the X1 or XT1 clock, wait for the clock oscillation to stabilize, and then set switching via software (see **6.6.2 Example of setting X1 oscillation clock** and **6.6.3 Example of setting XT1 oscillation clock**).

Note 1. The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on chip oscillator clock.

Note 2. When releasing a reset, confirm the oscillation stabilization time for the X1 clock using the oscillation stabilization time counter status register (OSTC).

Note 3. For the reset processing time, see **CHAPTER 25 POWER-ON-RESET CIRCUIT**.

Caution It is not necessary to wait for the oscillation stabilization time when an external clock input from the EXCLK pin is used.

6.6 Controlling Clock

6.6.1 Example of setting high-speed on-chip oscillator

After a reset release, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock. The frequency of the high-speed on-chip oscillator can be selected from 24, 16, 12, 8, 6, 4, 3, 2, and 1 MHz by using FRQSEL0 to FRQSEL3 of the option byte (000C2H). In addition, Oscillation can be changed by the high-speed on-chip oscillator frequency select register (HOCODIV).

[Option byte setting]

Address: 000C2H

	7	6	5	4	3	2	1	0
Option byte (000C2H)	CMODE1 0/1	CMODE0 0/1	1	1	FRQSEL3 0/1	FRQSEL2 0/1	FRQSEL1 0/1	FRQSEL0 0/1

CMODE1	CMODE0	Setting of flash operation mode	
0	0	LV (low-voltage main) mode	V_{DD} = 1.6 V to 3.6 V @ 1 MHz to 4 MHz
1	0	LS (low-speed main) mode	V_{DD} = 1.8 V to 3.6 V @ 1 MHz to 8 MHz
1	1	HS (high-speed main) mode	V_{DD} = 2.4 V to 3.6 V @ 1 MHz to 16 MHz V_{DD} = 2.7 V to 3.6 V @ 1 MHz to 24 MHz
Other than above		Setting prohibited	

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator
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 high-speed on-chip oscillator clock frequency	
			FRQSEL3 = 0	FRQSEL3 = 1
0	0	0	f_{IH} = 24 MHz	Setting prohibited
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 X1 oscillation clock

After a reset release, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock. To subsequently change the clock to the X1 oscillation clock, set the oscillator and start oscillation by using the oscillation stabilization time select register (OSTS), clock operation mode control register (CMC), and clock operation status control register (CSC) and wait for oscillation to stabilize by using the oscillation stabilization time counter status register (OSTC). After the oscillation stabilizes, set the X1 oscillation clock to f_{CLK} by using the system clock control register (CKC).

[Register settings] Set the register in the order of <1> to <5> below.

<1> Set (1) the OSCSEL bit of the CMC register, except for the cases where the fx is equal to or more than 10 MHz, in such cases set (1) the AMPH bit, to operate the X1 oscillator.

	7	6	5	4	3	2	1	0
CMC	EXCLK 0	OSCSEL 1	EXCLKS 0	OSCSELS 0	0	AMPHS1 0	AMPHS0 0	AMPH 0/1

<2> Using the OSTS register, select the oscillation stabilization time of the X1 oscillator at releasing of the STOP mode.

Example: Setting values when a wait of at least 102 μs is set based on 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 (0) the MSTOP bit of the CSC register to start oscillating the X1 oscillator.

	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 stabilize.

Example: Wait until the bits reach the following values when a wait of at least 102 μs is set based on 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 oscillation clock as 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 Set the HOCODIV register within the operable voltage range of the flash operation mode set in the option byte (000C2H) before and after the frequency change.

Option Byte (000C2H) Value		Flash Operation Mode	Operating Frequency Range	Operating Voltage Range
CMODE1	CMODE0			
0	0	LV (low-voltage main) mode	1 to 4 MHz	1.6 to 3.6 V
1	0	LS (low-speed main) mode	1 to 8 MHz	1.8 to 3.6 V
1	1	HS (high-speed main) mode	1 to 16 MHz	2.4 to 3.6 V
			1 to 24 MHz	2.7 to 3.6 V
Setting prohibited		Other than above		

6.6.3 Example of setting XT1 oscillation clock

After a reset release, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock. To subsequently change the clock to the XT1 oscillation clock, set the oscillator and start oscillation by using the subsystem clock supply mode control register (OSMC), clock operation mode control register (CMC), and clock operation status control register (CSC), set the XT1 oscillation clock to f_{CLK} by using the system clock control register (CKC).

[Register settings] Set the register in the order of <1> to <5> below.

<1> To run only the real-time clock 2 and 12-bit interval timer on the subsystem clock (ultra-low current consumption) in the STOP mode or in HALT mode during CPU operation on 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	x	x	0	0

<2> Set (1) the OSCSELS bit of the CMC register to operate the XT1 oscillator.

	7	6	5	4	3	2	1	0
CMC	EXCLK 0	OSCSEL 0	EXCLKS 0	OSCSELS 1	0	AMPHS1 0/1	AMPHS0 0/1	AMPH 0

AMPHS0 and AMPHS1 bits: These bits are used to specify the oscillation mode of the XT1 oscillator.

<3> Clear (0) the XTSTOP bit of the CSC register to start oscillating the XT1 oscillator.

	7	6	5	4	3	2	1	0
CSC	MSTOP 1	XTSTOP 0	0	0	0	0	MIOEN 0	HIOSTOP 0

<4> Use the timer function or another function to wait for oscillation of the subsystem clock to stabilize by using software.

<5> Use the CSS bit of the CKC register to specify the XT1 oscillation clock as the CPU/peripheral hardware clock.

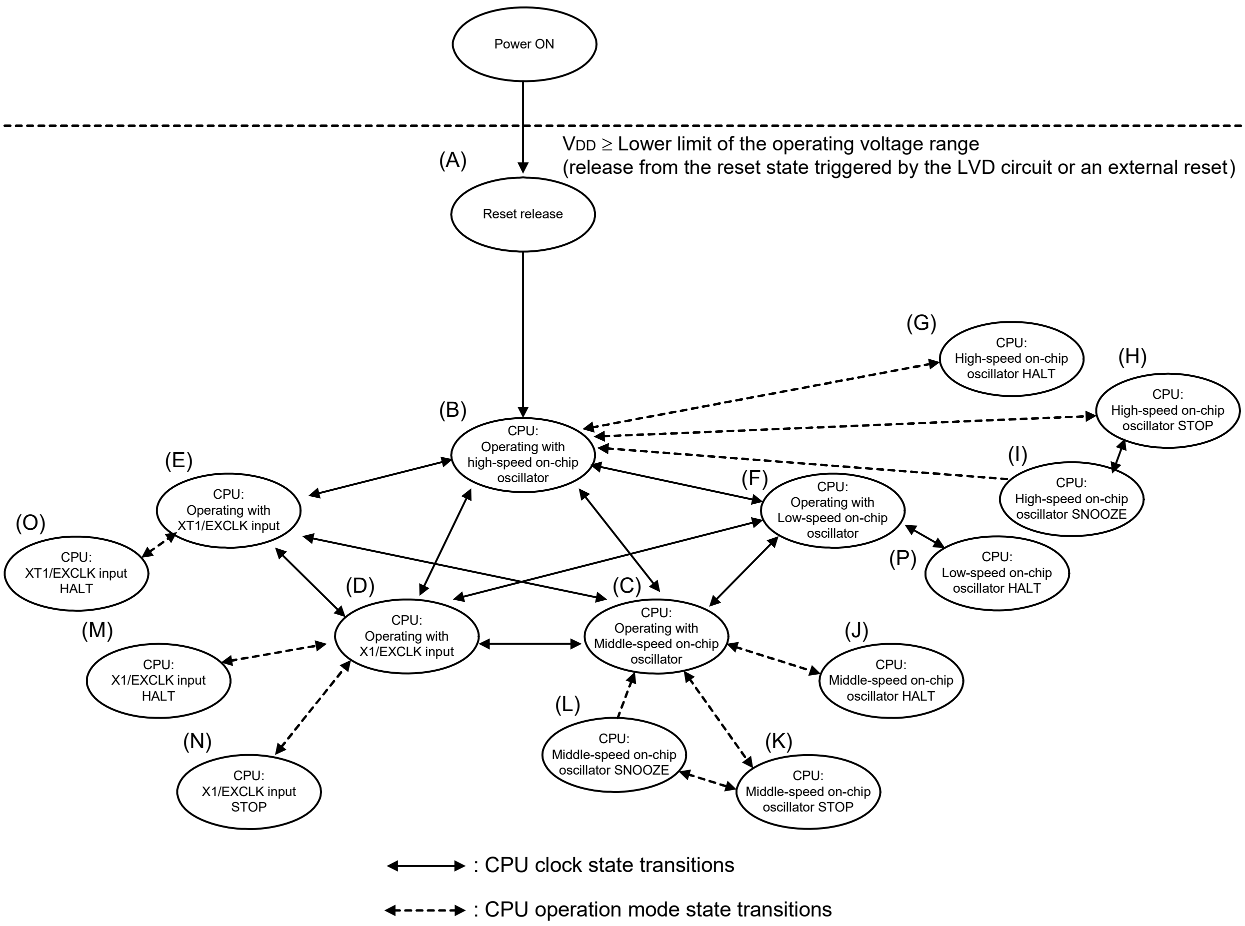
	7	6	5	4	3	2	1	0
CKC	CLS 0	CSS 1	MCS 0	MCM0 0	0	0	MCS1 0	MCM1 0

Remark x: Undefined

6.6.4 CPU clock status transition diagram

Figure 6 - 21 shows the CPU clock status transition diagram of this product.

Figure 6 - 21 CPU Clock Status Transition Diagram



Tables 6 - 3 to **6 - 6** show transition of the CPU clock and examples of setting the SFR registers.

Table 6 - 3 CPU Clock Transition and SFR Register Setting Examples (1/4)

(1) CPU operating with high-speed on-chip oscillator clock (B) after reset release (A)
Target state transition: (A) → (B)

Clock After Change	SFR Register Setting
High-speed on-chip oscillator clock	SFR registers do not have to be set (default status after reset release).

(2) Changing to high-speed on-chip oscillator clock operation (B)
Target state transition: (C) → (B), (D) → (B), (E) → (B), (F) → (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Clock After Change	CSC Register	Waiting for Oscillation Stabilization	CKC Register		
	HIOSTOP		CSS	MCM0	MCM1
High-speed on-chip oscillator clock	0	65 μs	0	0	0

Unnecessary if the CPU is operating with the high-speed on-chip oscillator clock

(3) Changing to middle-speed on-chip oscillator clock operation (C)
Target state transition: (B) → (C), (D) → (C), (E) → (C), (F) → (C)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Clock After Change	CSC Register	Waiting for Oscillation Stabilization	CKC Register		
	MIOEN		CSS	MCM0	MCM1
Middle-speed on-chip oscillator clock	1	4 μs	0	0	1

Unnecessary if the CPU is operating with the middle-speed on-chip oscillator clock

Remark (A) to (P) in Tables 6 - 3 to 6 - 6 correspond to (A) to (P) in Figure 6 - 21.

Table 6 - 4 CPU Clock Transition and SFR Register Setting Examples (2/4)

(4) Changing the CPU to high-speed system clock operation (D)

Target state transition: (B) → (D), (C) → (D), (E) → (D), (F) → (D)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Clock After Change	CMC Register Note 1			OSTS Register	CSC Register	OSTC Register	CKC Register	
	EXCLK	OSCSEL	AMPH		MSTOP		CSS	MCM0
Changing to X1 clock: 1 MHz ≤ fx ≤ 10 MHz	0	1	0	Note 2	0	Must be checked	0	1
Changing to X1 clock: 10 MHz < fx ≤ 20 MHz	0	1	1	Note 2	0	Must be checked	0	1
Changing to external main clock	1	1	×	Note 2	0	Need not be checked	0	1

CMC Register (EXCLK, OSCSEL, AMPH): Unnecessary if these registers are already set

OSTS, CSC, OSTC Registers: Unnecessary if the CPU is operating with the high-speed system clock

Note 1. The clock operation mode control register (CMC) can be changed only once after reset release. This setting is not necessary if it has already been set.

Note 2. Set the oscillation stabilization time as follows.

- Desired the oscillation stabilization time counter status register (OSTC) oscillation stabilization time ≤ Oscillation stabilization time set by the oscillation stabilization time select register (OSTS)

Caution **Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 34 ELECTRICAL SPECIFICATIONS).**

Remark (A) to (P) in Tables 6 - 3 to 6 - 6 correspond to (A) to (P) in Figure 6 - 21.

Table 6 - 5 CPU Clock Transition and SFR Register Setting Examples (3/4)

(5) Changing the CPU to subsystem clock operation (E)
Target state transition: (B) → (E), (C) → (E), (D) → (E)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Clock After Change	CMC Register Note				CSC Register	Waiting for Oscillation Stabilization	CKC Register
	EXCLKS	OSCSELS	AMPHS1	AMPHS0	XTSTOP		CSS
Changing to XT1 clock	0	1	0/1	0/1	0	Necessary	1
Changing to external sub clock	1	1	×	×	×	Necessary	1

$\underbrace{\text{CMC Register}}_{\text{Unnecessary if these registers are already set}}$

$\underbrace{\text{CSC Register, Waiting for Oscillation Stabilization}}_{\text{Unnecessary if the CPU is operating with the subsystem clock}}$

Note The clock operation mode control register (CMC) can be written only once by an 8-bit memory manipulation instruction after reset release.

(6) Changing to low-speed on-chip oscillator clock operation (F)
Target state transition: (B) → (F), (C) → (F), (D) → (F)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Clock After Change	CKSEL	Oscillation accuracy stabilization time	CKC Register
	SELLOSC		CSS
Changing to low-speed on-chip oscillator	1	210 μs	1

$\underbrace{\text{Oscillation accuracy stabilization time}}_{\text{Unnecessary if the CPU is operating with the low-speed on-chip oscillator clock}}$

Remark 1. ×: don't care
Remark 2. (A) to (P) in Tables 6 - 3 to 6 - 6 correspond to (A) to (P) in Figure 6 - 21.

Table 6 - 6 CPU Clock Transition and SFR Register Setting Examples (4/4)

(7) Changing from CPU operation mode (B), (C), (D), (E), and (F) to HALT mode (G), (J), (M), (O), and (P)
Target state transition: (B) → (G), (C) → (J), (D) → (M), (E) → (O), (F) → (P)

Mode After Change	Setting
HALT mode	Executing HALT instruction

(8) Changing from CPU operation mode (B), (C), and (D) to STOP mode (H), (K), and (N)
Target state transition: (B) → (H), (C) → (K), (D) → (N)

(Setting sequence) →

Mode After Change	Setting		
STOP mode	Stopping peripheral functions that cannot operate in STOP mode	Sets the OSTS register	Executing STOP instruction

Settings are unnecessary if the CPU does not enter STOP mode while it is operating with the high-speed system clock

(9) Changing from STOP mode (H) and (K), to SNOOZE mode (I) and (L)

For details about the setting for switching from the STOP mode to the SNOOZE mode, see **14.8 SNOOZE Mode Function**, **17.5.7 SNOOZE mode function**, and **17.7.3 SNOOZE mode function**.

Remark (A) to (P) in Tables 6 - 3 to 6 - 6 correspond to (A) to (P) in Figure 6 - 21.

6.6.5 Condition before changing CPU clock and processing after changing CPU clock

Condition before changing the CPU clock and processing after changing the CPU clock are shown below.

Table 6 - 7 Changing CPU Clock (1/5)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
High-speed on-chip oscillator clock	Middle-speed on-chip oscillator clock	Oscillation of middle-speed on-chip oscillator	Operating current can be reduced by stopping high-speed on-chip oscillator (HIOSTOP = 1) after checking that the CPU clock is changed.
	X1 clock	Stabilization of X1 oscillation • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time	
	External main system clock	Enabling input of external clock from the EXCLK pin • OSCSEL = 1, EXCLK = 1, MSTOP = 0	
	XT1 clock	Stabilization of XT1 oscillation • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • After elapse of oscillation stabilization time	
	External subsystem clock	Enabling input of external clock from the EXCLKS pin • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	
	Low-speed on-chip oscillator clock	Selection of low-speed on-chip oscillator • SELLOSC = 1	
Middle-speed on-chip oscillator clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator • HIOSTOP = 0	Operating current can be reduced by stopping middle-speed on-chip oscillator (MIOEN = 0) after checking that the CPU clock is changed.
	X1 clock	Stabilization of X1 oscillation • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time	
	External main system clock	Enabling input of external clock from the EXCLK pin • OSCSEL = 1, EXCLK = 1, MSTOP = 0	
	XT1 clock	Stabilization of XT1 oscillation • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • After elapse of oscillation stabilization time	
	External subsystem clock	Enabling input of external clock from the EXCLKS pin • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	
	Low-speed on-chip oscillator clock	Selection of low-speed on-chip oscillator • SELLOSC = 1	

Table 6 - 8 Changing CPU Clock (2/5)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
X1 clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • After elapse of oscillation stabilization time	X1 oscillation can be stopped (MSTOP = 1) after checking that the CPU clock is changed.
	Middle-speed on-chip oscillator clock	Oscillation of middle-speed on-chip oscillator • MIOEN = 1	
	External main system clock	Transition not possible	—
	XT1 clock	Stabilization of XT1 oscillation • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • After elapse of oscillation stabilization time	X1 oscillation can be stopped (MSTOP = 1) after checking that the CPU clock is changed.
	External subsystem clock	Enabling input of external clock from the EXCLKS pin • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	
	Low-speed on-chip oscillator clock	Stopped XT1 oscillation Selection of low-speed on-chip oscillator • SELLOSC = 1	
External main system clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • After elapse of oscillation stabilization time	External main system clock input can be disabled (MSTOP = 1) after checking that the CPU clock is changed.
	Middle-speed on-chip oscillator clock	Oscillation of middle-speed on-chip oscillator • MIOEN = 1	
	X1 clock	Transition not possible	—
	XT1 clock	Stabilization of XT1 oscillation • OSCSELS = 1, EXCLKS = 0, XTSTOP = 0 • After elapse of oscillation stabilization time	External main system clock input can be disabled (MSTOP = 1) after checking that the CPU clock is changed.
	External subsystem clock	Enabling input of external clock from the EXCLKS pin • OSCSELS = 1, EXCLKS = 1, XTSTOP = 0	
	Low-speed on-chip oscillator clock	Stopped XT1 oscillation Selection of low-speed on-chip oscillator • SELLOSC = 1	

Table 6 - 9 Changing CPU Clock (3/5)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
XT1 clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator and selection of high-speed on-chip oscillator clock as 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	Oscillation of middle-speed on-chip oscillator and selection of middle-speed on-chip oscillator clock as main system clock • MIOEN = 1, MCS = 0, MCS1 = 1	
	X1 clock	Stabilization of X1 oscillation and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time • MCS = 1	
	External main system clock	Enabling input of external clock from the EXCLK pin and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	
	External subsystem clock	Transition not possible	—
	Low-speed on-chip oscillator clock	Transition not possible	

Table 6 - 10 Changing CPU Clock (4/5)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
External subsystem clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator and selection of high-speed on-chip oscillator clock as 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	Oscillation of middle-speed on-chip oscillator and selection of middle-speed on-chip oscillator clock • MIOEN = 1, MCS = 0, MCS1 = 1	
	X1 clock	Stabilization of X1 oscillation and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time • MCS = 1	
	External main system clock	Enabling input of external clock from the EXCLK pin and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	
	XT1 clock	Transition not possible	—
	External subsystem clock	Transition not possible	

Table 6 - 11 Changing CPU Clock (5/5)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
Low-speed on-chip oscillator clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator and selection of high-speed on-chip oscillator clock as 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	Oscillation of middle-speed on-chip oscillator and selection of middle-speed on-chip oscillator clock • MIOEN = 1, MCS = 0, MCS1 = 1	
	X1 clock	Stabilization of X1 oscillation and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time • MCS = 1	
	External main system clock	Enabling input of external clock from the EXCLK pin and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	
	XT1 clock	Transition not possible	—
	Low-speed on-chip oscillator clock	Transition not possible	

6.6.6 Time required for switchover of CPU clock and main system clock

By setting bits 0, 4, 6 (MCM0, MCM1,CSS) of the system clock control register (CKC), the CPU clock can be switched (between the main system clock and the subsystem clock), main system clock can be switched (between the on-chip oscillator clock and the high-speed system clock), and on-chip oscillator clock can be switched (between the high-speed on-chip oscillator clock and the middle-speed on-chip oscillator clock).

The actual switchover operation is not performed immediately after rewriting to the CKC register; operation continues on the pre-switchover clock for several clocks (see **Tables 6 - 12** to **6 - 15**).

Whether the CPU is operating on the main system clock or the subsystem clock can be ascertained using bit 7 (CLS) of the CKC register. Whether the main system clock is operating on the high-speed system clock or main on-chip oscillator clock can be ascertained using bit 5 (MCS) of the CKC register. Whether the main on-chip oscillator clock is operating on the high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock can be ascertained using bit 1 (MCS1) of the CKC register.

When the CPU clock is switched, the peripheral hardware clock is also switched.

Table 6 - 12 Maximum Time Required for Main System Clock Switchover

Clock A	Switching directions	Clock B	Remark
f_{OCO}	↔	f_{MX}	See **Table 6 - 13**
f_{IH}	↔	f_{IM}	See **Table 6 - 14**
f_{MAIN}	↔	f_{SUB}	See **Table 6 - 15**

Table 6 - 13 Maximum Number of Clocks Required for $f_{OCO} \leftrightarrow f_{IM}$

Set Value Before Switchover		Set Value After Switchover	
MCM1		MCM1	
		0 ($f_{MAIN} = f_{OCO}$)	1 ($f_{MAIN} = f_{MX}$)
0 ($f_{MAIN} = f_{OCO}$)	$f_{MX} \geq f_{OCO}$		2 clocks
	$f_{MX} < f_{OCO}$		2 f_{OCO}/f_{MX} clocks
1 ($f_{MAIN} = f_{MX}$)	$f_{MX} \geq f_{OCO}$	2 f_{MX}/f_{OCO} clocks	
	$f_{MX} < f_{OCO}$	2 clocks	

Table 6 - 14 Maximum Number of Clocks Required for $f_{IH} \leftrightarrow f_{IM}$

Set Value Before Switchover		Set Value After Switchover	
MCM1		MCM1	
		0 ($f_{MAIN} = f_{IH}$)	1 ($f_{MAIN} = f_{IM}$)
0 ($f_{MAIN} = f_{IH}$)	$f_{IM} \geq f_{IH}$		2 clocks
	$f_{IM} < f_{IH}$		1 + f_{IH}/f_{IM} clock
1 ($f_{MAIN} = f_{IH}$)	$f_{IM} \geq f_{IH}$	2 f_{IM}/f_{IH} clocks	
	$f_{IM} < f_{IH}$	2 clocks	

Table 6 - 15 Maximum Number of Clocks Required for $f_{MAIN} \leftrightarrow f_{SUB}$

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} clock
1 ($f_{CLK} = f_{SUB}$)	3 clock	

Remark 1. The number of clocks listed in Tables 6 - 13 to 6 - 15 is the number of CPU clocks before switchover.

Remark 2. Calculate the number of clocks in Tables 6 - 13 to 6 - 15 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) to the high-speed system clock (@ oscillation with f_{IH} = 8 MHz, f_{MX} =10 MHz)

$1 + f_{IH}/f_{MX} = 1 + 8/10 = 1 + 0.8 = 1.8 \rightarrow 2$ clocks

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 - 16 Conditions Before the Clock Oscillation Is Stopped and Flag Settings

Clock	Conditions Before Clock Oscillation Is Stopped (External Clock Input Disabled)	Flag Settings of SFR Register
High-speed on-chip oscillator clock	MCS1 = 1, MCS = 1 or CLS = 1 (The CPU is operating on 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 on a clock other than the middle-speed on-chip oscillator clock.)	MIOEN = 0
X1 clock	MCS = 0 or CLS = 1 (The CPU is operating on a clock other than the high-speed system clock.)	MSTOP = 1
External main system clock		
XT1 clock	CLS = 0 (The CPU is operating on a clock other than the subsystem clock.)	XTSTOP = 1
External subsystem clock		
Low-speed on-chip oscillator clock	CLS = 0 (The CPU is operating on a clock other than the Low-speed on-chip oscillator clock.)	SELLOSC = 0

6.7 Resonator and Oscillator Constants

For the resonators for which operation has been verified and their oscillator constants, 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 the RL78 microcontroller so that the internal operation conditions are within the specifications of the DC and AC characteristics.

Figure 6 - 22 Example of External Circuit

CHAPTER 7 TIMER ARRAY UNIT

The number of timer array unit channels is shown below.

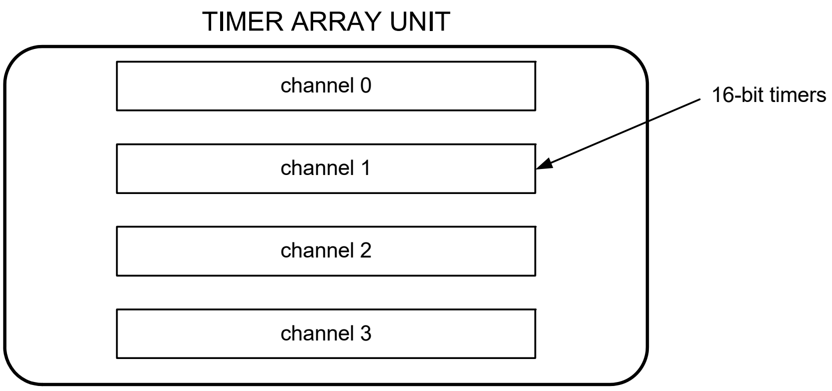
Channel	20-pin	30-pin	24, 32, 48-pin
Channel 0	√	√	√
Channel 1	√	√	√
Channel 2	√	√	√
Channel 3	√	√	√

Caution 1. The presence or absence of timer I/O pins differs depending on the product. For details, see Table 7 - 2 Timer I/O Pins provided in Each Product.

Caution 2. Most of the following descriptions in this chapter use the 48-pin products as an example.

The timer array unit has four 16-bit timers.

Each 16-bit timer is called a channel and can be used as an independent timer. In addition, two or more “channels” can be used to create a high-accuracy timer.



For details about each function, see the table below.

Independent channel operation function	Simultaneous channel operation function
• Interval timer (→ refer to **7.8.1**) • Square wave output (→ refer to **7.8.1**) • External event counter (→ refer to **7.8.2**) • Divider Note (→ refer to **7.8.3**) • Input pulse interval measurement (→ refer to **7.8.4**) • Measurement of high-/low-level width of input signal (→ refer to **7.8.5**) • Delay counter (→ refer to **7.8.6**)	• One-shot pulse output (→ refer to **7.9.1**) • PWM output (→ refer to **7.9.2**) • Multiple PWM output (→ refer to **7.9.3**)

Note Only channel 0.

It is possible to use the 16-bit timer of channels 1 and 3 as two 8-bit timers (higher and lower). The functions that can use channels 1 and 3 as 8-bit timers are as follows:

- Interval timer (upper or lower 8-bit timer)/square wave output (lower 8-bit timer only)
- External event counter (lower 8-bit timer only)
- Delay counter (lower 8-bit timer only)

7.1 Functions of Timer Array Unit

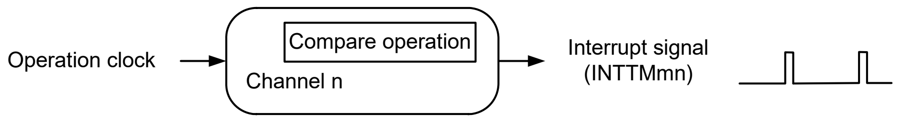
Timer array unit has the following functions.

7.1.1 Independent channel operation function

By operating a channel independently, it can be used for the following purposes without being affected by the operation mode of other channels.

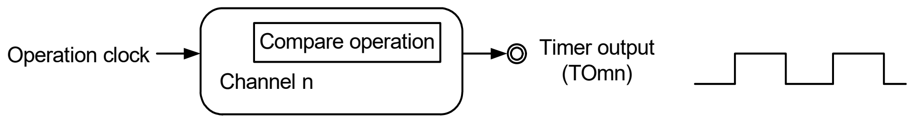
(1) Interval timer

Each timer of a unit can be used as a reference timer that generates an interrupt (INTTMmn) at fixed intervals.



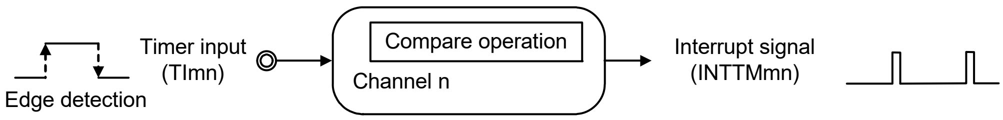
(2) Square wave output

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).



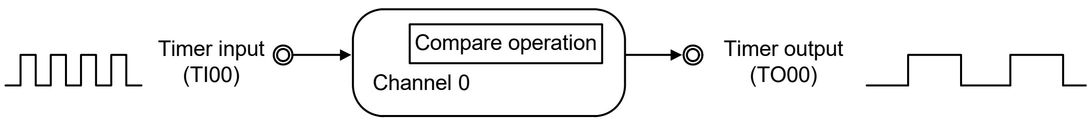
(3) External event counter

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.



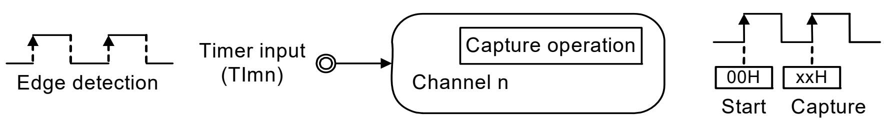
(4) Divider function (channel 0 only)

A clock input from a timer input pin (TI00) is divided and output from an output pin (TOm0).



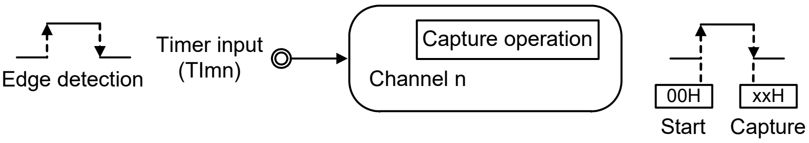
(5) Input pulse interval measurement

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.



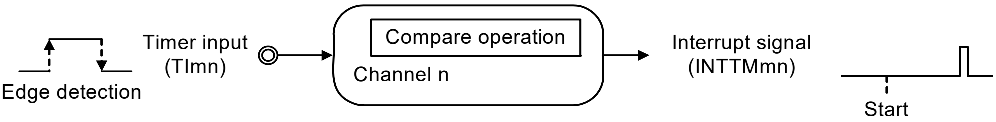
(6) Measurement of high-/low-level width of input signal

Counting is started by a single edge of the signal input to the timer input pin (TImn), and the count value is captured at the other edge. In this way, the high-level or low-level width of the input signal can be measured.



(7) Delay counter

Counting is started at the valid edge of the signal input to the timer input pin (TImn), and an interrupt is generated after any delay period.



Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

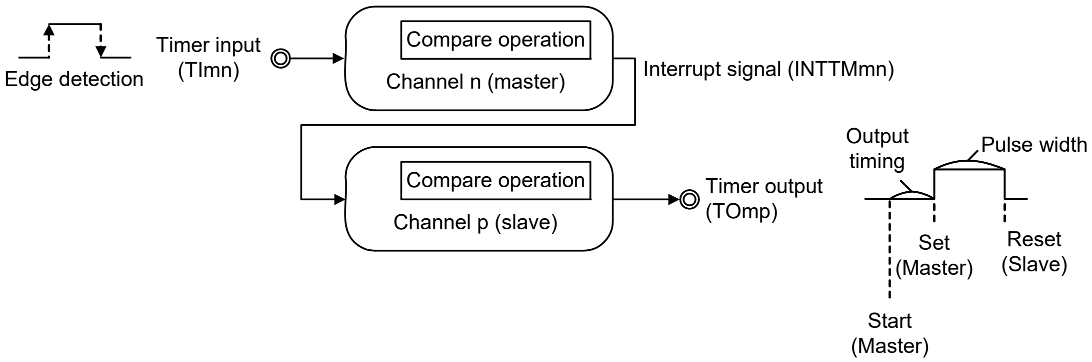
Remark 2. The presence or absence of timer I/O pins of channel 0 to 3 depends on the product. See **Table 7 - 2 Timer I/O Pins provided in Each Product** for details.

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 purposes.

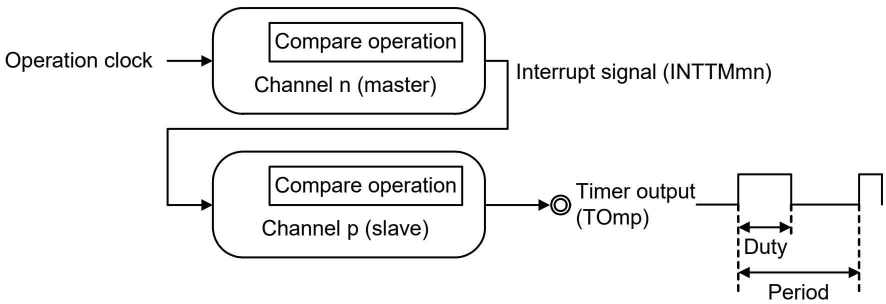
(1) One-shot pulse output

Two channels are used as a set to generate a one-shot pulse with a specified output timing and a specified pulse width.



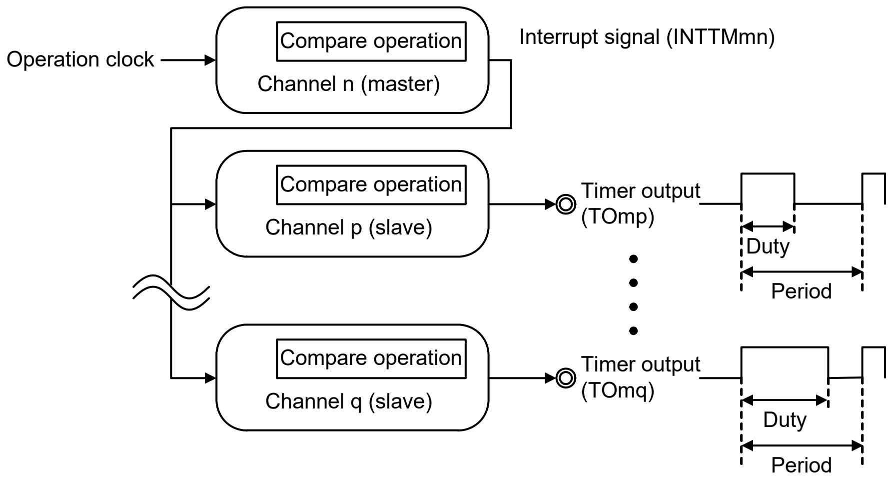
(2) PWM (Pulse Width Modulation) output

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 three 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 rules of simultaneous channel operation function.**

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3),
p, q: Slave channel number (n < p < q ≤ 3)

7.1.3 8-bit timer operation function (channels 1 and 3 only)

The 8-bit timer operation function makes it possible to use a 16-bit timer channel in a configuration consisting of two 8-bit timer channels. This function can only be used for channels 1 and 3.

Caution **There are several rules for using 8-bit timer operation function.**
For details, see 7.4.2 Basic rules of 8-bit timer operation function (channels 1 and 3 only).

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 count register mn (TCRmn)
Register	Timer data register mn (TDRmn)
Timer input	TI00 to TI03 Note 1
Timer output	TO00 to TO03 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 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) • Noise filter enable register 1 (NFEN1) • Port mode control register (PMCxx) Note 2 • Port mode register (PMxx) Note 2 • Port register (Pxx) Note 2

Note 1. The presence or absence of timer I/O pins of channel 0 to 3 depends on the product. See **Table 7 - 2 Timer I/O Pins provided in Each Product** for details.

Note 2. The port mode control register (PMCxx), port mode registers (PMxx) and port registers (Pxx) to be set differ depending on the product. For details, see **4.5 Register Settings When Using Alternate Function**.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

The presence or absence of timer I/O pins in each timer array unit channel depends on the product.

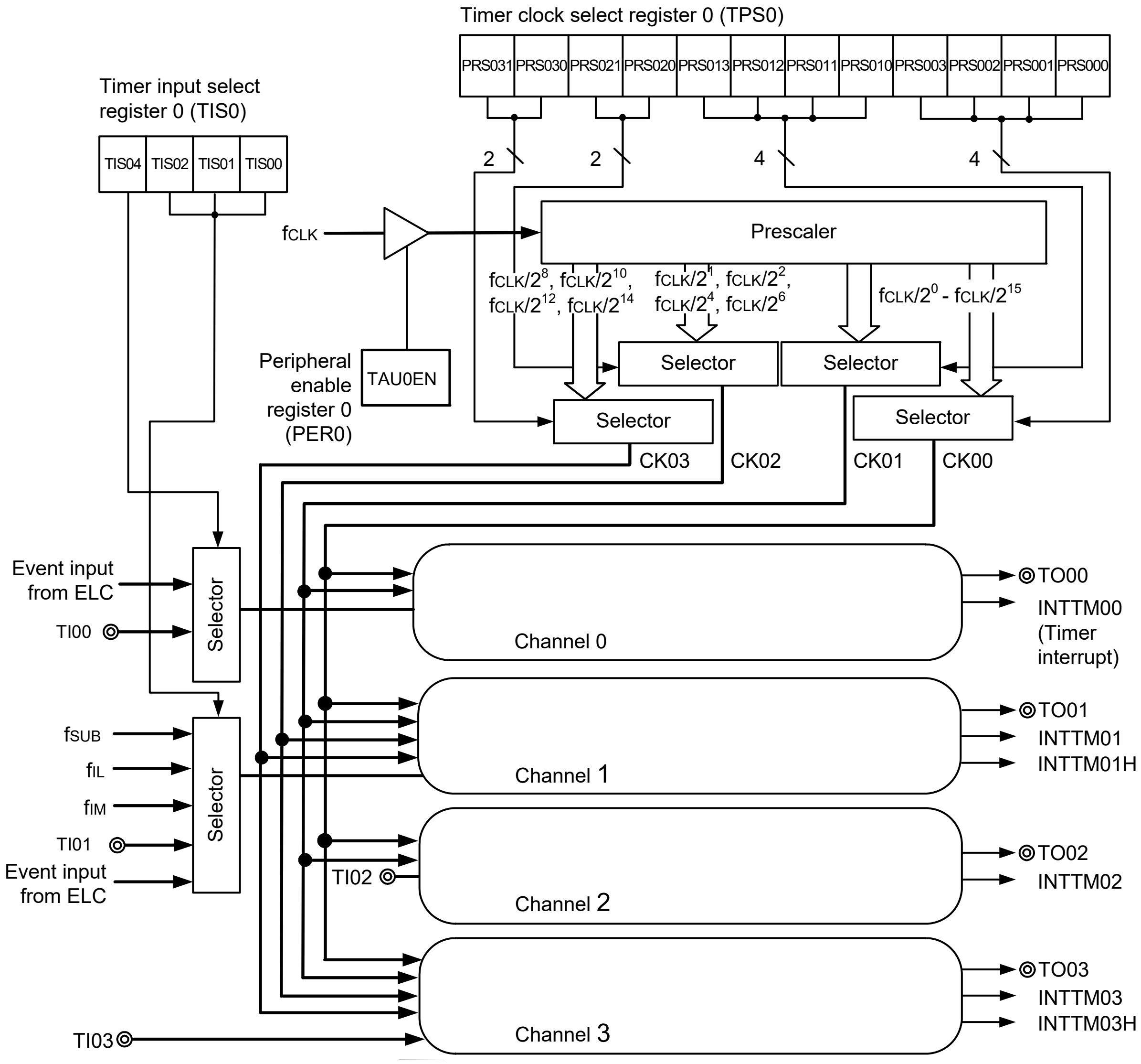
Table 7 - 2 Timer I/O Pins provided in Each Product

Timer array unit channels		I/O Pins of Each Product			
		20-pin	24-pin	30-pin	32, 48-pin
Unit 0	Channel 0	TI00, TO00			
	Channel 1	TI01, TO01			
	Channel 2	—	TI02/TO02		
	Channel 3	—	TI02/TO02	—	TI03/TO03

Remark When timer input and timer output are shared by the same pin, either only timer input or only timer output can be used.

Tables 7 - 1 and 7 - 2 show the block diagrams of the timer array unit.

Figure 7 - 1 Entire Configuration of Timer Array Unit 0



Remark

fSUB: Subsystem clock frequency

fIL: Low-speed on-chip oscillator clock frequency

fIM: Middle-speed on-chip oscillator clock frequency

Figure 7 - 2 Internal Block Diagram of Channel 0 of Timer Array Unit 0

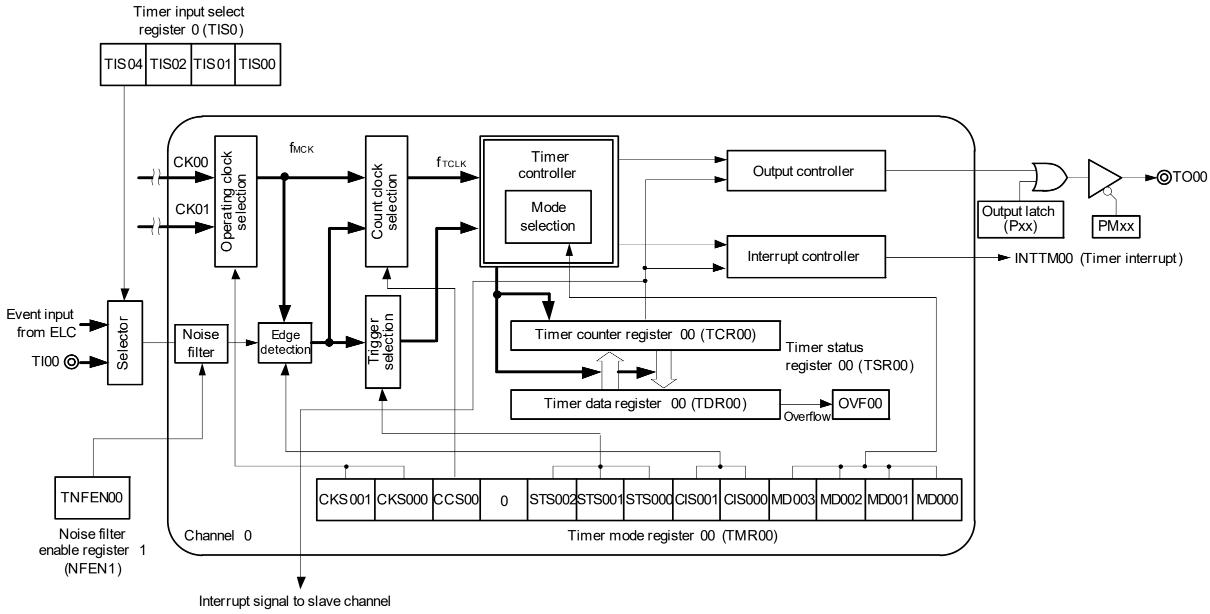


Figure 7 - 3 Internal Block Diagram of Channel 1 of Timer Array Unit 0

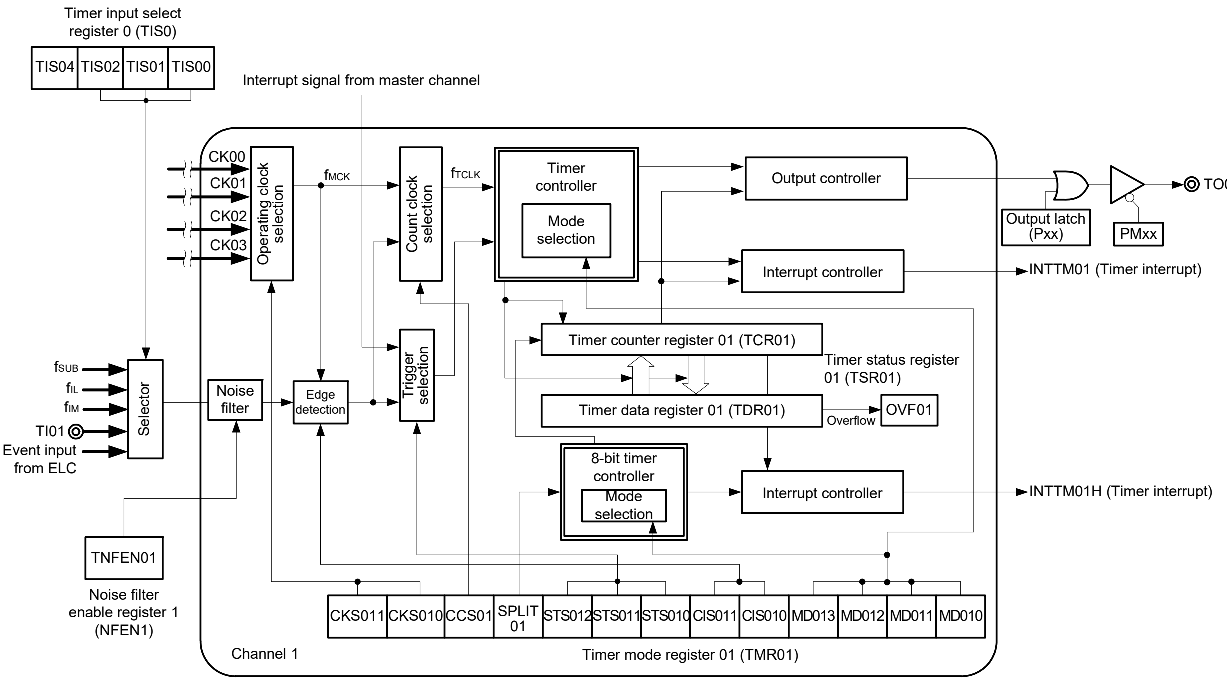


Figure 7 - 4 Internal Block Diagram of Channel 2 of Timer Array Unit 0

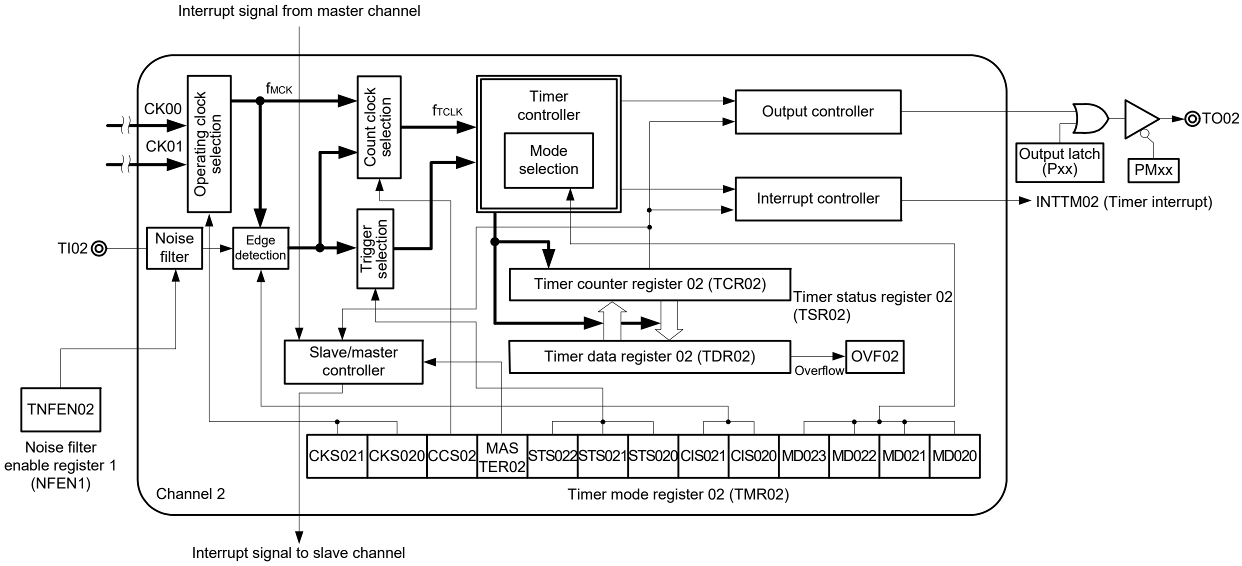
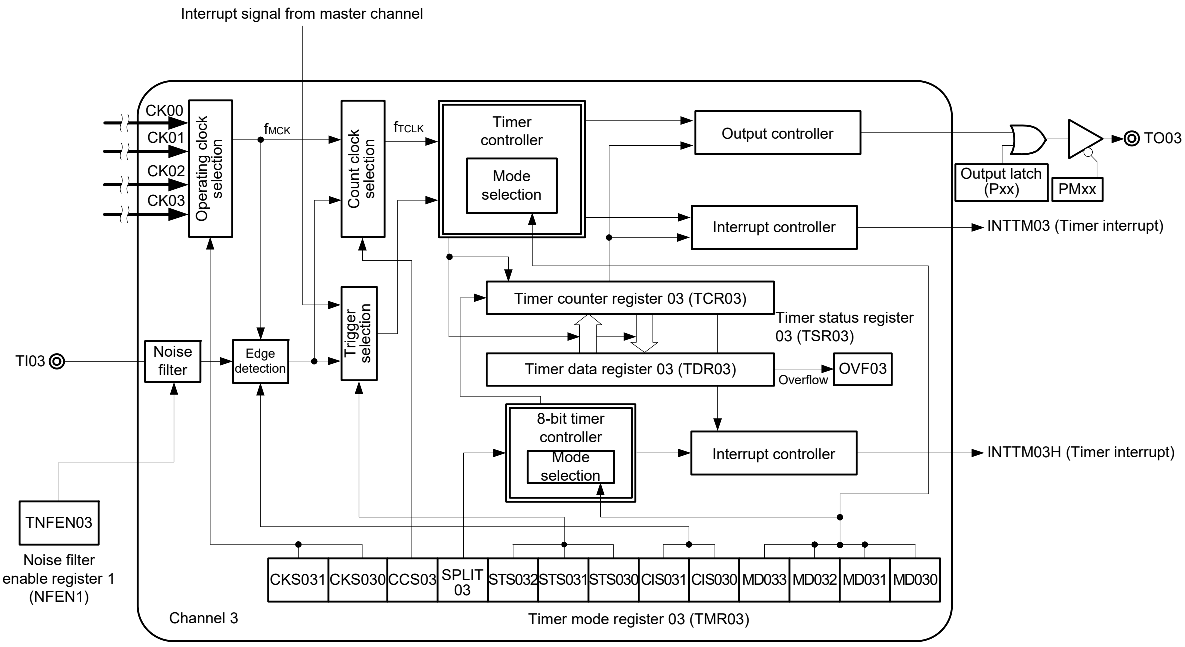


Figure 7 - 5 Internal Block Diagram of Channel 3 of Timer Array Unit 0



7.2.1 Timer count register mn (TCRmn)

The TCRmn register is a 16-bit read-only register and is used to count clocks.

The value of this counter is incremented or decremented in synchronization with the rising edge of a count clock. Whether the counter is incremented or decremented depends on the operation mode that is selected by the MDmn3 to MDmn0 bits of timer mode register mn (TMRmn) (refer to **7.3.4 Timer mode register mn (TMRmn)**).

Figure 7 - 6 Format of Timer count register mn (TCRmn)

Address: F0180H, F0181H (TCR00) to F0186H, F0187H (TCR03) After reset: FFFFH R

	F0181H (TCR00)								F0180H (TCR00)							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCRmn																

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

The count value can be read by reading timer count register mn (TCRmn).
The count value is set to FFFFH in the following cases.

- When the reset signal is generated
- When the TAU0RES 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 count mode
- When counting of the master/slave channel has been completed in the one-shot pulse output mode
- When counting of the slave channel has been completed in the multiple PWM output mode

The count value is cleared to 0000H in the following cases.

- When the start trigger is input in the capture mode
- When capturing has been completed in the capture mode

Caution The count value is not captured to timer data register mn (TDRmn) even when the TCRmn register is read.

The TCRmn register read value differs as follows according to operation mode changes and the operating status.

Table 7 - 3 Timer Count Register mn (TCRmn) Read Value in Various Operation Modes

Operation Mode	Count Mode	Timer count register mn (TCRmn) Read Value Note			
		Value if the operation mode was changed after releasing reset	Value if the Operation was restarted after count operation paused (TTmn = 1)	Value if the operation mode was changed after count operation paused (TTmn = 1)	Value when waiting for a start trigger after one count
Interval timer mode	Count down	FFFFH	Value if stop	Undefined	—
Capture mode	Count up	0000H	Value if stop	Undefined	—
Event counter mode	Count down	FFFFH	Value if stop	Undefined	—
One-count mode	Count down	FFFFH	Value if stop	Undefined	FFFFH
Capture & one-count mode	Count up	0000H	Value if stop	Undefined	Capture value of TDRmn register + 1

Note This indicates the value read from the TCRmn register when channel n has stopped operating as a timer (TEmn = 0) and has been enabled to operate as a counter (TSmn = 1). The read value is held in the TCRmn register until the count operation starts.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

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 SPLIT 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.

Reset signal generation clears this register to 0000H.

Figure 7 - 7 Format of Timer data register mn (TDRmn) (n = 0, 2)

Address: FFF18H, FFF19H (TDR00), FFF64H, FFF65H (TDR02) After reset: 0000H R/W

	FFF19H (TDR00)								FFF18H (TDR00)							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDRmn																

Figure 7 - 8 Format of Timer data register mn (TDRmn) (n = 1, 3)

Address: FFF1AH, FFF1BH (TDR01), FFF66H, FFF67H (TDR03) After reset: 00H R/W

	FFF1BH (TDR01H)								FFF1AH (TDR01L)							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDRmn																

(i) When timer data register mn (TDRmn) is used as compare register

Counting down is started from the value set to the TDRmn register. When the count value reaches 0000H, an interrupt signal (INTTMmn) is generated. The TDRmn register holds its value until it is rewritten.

Caution The TDRmn register does not perform a capture operation even if a capture trigger is input, when it is set to the compare function.

(ii) When timer data register mn (TDRmn) is used as capture register

The count value of timer count register mn (TCRmn) is captured to the TDRmn register when the capture trigger is input.

A valid edge of the TImn pin can be selected as the capture trigger. This selection is made by timer mode register mn (TMRmn).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

7.3 Registers Controlling Timer Array Unit

Timer array unit is controlled by the following registers.

- 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 output enable register m (TOEm)
- Timer output register m (TOm)
- Timer output level register m (TOLm)
- Timer output mode register m (TOMm)
- Noise filter enable register 1 (NFEN1)
- Port mode control register (PMCxx)
- Port mode register (PMxx)
- Port register (Pxx)

Caution Which registers and bits are included depends on the product. Be sure to set bits that are not mounted to their initial values.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

7.3.1 Peripheral enable register 0 (PER0)

This registers is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the timer array unit 0 is used, be sure to set bit 0 (TAU0EN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7 - 9 Format of Peripheral enable register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	4	3	<2>	1	<0>
PER0	RTCWEN	0	ADCEN	0	0	SAU0EN	0	TAU0EN

TAU0EN	Control of timer array 0 unit input clock
0	Stops supply of input clock. • SFR used by the timer array unit 0 cannot be written.
1	Supplies input clock. • SFR used by the timer array unit 0 can be read/written.

Caution 1. When setting the timer array unit, be sure to set the following registers first while the TAUmEN bit is set to 1. If TAUmEN = 0, writing to the control registers of the timer array unit is ignored (except for timer input select register 0 (TIS0), noise filter enable register 1 (NFEN1), port mode control register 3 (PMC3), port mode registers 3, 5 (PM3, PM5), and port registers 3, 5 (P3, P5)).

- **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)**

Caution 2. Be sure to clear the following bits to 0.
Bits 1, 3, 4, and 6

7.3.2 Peripheral reset control register 0 (PRR0)

This register is used for individual reset control of each peripheral hardware.
This MCU controls reset and reset release of each peripheral hardware supported by the PRR0 register.
To reset timer array unit 0, be sure to set bit 0 (TAU0RES) to 1.
The PRR0 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 7 - 10 Format of Peripheral reset control register 0 (PRR0)

Address: F00F1H After reset: 00H R/W

Symbol	7	6	<5>	4	3	<2>	1	<0>
PRR0	0	0	ADCRES	0	0	SAU0RES	0	TAU0RES

TAU0RES	Reset control of timer array unit 0
0	Timer array unit reset release
1	Timer array unit reset state

7.3.3 Timer clock select register m (TPSm)

The TPSm register is a 16-bit register that is used to select two types or four types of operation 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 3):
All channels for which CKm0 is selected as the operation clock (CKSmn1, CKSmn0 = 0, 0) are stopped (TEmn = 0).
If the PRSm10 to PRSm13 bits can be rewritten (n = 0 to 3):
All channels for which CKm2 is selected as the operation clock (CKSmn1, CKSmn0 = 0, 1) are stopped (TEmn = 0).
If the PRSm20 and PRSm21 bits can be rewritten (n = 1, 3):
All channels for which CKm1 is selected as the operation clock (CKSmn1, CKSmn0 = 1, 0) 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 operation clock (CKSmn1, CKSmn0 = 1, 1) are stopped (TEmn = 0).

The TPSm register can be set by a 16-bit memory manipulation instruction.
Reset signal generation clears this register to 0000H.

Figure 7 - 11 Format of Timer clock select register m (TPSm) (1/2)

Address: F01B6H, F01B7H (TPS0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPSm	0	0	PRSm 31	PRSm 30	0	0	PRSm 21	PRSm 20	PRSm 13	PRSm 12	PRSm 11	PRSm 10	PRSm 03	PRSm 02	PRSm 01	PRSm 00

PRS mk3	PRS mk2	PRS mk1	PRS mk0		Selection of operation clock (CKmk) Note (k = 0, 1)				
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz	24 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	12 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	6 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	3 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz	1.5 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156 kHz	313 kHz	625 kHz	750 kHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz	375 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	187.5 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	93.8 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	46.9 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	23.4 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	11.7 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	5.86 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz	2.93 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz	1.46 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61.0 Hz	153 Hz	305 Hz	610 Hz	732 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop timer array unit (TTm = 000FH).
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 clear bits 15, 14, 11, 10 to "0".

Caution 2. If f_{CLK} (undivided) is selected as the operation clock (CKmk) and TDRnm is set to 0000H (n = 0, m = 0 to 3), 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 = 1 to 15). For details, see **7.5.1 Count clock (f_{TCLK})**.

Figure 7 - 12 Format of Timer clock select register m (TPSm) (2/2)

Address: F01B6H, F01B7H (TPS0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPSm	0	0	PRSm 31	PRSm 30	0	0	PRSm 21	PRSm 20	PRSm 13	PRSm 12	PRSm 11	PRSm 10	PRSm 03	PRSm 02	PRSm 01	PRSm 00

PRS m21	PRS m20		Selection of operation clock (CKm2) Note				
			f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz
0	0	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	12 MHz
0	1	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	6 MHz
1	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz	1.5 MHz
1	1	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156.2 kHz	313 kHz	375 kHz

PRS m31	PRS m30		Selection of operation clock (CKm3) Note				
			f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz
0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	93.8 kHz
0	1	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	23.4 kHz
1	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	5.86 kHz
1	1	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz	1.46 kHz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop timer array unit (TTm = 000FH).
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 clear bits 15, 14, 11, 10 to "0".

By using channels 1 and 3 in the 8-bit timer mode and specifying CKm2 or CKm3 as the operation clock, the interval times shown in Table 7 - 4 can be achieved by using the interval timer function.

Table 7 - 4 Interval Times Available for Operation Clock CKSm2 or CKSm3

Clock		Interval time Note (f_{CLK} = 20 MHz)			
		16 μs	160 μs	1.6 ms	16 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 $f_{CLK}/2^r$ selected with the TPSm register, see **7.5.1 Count 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 operation clock (f_{MCK}), select the count clock, select the master/slave, select the 16 or 8-bit timer (only for channels 1 and 3), specify the start trigger and capture trigger, select the valid edge of the timer input, and specify the operation mode (interval, capture, event counter, one-count, or capture and one-count).

Rewriting the TMRmn register is prohibited when the register is in operation (when TEmn = 1). However, bits 7 and 6 (CISmn1, CISmn0) can be rewritten even while the register is operating with some functions (when TEmn = 1) (for details, see **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.

Reset signal generation clears this register to 0000H.

Caution The bits mounted depend on the channels in the bit 11 of TMRmn register.
TMRm2: MASTERmn bit (n = 2)
TMRm1, TMRm3: SPLITmn bit (n = 1, 3)
TMRm0: Fixed to 0

Figure 7 - 13 Format of Timer mode register mn (TMRmn) (1/4)

Address: F0190H, F0191H (TMR00) to F0196H, F0197H (TMR03) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2)	CKSmn1	CKSmn0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKSmn1	CKSmn0	0	CCSmn	SPLITmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0)	CKSmn1	CKSmn0	0	CCSmn	0 Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

CKSmn1	CKSmn0	Selection of operation clock (f_{MCK}) of channel n
0	0	Operation clock CKm0 set by timer clock select register m (TPSm)
0	1	Operation clock CKm2 set by timer clock select register m (TPSm)
1	0	Operation clock CKm1 set by timer clock select register m (TPSm)
1	1	Operation clock CKm3 set by timer clock select register m (TPSm)
Operation clock (f_{MCK}) is used by the edge detector. A count clock (f_{TCLK}) and a sampling clock are generated depending on the setting of the CCSmn bit. The operation clocks CKm2 and CKm3 can only be selected for channels 1 and 3.		

CCSmn	Selection of count clock (f_{TCLK}) of channel n
0	Operation clock (f_{MCK}) specified by the CKSmn0 and CKSmn1 bits
1	Valid edge of input signal input from the TImn pin In channels 0 and 1, valid edge of input signal selected by TIS0
Count clock (f_{TCLK}) is used for the counter, output controller, and interrupt controller.	

Note Bit 11 is fixed at 0 of read only, write is ignored.

Caution 1. Be sure to clear bits 13, 5, and 4 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 CKSmn0 and CKSmn1 bits (f_{MCK}) or the valid edge of the signal input from the TImn pin is selected as the count clock (f_{TCLK}).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 7 - 14 Format of Timer mode register mn (TMRmn) (2/4)

Address: F0190H, F0191H (TMR00) to F0196H, F0197H (TMR03) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2)	CKSmn1	CKSmn0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKSmn1	CKSmn0	0	CCSmn	SPLITmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0)	CKSmn1	CKSmn0	0	CCSmn	0 Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

(Bit 11 of TMRmn (n = 2))

MASTERmn	Selection between using channel n independently or simultaneously with another channel (as a slave or master)
0	Operates in independent channel operation function or as slave channel in simultaneous channel operation function.
1	Operates as master channel in simultaneous channel operation function.
Only the channel 2 can be set as a master channel (MASTERmn = 1). Be sure to use channel 0 is fixed to 0 (regardless of the bit setting, channel 0 operates as master, because it is the highest channel). Clear the MASTERmn bit to 0 for a channel that is used with the independent channel operation function.	

(Bit 11 of TMRmn (n = 1, 3))

SPLITmn	Selection of 8 or 16-bit timer operation for channels 1 and 3
0	Operates as 16-bit timer. (Operates in independent channel operation function or as slave channel in simultaneous channel operation function.)
1	Operates as 8-bit timer.

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

Note Bit 11 is fixed at 0 of read only, write is ignored.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 7 - 15 Format of Timer mode register mn (TMRmn) (3/4)

Address: F0190H, F0191H (TMR00) to F0196H, F0197H (TMR03) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2)	CKSmn1	CKSmn0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKSmn1	CKSmn0	0	CCSmn	SPLITmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0)	CKSmn1	CKSmn0	0	CCSmn	0 Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

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 the STSmn2 to STSmn0 bits is other than 010B, set the CISmn1 to CISmn0 bits to 10B.		

Note Bit 11 is fixed at 0 of read only, write is ignored.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 7 - 16 Format of Timer mode register mn (TMRmn) (4/4)

Address: F0190H, F0191H (TMR00) to F0196H, F0197H (TMR03) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2)	CKSm n1	CKSm n0	0	CCSm n	MAST ERmn	STSm n2	STSm n1	STSm n0	CISmn 1	CISmn 0	0	0	MDmn 3	MDmn 2	MDmn 1	MDmn 0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 1, 3)	CKSm n1	CKSm n0	0	CCSm n	SPLIT mn	STSm n2	STSm n1	STSm n0	CISmn 1	CISmn 0	0	0	MDmn 3	MDmn 2	MDmn 1	MDmn 0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0)	CKSm n1	CKSm n0	0	CCSm n	0 Note 1	STSm n2	STSm n1	STSm n0	CISmn 1	CISmn 0	0	0	MDmn 3	MDmn 2	MDmn 1	MDmn 0

MD mn3	MD mn2	MD mn1	Operation mode of channel n	Corresponding function	Count operation of TCR
0	0	0	Interval timer mode	Interval timer / Square wave output / Divider function / PWM output (master)	Counting down
0	1	0	Capture mode	Input pulse interval measurement	Counting up
0	1	1	Event counter mode	External event counter	Counting down
1	0	0	One-count mode	Delay counter / One-shot pulse output / PWM output (slave)	Counting down
1	1	0	Capture & one-count mode	Measurement of high-/low-level width of input signal	Counting up
Other than above			Setting prohibited		
The operation of each mode varies depending on MDmn0 bit (see table below).					

Operation mode (Value set by the MDmn3 to MDmn1 bits (see table above))	MDm n0	Setting of starting counting and interrupt
• Interval timer mode (0, 0, 0) • Capture mode (0, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
	1	Timer interrupt is generated when counting is started (timer output also changes).
• Event counter mode (0, 1, 1)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• One-count mode Note 2 (1, 0, 0)	0	Start trigger is invalid during counting operation. At that time, interrupt is not generated.
	1	Start trigger is valid during counting operation Note 3. At that time, interrupt is not generated.
• Capture & one-count mode (1, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either). Start trigger is invalid during counting operation. At that time, interrupt is not generated.

Note 1. Bit 11 is fixed at 0 of read only, write 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).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

7.3.5 Timer status register mn (TSRmn)

The TSRmn register indicates the overflow status of the counter of channel n.

The TSRmn register is valid only in the capture mode (MDmn3 to MDmn1 = 010B) and capture & one-count mode (MDmn3 to MDmn1 = 110B). See **Table 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 lower 8 bits of the TSRmn register can be set with an 8-bit memory manipulation instruction with TSRmnL.

Reset signal generation clears this register to 0000H.

Figure 7 - 17 Format of Timer status register mn (TSRmn)

Address: F01A0H, F01A1H (TSR00) to F01A6H, F01A7H (TSR03) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OVF

OVF	Counter overflow status of channel n
0	Overflow does not occur.
1	Overflow occurs.
When OVF = 1, this flag is cleared (OVF = 0) when the next value is captured without overflow.	

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

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 lower 8 bits of the TEm register can be set with a 1-bit or 8-bit memory manipulation instruction with TEmL.
Reset signal generation clears this register to 0000H.

Figure 7 - 18 Format of Timer channel enable status register m (TEm)

Address: F01B0H, F01B1H (TE0) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEm	0	0	0	0	TEHm3	0	TEHm1	0	0	0	0	0	TEm3	TEm2	TEm1	TEm0

TEHm3	Indication of whether operation of the higher 8-bit timer is enabled or stopped when channel 3 is in the 8-bit timer mode
0	Operation is stopped.
1	Operation is enabled.

TEHm1	Indication of whether operation of the higher 8-bit timer is enabled or stopped when channel 1 is in the 8-bit timer mode
0	Operation is stopped.
1	Operation is enabled.

TEmn	Indication of operation enable/stop status of channel n
0	Operation is stopped.
1	Operation is enabled.
This bit displays whether operation of the lower 8-bit timer for TEm1 and TEm3 is enabled or stopped when channel 1 or 3 is in the 8-bit timer mode.	

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

7.3.7 Timer channel start register m (TSm)

The TSm register is a trigger register that is used to initialize timer count register mn (TCRmn) and start the counting operation of each channel.

When a bit of this register is set to 1, the corresponding bit of timer channel enable status register m (TEm) is set to 1. The TSmn, TSHm1, TSHm3 bits are immediately cleared when operation is enabled (TEmn, TEHm1, TEHm3 = 1), because they are trigger bits.

The TSm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TSm register can be set with a 1-bit or 8-bit memory manipulation instruction with TSmL.

Reset signal generation clears this register to 0000H.

Figure 7 - 19 Format of Timer channel start register m (TSm)

Address: F01B2H, F01B3H (TS0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSm	0	0	0	0	TSHm3	0	TSHm1	0	0	0	0	0	TSm3	TSm2	TSm1	TSm0

TSHm3	Trigger to enable operation (start operation) of the higher 8-bit timer when channel 3 is in the 8-bit timer mode
0	No trigger operation
1	The TEHm3 bit is set to 1 and the count operation becomes enabled. The TCRm3 register count operation start in the interval timer mode in the count operation enabled state (see **Table 7 - 6** in **7.5.2 Start timing of counter**).

TSHm1	Trigger to enable operation (start operation) of the higher 8-bit timer when channel 1 is in the 8-bit timer mode
0	No trigger operation
1	The TEHm1 bit is set to 1 and the count operation becomes enabled. The TCRm1 register count operation start in the interval timer mode in the count operation enabled state (see **Table 7 - 6** in **7.5.2 Start timing of counter**).

TSmn	Operation enable (start) trigger of channel n
0	No trigger operation
1	The TEmn bit is set to 1 and the count operation becomes enabled. The TCRmn register count operation start in the count operation enabled state varies depending on each operation mode (see **Table 7 - 6** in **7.5.2 Start timing of counter**). This bit is the trigger to enable operation (start operation) of the lower 8-bit timer for TSm1 and TSm3 when channel 1 or 3 is in the 8-bit timer mode.

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

Caution 1. Be sure to clear bits 15 to 12, 10, 8 to 4 to "0"

Caution 2. When switching from a function that does not use TImn pin input to one that does, the following wait period is required from when timer mode register mn (TMRmn) is set until the TSmn (TSHm1, TSHm3) bit is set to 1.
When the TImn pin noise filter is enabled (TNFENmn = 1): Four cycles of the operation clock (f_{MCK})
When the TImn pin noise filter is disabled (TNFENmn = 0): Two cycles of the operation clock (f_{MCK})

Remark 1. When the TSm register is read, 0 is always read.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

7.3.8 Timer channel stop register m (TTm)

The TTm register is a trigger register that is used to stop the counting operation of each channel.

When a bit of this register is set to 1, the corresponding bit of timer channel enable status register m (TEm) is cleared to 0. The TTmn, TTHm1, TTHm3 bits are immediately cleared when operation is stopped (TEmn, TEHm1, TEHm3 = 0), because they are trigger bits.

The TTm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TTm register can be set with a 1-bit or 8-bit memory manipulation instruction with TTmL.

Reset signal generation clears this register to 0000H.

Figure 7 - 20 Format of Timer channel stop register m (TTm)

Address: F01B4H, F01B5H (TT0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTm	0	0	0	0	TTHm3	0	TTHm1	0	0	0	0	0	TTm3	TTm2	TTm1	TTm0

TTHm3	Trigger to stop operation of the higher 8-bit timer when channel 3 is in the 8-bit timer mode
0	No trigger operation
1	TEHm3 bit is cleared to 0 and the count operation is stopped.

TTHm1	Trigger to stop operation of the higher 8-bit timer when channel 1 is in the 8-bit timer mode
0	No trigger operation
1	TEHm1 bit is cleared to 0 and the count operation is stopped.

TTmn	Operation stop trigger of channel n
0	No trigger operation
1	TEmn bit is cleared to 0 and the count operation is stopped. The TTm1 and TTm3 bits respectively trigger stoppage of operation of the lower 8-bit timers in channels 1 and 3 when these channels are in the 8-bit timer mode.

Caution Be sure to clear bits 15 to 12, 10, 8 to 4 of the TTm register to "0".

Remark 1. When the TTm register is read, 0 is always read.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

7.3.9 Timer input select register 0 (TIS0)

The TIS0 register is used to select the channels 0 and 1 timer input.

The TIS0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7 - 21 Format of Timer input select register 0 (TIS0)

Address: F0074H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	TIS04	0	TIS02	TIS01	TIS00

TIS04	Selection of timer input used with channel 0
0	Input signal of timer input pin (TI00)
1	Event input signal from ELC

TIS02	TIS01	TIS00	Selection of timer input used with channel 1
0	0	0	Input signal of timer input pin (TI01)
0	0	1	Event input signal from ELC
0	1	0	Input signal of timer input pin (TI01)
0	1	1	Middle-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 above			Setting prohibited

Caution 1. The widths at high and low level of the timer input to be selected must both be at least 1/f_{MCK} + 10 ns. When f_{SUB} is selected for the f_{CLK} (CSS in CKC register = 1), the TIS02 bit cannot be set to 1.

Caution 2. When selecting an event input signal from the ELC using timer input select register 0 (TIS0), select f_{CLK} using timer clock select register 0 (TPS0).

7.3.10 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 lower 8 bits of the TOEm register can be set with a 1-bit or 8-bit memory manipulation instruction with TOEmL.

Reset signal generation clears this register to 0000H.

Figure 7 - 22 Format of Timer output enable register m (TOEm)

Address: F01BAH, F01BBH (TOE0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOEm	0	0	0	0	0	0	0	0	0	0	0	0	TOEm3	TOEm2	TOEm1	TOEm0

TOEmn	Timer output enable/disable of channel n
0	Timer output is disabled. Timer operation is not applied to the TOmn bit and the output is fixed. Writing to the TOmn bit is enabled and the level set in the TOmn bit is output from the TOmn pin.
1	Timer output is enabled. Timer operation is applied to the TOmn bit and an output waveform is generated. Writing to the TOmn bit is ignored.

Caution Be sure to clear bits 15 to 4 to "0".

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

7.3.11 Timer output register m (TOm)

The TOm register is a buffer register of timer output of each channel.
The value of each bit in this register is output from the timer output pin (TOmn) of each channel.
The TOmn bit oh this register can be rewritten by software only when timer output is disabled (TOEmn = 0).
When timer output is enabled (TOEmn = 1), rewriting this register by software is ignored, and the value is changed only by the timer operation.
To use the TI00, TO00, TI01/TO01, TI02/TO02, TI03/TO03 pins 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 lower 8 bits of the TOm register can be set with an 8-bit memory manipulation instruction with TOmL.
Reset signal generation clears this register to 0000H.

Figure 7 - 23 Format of Timer output register m (TOm)

Address: F01B8H, F01B9H (TO0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOm	0	0	0	0	0	0	0	0	0	0	0	0	TOm3	TOm2	TOm1	TOm0

TOmn	Timer output of channel n
0	Timer output value is 0.
1	Timer output value is 1.

Caution Be sure to clear bits 15 to 4 to "0".

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

7.3.12 Timer output level register m (TOLm)

The TOLm register is a register that controls the timer output level of each channel.

The setting of the inverted output of channel n by this register is reflected at the timing of set or reset of the timer output signal while the timer output is enabled (TOEmn = 1) in the Slave channel output mode (TOMmn = 1). In the master channel output mode (TOMmn = 0), this register setting is invalid.

The TOLm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOLm register can be set with an 8-bit memory manipulation instruction with TOLmL.

Reset signal generation clears this register to 0000H.

Figure 7 - 24 Format of Timer output level register m (TOLm)

Address: F01BCH, F01BDH (TOL0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOLm	0	0	0	0	0	0	0	0	0	0	0	0	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 clear bits 15 to 4, and 0 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), n: Channel number (n = 0 to 3)

7.3.13 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 lower 8 bits of the TOMm register can be set with an 8-bit memory manipulation instruction with TOMmL.
Reset signal generation clears this register to 0000H.

Figure 7 - 25 Format of Timer output mode register m (TOMm)

Address: F01BEH, F01BFH (TOM0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOMm	0	0	0	0	0	0	0	0	0	0	0	0	TOMm3	TOMm2	TOMm1	0

TOMmn	Control of timer output mode of channel n
0	Master channel output mode (to produce toggle output by timer interrupt request signal (INTTMmn))
1	Slave channel output mode (output is set by the timer interrupt request signal (INTTMmn) of the master channel, and reset by the timer interrupt request signal (INTTM0p) of the slave channel)

Caution Be sure to clear bits 15 to 4, and 0 to “0”.

Remark m: Unit number (m = 0)
n: Channel number
n = 0 to 3 (n = 0, 2 for master channel)
p: Slave channel number
n = 0, p = 1, 2, 3
n = 2, p = 3
(For details of the relation between the master channel and slave channel, refer to **7.4.1 Basic rules of simultaneous channel operation function**.)

7.3.14 Noise filter enable register 1 (NFEN1)

The NFEN1 register is used to set whether the noise filter can be used for the timer input signal to each channel.
Enable the noise filter by setting the corresponding bits to 1 on the pins in need of noise removal. When the noise filter is 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 OFF, only synchronization is performed with the operation clock of target channel (f_{MCK}) Note.
The NFEN1 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Note For details, see **7.5.1 (2) When valid edge of input signal via the TImn pin is selected (CCSmn = 1)**, **7.5.2 Start timing of counter**, and **7.7 Timer Input (TImn) Control**.

Figure 7 - 26 Format of Noise filter enable register 1 (NFEN1)

Address: F0071H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN1	0	0	0	0	TNFEN03	TNFEN02	TNFEN01	TNFEN00

TNFEN03	Enable/disable using noise filter of TI03 pin or RxD0 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN02	Enable/disable using noise filter of TI02 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN01	Enable/disable using noise filter of TI01 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN00	Enable/disable using noise filter of TI00 pin input signal
0	Noise filter OFF
1	Noise filter ON

Remark The presence or absence of timer I/O pins of channel 0 to 3 depends on the product. See **Table 7 - 2 Timer I/O Pins provided in Each Product** for details.

7.3.15 Registers controlling port functions of pins to be used for timer I/O

Using port pins for the timer array unit functions requires setting of the registers that control the port functions multiplexed on the target pins (port mode register (PMxx), port register (Pxx), and port mode control register (PMCxx)). For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.2 Port registers (Pxx)**, and **4.3.6 Port mode control registers (PMCxx)**.

The port mode register (PMxx), port register (Pxx), and port mode control register (PMCxx) to be set depend on the product. For details, see **4.5 Register Settings When Using Alternate Function**.

When using the ports (such as P30/TO01) to be shared with the timer output pin for timer output, set the port mode control register (PMCxx) bit, port mode register (PMxx) bit and port register (Pxx) bit corresponding to each port to 0.

Example: When using P30/TO01 for timer output
Set the PMC30 bit of port mode control register 3 to 0.
Set the PM30 bit of port mode register 3 to 0.
Set the P30 bit of port register 3 to 0.

When using the ports (such as P30/TI00) to be shared with the timer input pin for timer input, set the port mode register (PMxx) bit corresponding to each port to 1. And set the port mode control register (PMCxx) bit corresponding to each port to 0. At this time, the port register (Pxx) bit may be 0 or 1.

Example: When using P30/TI00 for timer input
Set the PMC30 bit of port mode control register 3 to 0.
Set the PM30 bit of port mode register 3 to 1.
Set the P30 bit of port register 3 to 0 or 1.

7.4 Basic Rules of Timer Array Unit

7.4.1 Basic rules of simultaneous channel operation function

When simultaneously using multiple channels, namely, a combination of a master channel (a reference timer mainly counting the cycle) and slave channels (timers operating according to the master channel), the following rules apply.

(1) Only an even channel (channel 0, 2) can be set as a master channel.
(2) Any channel, except channel 0, can be set as a slave channel.
(3) The slave channel must be lower than the master channel.
Example: If channel 0 is set as a master channel, channel 1 or those that follow (channels 1, 2, 3) can be set as a slave channel.

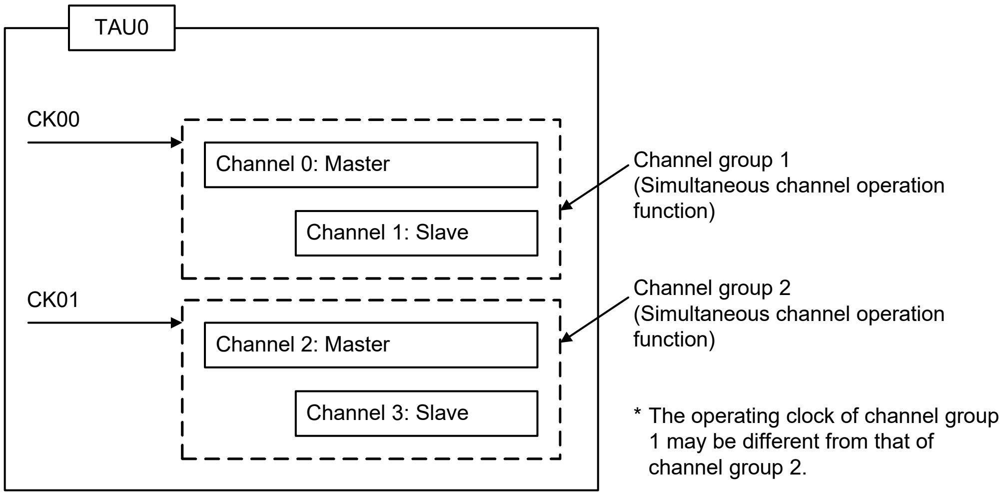
(4) Two or more slave channels can be set for one master channel.
(5) When two or more master channels are to be used, slave channels with a master channel between them may not be set.
Example: If channels 0 and 2 are set as master channels, channels 1 can be set as the slave channel of master channel 0. Channel 3 cannot be set as the slave channel of master channel 0.

(6) The operating clock for a slave channel in combination with a master channel must be the same as that of the master channel. The CKSmn0, CKSmn1 bits (bit 15, 14 of timer mode register mn (TMRmn)) of the slave channel that operates in combination with the master channel must be the same value as that of the master channel.
(7) A master channel can transmit INTTMmn (interrupt), start software trigger, and count clock to the lower channels.
(8) A slave channel can use INTTMmn (interrupt), a start software trigger, or the count clock of the master channel as a source clock, but cannot transmit its own INTTMmn (interrupt), start software trigger, or count clock to channels with lower channel numbers.
(9) A master channel cannot use INTTMmn (interrupt), a start software trigger, or the count clock from the other higher master channel as a source clock.
(10) To simultaneously start channels that operate in combination, the channel start trigger bit (TSmn) of the channels in combination must be set at the same time.
(11) During the counting operation, a TSmn bit of a master channel or TSmn bits of all channels which are operating simultaneously can be set. It cannot be applied to TSmn bits of slave channels alone.
(12) To stop the channels in combination simultaneously, the channel stop trigger bit (TTmn) of the channels in combination must be set at the same time.
(13) CKm2/CKm3 cannot be selected while channels are operating simultaneously, because the operating clocks of master channels and slave channels have to be synchronized.
(14) Timer mode register m0 (TMRm0) has no master bit (it is fixed as “0”). However, as channel 0 is the highest channel, it can be used as a master channel during simultaneous operation.

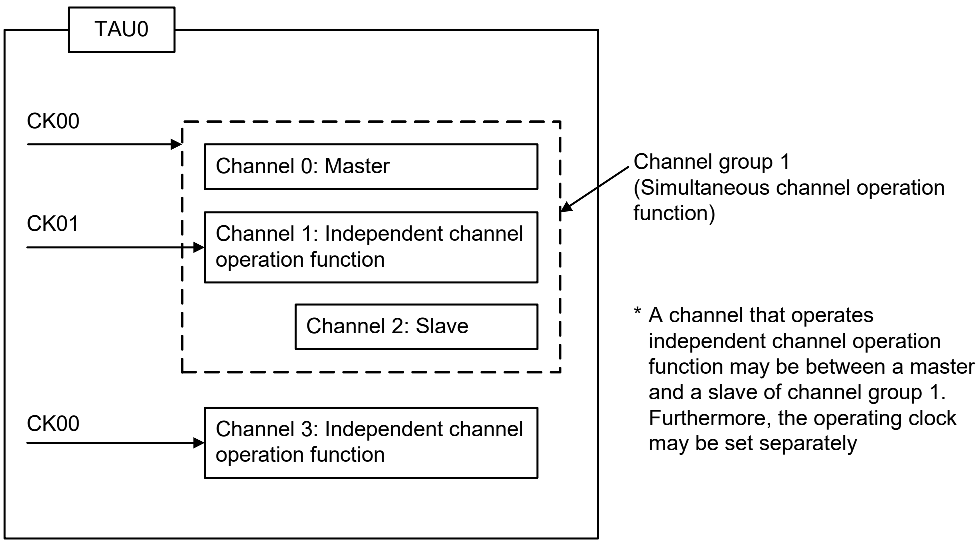
The rules of the simultaneous channel operation function are applied in a channel group (a master channel and slave channels forming one simultaneous channel operation function).
If two or more channel groups that do not operate in combination are specified, the basic rules of the simultaneous channel operation function in **7.4.1 Basic rules of simultaneous channel operation function** do not apply to the channel groups.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Example 1



Example 2



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

The 8-bit timer operation function makes it possible to use a 16-bit timer channel in a configuration consisting of two 8-bit timer channels.

This function can only be used for channels 1 and 3, and there are several rules for using it.

The basic rules for this function are as follows:

(1) The 8-bit timer operation function applies only to channels 1 and 3.

(2) When using 8-bit timers, set the SPLIT bit of timer mode register mn (TMRmn) to 1.

(3) The higher 8 bits can be operated as the interval timer function.

(4) At the start of operation, the higher 8 bits output INTTMm1H/INTTMm3H (an interrupt) (which is the same operation performed when MDmn0 is set to 1).

(5) The operation clock of the higher 8 bits is selected according to the CKSmn1 and CKSmn0 bits of the lower-bit TMRmn register.

(6) For the higher 8 bits, the TSHm1/TSHm3 bit is manipulated to start channel operation and the TTHm1/TTHm3 bit is manipulated to stop channel operation. The channel status can be checked using the TEHm1/TEHm3 bit.

(7) The lower 8 bits operate according to the TMRmn register settings. The following three functions support operation of the lower 8 bits:
- Interval timer function
- External event counter function
- Delay count function

(8) For the lower 8 bits, the TSm1/TSm3 bit is manipulated to start channel operation and the TTm1/TTm3 bit is manipulated to stop channel operation. The channel status can be checked using the TEm1/TEm3 bit.

(9) During 16-bit operation, manipulating the TSHm1, TSHm3, TTHm1, and TTHm3 bits is invalid. The TSm1, TSm3, TTm1, and TTm3 bits are manipulated to operate channels 1 and 3. The TEHm3 and TEHm1 bits are not changed.

(10) For the 8-bit timer function, the simultaneous operation functions (one-shot pulse, PWM, and multiple PWM) cannot be used.

Remark m: Unit number (m = 0), n: Channel number (n = 1, 3)

7.5 Operation of Counter

7.5.1 Count clock (fTCLK)

The count clock (fTCLK) of the timer array unit can be selected between following by CCSmn bit of timer mode register mn (TMRmn).

- Operation clock (fMCK) specified by the CKSmn0 and CKSmn1 bits
- Valid edge of input signal input from the TImn pin

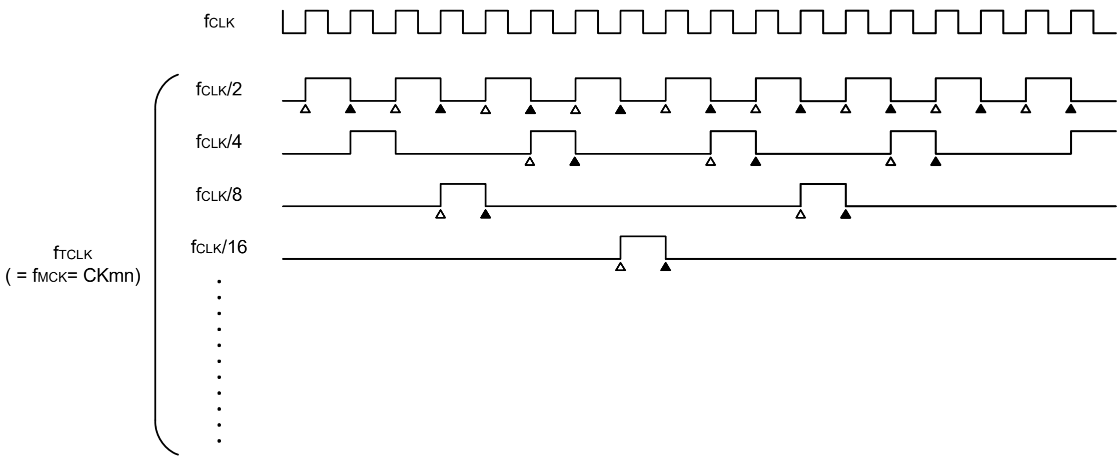
Because the timer array unit is designed to operate in synchronization with fCLK, the timings of the count clock (fTCLK) are shown below.

(1) When operation clock (fMCK) specified by the CKSmn0 and CKSmn1 bits is selected (CCSmn = 0)

The count 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 count register mn (TCRmn) delayed by one period of fCLK from rising edge of the count clock, because of synchronization with fCLK. But, this is described as “counting at rising edge of the count clock”, as a matter of convenience.

Figure 7 - 27 Timing of fCLK and count clock (fTCLK) (When CCSmn = 0)



Remark 1. △ : Rising edge of the count clock

▲ : Synchronization, increment/decrement of counter

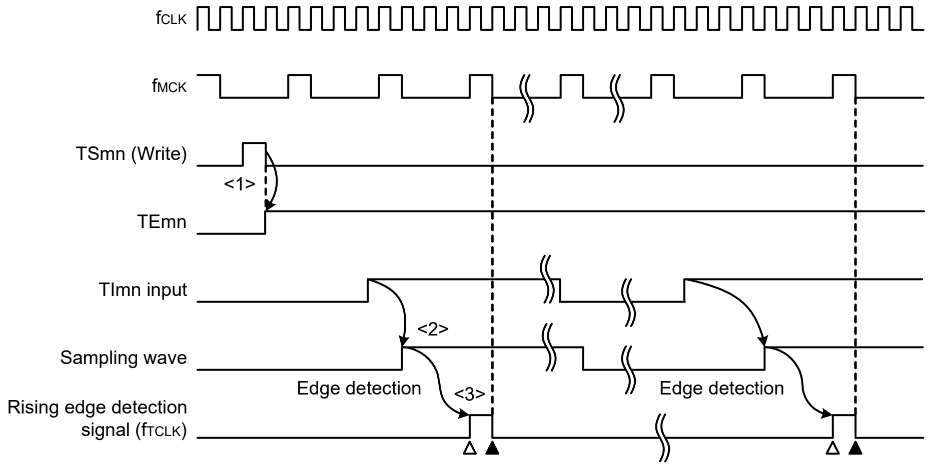
Remark 2. fCLK: CPU/peripheral hardware clock

(2) When valid edge of input signal via the TImn pin is selected (CCSmn = 1)

The count clock (f_{TCLK}) becomes the signal that detects valid edge of input signal via the TImn pin and synchronizes next rising f_{MCK}. The count clock (f_{TCLK}) is delayed for 1 to 2 period of f_{MCK} from the input signal via the TImn pin (when a noise filter is used, the delay becomes 3 to 4 clock).

Counting of timer count register mn (TCRmn) delayed by one period of f_{CLK} from rising edge of the count clock, because of synchronization with f_{CLK}. But, this is described as “counting at valid edge of input signal via the TImn pin”, as a matter of convenience.

Figure 7 - 28 Timing of f_{CLK} and count clock (f_{TCLK}) (When CCSmn = 1, noise filter unused)



<1> Setting TSmn bit to 1 enables the timer to be started and to become wait state for valid edge of input signal via the TImn pin.
<2> The rise of input signal via the TImn pin is sampled by f_{MCK}.
<3> The edge is detected by the rising of the sampled signal and the detection signal (count clock) is output.

Remark 1. △ : Rising edge of the count clock
▲ : Synchronization, increment/decrement of counter

Remark 2. f_{CLK}: CPU/peripheral hardware clock
f_{MCK}: Operation clock of channel n

Remark 3. The waveform of the input signal via TImn pin of the input pulse interval measurement, the measurement of high/low width of input signal, and the delay counter, and the one-shot pulse output are the same as that shown in Figure 7 - 28.

7.5.2 Start timing of counter

Timer count register mn (TCRmn) becomes enabled to operation by setting of TSmn bit of timer channel start register m (TSm).

Operations from count operation enabled state to timer count Register mn (TCRmn) count start is shown in Table 7 - 6.

Table 7 - 6 Operations from Count Operation Enabled State to Timer count Register mn (TCRmn) Count Start

Timer operation mode	Operation when TSmn = 1 is set
• Interval timer mode	No operation is carried out from start trigger detection (TSmn=1) until count clock generation. The first count clock loads the value of the TDRmn register to the TCRmn register and the subsequent count clock performs count down operation (see **7.5.3 (1) Operation of interval timer mode**).
• Event counter mode	Writing 1 to the TSmn bit loads the value of the TDRmn register to the TCRmn register. If detect edge of TImn input, the subsequent count clock performs count down operation (see **7.5.3 (2) Operation of event counter mode**).
• Capture mode	No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to the TCRmn register and the subsequent count clock performs count up operation (see **7.5.3 (3) Operation of capture mode (input pulse interval measurement)**).
• One-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads the value of the TDRmn register to the TCRmn register and the subsequent count clock performs count down operation (see **7.5.3 (4) Operation of one-count mode**).
• Capture & one-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to the TCRmn register and the subsequent count clock performs count up operation (see **7.5.3 (5) Operation of capture & one-count mode (high-level width measurement)**).

7.5.3 Operation of counter

Here, the counter operation in each mode is explained.

(1) Operation of interval timer mode

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit. Timer count register mn (TCRmn) holds the initial value until count clock generation.

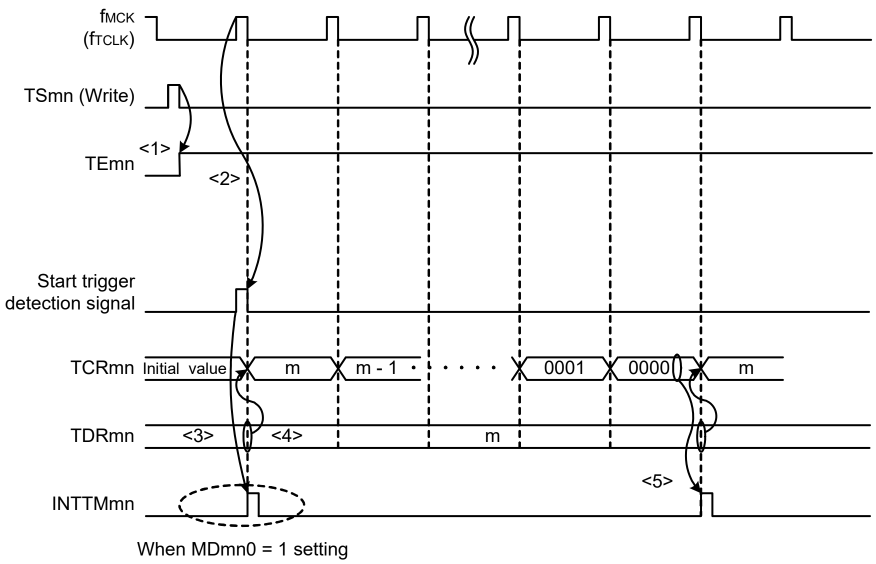
<2> A start trigger is generated at the first count clock after operation is enabled.

<3> When the MDmn0 bit is set to 1, INTTMmn is generated by the start trigger.

<4> By the first count clock after the operation enable, the value of timer data register mn (TDRmn) is loaded to the TCRmn register and counting starts in the interval timer mode.

<5> When the TCRmn register counts down and its count value is 0000H, INTTMmn is generated and the value of timer data register mn (TDRmn) is loaded to the TCRmn register and counting keeps on.

Figure 7 - 29 Operation Timing (In Interval Timer Mode)



Caution **In the first cycle operation of count clock after writing the TSmn bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.**

Remark fMCK, the start trigger detection signal, and INTTMmn become active between one clock in synchronization with fCLK.

(2) Operation of event counter mode

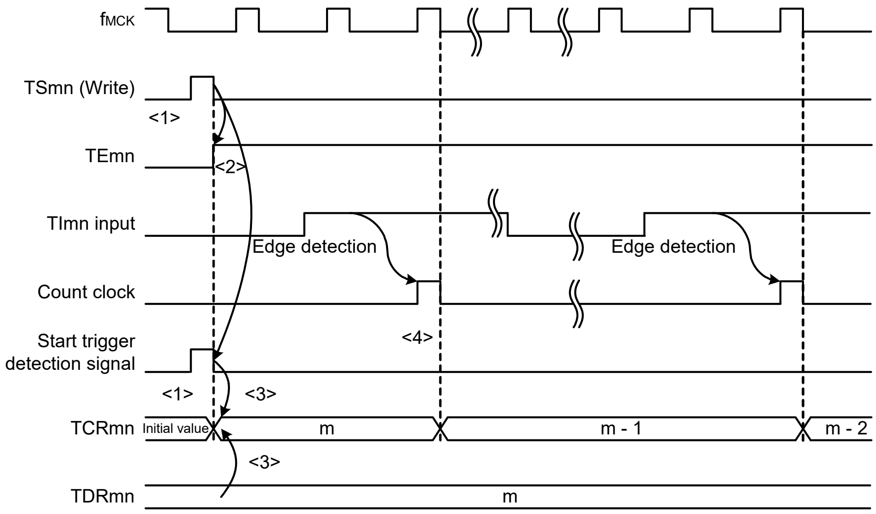
<1> Timer count register mn (TCRmn) holds its initial value while operation is stopped (TEmn = 0).

<2> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.

<3> As soon as 1 has been written to the TSmn bit and 1 has been set to the TEmn bit, the value of timer data register mn (TDRmn) is loaded to the TCRmn register to start counting.

<4> After that, the TCRmn register value is counted down according to the count clock of the valid edge of the TImn input.

Figure 7 - 30 Operation Timing (In Event Counter Mode)



Remark Figure 7 - 30 shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (f_{MCK}).

(3) Operation of capture mode (input pulse interval measurement)

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.

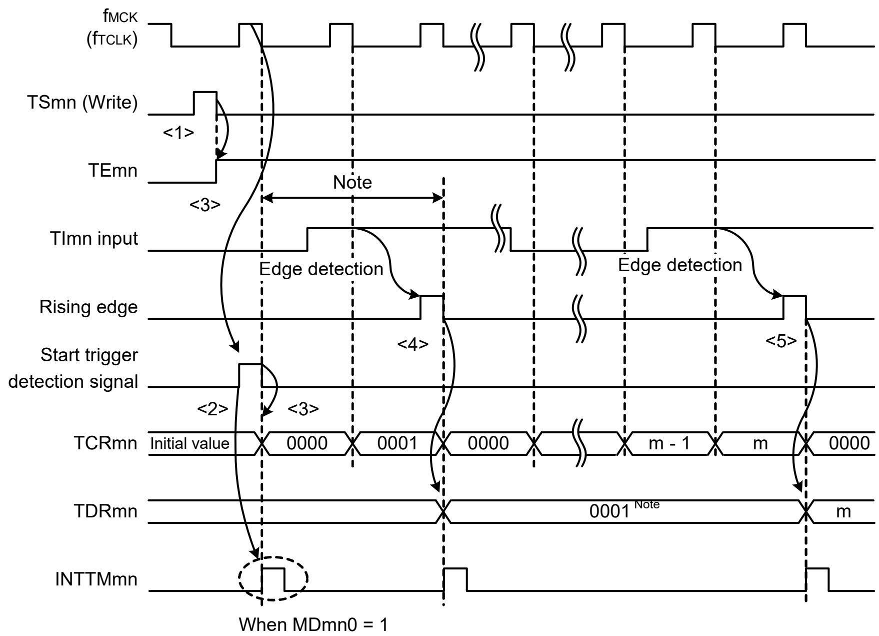
<2> Timer count register mn (TCRmn) holds the initial value until count clock generation.

<3> A start trigger is generated at the first count clock after operation is enabled. And the value of 0000H is loaded to the TCRmn register and counting starts in the capture mode. (When the MDmn0 bit is set to 1, INTTMmn is generated by the start trigger.)

<4> On detection of the valid edge of the TImn input, the value of the TCRmn register is captured to timer data register mn (TDRmn) and INTTMmn is generated. However, this capture value is no meaning. The TCRmn register keeps on counting from 0000H.

<5> On next detection of the valid edge of the TImn input, the value of the TCRmn register is captured to timer data register mn (TDRmn) and INTTMmn is generated.

Figure 7 - 31 Operation Timing (In Capture Mode: Input Pulse Interval Measurement)



Note If a clock has been input to TImn (the trigger exists) when capturing starts, counting starts when a trigger is detected, even if no edge is detected. Therefore, the first captured value (<4>) does not determine a pulse interval (in the above figure, 0001 just indicates two clock cycles but does not determine the pulse interval) and so the user can ignore it.

Caution **In the first cycle operation of count clock after writing the TSmn bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.**

Remark 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 count clock (fMCK).

(4) Operation of one-count mode

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.

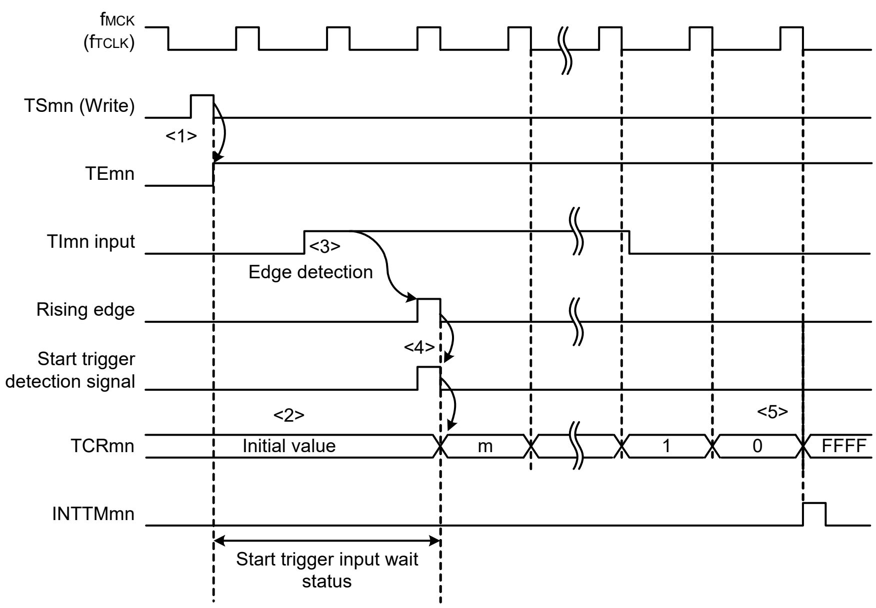
<2> Timer count register mn (TCRmn) holds the initial value until start trigger generation.

<3> Rising edge of the TImn input is detected.

<4> On start trigger detection, the value of timer data register mn (TDRmn) is loaded to the TCRmn register and count starts.

<5> When the TCRmn register counts down and its count value is 0000H, INTTMmn is generated and the value of the TCRmn register becomes FFFFH and counting stops.

Figure 7 - 32 Operation Timing (In One-count Mode)



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 count clock (fMCK).

(5) Operation of capture & one-count mode (high-level width measurement)

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit of timer channel start register m (TSm).

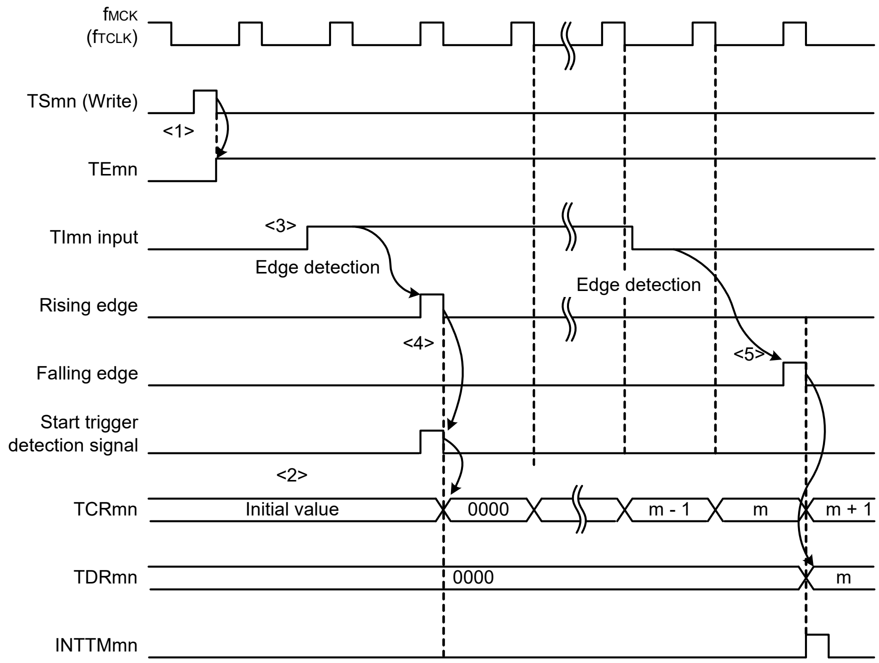
<2> Timer count register mn (TCRmn) holds the initial value until start trigger generation.

<3> Rising edge of the TImn input is detected.

<4> On start trigger detection, the value of 0000H is loaded to the TCRmn register and count starts.

<5> On detection of the falling edge of the TImn input, the value of the TCRmn register is captured to timer data register mn (TDRmn) and INTTMmn is generated.

Figure 7 - 33 Operation Timing (In Capture & One-count Mode: High-level Width Measurement)

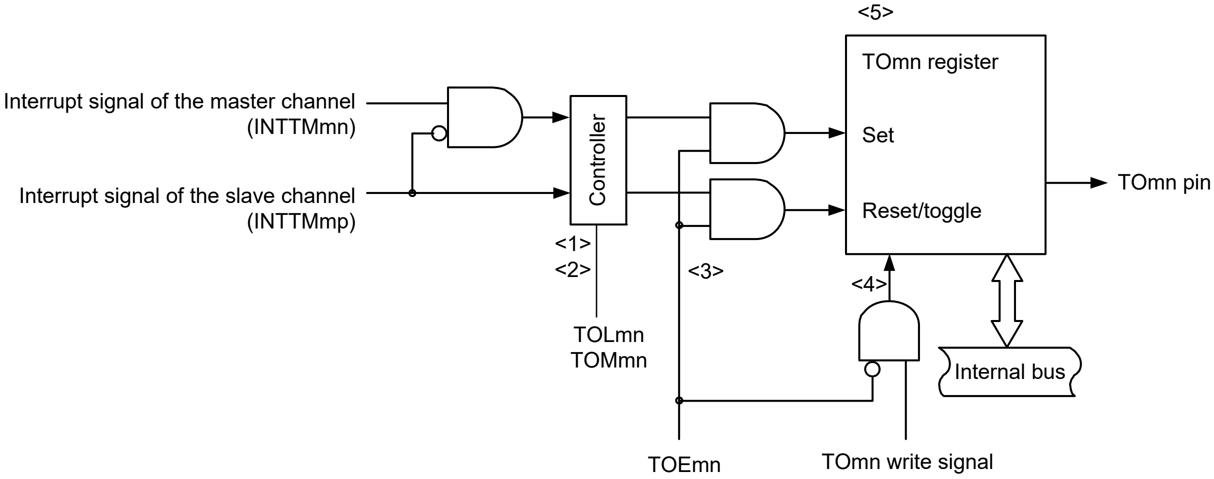


Remark Figure 7 - 33 shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (fMCK).

7.6 Channel Output (TOmn pin) Control

7.6.1 TOmn pin output circuit configuration

Figure 7 - 34 Output Circuit Configuration



The following describes the TOmn pin output circuit.

<1> When TOMmn = 0 (master channel output mode), the set value of timer output level register m (TOLm) is ignored and only INTTM0p (slave channel timer interrupt) is transmitted to timer output register m (TOm).

<2> When TOMmn = 1 (slave channel output mode), both INTTMmn (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TOm register.
At this time, the TOLm register becomes valid and the signals are controlled as follows:

When TOLmn = 0: Forward operation (INTTMmn → set, INTTM0p → reset)
When TOLmn = 1: Reverse operation (INTTMmn → reset, INTTM0p → set)

When INTTMmn and INTTM0p are simultaneously generated, (0% output of PWM), INTTM0p (reset signal) takes priority, and INTTMmn (set signal) is masked.

<3> While timer output is enabled (TOEmn = 1), INTTMmn (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TOm register. Writing to the TOm register (TOmn write signal) becomes invalid.
When TOEmn = 1, the TOmn pin output never changes with signals other than interrupt signals.
To initialize the TOmn pin output level, it is necessary to set timer operation is stopped (TOEmn = 0) and to write a value to the TOm register.

<4> While timer output is disabled (TOEmn = 0), writing to the TOmn bit to the target channel (TOmn write signal) becomes valid. When timer output is disabled (TOEmn = 0), neither INTTMmn (master channel timer interrupt) nor INTTM0p (slave channel timer interrupt) is transmitted to the TOm register.

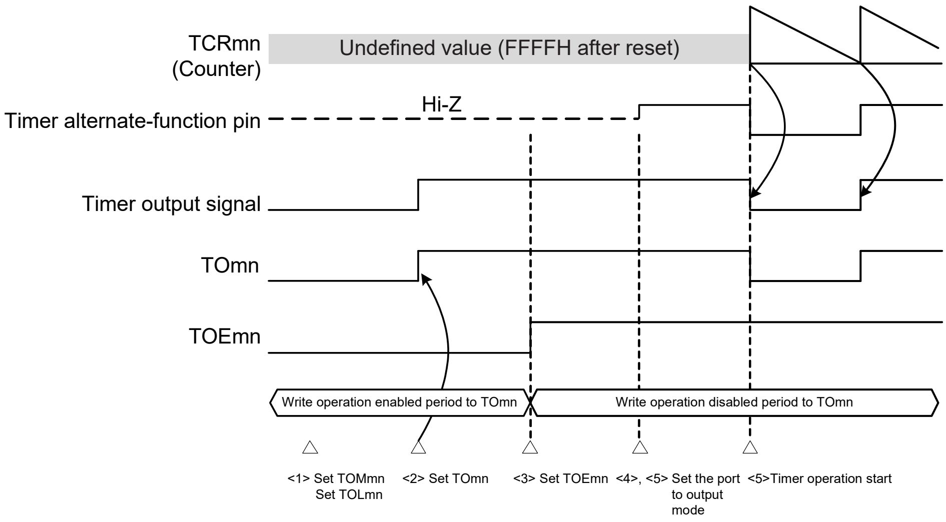
<5> The TOm register can always be read, and the TOmn pin output level can be checked.

Remark m: Unit number (m = 0)
n: Channel number
n = 0 to 3 (n = 0, 2 for master channel)
p: Slave channel number
n = 0: p = 1, 2, 3
n = 2: p = 3

7.6.2 TOmn Pin Output Setting

The following figure shows the procedure and status transition of the TOmn output pin from initial setting to timer operation start.

Figure 7 - 35 Status Transition from Timer Output Setting to Operation Start



<1> The operation mode of timer output is set.

- TOMmn bit (0: Master channel output mode, 1: Slave channel output mode)
- TOLmn bit (0: Positive logic output, 1: Negative logic output)

<2> The timer output signal is set to the initial status by setting timer output register m (TOm).

<3> The timer output operation is enabled by writing 1 to the TOEmn bit (writing to the TOm register is disabled).

<4> The port is set to digital I/O by port mode control register (PMCxx) (see **7.3.15 Registers controlling port functions of pins to be used for timer I/O**).

<5> The port I/O setting is set to output (see **7.3.15 Registers controlling port functions of pins to be used for timer I/O**).

<6> The timer operation is enabled (TSmn = 1).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

7.6.3 Cautions on Channel Output Operation

(1) Changing values set in the registers TOm, TOEm, TOLm, and TOMm during timer operation
Since the timer operations (operations of timer count register mn (TCRmn) and timer data register mn (TDRmn)) are independent of the TOmn output circuit and changing the values set in timer output register m (TOm), timer output enable register m (TOEm), 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 by 7.8 and 7.9.
When the values set to the TOEm and TOLm 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), n: Channel number (n = 0 to 3)

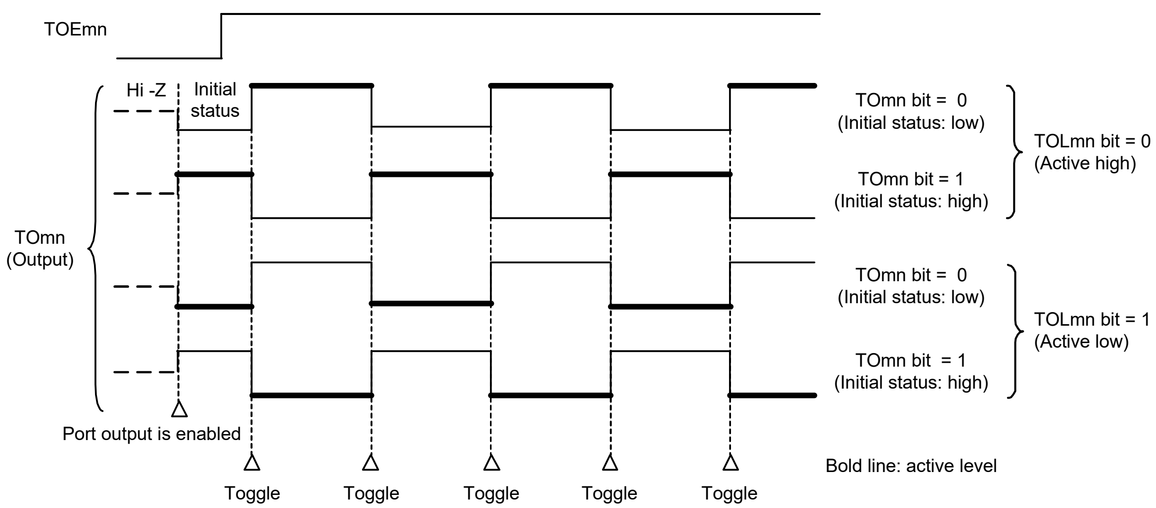
(2) Default level of TOmn pin and output level after timer operation start

The change in the output level of the TOmn pin when timer output register m (TOm) is written while timer output is disabled (TOEmn = 0), the initial level is changed, and then timer output is enabled (TOEmn = 1) before port output is enabled, is shown below.

(a) When operation starts with master channel output mode (TOMmn = 0) setting

The setting of timer output level register m (TOLm) is invalid when master channel output mode (TOMmn = 0). When the timer operation starts after setting the default level, the toggle signal is generated and the output level of the TOmn pin is reversed.

Figure 7 - 36 TOmn Pin Output Status at Toggle Output (TOMmn = 0)

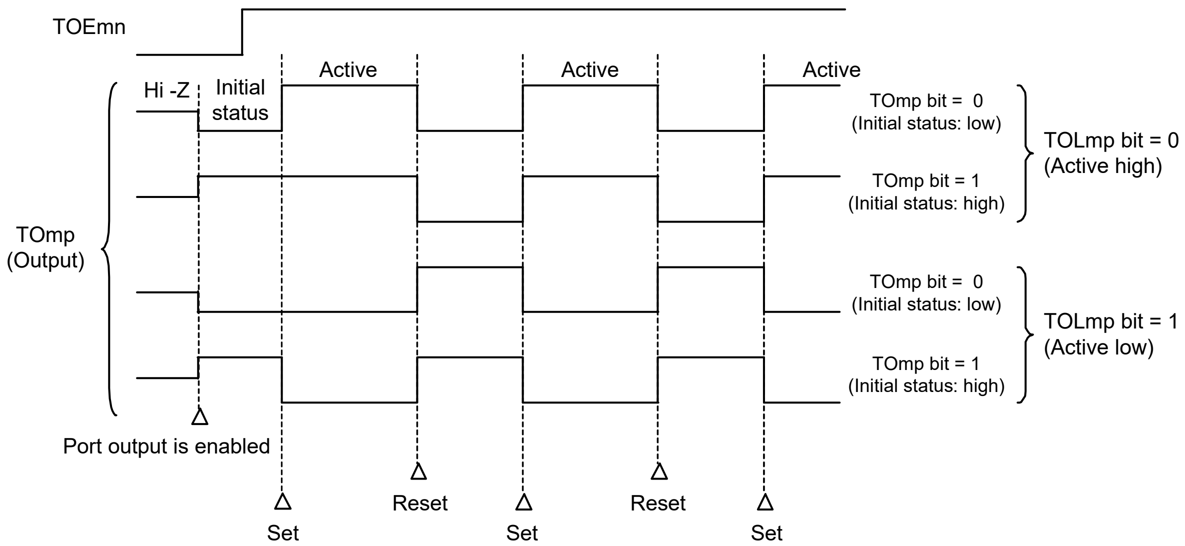


Remark 1. Toggle: Reverse TOmn pin output status

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

(b) When operation starts with slave channel output mode (TOMmn = 1) setting (PWM output))
When slave channel output mode (TOMmn = 1), the active level is determined by timer output level register m (TOLm) setting.

Figure 7 - 37 TOmn Pin Output Status at PWM Output (TOMmn = 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), n: Channel number (p = 1 to 3)

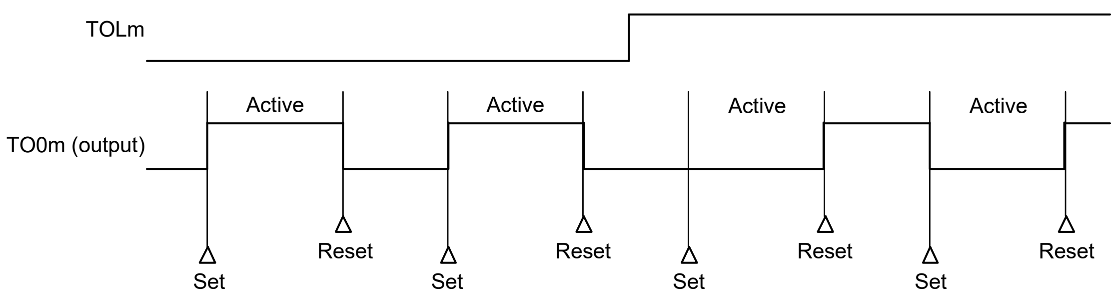
(3) Operation of TOmn pin in slave channel output mode (TOMmn = 1)

(a) When timer output level register m (TOLm) setting has been changed during timer operation

When the TOLm register setting has been changed during timer operation, the setting becomes valid at the generation timing of the TOmn pin change condition. Rewriting the TOLm register does not change the output level of the TOmn pin.

The operation when TOMmn is set to 1 and the value of the TOLm register is changed while the timer is operating (TEmn = 1) is shown below.

Figure 7 - 38 Operation when TOLm Register Has Been 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), n: Channel number (n = 0 to 3)

(b) Set/reset timing

To realize 0%/100% output at PWM output, the TOmn pin/TOmn bit set timing at master channel timer interrupt (INTTMmn) generation is delayed by 1 count clock by the slave channel.

If the set condition and reset condition are generated at the same time, a higher priority is given to the latter.

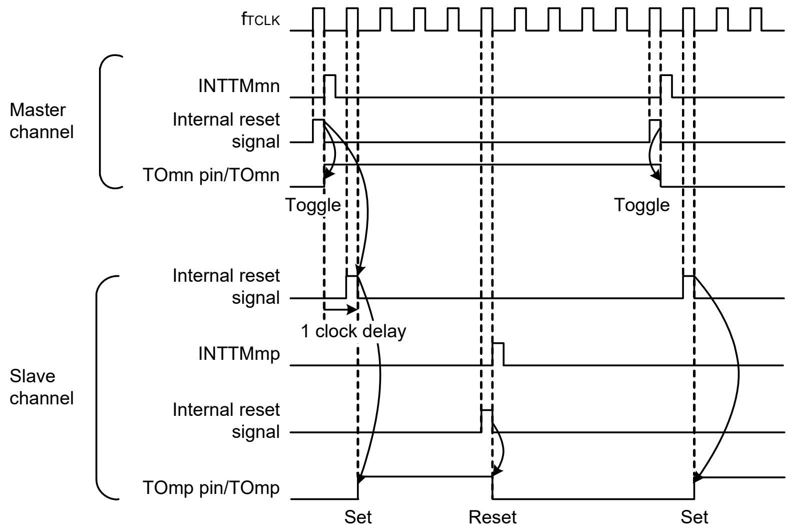
Figure 7 - 39 shows the set/reset operating statuses where the master/slave channels are set as follows.

Master channel: TOEmn = 1, TOMmn = 0, TOLmn = 0

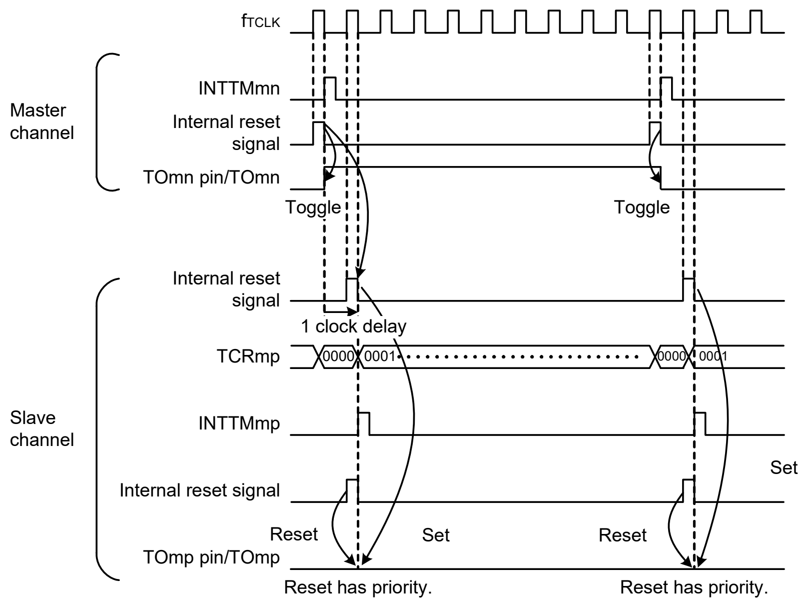
Slave channel: TOEmp = 1, TOMmp = 1, TOLmp = 0

Figure 7 - 39 Set/Reset Timing Operating Statuses

(1) Basic operation timing



(2) Operation timing when 0% duty



Remark 1. Internal reset signal:TOmn pin reset/toggle signal
Internal set signal: TOmn pin set signal

Remark 2. m: Unit number (m = 0)
n: Channel number
n = 0 to 3 (n = 0, 2 for master channel)
p: Slave channel number
n = 0: p = 1, 2, 3
n = 2: p = 3

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 - 40 Example of TO0n Bit Collective Manipulation

Before writing

TO0	0	0	0	0	0	0	0	0	0	0	0	0	TO03 1	TO02 0	TO01 1	TO00 0
TOE0	0	0	0	0	0	0	0	0	0	0	0	0	TOE03 0	TOE02 0	TOE01 0	TOE00 1

Data to be written

0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

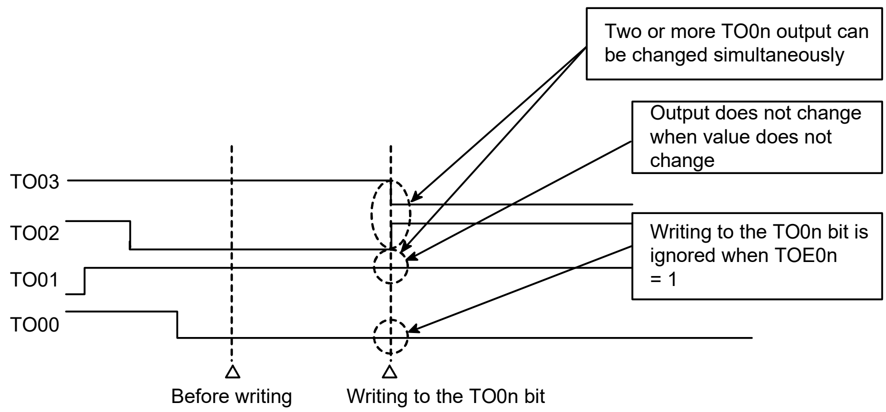
After writing

TO0	0	0	0	0	0	0	0	0	0	0	0	0	TO03 0	TO02 1	TO01 1	TO00 0

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 - 41 TO0n Pin Statuses by Collective Manipulation of TO0n Bit



Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

7.6.5 Timer Interrupt and TOmn Pin Output at Operation Start

In the interval timer mode or capture mode, the MDmn0 bit in timer mode register mn (TMRmn) sets whether or not to generate a timer interrupt at count start.

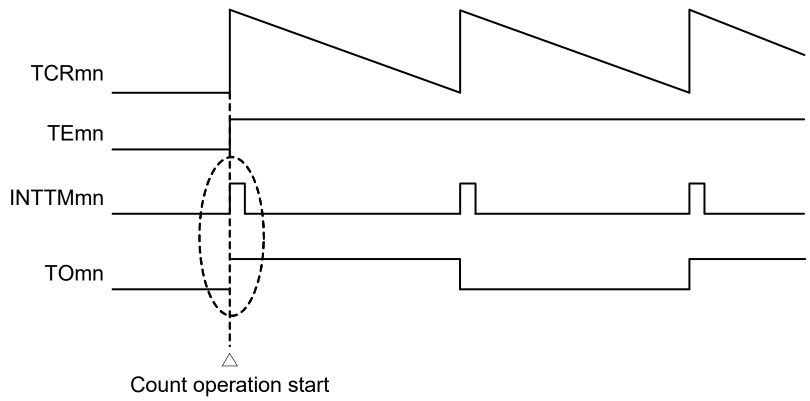
When MDmn0 is set to 1, the count operation start timing can be known by the timer interrupt (INTTMmn) generation.

In the other modes, neither timer interrupt at count operation start nor TOmn output is controlled.

Figure 7 - 42 shows operation examples when the interval timer mode (TOEmn = 1, TOMmn = 0) is set.

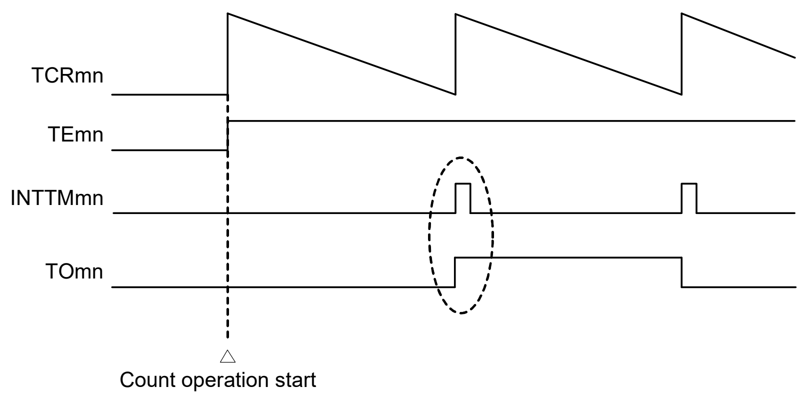
Figure 7 - 42 Operation examples of timer interrupt at count operation start and TOmn output

(a) When MDmn0 is set to 1



When MDmn0 is set to 1, a timer interrupt (INTTMmn) is output at count operation start, and TOmn performs a toggle operation.

(b) When MDmn0 is set to 0



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), n: Channel number (n = 0 to 3)

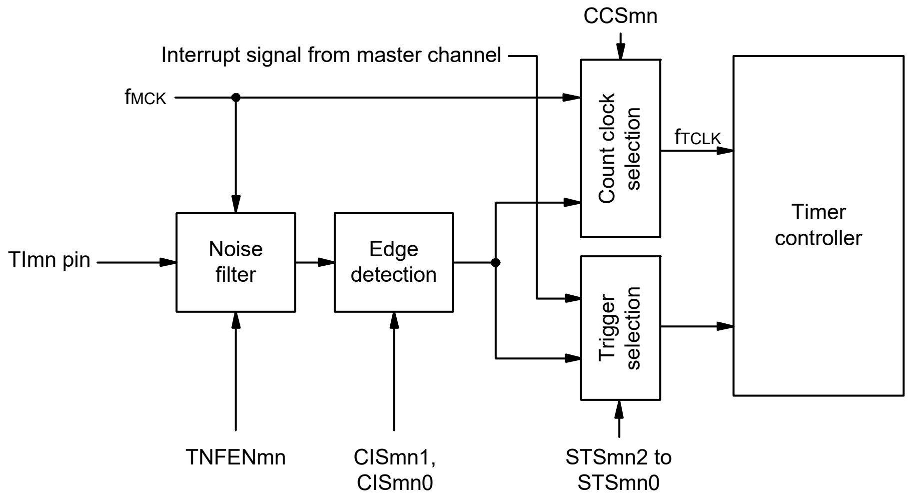
7.7 Timer Input (TImn) Control

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.

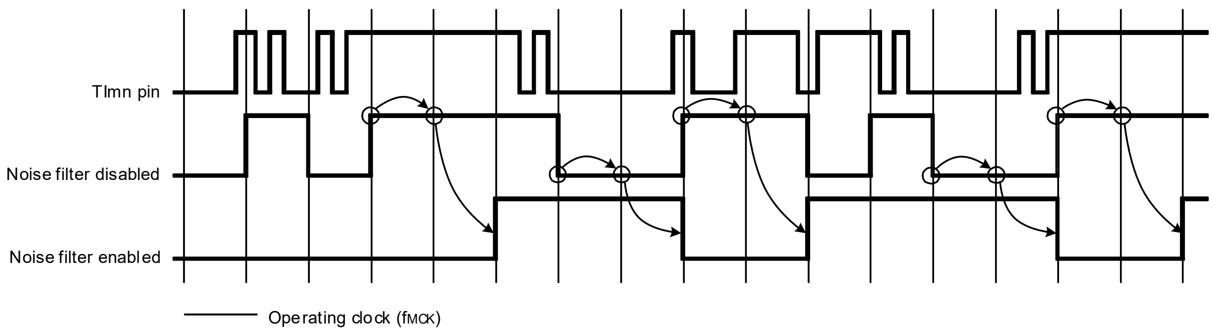
Figure 7 - 43 Input Circuit Configuration



7.7.2 Noise filter

When the noise filter is disabled, the input signal is only synchronized with the operating clock (f_{MCK}) for channel n. When the noise filter is enabled, after synchronization with the operating clock (f_{MCK}) 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 - 44 Sampling Waveforms through TImn Input Pin with Noise Filter Enabled and Disabled



Caution The input waveforms to the TImn pin are shown to explain the operation when the noise filter is enabled or disabled. When actually inputting waveforms, input them according to the TImn input high-level and low-level widths listed in 34.4 AC Characteristics.

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 bits 12 (CCSmn), 9 (STSmn1), and 8 (STSmn0) in the timer mode register mn (TMRmn) are 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 bits 12 (CCSmn), 9 (STSmn1), and 8 (STSmn0) 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 interval timer/square wave output

(1) Interval timer

The timer array unit can be used as a reference timer that generates INTTMmn (timer interrupt) at fixed intervals.

The interrupt generation period can be calculated by the following expression.

Generation period of INTTMmn (timer interrupt) = Period of count clock × (Set value of TDRmn + 1)

(2) Operation as square wave output

TOmn performs a toggle operation as soon as INTTMmn has been generated, and outputs a square wave with a duty factor of 50%.

The period and frequency for outputting a square wave from TOmn can be calculated by the following expressions.

• Period of square wave output from TOmn = Period of count clock × (Set value of TDRmn + 1) × 2
• Frequency of square wave output from TOmn = Frequency of count clock/{(Set value of TDRmn + 1) × 2}

Timer count register mn (TCRmn) operates as a down counter in the interval timer mode.

The TCRmn register loads the value of timer data register mn (TDRmn) at the first count clock after the channel start trigger bit (TSmn, TSHm1, TSHm3) of timer channel start register m (TSm) is set to 1. If the MDmn0 bit of timer mode register mn (TMRmn) is 0 at this time, INTTMmn is not output and TOmn is not toggled. If the MDmn0 bit of the TMRmn register is 1, INTTMmn is output and TOmn is toggled.

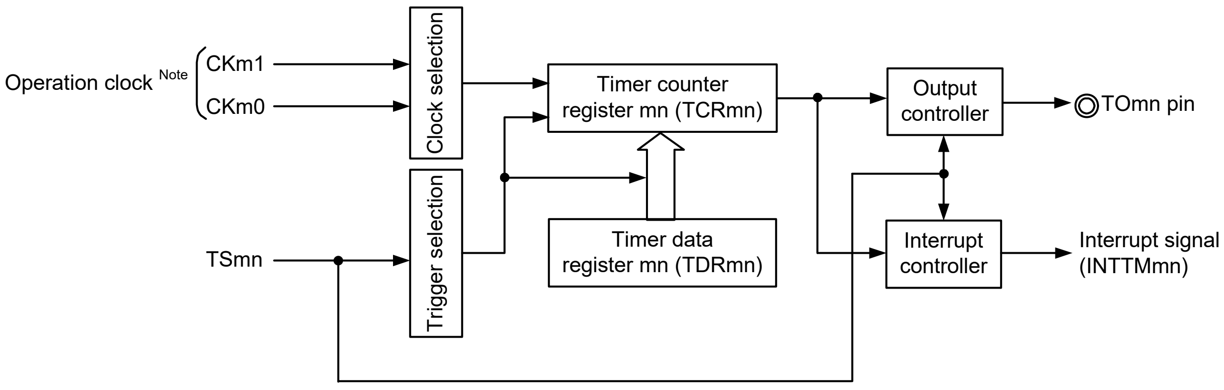
After that, the TCRmn register count down in synchronization with the count clock.

When TCRmn = 0000H, INTTMmn is output and TOmn is toggled at the next count clock. At the same time, the TCRmn register loads the value of the TDRmn register again. After that, the same operation is repeated.

The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid from the next period.

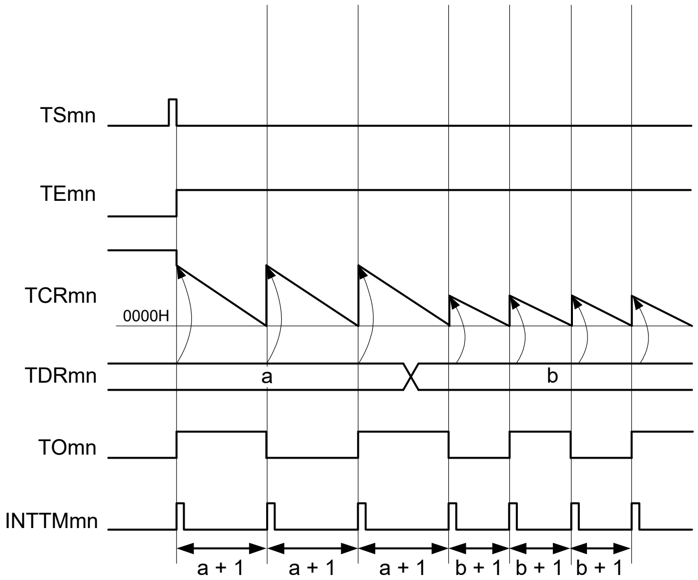
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 7 - 45 Block Diagram of Operation as Interval Timer/Square Wave Output



Note When channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Figure 7 - 46 Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MDmn0 = 1)



Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

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 count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)
TOmn: TOmn pin output signal

Figure 7 - 47 Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (1/2)

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	M/S Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	1/0	0	0	0/1	0	0	0	0	0	0	0	0	0	0	1/0

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
0: Neither generates INTTMmn nor inverts timer output when counting is started.
1: Generates INTTMmn and inverts timer output when counting is started.

Selection of TImn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Setting of MASTERmn bit (channel 2)
0: Independent channel operation function.
Setting of SPLITmn bit (channels 1, 3)
0: 16-bit timer mode
1: 8-bit timer mode

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected using channels 1 and 3 in the 8-bit timer mode).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected using channels 1 and 3 in the 8-bit timer mode).

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	1/0

0: Outputs 0 from TOmn.
1: Outputs 1 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	1/0

0: Stops the TOmn output operation by counting operation.
1: Enables the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn
	0

0: Cleared to 0 when TOMmn = 0 (master channel output mode)

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn
	0

0: Sets master channel output mode.

Note TMRm2: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 7 - 48 Operation Procedure of Interval Timer/Square Wave Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit m is stopped (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Input clock supply for timer array unit m is supplied (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets timer mode register mn (TMRmn) (determines operation mode of channel). Sets interval (period) value to timer data register mn (TDRmn).	Channel stops operating. (Clock is supplied and some power is consumed.)
	To use the TOmn output Clears the TOMmn bit of timer output mode register m (TOMm) to 0 (master channel output mode). Clears the TOLmn bit to 0. Sets the TOmn bit and determines default level of the TOmn output. → Sets the TOEmn bit to 1 and enables operation of TOmn. → Clears the port register and port mode register to 0. →	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. TOmn does not change because channel stops operating. The TOmn pin outputs the TOmn set level.
Operation start	(Sets the TOEmn bit to 1 only if using TOmn output and resuming operation.). Sets the TSmn (TSHm1, TSHm3) bit to 1. → The TSmn (TSHm1, TSHm3) bit automatically returns to 0 because it is a trigger bit.	TEmn (TEHm1, TEHm3) = 1, and count operation starts. Value of the TDRmn register is loaded to timer count register mn (TCRmn). INTTMmn is generated and TOmn performs toggle operation if the MDmn0 bit of the TMRmn register is 1.
During operation	Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TOm and TOEm registers can be changed. Set values of the TMRmn register, TOMmn, and TOLmn bits cannot be changed.	Counter (TCRmn) counts down. When count value reaches 0000H, the value of the TDRmn register is loaded to the TCRmn register again and the count operation is continued. By detecting TCRmn = 0000H, INTTMmn is generated and TOmn performs toggle operation. After that, the above operation is repeated.
Operation stop	The TTmn (TTHm1, TTHm3) bit is set to 1. → The TTmn (TTHm1, TTHm3) bit automatically returns to 0 because it is a trigger bit.	TEmn (TEHm1, TEHm3), and count operation stops. The TCRmn register holds count value and stops. The TOmn output is not initialized but holds current status.
	The TOEmn bit is cleared to 0 and value is set to the TOmn bit. →	The TOmn pin outputs the TOmn bit set level.

Operation is resumed.

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

Figure 7 - 49 Operation Procedure of Interval Timer/Square Wave Output Function (2/2)

	Software Operation	Hardware Status
TAU stop	To hold the TOmn pin output level Clears the TOmn bit to 0 after the value to be held is set to the port register. →	The TOmn pin output level is held by port function.
	When holding the TOmn pin output level is not necessary Setting not required.	
	The TAUmEN bit of the PER0 register is cleared to 0. →	Input clock supply for timer array unit m is stopped
	To initialize all circuits, set the TAU0RES bit in the PRR0 register to 1. →	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), n: Channel number (n = 0 to 3)

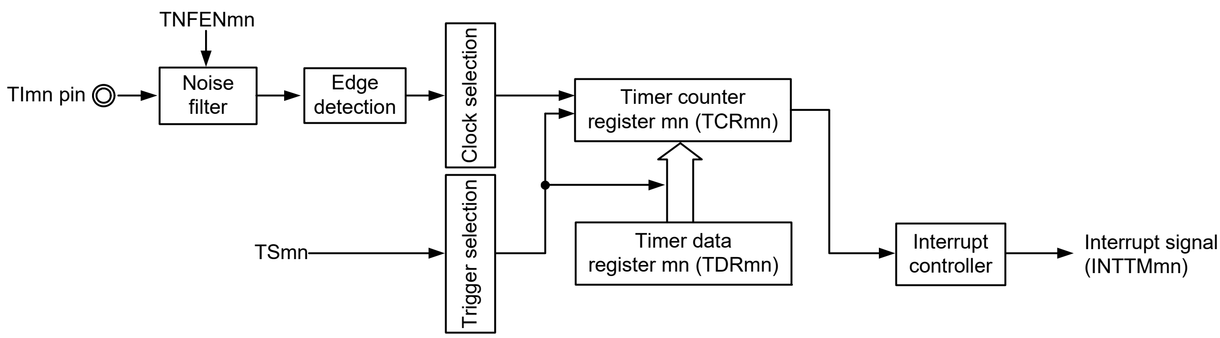
7.8.2 Operation as external event counter

The timer array unit can be used as an external event counter that counts the number of times the valid input edge (external event) is detected in the TImn pin. When a specified count value is reached, the event counter generates an interrupt. The specified number of counts can be calculated by the following expression.

Specified number of counts = Set value of TDRmn + 1

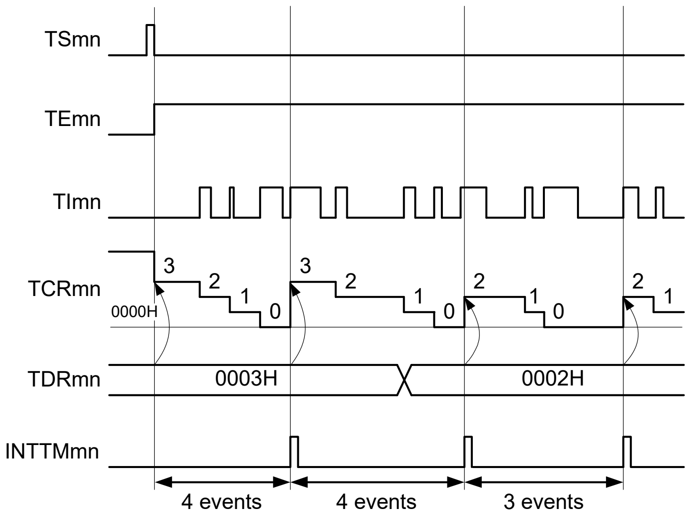
Timer count register mn (TCRmn) operates as a down counter in the event counter mode.
The TCRmn register loads the value of timer data register mn (TDRmn) by setting any channel start trigger bit (TSmn, TSHm1, TSHm3) of timer channel start register m (TSm) to 1.
The TCRmn register counts down each time the valid input edge of the TImn pin has been detected. When TCRmn = 0000H, the TCRmn register loads the value of the TDRmn register again, and outputs INTTMmn.
After that, the above operation is repeated.
An irregular waveform that depends on external events is output from the TOmn pin. Stop the output by setting the TOEmn bit of timer output enable register m (TOEm) to 0.
The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid during the next count period.

Figure 7 - 50 Block Diagram of Operation as External Event Counter



Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 7 - 51 Example of Basic Timing of Operation as External Event Counter



Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

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 count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)

Figure 7 - 52 Example of Set Contents of Registers in External Event Counter Mode (1/2)

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	M/S Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	1/0	0	1	0/1	0	0	0	1/0	1/0	0	0	0	1	1	0

Operation mode of channel n
011B Event count mode

Setting of operation when counting is started
0: Neither generates INTTMmn nor inverts timer output when counting is started.

Selection of TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
000B: Selects only software start.

Setting of MASTERmn bit (channel 2)
0: Independent channel operation function.
Setting of SPLITmn bit (channels 1, 3)
0: 16-bit timer mode
1: 8-bit timer mode

Count clock selection
1: Selects the TImn pin input valid edge.

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected using channels 1 and 3 in the 8-bit timer mode).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected using channels 1 and 3 in the 8-bit timer mode).

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	0

0: Outputs 0 from TOmn

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	0

0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn
	0

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn
	0

0: Sets master channel output mode.

Note TMRm2: MASTERmn bit
TMRm1, TMRm3 SPLITmn bit
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 7 - 53 Operation Procedure When External Event Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit m is stopped (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Input clock supply for timer array unit m is supplied. 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 register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). Sets number of counts to timer data register mn (TDRmn). Clears the TOEmn bit of timer output enable register m (TOEm) to 0.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and count operation starts. Value of the TDRmn register is loaded to timer count register mn (TCRmn) and detection of the TImn pin input edge is awaited.
During operation	Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TMRmn register, TOMmn, TOLmn, TOmn, and TOEmn bits cannot be changed.	Counter (TCRmn) counts down each time input edge of the TImn pin has been detected. When count value reaches 0000H, the value of the TDRmn register is loaded to the TCRmn register again, and the count operation is continued. By detecting TCRmn = 0000H, the INTTMmn output is generated. After that, the above operation is repeated.
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0. → To initialize all circuits, set the TAU0RES bit in the PRR0 register to 1. →	Input clock supply for timer array unit m is stopped All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

7.8.3 Operation as frequency divider (channel 0 of unit 0 only)

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 count register 00 (TCR00) operates as a down counter in the interval timer mode.
After the channel start trigger bit (TS00) of timer channel start register 0 (TS0) is set to 1, the TCR00 register loads the value of timer data register 00 (TDR00) when the TI00 valid edge is detected.
If the MD000 bit of timer mode register 00 (TMR00) is 0 at this time, INTTM00 is not output and TO00 is not toggled. If the MD000 bit of timer mode register 00 (TMR00) is 1, INTTM00 is output and TO00 is toggled.
After that, the TCR00 register counts down at the valid edge of the TI00 pin. When TCR00 = 0000H, it toggles TO00. At the same time, the TCR00 register loads the value of the TDR00 register again, and continues counting.
If detection of both the edges of the TI00 pin is selected, the duty factor error of the input clock affects the divided clock period of the TO00 output.
The period of the TO00 output clock includes a sampling error of one period of the operation clock.

Clock period of TO00 output = Ideal TO00 output clock period ± Operation clock period (error)

The TDR00 register can be rewritten at any time. The new value of the TDR00 register becomes valid during the next count period.

Figure 7 - 54 Block Diagram of Operation as Frequency Divider

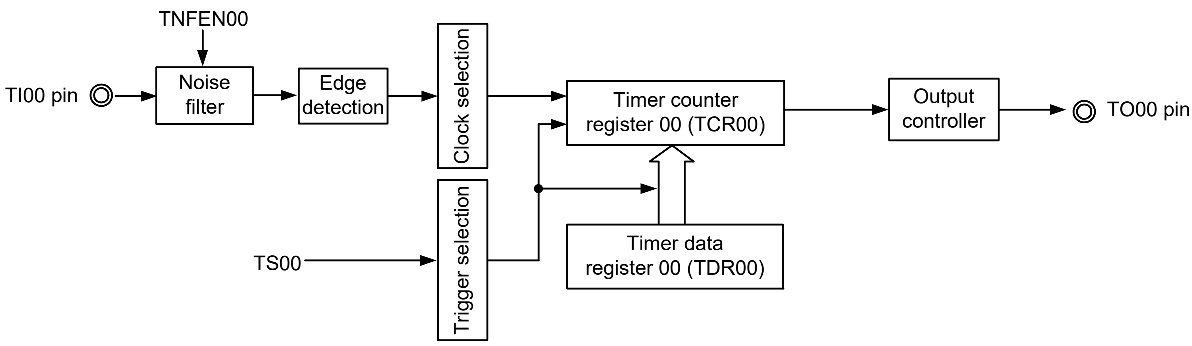
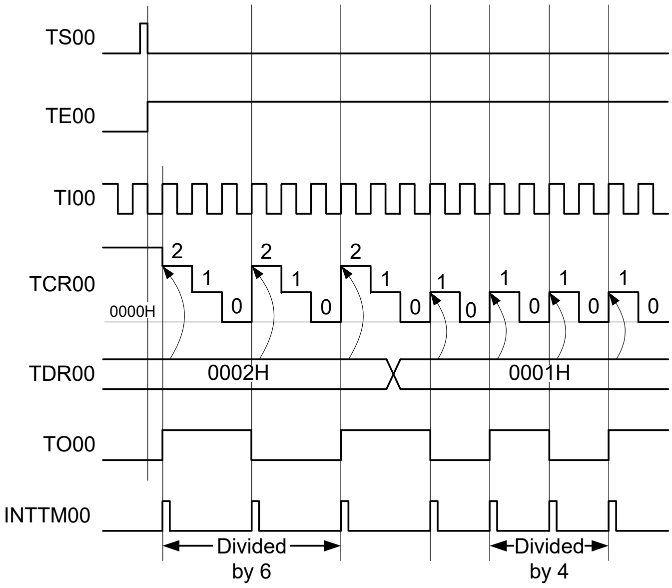


Figure 7 - 55 Example of Basic Timing of Operation as Frequency Divider (MD000 = 1)



Remark TS00: Bit n of timer channel start register 0 (TS0)

TE00: Bit n of timer channel enable status register 0 (TE0)

TI00: TI00 pin input signal

TCR00: Timer count register 00 (TCR00)

TDR00: Timer data register 00 (TDR00)

TO00: TO00 pin output signal

Figure 7 - 56 Example of Set Contents of Registers During Operation as Frequency Divider

(a) Timer mode register 00 (TMR00)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRm0	CKSmn1	CKSmn0		CCS00		STS002	STS001	STS000	CIS001	CIS000			MD003	MD002	MD001	MD000
	1/0	0	0	1	0	0	0	0	1/0	1/0	0	0	0	0	0	1/0

Operation mode of channel 0
000B: Interval timer

Setting of operation when counting is started
0: Neither generates INTTM00 nor inverts timer output when counting is started.
1: Generates INTTM00 and inverts timer output when counting is started.

Selection of TI00 pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
000B: Selects only software start.

Count clock selection
1: Selects the TI00 pin input valid edge.

Operation clock (fMCK) selection
00B: Selects CK00 as operation clock of channel 0.
10B: Selects CK01 as operation clock of channel 0.

(b) Timer output register 0 (TO0)

	Bit 0
TO0	TOmn
	0

0: Outputs 0 from TO00.
1: Outputs 1 from TO00.

(c) Timer output enable register 0 (TOE0)

	Bit n
TOE0	TOEmn
	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	TOLmn
	0

0: Cleared to 0 when master channel output mode (TOM00 = 0)

(e) Timer output mode register 0 (TOM0)

	Bit 0
TOM0	TOMmn
	0

0: Sets master channel output mode.

Figure 7 - 57 Operation Procedure When Frequency Divider Function Is Used

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit 0 is stopped (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Input clock supply for timer array unit 0 is supplied. 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 CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register 00 (TMR00) (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.)
	Clears the TOM00 bit of timer output mode register 0 (TOM0) to 0 (master channel output mode). Clears 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.
	Clears the port register and port mode register to 0. →	The TO00 pin outputs the TO00 set level.
Operation start	Sets the TOE00 bit to 1 (only when operation is resumed). → Sets the TS00 bit to 1. The TS00 bit automatically returns to 0 because it is a trigger bit.	TE00 = 1, and count operation starts. Value of the TDR00 register is loaded to timer count 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 because it is a trigger bit.	TE00 = 0, and count operation stops. The TCR00 register holds count value and stops. The TO00 output is not initialized but holds current status.
	The TOE00 bit is cleared to 0 and value is set to the TO00 bit. →	The TO00 pin outputs the TO00 set level.
TAU stop	To hold the TO00 pin output level Clears 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 cleared to 0. →	Input clock supply for timer array unit 0 is stopped
	To initialize all circuits, set the TAU0RES bit in the PRR0 register to 1. →	All circuits are initialized and SFR of each channel is also initialized. (The TO00 bit is cleared to 0 and the TO00 pin is set to port mode).

Operation is resumed.

7.8.4 Operation as input pulse interval measurement

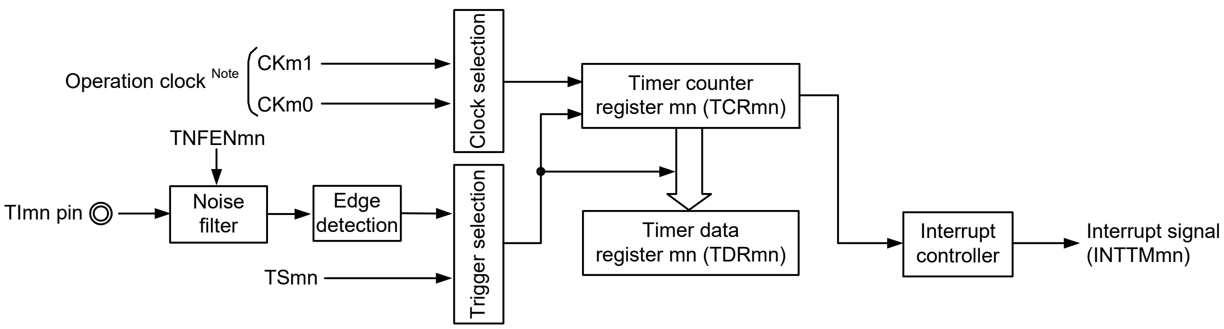
The count value can be captured at the TImn valid edge and the interval of the pulse input to TImn can be measured. In addition, the count value can be captured by using software operation (TSmn = 1) as a capture trigger while the TEmn bit is set to 1.
The pulse interval can be calculated by the following expression.

TImn input pulse interval = Period of count clock × ((10000H × TSRmn: OVF) + (Capture value of TDRmn + 1))

Caution The TImn pin input is sampled using the operating clock selected with the CKSmn bit of timer mode register mn (TMRmn), so an error of up to one operating clock cycle occurs.

Timer count register mn (TCRmn) operates as an up counter in the capture mode.
When the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, the TCRmn register counts up from 0000H in synchronization with the count clock.
When the TImn pin input valid edge is detected, the count value of the TCRmn register is transferred (captured) to timer data register mn (TDRmn) and, at the same time, the TCRmn register is cleared to 0000H, and the INTTMmn is output. If the counter overflows at this time, the OVF bit of timer status register mn (TSRmn) is set to 1. If the counter does not overflow, the OVF bit is cleared. After that, the above operation is repeated.
As soon as the count value has been captured to the TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.
If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSRmn register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.
Set the STSmn2 to STSmn0 bits of the TMRmn register to 001B to use the valid edges of TImn as a start trigger and a capture trigger.

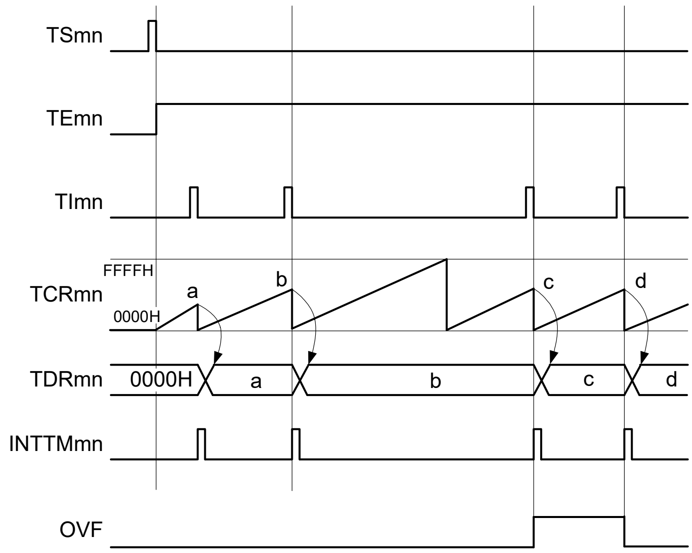
Figure 7 - 58 Block Diagram of Operation as Input Pulse Interval Measurement



Note When channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 7 - 59 Example of Basic Timing of Operation as Input Pulse Interval Measurement (MDmn0 = 0)



Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

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 count register mn (TCRmn)

TDRmn: Timer data register mn (TDRmn)

OVF: Bit 0 of timer status register mn (TSRmn)

Figure 7 - 60 Example of Set Contents of Registers to Measure Input Pulse Interval

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	M/S Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	0	0	0	0	1	1/0	1/0	0	0	0	1	0	1/0

Operation mode of channel n
010B: Capture mode

Setting of operation when counting is started
- 0: Does not generate INTTMmn when counting is started.
- 1: Generates INTTMmn when counting is started.

Selection of TImn pin input edge
- 00B: Detects falling edge.
- 01B: Detects rising edge.
- 10B: Detects both edges.
- 11B: Setting prohibited

Capture trigger selection
- 001B: Selects the TImn pin input valid edge.

Setting of MASTERmn bit (channel 2)
- 0: Independent channel operation function.

Setting of SPLITmn bit (channels 1, 3)
- 0: 16-bit timer mode

Count clock selection
- 0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
- 00B: Selects CKm0 as operation clock of channel n.
- 10B: Selects CKm1 as operation clock of channel n.
- 01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).
- 11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).

(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: Cleared to 0 when master channel output mode (TOMmn = 0)

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn
	0

0: Sets master channel output mode.

Note TMRm2: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 7 - 61 Operation Procedure When Input Pulse Interval Measurement Function Is Used

Operation is resumed.

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit m is stopped (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1.	Input clock supply for timer array unit m is supplied. 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 register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel).	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets TSmn bit to 1. The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and count operation starts. Timer count register mn (TCRmn) is cleared to 0000H. 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 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops. The OVF bit of the TSRmn register is also held.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0. To initialize all circuits, set the TAU0RES bit in the PRR0 register to 1.	Input clock supply for timer array unit m is stopped All circuits are initialized and SFR of each channel is also initialized.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

7.8.5 Operation as input signal high-/low-level width measurement

By starting counting at one edge of the TImn pin input and capturing the number of counts at another edge, the signal width (high-level width/low-level width) of TImn can be measured. The signal width of TImn can be calculated by the following expression.

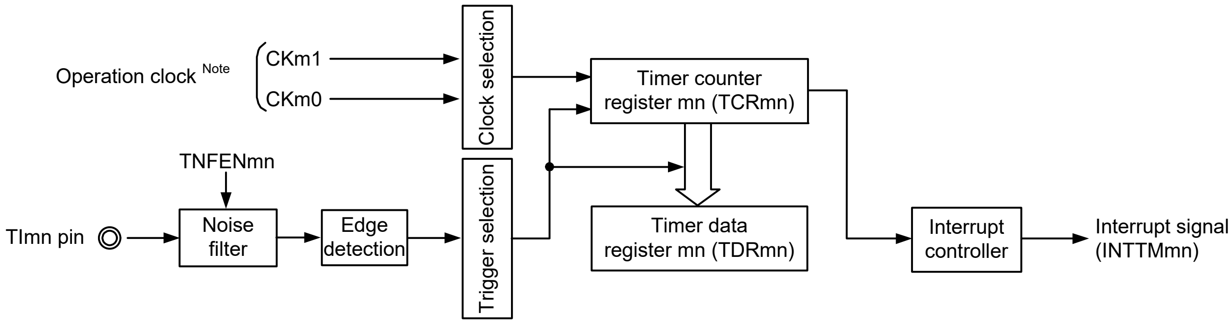
Signal width of TImn input = Period of count clock × ((10000H × TSRmn: OVF) + (Capture value of TDRmn + 1))

Caution The TImn pin input is sampled using the operating clock selected with the CKSmn bit of timer mode register mn (TMRmn), so an error equivalent to one operation clock occurs.

Timer count register mn (TCRmn) operates as an up counter in the capture & one-count mode.
When the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, the TEmn bit is set to 1 and the TImn pin start edge detection wait status is set.
When the TImn pin input start edge (rising edge of the TImn pin input when the high-level width is to be measured) is detected, the counter counts up from 0000H in synchronization with the count clock. When the valid capture edge (falling edge of the TImn pin input when the high-level width is to be measured) is detected later, the count value is transferred to timer data register mn (TDRmn) and, at the same time, INTTMmn is output. If the counter overflows at this time, the OVF bit of timer status register mn (TSRmn) is set to 1. If the counter does not overflow, the OVF bit is cleared. The TCRmn register stops at the value "value transferred to the TDRmn register + 1", and the TImn pin start edge detection wait status is set. After that, the above operation is repeated.
As soon as the count value has been captured to the TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.
If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSRmn register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.
Whether the high-level width or low-level width of the TImn pin is to be measured can be selected by using the CISmn1 and CISmn0 bits of the TMRmn register.
Because this function is used to measure the signal width of the TImn pin input, the TSmn bit cannot be set to 1 while the TEmn bit is 1.

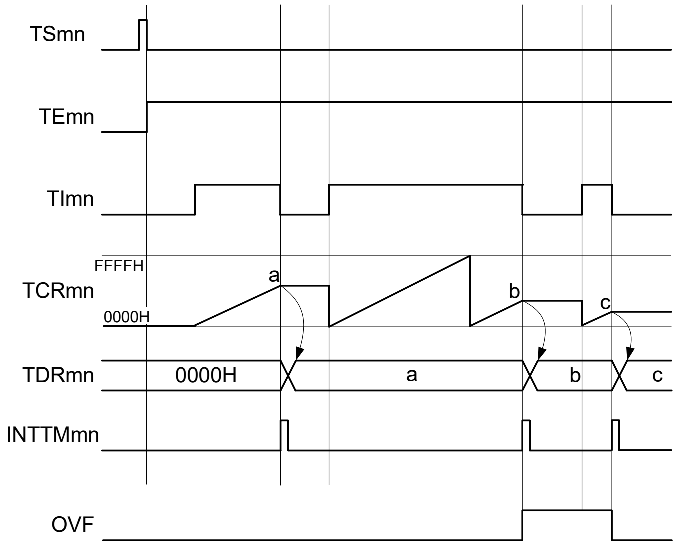
CISmn1, CISmn0 of TMRmn register = 10B: Low-level width is measured.
CISmn1, CISmn0 of TMRmn register = 11B: High-level width is measured.

Figure 7 - 62 Block Diagram of Operation as Input Signal High-/Low-Level Width Measurement



Note For channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Figure 7 - 63 Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement



Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

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 count register mn (TCRmn)

TDRmn: Timer data register mn (TDRmn)

OVF: Bit 0 of timer status register mn (TSRmn)

Figure 7 - 64 Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	M/S Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	0	0	0	1	0	1	1/0	0	0	1	1	0	0

Operation mode of channel n
110B: Capture & one-count

Setting of operation when counting is started
0: Does not generate INTTMmn when counting is started.

Selection of TImn pin input edge
10B: Both edges (to measure low-level width)
11B: Both edges (to measure high-level width)

Start trigger selection
010B: Selects the TImn pin input valid edge.

Setting of MASTERmn bit (channel 2)
0: Independent channel operation function.
Setting of SPLITmn bit (channels 1, 3)
0: 16-bit timer mode

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	0

0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn
	0

0: Cleared to 0 when master channel output mode (TOMmn = 0)

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn
	0

0: Sets master channel output mode.

Note TMRm2: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 7 - 65 Operation Procedure When Input Signal High-/Low-Level Width Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit m is stopped (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Input clock supply for timer array unit m is supplied. 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 register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). Clears the TOEmn bit to 0 and stops operation of TOmn.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and the TImn pin start edge detection wait status is set.
	Detects the TImn pin input count start valid edge. →	Clears timer count register mn (TCRmn) to 0000H and starts counting up.
During operation	Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TMRmn register, TOMmn, TOLmn, TOmn, and TOEmn bits cannot be changed.	When the TImn pin start edge is detected, the counter (TCRmn) counts up from 0000H. If a capture edge of the TImn pin is detected, the count value is transferred to timer data register mn (TDRmn) and INTTMmn is generated. If an overflow occurs at this time, the OVF bit of timer status register mn (TSRmn) is set; if an overflow does not occur, the OVF bit is cleared. The TCRmn register stops the count operation until the next TImn pin start edge is detected.
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops. The OVF bit of the TSRmn register is also held.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0. → To initialize all circuits, set the TAU0RES bit in the PRR0 register to 1. →	Input clock supply for timer array unit m is stopped All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

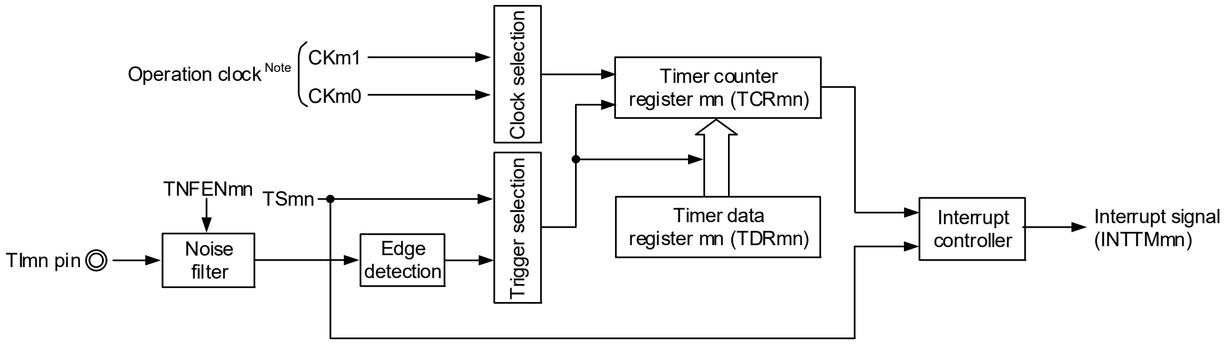
7.8.6 Operation as delay counter

It is possible to start counting down when the valid edge of the TImn pin input is detected (an external event), and then generate INTTMmn (a timer interrupt) after any specified interval.
It can also generate INTTMmn (timer interrupt) at any interval by making a software set TSmn = 1 and the count down start during the period of TEmn = 1.
The interrupt generation period can be calculated by the following expression.

Generation period of INTTMmn (timer interrupt) = Period of count clock × (Set value of TDRmn + 1)

Timer count register mn (TCRmn) operates as a down counter in the one-count mode.
When the channel start trigger bit (TSmn, TSHm1, TSHm3) of timer channel start register m (TSm) is set to 1, the TEmn, TEHm1, TEHm3 bits are set to 1 and the TImn pin input valid edge detection wait status is set.
Timer count register mn (TCRmn) starts operating upon TImn pin input valid edge detection and loads the value of timer data register mn (TDRmn). The TCRmn register counts down from the value of the TDRmn register it has loaded, in synchronization with the count clock. When TCRmn = 0000H, it outputs INTTMmn and stops counting until the next TImn pin input valid edge is detected.
The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid from the next period.

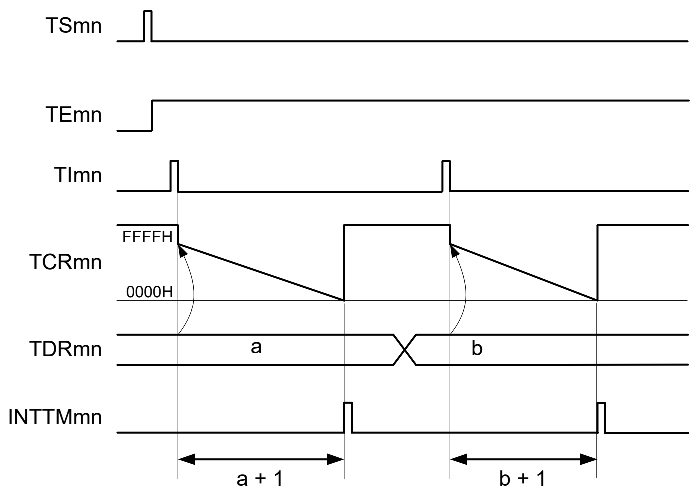
Figure 7 - 66 Block Diagram of Operation as Delay Counter



Note For using channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 7 - 67 Example of Basic Timing of Operation as Delay Counter



Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

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 count register mn (TCRmn)

TDRmn: Timer data register mn (TDRmn)

Figure 7 - 68 Example of Set Contents of Registers to Delay Counter

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	M/S Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	1/0	0	0	0/1	0	0	1	1/0	1/0	0	0	1	0	0	1/0

Operation mode of channel n
100B: One-count mode

Start trigger during operation
0: Trigger input is invalid.
1: Trigger input is valid.

Selection of TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
001B: Selects the TImn pin input valid edge.

Setting of MASTERmn bit (channel 2)
0: Independent channel operation function.
Setting of SPLITmn bit (channels 1, 3)
0: 16-bit timer mode
1: 8-bit timer mode

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected using channels 1 and 3 in the 8-bit timer mode).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected using channels 1 and 3 in the 8-bit timer mode).

(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when master channel output mode (TOMmn = 0)

(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets master channel output mode.

(f)

Note TMRm2: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 7 - 69 Operation Procedure When Delay Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit m is stopped (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1.	Input clock supply for timer array unit m is supplied. 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 register 1 (NFEN1) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). INTTMmn output delay is set to timer data register mn (TDRmn). Clears the TOEmn bit to 0 and stops operation of TOmn.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and the start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit is set to 1) wait status 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 count register mn (TCRmn).
During operation	Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used.	The counter (TCRmn) counts down. When 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 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0. To initialize all circuits, set the TAU0RES bit in the PRR0 register to 1.	Input clock supply for timer array unit m is stopped All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

7.9 Simultaneous Channel Operation Function of Timer Array Unit

7.9.1 Operation as one-shot pulse output function

By using two channels as a set, a one-shot pulse having any delay pulse width can be generated from the signal input to the TImn pin.

The delay time and pulse width can be calculated by the following expressions.

Delay time = {Set value of TDRmn (master) + 2} × Count clock period Pulse width = {Set value of TDRmp (slave)} × Count clock period

The master channel operates in the one-count mode and counts the delays. Timer count register mn (TCRmn) of the master channel starts operating upon start trigger detection and loads the value of timer data register mn (TDRmn).

The TCRmn register counts down from the value of the TDRmn register it has loaded, in synchronization with the count clock. When TCRmn = 0000H, it outputs INTTMmn and stops counting until the next start trigger is detected.

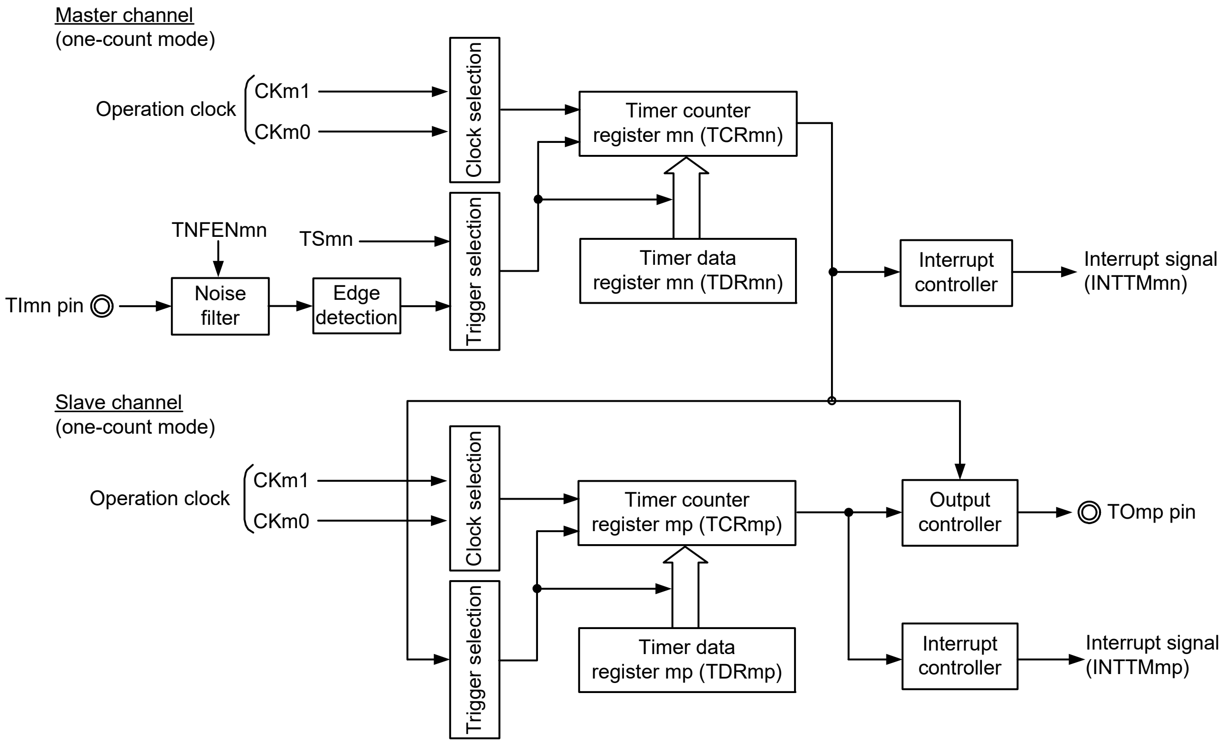
The slave channel operates in the one-count mode and counts the pulse width. The TCRmp register of the slave channel starts operation using INTTMmn of the master channel as a start trigger, and loads the value of the TDRmp register. The TCRmp register counts down from the value of The TDRmp register it has loaded, in synchronization with the count value. When count value = 0000H, it outputs INTTMmp and stops counting until the next start trigger (INTTMmn of the master channel) is detected. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.

Instead of using the TImn pin input, a one-shot pulse can also be output using the software operation (TSmn = 1) as a start trigger.

Caution Since the timing for loading of the TDRmn register of the master channel will be different from that for loading of the TDRmp register of the slave channel, writing to the TDRmn or TDRmp register while counting is in progress may lead to contention that causes an illegal waveform to be output. Only write new values to the TDRmn register after INTTMmn has been generated and to the TDRmp register after INTTMmp has been generated.

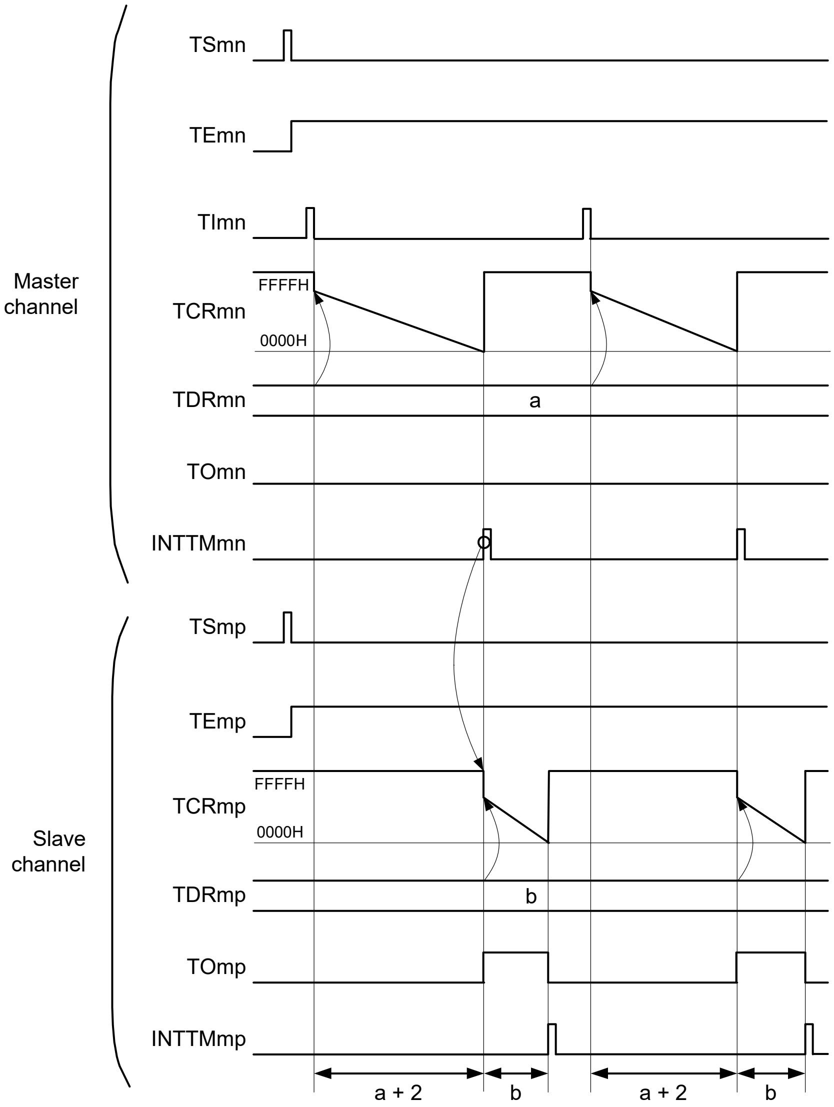
Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n = 0: p = 1, 2, 3, n = 2: p = 3)

Figure 7 - 70 Block Diagram of Operation as One-Shot Pulse Output Function



Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n = 0: p = 1, 2, 3, n = 2: p = 3)

Figure 7 - 71 Example of Basic Timing of Operation as One-Shot Pulse Output Function



Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n = 0: p = 1, 2, 3, n = 2: p = 3)

Remark 2. TSmn, TSmp: Bit n, p of timer channel start register m (TSm)
TEmn, TEmp: Bit n, p of timer channel enable status register m (TEm)
TImn, TImp: TImn and TImp pins input signal
TCRmn, TCRmp: Timer count registers mn, mp (TCRmn, TCRmp)
TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)
TOmn, TOmp: TOmn and TOmp pins output signal

Figure 7 - 72 Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Master Channel)

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1 1/0	CKSmn0 0	0	CCSmn 0	MASTERmn Note 1	STSmn2 0	STSmn1 0	STSmn0 1	CISmn1 1/0	CISmn0 1/0	0	0	MDmn3 1	MDmn2 0	MDmn1 0	MDmn0 0

Operation mode of channel n
100B: One-count mode

Start trigger during operation
0: Trigger input is invalid.

Selection of TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
001B: Selects the TImn pin input valid edge.

Setting of the MASTERmn bit (channel 2)
1: Master channel.

Count clock selection
0: Selects operation clock (fMCK).

Operation clock (fMCK) selection
00B: Selects CKm0 as operation clock of channels n.
10B: Selects CKm1 as operation clock of channels n.

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn 0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn 0

0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn 0

0: Cleared to 0 when master channel output mode (TOMmn = 0)

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn 0

0: Sets master channel output mode.

Note TMRm2: MASTERmn = 1
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)

Figure 7 - 73 Example of Set Contents of Registers
When One-Shot Pulse Output Function Is Used (Slave Channel)

(a) Timer mode register mp (TMRmp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmp	CKSmp1	CKSmp0		CCSmp	M/S Note	STSmp2	STSmp1	STSmp0	CISmp1	CISmp0			MDmp3	MDmp2	MDmp1	MDmp0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0

Operation mode of channel p
100B: One-count mode

Start trigger during operation
0: Trigger input is invalid.

Selection of TImp pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMmn of master channel.

Setting of MASTERmn bit (channel 2)
0: Slave channel
Setting of SPLITmp bit (channels 1, 3)
0: 16-bit timer mode

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel p.
10B: Selects CKm1 as operation clock of channel p.
* Make the same setting as master channel.

(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: MASTERmp bit
TMRm1, TMRm3: SPLITmp bit

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n = 0: p = 1, 2, 3, n = 2: p = 3)

Figure 7 - 74 Operation Procedure of One-Shot Pulse Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit m is stopped (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable registers 0 (PER0) to 1. →	Input clock supply for timer array unit m is supplied. 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 (NFEN1) 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. → Clears 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.

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

Figure 7 - 75 Operation Procedure of One-Shot Pulse Output Function (2/2)

	Software Operation	Hardware Status
Operation start	Sets the TOEmp bit (slave) to 1 (only when operation is resumed). The TSmn (master) and TSmp (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. → The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	The TEmn and TEmp bits are set to 1 and the master channel enters the 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 status. 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 count 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 valid edge is input to the TImn pin. The slave channel, triggered by INTTMmn of the master channel, loads the value of the TDRmp register to the TCRmp register, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. → The TTmn (master) and TTmp (slave) bits are set to 1 at the same time.	TEmn, TEmp = 0, and count operation stops. The TCRmn and TCRmp registers hold count value and stop. The TOmp output is not initialized but holds current status.
	The TOEmp bit of slave channel is cleared to 0 and value is set to the TOmp bit. →	The TOmp pin outputs the TOmp set level.
TAU stop	To hold the TOmp pin output level Clears the TOmp bit to 0 after the value to be held is set to the port register. → When holding the TOmp pin output level is not necessary Setting not required.	The TOmp pin output level is held by port function.
	The TAUmEN bit of the PER0 register is cleared to 0. → To initialize all circuits, set the TAU0RES bit in the PRR0 register to 1. →	Input clock supply for timer array unit m is stopped All circuits are initialized and SFR of each channel is also initialized. (The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)

Operation is resumed.

Note Do not set the TSmn bit of the slave channel to 1.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n = 0: p = 1, 2, 3, n = 2: p = 3)

7.9.2 Operation as PWM function

Two channels can be used as a set to generate a pulse of any period and duty factor.
The period and duty factor of the output pulse can be calculated by the following expressions.

Pulse period = {Set value of TDRmn (master) + 1} × Count clock period
Duty factor [%] = {Set value of TDRmp (slave)}/{Set value of TDRmn (master) + 1} × 100
0% output: Set value of TDRmp (slave) = 0000H
100% output: Set value of TDRmp (slave) ≥ {Set value of TDRmn (master) + 1}

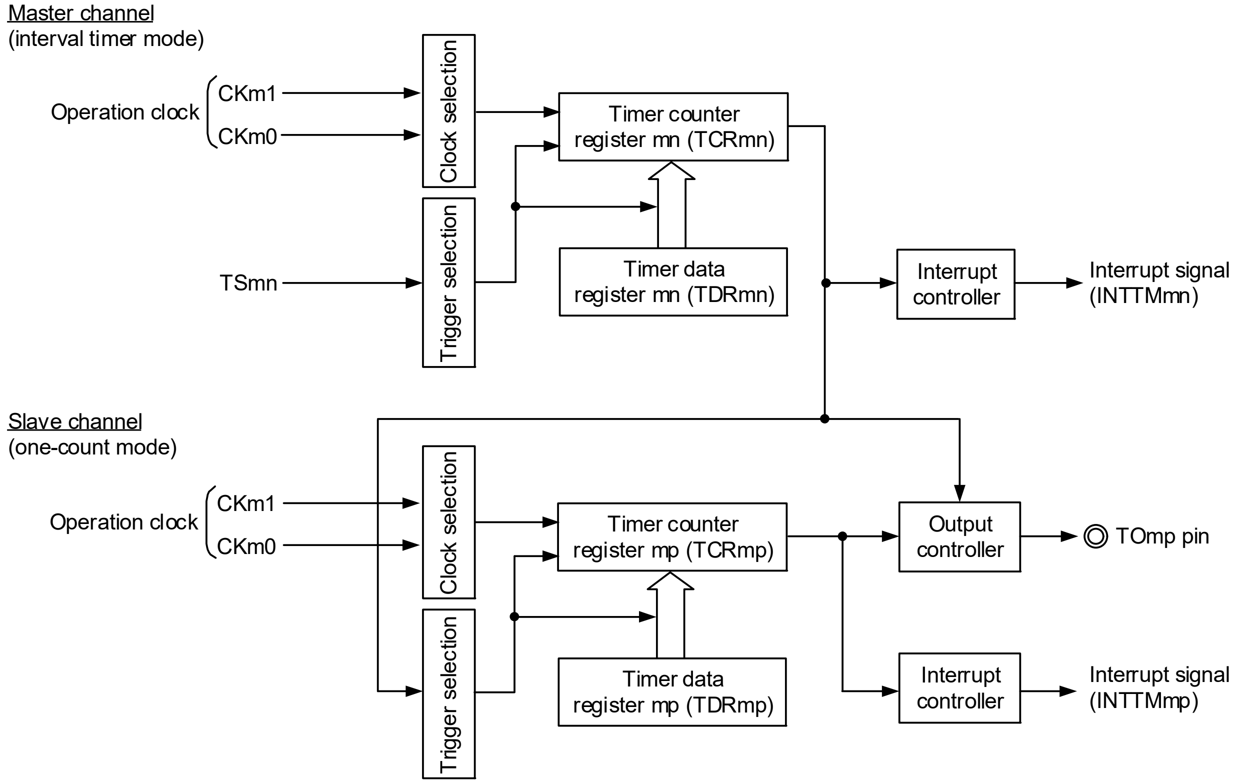
Remark The duty factor exceeds 100% if the set value of TDRmp (slave) > (set value of TDRmn (master) + 1), it summarizes to 100% output.

The master channel operates in the interval timer mode. If the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, an interrupt (INTTMmn) is output, the value set to timer data register mn (TDRmn) is loaded to timer count register mn (TCRmn), and the counter counts down in synchronization with the count clock. When the counter reaches 0000H, INTTMmn is output, the value of the TDRmn register is loaded again to the TCRmn register, and the counter counts down. This operation is repeated until the channel stop trigger bit (TTmn) of timer channel stop register m (TTm) is set to 1.
If two channels are used to output a PWM waveform, the period until the master channel counts down to 0000H is the PWM output (TOmp) cycle.
The slave channel operates in one-count mode. By using INTTMmn from the master channel as a start trigger, the TCRmp register loads the value of the TDRmp register and the counter counts down to 0000H. When the counter reaches 0000H, it outputs INTTMmp and waits until the next start trigger (INTTMmn from the master channel) is generated.
If two channels are used to output a PWM waveform, the period until the slave channel counts down to 0000H is the PWM output (TOmp) duty.
PWM output (TOmp) goes to the active level one clock after the master channel generates INTTMmn and goes to the inactive level when the TCRmp register of the slave channel becomes 0000H.

Caution To rewrite both timer data register mn (TDRmn) of the master channel and the TDRmp register of the slave channel, a write access is necessary two times. The timing at which the values of the TDRmn and TDRmp registers are loaded to the TCRmn and TCRmp registers is upon occurrence of INTTMmn of the master channel. Thus, when rewriting is performed split before and after occurrence of INTTMmn of the master channel, the TOmp pin cannot output the expected waveform. To rewrite both the TDRmn register of the master and the TDRmp register of the slave, therefore, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel.

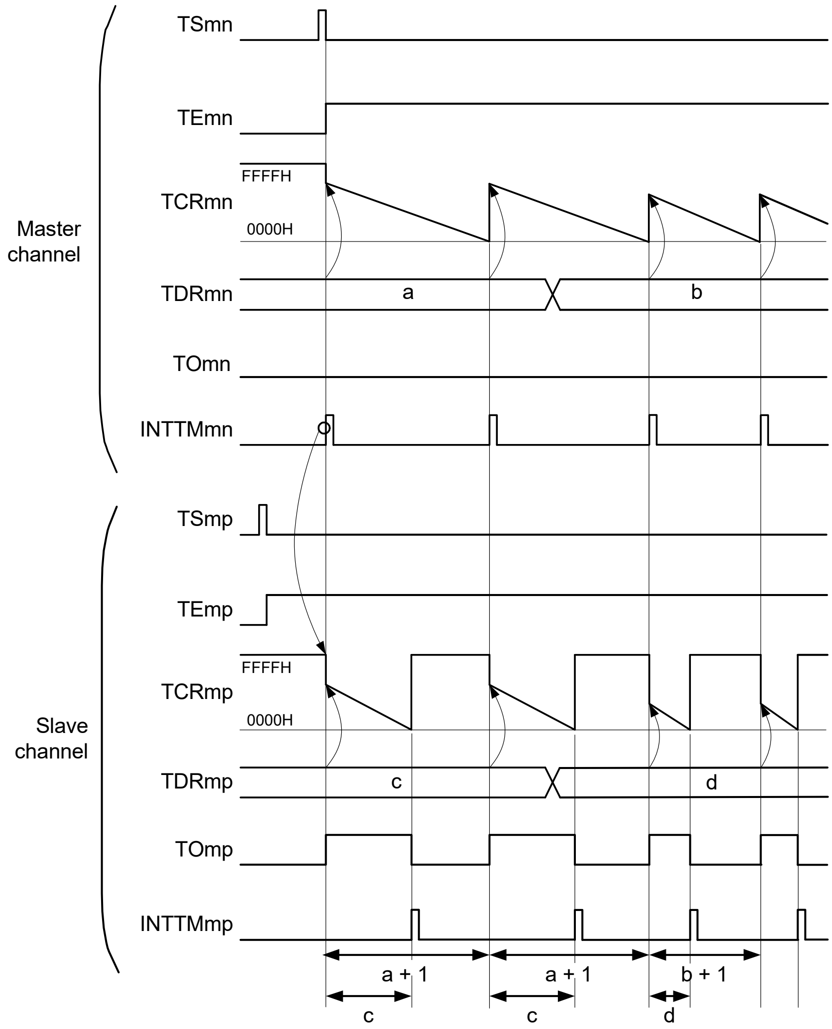
Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n = 0: p = 1, 2, 3, n = 2: p = 3)

Figure 7 - 76 Block Diagram of Operation as PWM Function



Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n = 0: p = 1, 2, 3, n = 2: p = 3)

Figure 7 - 77 Example of Basic Timing of Operation as PWM Function



Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n = 0: p = 1, 2, 3, n = 2: p = 3)

Remark 2. TSmn, TSmp: Bit n, p of timer channel start register m (TSm)
TEmn, TEmp: Bit n, p of timer channel enable status register m (TEm)
TCRmn, TCRmp: Timer count registers mn, mp (TCRmn, TCRmp)
TDRmn, TDRmp: Timer data registers mn, mp (TDRmn, TDRmp)
TOmn, TOmp: TOmn and TOmp pins output signal

Figure 7 - 78 Example of Set Contents of Registers When PWM Function (Master Channel) Is Used

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	MASTERmn Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
1: Generates INTTMmn when counting is started.

Selection of TImn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Setting of the MASTERmn bit (channel 2)
1: Master channel.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.

(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn 0	0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn 0	0: Cleared to 0 when master channel output mode (TOMmn = 0)

(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn 0	0: Sets master channel output mode.

Note TMRm2: MASTERmn = 1
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)

Figure 7 - 79 Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used

(a) Timer mode register mp (TMRmp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmp	CKSmp1	CKSmp0		CCSmp	M/S Note	STSmp2	STSmp1	STSmp0	CISmp1	CISmp0			MDmp3	MDmp2	MDmp1	MDmp0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

Operation mode of channel p
100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

Selection of TImp pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMmn of master channel.

Setting of MASTERmn bit (channel 2)
0: Slave channel
Setting of SPLITmp bit (channels 1, 3)
0: 16-bit timer mode

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel p.
10B: Selects CKm1 as operation clock of channel p.
* Make the same setting as master channel.

(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: MASTERmp bit
TMRm1, TMRm3: SPLITmp bit

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n = 0: p = 1, 2, 3, n = 2: p = 3)

Figure 7 - 80 Operation Procedure When PWM Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit m is stopped (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Input clock supply for timer array unit m is supplied. 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. → Sets the TOEmp bit to 1 and enables operation of TOmp. → Clears 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.

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

Figure 7 - 81 Operation Procedure When PWM Function Is Used (2/2)

	Software Operation	Hardware Status
Operation start	Sets the TOEmp bit (slave) to 1 (only when operation is resumed). The TSmn (master) and TSmp (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. → The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	TEmn = 1, TEmp = 1 When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	Set values of the TMRmn and TMRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. Set values of the TDRmn and TDRmp registers can be changed after INTTMmn of the master channel is generated. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used.	The counter of the master channel loads the TDRmn register value to timer count register mn (TCRmn), and counts down. When the count value reaches TCRmn = 0000H, INTTMmn output is generated. At the same time, the value of the TDRmn register is loaded to the TCRmn register, and the counter starts counting down again. At the slave channel, the value of the TDRmp register is loaded to the TCRmp register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. → The TTmn and TTmp bits automatically return to 0 because they are trigger bits.	TEmn, TEmp = 0, and count operation stops. The TCRmn and TCRmp registers hold count value and stop. The TOmp output is not initialized but holds current status.
	The TOEmp bit of slave channel is cleared to 0 and value is set to the TOmp bit. →	The TOmp pin outputs the TOmp set level.
TAU stop	To hold the TOmp pin output level Clears the TOmp bit to 0 after the value to be held is set to the port register. → When holding the TOmp pin output level is not necessary Setting not required.	The TOmp pin output level is held by port function.
	The TAUmEN bit of the PER0 register is cleared to 0. → To initialize all circuits, set the TAU0RES bit in the PRR0 register to 1. →	Input clock supply for timer array unit m is stopped All circuits are initialized and SFR of each channel is also initialized. (The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n = 0: p = 1, 2, 3, n = 2: p = 3)

7.9.3 Operation as multiple PWM output function

By extending the PWM function and using multiple slave channels, many PWM waveforms with different duty values can be output.
For example, when using two slave channels, the period and duty factor of an output pulse can be calculated by the following expressions.

Pulse period = {Set value of TDRmn (master) + 1} × Count clock period
Duty factor 1 [%] = {Set value of TDRmp (slave 1)}/{Set value of TDRmn (master) + 1} × 100
Duty factor 2 [%] = {Set value of TDRmq (slave 2)}/{Set value of TDRmn (master) + 1} × 100

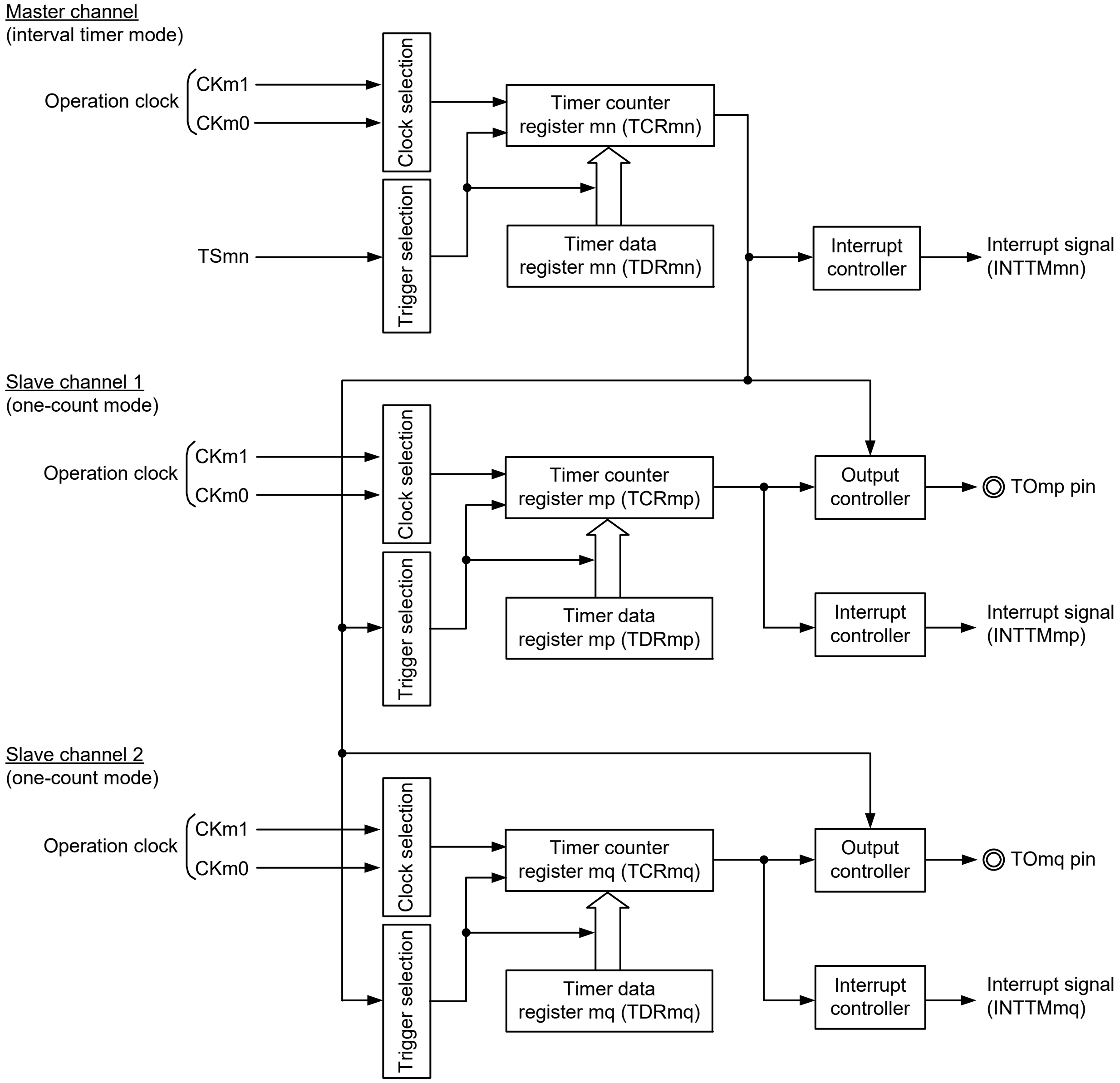
Remark Although the duty factor exceeds 100% if the set value of TDRmp (slave 1) > {set value of TDRmn (master) + 1} or if the {set value of TDRmq (slave 2)} > {set value of TDRmn (master) + 1}, it is summarized into 100% output.

Timer count register mn (TCRmn) of the master channel operates in the interval timer mode and counts the periods.
The TCRmp register of the slave channel 1 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmp pin. The TCRmp register loads the value of timer data register mp (TDRmp), using INTTMmn of the master channel as a start trigger, and starts counting down. When TCRmp = 0000H, TCRmp outputs INTTMmp and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.
In the same way as the TCRmp register of the slave channel 1, the TCRmq register of the slave channel 2 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmq pin. The TCRmq register loads the value of the TDRmq register, using INTTMmn of the master channel as a start trigger, and starts counting down. When TCRmq = 0000H, the TCRmq register outputs INTTMmq and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of TOmq becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmq = 0000H.
When channel 0 is used as the master channel as above, up to three types of PWM signals can be output at the same time.

Caution To rewrite both timer data register mn (TDRmn) of the master channel and the TDRmp register of the slave channel 1, write access is necessary at least twice. Since the values of the TDRmn and TDRmp registers are loaded to the TCRmn and TCRmp registers after INTTMmn is generated from the master channel, if rewriting is performed separately before and after generation of INTTMmn from the master channel, the TOmp pin cannot output the expected waveform. To rewrite both the TDRmn register of the master and the TDRmp register of the slave, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel (This applies also to the TDRmq register of the slave channel 2).

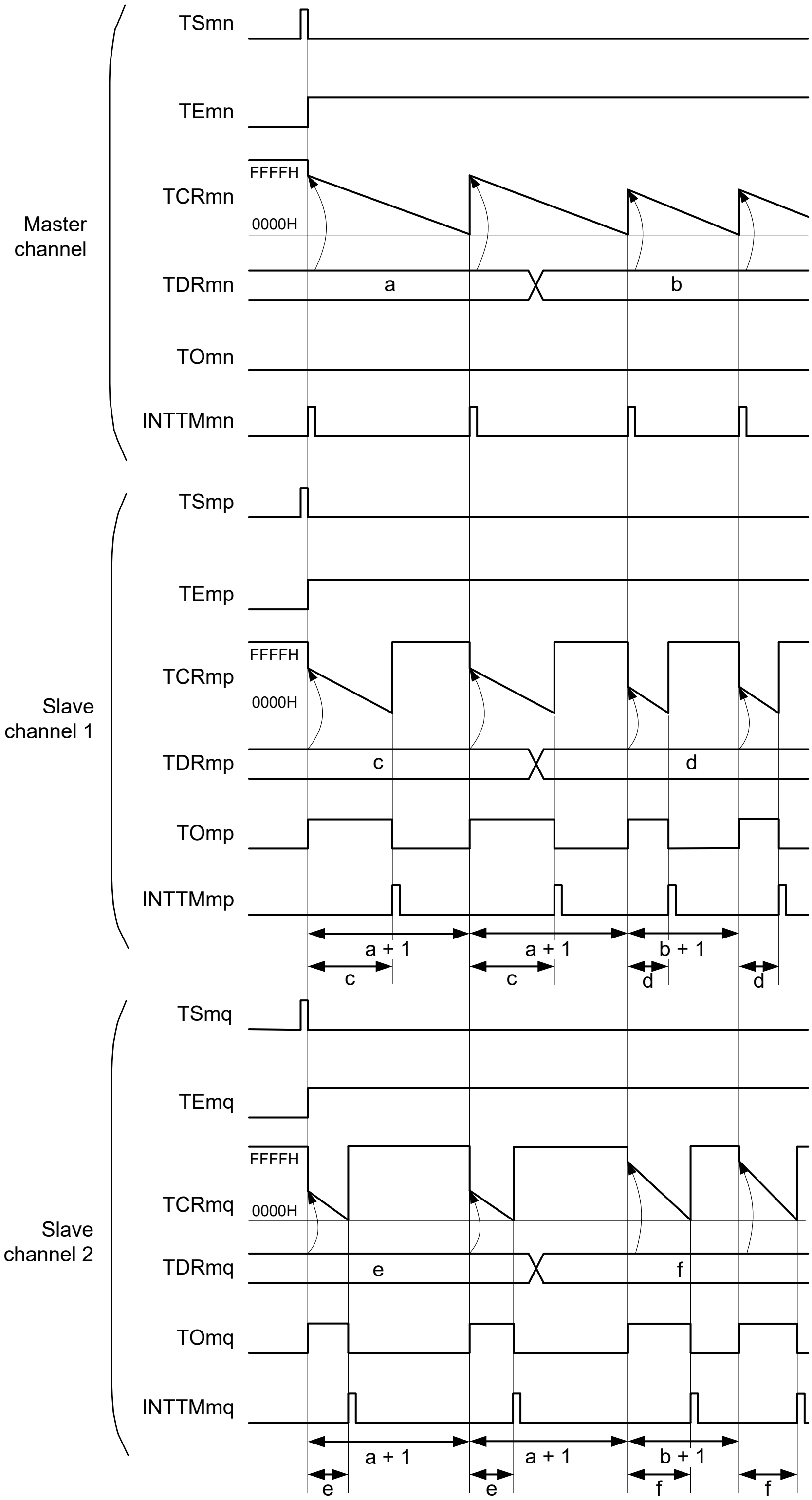
Remark m: Unit number (m = 0), n: Channel number (n = 0)
p: Slave channel number 1, q: Slave channel number 2
$n < p < q \leq 3$ (Where p and q are integers greater than n)

Figure 7 - 82 Block Diagram of Operation as Multiple PWM Output Function (output two types of PWMs)



Remark m: Unit number (m = 0), n: Channel number (n = 0)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 3 (Where p and q are integers greater than n)

Figure 7 - 83 Example of Basic Timing of Operation as Multiple PWM Output Function (Output two types of PWMs)



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

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 3 (Where p and q are integers greater than n)

Remark 2. TSmn, TSmp, TSmq: Bit n, p, q of timer channel start register m (TSm)
TEmn, TEmp, TEmq: Bit n, p, q of timer channel enable status register m (TEm)
TCRmn, TCRmp, TCRmq: Timer count registers mn, mp, mq (TCRmn, TCRmp, TCRmq)
TDRmn, TDRmp, TDRmq: Timer data registers mn, mp, mq (TDRmn, TDRmp, TDRmq)
TOmn, TOmp, TOmq: TOmn, TOmp, and TOmq pins output signal

Figure 7 - 84 Example of Set Contents of Registers When Multiple PWM Output Function (Master Channel) Is Used

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	MASTERmn Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
1: Generates INTTMmn when counting is started.

Selection of TImn pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Setting of MASTERmn bit (channel 2)
1: Master channel.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	0

0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn
	0

0: Cleared to 0 when master channel output mode (TOMmn = 0)

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn
	0

0: Sets master channel output mode.

Note TMRm2: MASTERmn = 1
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0)

Figure 7 - 85 Example of Set Contents of Registers
When Multiple PWM Output Function (Slave Channel) Is Used (output two types of PWMs)

(a) Timer mode register mp, mq (TMRmp, TMRmq)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmp	CKSmp1	CKSmp0		CCSmp	M/S Note	STSmp2	STSmp1	STSmp0	CISmp1	CISmp0			MDmp3	MDmp2	MDmp1	MDmp0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmq	CKSmq1	CKSmq0		CCSmq	M/S Note	STSmq2	STSmq1	STSmq0	CISmq1	CISmq0			MDmq3	MDmq2	MDmq1	MDmq0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

Operation mode of channel p, q
100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

Selection of TImp and TImq pins input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMmn of master channel.

Setting of MASTERmp and MASTERmq bits (channel 2)
0: Slave channel
Setting of SPLITmp and SPLITmq bits (channels 1, 3)
0: 16-bit timer mode.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel p, q.
10B: Selects CKm1 as operation clock of channel p, q.
* Make the same setting as master channel.

(b) Timer output register m (TOm)

	Bit q	Bit p
TOm	TOmq	TOmp
	1/0	1/0

0: Outputs 0 from TOmp or TOmq.
1: Outputs 1 from TOmp or TOmq.

(c) Timer output enable register m (TOEm)

	Bit q	Bit p
TOEm	TOEmq	TOEmp
	1/0	1/0

0: Stops the TOmp or TOmq output operation by counting operation.
1: Enables the TOmp or TOmq output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit q	Bit p
TOLm	TOLmq	TOLmp
	1/0	1/0

0: Positive logic output (active-high)
1: Negative logic output (active-low)

(e) Timer output mode register m (TOMm)

	Bit q	Bit p
TOMm	TOMmq	TOMmp
	1	1

1: Sets the slave channel output mode.

Note TMRm2: MASTERmp, MASTERmq bit
TMRm1, TMRm3: SPLITmp, SPLITmq bit

Remark m: Unit number (m = 0), n: Channel number (n = 0)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 3 (Where p and q are integers greater than n)

Figure 7 - 86 Operation Procedure When Multiple PWM Output Function Is Used (output two types of PWMs) (1/2)

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit m is stopped (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Input clock supply for timer array unit m is supplied. 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, mq (TMRmn, TMRmp, TMRmq) of each channel to be used (determines operation mode of channels). An interval (period) value is set to timer data register mn (TDRmn) of the master channel, and a duty factor is set to the TDRmp and TDRmq registers of the slave channels.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channels. The TOMmp and TOMmq bits of timer output mode register m (TOMm) are set to 1 (slave channel output mode). Clears the TOLmp and TOLmq bits to 0. Sets the TOmp and TOmq bits and determines default level of the TOmp and TOmq outputs. → Sets the TOEmp and TOEmq bits to 1 and enables operation of TOmp and TOmq. → Clears the port register and port mode register to 0. →	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. TOmp and TOmq do not change because channels stop operating. The TOmp and TOmq pins output the TOmp and TOmq set levels.

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

Figure 7 - 87 Operation Procedure When Multiple PWM Output Function Is Used (output two types of PWMs) (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	(Sets the TOEmp and TOEmq (slave) bits to 1 only when resuming operation.) The TSmn bit (master), and TSmp and TSmq (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. → The TSmn, TSmp, and TSmq bits automatically return to 0 because they are trigger bits.	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 TSRmq registers are not used.	The counter of the master channel loads the TDRmn register value to timer count register mn (TCRmn) and counts down. When the count value reaches TCRmn = 0000H, INTTMmn output is generated. At the same time, the value of the TDRmn register is loaded to the TCRmn register, and the counter starts counting down again. At the slave channel 1, the values of the TDRmp register are transferred to the TCRmp register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output levels of TOmp become active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. At the slave channel 2, the values of the TDRmq register are transferred to TCRmq register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output levels of TOmq become active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmq = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn bit (master), TTmp, and TTmq (slave) bits are set to 1 at the same time. → The TTmn, TTmp, and TTmq bits automatically return to 0 because they are trigger bits.	TEmn, TEmp, TEmq = 0, and count operation stops. The TCRmn, TCRmp, and TCRmq registers hold count value and stop. The TOmp and TOmq output are not initialized but hold current status.
	The TOEmp and TOEmq bits of slave channels are cleared to 0 and value is set to the TOmp and TOmq bits. →	The TOmp and TOmq pins output the TOmp and TOmq set levels.
TAU stop	To hold the TOmp and TOmq pin output levels Clears the TOmp and TOmq bits to 0 after the value to be held is set to the port register. → 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 cleared to 0. → To initialize all circuits, set the TAU0RES bit in the PRR0 register to 1. →	Input clock supply for timer array unit m is stopped All circuits are initialized and SFR of each channel is also initialized. (The TOmp and TOmq bits are cleared to 0 and the TOmp and TOmq pins are set to port mode.)

Remark m: Unit number (m = 0), n: Channel number (n = 0)
p: Slave channel number, q: Slave channel number
$n < p < q \leq 3$ (Where p and q are integer greater than n)

7.10 Cautions When Using Timer Array Unit

7.10.1 Cautions When Using Timer output

Depends on products, a pin is assigned a timer output and other alternate functions. In this case, outputs of the other alternate functions must be set in initial status.

For details, see **4.5 Register Settings When Using Alternate Function**.

CHAPTER 8 REAL-TIME CLOCK 2

8.1 Functions of Real-Time Clock 2

The real-time clock 2 (RTC2) has the following functions.

- Counters of year, month, day of the week, date, hour, minute, and second, that can count up to 99 years (with leap year correction function)
- Constant-period interrupt function (period: 0.5 seconds, 1 second, 1 minute, 1 hour, 1 day, 1 month)
- Alarm interrupt function (alarm: day of the week, hour, and minute)
- Pin output function of 1 Hz

A real-time clock interrupt signal (INTRTC) can be used for wakeup from STOP mode or as a trigger for SNOOZE mode of the A/D converter.

Caution The year, month, week, day, hour, minute and second can only be counted when a subsystem clock generator and RTC2/other clock (f_{SXR} = 32.768 kHz) is selected as the operation clock of the real-time clock 2. When the low-speed oscillation clock (f_{IL} = 15 kHz) is selected, only the constant-period interrupt function is available.
However, the constant-period interrupt interval when f_{IL} is selected will be calculated with the constant-period (the value selected with RTCC0 register) × f_{SXR}/f_{IL}.

8.2 Configuration of Real-time Clock 2

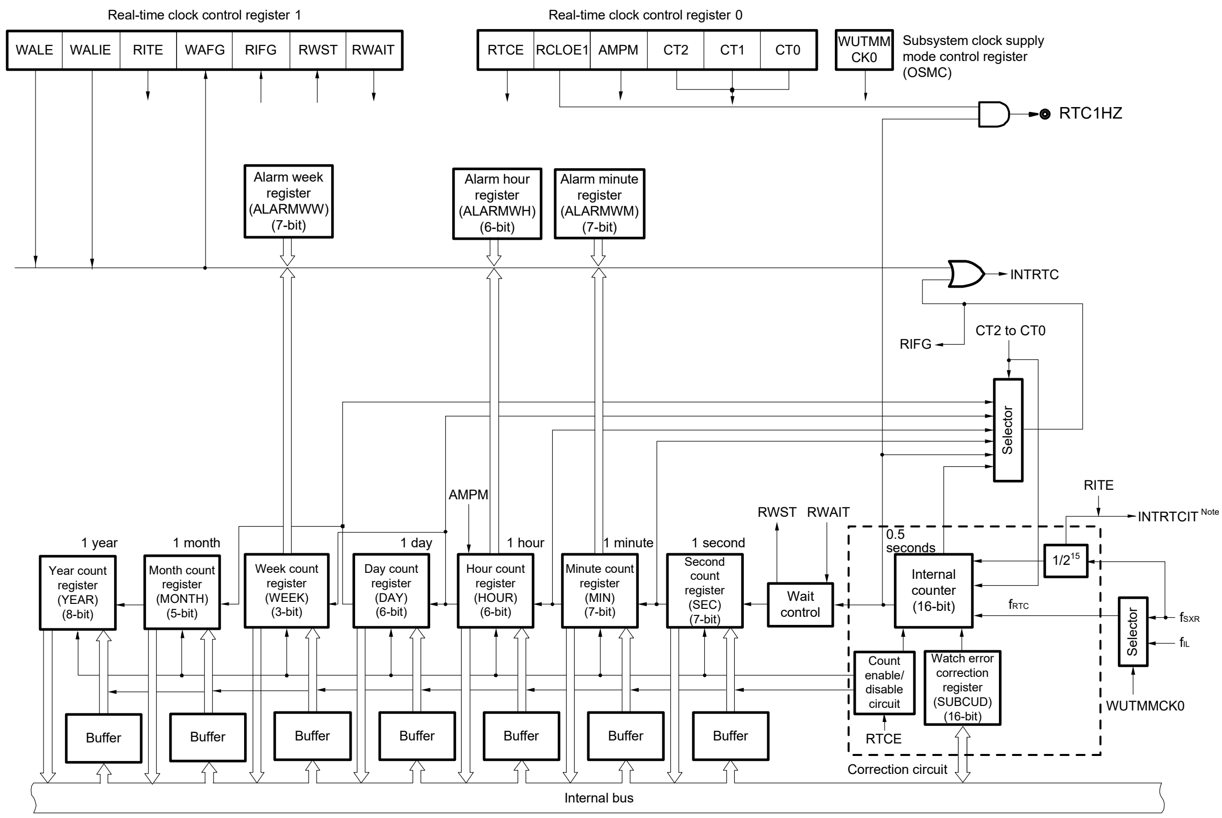
The real-time clock 2 includes the following hardware.

Table 8 - 1 Configuration of Real-time Clock 2

Item	Configuration
Counter	Counter (16-bit)
Control registers	Peripheral enable register 0 (PER0)
	Peripheral enable register 2 (PER2)
	Subsystem clock supply mode control register (OSMC)
	Real-time clock control register 0 (RTCC0)
	Real-time clock control register 1 (RTCC1)
	Second count register (SEC)
	Minute count register (MIN)
	Hour count register (HOUR)
	Day count register (DAY)
	Week count register (WEEK)
	Month count register (MONTH)
	Year count register (YEAR)
	Watch error correction register (SUBCUD)
	Alarm minute register (ALARMWM)
	Alarm hour register (ALARMWH)
	Alarm week register (ALARMWW)

Figure 8 - 1 shows the Block Diagram of Real-time Clock 2.

Figure 8 - 1 Block Diagram of Real-time Clock 2



Note This interrupt indicates the timing at which the correction value is captured from the clock error correction register (SUBCUD). The correction value is captured at an interval of 1 second (on f_{SXR} basis).

8.3 Registers Controlling Real-time Clock 2

The real-time clock 2 is controlled by the following registers.

• Peripheral enable register 0 (PER0)
• Peripheral enable register 2 (PER2)
• Subsystem clock supply mode control register (OSMC)
• Real-time clock control register 0 (RTCC0)
• Real-time clock control register 1 (RTCC1)
• Second count register (SEC)
• Minute count register (MIN)
• Hour count register (HOUR)
• Day count register (DAY)
• Week count register (WEEK)
• Month count register (MONTH)
• Year count register (YEAR)
• Watch error correction register (SUBCUD)
• Alarm minute register (ALARMWM)
• Alarm hour register (ALARMWH)
• Alarm week register (ALARMWW)
• Port mode register 5 (PM5)
• Port register 5 (P5)

The following shows the register states depending on reset sources.

Reset Source	System-related Register Note 1	Calendar-related Register Note 2
POR	Reset	Not reset
External reset	Retained	Retained
WDT	Retained	Retained
TRAP	Retained	Retained
LVD	Retained	Retained
Other internal reset	Retained	Retained

Note 1. RTCC0, RTCC1, SUBCUD
Note 2. SEC, MIN, HOUR, DAY, WEEK, MONTH, YEAR, ALARMWM, ALARMWH, ALARMWW, (counter)

Reset generation 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 supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the real-time clock 2 registers are manipulated, be sure to set bit 7 (RTCWEN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 8 - 2 Format of Peripheral enable register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	4	3	<2>	1	<0>
PER0	RTCWEN	0	ADCEN	0	0	SAU0EN	0	TAU0EN

RTCWEN	Control of internal clock supply to real-time clock 2
0	Stops input clock supply. • SFR used by the real-time clock 2 cannot be written. • The real-time clock 2 can operate.
1	Enables input clock supply. • SFR used by the real-time clock 2 can be read/written. • The real-time clock 2 can operate.

Caution 1. The clock error correction register (SUBCUD) can be written/read by setting RTCWEN = 1 in peripheral enable register 0 (PER0) or FMCEN = 1 in peripheral enable register 1 (PER1).

Caution 2. When using the real-time clock 2, be sure to set RTCWEN = 1 while oscillation of the count clock (f_{RTC}) is stable before setting the following registers. When RTCEN = 0, writing to the real-time clock 2 control registers is ignored and all the read values are the set values when RTCWEN = 1 (except for the subsystem clock supply mode control register (OSMC)).

- **Real-time clock control register 0 (RTCC0)**
- **Real-time clock control register 1 (RTCC1)**
- **Second count register (SEC)**
- **Minute count register (MIN)**
- **Hour count register (HOUR)**
- **Day count register (DAY)**
- **Week count register (WEEK)**
- **Month count register (MONTH)**
- **Year count register (YEAR)**
- **Watch error correction register (SUBCUD)**
- **Alarm minute register (ALARMWM)**
- **Alarm hour register (ALARMWH)**
- **Alarm week register (ALARMWW)**

Caution 3. Be sure to clear the following bits to 0.
Bits 1, 3, 4, and 6

8.3.2 Peripheral enable register 2 (PER2)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

Of the registers that are used to control the real-time clock 2, the clock error correction register (SUBCUD) can be set by setting bit 6 (FMCEN) of this register to 1.

The PER2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 8 - 3 Format of Peripheral enable register 2 (PER2)

Address: F00FCH After reset: 00H R/W

Symbol	<7>	6	<5>	4	3	2	1	0
PER2	TMKAEN	FMCEN	DOCEN	0	0	0	0	0

FMCEN	Control of internal clock supply to frequency measurement circuit
0	Stops input clock supply. • SFR used by the frequency measurement circuit cannot be written. • SUBCUD register used by the real-time clock 2 cannot be written. • The frequency measurement circuit is in the reset status.
1	Enables input clock supply. • SFR used by the frequency measurement circuit can be read/written. • SUBCUD register used by the real-time clock 2 can be read/written.

Caution 1. The clock error correction register (SUBCUD) can be read or written by setting RTCWEN of peripheral enable register 0 (PER0) to 1 or setting FMCEN of peripheral enable register 1 (PER1) to 1.

Caution 2. Be sure to set bits 0 to 4 to 0.

8.3.3 Subsystem clock supply mode control register (OSMC)

This register is used to reduce power consumption by stopping unnecessary clock functions.
If the RTCLPC bit is set to 1, power consumption can be reduced, because clock supply to the peripheral functions, except the real-time clock 2, 12-bit interval timer, 8-bit interval timer, clock output/buzzer output controller is stopped in STOP mode or HALT mode while subsystem clock is selected as CPU clock.
In addition, the OSMC register can be used to select the operation clock of the real-time clock 2, frequency measurement circuit, 12-bit interval timer, 8-bit interval timer, and clock output/buzzer output controller.
The OSMC register can be set by an 8-bit memory manipulation instruction.
Reset signal generation sets this register to undefined.

Figure 8 - 4 Format of Subsystem clock supply mode control register (OSMC)

Address: F00F3H After reset: Undefined R/W Note 1

Symbol	<7>	6	5	<4>	3	2	1	0
OSMC	RTCLPC	0	0	WUTMMCK0	x	x	0	0

RTCLPC Note 7	Setting in STOP mode or HALT mode while subsystem clock is selected as CPU clock
0	Enables supply of subsystem clock to peripheral functions (See **Tables 23 - 1** to **23 - 4** for peripheral functions whose operations are enabled.)
1	Stops supply of subsystem clock to peripheral functions other than the real-time clock 2, 12-bit interval timer, 8-bit interval timer, and clock output/buzzer output controller.

WUTMMCK0	Selection of operation clock for real-time clock 2, frequency measurement circuit, 12-bit interval timer, 8-bit interval timer, and clock output/buzzer output controller
0	• Subsystem clock Note 2
1	• Low-speed on-chip oscillator clock Notes 3, 4, 5, 6

FMTRGSEL Note 10	WUTMMCK0	Operating clock selection for frequency measurement circuit count operation/stop trigger clock and real-time clock 2
0	0	f_{SX} selected for the frequency measurement circuit/real-time clock 2
0	1	f_{IL} selected for the real-time clock 2 (constant-period interrupt function) Note 8
1	0	Setting prohibited
1	1	f_{IL} selected for the frequency measurement circuit Note 9

Note 1. Be sure to set bits 0, 1, 5, and 6 to 0. Bits 2 and 3 are read-only, write is ignored.

Note 2. Do not set the WUTMMCK0 bit to 0 and the FMTRGSEL bit in the FMCKS register to 1.

Note 3. Do not set the WUTMMCK0 bit to 1 while the sub clock is oscillating.

Note 4. Switching between the subsystem clock and the low-speed on-chip oscillator clock can be enabled by the WUTMMCK0 bit only when all of the real-time clock 2, frequency measurement circuit, 12-bit interval timer, 8-bit interval timer, and clock output/buzzer output functions are stopped.

Note 5. f_{IL} can be selected as the operating clock of the real-time clock 2 when the WUTMMCK0 bit is 1 and the FMTRGSEL bit in the FMCKS register is 0. Also, only the constant-period interrupt function of the real-time clock 2 can be used at this time; the clock count function cannot be used.

Note 6. $f_{IL}/2$ can be selected as the operating clock of the frequency measurement circuit when the WUTMMCK0 bit is 1 and the FMTRGSEL bit in the FMCKS register is 1.

Note 7. When the sub clock is selected (SELLOSC = 0) by bit 0 (SELLOSC) of the CKSEL register and RTCLPC is set to 1, the subsystem clock is stopped. However, when the low-speed on-chip oscillator clock is selected (SELLOSC = 1) and RTCLPC is set to 1, the subsystem clock is not stopped.

Note 8. The frequency measurement function cannot be used.

Note 9. The real-time clock 2 cannot be used.

Note 10. Bit 4 of the frequency measurement circuit clock select register (FMCKS)

Remark x: Undefined

8.3.4 Real-time clock control register 0 (RTCC0)

The RTCC0 register is an 8-bit register that is used to start or stop the real-time clock 2 operation, and set a 12- or 24-hour system and the constant-period interrupt function.

The RTCC0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Internal reset generated by the power-on-reset circuit clears this register to 00H.

Figure 8 - 5 Format of Real-time clock control register 0 (RTCC0)

Address: FFF9DH After reset: 00H R/W

Symbol	<7>	6	<5>	4	3	2	1	0
RTCC0	RTCE	0	RCLOE1	0	AMPM	CT2	CT1	CT0

RTCE Note 1	Real-time clock 2 operation control
0	Stops counter operation.
1	Starts counter operation.

RCLOE1 Note 2	RTC1HZ pin output control
0	Disables output of the RTC1HZ pin (1 Hz).
1	Enables output of the RTC1HZ pin (1 Hz).
Output of 1 Hz is not output because the clock counter does not operate when RTCE = 0.	

AMPM	12-/24-hour system select
0	12-hour system (a.m. and p.m. are displayed.)
1	24-hour system
• When changing the value of the AMPM bit while the clock counter operates (RTCE = 1), set the RWAIT bit (bit 0 of real-time clock control register 1 (RTCC1)) before rewriting the value. When the value of the AMPM bit is changed, the value of the hour count register (HOUR) changes to the value corresponding to the set hour system. When the AMPM value is 0, the 12-hour system is displayed. When the value is 1, the 24-hour system is displayed. • Table 8 - 3 shows the displayed time digits.	

CT2	CT1	CT0	Constant-period interrupt (INTRTC) selection
0	0	0	Does not use fixed-cycle interrupt function.
0	0	1	Once per 0.5 s (synchronized with second count up)
0	1	0	Once per 1 s (same time as second count up)
0	1	1	Once per 1 m (second 00 of every minute)
1	0	0	Once per 1 hour (minute 00 and second 00 of every hour)
1	0	1	Once per 1 day (hour 00, minute 00, and second 00 of every day)
1	1	×	Once per 1 month (Day 1, hour 00 a.m., minute 00, and second 00 of every month)
When changing the values of the CT2 to CT0 bits while the counter operates (RTCE = 1), rewrite the values of the CT2 to CT0 bits after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, after rewriting the values of the CT2 to CT0 bits, enable interrupt servicing after clearing the RIFG and RTCIF flags.			

Table 8 - 2 Relation between RTCE and RCLOE1 Settings and Status

Register Settings Status		Status	
RTCE	RCLOE1	Real-time clock 2	RTC1HZ pin output
0	×	Counting stopped	No output
1	0	Count operation	No output
	1	Count operation	1-Hz output

Note 1. When shifting to STOP mode immediately after setting RTCE to 1, use the procedure shown in Figure 8 - 21 Procedure for Shifting to HALT/STOP Mode After Setting RTCE bit to 1.

Note 2. When the RCLOE1 bit is set while the clock counter operates (RTCE = 1), a glitch may be output to the 1-Hz output pin (RTC1HZ).

Caution **Be sure to clear bits 6 and 4 to 0.**

Remark ×: don't care

8.3.5 Real-time clock control register 1 (RTCC1)

The RTCC1 register is an 8-bit register that is used to control the alarm interrupt function and the wait time of the counter.

The RTCC1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Internal reset generated by the power-on-reset circuit clears this register to 00H.

Figure 8 - 6 Format of Real-time clock control register 1 (RTCC1) (1/3)

Address: FFF9EH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	<1>	<0>
RTCC1	WALE	WALIE	RITE	WAFG	RIFG	0	RWST	RWAIT

WALE	Alarm operation control
0	Match operation is invalid.
1	Match operation is valid.
When setting a value to the WALE bit while the counter operates (RTCE = 1) and WALIE = 1, rewrite the WALE bit after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG and RTCIF flags after rewriting the WALE bit. When setting each alarm register (WALIE flag of real-time clock control register 1 (RTCC1), the alarm minute register (ALARMWM), the alarm hour register (ALARMWH), and the alarm week register (ALARMWW)), set match operation to be invalid (0) for the WALE bit.	

WALIE	Control of alarm interrupt (INTRTC) function operation
0	Does not generate interrupt on matching of alarm.
1	Generates interrupt on matching of alarm.

Caution If writing is performed to RTCC1 with a 1-bit manipulation instruction, the RIFG and WAFG flags may be cleared. Therefore, to perform writing to RTCC1, be sure to use an 8- bit manipulation instruction. To prevent the RIFG and WAFG flags from being cleared during writing, set 1 (writing disabled) to the corresponding bit. If the RIFG and WAFG flags are not used and the value may be changed, RTCC1 may be written by using a 1-bit manipulation instruction.

Figure 8 - 7 Format of Real-time clock control register 1 (RTCC1) (2/3)

Address: FFF9EH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	<1>	<0>
RTCC1	WALE	WALIE	RITE	WAFG	RIFG	0	RWST	RWAIT

RITE	Control of correction timing signal interrupt (INTRTIT) function operation
0	Does not generate interrupt of correction timing signal.
1	Generates interrupt of correction timing signal.

WAFG	Alarm detection status flag
0	Alarm mismatch
1	Detection of matching of alarm
This is a status flag that indicates detection of matching with the alarm. It is valid only when WALE = 1 and is set to 1 one clock (32.768 kHz) after matching of the alarm is detected. This flag is cleared when 0 is written to it.	

RIFG	Constant-period interrupt status flag
0	Constant-period interrupt is not generated.
1	Constant-period interrupt is generated.
This flag indicates the status of generation of the constant-period interrupt. When the constant-period interrupt is generated, it is set to 1. This flag is cleared when 0 is written to it. Writing 1 to it is invalid.	

Caution **If writing is performed to RTCC1 with a 1-bit manipulation instruction, the RIFG and WAFG flags may be cleared. Therefore, to perform writing to RTCC1, be sure to use an 8- bit manipulation instruction. To prevent the RIFG and WAFG flags from being cleared during writing, set 1 (writing disabled) to the corresponding bit. If the RIFG and WAFG flags are not used and the value may be changed, RTCC1 may be written by using a 1-bit manipulation instruction.**

Figure 8 - 8 Format of Real-time clock control register 1 (RTCC1) (3/3)

Address: FFF9EH After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	2	<1>	<0>
RTCC1	WALE	WALIE	RITE	WAFG	RIFG	0	RWST	RWAIT

RWST	Wait status flag of real-time clock 2
0	Counter is operating.
1	Mode to read or write counter value.
This status flag indicates whether the setting of the RWAIT bit is valid. Before reading or writing the counter value, confirm that the value of this flag is 1. Even if the RWAIT bit is set to 0, the RWST bit is not set to 0 while writing to the counter. After writing is completed, the RWST bit is set to 0.	

RWAIT	Wait control of real-time clock 2
0	Sets counter operation.
1	Stops SEC to YEAR counters. Mode to read or write counter value.
This bit controls the operation of the counter. Be sure to write 1 to it to read or write the counter value. As the counter (16-bit) is continuing to run, complete reading or writing within one second and turn back to 0. <R> When reading or writing to the counter is required while generation of the alarm interrupt is enabled, first set the CT2 to CT0 bits to 010B (generating the constant-period interrupt once per 1 second). Then, complete the processing from setting the RWAIT bit to 1 to setting it to 0 before generation of the next constant-period interrupt. After RWAIT is set to 1, it takes up to one f_{RTC} clock cycle before reading/writing the count value is enabled (RWST = 1). When the internal counter (16-bit) overflowed while RWAIT = 1, it keeps the event of overflow until RWAIT = 0, then counts up.[Notes 1, 2] However, when it wrote a value to second count register, it will not keep the overflow event.	

Note 1. When the RWAIT bit is set to 1 within one cycle of f_{RTC} clock after setting the RTCE bit to 1, the RWST bit being set to 1 may take up to two cycles of the operating clock (f_{RTC}).

Note 2. When the RWAIT bit is set to 1 within one cycle of f_{RTC} clock after release from the standby mode (HALT mode, STOP mode, or SNOOZE mode), the RWST bit being set to 1 may take up to two cycles of the operating clock (f_{RTC}).

Caution If writing is performed to RTCC1 with a 1-bit manipulation instruction, the RIFG and WAFG flags may be cleared. Therefore, to perform writing to RTCC1, be sure to use an 8- bit manipulation instruction. To prevent the RIFG and WAFG flags from being cleared during writing, set 1 (writing disabled) to the corresponding bit. If the RIFG and WAFG flags are not used and the value may be changed, RTCC1 may be written by using a 1-bit manipulation instruction.

Remark 1. Constant-period interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the constant-period interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

Remark 2. Writing to the second count register (SEC) clears the 16-bit internal counter.

8.3.6 Second count register (SEC)

The SEC register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of seconds.

It is a decimal counter that counts up when the counter (16-bit) overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 fRTC clocks 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.

Reset signal generation not clears this register to default value.

Figure 8 - 9 Format of Second count register (SEC)

Address: FFF92H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1

Caution When reading or writing to SEC while the clock counter operates (RTCE = 1), be sure to use the flows shown in 8.4.3 Reading the counters of real-time clock 2 and 8.4.4 Writing to the counters of real-time clock 2.

Remark Writing to the second count register (SEC) clears the 16-bit internal counter.

8.3.7 Minute count register (MIN)

The MIN register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of minutes.

It is a decimal counter that counts up when the second counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 fRTC clocks 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.

Reset signal generation not clears this register to default value.

Figure 8 - 10 Format of Minute count register (MIN)

Address: FFF93H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1

Caution When reading or writing to MIN while the clock counter operates (RTCE = 1), be sure to use the flows shown in 8.4.3 Reading the counters of real-time clock 2 and 8.4.4 Writing to the counters of real-time clock 2.

8.3.8 Hour count register (HOUR)

The HOUR register is an 8-bit register that takes a value of 00 to 23 or 01 to 12 and 21 to 32 (decimal) and indicates the count value of hours.
It is a decimal counter that counts up when the minute counter overflows.
When data is written to this register, it is written to a buffer and then to the counter up to 2 f_{RTC} clocks later. Even if the minute count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Specify a decimal value of 00 to 23, 01 to 12, or 21 to 32 by using BCD code according to the time system specified using bit 3 (AMPM) of real-time clock control register 0 (RTCC0).
If the AMPM bit value is changed, the values of the HOUR register change according to the specified time system.
The HOUR register can be set by an 8-bit memory manipulation instruction.
Reset signal generation not clears this register to default value.

Figure 8 - 11 Format of Hour count register (HOUR)

Address: FFF94H After reset: 12H R/W

Symbol	7	6	5	4	3	2	1	0
HOUR	0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1

Caution 1. Bit 5 (HOUR20) of the HOUR register indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

Caution 2. When reading or writing to HOUR while the clock counter operates (RTCE = 1), be sure to use the flows shown in 8.4.3 Reading the counters of real-time clock 2 and 8.4.4 Writing to the counters of real-time clock 2.

Table 8 - 3 shows the relationship between the setting value of the AMPM bit, the hour count register (HOUR) value, and time.

Table 8 - 3 Displayed Time Digits

24-Hour Display (AMPM = 1)		12-Hour Display (AMPM = 1)	
Time	HOUR Register	Time	HOUR Register
0	00 H	12 a.m.	12 H
1	01 H	1 a.m.	01 H
2	02 H	2 a.m.	02 H
3	03 H	3 a.m.	03 H
4	04 H	4 a.m.	04 H
5	05 H	5 a.m.	05 H
6	06 H	6 a.m.	06 H
7	07 H	7 a.m.	07 H
8	08 H	8 a.m.	08 H
9	09 H	9 a.m.	09 H
10	10 H	10 a.m.	10 H
11	11 H	11 a.m.	11 H
12	12 H	12 p.m.	32 H
13	13 H	1 p.m.	21 H
14	14 H	2 p.m.	22 H
15	15 H	3 p.m.	23 H
16	16 H	4 p.m.	24 H
17	17 H	5 p.m.	25 H
18	18 H	6 p.m.	26 H
19	19 H	7 p.m.	27 H
20	20 H	8 p.m.	28 H
21	21 H	9 p.m.	29 H
22	22 H	10 p.m.	30 H
23	23 H	11 p.m.	31 H

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.

8.3.9 Day count register (DAY)

The DAY register is an 8-bit register that takes a value of 1 to 31 (decimal) and indicates the count value of days.
It is a decimal counter that count ups when the hour counter overflows.
This counter counts as follows.

[DAY count values]
• 01 to 31 (January, March, May, July, August, October, December)
• 01 to 30 (April, June, September, November)
• 01 to 29 (February, leap year)
• 01 to 28 (February, normal year)

When data is written to this register, it is written to a buffer and then to the counter up to 2 f_{RTC} clocks 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.
Reset signal generation not clears this register to default value.

Figure 8 - 12 Format of Day count register (DAY)

Address: FFF96H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1

Caution When reading or writing to DAY while the clock counter operates (RTCE = 1), be sure to use the flows shown in 8.4.3 Reading the counters of real-time clock 2 and 8.4.4 Writing to the counters of real-time clock 2.

8.3.10 Week count register (WEEK)

The WEEK register is an 8-bit register that takes a value of 0 to 6 (decimal) and indicates the count value of weekdays.

It is a decimal counter that counts up when a carry to the date counter occurs.

When data is written to this register, it is written to a buffer and then to the counter up to 2 f_{RTC} clocks 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.

Reset signal generation not clears this register to default value.

Figure 8 - 13 Format of Week count register (WEEK)

Address: FFF95H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WEEK	0	0	0	0	0	WEEK4	WEEK2	WEEK1

Caution 1. The value corresponding to the month count register (MONTH) or the day count register (DAY) is not stored in the week count register (WEEK) automatically. After reset release, set the week count register as follow.

Day	WEEK
Sunday	00 H
Monday	01 H
Tuesday	02 H
Wednesday	03 H
Thursday	04 H
Friday	05 H
Saturday	06 H

Caution 2. When reading or writing to WEEK while the clock counter operates (RTCE = 1), be sure to use the flows shown in 8.4.3 Reading the counters of real-time clock 2 and 8.4.4 Writing to the counters of real-time clock 2.

8.3.11 Month count register (MONTH)

The MONTH register is an 8-bit register that takes a value of 1 to 12 (decimal) and indicates the count value of months.
It is a decimal counter that count ups when the date counter overflows.
When data is written to this register, it is written to a buffer and then to the counter up to 2 f_{RTC} clocks 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.
Reset signal generation not clears this register to default value.

Figure 8 - 14 Format of Month count register (MONTH)

Address: FFF97H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

Caution When reading or writing to MONTH while the clock counter operates (RTCE = 1), be sure to use the flows shown in 8.4.3 Reading the counters of real-time clock 2 and 8.4.4 Writing to the counters of real-time clock 2.

8.3.12 Year count register (YEAR)

The YEAR register is an 8-bit register that takes a value of 0 to 99 (decimal) and indicates the count value of years.
It is a decimal counter that counts up when the month count register (MONTH) overflows.
Values 00, 04, 08, …, 92, and 96 indicate a leap year.
When data is written to this register, it is written to a buffer and then to the counter up to 2 f_{RTC} clocks 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.
Reset signal generation not clears this register to default value.

Figure 8 - 15 Format of Year count register (YEAR)

Address: FFF98H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

Caution When reading or writing to YEAR while the clock counter operates (RTCE = 1), be sure to use the flows shown in 8.4.3 Reading the counters of real-time clock 2 and 8.4.4 Writing to the counters of real-time clock 2.

8.3.13 Watch error correction register (SUBCUD)

This register is used to correct the clock with a minimum resolution and accuracy of 0.96 ppm when it is slow or fast by changing the counter value every second.

F8 to F0 of SUBCUD are a 9-bit fixed point type (two's complement) register. See **Table 8 - 5 Clock Error Correction Values**.

The SUBCUD register can be set by an 16-bit memory manipulation instruction.

Internal reset generated by the power-on-reset circuit clears this register to 0020H.

Figure 8 - 16 Format of Watch error correction register (SUBCUD)

Address: F0310H After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SUBCUD	F15	0	0	0	0	0	0	F8	F7	F6	F5	F4	F3	F2	F1	F0

F15	Clock error correction enable
0	Stops clock error correction.
1	Enables clock error correction.

The range of value that can be corrected by using the clock error correction register (SUBCUD) is shown in Table 8 - 4.

Table 8 - 4 Correctable Range of Crystal Resonator Oscillation Frequency Deviation

Item	Value
Correctable range	-274.6 ppm to +212.6 ppm
Maximum quantization error	±0.48 ppm
Minimum resolution	0.96 ppm

Table 8 - 5 Clock Error Correction Values

SUBCUD										Target Correction Values
F15	F8	F7	F6	F5	F4	F3	F2	F1	F0	
1	1	0	0	0	0	0	0	0	0	-274.6 ppm
	1	0	0	0	0	0	0	0	1	-273.7 ppm
	1	0	0	0	0	0	0	1	0	-272.7 ppm
	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
	1	1	1	1	1	1	1	0	1	-33.3 ppm
	1	1	1	1	1	1	1	1	0	-32.4 ppm
	1	1	1	1	1	1	1	1	1	-31.4 ppm
	0	0	0	0	0	0	0	0	0	-30.5 ppm
	0	0	0	0	0	0	0	0	1	-29.6 ppm
	0	0	0	0	0	0	0	1	0	-28.6 ppm
	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
	0	0	0	0	1	1	1	1	1	-0.95 ppm
	0	0	0	1	0	0	0	0	0	0 ppm
	0	0	0	1	0	0	0	0	1	0.95 ppm
	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
	0	1	1	1	1	1	1	0	1	210.7 ppm
	0	1	1	1	1	1	1	1	0	211.7 ppm
	0	1	1	1	1	1	1	1	1	212.6 ppm
0	×	×	×	×	×	×	×	×	×	Clock error correction stopped

The F8 to F0 value of the SUBCUD register is calculated from the target correction value by using the following expression.

$$\text{SUBCUD[8:0]} = \left(\frac{\text{Target correction value [ppm]} \times 2^{15}}{10^{6}} \right)_{\text{2's complement (9-bit fixed point type)}} + 0001.00000\text{B}$$

Caution The target correction value is the oscillation frequency deviation (unit: [ppm]) of the crystal resonator. For calculating the correction value, see 8.4.8 Example of watch error correction of real-time clock 2.

Examples 1. When target correction value = 18.3 [ppm]

SUBCUD[8:0] = $(18.3 \times 2^{15}/10^{6})$ 2's complement (9-bit fixed point type) + 0001.00000B
= (0.59375) 2's complement (9-bit fixed point type) + 0001.00000B
= 0000.10011B + 0001.00000B
= 0001.10011B

Examples 2. When target correction value = -18.3 [ppm]

SUBCUD[8:0] = $(-18.3 \times 2^{15}/10^{6})$ 2's complement (9-bit fixed point type) + 0001.00000B
= (-0.59965) 2's complement (9-bit fixed point type) + 0001.00000B
= 1111.01101B + 0001.00000B
= 0000.01101B

8.3.14 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.
Reset signal generation not clears this register to default value.

Figure 8 - 17 Format of Alarm minute register (ALARMWM)

Address: FFF9AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

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.

8.3.15 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.
Reset signal generation not clears this register to default value.

Figure 8 - 18 Format of Alarm hour register (ALARMWH)

Address: FFF9BH After reset: 12H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWH	0	0	WH20	WH10	WH8	WH4	WH2	WH1

Caution 1. Set a decimal value of 00 to 23 or 01 to 12 and 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Caution 2. Bit 5 (WH20) of the ALARMWH register indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

8.3.16 Alarm week register (ALARMWW)

This register is used to set date of alarm.

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

Reset signal generation not clears this register to default value.

Figure 8 - 19 Format of Alarm week register (ALARMWW)

Address: FFF9CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

Table 8 - 6 shows an example of setting the alarm.

Table 8 - 6 Setting Alarm

Time of Alarm	Day							12-Hour Display				24-Hour Display			
	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Hour 10	Hour 1	Minute 10	Minute 1	Hour 10	Hour 1	Minute 10	Minute 1
	WW0	WW1	WW2	WW3	WW4	WW5	WW6								
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

8.3.17 Registers controlling port functions of pins to be used for real-time clock

Using a port pin for real-time clock output requires setting of the registers that control the port functions multiplexed on the target pin (port mode register (PMxx), port register (Pxx)). For details, see **4.3.1 Port mode registers (PMxx)** and **4.3.2 Port registers (Pxx)**.

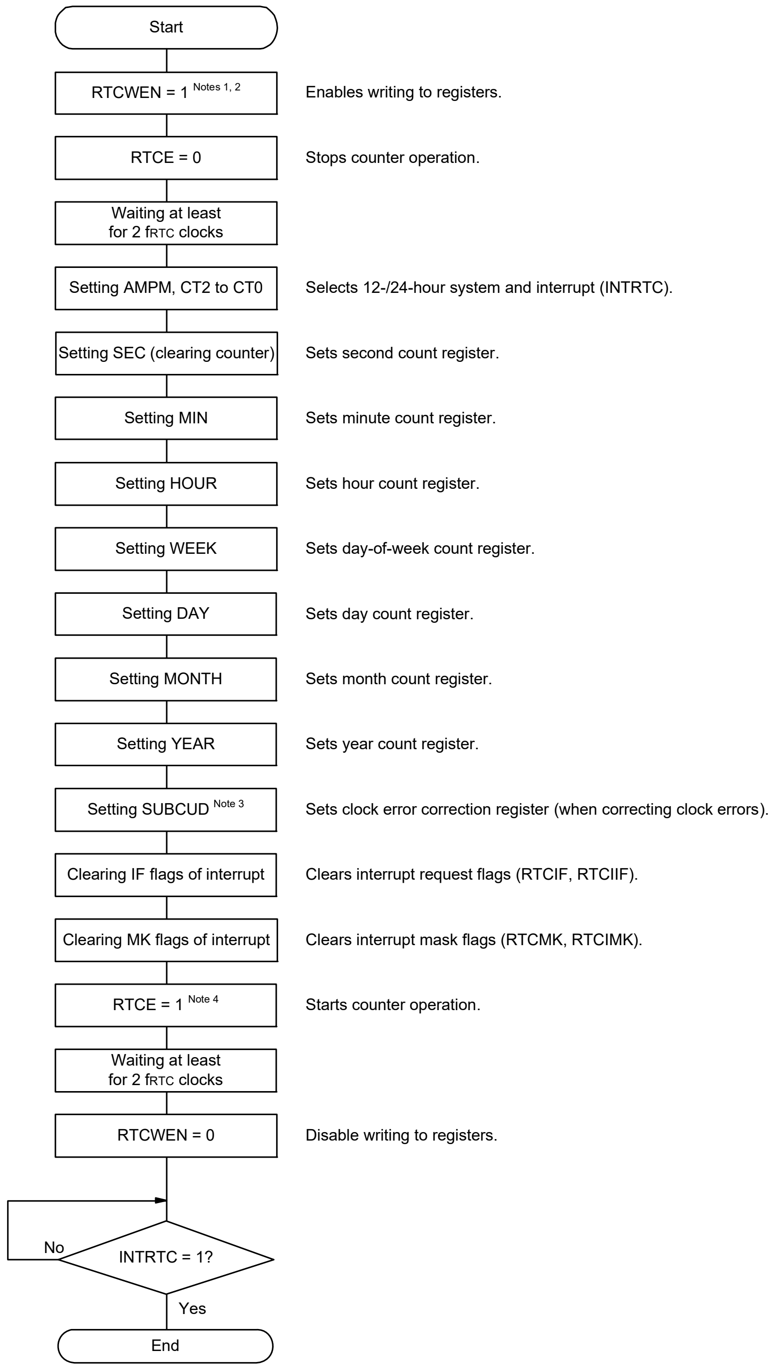
Specifically, using a port pin with a multiplexed real-time clock output function (P50/RTC1HZ) for real-time clock output, requires setting the corresponding bits in the port mode register (PMxx) and port register (Pxx) to 0.

Example: When P50/RTC1HZ is to be used for clock or buzzer real-time clock output
Set the PM50 bit of port mode register 5 to 0.
Set the P50 bit of port register 5 to 0.

8.4 Real-time Clock 2 Operation

8.4.1 Starting operation of real-time clock 2

Figure 8 - 20 Procedure for Starting Operation of Real-time Clock 2



Note 1. When the RTC register is not accessed, set RTCWEN = 0 to prevent writing to the clock counter inadvertently.

Note 2. First set the RTCWEN bit to 1, while oscillation of the count clock (f_{RTC}) is stable.

Note 3. Set up the SUBCUD register only if the watch error must be corrected. For details about how to calculate the correction value, see **8.4.8 Example of watch error correction of real-time clock 2**.

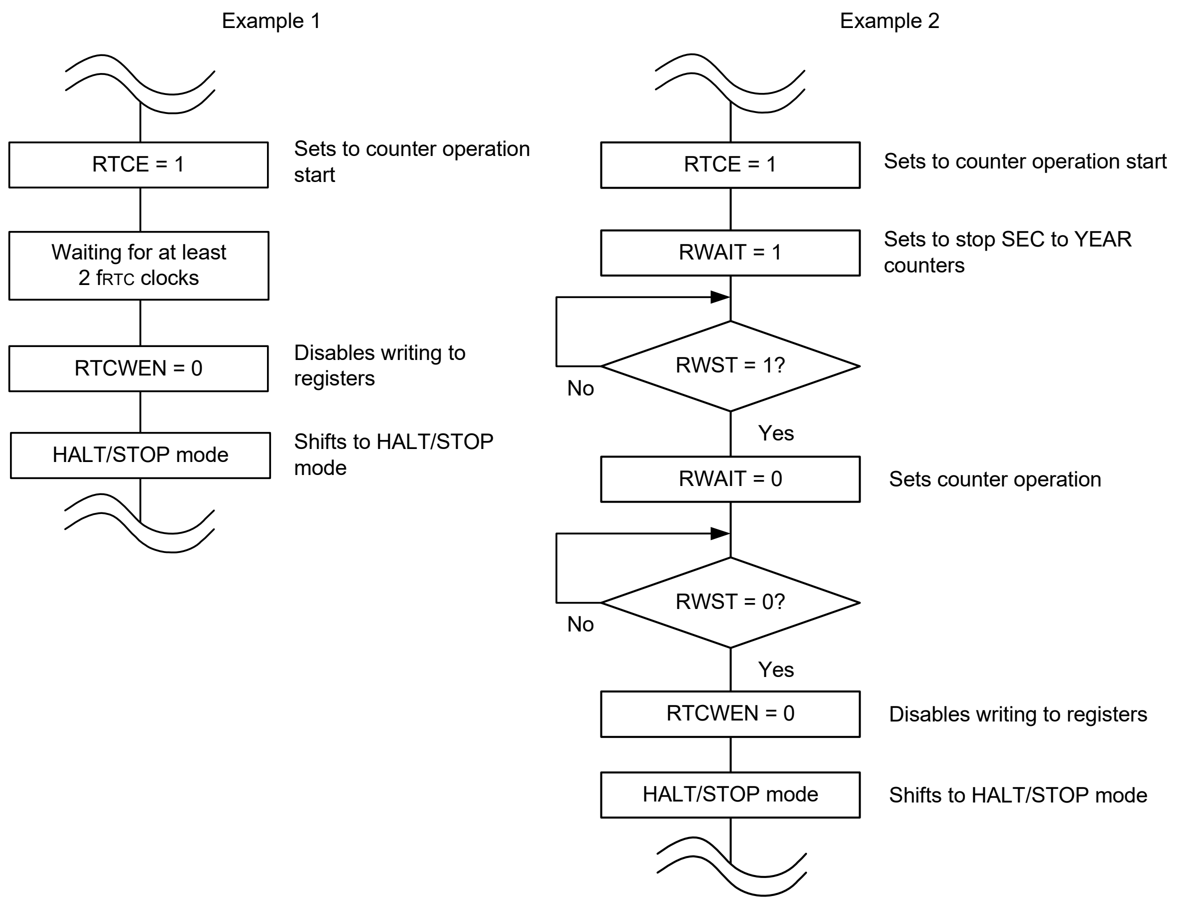
Note 4. Confirm the procedure described in **8.4.2 Shifting to HALT/STOP mode after starting operation** when shifting to HALT/STOP mode without waiting for INTRTC = 1 after RTCE = 1.

8.4.2 Shifting to HALT/STOP mode after starting operation

Perform one of the following processing when shifting to HALT/STOP mode immediately after setting the RTCE bit to 1.
However, after setting the RTCE bit to 1, this processing is not required when shifting to STOP mode after the first INTRTC interrupt has occurred.

(1) Shifting to HALT/STOP mode when at least two count clocks (f_{RTC}) have elapsed after setting the RTCE bit to 1 (see **Figure 8 - 21**, **Example 1**).
(2) Checking by polling the RWST bit to become 1, after setting the RTCE bit to 1 and then setting the RWAIT bit to 1. Afterward, setting the RWAIT bit to 0 and shifting to HALT/STOP mode after checking again by polling that the RWST bit has become 0 (see **Figure 8 - 21**, **Example 2**).

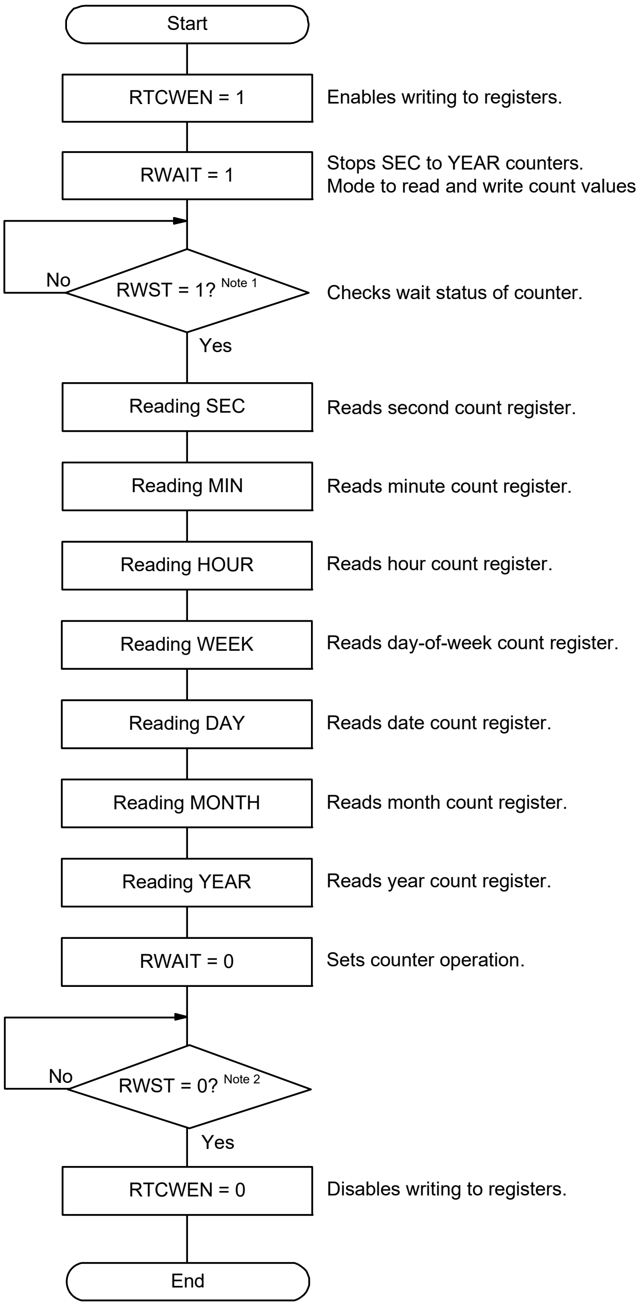
Figure 8 - 21 Procedure for Shifting to HALT/STOP Mode After Setting RTCE bit to 1



8.4.3 Reading the counters of real-time clock 2

Read the counter during counter operation (RTCE = 1) after setting RWAIT to 1 first.
Set RWAIT to 0 after completion of reading the counter.

Figure 8 - 22 Procedure for Reading Real-time Clock 2



Note 1. When the counter is stopped (RTCE = 0), RWST is not set to 1.
Note 2. Be sure to confirm that RWST = 0 before setting STOP mode.

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

<R> **Caution Complete the series of operations of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second. When reading to the counter is required while generation of the alarm interrupt is enabled, first set the CT2 to CT0 bits to 010B (generating the constant-period interrupt once per 1 second). Then, complete the processing from setting the RWAIT bit to 1 to setting it to 0 before generation of the next constant-period interrupt.**

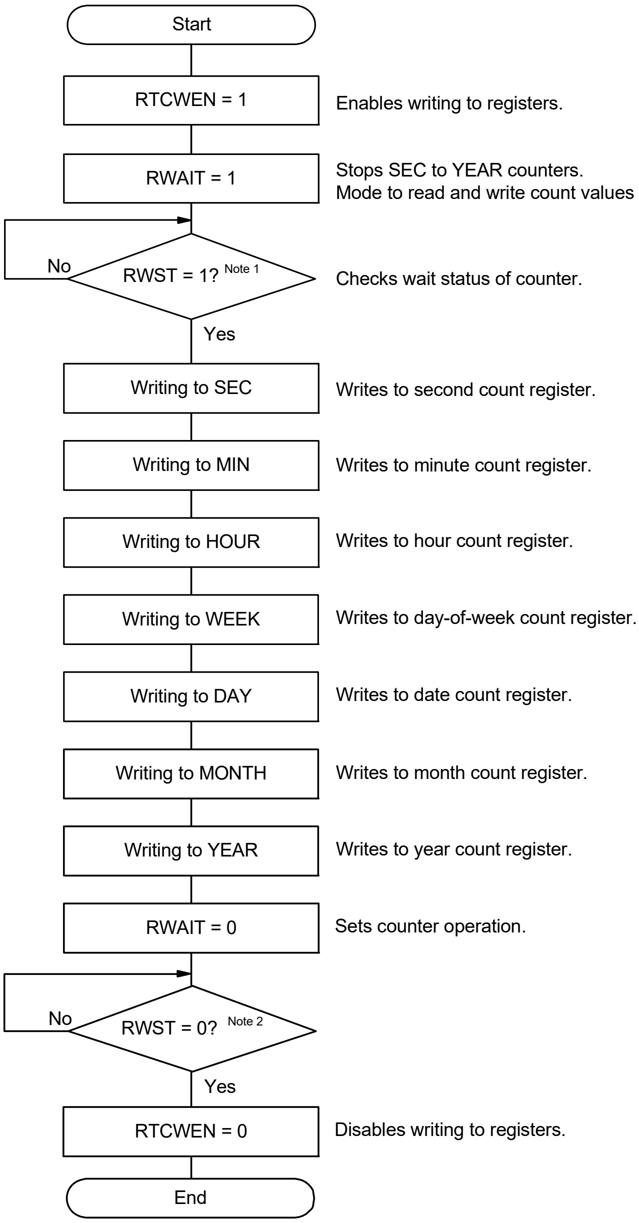
Remark SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR may be read in any sequence. All the registers do not have to be read and only some registers may be read.

8.4.4 Writing to the counters of real-time clock 2

Write to the counter during counter operation (RTCE = 1) after setting RWAIT to 1 first.

Set RWAIT to 0 after completion of writing to the counter.

Figure 8 - 23 Procedure for Writing to Real-time Clock 2



Note 1. When the counter is stopped (RTCE = 0), RWST is not set to 1.

Note 2. Be sure to confirm that RWST = 0 before setting STOP mode.

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

<R> **Caution 1. Complete the series of operations of setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second. When writing to the counter is required while generation of the alarm interrupt is enabled, first set the CT2 to CT0 bits to 010B (generating the constant-period interrupt once per 1 second). Then, complete the processing from setting the RWAIT bit to 1 to setting it to 0 before generation of the next constant-period interrupt.**

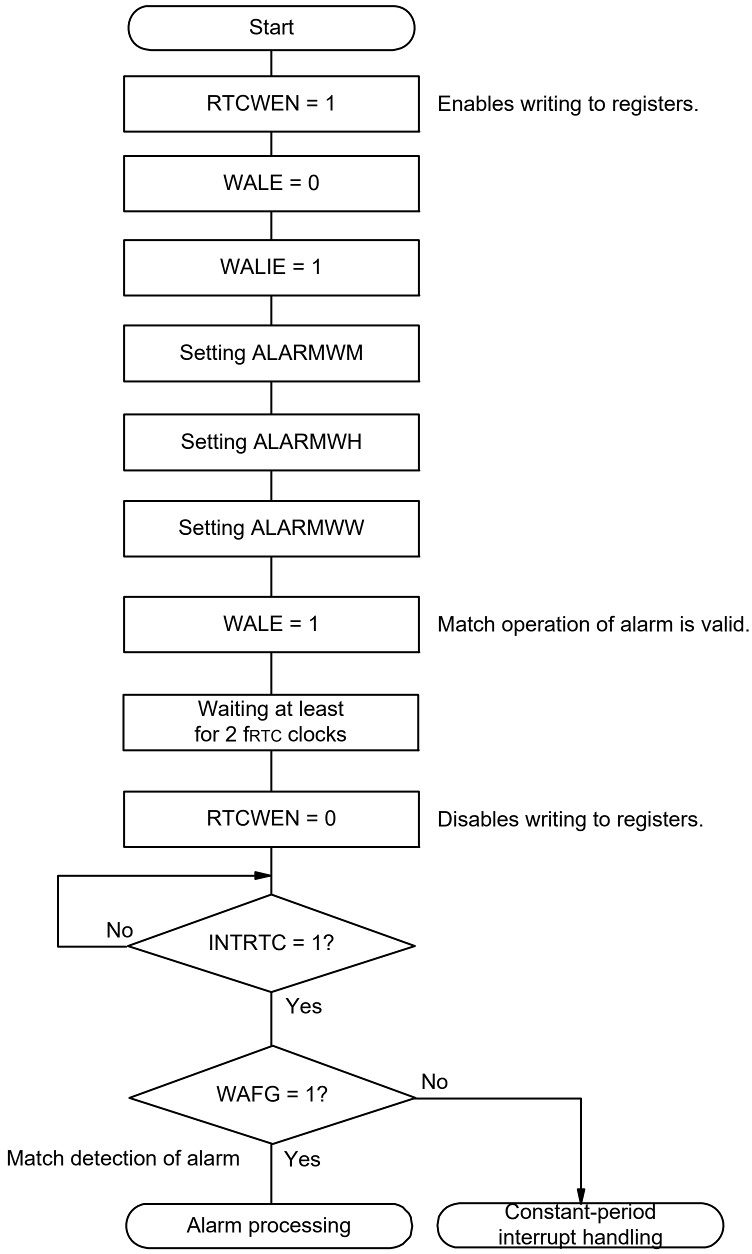
<R> **Caution 2. When changing the values of the SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR registers while counting is in progress (RTCE = 1), rewrite the registers after disabling interrupt processing of INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG and RTCIF flags after rewriting the registers.**

Remark SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR may be written in any sequence. All the registers do not have to be set and only some registers may be written.

8.4.5 Setting alarm of real-time clock 2

Set the alarm time after setting 0 to WALE (alarm operation invalid) first.

Figure 8 - 24 Alarm Setting Procedure

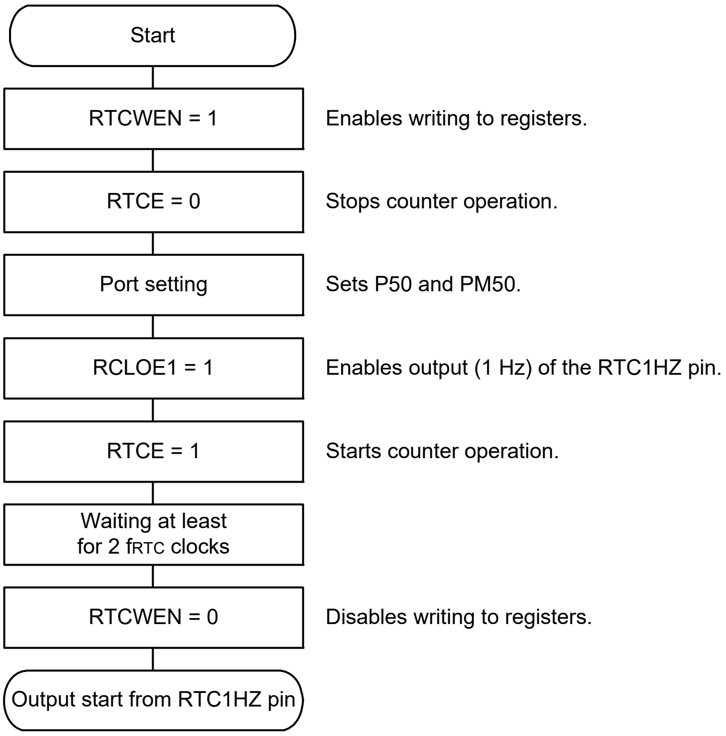


Remark 1. ALARMWM, ALARMWH, and ALARMWW may be written in any sequence.

Remark 2. Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

8.4.6 1 Hz output of real-time clock 2

Figure 8 - 25 1 Hz Output Setting Procedure



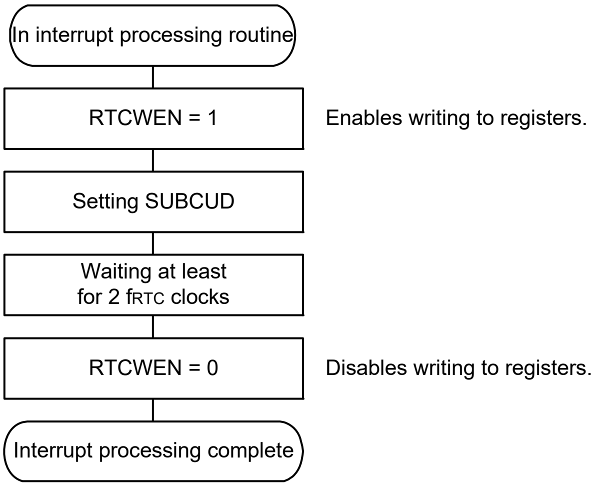
Caution The 1-Hz output function of the real-time clock 2 is not provided in 20-, 24-, 30-, and 32-pin products.

8.4.7 Clock error correction register setting procedure

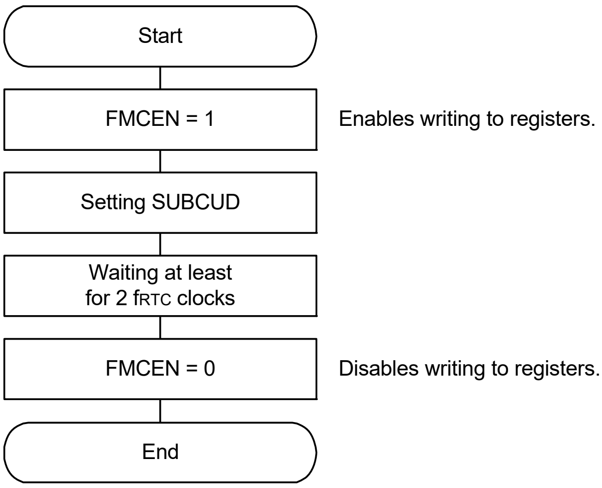
To set the clock error correction register (SUBCUD), perform either of the following procedures.
To prevent writing to the clock register inadvertently, the SUBCUD register should be rewritten by setting FMCEN to enable writing as in step (2).
If rewriting of the clock error correction register (SUBCUD) and the correction timing conflict, the RTC2 may not be corrected properly. To avoid conflicts between the correction timing and rewriting of the SUBCUD register, be sure to start rewriting of the SUBCUD register after generation of the correction timing interrupt (INTRTIT) or constant-period interrupt (INTRTC) that is generated synchronously with the correction timing, and complete rewriting of this register before the next correction timing is generated (0.5 seconds or less).

Caution The process from generation of a correction timing signal interrupt (INTRTIT) to the interrupt response and SUBCUD setting should be completed within 1 second (before the next timing of correction every second).

(1) Set the clock error correction register after setting RTCWEN to 1 first. Then set RTCWEN to 0.



(2) Set the clock error correction register after setting FMCEN to 1 first. Then set FMCEN to 0.



8.4.8 Example of watch error correction of real-time clock 2

The clock can be corrected every second with a minimum resolution and accuracy of 0.96 ppm when it is slow or fast, by setting a value to the clock error correction register.
The following shows how to calculate the target correction value, and how to calculate the F8 to F0 values of the clock error correction register from the target correction value.

Calculating the target correction value 1

(When using output frequency of the RTC1HZ pin)

[Measuring the oscillation frequency]
The oscillation frequency [Note] of each product is measured by outputting 1 Hz from the RTC1HZ pin when the F15 of the watch error correction register (SUBCUD) is set to 0 (stops the watch error correction).

Note See **8.4.6 1 Hz output of real-time clock 2** for the setting procedure of outputting about 1 Hz from the RTC1HZ pin.

[Measuring the oscillation frequency]
(When the output frequency from the RTCCL pin is 0.9999817 Hz)

Oscillation frequency = 32768 × 0.9999817 ≈ 32767.40 Hz

Assume the target frequency to be 32768 Hz. Then the target correction value is calculated as follows.

Target correction value = (Oscillation frequency - Target frequency) ÷ Target frequency
= (32767.40 - 32768.00) ÷ 32768.00
≈ -18.3 ppm

Remark 1. The oscillation frequency is the input clock (f_{RTC}). It can be calculated from the output frequency of the RTC1HZ pin × 32768 when stops the watch error correction.
Remark 2. The target correction value is the oscillation frequency deviation (unit: [ppm]) of the crystal resonator.
Remark 3. The target frequency is the frequency resulting after watch error correction performed.

Calculating the target correction value 2

(When using the frequency measurement circuit)

[Measuring the oscillation frequency]

The oscillation frequency Note of each product is measured using the frequency measurement circuit. The oscillation frequency can be calculated using the following expression.

$$\text{Oscillation frequency} = \frac{f_{MX}\text{ frequency [Hz]} \times \text{Operating trigger division ratio}}{(\text{Frequency measurement count register H and L (FMCRH and FMCRL) value})_{\text{Binary}}} \text{ [Hz]}$$

Note For the procedure to operate the frequency measurement circuit, refer to **9.4.1 Setting Frequency Measurement Circuit**.

[Calculating the target correction value]

(When the value of frequency measurement count register H and L is 9999060D)

- High-speed system clock frequency (f_{MX}) = 10 MHz
- When FMDIV2 to FMDIV0 in the frequency measurement control register = 111B (operating trigger division ratio = 2^{15}), the oscillation frequency is as follows.

Oscillation frequency = f_{MX} frequency [Hz] ×Operating trigger division ratio ÷ (FMCRH, FMCRL) value
= 10 × 106 × 2^{15} ÷ 9999060D
= 32771.0804816 Hz

Assume the target frequency to be 32768 Hz. Then the target correction value is calculated as follows.

Target correction value = Oscillation frequency ÷ Target frequency - 1
= 32771.0804816 ÷ 32768 - 1
≈ 94.0 ppm

Remark 1. The operating trigger division ratio is the f_{SUB} division ratio set by FMDIV2 to FMDIV0 of the frequency measurement control register. The operating trigger division ratio is 28 when FMDIV2 to FMDIV0 = 000B, and 2^{15} when FMDIV2 to FMDIV0 = 111B.

Remark 2. The target correction value is the oscillation frequency deviation (unit: [ppm]) of the crystal resonator.

Remark 3. The target frequency is the frequency resulting after watch error correction performed.

Calculating the F8 to F0 value of the watch error correction register

The F8 to F0 value of the SUBCUD register is calculated from the target correction value by using the following expression.

$$\text{SUBCUD[8:0]} = \left[\frac{\text{Target correction value [ppm]} \times 2^{15}}{10^{6}}\right]_{\text{2's complement (9-bit fixed point type)}} + 0001.00000\text{B}$$

Examples 1. When target correction value = -18.3 [ppm]

SUBCUD[8:0] = $(-18.3 \times 2^{15}/10^{6})_{\text{2's complement (9-bit fixed point type)}}$ + 0001.00000B
= $(-0.59965)_{\text{2's complement (9-bit fixed point type)}}$ + 0001.00000B
= 1111.01101B + 0001.00000B
= 0000.01101B

Examples 2. When target correction value = 94.0 [ppm]

SUBCUD[8:0] = $(94.0 \times 2^{15}/10^{6})_{\text{2's complement (9-bit fixed point type)}}$ + 0001.00000B
= $(+3.08019)_{\text{2's complement (9-bit fixed point type)}}$ + 0001.00000B
= 0011.00011B + 0001.00000B
= 0100.00011B

CHAPTER 9 FREQUENCY MEASURE CIRCUIT

9.1 Frequency Measurement Circuit

The frequency measurement circuit is used to measure the frequency of the subsystem clock generator and RTC2/other clock (f_{SXR}) or low-speed on-chip oscillator clock (f_{IL}), by inputting the high-accuracy reference clock externally.

9.2 Configuration of Frequency Measurement Circuit

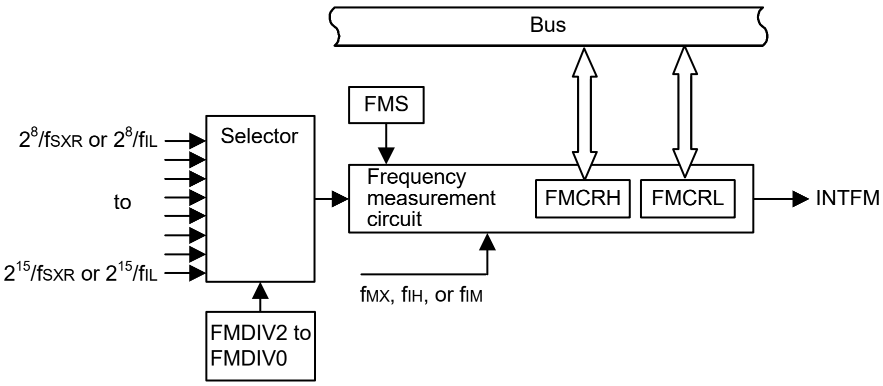
The frequency measurement circuit includes the following hardware.

Table 9 - 1 Configuration of Frequency Measurement Circuit

Item	Configuration
Counter	Counter (32-bit)
Control registers	Peripheral enable register 2 (PER2)
	Subsystem clock supply mode control register (OSMC)
	Frequency measurement count register L (FMCRL)
	Frequency measurement count register H (FMCRH)
	Frequency measurement control register (FMCTL)
	Frequency measurement clock select register (FMCKS)

Figure 9 - 1 shows the Frequency Measurement Circuit Diagram.

Figure 9 - 1 Frequency Measurement Circuit Diagram



9.3 Registers Controlling Frequency Measurement Circuit

The frequency measurement circuit is controlled by the following registers.

- Peripheral enable register 2 (PER2)
- Subsystem clock supply mode control register (OSMC)
- Frequency measurement count register L (FMCRL)
- Frequency measurement count register H (FMCRH)
- Frequency measurement control register (FMCTL)
- Frequency measurement clock select register (FMCKS)

9.3.1 Peripheral enable register 2 (PER2)

This register is used to enable or disable supplying the clock to the register used for the frequency measurement circuit. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

Of the registers that are used to control the frequency measurement circuit and the real-time clock 2, the clock error correction register (SUBCUD) can be set by setting bit 6 (FMCEN) of this register to 1.

The PER2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 9 - 2 Format of Peripheral enable register 2 (PER2)

Address: F00FCH After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
PER2	TMKAEN	FMCEN	DOCEN	0	0	0	0	0

FMCEN	Frequency measurement circuit
0	Stops input clock supply. • SFR used by the frequency measurement circuit cannot be written. • SUBCUD register used by the real-time clock 2 cannot be written. • The frequency measurement circuit is in the reset status.
1	Enables input clock supply. • SFR used by the frequency measurement circuit can be read/written. • SUBCUD register used by the real-time clock 2 can be read and written.

Caution 1. The clock error correction register (SUBCUD) can be read or written by setting RTCWEN of peripheral enable register 0 (PER0) to 1 or setting FMCEN of peripheral enable register 2 (PER2) to 1.

Caution 2. Be sure to set bits 0 to 4 to 0.

9.3.2 Subsystem clock supply mode control register (OSMC)

This register is used to reduce power consumption by stopping unnecessary clock functions.
If the RTCLPC bit is set to 1, power consumption can be reduced, because clock supply to the peripheral functions other than the real-time clock 2, 12-bit interval timer, 8-bit interval timer, and clock output/buzzer output controller is stopped in STOP mode or in HALT mode while the subsystem clock is selected as the CPU clock.
In addition, the OSMC register is used to select the subsystem clock generator and RTC2/other clock (f_{SXR}) or low-speed on-chip oscillator clock (f_{IL}) as the operating clock of the real-time clock 2, 12-bit interval timer, clock output/buzzer output controller, 8-bit interval timer, and frequency measurement circuit.
The OSMC register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 9 - 3 Format of Subsystem clock supply mode control register (OSMC)

Address: F00F3H After reset: Undefined R/W Note 1

Symbol	<7>	6	5	<4>	3	2	1	0
OSMC	RTCLPC	0	0	WUTMMCK0	x	x	0	0

RTCLPC Note 7	Setting in STOP mode or HALT mode while subsystem clock is selected as CPU clock
0	Enables supply of subsystem clock to peripheral functions (See **Tables 23 - 1** to **23 - 4** for peripheral functions whose operations are enabled.)
1	Stops supply of subsystem clock to peripheral functions other than the real-time clock 2, 12-bit interval timer, 8-bit interval timer, and clock output/buzzer output controller.

WUTMMCK0	Selection of operation clock for real-time clock 2, frequency measurement circuit, 12-bit interval timer, 8-bit interval timer, and clock output/buzzer output controller
0	• Subsystem clock Note 2
1	• Low-speed on-chip oscillator clock Notes 3, 4, 5, 6

FMTRGSEL Note 10	WUTMMCK0	Operating clock selection for frequency measurement circuit count operation/stop trigger clock and real-time clock 2
0	0	fSX selected for the frequency measurement circuit/real-time clock 2
0	1	fIL selected for the real-time clock 2 (constant-period interrupt function) Note 8
1	0	Setting prohibited
1	1	fIL selected for the frequency measurement circuit Note 9

Note 1. Be sure to set bits 0, 1, 5, and 6 to 0. Bits 2 and 3 are read-only, write is ignored.

Note 2. Do not set the WUTMMCK0 bit to 0 and the FMTRGSEL bit in the FMCKS register to 1.

Note 3. Do not set the WUTMMCK0 bit to 1 while the sub clock is oscillating.

Note 4. Switching between the subsystem clock and the low-speed on-chip oscillator clock can be enabled by the WUTMMCK0 bit only when all of the real-time clock 2, frequency measurement circuit, 12-bit interval timer, 8-bit interval timer, and clock output/buzzer output functions are stopped.

Note 5. fIL can be selected as the operating clock of the real-time clock 2 when the WUTMMCK0 bit is 1 and the FMTRGSEL bit in the FMCKS register is 0. Also, only the constant-period interrupt function of the real-time clock 2 can be used at this time; the clock count function cannot be used.

Note 6. fIL can be selected as the operating clock of the frequency measurement circuit when the WUTMMCK0 bit is 1 and the FMTRGSEL bit in the FMCKS register is 1.

Note 7. When the sub clock is selected (SELLOSC = 0) by bit 0 (SELLOSC) of the CKSEL register and RTCLPC is set to 1, the subsystem clock is stopped. However, when the low-speed on-chip oscillator clock is selected (SELLOSC = 1) and RTCLPC is set to 1, the subsystem clock is not stopped.

Note 8. The frequency measurement function cannot be used.

Note 9. The real-time clock 2 cannot be used.

Note 10. Bit 4 of the frequency measurement circuit clock select register (FMCKS)

Remark x: Undefined

9.3.3 Frequency measurement count register L (FMCRL)

This register represents the lower 16 bits of the frequency measurement count register (FMCR) in the frequency measurement circuit.

The FMCRL register can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears the FMCRL register to 0000H.

Figure 9 - 4 Format of Frequency measurement count register L (FMCRL)

Address:F0312H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FMCRL																

Caution 1. Do not read the value of FMCRL when FMS = 1.
Caution 2. Read the value of FMCRL after the frequency measurement complete interrupt is generated.

9.3.4 Frequency measurement count register H (FMCRH)

This register represents the upper 16 bits of the frequency measurement count register (FMCR) in the frequency measurement circuit.

The FMCRH register can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears the FMCRH register to 0000H.

Figure 9 - 5 Format of Frequency measurement count register H (FMCRH)

Address:F0314H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FMCRH																

Caution 1. Do not read the value of FMCRH when FMS = 1.
Caution 2. Read the value of FMCRH after the frequency measurement complete interrupt is generated.

Figure 9 - 6 Frequency Measurement Count Register (FMCRH, FMCRL)

	31 – 16	15 – 0
FMCR	FMCRH	FMCRL

9.3.5 Frequency measurement control register (FMCTL)

The FMCTL register is used to set the operation of the frequency measurement circuit. This register is used to start operation and set the period of frequency measurement.

The FMCTL register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the FMCTL register to 00H.

Figure 9 - 7 Format of Frequency measurement control register (FMCTL)

Address:F0316H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
FMCTL	FMS	0	0	0	0	FMDIV2	FMDIV1	FMDIV0

FMS	Frequency measurement circuit operation enable
0	Stops the frequency measurement circuit.
1	Operates the frequency measurement circuit. Starts counting on the rising edge of the operating clock and stops counting on the next rising edge of the operating clock.

FMDIV2	FMDIV1	FMDIV0	Frequency measurement period setting
0	0	0	$2^8/f_{SXR}$ or $2^8/f_{IL}$ (7.8125 ms)
0	0	1	$2^9/f_{SXR}$ or $2^9/f_{IL}$ (15.625 ms)
0	1	0	$2^{10}/f_{SXR}$ or $2^{10}/f_{IL}$ (31.25 ms)
0	1	1	$2^{11}/f_{SXR}$ or $2^{11}/f_{IL}$ (62.5 ms)
1	0	0	$2^{12}/f_{SXR}$ or $2^{12}/f_{IL}$ (0.125 s)
1	0	1	$2^{13}/f_{SXR}$ or $2^{13}/f_{IL}$ (0.25 s)
1	1	0	$2^{14}/f_{SXR}$ or $2^{14}/f_{IL}$ (0.5 s)
1	1	1	$2^{15}/f_{SXR}$ or $2^{15}/f_{IL}$ (1s)

Caution Do not read the value of the FMDIV2 to FMDIV0 bits when FMS = 1.

Remark The frequency measurement resolution can be calculated by the formula below.

- Frequency measurement resolution = 10^6/(frequency measurement period × reference clock frequency (f_{MX}) [Hz]) [ppm]

  Example 1) When FMDIV2 to FMDIV0 = 000B and f_{MX} = 20 MHz, measurement resolution = 6.4 ppm

  Example 2) When FMDIV2 to FMDIV0 = 111B and f_{MX} = 1 MHz, measurement resolution = 1 ppm

9.3.6 Frequency measurement clock select register (FMCKS)

The FMCKS register is used to select the operating clock and frequency count clock to be input to the frequency measurement circuit.

The FMCKS register can be used by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the FMCKS register to 00H.

Figure 9 - 8 Format of Frequency measurement clock select register (FMCKS)

Address: F007AH After reset: 00H R/W

Symbol	7	6	5	<4>	3	2	1	0
FMCKS	0	0	0	FMTRGSEL	0	0	FMCKSEL1	FMCKSEL0

FMTRGSEL	WUTMMCK0 Note 3	Selection of frequency measurement circuit count operation/ stop trigger clock/real-time clock 2 operating clock
0	0	f_{SXR} selected for the frequency measurement circuit/real-time clock 2
0	1	f_{IL} selected for the real-time clock 2 (constant-period interrupt function) Note 1
1	0	Setting prohibited
1	1	f_{IL} selected for the frequency measurement circuit Note 2

FMCKSEL1	FMCKSEL0	Selection of frequency count clock
0	0	f_{MX} selected
0	1	f_{IM} selected
1	×	f_{IH} selected

Note 1. The frequency measurement function cannot be used.

Note 2. The real-time clock 2 cannot be used.

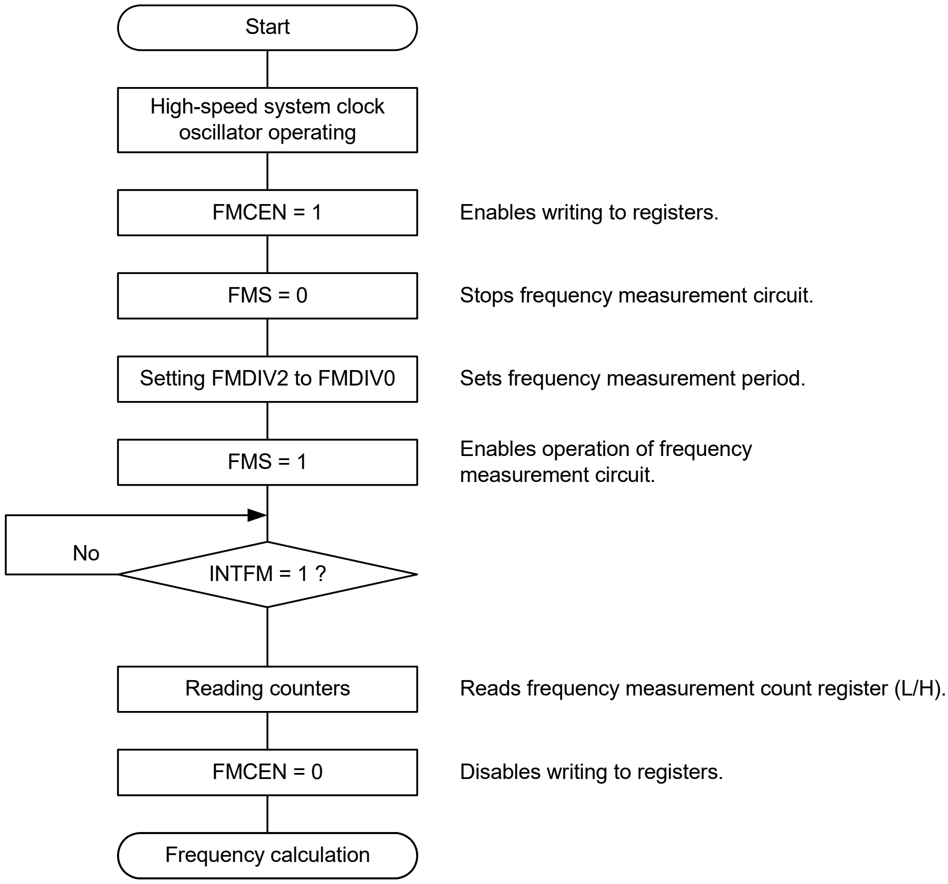
Note 3. Refer to the description on the WUTMMCK0 bit in the OSMC register.

9.4 Frequency Measurement Circuit Operation

9.4.1 Setting Frequency Measurement Circuit

Set frequency measurement circuit after setting 0 to FMS first.

Figure 9 - 9 Procedure for Setting Frequency Measurement Circuit Using Reference Clock



Caution After the frequency measurement count register (L/H) is read, be sure to set FMCEN to 0.

The f_{SXR} or f_{IL} oscillation frequency is calculated by using the following expression.

$$f_{SXR} \text{ or } f_{IL} \text{ oscillation frequency} = \frac{\text{Reference clock frequency [Hz]} \times \text{operation trigger division ratio}}{\text{Frequency measurement count register value (FMCR)}} \text{ [Hz]}$$

For example, when the frequency is measured under the following conditions

- Count clock frequency: f_{MX} = 10 MHz
- Frequency measurement period setting register: FMDIV2 to FMDIV0 = 111B (operation trigger division ratio: 2^{15})

and the measurement result is as follows,

- Frequency measurement count register: FMCR = 10000160D

the f_{SXR} or f_{IL} oscillation frequency is obtained as below.

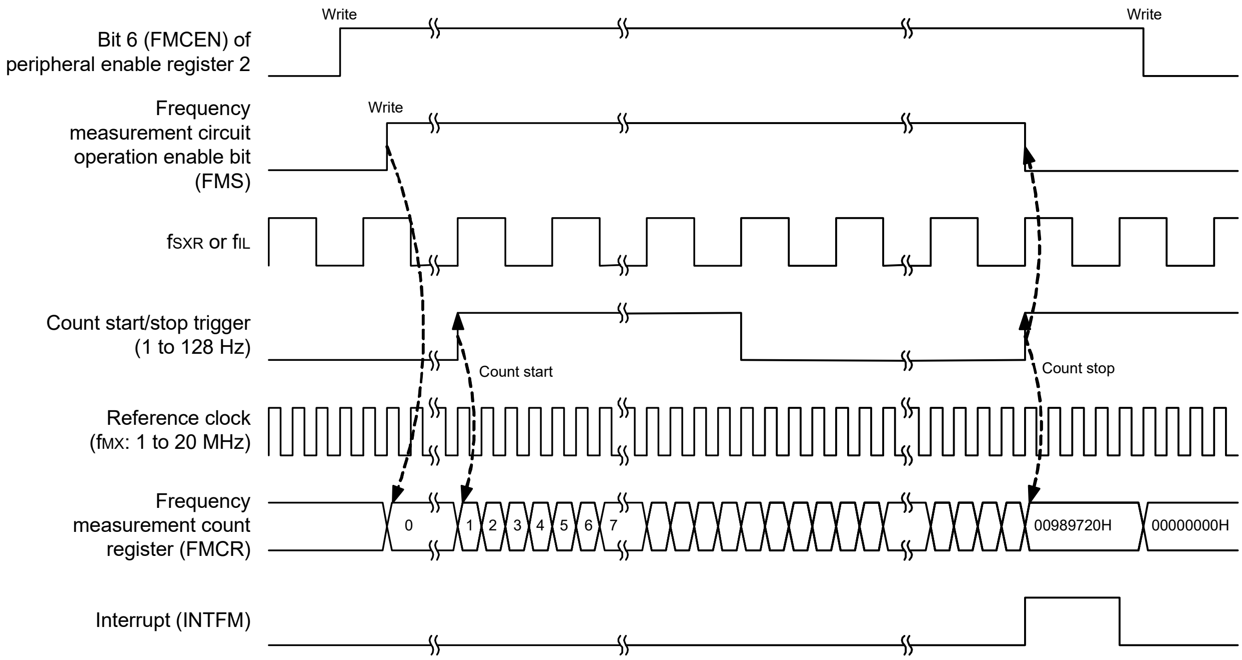
$$f_{SXR} \text{ or } f_{IL} \text{ oscillation frequency} = \frac{(10 \times 10^6) \times 2^{15}}{10000160} = 32767.47572 \text{ [Hz]}$$

9.4.2 Frequency Measurement Circuit Operation Timing

The operation timing of the frequency measurement circuit is shown in Figure 9 - 10.

After the frequency measurement circuit operation enable bit (FMS) is set to 1, counting is started by the count start trigger set with the frequency measurement period setting bits (FMDIV2 to FMDIV0) and stopped by the next trigger. After counting is stopped, the count value is retained, and the frequency measurement circuit operation enable bit (FMS) is reset to 0. An interrupt is also generated for one clock of fSUB. After the operation of the frequency measurement circuit is completed (FMS = 0) and the frequency measurement count register (L/H) is read, be sure to set bit 6 (FMCEN) of peripheral enable register 2 to 0.

Figure 9 - 10 Frequency Measurement Circuit Operation Timing



CHAPTER 10 12-BIT INTERVAL TIMER

10.1 Functions of 12-bit Interval Timer

An interrupt (INTIT) is generated at any previously specified time interval. It can be utilized for wakeup from STOP mode and triggering an A/D converter's SNOOZE mode.

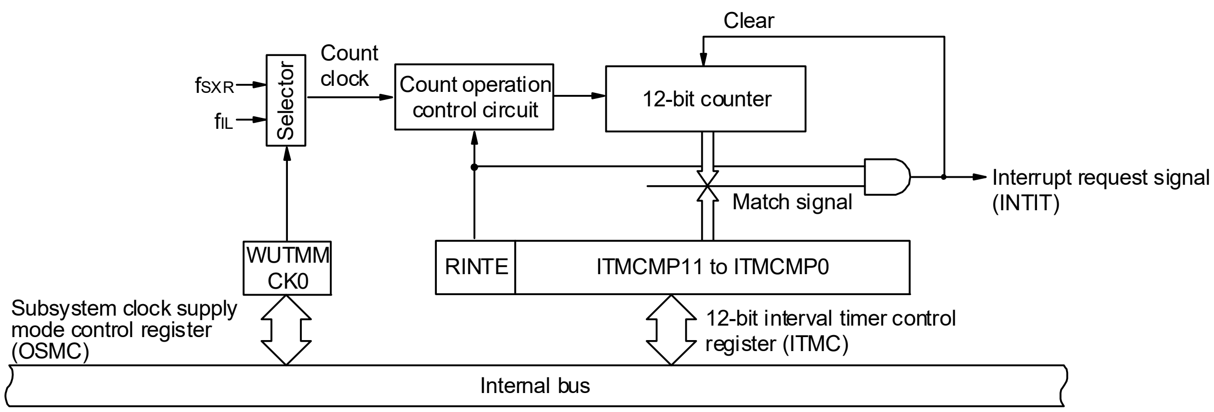
10.2 Configuration of 12-bit Interval Timer

The 12-bit interval timer includes the following hardware.

Table 10 - 1 Configuration of 12-bit Interval Timer

Item	Configuration
Counter	12-bit counter
Control registers	Peripheral enable register 2 (PER2)
	Peripheral reset control register 2 (PRR2)
	Subsystem clock supply mode control register (OSMC)
	12-bit interval timer control register (ITMC)

Figure 10 - 1 Block Diagram of 12-bit Interval Timer



Caution The subsystem clock generator and RTC/other clock (f_{SXR}) can be selected as the count clock in 30 and 48-pin products.

10.3 Registers Controlling 12-bit Interval Timer

The 12-bit interval timer is controlled by the following registers.

- Peripheral enable register 2 (PER2)
- Peripheral reset control register 2 (PRR2)
- Subsystem clock supply mode control register (OSMC)
- 12-bit interval timer control register (ITMC)

10.3.1 Peripheral enable register 2 (PER2)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.
When the 12-bit interval timer is used, be sure to set bit 7 (TMKAEN) of this register to 1.
The PER2 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 10 - 2 Format of Peripheral enable register 2 (PER2)

Address: F00FCH After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
PER2	TMKAEN	FMCEN	DOCEN	0	0	0	0	0

TMKAEN	Control of 12-bit interval timer input clock supply
0	Stops input clock supply. • SFR used by the 12-bit interval timer cannot be written. The read value is 0H. However, the SFR is not initialized. Note
1	Enables input clock supply. • SFR used by the 12-bit interval timer can be read and written.

Note To initialize the 12-bit interval timer and the SFR used by the 12-bit interval timer, use bit 7 (TMKARES) of PRR2.

Caution 1. Be sure to clear the following bits to 0.
Bits 0 to 4

Caution 2. Do not change the target bit in the PER2 register while operation of each peripheral function is enabled. Change the setting specified by PER2 while operation of each peripheral function assigned to PER2 is stopped (except for FMCEN).

10.3.2 Peripheral reset control register 2 (PRR2)

This register is used for individual reset control of each peripheral hardware.
This MCU controls reset and reset release of each peripheral hardware supported by the PRR2 register.
To reset the 12-bit interval timer, be sure to set bit 7 (TMKARES) to 1.
The PRR2 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 10 - 3 Format of Peripheral reset control register 2 (PRR2)

Address: F00FDH After reset: 00H R/W

Symbol	<7>	6	<5>	4	3	2	1	0
PRR2	TMKARES	0	DOCRES	0	0	0	0	0

TMKARES	Reset control of 12-bit interval timer
0	12-bit interval timer reset release
1	12-bit interval timer reset state

10.3.3 Subsystem clock supply mode control register (OSMC)

The WUTMMCK0 bit can be used to select the 12-bit interval timer operation clock.
In addition, by stopping clock functions that are unnecessary, the RTCLPC bit can be used to reduce power consumption. For details about setting the RTCLPC bit, see **CHAPTER 6 CLOCK GENERATOR**.
The OSMC register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 10 - 4 Format of Subsystem clock supply mode control register (OSMC)

Address: F00F3H After reset: Undefined R/W Note 1

Symbol	<7>	6	5	<4>	3	2	1	0
OSMC	RTCLPC	0	0	WUTMMCK0	x	x	0	0

RTCLPC Note 7	Setting in STOP mode or HALT mode while subsystem clock is selected as CPU clock
0	Enables supply of subsystem clock to peripheral functions (See **Tables 23 - 1** to **23 - 4** for peripheral functions whose operations are enabled.)
1	Stops supply of subsystem clock to peripheral functions other than the real-time clock 2, 12-bit interval timer, 8-bit interval timer, and clock output/buzzer output controller.

WUTMMCK0	Selection of operation clock for real-time clock 2, frequency measurement circuit, 12-bit interval timer, 8-bit interval timer, and clock output/buzzer output controller
0	• Subsystem clock Note 2
1	• Low-speed on-chip oscillator clock Notes 3, 4, 5, 6

FMTRGSEL Note 10	WUTMMCK0	Operating clock selection for frequency measurement circuit count operation/stop trigger clock and real-time clock 2
0	0	f_{SX} selected for the frequency measurement circuit/real-time clock 2
0	1	f_{IL} selected for the real-time clock 2 (constant-period interrupt function) Note 8
1	0	Setting prohibited
1	1	f_{IL} selected for the frequency measurement circuit Note 9

Note 1. Be sure to set bits 0, 1, 5, and 6 to 0. Bits 2 and 3 are read-only, write is ignored.

Note 2. Do not set the WUTMMCK0 bit to 0 and the FMTRGSEL bit in the FMCKS register to 1.

Note 3. Do not set the WUTMMCK0 bit to 1 while the sub clock is oscillating.

Note 4. Switching between the subsystem clock and the low-speed on-chip oscillator clock can be enabled by the WUTMMCK0 bit only when all of the real-time clock 2, frequency measurement circuit, 12-bit interval timer, 8-bit interval timer, and clock output/buzzer output functions are stopped.

Note 5. f_{IL} can be selected as the operating clock of the real-time clock 2 when the WUTMMCK0 bit is 1 and the FMTRGSEL bit in the FMCKS register is 0. Also, only the constant-period interrupt function of the real-time clock 2 can be used at this time; the clock count function cannot be used.

Note 6. $f_{IL}/2$ can be selected as the operating clock of the frequency measurement circuit when the WUTMMCK0 bit is 1 and the FMTRGSEL bit in the FMCKS register is 1.

Note 7. When the sub clock is selected (SELLOSC = 0) by bit 0 (SELLOSC) of the CKSEL register and RTCLPC is set to 1, the subsystem clock is stopped. However, when the low-speed on-chip oscillator clock is selected (SELLOSC = 1) and RTCLPC is set to 1, the subsystem clock is not stopped.

Note 8. The frequency measurement function cannot be used.

Note 9. The real-time clock 2 cannot be used.

Note 10. Bit 4 of the frequency measurement circuit clock select register (FMCKS)

Remark x: Undefined

10.3.4 12-bit interval timer control register (ITMC)

This register is used to set up the starting and stopping of the 12-bit interval timer operation and to specify the timer compare value.

The ITMC register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0FFFH.

Figure 10 - 5 Format of 12-bit interval timer control register (ITMC)

Address: FFF90H After reset: 0FFFH R/W

Symbol	15	14	13	12	11 to 0
ITMC	RINTE	0	0	0	ITCMP11 to ITCMP0

RINTE	12-bit interval timer operation control
0	Count operation stopped (count clear)
1	Count operation started

ITCMP11 to ITCMP0	Specification of the 12-bit interval timer compare value
001H	These bits generate a fixed-cycle interrupt (count clock cycles x (ITCMP setting + 1)).
• • •	
FFFH	
000H	Setting prohibit

Example interrupt cycles when 001H or FFFH is specified for ITCMP11 to ITCMP0
- ITCMP11 to ITCMP0 = 001H, count clock: when f_{SUB} = 32.768 kHz
  1/32.768 [kHz] × (1 + 1) = 0.06103515625 [ms] ≅ 61.03 [μs]
- ITCMP11 to ITCMP0 = FFFH, count clock: when f_{SUB} = 32.768 kHz
  1/32.768 [kHz] × (4095 + 1) = 125 [ms]

Caution 1. Before changing the RINTE bit from 1 to 0, use the interrupt mask flag register to disable the INTIT interrupt servicing. When the operation starts (from 0 to 1) again, clear the ITIF flag, and then enable the interrupt servicing.

Caution 2. The value read from the RINTE bit is applied one count clock cycle after setting the RINTE bit.

Caution 3. When setting the RINTE bit after returned from standby mode and entering standby mode again, confirm that the written value of the RINTE bit is reflected, or wait that more than one clock of the count clock has elapsed after returned from standby mode. Then enter standby mode.

Caution 4. Only change the setting of the ITCMP11 to ITCMP0 bits when RINTE = 0.
However, it is possible to change the settings of the ITCMP11 to ITCMP0 bits at the same time as when changing RINTE from 0 to 1 or 1 to 0.

10.4 12-bit Interval Timer Operation

10.4.1 12-bit interval timer operation timing

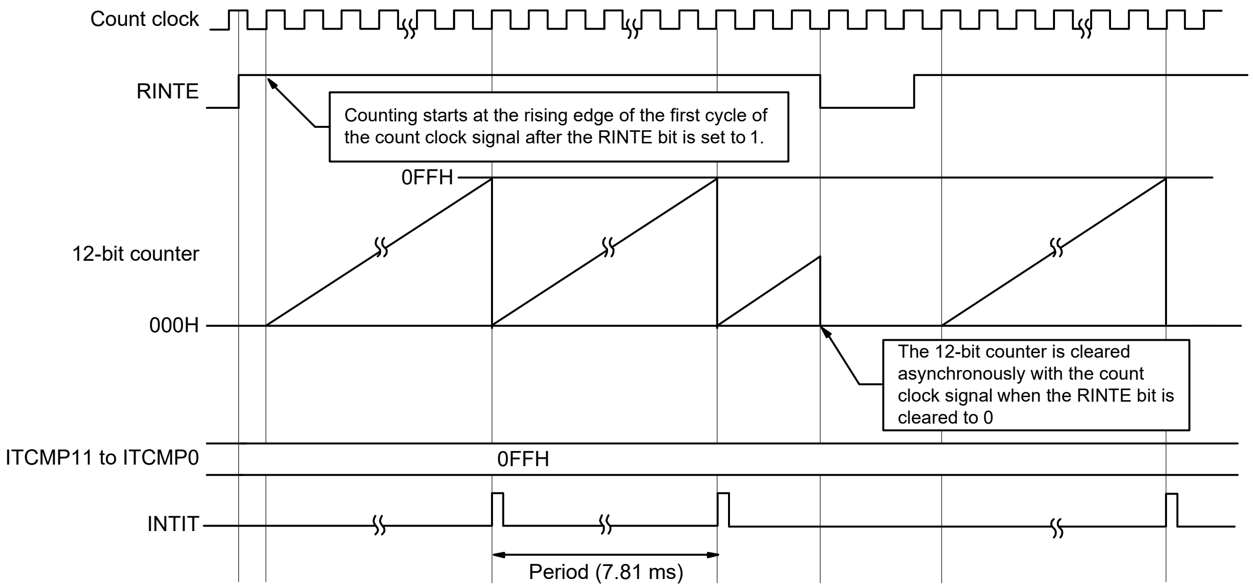
The count value specified for the ITCMP11 to ITCMP0 bits is used as an interval to operate an 12-bit interval timer that repeatedly generates interrupt requests (INTIT).

When the RINTE bit is set to 1, the 12-bit counter starts counting.

When the 12-bit counter value matches the value specified for the ITCMP11 to ITCMP0 bits, the 12-bit counter value is cleared to 0, counting continues, and an interrupt request signal (INTIT) is generated at the same time.

The basic operation of the 12-bit interval timer is as follows.

Figure 10 - 6 12-bit Interval Timer Operation Timing (ITCMP11 to ITCMP0 = 0FFH, count clock: f_{SUB} = 32.768 kHz)

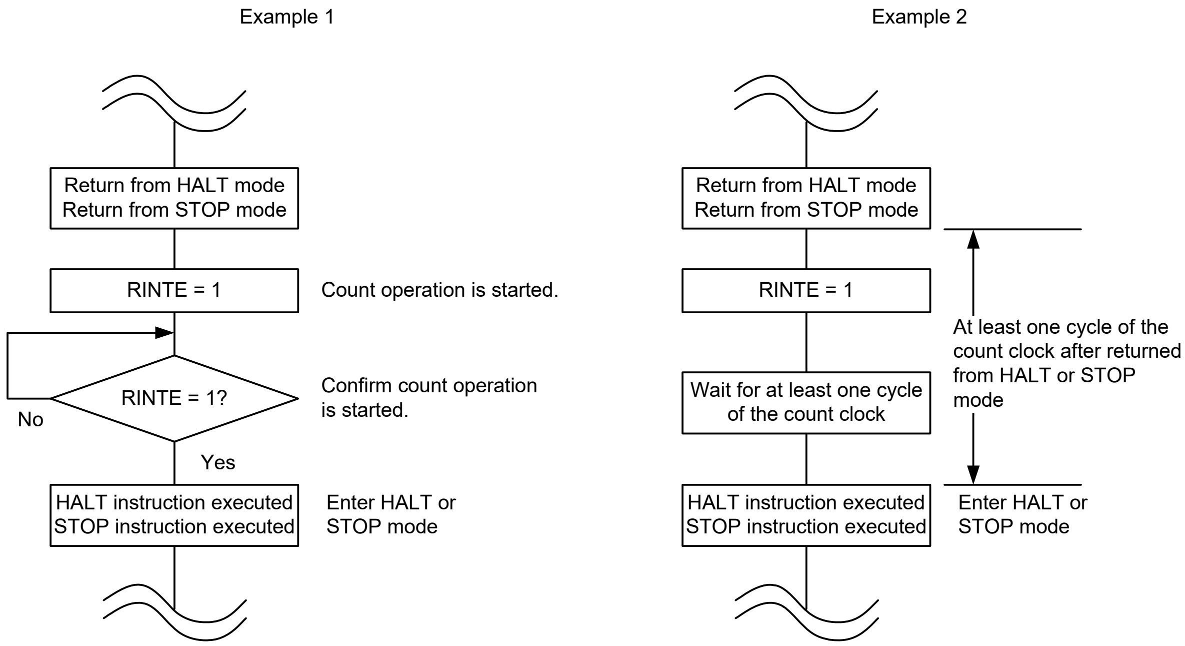


10.4.2 Start of count operation and re-enter to HALT/STOP mode after returned from HALT/STOP mode

When setting the RINTE bit after returned from HALT or STOP mode and entering HALT or STOP mode again, write 1 to the RINTE bit, and confirm the written value of the RINTE bit is reflected or wait for at least one cycle of the count clock.
Then, enter HALT or STOP mode.

- After setting RINTE to 1, confirm by polling that the RINTE bit has become 1, and then enter HALT or STOP mode (see **Example 1** in **Figure 10 - 7**).
- After setting RINTE to 1, wait for at least one cycle of the count clock and then enter HALT or STOP mode (see **Example 2** in **Figure 10 - 7**).

Figure 10 - 7 Procedure of entering to HALT or STOP mode after setting RINTE to 1



CHAPTER 11 8-BIT INTERVAL TIMER

The 8-bit interval timer has two 8-bit timers (channel 0 and channel 1) which operate independently. These timers can be connected to operate as a 16-bit timer.
The 8-bit interval timer contains two units, 8-bit interval timer_0 and 8-bit interval timer_1, which have the same function. This chapter describes these units as the 8-bit interval timer unless there are differences among them.

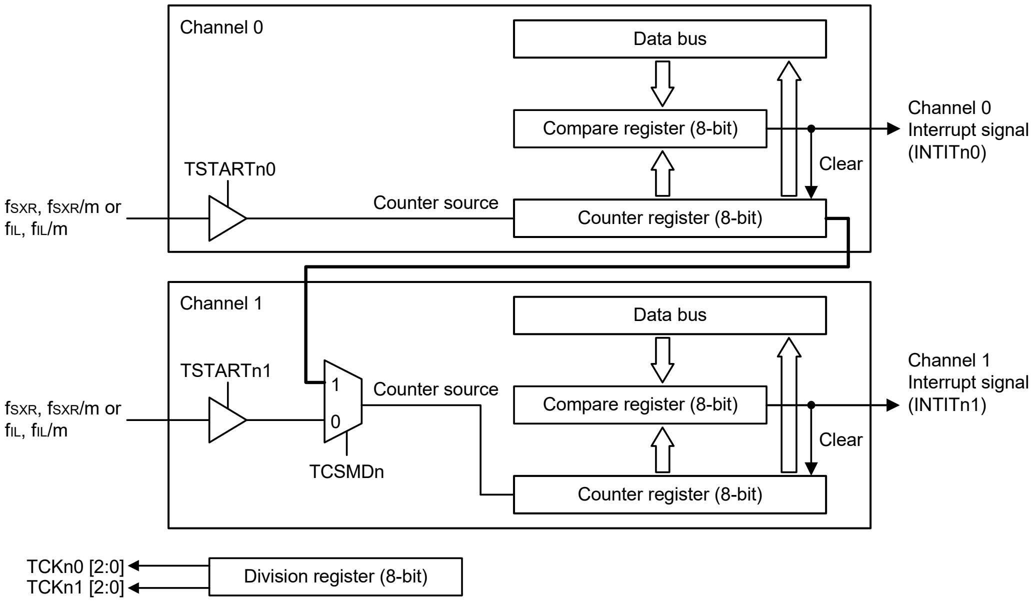
11.1 Overview

The 8-bit interval timer is an 8-bit timer that operates using the f_{SXR} or f_{IL} clock that is asynchronous with the CPU. Table 11 - 1 lists the 8-Bit Interval Timer Specifications and Figure 11 - 1 shows the 8-Bit Interval Timer Block Diagram.

Table 11 - 1 8-Bit Interval Timer Specifications

Item	Description
Count source (operating clock)	• f_{SXR}, $f_{SXR}/2$, $f_{SXR}/4$, $f_{SXR}/8$, $f_{SXR}/16$, $f_{SXR}/32$, $f_{SXR}/64$, $f_{SXR}/128$ • f_{IL}, $f_{IL}/2$, $f_{IL}/4$, $f_{IL}/8$, $f_{IL}/16$, $f_{IL}/32$, $f_{IL}/64$, $f_{IL}/128$
Operating mode	• 8-bit counter mode Channel 0 and channel 1 operate independently as an 8-bit counter • 16-bit counter mode Channel 0 and channel 1 are connected to operate as a 16-bit counter
Interrupt	• Output when the counter matches the compare value

Figure 11 - 1 8-Bit Interval Timer Block Diagram



TSTARTni (i = 0, 1), TCSMDn, TCLKENn: Bits in TRTCRn register

TCKni [2:0]: Bit in TRTMDn register

Remark m = 2, 4, 8, 16, 32, 64, 128

n = 0, 1

11.2 I/O Pins

The 8-bit interval timer does not have an I/O pin.

11.3 Registers

Table 11 - 2 lists the 8-Bit Interval Timer Register Configuration.

Table 11 - 2 Registers

Register Name	Symbol
8-bit interval timer counter register 00	TRT00 Note 1
8-bit interval timer counter register 01	TRT01 Note 1
8-bit interval timer counter register 0	TRT0 Note 2
8-bit interval timer compare register 00	TRTCMP00 Note 1
8-bit interval timer compare register 01	TRTCMP01 Note 1
8-bit interval timer compare register 0	TRTCMP0 Note 2
8-bit interval timer control register 0	TRTCR0
8-bit interval timer division register 0	TRTMD0
8-bit interval timer counter register 10	TRT10 Note 1
8-bit interval timer counter register 11	TRT11 Note 1
8-bit interval timer counter register 1	TRT1 Note 2
8-bit interval timer compare register 10	TRTCMP10 Note 1
8-bit interval timer compare register 11	TRTCMP11 Note 1
8-bit interval timer compare register 1	TRTCMP1 Note 2
8-bit interval timer control register 1	TRTCR1
8-bit interval timer division register 1	TRTMD1

Note 1. Can be accessed only when the TCSMDn bit in the TRTCRn register = 0.
Note 2. Can be accessed only when the TCSMDn bit in the TRTCRn register = 1.

Remark n = 0, 1

11.3.1 8-bit interval timer counter register ni (TRTni) (n = 0, 1, i = 0, 1)

This is the 8-bit interval timer counter register. It is used as a counter that counts up based on the count clock.
The TRTni register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 11 - 2 Format of 8-bit interval timer counter register ni (TRTni)

Address: F0500H (TRT00), F0501H (TRT01), F0508H (TRT10), F0509H (TRT11) After reset: 00H R Notes 1, 2

Symbol	7	6	5	4	3	2	1	0
TRTni								

Note 1. The TRTni register is set to 00H two cycles of the count clock after the compare register TRTCMPni is write-accessed. Refer to **11.4.4 Timing for Updating Compare Register Values**.

Note 2. Can be accessed only when the mode select bit (TCSMDn) in the 8-bit interval timer control register n (TRTCRn) is 0.

11.3.2 8-bit interval timer counter register n (TRTn) (n = 0, 1)

This is a 16-bit counter register when the 8-bit interval timer is used in 16-bit interval timer mode.
The TRTn register can be set by a 16-bit memory manipulation instruction.
Reset signal generation sets this register to 0000H.

Figure 11 - 3 Format of 8-bit interval timer counter register n (TRTn)

Address: F0500H (TRT0), F0508H (TRT1) After reset: 0000H R Notes 1, 2

	F0501H (TRT01) F0509H (TRT11)								F0500H (TRT00) F0508H (TRT10)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRTn																

Note 1. The TRTn register is set to 0000H two cycles of the count clock after the compare register TRTCMPn is write-accessed. Refer to **11.4.4 Timing for Updating Compare Register Values**.

Note 2. Can be accessed only when the mode select bit (TCSMDn) in the 8-bit interval timer control register n (TRTCRn) is 0.

11.3.3 8-bit interval timer compare register ni (TRTCMPni) (n = 0, 1, i = 0, 1)

This is the 8-bit interval timer compare value register.
The TRTCMPni register can be set by an 8-bit memory manipulation instruction.
Reset signal generation sets this register to FFH.
The setting range is 01H to FFH Note 1.
This register is used to store the compare value of registers TRTn0 and TRTn1 (counters).
Write-access clears the count value (TRTn0, TRTn1) to 00H.
Refer to **11.4.4 Timing for Updating Compare Register Values** for the timing of rewriting the compare value.

Figure 11 - 4 Format of 8-bit interval timer compare register ni (TRTCMPni)

Address: F0350H (TRTCMP00), F0351H (TRTCMP01), F0358H (TRTCMP10), F0359H (TRTCMP11)   After reset: FFH   R/W Notes 1, 2

Symbol	7	6	5	4	3	2	1	0
TRTCMPni								

Note 1. The TRTCMPni register must not be set to 00H.
Note 2. Can be accessed only when the mode select bit (TCSMDn) in the 8-bit interval timer control register n (TRTCRn) is 0.

11.3.4 8-bit interval timer compare register n (TRTCMPn) (n = 0, 1)

This is a compare value register when the 8-bit interval timer is used in 16-bit interval timer mode.
The TRTCMPn register can be set by a 16-bit memory manipulation instruction.
Reset signal generation sets this register to FFFFH.
The setting is 0001H to FFFFH Note 1.
This register is used to store the compare value of the TRTn register (counter).
Write-access clears the count value (TRTn) to 0000H.
Refer to **11.4.4 Timing for Updating Compare Register Values** for the timing of rewriting the compare value.

Figure 11 - 5 Format of 8-bit interval timer compare register n (TRTCMPn)

Address: F0350H (TRTCMP0), F0358H (TRTCMP1)   After reset: FFFFH   R/W Note 2

	F0351H (TRTCMP01) F0359H (TRTCMP11)								F0350H (TRTCMP00) F0358H (TRTCMP10)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRTCMPn																

Note 1. The TRTCMPn register must not be set to 0000H.
Note 2. Can be accessed only when the mode select bit (TCSMDn) in the 8-bit interval timer control register n (TRTCRn) is 1.

11.3.5 8-bit interval timer control register n (TRTCRn) (n = 0, 1)

This register is used to start and stop counting by the 8-bit interval timer and to switch between using the 8-bit interval timer as an 8-bit counter or a 16-bit counter.

The TRTCRn register can be set by a 1-bit or 8-bit manipulation instruction.

Reset signal generation resets this register to 00H.

Figure 11 - 6 Format of 8-bit interval timer control register n (TRTCRn)

Address: F0352H (TRTCR0), F035AH (TRTCR1) After reset: 00H R/W Note 3

Symbol	7	6	5	4	3	<2>	1	<0>
TRTCRn	TCSMDn	0	0	TCLKENn	0	TSTARTn1	0	TSTARTn0

TCSMDn	Mode select
0	Operates as 8-bit counter
1	Operates as 16-bit counter (channel 0 and channel 1 are connected)
Refer to **11.4 Operation** for details.	

TCLKENn	8-bit interval timer clock enable Note 1
0	Clock is stopped
1	Clock is supplied

TSTARTn1	8-bit interval timer 1 count start Notes 1, 2
0	Count stops
1	Count starts
In 8-bit interval timer mode, writing 1 to the TSTARTn1 bit starts the TRTn1 count and writing 0 stops the count. In 16-bit interval timer mode, this bit is invalid because it is not used. Refer to **11.4 Operation** for details.	

TSTARTn0	8-bit interval timer 0 count start Notes 1, 2
0	Count stops
1	Count starts
In 8-bit interval timer mode, writing 1 to the TSTARTn0 bit starts the TRTn0 count and writing 0 stops the count. In 16-bit interval timer mode, writing 1 to the TSTARTn0 bit starts the TRTn count and writing 0 stops the count. Refer to **11.4 Operation** for details.	

Note 1. Be sure to set the TCLKENn bit to 1 before setting the 8-bit interval timer. To stop the clock, set TSTARTn0 and TSTARTn1 to 0 and then set the TCLKENn bit to 0 after one or more cycles of the operating clock (f_{SXR} or f_{IL}) have elapsed. Refer to **11.5.3 8-Bit Interval Timer Setting Procedure** for details.

Note 2. Refer to **11.5.1 Changing Settings of Operating Mode** for the notes on using bits TSTARTn0, TSTARTn1, and TCSMDn.

Note 3. Bits 6, 5, 3, and 1 are read-only. When writing, write 0. When reading, 0 is read.

11.3.6 8-bit interval timer division register n (TRTMDn) (n = 0, 1)

This register is used to select the division ratio of the count source used by the 8-bit interval timer.
The TRTMDn register can be set by an 8-bit memory manipulation instruction.
Reset signal generation sets this register to 00H.

Figure 11 - 7 Format of 8-bit interval timer division register n (TRTMDn)

Address: F0353H (TRTMD0), F035BH (TRTMD1) After reset: 00H R/W Note 4

Symbol	7	6	5	4	3	2	1	0
TRTMDn	—	TCKn1			—	TCKn0		

TCKn1			8-bit interval timer 1 division select Notes 1, 2, 3
Bit 6	Bit 5	Bit 4	
0	0	0	fSXR or fIL
0	0	1	fSXR/2 or fIL/2
0	1	0	fSXR/4 or fIL/4
0	1	1	fSXR/8 or fIL/8
1	0	0	fSXR/16 or fIL/16
1	0	1	fSXR/32 or fIL/32
1	1	0	fSXR/64 or fIL/64
1	1	1	fSXR/128 or fIL/128
In 8-bit interval timer mode, TRTn1 counts using the count source set in TCKn1. In 16-bit interval timer mode, set these bits to 000 because they are not used. Refer to **11.4 Operation** for details.			

TCKn0			8-bit interval timer 0 division select Notes 1, 2, 3
Bit 2	Bit 1	Bit 0	
0	0	0	fSXR or fIL
0	0	1	fSXR/2 or fIL/2
0	1	0	fSXR/4 or fIL/4
0	1	1	fSXR/8 or fIL/8
1	0	0	fSXR/16 or fIL/16
1	0	1	fSXR/32 or fIL/32
1	1	0	fSXR/64 or fIL/64
1	1	1	fSXR/128 or fIL/128
In 8-bit interval timer mode, TRTn0 counts using the count source set in TCKn0. In 16-bit interval timer mode, TRTn counts using the count source set in TCKn0. Refer to **11.4 Operation** for details.			

Note 1. Do not switch the count source during count operation. When switching the count source, set these bits while the TSTARTni bit in the TRTCRn register is 0 (count stops).
Note 2. Set TCKni of the unused channel to 000B.
Note 3. Be sure to set the TCKni (i = 0, 1) bit before setting the TRTCMPni register.
Note 4. Bits 7 and 3 are read-only. When writing, write 0. When reading, 0 is read.

11.4 Operation

11.4.1 Count Mode

The following two modes are supported: 8-bit counter mode and 16-bit counter mode. Table 11 - 3 lists the Registers and Settings Used in 8-Bit Counter Mode and Table 11 - 4 lists the Registers and Settings Used in 16-Bit Counter Mode.

Table 11 - 3 Registers and Settings Used in 8-Bit Counter Mode

Register Name (Symbol)	Bit	Function
8-bit interval timer counter register n0 (TRTn0)	b7 to b0	8-bit counter of channel 0. The count value can be read.
8-bit interval timer counter register n1 (TRTn1)	b7 to b0	8-bit counter of channel 1. The count value can be read.
8-bit interval timer compare register n0 (TRTCMPn0)	b7 to b0	8-bit compare value of channel 0. Set the compare value.
8-bit interval timer compare register n1 (TRTCMPn1)	b7 to b0	8-bit compare value of channel 1. Set the compare value.
8-bit interval timer control register n (TRTCRn)	TSTARTn0	Select whether to start/stop the count of channel 0.
	TSTARTn1	Select whether to start/stop the count of channel 1.
	TCLKENn	Set to 1.
	TCSMDn	Set to 0.
8-bit interval timer division register n (TRTMDn)	TCKn0	Select the count clock of channel 0.
	TCKn1	Select the count clock of channel 1.

Remark n = 0, 1

Table 11 - 4 Registers and Settings Used in 16-Bit Counter Mode

Register Name (Symbol)	Bit	Function
8-bit interval timer counter register n (TRTn)	b15 to b0	16-bit counter. The count value can be read.
8-bit interval timer compare register n (TRTCMPn)	b15 to b0	16-bit compare value. Set the compare value.
8-bit interval timer control register n (TRTCRn)	TSTARTn0	Select whether to control starting/stopping the count.
	TSTARTn1	Set to 0.
	TCLKENn	Set to 1.
	TCSMDn	Set to 1.
8-bit interval timer division register n (TRTMDn)	TCKn0	Select the count clock.
	TCKn1	Set to 000B.

Remark n = 0, 1

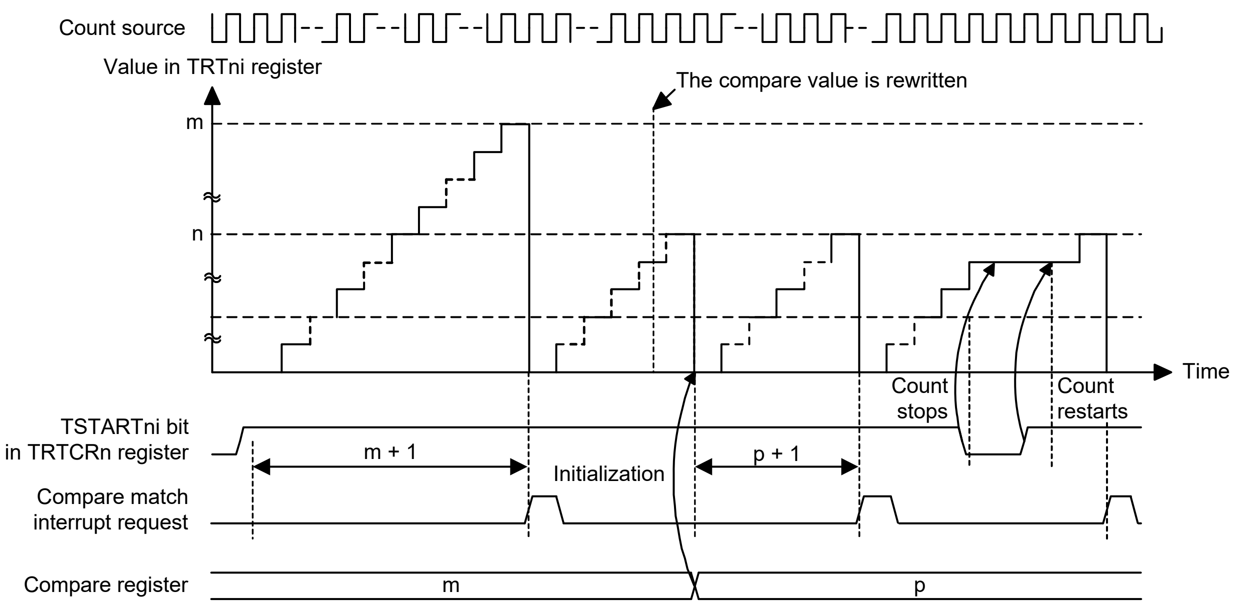
11.4.2 Timer Operation

The counter is incremented by cycles of the source selected by the TCKni (n = 0, 1, i = 0, 1) bits in the division register (TRTMDn). The counter value is incremented by one each time a cycle from the source is input. After the counter value reaches the value for comparison, matching of the values is detected on the input of the next cycle from the source, after which an interrupt is generated. The interrupt request takes the form of a single pulse synchronized with the source for counting.

Note that, when the TSTARTni bit in the TRTCRn register is set to 0 and counting is stopped at 00h, the interrupt signal (INTITnm) will be fixed to the high level. If the interrupt signal (INTITnm) is in use as an activating source for the data transfer controller or a source of events for the ELC in such cases, the activating source or event signal is continually generated.

When operation is stopped, the counter retains the value counted immediately before it stopped. To clear the value, set a value for comparison in the TRTCMPni register again. After writing to the TRTCMPni register, the counter value is cleared after two cycles of the source for counting.

Figure 11 - 8 Example of Timer Operation



Remark n = 0, 1 i = 0, 1 m, p: Values set in TRTCMPni register

However, the initial 00H count interval when starting count varies as follows according to the timing 1 is written in the TSTARTni (i = 0, 1) bit of the TRTCR register.

- When the count source (fSXR or fIL) is selected
  Maximum: Two cycles of the count source
  Minimum: One cycle of the count source
- When the count source (fSXR/2^m or fIL/2^m) is selected
  Maximum: One cycle of the count source
  Minimum: One cycle of the selected clock (fSXR or fIL)

When the count value matches the compare value, the count value is cleared by the next count source. When the compare value in the TRTCMPni register is rewritten, the count value is also cleared two cycles of the count source after writing.
Table 11 - 5 lists the Interrupt Sources in 8-Bit/16-Bit Count Mode.

Table 11 - 5 Interrupt Sources in 8-Bit/16-Bit Count Mode

Interrupt Name	8-Bit Count Mode Source	16-Bit Count Mode Source
INTITn0	Rising edge of the next count source after compare match of channel 0	Rising edge of the next count source after compare match
INTITn1	Rising edge of the next count source after compare match of channel 1	Not generated

Remark n = 0, 1

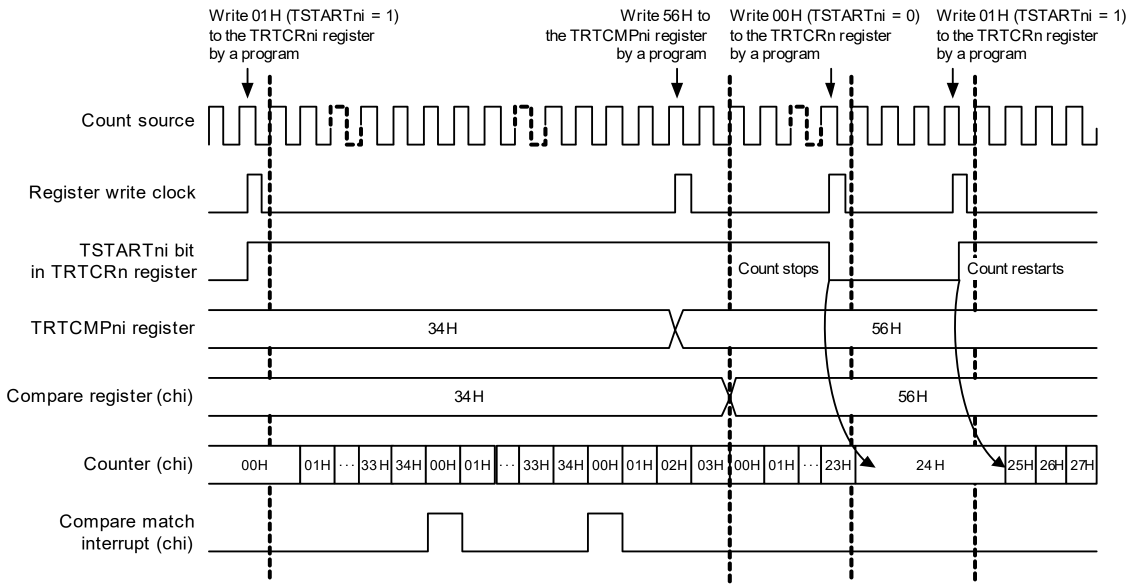
11.4.3 Start/Stop Timing

11.4.3.1 When Count Source (fSXR) is Selected

After 1 is written to the TSTARTni (n = 0, 1, i = 0, 1) bit in the TRTCRn register, the count is started by the next subsystem clock (fSXR), and then the counter is incremented from 00H to 01H by the next count source (fSXR). Likewise, after 0 is written to the TSTARTni bit, the count is stopped after the counter is incremented by the subsystem clock (fSXR).

Figure 11 - 9 shows the timing for starting/stopping count operation, and Figure 11 - 10 shows the timing of count stop → compare setting (count clearing) → count start. Figure 11 - 9 and Figure 11 - 10 show the update timing in 8-bit counter mode, but operation is performed at the same timing even in 16-bit counter mode.

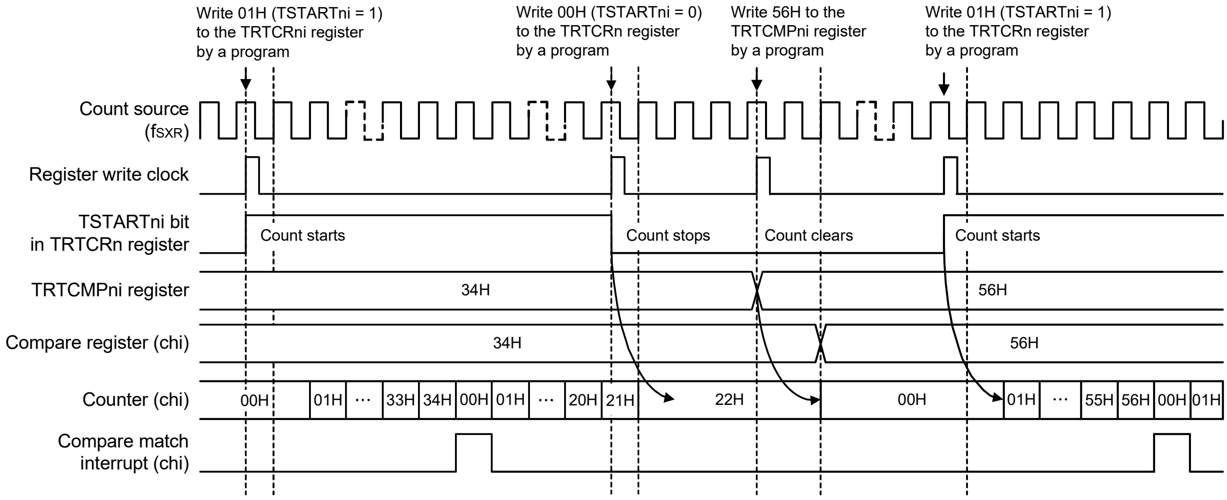
Figure 11 - 9 Example of Count Start/Stop Operation (fSXR Selected)



The TCSMDn bit in the TRTCRn register is set to 0 (8-bit counter operation)

Remark n = 0, 1 i = 0, 1

Figure 11 - 10 Example of Count Stop → Count Clearing → Count Start Operation (fSXR Selected)



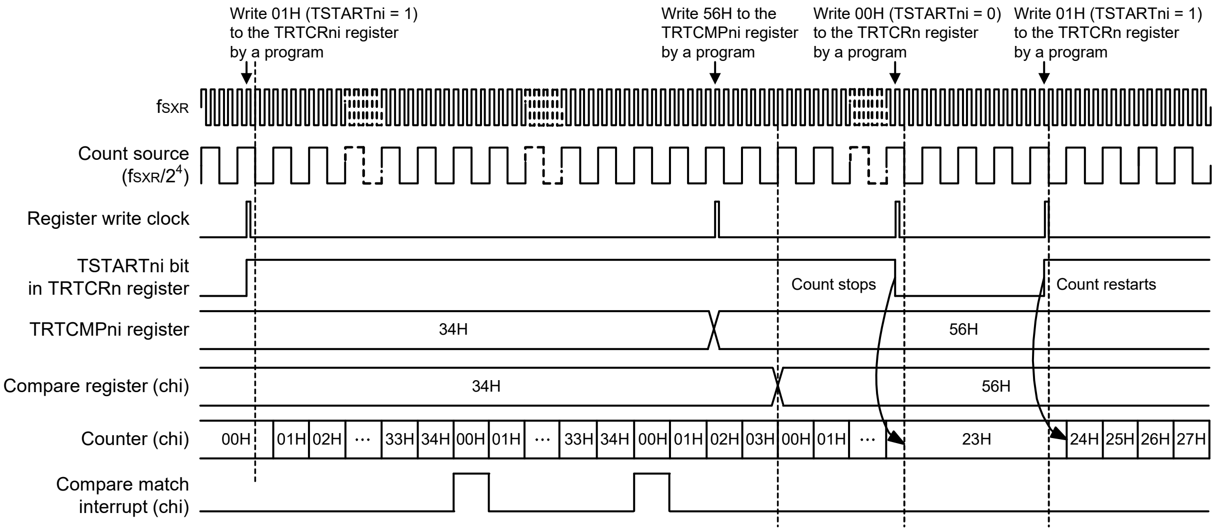
The TCSMDn bit in the TRTCRn register is set to 0 (8-bit counter operation)

Remark n = 0, 1 i = 0, 1

11.4.3.2 When Count Source (fSXR/2m) is Selected

After 1 is written to the TSTARTni (n = 0, 1, i = 0, 1) bit in the TRTCRn register, the count is started with the next subsystem clock (fSXR), and then the counter is incremented from 00H to 01H by the next count source (fSXR/2m). Likewise, after 0 is written to the TSTARTni bit, the count is stopped with the subsystem clock (fSXR).
However, the first period to count 00H when the timer starts counting is shorter than one cycle of the count source as below, depending on the timing for writing to the TSTARTni bit and the timing of the next count source.
Minimum: One cycle of the subsystem clock (fSXR)
Maximum: One cycle of the count source
Figure 11 - 11 shows the timing for starting/stopping count operation, and Figure 11 - 12 shows the timing of count stop → compare setting (count clearing) → count start. Figure 11 - 11 and Figure 11 - 12 show the update timing in
8-bit counter mode, but operation is performed at the same timing even in 16-bit counter mode.

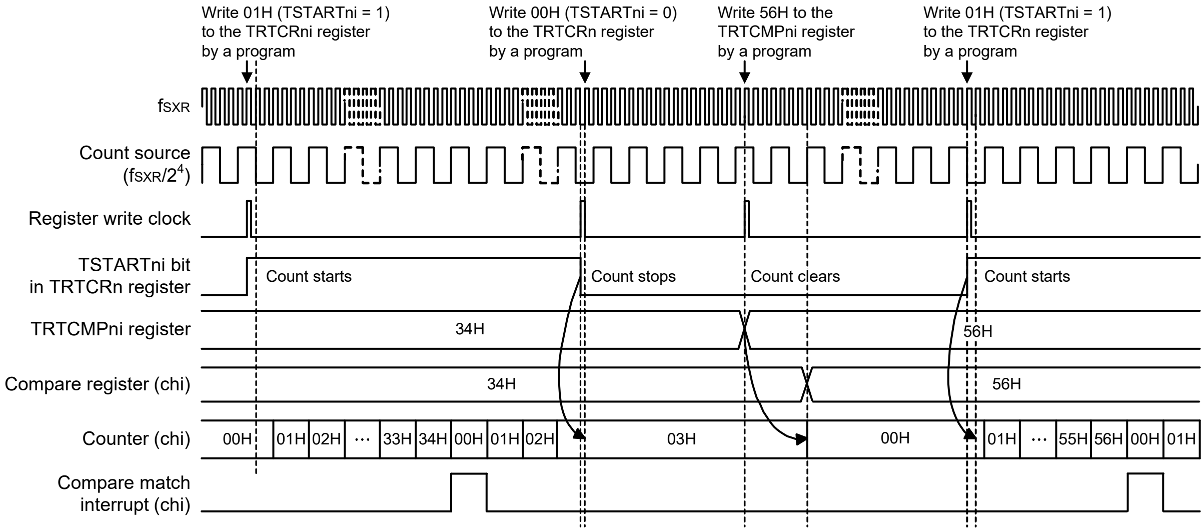
Figure 11 - 11 Example of Count Start/Stop Operation (fSXR/2m Selected)



The TCSMDn bit in the TRTCRn register is set to 0 (8-bit counter operation)

Remark n = 0, 1 i = 0, 1

Figure 11 - 12 Example of Count Stop → Count Clearing → Count Start Operation (fSXR/2m Selected)



The TCSMDn bit in the TRTCRn register is set to 0 (8-bit counter operation)

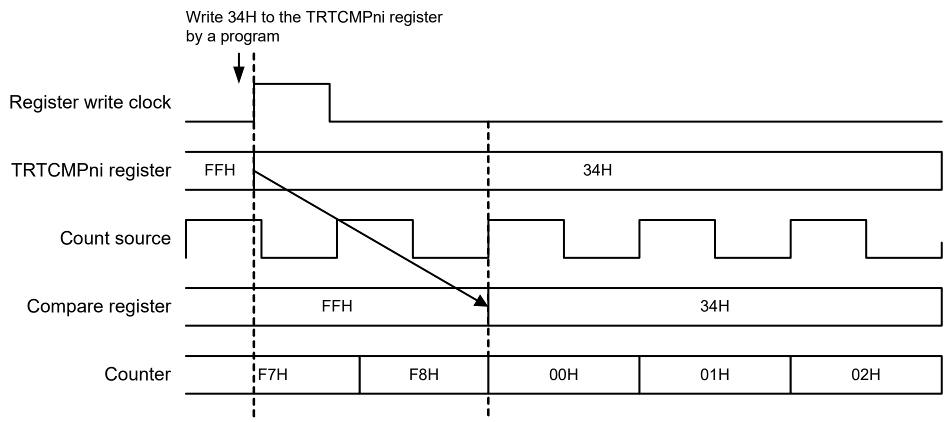
Remark n = 0, 1 i = 0, 1

11.4.4 Timing for Updating Compare Register Values

The timing for updating the value of the TRTCMPni (n = 0, 1, i = 0, 1) register is the same, regardless of the value of the TSTARTni bit in the TRTCRn register. After TRTCMPni is write-accessed, the value is stored in the compare register after two cycles of the count source. When stored in the compare register, the count value is cleared and set (8-bit count mode: 00H, 16-bit count mode: 0000H).

Figure 11 - 13 shows the timing of rewrite operation. This figure shows the update timing in 8-bit count mode, but operation is performed at the same timing in 16-bit count mode.

Figure 11 - 13 Timing of Compare Value Rewrite Operation

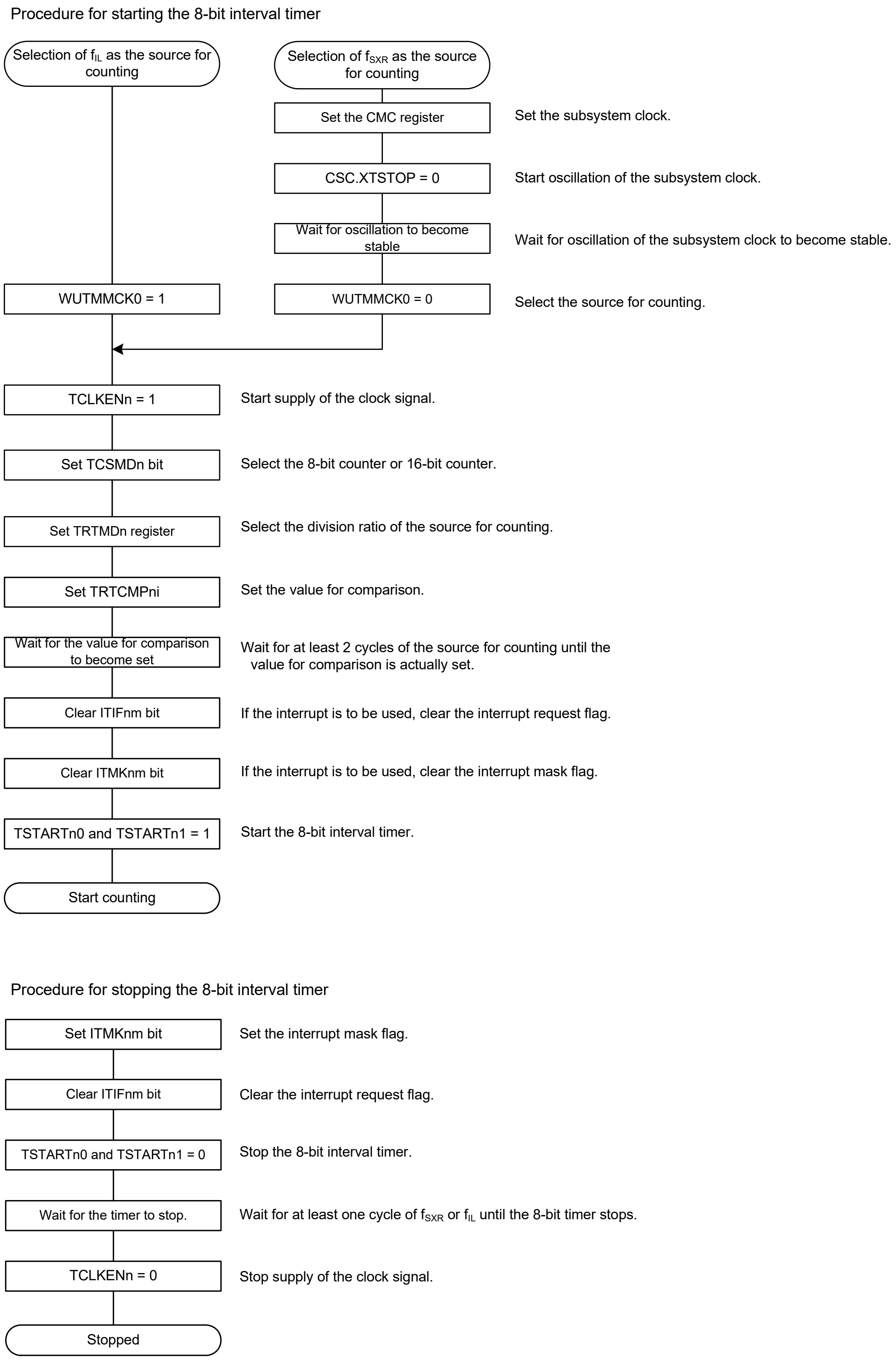


Remark n = 0, 1 i = 0, 1

11.4.5 Procedure for Setting the 8-bit Interval Timer

The procedure for starting and stopping the 8-bit interval timer is shown below.

Figure 11 - 14 Procedure for Starting and Stopping the 8-bit Interval Timer



11.5 Notes on 8-Bit Interval Timer

11.5.1 Changing Settings of Operating Mode

The settings of bits TCSMDn and TCKni (n = 0, 1, i = 0, 1) must be changed while the TSTARTni bit in the TRTCRn register is 0 (count stops). After the value of the TSTARTni bit is rewritten from 1 to 0 (count stops), allow at least one cycle of f_{SXR} or f_{IL} to elapse before accessing the registers (TRTCRn and TRTMDn) associated with the 8-bit interval timer.

11.5.2 Accessing Compare Registers

Do not write to the same compare registers (TRTCMPn0, TRTCMPn1, and TRTCMPn) successively. When writing successively, allow at least two cycles of the count source between writes.
Writing to the compare register (TRTCMPn0, TRTCMPn1, TRTCMPn) must proceed while the source to drive counting is made to oscillate by setting the 8-bit interval timer clock enable bit (TCLKENn) to 1.

11.5.3 8-Bit Interval Timer Setting Procedure

To supply the clock, set the 8-bit interval timer clock enable bit (TCLKENn) in the 8-bit interval timer control register (TRTCRn) to 1 and then set the TSTARTni bit. Do not set bits TCLKENn and TSTARTni at the same time.
To stop the clock, set TSTARTni to 0 and then allow at least one cycle of f_{SXR} or f_{IL} to elapse before setting the TCLKENn bit to 0.

CHAPTER 12 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

The number of the output pins of the clock buzzer output/buzzer output control circuit differs depending on the product.

Output Pin	20, 24, 30, 32-pin	48-pin
PCLBUZ0	√	√
PCLBUZ1	—	√

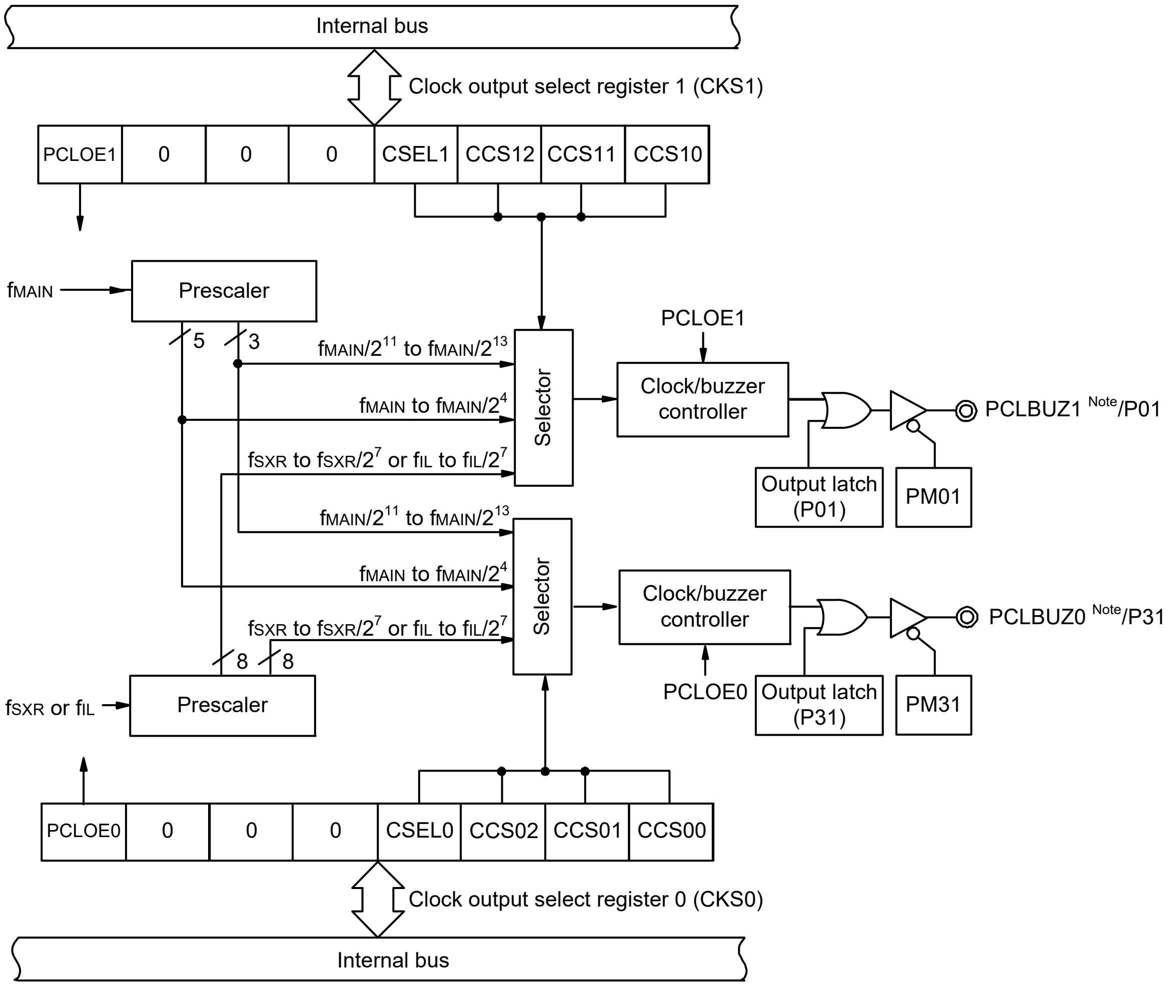
Caution Most of the following descriptions in this chapter use the 48-pin products as an example.

12.1 Functions of Clock Output/Buzzer Output Controller

The clock output controller is intended for clock output for supply to peripheral ICs.
Buzzer output is a function to output a square wave of buzzer frequency.
One pin can be used to output a clock or buzzer sound.
Two output pins, PCLBUZ0 and PCLBUZ1, are available.
The PCLBUZn pin outputs a clock selected by clock output select register n (CKSn).
Figure 12 - 1 shows the Block Diagram of Clock Output/Buzzer Output Controller.

Remark n = 0, 1

Figure 12 - 1 Block Diagram of Clock Output/Buzzer Output Controller



Note For output frequencies available from PCLBUZ0 and PCLBUZ1, refer to **34.4 AC Characteristics**.

12.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

Table 12 - 1 Configuration of Clock Output/Buzzer Output Controller

Item	Configuration
Control registers	Clock output select registers n (CKSn) Subsystem clock supply mode control register (OSMC) Port mode registers 0, 3 (PM0, PM3) Port registers 0, 3 (P0, P3)

12.3 Registers Controlling Clock Output/Buzzer Output Controller

The following register is used to control the clock output/buzzer output controller.

- Clock output select registers n (CKSn)
- Subsystem clock supply mode control register (OSMC)
- Port mode registers 0, 3 (PM0, PM3)
- Port registers 0, 3 (P0, P3)

12.3.1 Clock output select registers n (CKSn)

This register set output enable/disable for clock output or for the buzzer frequency output pin (PCLBUZn), and set the output clock.

Select the clock to be output from the PCLBUZn pin by using the CKSn register.

The CKSn register is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 12 - 2 Format of Clock output select registers n (CKSn)

Address: FFFA5H (CKS0), FFFA6H (CKS1) After reset: 00H 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 disable (default)
1	Output enable

CSELn	CCSn2	CCSn1	CCSn0	PCLBUZn pin output clock selection				
					f_{MAIN} = 5 MHz	f_{MAIN} = 10 MHz	f_{MAIN} = 20 MHz	f_{MAIN} = 24 MHz
0	0	0	0	f_{MAIN}	5 MHz	10 MHz Note	Setting prohibited Note	Setting prohibited Note
0	0	0	1	$f_{MAIN}/2$	2.5 MHz	5 MHz	10 MHz Note	Setting prohibited Note
0	0	1	0	$f_{MAIN}/2^2$	1.25 MHz	2.5 MHz	5 MHz	6 MHz
0	0	1	1	$f_{MAIN}/2^3$	625 kHz	1.25 MHz	2.5 MHz	3 MHz
0	1	0	0	$f_{MAIN}/2^4$	312.5 kHz	625 kHz	1.25 MHz	1.5 MHz
0	1	0	1	$f_{MAIN}/2^{11}$	2.44 kHz	4.88 kHz	9.77 kHz	11.7 kHz
0	1	1	0	$f_{MAIN}/2^{12}$	1.22 kHz	2.44 kHz	4.88 kHz	5.86 kHz
0	1	1	1	$f_{MAIN}/2^{13}$	610 Hz	1.22 kHz	2.44 kHz	2.93 kHz
1	0	0	0	f_{SXR} or f_{IL}	32.768 kHz or 15 kHz			
1	0	0	1	$f_{SXR}/2$ or $f_{IL}/2$	16.384 kHz or 7.5 kHz			
1	0	1	0	$f_{SXR}/2^2$ or $f_{IL}/2^2$	8.192 kHz or 3.75 kHz			
1	0	1	1	$f_{SXR}/2^3$ or $f_{IL}/2^3$	4.096 kHz or 1.875 kHz			
1	1	0	0	$f_{SXR}/2^4$ or $f_{IL}/2^4$	2.048 kHz or 938 Hz			
1	1	0	1	$f_{SXR}/2^5$ or $f_{IL}/2^5$	1.024 kHz or 469 Hz			
1	1	1	0	$f_{SXR}/2^6$ or $f_{IL}/2^6$	512 Hz or 234 Hz			
1	1	1	1	$f_{SXR}/2^7$ or $f_{IL}/2^7$	256 Hz or 117 Hz			

Note Use the output clock within a range of 8 MHz. See **34.4 AC Characteristics** for details.

Caution Change the output clock after disabling clock output (PCLOEn = 0).

Remark 1. n = 0, 1

Remark 2. f_{MAIN}: Main system clock frequency

f_{SXR}: Subsystem clock oscillation circuit and RTC2/other clock frequency

f_{IL}: Low-speed on-chip oscillator clock frequency

12.3.2 Registers controlling port functions of pins to be used for clock or buzzer output

Using a port pin for clock or buzzer output requires setting of the registers that control the port functions multiplexed on the target pin (port mode register (PMxx), port register (Pxx)). For details, see **4.3.1 Port mode registers (PMxx)** and **4.3.2 Port registers (Pxx)**.

Specifically, using a port pin with a multiplexed clock or buzzer output function (e.g. P31/PCLBUZ0) for clock or buzzer output, requires setting the corresponding bits in the port register (Pxx), port mode register (PMxx), and port mode control register (PMCxx) to 0.

Example: When P31/PCLBUZ0 is to be used for clock or buzzer output
Set the P31 bit of port register 3 to 0.
Set the PM31 bit of port mode register 3 to 0.
Set the PMC31 bit of port mode control register 3 to 0.

12.4 Operations of Clock Output/Buzzer Output Controller

One pin can be used to output a clock or buzzer sound.
The PCLBUZ0 pin outputs a clock/buzzer selected by clock output select register 0 (CKS0).
The PCLBUZ1 pin outputs a clock/buzzer selected by clock output select register 1 (CKS1).

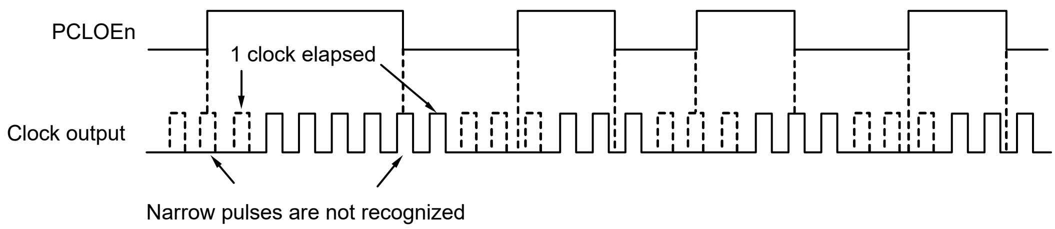
12.4.1 Operation as output pin

The PCLBUZn pin is output as the following procedures.

<1> Set 0 in the bit of the port mode register (PMxx) and port register (Px) which correspond to the port which has a pin used as the PCLBUZ0 pin.
<2> Select the output frequency with bits 0 to 3 (CCSn0 to CCSn2, CSELn) of the clock output select register (CKSn) of the PCLBUZn pin (output in disabled status).
<3> Set bit 7 (PCLOEn) of the CKSn register to 1 to enable clock/buzzer output.

Remark 1. The controller used for outputting the clock starts or stops outputting the clock one clock after enabling or disabling clock output (PCLOEn bit) is switched. At this time, pulses with a narrow width are not output. Figure 12 - 3 shows enabling or stopping output using the PCLOEn bit and the timing of outputting the clock.
Remark 2. n = 0, 1

Figure 12 - 3 Timing of Outputting Clock from PCLBUZn Pin



12.5 Cautions of clock output/buzzer output controller

When the main system clock is selected for the PCLBUZn output (CSELn = 0), if STOP mode is entered within 1.5 clock cycles output from the PCLBUZn pin after the output is disabled (PCLOEn = 0), the PCLBUZn output width becomes shorter.

CHAPTER 13 WATCHDOG TIMER

13.1 Functions of Watchdog Timer

The counting operation of the watchdog timer is set by the option byte (000C0H).
The watchdog timer operates on the low-speed on-chip oscillator clock (f_{IL}).
The watchdog timer is used to detect an inadvertent program loop. If a program loop is detected, an internal reset signal is generated.
Program loop is detected in the following cases.

- If the watchdog timer counter overflows
- If a 1-bit manipulation instruction is executed on the watchdog timer enable register (WDTE)
- If data other than “ACH” is written to the WDTE register
- If data is written to the WDTE register during a window close period

When a reset occurs due to the watchdog timer, bit 4 (WDTRF) of the reset control flag register (RESF) is set to 1.
For details of the RESF register, see **CHAPTER 24 RESET FUNCTION**.
When 75% of the overflow time + 1/2 f_{IL} is reached, an interval interrupt can be generated.

13.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 13 - 1 Configuration of Watchdog Timer

Item	Configuration
Counter	Internal counter (17 bits)
Control register	Watchdog timer enable register (WDTE)

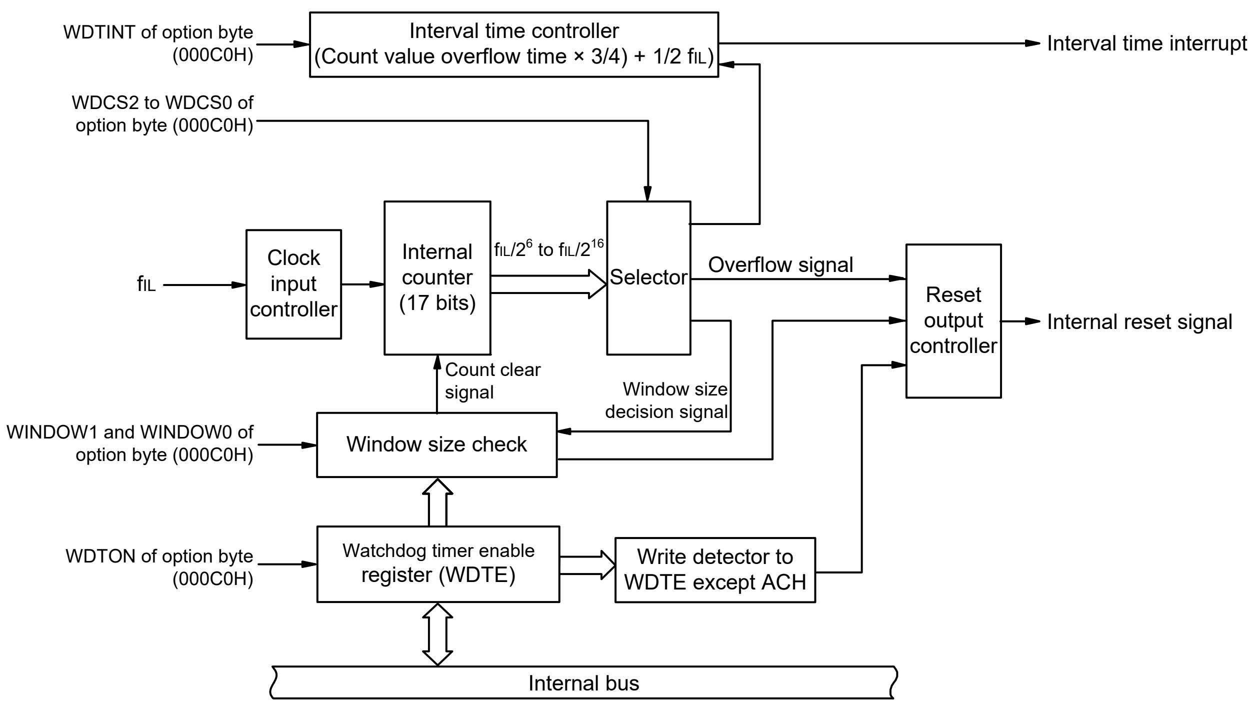
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the option byte.

Table 13 - 2 Setting of Option Bytes and Watchdog Timer

Setting of Watchdog Timer	Option Byte (000C0H)
Watchdog timer interval interrupt	Bit 7 (WDTINT)
Window open period	Bits 6 and 5 (WINDOW1, WINDOW0)
Controlling counter operation of watchdog timer	Bit 4 (WDTON)
Overflow time of watchdog timer	Bits 3 to 1 (WDCS2 to WDCS0)
Controlling counter operation of watchdog timer (in HALT/STOP mode)	Bit 0 (WDSTBYON)

Remark For the option byte, see **CHAPTER 29 OPTION BYTE**.

Figure 13 - 1 Block Diagram of Watchdog Timer



Remark fIL: Low-speed on-chip oscillator clock

13.3 Register Controlling Watchdog Timer

The watchdog timer is controlled by the watchdog timer enable register (WDTE).

13.3.1 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.

Reset signal generation sets this register to 9AH or 1AH Note.

Figure 13 - 2 Format of Watchdog timer enable register (WDTE)

Address: FFFABH After reset: 1AH/9AH Note R/W

Symbol	7	6	5	4	3	2	1	0
WDTE								

Note The WDTE register reset value differs depending on the WDTON bit setting value of the option byte (000C0H). To operate watchdog timer, set the WDTON bit to 1.

WDTON Bit Setting Value	WDTE Register Reset Value
0 (watchdog timer count operation disabled)	1AH
1 (watchdog timer count operation enabled)	9AH

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/1AH (this differs from the written value (ACH)).

13.4 Operation of Watchdog Timer

13.4.1 Controlling operation of watchdog timer

1. When the watchdog timer is used, its operation is specified by the option byte (000C0H).

• Enable counting operation of the watchdog timer by setting bit 4 (WDTON) of the option byte (000C0H) to 1 (the counter starts operating after a reset release) (for details, see **CHAPTER 29**).

WDTON	Watchdog Timer Counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

• Set an overflow time by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H) (for details, see **13.4.2** and **CHAPTER 29**).
• Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (000C0H) (for details, see **13.4.3** and **CHAPTER 29**).

2. After a reset release, the watchdog timer starts counting.
3. By writing "ACH" to the watchdog timer enable register (WDTE) after the watchdog timer starts counting and before the overflow time set by the option byte, the watchdog timer is cleared and starts counting again.
4. After that, write the WDTE register the second time or later after a reset release during the window open period. If the WDTE register is written during a window close period, an internal reset signal is generated.
5. If the overflow time expires without "ACH" written to the WDTE register, an internal reset signal is generated.
   An internal reset signal is generated in the following cases.

• If a 1-bit manipulation instruction is executed on the WDTE register
• If data other than "ACH" is written to the WDTE register

Caution 1.When data is written to the watchdog timer enable register (WDTE) for the first time after reset release, 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 2 clocks (f_{IL}) may occur before the watchdog timer is cleared.
Caution 3. The watchdog timer can be cleared immediately before the count value overflows.

Caution 4. The operation of the watchdog timer in the HALT and STOP modes differs as follows depending on the set value of bit 0 (WDSTBYON) of the 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.

13.4.2 Setting overflow time of watchdog timer

Set the overflow time of the watchdog timer by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H).

If an overflow occurs, an internal reset signal is generated. The present count is cleared and the watchdog timer starts counting again by writing “ACH” to the watchdog timer enable register (WDTE) during the window open period before the overflow time.

The following overflow times can be set.

Table 13 - 3 Setting of Overflow Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer (f_{IL} = 17.25 kHz (MAX.))
0	0	0	$2^6/f_{IL}$ (3.71 ms)
0	0	1	$2^7/f_{IL}$ (7.42 ms)
0	1	0	$2^8/f_{IL}$ (14.84 ms)
0	1	1	$2^9/f_{IL}$ (29.68 ms)
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)
1	0	1	$2^{13}/f_{IL}$ (474.89 ms)Note
1	1	0	$2^{14}/f_{IL}$ (949.79 ms)Note
1	1	1	$2^{16}/f_{IL}$ (3799.18 ms)Note

Note When the overflow time of the watchdog timer is set to $2^{13}/f_{IL}$, $2^{14}/f_{IL}$, or $2^{16}/f_{IL}$ and the interval interrupt is in use, clear 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.

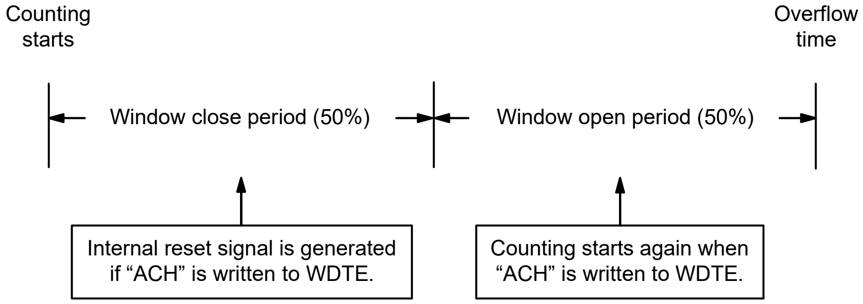
Remark f_{IL}: Low-speed on-chip oscillator clock frequency

13.4.3 Setting window open period of watchdog timer

Set the window open period of the watchdog timer by using bits 6 and 5 (WINDOW1, WINDOW0) of the option byte (000C0H). The outline of the window is as follows.

- If “ACH” is written to the watchdog timer enable register (WDTE) during the window open period, the watchdog timer is cleared and starts counting again.
- Even if “ACH” is written to the WDTE register during the window close period, an abnormality is detected and an internal reset signal is generated.

Example: If the window open period is 50%



Caution When data is written to the WDTE register for the first time after reset release, the watchdog timer 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 13 - 4 Setting Window Open Period of Watchdog Timer

WINDOW1	WINDOW0	Window Open Period of Watchdog Timer
0	0	Setting prohibited
0	1	50%
1	0	75%Note
1	1	100%

Note When the window open period is set to 75%, clearing the counter of the watchdog timer (writing ACH to WDTE) must proceed outside the corresponding period from among those listed below, over which clearing of the counter is prohibited (for example, confirming that the interval timer interrupt request flag (WDTIIF) of the watchdog timer is set).

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time (f_{IL} = 17.25 kHz (MAX.))	Period over which clearing the counter is prohibited when the window open period is set to 75%
0	0	0	$2^6/f_{IL}$ (3.71 ms)	1.85 ms to 2.51 ms
0	0	1	$2^7/f_{IL}$ (7.42 ms)	3.71 ms to 5.02 ms
0	1	0	$2^8/f_{IL}$ (14.84 ms)	7.42 ms to 10.04 ms
0	1	1	$2^9/f_{IL}$ (29.68 ms)	14.84 ms to 20.08 ms
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)	56.36 ms to 80.32 ms
1	0	1	$2^{13}/f_{IL}$ (474.90 ms)	237.44 ms to 321.26 ms
1	1	0	$2^{14}/f_{IL}$ (949.80 ms)	474.89 ms to 642.51 ms
1	1	1	$2^{16}/f_{IL}$ (3799.19 ms)	1899.59 ms to 2570.04 ms

Caution When bit 0 (WDSTBYON) of the option byte (000C0H) = 0, the window open period is 100% regardless of the values of the WINDOW1 and WINDOW0 bits.

Remark If the overflow time is set to $2^9/f_{IL}$, the window close time and open time are as follows.

	Setting of Window Open Period		
	50%	75%	100%
Window close time	0 to 20.08 ms	0 to 10.04 ms	None
Window open time	20.08 to 29.68 ms	10.04 to 29.68 ms	0 to 29.68 ms

<When window open period is 50%>

- Overflow time:
  $2^9/f_{IL}$ (MAX.) = $2^9/17.25$ kHz (MAX.) = 29.68 ms
- Window close time:
  0 to $2^9/f_{IL}$ (MIN.) × (1 - 0.5) = 0 to $2^9/12.75$ kHz × 0.5 = 0 to 20.08 ms
- Window open time:
  $2^9/f_{IL}$ (MIN.) × (1 - 0.5) to $2^9/f_{IL}$ (MAX.) = $2^9/12.75$ kHz × 0.5 to $2^9/17.25$ kHz = 20.08 to 29.68 ms

13.4.4 Setting watchdog timer interval interrupt

Setting bit 7 (WDTINT) of an option byte (000C0H) can generate an interval interrupt (INTWDTI) when 75% of the overflow time + 1/2 f_{IL} is reached.

Table 13 - 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 + 1/2 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.

CHAPTER 14 A/D CONVERTER

The number of analog input channels of the A/D converter differs, depending on the product.

			20-pin	24-pin	30-pin	32-pin	48-pin
Analog input channels		Total	6 ch		12 ch		17 ch
	High accuracy channel	Pins based on input buffer power supply AV_{DD}	6 ch (ANI2 to ANI4, ANI11 to ANI3)		12 ch (ANI2 to ANI13)		14 ch (ANI0 to ANI13)
	Standard channel	Pins based on input buffer power supply V_{DD}	0 ch				3 ch (ANI16 to ANI18)

Remark Most of the following descriptions in this chapter use the 48-pin as an example.

14.1 Function of A/D Converter

The A/D converter converts analog input signals into digital values, and is configured to control analog inputs, including up to 17 channels of A/D converter analog inputs (ANI0 to ANI13 and ANI16 to ANI18). 12-bit resolution or 8-bit resolution can also be selected by using the ADTYP bit of A/D converter mode register 2 (ADM2).
The A/D converter has the following function.

• 12-bit or 8-bit resolution A/D conversion
12-bit or 8-bit resolution A/D conversion is carried out repeatedly for one analog input channel selected from ANI0 to ANI13 and ANI16 to ANI18. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated (when in the select mode).

Caution The valid resolution differs depending on the voltage conditions of AV_{DD} and AV_{REFP}.
For details, see 34.6.1 A/D converter characteristics.

Remark When using the converter with a resolution of 10 bits, select the 12-bit resolution mode (ADTYP = 0).
Use the higher 10 bits of the conversion result. Do not use the lower 2 bits.

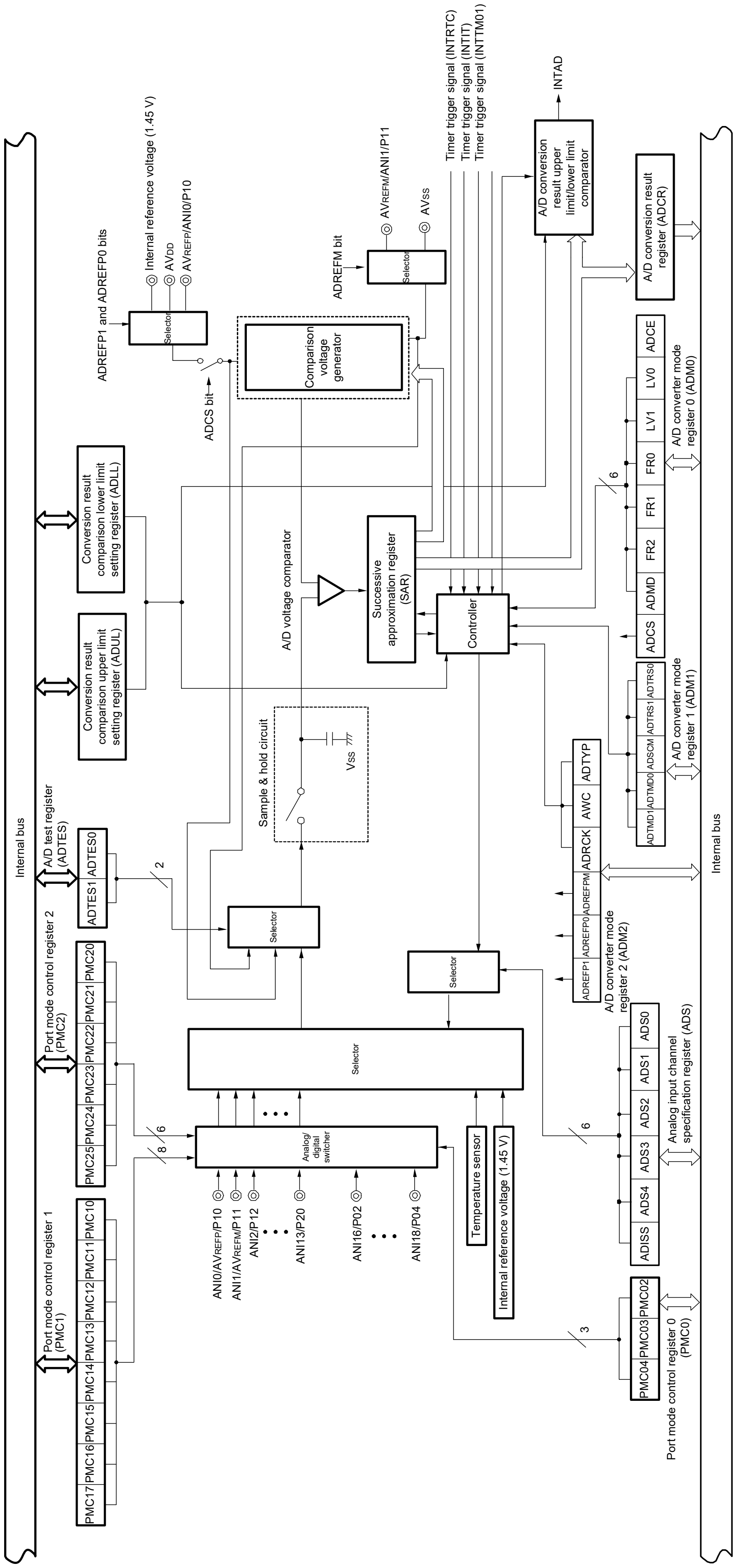
Various A/D conversion modes can be specified by using the mode combinations below.

Trigger mode	Software trigger	Conversion is started by software manipulation.
	Hardware trigger no-wait mode	Conversion is started by detecting a hardware trigger.
	Hardware trigger wait mode	The power is turned on by detecting a hardware trigger while the system is off and in the conversion standby state, and conversion is then started automatically after the A/D power supply stabilization wait time passes. 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.
Conversion operation mode	One-shot conversion mode	A/D conversion is performed on the selected channel once. Sequential four channels of ANI0 to ANI13 can be selected as analog input.
	Sequential conversion mode	A/D conversion is sequentially performed on the selected channels until it is stopped by software.

Operation Mode Note	Number of Sampling Clock	
Normal 1	11 f_{AD}	Set a value to the number of sampling clocks, at which the sampling capacitor is fully charged, depending on the output impedance of the analog input source.
Normal 2	23 f_{AD}	
Low-voltage 1	33 f_{AD}	
Low-voltage 2	187 f_{AD}	

Note The operation modes selectable differ depending on the analog input channel, AV_{DD} voltage, trigger mode, and f_{CLK}. For details, **Tables 14 - 3** to **14 - 6 A/D Conversion Time Selection**.

Figure 14 - 1 Block Diagram of A/D Converter



Remark Analog input pin for Figure 14 - 1 when a 48-pin product is used.

14.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

(1) ANI0-ANI13 and ANI16-ANI18 pins

These are the analog input pins of the 17 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 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 output from the voltage tap of the comparison voltage generator with the sampled voltage value. If the analog input voltage is found to be greater than the reference voltage (1/2 AV_{REF}) as a result of the comparison, the most significant bit (MSB) of the successive approximation register (SAR) is set. If the analog input voltage is less than the reference voltage (1/2 AV_{REF}), the MSB bit of the SAR is reset.
After that, bit 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 been already 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 according to 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.
When performing A/D conversion at a resolution of 8 bits, the comparison continues until bit 4 of the SAR register.

Remark AV_{REF}: The + side reference voltage of the A/D converter
(This can be selected from AV_{REFP}, the internal reference voltage (1.45 V), and AV_{DD}.)

(4) Comparison voltage generator

The comparison voltage generator generates the comparison voltage input from an analog input pin.

(5) Successive approximation register (SAR)

The SAR register is a register that sets voltage tap data whose values from the comparison voltage generator match the voltage values of the analog input pins, 1 bit at a time starting from the most significant bit (MSB).

If data is set in the SAR register all the way to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register (conversion results) are held in the A/D conversion result register (ADCR). When all the specified A/D conversion operations have ended, an A/D conversion end interrupt request signal (INTAD) is generated.

(6) 12-bit A/D conversion result register (ADCR)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCR register holds the A/D conversion result in its lower 12 bits (the higher 4 bits are fixed to 0).

(7) 8-bit A/D conversion result register (ADCRH)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCRH register stores the higher 8 bits of the A/D conversion result.

(8) Controller

This circuit controls the conversion time of an input analog signal that is to be converted into a digital signal, as well as starting and stopping of the conversion operation. When A/D conversion has been completed, this controller generates an interrupt request signal (INTAD) through the A/D conversion result upper limit/lower limit comparator.

(9) AVREFP pin

This pin inputs an external reference voltage (AVREFP).

If using AVREFP as the + side 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 ANI13 and ANI16 to ANI18 are converted to digital signals based on the voltage applied between AVREFP and the – side reference voltage (AVREFM/AVSS).

In addition to AVREFP, it is possible to select AVDD or the internal reference voltage (1.45 V) as the + side reference voltage of the A/D converter.

(10) AVREFM pin

This pin inputs an external reference voltage (AVREFM). If using AVREFM as the – side reference voltage of the A/D converter, set the ADREFM bit of the ADM2 register to 1.

In addition to AVREFM, it is possible to select AVSS as the – side reference voltage of the A/D converter.

14.3 Registers Controlling A/D Converter

The A/D converter uses the following registers.

- 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 A/D conversion result register (ADCR)
- 8-bit A/D conversion result register (ADCRH)
- Analog input channel specification register (ADS)
- Conversion result comparison upper limit setting register (ADUL)
- Conversion result comparison lower limit setting register (ADLL)
- A/D test register (ADTES)
- Port mode control registers 0 to 2 (PMC0 to PMC2)
- Port mode registers 0 to 2 (PM0 to PM2)

14.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the A/D converter is used, be sure to set bit 5 (ADCEN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 14 - 2 Format of Peripheral enable register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	4	3	<2>	1	<0>
PER0	RTCWEN	0	ADCEN	0	0	SAU0EN	0	TAU0EN

ADCEN	Control of A/D converter input clock supply
0	Stops input clock supply. • SFR used by the A/D converter cannot be written.
1	Enables input clock supply. • SFR used by the A/D converter can be read/written.

Caution 1. When setting the A/D converter, be sure to set the following registers first while the ADCEN bit is set to 1.

If ADCEN = 0, writing to the A/D converter control registers is ignored (except for port mode registers 0 to 2(PM0 to PM2), and port mode control registers 0 to 2 (PMC0 to PMC2).

- **A/D converter mode register 0 (ADM0)**
- **A/D converter mode register 1 (ADM1)**
- **A/D converter mode register 2 (ADM2)**
- **12-bit A/D conversion result register (ADCR)**
- **8-bit A/D conversion result register (ADCRH)**
- **Analog input channel specification register (ADS)**
- **Conversion result comparison upper limit setting register (ADUL)**
- **Conversion result comparison lower limit setting register (ADLL)**
- **A/D test register (ADTES)**

Caution 2. Be sure to clear the following bits to 0.

Bits 1, 3, 4, and 6

14.3.2 Peripheral reset control register 0 (PRR0)

This register is used for individual reset control of each peripheral hardware.
This MCU controls reset and reset release of each peripheral hardware supported by the PRR0 register.
To reset the A/D converter, be sure to set bit 5 (ADCRES) to 1.
The PRR0 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 14 - 3 Format of Peripheral reset control register 0 (PRR0)

Address: F00F1H After reset: 00H R/W

Symbol	7	6	<5>	4	3	<2>	1	<0>
PRR0	0	0	ADCRES	0	0	SAU0RES	0	TAU0RES

ADCRES	Reset control of A/D converter
0	A/D converter reset release
1	A/D converter reset state

14.3.3 A/D converter mode register 0 (ADM0)

This register sets the conversion time for analog input to be A/D converted, and starts/stops conversion.

The ADM0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 14 - 4 Format of A/D converter mode register 0 (ADM0)

Address: FFF30H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM0	ADCS	ADMD	FR2 Note 1	FR1 Note 1	FR0 Note 1	LV1 Note 1	LV0 Note 1	ADCE

ADCS	A/D conversion operation control
0	Stops conversion operation [When read] Conversion stopped/standby status
1	Enables conversion operation [When read] While in the software trigger mode: Conversion operation status While in the hardware trigger wait mode: A/D power supply stabilization wait status + conversion operation status

ADMD	Specification of the A/D conversion channel selection mode
0	Select mode
1	Scan mode

ADCE	A/D voltage comparator operation control Note 2
0	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 **Tables 14 - 3** to **14 - 6 A/D Conversion Time Selection**.

Note 2. While in the software trigger mode or hardware trigger no-wait mode, the operation of the A/D voltage comparator is controlled by the ADCS and ADCE bits, and it takes stabilization wait status from the start of operation for the operation to stabilize. Therefore, when the ADCS bit is set to 1 after stabilization wait status or more has elapsed from the time ADCE bit is set to 1, the conversion result at that time has priority over the first conversion result. If the ADCS bit was set to 1 before the stabilization time elapsed, ignore the first conversion data.

[Stabilization wait status]

If a high-accuracy channel is selected as the analog input channel: 0.5 μs

If a test mode setting (ADTES1 bit of ADTES register = 1) is selected: 0.5 μs

If a standard channel is selected as the analog input channel: 2 μs

If a temperature sensor output voltage/internal reference voltage are selected as the analog input channel: (ADISS bit of ADS register = 1): 2 μs

Caution 1. Change the ADMD, FR2 to FR0, LV1, and LV0 bits while in the conversion stopped status (ADCS = 0, ADCE = 0).

Caution 2. Setting ADCS = 1, ADCE = 0 is prohibited.

Caution 3. Do not change the ADCE and ADCS bits from 0 to 1 at the same time by using an 8-bit manipulation instruction. Be sure to set these bits in the order described in 14.7 A/D Converter Setup Flowchart.

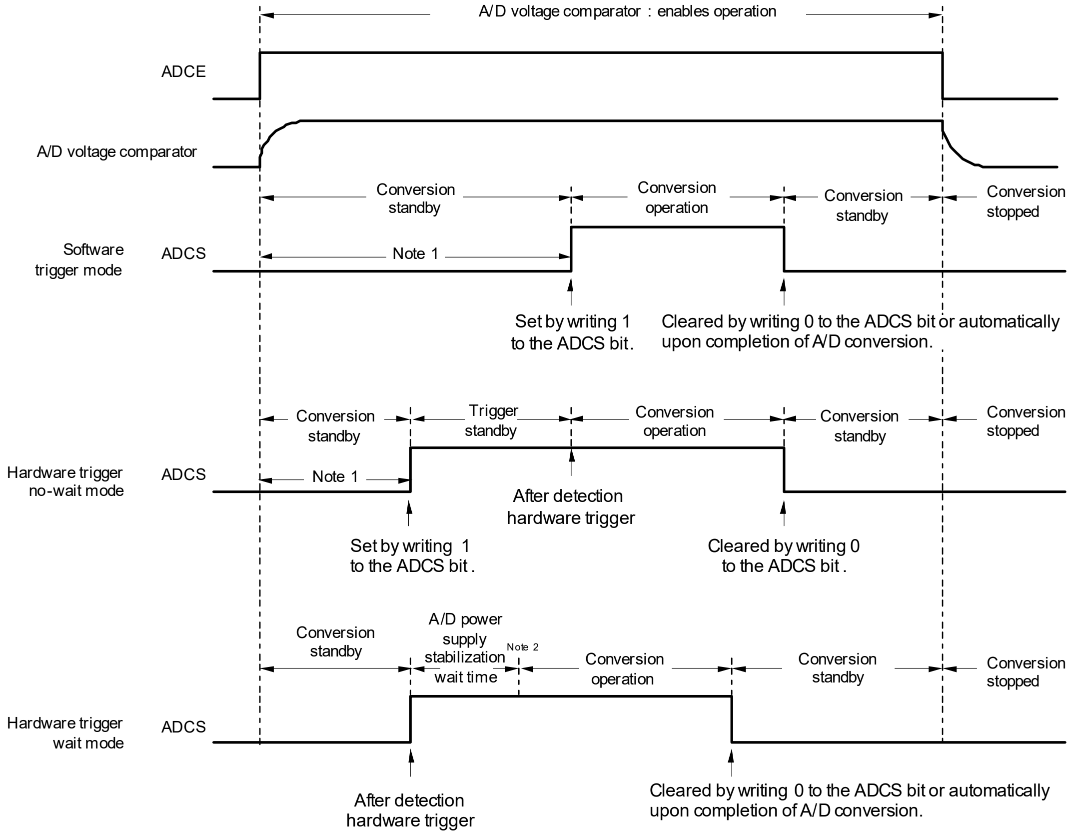
Table 14 - 1 Settings of ADCS and ADCE Bits

ADCS	ADCE	A/D Conversion Operation
0	0	Conversion stopped state
0	1	Conversion standby state
1	0	Setting prohibited
1	1	Conversion-in-progress state

Table 14 - 2 Setting and Clearing Conditions for ADCS Bit

A/D Conversion Mode			Set Conditions	Clear Conditions
Software trigger	Select mode	Sequential conversion mode	When 1 is written to ADCE and ADCS	When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when A/D conversion ends.
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when conversion ends on the specified four channels.
Hardware trigger no-wait mode	Select mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		When 0 is written to ADCS
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		When 0 is written to ADCS
Hardware trigger wait mode	Select mode	Sequential conversion mode	When 1 is written to ADCE and a hardware trigger is input	When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when A/D conversion ends.
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when conversion ends on the specified four channels.

Figure 14 - 5 Timing Chart When A/D Voltage Comparator Is Used



Note 1. While in the software trigger mode or hardware trigger no-wait mode, the time from the rising of the ADCE bit to the rising of the ADCS bit must be following time or longer to stabilize the internal circuit.
[Stabilization wait status]
If a high-accuracy channel is selected as the analog input channel: 0.5 μs
If a test mode setting (ADTES1 bit of ADTES register = 1) is selected: 0.5 μs
If a standard channel is selected as the analog input channel: 2 μs
If a temperature sensor output voltage/internal reference voltage are selected as the analog input channel: (ADISS bit of ADS register = 1): 2 μs

Note 2. For the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 2, and 3 in scan mode, the A/D power supply stabilization wait time do not occur after a hardware trigger is detected.

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 status.

Caution 2. While in the one-shot conversion mode of the hardware trigger no-wait mode, the ADCS flag is not automatically cleared to 0 when A/D conversion ends. Instead, 1 is retained.

Caution 3. Only rewrite the value of the ADCE bit when ADCS = 0 (while in the conversion stopped/conversion standby status).

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 + conversion start time + A/D conversion time
Hardware trigger wait mode: 2 f_{CLK} clock + conversion start time + A/D power supply stabilization wait time + A/D conversion time

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 14 - 3 A/D Conversion Time Selection (1/4)
(1) 12-bit resolution mode (ADTYP = 0) When there is no stabilization wait time (software trigger mode/hardware trigger no-wait mode)

A/D Converter Mode Register 0 (ADM0)					Conversion Clock (f_{AD})	Number of Conversion Clock (Number of Sampling Clock)	Conversion Clock Number (f_{CLK})	Conversion Time Selection				
								AV_{DD} = 1.6 to 3.6 V	AV_{DD} = 1.6 to 3.6 V	AV_{DD} = 1.8 to 3.6 V	AV_{DD} = 2.4 to 3.6 V	AV_{DD} = 2.7 to 3.6 V
FR2	FR1	FR0	LV1	LV0				f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 24 MHz
0	0	0	0	0	f_{CLK}/32	54 f_{AD} (number of sampling clock: 11 f_{AD})	1728/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	72 μs Note
0	0	1			f_{CLK}/16		864/f_{CLK}				54 μs Note	36 μs Note
0	1	0			f_{CLK}/8		432/f_{CLK}			54 μs Note	27 μs Note	18 μs Note
0	1	1			f_{CLK}/6		324/f_{CLK}			40.5 μs Note	20.25 μs Note	13.5 μs Note
1	0	0			f_{CLK}/5		270/f_{CLK}			33.75 μs Note	16.875 μs Note	11.25 μs Note
1	0	1			f_{CLK}/4		216/f_{CLK}		54 μs Note	27 μs Note	13.5 μs Note	9 μs Note
1	1	0			f_{CLK}/2		108/f_{CLK}		27 μs Note	13.5 μs Note	6.75 μs Note	4.5 μs Note
1	1	1			f_{CLK}/1		54/f_{CLK}	54 μs Note	13.5 μs Note	6.75 μs Note	3.375 μs Note	Setting prohibited
0	0	0	0	1	f_{CLK}/32	66 f_{AD} (number of sampling clock: 23 f_{AD})	2112/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	88 μs
0	0	1			f_{CLK}/16		1056/f_{CLK}				66 μs	44 μs
0	1	0			f_{CLK}/8		528/f_{CLK}			66 μs Note	33 μs	22 μs
0	1	1			f_{CLK}/6		396/f_{CLK}			49.5 μs Note	24.75 μs	16.5 μs
1	0	0			f_{CLK}/5		330/f_{CLK}			41.25 μs Note	20.625 μs	13.75 μs
1	0	1			f_{CLK}/4		264/f_{CLK}		66 μs Note	33 μs Note	16.5 μs	11 μs
1	1	0			f_{CLK}/2		132/f_{CLK}		33 μs Note	16.5 μs Note	8.25 μs	5.5 μs
1	1	1			f_{CLK}/1		66/f_{CLK}	66 μs Note	16.5 μs Note	8.25 μs Note	4.125 μs	Setting prohibited
0	0	0	1	0	f_{CLK}/32	76 f_{AD} (number of sampling clock: 33 f_{AD})	2432/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	101.33 μs
0	0	1			f_{CLK}/16		1216/f_{CLK}				76 μs	50.67 μs
0	1	0			f_{CLK}/8		608/f_{CLK}			76 μs	38 μs	25.33 μs
0	1	1			f_{CLK}/6		456/f_{CLK}			57 μs	28.5 μs	19 μs
1	0	0			f_{CLK}/5		380/f_{CLK}			47. 5 μs	23.75 μs	15.83 μs
1	0	1			f_{CLK}/4		304/f_{CLK}		76 μs Note	38 μs	19 μs	12.67 μs
1	1	0			f_{CLK}/2		152/f_{CLK}		38 μs Note	19 μs	9.5 μs	6.33 μs
1	1	1			f_{CLK}/1		76/f_{CLK}	76 μs Note	19 μs Note	9.5 μs	4.75 μs	Setting prohibited
0	0	0	1	1	f_{CLK}/32	230 f_{AD} (number of sampling clock: 187 f_{AD})	7360/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	306.67 μs
0	0	1			f_{CLK}/16		3680/f_{CLK}				230 μs	153.33 μs
0	1	0			f_{CLK}/8		1840/f_{CLK}			230 μs	115 μs	76.67 μs
0	1	1			f_{CLK}/6		1380/f_{CLK}			172.5 μs	86.25 μs	57.5 μs
1	0	0			f_{CLK}/5		1150/f_{CLK}			143.75 μs	71. 875 μs	47.92 μs
1	0	1			f_{CLK}/4		920/f_{CLK}		230 μs	115 μs	57.5 μs	38.33 μs
1	1	0			f_{CLK}/2		460/f_{CLK}		115 μs	57.5 μs	28.75 μs	19.17 μs
1	1	1			f_{CLK}/1		230/f_{CLK}	230 μs	57.5 μs	28.75 μs	14.375 μs	Setting prohibited

Note When using ANI16 to ANI18, setting this value is prohibited.

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 34.6.1 A/D converter characteristics.

Caution 2. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, while in the conversion stopped/conversion standby status (ADCS = 0, ADCE = 0).

Caution 3. The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

Caution 4. When software trigger mode/hardware trigger no-wait mode, specify the conversion time so that the following conditions are satisfied:

- **f_{AD} is used within a range of 1 to 16 MHz.**
- **When using ANI16 to ANI18, the A/D converter is used in the following range of AV_{DD}, in accordance with the settings of the LV1 and LV0 bits:**
  When LV1 = 0, LV0 = 0: Setting prohibited
  When LV1 = 0, LV0 = 1: 2.4 V ≤ AV_{DD} ≤ 3.6 V
  When LV1 = 1, LV0 = 0: 1.8 V ≤ AV_{DD} ≤ 3.6 V
  When LV1 = 1, LV0 = 1: 1.6 V ≤ AV_{DD} ≤ 3.6 V
- **If temperature sensor output voltage or internal reference voltage (ADISS bit of ADS register = 1) is specified for the analog input channel, the A/D converter is used in the following range of AV_{DD}:**
  When LV1 = 0, LV0 = 0: Setting prohibited
  When other than LV1 = 0, LV0 = 0: 2.4 V ≤ AV_{DD} ≤ 3.6 V

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 14 - 4 A/D Conversion Time Selection (2/4)

(2) 12-bit resolution mode (ADTYP = 0) When there is A/D power supply stabilization wait time (hardware trigger wait mode (except second and subsequent conversion in sequential conversion mode and conversion of channel specified by scan 1, 2, and 3 in scan mode Note 1))

A/D Converter Mode Register 0 (ADM0)					Conversion Clock (f_{AD})	Number of A/D power supply stabilization wait time	Number of Conversion Clock (Number of Sampling Clock)	A/D Power Supply Stabilization Wait Time +Conversion Time	Conversion Time Selection				
									AV_{DD} = 1.6 to 3.6 V	AV_{DD} = 1.6 to 3.6 V	AV_{DD} = 1.8 to 3.6 V	AV_{DD} = 2.4 to 3.6 V	AV_{DD} = 2.7 to 3.6 V
FR2	FR1	FR0	LV1	LV0					f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 24 MHz
0	0	0	0	0	f_{CLK}/32	4 f_{CLK}	54 f_{AD} (number of sampling clock: 11 f_{AD})	1732/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	72.1667 μs Note 2
0	0	1			f_{CLK}/16			868/f_{CLK}				54.25 μs Note 2	36.1667 μs Note 2
0	1	0			f_{CLK}/8			436/f_{CLK}			54.5 μs Note 2	27.25 μs Note 2	18.1667 μs Note 2
0	1	1			f_{CLK}/6			328/f_{CLK}			41 μs Note 2	20.5 μs Note 2	13.6667 μs Note 2
1	0	0			f_{CLK}/5			274/f_{CLK}			34.25 μs Note 2	17.125 μs Note 2	11.4167 μs Note 2
1	0	1			f_{CLK}/4			220/f_{CLK}		55 μs Note 2	27.5 μs Note 2	13.75 μs Note 2	9.1667 μs Note 2
1	1	0			f_{CLK}/2			112/f_{CLK}		28 μs Note 2	14 μs Note 2	7 μs Note 2	4.6667 μs Note 2
1	1	1			f_{CLK}/1	2 f_{CLK}		56/f_{CLK}	56 μs Note 2	14 μs Note 2	7 μs Note 2	3.5 μs Note 2	Setting prohibited
0	0	0	0	1	f_{CLK}/32	58 f_{CLK}	66 f_{AD} (number of sampling clock: 23 f_{AD})	2170/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	90.4167 μs
0	0	1			f_{CLK}/16			1114/f_{CLK}				69.625 μs	46.4167 μs
0	1	0			f_{CLK}/8			586/f_{CLK}			73.25 μs Note 2	36.625 μs	24.4167 μs
0	1	1			f_{CLK}/6			454/f_{CLK}			56.75 μs Note 2	28.375 μs	18.9167 μs
1	0	0			f_{CLK}/5			388/f_{CLK}			48.5 μs Note 2	24.25 μs	16.1667 μs
1	0	1			f_{CLK}/4			322/f_{CLK}		80.5 μs Note 2	40.25 μs Note 2	20.125 μs	13.4167 μs
1	1	0			f_{CLK}/2			190/f_{CLK}		47.5 μs Note 2	23.75 μs Note 2	11.875 μs	7.9167 μs
1	1	1			f_{CLK}/1	29 f_{CLK}		95/f_{CLK}	95 μs Note 2	23.75 μs Note 2	11.875 μs Note 2	5.9375 μs	Setting prohibited
0	0	0	1	0	f_{CLK}/32	15 f_{CLK}	76 f_{AD} (number of sampling clock: 33 f_{AD})	2447/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	101.958 μs Note 2
0	0	1			f_{CLK}/16			1231/f_{CLK}				76.9375 μs Note 2	51.292 μs Note 2
0	1	0			f_{CLK}/8			623/f_{CLK}			77.875 μs	38.9375 μs Note 2	25.958 μs Note 2
0	1	1			f_{CLK}/6			471/f_{CLK}			58.875 μs	29.4375 μs Note 2	19.625 μs Note 2
1	0	0			f_{CLK}/5			395/f_{CLK}			49.375 μs	24.6875 μs Note 2	16.458 μs Note 2
1	0	1			f_{CLK}/4			319/f_{CLK}		79.75 μs Note 2	39.875 μs	19.9375 μs Note 2	13.292 μs Note 2
1	1	0			f_{CLK}/2			167/f_{CLK}		41.75 μs Note 2	20.875 μs	10.4375 μs Note 2	6.958 μs Note 2
1	1	1			f_{CLK}/1			91/f_{CLK}	91 μs Note 2	22.75 μs Note 2	11.375 μs	5.6875 μs Note 2	Setting prohibited
0	0	0	1	1	f_{CLK}/32	8 f_{CLK}	230 f_{AD} (number of sampling clock: 187 f_{AD})	7368/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	307 μs Note 2
0	0	1			f_{CLK}/16			3688/f_{CLK}				230.5 μs Note 2	153.67 μs Note 2
0	1	0			f_{CLK}/8			1848/f_{CLK}			231 μs Note 2	115.5 μs Note 2	77 μs Note 2
0	1	1			f_{CLK}/6			1388/f_{CLK}			173.5 μs Note 2	86.75 μs Note 2	57.83 μs Note 2
1	0	0			f_{CLK}/5			1158/f_{CLK}			144.75 μs Note 2	72.375 μs Note 2	48.25 μs Note 2
1	0	1			f_{CLK}/4			928/f_{CLK}		232 μs	116 μs Note 2	58 μs Note 2	38.67 μs Note 2
1	1	0			f_{CLK}/2			468/f_{CLK}		117 μs	58.5 μs Note 2	29.25 μs Note 2	19.5 μs Note 2
1	1	1			f_{CLK}/1			238/f_{CLK}	238 μs	59.5 μs	29.75 μs Note 2	14.875 μs Note 2	Setting prohibited

Note 1. For the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 2, and 3 in scan mode, the conversion start time and stabilization wait time for A/D power supply do not occur after a hardware trigger is detected (see **Table 14 - 3**).

Note 2. When using ANI16 to ANI18, setting this value is prohibited.

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 34.6.1 A/D converter characteristics.

Caution 2. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, while in the conversion stopped (ADCS = 0, ADCE = 0).

Caution 3. The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

Caution 4. When hardware trigger wait mode, specify the conversion time so that the following conditions are satisfied:

- **f_{AD} is used within a range of 1 to 16 MHz.**
- **When using ANI16 to ANI18, the A/D converter is used in the following conditions:**
  When LV1 = 0, LV0 = 0: Setting prohibited
  When LV1 = 0, LV0 = 1: 2.4 V ≤ AV_{DD} ≤ 3.6 V
  When LV1 = 1, LV0 = 0: 1.8 V ≤ AV_{DD} ≤ 3.6 V, 1 MHz ≤ f_{CLK} ≤ 8 MHz
  When LV1 = 1, LV0 = 1: 1.6 V ≤ AV_{DD} ≤ 3.6 V, 1 MHz ≤ f_{CLK} ≤ 4 MHz
- **If temperature sensor output voltage or internal reference voltage (ADISS bit of ADS register = 1) is specified for the analog input channel, the A/D converter is used in the following conditions:**
  When LV1 = 0, LV0 = 0: Setting prohibited
  When LV1 = 0, LV0 = 1: 2.4 V ≤ AV_{DD} ≤ 3.6 V
  When LV1 = 1, LV0 = 0: 2.4 V ≤ AV_{DD} ≤ 3.6 V, 1 MHz ≤ f_{CLK} ≤ 8 MHz
  When LV1 = 1, LV0 = 1: 2.4 V ≤ AV_{DD} ≤ 3.6 V, 1 MHz ≤ f_{CLK} ≤ 4 MHz

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 14 - 5 A/D Conversion Time Selection (3/4)

(3) 8-bit resolution mode (ADTYP = 1) When there is no stabilization wait time (software trigger mode/hardware trigger no-wait mode)

A/D Converter Mode Register 0 (ADM0)					Conversion Clock (fAD)	Number of Conversion Clock (Number of Sampling Clock)	Conversion Clock Number (fCLK)	Conversion Time Selection				
								AVDD = 1.6 to 3.6 V	AVDD = 1.6 to 3.6 V	AVDD = 1.8 to 3.6 V	AVDD = 2.4 to 3.6 V	AVDD = 2.7 to 3.6 V
FR2	FR1	FR0	LV1	LV0				fCLK = 1 MHz	fCLK = 4 MHz	fCLK = 8 MHz	fCLK = 16 MHz	fCLK = 24 MHz
0	0	0	0	0	fCLK/32	41 fAD (number of sampling clock: 11 fAD)	1312/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	54.6667 μs Note
0	0	1			fCLK/16		656/fCLK				41 μs Note	27.3333 μs Note
0	1	0			fCLK/8		328/fCLK			41 μs Note	20.5 μs Note	13.6667 μs Note
0	1	1			fCLK/6		246/fCLK			30.75 μs Note	15.375 μs Note	10.25 μs Note
1	0	0			fCLK/5		205/fCLK			25.625 μs Note	12.8125 μs Note	8.5417 μs Note
1	0	1			fCLK/4		164/fCLK		41 μs Note	20.5 μs Note	10.25 μs Note	6.8333 μs Note
1	1	0			fCLK/2		82/fCLK		20.5 μs Note	10.25 μs Note	5.125 μs Note	3.4167 μs Note
1	1	1			fCLK/1		41/fCLK	41 μs Note	10.25 μs Note	5.125 μs Note	2.5625 μs Note	Setting prohibited
0	0	0	0	1	fCLK/32	53 fAD (number of sampling clock: 23 fAD)	1696/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	70.6667 μs
0	0	1			fCLK/16		848/fCLK				53 μs	35.3333 μs
0	1	0			fCLK/8		424/fCLK			53 μs Note	26.5 μs	17.6667 μs
0	1	1			fCLK/6		318/fCLK			39.75 μs Note	19.875 μs	13.25 μs
1	0	0			fCLK/5		265/fCLK			33.125 μs Note	16.5625 μs	11.0417 μs
1	0	1			fCLK/4		212/fCLK		53 μs Note	26.5 μs Note	13.25 μs	8.8333 μs
1	1	0			fCLK/2		106/fCLK		26.5 μs Note	13.25 μs Note	6.625 μs	4.4167 μs
1	1	1			fCLK/1		53/fCLK	53 μs Note	13.25 μs Note	6.625 μs Note	3.3125 μs	Setting prohibited
0	0	0	1	0	fCLK/32	63 fAD (number of sampling clock: 33 fAD)	2016/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	84.00 μs
0	0	1			fCLK/16		1008/fCLK				63 μs	42.00 μs
0	1	0			fCLK/8		504/fCLK			63 μs	31.5 μs	21.00 μs
0	1	1			fCLK/6		378/fCLK			47.25 μs	23.625 μs	15.75 μs
1	0	0			fCLK/5		315/fCLK			39.375 μs	19.6875 μs	13.13 μs
1	0	1			fCLK/4		252/fCLK		63 μs Note	31.5 μs	15.75 μs	10.50 μs
1	1	0			fCLK/2		126/fCLK		31.5 μs Note	15.75 μs	7.875 μs	5.25 μs
1	1	1			fCLK/1		63/fCLK	63 μs Note	15.75 μs Note	7.875 μs	3.9375 μs	Setting prohibited
0	0	0	1	1	fCLK/32	217 fAD (number of sampling clock: 187 fAD)	6944/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	289.33 μs
0	0	1			fCLK/16		3472/fCLK				217 μs	144.67 μs
0	1	0			fCLK/8		1736/fCLK			217 μs	108.5 μs	72.33 μs
0	1	1			fCLK/6		1302/fCLK			162.75 μs	81.375 μs	54.25 μs
1	0	0			fCLK/5		1085/fCLK			135.625 μs	67.8125 μs	45.21 μs
1	0	1			fCLK/4		868/fCLK		217 μs	108.5 μs	54.25 μs	36.17 μs
1	1	0			fCLK/2		434/fCLK		108.5 μs	54.25 μs	27.125 μs	18.08 μs
1	1	1			fCLK/1		217/fCLK	217 μs	54.25 μs	27.125 μs	13.5625 μs	Setting prohibited

Note When using ANI16 to ANI18, setting this value is prohibited.

Caution 1. **The A/D conversion time must also be within the relevant range of conversion times (tCONV) described in 34.6.1 A/D converter characteristics.**

Caution 2. **When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, while in the conversion stopped (ADCS = 0, ADCE = 0).**

Caution 3. **The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.**

Caution 4. **When software trigger mode/hardware trigger no-wait mode, specify the conversion time so that the following conditions are satisfied:**

- **fAD is used within a range of 1 to 16 MHz.**
- **When using ANI16 to ANI18, the A/D converter is used in the following range of AVDD, in accordance with the settings of the LV1 and LV0 bits:**
  When LV1 = 0, LV0 = 0: Setting prohibited
  When LV1 = 0, LV0 = 1: 2.4 V ≤ AVDD ≤ 3.6 V
  When LV1 = 1, LV0 = 0: 1.8 V ≤ AVDD ≤ 3.6 V
  When LV1 = 1, LV0 = 1: 1.6 V ≤ AVDD ≤ 3.6 V
- **If temperature sensor output voltage or internal reference voltage (ADISS bit of ADS register = 1) is specified for the analog input channel, the A/D converter is used in the following range of AVDD:**
  When LV1 = 0, LV0 = 0: Setting prohibited
  When other than LV1 = 0, LV0 = 0: 2.4 V ≤ AVDD ≤ 3.6 V

Remark fCLK: CPU/peripheral hardware clock frequency

Table 14 - 6 A/D Conversion Time Selection (4/4)

(4) 8-bit resolution mode (ADTYP = 1) When there is A/D power supply stabilization wait time (hardware trigger wait mode (except second and subsequent conversion in sequential conversion mode and conversion of channel specified by scan 1, 2, and 3 in scan mode Note 1))

A/D Converter Mode Register 0 (ADM0)					Conversion Clock (fAD)	Number of A/D power supply stabilization wait time	Number of Conversion Clock (Number of Sampling Clock)	A/D Power Supply Stabilization Wait Time +Conversion Time	Conversion Time Selection				
									AVDD = 1.6 to 3.6 V	AVDD = 1.6 to 3.6 V	AVDD = 1.8 to 3.6 V	AVDD = 2.4 to 3.6 V	AVDD = 2.7 to 3.6 V
FR2	FR1	FR0	LV1	LV0					fCLK = 1 MHz	fCLK = 4 MHz	fCLK = 8 MHz	fCLK = 16 MHz	fCLK = 24 MHz
0	0	0	0	0	fCLK/32	4 fCLK	41 fAD (number of sampling clock: 11 fAD)	1316/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	54.8333 μs Note 2
0	0	1			fCLK/16			660/fCLK				41.25 μs Note 2	27.5000 μs Note 2
0	1	0			fCLK/8			332/fCLK			41.5 μs Note 2	20.75 μs Note 2	13.8333 μs Note 2
0	1	1			fCLK/6			250/fCLK			31.25 μs Note 2	15.625 μs Note 2	10.4167 μs Note 2
1	0	0			fCLK/5			209/fCLK			26.125 μs Note 2	13.0625 μs Note 2	8.7083 μs Note 2
1	0	1			fCLK/4			168/fCLK		42 μs Note 2	21 μs Note 2	10.5 μs Note 2	7.0000 μs Note 2
1	1	0			fCLK/2			86/fCLK		21.5 μs Note 2	10.75 μs Note 2	5.375 μs Note 2	3.5833 μs Note 2
1	1	1			fCLK/1	2 fCLK		43/fCLK	43 μs Note 2	10.75 μs Note 2	5.375 μs Note 2	2.6875 μs Note 2	Setting prohibited
0	0	0	0	1	fCLK/32	58 fCLK	53 fAD (number of sampling clock: 23 fAD)	1754/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	73.0833 μs Note 2
0	0	1			fCLK/16			906/fCLK				56.625 μs	37.7500 μs Note 2
0	1	0			fCLK/8			482/fCLK			60.25 μs Note 2	30.125 μs	20.0833 μs Note 2
0	1	1			fCLK/6			376/fCLK			47 μs Note 2	23.5 μs	15.6667 μs Note 2
1	0	0			fCLK/5			323/fCLK			40.375 μs Note 2	20.1875 μs	13.4583 μs Note 2
1	0	1			fCLK/4			270/fCLK		67.5 μs Note 2	33.75 μs Note 2	16.875 μs	11.2500 μs Note 2
1	1	0			fCLK/2			164/fCLK		41 μs Note 2	20.5 μs Note 2	10.25 μs	6.8333 μs Note 2
1	1	1			fCLK/1	29 fCLK		82/fCLK	82 μs Note 2	20.5 μs Note 2	10.25 μs Note 2	5.125 μs	Setting prohibited
0	0	0	1	0	fCLK/32	15 fCLK	63 fAD (number of sampling clock: 33 fAD)	2031/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	84.625 μs Note 2
0	0	1			fCLK/16			1023/fCLK				63.9375 μs Note 2	42.625 μs Note 2
0	1	0			fCLK/8			519/fCLK			64.875 μs	32.4375 μs Note 2	21.625 μs Note 2
0	1	1			fCLK/6			393/fCLK			49.125 μs	24.5625 μs Note 2	16.375 μs Note 2
1	0	0			fCLK/5			330/fCLK			41.25 μs	20.625 μs Note 2	13.75 μs Note 2
1	0	1			fCLK/4			267/fCLK		66.75 μs Note 2	33.375 μs	16.6875 μs Note 2	11.125 μs Note 2
1	1	0			fCLK/2			141/fCLK		35.25 μs Note 2	17.625 μs	8.8125 μs Note 2	5.875 μs Note 2
1	1	1			fCLK/1			78/fCLK	78 μs Note 2	19.5 μs Note 2	9.75 μs	4.875 μs Note 2	Setting prohibited
0	0	0	1	1	fCLK/32	8 fCLK	217 fAD (number of sampling clock: 187 fAD)	6952/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	289.67 μs Note 2
0	0	1			fCLK/16			3480/fCLK				217.5 μs Note 2	145 μs Note 2
0	1	0			fCLK/8			1744/fCLK			218 μs Note 2	109 μs Note 2	72.67 μs Note 2
0	1	1			fCLK/6			1310/fCLK			163.75 μs Note 2	81.875 μs Note 2	54.58 μs Note 2
1	0	0			fCLK/5			1093/fCLK			136.625 μs Note 2	68.3125 μs Note 2	45.54 μs Note 2
1	0	1			fCLK/4			876/fCLK		219 μs	109.5 μs Note 2	54.75 μs Note 2	36.5 μs Note 2
1	1	0			fCLK/2			442/fCLK		110.5 μs	55.25 μs Note 2	27.625 μs Note 2	18.42 μs Note 2
1	1	1			fCLK/1			225/fCLK	225 μs	56.25 μs	28.125 μs Note 2	14.0625 μs Note 2	Setting prohibited

Note 1. For the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 2, and 3 in scan mode, the conversion start time and stabilization wait time for A/D power supply do not occur after a hardware trigger is detected (see **Table 14 - 5**).

Note 2. When using ANI16 to ANI18, setting this value is prohibited.

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 34.6.1 A/D converter characteristics.

Caution 2. When rewriting the FR2 to FR0, LV1, and LV0 bits to other than the same data, while in the conversion stopped (ADCS = 0, ADCE = 0).

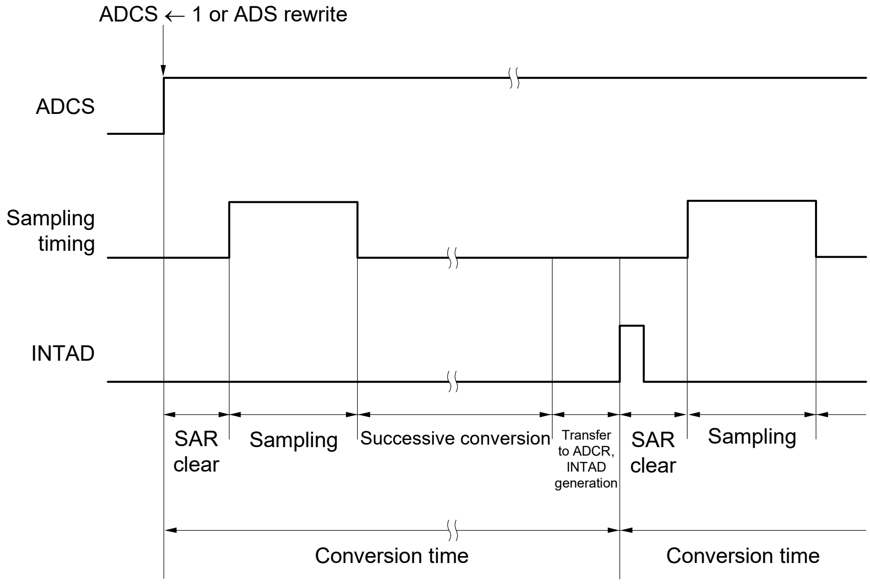
Caution 3. The above conversion time does not include clock frequency errors. Select conversion time, taking clock frequency errors into consideration.

Caution 4. When hardware trigger wait mode, specify the conversion time so that the following conditions are satisfied:

- **f_{AD} is used within a range of 1 to 16 MHz.**
- **When using ANI16 to ANI18, the A/D converter is used in the following conditions:**
  When LV1 = 0, LV0 = 0: Setting prohibited
  When LV1 = 0, LV0 = 1: 2.4 V ≤ AV_{DD} ≤ 3.6 V
  When LV1 = 1, LV0 = 0: 1.8 V ≤ AV_{DD} ≤ 3.6 V, 1 MHz ≤ f_{CLK} ≤ 8 MHz
  When LV1 = 1, LV0 = 1: 1.6 V ≤ AV_{DD} ≤ 3.6 V, 1 MHz ≤ f_{CLK} ≤ 4 MHz
- **If temperature sensor output voltage or internal reference voltage (ADISS bit of ADS register = 1) is specified for the analog input channel, the A/D converter is used in the following conditions:**
  When LV1 = 0, LV0 = 0: Setting prohibited
  When LV1 = 0, LV0 = 1: 2.4 V ≤ AV_{DD} ≤ 3.6 V
  When LV1 = 1, LV0 = 0: 2.4 V ≤ AV_{DD} ≤ 3.6 V, 1 MHz ≤ f_{CLK} ≤ 8 MHz
  When LV1 = 1, LV0 = 1: 2.4 V ≤ AV_{DD} ≤ 3.6 V, 1 MHz ≤ f_{CLK} ≤ 4 MHz

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Figure 14 - 6 A/D Converter Sampling and A/D Conversion Timing (Example for Software Trigger Mode)



14.3.4 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.
Reset signal generation clears this register to 00H.

Figure 14 - 7 Format of A/D converter mode register 1 (ADM1)

Address: FFF32H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADM1	ADTMD1	ADTMD0	ADSCM	0	0	0	ADTRS1	ADTRS0

ADTMD1	ADTMD0	Selection of the A/D conversion trigger mode
0	×	Software trigger mode
1	0	Hardware trigger no-wait mode
1	1	Hardware trigger wait mode

ADSCM	Specification of the A/D conversion mode
0	Sequential conversion mode
1	One-shot conversion mode

ADTRS1	ADTRS0	Selection of the hardware trigger signal
0	0	End of timer channel 1 count or capture interrupt signal (INTTM01)
0	1	Event signal selected by ELC
1	0	Real-time clock 2 interrupt signal (INTRTC)
1	1	12-bit interval timer interrupt signal (INTIT)

Caution 1. Only rewrite the value of the ADM1 register while conversion operation is stopped (which is indicated by the ADCS and ADCE bits of A/D converter mode register 0 (ADM0) being 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 f_{CLK} clock + conversion start time + A/D conversion time
Hardware trigger wait mode: 2 f_{CLK} clock + conversion start time + A/D power supply stabilization wait time + A/D conversion time

Caution 3. In modes other than SNOOZE function, input of the next INTRTC or INTIT will not be recognized as a valid hardware trigger for up to four f_{CLK} cycles after the first INTRTC or INTIT is input.

Remark 1. ×: don't care
Remark 2. f_{CLK}: CPU/peripheral hardware clock frequency

14.3.5 A/D converter mode register 2 (ADM2)

This register is used to select the A/D converter reference voltage, check the upper limit and lower limit A/D conversion result values, select the resolution, and specify whether to use the SNOOZE mode.

The ADM2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 14 - 8 Format of A/D converter mode register 2 (ADM2) (1/2)

Address: F0010H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	1	<0>
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

ADREFP1	ADREFP0	Selection of the + side reference voltage source of the A/D converter
0	0	Supplied from AV_{DD}
0	1	Supplied from AV_{REFP}/ANI0
1	0	Supplied from the internal reference voltage (1.45 V) Note
1	1	Setting prohibited

• When ADREFP1 or ADREFP0 bit is rewritten, this must be configured in accordance with the following procedures.
(1) Set ADCE = 0
(2) Change the values of ADREFP1 and ADREFP0
(3) Stabilization wait time (A)
(4) Set ADCE = 1
(5) Stabilization wait time (B)

The stabilization wait time indicated by (3) is required when the value of the ADREFP1 and ADREFP0 bits is changed.
When ADREFP1 and ADREFP0 are changed to 1 and 0: A = 10 μs
When ADREFP1 and ADREFP0 are changed to 0 and 0 or 0 and 1: A = 1 μs
The stabilization wait time indicated by (5) is required when the value of the ADCE bit is changed to 1.
If a high-accuracy channel is selected as the analog input channel: 0.5 μs
If a test mode setting (ADTES1 bit of ADTES register = 1) is selected: 0.5 μs
If a standard channel is selected as the analog input channel: 2 μs
If a temperature sensor output voltage/internal reference voltage are selected as the analog input channel: (ADISS bit of ADS register = 1): 2 μs
After (5) stabilization time, start the A/D conversion.

• When ADREFP1 and ADREFP0 are set to 1 and 0, respectively, A/D conversion cannot be performed on the temperature sensor output voltage and internal reference voltage.
Be sure to perform A/D conversion while ADISS = 0.

Note The operating voltage of the microcontroller must be at least 1.8 V if you wish to use this setting.

Caution 1. Rewrite the value of the ADM2 register while conversion 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 on the subsystem clock. Also, if the ADREFP1 bit is set to 1, the A/D converter reference voltage current (I_{ADREF}) indicated in 34.3.2 Supply current characteristics will be added to the current consumption when shifting to HALT mode while the CPU is operating on the main system clock.

Caution 3. When using AV_{REFP} and AV_{REFM}, specify ANI0 and ANI1 as the analog input channels and specify input mode by using the port mode register.

Figure 14 - 9 Format of A/D converter mode register 2 (ADM2) (2/2)

Address: F0010H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	1	<0>
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

ADREFM	Selection of the – side reference voltage source of the A/D converter
0	Supplied from AV_{SS}
1	Supplied from AV_{REFM}/ANI1

ADRCK	Checking the upper limit and lower limit conversion result values
0	The A/D conversion end interrupt request signal (INTAD) is output when the ADLL register ≤ the ADCR register ≤ the ADUL register (AREA1).
1	The A/D conversion end interrupt request signal (INTAD) is output when the ADCR register < the ADLL register (AREA2) or the ADUL register < the ADCR register (AREA3).
Figure 14 - 10 shows the generation range of the A/D conversion end interrupt request signal (INTAD) for AREA1 to AREA3.	

AWC	Specification of the SNOOZE mode
0	Do not use the SNOOZE mode function.
1	Use the SNOOZE mode function.
When there is a hardware trigger signal in the STOP mode, the STOP mode is exited, and A/D conversion is performed without operating the CPU (the SNOOZE mode). • The SNOOZE mode function can only be specified when the high-speed on-chip oscillator clock is selected for the CPU/peripheral hardware clock (f_{CLK}). If any other clock is selected, specifying this mode is prohibited. • Using the SNOOZE mode function in the software trigger mode or hardware trigger no-wait mode is prohibited. • Using the SNOOZE mode function in the sequential conversion mode is prohibited. • When using the SNOOZE mode function, specify a hardware trigger interval of at least "shift time to SNOOZE mode Note 1 + A/D power supply stabilization wait time + A/D conversion time +2 f_{CLK} clock" • Even when using 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 or normal operation.	

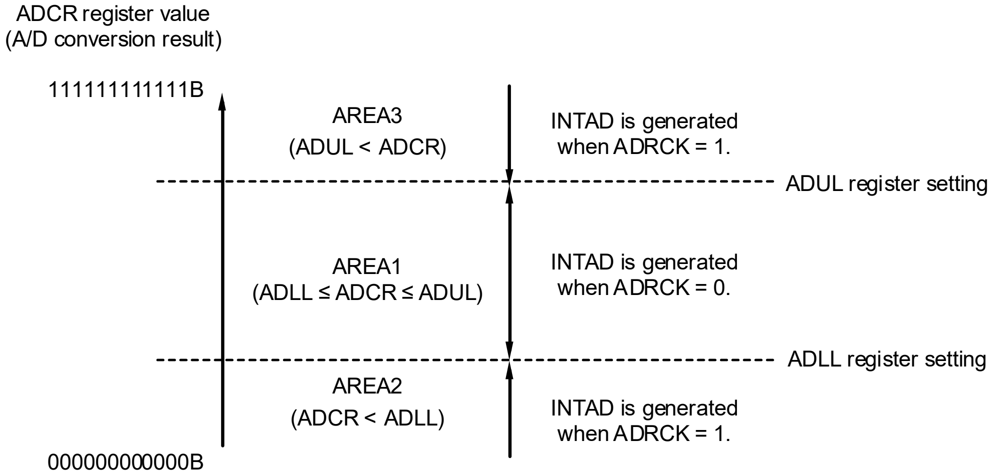
ADTYP	Selection of the A/D conversion resolution
0	12-bit resolution Note 2
1	8-bit resolution

Note 1. Refer to "Transition time from STOP mode to SNOOZE mode:" in **23.3.3 SNOOZE mode**.

Note 2. The valid resolution differs depending on the voltage conditions of AV_{DD} and AV_{REFP}. For details, see **34.6.1 A/D converter characteristics**.

Caution **Rewrite the value of the ADM2 register while conversion operation is stopped (ADCS = 0, ADCE = 0).**

Figure 14 - 10 ADRCK Bit Interrupt Signal Generation Range



Remark If INTAD does not occur, the A/D conversion result is not stored in the ADCR or ADCRH register.

14.3.6 12-bit A/D conversion result register (ADCR)

This register is a 16-bit register that stores the A/D conversion result. The higher 4 bits are fixed to 0. Each time A/D conversion ends, each time A/D conversion ends, the value of ADSAR[11:0] is stored in the A/D conversion result register (note that whether to store this value is determined by the setting of the ADRCK bit of the ADM2 register and by the settings of the ADUL and ADLL registers). The higher 4 bits of the conversion result are stored in FFF1FH and the lower 8 bits are stored in the lower 4 bits of FFF1EH Note.

The ADCR register can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Note If the A/D conversion result is outside the range specified by using the A/D conversion comparison function (the value specified by the ADRCK bit of the ADM2 register and ADUL/ADLL registers; see **Figure 14 - 10**), the result is not stored.

Caution The valid resolution differs depending on the voltage conditions of AV_{DD} and AV_{REFP}. For details, see 34.6.1 A/D converter characteristics.

Remark 1. When using the converter with a resolution of 10 bits, select the 12-bit resolution mode (ADTYP = 0). Use the higher 10 bits of the conversion result. Do not use the lower 2 bits.

Remark 2. When using the converter with a resolution of 8 bits, select the 8-bit resolution mode (ADTYP = 1). Do not use the lower 4 bits of the ADCR register.
The higher 8 bits of the conversion result can be read by using the ADCRH register.

Figure 14 - 11 Format of 12-bit A/D conversion result register (ADCR)

Address: FFF1EH, FFF1FH After reset: 0000H R

Symbol	FFF1FH								FFF1EH							
ADCR	0	0	0	0												

Caution 1. When writing to the A/D converter mode register 0 (ADM0) and analog input channel specification register (ADS), the contents of the ADCR register may become undefined. Read the conversion result following conversion completion before writing to the ADM0 and ADS registers. Using timing other than the above may cause an incorrect conversion result to be read.

Caution 2. If INTAD does not occur, the A/D conversion result is not stored in the ADCR register.

14.3.7 8-bit A/D conversion result register (ADCRH)

This register is an 8-bit register that indicates bits [11:4] of the ADCR register. The higher 8 bits of 12-bit resolution are stored [Note].

The ADCRH register can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Note If the A/D conversion result is outside the range specified by using the A/D conversion comparison function (the value specified by the ADRCK bit of the ADM2 register and ADUL/ADLL registers; see **Figure 14 - 10**), the result is not stored.

Figure 14 - 12 Format of 8-bit A/D conversion result register (ADCRH)

Address: FFF1FH [Note] After reset: 00H R

Symbol	FFF1FH								FFF1EH							
ADCRH	0	0	0	0												
					ADCRH											

Note The ADCRH data (the lower 4 bits of FFF1FH + the higher 4 bits of FFF1EH) is to be read as a FFF1FH address.

Caution 1. When writing to the A/D converter mode register 0 (ADM0) and the analog input channel specification register (ADS), the contents of the ADCRH register may become undefined. Read the conversion result following conversion completion before writing to the ADM0 and ADS registers. Using timing other than the above may cause an incorrect conversion result to be read.

Caution 2. If INTAD does not occur, the A/D conversion result is not stored in the ADCRH register.

14.3.8 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.

Reset signal generation clears this register to 00H.

Figure 14 - 13 Format of Analog input channel specification register (ADS)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

• Select mode (ADMD = 0)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P10/ANI0/AVREFP pin
0	0	0	0	0	1	ANI1	P11/ANI1/AVREFM pin
0	0	0	0	1	0	ANI2	P12/ANI2 pin
0	0	0	0	1	1	ANI3	P13/ANI3 pin
0	0	0	1	0	0	ANI4	P14/ANI4 pin
0	0	0	1	0	1	ANI5	P15/ANI5 pin
0	0	0	1	1	0	ANI6	P16/ANI6 pin
0	0	0	1	1	1	ANI7	P17/ANI7 pin
0	0	1	0	0	0	ANI8	P25/ANI8 pin
0	0	1	0	0	1	ANI9	P24/ANI9 pin
0	0	1	0	1	0	ANI10	P23/ANI10 pin
0	0	1	0	1	1	ANI11	P22/ANI11 pin
0	0	1	1	0	0	ANI12	P21/ANI12 pin
0	0	1	1	0	1	ANI13	P20/ANI13 pin
0	1	0	0	0	1	ANI17	P02/ANI17 pin
0	1	0	0	1	0	ANI18	P04/ANI18 pin
1 Note 1	0	0	0	0	0 Note 2	—	Temperature sensor output
1 Note 1	0	0	0	0	1 Note 3	—	Internal reference voltage output (1.45 V)
Other than the above						Setting prohibited	

Note 1. When setting the ADISS bit to 1 after having previously set it to 1 (i.e. in the sequence ADISS = 1 → 0 → 1), at least 200 μs must elapse after the setting of the ADISS bit was 1 before it is again set to 1.

Note 2. When setting the ADS0 bit to 0 after having previously set it to 0 while the setting of the ADISS bit is 1 (i.e. in the sequence (ADISS, ADS0) = (1,0) → (1,1) → (1,0)), at least 100 μs must elapse after the setting of the ADS0 bit was 0 before it is again set to 0.

Note 3. When setting the ADS0 bit to 1 after having previously set it to 1 while the setting of the ADISS bit is 1 (i.e. in the sequence (ADISS, ADS0) = (1,1) → (1,0) → (1,1)), at least 100 μs must elapse after the setting of the ADS0 bit was 1 before it is again set to 1.

• Scan mode (ADMD = 1)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel			
						Scan 0	Scan 1	Scan 2	Scan 3
0	0	0	0	0	0	ANI0	ANI1	ANI2	ANI3
0	0	0	0	0	1	ANI1	ANI2	ANI3	ANI4
0	0	0	0	1	0	ANI2	ANI3	ANI4	ANI5
0	0	0	0	1	1	ANI3	ANI4	ANI5	ANI6
0	0	0	1	0	0	ANI4	ANI5	ANI6	ANI7
0	0	0	1	0	1	ANI5	ANI6	ANI7	ANI8
0	0	0	1	1	0	ANI6	ANI7	ANI8	ANI9
0	0	0	1	1	1	ANI7	ANI8	ANI9	ANI10
0	0	1	0	0	0	ANI8	ANI9	ANI10	ANI11
0	0	1	0	0	1	ANI9	ANI10	ANI11	ANI12
0	0	1	0	1	0	ANI10	ANI11	ANI12	ANI13
Other than the above						Setting prohibited			

Caution 1. Be sure to clear bits 5 and 6 to 0.

Caution 2. Use port mode registers 0 to 2 (PM0 to PM2) to select input mode for the port that is set to analog input by the PMCx register.

Caution 3. Do not set the pin that is set by Port mode control registers 0 to 2 (PMC0 to PMC2) as digital I/O by the ADS register.

Caution 4. Rewrite the value of the ADISS bit while conversion operation is stopped (ADCS = 0, ADCE = 0).

Caution 5. If using AV_{REFP} as the + side reference voltage source of the A/D converter, do not select ANI0 as an A/D conversion channel.

Caution 6. If using AV_{REFM} as the – side reference voltage source of the A/D converter, do not select ANI1 as an A/D conversion channel.

Caution 7. If ADISS is set to 1, the internal reference voltage (1.45 V) cannot be used for the + side reference voltage source. Also, after setting the ADISS to 1, the result of the first conversion cannot be used. For details about the setting flow, see 14.7.4 Setup when temperature sensor output voltage/internal reference voltage is selected (example for software trigger mode and one-shot conversion mode).

Caution 8. Do not set the ADISS bit to 1 when shifting to STOP mode, or to HALT mode while the CPU is operating on the subsystem clock. Also, if the ADISS bit is set to 1, the temperature sensor operating current indicated in 34.3.2 Supply current characteristics (I_{TMPS}) will be added to the current consumption when shifting to HALT mode while the CPU is operating on the main system clock.

Caution 9. Ignore the conversion result if the corresponding ANI pin does not exist in the product used.

14.3.9 Conversion result comparison upper limit setting register (ADUL)

This register is used to specify the setting for checking the upper limit of the A/D conversion results.

The A/D conversion results and ADUL register value are compared, and interrupt signal (INTAD) generation is controlled in the range specified for the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in **Figure 14 - 10**).

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

Reset signal generation sets this register to FFH.

Figure 14 - 14 Format of Conversion result comparison upper limit setting register (ADUL)

Address: F0011H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
ADUL	ADUL7	ADUL6	ADUL5	ADUL4	ADUL3	ADUL2	ADUL1	ADUL0

Caution 1. When 12-bit resolution A/D conversion is selected, the higher 8 bits of the 12-bit A/D conversion result register (ADCR) are compared with the ADUL register.

Caution 2. Only rewrite the value of the ADUL register and ADLL register while conversion operation is stopped (which is indicated by the ADCE bit of A/D converter mode register 0 (ADM0) being 0).

Caution 3. Make sure that ADUL > ADLL when setting these registers.

14.3.10 Conversion result comparison lower limit setting register (ADLL)

This register is used to specify the setting for checking the lower limit of the A/D conversion results.

The A/D conversion results and ADLL register value are compared, and interrupt signal (INTAD) generation is controlled in the range specified for the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in **Figure 14 - 10**).

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

Reset signal generation clears this register to 00H.

Figure 14 - 15 Format of Conversion result comparison lower limit setting register (ADLL)

Address: F0012H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADLL	ADLL7	ADLL6	ADLL5	ADLL4	ADLL3	ADLL2	ADLL1	ADLL0

Caution 1. When 12-bit resolution A/D conversion is selected, the higher 8 bits of the 12-bit A/D conversion result register (ADCR) are compared with the ADLL register.

Caution 2. Only rewrite the value of the ADUL register and ADLL register while conversion operation is stopped (which is indicated by the ADCE bit of A/D converter mode register 0 (ADM0) being 0).

Caution 3. Make sure that ADUL > ADLL when setting these registers.

14.3.11 A/D test register (ADTES)

This register is used to select the + side reference voltage or - side reference voltage for the A/D converter, an analog input channel (ANIxx), the temperature sensor output voltage, or the internal reference voltage (1.45 V) as the target for A/D conversion. When using as the A/D test function, set as follows.

• For zero-scale measurement, select the − side reference voltage as the target for conversion.

• For full-scale measurement, select the + side reference voltage as the target for conversion.

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

Reset signal generation clears this register to 00H.

Figure 14 - 16 Format of A/D test register (ADTES)

Address: F0013H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	0	0	ADTES1	ADTES0

ADTES1	ADTES0	A/D conversion target
0	0	ANIxx/temperature sensor output voltage/internal reference voltage (1.45 V) (This is specified using the analog input channel specification register (ADS).)
1	0	- side reference voltage (setting at ADREFM bit of ADM2 register)
1	1	+ side reference voltage (setting at ADREFP1, ADREFP0 bits of the ADM2 register)
Other than the above		Setting prohibited

Caution For details on the A/D test function, refer to CHAPTER 27 SAFETY FUNCTIONS.

14.3.12 Registers controlling port function of analog input pins

Set up the registers for controlling the functions of the ports shared with the analog input pins of the A/D converter (port mode registers (PMxx) and port mode control registers (PMCxx)).
For details, see as follows.

- **4.3.1 Port mode registers (PMxx)**
- **4.3.6 Port mode control registers (PMCxx)**

When using the ANI0 to ANI13 and ANI16 to ANI18 pins for analog input of the A/D converter, set the port mode register (PMxx) bit and port mode control register (PMCxx) bit corresponding to each port to 1.

14.4 A/D Converter Conversion Operations

The A/D converter conversion operations are described below.

<1> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<2> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the sampled voltage is held until the A/D conversion operation has ended.

<3> Bit 11 of the successive approximation register (SAR) is set. 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 bit of the SAR register remains set to 1. If the analog input is smaller than (1/2) AVREF, the MSB bit 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 according to the preset value of bit 11, 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> Upon completion of the comparison of 12 bits, an effective digital result value remains in the SAR register, and the result value is transferred to the A/D conversion result register (ADCR, ADCRH) and then latched.

At the same time, the A/D conversion end interrupt request (INTAD) can also be generated [Note 1].

<8> Repeat steps <1> to <7>, until the ADCS bit is cleared to 0 [Note 2].

To stop the A/D converter, clear the ADCS bit to 0.

Note 1. If the A/D conversion result is outside the A/D conversion result range specified by the ADRCK bit and the ADUL and ADLL registers (see **Figure 14 - 10**), the A/D conversion result interrupt request signal is not generated and no A/D conversion results are stored in the ADCR and ADCRH registers.

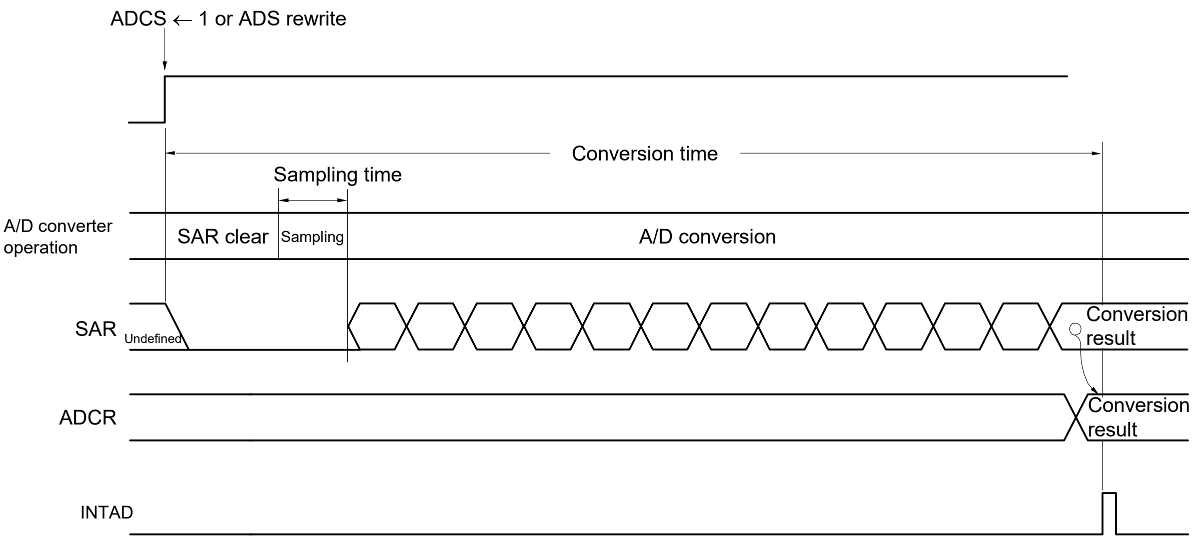
Note 2. While in the sequential conversion mode, the ADCS flag is not automatically cleared to 0. This flag is not automatically cleared to 0 while in the one-shot conversion mode of the hardware trigger no-wait mode, either. Instead, 1 is retained.

Remark 1. Two types of the A/D conversion result registers are available.

- ADCR register (16 bits): Store 12-bit A/D conversion value
- ADCRH register (8 bits): Store 8-bit A/D conversion value

Remark 2. AVREF: The + side reference voltage of the A/D converter. This can be selected from AVREFP, the internal reference voltage (1.45 V), and AVDD.

Figure 14 - 17 Conversion Operation of A/D Converter (Software Trigger 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 bit 7 (ADCS) of the A/D converter mode register 0 (ADM0) to 0.

Rewriting and overwriting to the analog input channel specification register (ADS) during A/D conversion interrupts the current conversion after which A/D conversion of the analog input specified by the ADS register proceeds. Data from the A/D conversion that was in progress are discarded.

Reset signal generation clears the A/D conversion result register (ADCR, ADCRH) to 0000H or 00H.

14.5 Input Voltage and Conversion Results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI13 and ANI16 to ANI18) and the theoretical A/D conversion result (stored in the 12-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$\mathrm{ADCR} = \mathrm{INT}\left(\frac{V_{AIN}}{AV_{REF}} \times 4096 + 0.5\right)$$

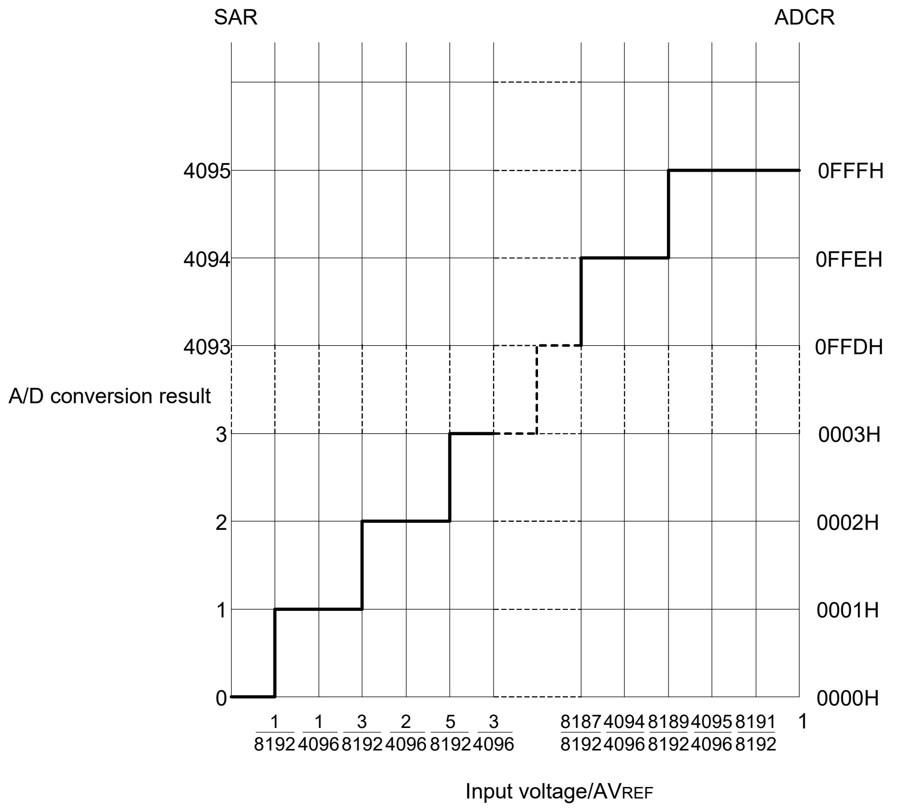
or

$$(\mathrm{ADCR} - 0.5) \times \frac{AV_{REF}}{4096} \le V_{AIN} < (\mathrm{ADCR} + 0.5) \times \frac{AV_{REF}}{4096}$$

where, INT( ): Function which returns integer part of value in parentheses
V_{AIN}: Analog input voltage
AV_{REF}: AV_{REF} pin voltage
ADCR: A/D conversion result register (ADCR) value

Figure 14 - 18 shows the Relationship Between Analog Input Voltage and A/D Conversion Result.

Figure 14 - 18 Relationship Between Analog Input Voltage and A/D Conversion Result



Remark AV_{REF}: The + side reference voltage of the A/D converter. This can be selected from AV_{REFP}, the internal reference voltage (1.45 V), and AV_{DD}.

14.6 A/D Converter Operation Modes

The operation of each A/D converter mode is described below. In addition, the procedure for specifying each mode is described in **14.7 A/D Converter Setup Flowchart**.

14.6.1 Software trigger mode (select mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time Note, the ADCS bit of the ADM0 register is set to 1 to perform the A/D conversion of the analog input specified by the analog input channel specification register (ADS).

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts.

<4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

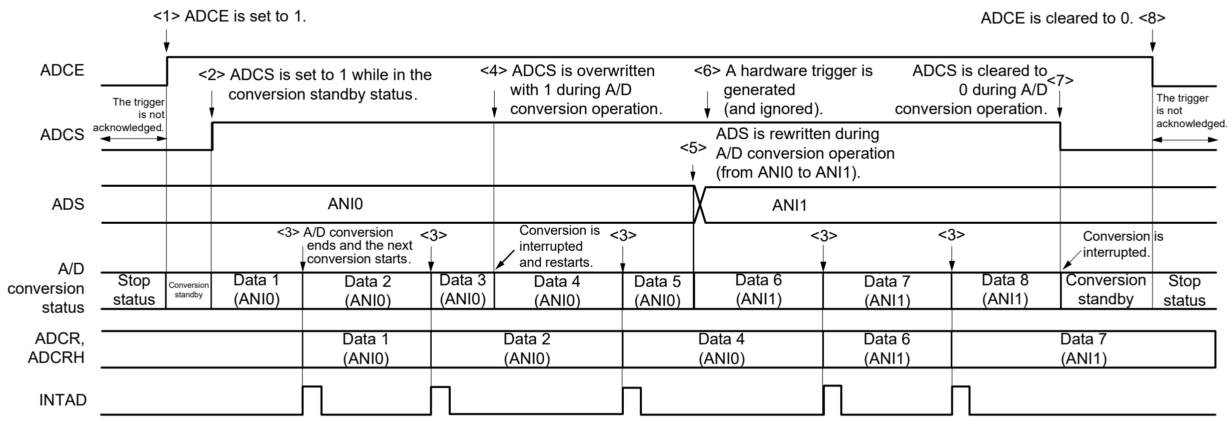
<6> Even if a hardware trigger is input during conversion operation, A/D conversion does not start.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Note If a high-accuracy channel is selected as the analog input channel: Stabilization wait time = 0.5 μs
If a standard channel is selected as the analog input channel: Stabilization wait time = 2 μs

Figure 14 - 19 Example of Software Trigger Mode (Select Mode, Sequential Conversion Mode) Operation Timing



14.6.2 Software trigger mode (select mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time Note, the ADCS bit of the ADM0 register is set to 1 to perform the A/D conversion of the analog input specified by the analog input channel specification register (ADS).

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the system enters the A/D conversion standby status.

<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

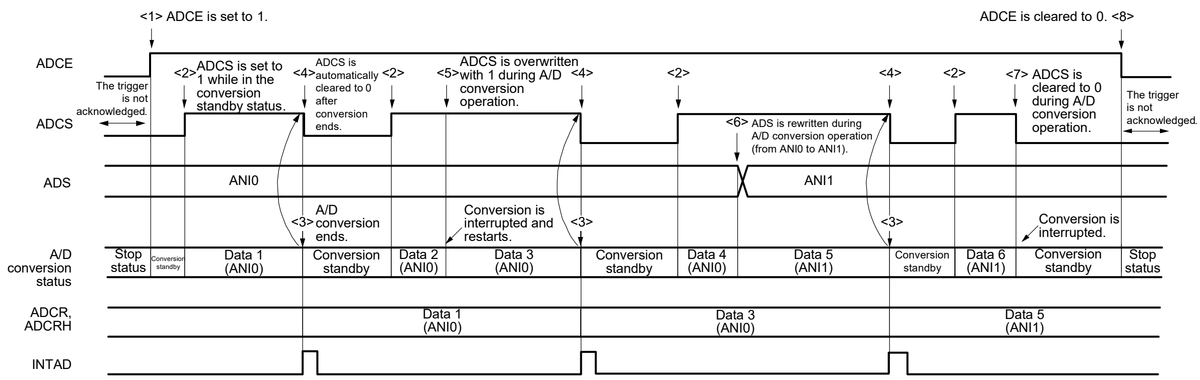
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby status.

Note If a high-accuracy channel is selected as the analog input channel: Stabilization wait time = 0.5 μs
If a standard channel is selected as the analog input channel: Stabilization wait time = 2 μs

Figure 14 - 20 Example of Software Trigger Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



14.6.3 Software trigger mode (scan mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time Note, the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts (until all four channels are finished).

<4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

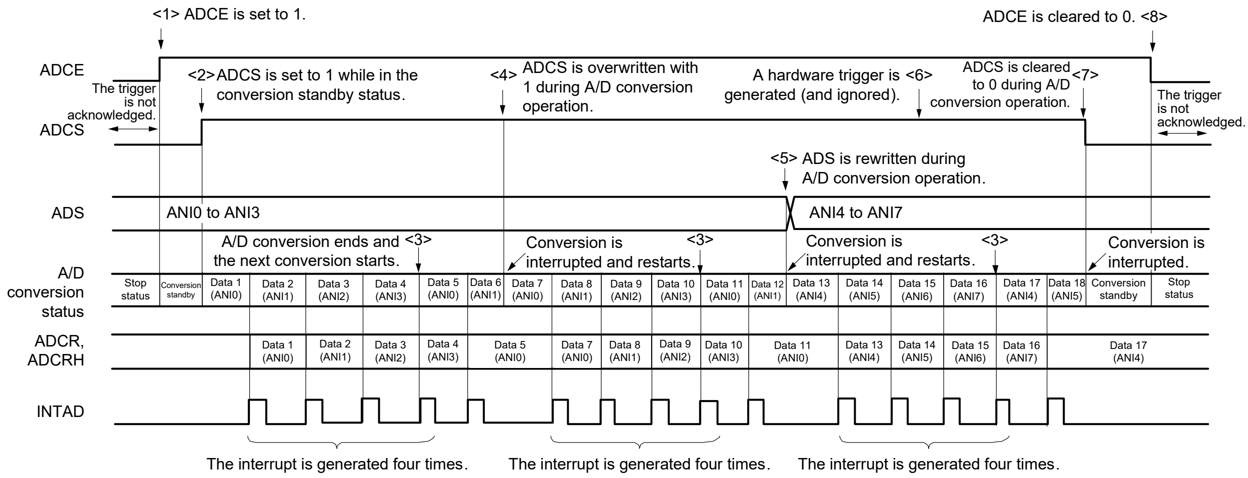
<6> Even if a hardware trigger is input during conversion operation, A/D conversion does not start.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Note If a high-accuracy channel is selected as the analog input channel: Stabilization wait time = 0.5 μs
If a standard channel is selected as the analog input channel: Stabilization wait time = 2 μs

Figure 14 - 21 Example of Software Trigger Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



14.6.4 Software trigger mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time Note, the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion of the four channels ends, the ADCS bit is automatically cleared to 0, and the system enters the A/D conversion standby status.

<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

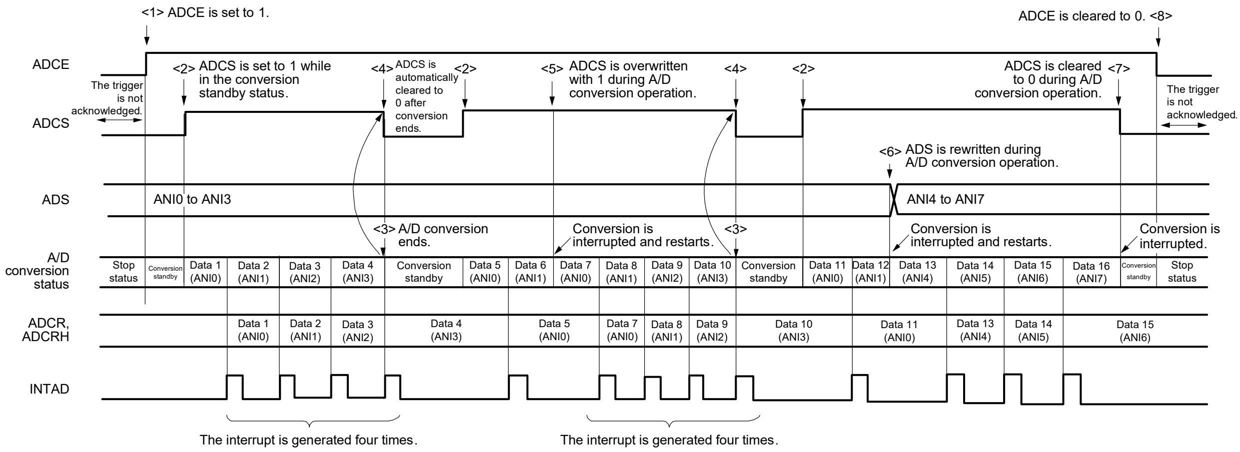
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby status.

Note If a high-accuracy channel is selected as the analog input channel: Stabilization wait time = 0.5 μs
If a standard channel is selected as the analog input channel: Stabilization wait time = 2 μs

Figure 14 - 22 Example of Software Trigger Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing



14.6.5 Hardware trigger no-wait mode (select mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time Note, the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).

<4> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts.

<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

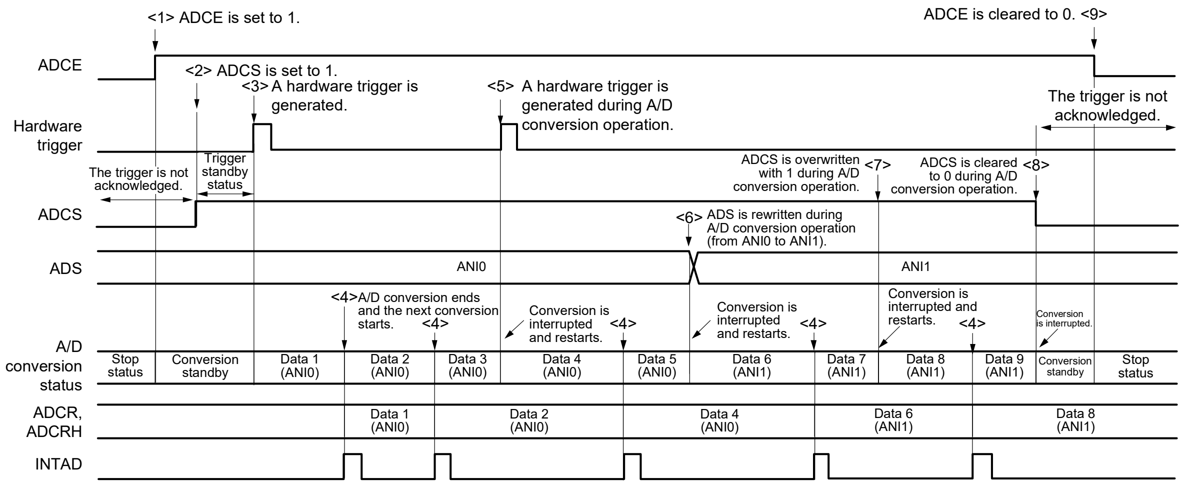
<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<9> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Note If a high-accuracy channel is selected as the analog input channel: Stabilization wait time = 0.5 μs
If a standard channel is selected as the analog input channel: Stabilization wait time = 2 μs

Figure 14 - 23 Example of Hardware Trigger No-Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



14.6.6 Hardware trigger no-wait mode (select mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time Note, the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).

<4> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<5> After A/D conversion ends, the ADCS bit remains set to 1, and the system enters the A/D conversion standby status.

<6> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

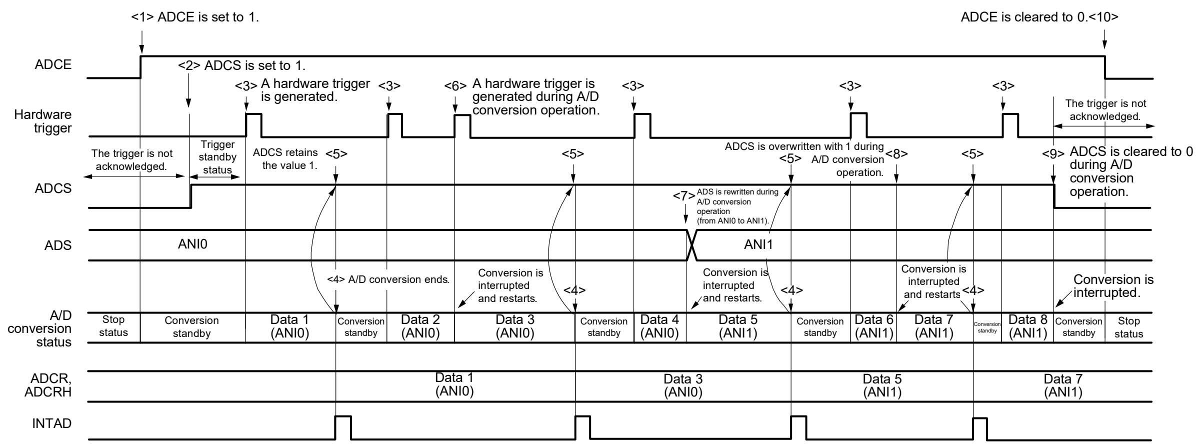
<8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<9> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<10> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Note If a high-accuracy channel is selected as the analog input channel: Stabilization wait time = 0.5 μs
If a standard channel is selected as the analog input channel: Stabilization wait time = 2 μs

Figure 14 - 24 Example of Hardware Trigger No-Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



14.6.7 Hardware trigger no-wait mode (scan mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time Note, the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<4> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts.

<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

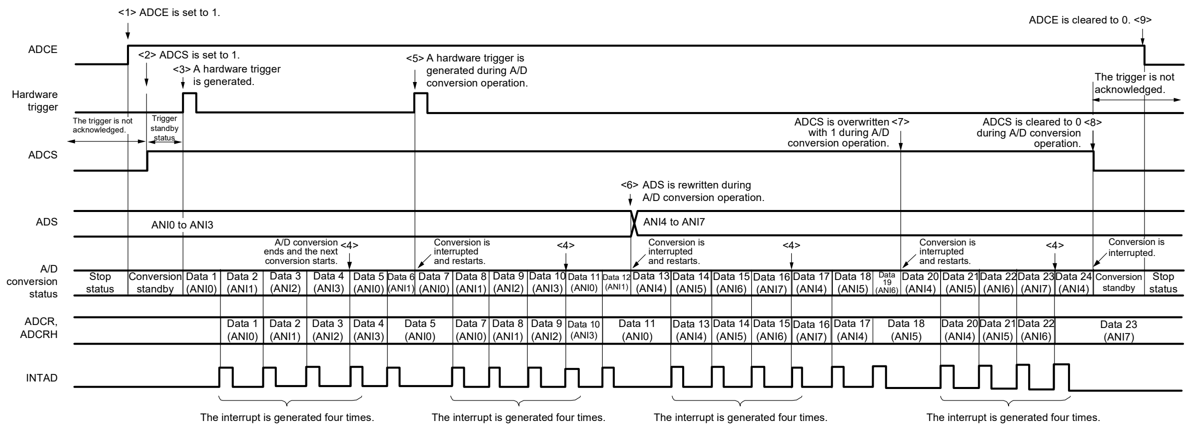
<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<9> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Note If a high-accuracy channel is selected as the analog input channel: Stabilization wait time = 0.5 μs
If a standard channel is selected as the analog input channel: Stabilization wait time = 2 μs

Figure 14 - 25 Example of Hardware Trigger No-Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



14.6.8 Hardware trigger no-wait mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time Note, the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<4> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<5> After A/D conversion of the four channels ends, the ADCS bit remains set to 1, and the system enters the A/D conversion standby status.

<6> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

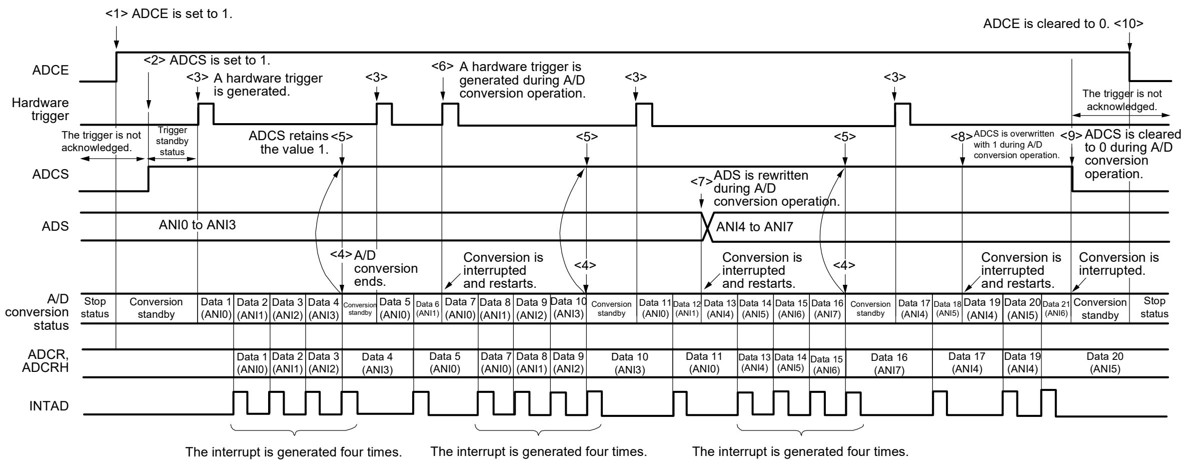
<8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<9> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<10> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Note If a high-accuracy channel is selected as the analog input channel: Stabilization wait time = 0.5 μs
If a standard channel is selected as the analog input channel: Stabilization wait time = 2 μs

Figure 14 - 26 Example of Hardware Trigger No-Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing



14.6.9 Hardware trigger wait mode (select mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the hardware trigger standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input.

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts. (At this time, no hardware trigger is necessary.)

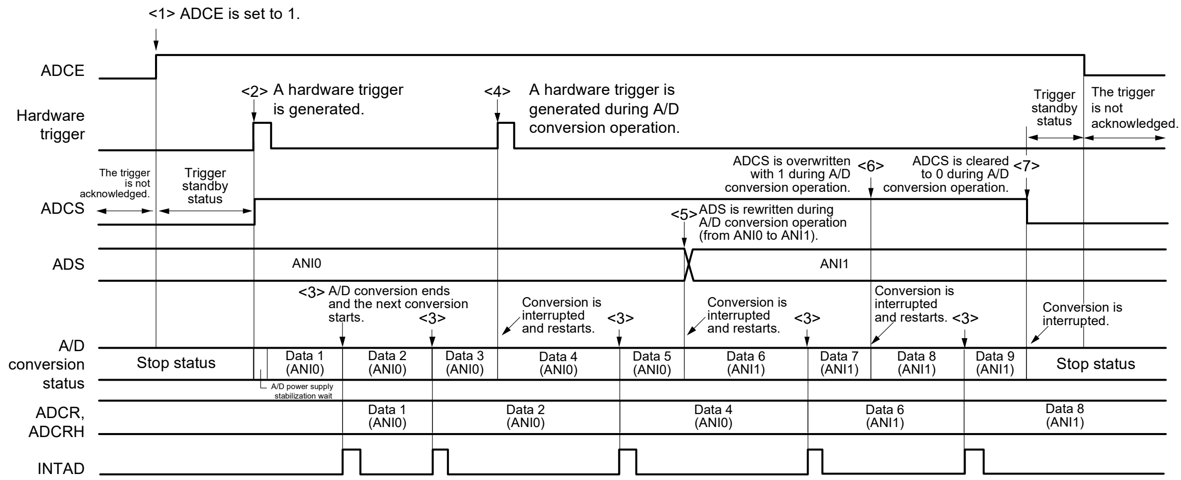
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<6> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 14 - 27 Example of Hardware Trigger Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



14.6.10 Hardware trigger wait mode (select mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the hardware trigger standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input.

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the stop status.

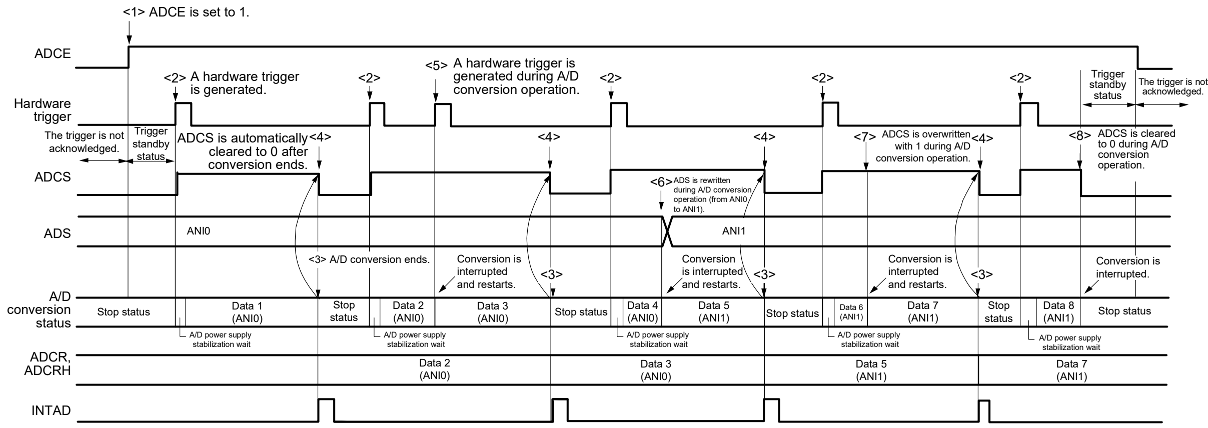
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is initialized.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 14 - 28 Example of Hardware Trigger Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



14.6.11 Hardware trigger wait mode (scan mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input. A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts.

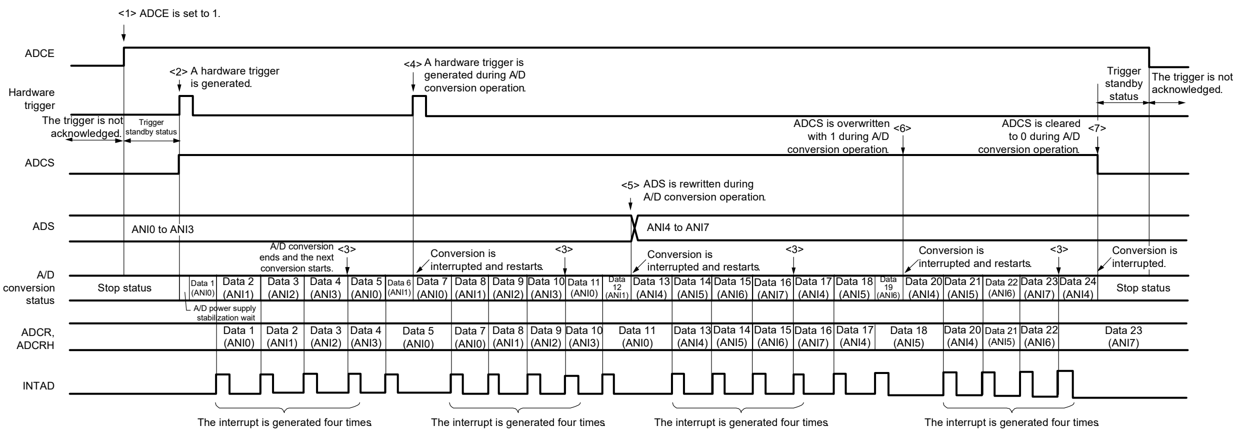
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<6> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 14 - 29 Example of Hardware Trigger Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



14.6.12 Hardware trigger wait mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input. A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the stop status.

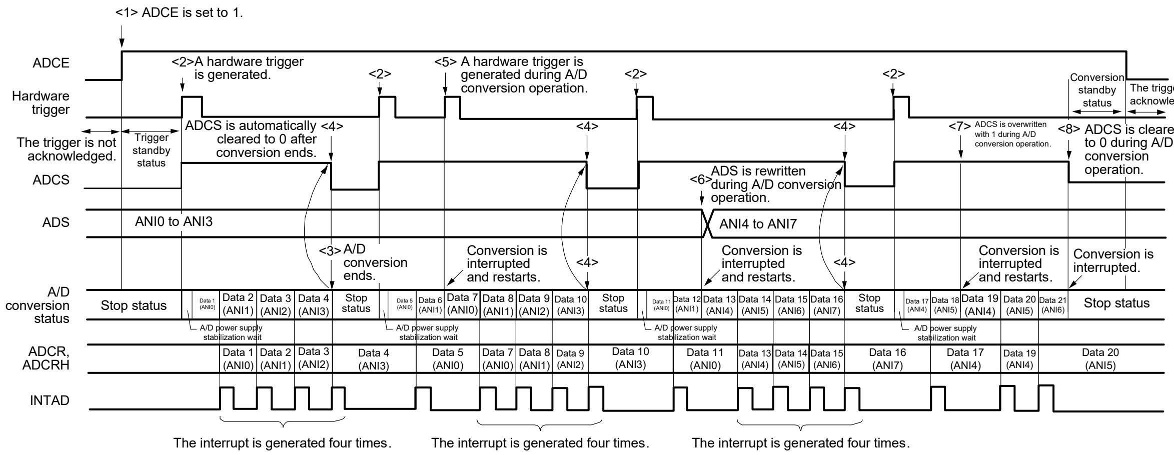
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts on the first channel. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 14 - 30 Example of Hardware Trigger Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing



14.7 A/D Converter Setup Flowchart

The A/D converter setup flowchart in each operation mode is described below.

14.7.1 Setting up software trigger mode

Figure 14 - 31 Setting up Software Trigger Mode

Step	Description
Start of setup	
PER0 register setting	The ADCEN bit of the PER0 register is set (1), and supplying the clock starts.
PMC register setting	The ports are set to analog input.
PM register setting	The ports are set to the input mode.
• ADM0 register setting • ADM1 register setting • ADM2 register setting • ADUL/ADLL register setting • ADS register setting (The order of the settings is irrelevant.)	• ADM0 register FR2 to FR0, LV1, and LV0 bits: These are used to specify the A/D conversion time. ADMD bit: Select mode/scan mode • ADM1 register ADTMD1 and ADTMD0 bits: These are used to specify the software trigger mode. ADSCM bit: Sequential conversion mode/one-shot conversion mode • ADM2 register ADREFP1, ADREFP0, and ADREFM bits: These are used to select the reference voltage source. ADRCK bit: This is used to select the range for the A/D conversion result comparison value generated by the interrupt signal from AREA1, AREA3, and AREA2. ADTYP bit: 8-bit/12-bit resolution • ADUL/ADLL register These are used to specify the upper limit and lower limit A/D conversion result comparison values. • ADS register ADS4 to ADS0 bits: These are used to select the analog input channels.
Reference voltage stabilization wait time count A	The reference voltage stabilization wait time count A is required when the value of the ADREFP1 and ADREFP0 bits is changed. If change the ADREFP1 and ADREFP0 = 1, 0: A = 10 μs If change the ADREFP1 and ADREFP0 = 0, 0 or 0, 1: A = 1 μs
ADCE bit setting	The ADCE bit of the ADM0 register is set (1), and the system enters the A/D conversion standby status.
Reference voltage stabilization wait time count B	If a high-accuracy channel is selected as the analog input channel: B = 0.5 μs If a standard channel is selected as the analog input channel: B = 2 μs
ADCS bit setting	After counting up to the reference voltage stabilization wait time count B ends, the ADCS bit of the ADM0 register is set (1), and A/D conversion starts.
Start of A/D conversion	
	The A/D conversion operations are performed.
End of A/D conversion	The A/D conversion end interrupt (INTAD) is generated. Note
Storage of conversion results in the ADCR and ADCRH registers	The conversion results are stored in the ADCR and ADCRH registers

Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no A/D conversion end interrupt request signal (INTAD) being generated. In this case, the results are not stored in the ADCR, ADCRH registers.

14.7.2 Setting up hardware trigger no-wait mode

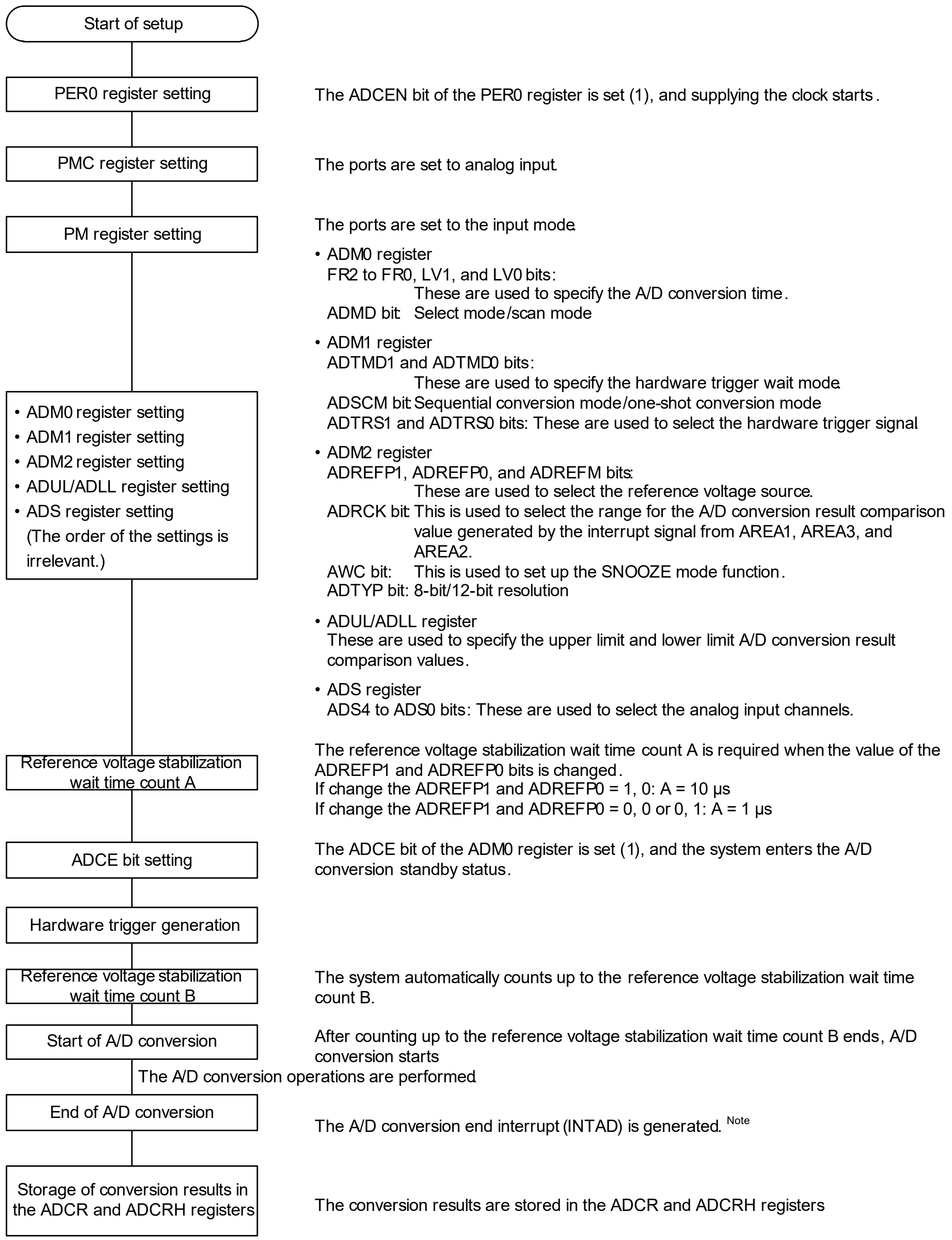
Figure 14 - 32 Setting up Hardware Trigger No-Wait Mode

Start of setup	
PER0 register setting	The ADCEN bit of the PER0 register is set (1), and supplying the clock starts.
PMC register setting	The ports are set to analog input.
PM register setting	The ports are set to the input mode.
• ADM0 register setting • ADM1 register setting • ADM2 register setting • ADUL/ADLL register setting • ADS register setting (The order of the settings is irrelevant.)	• ADM0 register FR2 to FR0, LV1, and LV0 bits: These are used to specify the A/D conversion time. ADMD bit: Select mode/scan mode • ADM1 register ADTMD1 and ADTMD0 bits: These are used to specify the hardware trigger no-wait mode. ADSCM bit: Sequential conversion mode/one-shot conversion mode • ADM2 register ADREFP1, ADREFP0, and ADREFM bits: These are used to select the reference voltage source. ADRCK bit: This is used to select the range for the A/D conversion result comparison value generated by the interrupt signal from AREA1, AREA3, and AREA2. ADTYP bit: 8-bit/12-bit resolution • ADUL/ADLL register These are used to specify the upper limit and lower limit A/D conversion result comparison values. • ADS register ADS4 to ADS0 bits: These are used to select the analog input channels.
Reference voltage stabilization wait time count A	The reference voltage stabilization wait time count A is required when the value of the ADREFP1 and ADREFP0 bits is changed. If change the ADREFP1 and ADREFP0 = 1, 0: A = 10 μs If change the ADREFP1 and ADREFP0 = 0, 0 or 0, 1: A = 1 μs
ADCE bit setting	The ADCE bit of the ADM0 register is set (1), and the system enters the A/D conversion standby status.
Reference voltage stabilization wait time count B	If a high-accuracy channel is selected as the analog input channel: B = 0.5 μs If a standard channel is selected as the analog input channel: B = 2 μs
ADCS bit setting	After counting up to the reference voltage stabilization wait time count B ends, the ADCS bit of the ADM0 register is set (1), and the system enters the hardware trigger standby status.
Hardware trigger standby status	
Start of A/D conversion by generating a hardware trigger	
The A/D conversion operations are performed.	
End of A/D conversion	The A/D conversion end interrupt (INTAD) is generated. Note
Storage of conversion results in the ADCR and ADCRH registers	The conversion results are stored in the ADCR and ADCRH registers

Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no A/D conversion end interrupt request signal (INTAD) being generated. In this case, the results are not stored in the ADCR, ADCRH registers.

14.7.3 Setting up hardware trigger wait mode

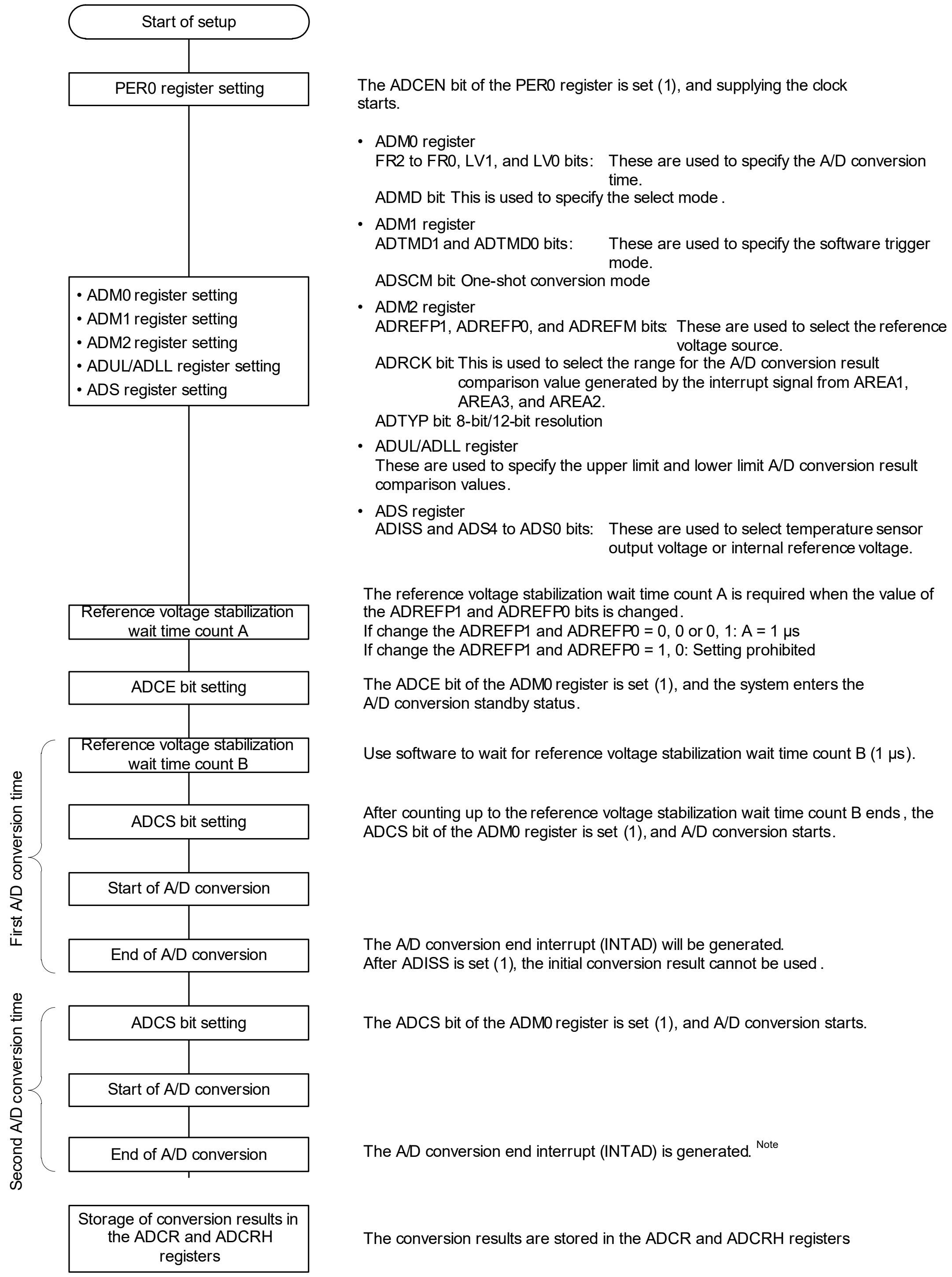
Figure 14 - 33 Setting up Hardware Trigger Wait Mode



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no A/D conversion end interrupt request signal (INTAD) being generated. In this case, the results are not stored in the ADCR, ADCRH registers.

14.7.4 Setup when temperature sensor output voltage/internal reference voltage is selected (example for software trigger mode and one-shot conversion mode)

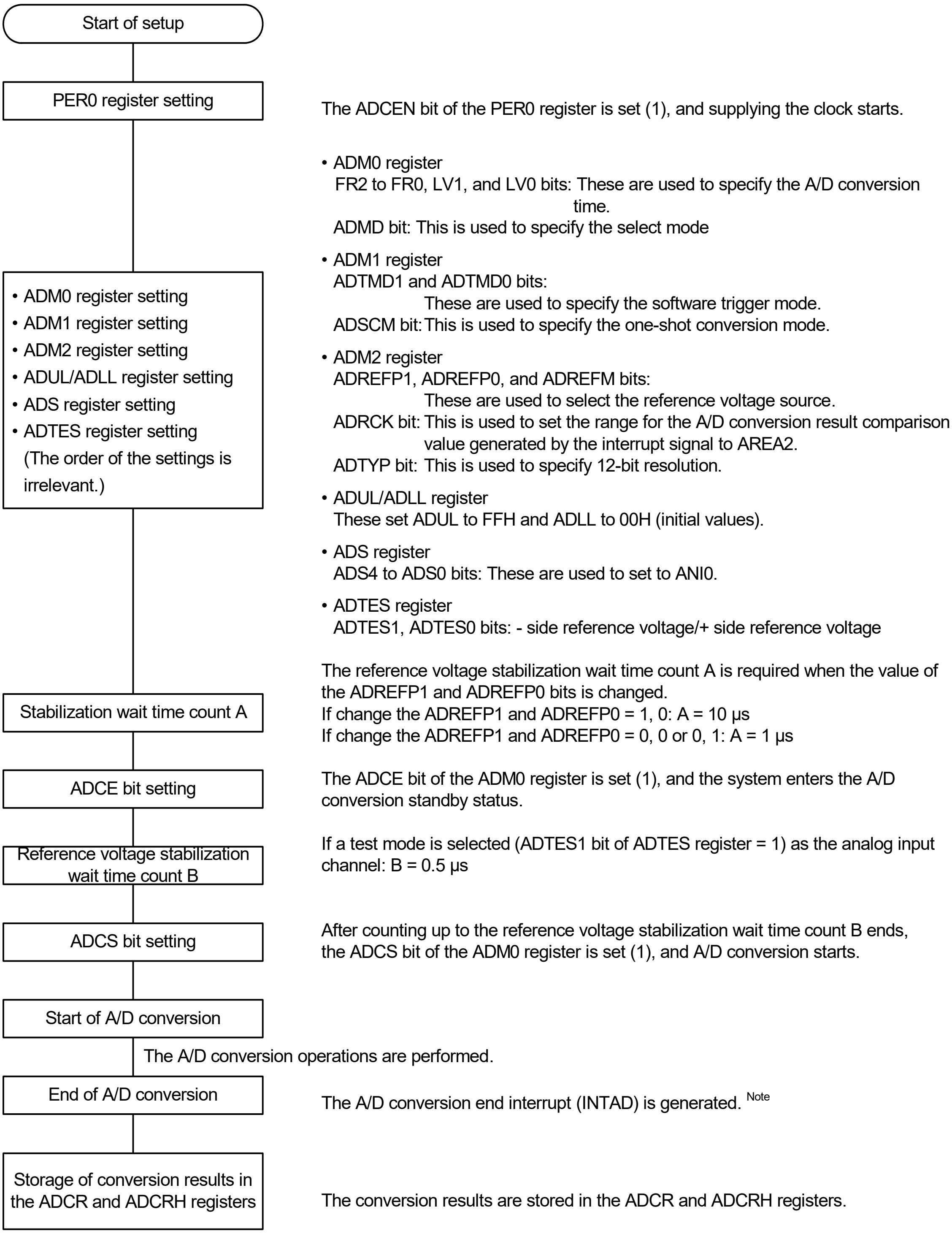
Figure 14 - 34 Setup when Temperature Sensor Output Voltage/Internal Reference Voltage is Selected



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no A/D conversion end interrupt request signal (INTAD) being generated. In this case, the results are not stored in the ADCR, ADCRH registers.

14.7.5 Setting up test mode

Figure 14 - 35 Setting up Test Trigger Mode



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no A/D conversion end interrupt request signal (INTAD) being generated. In this case, the results are not stored in the ADCR, ADCRH registers.

Caution For the procedure for testing the A/D converter, see 27.3.8 A/D test function.

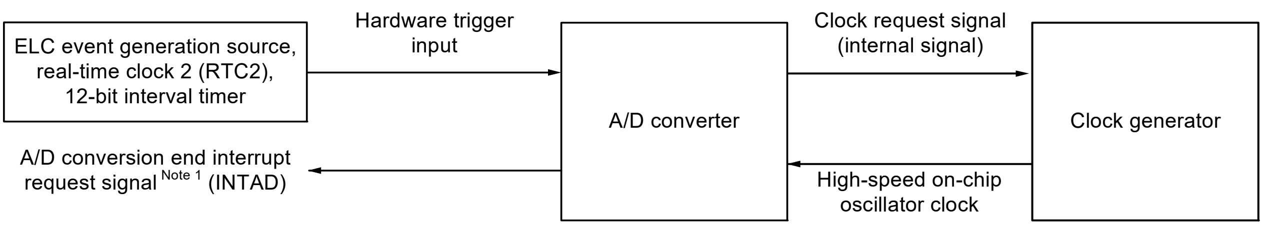
14.8 SNOOZE Mode Function

In the SNOOZE mode, A/D conversion is triggered by inputting a hardware trigger in the STOP mode. Normally, A/D conversion is stopped while in the STOP mode, but, by using the SNOOZE mode, A/D conversion can be performed without operating the CPU by inputting a hardware trigger. This is effective for reducing the operation current.
If the A/D conversion result range is specified using the ADUL and ADLL registers, A/D conversion results can be judged at a certain interval of time in SNOOZE mode. Using this function enables power supply voltage monitoring and input key judgment based on A/D inputs.

In the SNOOZE mode, only the following two conversion modes can be used:
• Hardware trigger wait mode (select mode, one-shot conversion mode)
• Hardware trigger wait mode (scan mode, one-shot conversion mode)

Caution The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock (f_{IH}) or middle-speed on-chip oscillator clock (f_{IM}) is selected for f_{CLK}.

Figure 14 - 36 Block Diagram When Using SNOOZE Mode Function



When using the SNOOZE mode function, the initial setting of each register is specified before switching to the STOP mode. (For details about these settings, see **14.7.3 Setting up hardware trigger wait mode** [Note 2].) Just before switching to the STOP mode, set bit 2 (AWC) of A/D converter mode register 2 (ADM2) to 1. After the initial settings are specified, bit 0 (ADCE) of A/D converter mode register 0 (ADM0) is set to 1.
If a hardware trigger is input after switching to the STOP mode, the high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock is supplied to the A/D converter. After supplying the high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock, the system 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 ends differs depending on whether an interrupt signal is generated [Note 1].

Note 1. Depending on the setting of the A/D conversion result comparison function (ADRCK bit, ADUL/ADLL register), there is a possibility of no interrupt signal being generated.
Note 2. Be sure to set the ADM1 register to E1H, E2H or E3H.

Remark 1. The hardware trigger is the event selected by the ELC, or INTRTC or INTIT.
INTRTC can be used as an ELC event generation source or directly used as a trigger.
Remark 2. Specify the hardware trigger by using A/D Converter Mode Register 1 (ADM1).

(1) If an interrupt is generated after A/D conversion ends

If the A/D conversion result value is inside the range of values specified by the A/D conversion result comparison function (which is set up by using the ADRCK bit and ADUL/ADLL register), the A/D conversion end interrupt request signal (INTAD) is generated.

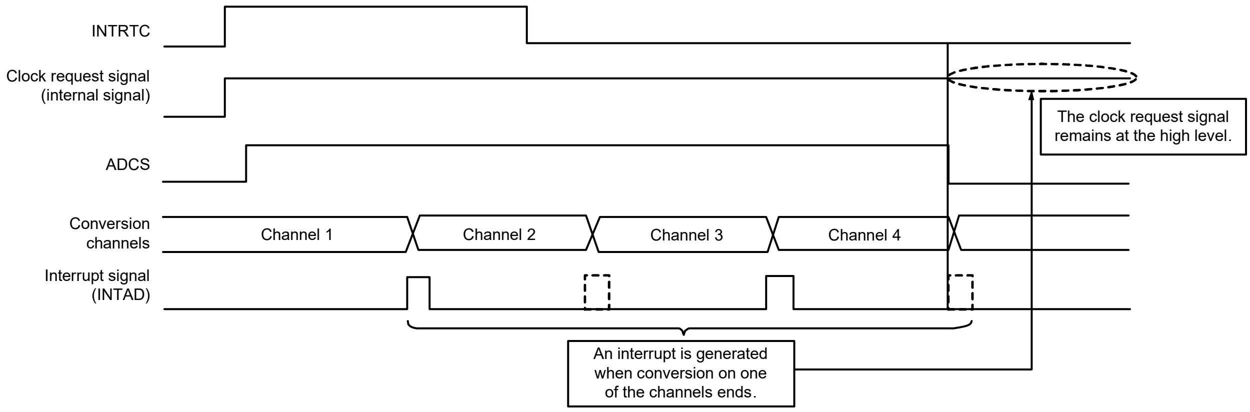
• While in the select mode

When A/D conversion ends and an A/D conversion end interrupt request signal (INTAD) is generated, the A/D converter returns to normal operation mode from SNOOZE mode. At this time, be sure to clear bit 2 (AWC = 0: SNOOZE mode release) of A/D converter mode register 2 (ADM2). If the AWC bit is left set to 1, A/D conversion will not start normally in the subsequent SNOOZE or normal operation mode.

• While in the scan mode

If even one A/D conversion end interrupt request signal (INTAD) is generated during A/D conversion of the four channels, the clock request signal remains at the high level, and the A/D converter switches from the SNOOZE mode to the normal operation mode. At this time, be sure to clear bit 2 (AWC = 0: SNOOZE mode release) of 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 14 - 37 Operation Example When Interrupt Is Generated After A/D Conversion Ends (While in Scan Mode)



(2) If no interrupt is generated after A/D conversion ends

If the A/D conversion result value is outside the range of values specified by the A/D conversion result comparison function (which is set up by using the ADRCK bit and ADUL/ADLL register), the A/D conversion end interrupt request signal (INTAD) is not generated.

- While in the select mode

  If the A/D conversion end interrupt request signal (INTAD) is not generated after A/D conversion ends, the clock request signal (an internal signal) is automatically set to the low level, and supplying the high-speed on-chip oscillator clock stops. If a hardware trigger is input later, A/D conversion work is again performed in the SNOOZE mode.

- While in the scan mode

  If the A/D conversion end interrupt request signal (INTAD) is not generated even once during A/D conversion of the four channels, the clock request signal (an internal signal) is automatically set to the low level after A/D conversion of the four channels ends, and supplying the high-speed on-chip oscillator clock stops. If a hardware trigger is input later, A/D conversion work is again performed in the SNOOZE mode.

Figure 14 - 38 Operation Example When No Interrupt Is Generated After A/D Conversion Ends (While in Scan Mode)

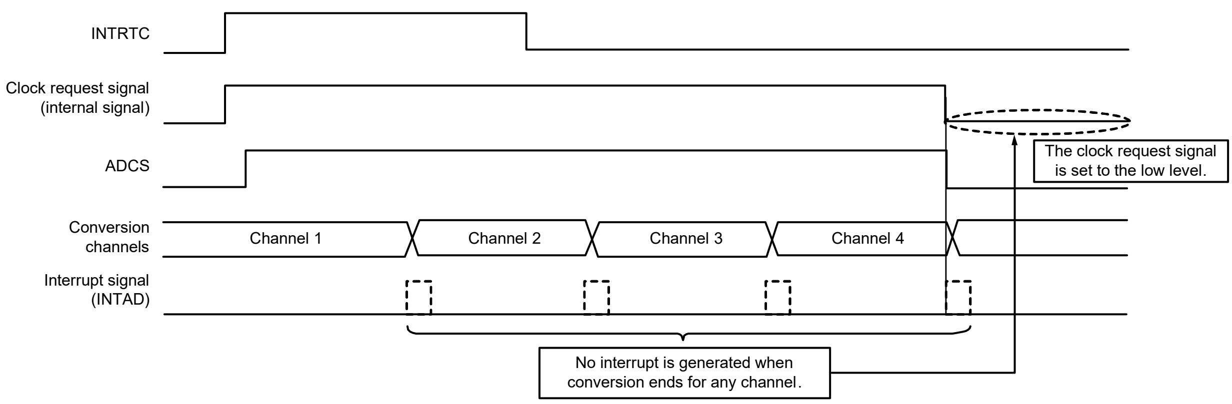
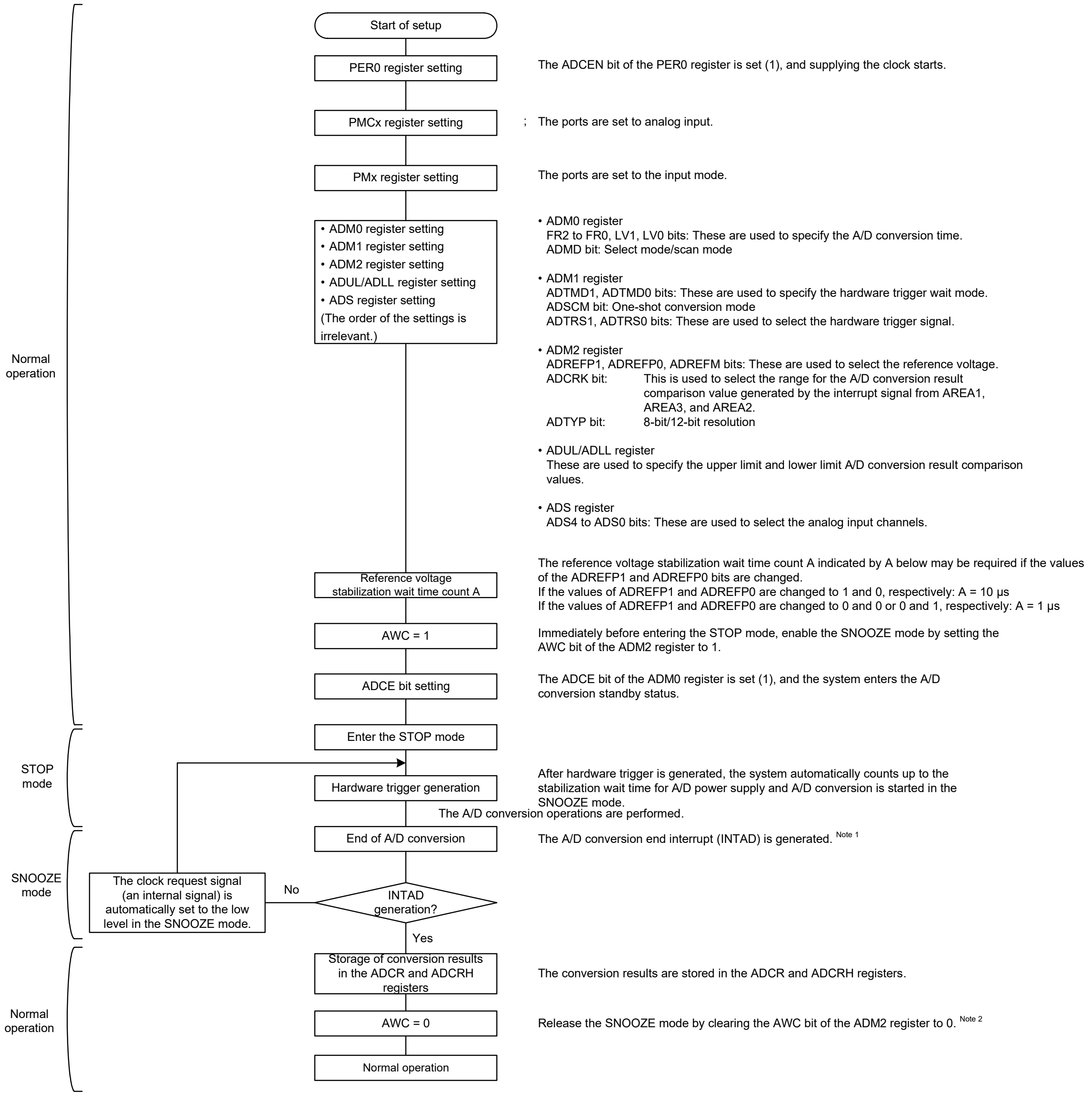


Figure 14 - 39 Flowchart for Setting up SNOOZE Mode



Note 1. If the A/D conversion end interrupt request signal (INTAD) is not generated by setting ADRCK bit and ADUL/ADLL register, the result is not stored in the ADCR and ADCRH registers.
The system 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 spite of the subsequent SNOOZE or normal operation mode. Be sure to clear the AWC bit to 0.

14.9 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

1LSB is as follows when the resolution is 10 bits.

1 LSB = $1/2^{12}$ = 1/4096
= 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, and differential linearity errors that are combinations of these express the overall error.
Note that the quantization error is not included in the overall error in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, a ±1/2LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2LSB is converted to the same digital code, so a quantization error cannot be avoided.
Note that the quantization error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

Figure 14 - 40 Overall Error

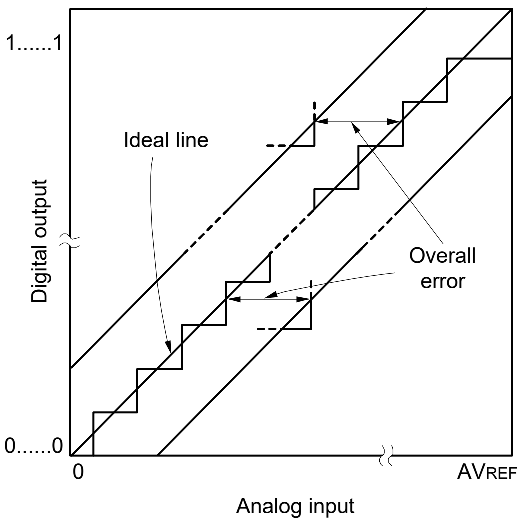
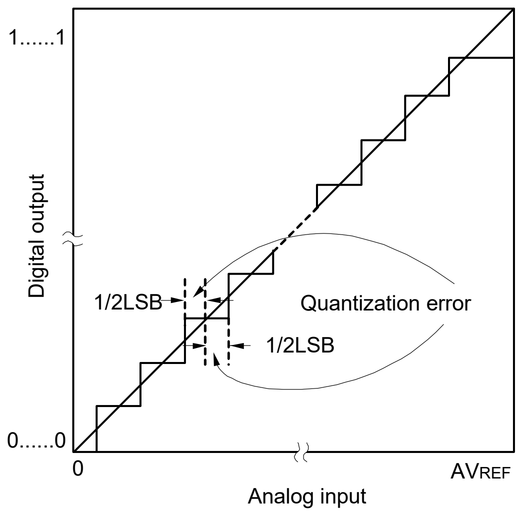


Figure 14 - 41 Quantization Error



(4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2LSB) when the digital output changes from 0......000 to 0......001.

If the actual measurement value is greater than the theoretical value, it shows the difference between the actual measurement value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 0……001 to 0……010.

(5) Full-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (Full-scale – 3/2LSB) when the digital output changes from 1......110 to 1......111.

(6) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

(7) Differential linearity error

While the ideal width of code output is 1LSB, this indicates the difference between the actual measurement value and the ideal value.

Figure 14 - 42 Zero-Scale Error

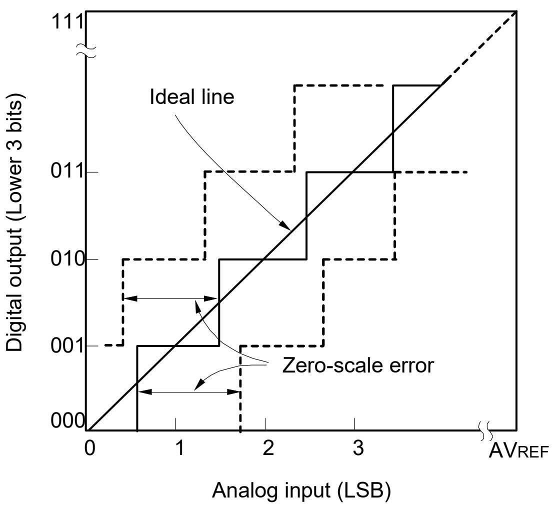


Figure 14 - 43 Full-Scale Error

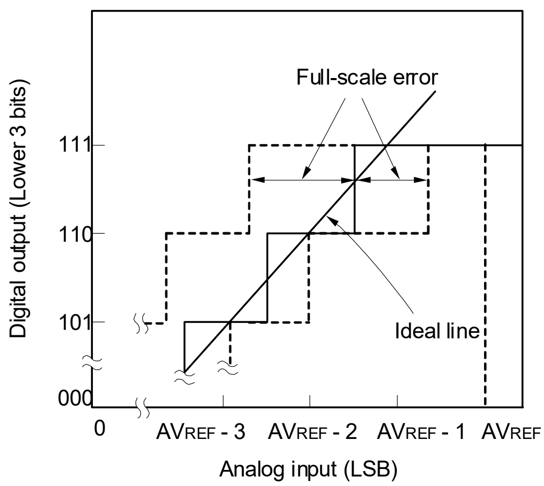


Figure 14 - 44 Integral Linearity Error

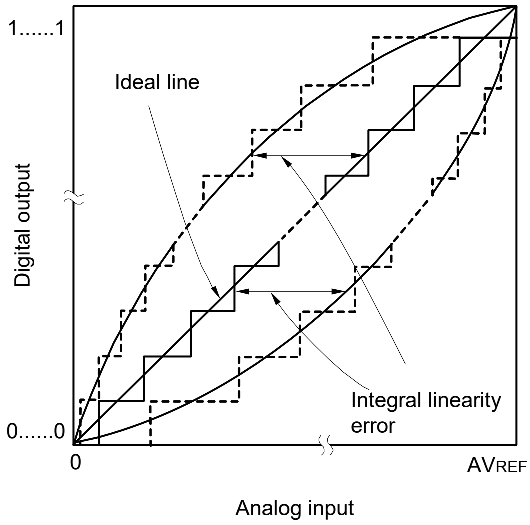
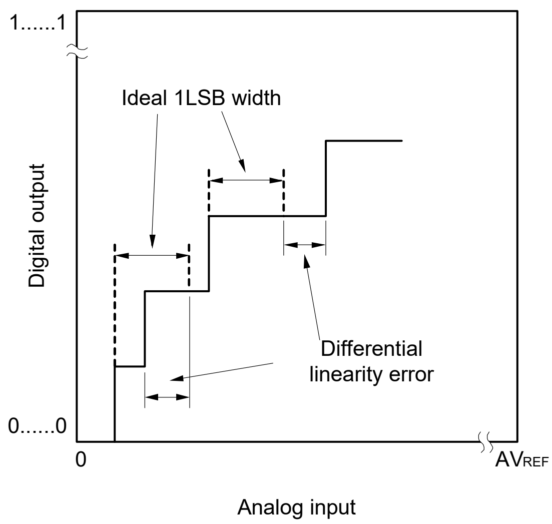


Figure 14 - 45 Differential Linearity Error



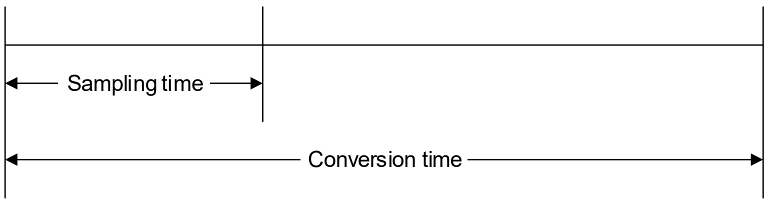
(8) Conversion time

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.



14.10 Cautions for A/D Converter

(1) Operating current in STOP mode

Shift to STOP mode after stopping the A/D converter (by setting bit 7 (ADCS) of A/D converter mode register 0 (ADM0) to 0). The operating current can be reduced by setting bit 0 (ADCE) of the ADM0 register to 0 at the same time.

To restart from the standby status, clear bit 0 (ADIF) of interrupt request flag register 1H (IF1H) to 0 and start operation.

(2) Input range of ANI0 to ANI13 and ANI16 to ANI18 pins

Observe the rated range of the ANI0 to ANI13 and ANI16 to ANI18 pins input voltage. If a voltage exceeding AV_{DD} and AV_{REFP} or a voltage below AV_{SS} and AV_{REFM} (even in the range of absolute maximum ratings) is input to an analog input channel, the converted value of that channel becomes undefined. In addition, the converted values of the other channels may also be affected.

When internal reference voltage (1.45 V) is selected reference voltage source for the + side of the A/D converter, do not input voltages exceeding the internal reference voltage (1.45 V) to a pin selected by the ADS register. However, the input of voltages exceeding the internal reference voltage (1.45 V) to pins not selected by the ADS register does not create a problem.

(3) Conflicting operations

<1> Conflict between the A/D conversion result register (ADCR, ADCRH) write and the ADCR or ADCRH register read by instruction upon the end of conversion
The ADCR or ADCRH register read has priority. After the read operation, the new conversion result is written to the ADCR or ADCRH registers.

<2> Conflict between the ADCR or ADCRH register write, the A/D converter mode register 0 (ADM0) write, and the analog input channel specification register (ADS) write upon the end of conversion
The ADM0 or ADS register write has priority. The ADCR or ADCRH register write is not performed, nor is the conversion end interrupt signal (INTAD) generated.

(4) Noise countermeasures

To maintain the 12-bit resolution, attention must be paid to noise input to the AV_{REFP}, AV_{DD} ANI0 to ANI13 and ANI16 to ANI18 pins.

<1> Connect a capacitor with a low equivalent resistance and a good frequency response (capacitance of about 0.01 μF) via the shortest possible run of relatively thick wiring to the power supply.

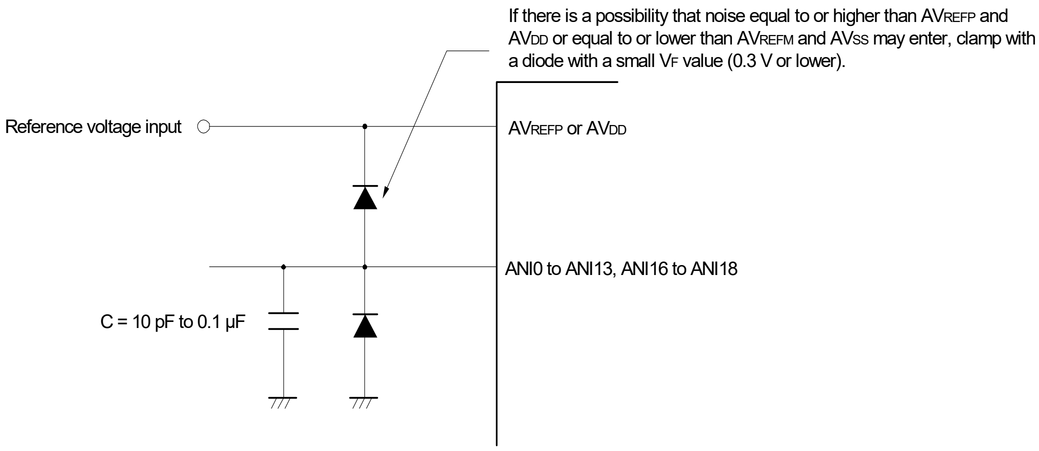
<2> The higher the output impedance of the analog input source, the greater the influence. To reduce the noise, connecting external capacitor as shown in Figure 14 - 46 is recommended.

<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.

<5> Separate digital and analog signals so that they do not cross or approach each other.

Figure 14 - 46 Analog Input Pin Connection



(5) Analog input (ANIn) pins

<1> ANI0 to ANI13 pins (high-accuracy channel) are also used as P10 to P17 and P20 to P25 pins.
When A/D conversion is performed with any of the high-accuracy channel (ANI0 to ANI6) pins selected, do not change the output value P150 to P156 while conversion is in progress; otherwise the conversion accuracy may be degraded.

<2> If a pin adjacent to the pin whose value is being A/D converted is used as a digital I/O port pin, the A/D conversion value might differ from the expected value due to coupling noise. To prevent coupling noise, make sure that pulses whose voltage suddenly change, such as digital pulses, are not input or output to a pin adjacent to the pin whose value is being A/D converted.

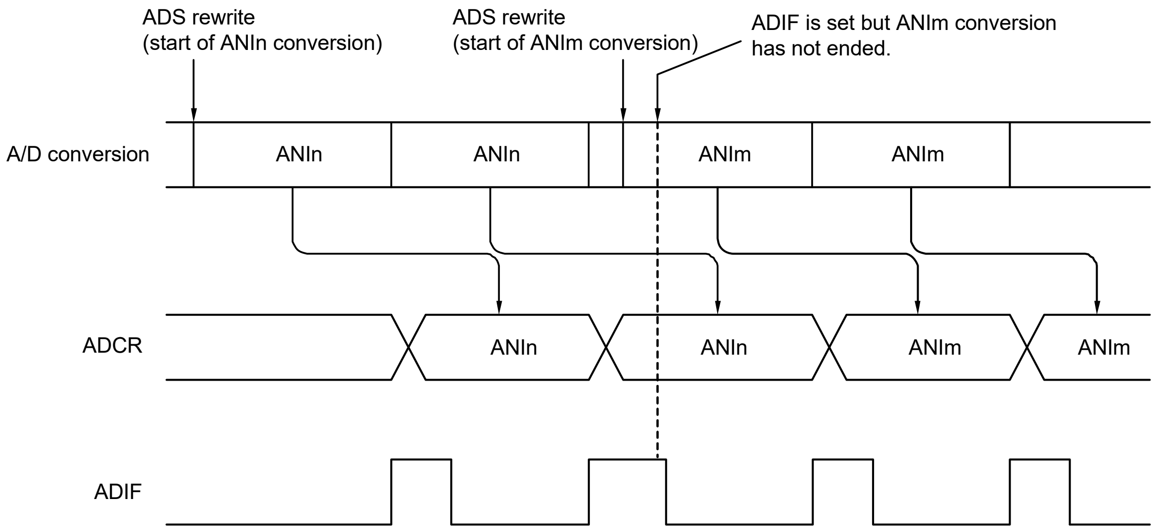
(6) Input impedance of analog input (ANIn) pins

This A/D converter charges a sampling capacitor for sampling during sampling time.
Therefore, only a leakage current flows when sampling is not in progress, and a current that charges the capacitor flows during sampling. Consequently, the input impedance fluctuates depending on whether sampling is in progress, and on the other states.
However, in order to perform sampling accurately, the output impedance of the analog input source should be 1 kΩ or lower. If it is not possible to keep the output impedance below this level, it is recommended to either extend the sampling time or connect a capacitor of about 0.1 μF to the ANI0 to ANI13 and ANI16 to ANI18 pins. (See **Figure 14 - 46** for details.)
Also, if the ADCS bit is set to 0 or a reconversion is started during A/D conversion, the sampling capacitor will be insufficiently charged. This means that charging will start with an undefined conversion voltage from the next conversion in the case of setting the ADCS bit to 0, or from the current conversion in the case of starting a reconversion. To ensure that the capacitor is fully charged, therefore, either reduce the output impedance of the analog input source or specify a sufficiently long sampling time, irrespective of the analog signal voltage variation.

(7) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the analog input channel specification register (ADS) is changed.
Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and ADIF flag for the pre-change analog input may be set just before the ADS register rewrite. Caution is therefore required since, at this time, when ADIF flag is read immediately after the ADS register rewrite, ADIF flag is set despite the fact A/D conversion for the post-change analog input has not ended.
When A/D conversion is stopped and then resumed, clear ADIF flag before the A/D conversion operation is resumed.

Figure 14 - 47 Timing of A/D Conversion End Interrupt Request Generation



(8) Conversion results just after A/D conversion start

While in the software trigger mode or hardware trigger no-wait mode, the first A/D conversion value immediately after A/D conversion starts may not fall within the rating range if the ADCS bit is set to 1 within the stabilization wait time after the ADCE bit was set to 1. Take measures such as polling the A/D conversion end interrupt request (INTAD) and removing the first conversion result.

[Stabilization wait status]

If a high-accuracy channel is selected as the analog input channel: 0.5 μs

If a test mode setting (ADTES1 bit of ADTES register = 1) is selected: 0.5 μs

If a standard channel is selected as the analog input channel: 2 μs

If a temperature sensor output/internal reference voltage output are selected as the analog input channel:

(ADISS bit of ADS register = 1): 2 μs

(9) A/D conversion result register (ADCR, ADCRH) read operation

When a write operation is performed to A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), and port mode control register (PMCx), the contents of the ADCR and ADCRH registers may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, or PMCx register. Using a timing other than the above may cause an incorrect conversion result to be read.

(10) Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 14 - 48 Internal Equivalent Circuit of ANIn Pin

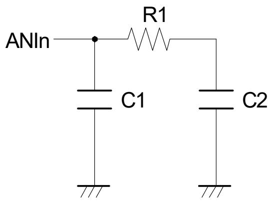


Table 14 - 7 Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AVDD, AVREFP	ANIn pin	R1[kΩ]	C1[pF]	C2[pF]
2.4 V ≤ AVDD ≤ 3.6 V	ANI0 to ANI13	7.4	8	6.3
	ANI16 to ANI18	12.3	8	7.4
1.8 V ≤ AVDD ≤ 3.6 V	ANI0 to ANI13	11	8	6.3
	ANI16 to ANI18	41	8	7.4
1.6 V ≤ AVDD ≤ 3.6 V	ANI0 to ANI13	510	8	6.3
	ANI16 to ANI18	650	8	7.4

Remark The resistance and capacitance values shown in Table 14 - 7 are not guaranteed values.

(11) Starting the A/D converter

Start the A/D converter after the AVREFP and AVDD voltages stabilize.

(12) Always set the AVDD and VDD to the same potential when the A/D voltage comparator operates.

CHAPTER 15 COMPARATOR

15.1 Functions of Comparator

The comparator has the following functions.

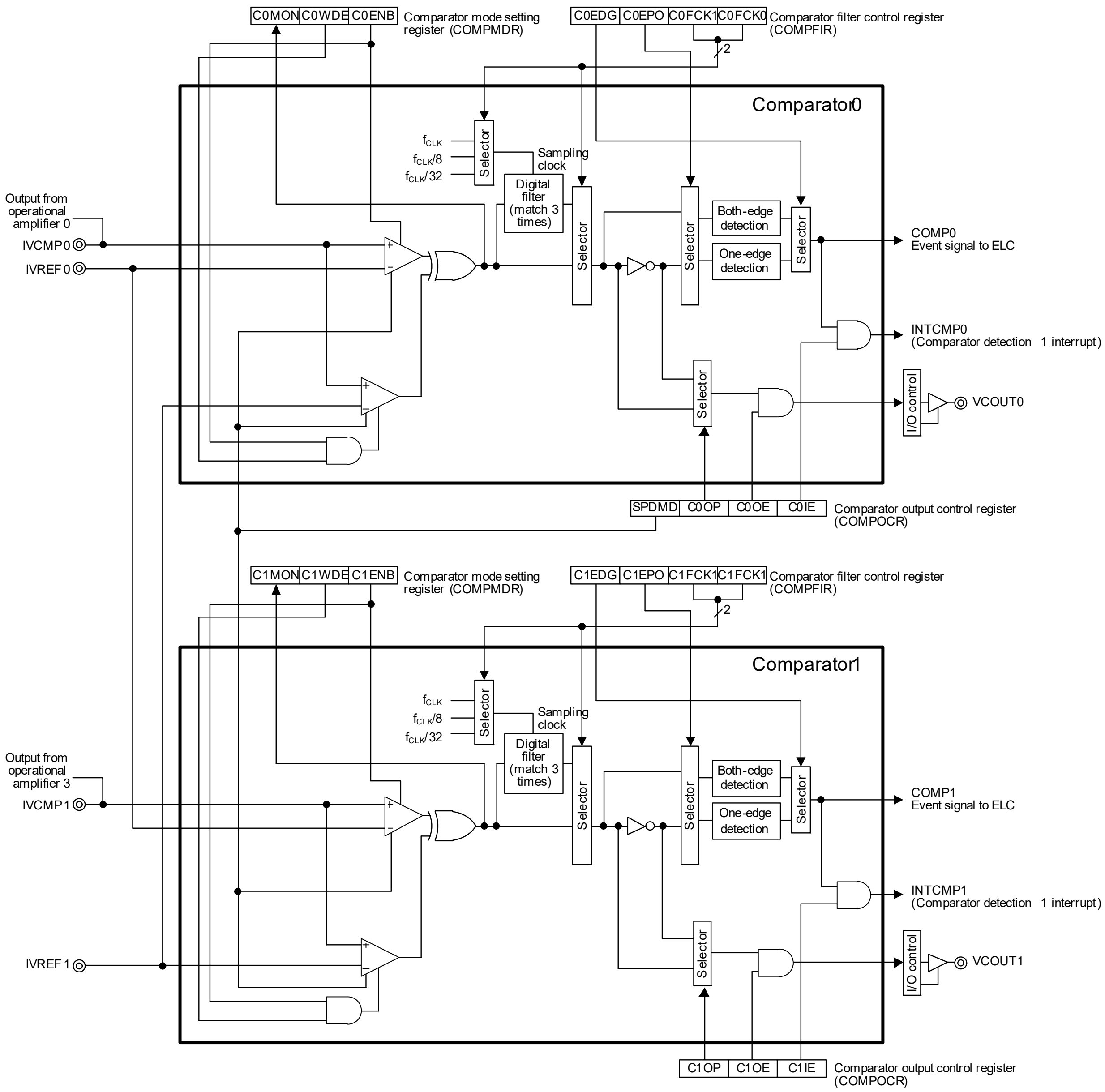
- Comparator high-speed mode, comparator low-speed mode, comparator high-speed window mode, or comparator low-speed window mode can be selected.
- The external reference voltage input is used as the reference voltage.
- The canceling width of the noise canceling digital filter can be selected.
- An interrupt signal can be generated by detecting an active edge of the comparator output.
- An event link controller (ELC) event signal can be output by detecting an active edge of the comparator output.

Caution The positive side input signal of the comparator 0 is connected to the output from operational amplifier 0. Do not input a signal from the positive side input pin (P14/ANI4/IVCMP0/AMP0O) of comparator 0 while operational amplifier 0 is in use.
Similarly, do not input a signal to the positive side input pin (P20/ANI13/IVCMP1/AMP3O) while operational amplifier 3 is in use.

15.2 Configuration of Comparator

Figure 15 - 1 shows the Comparator Block Diagram.

Figure 15 - 1 Comparator Block Diagram



15.3 Registers Controlling Comparator

Table 15 - 1 lists the Registers Controlling Comparator.

Table 15 - 1 Registers Controlling Comparator

Register Name	Symbol
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 control register 1, 2, 3	PMC1, PMC2, PMC3
Port mode registers 1, 2, 3, 5	PM1, PM2, PM3, PM5
Port registers 1, 2, 3, 5	P1, P2, P3, P5

15.3.1 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable use of each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the comparator is used, be sure to set bit 5 (CMPEN) of this register to 1.

The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 15 - 2 Format of Peripheral enable register 1 (PER1)

Address: F00FAH After reset: 00H R/W

Symbol	7	6	<5>	4	<3>	2	1	0
PER1	0	0	CMPEN	0	DTCEN	0	0	0

CMPEN	Control of comparator input clock
0	Stops input clock supply. • SFR used by the comparator cannot be written.
1	Supplies input clock. • SFR used by the comparator can be read/written.

Caution 1. When setting the comparator, be sure to set the CMPEN bit to 1 first.
If CMPEN = 0, writing to a control register of the comparator is ignored (except for port mode control registers 1, 2, 3 (PMC1, PMC2, PMC3), port mode registers 1, 2, 3, 5 (PM1, PM2, PM3, PM5), and port registers 1, 2, 3, 5 (P1, P2, P3, P5)).

Caution 2. Be sure to clear bits 0 to 2, 4, 6, and 7 to 0.

15.3.2 Peripheral reset control register 1 (PRR1)

This register is used for individual reset control of each peripheral hardware.
This MCU controls reset and reset release of each peripheral hardware supported by the PRR1 register.
To reset the comparator, be sure to set bit 5 (CMPRES) to 1.
The PRR1 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 15 - 3 Format of Peripheral reset control register 1 (PRR1)

Address: F00FBH After reset: 00H R/W

Symbol	7	6	<5>	4	3	2	1	0
PRR1	0	0	CMPRES	0	0	0	0	0

CMPRES	Reset control of comparator
0	Comparator reset release
1	Comparator reset state

15.3.3 Comparator mode setting register (COMPMDR)

Figure 15 - 4 Format of Comparator mode setting register (COMPMDR)

Address: F0340H After reset: 00H R/W

Symbol	7	6	5	<4>	3	2	1	<0>
COMPMDR	C1MON	0	C1WDE	C1ENB	C0MON	0	C0WDE	C0ENB

C1MON	Comparator 1 monitor flag Notes 1, 2
0	In standard mode: VCMP1 < IVREF0 In window mode: IVCMP1 < IVREF0 or IVCMP1< IVREF1
1	In standard mode: IVCMP1 > IVREF0 In window mode: IVREF0 < IVCMP1 < IVREF1

C1WDE	Comparator 1 window mode selection Note 3
0	Comparator 1 standard mode
1	Comparator 1 window mode

C1ENB	Comparator 1 operation enable
0	Comparator 1 operation disabled
1	Comparator 1 operation enabled

C0MON	Comparator 0 monitor flag Notes 1, 2
0	In standard mode: VCMP0 < IVREF0 In window mode: IVCMP0 < IVREF0 or IVCMP0 > IVREF1
1	In standard mode: VCMP0 > IVREF0 In window mode: IVREF0 < IVCMP0 < IVREF1

C0WDE	Comparator 0 window mode selection Note 3
0	Comparator 0 standard mode
1	Comparator 0 window mode

C0ENB	Comparator 0 operation enable
0	Comparator 0 operation disabled
1	Comparator 0 operation enabled

Note 1. The initial 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 2. The value written to this bit is ignored.

Note 3. When using window mode, make sure that IVREF1 > IVREF0 for use as the reference voltage.

15.3.4 Comparator filter control register (COMPFIR)

Figure 15 - 5 Format of Comparator filter control register (COMPFIR)

Address: F0341H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
COMPFIR	C1EDG	C1EPO	C1FCK1	C1FCK0	C0EDG	C0EPO	C0FCK1	C0FCK0

C1EDG	Comparator 1 edge detection selection Note 1
0	Interrupt request by comparator 1 one-edge detection
1	Interrupt request by comparator 1 both-edge detection

C1EPO	Comparator 1 edge polarity switching Note 1
0	Interrupt request at comparator 1 rising edge
1	Interrupt request at comparator 1 falling edge

C1FCK1	C1FCK0	Comparator 1 filter selection Note 1
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 2
0	Interrupt request by comparator 0 one-edge detection
1	Interrupt request by comparator 0 both-edge detection

C0EPO	Comparator 0 edge polarity switching Note 2
0	Interrupt request at comparator 0 rising edge
1	Interrupt request at comparator 0 falling edge

C0FCK1	C0FCK0	Comparator 0 filter selection Note 2
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 $f_{CLK}/32$

Note 1. If bits C1FCK1 to C1FCK0, C1EPO, and C1EDG are changed, a comparator 1 interrupt and an event signal to the ELC may be generated. Change these bits only after setting the ELSELR19 register for the ELC to 0 (not linked to comparator 1 output). In addition, clear bit 5 (CMPIF1) in interrupt request flag register 1H (IF1H) to 0.
If bits C1FCK1 to C1FCK0 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 to the ELC.

Note 2. If bits C0FCK1 to C0FCK0, C0EPO, and C0EDG are changed, a comparator 0 interrupt and an event signal to the ELC may be generated. Change these bits only after setting the ELSELR18 register for the ELC to 0 (not linked to comparator 0 output). In addition, clear bit 4 (CMPIF0) in interrupt request flag register 1H (IF1H) to 0.
If bits C0FCK1 to C0FCK0 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 to the ELC.

15.3.5 Comparator output control register (COMPOCR)

Figure 15 - 6 Format of Comparator output control register (COMPOCR)

Address: F0342H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
COMPOCR	SPDMD	C1OP	C1OE	C1IE	0	C0OP	C0OE	C0IE

SPDMD	Comparator speed selection [Note 1]
0	Comparator low-speed mode
1	Comparator high-speed mode

C1OP	VCOUT1 output polarity selection
0	Comparator 1 output is output to VCOUT1
1	Inverted comparator 1 output is output to 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 2]
0	Comparator 1 interrupt request disabled
1	Comparator 1 interrupt request enabled

C0OP	VCOUT0 output polarity selection
0	Comparator 0 output is output to VCOUT0
1	Inverted comparator 0 output is output to 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 3]
0	Comparator 0 interrupt request disabled
1	Comparator 0 interrupt request enabled

Note 1. When rewriting the SPDMD bit, be sure to set the CiENB bit (i = 0 or 1) in the COMPMDR register to 0 in advance.

Note 2. If C1IE is changed from 0 (interrupt request disabled) to 1 (interrupt request enabled), since bit 5 (CMPIF1) in interrupt request flag register 1H (IF1H) may set to 1 (interrupt requested), clear bit 5 (CMPIF1) in interrupt request flag register 1H (IF1H) to 0 before using an interrupt.

Note 3. If C0IE is changed from 0 (interrupt request disabled) to 1 (interrupt request enabled), since bit 4 (CMPIF0) in interrupt request flag register 1H (IF1H) may set to 1 (interrupt requested), clear bit 4 (CMPIF0) in interrupt request flag register 1H (IF1H) to 0 before using an interrupt.

15.3.6 Registers controlling port functions of analog input pins

When using the IVCMP0, IVCMP1, IVREF0, and IVREF1 pins for analog input of the comparator, set the port mode register (PMxx) bit and port mode control register (PMCxx) bit to 1.

When using the VCOUT0 and VCOUT1 functions, set the registers (port mode register (PMxx) and port register (Pxx) that control the port functions shared with the target channels. For details, see **4.3.1 Port mode registers (PMxx)** and **4.3.2 Port registers (Pxx)**.

15.4 Operation

Comparator 0 and comparator 1 operate independently. Their setting methods and operations are the same. Table 15 - 2 lists the Procedure for Setting Comparator Associated Registers.

Table 15 - 2 Procedure for Setting Comparator Associated Registers

Step	Register	Bit	Setting Value	
1	PRR1	CMPRES	0 (reset release of comparator i)	
2	PER1	CMPEN	1 (input clock supply)	
3	PMCxx	PMC14, PMC20, PMC30, PMC31	Select the function of pins IVCMPi and IVREFi. • Set the PMC14, PMC20, PMC30, and PMC31 bits to 1 (analog input). • Set the PM14, PM20, PM30, and PM31 bits to 1 (input mode).	
	PMxx	PM14, PM20, PM30, PM31		
4	COMPOCR	SPDMD	Select the comparator response speed (0: Low-speed mode/1: High-speed mode). Note 1	
5	COMPMDR	CiWDE	0 (standard mode)	1 (window mode)
6	COMPMDR	CiENB	1 (operation enabled)	
7	Wait for comparator stabilization time t_{CMP}.			
8	COMPFIR	CiFCK1 to CiFCK0	Select whether the digital filter is used or not and the sampling clock.	
		CiEOP, CiEDG	Select the edge detection condition for an interrupt request (rising edge/falling edge/both edges).	
9	COMPOCR	CiOP, CiOE	Set the VCOUTi output (select the polarity and set output enabled or disabled).	
		CiIE	Set the interrupt request output enabled or disabled.	
10	Select the port logic output pin: VCOUTi Refer to **15.4.4 Comparator i Output (i = 0 or 1)**.			
11	PR1H	CMPPR0i, CMPPR1i	When using an interrupt: Select the interrupt priority level.	
12	MK1H	CMPMKi	When using an interrupt: Select the interrupt masking.	
13	IF1H	CMPIFi	When using an interrupt: 0 (no interrupt requested: initialization) Note 2	

Note 1. Comparator 0 and comparator 1 cannot be set independently.

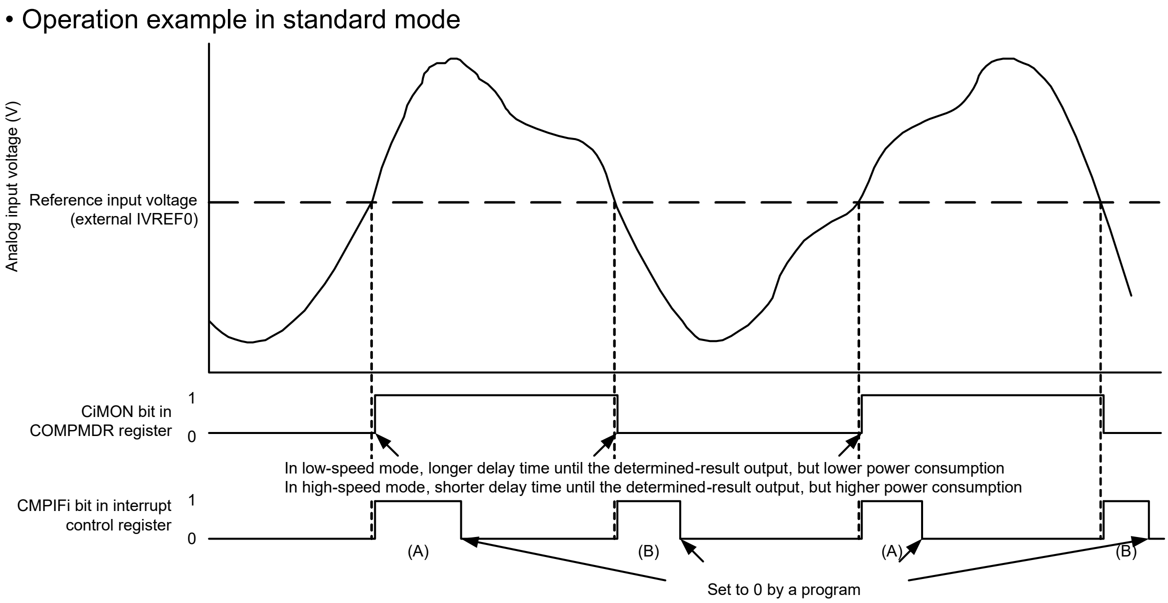
Note 2. After the setting of the comparator, an unnecessary interrupt may occur until operation becomes stable, so initialize the interrupt flag.

Caution **To initialize all circuits of converter i, set the CMPRES bit in the PRR1 register to 1.**

Remark i = 0, 1

Figures 15 - 7 shows an operation example of comparator i (i = 0, 1) in standard mode. 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 IVREF0 pin is selected for use as the reference voltage.

Figure 15 - 7 Comparator i (i = 0 or 1) Operation Example in Standard Mode

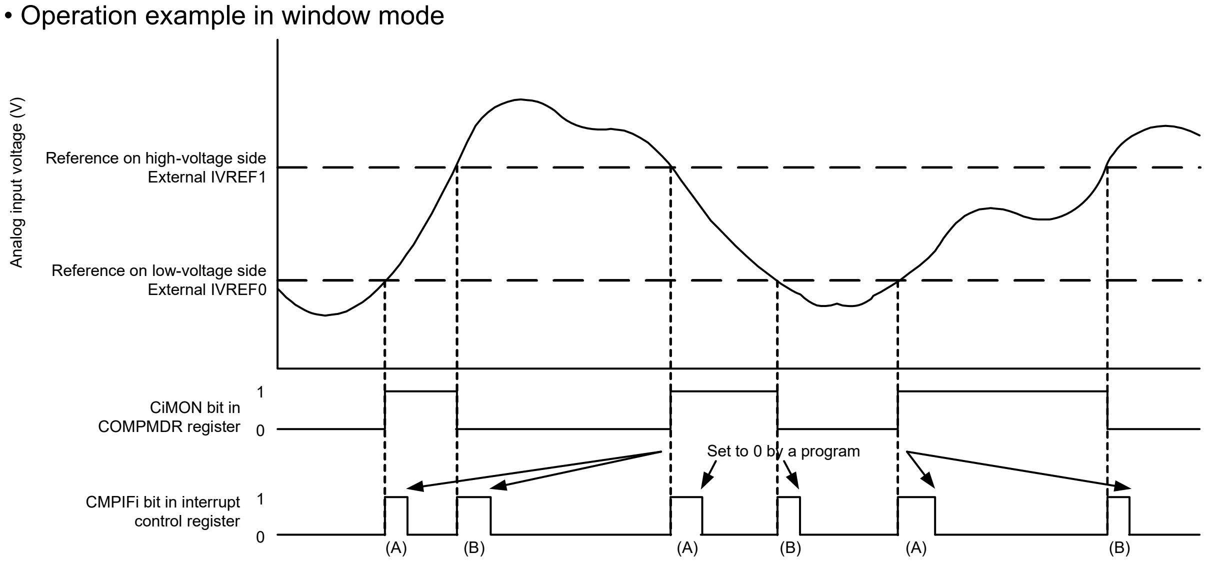


Caution **The above diagram applies when CiFCK1 to CiFCK0 in the COMPFIR register = 00B (no filter) and CiEDG = 1 (both edges). When CiEDG = 0 and CiEPO = 0 (rising edge), CMPIFi changes as shown by (A) only. When CiEDG = 0 and CiEPO = 1 (falling edge), CMPIFi changes as shown by (B) only.**

Figures 15 - 8 shows an operation example of comparator i (i = 0, 1) in window mode. During this mode, in both low-speed and high-speed mode, the CiMON bit in the COMPMDR register is set to 1 when the analog input voltage meets the following condition, and the CiMON bit is set to 0 when the analog input voltage does not meet the following condition. The low-side reference voltage is the input voltage to the IVREF0 pin and the high-side reference voltage is the input voltage to the IVREF1 pin.

"Low-side reference voltage < analog input voltage < high-side reference voltage"

Figure 15 - 8 Comparator i (i = 0 or 1) Operation Example in Window Mode



Caution **The above diagram applies when CiFCK1 to CiFCK0 in the COMPFIR register = 00B (no filter) and CiEDG = 1 (both edges). When CiEDG = 0 and CiEPO = 0 (rising edge), CMPIFi changes as shown by (A) only. When CiEDG = 0 and CiEPO = 1 (falling edge), CMPIFi changes as shown by (B) only.**

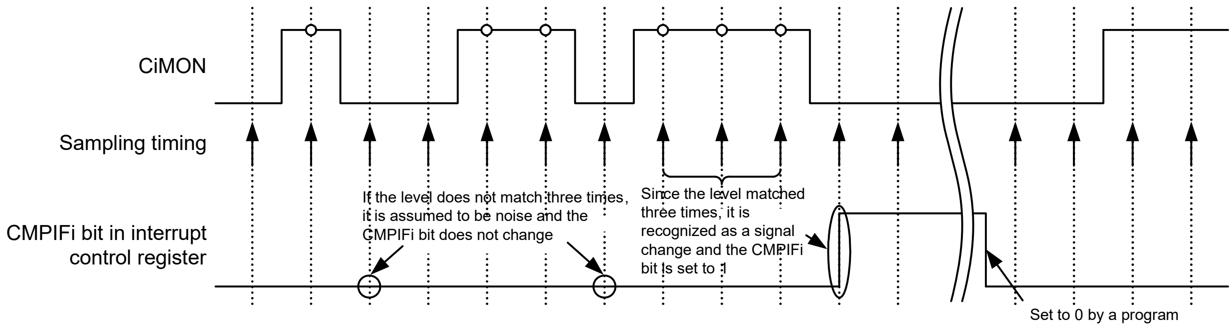
When using the comparator i interrupt, set CiIE in the COMPOCR register to 1 (interrupt request enabled). If the comparison result changes at this time, a comparator i interrupt request is generated. For details on interrupt requests, refer to **15.4.2 Comparator i (i = 0 or 1) Interrupts**.

15.4.1 Comparator i Digital Filter (i = 0 or 1)

Comparator i contains a digital filter. The sampling clock can be selected by bits CiFCK1 - 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 15 - 9 shows the Comparator i (i = 0 or 1) Digital Filter and Interrupt Operation Example.

Figure 15 - 9 Comparator i (i = 0 or 1) Digital Filter and Interrupt Operation Example



Caution The above operation example applies when bits CiFCK1 to CiFCK0 in the COMPFIR register is 01B, 10B, or 11B (digital filter enabled).

15.4.2 Comparator i (i = 0 or 1) 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.

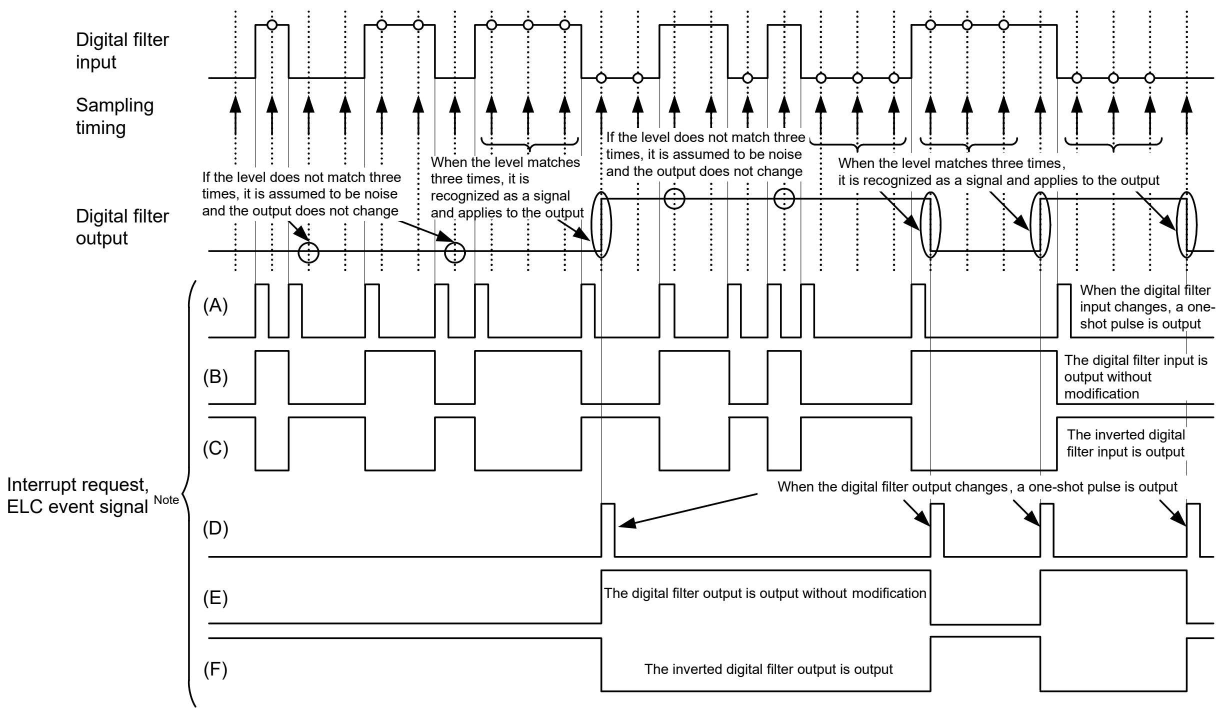
When using the comparator i interrupt, set the CiIE bit in the COMPOCR register to 1 (interrupt request output enabled). The condition for interrupt request generation can be set by the COMPFIR register. The comparator outputs can also be passed through the digital filter. Three different sampling clocks can be selected for the digital filter.

For details on the register setting and interrupt request generation, refer to **15.3.4 Comparator filter control register (COMPFIR)** and **15.3.5 Comparator output control register (COMPOCR)**.

15.4.3 Event signal output to event link controller (ELC)

An event signal to the ELC 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 to the ELC are always output regardless of the CiIE bit in the COMPOCR register. Set registers ELSELR18 and ELSELR19 for the ELC to select the event output destination and to stop linking events.

Figure 15 - 10 Digital Filter and Interrupt Request/Event Signal Output to the ELC Operation



Note When the CiIE bit (i = 0, 1) is 1, the same waveform is generated for an interrupt request and an ELC event signal. When the CiIE bit (i = 0, 1) 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 (i = 0, 1) 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 (i = 0, 1) 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 edges), (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 (falling edge).

15.4.4 Comparator i Output (i = 0 or 1)

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 output polarity (non-inverted output or inverted output) and output enabled or disabled. For the correspondence between the register setting and the comparator output, refer to **15.3.5 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 15 - 2 Procedure for Setting Comparator Associated Registers).
<2> Select the polarity of the VCOUT0 or VCOUT1 output and enable the output (step 10 as listed in Table 15 - 2 Procedure for Setting Comparator Associated Registers).
<3> Set the corresponding port mode control register bit for 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).

15.4.5 Stopping or Supplying Comparator Clock

To stop the comparator clock by setting peripheral enable register 1 (PER1), use the following procedure:

<1> Set the CiIE bit (i = 0, 1) 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 bit in the IF1H register (clear any unnecessary 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.

- The comparator is set to an interrupt request on one-edge detection (CnEDG = 0), an interrupt request at the rising edge for the comparator, and IVCMP > IVREF

- The comparator is set to an interrupt request on one-edge detection (CnEDG = 0), an interrupt request at the falling edge for the comparator, and IVCMP < IVREF

CHAPTER 16 OPERATIONAL AMPLIFIER

The number of operational amplifier input and output pins differs depending on the product.

Unit	I/O Pin	48-Pin	32-Pin	30-Pin	24-Pin	20-Pin
Unit 0 (Operational amplifier 0)	AMP0+, AMP0- (input)	√	√	√	√	√
	AMP0O (output)	√	√	√	√	√
Unit 1 (Operational amplifier 1)	AMP1+, AMP1- (input)	√	√	√	×	×
	AMP1O (output)	√	√	√	×	×
Unit 2 (Operational amplifier 2)	AMP2+, AMP2- (input)	√	√	√	×	×
	AMP2O (output)	√	√	√	×	×
Unit 3 (Operational amplifier 3)	AMP3+, AMP3- (input)	√	√	√	√	√
	AMP3O (output)	√	√	√	√	√

16.1 Functions of Operational Amplifier

Operational amplifiers can be used to amplify small analog input voltages and output the amplified voltages. This MCU has a total of four differential operational amplifier units with two input pins and one output pin.

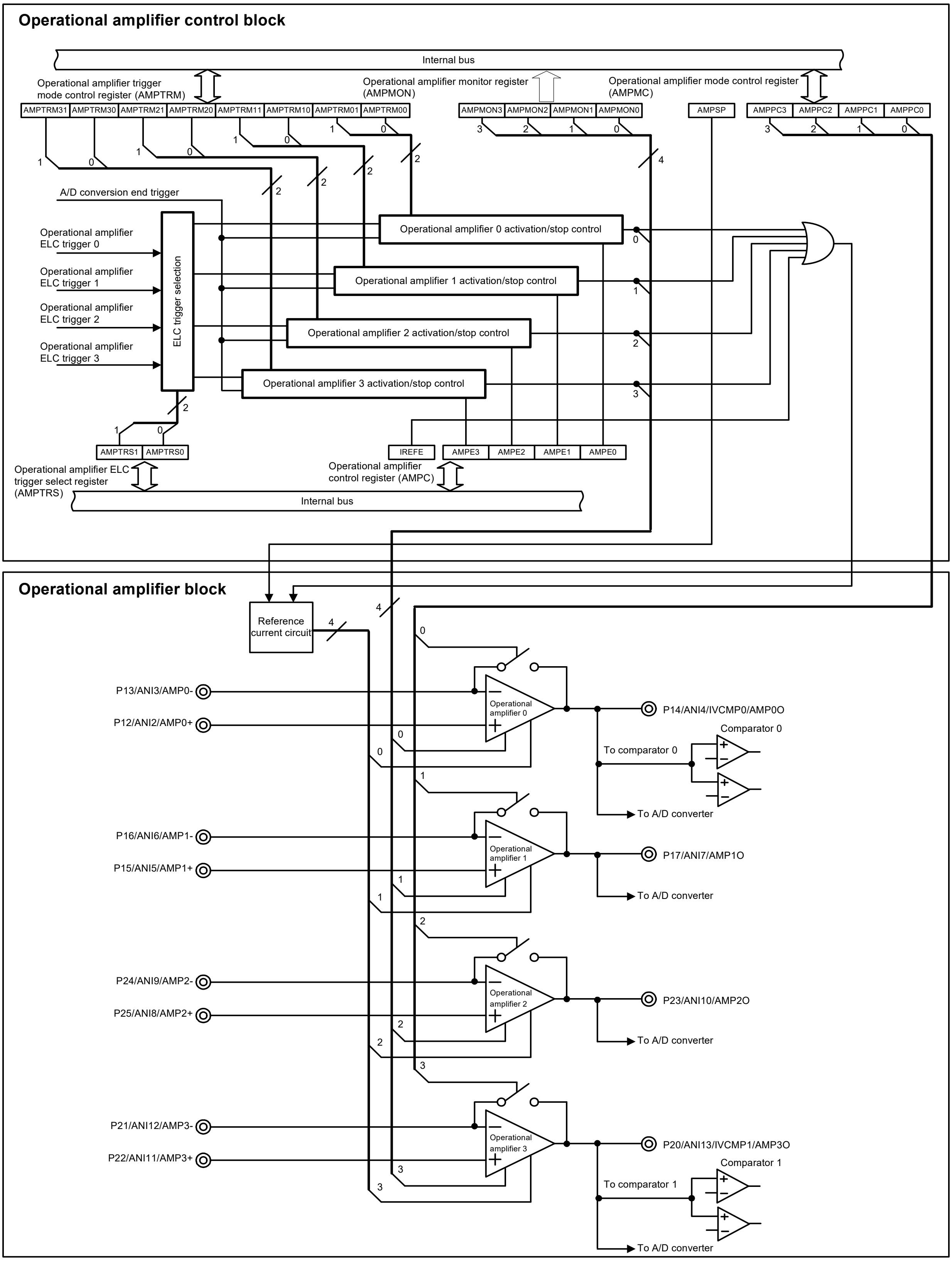
The operational amplifiers have the following functions.

- Among the four units, operational amplifiers 0 and 3 are internally connected to the positive side signal input of comparator 0 and the positive side signal input of comparator 1, respectively.
- The output signals from all units can be used for the input signals to the A/D converter.
- High-speed mode (high-current consumption) and low-power mode (slow-speed response) are supported and either mode can be selected based on trade-offs between the response speed and current consumption.
- Operation can be started by each trigger from the ELC, and operation can also be started by an ELC trigger even in STOP mode.
- Operation can be stopped by an A/D conversion end trigger.

16.2 Configuration of Operational Amplifier

Figure 16 - 1 shows a Block Diagram of Operational Amplifier.

Figure 16 - 1 Block Diagram of Operational Amplifier



16.3 Registers Controlling Operational Amplifier

Table 16 - 1 lists the Registers Used to Control the Operational Amplifiers.

Table 16 - 1 Registers Used to Control the Operational Amplifiers

Item	Configuration
Control registers	Operational amplifier mode control register (AMPMC)
	Operational amplifier trigger mode control register (AMPTRM)
	Operational amplifier ELC trigger select register (AMPTRS)
	Operational amplifier control register (AMPC)
	Operational amplifier monitor register (AMPMON)
	Port mode register 1 (PM1)
	Port mode register 2 (PM2)
	Port mode control register 1 (PMC1)
	Port mode control register 2 (PMC2)

16.3.1 Operational amplifier mode control register (AMPMC)

Figure 16 - 2 Format of Operational amplifier mode control register (AMPMC)

Address: F0348H After reset: 00H R/W

Symbol	<7>	6	5	4	<3>	<2>	<1>	<0>
AMPMC	AMPSP	0	0	0	AMPPC3	AMPPC2	AMPPC1	AMPPC0

AMPSP	Operation mode selection (common to all units)
0	Low-power mode (low-speed)
1	High-speed mode

AMPPCn	Operational amplifier precharge control status
0	Precharging is stopped
1	Precharging is enabled
When precharging is enabled, the negative input pin and output pin of operational amplifier n are short-circuited and connected to the voltage follower.	

Caution 1. Set AMPSP bit while the value of the AMPC register is 00H (operational amplifier and operational amplifier reference current circuit are stopped).

Caution 2. Be sure to set bits that are not used in this register to the initial value.

Remark n: Unit number (n = 0, 1, 2, 3)

16.3.2 Operational amplifier trigger mode control register (AMPTRM)

Figure 16 - 3 Format of Operational amplifier trigger mode control register (AMPTRM)

Address: F0349H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
AMPTRM	AMPTRM31	AMPTRM30	AMPTRM21	AMPTRM20	AMPTRM11	AMPTRM10	AMPTRM01	AMPTRM00

AMPTRMn1	AMPTRMn0	Operational amplifier function activation/stop trigger control Note 3
0	0	Software trigger mode • The operational amplifier can be activated/stopped by setting the AMPC register • The operational amplifier cannot be activated by an ELC trigger • The operational amplifier cannot be controlled by an A/D conversion end trigger
0	1	ELC trigger mode • The operational amplifier can be set to wait for an ELC trigger or stopped by setting the AMPC register • The operational amplifier can be activated by an ELC trigger Note 1 • The operational amplifier cannot be controlled by an A/D conversion end trigger
1	0	Setting prohibited
1	1	ELC and A/D trigger mode • The operational amplifier can be set to wait for an ELC trigger or stopped by setting the AMPC register • The operational amplifier can be activated by an ELC trigger Note 1 • The operational amplifier can be stopped by an A/D conversion end trigger Note 2

Note 1. When using an ELC trigger to activate the operational amplifier, first specify various settings related to the event link controller (ELC), set the AMPTRS register, and then use the AMPC register to set the operation control bit of the operational amplifier to be activated to 1 (operational amplifier wait state is enabled).

Note 2. An A/D conversion end trigger is always generated at the end of A/D conversion.

Note 3. When changing the set values of AMPTRMn1 and AMPTRMn0, make sure that the AMPEn bit in the AMPC register is 0 (operation amplifier is stopped).

Remark n: Unit number (n = 0, 1, 2, 3)

16.3.3 Operational amplifier ELC trigger select register (AMPTRS)

Figure 16 - 4 Format of Operational amplifier ELC trigger select register (AMPTRS)

Address: F034AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
AMPTRS	0	0	0	0	0	0	AMPTRS1	AMPTRS0

AMPTRS1	AMPTRS0	ELC trigger selection Note
0	0	Operational amplifier 0: Operational amplifier ELC trigger 0 Operational amplifier 1: Operational amplifier ELC trigger 1 Operational amplifier 2: Operational amplifier ELC trigger 2 Operational amplifier 3: Operational amplifier ELC trigger 3
0	1	Operational amplifier 0: Operational amplifier ELC trigger 0 Operational amplifier 1: Operational amplifier ELC trigger 0 Operational amplifier 2: Operational amplifier ELC trigger 1 Operational amplifier 3: Operational amplifier ELC trigger 1
1	0	Setting prohibited
1	1	Operational amplifier 0: Operational amplifier ELC trigger 0 Operational amplifier 1: Operational amplifier ELC trigger 0 Operational amplifier 2: Operational amplifier ELC trigger 0 Operational amplifier 3: Operational amplifier ELC trigger 0

Note Do not change the value of the AMPTRS register after setting the AMPTRM register.

Caution Be sure to set bits that are not used in this register to the initial value.

16.3.4 Operational amplifier control register (AMPC)

Figure 16 - 5 Format of Operational amplifier control register (AMPC)

Address: F034BH After reset: 00H R/W

Symbol	<7>	6	5	4	<3>	<2>	<1>	<0>
AMPC	IREFE	0	0	0	AMPE3	AMPE2	AMPE1	AMPE0

IREFE	Operation control of operational amplifier reference current circuit
0	Operational amplifier reference current circuit is stopped
1	Operation of operational amplifier reference current circuit is enabled

AMPEn	Operation control of operational amplifier
0	Operation amplifier is stopped
1	Software trigger mode: Operation of operational amplifier is enabled Note ELC trigger mode or ELC and A/D trigger mode: Wait for ELC is enabled

Note Operation of the operational amplifier reference current circuit is also enabled regardless of the IREFE bit setting.
Be sure to set bits 2 and 1 in 24-pin and 20-pin products to 0. Be sure to set the bits to 0 for a unit that is not to be used.

Caution Be sure to set bits that are not used in this register to the initial value.

Remark n: Unit number (n = 0, 1, 2, 3)

16.3.5 Operational amplifier monitor register (AMPMON)

Figure 16 - 6 Format of Operational amplifier monitor register (AMPMON)

Address: F034CH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
AMPMON	0	0	0	0	AMPMON3	AMPMON2	AMPMON1	AMPMON0

AMPMONn	Operational amplifier status
0	Operational amplifier n is stopped
1	Operational amplifier n is operating

Caution 1. This register is used to asynchronously reflect whether each operational amplifier is operating/stopped. To determine the operational amplifier state, read this register continuously and confirm that the bit state has changed. After that, read this register again for confirmation and determine whether the operational amplifier state has changed.
When an ELC trigger or A/D conversion end trigger synchronized with the clock or a software trigger in the other interrupt routine is used to control the operational amplifier, the timing to operate/stop the operational amplifier can be estimated, such as for checking normal operation. In this case, read this register after one CPU/peripheral clock cycle when the corresponding trigger or interrupt affecting the operational amplifier state has occurred.

Caution 2. Be sure to set bits that are not used in this register to the initial value.

Remark n: Unit number (n = 0, 1, 2, 3)

16.3.6 Registers controlling port function of analog input pins

To use the AMP0+, AMP0-, AMP0O, AMP1+, AMP1-, AMP1O, AMP2+, AMP2-, AMP2O, AMP3+, AMP3-, and AMP3O pins as analog I/O of the operational amplifier, set the corresponding bits in port mode registers 1 and 2 (PM1, PM2) and port mode control registers (PMC1, PMC2) to 1.

For details, refer to **4.3.1 Port mode registers (PMxx)** and **4.3.6 Port mode control registers (PMCxx)**.

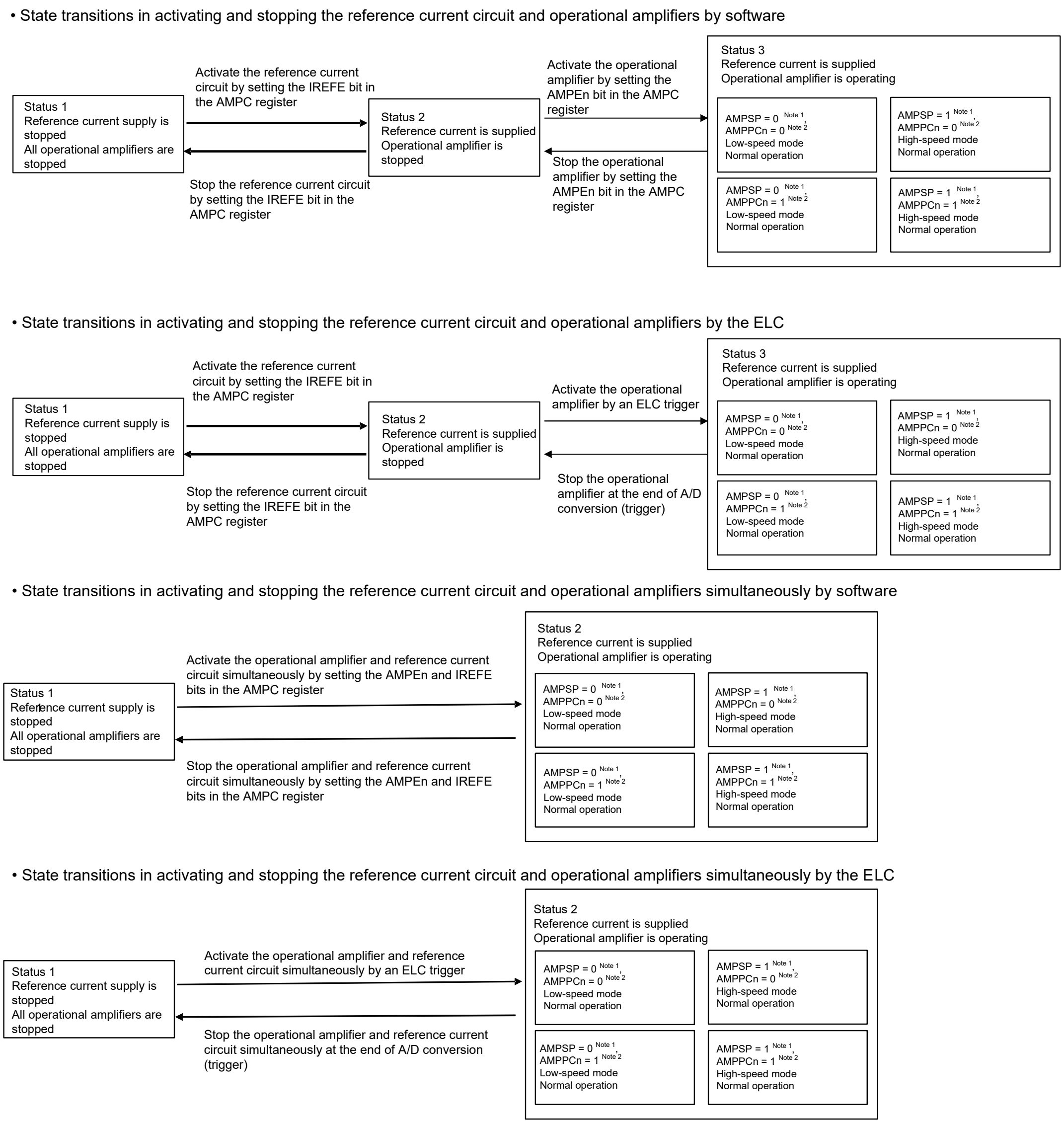
When using the operational amplifier function, set the used operational amplifier I/O pins to input pins using the port mode registers (PM1, PM2) and analog pins using the port mode control registers (PMC1, PMC2).

16.4 Operation

16.4.1 State Transitions

Figure 16 - 7 shows state transitions when the operational amplifier and reference current circuit are activated or stopped using the operational amplifier control circuit.

Figure 16 - 7 Operational Amplifier State Transitions



Note 1. Set the AMPSP bit in the AMPMC register and the AMPTRS and AMPTRM registers in status 1.
Note 2. Set the AMPPCn bit in the AMPMC register in status 3.

Caution **State transitions other than those shown in figure 16-7 are prohibited in activating and stopping the operational amplifiers and reference current circuit.**

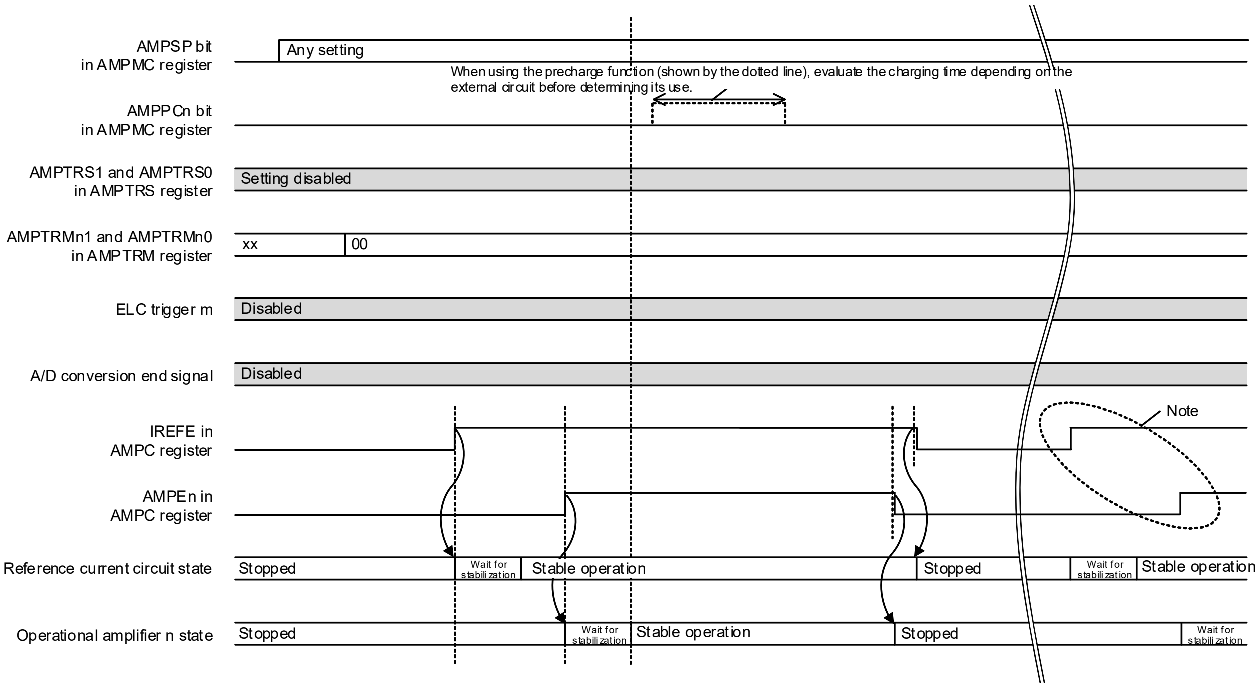
Remark 1. A stabilization wait time is necessary after supply of the reference current and operation of the operational amplifier are set before each operation actually starts. For details on the stabilization wait time, refer to **CHAPTER 34 ELECTRICAL SPECIFICATIONS**.

Remark 2. An ELC trigger and end of A/D conversion can be used to activate/stop only the operational amplifier that is preset to be used by setting the AMPTRM register.

16.4.2 Operational Amplifier Control Operation

Figures 16 - 8 to 16 - 11 show Operational Amplifier Control Operation.

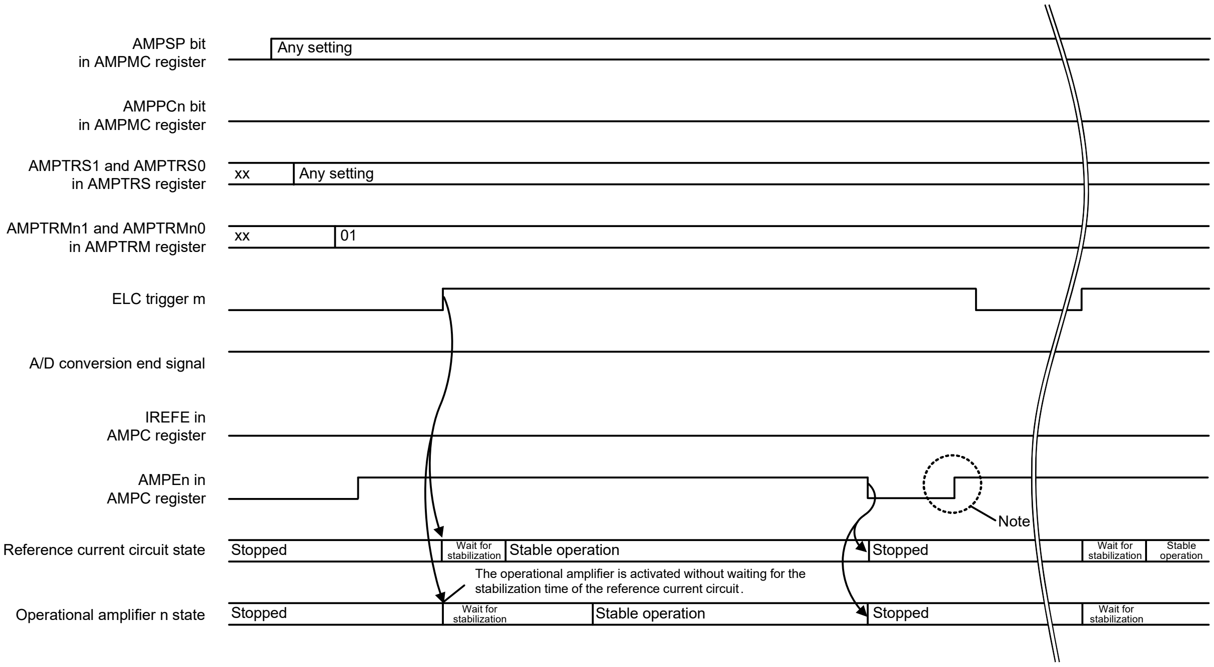
Figure 16 - 8 Operational Amplifier Control Operation (Software trigger mode is used for control) (When the reference current circuit and operational amplifier are activated/stopped by software trigger mode)



Note When operating/stopping the operational amplifier continuously, set the IREFE and AMPEN bits again as in the first setting after the operational amplifier is stopped.

Remark n: Unit number (n = 0, 1, 2, 3)
m: ELC trigger used to control operational amplifier unit n selected by the AMPTRS register

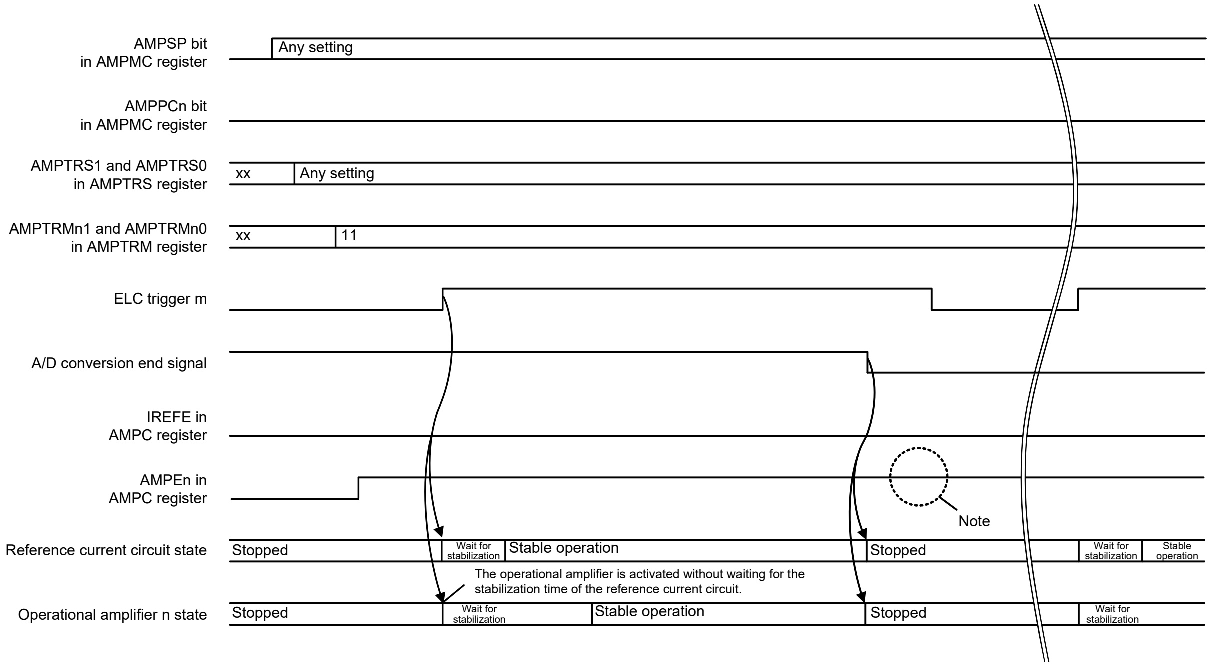
Figure 16 - 9 Operational Amplifier Control Operation (ELC trigger mode is used for activation) (When the reference current circuit and operational amplifier are activated by an ELC trigger and stopped by setting the SFR)



Note When operating/stopping the operational amplifier continuously, use the AMPEN bit again as in the first setting, and set the operational amplifier to wait for an ELC trigger after it is stopped.

Remark n: Unit number (n = 0, 1, 2, 3)
m: ELC trigger used to control operational amplifier unit n selected by the AMPTRS register
Set the function used for ELC event generation and the peripheral function to be linked (the ELSELR register) in advance.

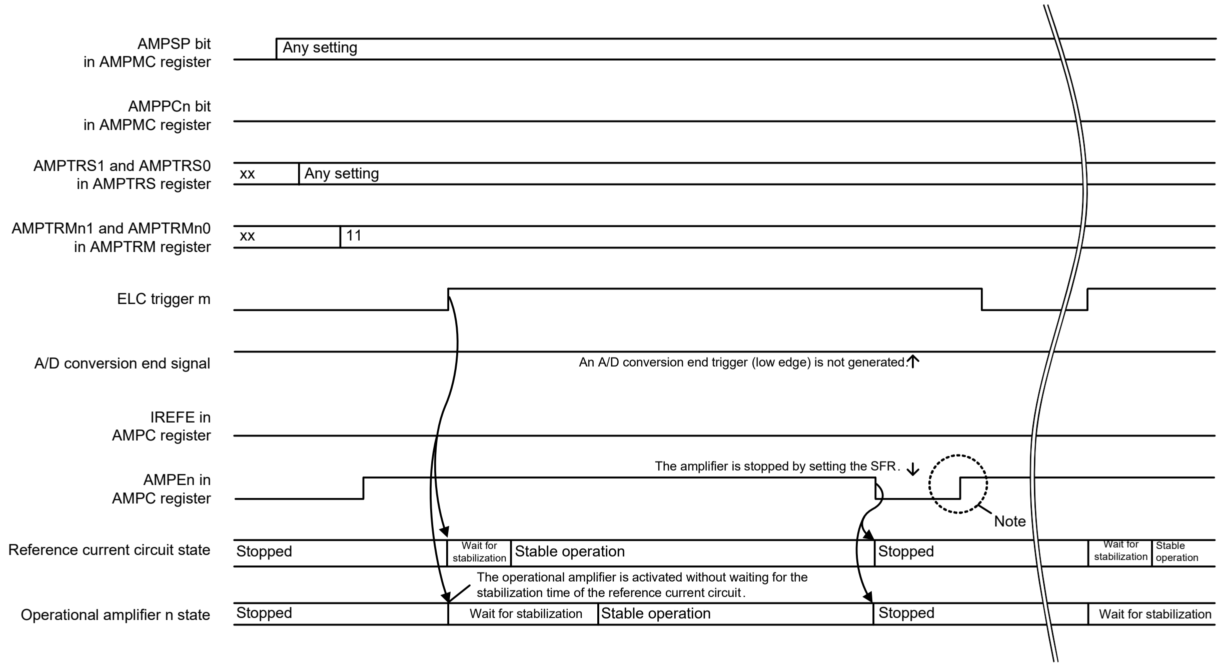
Figure 16 - 10 Operational Amplifier Control Operation (ELC and A/D trigger mode (1)) (When the reference current circuit and operational amplifier are activated by an ELC trigger and stopped by an A/D conversion end (trigger))



Note When operating/stopping the operational amplifier continuously, it is not necessary to set the registers again because the operational amplifier waits for an ELC trigger after it is stopped.

Remark n: Unit number (n = 0, 1, 2, 3)
m: ELC trigger used to control operational amplifier unit n selected by the AMPTRS register
Set the function used for ELC event generation and the peripheral function to be linked (the ELSELR register) in advance.

Figure 16 - 11 Operational Amplifier Control Operation (ELC and A/D trigger mode (2))
(When the reference current circuit and operational amplifier are stopped by setting the SFR under the setting that they can be activated by an ELC trigger and stopped by an A/D conversion end (trigger))



Note When operating/stopping the operational amplifier continuously, use the AMPEN bit again as in the first setting, and set the operational amplifier to wait for an ELC trigger after it is stopped.

Remark n: Unit number (n = 0, 1, 2, 3)
m: ELC trigger used to control operational amplifier unit n selected by the AMPTRS register
Set the function used for ELC event generation and the peripheral function to be linked (the ELSELR register) in advance.

16.4.3 Software trigger mode

The following flowchart shows the procedure to operate and stop the operational amplifier using a software trigger with examples of register settings.

Step	Description
PMCxx = 1	Set the port mode control registers (PMC1, PMC2) (Set the AMPn+, AMPn-, and AMPnO pins to analog mode)
↓ PMxx = 1	Set the port mode registers (PM1, PM2) (Set the AMPn+, AMPn-, and AMPnO pins to input)
↓ AMPC = 00H	Set all operational amplifiers units and the operational amplifier reference current circuit to be stopped
↓ AMPSP = 0 or 1 Note 1	Set the AMPSP bit to any value
↓ AMPTRMn1, AMPTRMn0 = 00	Set operational amplifier unit n to software trigger mode
↓ IREFE = 1	Operate the operational amplifier reference current circuit (can be skipped Note 2)
↓ Stabilization wait time	Wait for the stabilization wait time <1> Note 2
↓ AMPEn = 1	Generate a software trigger for operational amplifier unit n (start operation of the operational amplifier)
↓ Stabilization wait time	Wait for the stabilization wait time <2> Note 2
↓ AMPPCn = 1	If precharging is to be used, this starts precharging of the external capacitor connected to the operational amplifier n. If precharging is not to be used, set the AMPPCn bit to 0.
↓ Precharging period	Wait for the precharging period
↓ AMPPCn = 0	Complete precharging of the external capacitor of operational amplifier unit n
↓ The operational amplifier is operating	
↓ AMPEn = 0	Stop operation of operational amplifier unit n
↓ IREFE = 0 Note 3	Stop the operational amplifier reference current circuit
↓ The operational amplifier is stopped	

Note 1. Set AMPSP bit while the value of the AMPC register is 00H (operational amplifier and operational amplifier reference current circuit are stopped).

(**Notes** and **Caution** continue on the next page.)

Note 2. When the reference current circuit for the operational amplifiers is operating, the total of the stabilization wait times <1> and <2> must satisfy the relevant specification for operation stabilization wait time when the operational amplifier and reference current circuit are activated simultaneously in **34.6.4 Operational amplifier characteristics**.
When the reference current circuit for the operational amplifiers is not operating, stabilization wait time <2> must satisfy the relevant specification for operation stabilization wait time when only the operational amplifier is activated in **34.6.4 Operational amplifier characteristics**.

Note 3. IREFE = 0 and AMPEn = 0 can be set at the same time.

Caution For details on the stabilization wait time, refer to CHAPTER 34 ELECTRICAL SPECIFICATIONS.

16.4.4 ELC trigger mode

The following flowchart shows the procedure to operate the operational amplifier using an ELC trigger with examples of register settings.

This is an example of processing when the operational amplifier is activated by an ELC trigger and stopped by software repeatedly.

Step	Description
Set the ELC	
↓ Set the peripheral function used for ELC event generation	
↓ PMCxx = 1	Set the port mode control registers (PMC1, PMC2) (Set the AMPn+, AMPn-, and AMPnO pins to analog mode)
↓ PMxx = 1	Set the port mode registers (PM1, PM2) (Set the AMPn+, AMPn-, and AMPnO pins to input)
↓ AMPC = 00H	Set all operational amplifiers units and the operational amplifier reference current circuit to be stopped
↓ AMPSP = 0 or 1 Note 1	Set the AMPSP bit to any value
↓ AMPTRS1, AMPTRS0 = Any value Note 2	Select an ELC trigger
↓ AMPTRMn1, AMPTRMn0 = 01 Note 2	Set operational unit n to ELC trigger mode
↓ AMPEn = 1 Note 3	Set operational amplifier unit n to wait for an ELC trigger * Set each peripheral function so that an ELC trigger is generated after AMPEn is set to 1.
↓ Wait for an ELC trigger	
↓ Generate an ELC trigger	Start operation of operational amplifier unit n
↓ The operational amplifier is operating	
↓ AMPEn = 0	Set operation of operational amplifier unit n to be stopped
↓ The operational amplifier is stopped	
↓ AMPEn = 1	Set operational amplifier unit n to wait for an ELC trigger
↓ Wait for an ELC trigger	The operational amplifier is stopped until the next ELC trigger is generated

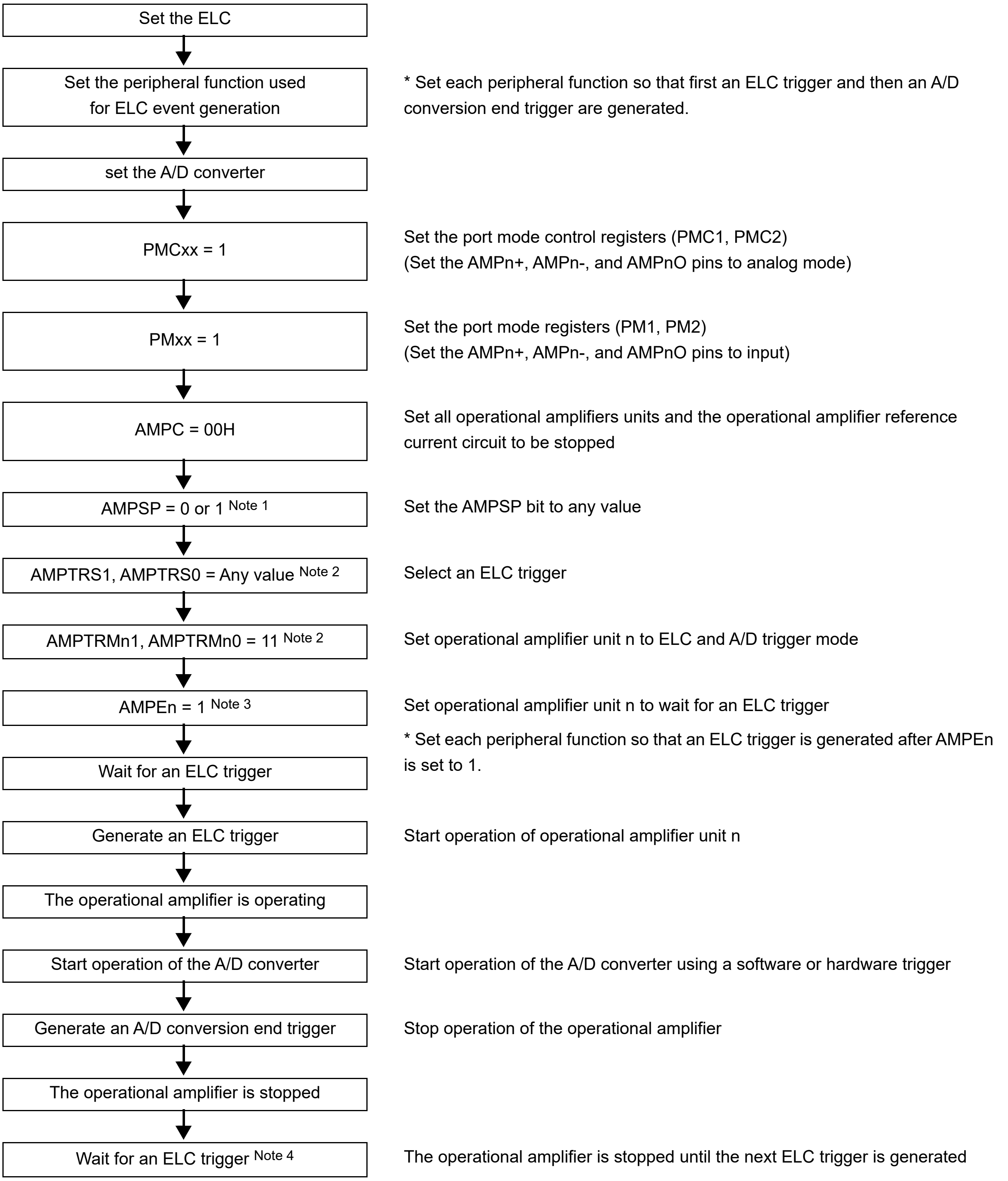
Note 1. Set AMPSP bit while the value of the AMPC register is 00H (operational amplifier and operational amplifier reference current circuit are stopped).

Note 2. Set these bits while the AMPEn bit in the AMPC register is 0.

Note 3. To operate the operational amplifier reference current circuit continuously, set the IREFE bit in the AMPC register to 1 at this timing.

16.4.5 ELC and A/D Trigger Mode

The following flowchart shows the procedure to activate the operational amplifier using an ELC trigger and to stop the operational amplifier using an A/D conversion end trigger with examples of register settings. This is an example of processing when the operational amplifier is activated by an ELC trigger and stopped by an A/D conversion end trigger repeatedly.



Note 1. Set AMPSP bit while the value of the AMPC register is 00H (operational amplifier and operational amplifier reference current circuit are stopped).

Note 2. Set these bits while the AMPEn bit in the AMPC register is 0.

Note 3. Set this bit while the peripheral function used for ELC trigger event generation and the A/D converter are stopped. To operate the operational amplifier reference current circuit continuously, set the IREFE bit in the AMPC register to 1 at this timing.

Note 4. To stop wait operation for a trigger, set the AMPEn bit in the AMPC register to 0.
To forcibly stop the operational amplifier after it is activated by an ELC trigger, also set the AMPEn bit in the AMPC register to 0.

16.5 Usage Notes on Operational Amplifier

(1) When using the operational amplifier function, set the used operational amplifier I/O pins to input pins using the port mode registers (PM1, PM2) and analog pins using the port mode control registers (PMC1, PMC2).

(2) When connecting bypass capacitors to the AVDD and AVSS pins that are the power supply pins for the operational amplifier function, place them as close to the chip as possible (to keep the wiring short) and prevent noise from the device, board, and peripheral components.

(3) In addition to SFR settings, the operational amplifier function can be activated by an ELC trigger and stopped at the end of A/D conversion. The reference current circuit can be stopped at the end of A/D conversion.
Therefore, design applications (circuits and programs) conforming to the operation flows in order to prevent these asynchronous triggers from causing conflicts between activation/stop control (conflicting control).

(4) For the pins multiplexed with positive and negative input for the operational amplifier function and analog input for the A/D converter, while they are used as operational amplifier positive and negative pins, do not perform A/D conversion on the analog input pins multiplexed these pins.

(5) The output from operational amplifier 0 is connected to the positive side signal input of comparator 0 (IVCMP0) and an analog input pin (ANI4). Do not input a signal to P14/ANI4/IVCMP0/AMP0O while operational amplifier 0 is in use. Similarly, do not input a signal to P20/ANI13/IVCMP1/AMP3O while operational amplifier 3 is in use.

CHAPTER 17 SERIAL ARRAY UNIT

The serial array unit has two serial channels. Each channel can achieve simplified SPI (CSI[Note]), UART, and simplified I²C communications.

Function assignment of each channel supported by the RL78/I1D is as shown below.

<R> **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.

- 20-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	—		—

- 24, 32-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01

- 30-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	—		—

- 48-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00 (supporting slave select input function)	UART0	IIC00
	1	CSI01		IIC01

Caution **Most of the following descriptions in this chapter use the unit and channel configuration of the 48-pin products as an example.**

17.1 Functions of Serial Array Unit

Each serial interface supported by the RL78/I1D has the following features.

17.1.1 Simplified SPI (CSI00, CSI01)

Data is transmitted or received in synchronization with the serial clock (SCK) output from the master channel.
Simplified SPI is a clock synchronous communications function that uses three 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 **17.5 Operation of Simplified SPI (CSI00, CSI01) 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

CSI00 supports the SNOOZE mode. When SCK input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. Only the following simplified SPI (CSI) can be specified.
• CSI00

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics. For details, see **CHAPTER 34 ELECTRICAL SPECIFICATIONS**.

17.1.2 UART (UART0)

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel).

For details about the settings, see **17.7 Operation of UART (UART0) Communication**.

[Data transmission/reception]

• Data length of 7, 8, or 9 bits
• Select the MSB/LSB first
• Level setting of transmit/receive data and select of reverse
• Parity bit appending and parity check functions
• Stop bit appending

[Interrupt function]

• Transfer end interrupt/buffer empty interrupt
• Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

• Framing error, parity error, or overrun error

In addition, UART reception of the following channels supports the SNOOZE mode. When RxD input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. Only the following UART can be specified when the high-speed on-chip oscillator clock (f_{IH}) is selected for the CPU/peripheral hardware clock (f_{CLK}) in the SNOOZE mode.

• UART0

17.1.3 Simplified I2C (IIC00, IIC01)

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 **17.8 Operation of Simplified I2C (IIC00, IIC01) 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

<R> • 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 the processing flow in **17.8.3 (2)** for details.

17.2 Configuration of Serial Array Unit

The serial array unit includes the following hardware.

Table 17 - 1 Configuration of Serial Array Unit

Item	Configuration
Shift register	8 bits or 9 bits
Buffer register	Lower 8 bits or 9 bits of serial data register mn (SDRmn) Note
Serial clock I/O	SCK00, SCK01 pins (for simplified SPI) and SCL00, SCL01 pins (for simplified I^2C)
Serial data input	SI00, SI01 pins (for simplified SPI), RxD0 pin (for UART)
Serial data output	SO00, SO01 pins (for simplified SPI), TxD0 pin (for UART)
Serial data I/O	SDA00, SDA01 pins (for simplified I^2C)
Slave select input	$\overline{\text{SSI00}}$ pin (for slave select input function)
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 register 5 (PIM5) • Port output mode register 5 (POM5) • Port mode registers 5, 6 (PM5, PM6) • Port registers 5, 6 (P5, P6)

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

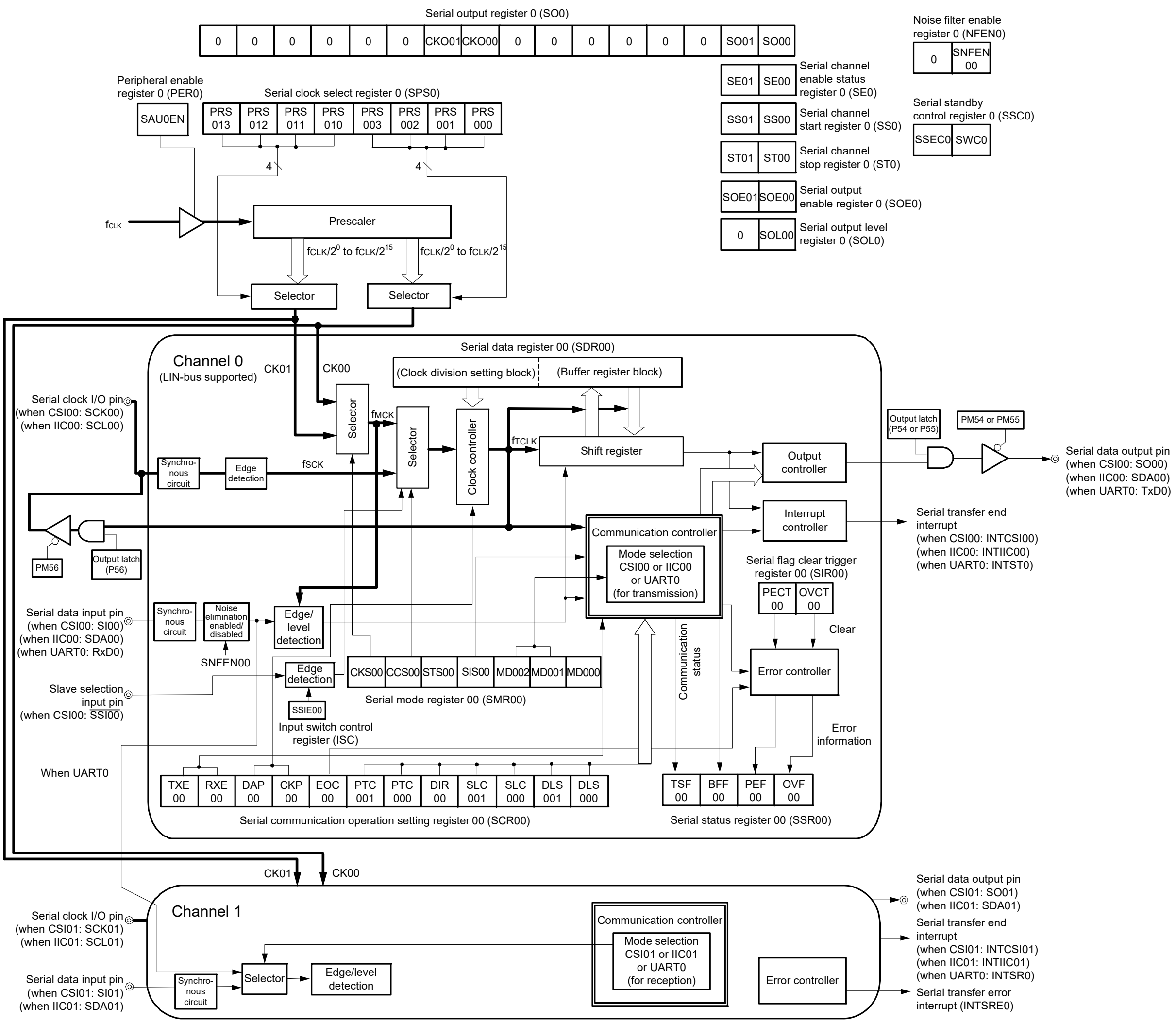
Note The lower 8 bits of serial data register mn (SDRmn) can be read or written as the following SFRs, depending on the communication mode.

- 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), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), q: UART number (q = 0), r: IIC number (r = 00, 01)

Figure 17 - 1 shows the Block Diagram of Serial Array Unit 0.

Figure 17 - 1 Block Diagram of Serial Array Unit 0



17.2.1 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 0 to 8) 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 8/9 bits of serial data register mn (SDRmn).

	8	7	6	5	4	3	2	1	0
Shift register									

Note Only the following UARTs can be specified for the 9-bit data length.
• UART0

17.2.2 Lower 8/9 bits of the serial data register mn (SDRmn)

The SDRmn register 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 operation clock (f_{MCK}).

When data is received, parallel data converted by the shift register is stored in the lower 8/9 bits. When data is to be transmitted, set transmit data to be transferred to the shift register to the lower 8/9 bits.

The data stored in the lower 8/9 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 SDRmn register)
• 8-bit data length (stored in bits 0 to 7 of SDRmn register)
• 9-bit data length (stored in bits 0 to 8 of SDRmn register) [Note 1]

The SDRmn register can be read or written in 16-bit units.

The lower 8/9 bits of the SDRmn register can be read or written [Note 2] as the following SFRs, 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)

Reset signal generation clears the SDRmn register to 0000H.

Note 1. Only the following UARTs can be specified for the 9-bit data length.
• UART0

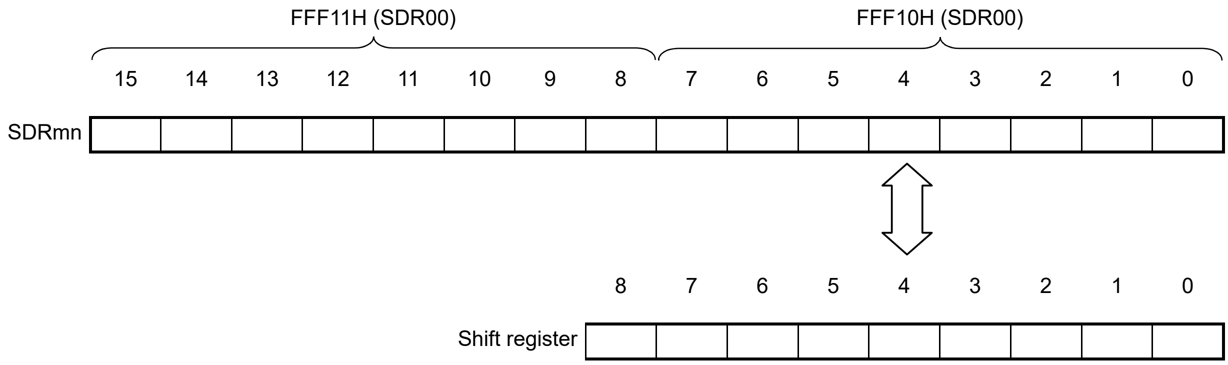
Note 2. Using an 8-bit memory manipulation instruction to write to the SDRmn[7:0] bits while operation is stopped (SEmn = 0) is prohibited (doing so clears the SDRmn[15:9] bits to 0).

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), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), q: UART number (q = 0), r: IIC number (r = 00, 01)

Figure 17 - 2 Format of Serial data register mn (SDRmn) (mn = 00, 01)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01) After reset: 0000H R/W



Remark For the function of the higher 7 bits of the SDRmn register, see **17.3 Registers Controlling Serial Array Unit**.

17.3 Registers Controlling Serial Array Unit

Serial array unit is controlled by the following registers.

- 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 level register m (SOLm)
- Serial output register m (SOm)
- Serial standby control register m (SSCm)
- Input switch control register (ISC)
- Noise filter enable register 0 (NFEN0)
- Port input mode register 5 (PIM5)
- Port output mode register 5 (POM5)
- Port mode registers 5, 6 (PM5, PM6)
- Port registers 5, 6 (P5, P6)

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

17.3.1 Peripheral enable register 0 (PER0)

PER0 is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.
When serial array unit 0 is used, be sure to set bit 2 (SAU0EN) of this register to 1.
The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears the PER0 register to 00H.

Figure 17 - 3 Format of Peripheral enable register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	4	3	<2>	1	<0>
PER0	RTCWEN	0	ADCEN	0	0	SAU0EN	0	TAU0EN

SAUmEN	Control of serial array unit m input clock supply
0	Stops supply of input clock. • SFR used by serial array unit m cannot be written.
1	Enables input clock supply. • SFR used by serial array unit m can be read/written.

Caution 1. When setting serial array unit m, be sure to first set the following registers with the SAUmEN bit set to 1. If SAUmEN = 0, writing to a control register of serial array unit m is ignored (except for the input switch control register (ISC), noise filter enable register 0 (NFEN0), port input mode register 5 (PIM5), port output mode register 5 (POM5), port mode registers 5, 6 (PM5, PM6), and port registers 5, 6 (P5, P6).

- **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)**

Caution 2. Be sure to clear the following bits to 0.
Bits 1, 3, 4, and 6

17.3.2 Peripheral reset control register 0 (PRR0)

This register is used for individual reset control of each peripheral hardware.
This MCU controls reset and reset release of each peripheral hardware supported by the PRR0 register.
To reset the serial array unit, be sure to set bit 2 (SAU0RES) to 1.
The PRR0 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears the PRR0 register to 00H.

Figure 17 - 4 Format of Peripheral reset control register 0 (PRR0)

Address: F00F1H After reset: 00H R/W

Symbol	7	6	<5>	4	3	<2>	1	<0>
PRR0	0	0	ADCRES	0	0	SAU0RES	0	TAU0RES

SAU0RES	Reset control of serial array unit
0	Serial array unit reset release
1	Serial array unit reset state

17.3.3 Serial clock select register m (SPSm)

The SPSm register is a 16-bit register that is used to select two types of operation clocks (CKm0, CKm1) that are commonly supplied to each channel. CKm1 is selected by bits 7 to 4 of the SPSm register, and CKm0 is selected by bits 3 to 0.
Rewriting the SPSm register is prohibited when the register is in operation (when SEmn = 1).
The SPSm register can be set by a 16-bit memory manipulation instruction.
The lower 8 bits of the SPSm register can be set with an 8-bit memory manipulation instruction with SPSmL.
Reset signal generation clears the SPSm register to 0000H.

Figure 17 - 5 Format of Serial clock select register m (SPSm)

Address: F0126H, F0127H (SPS0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPSm	0	0	0	0	0	0	0	0	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRS mk3	PRS mk2	PRS mk1	PRS mk0	Section of operation clock (CKmk) Note					
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz	24 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	12 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	6 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	3 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz	1.5 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156 kHz	313 kHz	625 kHz	750 kHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz	375 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	187.5 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	93.8 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	46.9 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	23.4 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	11.7 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	5.86 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz	2.93 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz	1.46 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61 Hz	153 Hz	305 Hz	610 Hz	732 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 clear bits 15 to 8 to 0.

Remark 1. f_{CLK}: CPU/peripheral hardware clock frequency
Remark 2. m: Unit number (m = 0)
Remark 3. k = 0, 1

17.3.4 Serial mode register mn (SMRmn)

The SMRmn register is a register that sets an operation mode of channel n. It is also used to select an operation clock (f_{MCK}), specify whether the serial clock (f_{SCK}) may be input or not, set a start trigger, an operation mode (simplified SPI (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.

Reset signal generation sets the SMRmn register to 0020H.

Figure 17 - 6 Format of Serial mode register mn (SMRmn) (1/2)

Address: F0110H, F0111H (SMR00) to F0112H, F0113H (SMR01), After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn	0	0	0	0	0	STSmn Note	0	SISmn0 Note	1	0	0	MDmn2	MDmn1	MDmn0

CKSmn	Selection of operation clock (f_{MCK}) of channel n
0	Operation clock CKm0 set by the SPSm register
1	Operation clock CKm1 set by the SPSm register
Operation clock (f_{MCK}) is used by the edge detector. In addition, depending on the setting of the CCSmn bit and the higher 7 bits of the SDRmn register, a transfer clock (f_{TCLK}) is generated.	

CCSmn	Selection of transfer clock (f_{TCLK}) of channel n
0	Divided operation clock f_{MCK} specified by the CKSmn bit
1	Clock input f_{SCK} from the SCKp pin (slave transfer in simplified SPI (CSI) mode)
Transfer clock f_{TCLK} is used for the shift register, communication controller, output controller, interrupt controller, and error controller. When CCSmn = 0, the division ratio of operation clock (f_{MCK}) is set by the higher 7 bits of the SDRmn register.	

STSmn Note	Selection of start trigger source
0	Only software trigger is valid (selected for simplified SPI (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.	

Note The SMR01 register only.

Caution Be sure to clear bits 13 to 9, 7, 4, and 3 (or bits 13 to 6, 4, and 3 for the SMR00 register) to 0. Be sure to set bit 5 to 1.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), q: UART number (q = 0), r: IIC number (r = 00, 01)

Figure 17 - 7 Format of Serial mode register mn (SMRmn) (2/2)

Address: F0110H, F0111H (SMR00) to F0112H, F0113H (SMR01) After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn Note	0	SIS mn0 Note	1	0	0	MD mn2	MD mn1	MD mn0

SIS mn0 Note	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.

MD mn2	MD mn1	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

MD mn0	Selection of interrupt source of channel n
0	Transfer end interrupt
1	Buffer empty interrupt (Occurs when data is transferred from the SDRmn register to the shift register.)
For successive transmission, the next transmit data is written by setting the MDmn0 bit to 1 when SDRmn data has run out.	

Note The SMR01 register only.

Caution Be sure to clear bits 13 to 9, 7, 4, and 3 (or bits 13 to 6, 4, and 3 for the SMR00 register) to 0. Be sure to set bit 5 to 1.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), q: UART number (q = 0), r: IIC number (r = 00, 01)

17.3.5 Serial communication operation setting register mn (SCRmn)

The SCRmn register is a communication operation setting register of channel n. It is used to set a data transmission/reception mode, phase of data and clock, whether an error signal is to be masked or not, parity bit, start bit, stop bit, and data length.

Rewriting the SCRmn register is prohibited when the register is in operation (when SEmn = 1).

The SCRmn register can be set by a 16-bit memory manipulation instruction.

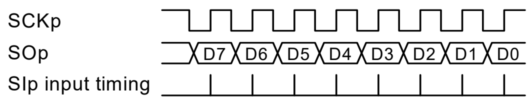
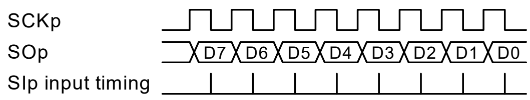
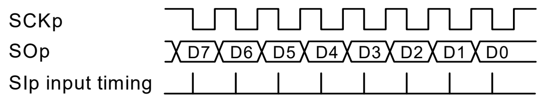
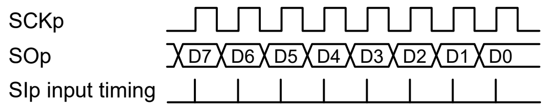
Reset signal generation sets the SCRmn register to 0087H.

Figure 17 - 8 Format of Serial communication operation setting register mn (SCRmn) (1/2)

Address: F0118H, F0119H (SCR00) to F011AH, F011BH (SCR01) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn	0	EOCmn	PTCmn1	PTCmn0	DIRmn	0	SLCmn1 Note 1	SLCmn0	0	1	DLSmn1	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
Be sure to set DAPmn, CKPmn = 0, 0 in the UART mode and simplified I^2C mode.			

EOCmn	Mask control of error interrupt signal (INTSREx (x = 0))
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 = 0 in the simplified SPI (CSI) mode, simplified I^2C mode, and during UART transmission Note 2.	

Note 1. The SCR00 register only.

Note 2. When using CSImn not with EOCmn = 0, error interrupt INTSREn may be generated.

Caution Be sure to clear bits 3, 6, and 11 to 0 (Also clear bit 5 of the SCR01 register to 0). Be sure to set bit 2 to 1.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

Figure 17 - 9 Format of Serial communication operation setting register mn (SCRmn) (2/2)

Address: F0118H, F0119H (SCR00) to F011AH, F011BH (SCR01) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLCm n1 Note 1	SLC mn0	0	1	DLSm n1	DLS mn0

PTC mn1	PTC mn0	Setting of parity bit in UART mode	
		Transmission	Reception
0	0	Does not output the parity bit.	Receives without parity
0	1	Outputs 0 parity Note 2.	No parity judgment
1	0	Outputs even parity.	Judged as even parity.
1	1	Outputs odd parity.	Judges as odd parity.
Be sure to set PTCmn1, PTCmn0 = 0, 0 in the simplified SPI (CSI) mode and simplified I^{2}C mode.			

DIR mn	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.
Be sure to clear DIRmn = 0 in the simplified I^{2}C mode.	

SLCmn1 Note 1	SLC mn0	Setting of stop bit in UART mode
0	0	No stop bit
0	1	Stop bit length = 1 bit
1	0	Stop bit length = 2 bits (mn = 00 only)
1	1	Setting prohibited
When the transfer end interrupt is selected, the interrupt is generated when all stop bits have been completely transferred. Set 1 bit (SLCmn1, SLCmn0 = 0, 1) during UART reception and in the simplified I^{2}C mode. Set no stop bit (SLCmn1, SLCmn0 = 0, 0) in the simplified SPI (CSI) mode. Set 1 bit (SLCmn1, SLCmn0 = 0, 1) or 2 bits (SLCmn1, SLCmn0 = 1, 0) during UART transmission.		

DLSmn1	DLS mn0	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
Be sure to set DLSmn1, DLSmn0 = 1, 1 in the simplified I^{2}C mode.		

Note 1. The SCR00 register only.
Note 2. 0 is always added regardless of the data contents.

Caution Be sure to clear bits 3, 6, and 11 to 0 (Also clear bit 5 of the SCR01 register to 0). Be sure to set bit 2 to 1.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01)

17.3.6 Serial data register mn (SDRmn)

The SDRmn register is the transmit/receive data register (16 bits) of channel n.
Bits 8 to 0 (lower 9 bits) of SDR00 and SDR01 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 operation clock (f_{MCK}).
If the CCSmn bit of serial mode register mn (SMRmn) is cleared to 0, the clock set by dividing the operating clock by the higher 7 bits of the SDRmn register is used as the transfer clock.
If the CCSmn bit of serial mode register mn (SMRmn) is set to 1, set bits 15 to 9 (upper 7 bits) of SDR00 and SDR01 to 0000000B. The input clock f_{SCK} (slave transfer in simplified SPI (CSI) mode) from the SCKp pin is used as the transfer clock.
The lower 8/9 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 8/9 bits, and during transmission, the data to be transmitted to the shift register is set to the lower 8/9 bits.
The SDRmn register can be read or written in 16-bit units.
However, the higher 7 bits can be written or read only when the operation is stopped (SEmn = 0). During operation (SEmn = 1), a value is written only to the lower 8/9 bits of the SDRmn register. When the SDRmn register is read during operation, the higher 7 bits are always read as 0.
Reset signal generation clears the SDRmn register to 0000H.

Figure 17 - 10 Format of Serial data register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01) After reset: 0000H R/W

FFF11H (SDR00) FFF10H (SDR00)

l	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn																

SDRmn[15:9]							Transfer clock set by dividing the operating clock
0	0	0	0	0	0	0	$f_{MCK}/2$
0	0	0	0	0	0	1	$f_{MCK}/4$
0	0	0	0	0	1	0	$f_{MCK}/6$
0	0	0	0	0	1	1	$f_{MCK}/8$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	1	1	1	0	$f_{MCK}/254$
1	1	1	1	1	1	1	$f_{MCK}/256$

Caution 1. Setting SDRmn[15:9] = (0000000B, 0000001B) is prohibited when UART is used.

Caution 2. Setting SDRmn[15:9] = 0000000B is prohibited when simplified I²C is used. Set SDRmn[15:9] to 0000001B or greater.

Caution 3. Using an 8-bit memory manipulation instruction to write to the SDRmn[7:0] bits while operation is stopped (SEmn = 0) is prohibited (doing so clears the SDRmn[15:9] bits to 0).

Remark 1. For the function of the lower 8/9 bits of the SDRmn register, see **17.2 Configuration of Serial Array Unit**.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1)

17.3.7 Serial flag clear trigger register mn (SIRmn)

The SIRmn register is a trigger register that is used to clear each error flag of channel n.
When each bit (FECTmn, PECTmn, OVCTmn) of this register is set to 1, the corresponding bit (FEFmn, PEFmn, OVFmn) of serial status register mn is cleared to 0. Because the SIRmn register is a trigger register, it is cleared immediately when the corresponding bit of the SSRmn register is cleared.
The SIRmn register can be set by a 16-bit memory manipulation instruction.
The lower 8 bits of the SIRmn register can be set with an 8-bit memory manipulation instruction with SIRmnL.
Reset signal generation clears the SIRmn register to 0000H.

Figure 17 - 11 Format of Serial flag clear trigger register mn (SIRmn)

Address: F0108H, F0109H (SIR00) to F010AH, F010BH (SIR01) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	FECTmn Note	PECTmn	OVCTmn

FECTmn Note	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 The SIR01 register only.

Caution Be sure to clear bits 15 to 3 (or bits 15 to 2 for the SIR00 register) to 0.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1)
Remark 2. When the SIRmn register is read, 0000H is always read.

17.3.8 Serial status register mn (SSRmn)

The SSRmn register is a register that indicates the communication status and error occurrence status of channel n. The errors indicated by this register are a framing error, parity error, and overrun error.
The SSRmn register can be read by a 16-bit memory manipulation instruction.
The lower 8 bits of the SSRmn register can be set with an 8-bit memory manipulation instruction with SSRmnL.
Reset signal generation clears the SSRmn register to 0000H.

Figure 17 - 12 Format of Serial status register mn (SSRmn) (1/2)

Address: F0100H, F0101H (SSR00) to F0102H, F0103H (SSR01) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn Note	PEF mn	OVF mn

TSF mn	Communication status indication flag of channel n
0	Communication is stopped or suspended.
1	Communication is in progress.

<Clear conditions>
- 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.

BFF mn	Buffer register status indication flag of channel n
0	Valid data is not stored in the SDRmn register.
1	Valid data is stored in the SDRmn register.

<Clear conditions>
- Transferring transmit data from the SDRmn register to the shift register ends during transmission.
- Reading receive data from the SDRmn register ends during reception.
- The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is enabled).

<Set conditions>
- Transmit data is written to the SDRmn register while the TXEmn bit of the SCRmn register is set to 1 (transmission or transmission and reception mode in each communication mode).
- Receive data is stored in the SDRmn register while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode).
- A reception error occurs.

Note The SSR01 register only.

Caution **When simplified SPI (CSI) reception proceeds in the SNOOZE mode (SWCm = 1), the BFFmn flag will not change.**

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 17 - 13 Format of Serial status register mn (SSRmn) (2/2)

Address: F0100H, F0101H (SSR00) to F0102H, F0103H (SSR01) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSFmn	BFFmn	0	0	FEFmn Note	PEFmn	OVFmn

FEFmn Note	Framing error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception).

<Clear condition>
- 1 is written to the FECTmn bit of the SIRmn register.

<Set condition>
- A stop bit is not detected when UART reception ends.

PEFmn	Parity error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception) or ACK is not detected (during I^2C transmission).

<Clear condition>
- 1 is written to the PECTmn bit of the SIRmn register.

<Set 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 channel at the ACK reception timing during I^2C transmission (ACK is not detected).

OVFmn	Overrun error detection flag of channel n
0	No error occurs.
1	An error occurs

<Clear condition>
- 1 is written to the OVCTmn bit of the SIRmn register.

<Set 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 and reception mode in each communication mode).
- Transmit data is not ready for slave transmission or transmission and reception in simplified SPI (CSI) mode.

Note The SSR01 register only.

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 simplified SPI (CSI) reception proceeds in the SNOOZE mode (SWCm = 1), the OVFmn flag will not change.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

17.3.9 Serial channel start register m (SSm)

The SSm register is a trigger register that is used to enable starting communication/count by each channel.

When 1 is written a bit of this register (SSmn), the corresponding bit (SEmn) of serial channel enable status register m (SEm) is set to 1 (Operation is enabled). Because the SSmn bit is a trigger bit, it is cleared immediately when SEmn = 1.

The SSm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSm register can be set with an 1-bit or 8-bit memory manipulation instruction with SSmL.

Reset signal generation clears the SSm register to 0000H.

Figure 17 - 14 Format of Serial channel start register m (SSm)

Address: F0122H, F0123H (SS0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS01	SS00

SSmn	Operation start trigger of channel n
0	No trigger operation
1	Sets the SEmn bit to 1 and enters the communication wait status Note.

Note If set the SSmn = 1 to during a communication operation, will wait status to stop the communication. At this time, holding status value of control register and shift register, SCKmn and SOmn pins, and FEFmn, PEFmn, OVFmn flags.

Caution 1. Be sure to clear bits 15 to 2 of the SS0 register to 0.

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 4 or more fMCK clocks have elapsed.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1)

Remark 2. When the SSm register is read, 0000H is always read.

17.3.10 Serial channel stop register m (STm)

The STm register is a trigger register that is used to enable stopping communication/count by each channel.
When 1 is written a bit of this register (STmn), the corresponding bit (SEmn) of serial channel enable status register m (SEm) is cleared to 0 (operation is stopped). Because the STmn bit is a trigger bit, it is cleared immediately when SEmn = 0.
The STm register can set written by a 16-bit memory manipulation instruction.
The lower 8 bits of the STm register can be set with a 1-bit or 8-bit memory manipulation instruction with STmL.
Reset signal generation clears the STm register to 0000H.

Figure 17 - 15 Format of Serial channel stop register m (STm)

Address: F0124H, F0125H (ST0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ST01	ST00

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 Holding status value of the control register and shift register, the SCKmn and SOmn pins, and FEFmn, PEFmn, OVFmn flags.

Caution Be sure to clear bits 15 to 2 of the ST0 register to 0.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1)
Remark 2. When the STm register is read, 0000H is always read.

17.3.11 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 a bit of serial channel start register m (SSm), the corresponding bit of this register is set to 1.

When 1 is written a bit of serial channel stop register m (STm), the corresponding bit is cleared to 0.

Channel n that is enabled to operate cannot rewrite by software the value of the CKOmn bit (serial clock output of channel n) of serial output register m (SOm) to be described below, and a value reflected by a communication operation is output from the serial clock pin.

Channel n that stops operation can set the value of the CKOmn bit of the SOm register by software and output its value from the serial clock pin. In this way, any waveform, such as that of a start condition/stop condition, can be created by software.

The SEm register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the SEm register can be set with a 1-bit or 8-bit memory manipulation instruction with SEmL.

Reset signal generation clears the SEm register to 0000H.

Figure 17 - 16 Format of Serial channel enable status register m (SEm)

Address: F0120H, F0121H (SE0) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SE01	SE00

SEmn	Indication of operation enable/stop status of channel n
0	Operation stops
1	Operation is enabled.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

17.3.12 Serial output enable register m (SOEm)

The SOEm register is a register that is used to enable or stop output of the serial communication operation of each channel.

Channel n that enables serial output cannot rewrite by software the value of the SOmn bit of serial output register m (SOm) to be described below, and a value reflected by a communication operation is output from the serial data output pin.

For channel n, whose serial output is stopped, the SOmn bit value of the SOm register can be set by software, and that value can be output from the serial data output pin. In this way, any waveform of the start condition and stop condition can be created by software.

The SOEm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SOEm register can be set with a 1-bit or 8-bit memory manipulation instruction with SOEmL.

Reset signal generation clears the SOEm register to 0000H.

Figure 17 - 17 Format of Serial output enable register m (SOEm)

Address: F012AH, F012BH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE01	SOE00

SOEmn	Serial output enable/stop of channel n
0	Stops output by serial communication operation.
1	Enables output by serial communication operation.

Caution Be sure to clear bits 15 to 2 of the SOE0 register, bits 15 to 2 to 0.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

17.3.13 Serial output register m (SOm)

The SOm register is a buffer register for serial output of each channel.
The value of the SOmn bit of this register is output from the serial data output pin of channel n.
The value of the CKOmn bit of this register is output from the serial clock output pin of channel n.
The SOmn bit of this register can be rewritten by software only when serial output is disabled (SOEmn = 0). When serial output is enabled (SOEmn = 1), rewriting by software is ignored, and the value of the register can be changed only by a serial communication operation.
The CKOmn bit of this register can be rewritten by software only when the channel operation is stopped (SEmn = 0). While channel operation is enabled (SEmn = 1), rewriting by software is ignored, and the value of the CKOmn bit can be changed only by a serial communication operation.
To use a pin for the serial interface as a port function pin other than a serial interface function pin, set the corresponding the CKOmn and SOmn bits to 1.
The SOm register can be set by a 16-bit memory manipulation instruction.
Reset signal generation clears the SOm register to 0303H.

Figure 17 - 18 Format of Serial output register m (SOm)

Address: F0128H, F0129H After reset: 0303H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	0	0	CKO01	CKO00	0	0	0	0	0	0	SO01	SO00

CKOmn	Serial clock output of channel n
0	Serial clock output value is 0.
1	Serial clock output value is 1.

SOmn	Serial data output of channel n
0	Serial data output value is 0.
1	Serial data output value is 1.

Caution Be sure to clear bits 15 to 10 and 7 to 2 of the SO0 register to 0.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1)

17.3.14 Serial output level register m (SOLm)

The SOLm register is a register that is used to set inversion of the data output level of each channel.
This register can be set only in the UART mode. Be sure to set 0 for corresponding bit in the simplified SPI (CSI) mode and simplified I²C 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 lower 8 bits of the SOLm register can be set with an 8-bit memory manipulation instruction with SOLmL.
Reset signal generation clears the SOLm register to 0000H.

Figure 17 - 19 Format of Serial output level register m (SOLm)

Address: F0134H, F0135H (SOL0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL00

SOLmn	Selects inversion of the level of the transmit data of channel n in UART mode
0	Communication data is output as is.
1	Communication data is inverted and output.

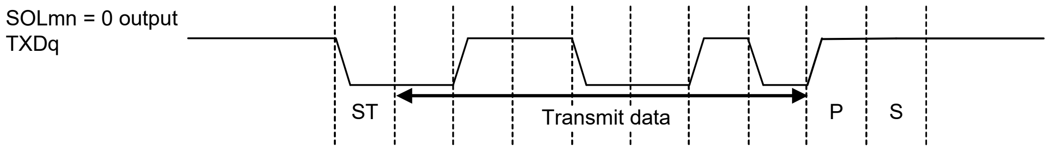
Caution Be sure to clear bits 15 to 1 of the SOL0 register to 0.

Remark m: Unit number (m = 0), n: Channel number (n = 0)

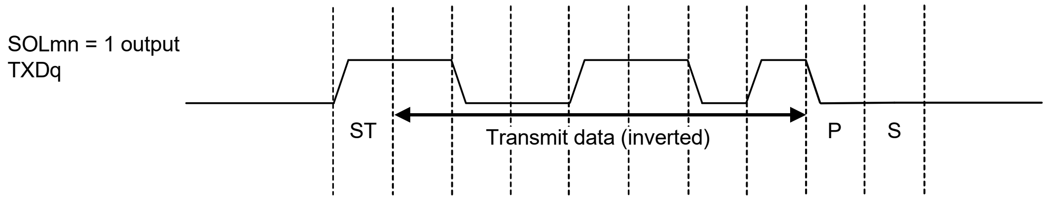
Figure 17 - 20 shows examples in which the level of transmit data is reversed during UART transmission.

Figure 17 - 20 Examples of Reverse Transmit Data

(a) Non-reverse Output (SOLmn = 0)



(b) Reverse Output (SOLmn = 1)



Remark m: Unit number (m = 0), n: Channel number (n = 0)

17.3.15 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 SSCm register can be set by a 16-bit memory manipulation instruction.
The lower 8 bits of the SSCm register can be set with an 8-bit memory manipulation instruction with SSCmL.
Reset signal generation clears the SSCm register to 0000H.

Caution The maximum transfer rate in the SNOOZE mode is as follows.
• When using CSI00: Up to 1 Mbps
• When using UART0: 4800 bps only

Figure 17 - 21 Format of Serial standby control register m (SSCm)

Address: F0138H, F0139H (SSC0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSCm	0	0	0	0	0	0	0	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).
1	Disable the generation of error interrupts (INTSRE0).
• 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, clear the SSECm bit to 0. • Setting SSECm, SWCm = 1, 0 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 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.	

Caution Setting SSECm, SWCm = 1, 0 is prohibited.

Figure 17 - 22 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.

17.3.16 Input switch control register (ISC)

The SSIE0 bit controls the $\overline{\text{SSI00}}$ pin input of channel 0 during CSI00 communication and in slave mode.

While a high level is being input to the $\overline{\text{SSI00}}$ pin, no transmission/reception operation is performed even if a serial clock is input. While a low level is being input to the $\overline{\text{SSI00}}$ pin, a transmission/reception operation is performed according to each mode setting if a serial clock is input.

The ISC register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the ISC register to 00H.

Figure 17 - 23 Format of Input switch control register (ISC)

Address: F0073H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	SSIE00	0	0	0	0	0	0	0

SSIE00	Channel 0 $\overline{\text{SSI00}}$ input setting in CSI00 communication and slave mode
0	Disables $\overline{\text{SSI00}}$ pin input.
1	Enables $\overline{\text{SSI00}}$ pin input.

Caution Be sure to clear bits 6 to 0 to 0.

17.3.17 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 clearing the corresponding bit of this register to 0.

Enable the noise filter of the pin used for UART communication, by setting the corresponding bit of this register to 1.

When the noise filter is enabled, after synchronization is performed with the operation clock (f_{MCK}) of the target channel, 2-clock match detection is performed. When the noise filter is OFF, only synchronization is performed with the Operation clock of target channel (f_{MCK}).

The NFEN0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the NFEN0 register to 00H.

Figure 17 - 24 Format of Noise filter enable register 0 (NFEN0)

Address: F0070H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN0	0	0	0	0	0	0	0	SNFEN00

SNFEN00	Use of noise filter of RxD0 pin
0	Noise filter OFF
1	Noise filter ON
Set the SNFEN00 bit to 1 to use the RxD0 pin. Clear the SNFEN00 bit to 0 to use the other than RxD0 pin.	

Caution Be sure to clear bits 7 to 1 to 0.

17.3.18 Registers controlling port functions of serial input/output pins

Using the serial array unit requires setting of the registers that control the port functions multiplexed on the target channel (port mode register (PMxx), port register (Pxx), port input mode register (PIMxx), and port output mode register (POMxx)).

For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.2 Port registers (Pxx)**, **4.3.4 Port input mode registers (PIMxx)**, and **4.3.5 Port output mode registers (POMxx)**.

Specifically, using a port pin with a multiplexed serial data or serial clock output function (e.g. P54/SO00/TxD0/INTP1/TOOLTxD) for serial data or serial clock output, requires setting the corresponding bits in the port mode register (PMxx) to 0, and the corresponding bit in the port register (Pxx) to 1.

When using the port pin in N-ch open-drain output (V_{DD} tolerance) mode, set the corresponding bit in the port output mode register (POMxx) to 1. When connecting an external device operating on a different potential (1.8 V, 2.5 V, or 3 V), see **4.4.4 Handling different potential (1.8 V, 2.5 V) by using I/O buffers**.

Example When P54/SO00/TxD0/INTP1/TOOLTxD is to be used for serial data output
Set the PM54 bit of port mode register 5 to 0.
Set the P54 bit of port register 5 to 1.

Specifically, using a port pin with a multiplexed serial data or serial clock input function (e.g. P55/SI00/RxD0/SDA00/INTP2/TOOLRxD) for serial data or serial clock input, requires setting the corresponding bit in the port mode register (PMxx) to 1. In this case, the corresponding bit in the port register (Pxx) can be set to 0 or 1.

When the TTL input buffer is selected, set the corresponding bit in the port input mode register (PIMxx) to 1. When connecting an external device operating on a different potential (1.8 V, 2.5 V or 3 V), see **4.4.4 Handling different potential (1.8 V, 2.5 V) by using I/O buffers**.

Example When P55/SI00/RxD0/SDA00/INTP2/TOOLRxD is to be used for serial data input
Set the PM55 bit of port mode register 5 to 1.
Set the P55 bit of port register 5 to 0 or 1.

17.4 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.

17.4.1 Stopping the operation by units

The stopping of the operation by units is set by using peripheral enable register 0 (PER0).
The PER0 register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.
To stop the operation of serial array unit 0, set bit 2 (SAU0EN) to 0.

Figure 17 - 25 Peripheral Enable Register 0 (PER0) Setting When Stopping the Operation by Units

(a) Peripheral enable register 0 (PER0)... Set only the bit of SAUm to be stopped to 0.

	7	6	5	4	3	2	1	0
PER0	RTCWEN ×	0	ADCEN ×	0	0	SAU0EN 0/1	0	TAU0EN ×

Control of SAUm input clock
0: Stops supply of input clock
1: Supplies input clock

Caution 1. If SAUmEN = 0, writing to a control register of serial array unit m is ignored.
Note that this does not apply to the following registers.
• Input switch control register (ISC)
• Noise filter enable register 0 (NFEN0)
• Port input mode register 5 (PIM5)
• Port output mode register 5 (POM5)
• Port mode registers 5, 6 (PM5, PIM6)
• Port registers 5, 6 (P5, P6)

Caution 2. Be sure to clear the following bits to 0.
Bits 1, 3, 5, 4, and 6

Remark ×: Bits not used with serial array units (depending on the settings of other peripheral functions)
0/1: Set to 0 or 1 depending on the usage of the user

17.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

Figure 17 - 26 Each Register Setting When Stopping the Operation by Channels

(a) Serial channel stop register m (STm)... This register is a trigger register that is used to enable stopping communication/count by each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	STm1 0/1	STm0 0/1

1: Clears the SEmn bit to 0 and stops the communication operation

* Because the STmn bit is a trigger bit, it is cleared immediately when SEmn = 0.

(b) Serial Channel Enable Status Register m (SEm)... This register indicates whether data transmission/ reception operation of each channel is enabled or stopped.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SEm1 0/1	SEm0 0/1

0: Operation stops

* The SEm register is a read-only status register, whose operation is stopped by using the STm register.
With a channel whose operation is stopped, the value of the CKOmn bit of the SOm register can be set by software.

(c) Serial output enable register m (SOEm)... This register is a register that is used to enable or stop output of the serial communication operation of each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 0/1	SOEm0 0/1

0: Stops output by serial communication operation

* For channel n, whose serial output is stopped, the SOmn bit value of the SOm register can be set by software.

(d) Serial output register m (SOm)... This register is a buffer register for serial output of each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 0/1	CKOm0 0/1	0	0	0	0	0	0	SOm1 0/1	SOm0 0/1

1: Serial clock output value is "1"

1: Serial data output value is "1"

* When using pins corresponding to each channel as port function pins, set the corresponding CKOmn, SOmn bits to "1".

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1)

Remark 2. ▇: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

17.5 Operation of Simplified SPI (CSI00, CSI01) Communication

This is a clocked communication function that uses three lines: serial clock (SCK) and serial data (SI and SO) lines.

[Data transmission/reception]
- Data length of 7 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

CSI00 supports the SNOOZE mode. When SCK input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. Only following simplified SPI (CSI) can be specified.
- CSI00

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics. For details, see **CHAPTER 34 ELECTRICAL SPECIFICATIONS**.

The channels supporting simplified SPI (CSI00, CSI01) are channels 0 and 1 of SAU0.

• 20-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	—		—

• 24, 32-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01

• 30-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	—		—

• 48-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00 (supporting slave select input function)	UART0	IIC00
	1	CSI01		IIC01

Simplified SPI (CSI00, CSI01) performs the following seven types of communication operations.

• Master transmission (See **17.5.1**.)
• Master reception (See **17.5.2**.)
• Master transmission/reception (See **17.5.3**.)
• Slave transmission (See **17.5.4**.)
• Slave reception (See **17.5.5**.)
• Slave transmission/reception (See **17.5.6**.)
• SNOOZE mode function (CSI00 only) (See **17.5.7**.)

17.5.1 Master transmission

Master transmission is that the RL78 microcontroller outputs a transfer clock and transmits data to another device.

Simplified SPI	CSI00	CSI01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCK00, SO00	SCK01, SO01
Interrupt	INTCSI00	INTCSI01
	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}: System 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 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 in the electrical specifications (see **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

(1) Register setting

Figure 17 - 27 Example of Contents of Registers for Master Transmission of Simplified SPI (CSI00, CSI01)

(a) Serial mode register mn (SMRmn)

	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	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (fMCK) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **17.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting (Operation clock (fMCK) division setting)							0	Transmit data (Transmit data setting)							

$\underbrace{\text{bits 7 to 0}}_{\text{SIOp}}$

(d) Serial output register m (SOm)... Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 0/1	CKOm0 0/1	0	0	0	0	0	0	SOm1 0/1	SOm0 0/1

Communication starts when these bits are 1 if the clock phase is non-reversed (the CKPmn bit of the SCRmn = 0). If the clock phase is reversed (CKPmn = 1), communication starts when these bits are 0.

(e) Serial output enable register m (SOEm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 0/1	SOEm0 0/1

(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 0/1	SSm0 0/1

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Remark 2. ☐: Setting is fixed in the simplified SPI (CSI) master transmission mode,
▒: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 17 - 28 Initial Setting Procedure for Master Transmission

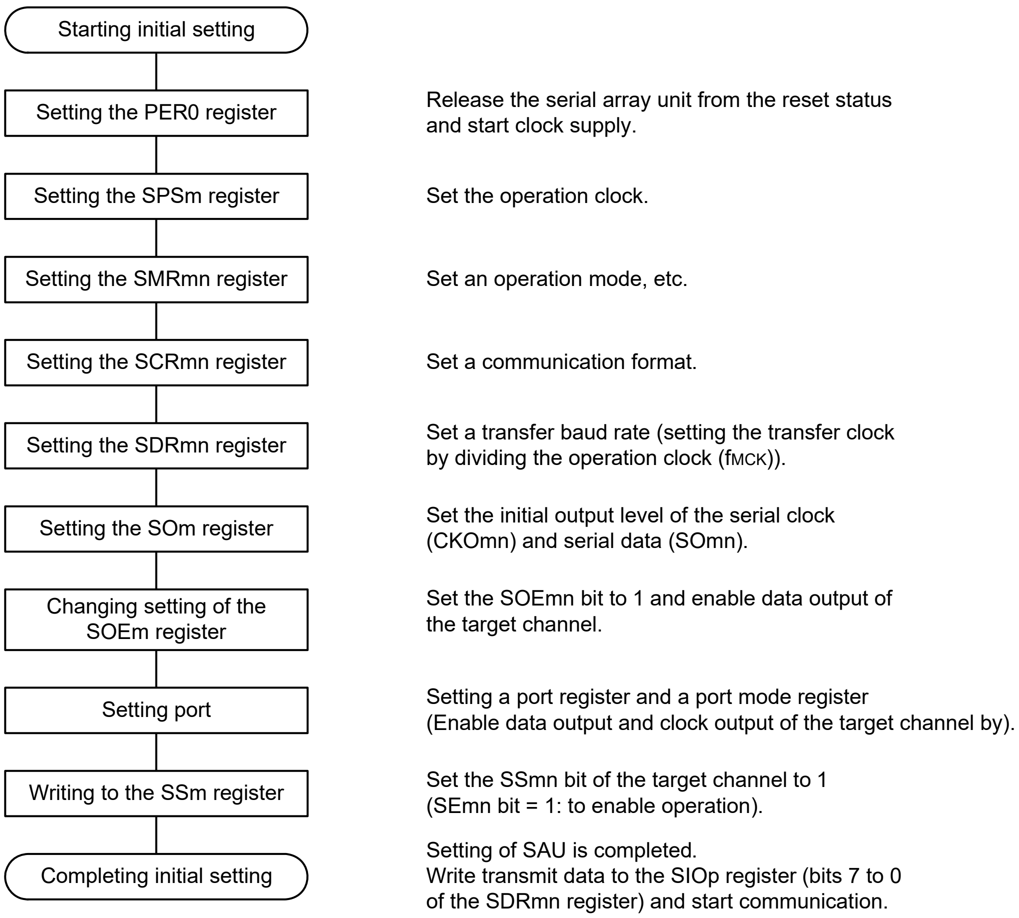


Figure 17 - 29 Procedure for Stopping Master Transmission

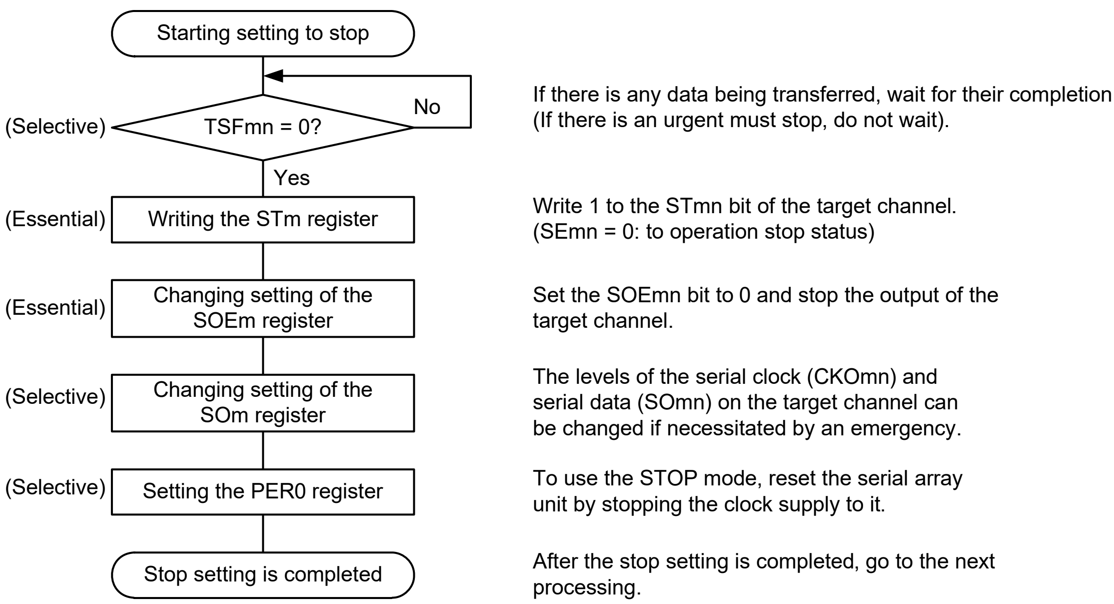
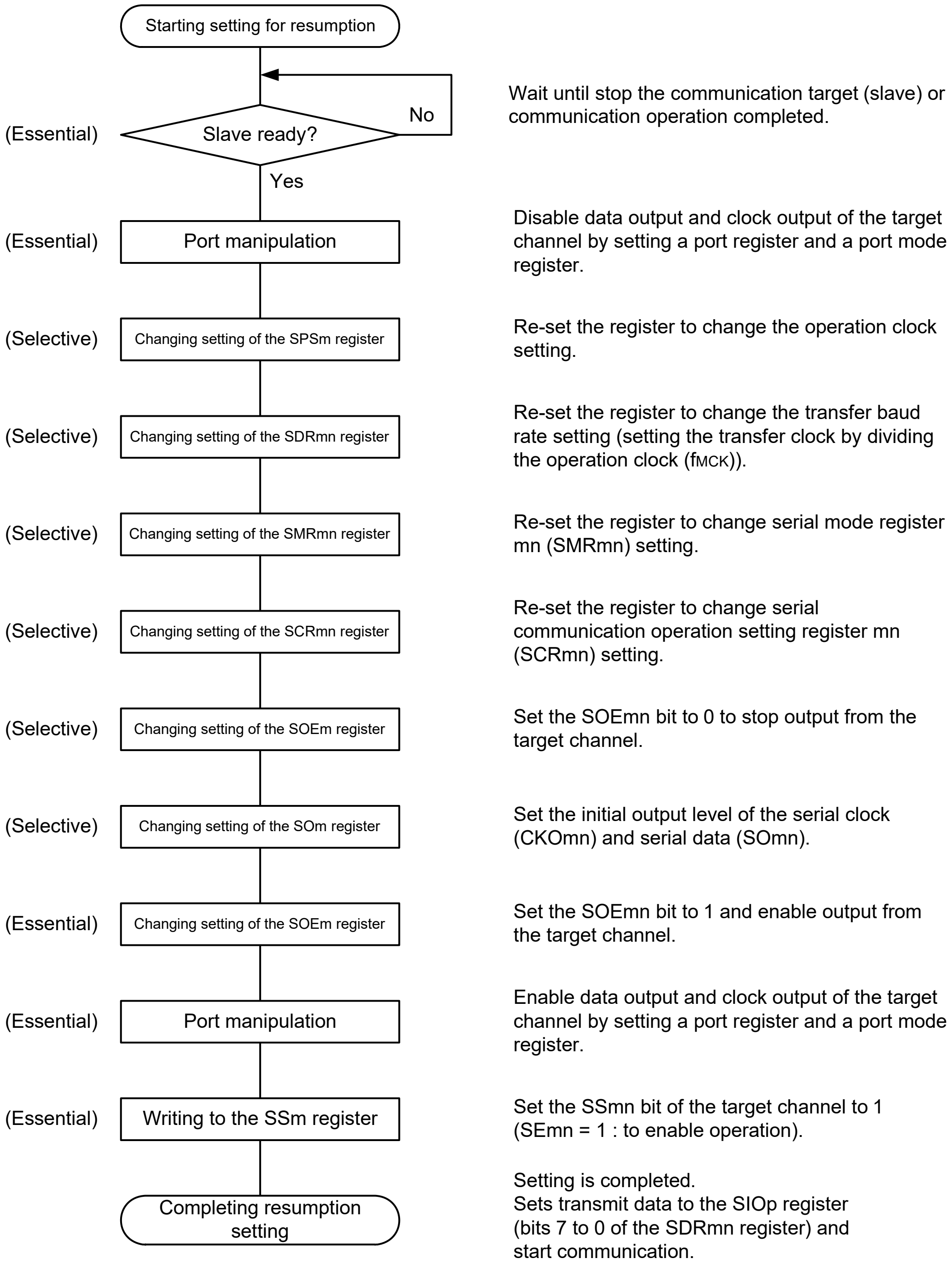


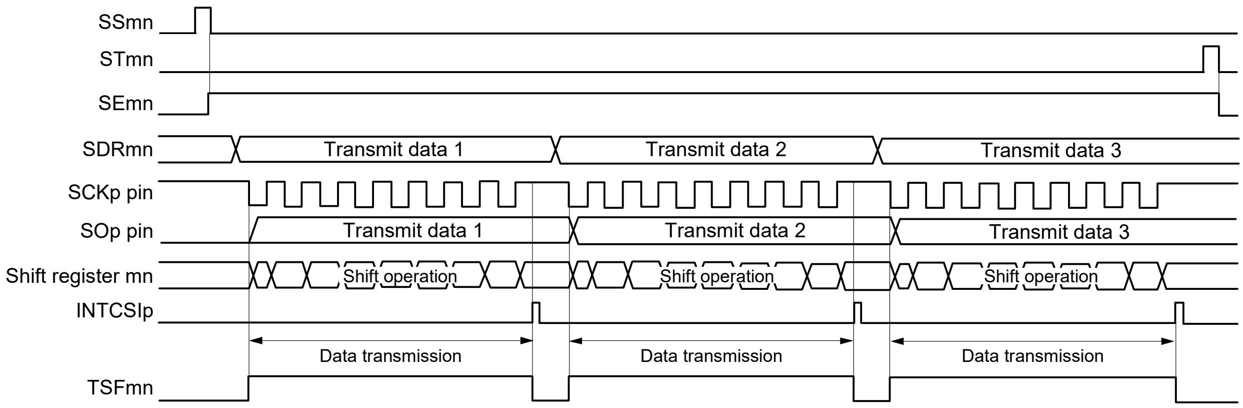
Figure 17 - 30 Procedure for Resuming Master Transmission



Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

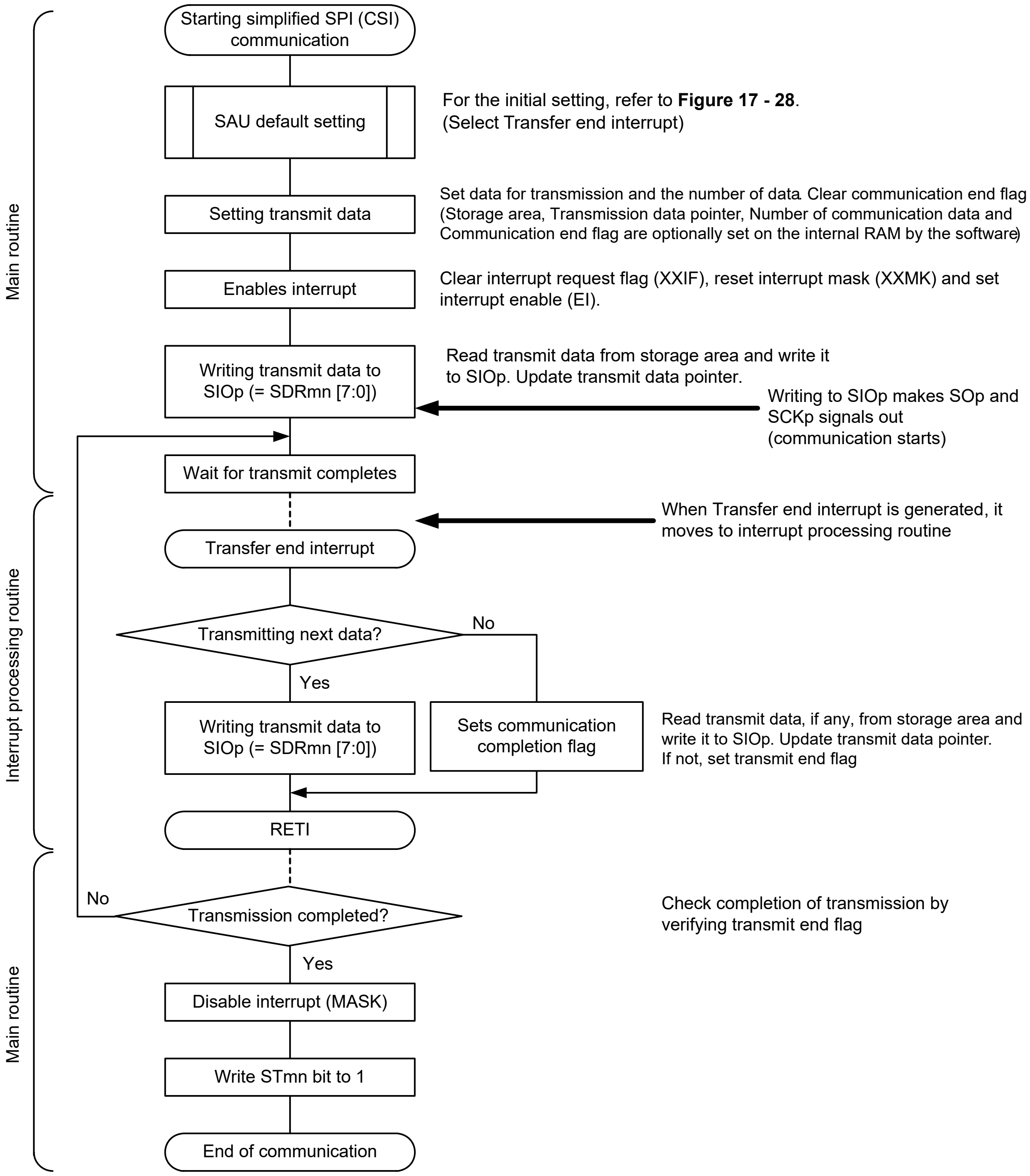
(3) Processing flow (in single-transmission mode)

Figure 17 - 31 Timing Chart of Master Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



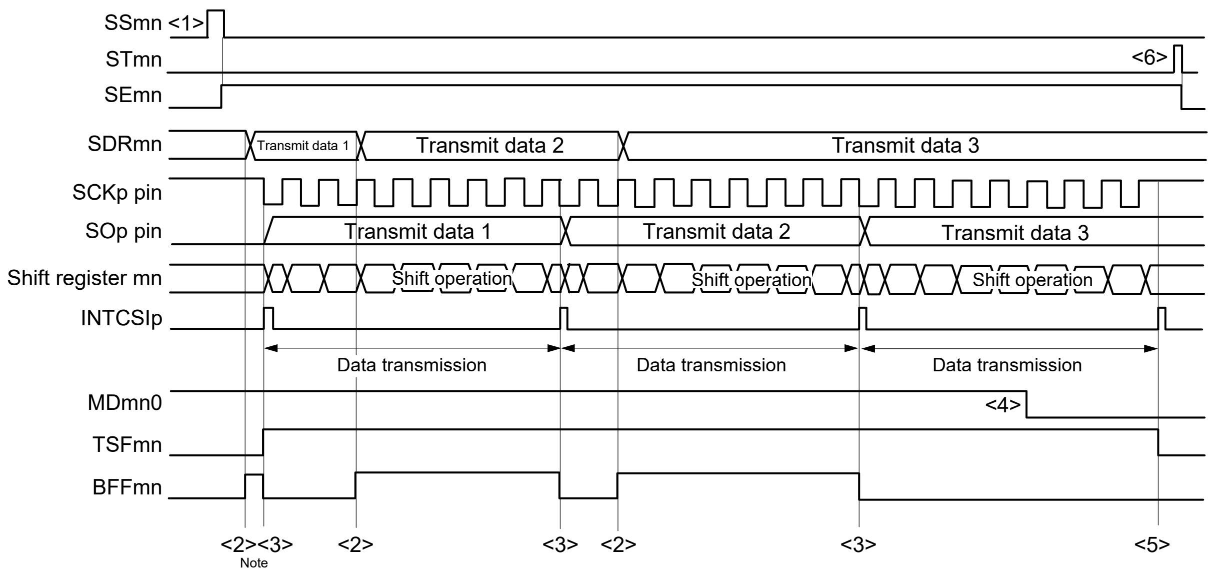
Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 17 - 32 Flowchart of Master Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 17 - 33 Timing Chart of Master Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)

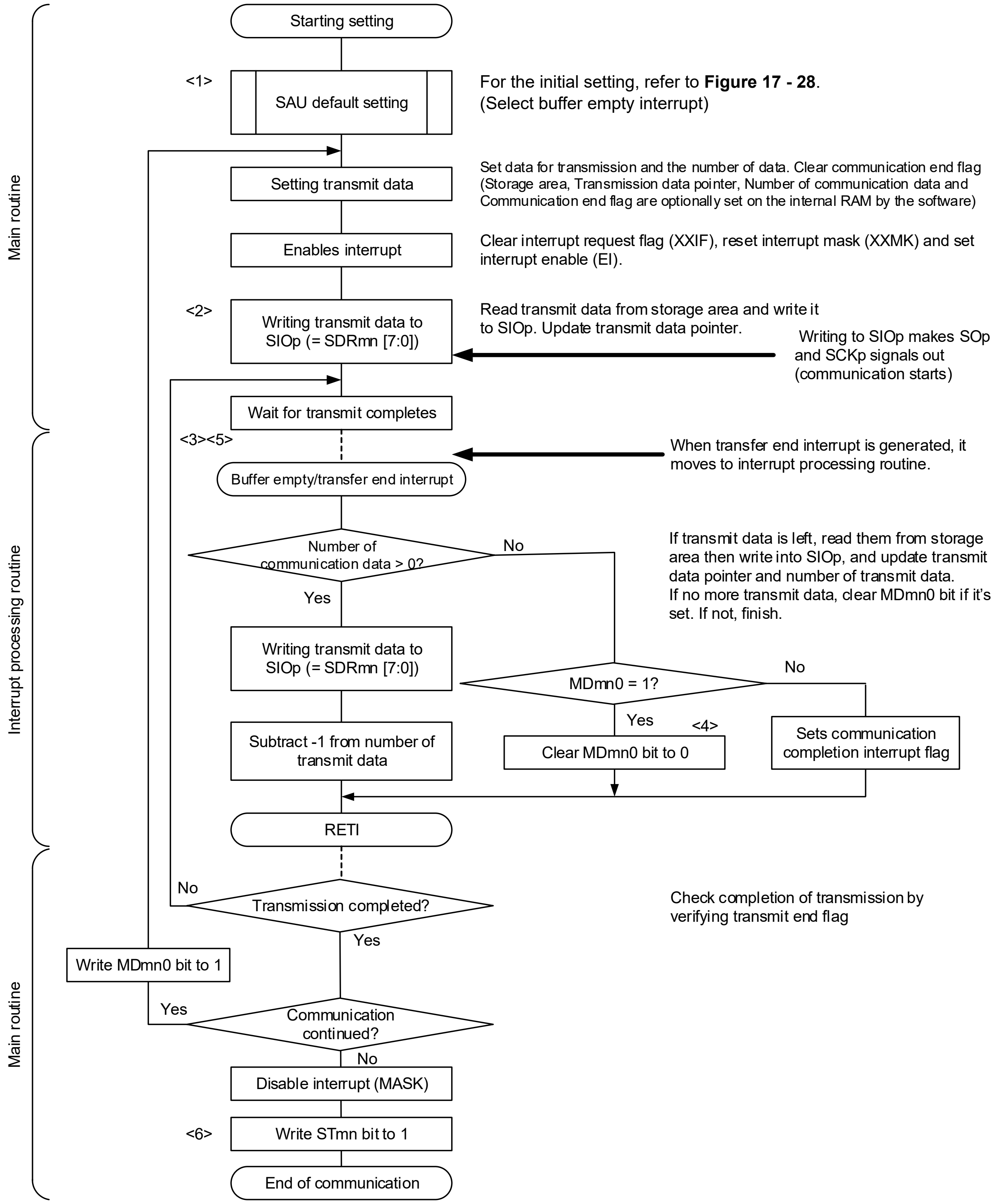


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 17 - 34 Flowchart of Master Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in Figure 17 - 33 Timing Chart of Master Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0).

17.5.2 Master reception

Master reception is that the RL78 microcontroller outputs a transfer clock and receives data from other device.

Simplified SPI	CSI00	CSI01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCK00, SI00	SCK01, SI01
Interrupt	INTCSI00	INTCSI01
	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}: System 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 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 in the electrical specifications (see **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

(1) Register setting

Figure 17 - 35 Example of Contents of Registers for Master Reception of Simplified SPI (CSI00, CSI01)

(a) Serial mode register mn (SMRmn)

	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	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **17.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting (Operation clock (f_{MCK}) division setting)							0	Receive data (Write FFH as dummy data.)							

SIOp (bits 7 to 0)

(d) Serial output register m (SOm)... Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 0/1	CKOm0 0/1	0	0	0	0	0	0	SOm1 ×	SOm0 ×

Communication starts when these bits are 1 if the clock phase is non-reversed (the CKPmn bit of the SCRmn = 0). If the clock phase is reversed (CKPmn = 1), communication starts when these bits are 0.

(e) Serial output enable register m (SOEm)... The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 ×	SOEm0 ×

(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 0/1	SSm0 0/1

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Remark 2. ■: Setting is fixed in the simplified SPI (CSI) master reception mode,
■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 17 - 36 Initial Setting Procedure for Master Reception

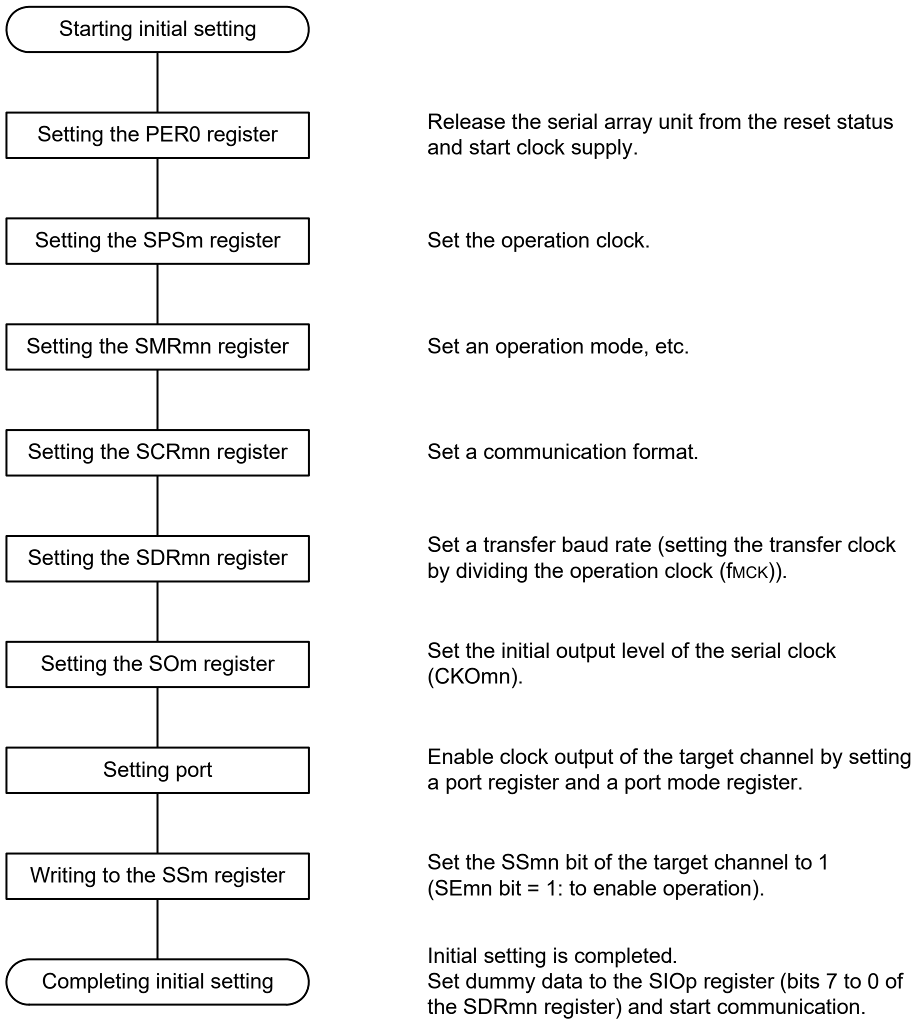


Figure 17 - 37 Procedure for Stopping Master Reception

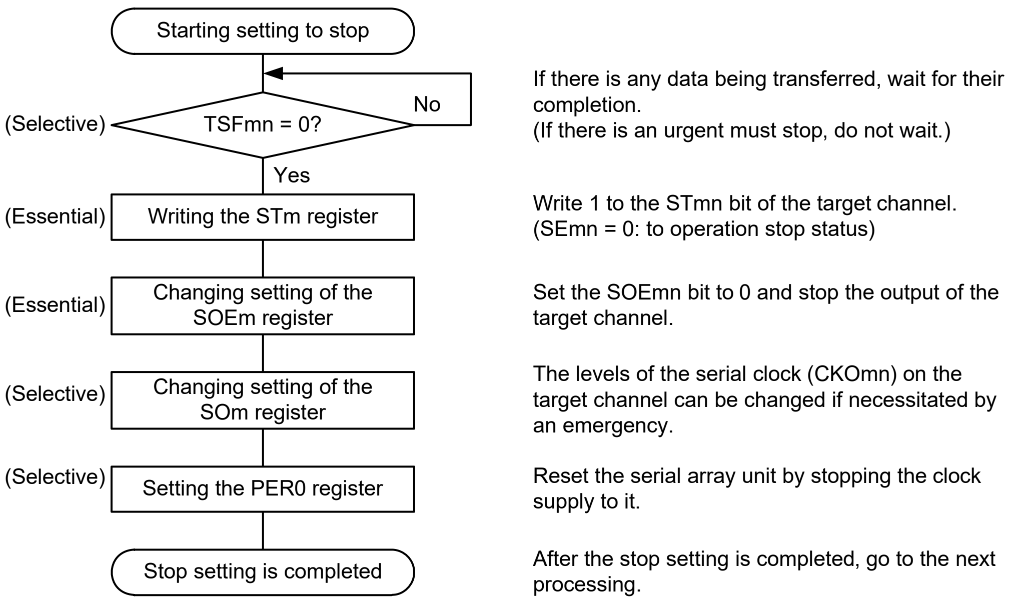
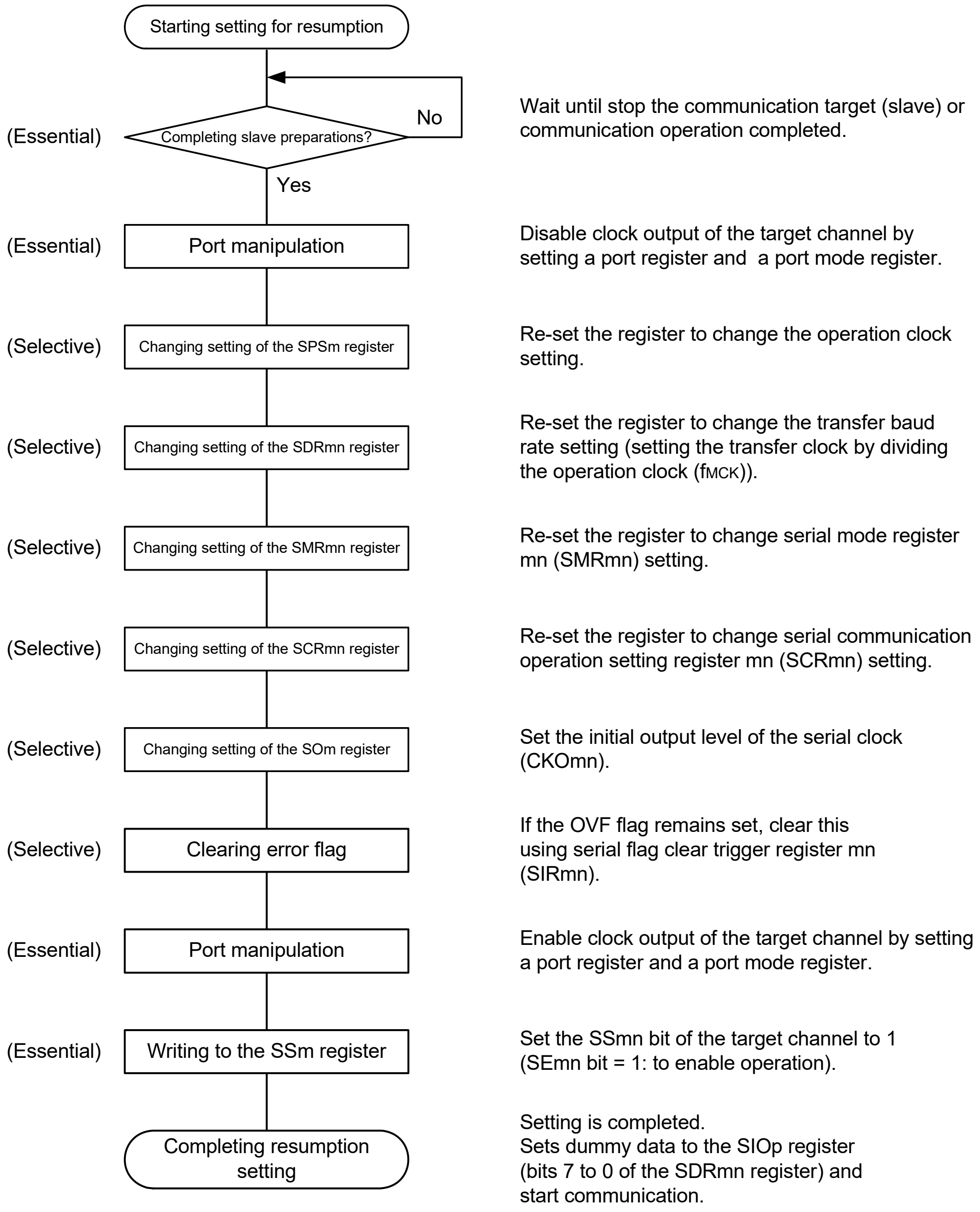


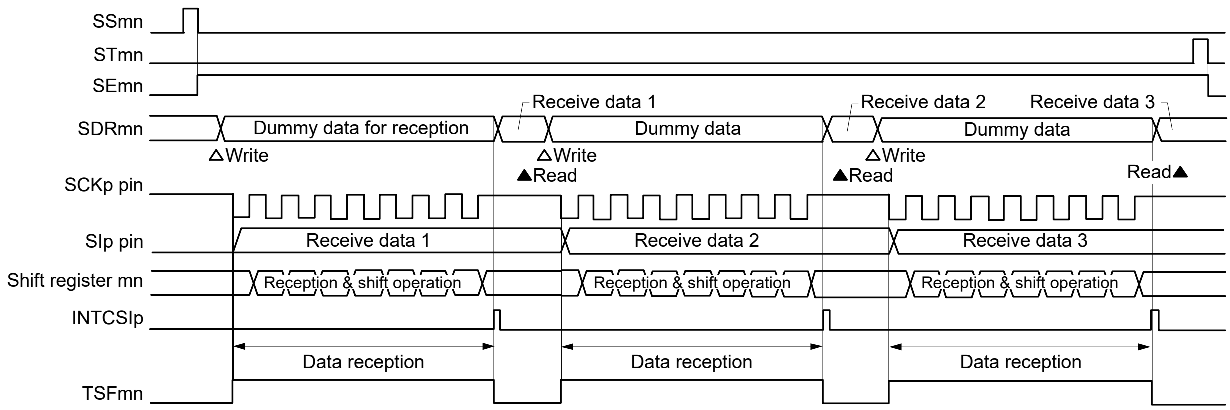
Figure 17 - 38 Procedure for Resuming Master Reception



Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

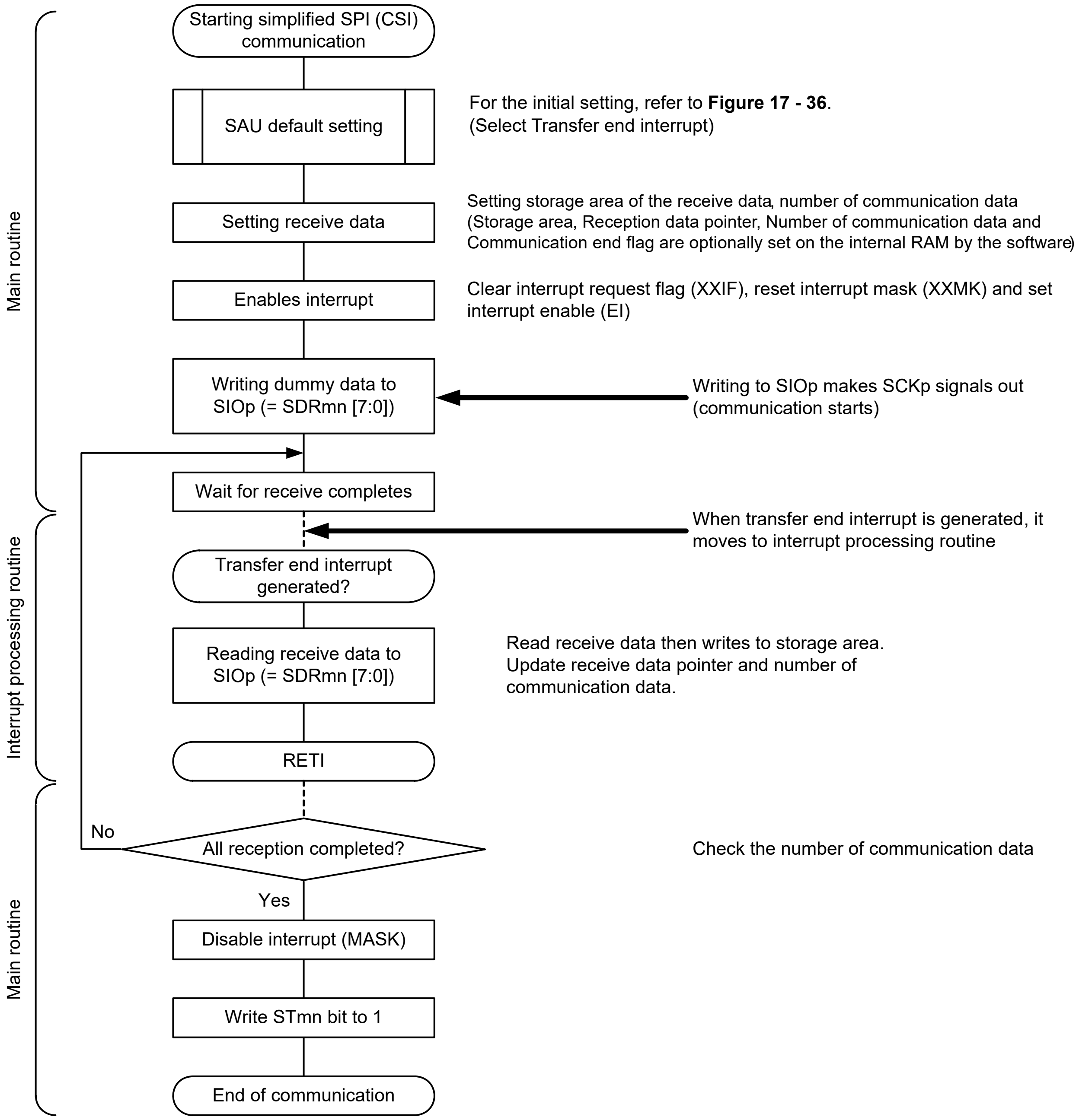
(3) Processing flow (in single-reception mode)

Figure 17 - 39 Timing Chart of Master Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



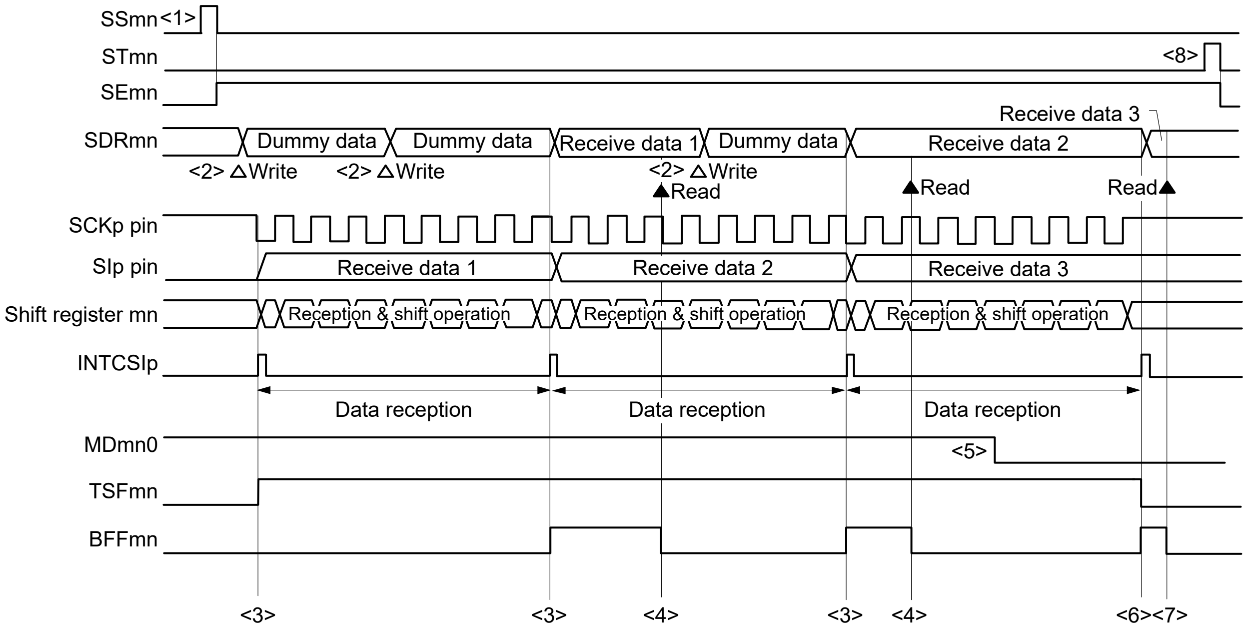
Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 17 - 40 Flowchart of Master Reception (in Single-Reception Mode)



(4) Processing flow (in continuous reception mode)

Figure 17 - 41 Timing Chart of Master Reception (in Continuous Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

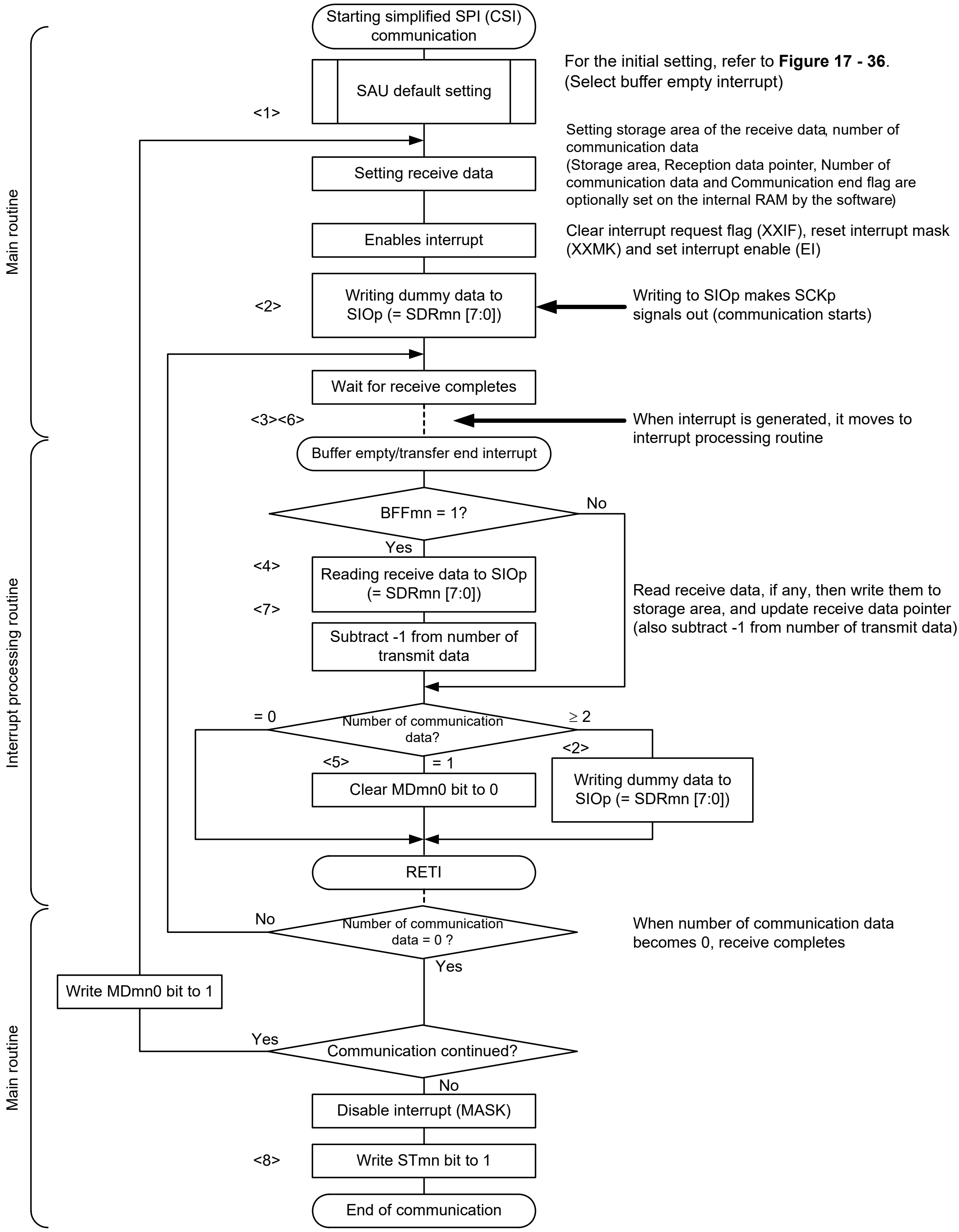


Caution **The MDmn0 bit can be rewritten even during operation.**
However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in Figure 17 - 42 Flowchart of Master Reception (in Continuous Reception Mode).

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 17 - 42 Flowchart of Master Reception (in Continuous Reception Mode)



Remark <1> to <8> in the figure correspond to <1> to <8> in Figure 17 - 41 Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

17.5.3 Master transmission/reception

Master transmission/reception is that the RL78 microcontroller outputs a transfer clock and transmits/receives data to/from another device.

Simplified SPI (CSI)	CSI00	CSI01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01
Interrupt	INTCSI00	INTCSI01
	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}: System clock frequency	
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data I/O starts at the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: 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 in the electrical specifications (see **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

(1) Register setting

Figure 17 - 43 Example of Contents of Registers for Master Transmission/Reception of Simplified SPI (CSI00, CSI01)

(a) Serial mode register mn (SMRmn)

	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	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (fMCK) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **17.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 18-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting (Operation clock (fMCK) division setting)							0	Transmit data setting/receive data register							

Bits 7 to 0: SIOp

(d) Serial output register m (SOm)... Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 0/1	CKOm0 0/1	0	0	0	0	0	0	SOm1 0/1	SOm0 0/1

Communication starts when these bits are 1 if the clock phase is non-reversed (the CKPmn bit of the SCRmn = 0). If the clock phase is reversed (CKPmn = 1), communication starts when these bits are 0.

(e) Serial output enable register m (SOEm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 0/1	SOEm0 0/1

(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 0/1	SSm0 0/1

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Remark 2. ■: Setting is fixed in the simplified SPI (CSI) master transmission/reception mode,
▒: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 17 - 44 Initial Setting Procedure for Master Transmission/Reception

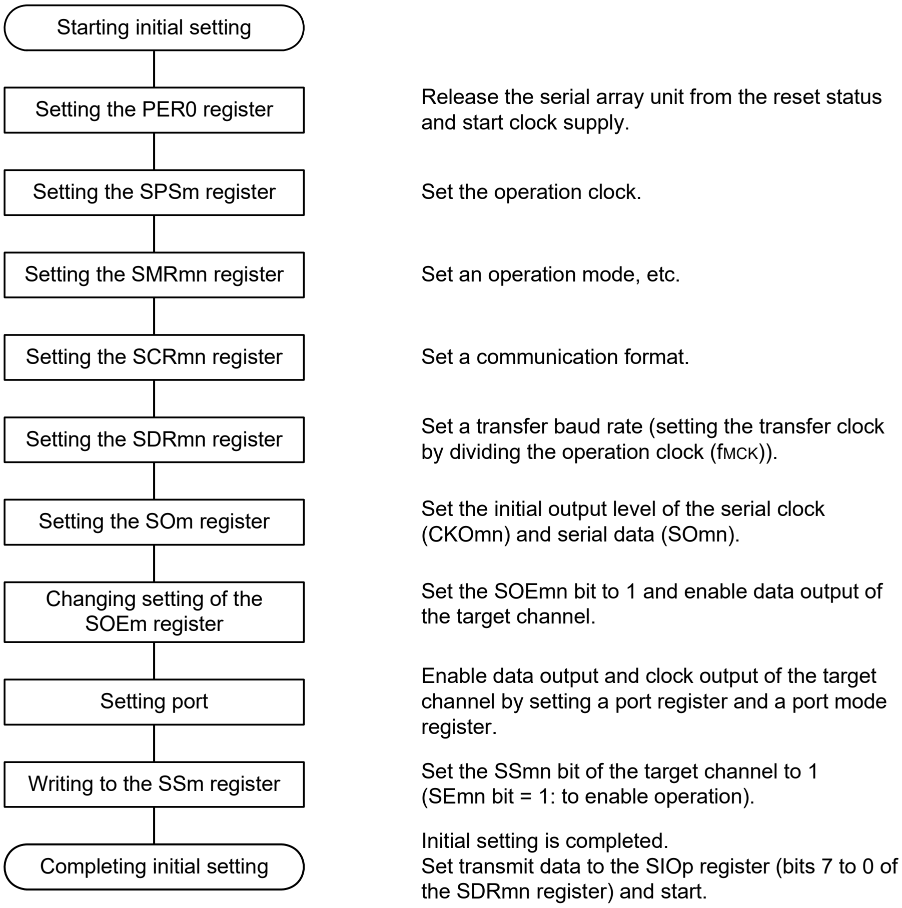


Figure 17 - 45 Procedure for Stopping Master Transmission/Reception

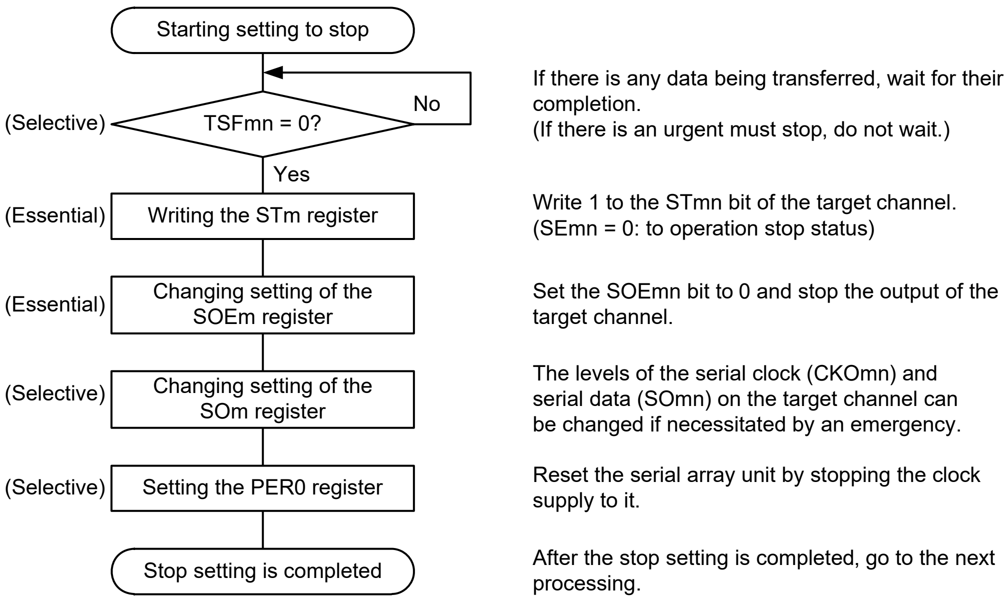
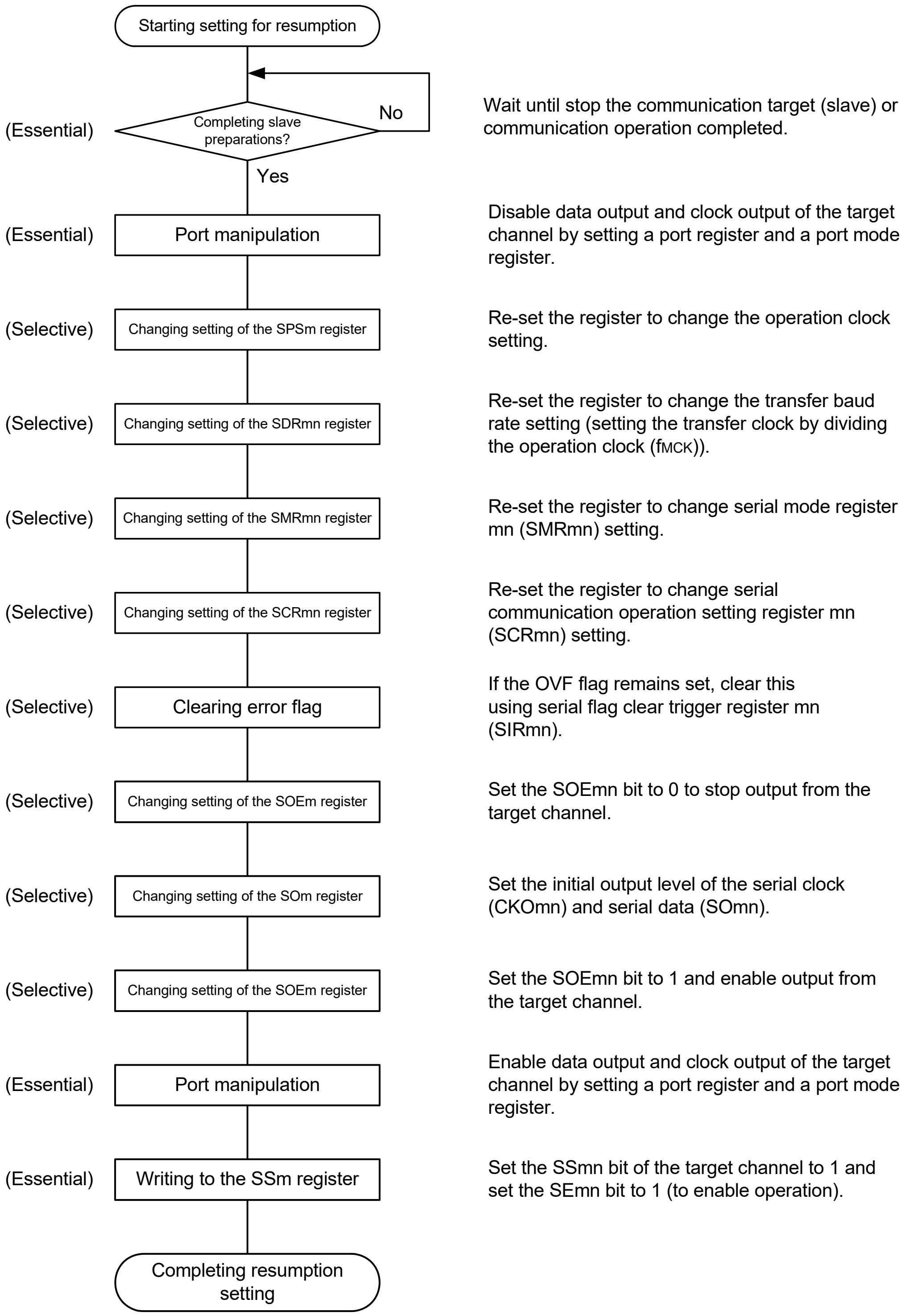
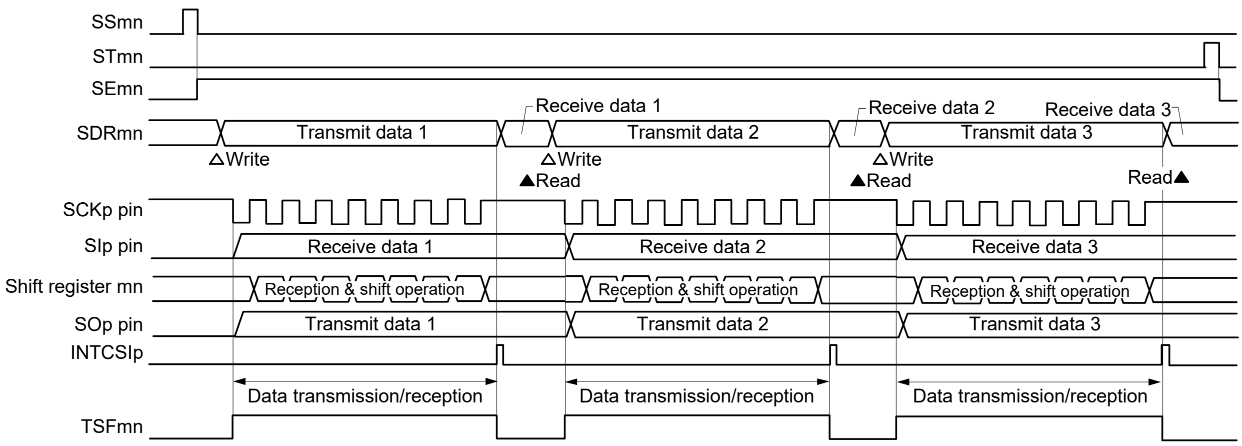


Figure 17 - 46 Procedure for Resuming Master Transmission/Reception



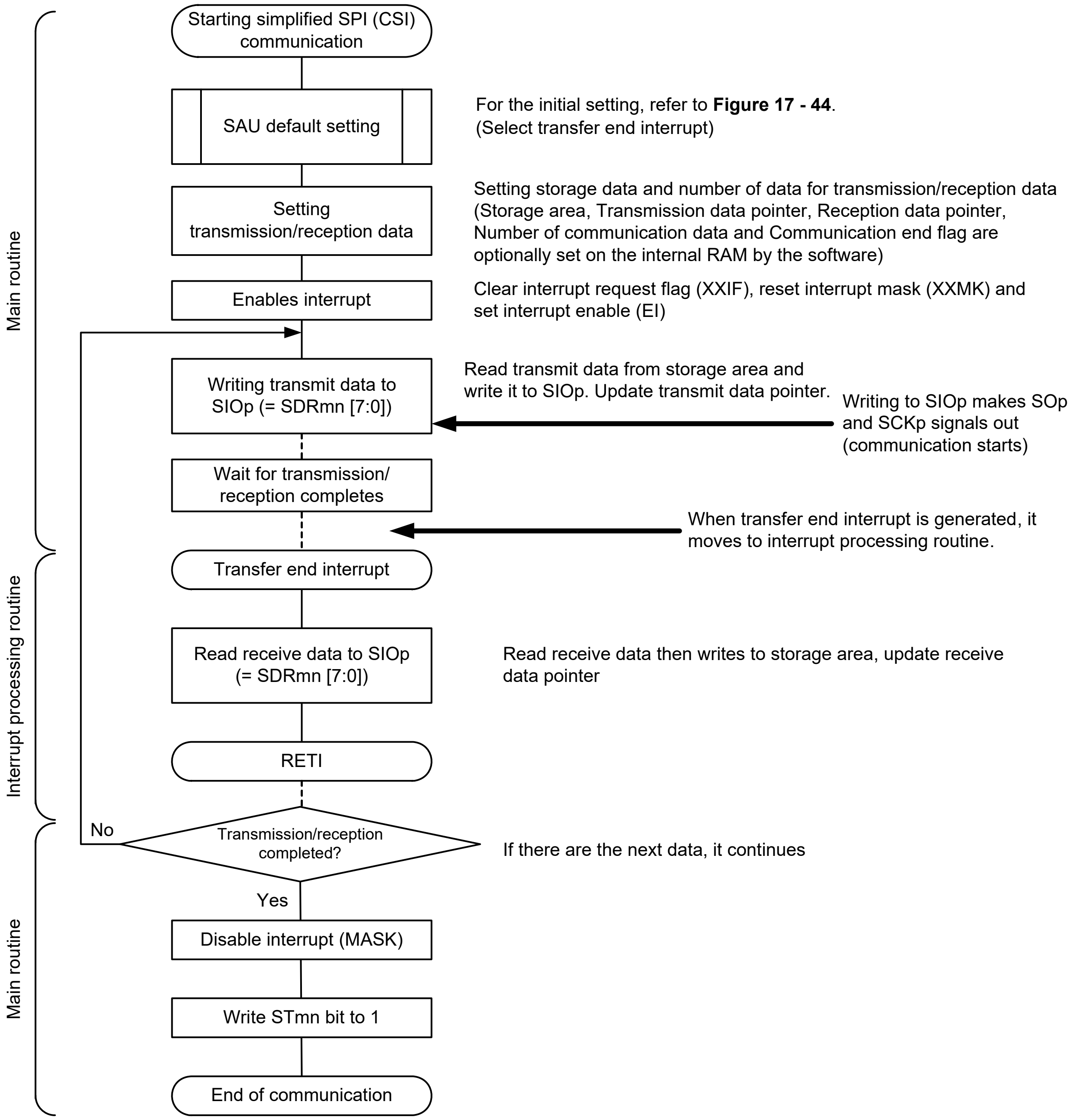
(3) Processing flow (in single-transmission/reception mode)

Figure 17 - 47 Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



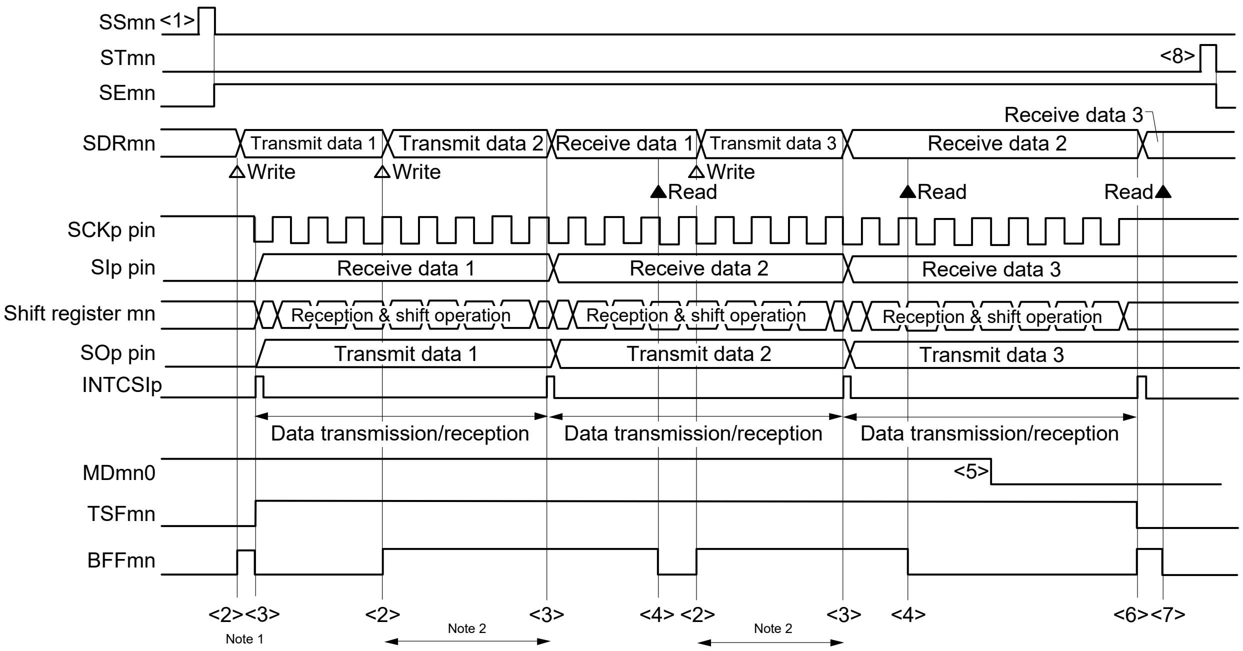
Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 17 - 48 Flowchart of Master Transmission/Reception (in Single-Transmission/Reception Mode)



(4) Processing flow (in continuous transmission/reception mode)

Figure 17 - 49 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.

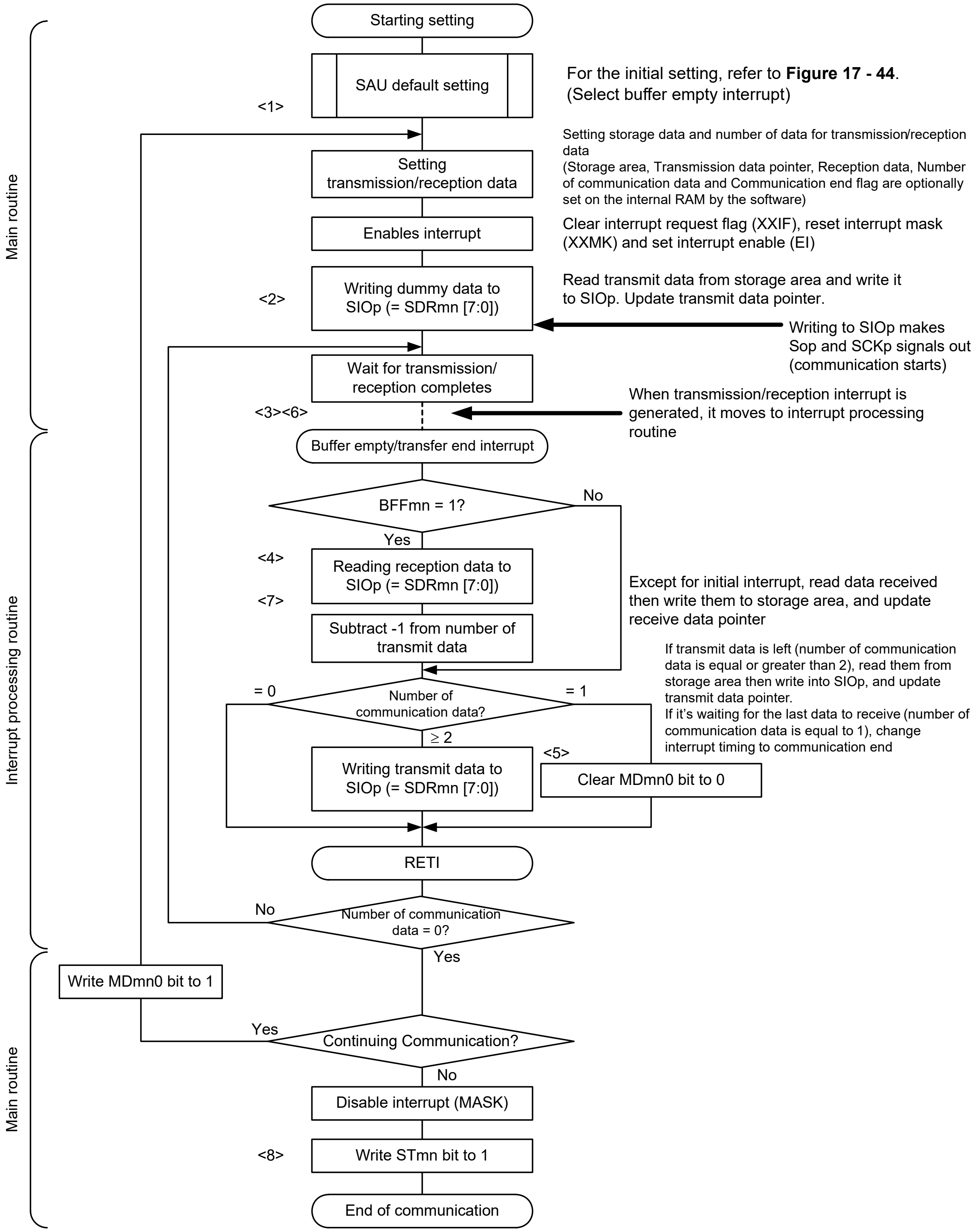
Note 2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution **The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.**

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in Figure 17 - 50 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode).

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 17 - 50 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



Remark <1> to <8> in the figure correspond to <1> to <8> in Figure 17 - 49 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

17.5.4 Slave transmission

Slave transmission is that the RL78 microcontroller transmits data to another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00	CSI01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCK00, SO00	SCK01, SO01
Interrupt	INTCSI00	INTCSI01
	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 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 and SCK01 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 in the electrical specifications (see **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark 1. f_{MCK}: Operation clock frequency of target channel

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

(1) Register setting

Figure 17 - 51 Example of Contents of Registers for Slave Transmission of Simplified SPI (CSI00, CSI01)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0/1

Operation clock (fMCK) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	1	0	0/1	0/1	0	0	0	0	0/1	0	0	0	0	1	1	0/1

Selection of the data and clock phase (For details about the setting, see **17.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	Transmit data setting							
									SIOp							

(d) Serial output register m (SOm)... Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm							CKOm1	CKOm0							SOm1	SOm0
	0	0	0	0	0	0	×	×	0	0	0	0	0	0	0/1	0/1

(e) Serial output enable register m (SOEm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm															SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1

(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm															SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Remark 2. ■: Setting is fixed in the simplified SPI (CSI) slave transmission mode,

■: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 17 - 52 Initial Setting Procedure for Slave Transmission

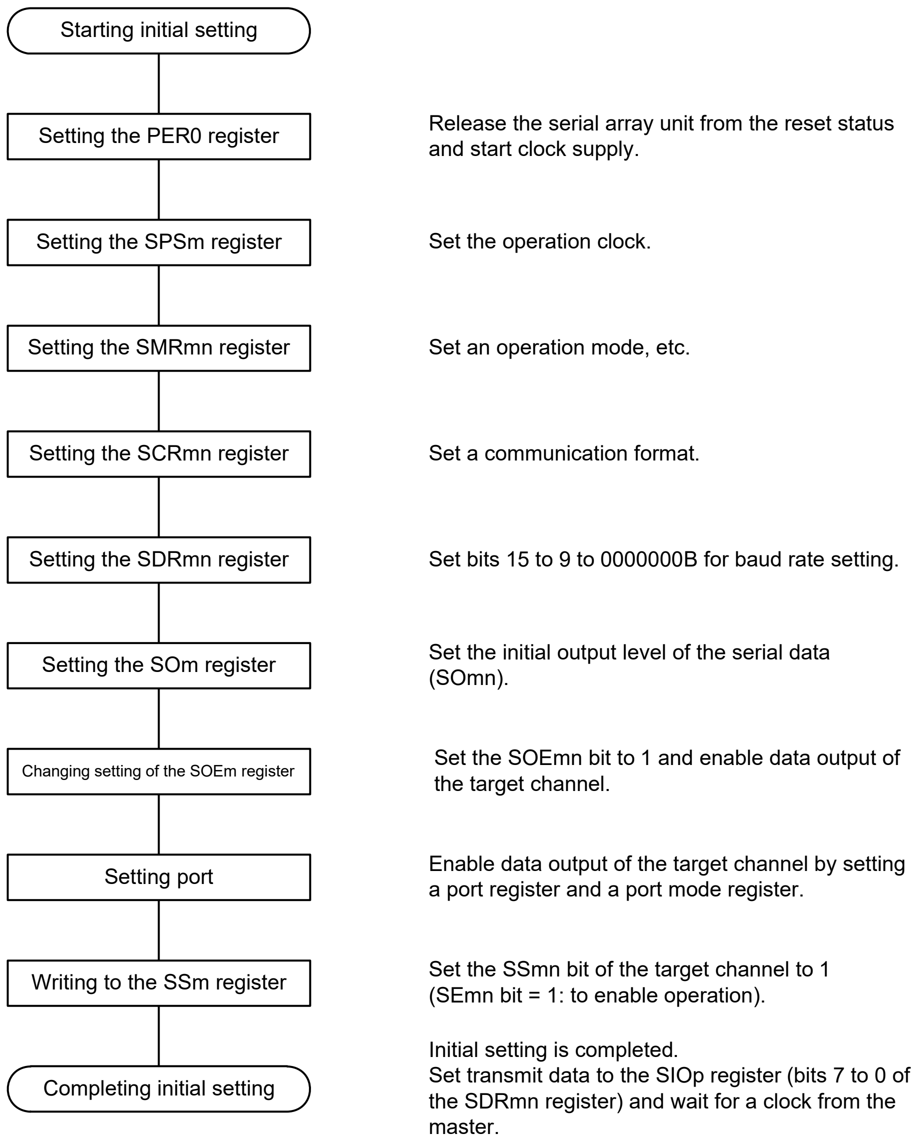


Figure 17 - 53 Procedure for Stopping Slave Transmission

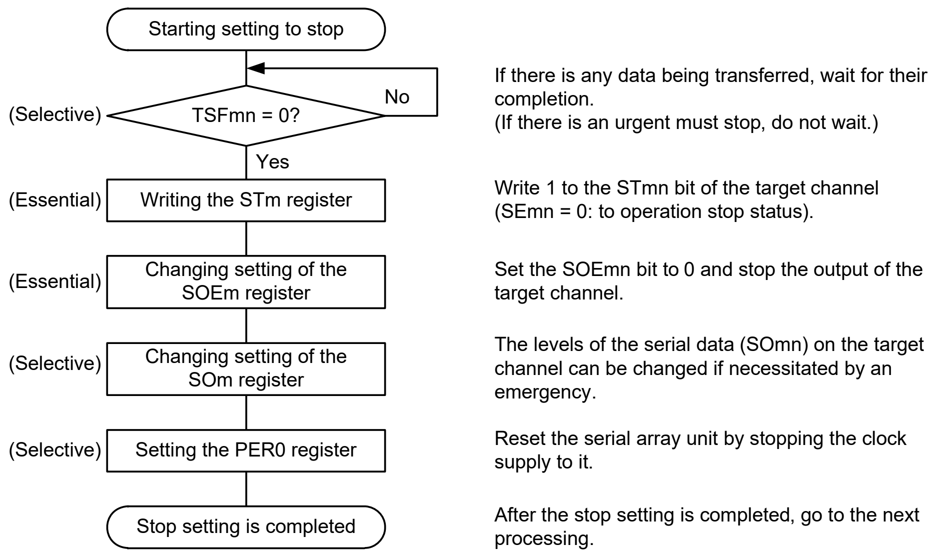
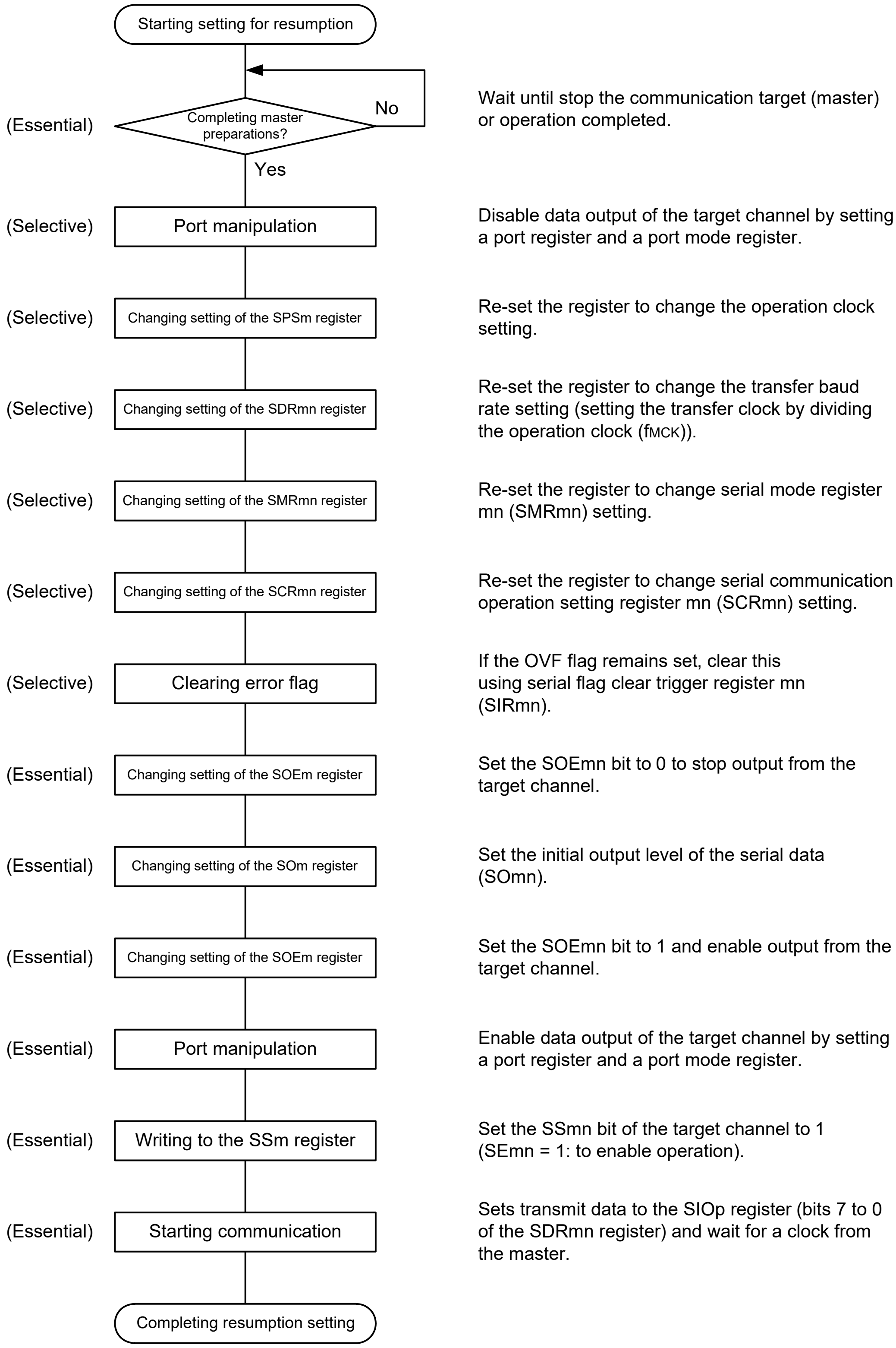


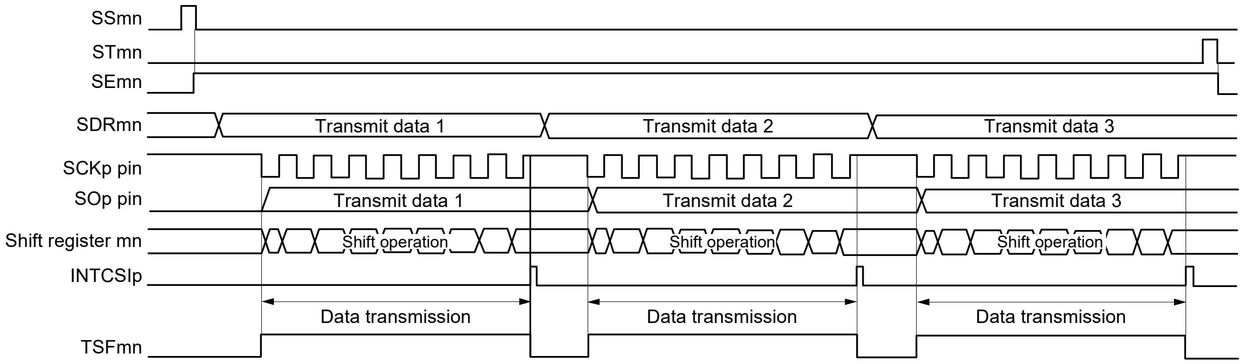
Figure 17 - 54 Procedure for Resuming Slave Transmission



Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

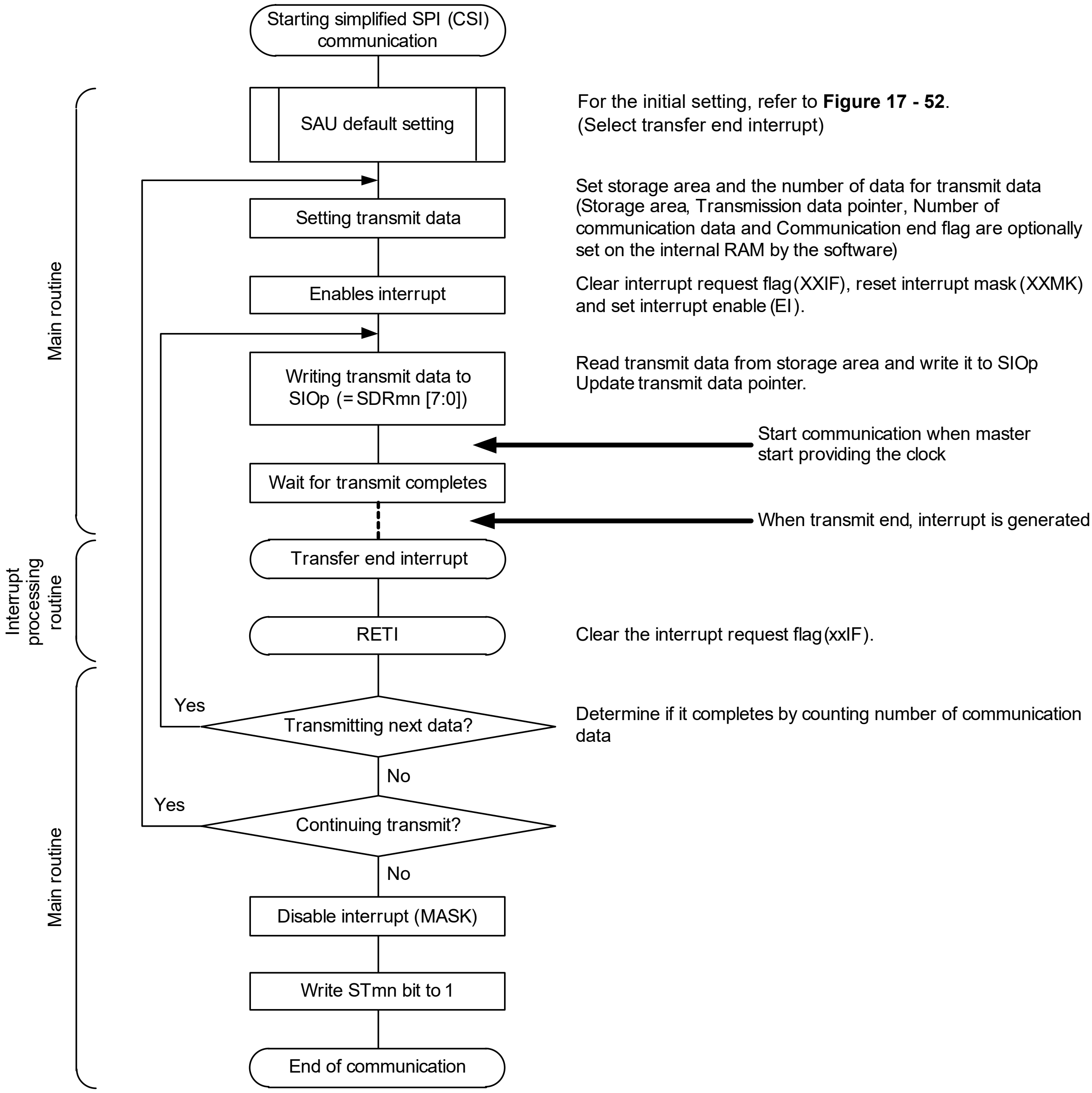
(3) Processing flow (in single-transmission mode)

Figure 17 - 55 Timing Chart of Slave Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



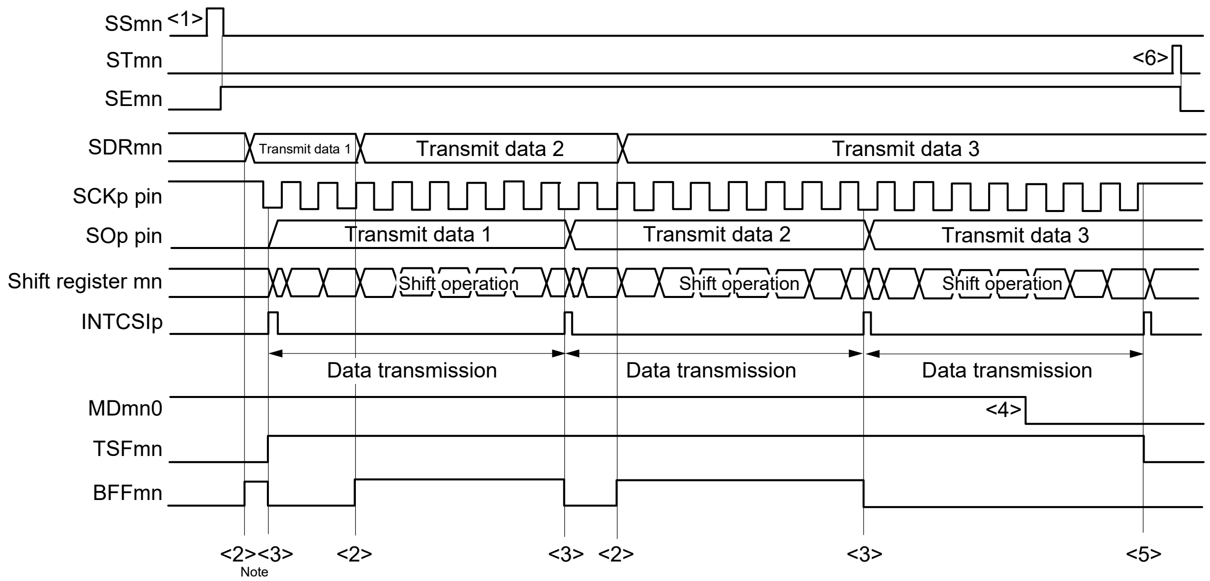
Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 17 - 56 Flowchart of Slave Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 17 - 57 Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)

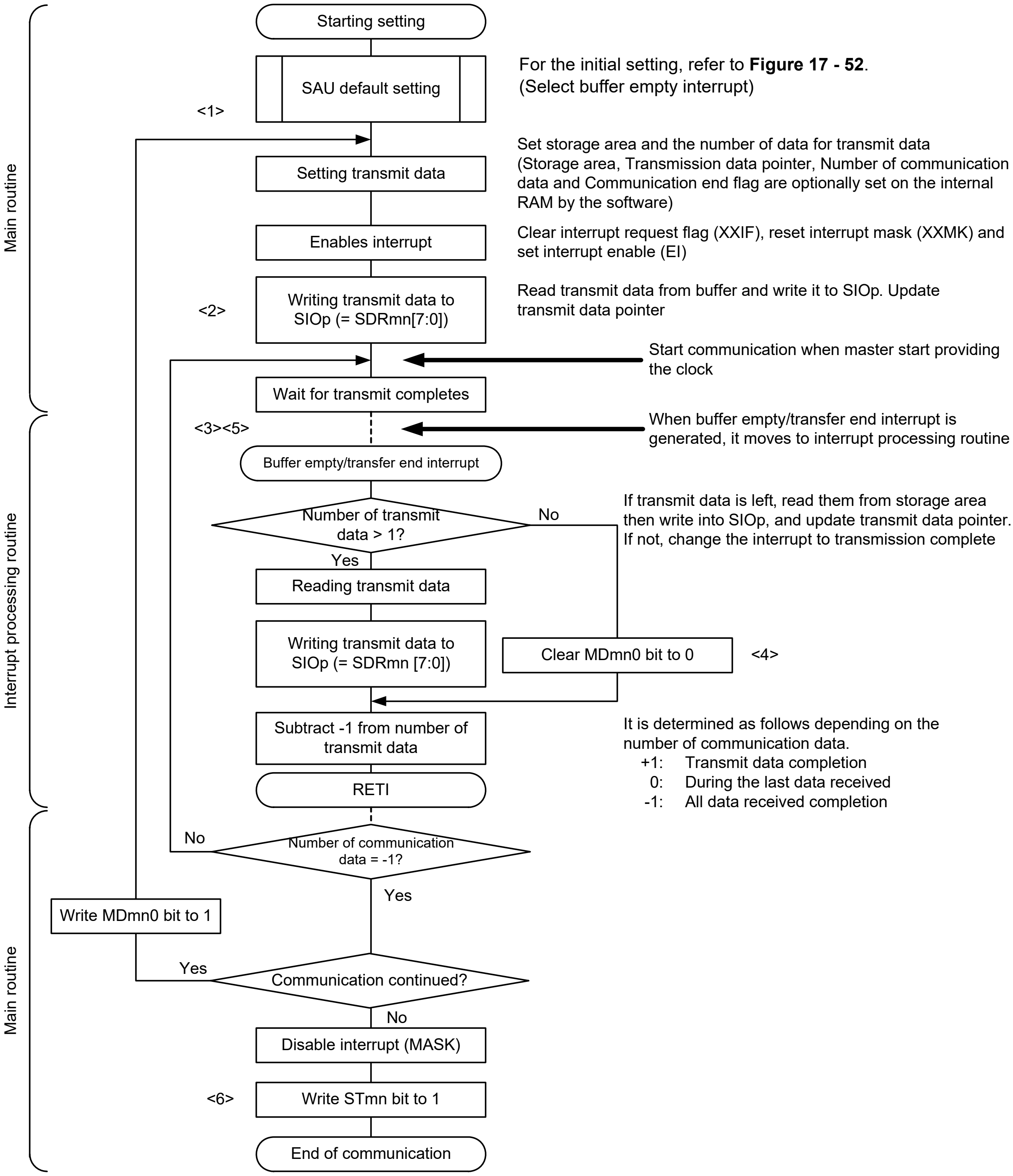


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution **The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.**

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 17 - 58 Flowchart of Slave Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in Figure 17 - 57 Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0).

17.5.5 Slave reception

Slave reception is that the RL78 microcontroller receives data from another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00	CSI01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCK00, SI00	SCK01, SI01
Interrupt	INTCSI00	INTCSI01
	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 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 and SCK011 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 in the electrical specifications (see **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark 1. f_{MCK}: Operation clock frequency of target channel

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

(1) Register setting

Figure 17 - 59 Example of Contents of Registers for Slave Reception of Simplified SPI (CSI00, CSI01)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **17.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15–9	8	7–0
SDRmn	0000000 Baud rate setting	0	Receive data (SIOp)

(d) Serial output register m (SOm)... The Register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 ×	CKOm0 ×	0	0	0	0	0	0	SOm1 ×	SOm0 ×

(e) Serial output enable register m (SOEm)... The Register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 ×	SOEm0 ×

(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 0/1	SSm0 0/1

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Remark 2. ■: Setting is fixed in the simplified SPI (CSI) slave reception mode,
■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 17 - 60 Initial Setting Procedure for Slave Reception

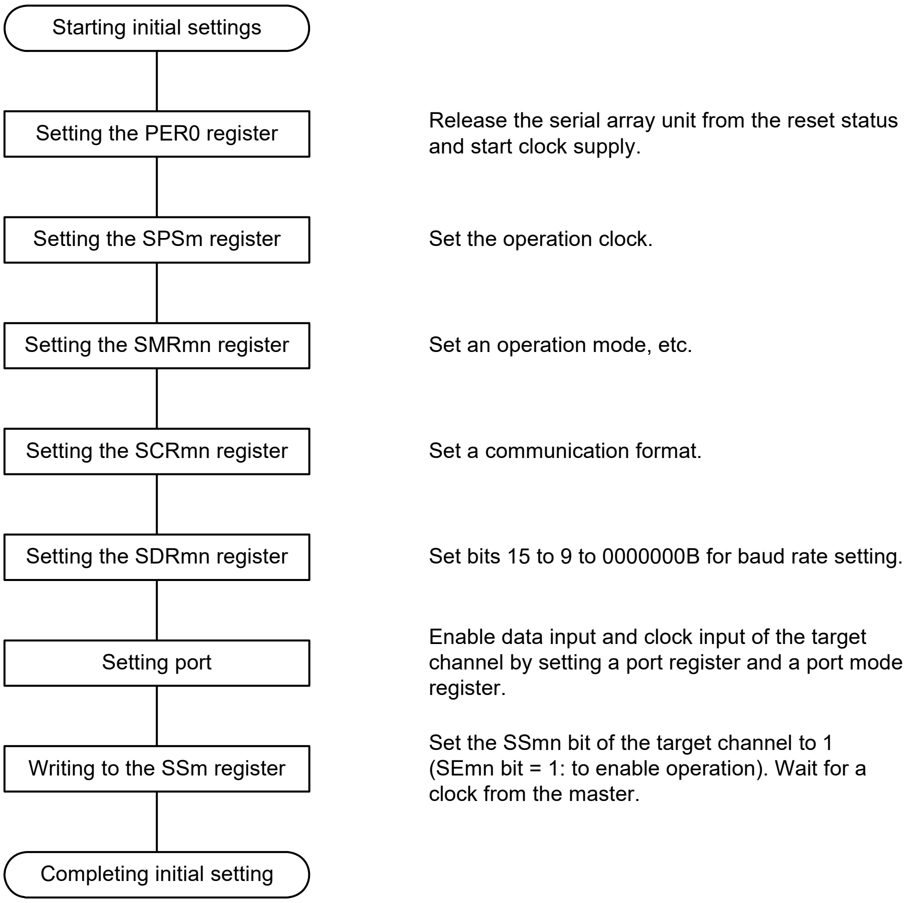


Figure 17 - 61 Procedure for Stopping Slave Reception

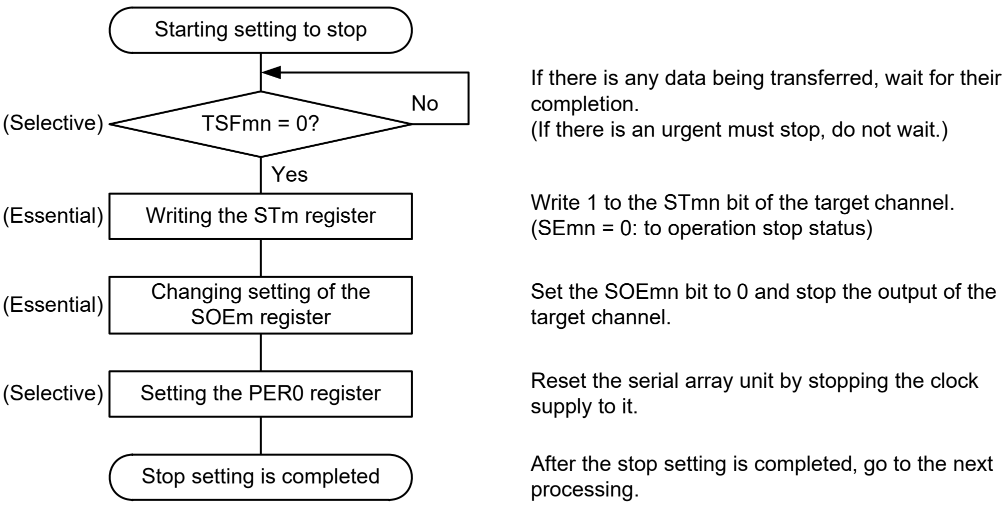
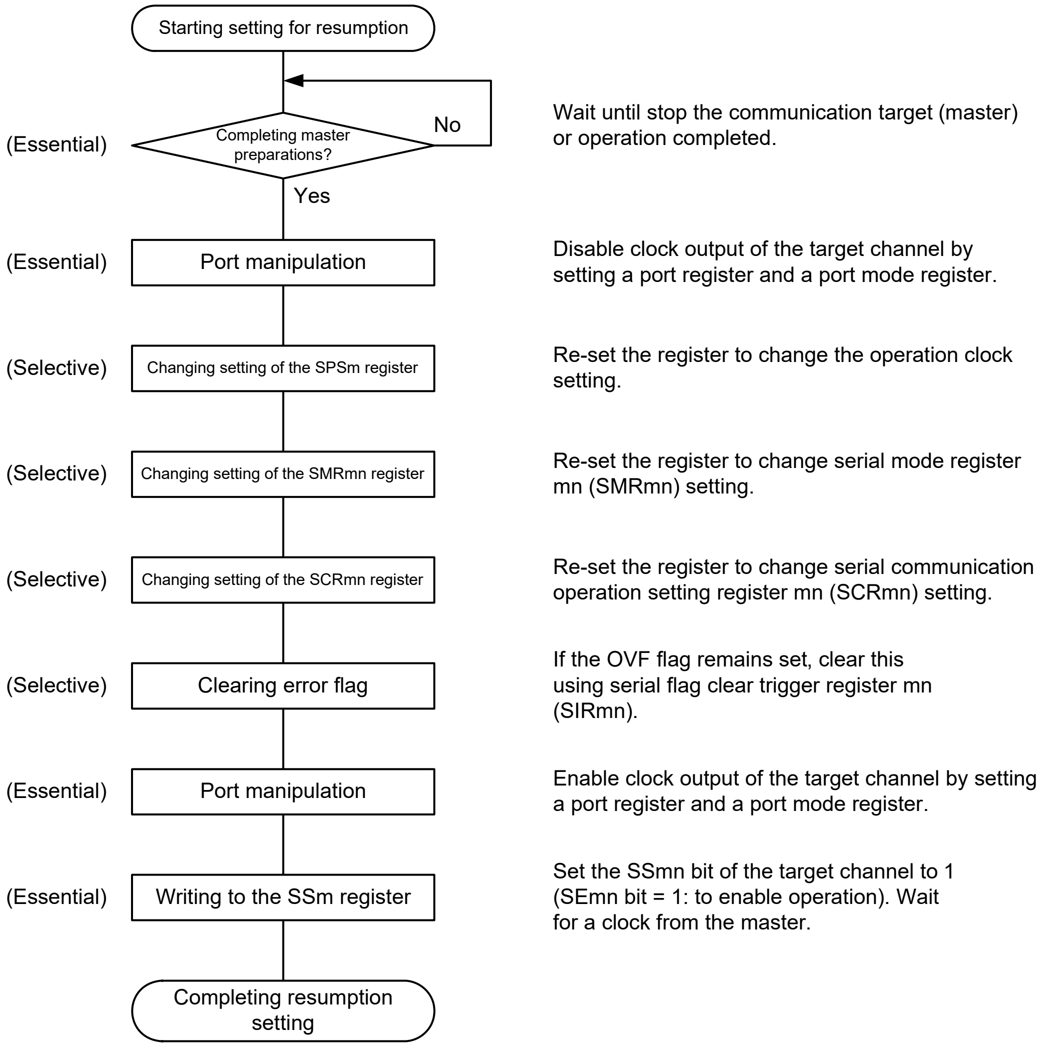


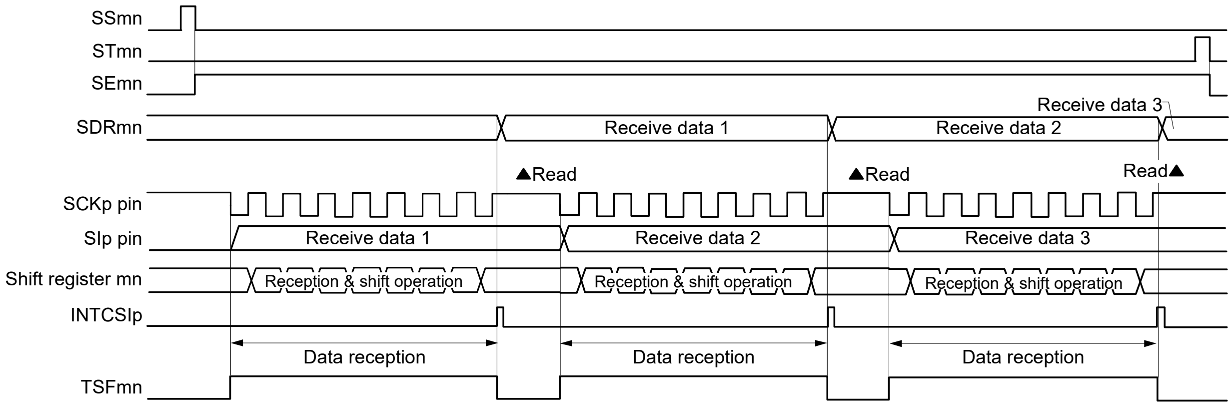
Figure 17 - 62 Procedure for Resuming Slave Reception



Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

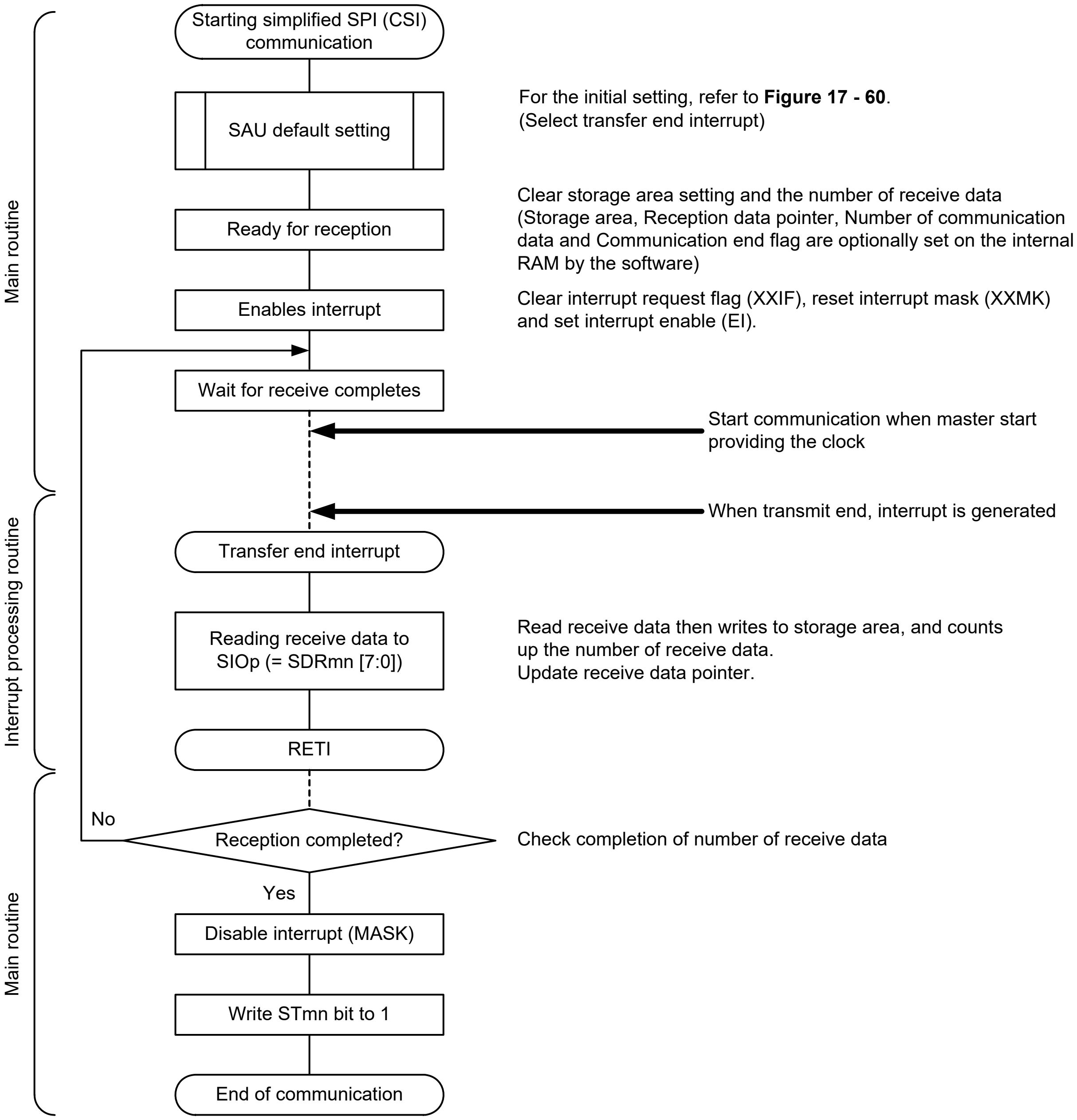
(3) Processing flow (in single-reception mode)

Figure 17 - 63 Timing Chart of Slave Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 17 - 64 Flowchart of Slave Reception (in Single-Reception Mode)



17.5.6 Slave transmission/reception

Slave transmission/reception is that the RL78 microcontroller transmits/receives data to/from another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00	CSI01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01
Interrupt	INTCSI00	INTCSI01
	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 from the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse	
Data direction	MSB or LSB first	

Note 1. Because the external serial clock input to the SCK00 and SCK01 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 in the electrical specifications (see **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark 1. f_{MCK}: Operation clock frequency of target channel

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

(1) Register setting

Figure 17 - 65 Example of Contents of Registers for Slave Transmission/Reception of Simplified SPI (CSI00, CSI01)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (fMCK) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **17.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	Transmit data setting/receive data register							

SIOp (bits 7 to 0)

(d) Serial output register m (SOm)... Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 ×	CKOm0 ×	0	0	0	0	0	0	SOm1 0/1	SOm0 0/1

(e) Serial output enable register m (SOEm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 0/1	SOEm0 0/1

(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 0/1	SSm0 0/1

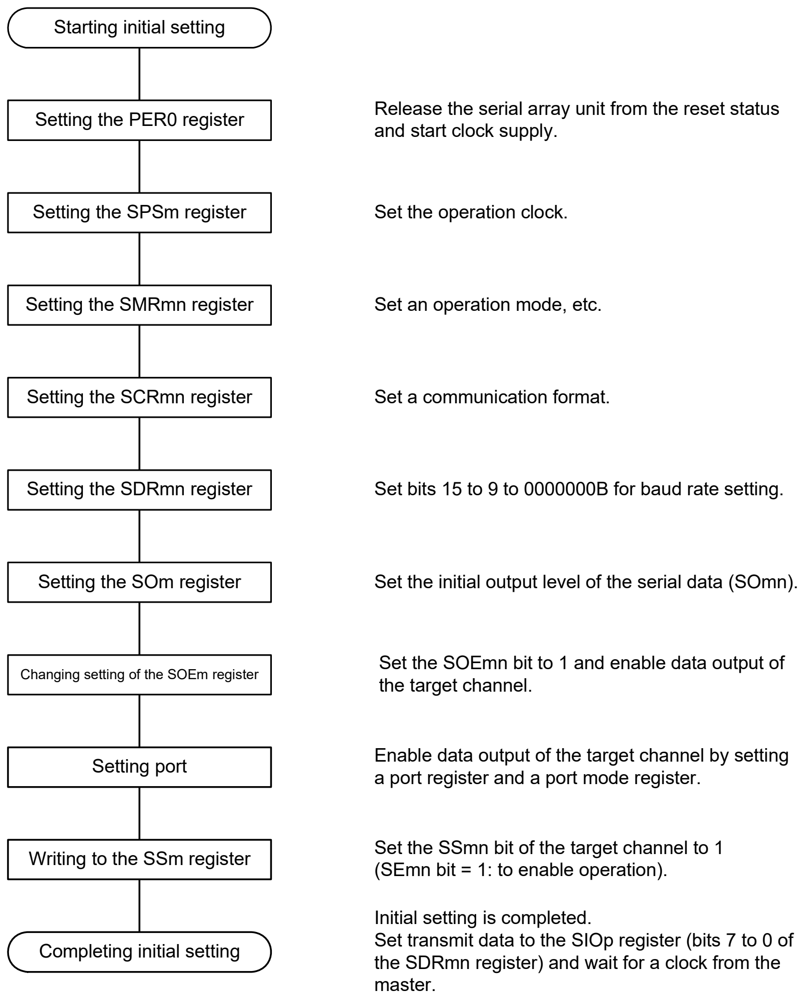
Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Remark 2. ■: Setting is fixed in the simplified SPI (CSI) master transmission/reception mode
▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 17 - 66 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 17 - 67 Procedure for Stopping Slave Transmission/Reception

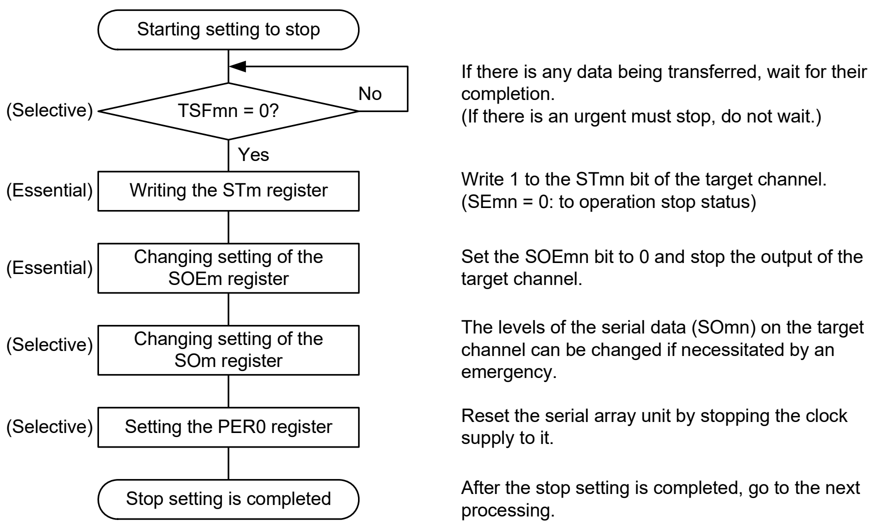
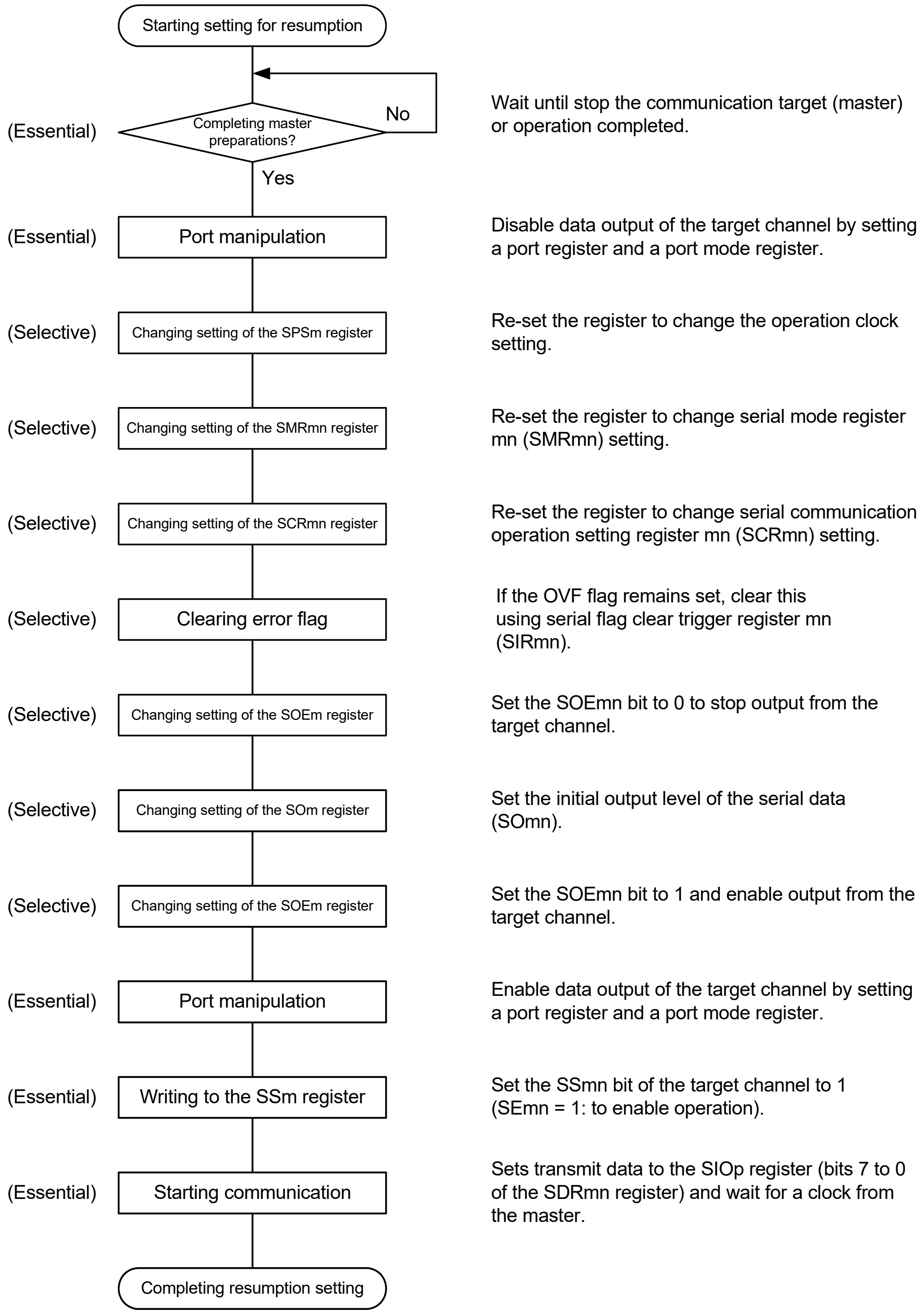


Figure 17 - 68 Procedure for Resuming Slave Transmission/Reception

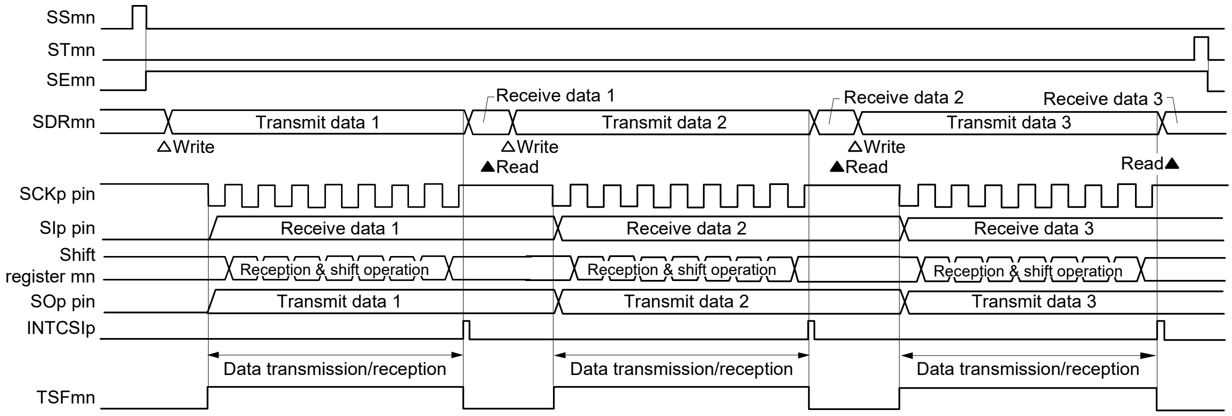


Caution 1. Be sure to set transmit data to the SIOp register before the clock from the master is started.

Caution 2. If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

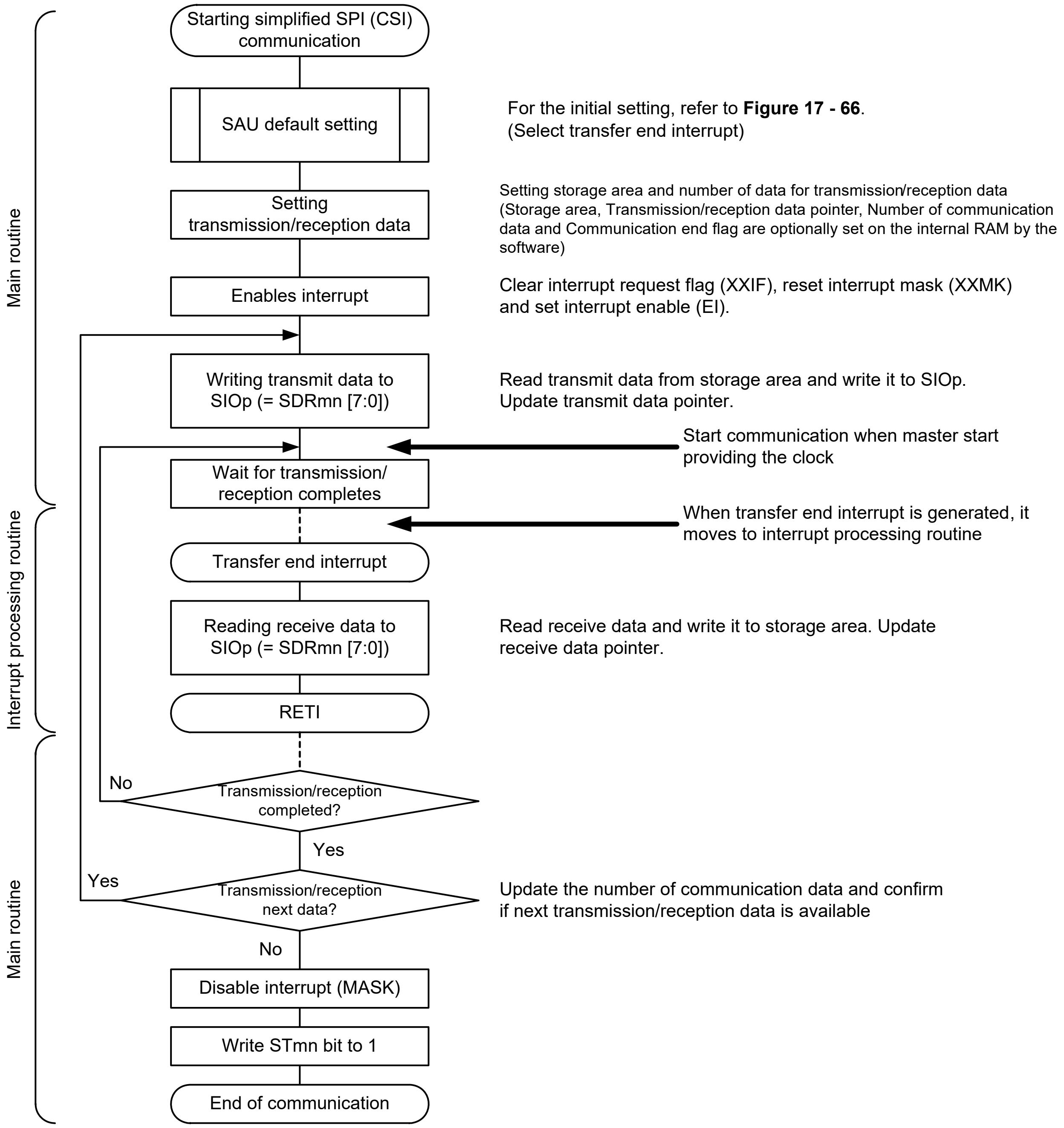
(3) Processing flow (in single-transmission/reception mode)

Figure 17 - 69 Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

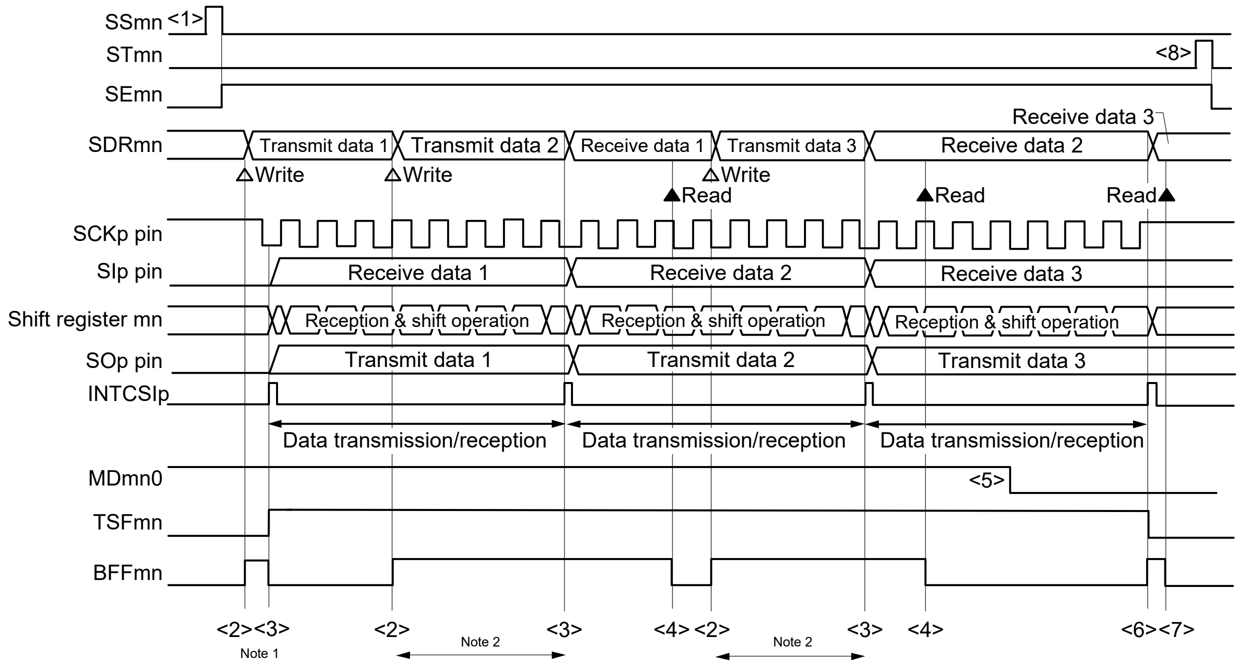
Figure 17 - 70 Flowchart of Slave Transmission/Reception (in Single-Transmission/Reception Mode)



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 17 - 71 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.

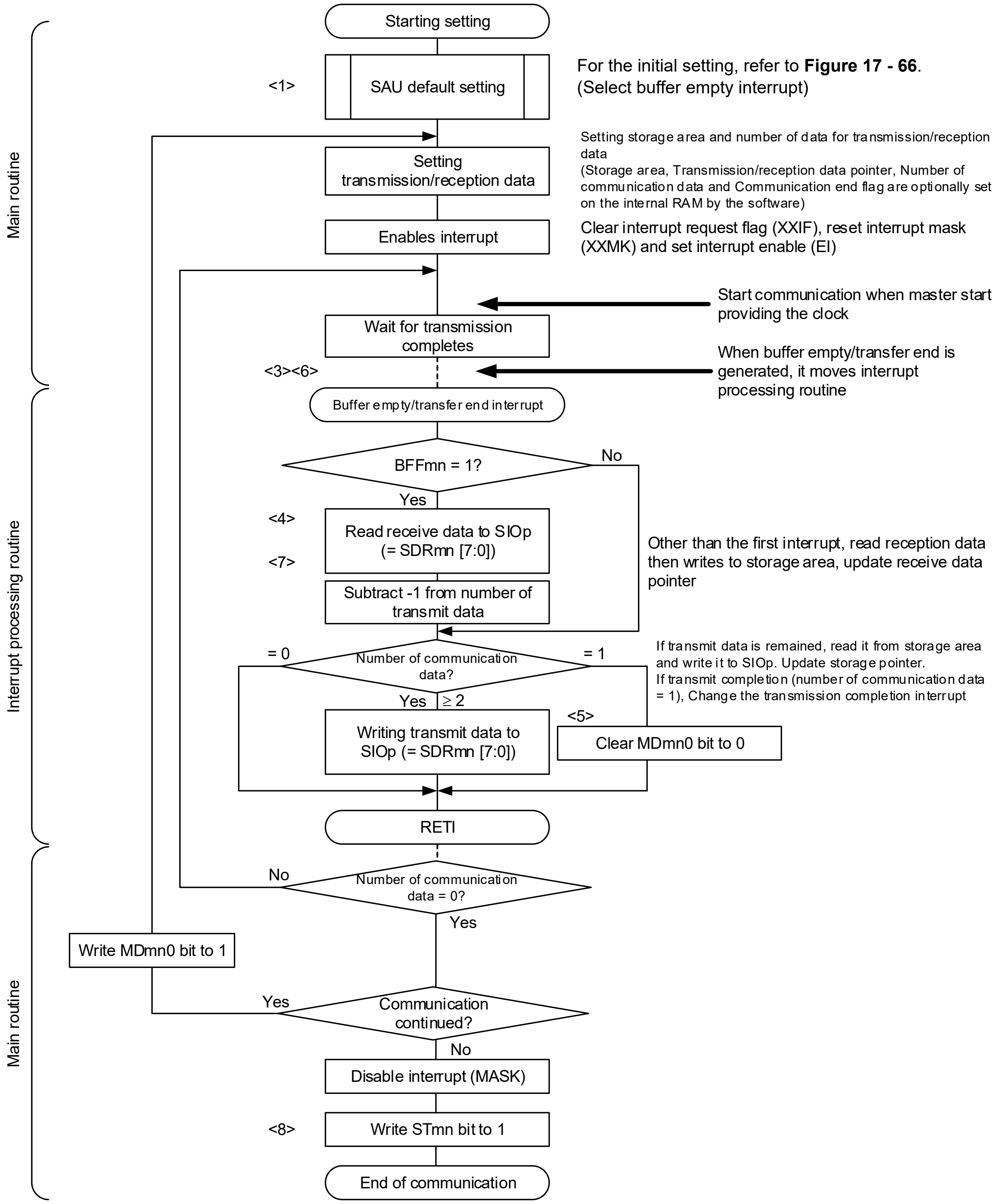
Note 2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution **The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.**

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in Figure 17 - 72 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode).

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 17 - 72 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 17 - 71 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

17.5.7 SNOOZE mode function

In the SNOOZE mode, the simplified SPI (CSI) proceeds with reception upon detection of input on the SCKp pin while in the STOP mode. Transfer through the simplified SPI (CSI) will normally stop in the STOP mode. However, using the SNOOZE mode enables reception by the simplified SPI (CSI) without CPU operations upon detection of input on the SCKp pin. Only the following channel can be set to the SNOOZE mode.

• CSI00

When using the simplified SPI (CSI) in SNOOZE mode, make the following setting before switching to the STOP mode (see **Figure 17 - 74** and **Figure 17 - 76 Flowchart of SNOOZE Mode Operation**).

• 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.

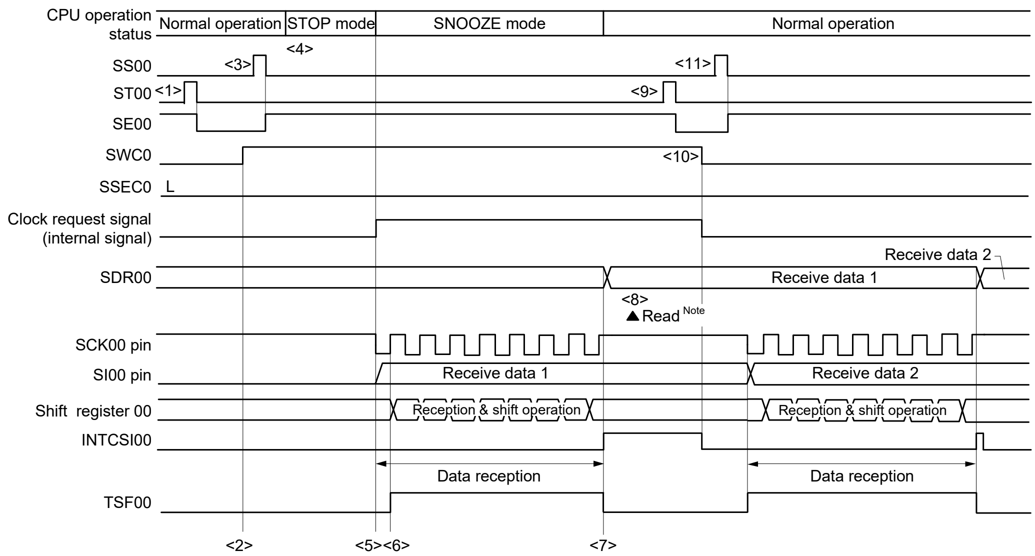
The CPU shifts to the SNOOZE mode on detecting the valid edge of the SCKp signal following a transition to the STOP mode. A CSIp starts reception on detecting input of the serial clock on the SCKp pin.

Caution 1. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock (fIH) or middle-speed on-chip oscillator clock (fIM) is selected for fCLK.

Caution 2. The maximum transfer rate when using CSIp in the SNOOZE mode is 1 Mbps.

(1) SNOOZE mode operation (once startup)

Figure 17 - 73 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.

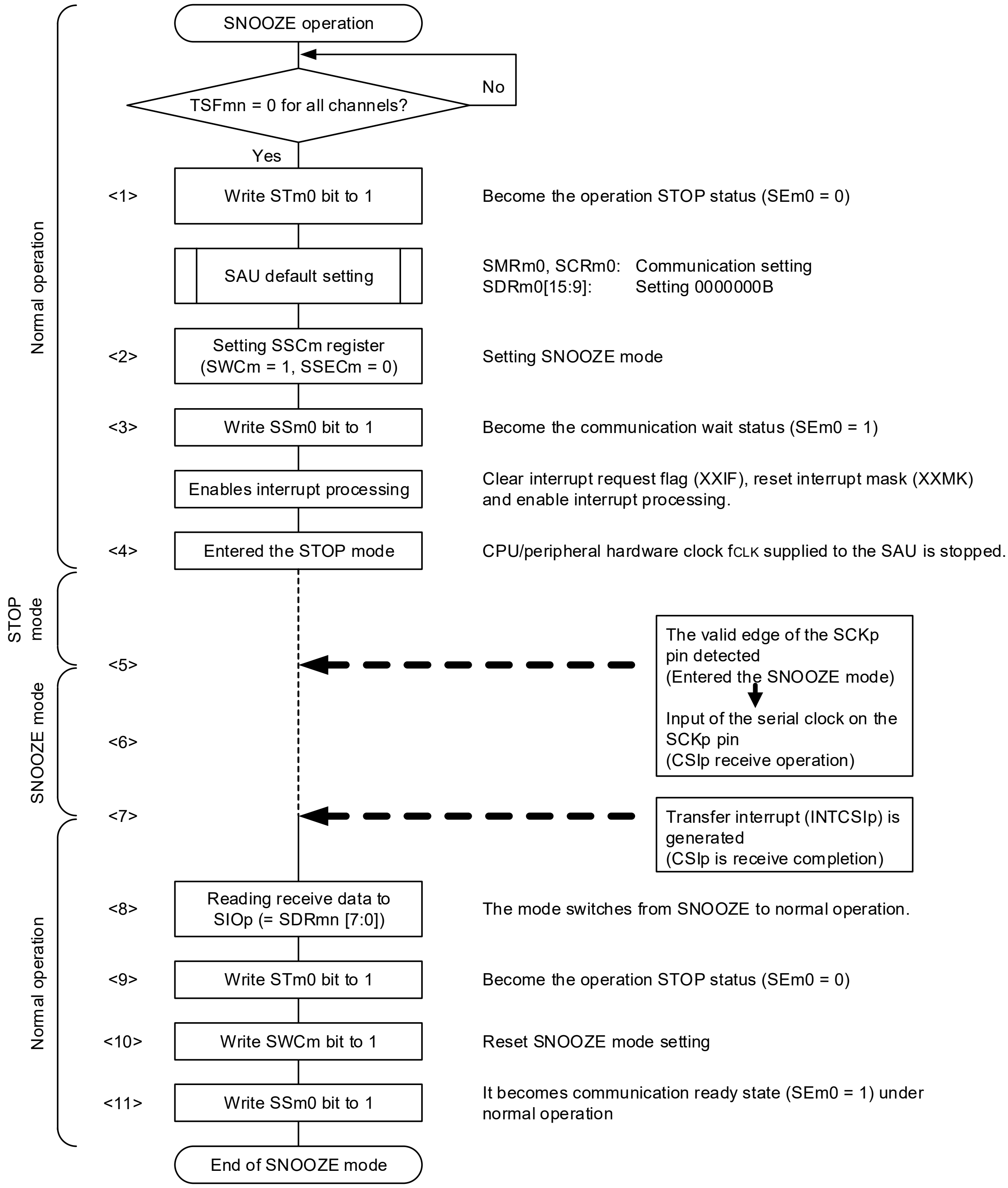
Caution 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (clear the SEm0 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

Caution 2. When SWCm = 1, the BFFm1 and OVFm1 flags will not change.

Remark 1. <1> to <11> in the figure correspond to <1> to <11> in Figure 17 - 74 Flowchart of SNOOZE Mode Operation (once startup).

Remark 2. m = 0; p = 00

Figure 17 - 74 Flowchart of SNOOZE Mode Operation (once startup)

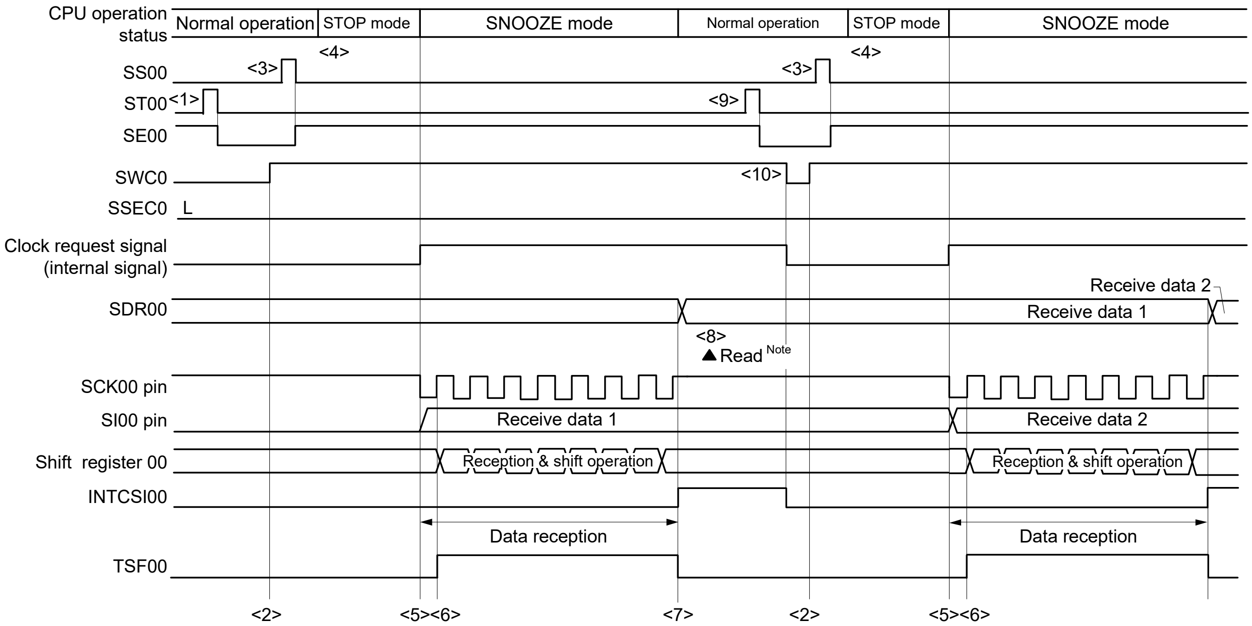


Remark 1. <1> to <11> in the figure correspond to <1> to <11> in Figure 17 - 73 Timing Chart of SNOOZE Mode Operation (once startup) (Type 1: DAPmn = 0, CKPmn = 0).

Remark 2. m = 0; p = 00

(2) SNOOZE mode operation (continuous startup)

Figure 17 - 75 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.

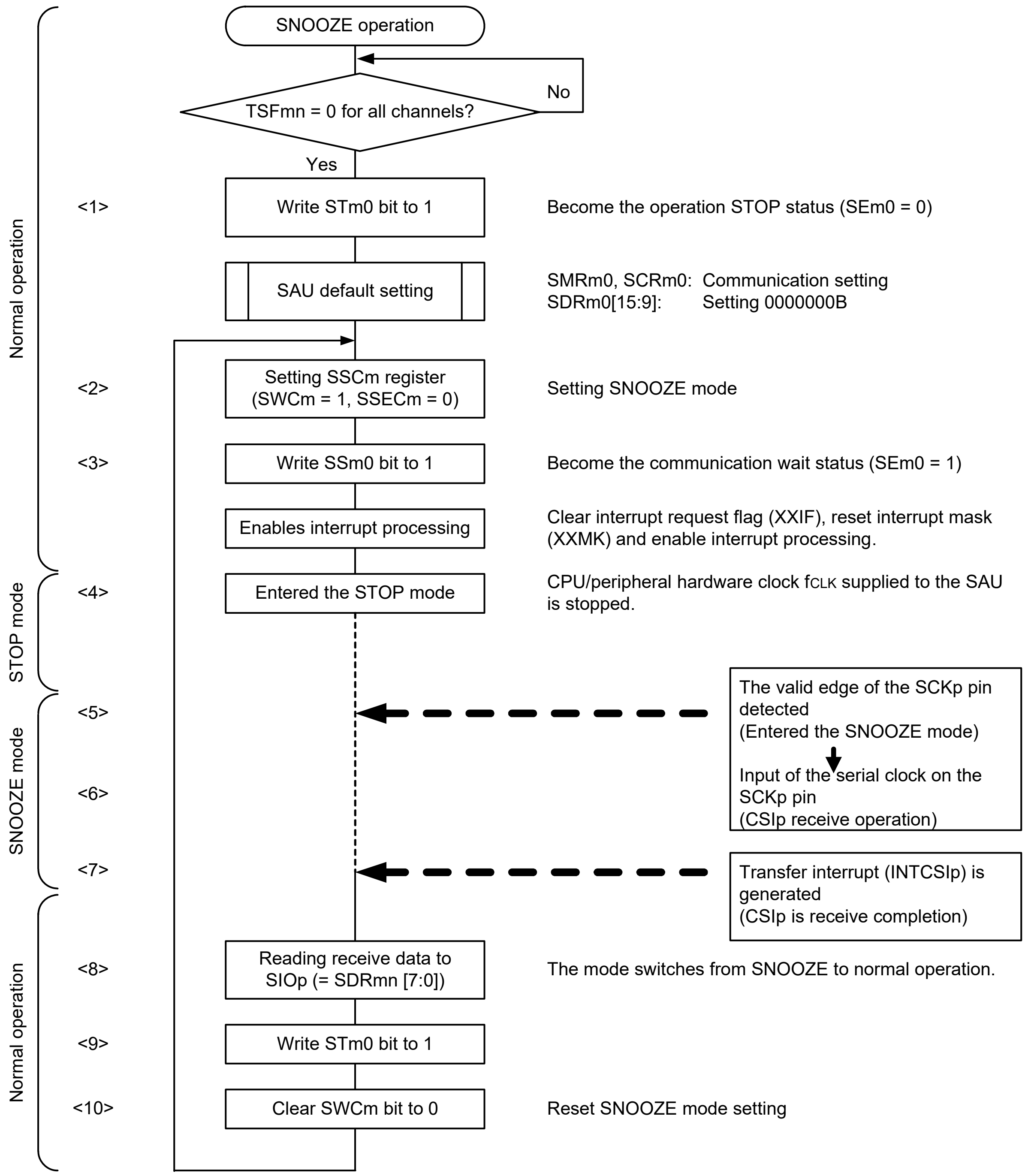
Caution 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (clear the SEm0 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE release).

Caution 2. When SWCm = 1, the BFFm1 and OVFm1 flags will not change.

Remark 1. <1> to <10> in the figure correspond to <1> to <10> in Figure 17 - 76 Flowchart of SNOOZE Mode Operation (continuous startup).

Remark 2. m = 0; p = 00

Figure 17 - 76 Flowchart of SNOOZE Mode Operation (continuous startup)



Remark 1. <1> to <10> in the figure correspond to <1> to <10> in Figure 17 - 75 Timing Chart of SNOOZE Mode Operation (continuous startup) (Type 1: DAPmn = 0, CKPmn = 0).

Remark 2. m = 0; p = 00

17.5.8 Calculating transfer clock frequency

The transfer clock frequency for simplified SPI (CSI00, CSI01) communication can be calculated by the following expressions.

(1) Master

(Transfer clock frequency) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [Hz]

(2) Slave

(Transfer clock frequency) = {Frequency of serial clock (SCK) supplied by master} Note [Hz]

Note The permissible maximum transfer clock frequency is $f_{MCK}/6$.

Remark The value of SDRmn[15:9] is the value of bits 15 to 9 of serial data register mn (SDRmn) (0000000B to 1111111B) and therefore is 0 to 127.

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 17 - 2 Selection of Operation Clock For Simplified SPI

SMRmn Register	SPSm Register								Operation Clock (f_{MCK}) Note	
CKSmn	PRSm13	PRSm12	PRSm11	PRSm10	PRSm03	PRSm02	PRSm01	PRSm00		f_{CLK} = 24 MHz
0	×	×	×	×	0	0	0	0	f_{CLK}	24 MHz
	×	×	×	×	0	0	0	1	$f_{CLK}/2$	12 MHz
	×	×	×	×	0	0	1	0	$f_{CLK}/2^2$	6 MHz
	×	×	×	×	0	0	1	1	$f_{CLK}/2^3$	3 MHz
	×	×	×	×	0	1	0	0	$f_{CLK}/2^4$	1.5 MHz
	×	×	×	×	0	1	0	1	$f_{CLK}/2^5$	750 kHz
	×	×	×	×	0	1	1	0	$f_{CLK}/2^6$	375 kHz
	×	×	×	×	0	1	1	1	$f_{CLK}/2^7$	187.5 kHz
	×	×	×	×	1	0	0	0	$f_{CLK}/2^8$	93.8 kHz
	×	×	×	×	1	0	0	1	$f_{CLK}/2^9$	46.9 kHz
	×	×	×	×	1	0	1	0	$f_{CLK}/2^{10}$	23.4 kHz
	×	×	×	×	1	0	1	1	$f_{CLK}/2^{11}$	11.7 kHz
	×	×	×	×	1	1	0	0	$f_{CLK}/2^{12}$	5.86 kHz
	×	×	×	×	1	1	0	1	$f_{CLK}/2^{13}$	2.93 kHz
	×	×	×	×	1	1	1	0	$f_{CLK}/2^{14}$	1.46 kHz
	×	×	×	×	1	1	1	1	$f_{CLK}/2^{15}$	732 Hz
1	0	0	0	0	×	×	×	×	f_{CLK}	24 MHz
	0	0	0	1	×	×	×	×	$f_{CLK}/2$	12 MHz
	0	0	1	0	×	×	×	×	$f_{CLK}/2^2$	6 MHz
	0	0	1	1	×	×	×	×	$f_{CLK}/2^3$	3 MHz
	0	1	0	0	×	×	×	×	$f_{CLK}/2^4$	1.5 MHz
	0	1	0	1	×	×	×	×	$f_{CLK}/2^5$	750 kHz
	0	1	1	0	×	×	×	×	$f_{CLK}/2^6$	375 kHz
	0	1	1	1	×	×	×	×	$f_{CLK}/2^7$	187.5 kHz
	1	0	0	0	×	×	×	×	$f_{CLK}/2^8$	93.8 kHz
	1	0	0	1	×	×	×	×	$f_{CLK}/2^9$	46.9 kHz
	1	0	1	0	×	×	×	×	$f_{CLK}/2^{10}$	23.4 kHz
	1	0	1	1	×	×	×	×	$f_{CLK}/2^{11}$	11.7 kHz
	1	1	0	0	×	×	×	×	$f_{CLK}/2^{12}$	5.86 kHz
	1	1	0	1	×	×	×	×	$f_{CLK}/2^{13}$	2.93 kHz
	1	1	1	0	×	×	×	×	$f_{CLK}/2^{14}$	1.46 kHz
	1	1	1	1	×	×	×	×	$f_{CLK}/2^{15}$	732 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).

Remark 1. ×: Don't care

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

17.5.9 Procedure for processing errors that occurred during simplified SPI (CSI00, CSI01) communication

The procedure for processing errors that occurred during simplified SPI (CSI00, CSI01) communication is described in Figure 17 - 77.

Figure 17 - 77 Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

17.6 Clock Synchronous Serial Communication with Slave Select Input Function

Channel 0 of SAU0 correspond to the clock synchronous serial communication with slave select input function.

[Data transmission/reception]
- Data length of 7 or 8 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable
- Level setting of transmit/receive data

[Clock control]
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel
- Maximum transfer rate [Note]

  During slave communication: Max. $f_{MCK}/6$

[Interrupt function]
- Transfer end interrupt/buffer empty interrupt

[Error detection flag]
- Overrun error

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics. For details, see **CHAPTER 34 ELECTRICAL SPECIFICATIONS**.

• 20-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	—		—

• 24, 32-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01

• 30-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	—		—

• 48-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00 (supporting slave select input function)	UART0	IIC00
	1	CSI01		IIC01

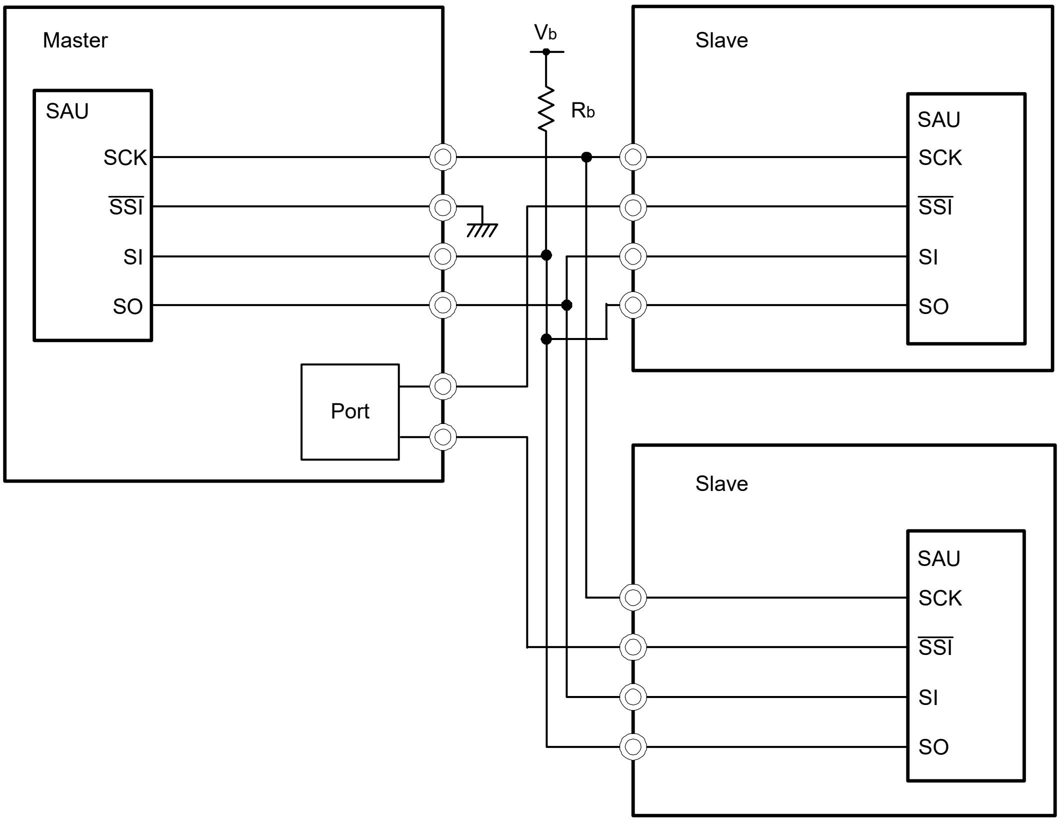
Slave select input function performs the following three types of communication operations.

• Slave transmission (See **17.6.1**.)
• Slave reception (See **17.6.2**.)
• Slave transmission/reception (See **17.6.3**.)

Multiple slaves can be connected to a master and communication can be performed by using the slave select input function. The master outputs a slave select signal to the slave (one) that is the other party of communication, and each slave judges whether it has been selected as the other party of communication and controls the SO pin output. When a slave is selected, transmit data can be communicated from the SO pin to the master. When a slave is not selected, the SO pin is set to high-level output. Therefore, in an environment where multiple slaves are connected, it is necessary set the SO pin to N-ch open-drain and pull up the node. Furthermore, when a slave is not selected, no transmission/reception operation is performed even if a serial clock is input from the master.

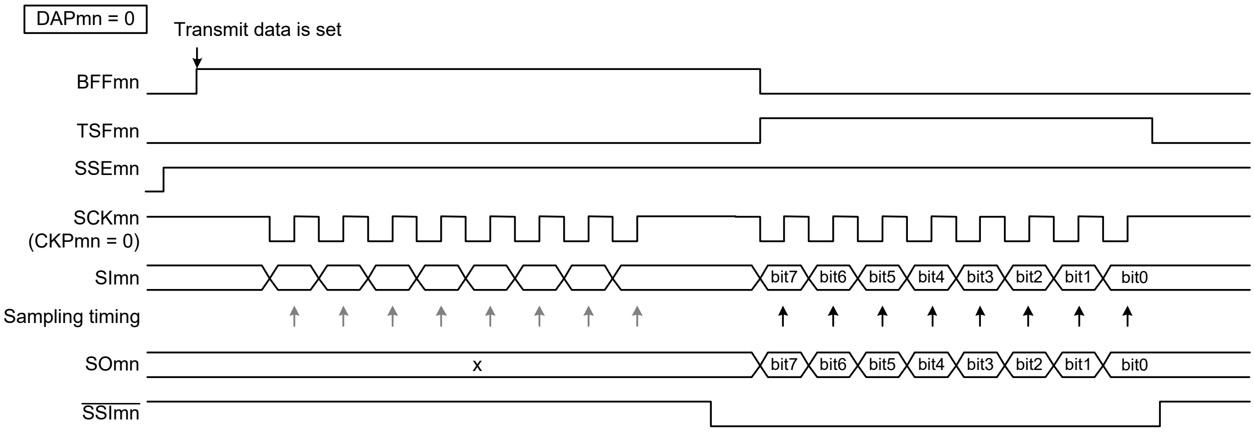
Caution Output the slave select signal by port manipulation.

Figure 17 - 78 Example of Slave Select Input Function Configuration

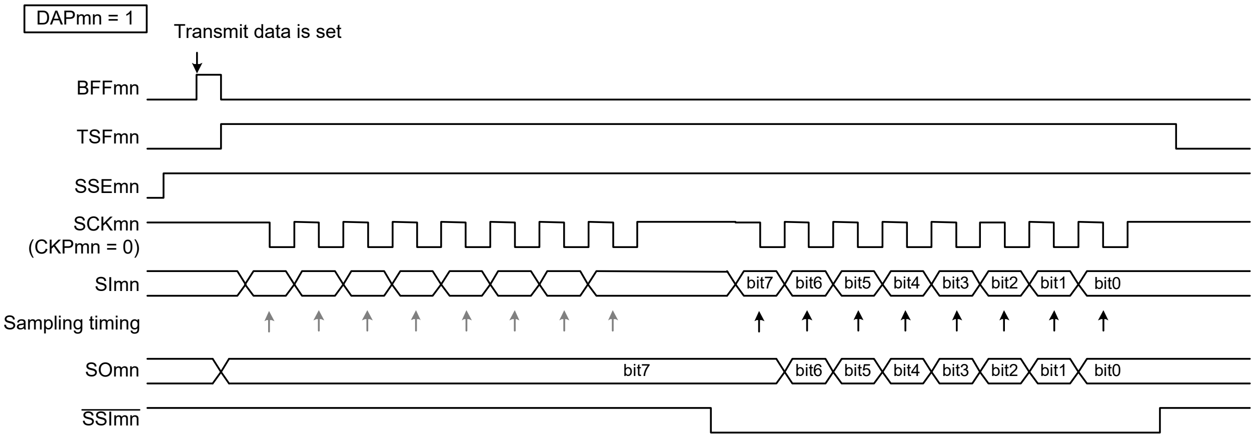


Caution Make sure $V_{DD} \geq V_b$.
Select the N-ch open-drain output (V_{DD} tolerance) mode for the SO00 pin.

Figure 17 - 79 Slave Select Input Function Timing Diagram



While $\overline{SSImn}$ is at high level, transmission is not performed even if the falling edge of SCKmn (serial clock) arrives, and neither is receive data sampled in synchronization with the rising edge.
When $\overline{SSImn}$ goes to low level, data is output (shifted) in synchronization with the falling edge of the serial clock and a reception operation is performed in synchronization with the rising edge.



If DAPmn = 1, when transmit data is set while $\overline{SSImn}$ is at high level, the first data (bit 7) is output to the data output. However, no shift operation is performed even if the rising edge of SCKmn (serial clock) arrives, and neither is receive data sampled in synchronization with the falling edge. When $\overline{SSImn}$ goes to low level, data is output (shifted) in synchronization with the next rising edge and a reception operation is performed in synchronization with the falling edge.

Remark m: Unit number (m = 0), n: Channel number (n = 0)

17.6.1 Slave transmission

Slave transmission is that the RL78 microcontroller transmits data to another device in the state of a transfer clock being input from another device.

Slave select Input function	CSI00
Target channel	Channel 0 of SAU0
Pins used	SCK00, SO00, $\overline{\text{SSI00}}$
Interrupt	INTCSI00
	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 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
Slave select Input function	Slave select input function operation selectable

Note 1. Because the external serial clock input to the SCK00 pin 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 in the electrical specifications (see **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark 1. f_{MCK}: Operation clock frequency of target channel

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0)

(1) Register setting

Figure 17 - 80 Example of Contents of Registers for Slave Transmission of Slave Select Input Function (CSI00) (1/2)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **17.3 Registers Controlling Serial Array Unit**)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15–9	8	7–0
SDRmn	0000000 Baud rate setting	0	Transmit data setting (SIOp)

(d) Serial output register m (SOm)... Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 ×	CKOm0 ×	0	0	0	0	0	0	SOm1 ×	SOm0 0/1

(e) Serial output enable register m (SOEm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 ×	SOEm0 0/1

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Remark 2. ■: Setting is fixed in the simplified SPI (CSI) slave transmission mode,
■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 17 - 81 Example of Contents of Registers for Slave Transmission of Slave Select Input Function (CSI00) (2/2)

(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 ×	SSm0 0/1

(g) Input switch control register (ISC)... $\overline{\text{SSI00}}$ input setting in CSI00 slave channel (channel 0 of unit 0).

	7	6	5	4	3	2	1	0
ISC	SSIE00 0/1	0	0	0	0	0	0	0

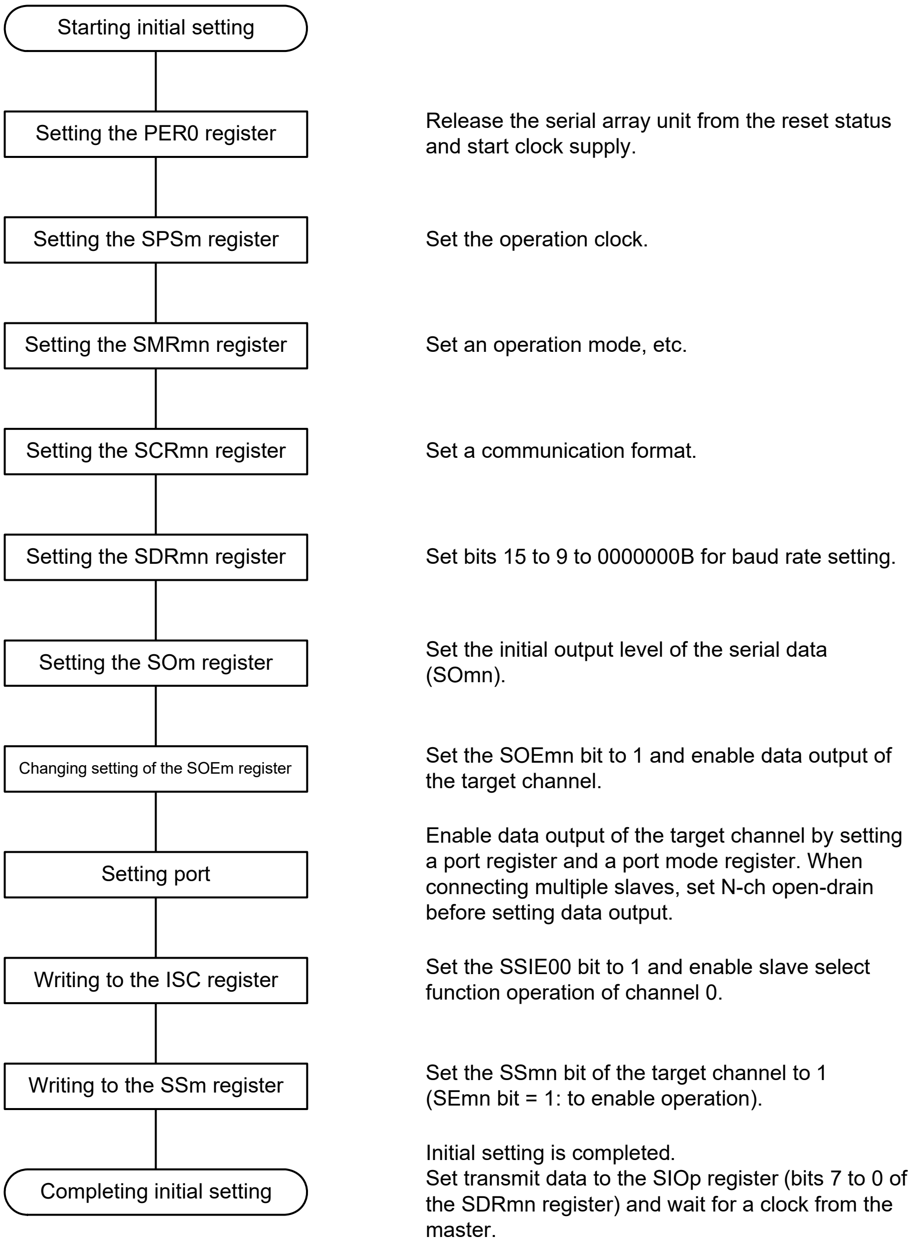
0: Disables the input value of the $\overline{\text{SSI00}}$ pin
1: Enables the input value of the $\overline{\text{SSI00}}$ pin

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Remark 2. ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

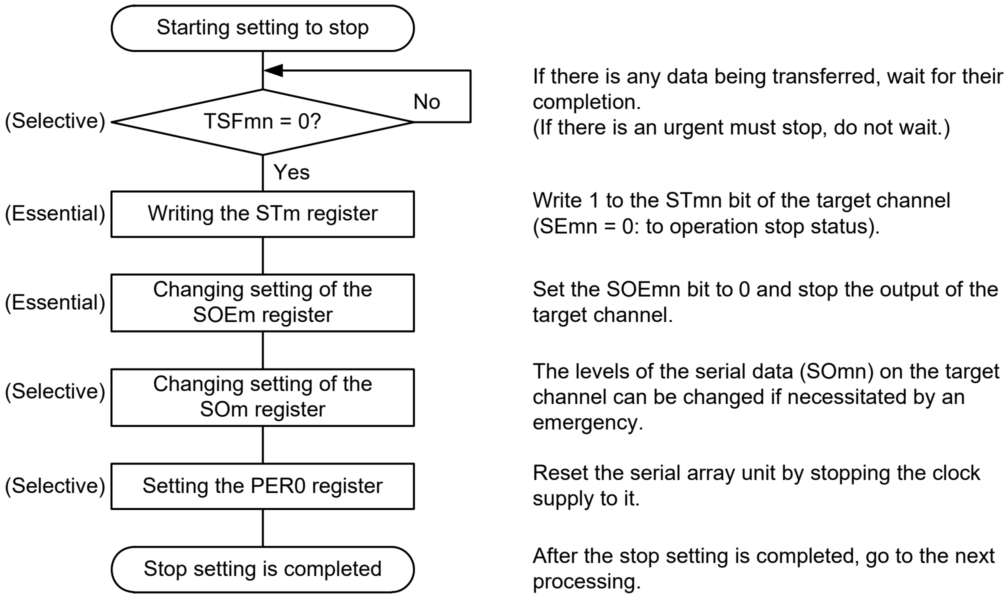
(2) Operation procedure

Figure 17 - 82 Initial Setting Procedure for Slave Transmission



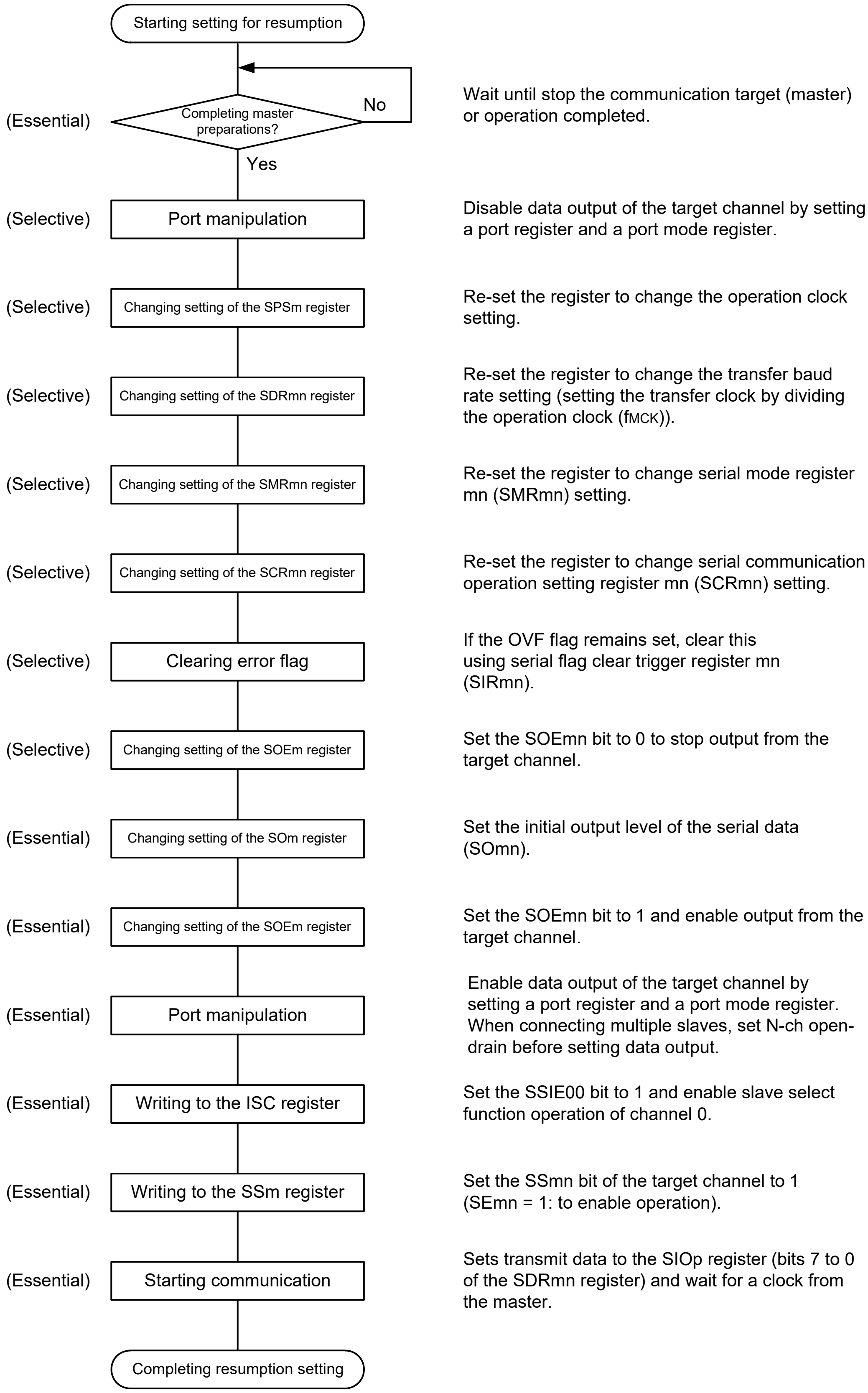
Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 17 - 83 Procedure for Stopping Slave Transmission



Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 17 - 84 Procedure for Resuming Slave Transmission

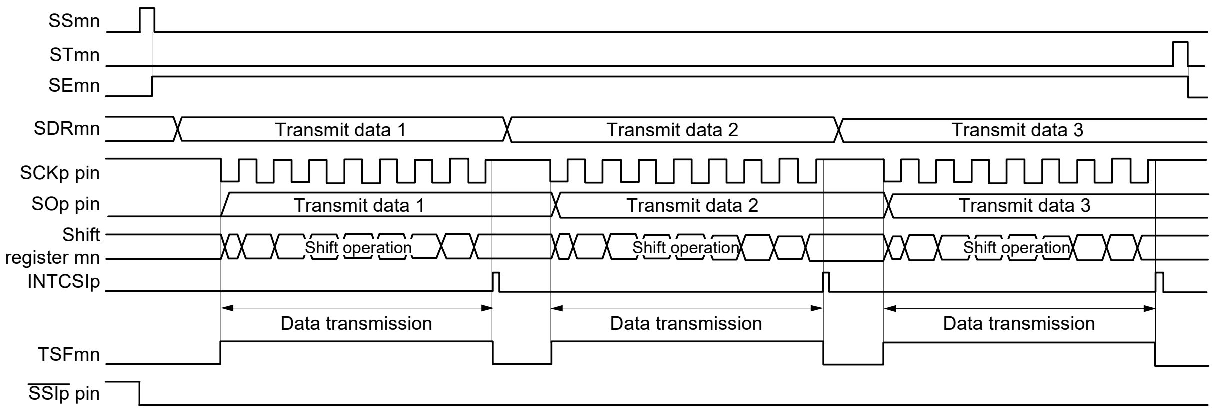


Remark 1. If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

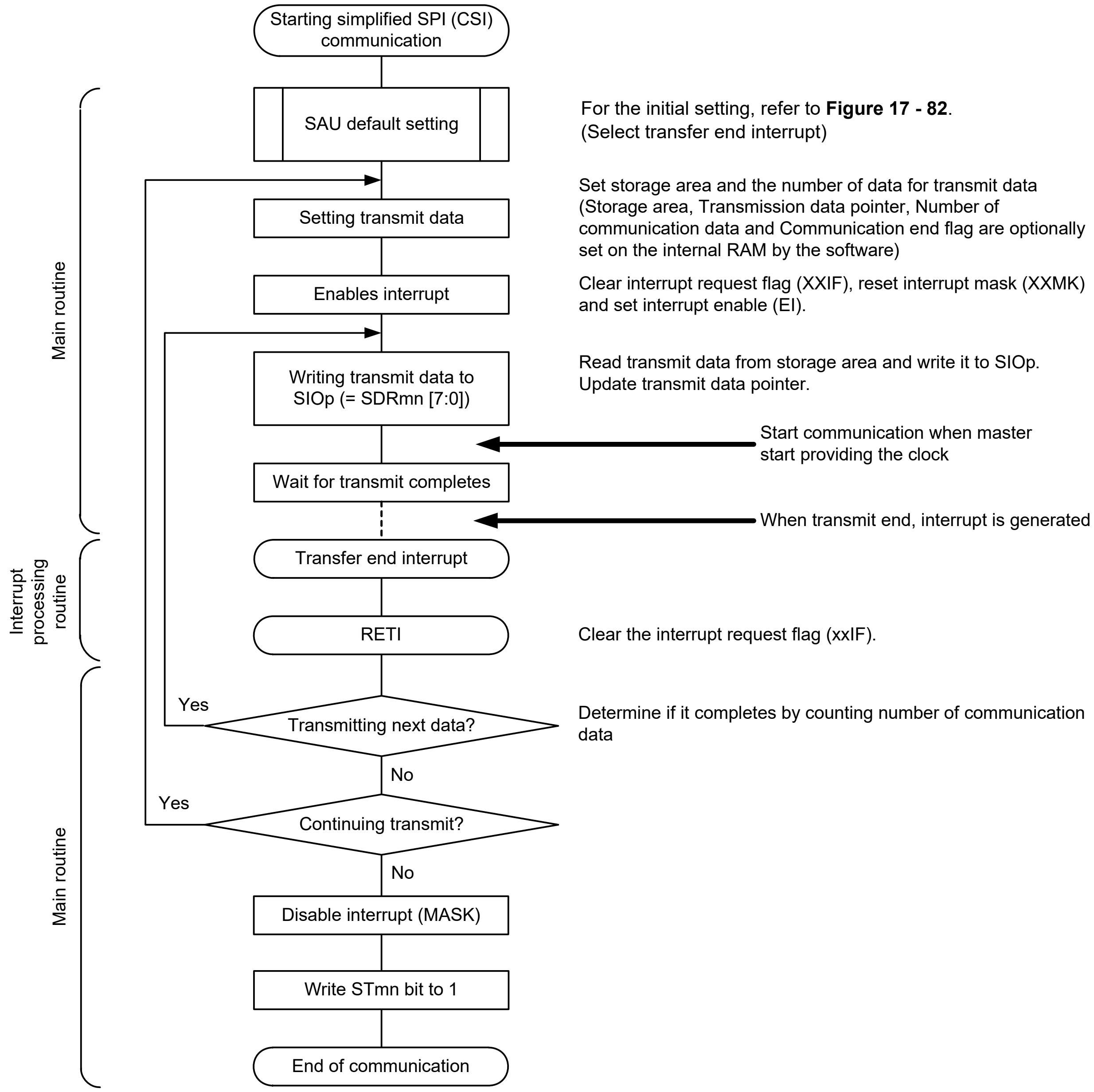
(3) Processing flow (in single-transmission mode)

Figure 17 - 85 Timing Chart of Slave Transmission (in Single-Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

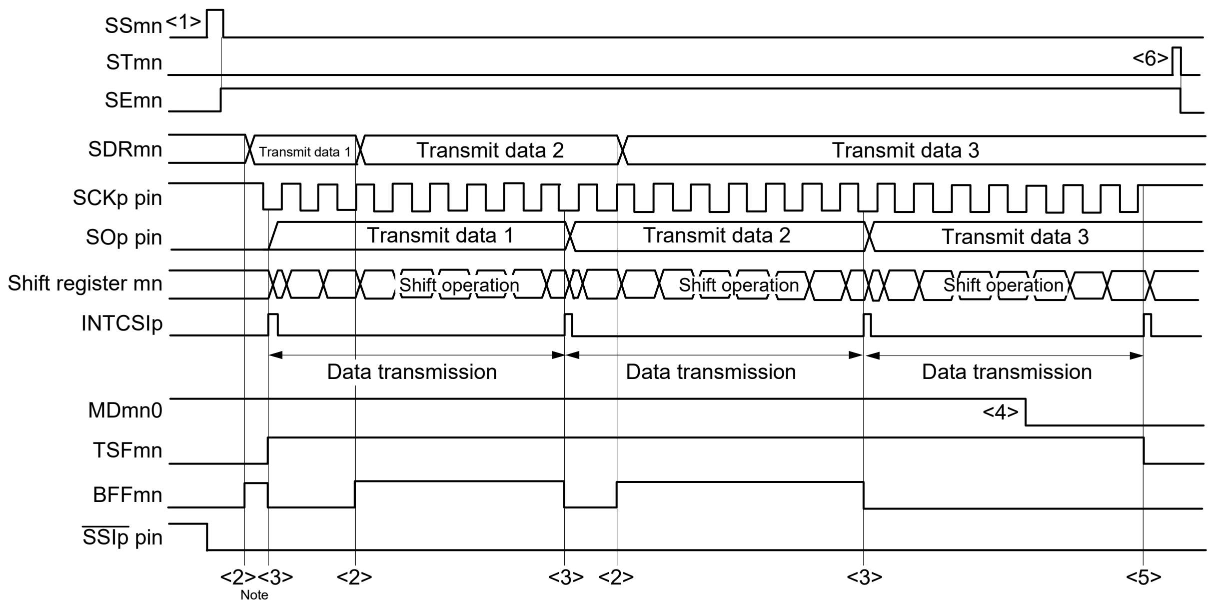
Figure 17 - 86 Flowchart of Slave Transmission (in Single-Transmission Mode)



Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

(4) Processing flow (in continuous transmission mode)

Figure 17 - 87 Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)

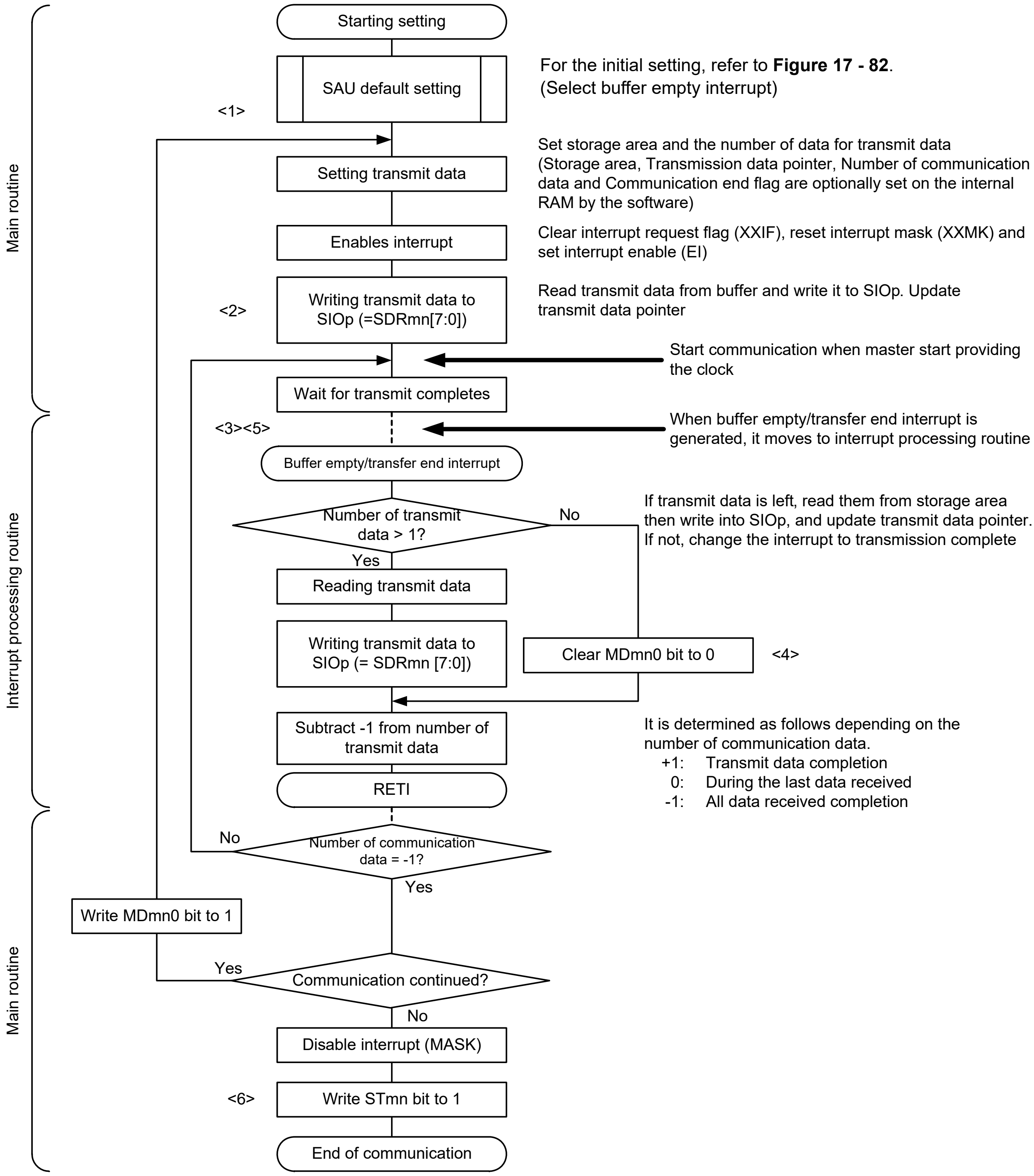


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution **The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.**

Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 17 - 88 Flowchart of Slave Transmission (in Continuous Transmission Mode)



Remark 1. <1> to <6> in the figure correspond to <1> to <6> in Figure 17 - 87 Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0).

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

17.6.2 Slave reception

Slave reception is that the RL78 microcontroller receives data from another device in the state of a transfer clock being input from another device.

Slave select input function	CSI00
Target channel	Channel 0 of SAU0
Pins used	SCK00, SI00, $\overline{\text{SSI00}}$
Interrupt	INTCSI00
	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 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
Slave select input function	Slave select input function operation selectable

Note 1. Because the external serial clock input to the SCK00 pin 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 in the electrical specifications (see **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark 1. f_{MCK}: Operation clock frequency of target channel

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0)

(1) Register setting

Figure 17 - 89 Example of Contents of Registers for Slave Reception of Slave Select Input Function (CSI00) (1/2)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0

Operation clock (fMCK) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **17.3 Registers Controlling Serial Array Unit**)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	Receive data (SIOp)							

(d) Serial output register m (SOm)... The Register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 ×	CKOm0 ×	0	0	0	0	0	0	SOm1 ×	SOm0 ×

(e) Serial output enable register m (SOEm)... The Register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 ×	SOEm0 ×

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Remark 2. ■: Setting is fixed in the simplified SPI (CSI) slave reception mode,
■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 17 - 90 Example of Contents of Registers for Slave Reception of Slave Select Input Function (CSI00) (2/2)

(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 ×	SSm0 0/1

(g) Input switch control register (ISC)... $\overline{\text{SSI00}}$ input setting in CSI00 slave channel (channel 0 of unit 0).

	7	6	5	4	3	2	1	0
ISC	SSIE00 0/1	0	0	0	0	0	0	0

0: Disables the input value of the $\overline{\text{SSI00}}$ pin
1: Enables the input value of the $\overline{\text{SSI00}}$ pin

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Remark 2. ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 17 - 91 Initial Setting Procedure for Slave Reception

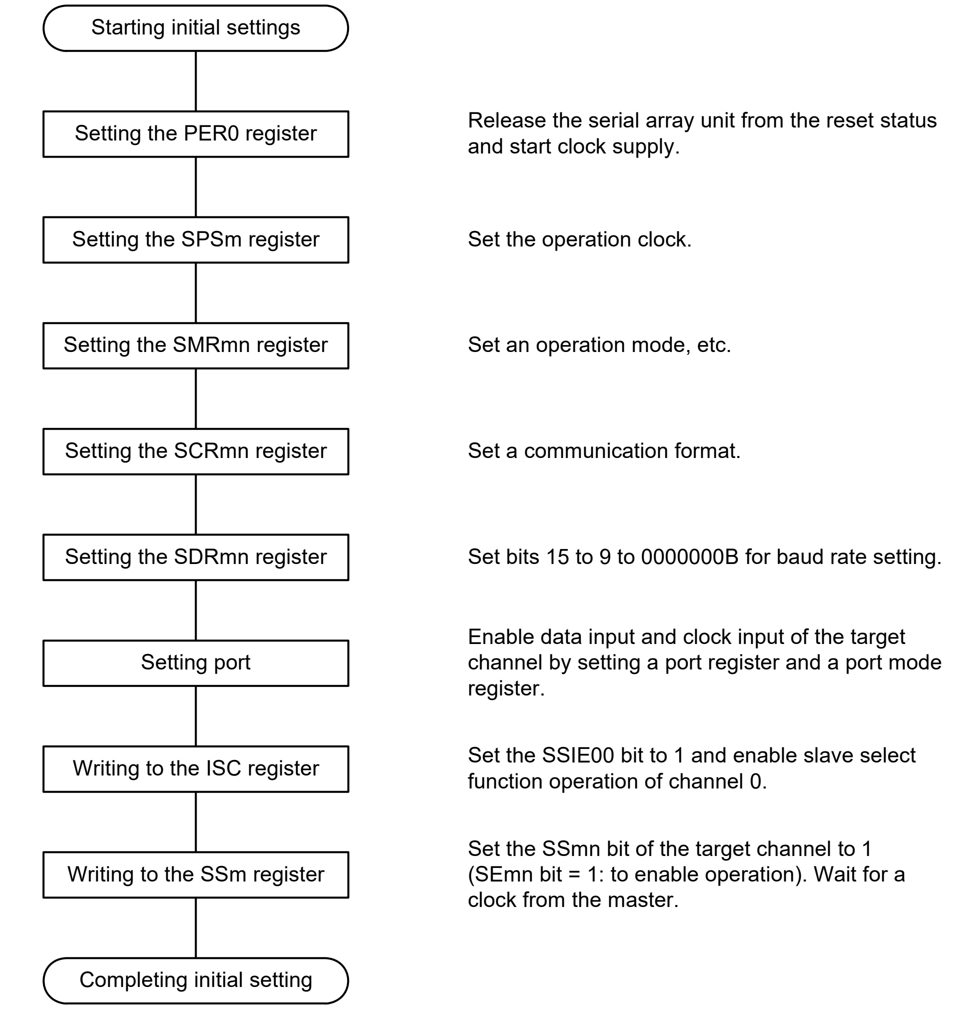
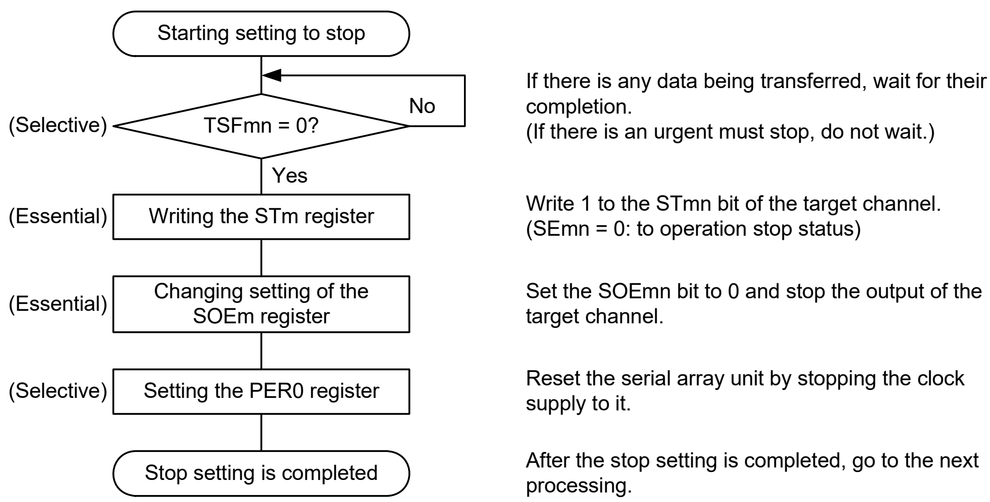
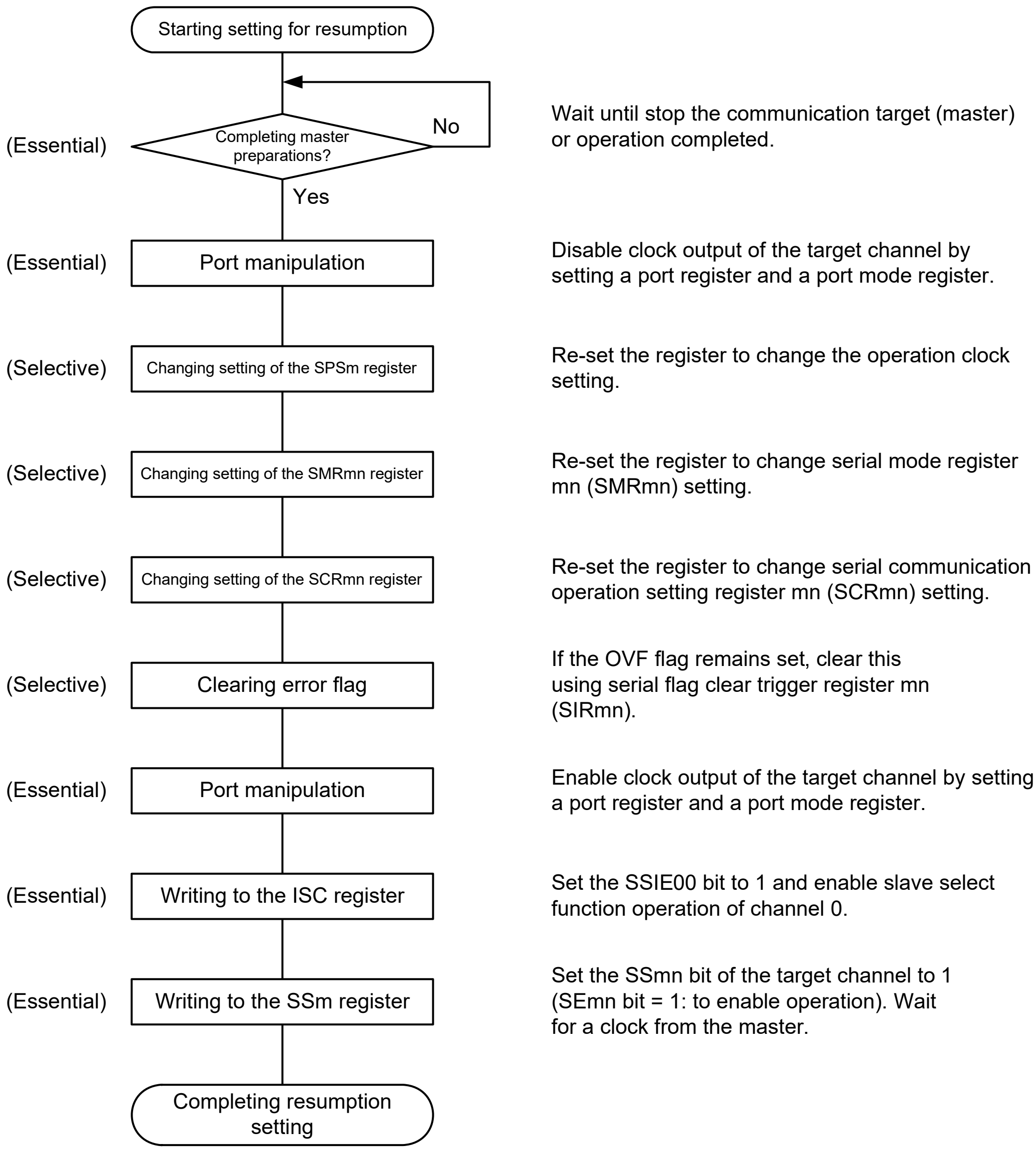


Figure 17 - 92 Procedure for Stopping Slave Reception



Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

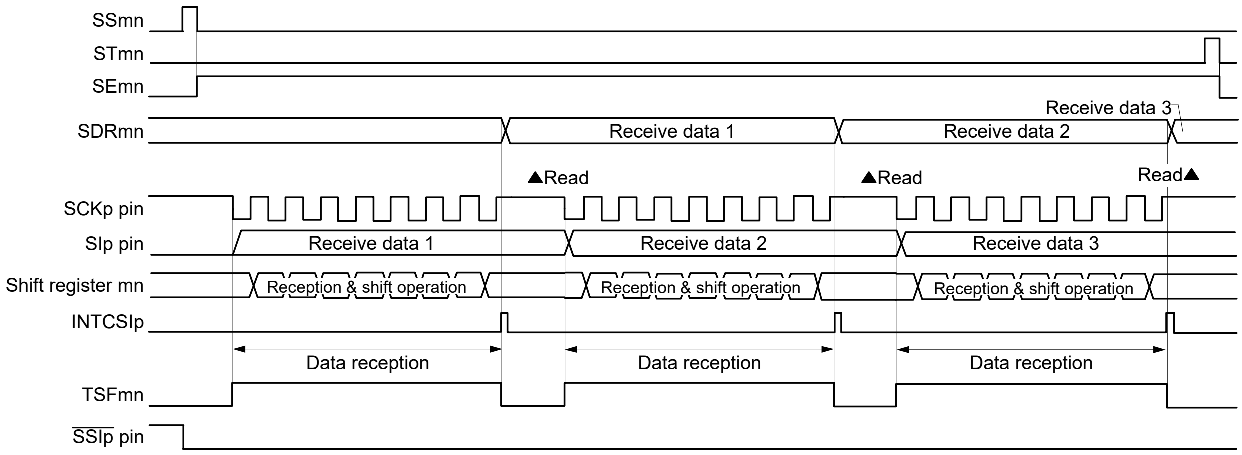
Figure 17 - 93 Procedure for Resuming Slave Reception



Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

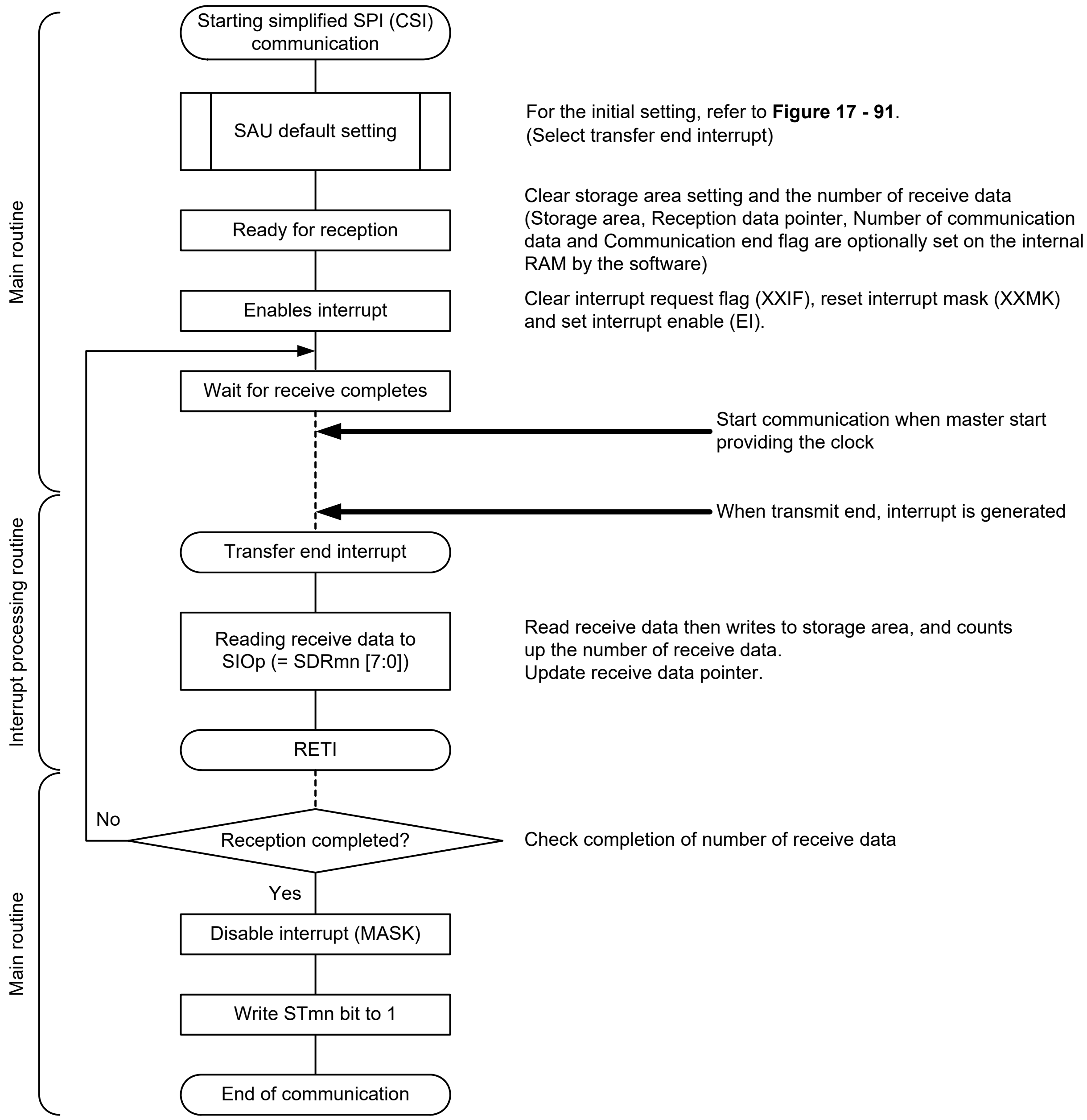
(3) Processing flow (in single-reception mode)

Figure 17 - 94 Timing Chart of Slave Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 17 - 95 Flowchart of Slave Reception (in Single-Reception Mode)



17.6.3 Slave transmission/reception

Slave transmission/reception is that the RL78 microcontroller transmits/receives data to/from another device in the state of a transfer clock being input from another device.

Slave select input function	CSI00
Target channel	Channel 0 of SAU0
Pins used	SCK00, SI00, SO00, $\overline{\text{SSI00}}$
Interrupt	INTCSI00
	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 from the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse
Data direction	MSB or LSB first
Slave select input function	Slave select input function operation selectable

Note 1. Because the external serial clock input to the SCK00 pin 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 in the electrical specifications (see **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark 1. f_{MCK}: Operation clock frequency of target channel

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0)

(1) Register setting

Figure 17 - 96 Example of Contents of Registers for Slave Transmission/Reception of Slave Select Input Function (CSI00) (1/2)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **17.3 Registers Controlling Serial Array Unit**)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	Transmit data setting/receive data register							

$\underbrace{\text{bits 7 to 0}}_{\text{SIOp}}$

(d) Serial output register m (SOm)... Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 ×	CKOm0 ×	0	0	0	0	0	0	SOm1 ×	SOm0 0/1

(e) Serial output enable register m (SOEm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 ×	SOEm0 0/1

Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Remark 2. ■: Setting is fixed in the simplified SPI (CSI) slave transmission/reception mode
▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 17 - 97 Example of Contents of Registers for Slave Transmission/Reception of Slave Select Input Function (CSI00) (2/2)

(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 ×	SSm0 0/1

(g) Input switch control register (ISC)... $\overline{\text{SSI00}}$ input setting in CSI00 slave channel (channel 0 of unit 0).

	7	6	5	4	3	2	1	0
ISC	SSIE00 0/1	0	0	0	0	0	0	0

0: Disables the input value of the $\overline{\text{SSI00}}$ pin
1: Enables the input value of the $\overline{\text{SSI00}}$ pin

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Remark 2. ▇: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

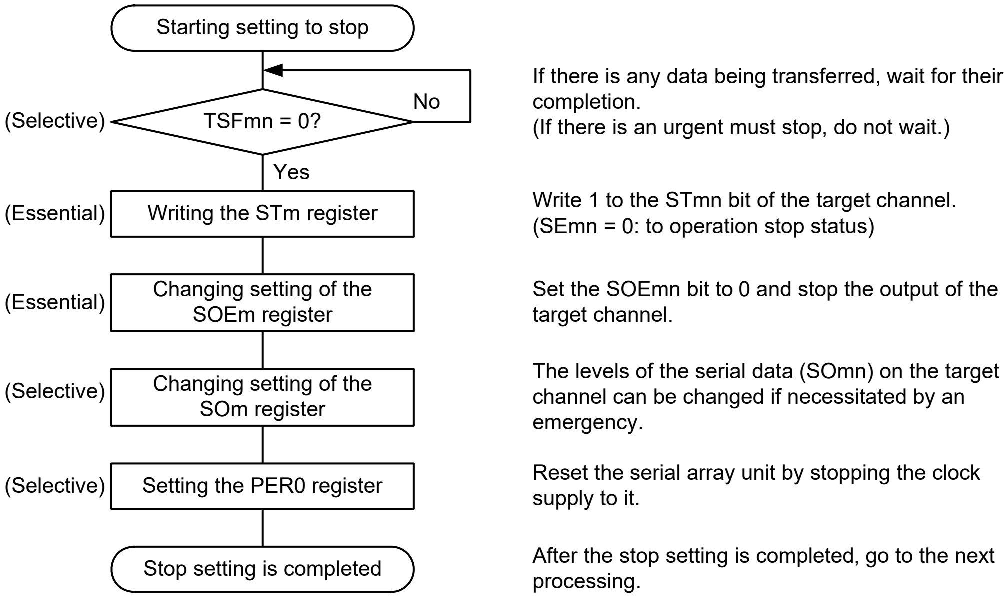
Figure 17 - 98 Initial Setting Procedure for Slave Transmission/Reception

Starting initial setting	
Setting the PER0 register	Release the serial array unit from the reset status and start clock supply.
Setting the SPSm register	Set the operation clock.
Setting the SMRmn register	Set an operation mode, etc.
Setting the SCRmn register	Set a communication format.
Setting the SDRmn register	Set bits 15 to 9 to 0000000B for baud rate setting.
Setting the SOm register	Set the initial output level of the serial data (SOmn).
Changing setting of the SOEm register	Set the SOEmn bit to 1 and enable data output of the target channel.
Setting port	Enable data output of the target channel by setting a port register and a port mode register. When connecting multiple slaves, set N-ch open-drain before setting data output.
Writing to the ISC register	Set the SSIE00 bit to 1 and enable slave select function operation of channel 0.
Writing to the SSm register	Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation).
Completing initial setting	Initial setting is completed. Set transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and wait for a clock from the master.

Caution **Be sure to set transmit data to the SIOp register before the clock from the master is started.**

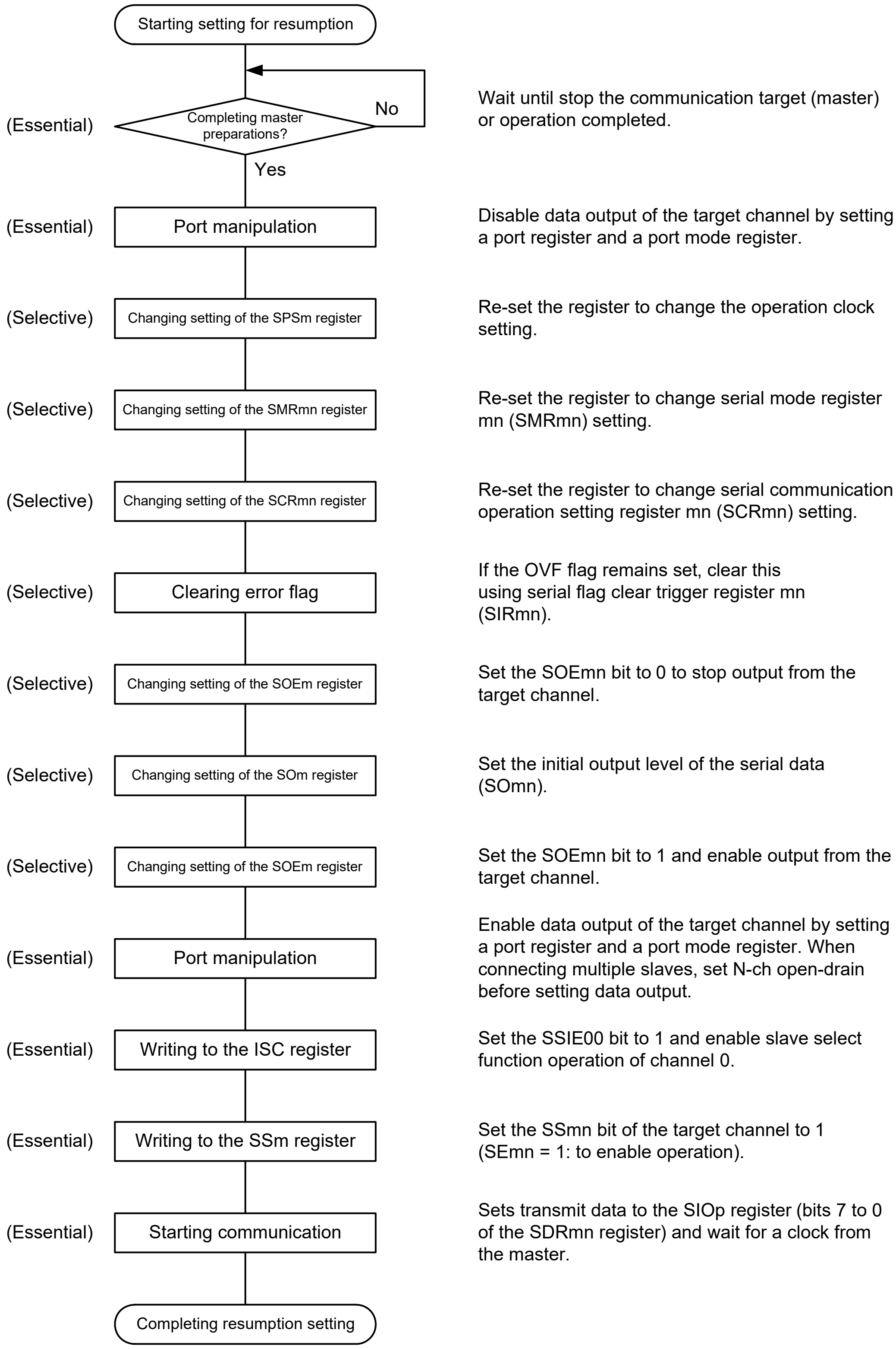
Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 17 - 99 Procedure for Stopping Slave Transmission/Reception



Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 17 - 100 Procedure for Resuming Slave Transmission/Reception

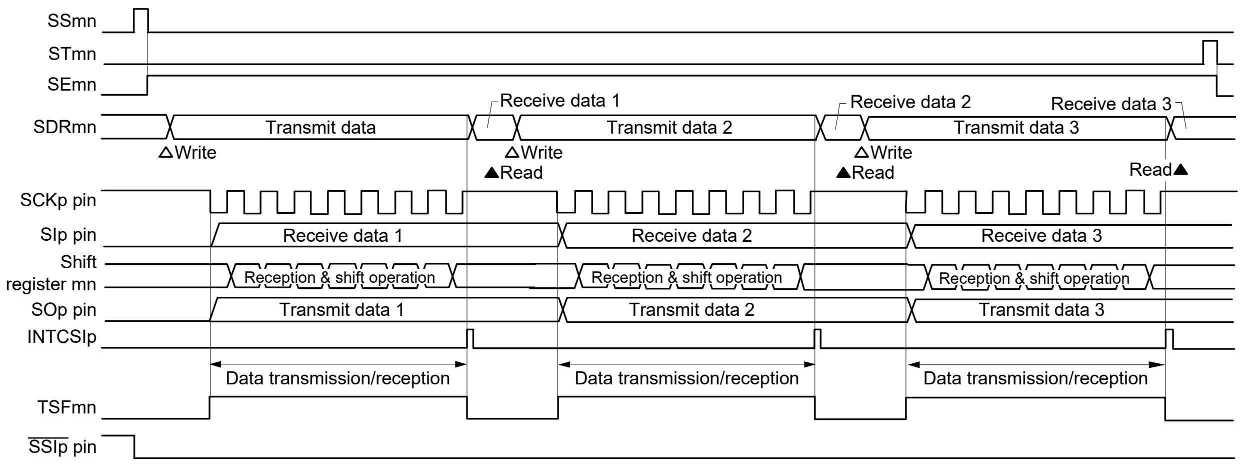


Caution 1. Be sure to set transmit data to the SIOp register before the clock from the master is started.

Caution 2. If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

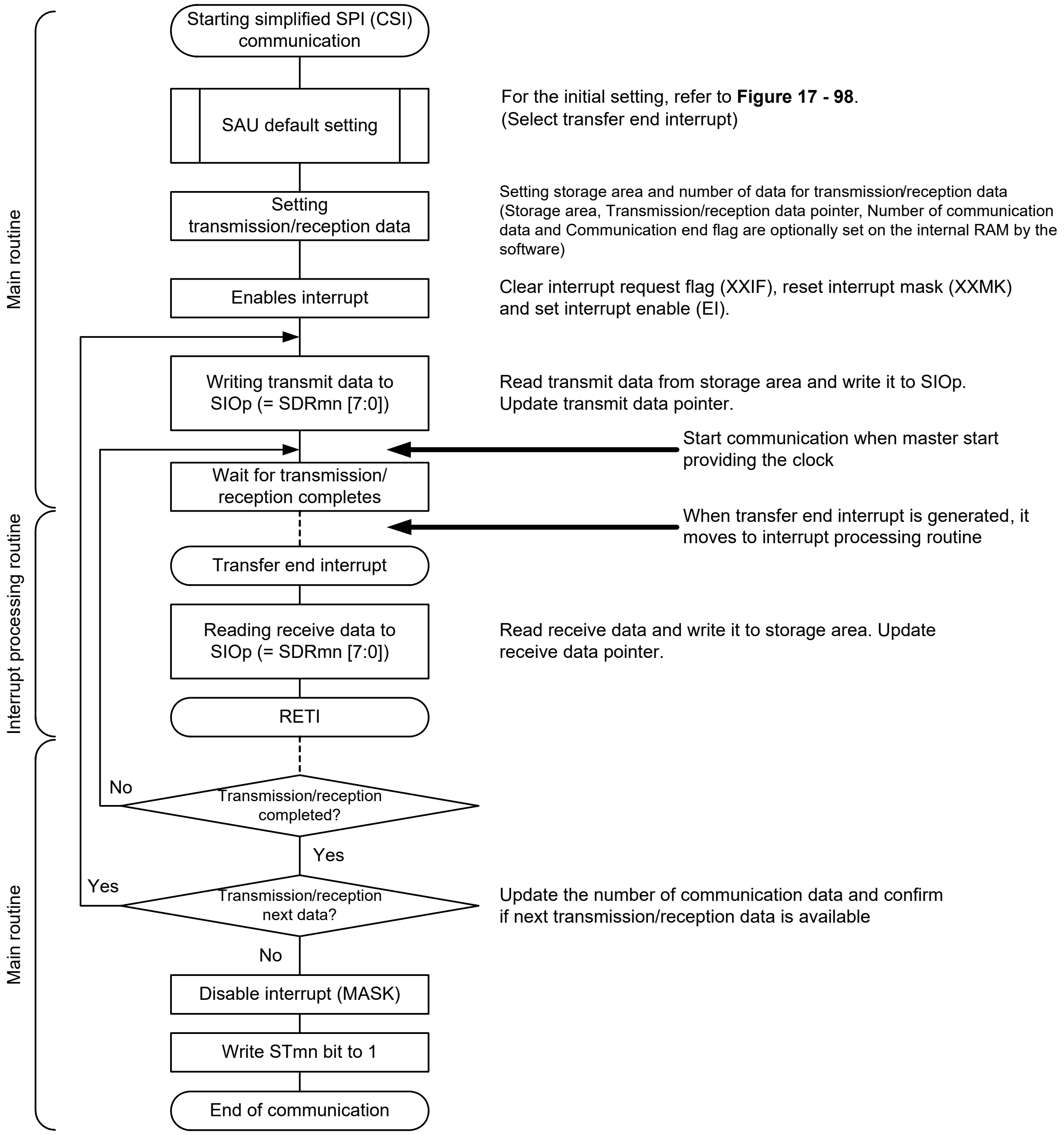
(3) Processing flow (in single-transmission/reception mode)

Figure 17 - 101 Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 17 - 102 Flowchart of Slave Transmission/Reception (in Single- Transmission/Reception Mode)

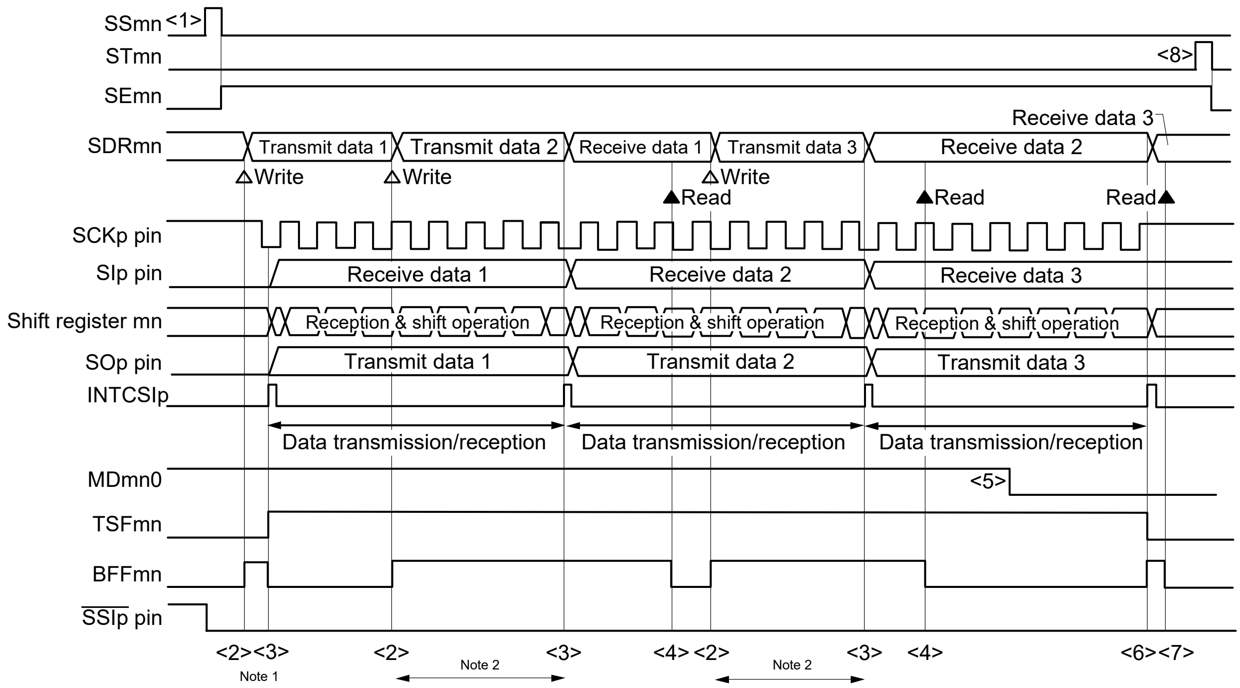


Caution **Be sure to set transmit data to the SIOp register before the clock from the master is started.**

Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

(4) Processing flow (in continuous transmission/reception mode)

Figure 17 - 103 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.

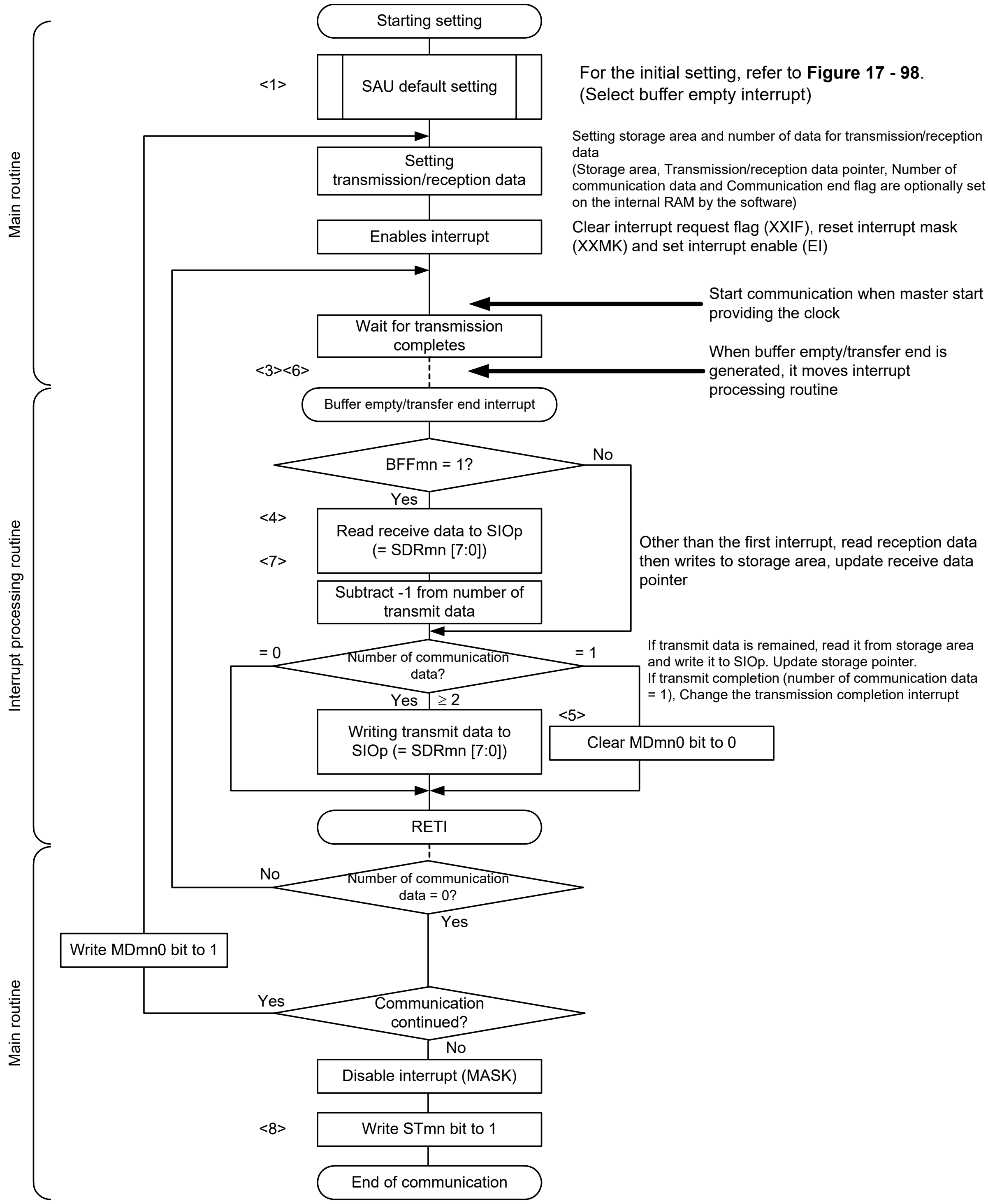
Note 2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution **The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.**

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in Figure 17 - 104 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode).

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 17 - 104 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. <1> to <8> in the figure correspond to <1> to <8> in Figure 17 - 103 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

17.6.4 Calculating transfer clock frequency

The transfer clock frequency for slave select input function (CSI00) communication can be calculated by the following expressions.

(1) 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 m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Table 17 - 3 Selection of Operation Clock For Slave Select Input Function

SMRmn Register	SPSm Register								Operation Clock (f_{MCK}) Note	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 24 MHz
0	×	×	×	×	0	0	0	0	f_{CLK}	24 MHz
	×	×	×	×	0	0	0	1	$f_{CLK}/2$	12 MHz
	×	×	×	×	0	0	1	0	$f_{CLK}/2^2$	6 MHz
	×	×	×	×	0	0	1	1	$f_{CLK}/2^3$	3 MHz
	×	×	×	×	0	1	0	0	$f_{CLK}/2^4$	1.5 MHz
	×	×	×	×	0	1	0	1	$f_{CLK}/2^5$	750 kHz
	×	×	×	×	0	1	1	0	$f_{CLK}/2^6$	375 kHz
	×	×	×	×	0	1	1	1	$f_{CLK}/2^7$	187.5 kHz
	×	×	×	×	1	0	0	0	$f_{CLK}/2^8$	93.8 kHz
	×	×	×	×	1	0	0	1	$f_{CLK}/2^9$	46.9 kHz
	×	×	×	×	1	0	1	0	$f_{CLK}/2^{10}$	23.4 kHz
	×	×	×	×	1	0	1	1	$f_{CLK}/2^{11}$	11.7 kHz
	×	×	×	×	1	1	0	0	$f_{CLK}/2^{12}$	5.86 kHz
	×	×	×	×	1	1	0	1	$f_{CLK}/2^{13}$	2.93 kHz
	×	×	×	×	1	1	1	0	$f_{CLK}/2^{14}$	1.46 kHz
	×	×	×	×	1	1	1	1	$f_{CLK}/2^{15}$	732 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).

Remark 1. ×: Don't care

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0)

17.6.5 Procedure for processing errors that occurred during slave select input function communication

The procedure for processing errors that occurred during slave select input function communication is described in Figure 17 - 105.

Figure 17 - 105 Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

Remark m: Unit number (m = 0), n: Channel number (n = 0)

17.7 Operation of UART (UART0) Communication

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel).

[Data transmission/reception]
• Data length of 7, 8, or 9 bits
• Select the MSB/LSB first
• 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

In addition, UART0 supports the SNOOZE mode. When RxD input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible. Only the following UART can be specified.
• UART0

UART0 uses channels 0 and 1 of SAU0.

• 20-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	—		—

• 24, 32-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01

• 30-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	—		—

• 48-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00 (supporting slave select input function)	UART0	IIC00
	1	CSI01		IIC01

Select any function for each channel. Only the selected function is possible. If UART0 is selected for channels 0 and 1 of unit 0, for example, these channels cannot be used for CSI00 and IIC00.

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 **17.7.1**.)

• UART reception (See **17.7.2**.)

17.7.1 UART transmission

UART transmission is an operation to transmit data from the RL78 microcontroller to another device asynchronously (start-stop synchronization).

Of two channels used for UART, the even channel is used for UART transmission.

UART	UART0
Target channel	Channel 0 of SAU0
Pins used	TxD0
Interrupt	INTST0
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	None
Transfer data length	7, 8, or 9 bits Note 1
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDRmn [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps] Note 2
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 bits
Data direction	MSB or LSB first

Note 1. Only the following UARTs can be specified for the 9-bit data length.
• UART0

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 34 ELECTRICAL SPECIFICATIONS**). For UART transmission, use the high-speed system clock, subsystem clock, or high-speed on-chip oscillator.

Remark 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0), mn = 00

(1) Register setting

Figure 17 - 106 Example of Contents of Registers for UART Transmission of UART (UART0) (1/2)

(a) Serial mode register mn (SMRmn)

	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	0	0	0	1	0	0	MDmn2 0	MDmn1 1	MDmn0 0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	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 0/1	PTCmn0 0/1	DIRmn 0/1	0	SLCmn1 0/1	SLCmn0 0/1	0	1	DLSmn1 0/1 Note 1	DLSmn0 0/1

Setting of parity bit
00B: No parity
01B: Appending 0 parity
10B: Appending Even parity
11B: Appending Odd parity

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of stop bit
01B: Appending 1 bit
10B: Appending 2 bits

(c) Serial data register mn (SDRmn) (lower 8 bits: TXDq)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0 Note 2	Transmit data setting							

$\underbrace{\text{bits 7 to 0}}_{\text{TXDq}}$

(d) Serial output level register m (SOLm)... Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOLm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOLm0 0/1

0: Non-reverse (normal) transmission
1: Reverse transmission

Note 1. Only provided for the SCR00 register. This bit is fixed to 1 for the other registers.

Note 2. When performing 9-bit communication, bits 0 to 8 of the SDRm0 register are used to specify the transmission data.
• UART0

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), q: UART number (q = 0), mn = 00?

Remark 2. ■: Setting is fixed in the UART transmission mode,
■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 17 - 107 Example of Contents of Registers for UART Transmission of UART (UART0) (2/2)

(e) Serial output register m (SOm)... Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 ×	CKOm0 ×	0	0	0	0	0	0	SOm1 ×	SOm0 0/1 Note

0: Serial data output value is "0"
1: Serial data output value is "1"

(f) Serial output enable register m (SOEm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 ×	SOEm0 0/1

(g) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 ×	SSm0 0/1

Note Before transmission is started, be sure to set to 1 when the SOLmn bit of the target channel is set to 0, and set to 0 when the SOLmn bit of the target channel is set to 1. The value varies depending on the communication data during communication operation.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), q: UART number (q = 0), mn = 00

Remark 2. ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 17 - 108 Initial Setting Procedure for UART Transmission

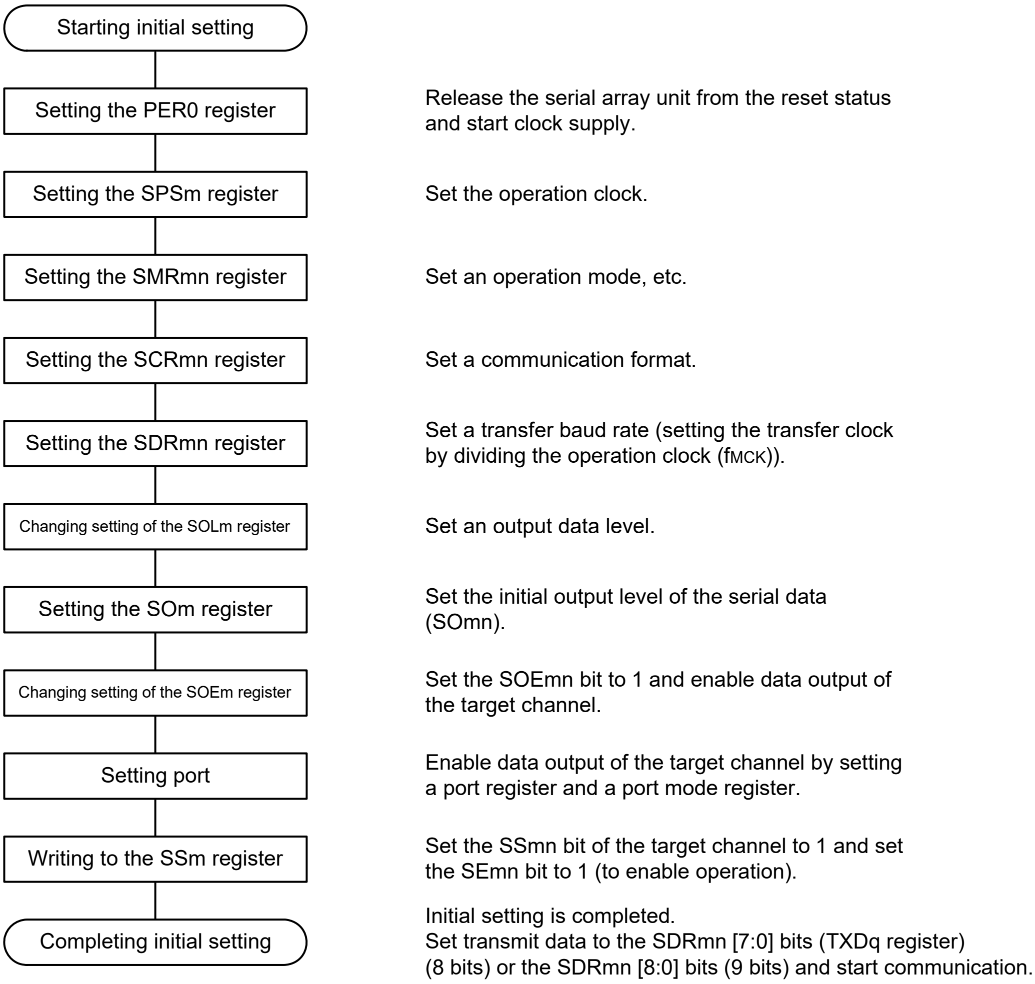


Figure 17 - 109 Procedure for Stopping UART Transmission

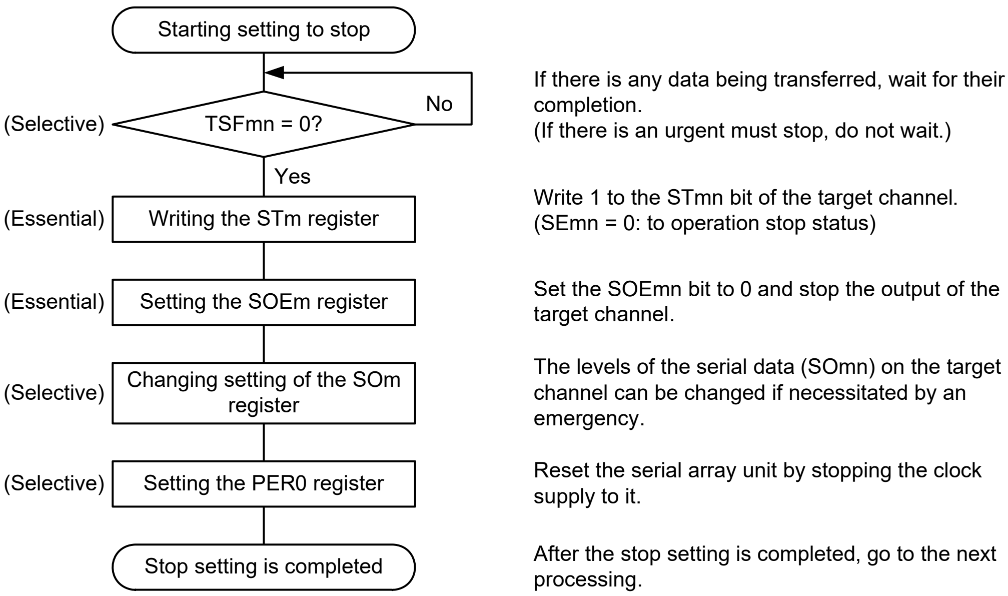
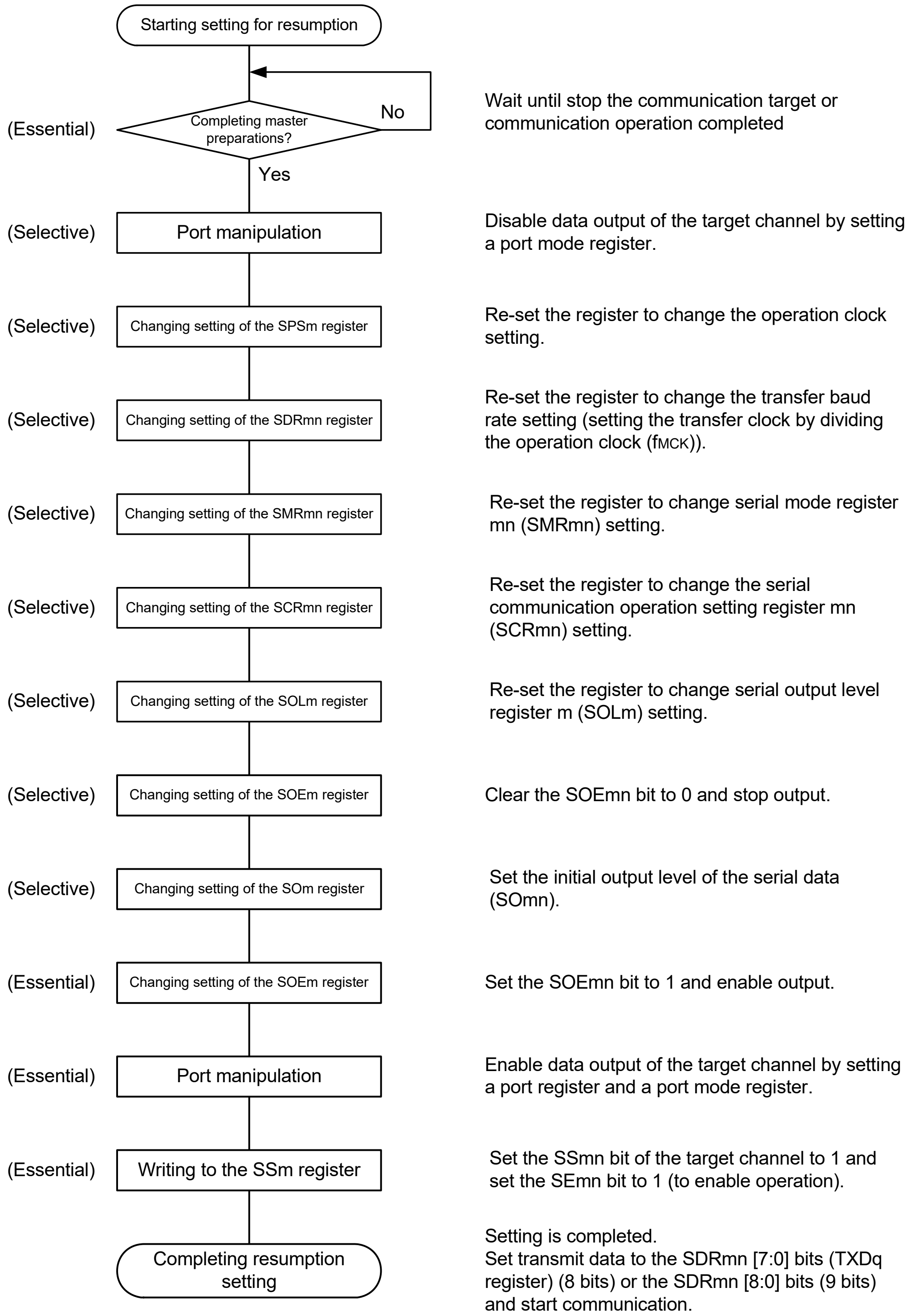


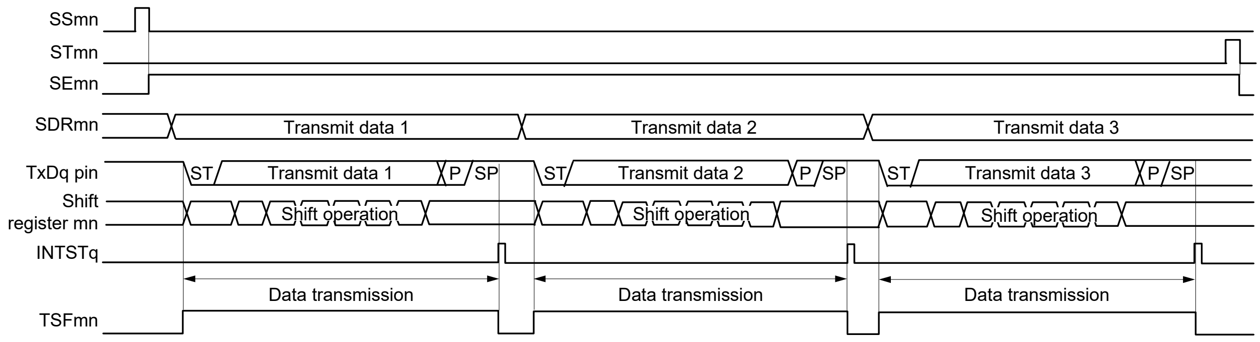
Figure 17 - 110 Procedure for Resuming UART Transmission



Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target stops or transmission finishes, and then perform initialization instead of restarting the transmission.

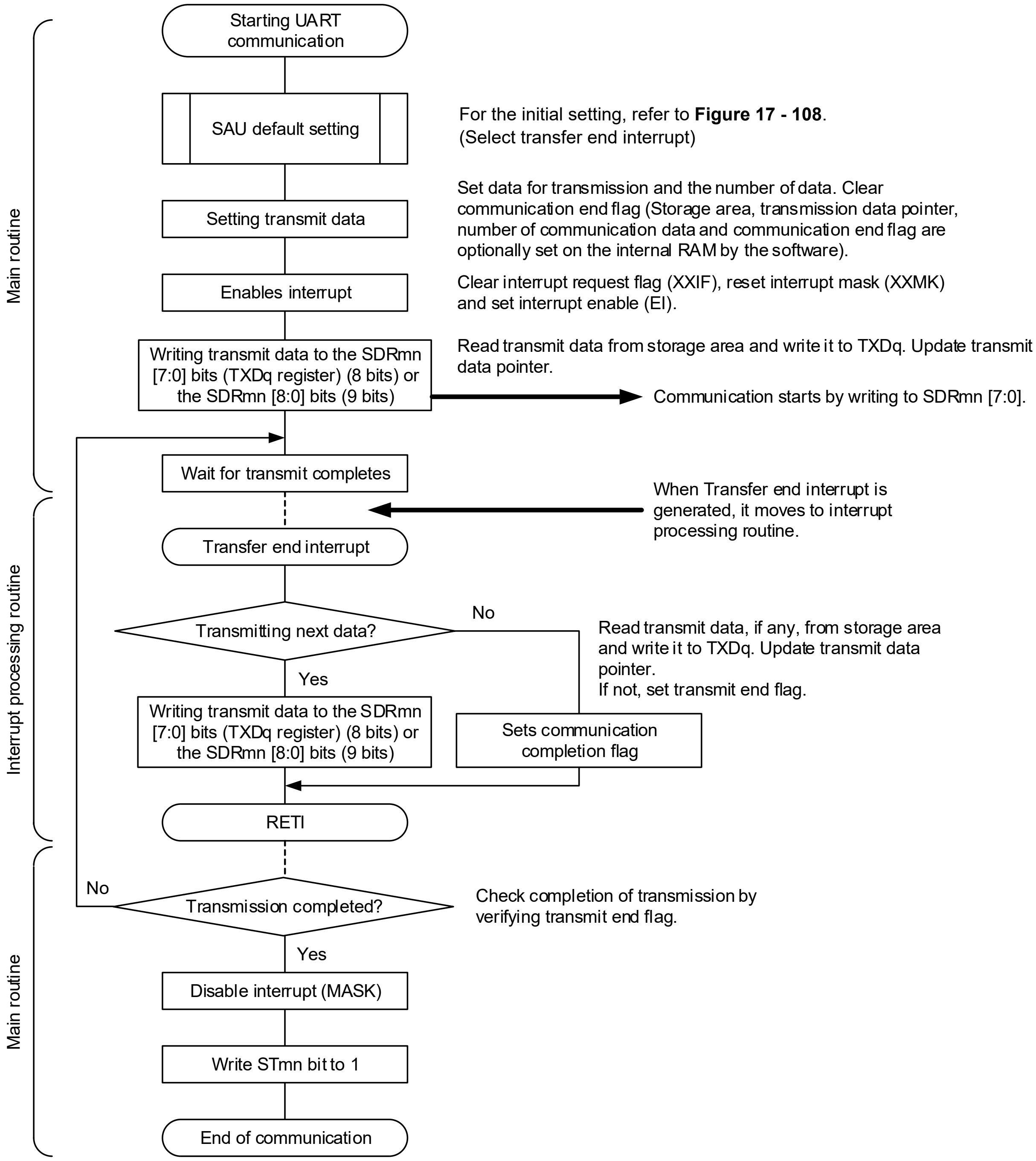
(3) Processing flow (in single-transmission mode)

Figure 17 - 111 Timing Chart of UART Transmission (in Single-Transmission Mode)



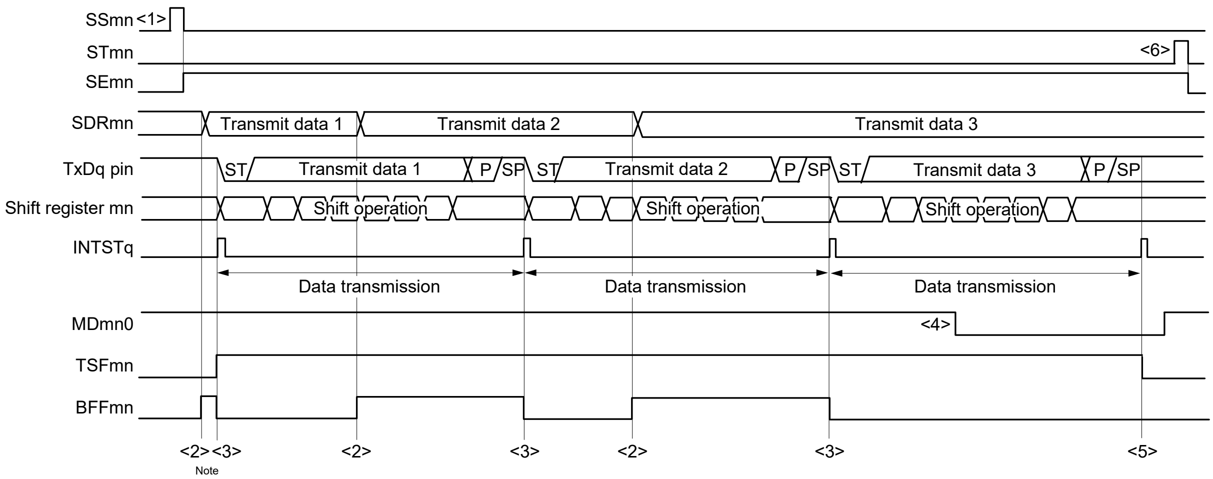
Remark m: Unit number (m = 0), n: Channel number (n = 0), q: UART number (q = 0), mn = 00

Figure 17 - 112 Flowchart of UART Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 17 - 113 Timing Chart of UART Transmission (in Continuous Transmission Mode)

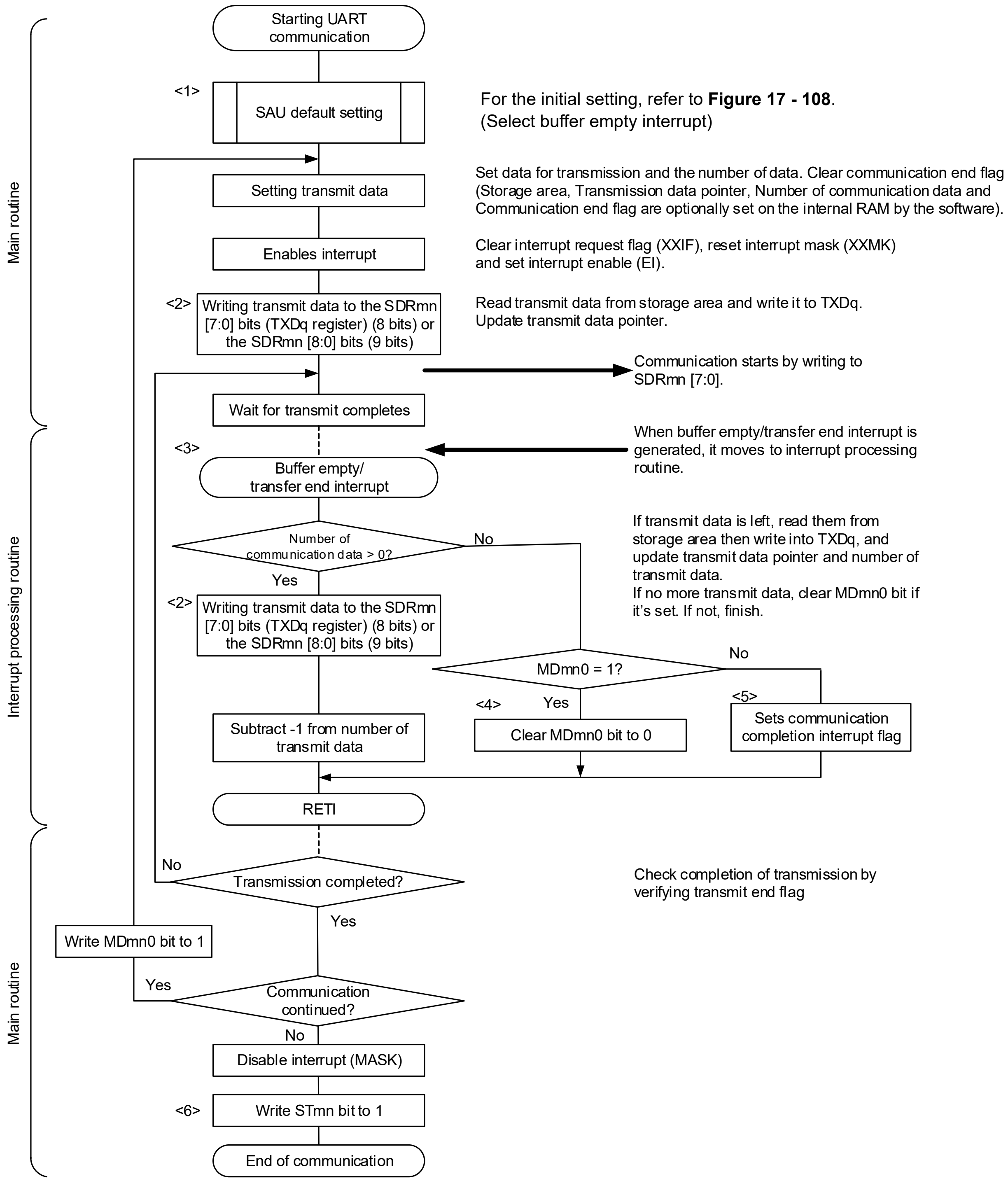


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SSRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), q: UART number (q = 0), mn = 00

Figure 17 - 114 Flowchart of UART Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in Figure 17 - 113 Timing Chart of UART Transmission (in Continuous Transmission Mode).

17.7.2 UART reception

UART reception is an operation wherein the RL78 microcontroller 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
Target channel	Channel 1 of SAU0
Pins used	RxD0
Interrupt	INTSR0
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error interrupt	INTSRE0
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) • Appending 0 parity (no parity check) • Appending even parity • Appending odd parity
Stop bit	Appending 1 bit
Data direction	MSB or LSB first

Note 1. Only the following UARTs can be specified for the 9-bit data length.
• UART0

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 34 ELECTRICAL SPECIFICATIONS**). For UART reception, use the high-speed system clock, subsystem clock, or high-speed on-chip oscillator.

Remark 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

Remark 2. m: Unit number (m = 0), n: Channel number (n = 1), mn = 01

(1) Register setting

Figure 17 - 115 Example of Contents of Registers for UART Reception of UART (UART0) (1/2)

(a) Serial mode register mn (SMRmn)

	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 1	0	SISmn0 0/1	1	0	0	MDmn2 0	MDmn1 1	MDmn0 0

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

0: Forward (normal) reception
1: Reverse reception

Operation mode of channel n
0: Transfer end interrupt

(b) Serial mode register mr (SMRmr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmr	CKSmr 0/1	CCSmr 0	0	0	0	0	0	0	0	0	1	0	0	MDmr2 0	MDmr1 1	MDmr0 0/1

Same setting value as CKSmn bit

Operation mode of channel r
0: Transfer end interrupt
1: Buffer empty interrupt

(c) Serial communication operation setting register mn (SCRmn)

	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 1	PTCmn1 0/1	PTCmn0 0/1	DIRmn 0/1	0	SLCmn1 0	SLCmn0 1	0	1	DLSmn1 0/1 Note 1	DLSmn0 0/1

Setting of parity bit
00B: No parity
01B: No parity judgment
10B: Appending Even parity
11B: Appending Odd parity

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length

(d) Serial data register mn (SDRmn) (lower 8 bits: RXDq)

	15–9	8	7–0
SDRmn	Baud rate setting	0 Note 2	Receive data register (RXDq)

Note 1. Only provided for the SCR01 register. This bit is fixed to 1 for the other registers.

Note 2. When performing 9-bit communication, bits 0 to 8 of the SDRm1 register are used to specify the receive data. Only the following UART can be specified for the 9-bit data length.
• UART0

Caution For the UART reception, be sure to set the SMRmr register of channel r that is to be paired with channel n.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 1), mn = 01, r: Channel number (r = n - 1), q: UART number (q = 0)

Remark 2. ■: Setting is fixed in the UART reception mode,
▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 17 - 116 Example of Contents of Registers for UART Reception of UART (UART0) (2/2)

(e) Serial output register m (SOm)... The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	CKOm1 ×	CKOm0 ×	0	0	0	0	0	0	SOm1 ×	SOm0 ×

(f) Serial output enable register m (SOEm)... The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 ×	SOEm0 ×

(g) Serial channel start register m (SSm)... Sets only the bits of the target channel is 1.

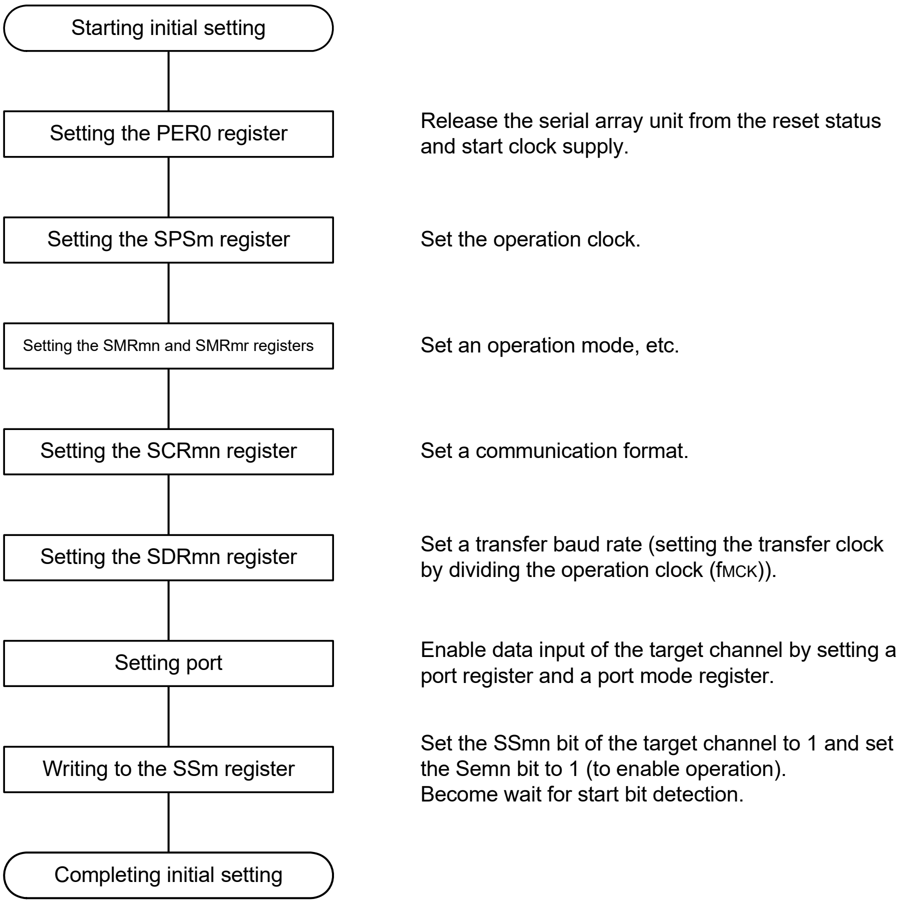
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 0/1	SSm0 ×

Remark 1. m: Unit number (m = 0)

Remark 2. ▇: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 17 - 117 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 4 or more f_{MCK} clocks have elapsed.**

Figure 17 - 118 Procedure for Stopping UART Reception

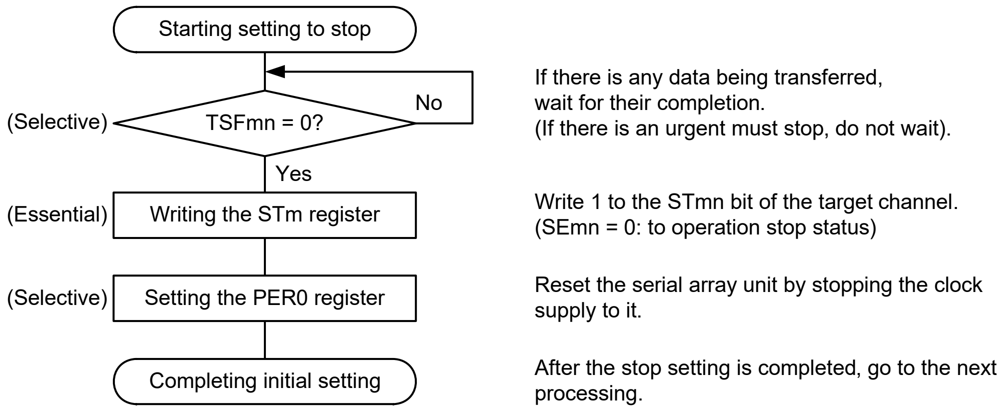
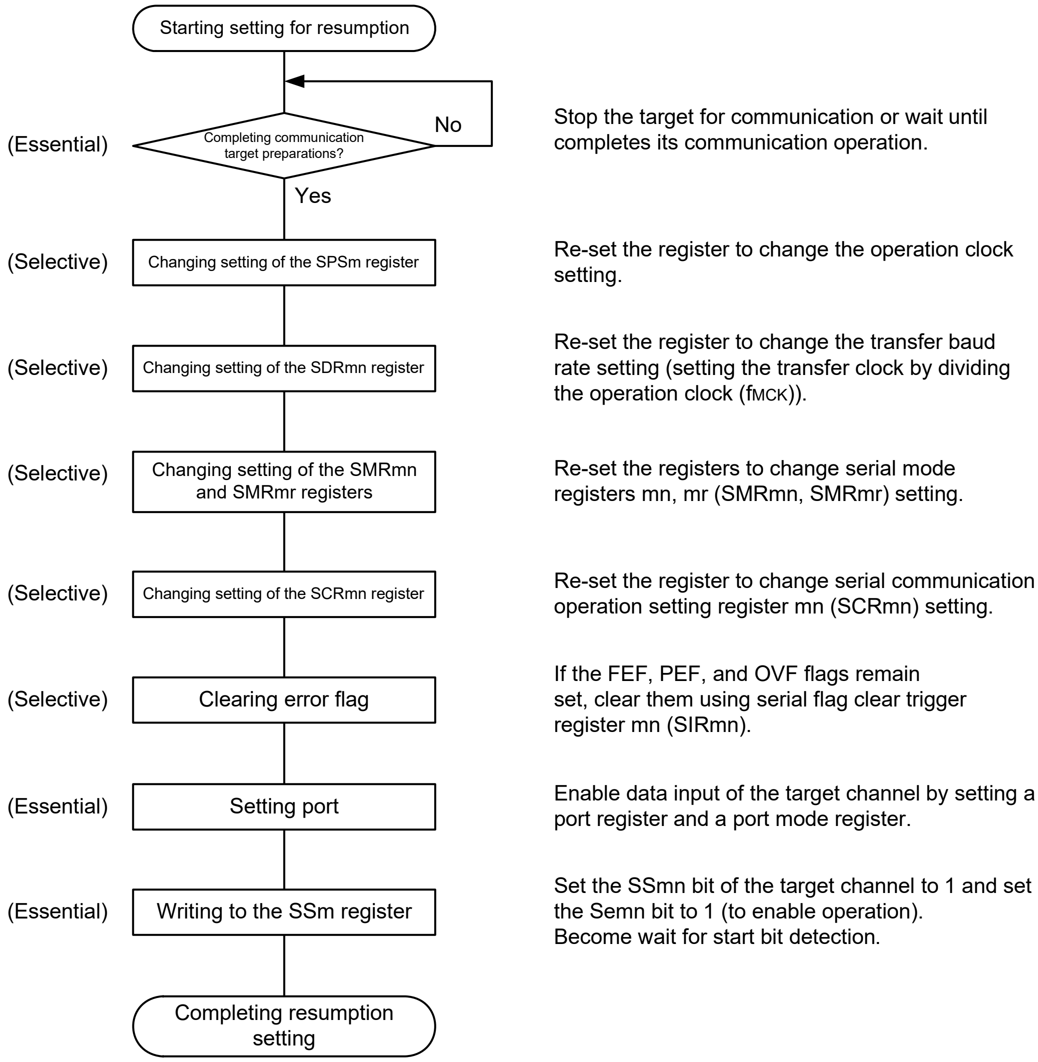


Figure 17 - 119 Procedure for Resuming UART Reception

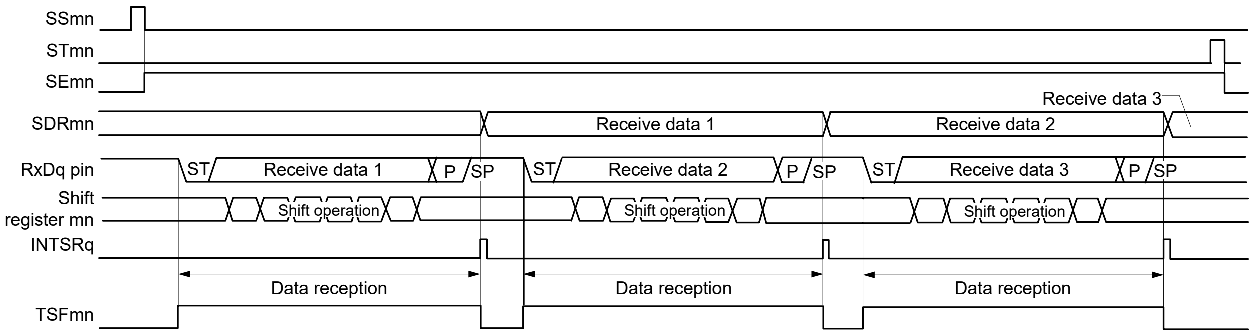


Caution **After is set RXEmn bit to 1 of SCRmn register, set the SSmn = 1 from an interval of at least four clocks of f_{MCK}.**

Remark If PER0 is rewritten while stopping the communication target and the clock supply is stopped, wait until the communication target stops or communication finishes, and then perform initialization instead of restarting the communication.

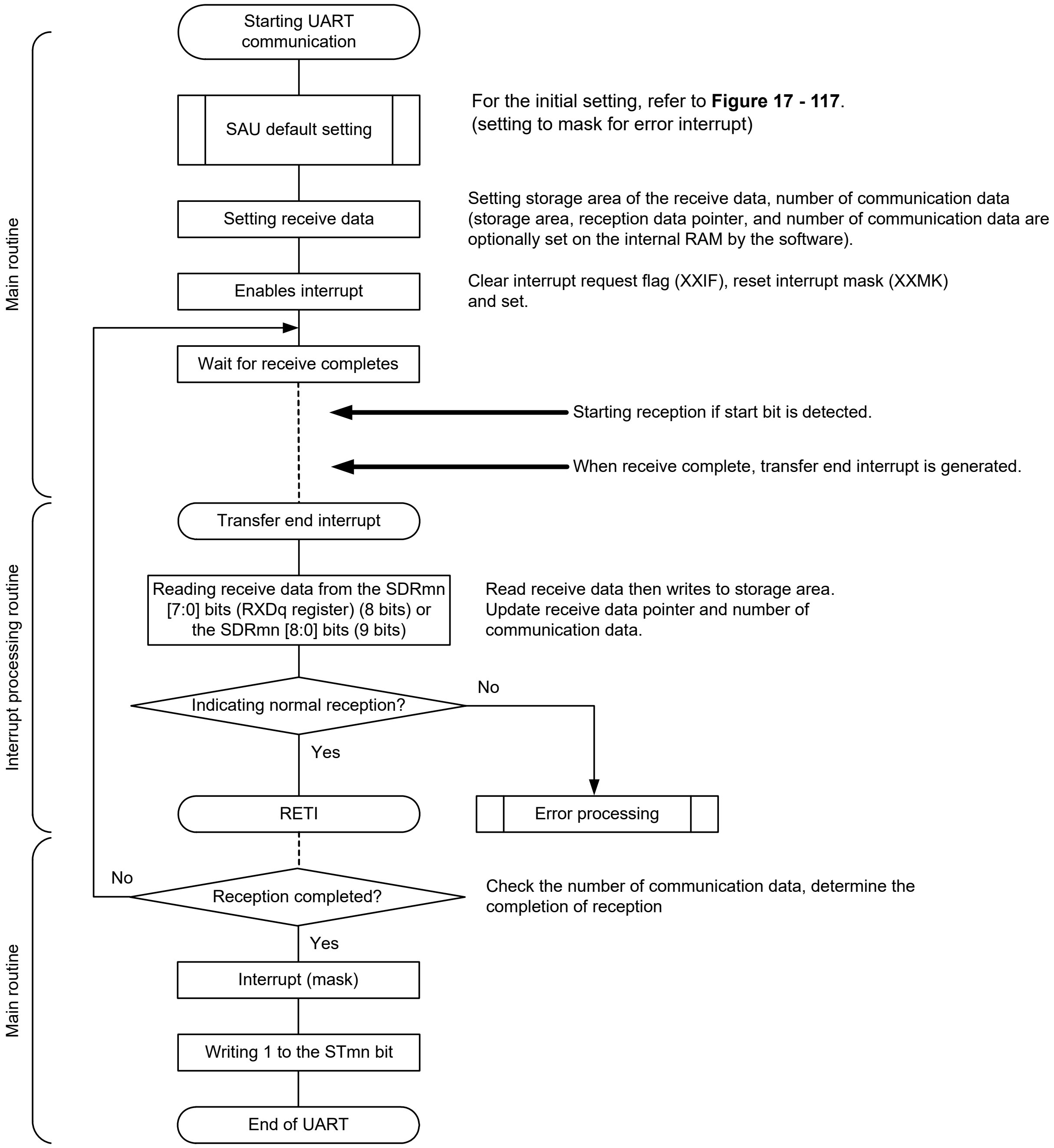
(3) Processing flow

Figure 17 - 120 Timing Chart of UART Reception



Remark m: Unit number (m = 0), n: Channel number (n = 1), mn = 01, r: Channel number (r = n - 1), q: UART number (q = 0)

Figure 17 - 121 Flowchart of UART Reception



17.7.3 SNOOZE mode function

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. Only the following UART can be specified.

• UART0

When using UARTq in the SNOOZE mode, make the following settings before entering the STOP mode (See **Figures 17 - 124** and **17 - 126 Flowchart of SNOOZE Mode Operation**).

• 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 17 - 4.
• Set the EOCmn and SSECmn bits. This is for enabling or stopping generation of an error interrupt (INTSRE0) 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 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. SNOOZE mode can be used only when the high-speed on-chip oscillator clock (f_{IH}) is selected as f_{CLK}.

Caution 2. The transfer rate in the SNOOZE mode is only 4800 bps.

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, or OVFmn flag before setting the SWC0 bit to 1 and read the value in bits 7 to 0 (RxDq register) of the SDRm1 register.

Caution 5. When using UART reception to transition from STOP mode to SNOOZE mode, use the high-speed on-chip oscillator.

Table 17 - 4 Baud Rate Setting for UART Reception in SNOOZE Mode

High-speed On-chip Oscillator (f_{IH})	Baud Rate for UART Reception in SNOOZE Mode			
	Baud Rate of 4800 bps			
	Operation Clock (f_{MCK})	SDRmn [15:9]	Maximum Permissible Value	Minimum Permissible Value
24 MHz ± 1.0% Note	$f_{CLK}/2^5$	79	1.60%	-2.18%
16 MHz ± 1.0% Note	$f_{CLK}/2^4$	105	2.27%	-1.53%
12 MHz ± 1.0% Note	$f_{CLK}/2^4$	79	1.60%	-2.19%
8 MHz ± 1.0% Note	$f_{CLK}/2^3$	105	2.27%	-1.53%
6 MHz ± 1.0% Note	$f_{CLK}/2^3$	79	1.60%	-2.19%
4 MHz ± 1.0% Note	$f_{CLK}/2^2$	105	2.27%	-1.53%
3 MHz ± 1.0% Note	$f_{CLK}/2^2$	79	1.60%	-2.19%
2 MHz ± 1.0% Note	$f_{CLK}/2$	105	2.27%	-1.54%
1 MHz ± 1.0% Note	f_{CLK}	105	2.27%	-1.57%

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.

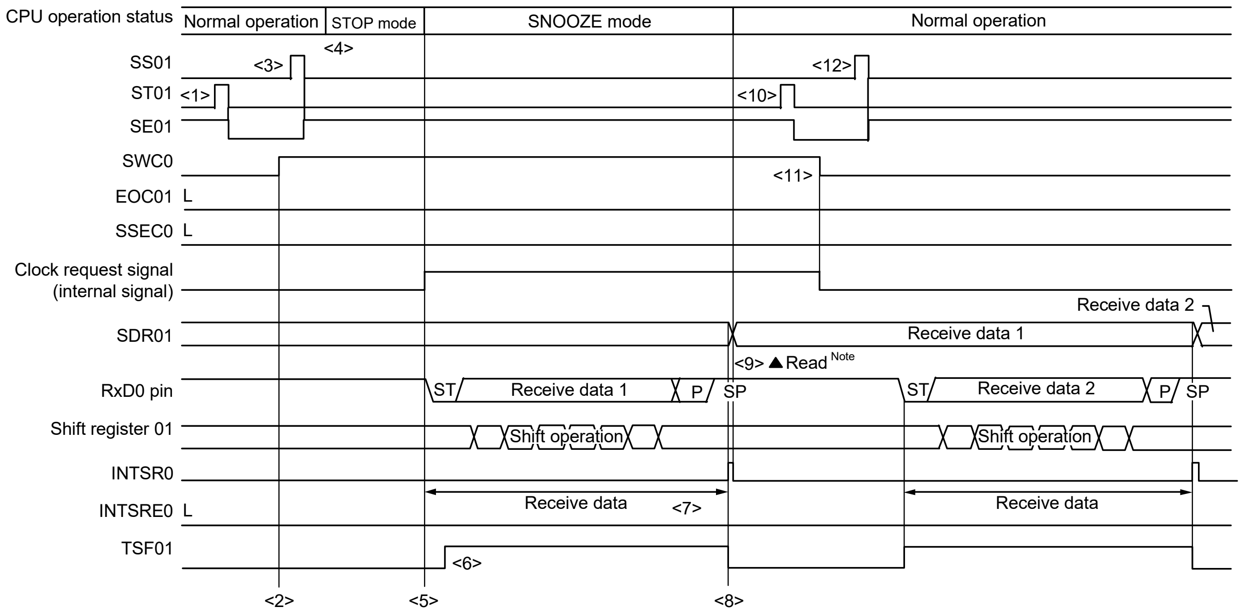
- In the case of f_{IH} ± 1.5%, perform (Maximum permissible value - 0.5%) and (Minimum permissible value + 0.5%) to the values in the above table.
- In the case of f_{IH} ± 2.0%, perform (Maximum permissible value - 1.0%) and (Minimum permissible value + 1.0%) to the values in the above table.

Remark 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.

(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 17 - 122 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)



Note Read the received data when SWCm is 1.

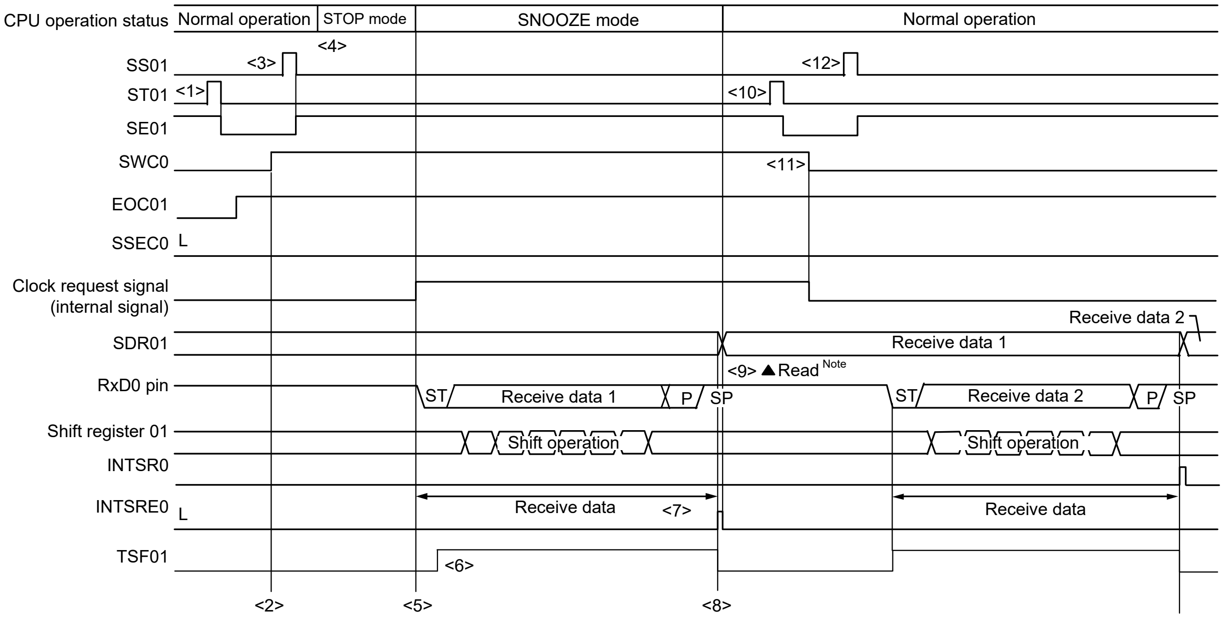
Caution Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

Remark 1. <1> to <12> in the figure correspond to <1> to <12> in Figure 17 - 124 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0).

Remark 2. m = 0; q = 0

(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 17 - 123 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)



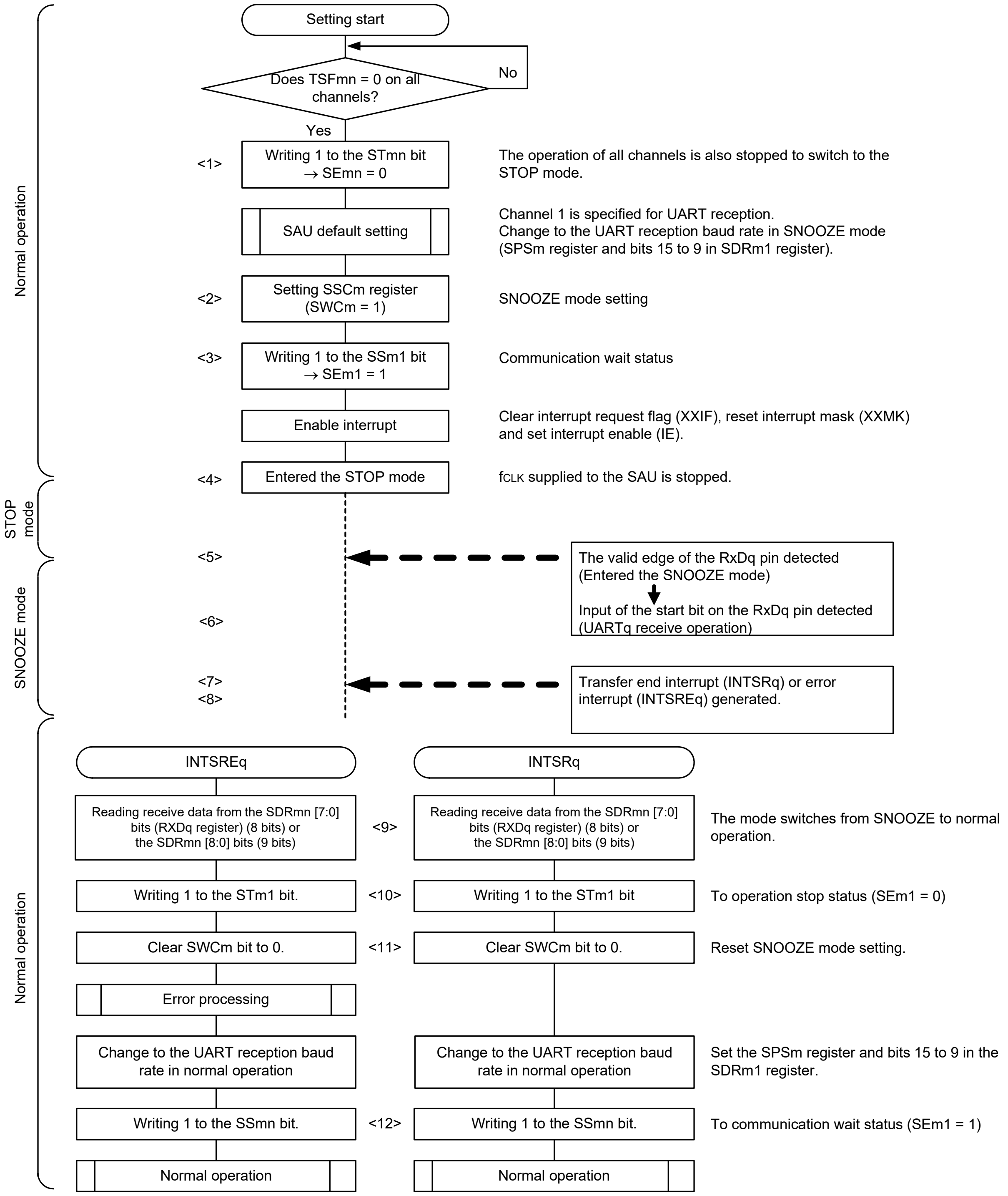
Note Read the received data when SWCm = 1.

Caution Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

Remark 1. <1> to <12> in the figure correspond to <1> to <12> in Figure 17 - 124 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0).

Remark 2. m = 0; q = 0

Figure 17 - 124 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)

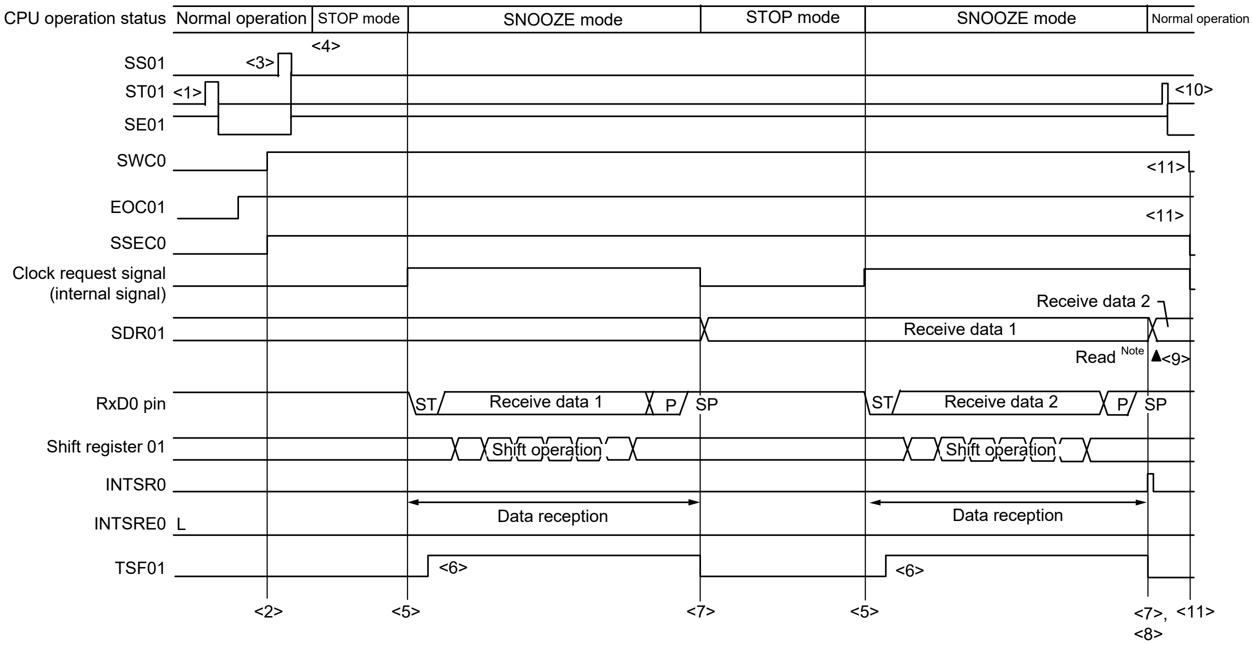


Remark 1. <1> to <12> in the figure correspond to <1> to <12> in Figure 17 - 122 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1) and Figure 17 - 123 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0).

Remark 2. m = 0; q = 0

(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 17 - 125 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)



Note Only read received data while SWCm = 1.

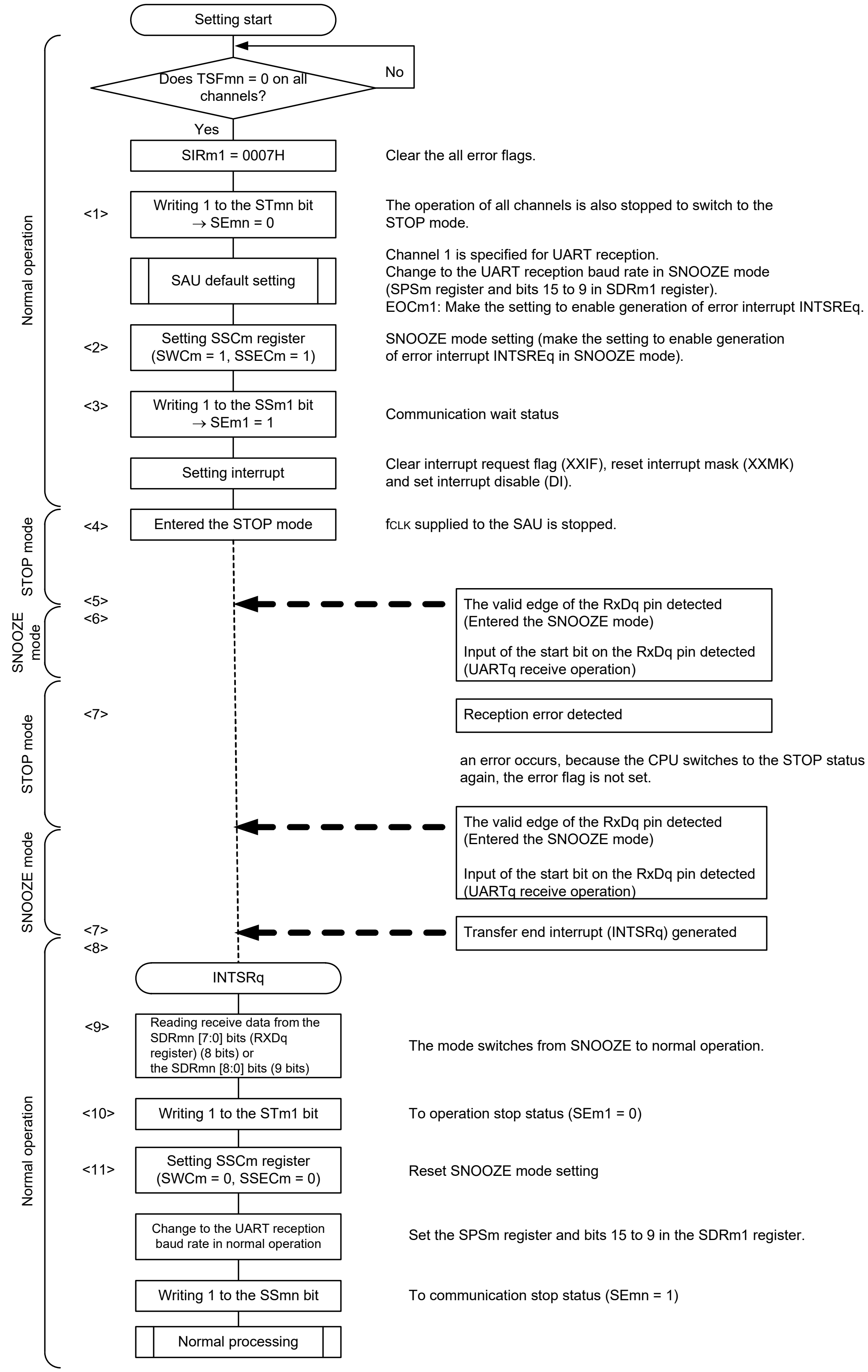
Caution 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

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, or OVFm1 flag 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 17 - 126 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1).

Remark 2. m = 0; q = 0

Figure 17 - 126 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)



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

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, or OVFm1 flag 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 17 - 125 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1).

Remark 2. m = 0; q = 0

17.7.4 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) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [bps]

Caution Setting serial data register mn (SDRmn) SDRmn[15:9] = (0000000B, 0000001B) is prohibited.

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), n: Channel number (n = 1), mn = 01

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 17 - 5 Selection of Operation Clock For UART

SMRmn Register	SPSm Register								Operation Clock (f_{MCK}) Note	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 24 MHz
0	×	×	×	×	0	0	0	0	f_{CLK}	24 MHz
	×	×	×	×	0	0	0	1	$f_{CLK}/2$	12 MHz
	×	×	×	×	0	0	1	0	$f_{CLK}/2^2$	6 MHz
	×	×	×	×	0	0	1	1	$f_{CLK}/2^3$	3 MHz
	×	×	×	×	0	1	0	0	$f_{CLK}/2^4$	1.5 MHz
	×	×	×	×	0	1	0	1	$f_{CLK}/2^5$	750 kHz
	×	×	×	×	0	1	1	0	$f_{CLK}/2^6$	375 kHz
	×	×	×	×	0	1	1	1	$f_{CLK}/2^7$	187.5 kHz
	×	×	×	×	1	0	0	0	$f_{CLK}/2^8$	93.8 kHz
	×	×	×	×	1	0	0	1	$f_{CLK}/2^9$	46.9 kHz
	×	×	×	×	1	0	1	0	$f_{CLK}/2^{10}$	23.4 kHz
	×	×	×	×	1	0	1	1	$f_{CLK}/2^{11}$	11.7 kHz
	×	×	×	×	1	1	0	0	$f_{CLK}/2^{12}$	5.86 kHz
	×	×	×	×	1	1	0	1	$f_{CLK}/2^{13}$	2.93 kHz
	×	×	×	×	1	1	1	0	$f_{CLK}/2^{14}$	1.46 kHz
	×	×	×	×	1	1	1	1	$f_{CLK}/2^{15}$	732 Hz
1	0	0	0	0	×	×	×	×	f_{CLK}	24 MHz
	0	0	0	1	×	×	×	×	$f_{CLK}/2$	12 MHz
	0	0	1	0	×	×	×	×	$f_{CLK}/2^2$	6 MHz
	0	0	1	1	×	×	×	×	$f_{CLK}/2^3$	3 MHz
	0	1	0	0	×	×	×	×	$f_{CLK}/2^4$	1.5 MHz
	0	1	0	1	×	×	×	×	$f_{CLK}/2^5$	750 kHz
	0	1	1	0	×	×	×	×	$f_{CLK}/2^6$	375 kHz
	0	1	1	1	×	×	×	×	$f_{CLK}/2^7$	187.5 kHz
	1	0	0	0	×	×	×	×	$f_{CLK}/2^8$	93.8 kHz
	1	0	0	1	×	×	×	×	$f_{CLK}/2^9$	46.9 kHz
	1	0	1	0	×	×	×	×	$f_{CLK}/2^{10}$	23.4 kHz
	1	0	1	1	×	×	×	×	$f_{CLK}/2^{11}$	11.7 kHz
	1	1	0	0	×	×	×	×	$f_{CLK}/2^{12}$	5.86 kHz
	1	1	0	1	×	×	×	×	$f_{CLK}/2^{13}$	2.93 kHz
	1	1	1	0	×	×	×	×	$f_{CLK}/2^{14}$	1.46 kHz
	1	1	1	1	×	×	×	×	$f_{CLK}/2^{15}$	732 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).

Remark 1. ×: Don't care
Remark 2. m: Unit number (m = 0), n: Channel number (n = 1), mn = 01

(2) Baud rate error during transmission

The baud rate error of UART (UART0) 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} = 24 MHz.

UART Baud Rate (Target Baud Rate)	f_{CLK} = 24 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Baud Rate	Error from Target Baud Rate
300 bps	$f_{CLK}/2^9$	77	300.48 bps	+0.16%
600 bps	$f_{CLK}/2^8$	77	600.96 bps	+0.16%
1200 bps	$f_{CLK}/2^7$	77	1201.92 bps	+0.16%
2400 bps	$f_{CLK}/2^6$	77	2403.85 bps	+0.16%
4800 bps	$f_{CLK}/2^5$	77	4807.69 bps	+0.16%
9600 bps	$f_{CLK}/2^4$	77	9615.38 bps	+0.16%
19200 bps	$f_{CLK}/2^3$	77	19230.8 bps	+0.16%
31250 bps	$f_{CLK}/2^3$	47	31250.0 bps	±0.0%
38400 bps	$f_{CLK}/2^2$	77	38461.5 bps	+0.16%
76800 bps	$f_{CLK}/2$	77	76923.1 bps	+0.16%
153600 bps	f_{CLK}	77	153846 bps	+0.16%
312500 bps	f_{CLK}	37	315789 bps	±1.05%

Remark m: Unit number (m = 0), n: Channel number (n = 0), mn = 00

(3) Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UART0) 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.

$$\text{(Maximum receivable baud rate)} = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times \text{Brate}$$

$$\text{(Minimum receivable baud rate)} = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times \text{Brate}$$

Brate: Calculated baud rate value at the reception side (See **17.7.4 (1) Baud rate calculation expression**.)

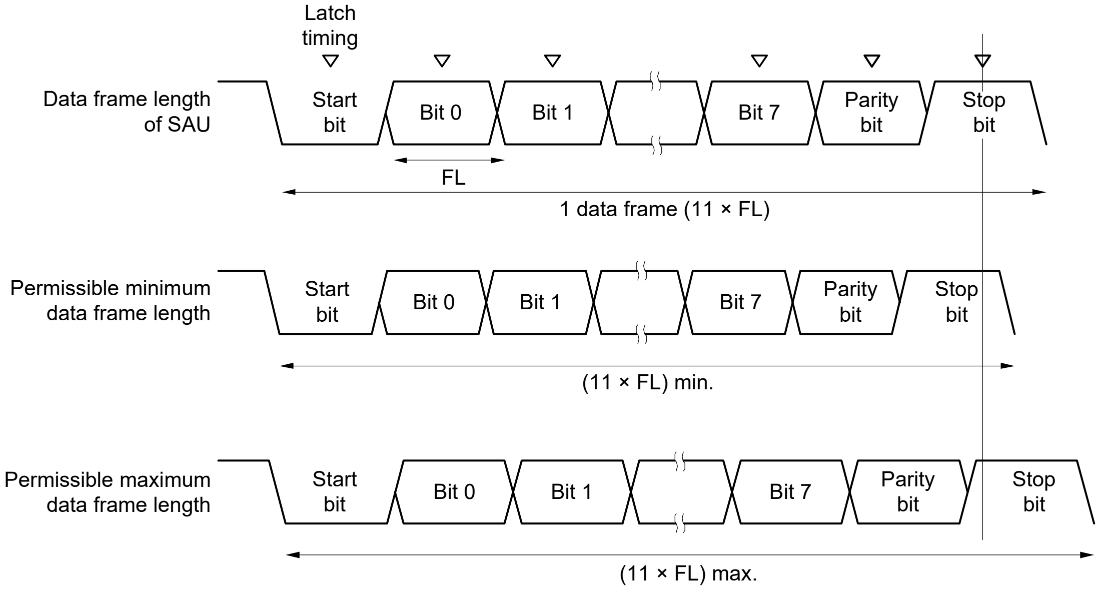
k: SDRmn[15:9] + 1

Nfr: 1 data frame length [bits]

= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 1), mn = 01

Figure 17 - 127 Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in Figure 17 - 127, 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.

17.7.5 Procedure for processing errors that occurred during UART (UART0) communication

The procedure for processing errors that occurred during UART (UART0) communication is described in Figures 17 - 128 and 17 - 129.

Figure 17 - 128 Processing Procedure in Case of Parity Error or Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn) →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

Figure 17 - 129 Processing Procedure in Case of Framing Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets the STmn bit of serial channel stop register m (STm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operating.	
Synchronization with other party of communication		Synchronization with the other party of communication is re-established and communication is resumed because it is considered that a framing error has occurred because the start bit has been shifted.
Sets the SSmn bit of serial channel start register m (SSm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate.	

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

17.8 Operation of Simplified I^2C (IIC00, IIC01) Communication

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This communication function is designed to execute single communication with devices such as EEPROM, flash memory, and A/D converter, and therefore, can be used only by the master.
Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.

[Data transmission/reception]
- Master transmission, master reception (only master function with a single master)
- ACK output function [Note] and ACK detection function
- Data length of 8 bits
  (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- 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 I^2C]
- Slave transmission, slave reception
- Multi-master function (arbitration loss detection function)
<R> • 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 the processing flow in **17.8.3 (2)** for details.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

The channel supporting simplified I^{2}C (IIC00, IIC01) is channels 0 and 1 of SAU0.

• 20-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^{2}C
0	0	CSI00	UART0	IIC00
	1	—		—

• 24, 32-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^{2}C
0	0	CSI00	UART0	IIC00
	1	CSI01		IIC01

• 30-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^{2}C
0	0	CSI00	UART0	IIC00
	1	—		—

• 48-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^{2}C
0	0	CSI00 (supporting slave select input)	UART0	IIC00
	1	CSI01		IIC01

Simplified I^{2}C (IIC00, IIC01) performs the following four types of communication operations.

- Address field transmission (See **17.8.1**.)
- Data transmission (See **17.8.2**.)
- Data reception (See **17.8.3**.)
- Stop condition generation (See **17.8.4**.)

17.8.1 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
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCL00, SDA00 Note 1	SCL01, SDA01 Note 1
Interrupt	INTIIC00	INTIIC01
	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 rate Note 2	Max. f_{MCK}/4 [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I^2C. • Max. 1 MHz (fast mode plus) • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)	
Data level	Non-reversed 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 I^2C, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) with the port output mode register (POMxx). For details, see **4.3 Registers Controlling Port Function** and **4.5 Register Settings When Using Alternate Function**.
When IIC00 is communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) also for the clock input/output pins (SCL00).
For details, see **4.4.4 Handling different potential (1.8 V, 2.5 V) by using I/O buffers**.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

(1) Register setting

Figure 17 - 130 Example of Contents of Registers for Address Field Transmission of Simplified I²C (IIC00, IIC01)

(a) Serial mode register mn (SMRmn)

	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

Operation clock (fMCK) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Operation mode of channel n
0: Transfer end interrupt

(b) Serial communication operation setting register mn (SCRmn)

	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	DLSmn0 1

Setting of parity bit
00B: No parity

Setting of stop bit
01B: Appending 1 bit (ACK)

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	Transmit data setting (address + R/W)							

SIOr (bits 7 to 0)

(d) Serial output register m (SOm)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOm1 0/1	SOm0 0/1

Start condition is generated by manipulating the SOmn bit.

(e) Serial output enable register m (SOEm)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 0/1	SOEm0 0/1

SOEmn = 0 until the start condition is generated, and SOEmn = 1 after generation.

(f) Serial channel start register m (SSm)... Sets only the bits of the target channel is 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SSm1 0/1	SSm0 0/1

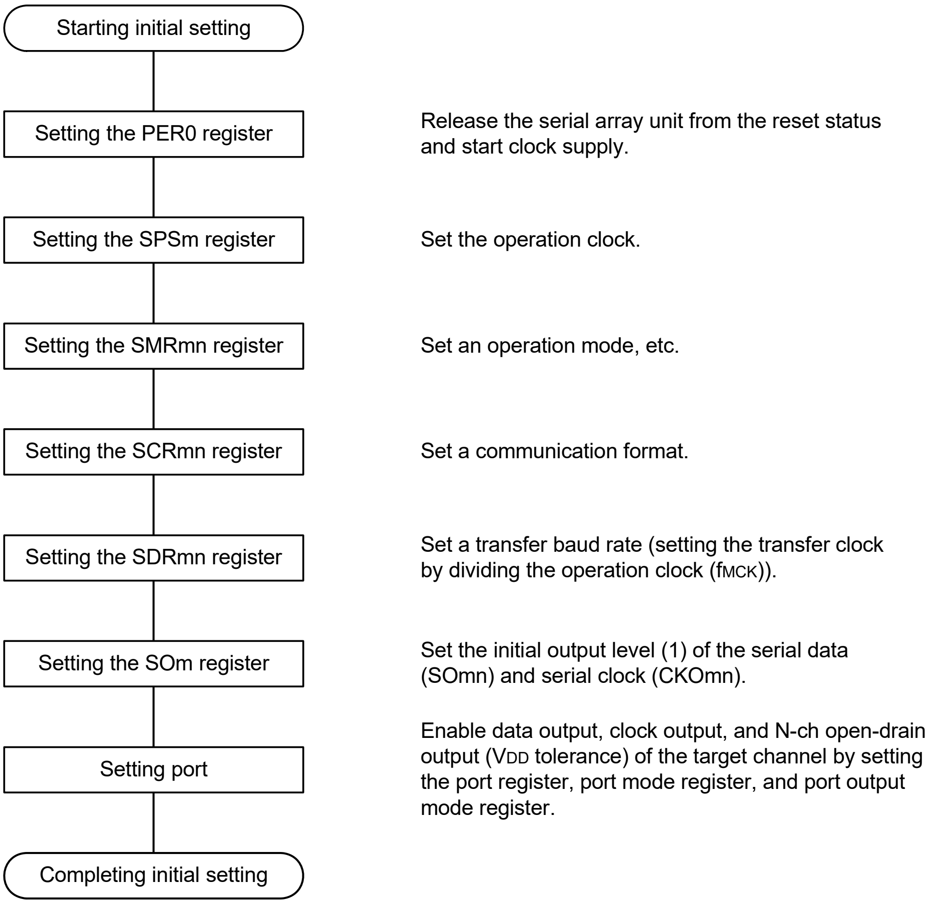
Note 1. Only provided for the SMR00 register.
Note 2. Only provided for the SCR00 register.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01), mn = 00, 01

Remark 2. ■: Setting is fixed in the IIC mode,
■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

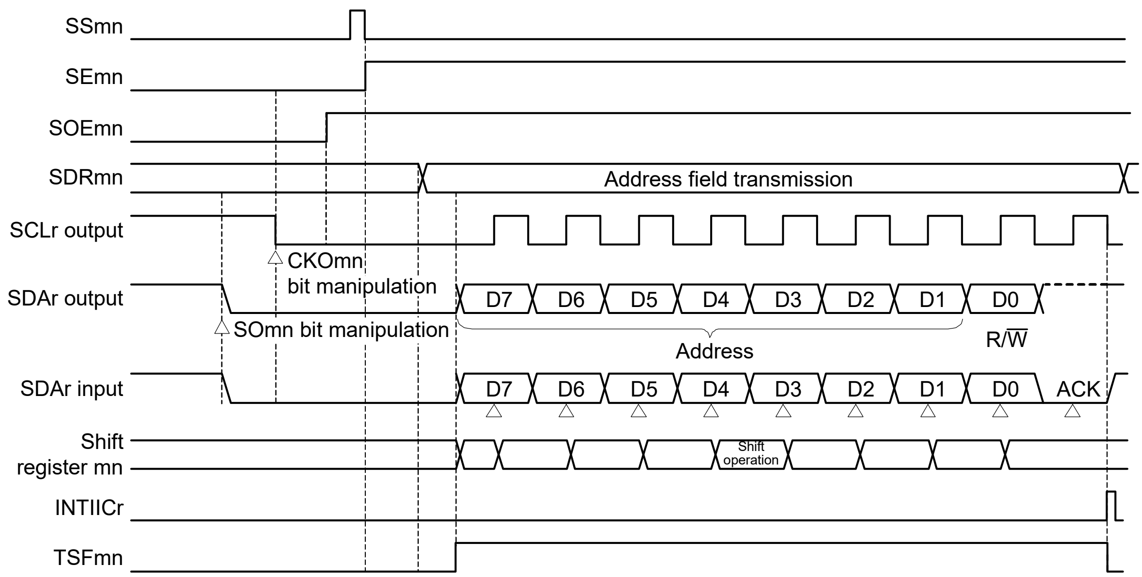
Figure 17 - 131 Initial Setting Procedure for Address Field Transmission



Remark At the end of the initial setting, the simplified I2C (IIC00, IIC01) must be set so that output is disabled and operations are stopped.

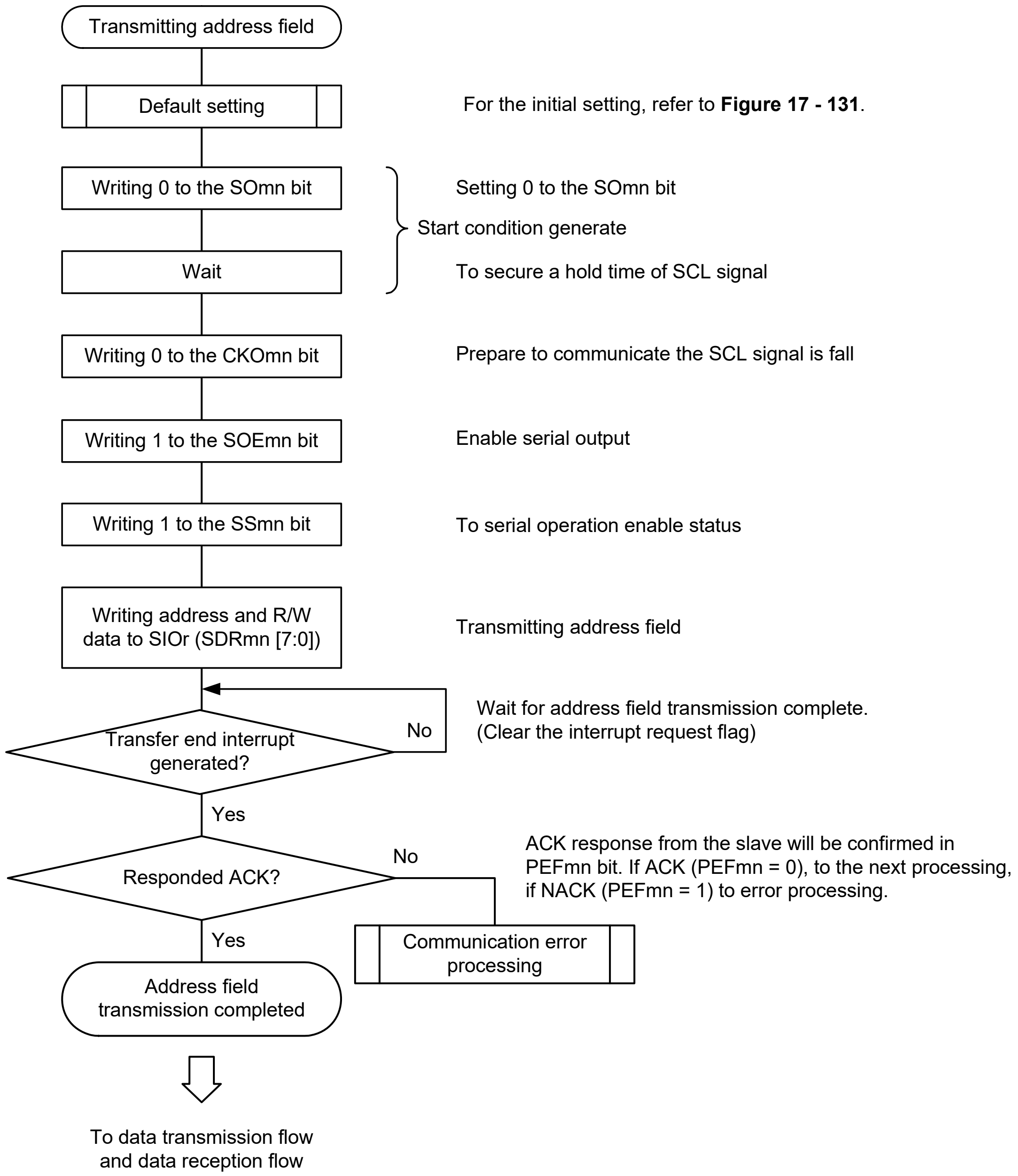
(3) Processing flow

Figure 17 - 132 Timing Chart of Address Field Transmission



Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01), mn = 00, 01

Figure 17 - 133 Flowchart of Address Field Transmission



17.8.2 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 I^{2}C	IIC00	IIC01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCL00, SDA00 Note 1	SCL01, SDA01 Note 1
Interrupt	INTIIC00	INTIIC01
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	ACK error flag (PEFmn)	
Transfer data length	8 bits	
Transfer rate Note 2	Max. f_{MCK}/4 [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I^{2}C. • 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 I^{2}C, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) with the port output mode register (POMxx). For details, see **4.3 Registers Controlling Port Function** and **4.5 Register Settings When Using Alternate Function**.
When IIC00 is communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) also for the clock input/output pins (SCL00).
For details, see **4.4.4 Handling different potential (1.8 V, 2.5 V) by using I/O buffers**.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

(1) Register setting

Figure 17 - 134 Example of Contents of Registers for Data Transmission of Simplified I²C (IIC00, IIC01)

(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	DLSmn0 1

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)... During data transmission/reception, valid only lower 8-bits (SIOr).

	15–9	8	7–0
SDRmn	Baud rate setting Note 3	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	0	0	CKOm1 0/1 Note 4	CKOm0 0/1 Note 4	0	0	0	0	0	0	SOm1 0/1 Note 4	SOm0 0/1 Note 4

(e) Serial output enable register m (SOEm)... Do not manipulate this register during data transmission/reception

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 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	0	0	SSm1 0/1	SSm0 0/1

Note 1. Only provided for the SMR01 register.
Note 2. Only provided for the SCR00 register.
Note 3. Because the setting is completed by address field transmission, setting is not required.
Note 4. The value varies depending on the communication data during communication operation.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01), mn = 00, 01

Remark 2. ■: Setting is fixed in the IIC mode,
■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Processing flow

Figure 17 - 135 Timing Chart of Data Transmission

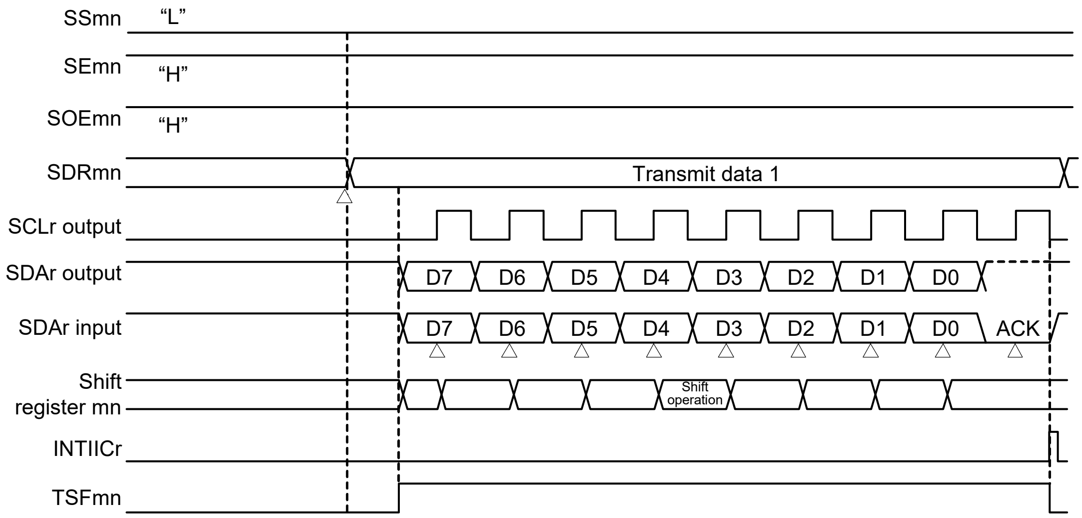
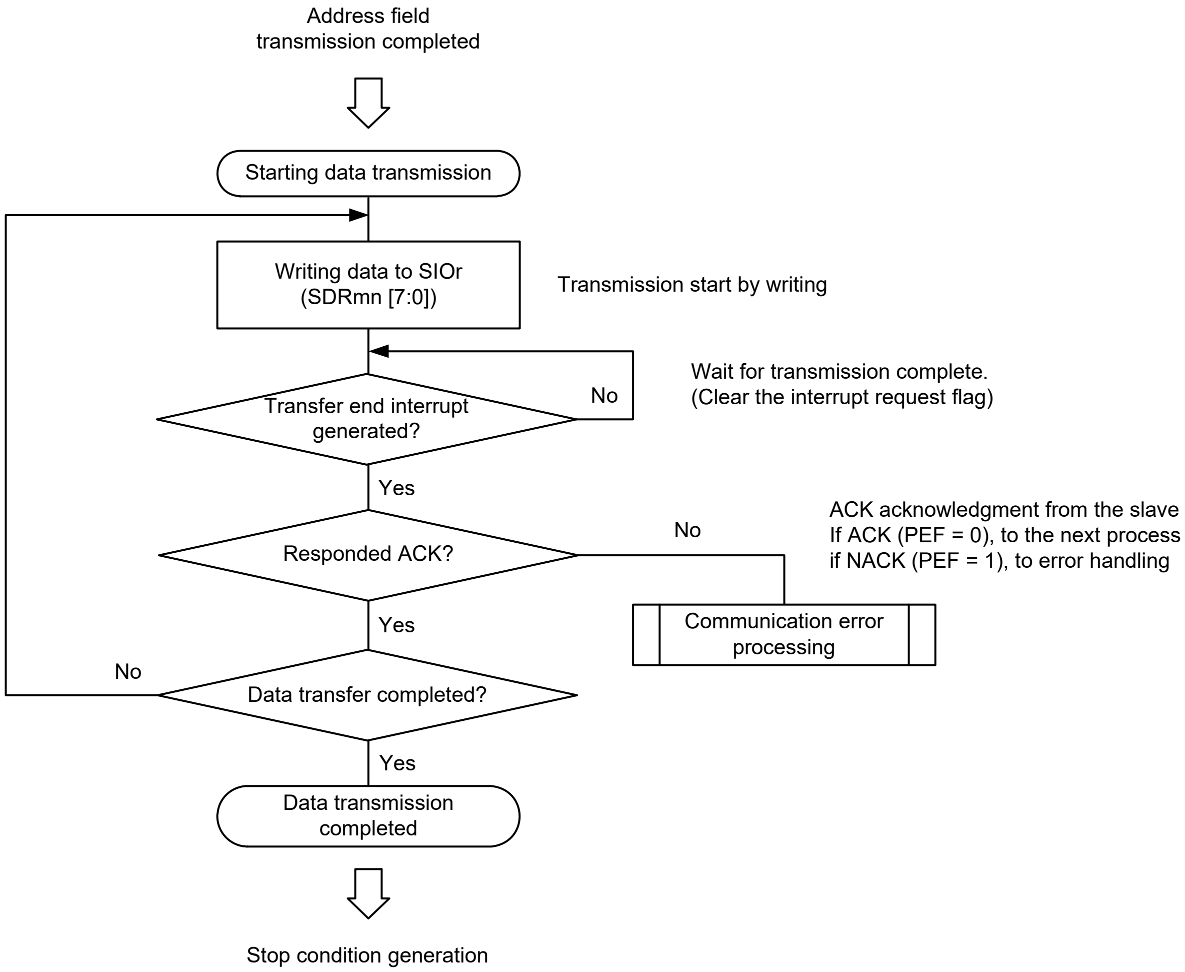


Figure 17 - 136 Flowchart of Data Transmission



17.8.3 Data reception

Data reception is an operation to receive data to the target for transfer (slave) after transmission of an address field. After all data are received to the slave, a stop condition is generated and the bus is released.

Simplified I²C	IIC00	IIC01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCL00, SDA00 Note 1	SCL01, SDA01 Note 1
Interrupt	INTIIC00	INTIIC01
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	Overrun error detection flag (OVFmn) only	
Transfer data length	8 bits	
Transfer rate Note 2	Max. $f_{MCK}/4$ [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of 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 (V_{DD} tolerance) mode (POMxx = 1) with the port output mode register (POMxx). For details, see **4.3 Registers Controlling Port Function** and **4.5 Register Settings When Using Alternate Function**.
When IIC00 is communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) also for the clock input/output pins (SCL00).
For details, see **4.4.4 Handling different potential (1.8 V, 2.5 V) by using I/O buffers**.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

(1) Register setting

Figure 17 - 137 Example of Contents of Registers for Data Reception of Simplified I²C (IIC00, IIC01)

(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	DLSmn0 1

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)

	15–9	8	7–0
SDRmn	Baud rate setting Note 3	0	Dummy transmit data setting (FFH)

$\underbrace{\text{bits 7–0}}_{\text{SIOr}}$

(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	0	0	CKOm1 0/1 Note 4	CKOm0 0/1 Note 4	0	0	0	0	0	0	SOm1 0/1 Note 4	SOm0 0/1 Note 4

(e) Serial output enable register m (SOEm)... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm1 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	0	0	SSm1 0/1	SSm0 0/1

Note 1. Only provided for the SMR01 register.
Note 2. Only provided for the SCR00 register.
Note 3. The baud rate setting is not required because the baud rate has already been set when the address field was transmitted.
Note 4. The value varies depending on the communication data during communication operation.

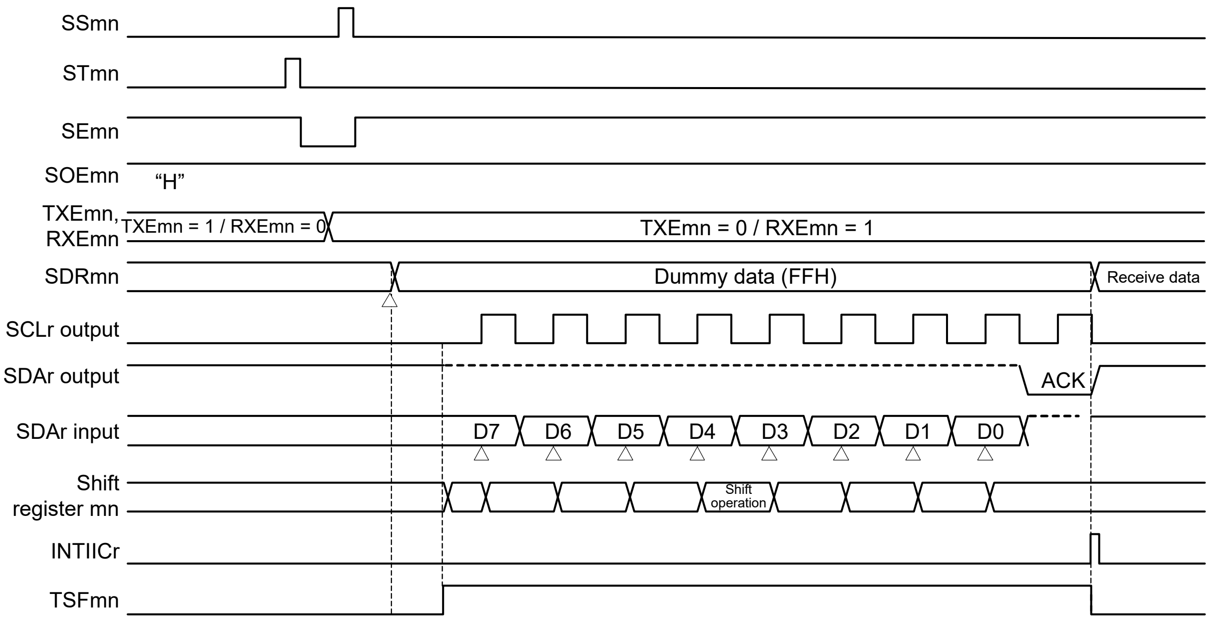
Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01), mn = 00, 01

Remark 2. ■: Setting is fixed in the IIC mode,
■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

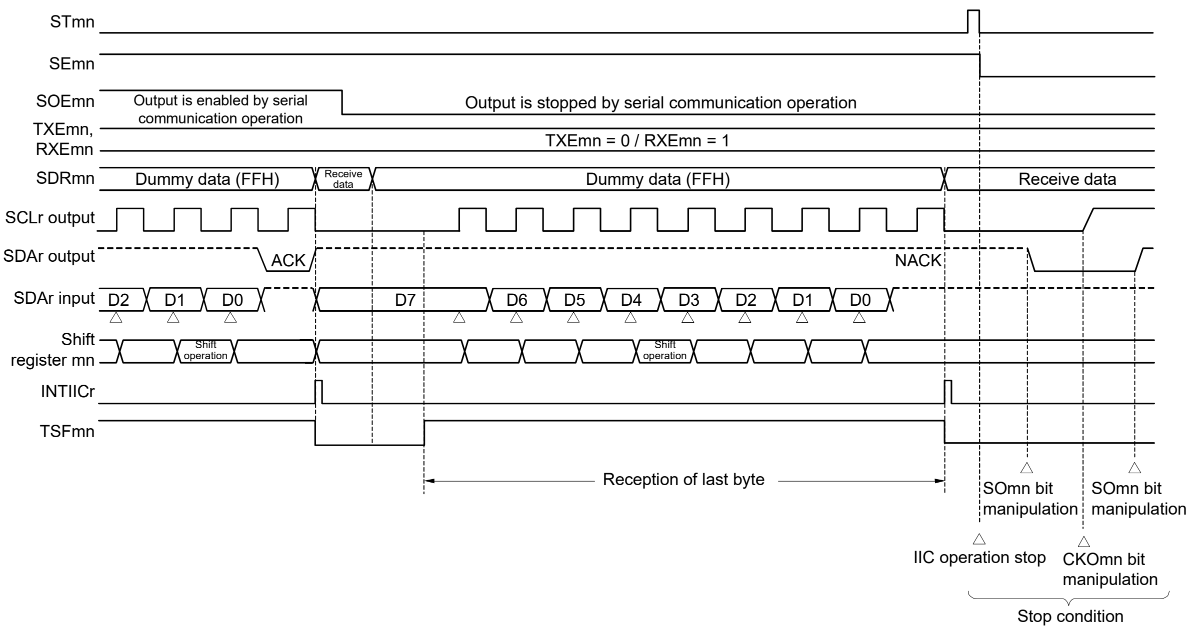
(2) Processing flow

Figure 17 - 138 Timing Chart of Data Reception

(a) When starting data reception

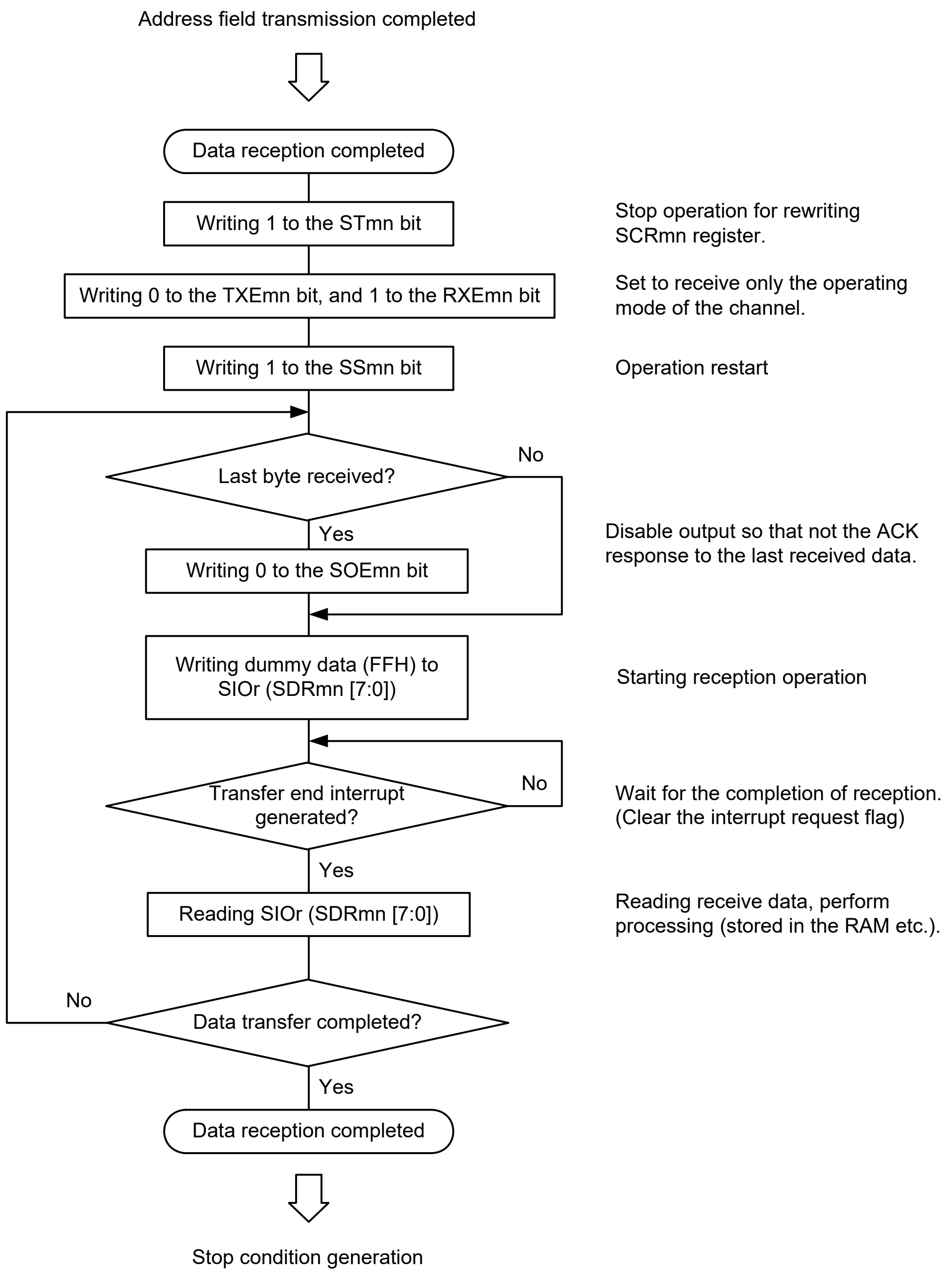


(b) When receiving last data



Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01), mn = 00, 01

Figure 17 - 139 Flowchart of Data Reception



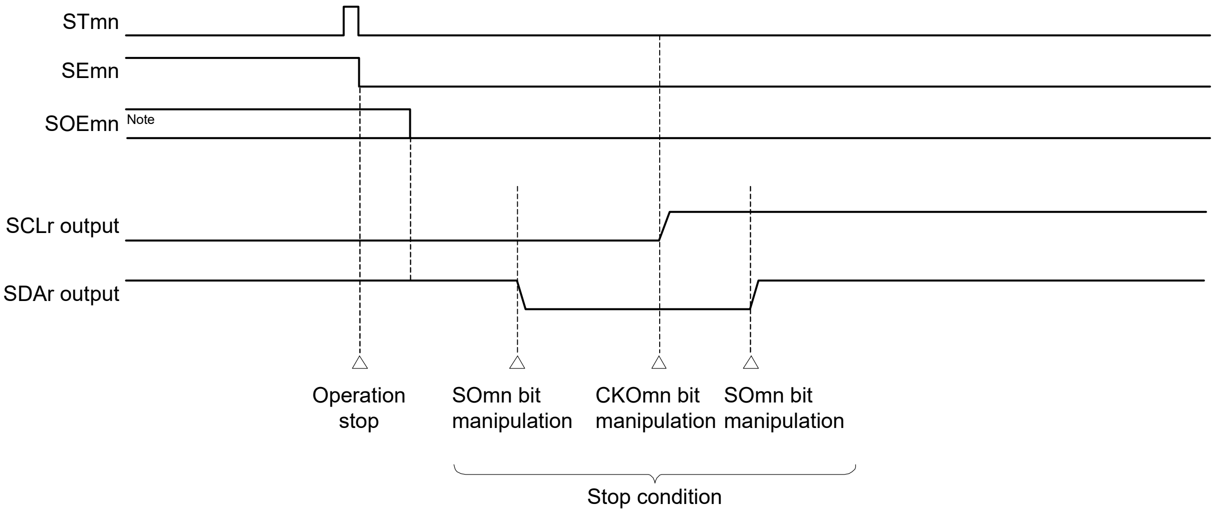
Caution **ACK is not output when the last data is received (NACK). Communication is then completed by setting 1 to the STmn bit of serial channel stop register m (STm) to stop operation and generating a stop condition.**

17.8.4 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.

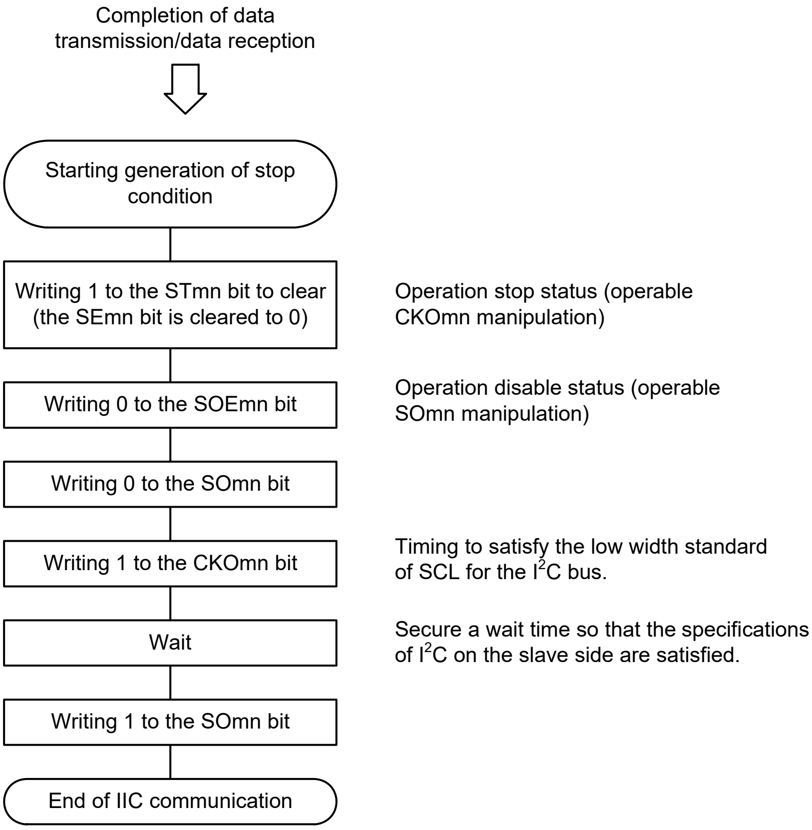
(1) Processing flow

Figure 17 - 140 Timing Chart of Stop Condition Generation



Note During a receive operation, the SOEmn bit of serial output enable register m (SOEm) is cleared to 0 before receiving the last data.

Figure 17 - 141 Flowchart of Stop Condition Generation



17.8.5 Calculating transfer rate

The transfer rate for simplified I^2C (IIC00, IIC01) communication can be calculated by the following expressions.

(Transfer rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2

Caution SDRmn[15:9] must not be set to 0000000B. Be sure to set a value of 0000001B or greater for SDRmn[15:9]. The duty ratio of the SCL signal output by the simplified I^2C is 50%. The I^2C 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 I^2C bus specifications. Make sure that the SDRmn[15:9] value satisfies the I^2C 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), n: Channel number (n = 0, 1), mn = 00, 01

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 17 - 6 Selection of Operation Clock For Simplified I^2C

SMRmn Register	SPSm Register								Operation Clock (f_{MCK}) Note	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 24 MHz
0	×	×	×	×	0	0	0	0	f_{CLK}	24 MHz
	×	×	×	×	0	0	0	1	$f_{CLK}/2$	12 MHz
	×	×	×	×	0	0	1	0	$f_{CLK}/2^2$	6 MHz
	×	×	×	×	0	0	1	1	$f_{CLK}/2^3$	3 MHz
	×	×	×	×	0	1	0	0	$f_{CLK}/2^4$	1.5 MHz
	×	×	×	×	0	1	0	1	$f_{CLK}/2^5$	750 kHz
	×	×	×	×	0	1	1	0	$f_{CLK}/2^6$	375 kHz
	×	×	×	×	0	1	1	1	$f_{CLK}/2^7$	187.5 kHz
	×	×	×	×	1	0	0	0	$f_{CLK}/2^8$	93.8 kHz
	×	×	×	×	1	0	0	1	$f_{CLK}/2^9$	46.9 kHz
	×	×	×	×	1	0	1	0	$f_{CLK}/2^{10}$	23.4 kHz
	×	×	×	×	1	0	1	1	$f_{CLK}/2^{11}$	11.7 kHz
1	0	0	0	0	×	×	×	×	f_{CLK}	24 MHz
	0	0	0	1	×	×	×	×	$f_{CLK}/2$	12 MHz
	0	0	1	0	×	×	×	×	$f_{CLK}/2^2$	6 MHz
	0	0	1	1	×	×	×	×	$f_{CLK}/2^3$	3 MHz
	0	1	0	0	×	×	×	×	$f_{CLK}/2^4$	1.5 MHz
	0	1	0	1	×	×	×	×	$f_{CLK}/2^5$	750 kHz
	0	1	1	0	×	×	×	×	$f_{CLK}/2^6$	375 kHz
	0	1	1	1	×	×	×	×	$f_{CLK}/2^7$	187.5 kHz
	1	0	0	0	×	×	×	×	$f_{CLK}/2^8$	93.8 kHz
	1	0	0	1	×	×	×	×	$f_{CLK}/2^9$	46.9 kHz
	1	0	1	0	×	×	×	×	$f_{CLK}/2^{10}$	23.4 kHz
	1	0	1	1	×	×	×	×	$f_{CLK}/2^{11}$	11.7 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), n: Channel number (n = 0, 1), mn = 00, 01

Here is an example of setting an I^2C transfer rate where $f_{MCK} = f_{CLK}$ = 24 MHz.

I^2C Transfer Mode (Desired Transfer Rate)	f_{CLK} = 24 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Transfer Rate	Error from Desired Transfer Rate
100 kHz	$f_{CLK}/2$	59	100 kHz	0.0%
400 kHz	f_{CLK}	31	375 kHz	6.25% Note
1 MHz	f_{CLK}	14	0.80 MHz	20.0% Note

Note The error cannot be set to about 0% because the duty ratio of the SCL signal is 50%.

17.8.6 Procedure for processing errors that occurred during simplified I²C (IIC00, IIC01) communication

The procedure for processing errors that occurred during simplified I²C (IIC00, IIC01) communication is described in **Figures 17 - 142** and **17 - 143**.

Figure 17 - 142 Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		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.

Figure 17 - 143 Processing Procedure in Case of ACK error in Simplified I²C Mode

Software Manipulation	Hardware Status	Remark
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets the STmn bit of serial channel stop register m (STm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operation.	Slave is not ready for reception because ACK is not returned. Therefore, a stop condition is created, the bus is released, and communication is started again from the start condition. Or, a restart condition is generated and transmission can be redone from address transmission.
Creates stop condition.		
Creates start condition.		
Sets the SSmn bit of serial channel start register m (SSm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate.	

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01), mn = 00, 01

CHAPTER 18 DATA OPERATION CIRCUIT (DOC)

18.1 Overview

The data operation circuit (DOC) is used to compare, add, and subtract 16-bit data.

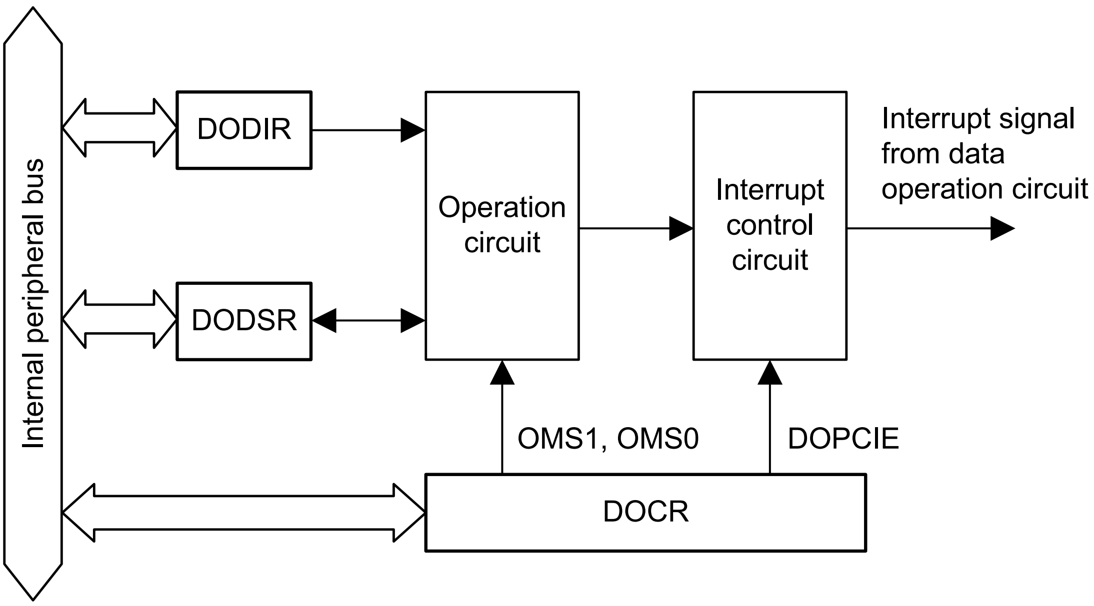
Table 18 - 1 lists the data operation circuit specifications and Figure 18 - 1 shows a block diagram of the data operation circuit.

16-bit data is compared and an interrupt can be generated when a selected condition applies.

Table 18 - 1 DOC Specifications

Item	Description
Data operation function	16-bit data comparison, addition, and subtraction
Lower power consumption function	Module stop state can be set.
Interrupts	• The compared values either match or mismatch • The result of data addition is greater than FFFFH • The result of data subtraction is less than 0000H

Figure 18 - 1 DOC Block Diagram



DOCR: DOC control register
DODIR: DOC data input register
DODSR: DOC data setting register

18.2 Registers Controlling Data Operation Circuit

Table 18 - 2 lists the registers used to control the data operation circuit.

Table 18 - 2 Registers Used to Control Data Operation Circuit

Register Name	Symbol
Peripheral enable register 2	PER2
Peripheral reset control register 2	PRR2
DOC control register	DOCR
DOC data input register	DODIR
DOC data setting register	DODSR

18.2.1 Peripheral enable register 2 (PER2)

The PER2 register is used to enable or disable supply of the clock to the peripheral hardware. Clock supply to a hardware that is not used is stopped in order to reduce the power consumption and noise.
When using the data operation circuit, be sure to set bit 5 (DOCEN) to 1.
The PER2 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 18 - 2 Format of Peripheral enable register 2 (PER2)

Address: F00FCH After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
PER2	TMKAEN	FMCEN	DOCEN	0	0	0	0	0

DOCEN	Control of data operation circuit input clock supply
0	Stops input clock supply. • SFR used by the data operation circuit cannot be written. The read value is 0H. However, the SFR is not initialized.
1	Supplies input clock. • SFR used by the data operation circuit can be read/written.

Note When initializing the data operation circuit and the SFR used by the data operation circuit, be sure to use bit 5 (DOCRES) of PRR2.

Caution 1. Be sure to set the following bits to 0.
Bits 0 to 4

Caution 2. Do not change the value of the target bit in the PER2 register while operation of the peripheral functions is enabled. The value set by PER2 should be changed while operation of the peripheral functions is stopped (except for FMCEN).

18.2.2 Peripheral reset control register 2 (PRR2)

The PRR2 register is used for individual reset control of each peripheral hardware.
This register is used to control reset and reset release of each hardware supported by the PRR2 register.
When resetting the data operation circuit, be sure to set bit 5 (DOCRES) to 1.
The PRR2 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 18 - 3 Format of Peripheral reset control register 2 (PRR2)

Address: F00FDH After reset: 00H R/W

Symbol	<7>	6	<5>	4	3	2	1	0
PRR2	TMKARES	0	DOCRES	0	0	0	0	0

DOCRES	Reset control of data operation circuit
0	The data operation circuit reset is released.
1	The data operation circuit is in the reset status.

18.2.3 DOC control register (DOCR)

Figure 18 - 4 Format of DOC control register (DOCR)

Address: F0511H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	3	<2>	<1>	<0>
DOCR	0	DOPCFCL	DOPCF	DOPCIE	0	DCSEL	OMS1	OMS0

DOPCFCL	DOPCF Clear
0	Maintains the DOPCF flag state.
1	Clears the DOPCF flag.
Setting this bit to 1 clears the DOPCF flag. The read value is 0.	

DOPCF	Data Operation Circuit Flag
0	1 is written to the DOPCFCL bit
1	• The condition selected by the DCSEL bit is met • A result of data addition is greater than FFFFH • A result of data subtraction is less than 0000H
Indicates the result of an operation.	

DOPCIE	Data Operation Circuit Interrupt Enable
0	Disables interrupts from the data operation circuit.
1	Enables interrupts from the data operation circuit.
Setting this bit to 1 enables interrupts from the data operation circuit.	

DCSEL	Detection Condition Select
0	Data mismatch is detected.
1	Data match is detected.
This bit is valid only when data comparison mode is selected. This bit selects the condition for detection in data comparison mode.	

OMS1	OMS0	Operating Mode Select
0	0	Data comparison mode
0	1	Data addition mode
1	0	Data subtraction mode
1	1	Setting prohibited
These bits select the operating mode of the data operation circuit.		

Caution Be sure to set bits 3 and 7 to 0.

18.2.4 DOC data input register (DODIR)

DODIR is a 16-bit readable/writable register in which 16-bit data for use in the operations are stored.

Figure 18 - 5 Format of DOC data input register (DODIR)

Address: F0512H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DODIR																

18.2.5 DOC data setting register (DODSR)

DODSR is a 16-bit readable/writable register. This register stores 16-bit data for use as a reference in data comparison mode. This register also stores the results of operations in data addition and data subtraction modes.

Figure 18 - 6 Format of DOC data setting register (DODSR)

Address: F0514H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DODSR																

18.3 Operation

18.3.1 Data Comparison Mode

Figure 18 - 7 shows an example of the steps involved in data comparison mode operation by the data operation circuit.

In data comparison mode, the data operation circuit operates as follows.

The following is an example of operation when DCSEL is set to 0 (data mismatch is detected as a result of data comparison).

(1) Writing 00B to the OMS1 and OMS0 bits in the DOCR register selects data comparison mode.

(2) The 16-bit reference data is set in the DODSR register.

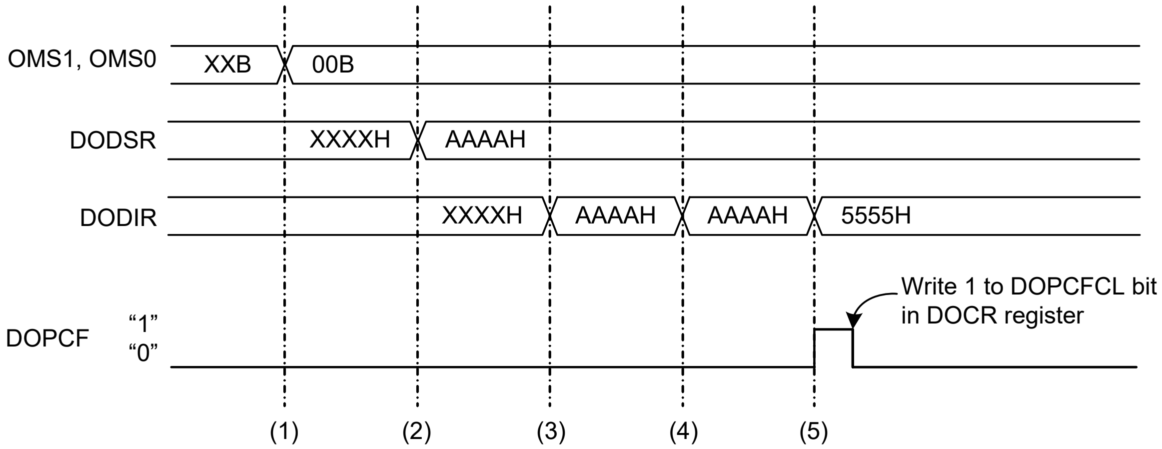
(3) 16-bit data for comparison is written to the DODIR register.

(4) Writing of 16-bit data continues until all data for comparison have been written to the DODIR register.

(5) If a value written to the DODIR register does not match that in the DODSR register [Note], the DOPCF flag in the DOCR register is set to 1. When the DOPCIE bit in the DOCR register is 1, a data operation circuit interrupt is also generated.

Note When DCSEL in the DOCR register = 0

Figure 18 - 7 Example of Operation in Data Comparison Mode



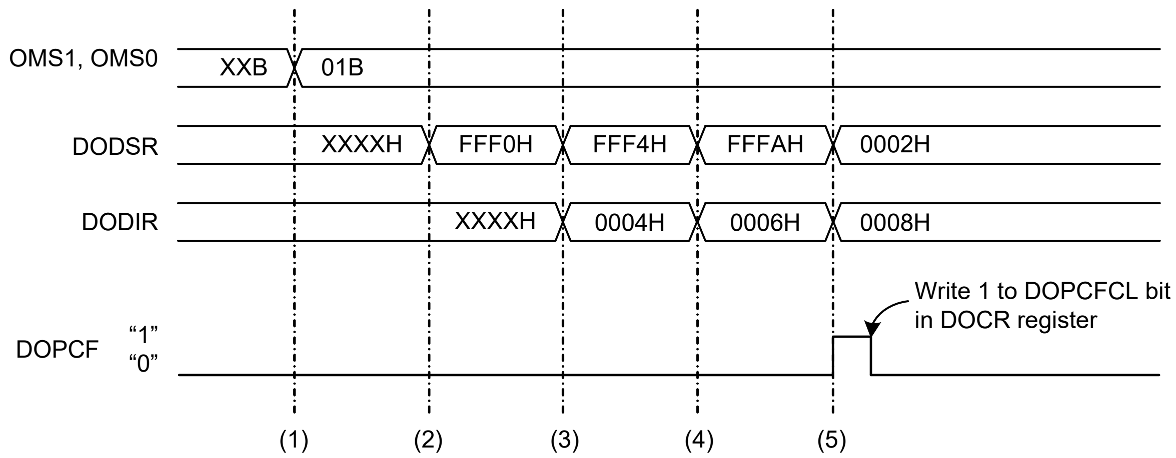
18.3.2 Data Addition Mode

Figure 18 - 8 shows an example of the steps involved in data addition mode operation by the data operation circuit.

In data addition mode, the data operation circuit operates as follows.

(1) Writing 01B to the OMS1 and OMS0 bits selects data addition mode.
(2) 16-bit data is set in the DODSR register as the initial value.
(3) 16-bit data to be added is written to the DODIR register. The result of the operation is stored in the DODSR register.
(4) Writing of 16-bit data continues until all data for addition have been written to the DODIR register.
(5) If the result of an operation is greater than FFFFH, the DOPCF flag in the DOCR register is set to 1. When the DOPCIE bit in the DOCR register is 1, a data operation circuit interrupt is also generated.

Figure 18 - 8 Example of Operation in Data Addition Mode



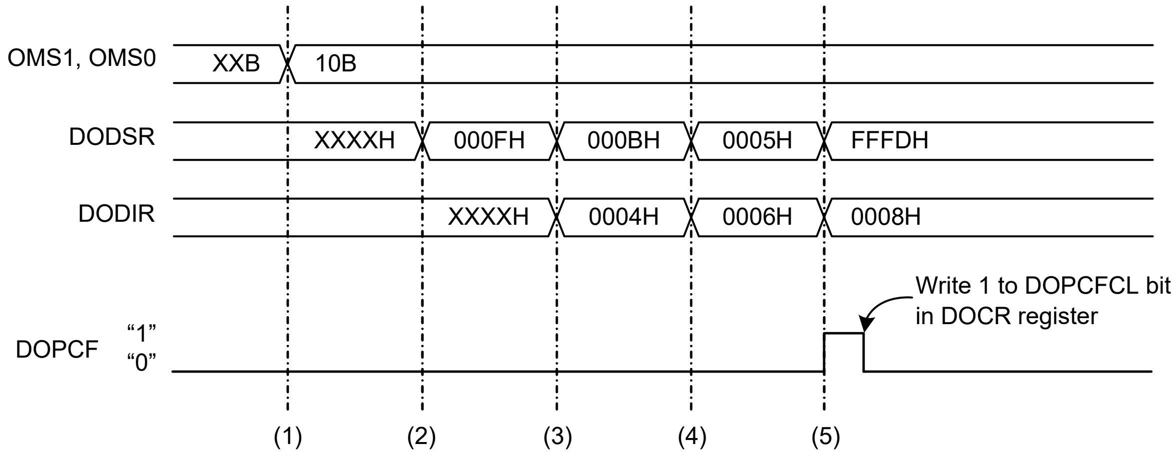
18.3.3 Data Subtraction Mode

Figure 18 - 9 shows an example of the steps involved in data subtraction mode operation by the data operation circuit.

In data subtraction mode, the data operation circuit operates as follows.

(1) Writing 10B to the OMS1 and OMS0 bits selects data subtraction mode.
(2) 16-bit data is set in the DODSR register as the initial value.
(3) 16-bit data to be subtracted is written to DODIR. The result of the operation is stored in DODSR.
(4) Writing of 16-bit data continues until all data for subtraction have been written to DODIR.
(5) If the result of an operation is less than 0000H, the DOPCF flag in the DOCR register is set to 1.
When the DOPCIE bit in the DOCR register is 1, a data operation circuit interrupt is also generated.

Figure 18 - 9 Example of Operation in Data Subtraction Mode



18.4 Interrupt Requests

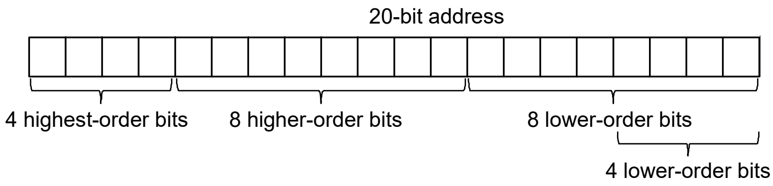
The data operation circuit generates the data operation circuit interrupt as an interrupt request. When an interrupt source is generated, the data operation circuit flag corresponding to the interrupt is set to 1. Table 18 - 3 describes the interrupt request.

Table 18 - 3 Interrupt Request from Data Operation Circuit

Interrupt Request	Data Operation Circuit Flag	Interrupt Generation Timing
Data operation circuit interrupt	DOPCF	• The compared values either match or mismatch • The result of data addition is greater than FFFFH • The result of data subtraction is less than 0000H

CHAPTER 19 DATA TRANSFER CONTROLLER (DTC)

The term “8 higher-order bits of the address” in this chapter 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) is a function that transfers data between memories without using the CPU. The DTC is activated by a peripheral function interrupt to perform data transfers. The DTC and CPU use the same bus, and the DTC takes priority over the CPU in using the bus.

Table 19 - 1 lists the DTC Specifications.

Table 19 - 1 DTC Specifications

Item		Specification
Activation sources		22 sources
Allocatable control data		24 sets
Address space which can be transferred	Address space	64 Kbytes (F0000H to FFFFFH), excluding general-purpose registers
	Sources	Special function register (SFR), RAM area (excluding general-purpose registers), code flash memory area to which access is possible via the mirror area [Note], data flash memory area [Note], extended special function register (2nd SFR)
	Destinations	Special function register (SFR), RAM area (excluding general-purpose registers), extended special function register (2nd SFR)
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 - 5 DTC Activation Sources and Vector Addresses**.
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 transfer.
Transfer start		When bits DTCENi0 to DTCENi7 in the DTCENi registers are 1 (activation enabled), data transfer is started each time the corresponding DTC activation sources are generated.
Transfer stop	Normal mode	• When bits DTCENi0 to DTCENi7 are set to 0 (activation disabled). • When the data transfer causing the DTCCTj register value to change from 1 to 0 is completed.
	Repeat mode	• When bits DTCENi0 to DTCENi7 are set to 0 (activation disabled). • When the data transfer causing the DTCCTj register value to change from 1 to 0 is completed while the RPTINT bit is 1 (interrupt generation enabled).

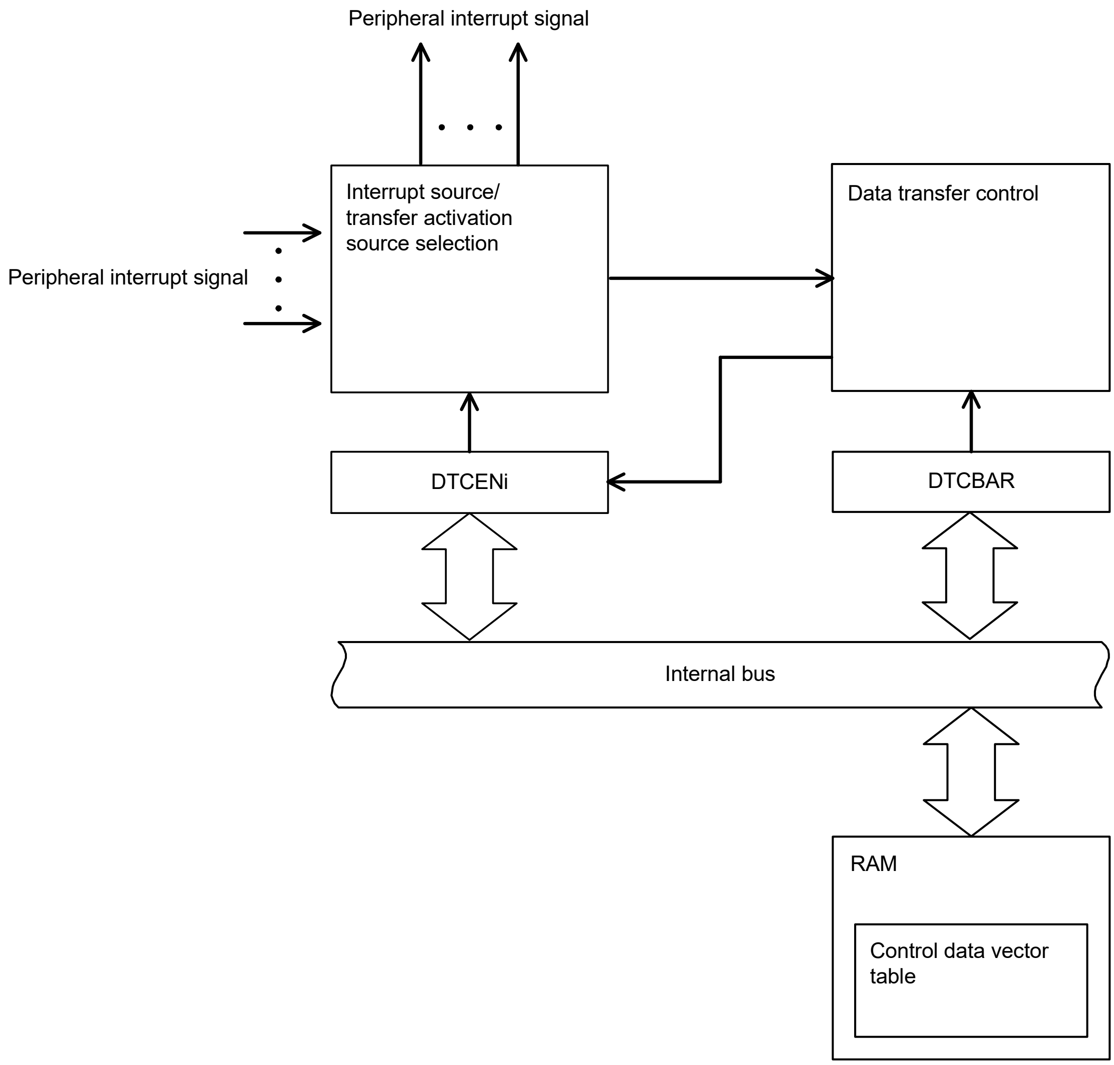
Note In the HALT mode or SNOOZE mode, these areas cannot be set as the sources for DTC transfer since the flash memory is stopped.

Remark i = 0 to 2, j = 0 to 23

19.2 Configuration of DTC

Figure 19 - 1 shows the DTC Block Diagram.

Figure 19 - 1 DTC Block Diagram



19.3 Registers Controlling DTC

Table 19 - 2 lists the Registers Controlling DTC.

Table 19 - 2 Registers Controlling DTC

Register Name	Symbol
Peripheral enable register 1	PER1
DTC activation enable register 0	DTCEN0
DTC activation enable register 1	DTCEN1
DTC activation enable register 2	DTCEN2
DTC base address register	DTCBAR

Table 19 - 3 lists DTC Control Data.
DTC control data is allocated in the DTC control data area in RAM.
The DTCBAR register is used to set the 256-byte area, including the DTC control data area and the DTC vector table area where the start address for control data is stored.

Table 19 - 3 DTC Control Data

Register Name	Symbol
DTC Control Register j	DTCCRj
DTC Block Size Register j	DTBLSj
DTC Transfer Count Register j	DTCCTj
DTC Transfer Count Reload Register j	DTRLDj
DTC Source Address Register j	DTSARj
DTC Destination Address Register j	DTDARj

Remark j = 0 to 23

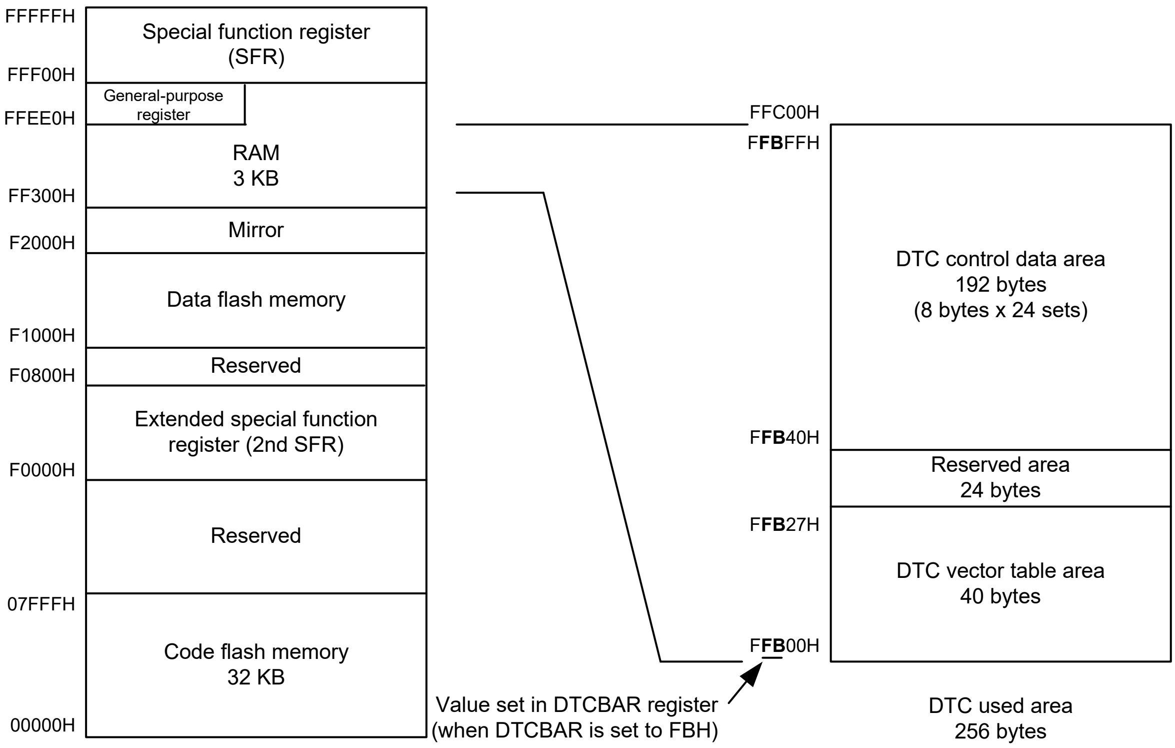
19.3.1 Allocation of DTC Control Data Area and DTC Vector Table Area

The DTCBAR register is used to set the 256-byte area where DTC control data and the vector table within the RAM area.

Figure 19 - 2 shows a Memory Map Example when DTCBAR Register is Set to FBH.

In the 192-byte DTC control data area, the space not used by the DTC can be used as RAM.

Figure 19 - 2 Memory Map Example when DTCBAR Register is Set to FBH



The areas where the DTC control data and vector table can be allocated differ depending on the 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 using the self-programming and data-flash functions.
R5F117xC (x = A, B, G): FF300H to FF709H

Caution 4. The internal RAM area in the following products cannot be used as the DTC control data area or DTC vector table area when using the on-chip debugging trace function.
R5F117xC (x = A, B, G): FF700H to FF8FFH

19.3.2 Control Data Allocation

Control data is allocated beginning with each start address in the order: Registers DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, and DTDARj (j = 0 to 23).
The higher 8 bits for start addresses 0 to 23 are set by the DTCBAR register, and the lower 8 bits are separately set according to the vector table assigned to each activation source.
Figure 19 - 3 shows the control data allocation.

Note 1. Change the data in registers DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, and DTDARj when the corresponding bit among bits DTCENi0 to DTCENi7 (i = 0 to 2) in the DTCENi register is set to 0 (activation disabled).

Note 2. Do not access DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, or DTDARj using a DTC transfer.

Figure 19 - 3 Control Data Allocation

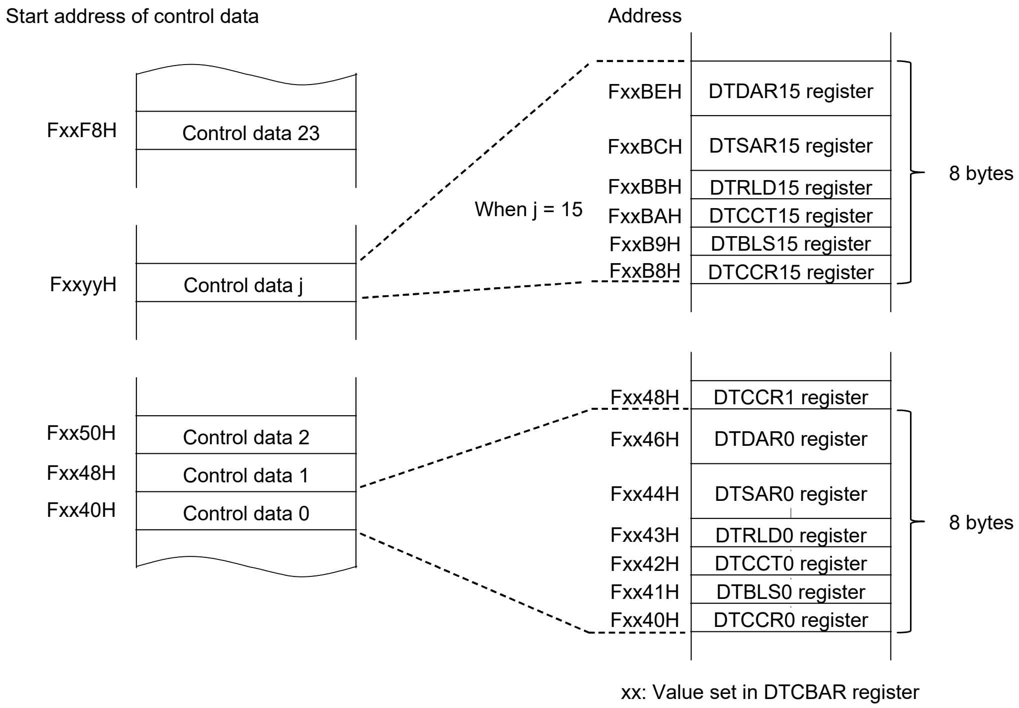


Table 19 - 4 Start Address of Control Data

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

xx: Value set in DTCBAR register

19.3.3 Vector Table

When the DTC is activated, one set of control data from among the 24 control data sets is selected according to the 8 lower-order bits of values read from the location in the vector table which is assigned to the corresponding activation source, and the selected control data are read from the DTC control data area.
The 8 higher-order bits of the vector address are set in the DTCBAR register, and 00H to 16H are allocated to the 8 lower-order bits corresponding to the activation source.
One byte of the vector table is assigned to each activation source, and values from 40H to F8H are stored in each location. When the DTC is activated, one of the 24 control data sets is selected based on the 4 lower-order bits of the address value.
Table 19 - 5 lists the DTC Activation Sources and Vector Addresses.

Note Change the start address of the DTC control data area to be set in the vector table when the corresponding bit among bits DTCENi0 to DTCENi7 (i = 0 to 2) in the DTCENi register is set to 0 (activation disabled).

Figure 19 - 4 Start Address of Control Data and Vector Table

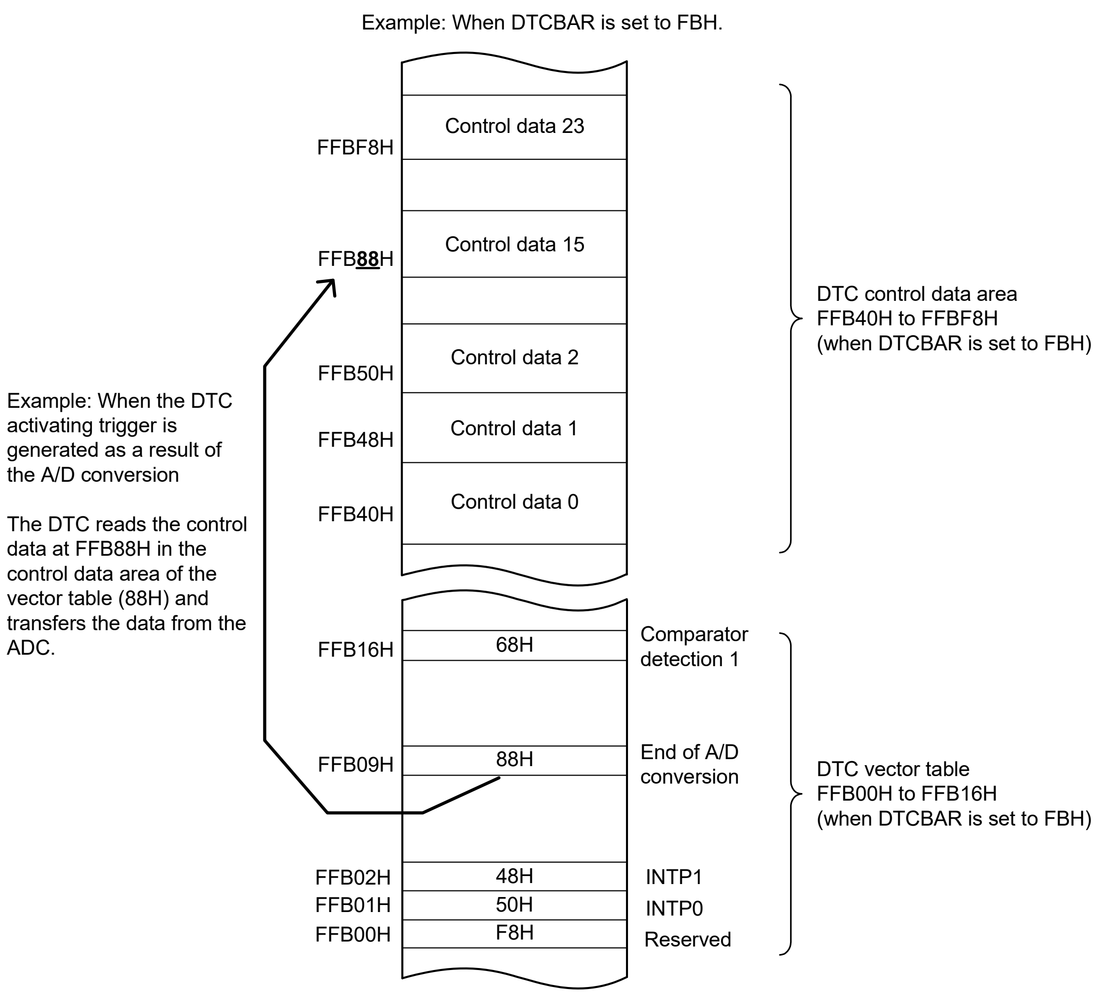


Table 19 - 5 DTC Activation Sources and Vector Addresses

DTC Activation Sources (Interrupt Request Source)	Source No.	Vector Address	Priority
Reserved	0	Address set in DTCBAR register +00H	Highest
INTP0	1	Address set in DTCBAR register +01H	
INTP1	2	Address set in DTCBAR register +02H	
INTP2	3	Address set in DTCBAR register +03H	
INTP3	4	Address set in DTCBAR register +04H	
INTP4	5	Address set in DTCBAR register +05H	
INTP5	6	Address set in DTCBAR register +06H	
INTP6	7	Address set in DTCBAR register +07H	
Key input	8	Address set in DTCBAR register +08H	
A/D conversion end	9	Address set in DTCBAR register +09H	
UART0 reception transfer end/CSI01 transfer end or buffer empty/IIC01 transfer end	10	Address set in DTCBAR register +0AH	
UART0 transmission transfer end/CSI00 transfer end or buffer empty/IIC00 transfer end	11	Address set in DTCBAR register +0BH	
End of channel 0 of timer array unit 0 count or capture	12	Address set in DTCBAR register +0CH	
End of channel 1 of timer array unit 0 count or capture	13	Address set in DTCBAR register +0DH	
End of channel 2 of timer array unit 0 count or capture	14	Address set in DTCBAR register +0EH	
End of channel 3 of timer array unit 0 count or capture	15	Address set in DTCBAR register +0FH	
12-bit interval timer	16	Address set in DTCBAR register +10H	
8-bit interval timer 00	17	Address set in DTCBAR register +11H	
8-bit interval timer 01	18	Address set in DTCBAR register +12H	
8-bit interval timer 10	19	Address set in DTCBAR register +13H	
8-bit interval timer 11	20	Address set in DTCBAR register +14H	
Comparator detection 0	21	Address set in DTCBAR register +15H	
Comparator detection 1	22	Address set in DTCBAR register +16H	Lowest

19.3.4 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise.
When using the DTC, be sure to set bit 3 (DTCEN) to 1.

The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 19 - 5 Format of Peripheral enable register 1 (PER1)

Address: F00FAH After reset: 00H R/W

Symbol	7	6	<5>	4	<3>	2	1	0
PER1	0	0	CMPEN	0	DTCEN	0	0	0

DTCEN	Control of DTC input clock supply
0	Stops input clock supply. • DTC cannot run.
1	Enables input clock supply. • DTC can run.

Caution Be sure to set bits to 0 to 2, 4, 6, and 7 to 0.

19.3.5 DTC control register j (DTCCRj) (j = 0 to 23)

The DTCCRj register is used to control the DTC operating mode.

Figure 19 - 6 Format of DTC control register j (DTCCRj)

Address: Refer to **19.3.2 Control Data Allocation**. After reset: Undefined 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 the RPTINT 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 the DAMOD bit is invalid when the MODE bit is 1 (repeat mode) and the RPTSEL bit is 0 (transfer destination is the repeat area).	

SAMOD	Transfer source address control
0	Fixed
1	Incremented
The setting of the SAMOD bit is invalid when the MODE bit is 1 (repeat mode) and the RPTSEL bit is 1 (transfer source is the repeat area).	

RPTSEL	Repeat area selection
0	Transfer destination is the repeat area
1	Transfer source is the repeat area
The setting of the RPTSEL bit is invalid when the MODE bit is 0 (normal mode).	

MODE	Transfer mode selection
0	Normal mode
1	Repeat mode

Caution Do not access the DTCCRj register using a DTC transfer.

19.3.6 DTC block size register j (DTBLSj) (j = 0 to 23)

This register is used to set the block size of the data to be transferred by one activation.

Figure 19 - 7 Format of DTC block size register j (DTBLSj)

Address: Refer to **19.3.2 Control Data Allocation**. After reset: Undefined 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.

19.3.7 DTC transfer count register j (DTCCTj) (j = 0 to 23)

This register is used to set the number of DTC data transfers. The value is decremented by 1 each time DTC transfer is activated once.

Figure 19 - 8 Format of DTC transfer count register j (DTCCTj)

Address: Refer to **19.3.2 Control Data Allocation**. After reset: Undefined 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.

19.3.8 DTC transfer count reload register j (DTRLDj) (j = 0 to 23)

This register is used to set the initial value of the transfer count register in repeat mode. Since the value of this register is reloaded to the DTCCT register in repeat mode, set the same value as the initial value of the DTCCT register.

Figure 19 - 9 Format of DTC transfer count reload register j (DTRLDj)

Address: Refer to **19.3.2 Control Data Allocation**. After reset: Undefined R/W

	7	6	5	4	3	2	1	0
DTRLDj	DTRLDj7	DTRLDj6	DTRLDj5	DTRLDj4	DTRLDj3	DTRLDj2	DTRLDj1	DTRLDj0

Caution Do not access the DTRLDj register using a DTC transfer.

19.3.9 DTC source address register j (DTSARj) (j = 0 to 23)

This register is used to specify the transfer source address for data transfer.
When the SZ bit in the DTCCRj register is set to 1 (16-bit transfer), the lowest bit is ignored and the address is handled as an even address.

Figure 19 - 10 Format of DTC source address register j (DTSARj)

Address: Refer to **19.3.2 Control Data Allocation**. After reset: Undefined R/W

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTSARj	DTS ARj15	DTS ARj14	DTS ARj13	DTS ARj12	DTS ARj11	DTS ARj10	DTS ARj9	DTS ARj8	DTS ARj7	DTS ARj6	DTS ARj5	DTS ARj4	DTS ARj3	DTS ARj2	DTS ARj1	DTS ARj0

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.

19.3.10 DTC destination address register j (DTDARj) (j = 0 to 23)

This register is used to specify the transfer destination address for data transfer.
When the SZ bit in the DTCCRj register is set to 1 (16-bit transfer), the lowest bit is ignored and the address is handled as an even address.

Figure 19 - 11 Format of DTC destination address register j (DTDARj)

Address: Refer to **19.3.2 Control Data Allocation**. After reset: Undefined R/W

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTDARj	DTD ARj15	DTD ARj14	DTD ARj13	DTD ARj12	DTD ARj11	DTD ARj10	DTD ARj9	DTD ARj8	DTD ARj7	DTD ARj6	DTD ARj5	DTD ARj4	DTD ARj3	DTD ARj2	DTD ARj1	DTD ARj0

Caution 1. Do not set the general-purpose register (FFEE0H to FFEFFH) space to the transfer source address.
Caution 2. Do not access the DTDARj register using a DTC transfer.

19.3.11 DTC activation enable register i (DTCENi) (i = 0 to 2)

This is an 8-bit register which enables or disables DTC activation by interrupt sources. Table 19 - 6 lists the Correspondences between Interrupt Sources and Bits DTCENi0 to DTCENi7.
The DTCENi register can be set by an 8-bit memory manipulation instruction and a 1-bit memory manipulation instruction.

Caution 1. Modify bits DTCENi0 to DTCENi7 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 function is assigned, be sure to set their values to 0.

Figure 19 - 12 Format of DTC activation enable register i (DTCENi) (i = 0 to 2)

Address: F02E8H (DTCEN0), F02E9H (DTCEN1), F02EAH (DTCEN2) After reset: 00H 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.	

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.	

Table 19 - 6 Correspondences between Interrupt Sources and Bits DTCENi0 to DTCENi7

Register	DTCENi7 Bit	DTCENi6 Bit	DTCENi5 Bit	DTCENi4 Bit	DTCENi3 Bit	DTCENi2 Bit	DTCENi1 Bit	DTCENi0 Bit
DTCEN0	Reserved	INTP0	INTP1	INTP2	INTP3	INTP4	INTP5	INTP6
DTCEN1	Key input	A/D conversion end	UART0 reception transfer end/ CSI01 transfer end or buffer empty/ IIC01 transfer end	UART0 transmission transfer end/ CSI00 transfer end or buffer empty/ IIC00 transfer end	End of channel 0 of timer array unit 0 count or capture	End of channel 1 of timer array unit 0 count or capture	End of channel 2 of timer array unit 0 count or capture	End of channel 3 of timer array unit 0 count or capture
DTCEN2	12-bit interval timer	8-bit interval timer 00	8-bit interval timer 01	8-bit interval timer 10	8-bit interval timer 11	Comparator detection 0	Comparator detection 1	Reserved

Caution **For the bits to which no function is assigned, be sure to set their values to 0.**

Remark i = 0 to 2

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 higher 8 bits to generate a 16-bit address.

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 notes on 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

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, control data is read from the DTC control data area to perform data transfers and control data after data transfer is written back to the DTC control data area. Twenty-four sets of control data can be stored in the DTC control data area, which allows 24 types of data transfers to be performed.
There are two transfer modes (normal mode and repeat mode) and two transfer sizes (8-bit transfer and 16-bit transfer). When the CHNE bit in the DTCCRj (j = 0 to 23) register is set to 1 (chain transfers enabled), multiple control data is read and data transfers are continuously performed by one activation source (chain transfers).
A transfer source address is specified by the 16-bit register DTSARj, and a transfer destination address is specified by the 16-bit register DTDARj.
The values in registers DTSARj and DTDARj are separately incremented or fixed according to the control data after the data transfer.

19.4.1 Activation Sources

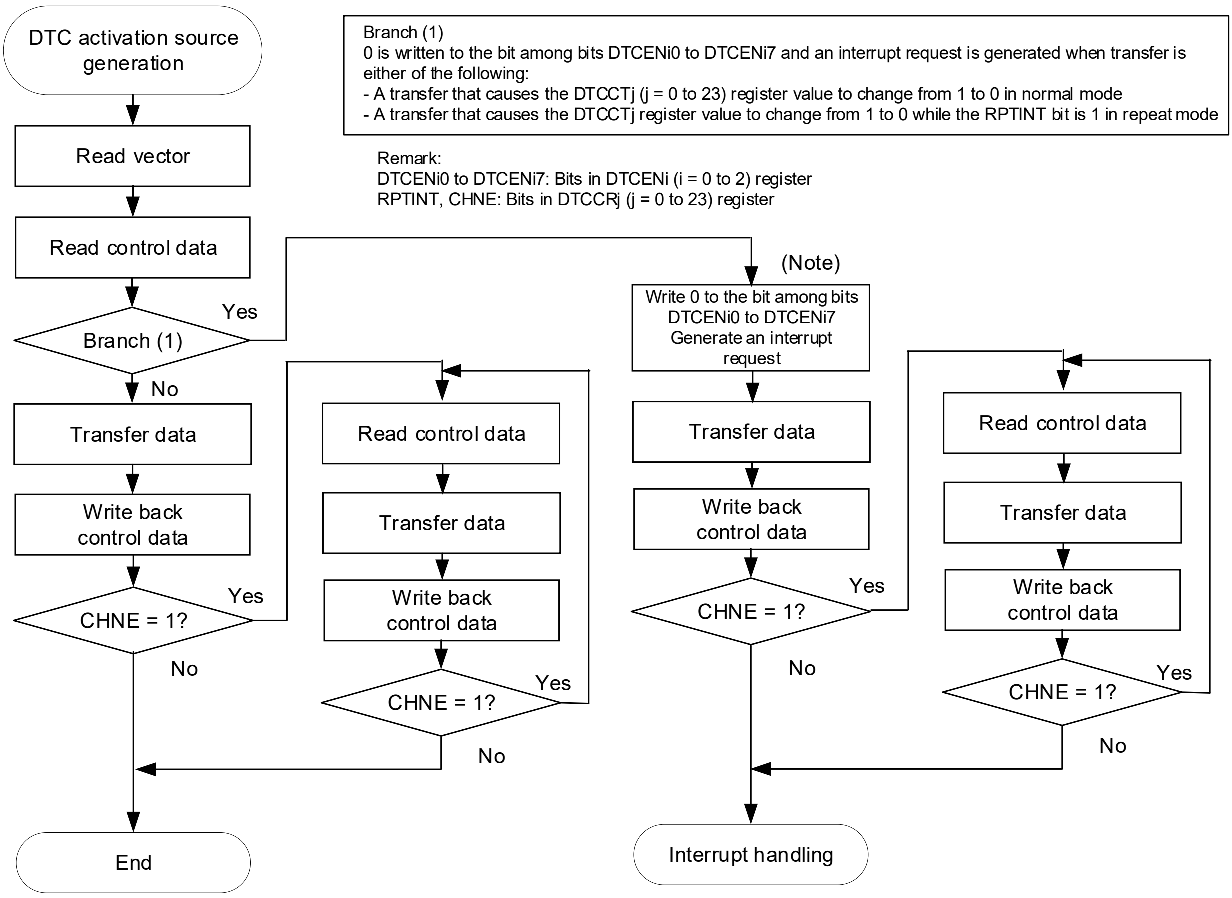
The DTC is activated by an interrupt signal from the peripheral functions. The interrupt signals to activate the DTC are selected with the DTCENi (i = 0 to 2) register.

The DTC sets the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi register to 0 (activation disabled) during operation when the setting of data transfer (the first transfer in chain transfers) is either of the following:

- A transfer that causes the DTCCTj (j = 0 to 23) register value to change to 0 in normal mode
- A transfer that causes the DTCCTj register value to change to 0 while the RPTINT bit in the DTCCRj register is 1 (interrupt generation enabled) in repeat mode

Figure 19 - 14 shows the DTC Internal Operation Flowchart.

Figure 19 - 14 DTC Internal Operation Flowchart



Note 0 is not written to the bit among bits DTCENi0 to DTCENi7 for data transfers activated by the setting to enable chain transfers (the CHNE bit is 1). Also, no interrupt request is generated.

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 DTCCTj (j = 0 to 23) register value to change to 0 is performed, the DTC generates an interrupt request corresponding to the activation source to the interrupt controller during DTC operation, and sets the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi (i = 0 to 2) register to 0 (activation disabled).
Table 19 - 7 shows Register Functions in Normal Mode. Figure 19 - 15 shows Data Transfers in Normal Mode.

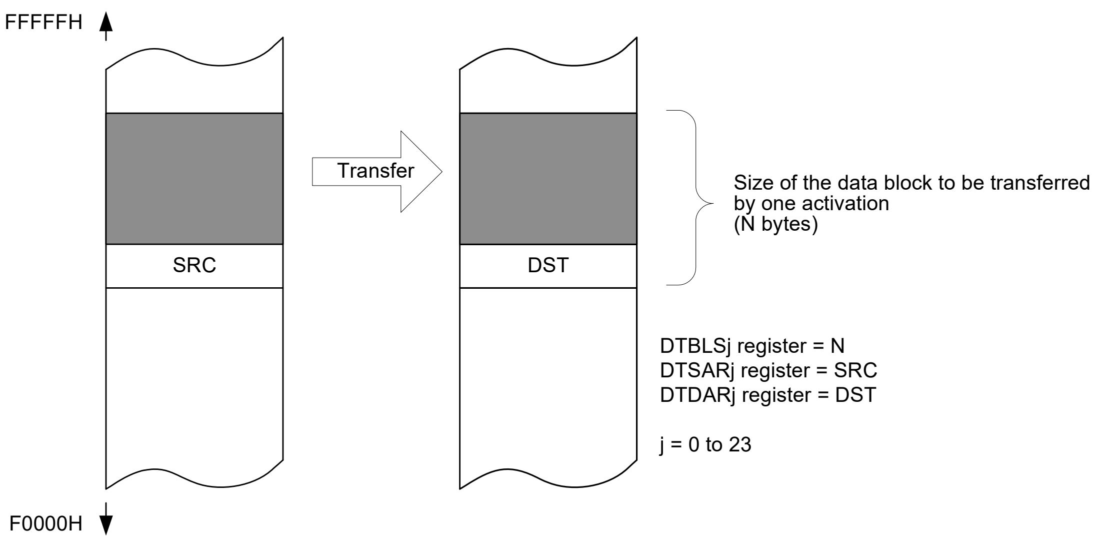
Table 19 - 7 Register Functions in Normal Mode

Register Name	Symbol	Function
DTC block size register j	DTBLSj	Size of the data block to be transferred by one activation
DTC transfer count register j	DTCCTj	Number of data transfers
DTC transfer count reload register j	DTRLDj	Not used [Note]
DTC source address register j	DTSARj	Data transfer source address
DTC destination address register j	DTDARj	Data transfer destination address

Note Initialize 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



DTCCR Register Setting				Source Address Control	Destination Address Control	Source Address after Transfer	Destination Address after Transfer
DAMOD	SAMOD	RPTSEL	MODE				
0	0	X	0	Fixed	Fixed	SRC	DST
0	1	X	0	Incremented	Fixed	SRC + N	DST
1	0	X	0	Fixed	Incremented	SRC	DST + N
1	1	X	0	Incremented	Incremented	SRC + N	DST + N

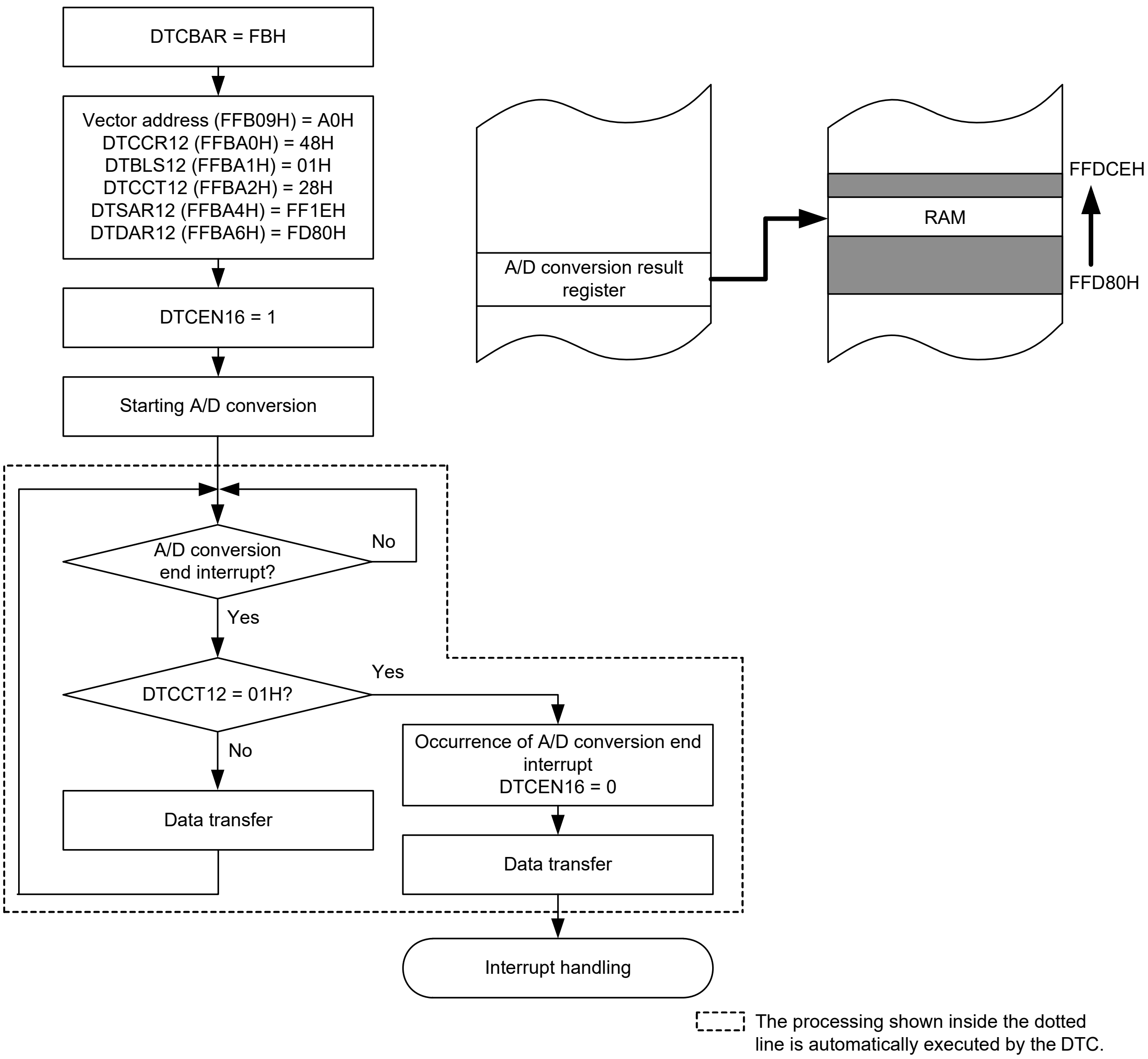
X: 0 or 1

(1) Example 1 of using normal mode: Consecutively capturing A/D conversion results

The DTC is activated by an A/D conversion end interrupt and the value of the A/D conversion result register is transferred to RAM.

- The vector address is FFB09H and control data is allocated at FFBA0H to FFBA7H
- Transfers 2-byte data of the A/D conversion result register (FFF1EH, FFF1FH) to 80 bytes of FFD80H to FFDCEH of RAM 40 times

Figure 19 - 16 Example 1 of using normal mode: Consecutively capturing A/D conversion results



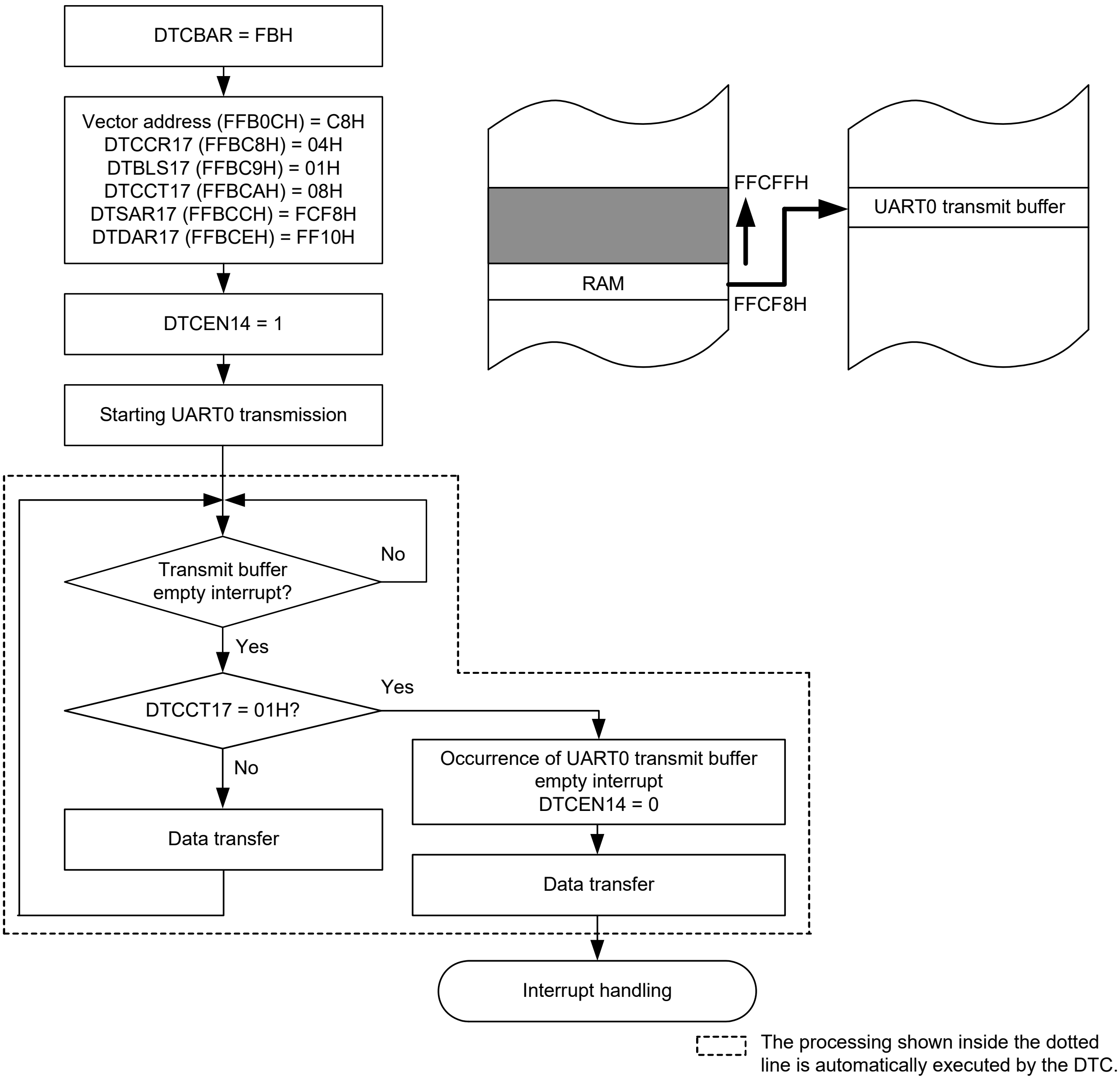
The value of the DTRLD12 register is not used because of normal mode, but initialize the register to 00H when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.

(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 FFB0BH and control data is allocated at FFBC8H to FFBCFH
- Transfers 8 bytes of FFCF8H to FFCFFH of RAM to the UART0 transmit buffer (FFF10H)

Figure 19 - 17 Example 2 of using normal mode: UART0 consecutive transmission



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 are automatically sent when the DTC is activated by a transmit buffer empty interrupt.

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 (i = 0 to 23) register and the address specified for the repeat area are initialized to continue transfers. When the data transfer causing the DTCCTj register value to change to 0 is performed while the RPTINT bit in the DTCCRj register is 1 (interrupt generation enabled), the DTC generates an interrupt request corresponding to the activation source to the interrupt controller during DTC operation, and sets the corresponding bit among bits DTCENi0 to DTCENi7 to 0 (activation disabled). When the RPTINT bit in the DTCCRj register is 0 (interrupt generation disabled), no interrupt request is generated even if the data transfer causing the DTCCTj register value to change to 0 is performed. Also, bits DTCENi0 to DTCENi7 are not set to 0. Table 19 - 8 lists Register Functions in Repeat Mode. Figure 19 - 18 shows Data Transfers in Repeat Mode.

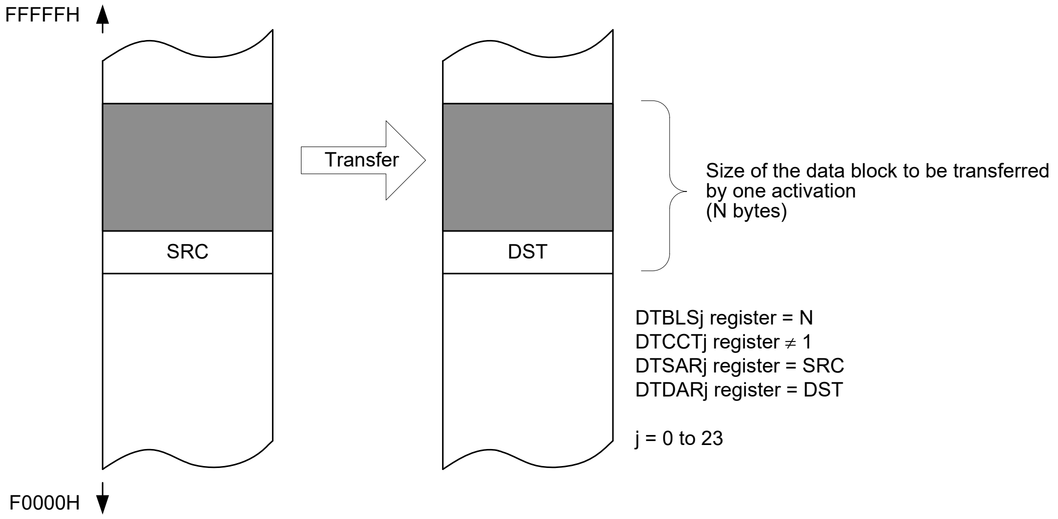
Table 19 - 8 Register Functions in Repeat Mode

Register Name	Symbol	Function
DTC block size register j	DTBLSj	Size of the data block to be transferred by one activation
DTC transfer count register j	DTCCTj	Number of data transfers
DTC transfer count reload register j	DTRLDj	This register value is reloaded to the DTCCT register (the number of transfers is initialized).
DTC source address register j	DTSARj	Data transfer source address
DTC destination address register j	DTDARj	Data transfer destination address

Remark j = 0 to 23

Figure 19 - 18 Data Transfers in Repeat Mode

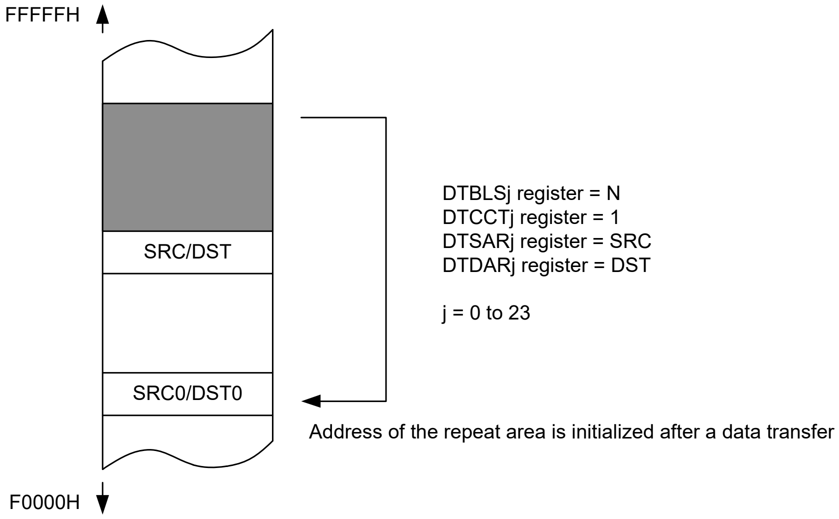
DTCCTj register ≠ 1



DTCCR Register Setting				Source Address Control	Destination Address Control	Source Address after Transfer	Destination Address after Transfer
DAMOD	SAMOD	RPTSEL	MODE				
0	X	1	1	Repeat area	Fixed	SRC + N	DST
1	X	1	1	Repeat area	Incremented	SRC + N	DST + N
X	0	0	1	Fixed	Repeat area	SRC	DST + N
X	1	0	1	Incremented	Repeat area	SRC + N	DST + N

X: 0 or 1

DTCCTj register = 1



DTCCR Register Setting				Source Address Control	Destination Address Control	Source Address after Transfer	Destination Address after Transfer
DAMOD	SAMOD	RPTSEL	MODE				
0	X	1	1	Repeat area	Fixed	SRC0	DST
1	X	1	1	Repeat area	Incremented	SRC0	DST + N
X	0	0	1	Fixed	Repeat area	SRC	DST0
X	1	0	1	Incremented	Repeat area	SRC + N	DST0

SRC0: Initial source address value
DST0: Initial destination address value
X: 0 or 1

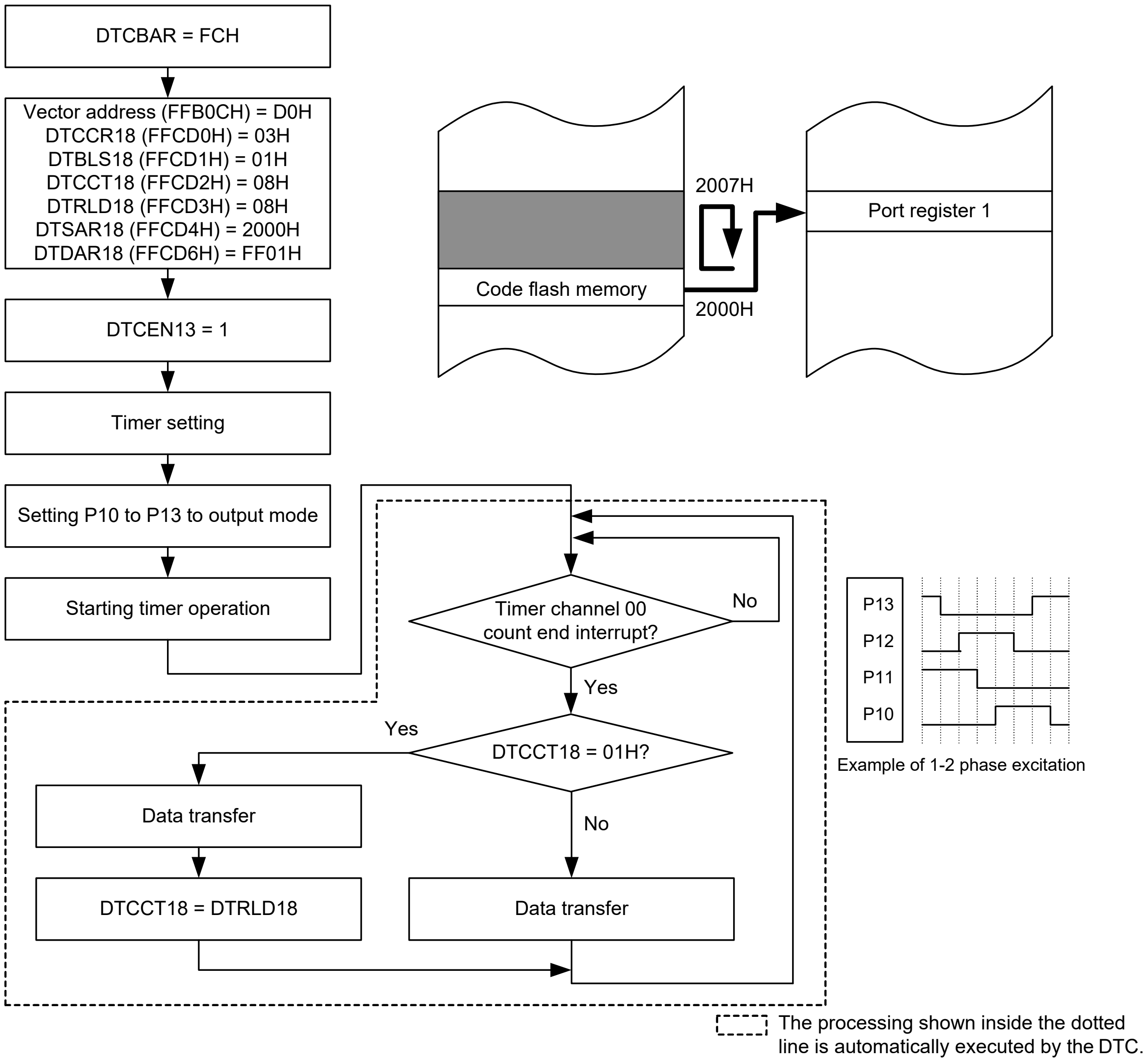
Caution 1. When repeat mode is used, the lower 8 bits of the initial value for the repeat area address must be 00H.
Caution 2. When repeat mode is used, the data size of the repeat area must be set to 255 bytes or less.

(1) Example 1 of using repeat mode: Outputting a stepping motor control pulse using ports

The DTC is activated using the interval timer function of channel 0 of timer array unit 0, and the pattern of the motor control pulse stored in the code flash memory is transferred to the general-purpose port.

- The vector address is FFB0CH and control data is allocated at FFCD0H to FFCD7H
- Transfers 8-byte data of 02000H to 02007H of the code flash memory from the mirror area (F2000H to F2007H) to port register 1 (FFF01H)
- A repeat mode interrupt is disabled

Figure 19 - 19 Example 1 of using repeat mode: Outputting a stepping motor control pulse using ports



To stop the output, stop the timer first and then clear DTCEN13.

19.4.4 Chain Transfers

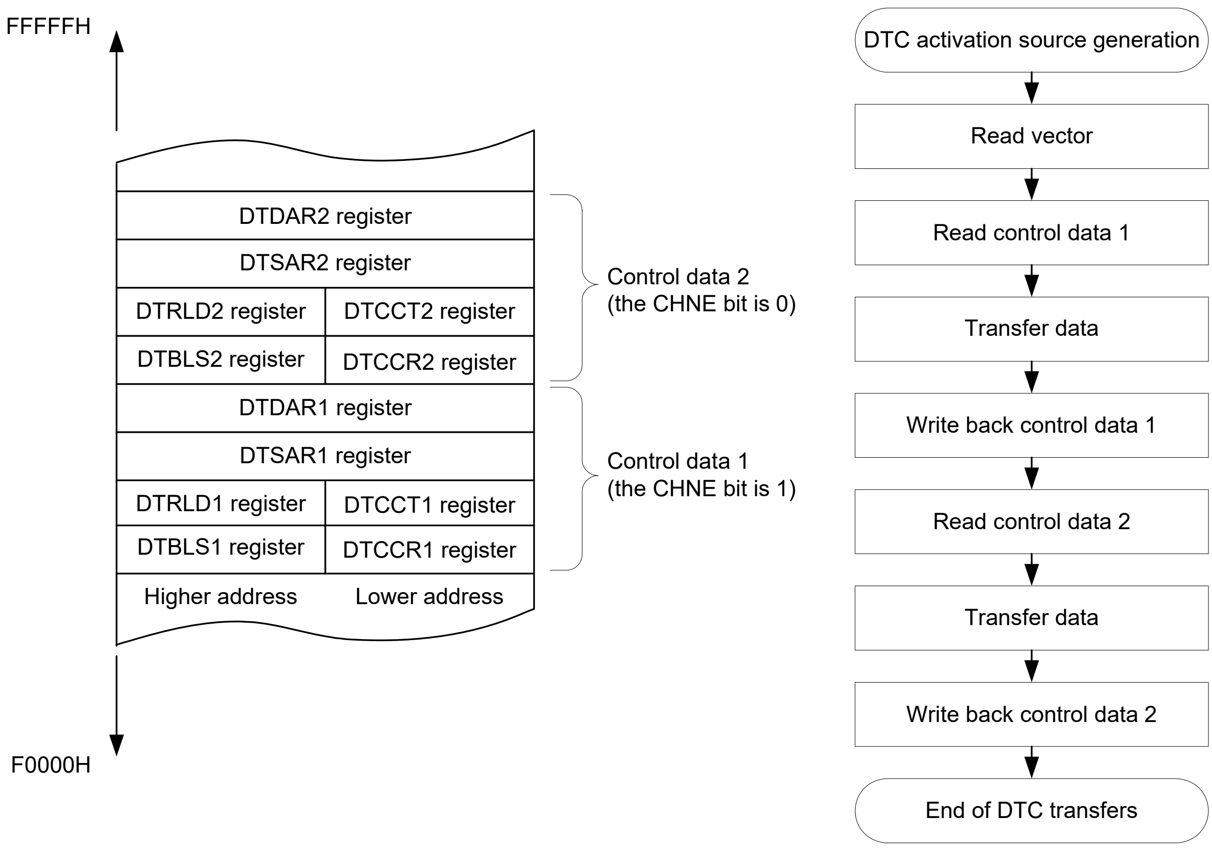
When the CHNE bit in the DTCCRj (j = 0 to 22) register is 1 (chain transfers enabled), multiple data transfers can be continuously performed by one activation source.

When the DTC is activated, one control data is selected according to the data read from the vector address corresponding to the activation source, and the selected control data is read from the DTC control data area. When the CHNE bit for the control data is 1 (chain transfers enabled), the next control data immediately following the current control data is read and transferred after the current transfer is completed. This operation is repeated until the data transfer with the control data for which the CHNE bit is 0 (chain transfers disabled) is completed.

When chain transfers are performed using multiple control data, the number of transfers set for the first control data is enabled, and the number of transfers set for the second and subsequent control data to be processed will be invalid.

Figure 19 - 20 shows Data Transfers during Chain Transfers.

Figure 19 - 20 Data Transfers during Chain Transfers



Note 1. Set the CHNE bit in the DTCCR23 register to 0 (chain transfers disabled).

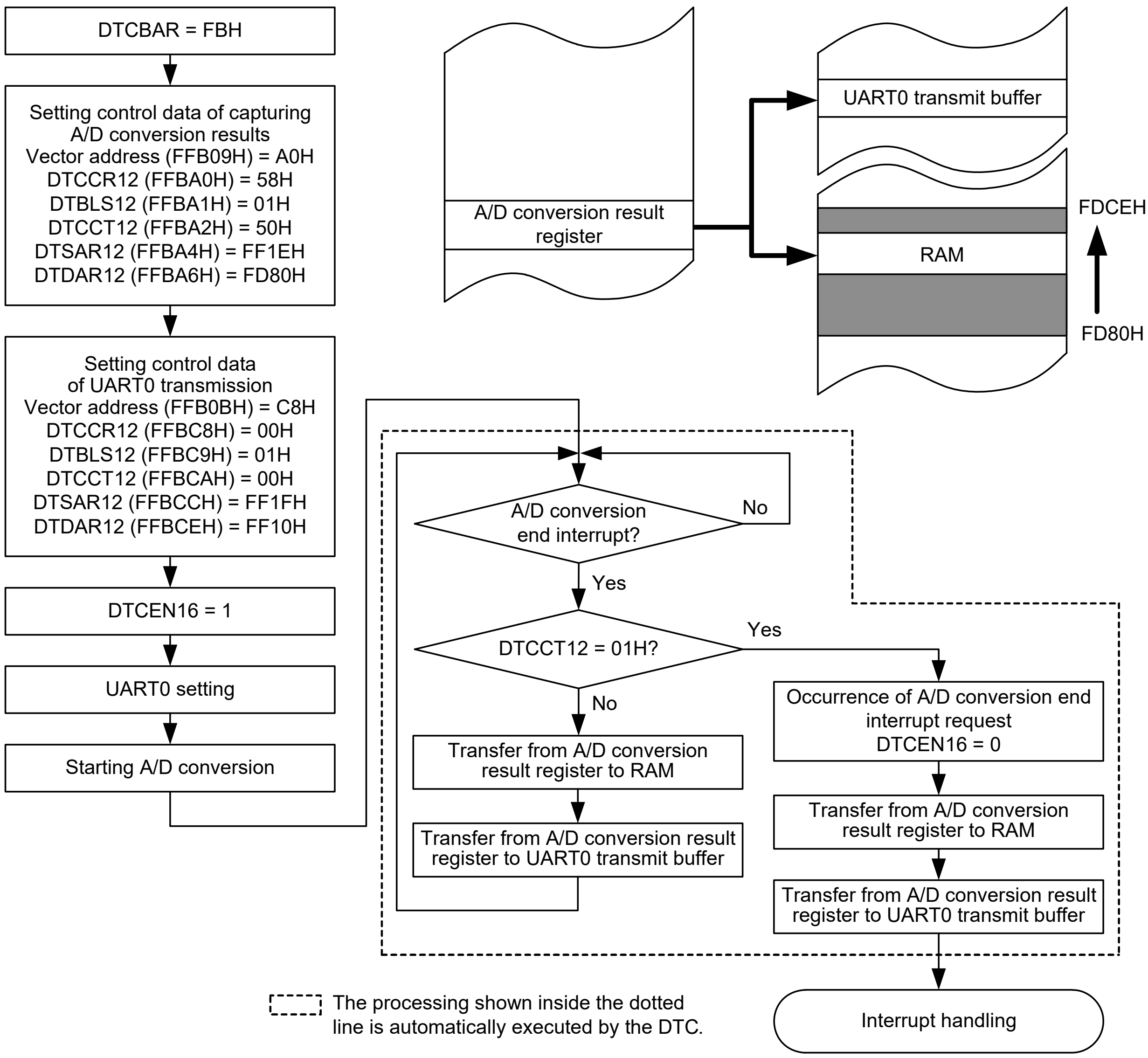
Note 2. During chain transfers, bits DTCENi0 to DTCENi7 (i = 0 to 2) in the DTCENi register are not set to 0 (activation disabled) for the second and subsequent transfers. Also, no interrupt request is generated.

(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 FFB09H
- Control data of capturing A/D conversion results is allocated at FFBA0H to FFBA7H
- Control data of UART0 transmission is allocated at FFBA8H at FFBAFH
- Transfers 2-byte data of the A/D conversion result register (FFF1FH, FFF1EH) to FFD80H to FFDCFH of RAM, and transfers the upper 1 byte (FFF1FH) of the A/D conversion result register to the UART transmit buffer (FFF10H)

Figure 19 - 21 Example of using chain transfers: Consecutively capturing A/D conversion results and UART0 transmission



19.5 Notes on DTC

19.5.1 Setting DTC Control Data and Vector Table

- Do not access the DTC extended special function register (2nd SFR), the DTC control data area, the DTC vector table area, or the general-register (FFEE0H to FFEFFH) space using a DTC transfer.
- Modify the DTC base address register (DTCBAR) while all DTC activation sources are set to activation disabled.
- Do not rewrite the DTC base address register (DTCBAR) twice or more.
- Modify the data of the DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, or DTDARj register when the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi (i = 0 to 2) register 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 DTCENi0 to DTCENi7 in the DTCENi (i = 0 to 2) register is 0 (activation disabled).
- Do not allocate RAM addresses which are used as a DTC transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

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 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 self-programming and data-flash functions.
  R5F117xC (x = A, B, G): FF300H to FF709H
- 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.
  R5F117xC (x = A, B, G): FF700H to FF8FFH
- Initialize the DTRLD register to 00H even in normal mode when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.

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 manipulation instructions for IFxx, MKxx, PRxx, and PSW, and an 8-bit manipulation instruction that has the ES register as operand
- Instruction for accessing the data flash memory
- Instruction of Multiply, Divide, Multiply & Accumulate (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 Space

When accessing the data flash space 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 ← The wait of three clock cycles occurs.
MOV A, ! Data Flash space

19.5.5 Number of DTC Execution Clock Cycles

Table 19 - 9 lists the Operations Following DTC Activation and Required Number of Cycles for each operation.

Table 19 - 9 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 - 10 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 - 11 Number of Clock Cycles Required for One Data Read/Write Operation**.

Table 19 - 10 Number of Clock Cycles Required for Control Data Write-Back Operation

DTCCR Register Setting				Address Setting		Control Register to be Written Back				Number of Clock Cycles
DAMOD	SAMOD	RPTSEL	MODE	Source	Destination	DTCCTj Register	DTRLDj Register	DTSARj Register	DTDARj Register	
0	0	X	0	Fixed	Fixed	Written back	Written back	Not written back	Not written back	1
0	1	X	0	Incremented	Fixed	Written back	Written back	Written back	Not written back	2
1	0	X	0	Fixed	Incremented	Written back	Written back	Not written back	Written back	2
1	1	X	0	Incremented	Incremented	Written back	Written back	Written back	Written back	3
0	X	1	1	Repeat area	Fixed	Written back	Written back	Written back	Not written back	2
1	X	1	1		Incremented	Written back	Written back	Written back	Written back	3
X	0	0	1	Fixed	Repeat area	Written back	Written back	Not written back	Written back	2
X	1	0	1	Incremented		Written back	Written back	Written back	Written back	3

Remark j = 0 to 23; X: 0 or 1

Table 19 - 11 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 states Note
Data write	1	—	—	1	1	1 + number of wait states Note

Note The number of wait states differs depending on the specifications of the register allocated to the extended special function register (2nd SFR) to be accessed.

19.5.6 DTC Response Time

Table 19 - 12 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 clocks.

Table 19 - 12 DTC Response Time

	Minimum Time	Maximum Time
Response Time	3 clocks	19 clocks

Note that the response from the DTC may be further delayed under the following cases. The number of delayed clock cycles differs depending on the conditions.

- When executing an instruction from the internal RAM
  Maximum response time: 20 clocks
- When executing a DTC pending instruction (refer to **19.5.3 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.
- When accessing the TRJ0 register that a wait occurs
  Maximum response time: Maximum response time for each condition + 1 clock

Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU/peripheral hardware clock)

19.5.7 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.
- When an 8-bit interval timer or the 12-bit interval timer is selected as a DTC activation source and DTC transfer is to proceed again following the completion of a previous DTC transfer activated by the same source, set the corresponding DTCENi0 to DTCENi7 bit in the DTCENi (i = 0 to 2) register to 1 (enabling activation) after one cycle of the operating clock for the timer.
- 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 is generated after completion of the transfer. Therefore, enable DTC activation after confirming the comparator monitor flag (CnMON) as necessary. (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 > IVREF
  - 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 < IVREF

19.5.8 Operation in Standby Mode Status

Status	DTC Operation
HALT mode	Operable (Operation is disabled while in the low power consumption RTC mode)
STOP mode	DTC activation sources can be accepted Note 2
SNOOZE mode	Operable Notes 1, 3, 4, 5

Note 1. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock (f_{IH}) or middle-speed on-chip oscillator clock (f_{IM}) is selected as f_{CLK}.

Note 2. In the STOP mode, detecting a DTC activation source enables transition to SNOOZE mode and DTC transfer. After completion of transfer, the system returns to the STOP mode. However, since the code flash memory and the data flash memory are stopped during the SNOOZE mode, the flash memory cannot be set as the transfer source.

Note 3. When a transfer end interrupt is set as a DTC activation source from the CSIp SNOOZE mode function, release the SNOOZE mode using the transfer end interrupt to start CPU processing after completion of DTC transfer, or use a chain transfer to set CSIp reception again (writing 1 to the STm0 bit, writing 0 to the SWCm bit, setting of the SSCm register, and writing 1 to the SSm0 bit).

Note 4. When a transfer end interrupt is set as a DTC activation source from the UARTq SNOOZE mode function, release the SNOOZE mode using the transfer end interrupt to start CPU processing after completion of DTC transfer, or use a chain transfer to set UARTq reception again (writing 1 to the STm1 bit, writing 0 to the SWCm bit, setting of the SSCm register, and writing 1 to the SSm1 bit).

Note 5. When an A/D conversion end interrupt is set as a DTC activation source from the A/D converter SNOOZE mode function, release the SNOOZE mode using the A/D conversion end interrupt to start CPU processing after completion of DTC transfer, or use a chain transfer to set the A/D converter SNOOZE mode function again (writing 0 to the AWC bit and then writing 1 to the AWC bit).

CHAPTER 20 EVENT LINK CONTROLLER (ELC)

20.1 Functions of ELC

The event link controller (ELC) mutually connects (links) events output from each peripheral function. By linking events, it becomes possible to coordinate operation between peripheral functions directly without going through the CPU.

The ELC has the following functions.

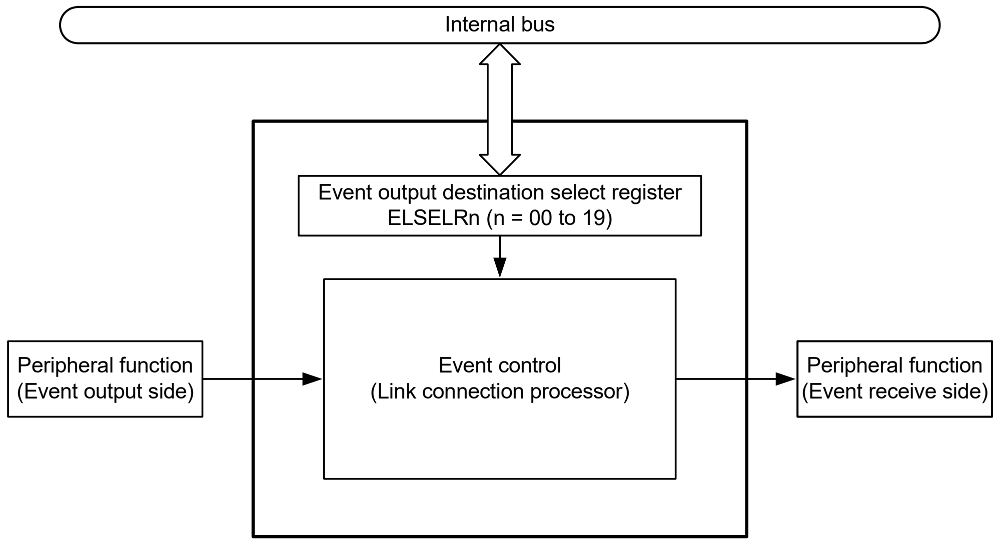
• Capable of directly linking event signals from 20 types of peripheral functions to specified peripheral functions

• Event signals can be used as activation sources for operating any one of seven types of peripheral functions

20.2 Configuration of ELC

Figure 20 - 1 shows the ELC Block Diagram.

Figure 20 - 1 ELC Block Diagram



20.3 Registers Controlling ELC

Table 20 - 1 lists the Registers Controlling ELC.

Table 20 - 1 Registers Controlling ELC

Register name	Symbol
Event output destination select register 00	ELSELR00
Event output destination select register 01	ELSELR01
Event output destination select register 02	ELSELR02
Event output destination select register 03 Note 1	ELSELR03
Event output destination select register 04 Note 2	ELSELR04
Event output destination select register 05 Note 3	ELSELR05
Event output destination select register 06 Note 2	ELSELR06
Event output destination select register 07 Note 4	ELSELR07
Event output destination select register 08	ELSELR08
Event output destination select register 09	ELSELR09
Event output destination select register 10	ELSELR10
Event output destination select register 11	ELSELR11
Event output destination select register 12	ELSELR12
Event output destination select register 13	ELSELR13
Event output destination select register 14	ELSELR14
Event output destination select register 15	ELSELR15
Event output destination select register 16	ELSELR16
Event output destination select register 17	ELSELR17
Event output destination select register 18	ELSELR18
Event output destination select register 19	ELSELR19

Note 1. Do not set any value other than the initial value (event link disabled) in 20-pin products.
Note 2. Do not set any value other than the initial value (event link disabled) in 32-, 30-, 24-, and 20-pin products.
Note 3. Do not set any value other than the initial value (event link disabled) in 32-, 24-, and 20-pin products.
Note 4. Do not set any value other than the initial value (event link disabled) in 30- and 20-pin products.

20.3.1 Event output destination select register n (ELSELRn) (n = 00 to 19)

An ELSELRn register links each event signal to an operation of an event-receiving peripheral function (link destination peripheral function) after reception.
Do not set multiple event inputs to the same event output destination (event receive side). The operation of the event-receiving peripheral function will become undefined, and event signals may not be received correctly. In addition, do not set the event link generation source and the event link output destination to the same function.
Set an ELSELRn register during a period when no event output peripheral functions are generating event signals and the function of the event output destination (event receive side) is stopped.
Table 20 - 2 lists the Correspondence Between ELSELRn (n = 00 to 19) Registers and Peripheral Functions, and Table 20 - 3 lists the Correspondence Between Values Set to ELSELRn (n = 00 to 19) Registers and Operation of Link Destination Peripheral Functions at Reception.

Figure 20 - 2 Format of Event output destination select register n (ELSELRn)

Address: F0240H (ELSELR00) to F0253H (ELSELR19) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ELSELRn	0	0	0	0	ELSELn3	ELSELn2	ELSELn1	ELSELn0

ELSELn3	ELSELn2	ELSELn1	ELSELn0	Event Link Selection
0	0	0	0	Event link disabled
0	0	0	1	Select operation of peripheral function 1 to link Note
0	0	1	0	Select operation of peripheral function 2 to link Note
0	0	1	1	Select operation of peripheral function 3 to link Note
0	1	0	0	Select operation of peripheral function 4 to link Note
0	1	0	1	Select operation of peripheral function 5 to link Note
0	1	1	0	Select operation of peripheral function 6 to link Note
0	1	1	1	Select operation of peripheral function 7 to link Note
Other than above				Setting prohibited

Note See **Table 20 - 3 Correspondence Between Values Set to ELSELRn (n = 00 to 19) Registers and Operation of Link Destination Peripheral Functions at Reception**.

Table 20 - 2 Correspondence Between ELSELRn (n = 00 to 19) Registers and Peripheral Functions

Register Name	Event Generator (Output Origin of Event Input n)	Event Description
ELSELR00	External interrupt edge detection 0	INTP0
ELSELR01	External interrupt edge detection 1	INTP1
ELSELR02	External interrupt edge detection 2	INTP2
ELSELR03	External interrupt edge detection 3	INTP3
ELSELR04	External interrupt edge detection 4	INTP4
ELSELR05	External interrupt edge detection 5	INTP5
ELSELR06	External interrupt edge detection 6	INTP6
ELSELR07	Key return signal detection	INTKR
ELSELR08	RTC2 fixed-cycle signal/Alarm match detection	INTRTC
ELSELR09	12-bit interval timer interval signal detection	INTIT
ELSELR10	8-bit interval timer channel 00 compare match or 16-bit interval timer channel 0 compare match (cascaded)	INTIT00
ELSELR11	8-bit interval timer channel 01 compare match	INTIT01
ELSELR12	8-bit interval timer channel 10 compare match or 16-bit interval timer channel 1 compare match (cascaded)	INTIT10
ELSELR13	8-bit interval timer channel 11 compare match	INTIT11
ELSELR14	TAU channel 00 count end/capture end	INTTM00
ELSELR15	TAU channel 01 count end/capture end	INTTM01
ELSELR16	TAU channel 02 count end/capture end	INTTM02
ELSELR17	TAU channel 03 count end/capture end	INTTM03
ELSELR18	Comparator detection 0	INTCMP0
ELSELR19	Comparator detection 1	INTCMP1

Table 20 - 3 Correspondence Between Values Set to ELSELRn (n = 00 to 19) Registers and Operation of Link Destination Peripheral Functions at Reception

Bits ELSELn3 to ELSELn0 in ELSELRn Register	Link Destination Number	Link Destination Peripheral Function	Operation When Receiving Event
0001B	1	A/D converter	A/D conversion starts
0010B	2	Timer input of timer array unit 0 channel 0 Note 1	Delay counter, input pulse interval measurement, external event counter
0011B	3	Timer input of timer array unit 0 channel 1 Note 2	Delay counter, input pulse interval measurement, external event counter
0100B	4	Operation amplifier 0	Operation starts
0101B	5	Operation amplifier 1	Operation starts
0110B	6	Operation amplifier 2	Operation starts
0111B	7	Operation amplifier 3	Operation starts

Note 1. To select the timer input of timer array unit 0 channel 0 as the link destination peripheral function, set the operating clock for channel 0 to f_{CLK} using timer clock select register 0 (TPS0), set the noise filter of the TI00 pin to OFF (TNFEN00 = 0) using noise filter enable register 1 (NFEN1), and then set the timer output used for channel 0 to an event input signal from the ELC using timer input select register 0 (TIS0).

Note 2. To select the timer input of timer array unit 0 channel 1 as the link destination peripheral function, set the operating clock for channel 1 to f_{CLK} using timer clock select register 0 (TPS0), set the noise filter of the TI01 pin to OFF (TNFEN01 = 0) using noise filter enable register 1 (NFEN1), and then set the timer output used for channel 1 to an event input signal from the ELC using timer input select register 0 (TIS0).

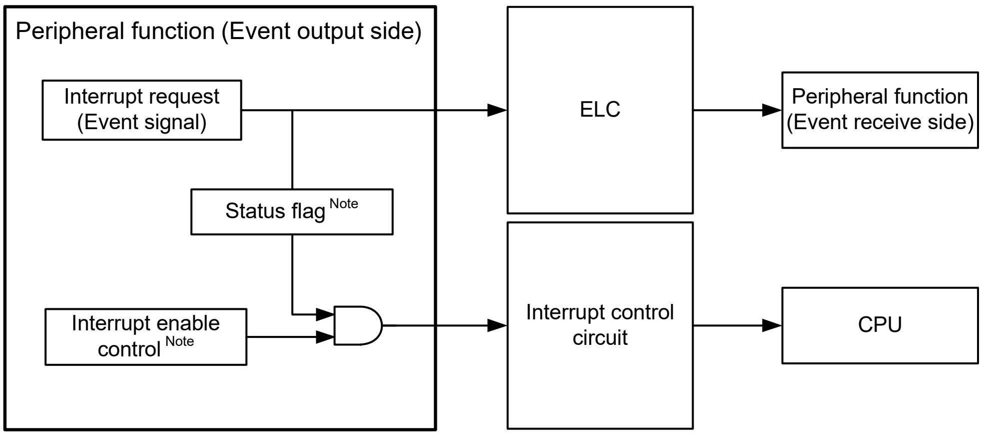
20.4 ELC Operation

The path for using an event signal generated by a peripheral function as an interrupt request to the interrupt control circuit is independent from the path for using it as an ELC event. Therefore, each event signal can be used as an event signal for operation of an event-receiving peripheral function, regardless of interrupt control.

Figure 20 - 3 shows the Relationship Between Interrupt Handling and ELC. The figure show an example of an interrupt request status flag and a peripheral function possessing the enable bits that control enabling/disabling of such interrupts.

A peripheral function which receives an event from the ELC will perform the operation corresponding to the event-receiving peripheral function after reception of an event (See **Table 20 - 3 Correspondence Between Values Set to ELSELRn (n = 00 to 19) Registers and Operation of Link Destination Peripheral Functions at Reception**).

Figure 20 - 3 Relationship Between Interrupt Handling and ELC



Note Not available depending on the peripheral function.

Table 20 - 4 lists the Response of Peripheral Functions That Receive Events.

Table 20 - 4 Response of Peripheral Functions That Receive Events

Event Receiver No.	Event Link Destination Function	Operation after Event Reception	Response
1	A/D converter	A/D conversion	An event from the ELC is directly used as a hardware trigger of A/D conversion.
2	Timer array unit 0 Timer input of channel 0	Delay counter Input pulse width measurement External event counter	The edge is detected 3 or 4 cycles of f_{CLK} after an ELC event is generated.
3	Timer array unit 0 Timer input of channel 1	Delay counter Input pulse width measurement External event counter	The edge is detected 3 or 4 cycles of f_{CLK} after an ELC event is generated.
4	Operation amplifier 0	Operation starts	An event from the ELC is directly used as a hardware trigger for the operational amplifier.
5	Operation amplifier 1	Operation starts	An event from the ELC is directly used as a hardware trigger for the operational amplifier.
6	Operation amplifier 2	Operation starts	An event from the ELC is directly used as a hardware trigger for the operational amplifier.
7	Operation amplifier 3	Operation starts	An event from the ELC is directly used as a hardware trigger for the operational amplifier.

CHAPTER 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.

		20-pin	24-pin	30-pin	32-pin	48-pin
Maskable interrupts	External	3	5	5	5	8
	Internal	22	22	24	24	24

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, PR10L, PR10H, PR11L, PR11H, PR12L).
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. Default priority, see **Table 21 - 1**.
A standby release signal is generated and STOP, HALT, and SNOOZE modes are released.
External interrupt requests and internal interrupt requests are provided as maskable interrupts.

(2) Software interrupt
This is a vectored interrupt generated by executing the BRK instruction. It is acknowledged even when interrupts are disabled. The software interrupt does not undergo interrupt priority control.

21.2 Interrupt Sources and Configuration

Interrupt sources include maskable interrupts and software interrupts. In addition, they also have up to seven reset sources (see **Table 21 - 1**). The vector codes that store the program start address when branching due to the generation of a reset or various interrupt requests are two bytes each, so interrupts jump to a 64 K address of 00000H to 0FFFFH.

Table 21 - 1 Interrupt Source List (1/2)

Interrupt Type	Default Priority Note 1	Interrupt Source: Name	Interrupt Source: Trigger	Internal/External	Vector Table Address	Basic Configuration Type Note 2	48-pin	32-pin	30-pin	24-pin	20-pin
Maskable	0	INTWDTI	Watchdog timer interval Note 3 (75% of overflow time + 1/2 f_{IL})	Internal	0004H	(A)	√	√	√	√	√
	1	INTLVI	Voltage detection Note 4		0006H		√	√	√	√	√
	2	INTP0	Pin input edge detection	External	0008H	(B)	√	√	√	√	√
	3	INTP1			000AH		√	√	√	√	√
	4	INTP2			000CH		√	√	√	√	√
	5	INTP3			000EH		√	√	√	√	—
	6	INTP4			0010H		√	—	—	—	—
	7	INTP5			0012H		√	—	√	—	—
	8	INTP6			0014H		√	—	—	—	—
	9	INTST0	UART0 transmission transfer end or buffer empty interrupt	Internal	0016H	(A)	√	√	√	√	√
		INTCSI00	CSI00 transfer end or buffer empty interrupt				√	√	√	√	√
		INTIIC00	IIC00 transfer end				√	√	√	√	√
	10	INTSR0	UART0 reception transfer end		0018H		√	√	√	√	√
		INTCSI01	CSI01 transfer end or buffer empty interrupt				√	√	—	√	—
		INTIIC01	IIC01 transfer end				√	√	—	√	—
	11	INTSRE0	UART0 reception communication error occurrence		001EH		√	√	√	√	√
	12	INTTM00	End of TAU channel 00 count or capture (at 16-bit operation or lower 8-bit operation)		0020H		√	√	√	√	√
	13	INTRTIT	RTC2 correction timing		0022H		√	√	√	—	—
	14	INTFM	Frequency measurement end		0024H		√	√	√	—	—
	15	INTTM01H	End of TAU channel 01 count or capture (at higher 8-bit operation)		0026H		√	√	√	√	√
	16	INTTM03H	End of TAU channel 03 count or capture (at higher 8-bit operation)		0028H		√	√	√	√	√
	17	INTTM01	End of TAU channel 01 count or capture (at 16-bit operation or lower 8-bit operation)		002AH		√	√	√	√	√
	18	INTTM02	End of TAU channel 02 count or capture (at 16-bit operation or lower 8-bit operation)		002CH		√	√	√	√	√
	19	INTTM03	End of TAU channel 03 count or capture (at 16-bit operation or lower 8-bit operation)		002EH		√	√	√	√	√
	20	INTAD	End of A/D conversion		0034H		√	√	√	√	√
	21	INTRTC	Fixed-cycle signal of real-time clock 2/alarm match detection		0036H		√	√	√	√	√
	22	INTIT	12-bit interval timer interval signal detection		0038H		√	√	√	√	√

Note 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 31 indicates the lowest priority.

Note 2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figures 21 - 1 and 21 - 2.

Note 3. When bit 7 (WDTINT) of the option byte (000C0H) is set to 1.

Note 4. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is cleared to 0.

Table 21 - 1 Interrupt Source List (2/2)

Interrupt Type	Default Priority Note 1	Interrupt Source: Name	Interrupt Source: Trigger	Internal/External	Vector Table Address	Basic Configuration Type Note 2	48-pin	32-pin	30-pin	24-pin	20-pin
Maskable	23	INTKR	Key return signal detection	External	003AH	(C)	√	√	—	√	—
	24	INTCMP0	Comparator detection 0	Internal	003CH	(A)	√	√	√	√	√
	25	INTCMP1	Comparator detection 1		003EH		√	√	√	√	√
	26	INTDOC	DOC operation result detection		0040H		√	√	√	√	√
	27	INTIT00	8-bit interval timer channel 00 compare match or 16-bit interval timer channel 0 compare match (cascaded)		0044H		√	√	√	√	√
	28	INTIT01	8-bit interval timer channel 01		0046H		√	√	√	√	√
	29	INTIT10	8-bit interval timer channel 10 compare match or 16-bit interval timer channel 1 compare match (cascaded)		0048H		√	√	√	√	√
	30	INTIT11	8-bit interval timer channel 11		004AH		√	√	√	√	√
	31	INTFL	Reserved		0052H		√	√	√	√	√
Software	—	BRK	Execution of BRK instruction	—	007EH	(D)	√	√	√	√	√
Reset	—	RESET	$\overline{\text{RESET}}$ pin input	—	0000H	—	√	√	√	√	√
		POR	Power-on-reset				√	√	√	√	√
		LVD	Voltage detection Note 3				√	√	√	√	√
		WDT	Overflow of watchdog timer				√	√	√	√	√
		TRAP	Execution of illegal instruction Note 4				√	√	√	√	√
		IAW	Illegal-memory access				√	√	√	√	√
		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 31 indicates the lowest priority.

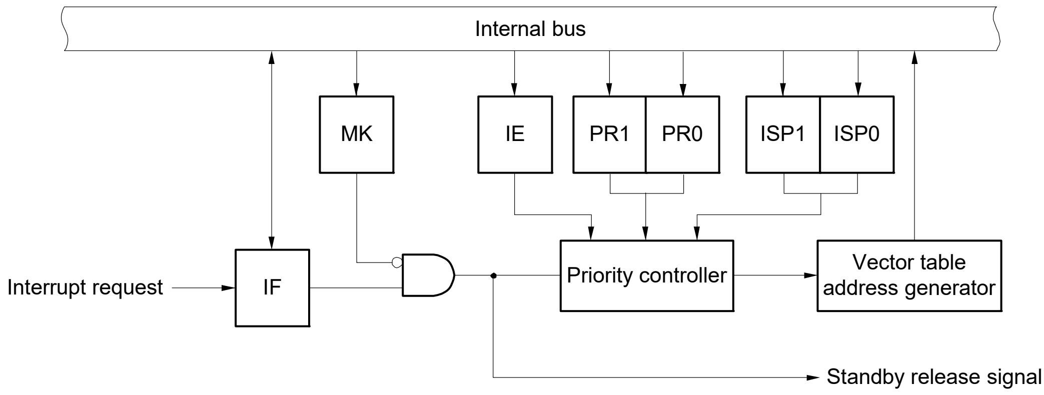
Note 2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figures 21 - 1 and 21 - 2.

Note 3. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is set to 1.

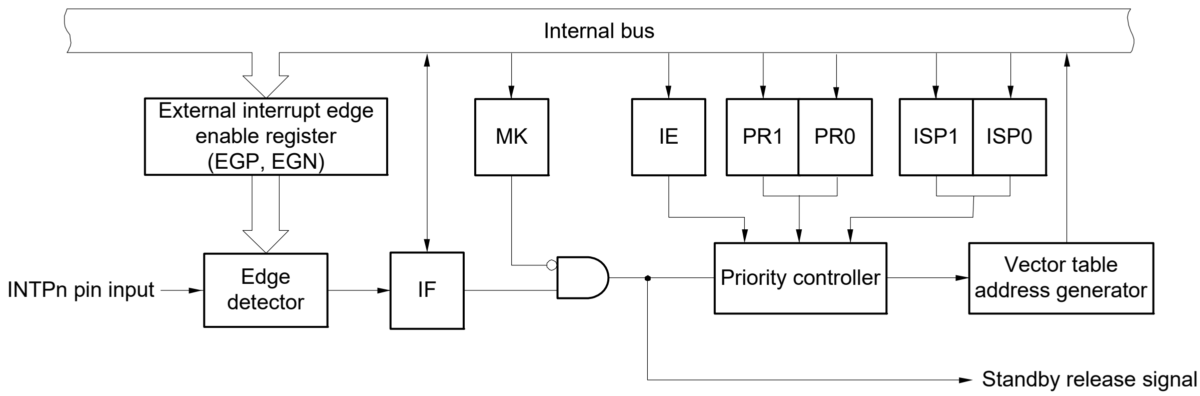
Note 4. When the instruction code in FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Figure 21 - 1 Basic Configuration of Interrupt Function (1/2)

(A) Internal maskable interrupt



(B) External maskable interrupt (INTPn)

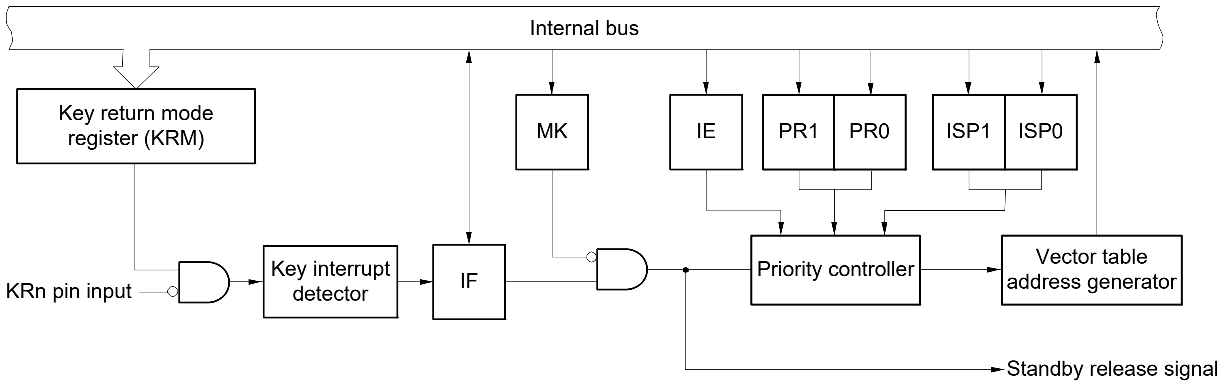


IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

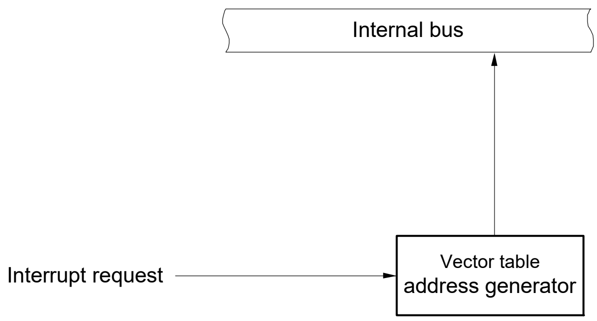
Remark 20-pin: n = 0 to 2
24, 30-pin: n = 0 to 3
32-pin: n = 0 to 3, 5
48-pin: n = 0 to 6

Figure 21 - 2 Basic Configuration of Interrupt Function (2/2)

(C) External maskable interrupt (INTKR)



(D) Software interrupt



IF: Interrupt request flag
IE: Interrupt enable flag
ISP0: In-service priority flag 0
ISP1: In-service priority flag 1
MK: Interrupt mask flag
PR0: Priority specification flag 0
PR1: Priority specification flag 1

Remark 24, 32-pin: n = 0 to 2
48-pin: n = 0 to 3

21.3 Registers Controlling Interrupt Functions

The following 6 types of registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L)
- Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L)
- Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR10L, PR10H, PR11L, PR11H, PR12L)
- External interrupt rising edge enable register (EGP0)
- External interrupt falling edge enable register (EGN0)
- Program status word (PSW)

Table 21 - 2 show a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 21 - 2 Flags Corresponding to Interrupt Request Sources (1/2)

Interrupt Source	Interrupt Request Flag	Register	Interrupt Mask Flag	Register	Priority Specification Flag	Register	48-pin	32-pin	30-pin	24-pin	20-pin
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L	√	√	√	√	√
INTLVI	LVIIF		LVIMK		LVIPR0, LVIPR1		√	√	√	√	√
INTP0	PIF0		PMK0		PPR00, PPR10		√	√	√	√	√
INTP1	PIF1		PMK1		PPR01, PPR11		√	√	√	√	√
INTP2	PIF2		PMK2		PPR02, PPR12		√	√	√	√	√
INTP3	PIF3		PMK3		PPR03, PPR13		√	√	√	√	—
INTP4	PIF4		PMK4		PPR04, PPR14		√	—	—	—	—
INTP5	PIF5		PMK5		PPR05, PPR15		√	—	√	—	—
INTP6	PIF6	IF0H	PMK6	MK0H	PPR06, PPR16	PR00H, PR10H	√	—	—	—	—
INTST0 Note 1	STIF0		STMK0		STPR00, STPR10		√	√	√	√	√
INTCSI00 Note 1	CSIIF00		CSIMK00		CSIPR000, CSIPR100		√	√	√	√	√
INTIIC00 Note 1	IICIF00		IICMK00		IICPR000, IICPR100		√	√	√	√	√
INTSR0 Note 2	SRIF0		SRMK0		SRPR00, SRPR10		√	√	√	√	√
INTCSI01 Note 2	CSIIF01		CSIMK01		CSIPR001, CSIPR101		√	√	—	√	—
INTIIC01 Note 2	IICIF01		IICMK01		IICPR001, IICPR101		√	√	—	√	—
INTSRE0	SREIF0		SREMK0		SREPR00, SREPR10		√	√	√	√	√
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100		√	√	√	√	√
INTRTIT	RTITIF		RTITMK		RTITPR0, RTITPR1		√	√	√	—	—
INTFM	FMIF	IF1L	FMMK	MK1L	FMPR0, FMPR1	PR01L, PR11L	√	√	√	—	—
INTTM01H	TMIF01H		TMMK01H		TMPR001H, TMPR101H		√	√	√	√	√
INTTM03H	TMIF03H		TMMK03H		TMPR003H, TMPR103H		√	√	√	√	√
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		√	√	√	√	√
INTIT	TMKAIF		TMKAMK		TMKAPR0, TMKAPR1		√	√	√	√	√
INTKR	KRIF		KRMK		KRPR0, KRPR1		√	√	—	√	—
INTCMP0	CMPIF0		CMPMK0		CMPPR00, CMPPR01		√	√	√	√	√
INTCMP1	CMPIF1		CMPMK1		CMPPR01, CMPPR11		√	√	√	√	√
INTDOC	DOCIF		DOCMK		DOCPR0, DOCPR1		√	√	√	√	√

Note 1. If one of the interrupt sources INTST0, INTCSI00, and INTIIC00 is generated, bit 1 of the IF0H register is set to 1. Bit 1 of the MK0H, PR00H, and PR10H registers supports these three interrupt sources.

Note 2. If one of the interrupt sources INTSR0, INTCSI01, and INTIIC01 is generated, bit 2 of the IF0H register is set to 1. Bit 2 of the MK0H, PR00H, and PR10H registers supports these three interrupt sources.

Table 21 - 2 Flags Corresponding to Interrupt Request Sources (2/2)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		48-pin	32-pin	30-pin	24-pin	20-pin
		Register		Register		Register					
INTIT00	ITIF00	IF2L	ITMK00	MK2L	ITPR000, ITPR100	PR02L, PR12L	√	√	√	√	√
INTIT01	ITIF01		ITMK01		ITPR001, ITPR101		√	√	√	√	√
INTIT10	ITIF10		ITMK10		ITPR010, ITPR110		√	√	√	√	√
INTIT11	ITIF11		ITMK11		ITPR011, ITPR111		√	√	√	√	√
INTFL	FLIF		FLMK		FLPR0, FLPR1		√	√	√	√	√

21.3.1 Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L)

The interrupt request flags are set to 1 when the corresponding interrupt request is generated or an instruction is executed. They are cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon reset signal generation.

When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is entered.

The IF0L, IF0H, IF1L, IF1H, and IF2L registers can be set by a 1-bit or 8-bit memory manipulation instruction. When the IF0L and IF0H registers and the IF1L and IF1H registers are combined to form 16-bit registers IF0 and IF1, they can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 21 - 3 Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L)

Address: FFFE0H After reset: 00H 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

Symbol	<7>	<6>	<5>	4	3	<2>	<1>	<0>
IF0H	RTITIF	TMIF00	SREIF0	0	0	SRIF0 CSIIF01 IICIF01	STIF0 CSIIF00 IICIF00	PIF6

Address: FFFE2H After reset: 00H R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	0	0	TMIF03	TMIF02	TMIF01	TMIF03H	TMIF01H	FMIF

Address: FFFE3H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1H	0	DOCIF	CMPIF1	CMPIF0	KRIF	TMKAIF	RTCIF	ADIF

Address: FFFD0H After reset: 00H R/W

Symbol	<7>	6	5	4	<3>	<2>	<1>	<0>
IF2L	FLIF	0	0	0	ITIF11	ITIF10	ITIF01	ITIF00

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

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 - 2. 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 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)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt.
The MK0L, MK0H, MK1L, MK1H, and MK2L registers can be set by a 1-bit or 8-bit memory manipulation instruction. When the MK0L and MK0H registers and the MK1L and MK1H registers are combined to form 16-bit registers MK0 and MK1, they can be set by a 16-bit memory manipulation instruction.
Reset signal generation sets these registers to FFH.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 21 - 4 Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L)

Address: FFFE4H After reset: FFH 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

Symbol	<7>	<6>	<5>	4	3	<2>	<1>	<0>
MK0H	RTITMK	TMMK00	SREMK0	1	1	SRMK0 CSIMK01 IICMK01	STMK0 CSIMK00 IICMK00	PMK6

Address: FFFE6H After reset: FFH R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	0	0	TMMK03	TMMK02	TMMK01	TMMK03H	TMMK01H	FMMK

Address: FFFE7H After reset: FFH R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1H	0	DOCMK	CMPMK1	CMPMK0	KRMK	TMKAMK	RTCMK	ADMK

Address: FFFD4H After reset: FFH R/W

Symbol	<7>	6	5	4	<3>	<2>	<1>	<0>
MK2L	FLMK	0	0	0	ITMK11	ITMK10	ITMK01	ITMK00

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 - 2. 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, PR10L, PR10H, PR11L, PR11H, PR12L)

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, or 2L).
The PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, and PR12L registers can be set by a 1-bit or 8-bit memory manipulation instruction. If the PR00L and PR00H registers, and the PR01L and PR01H registers, the PR10L and PR10H registers, and the PR11L and PR11H registers are combined to form 16-bit registers PR00, PR01, PR10, and PR11, they can be set by a 16-bit memory manipulation instruction.
Reset signal generation sets these registers to FFH.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 21 - 5 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR10L, PR10H, PR11L, PR11H, PR12L) (1/2)

Address: FFFE8H After reset: FFH 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

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

Symbol	<7>	<6>	<5>	4	3	<2>	<1>	<0>
PR00H	RTITPR0	TMPR000	SREPR00	1	1	SRPR00 CSIPR001 IICPR001	STPR00 CSIPR000 IICPR000	PPR06

Address: FFFEDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	4	3	<2>	<1>	<0>
PR10H	RTITPR1	TMPR100	SREPR10	1	1	SRPR10 CSIPR101 IICPR101	STPR10 CSIPR100 IICPR100	PPR16

Address: FFFEAH After reset: FFH R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
PR01L	0	0	TMPR003	TMPR002	TMPR001	TMPR003H	TMPR001H	FMPR0

Figure 21 - 6 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR10L, PR10H, PR11L, PR11H, PR12L) (2/2)

Address: FFFEEH After reset: FFH R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
PR11L	0	0	TMPR103	TMPR102	TMPR101	TMPR103H	TMPR101H	FMPR1

Address: FFFEBH After reset: FFH R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01H	0	DOCPR0	CMPPR01	CMPPR00	KRPR0	TMKAPR0	RTCPR0	ADPR0

Address: FFFEFH After reset: FFH R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11H	0	DOCPR1	CMPPR11	CMPPR10	KRPR1	TMKAPR1	RTCPR1	ADPR1

Address: FFFD8H After reset: FFH R/W

Symbol	<7>	6	5	4	<3>	<2>	<1>	<0>
PR02L	FLPR0	0	0	0	ITPR011	ITPR010	ITPR001	ITPR000

Address: FFFDCH After reset: FFH R/W

Symbol	<7>	6	5	4	<3>	<2>	<1>	<0>
PR12L	FLPR1	0	0	0	ITPR111	ITPR110	ITPR101	ITPR100

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 - 2. Be sure to set bits that are not available to the initial value.

21.3.4 External interrupt rising edge enable register (EGP0), external interrupt falling edge enable register (EGN0)

These registers specify the valid edge for INTP0 to INTP6.
The EGP0 and EGN0 registers can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears these registers to 00H.

Figure 21 - 7 Format of External Interrupt Rising Edge Enable Register (EGP0) and External Interrupt Falling Edge Enable Register (EGN0)

Address: FFF38H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	0	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FFF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	0	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 6)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 21 - 3 shows the Ports Corresponding to EGPn and EGNn Bits.

Table 21 - 3 Ports Corresponding to EGPn and EGNn Bits

Detection Enable Bit		Interrupt Request Signal	48-pin	32-pin	30-pin	24-pin	20-pin
EGP0	EGN0	INTP0	√	√	√	√	√
EGP1	EGN1	INTP1	√	√	√	√	√
EGP2	EGN2	INTP2	√	√	√	√	√
EGP3	EGN3	INTP3	√	√	√	√	—
EGP4	EGN4	INTP4	√	—	—	—	—
EGP5	EGN5	INTP5	√	—	√	—	—
EGP6	EGN6	INTP6	√	—	—	—	—

Caution **When the input port pins used for the external interrupt functions are switched to the output mode, the INTPn interrupt might be generated upon detection of a valid edge.**
When switching the input port pins to the output mode, set the port mode register (PMxx) to 0 after disabling the edge detection (by setting EGPn and EGNn to 0).

Remark 1. For edge detection port, see **2.1 Port Functions**.
Remark 2. n = 0 to 6

21.3.5 Program status word (PSW)

The program status word is a register used to hold the instruction execution result and the current status for an interrupt request. The IE flag that sets maskable interrupt enable/disable and the ISP0 and ISP1 flags that controls multiple interrupt servicing are mapped to the PSW.

Besides 8-bit read/write, this register can carry out operations using bit manipulation instructions and dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged, if the BRK instruction is executed, the contents of the PSW are automatically saved into a stack and the IE flag is reset to 0. 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 ISP0 and ISP1 flags. The PSW contents are also saved into the stack with the PUSH PSW instruction. They are restored from the stack with the RETI, RETB, and POP PSW instructions.

Reset signal generation sets PSW to 06H.

Figure 21 - 8 Configuration of Program Status Word

	7	6	5	4	3	2	1	0	After reset
PSW	IE	Z	RBS1	AC	RBS0	ISP1	ISP0	CY	06H

Z, RBS1, AC, RBS0, ISP1, ISP0, CY: Used when normal instruction is executed

ISP1	ISP0	Priority of interrupt currently being serviced
0	0	Enables interrupt of level 0 (while interrupt of level 1 or 0 is being serviced).
0	1	Enables interrupt of level 0 and 1 (while interrupt of level 2 is being serviced).
1	0	Enables interrupt of level 0 to 2 (while interrupt of level 3 is being serviced).
1	1	Enables all interrupts (waits for acknowledgment of an interrupt).

IE	Interrupt request acknowledgment enable/disable
0	Disabled
1	Enabled

21.4 Interrupt Servicing Operations

21.4.1 Maskable interrupt request acknowledgment

A maskable interrupt request becomes acknowledgeable when the interrupt request flag is set to 1 and the mask (MK) flag corresponding to that interrupt request is cleared to 0. A vectored interrupt request is acknowledged if interrupts are in the interrupt enabled state (when the IE flag is set to 1). However, a low-priority interrupt request is not acknowledged during servicing of a higher priority interrupt request.
The times from generation of a maskable interrupt request until vectored interrupt servicing is performed are listed in Table 21 - 4 below.
For the interrupt request acknowledgment timing, see **Figures 21 - 10** and **21 - 11**.

Table 21 - 4 Time from Generation of Maskable Interrupt Until Servicing

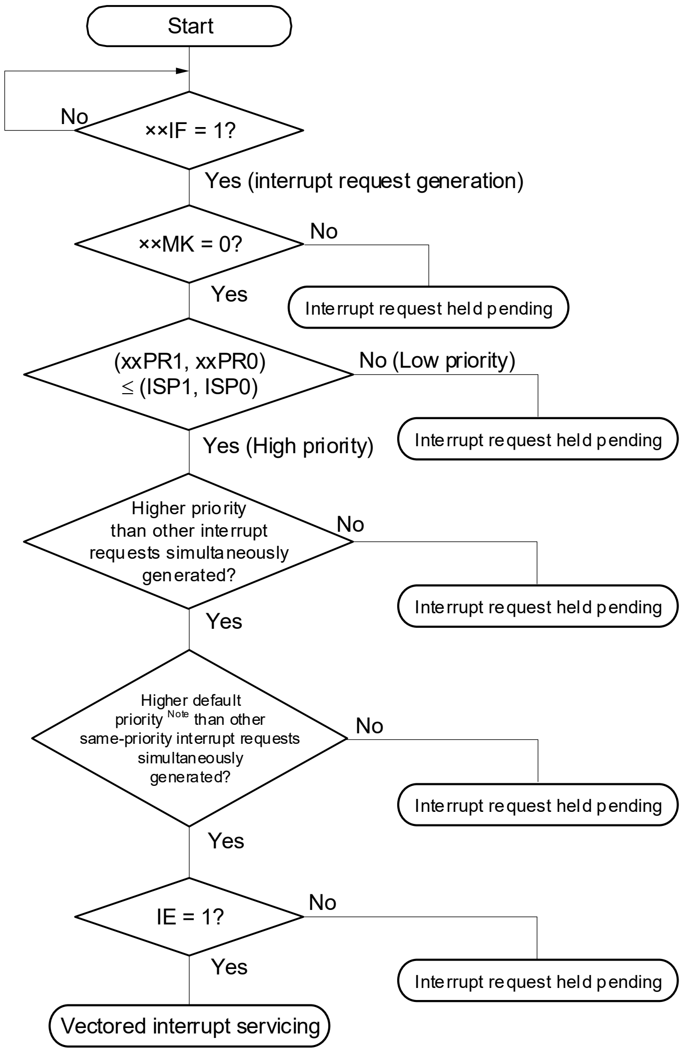
	Minimum Time	Maximum Time Note
Servicing time	9 clocks	16 clocks

Note Maximum time does not apply when an instruction from the internal RAM area is executed.

Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specification flag is acknowledged first. If two or more interrupts requests have the same priority level, the request with the highest default priority is acknowledged first.
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.
Figure 21 - 9 shows the Interrupt Request Acknowledgment Processing 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 (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 the loaded into the PC and branched.
Restoring from an interrupt is possible by using the RETI instruction.

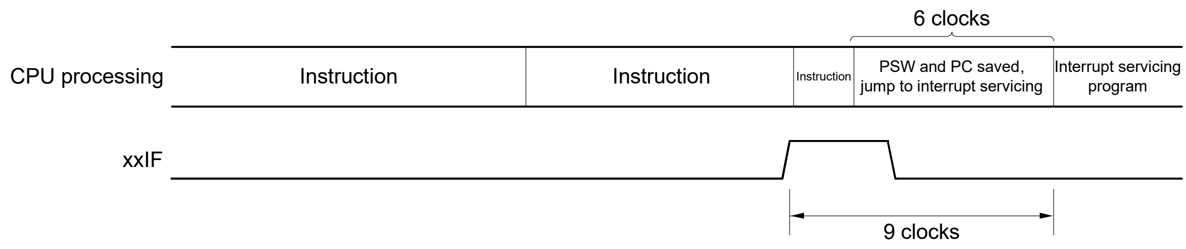
Figure 21 - 9 Interrupt Request Acknowledgment Processing Algorithm



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)
ISP0, ISP1:	Flag that indicates the priority level of the interrupt currently being serviced (see **Figure 21 - 8**)

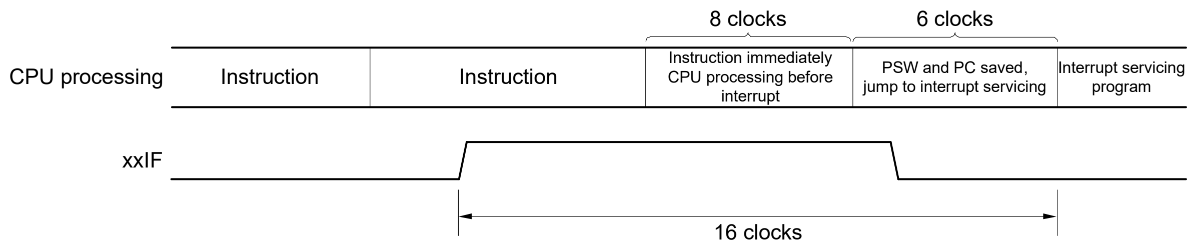
Note For the default priority, refer to **Table 21 - 1 Interrupt Source List**.

Figure 21 - 10 Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/fCLK (fCLK: CPU clock)

Figure 21 - 11 Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/fCLK (fCLK: CPU clock)

21.4.2 Software interrupt request acknowledgment

A software interrupt request is acknowledged by BRK instruction execution. Software interrupts cannot be disabled.
If a software interrupt request is acknowledged, the contents are saved into the stacks in the order of the program status word (PSW), then program counter (PC), the IE flag is reset (0), and the contents of the vector table (0007EH, 0007FH) are loaded into the PC and branched.
Restoring from a software interrupt is possible by using the RETB instruction.

Caution Can not use the RETI instruction for restoring from the software interrupt.

21.4.3 Multiple interrupt servicing

Multiple interrupt servicing occurs when another interrupt request is acknowledged during execution of an interrupt.
Multiple interrupt servicing does not occur unless the interrupt request acknowledgment enabled state is selected (IE = 1). When an interrupt request is acknowledged, interrupt request acknowledgment becomes disabled (IE = 0). Therefore, to enable multiple interrupt servicing, it is necessary to set (1) the IE flag with the EI instruction during interrupt servicing to enable interrupt acknowledgment.
Moreover, even if interrupts are enabled, multiple interrupt servicing may not be enabled, this being subject to interrupt priority control. Two types of priority control are available: default priority control and programmable priority control. Programmable priority control is used for multiple interrupt servicing.
In the interrupt enabled state, if an interrupt request with a priority equal to or higher than that of the interrupt currently being serviced is generated, it is acknowledged for multiple interrupt servicing. If an interrupt with a priority equal to or lower than that of the interrupt currently being serviced is generated during interrupt servicing, it is not acknowledged for multiple interrupt servicing. However, when setting the IE flag to 1 during the interruption at level 0, other level 0 interruptions can be allowed.
Interrupt requests that are not enabled because interrupts are in the interrupt disabled state or because they have a lower priority are held pending. When servicing of the current interrupt ends, the pending interrupt request is acknowledged following execution of at least one main processing instruction execution.
Table 21 - 5 shows Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing and Figures 21 - 12 and 21 - 13 show multiple interrupt servicing examples.

Table 21 - 5 Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing

Multiple Interrupt Request / Interrupt Being Serviced		Maskable Interrupt Request								Software Interrupt Request
		Priority Level 0 (PR = 00)		Priority Level 1 (PR = 01)		Priority Level 2 (PR = 10)		Priority Level 3 (PR = 11)		
		IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	
Maskable interrupt	ISP1 = 0 ISP0 = 0	√	×	×	×	×	×	×	×	√
	ISP1 = 0 ISP0 = 1	√	×	√	×	×	×	×	×	√
	ISP1 = 1 ISP0 = 0	√	×	√	×	√	×	×	×	√
	ISP1 = 1 ISP0 = 1	√	×	√	×	√	×	√	×	√
Software interrupt		√	×	√	×	√	×	√	×	√

Remark 1. √: Multiple interrupt servicing enabled

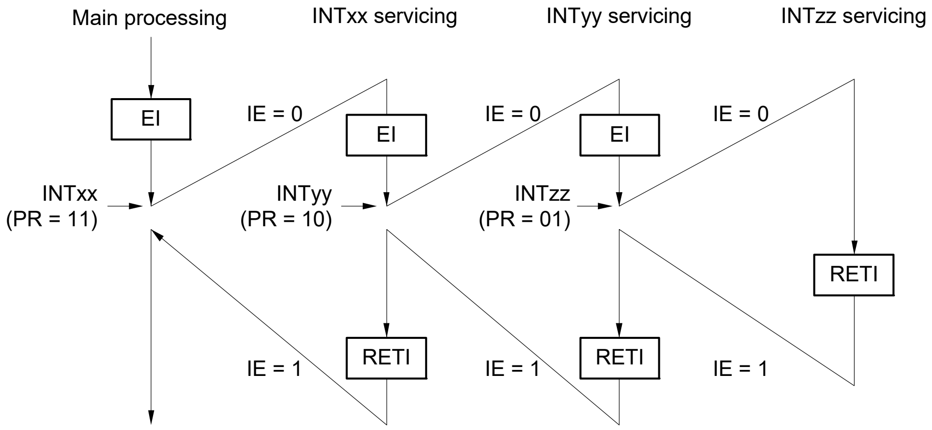
Remark 2. ×: Multiple interrupt servicing disabled

Remark 3. ISP0, ISP1, and IE are flags contained in the PSW.
ISP1 = 0, ISP0 = 0: An interrupt of level 1 or level 0 is being serviced.
ISP1 = 0, ISP0 = 1: An interrupt of level 2 is being serviced.
ISP1 = 1, ISP0 = 0: An interrupt of level 3 is being serviced.
ISP1 = 1, ISP0 = 1: Wait for An interrupt acknowledgment (all interrupts 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, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H registers.
PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)

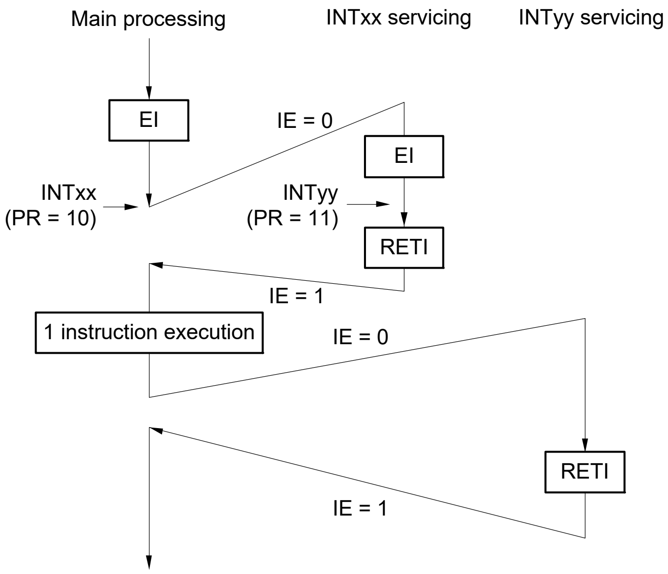
Figure 21 - 12 Examples of Multiple Interrupt Servicing (1/2)

Example 1. Multiple interrupt servicing occurs twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and multiple interrupt servicing takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledgment.

Example 2. Multiple interrupt servicing does not occur due to priority control

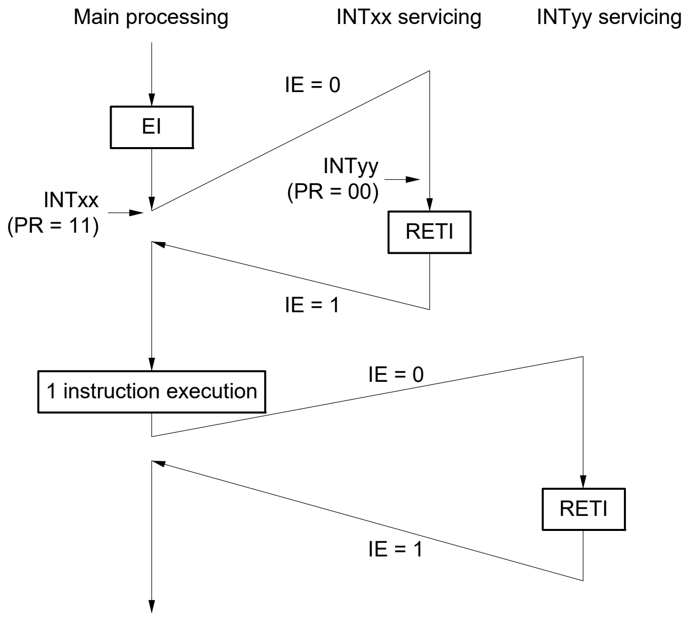


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00:	Specify level 0 with 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 - 13 Examples of Multiple Interrupt Servicing (2/2)

Example 3. Multiple interrupt servicing does not occur because interrupts are not enabled



Interrupts are not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with 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.

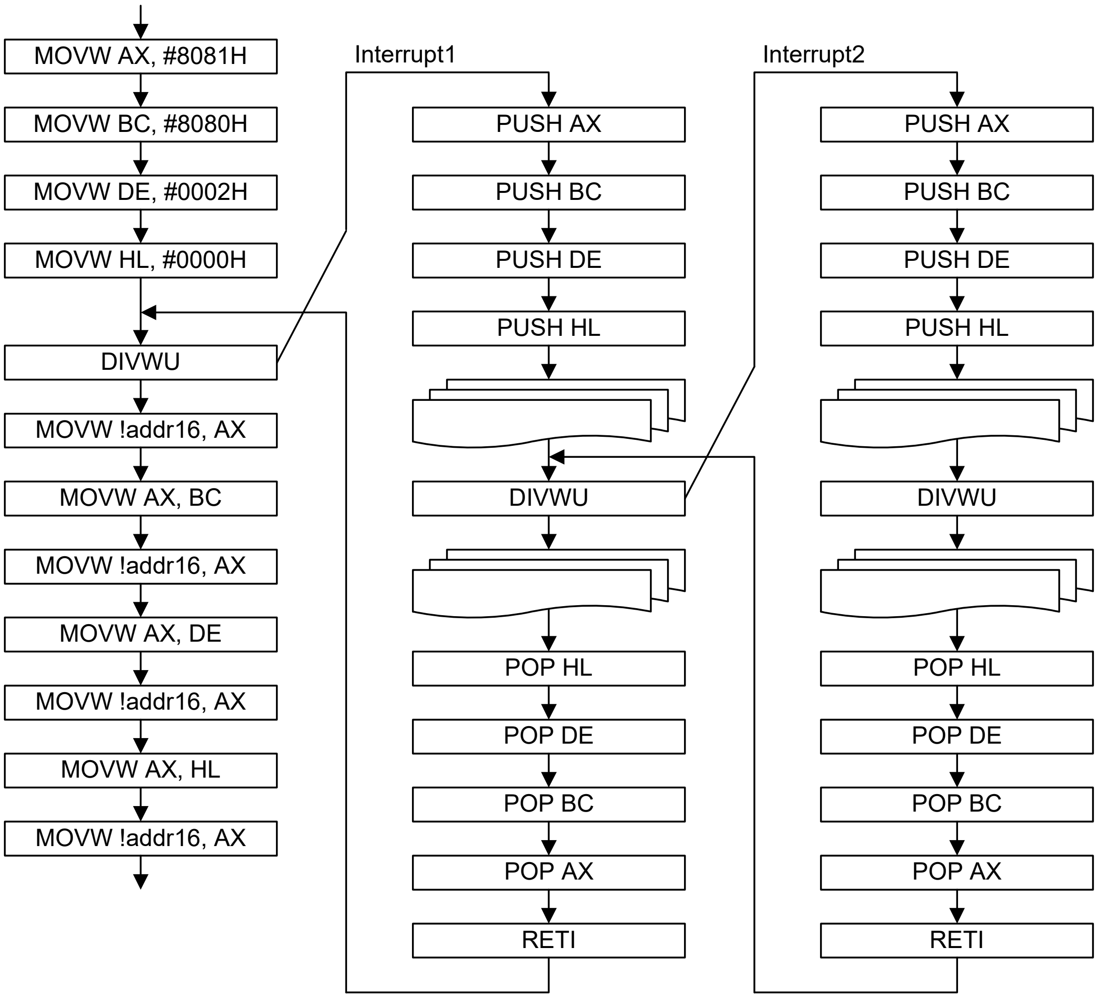
21.4.4 Interrupt servicing during division instruction

The RL78/I1D handles interrupts during the DIVHU/DIVWU instruction in order to enhance the interrupt response when a division instruction is executed.

• When an interrupt is generated while the DIVHU/DIVWU instruction is executed, the instruction is suspended
• After the instruction is suspended, the PC indicates the next instruction after DIVHU/DIVWU
• An interrupt is generated by the next instruction
• PC-3 is stacked to execute the DIVHU/DIVWU instruction again

Normal interrupt	Interrupts while Executing DIVHU/DIVWU Instruction
(SP-1) ← PSW	(SP-1) ← PSW
(SP-2) ← $(PC)_S$	(SP-2) ← $(PC-3)_S$
(SP-3) ← $(PC)_H$	(SP-3) ← $(PC-3)_H$
(SP-4) ← $(PC)_L$	(SP-4) ← $(PC-3)_L$
PC_S ← 0000	PC_S ← 0000
PC_H ← (Vector)	PC_H ← (Vector)
PC_L ← (Vector)	PC_L ← (Vector)
SP ← SP-4	SP ← SP-4
IE ← 0	IE ← 0

The AX, BC, DE, and HL registers are used for DIVHU/DIVWU. Use these registers by stacking them for interrupt servicing.



Caution Disable interrupts when executing the DIVHU or DIVWU instruction in an interrupt servicing routine.

Alternatively, unless they are executed in the RAM area, note that execution of a DIVHU or DIVWU instruction is possible even with interrupts enabled as long as a NOP instruction is added immediately after the DIVHU or DIVWU instruction in the assembly language source code. The following compilers automatically add a NOP instruction immediately after any DIVHU or DIVWU instruction output during the build process.

- **V. 1.71 and later versions of the CA78K0R (Renesas Electronics compiler), for both C and assembly language source code**
- **Service pack 1.40.3 and later versions of the EWRL78 (IAR compiler), for C language source code**
- **GNURL78 (KPIT compiler), for C language source code**

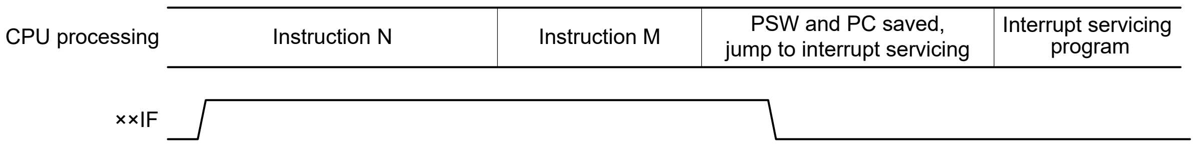
21.4.5 Interrupt request hold

There are instructions where, even if an interrupt request is issued while the instructions are being executed, interrupt request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, $addr20
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- SKH
- SKNH
- MULHU
- MULH
- MACHU
- MACH
- Write instructions for the IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H registers

Figure 21 - 14 shows the timing at which interrupt requests are held pending.

Figure 21 - 14 Interrupt Request Hold



Remark 1. Instruction N: Interrupt request hold instruction
Remark 2. Instruction M: Instruction other than interrupt request hold instruction

CHAPTER 22 KEY INTERRUPT FUNCTION

The number of key interrupt input channels differs, depending on the product.

	20, 30-pin	24, 32-pin	48-pin
Key interrupt input channels	—	3 ch	4 ch

22.1 Functions of Key Interrupt

A key interrupt (INTKR) can be generated by inputting a rising edge/falling edge to the key interrupt input pins (KR0 to KR3).

Table 22 - 1 Assignment of Key Interrupt Detection Pins

Key interrupt pins	Key return mode register 0 (KRM0)
KR0	KRM00
KR1	KRM01
KR2	KRM02
KR3	KRM03

Remark KR0 to KR2: 24, 32-pin
KR0 to KR3: 48-pin

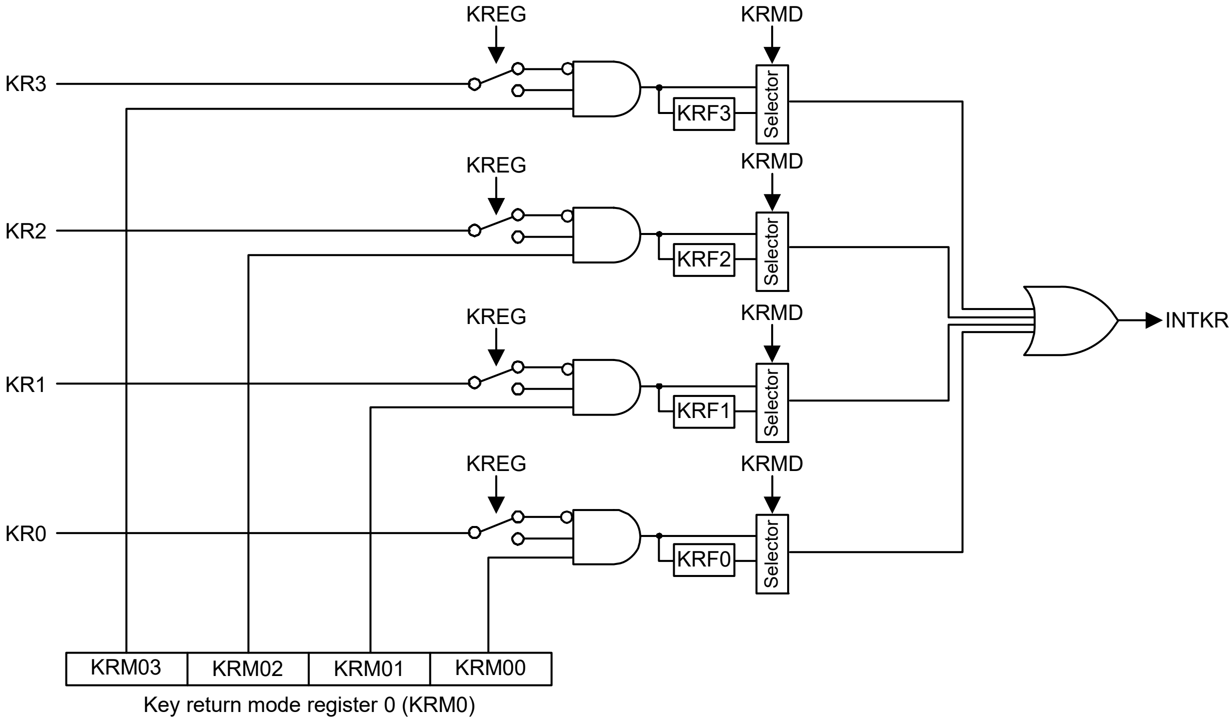
22.2 Configuration of Key Interrupt

The key interrupt includes the following hardware.

Table 22 - 2 Configuration of 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 3, 5 (PM3, PM5)

Figure 22 - 1 Block Diagram of Key Interrupt



Remark KR0 to KR2: 24, 32-pin
KR0 to KR3: 48-pin

22.3 Register Controlling Key Interrupt

The key interrupt function is controlled by the following registers.

• Key return control register (KRCTL)
• Key return mode register 0 (KRM0)
• Key return flag register (KRF)
• Port mode registers 3, 5 (PM3, PM5)

22.3.1 Key return control register (KRCTL)

This register controls the usage of the key return flags (KRF0 to KRF3) and sets the detection edge.
The KRCTL register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 22 - 2 Format of Key return control register (KRCTL)

Address: FFF34H After reset: 00H 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 KRF3)
0	Does not use key return flags
1	Uses key return flags

KREG	Selection of detection edge (KR0 to KR3)
0	Falling edge
1	Rising edge

22.3.2 Key return mode register 0 (KRM0)

The KRM0 register controls the KR0 to KR3 signals.
The KRM0 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 22 - 3 Format of Key return mode register 0 (KRM0)

Address: FFF37H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
KRM0	0	0	0	0	KRM03	KRM02	KRM01	KRM00

KRM0n	Key interrupt mode control
0	Does not detect key interrupt signal
1	Detects key interrupt signal

Caution 1. The on-chip pull-up resistors can be applied by setting the corresponding key interrupt input pins (bits) in pull-up resistor register 3 and 5 (PU3, PU5) to 1.

Caution 2. An interrupt will be generated if the target bit of the KRM0 register is set while a low level (KREG is set to 0) or a high level (KREG 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 34.4 AC Characteristics).

Caution 3. The pins not used in the key interrupt mode can be used as normal ports.

Remark n = 0 to 3

22.3.3 Key return flag register (KRF)

This register controls the key interrupt flags (KRF0 to KRF3).

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

Reset signal generation clears this register to 00H.

Figure 22 - 4 Format of Key return flag register (KRF)

Address: FFF35H After reset: 00H R/W Note

Symbol	7	6	5	4	3	2	1	0
KRF	0	0	0	0	KRF3	KRF2	KRF1	KRF0

KRFn	Key interrupt flag (n = 0 to 3)
0	No key interrupt signal has been detected.
1	A key interrupt signal has been detected.

Note Writing to 1 is invalid. To clear the KRFn bit, write 0 to the corresponding bit and 1 to the other bits using an 8-bit memory manipulation instruction.

22.3.4 Port mode registers 3, 5 (PM3, PM5)

These registers set the input and output of ports 3 and 5 in 1-bit units.
To use a key interrupt input (KR0 to KR3), set 1 to the bit of port mode register (PM3, PM5) corresponding to each port.
The PM3 and PM5 registers can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears these registers to FFH.

Figure 22 - 5 Format of Port mode registers 3, 5 (PM3, PM5)

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	1	PM33	PM32	PM31	PM30

Address: FFF25H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50

KRM0n	I/O mode selection for PMmn pin (m = 3, 5, n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

CHAPTER 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
HALT instruction execution sets the HALT mode. In the HALT mode, the CPU operation clock 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
STOP instruction execution sets 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 CPU operating current.
Because this mode can be cleared by an interrupt request, it enables intermittent operations to be carried out. However, because a wait time is required to secure the oscillation stabilization time after the STOP mode is released when the X1 clock is selected, select the HALT mode if it is necessary to start processing immediately upon interrupt request generation.

(3) SNOOZE mode
In the case of CSI0 or UART0 data reception, an A/D conversion request by the timer trigger signal (the interrupt request signal (INTRTC/INTIT) or ELC event input), and DTC start source, the STOP mode is exited, the CSI0 or UART0 data is received without operating the CPU, A/D conversion is performed, and DTC start source. 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 either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latches and output buffer statuses are also held.

Caution 1. The STOP mode can be used only when the CPU is operating on the main system clock. Do not set to the STOP mode while the CPU operates with the subsystem clock. The HALT mode can be used when the CPU is operating on either the main system clock or the subsystem clock.

Caution 2. When shifting to the STOP mode, be sure to stop the peripheral hardware operation operating with main system clock before executing STOP instruction (except SNOOZE mode setting unit).

Caution 3. When using CSI0, UART0, or the A/D converter in the SNOOZE mode, set up serial standby control register m (SSCm) and A/D converter mode register 2 (ADM2) before switching to the STOP mode. For details, see 17.3 Registers Controlling Serial Array Unit and 14.3 Registers Controlling A/D Converter.

Caution 4. The following sequence is recommended for operating current reduction of the A/D converter when the standby function is used: First clear bit 7 (ADCS) and bit 0 (ADCE) of A/D converter mode register 0 (ADM0) to 0 to stop the A/D conversion operation, and then execute the STOP instruction.

Caution 5. It can be selected by the option byte whether the low-speed on-chip oscillator continues oscillating or stops in the HALT or STOP mode. For details, see CHAPTER 29 OPTION BYTE.

23.2 Registers controlling standby function

The registers which control the standby function are described below.

- Subsystem clock supply mode control register (OSMC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)

Remark For details of registers described above, see **CHAPTER 6 CLOCK GENERATOR**. For registers which control the SNOOZE mode, **CHAPTER 14 A/D CONVERTER** and **CHAPTER 17 SERIAL ARRAY UNIT**.

23.3 Standby Function Operation

23.3.1 HALT mode

(1) HALT mode

The HALT mode is set by executing the HALT instruction. HALT mode can be set regardless of whether the CPU clock before the setting was the high-speed system clock, high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock, or subsystem clock.

The operating statuses in the HALT mode are shown below.

Caution Because the interrupt request signal is used to clear the HALT mode, if the interrupt mask flag is 0 (the interrupt processing is enabled) and the interrupt request flag is 1 (the interrupt request signal is generated), the HALT mode is not entered even if the HALT instruction is executed in such a situation.

Table 23 - 1 Operating Statuses in HALT Mode (1/2)

Item \ HALT Mode Setting			When HALT Instruction is Executed While CPU is Operating on Main System Clock			
			When CPU is Operating on High-speed On-chip Oscillator Clock (f_{IH})	When CPU is Operating on Middle-speed On-chip Oscillator Clock (f_{IM})	When CPU is Operating on X1 Clock (f_X)	When CPU is Operating on External Main System Clock (f_{EX})
System clock			Clock supply to the CPU is stopped			
	Main system clock	f_{IH}	Operation continues (cannot be stopped)	Operation disabled	Operation disabled	
		f_{IM}	Operation disabled	Operation continues (cannot be stopped)	Operation disabled	
		f_X	Operation disabled		Operation continues (cannot be stopped)	Cannot operate
		f_{EX}			Cannot operate	Operation continues (cannot be stopped)
	Subsystem clock	f_{XT}	Status before HALT mode was set is retained			
		f_{EXS}				
	Low-speed on-chip oscillator clock	f_{IL}	Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of 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 clock generator clock (f_{SX}, f_{SXR}) is operating.) WUTMMCK0 = 0, SELLOSC = 0, and WDTON = 0: Stop WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 1: Oscillates WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 0: Stop			
CPU			Operation stopped			
Code flash memory						
Data flash memory						
RAM			Operation stopped (Operable while in the DTC is executed)			
Port (latch)			Status before HALT mode was set is retained			
Timer array unit			Operable			
RTC2			Operable			
Frequency measurement function			Operable			
12-bit Interval timer						
8-bit Interval timer						
Watchdog timer			See **CHAPTER 13 WATCHDOG TIMER**.			
Clock output/buzzer output			Operable			
A/D converter						
Comparator						
Operational amplifier function						
Serial array unit (SAU)						
Data operation circuit (DOC)			Operable when registers are set by the DTC			
Data transfer controller (DTC)			Operable			
Event link controller (ELC)			Operable function blocks can be linked			
Power-on-reset function			Operable			
Voltage detection function						
External interrupt						
Key interrupt function						
CRC operation function	High-speed CRC					
	General-purpose CRC		In the calculation of the RAM area, operable when DTC is executed only			
Illegal-memory access detection function			Operable when DTC is executed only			
RAM parity error detection function						
RAM guard function						
SFR guard function						

Remark Operation stopped: Operation is automatically stopped before switching to HALT mode.
Operation disabled: Operation is stopped before switching to the HALT mode.
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 Statuses in HALT Mode (2/2)

Item (HALT Mode Setting)			When HALT Instruction is Executed While CPU is Operating on Subsystem Clock		
			When CPU is Operating on XT1 Clock (f_{XT})	When CPU is Operating on External Subsystem Clock (f_{EXS})	When CPU is Operating on 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 (cannot be stopped)	Cannot operate	Operation disabled
		f_{EXS}	Cannot operate	Operation continues (cannot be stopped)	Operation disabled
	Low-speed on-chip oscillator clock	f_{IL}	Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply mode control register (OSMC) WUTMMCK0 = 1: Oscillates (Setting of WUTMMCK0 = 1 and SELLOSC = 1 is prohibited when the subsystem clock generator clock (f_{SX}, f_{SXR}) is operating.) WUTMMCK0 = 0, SELLOSC = 0, and WDTON = 0: Stop WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 1: Oscillates WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 0: Stop		Operation continues (cannot be stopped)
CPU			Operation stopped		
Code flash memory					
Data flash memory					
RAM			Operation stopped (Operable while in the DTC is executed)		
Port (latch)			Status before HALT mode was set is retained		
Timer array unit			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operable
RTC2			Operable		
Frequency measurement function			Operation stopped		
12-bit Interval timer			Operable		
8-bit Interval timer					
Watchdog timer			See **CHAPTER 13 WATCHDOG TIMER**.		
Clock output/buzzer output			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		
A/D converter			Operation disabled		
Comparator			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operable
Operational amplifier function			Operable		
Serial array unit (SAU)			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operable
Data operation circuit (DOC)			Operable when registers are set by the DTC		
Data transfer controller (DTC)			Operates when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0).		Operable
Event link controller (ELC)			Operable function blocks can be linked		
Power-on-reset function			Operable		
Voltage detection function					
External interrupt					
Key interrupt function					
CRC operation function		High-speed CRC	Operation disabled		
		General-purpose CRC	In the calculation of the RAM area, operable when DTC is executed only		
Illegal-memory access detection function			Operable when DTC is executed only		
RAM parity error detection function					
RAM guard function					
SFR guard function					

Remark Operation stopped: Operation is automatically stopped before switching to the HALT mode.
Operation disabled: Operation is stopped before switching to the HALT mode.
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

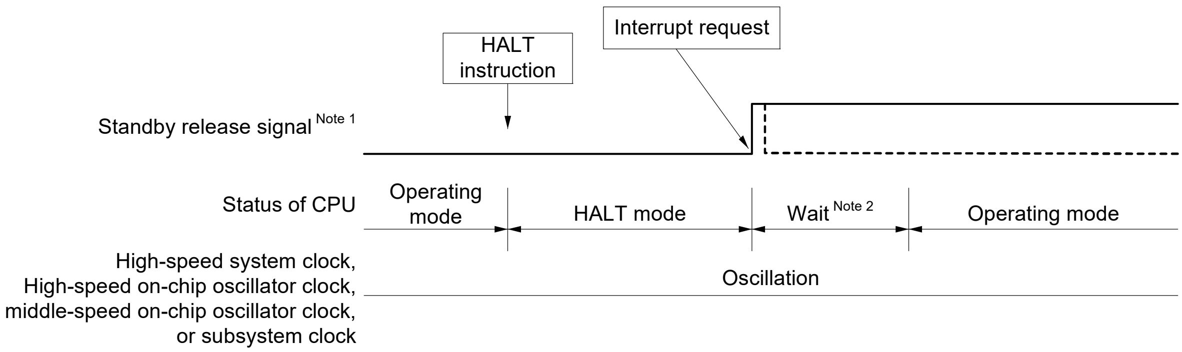
(2) HALT mode release

The HALT mode can be released by the following two sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the HALT mode is released. If interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 23 - 1 HALT Mode Release by Interrupt Request Generation



Note 1. For details of the standby release signal, see **Figure 21 - 1 Basic Configuration of Interrupt Function (1/2)**.

Note 2. Wait time for HALT mode release

- When vectored interrupt servicing is carried out
  - Main system clock: 15 to 16 clocks
  - Subsystem clock (RTCLPC = 0): 10 to 11 clocks
  - Subsystem clock (RTCLPC = 1): 11 to 12 clocks
- When vectored interrupt servicing is not carried out
  - Main system clock: 9 to 10 clocks
  - Subsystem clock (RTCLPC = 0): 4 to 5 clocks
  - Subsystem clock (RTCLPC = 1): 5 to 6 clocks

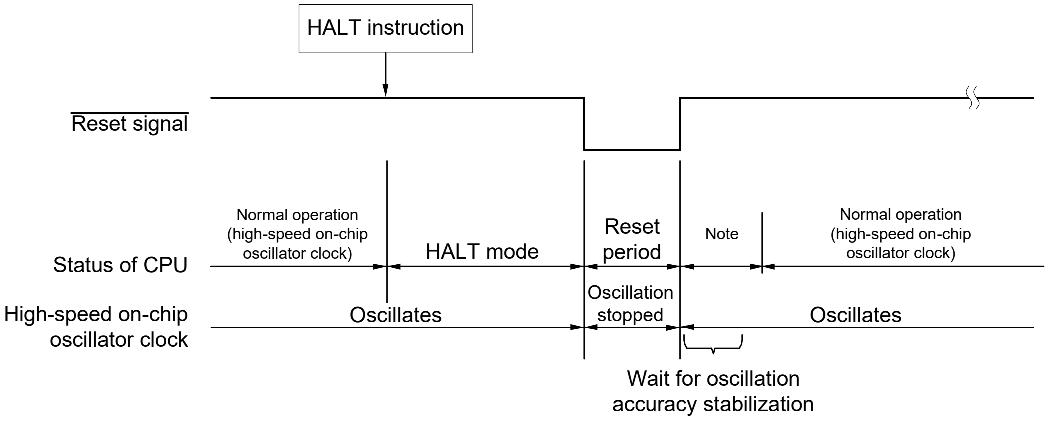
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) Release by reset signal generation

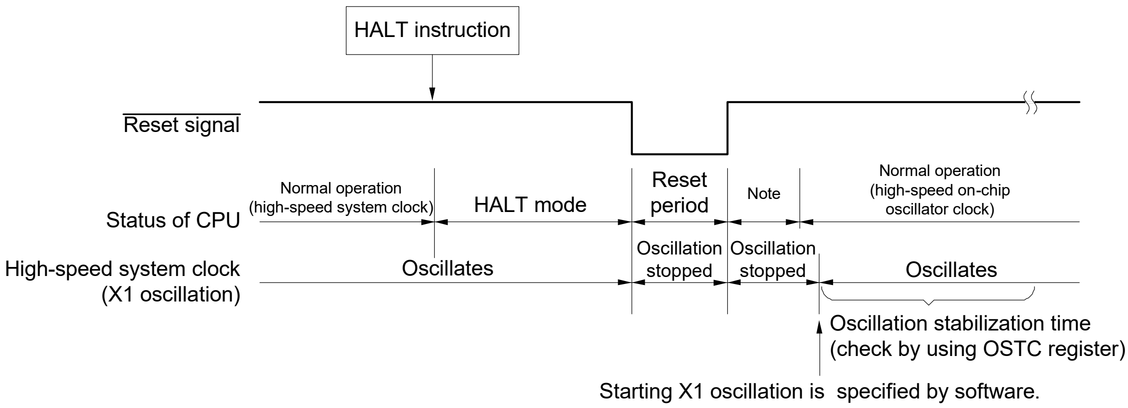
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 23 - 2 HALT Mode Release by Reset (1/2)

(1) When high-speed on-chip oscillator clock is used as CPU clock



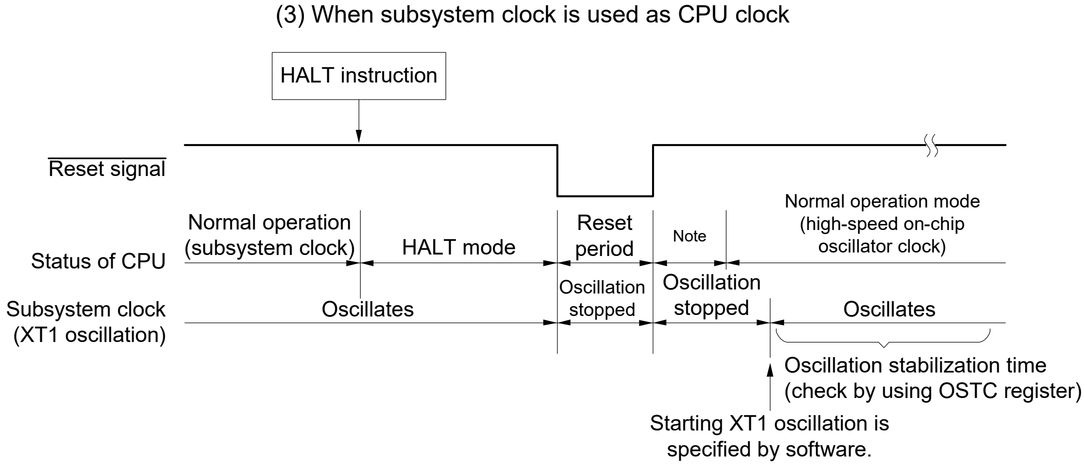
(2) When high-speed system clock is used as CPU clock



Note For the reset processing time, see **CHAPTER 24 RESET FUNCTION**.

For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 25 POWER-ON-RESET CIRCUIT**.

Figure 23 - 3 HALT Mode Release by Reset (2/2)



Note For the reset processing time, see **CHAPTER 24 RESET FUNCTION**.
For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 25 POWER-ON-RESET CIRCUIT**.

23.3.2 STOP mode

(1) STOP mode setting and operating statuses

The STOP mode is set by executing the STOP instruction, and it can be set only when the CPU clock before the setting was the high-speed on-chip oscillator clock, X1 clock, or external main system clock.

Caution Because the interrupt request signal is used to clear the STOP mode, if the interrupt mask flag is 0 (the interrupt processing is enabled) and the interrupt request flag is 1 (the interrupt request signal is generated), the STOP mode is immediately cleared if set when the STOP instruction is executed in such a situation.
Accordingly, once the STOP instruction is executed, the system returns to its normal operating mode after the elapse of release time from the STOP mode.

The operating statuses in the STOP mode are shown below.

Table 23 - 3 Operating Statuses in STOP Mode

Item \ STOP Mode Setting			When STOP Instruction is Executed While CPU is Operating on Main System Clock			
			When CPU is Operating on High-speed On-chip Oscillator Clock (f_{IH})	When CPU is Operating on Middle-speed On-chip Oscillator Clock (f_{IM})	When CPU is Operating on X1 Clock (f_X)	When CPU is Operating on External Main System Clock (f_{EX})
System clock			Clock supply to the CPU is stopped			
	Main system clock	f_{IH}	Stopped			
		f_{IM}	Stopped	Stopped	Stopped	
		f_X	Stopped			
		f_{EX}				
	Subsystem clock	f_{XT}	Status before STOP mode was set is retained			
		f_{EXS}				
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of 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 clock generator clock (f_{SX}, f_{SXR}) is operating.) WUTMMCK0 = 0, SELLOSC = 0, and WDTON = 0: Stop WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 1: Oscillates WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 0: Stop			
CPU			Operation stopped			
Code flash memory						
Data flash memory						
RAM						
Port (latch)			Status before STOP mode was set is retained			
Timer array unit			Operation disabled			
RTC2			Operable			
Frequency measurement function			Operation disabled			
12-bit Interval timer			Operable			
8-bit Interval timer						
Watchdog timer			See **CHAPTER 13 WATCHDOG TIMER**.			
Clock output/buzzer output			Operates when the subsystem clock is selected as the clock source for counting and the RTCLPC bit is 0 (operation is disabled when a clock other than the subsystem clock is selected and the RTCLPC bit is not 0).			
A/D converter			Wakeup operation is enabled (switching to SNOOZE mode)			
Comparator			Operable (only when the digital filter is not used)			
Operational amplifier function			Operable			
Serial array unit (SAU)			Wakeup operation is enabled only for CSIp and UARTq (switching to SNOOZE mode) Operation is disabled for anything other than CSIp and UARTq			
Data operation circuit (DOC)			Operation stopped			
Data transfer controller (DTC)			DTC activation source receiving operation enabled (switching to SNOOZE mode)			
Event link controller (ELC)			Operable function blocks can be linked			
Power-on-reset function			Operable			
Voltage detection function						
External interrupt						
Key interrupt function						
CRC operation function	High-speed CRC		Operation stopped			
	General-purpose CRC					
Illegal-memory access detection function						
RAM parity error detection function						
RAM guard function						
SFR guard function						

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

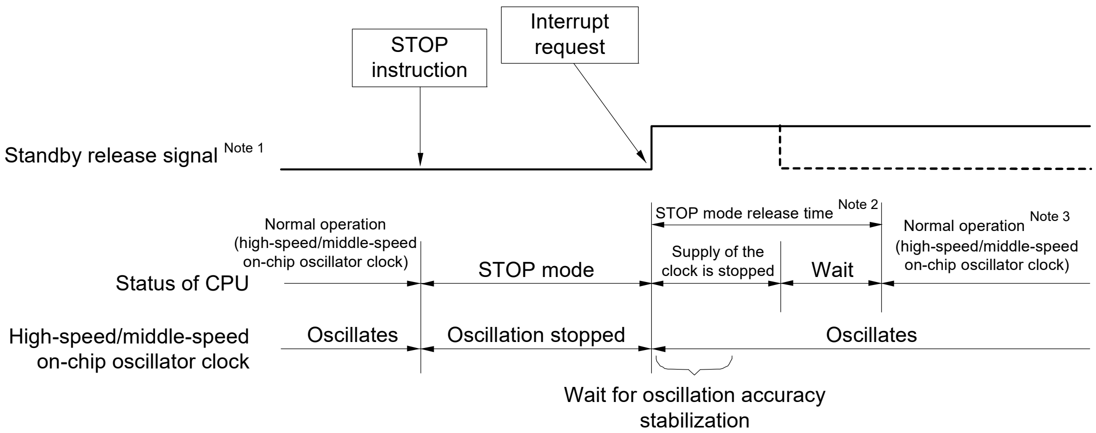
Remark Operation stopped: Operation is automatically stopped before switching to the STOP mode.
Operation disabled: Operation is stopped before switching to the STOP mode.
f_{IH}: High-speed on-chip oscillator clock f_{IL}: Low-speed on-chip oscillator clock
f_{IM}: Middle-speed on-chip oscillator clock f_X: X1 clock
f_{EX}: External main system clock f_{XT}: XT1 clock
f_{EXS}: External subsystem clock

(2) STOP mode release
The STOP mode can be released by the following two sources.

(a) Release by unmasked interrupt request
When an unmasked interrupt request is generated, the STOP mode is released. After the oscillation stabilization time has elapsed, if interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 23 - 4 STOP Mode Release by Interrupt Request Generation (1/2)

(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 (1/2)**.

Note 2. STOP mode release time
Supply of the clock is stopped:
When high-speed on-chip oscillator clock: 18 μs to 65 μs
When middle-speed on-chip oscillator clock: 22 μs to 31 μs (in HS mode)
2.2 μs to 3.4 μs (during operation at 4 MHz in LS mode)
2.9 μs to 4.2 μs (during operation at 2 MHz in LS mode)
4.2 μs to 5.9 μs (during operation at 1 MHz in LS mode)
4.2 μs to 5.9 μs (during operation at 1 MHz in LP mode)
Wait:
(common to the high-speed/middle-speed on-chip oscillator clock)
•When vectored interrupt servicing is carried out: 7 clocks
•When vectored interrupt servicing is not carried out: 1 clock

Note 3. Before switching the operating clock from the CPU/peripheral hardware clock (f_{CLK}) to the high-speed on-chip oscillator clock after using the middle-speed on-chip oscillator clock for the transition from STOP mode to normal mode, use software to set up waiting for the corresponding period from the list below.
In HS mode: 24 μs
In LS mode: 10 μs
In LP mode: 7 μs

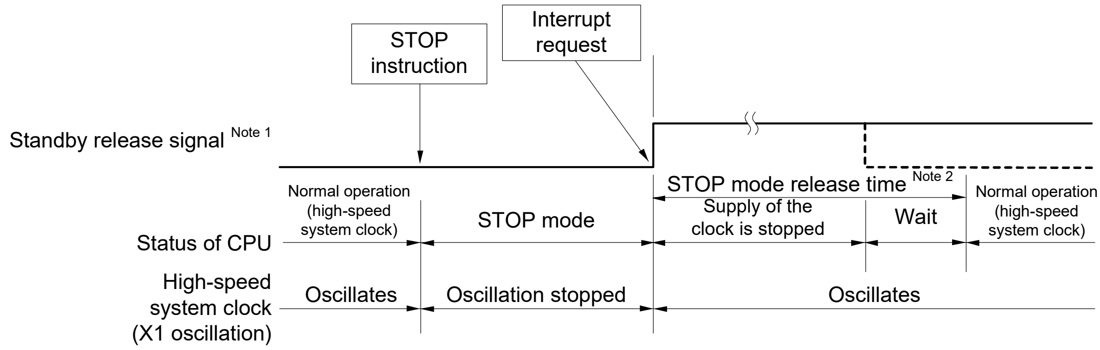
Caution To shorten oscillation stabilization time after the STOP mode is released when the CPU operates with the high-speed system clock (X1 oscillation), temporarily switch the CPU clock to the high-speed on-chip oscillator clock before the execution of the STOP instruction.

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 - 5 STOP Mode Release by Interrupt Request Generation (2/2)

(2) When high-speed system clock (X1 oscillation) is used as CPU clock

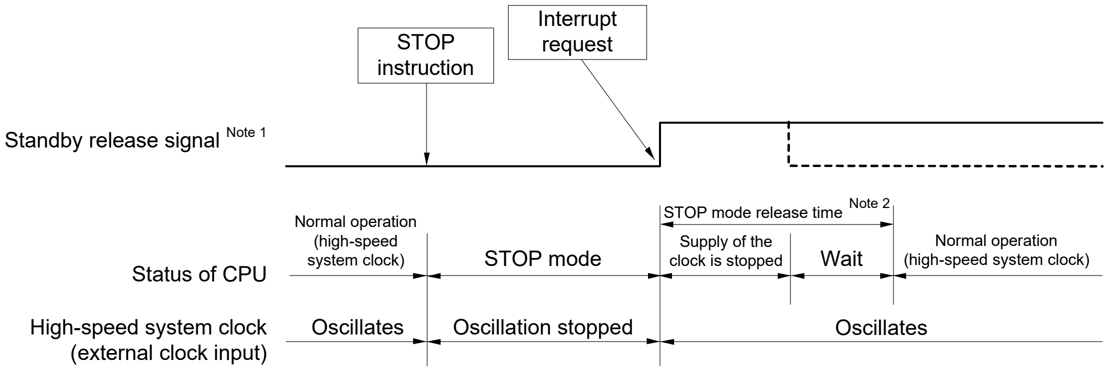


Note 1. For details of the standby release signal, see **Figure 21 - 1 Basic Configuration of Interrupt Function (1/2)**.

Note 2. STOP mode release time
Supply of the clock is stopped:
18 μs to whichever is longer 65 μs or the oscillation stabilization time (set by OSTS)

Wait:
- When vectored interrupt servicing is carried out: 10 to 11 clocks
- When vectored interrupt servicing is not carried out: 4 to 5 clocks

(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 (1/2)**.

Note 2. STOP mode release time
Supply of the clock is stopped:
18 μs to 65 μs

Wait:
- When vectored interrupt servicing is carried out: 7 clocks
- When vectored interrupt servicing is not carried out: 1 clock

Caution To reduce the oscillation stabilization time after release from the STOP mode while CPU operates based on the high-speed system clock (X1 oscillation), switch the clock to the high-speed on-chip oscillator clock temporarily before executing the STOP instruction.

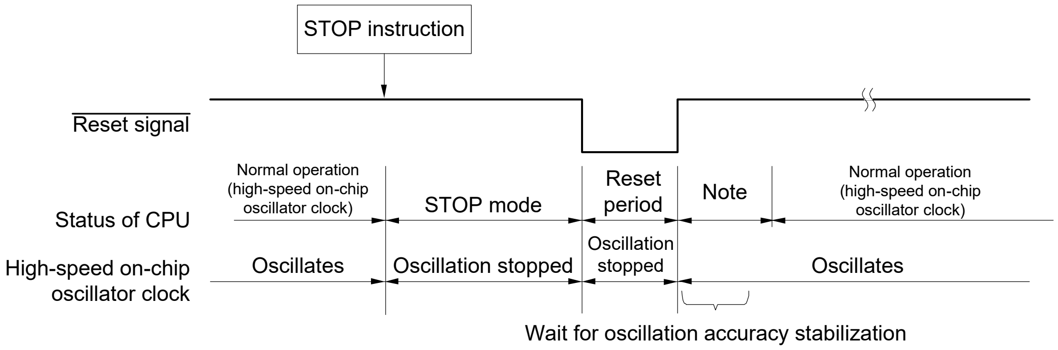
Remark 1. The clock supply stop time varies depending on the temperature conditions and STOP mode period.
Remark 2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(b) Release by reset signal generation

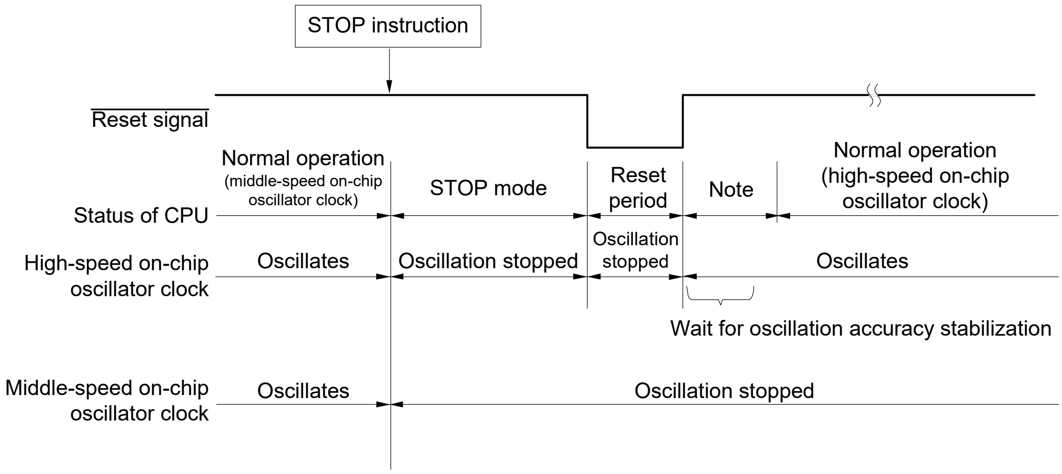
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 23 - 6 STOP Mode Release by Reset

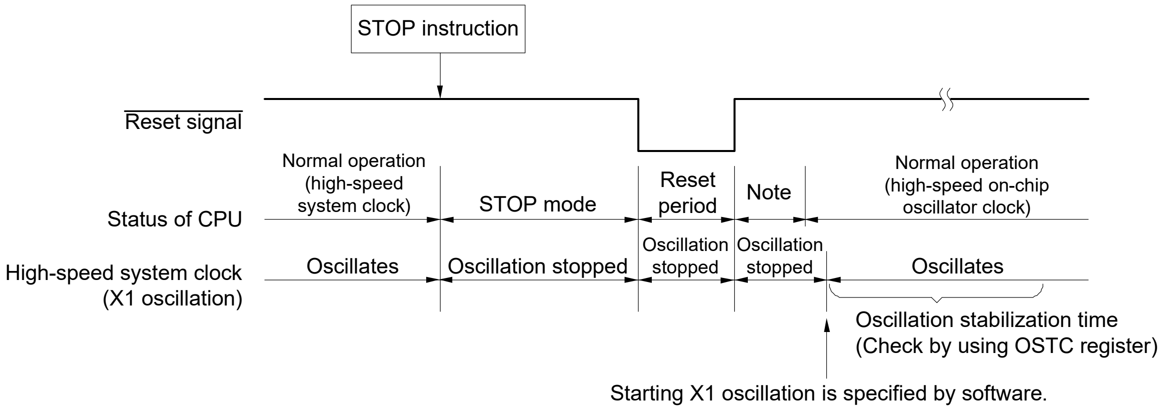
(1) When high-speed on-chip oscillator clock is used as CPU clock



(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 **CHAPTER 24 RESET FUNCTION**.
For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 25 POWER-ON-RESET CIRCUIT**.

23.3.3 SNOOZE mode

(1) SNOOZE mode setting and operating statuses

The SNOOZE mode can be set by the CSI0, UART0, A/D converter, and DTC. Note that this mode can only be specified if the CPU clock is the high-speed on-chip oscillator clock or middle-speed on-chip oscillator clock Note.

When using CSI0 or UART0 in the SNOOZE mode, set up serial standby control register m (SSCm) before switching to the STOP mode. For details, see **17.3 Registers Controlling Serial Array Unit**.

When using the A/D converter in the SNOOZE mode, set up A/D converter mode register 2 (ADM2) before switching to the STOP mode. For details, see **14.3 Registers Controlling A/D Converter**.

When DTC transfer is used in SNOOZE mode, before switching to the STOP mode, allow DTC activation by interrupt to be used. During STOP mode, detecting DTC activation by interrupt enables DTC transit to SNOOZE mode, automatically. For details, see **19.3 Registers Controlling DTC**.

Note When using UART reception to transition from STOP mode to SNOOZE mode, use the high-speed on-chip oscillator.

In SNOOZE mode transition, wait status to be only following time.

Transition time from STOP mode to SNOOZE mode:

When high-speed on-chip oscillator clock: 18 μs to 65 μs

When middle-speed on-chip oscillator clock: 22 μs to 31 μs (in HS mode)

Up to 3.4 μs (during operation at 4 MHz in LS mode)

Up to 4.2 μs (during operation at 2 MHz in LS mode)

Up to 5.9 μs (during operation at 1 MHz in LS mode)

Up to 5.9 μs (during operation at 1 MHz in LP mode)

Remark Transition time from STOP mode to SNOOZE mode varies depending on the temperature conditions and the STOP mode period.

Transition time from SNOOZE mode to normal operation:

When high-speed on-chip oscillator clock:

- When vectored interrupt servicing is carried out:
  - HS (High-speed main) mode: “5.2 μs to 9.4 μs” + 7 clocks
  - LS (Low-speed main) mode: “1.3 μs to 4.5 μs” + 7 clocks
  - LV (Low-voltage main) mode: “17.5 μs to 25.2 μs” + 7 clocks
- When vectored interrupt servicing is not carried out:
  - HS (High-speed main) mode: “5.2 μs to 9.4 μs” + 1 clock
  - LS (Low-speed main) mode: “1.3 μs to 4.5 μs” + 1 clock
  - LV (Low-voltage main) mode: “17.5 μs to 25.2 μs” + 1 clock

When middle-speed on-chip oscillator clock:

- When vectored interrupt servicing is carried out:
  - HS (High-speed main) mode: “6.0 μs to 10.3 μs” + 7 clocks
  - LS (Low-speed main) mode: “1.8 μs to 4.9 μs” + 7 clocks
  - LP (Low-power main) mode: “3.8 μs to 4.9 μs” + 7 clocks
- When vectored interrupt servicing is not carried out:
  - HS (High-speed main) mode: “6.0 μs to 10.3 μs” + 1 clock
  - LS (Low-speed main) mode: “1.8 μs to 4.9 μs” + 1 clock
  - LP (Low-power main) mode: “3.8 μs to 4.9 μs” + 1 clock

The operating statuses in the SNOOZE mode are shown next.

Table 23 - 4 Operating Statuses in SNOOZE Mode

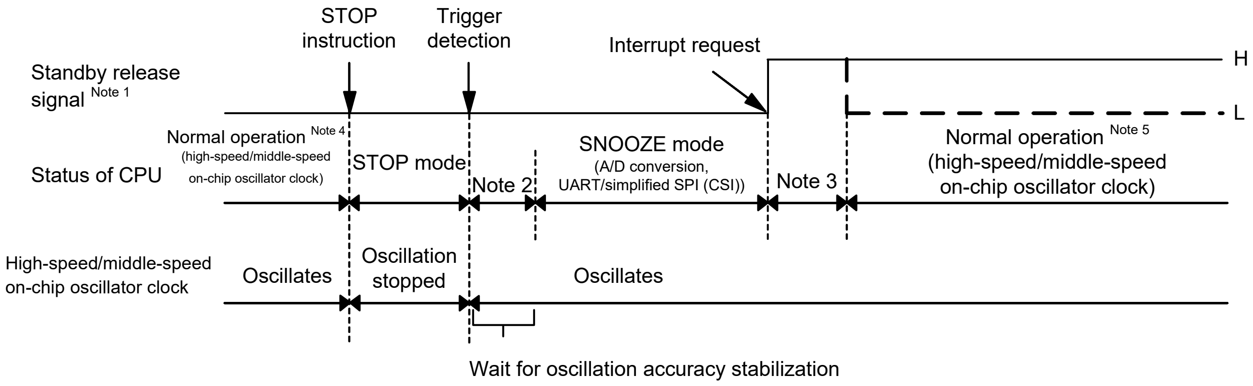
Item (STOP Mode Setting)			During STOP mode, receiving data signal from CSI0 and UART0, inputting timer trigger signal to A/D converter, and generating DTC activation by interrupt: When CPU is Operating on High-speed On-chip Oscillator Clock (f_{IH})	When CPU is Operating on Middle-speed On-chip Oscillator 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}	Use of the status while in the STOP mode continues	
		f_{EXS}		
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of 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 clock generator clock (f_{SX}, f_{SXR}) is operating.) WUTMMCK0 = 0, SELLOSC = 0, and WDTON = 0: Stop WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 1: Oscillates WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 0: Stop	
CPU			Operation stopped	
Code flash memory				
Data flash memory				
RAM			Operation stopped (Operable while in the DTC is executed)	
Port (latch)			Use of the status while in the STOP mode continues	
Timer array unit			Operation disabled	
RTC2			Operable	
Frequency measurement function			Operation disabled	
8-bit Interval timer			Operable	
12-bit Interval timer				
Watchdog timer			See **CHAPTER 13 WATCHDOG TIMER**.	
Clock output/buzzer output			Operates when the subsystem clock is selected as the clock source for counting and the RTCLPC bit is 0 (operation is disabled when a clock other than the f_{IL} or f_{SXR} is selected and the RTCLPC bit is not 0).	
A/D converter			Operable	
Comparator			Operable (when digital filter is not used)	
Operational amplifier function			Operable	
Serial array unit (SAU)			Operable CSI00 and UART0 only. Operation disabled other than CSI00 and UART0.	
Data operation circuit (DOC)			Operable when registers are set by the DTC	
Data transfer controller (DTC)			Operable	
Event link controller (ELC)			Operable function blocks can be linked	
Power-on-reset function			Operable	
Voltage detection function				
External interrupt				
Key interrupt function				
CRC operation function	High-speed CRC		Operation stopped	
	General-purpose CRC		Operation disabled	
Illegal-memory access detection function			Operable when executing the DTC	
RAM parity error detection function				
RAM guard function				
SFR guard function				

Remark Operation stopped: Operation is automatically stopped before switching to the STOP mode.
Operation disabled: Operation is stopped before switching to the STOP mode.

f_{IH}: High-speed on-chip oscillator clock
f_{IL}: Low-speed on-chip oscillator clock
f_{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 generator clock
f_{SXR}: Subsystem clock generator and RTC2/other clock

(2) Timing diagram when the interrupt request signal is generated in the SNOOZE mode

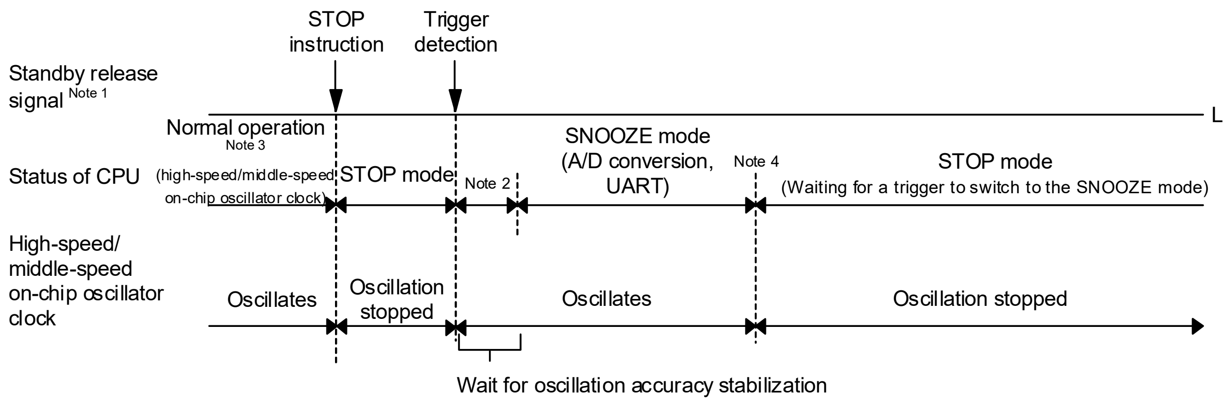
Figure 23 - 7 When the Interrupt Request Signal is Generated in the SNOOZE Mode



Note 1. For details of the standby release signal, see **Figure 21 - 1**.
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 (AWC = 1 or SWC = 1) immediately before switching to the STOP mode.
Note 5. Be sure to release the SNOOZE mode (AWC = 0 or SWC = 0) immediately after return to the normal operation.

(3) Timing diagram when the interrupt request signal is not generated in the SNOOZE mode

Figure 23 - 8 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**.
Note 2. Transition time from STOP mode to SNOOZE mode
Note 3. Enable the SNOOZE mode (AWC = 1 or SWC = 1) immediately before switching to the STOP mode.
Note 4. If a standby release signal is generated in response to an interrupt from a module which is not set to operate in the SNOOZE mode during a transition of the chip from SNOOZE mode to STOP mode, the high-speed on-chip oscillator clock may run slowly for up to 15 μs from when the CPU starts to operate. If the clock frequency accuracy specified in the electrical characteristics is required immediately after release from standby, wait for the number of cycles at the actual CPU clock frequency that is equivalent to 15 μs.

Remark For details of the SNOOZE mode function, see **CHAPTER 14 A/D CONVERTER** and **CHAPTER 17 SERIAL ARRAY UNIT**.

CHAPTER 24 RESET FUNCTION

The following seven operations are available to generate a reset signal.

(1) External reset input via $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer program loop detection
(3) Internal reset by comparison of supply voltage and detection voltage of power-on-reset (POR) circuit
(4) Internal reset by comparison of supply voltage of the voltage detector (LVD) and detection voltage
(5) Internal reset by execution of illegal instruction [Note]
(6) Internal reset by RAM parity error
(7) Internal reset by illegal-memory access

External and internal resets start program execution from the address at 0000H and 0001H when the reset signal is generated.
A reset is effected when a low level is input to the $\overline{\text{RESET}}$ pin, the watchdog timer overflows, or by POR and LVD circuit voltage detection, execution of illegal instruction [Note], RAM parity error or illegal-memory access, and each item of hardware is set to the status shown in Table 24 - 1.

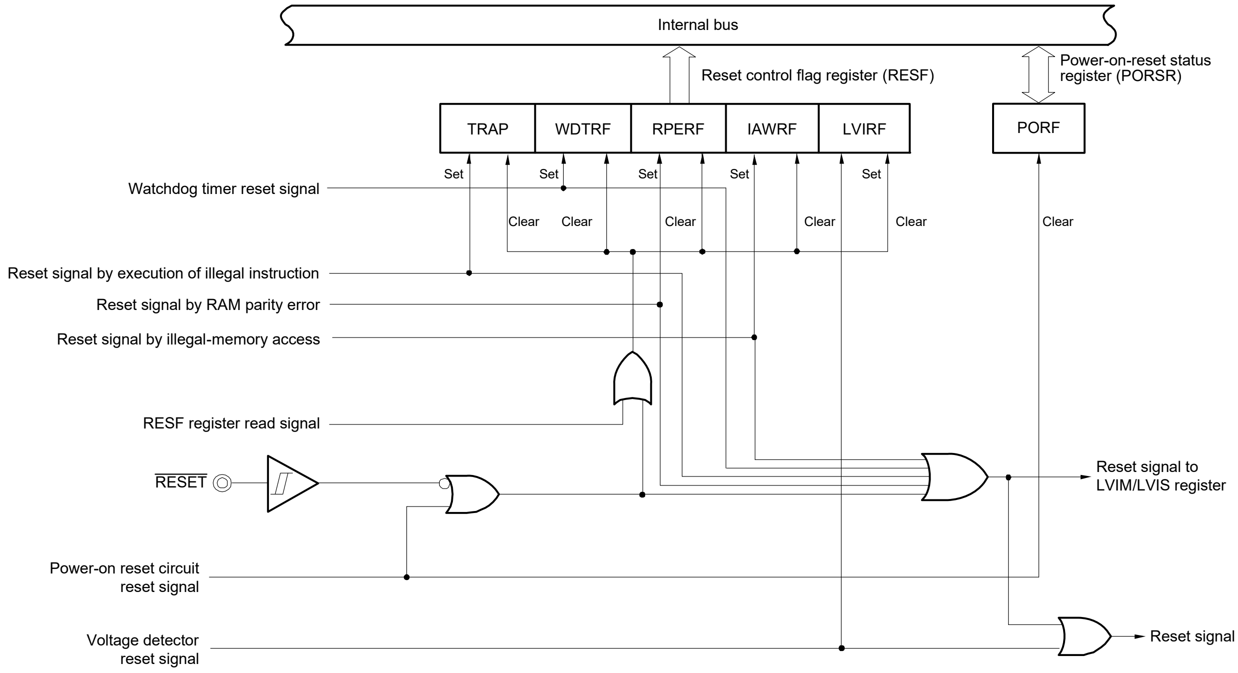
Note The illegal instruction is generated when instruction code FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Caution 1. 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 34.4 AC Characteristics, and then input a high level to the pin.

Caution 2. During generation of a reset signal, the X1 clock, XT1 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 and external subsystem clock input become invalid.

Caution 3. The port pins become the following state because each SFR and 2nd SFR are initialized after reset.
- **P40: High-impedance during the external reset period or reset period by the POR. High level during other types of reset or after receiving a reset signal (connected to the on-chip pull-up resistance).**
- **Ports other than P40: High-impedance during the reset period or after receiving a reset signal.**

Figure 24 - 1 Block Diagram of Reset Function



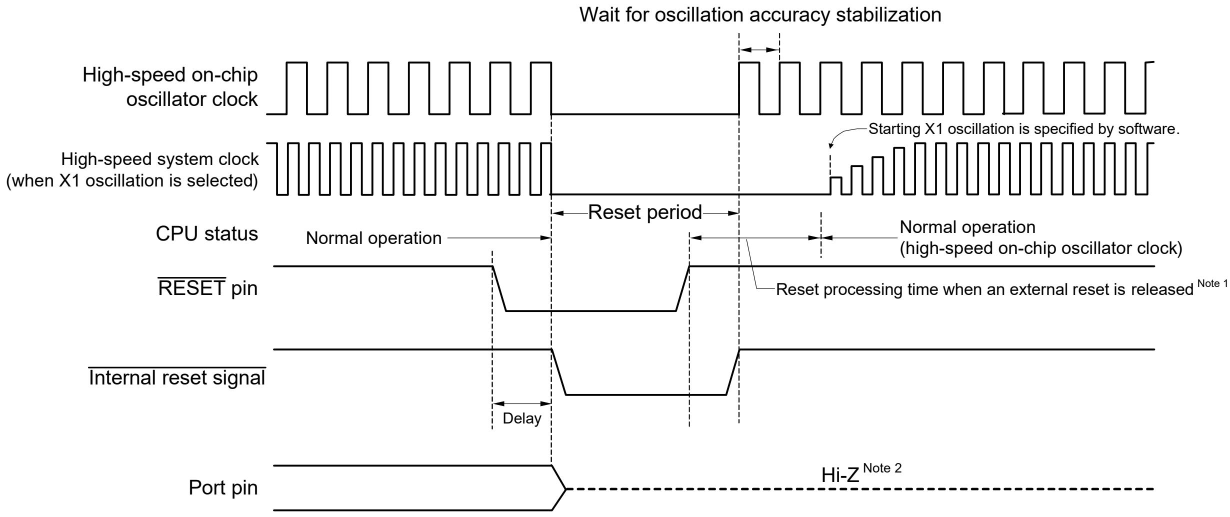
Caution An LVD circuit internal reset does not reset the LVD circuit.

Remark 1. LVIM: Voltage detection register
Remark 2. LVIS: Voltage detection level register

24.1 Timing of Reset Operation

This LSI 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.

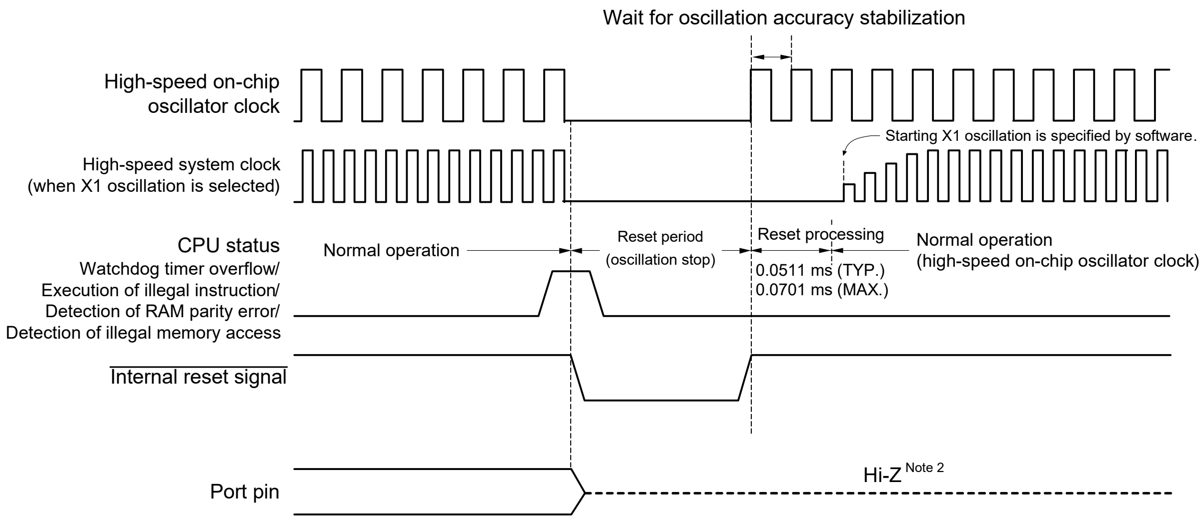
Figure 24 - 2 Timing of Reset by $\overline{\text{RESET}}$ Input



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

Release from the reset state is automatic in the case of a reset due to a watchdog timer overflow, 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 Watchdog Timer Overflow, Execution of Illegal Instruction, Detection of RAM Parity Error, or Detection of Illegal Memory Access



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

Note 1. Reset times (times for release from the external reset state)

After the first release of the POR:	0.672 ms (TYP.), 0.832 ms (MAX.) when the LVD is in use.
	0.399 ms (TYP.), 0.519 ms (MAX.) when the LVD is off.
After the second release of the POR:	0.531 ms (TYP.), 0.675 ms (MAX.) when the LVD is in use.
	0.259 ms (TYP.), 0.362 ms (MAX.) when the LVD is off.

After power is supplied, a voltage stabilization waiting time of about 0.99 ms (TYP.) and up to 2.30 ms (MAX.) is required before reset processing starts after release of the external reset.

Note 2. The state of P40 is as follows.

- High-impedance during the external reset period or reset period by the POR.
- High level during other types of reset or after receiving a reset signal (connected to the on-chip pull-up resistance).

Reset by POR and LVD circuit supply voltage detection is automatically released when $V_{DD} \geq V_{POR}$ or $V_{DD} \geq V_{LVD}$ after the reset. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts. For details, see **CHAPTER 25 POWER-ON-RESET CIRCUIT** or **CHAPTER 26 VOLTAGE DETECTOR**.

Remark V_{POR}: POR power supply rise detection voltage
V_{LVD}: LVD detection voltage

Table 24 - 1 Operation Statuses During Reset Period

Item			During Reset Period
System clock			Clock supply to the CPU is stopped.
	Main system clock	f_{IH}	Operation stopped
		f_{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}	Operable (the XT1 and XT2 pins are input port mode)
		f_{EXS}	Clock input invalid (the pin is input port mode)
	f_{IL}		Operation stopped
CPU			
Code flash memory			Operation stopped
Data flash memory			Operation stopped
RAM			Operation stopped
Port (latch)			High impedance [Note]
Timer array unit			Operation stopped
RTC2			Other than POR reset: Operable POR reset: Calendar operation possible, Operation of the RTCC0, RTCC1, and SUBCUD registers is stopped
Frequency measurement function			Operation stopped
12-bit Interval timer			
8-bit Interval timer			
Watchdog timer			
Clock output/buzzer output			
A/D converter			
Comparator			
Operational amplifier function			
Serial array unit (SAU)			
Data operation circuit (DOC)			
Data transfer controller (DTC)			
Event link controller (ELC)			
Power-on-reset function			Detection operation possible
Voltage detection function			Operation is possible in the case of an LVD reset and stopped in the case of other types of reset.
External interrupt			Operation stopped
Key interrupt function			
CRC operation function		High-speed CRC	
		General-purpose CRC	
Illegal-memory access detection function			
RAM parity error detection function			
RAM guard function			
SFR guard function			

Note P40 becomes the following state.

- 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 resistance).

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 Hardware Statuses After Reset Acknowledgment

Hardware		After Reset Acknowledgment Note
Program counter (PC)		The contents of the reset vector table (0000H, 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		06H
RAM	Data memory	Undefined
	General-purpose registers	Undefined

Note During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

Remark For the state of the special function register (SFR) after receiving a reset signal, 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 Register for Confirming Reset Source

24.2.1 Reset control flag register (RESF)

Many internal reset generation sources exist in the RL78 microcontroller. The reset control flag register (RESF) is used to store which source has generated the reset request.

The RESF register can be read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input, reset by power-on-reset (POR) circuit, and accessing SFR other than the RESF register after reading the RESF register clear TRAP, WDTRF, RPERF, IAWRF, and LVIRF flags.

Figure 24 - 4 Format of Reset control flag register (RESF)

Address: FFFA8H After reset: Undefined Note 1 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 t by RAM parity
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

IAWRF	Internal reset request t by illegal-memory access
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

LVIRF	Internal reset request by voltage detector (LVD)
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

Note 1. The value after reset varies depending on the reset source. See **Table 24 - 3**.

Note 2. The illegal instruction is generated when instruction code FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Caution 1. Do not read data by a 1-bit memory manipulation instruction.

Caution 2. When enabling RAM parity error resets (RPERDIS = 0), be sure to initialize the used RAM area at data access or the used RAM area + 10 bytes at execution of instruction from the RAM area.
Reset generation enables RAM parity error resets (RPERDIS = 0). For details, see 27.3.3 RAM parity error detection function.

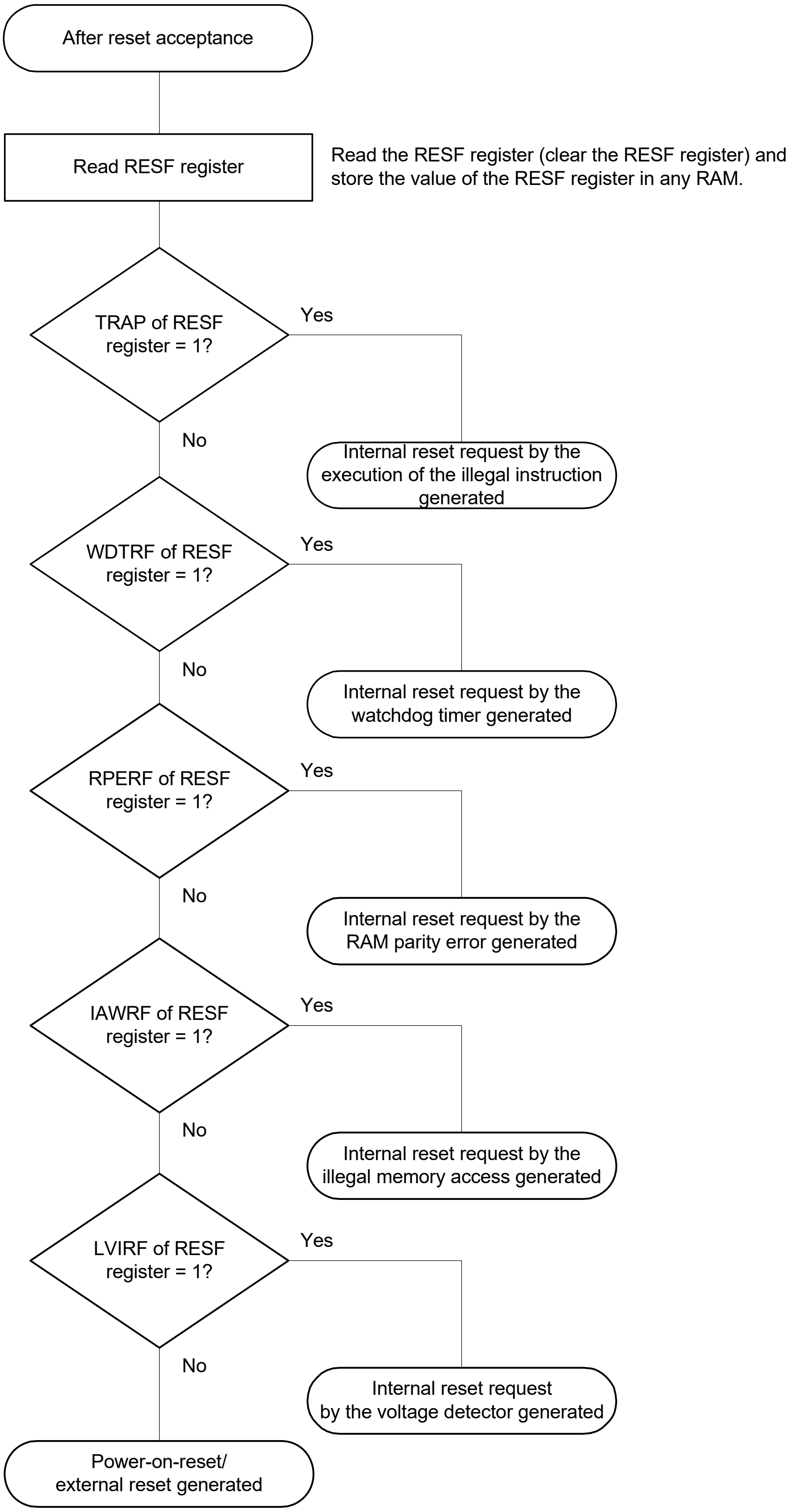
The status of the RESF register when a reset request is generated is shown in Table 24 - 3.

Table 24 - 3 RESF Register Status When Reset Request Is Generated

Flag \ Reset Source	$\overline{\text{RESET}}$ 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 LVD
TRAP	Cleared (0)	Cleared (0)	Set (1)	Held	Held	Held	Held
WDTRF			Held	Set (1)			
RPERF				Held	Set (1)		
IAWRF					Held	Set (1)	
LVIRF						Held	Set (1)

Accessing SFR other than the RESF register after reading the RESF register by an 8-bit memory manipulation instruction clears the RESF register automatically. Figure 24 - 5 shows the procedure for checking a reset source.

Figure 24 - 5 Example of Procedure for Checking 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.
Writing 1 to bit 0 (PORF) of the PORSR register enables this function. Writing 0 disables this function.
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 an 8-bit memory manipulation instruction.
Power-on reset signal generation clears this register to 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. If the PORF bit is set to 1, it guarantees that no power-on reset has occurred, but it does not guarantee that the RAM value is retained.

Figure 24 - 6 Format of Power-on-reset status register (PORSR)

Address: F00F9H After reset: 00H 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)

This register is used for individual reset control of each peripheral hardware.

This MCU controls reset and reset release of each peripheral hardware supported by the PRR0 register.

Figure 24 - 7 Format of Peripheral reset control register 0 (PRR0)

Address: F00F1H After reset: 00H R/W

Symbol	7	6	<5>	4	3	<2>	1	<0>
PRR0	0	0	ADCRES	0	0	SAU0RES	0	TAU0RES

PRR0n	Peripheral reset control on each peripheral hardware
0	Peripheral reset release
1	Peripherals reset state

Remark n = 0, 2, or 5

The controlled hardware by each bit are as follows.

Table 24 - 4 Controlled Hardware by Each Bit in PRR0

Bit	Bit Name	Controlled Hardware
0	TAU0RES	Timer array unit (Unit 0)
2	SAU0RES	Serial array unit (Unit 0)
5	ADCRES	A/D Converter

24.2.4 Peripheral reset control register 1 (PRR1)

This register is used for individual reset control of each peripheral hardware.

This MCU controls reset and reset release of each peripheral hardware supported by the PRR1 register.

Figure 24 - 8 Format of Peripheral reset control register 1 (PRR1)

Address: F00FBH After reset: 00H R/W

Symbol	7	6	<5>	4	3	2	1	0
PRR1	0	0	CMPRES	0	0	0	0	0

PRR1n	Peripheral reset control on each peripheral hardware
0	Peripheral reset release
1	Peripherals reset state

Remark n = 5

The controlled hardware by each bit are as follows.

Table 24 - 5 Macro Controlled by Each Bit in PRR1

Bit	Bit Name	Controlled Hardware
5	CMPRES	Comparators 0 and 1

24.2.5 Peripheral reset control register 2 (PRR2)

This register is used for individual reset control of each peripheral hardware.

This MCU controls reset and reset release of each peripheral hardware supported by the PRR2 register.

Figure 24 - 9 Format of Peripheral reset control register 2 (PRR2)

Address: F00FDH After reset: 00H R/W

Symbol	<7>	6	<5>	4	3	2	1	0
PRR2	TMKARES	0	DOCRES	0	0	0	0	0

PRR2n	Peripheral reset control on each peripheral hardware macro
0	Peripheral reset release
1	Peripherals reset state

Remark n = 5, 7

The controlled hardware by each bit are as follows.

Table 24 - 6 Macro Controlled by Each Bit in PRR2

Bit	Bit Name	Controlled Hardware
5	DOCRES	Data operation circuit (DOC)
7	TMKARES	12-bit interval timer

CHAPTER 25 POWER-ON-RESET CIRCUIT

25.1 Functions of Power-on-reset Circuit

The power-on-reset circuit (POR) has the following functions.

- Generates internal reset signal at power on.
  The reset signal is released when the supply voltage (V_{DD}) exceeds the detection voltage (V_{POR}). Note that the reset state must be retained until the operating voltage becomes in the range defined in **34.4 AC Characteristics**. This is done by utilizing the voltage detection circuit 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 is supplied, this LSI should be placed in the STOP mode, or in the reset state by utilizing the voltage detection circuit or externally input reset signal, before the operation voltage falls below the range defined in **34.4 AC Characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Caution If an internal reset signal is generated in the power-on-reset circuit, the reset control flag register (RESF) and the power-on-reset status register (PORSR) are cleared (00H).

Remark 1. The RL78 microcontroller 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 **CHAPTER 24 RESET FUNCTION**.

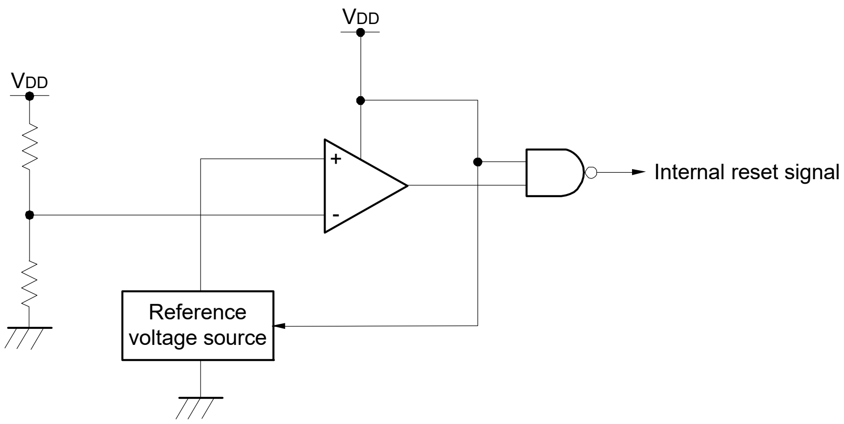
Remark 2. The occurrence of an internal reset in the power-on-reset circuit can be checked by the power-on reset status register (PORSR). For details on the PORSR register, refer to **CHAPTER 24 RESET FUNCTION**.

Remark 3. V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage
For details, see **34.6.5 POR circuit characteristics**.

25.2 Configuration of Power-on-reset Circuit

The block diagram of the power-on-reset circuit is shown in Figure 25 - 1.

Figure 25 - 1 Block Diagram of Power-on-reset Circuit

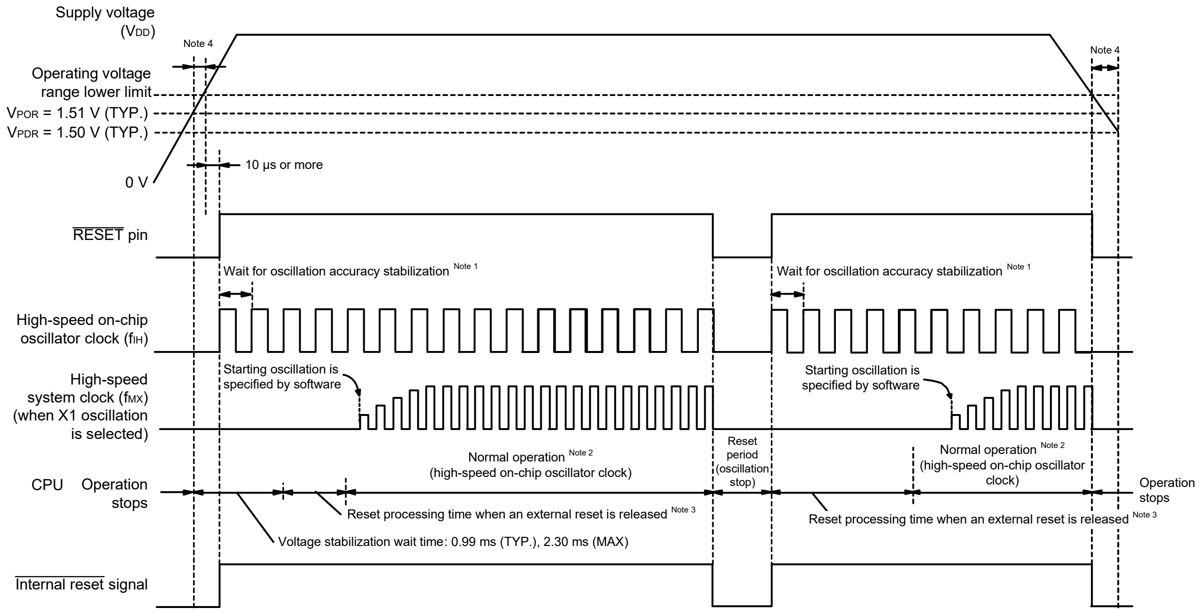


25.3 Operation of Power-on-reset Circuit

The timing of generation of the internal reset signal by the power-on-reset circuit and voltage detector is shown next.

Figure 25 - 2 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (1/3)

(1) When using an external reset by the $\overline{\text{RESET}}$ pin



Note 1. The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

Note 2. The high-speed on-chip oscillator clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

Note 3. The time until normal operation starts includes the following reset processing time when the external reset is released (after the first release of POR) after the $\overline{\text{RESET}}$ signal is driven high (1) as well as the voltage stabilization wait time after VPOR (1.51 V, TYP.) is reached.

Reset processing time when the external reset is released is shown below.

After the first release of POR: 0.672 ms (TYP.), 0.832 ms (MAX.) (when the LVD is in use)
0.399 ms (TYP.), 0.519 ms (MAX.) (when the LVD is off)

Reset processing time when the external reset is released after the second release of POR is shown below.

After the second release of POR: 0.531 ms (TYP.), 0.675 ms (MAX.) (when the LVD is in use)
0.259 ms (TYP.), 0.362 ms (MAX.) (when the LVD is off)

Note 4. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **34.4 AC Characteristics**. This is done by controlling the externally input reset signal.

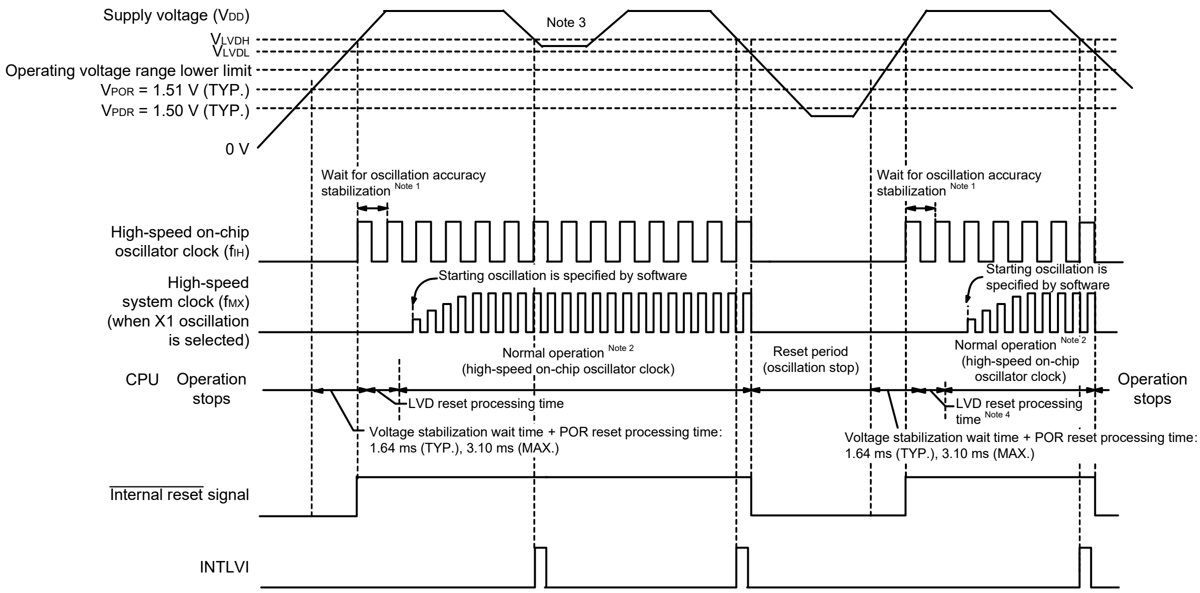
After power supply is turned off, this LSI should be placed in the STOP mode, or in the reset state by utilizing the voltage detection circuit or externally input reset signal, before the voltage falls below the operating range. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Remark VPOR: POR power supply rise detection voltage
VPDR: POR power supply fall detection voltage

Caution For power-on reset, be sure to use the externally input reset signal on the $\overline{\text{RESET}}$ pin when the LVD is off. For details, see CHAPTER 26 VOLTAGE DETECTOR.

Figure 25 - 3 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (2/3)

(2) LVD is interrupt & reset mode (option byte 000C1: LVIMDS1, LVIMDS0 = 1, 0)



Note 1. The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

Note 2. The high-speed on-chip oscillator clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

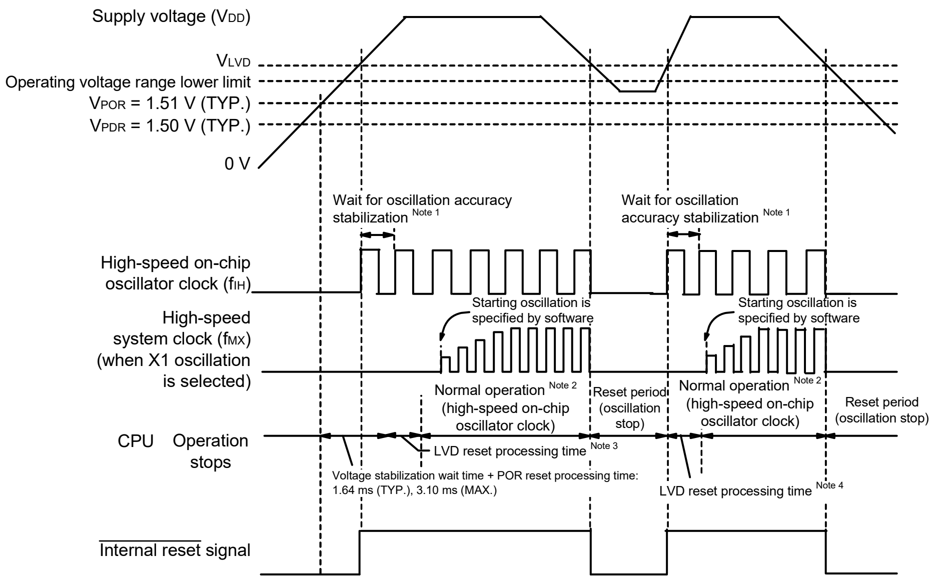
Note 3. After the interrupt request signal (INTLVI) is generated, the LVILV and LVIMD bits of the voltage detection level register (LVIS) are automatically set to 1. After INTLVI is generated, appropriate settings should be made according to **Figure 26 - 9 Setting Procedure for Operating Voltage Check and Reset**, taking into consideration that the supply voltage might return to the high voltage detection level (V_{LVDH}) or higher without falling below the low voltage detection level (V_{LVDL}).

Note 4. The time until normal operation starts includes the following LVD reset processing time after the LVD detection level (V_{LVDH}) is reached as well as the voltage stabilization wait + POR reset processing time after the V_{POR} (1.51 V, TYP.) is reached.
LVD reset processing time: 0 ms to 0.0701 ms (MAX.)

Remark V_{LVDH}, V_{LVDL}: LVD detection voltage
V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Figure 25 - 4 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (3/3)

(3) LVD reset mode (option byte 000C1H: LVIMDS1, LVIMDS0 = 1, 1)



Note 1. The internal reset processing time includes the oscillation accuracy stabilization time of the high-speed on-chip oscillator clock.

Note 2. The high-speed on-chip oscillator clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

Note 3. The time until normal operation starts includes the following LVD reset processing time after the LVD detection level (V_{LVD}) is reached as well as the voltage stabilization wait + POR reset processing time after the V_{POR} (1.51 V, TYP.) is reached.
LVD reset processing time: 0 ms to 0.0701 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 (LVD), the following LVD reset processing time is required after the LVD detection level (V_{LVD}) is reached.
LVD reset processing time: 0.0511 ms (TYP.), 0.0701 ms (MAX.)

Remark 1. V_{LVDH}, V_{LVDL}: LVD detection voltage
V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Remark 2. When the LVD interrupt mode is selected (option byte 000C1H: LVIMD1 = 0, LVIMD0 = 1), the time until normal operation starts after power is turned on is the same as the time specified in Note 3 of Figure 25 - 4 (3).

CHAPTER 26 VOLTAGE DETECTOR

26.1 Functions of Voltage Detector

The operation mode and detection voltages (V_{LVDH}, V_{LVDL}, V_{LVD}) for the voltage detector is set by using the option byte (000C1H). The detection voltages can be reset using the LVIS register. The voltage detector (LVD) has the following functions.

- The LVD circuit compares the supply voltage (V_{DD}) with the detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}), and generates an internal reset or internal interrupt signal.
- The detection level for the power supply detection voltage (V_{LVDH}, V_{LVDL}) can be selected as one of 14 levels (For details, see **26.3.2 Voltage detection level register (LVIS)** and **CHAPTER 29 OPTION BYTE**).
- Operable in STOP mode.
- After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **34.4 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 LSI 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 010C2H).

(a) Interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0)
The two detection voltages (V_{LVDH}, V_{LVDL}) are selected by the option byte 000C1H. The high-voltage detection level (V_{LVDH}) is used for releasing resets and generating interrupts. This level is also used for generating resets. The low-voltage detection level (V_{LVDL}) is used for generating resets.

(b) Reset mode (option byte LVIMDS1, LVIMDS0 = 1, 1)
The detection voltage (V_{LVD}) selected by the option byte 000C1H is used for triggering and ending resets. The detection voltages can be reset using the LVIS register.

(c) Interrupt mode (option byte LVIMDS1, LVIMDS0 = 0, 1)
The detection voltage (V_{LVD}) selected by the option byte 000C1H is used for generating interrupts/reset release. The detection voltages can be reset using the LVIS register.

The reset and internal interrupt signals are generated in each mode as follows.

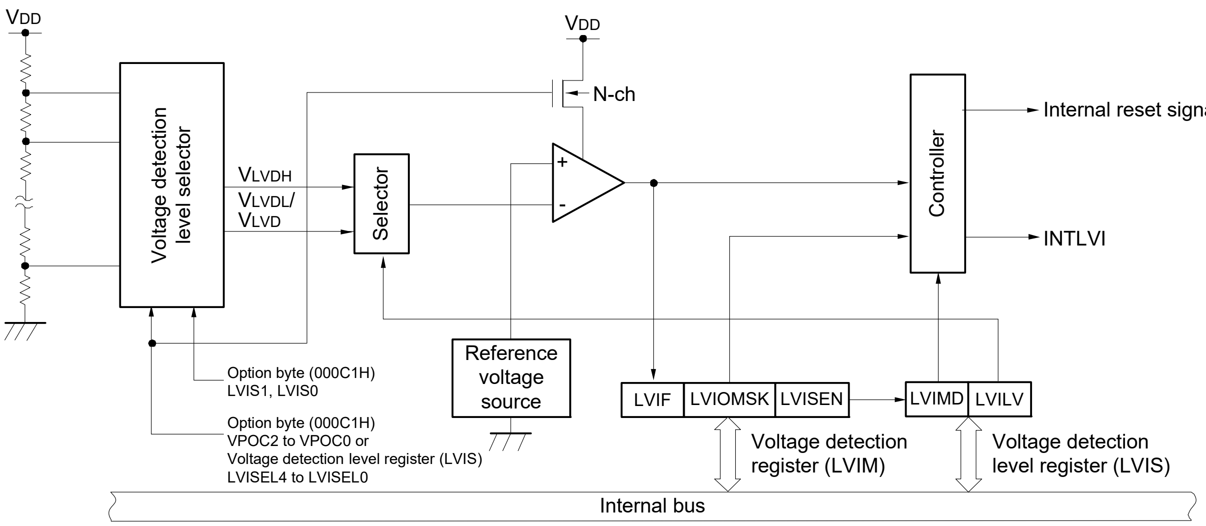
Interrupt & reset mode (LVIMDS1, LVIMDS0 = 1, 0)	Reset mode (LVIMDS1, LVIMDS0 = 1, 1)	Interrupt mode (LVIMDS1, LVIMDS0 = 0, 1)
Generates an interrupt request signal by detecting $V_{DD} < V_{LVDH}$ when the operating voltage falls, and an internal reset by detecting $V_{DD} < V_{LVDL}$. Releases an internal reset by detecting $V_{DD} \geq V_{LVDH}$.	Releases an internal reset by detecting $V_{DD} \geq V_{LVD}$. Generates an internal reset by detecting $V_{DD} < V_{LVD}$.	Retains the state of an internal reset by the LVD immediately after a reset until $V_{DD} \geq V_{LVD}$. Releases the LVD internal reset by detecting $V_{DD} \geq V_{LVD}$. Generates an interrupt request signal (INTLVI) by detecting $V_{DD} < V_{LVD}$ or $V_{DD} \geq V_{LVD}$ after the LVD internal reset is released.

While the voltage detector is operating, whether the supply voltage is more than or less than the detection level can be checked by reading the voltage detection flag (LVIF: bit 0 of the voltage detection register (LVIM)).
Bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of the RESF register, see **CHAPTER 24 RESET FUNCTION**.

26.2 Configuration of Voltage Detector

The block diagram of the voltage detector is shown in Figure 26 - 1.

Figure 26 - 1 Block Diagram of Voltage Detector



26.3 Registers Controlling Voltage Detector

The voltage detector is controlled by the following registers.

- Voltage detection register (LVIM)
- Voltage detection level register (LVIS)

26.3.1 Voltage detection register (LVIM)

This register is used to specify whether to enable or disable rewriting the voltage detection level register (LVIS), as well as to check the LVD output mask status.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 26 - 2 Format of Voltage detection register (LVIM)

Address: FFFA9H After reset: 00H Note 1 R/W Note 2

Symbol	<7>	6	5	4	3	2	<1>	<0>
LVIM	LVISEN	0	0	0	0	0	LVIOMSK	LVIF

LVISEN	Specification of whether to enable or disable rewriting the voltage detection level register (LVIS)
0	Disabling of rewriting the LVIS register (LVIOMSK = 0 (Mask of LVD output is invalid)
1	Enabling of rewriting the LVIS register Note 3 (LVIOMSK = 1 (Mask of LVD output is valid)

LVIOMSK	Mask status flag of LVD output
0	Mask of LVD output is invalid
1	Mask of LVD output is valid Note 3

LVIF	Voltage detection flag
0	Supply voltage (V_{DD}) ≥ detection voltage (V_{LVD}), or when LVD is off
1	Supply voltage (V_{DD}) < detection voltage (V_{LVD})

Note 1. The reset value changes depending on the reset source.
If the LVIS register is reset by LVD, it is not reset but holds the current value. In other reset, LVISEN is cleared to 0.

Note 2. Bits 0 and 1 are read-only.

Note 3. The LVIOMSK bit is automatically set to 1 for the following periods and generation of an LVD reset or interrupt is masked.

- Period when LVISEN = 1
  In either of the following cases, generation of an LVD reset or interrupt is masked only in interrupt & reset mode.
- Wait time until the LVD detection voltage stabilizes after an LVD interrupt is generated
- Wait time until the LVD detection voltage stabilizes after the value of the LVILV bit is changed

26.3.2 Voltage detection level register (LVIS)

This register selects the voltage detection level. The minimum supply voltage (LVD detection voltage) and LVD detection level settings that are set by the user option byte can be changed by software.
This register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation input sets this register to Note 1.

Caution Do not change the detection voltage in interrupt & reset mode.

Figure 26 - 3 Format of Voltage detection level register (LVIS)

Address: FFFAAH After reset: Note 1 R/W

Symbol	<7>	6	5	4	3	2	1	<0>
LVIS	LVIMD Note 2	0	LVISEL4 Note 6	LVISEL3	LVISEL2	LVISEL1	LVISEL0	LVILV Note 2

LVIMD Note 2	Operation mode of voltage detection
0	Interrupt mode
1	Reset mode

LVISEL4 Note 6	LVISEL3	LVISEL2	Minimum operating voltage (typical falling value) Note 5
0	0	1	1.84 V
0	1	0	2.45 V
0	1	1	2.75 V
1	1	1	1.53 V (LVD OFF)
Other than above			Setting prohibited

LVISEL1	LVISEL0	LVD detection level setting Note 5
0	0	Setting voltage according to LVISEL4/LVISEL3/LVISEL2 + 1.2 V Note 3
0	1	Setting voltage according to LVISEL4/LVISEL3/LVISEL2 + 0.2 V Note 3
1	0	Setting voltage according to LVISEL4/LVISEL3/LVISEL2 + 0.1 V Note 3
1	1	Setting voltage according to LVISEL4/LVISEL3/LVISEL2 Note 4

LVILV Note 2	LVD detection level
0	High-voltage detection level (V_{LVDH})
1	Low-voltage detection level (V_{LVDL} or V_{LVD})

Note 1. The reset value changes depending on the setting of the option byte.
After a reset is released, the values of VPOC2 to VPOC0 and LVIS1 and LVIS0 in the user option byte are reflected in LVISEL4 to LVISEL2, LVISEL1, and LVISEL0, respectively.
The reset values of LVIMD and LVIVL are set as follows.
When LVIMDS1, LVIMDS0 in the option byte = 1, 0: LVIMD = 0, LVILV = 0
When LVIMDS1, LVIMDS0 in the option byte = 1, 1: LVIMD = 1, LVILV = 1
When LVIMDS1, LVIMDS0 in the option byte = 0, 1: LVIMD = 0, LVILV = 1

Note 2. Writing “0” can only be allowed in the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0). Do not set LVIMD and LVILV in other cases. The value is switched automatically when reset or interrupt is generated in the interrupt & reset mode.

Note 3. Indicates an approximate detection value. For details on the actual detection voltage, refer to the LVD section in Electrical Specifications.

Note 4. Cannot be selected when LVIMDS1 and LVIMDS0 = 1 and 0.

Note 5. When changing LVISEL4 to LVISEL0 to use two or more LVD detection voltages, the setting value that indicates the highest voltage value among the LVD detection voltages to be used should be set in the VPOC2 to VPOC0 bits and LVIS1 and LVIS0 bits before using the voltages.

Note 6. Rewriting LVISEL4 is prohibited. Keep the initial value unchanged.

Caution 1. When rewriting the LVIMD and LVILV bits, use the procedure shown in Figure 26 - 9.

Caution 2. Specify the LVD operation mode and initial detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}) of each mode by using the option byte 000C1H. Figure 26 - 4 shows the format of the user option byte (000C1H/010C1H). For details about the option byte, see CHAPTER 29 OPTION BYTE.

Figure 26 - 4 Format of User Option Byte (000C1H/010C1H) (1/2)

Address: 000C1H/010C1H[Note]

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD setting (interrupt & reset mode)

Detection voltage			Option byte Setting Value						
V_{LVDH}		V_{LVDL}	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge	Falling edge						LVIMDS1	LVIMDS0
1.77 V	1.73 V	1.63 V	0	0	0	1	0	1	0
1.88 V	1.84 V					0	1		
2.92 V	2.86 V					0	0		
1.98 V	1.94 V	1.84 V		0	1	1	0		
2.09 V	2.04 V					0	1		
3.13 V	3.06 V					0	0		
2.61 V	2.55 V	2.45 V		1	0	1	0		
2.71 V	2.65 V					0	1		
2.92 V	2.86 V	2.75 V		1	1	1	0		
3.02 V	2.96 V					0	1		
—			Settings other than the above are prohibited						

• LVD setting (reset mode)

Detection voltage		Option byte Setting Value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
1.67 V	1.63 V	0	0	0	1	1	1	1
1.77 V	1.73 V		0	0	1	0		
1.88 V	1.84 V		0	1	1	1		
1.98 V	1.94 V		0	1	1	0		
2.09 V	2.04 V		0	1	0	1		
2.50 V	2.45 V		1	0	1	1		
2.61 V	2.55 V		1	0	1	0		
2.71 V	2.65 V		1	0	0	1		
2.81 V	2.75 V		1	1	1	1		
2.92 V	2.86 V		1	1	1	0		
3.02 V	2.96 V		1	1	0	1		
3.13 V	3.06 V		0	1	0	0		
—		Settings other than the above are prohibited						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Remark 1. For details on the LVD circuit, see **CHAPTER 26 VOLTAGE DETECTOR**.

Remark 2. The detection voltage is a TYP. value. For details, see **34.6.6 LVD circuit characteristics**.

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

Figure 26 - 4 Format of User Option Byte (000C1H/010C1H) (2/2)

Address: 000C1H/010C1H[Note]

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD setting (interrupt mode)

Detection voltage		Option byte Setting Value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
1.67 V	1.63 V	0	0	0	1	1	0	1
1.77 V	1.73 V		0	0	1	0		
1.88 V	1.84 V		0	1	1	1		
1.98 V	1.94 V		0	1	1	0		
2.09 V	2.04 V		0	1	0	1		
2.50 V	2.45 V		1	0	1	1		
2.61 V	2.55 V		1	0	1	0		
2.71 V	2.65 V		1	0	0	1		
2.81 V	2.75 V		1	1	1	1		
2.92 V	2.86 V		1	1	1	0		
3.02 V	2.96 V		1	1	0	1		
3.13 V	3.06 V		0	1	0	0		
—		Settings other than the above are prohibited						

Detection voltage		Option byte Setting Value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
—	—	1	×	×	×	×	×	1
—		Settings other than the above are prohibited						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Caution 1. Set bit 4 to 1.

Caution 2. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 34.4 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 LSI 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 010C2H).

Remark 1. ×: Don't care

Remark 2. For details on the LVD circuit, see **CHAPTER 26 VOLTAGE DETECTOR**.

Remark 3. The detection voltage is a TYP. value. For details, see **34.6.6 LVD circuit characteristics**.

26.4 Operation of Voltage Detector

26.4.1 When used as reset mode

Specify the operation mode (the reset mode (LVIMDS1, LVIMDS0 = 1, 1)) and the initial detection voltage (V_{LVD}) by using the option byte 000C1H. The detection voltages can be reset using the LVIS register.

The operation is started in the following initial setting state when the reset mode is set.

- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- The initial value of the voltage detection level select register (LVIS) is set to 81H.
  See **26.3.2 Voltage detection level register (LVIS)** for details on the initial value of the voltage detection level register (LVIS).
  Bit 7 (LVIMD) is 1 (reset mode).
  Bit 0 (LVILV) is 1 (low-voltage detection level: V_{LVD}).

• Operation in LVD reset mode

In the reset mode (option byte LVIMDS1, LVIMDS0 = 1, 1), the state of an internal reset by LVD is retained until the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}) after power is supplied. The internal reset is released when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}).

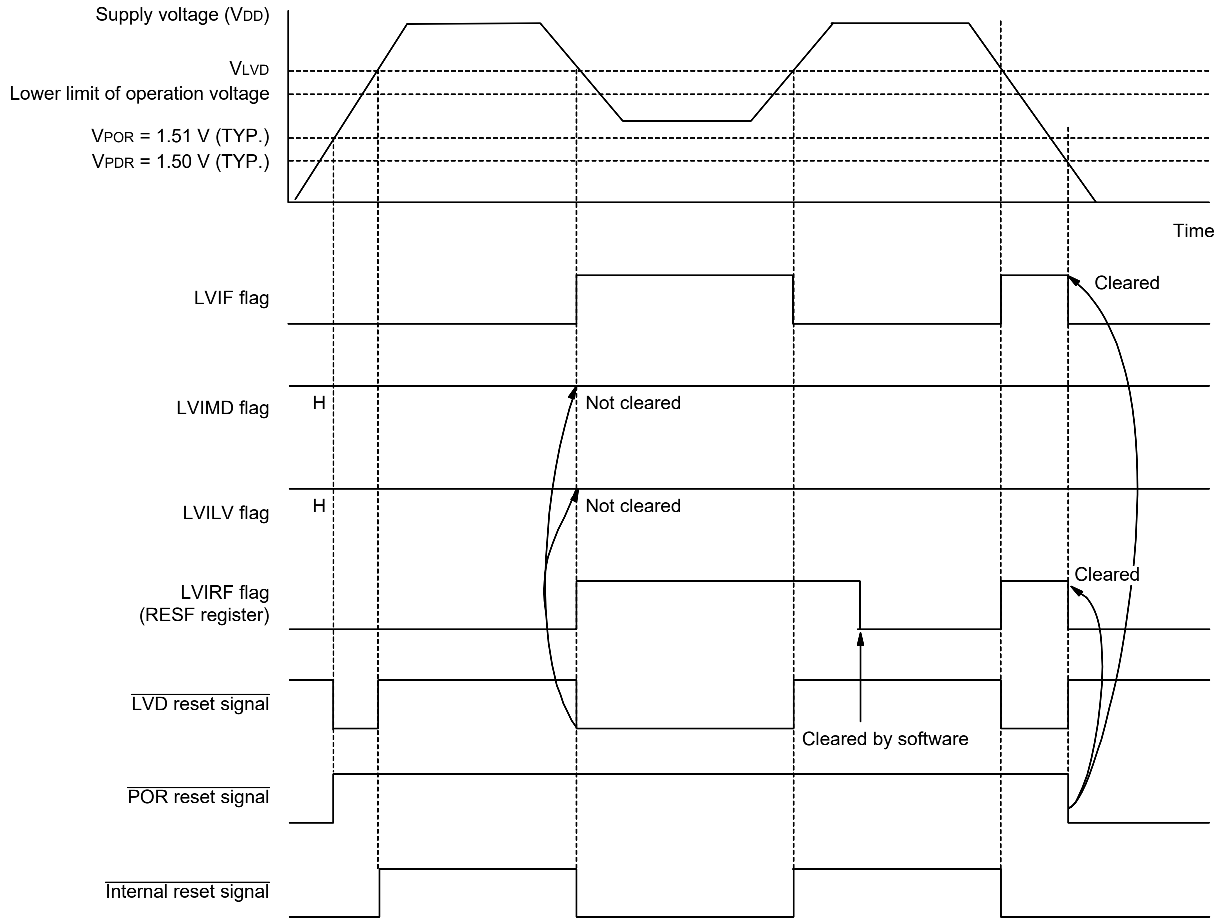
At the fall of the operating voltage, an internal reset by LVD is generated when the supply voltage (V_{DD}) falls below the voltage detection level (V_{LVD}).

The reset release voltage when an LVD reset is generated is the detection voltage set by the option byte or detection voltage set by the LVIS register, whichever is higher. The state of an internal reset by the LVD is retained until the supply voltage exceeds the voltage detection level.

The reset release voltage used for resets other than an LVD reset is the same voltage detection level set by the option byte.

Figure 26 - 5 shows the timing of the internal reset signal generated in the LVD reset mode.

Figure 26 - 5 Timing of Voltage Detector Internal Reset Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 1)



Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

26.4.2 When used as interrupt mode

Specify the operation mode (the interrupt mode (LVIMDS1, LVIMDS0 = 0, 1)) and the initial detection voltage (V_{LVD}) by using the option byte 000C1H. The detection voltages can be reset using the LVIS register.

The operation is started in the following initial setting state when the interrupt mode is set.

- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- See **26.3.2 Voltage detection level register (LVIS)** for details on the initial value of the voltage detection level register (LVIS).
  Bit 7 (LVIMD) is 0 (interrupt mode).
  Bit 0 (LVILV) is 1 (low-voltage detection level: V_{LVD}).

• Operation in LVD interrupt mode

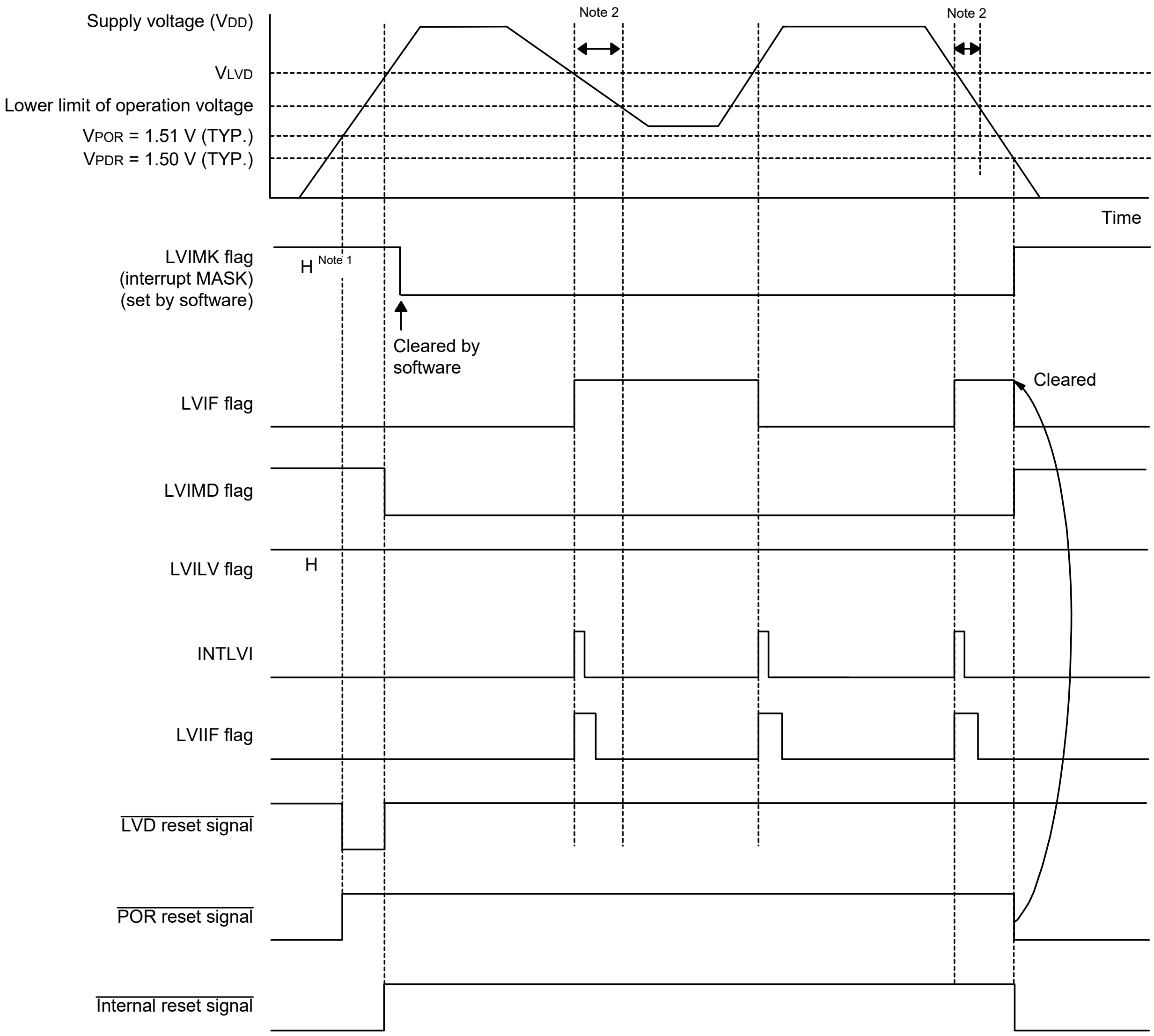
In the interrupt mode (option byte LVIMDS1, LVIMDS0 = 0, 1), the state of an internal reset by LVD is retained immediately after a reset until the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}) after power is supplied (after the first release of the POR). The LVD internal reset is released when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}).

After the LVD internal reset is released, an interrupt request signal (INTLVI) by the LVD is generated when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}). When the voltage falls, this LSI 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 **34.4 AC Characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

After the LVISEN bit is set to 1 (LVD is masked) by changing the detection level, if the supply voltage (V_{DD}) falls below the voltage detection level (V_{LVD}) when LVISEN is set to 0, an interrupt request signal (INTLVI) by the LVD is generated.

Figure 26 - 6 shows the timing of the interrupt request signal generated in the LVD interrupt mode.

Figure 26 - 6 Timing of Voltage Detector Internal Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 0, 1)



Note 1. The LVIMK flag is set to "1" by reset signal generation.

Note 2. When the voltage falls, this LSI 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 **34.4 AC Characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

26.4.3 When used as interrupt and reset mode

Specify the operation mode (the interrupt & reset (LVIMDS1, LVIMDS0 = 1, 0)) and the detection voltage (V_{LVDH}, V_{LVDL}) by using the option byte 000C1H. Do not manipulate the detection voltage using the LVIS register.

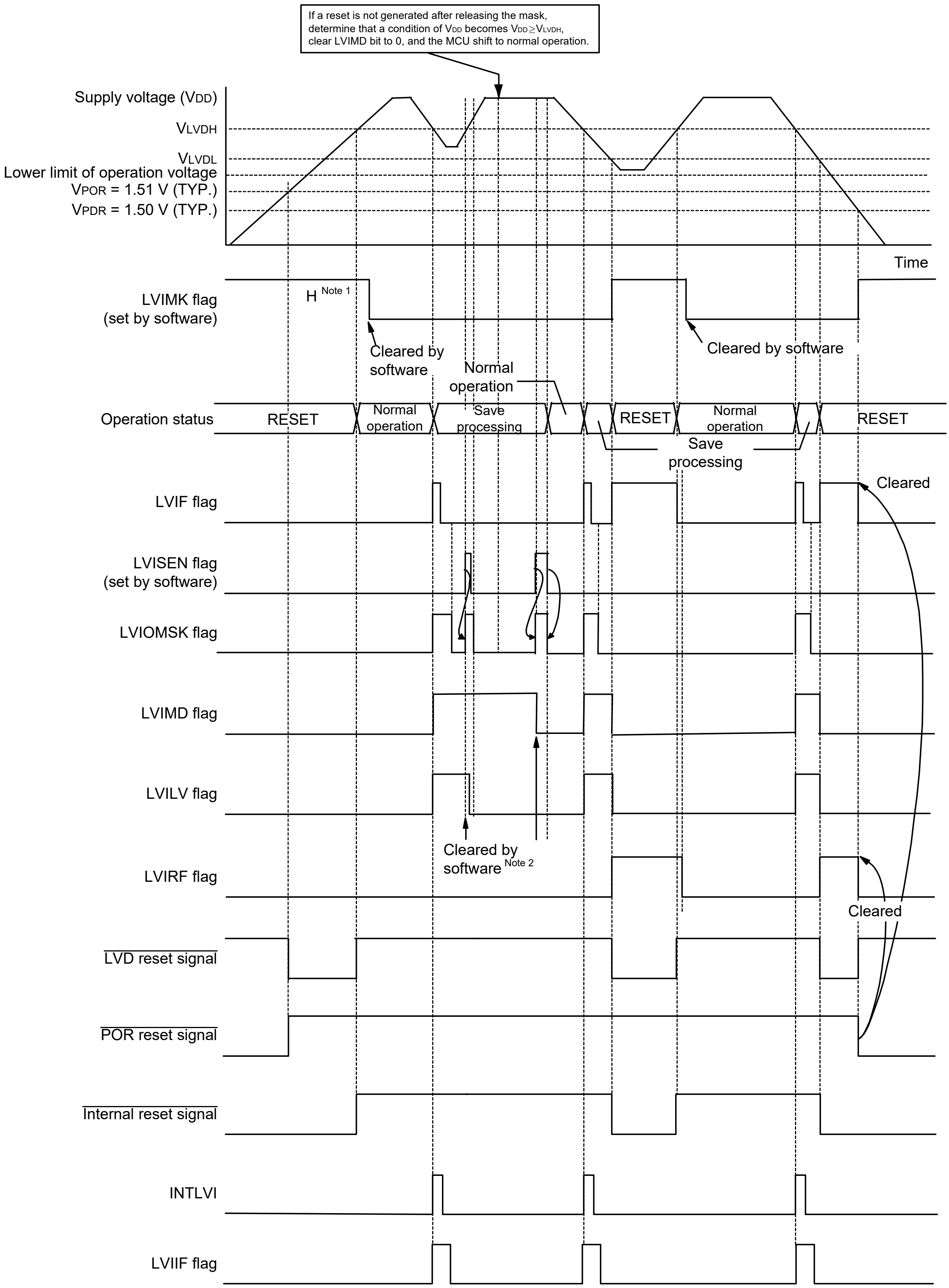
The operation is started in the following initial setting state when the interrupt & reset mode is set.

- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- See **26.3.2 Voltage detection level register (LVIS)** for details on the initial value of the voltage detection level register (LVIS).
  Bit 7 (LVIMD) is 0 (interrupt mode).
  Bit 0 (LVILV) is 0 (high-voltage detection level: V_{LVDH}).

• Operation in LVD interrupt & reset mode
In the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0), the state of an internal reset by LVD is retained until the supply voltage (V_{DD}) exceeds the high-voltage detection level (V_{LVDH}) after power is supplied. The internal reset is released when the supply voltage (V_{DD}) exceeds the high-voltage detection level (V_{LVDH}).
An interrupt request signal by LVD (INTLVI) is generated and arbitrary save processing is performed when the supply voltage (V_{DD}) falls below the high-voltage detection level (V_{LVDH}). After that, an internal reset by LVD is generated when the supply voltage (V_{DD}) falls below the low-voltage detection level (V_{LVDL}). After INTLVI is generated, an interrupt request signal is not generated even if the supply voltage becomes equal to or higher than the high-voltage detection voltage (V_{LVDH}) without falling below the low-voltage detection voltage (V_{LVDL}).
To use the LVD reset & interrupt mode, perform the processing according to **Figure 26 - 9 Setting Procedure for Operating Voltage Check and Reset**.

Figures 26 - 7 and 26 - 8 show the timing of the internal reset signal and interrupt signal generated in the LVD interrupt & reset mode.

Figure 26 - 7 Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (1/2)



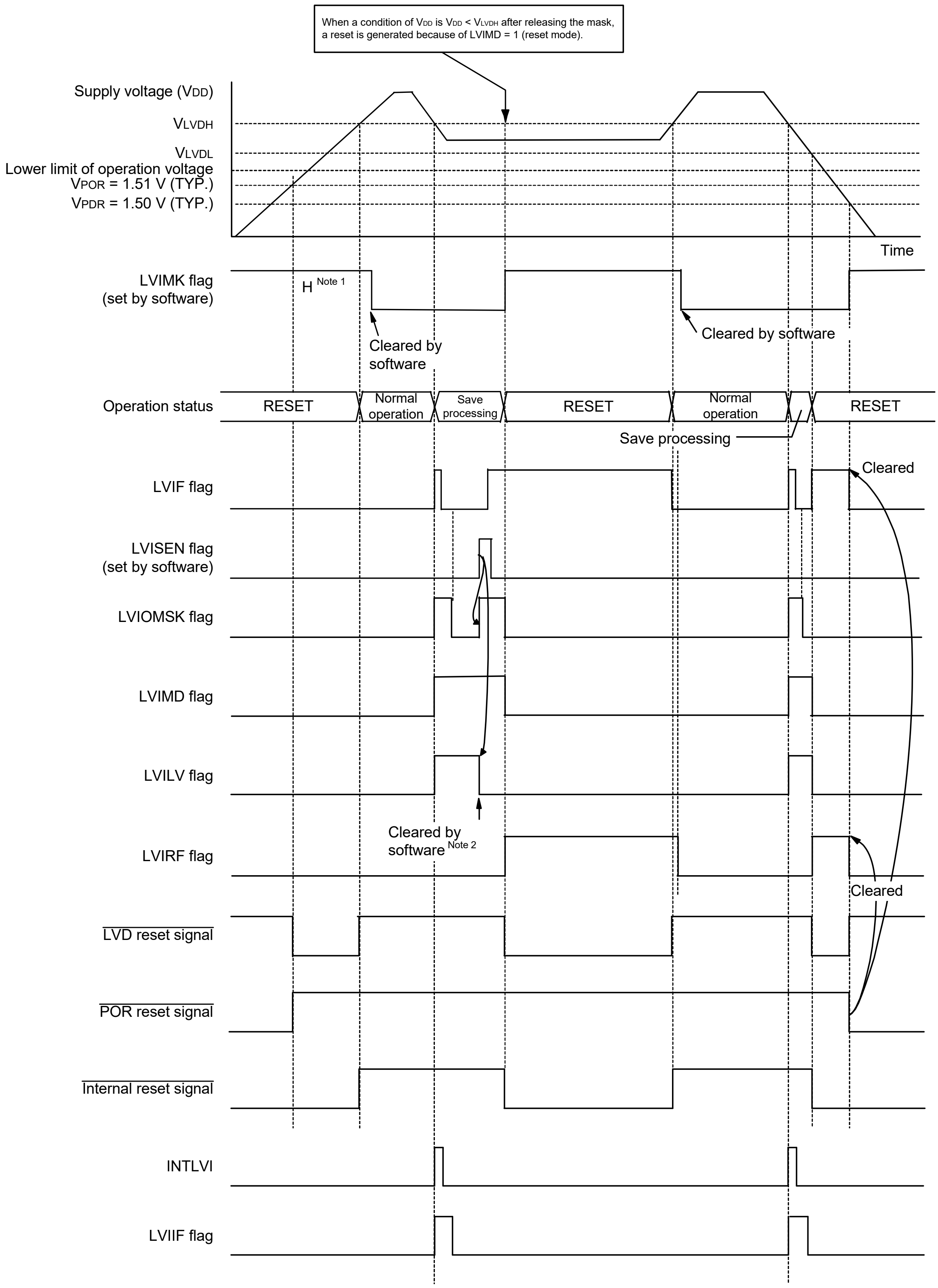
(**Notes** and **Remark** are listed on the next page.)

Note 1. The LVIMK flag is set to “1” by reset signal generation.

Note 2. After an interrupt is generated, perform the processing according to Figure 26 - 9 Setting Procedure for Operating Voltage Check and Reset in interrupt and reset mode.

Remark V_{POR}: POR power supply rise detection voltage
V_{POR}: POR power supply fall detection voltage

Figure 26 - 8 Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (2/2)



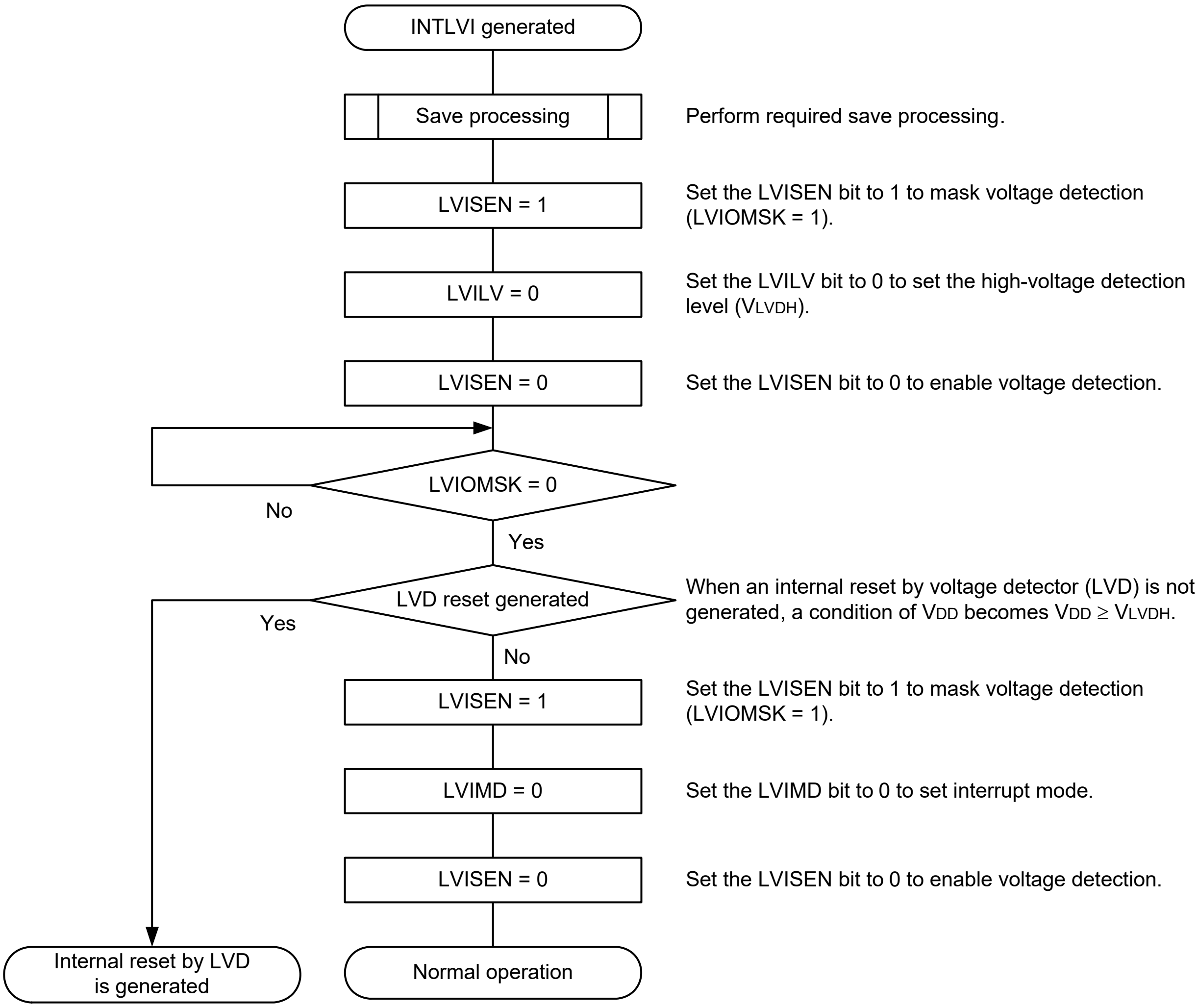
(**Notes** and **Remark** are listed on the next page.)

Note 1. The LVIMK flag is set to “1” by reset signal generation.

Note 2. After an interrupt is generated, perform the processing according to Figure 26 - 9 Setting Procedure for Operating Voltage Check and Reset in interrupt and reset mode.

Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Figure 26 - 9 Setting Procedure for Operating Voltage Check and Reset



26.5 Changing of LVD Detection Voltage Setting

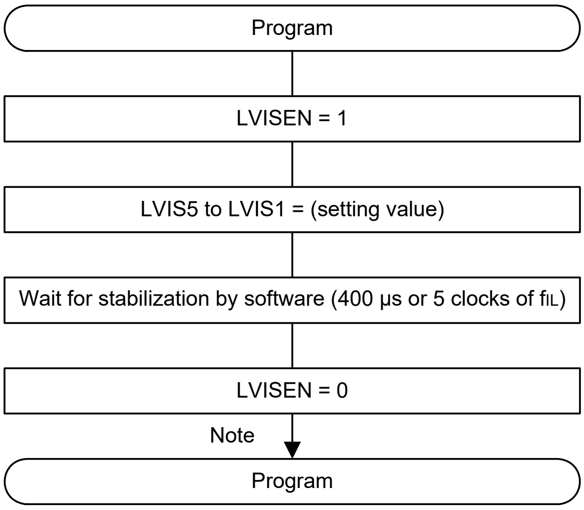
To change the LVD detection voltage by software, use the following procedure.

The LVD detection voltage can be changed in interrupt mode and reset mode.

In interrupt & reset mode, the value of the LVD detection voltage cannot be changed. Keep the initial value (set value in the option byte) unchanged.

To use two or more LVD detection voltages by changing LVISEL4 to LVISEL0 in the LVIS register by software, the highest voltage value of the used LVD detection voltages must be specified in the VPOC2 to VPOC0, LVIS1, and LVIS0 bits in the option byte (000C1H).

Figure 26 - 10 Changing of LVD Detection Voltage Setting

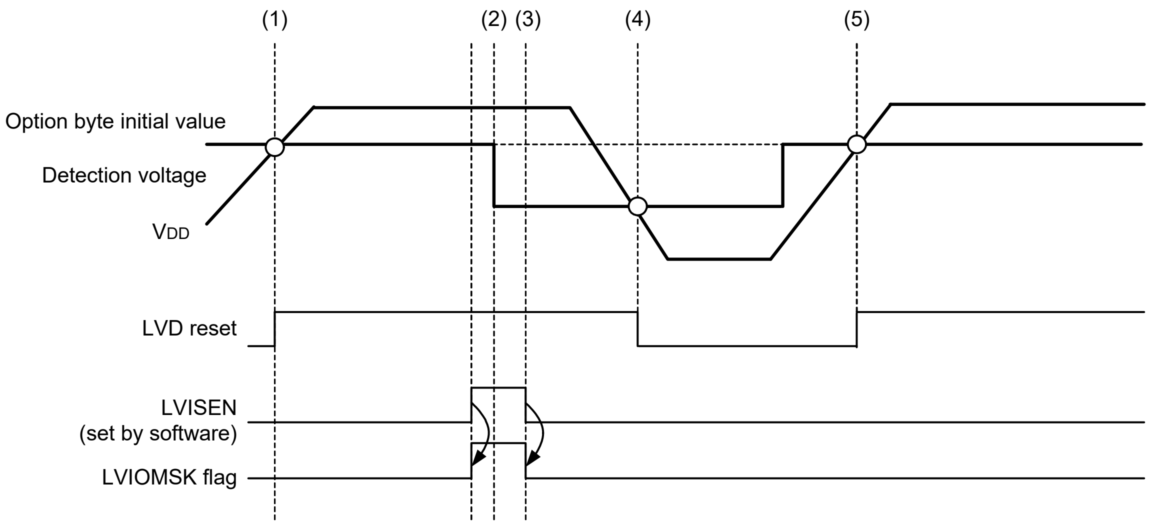


Note After LVISEN is set to 0, LVD is detected if $V_{LVD} > V_{DD}$, and a reset/interrupt is generated.

26.5.1 Changing of LVD detection voltage setting in LVD reset mode

Figure 26 - 11 shows an Example of Timing for Changing LVD Detection Voltage Setting in LVD Reset Mode.

Figure 26 - 11 Example of Timing for Changing LVD Detection Voltage Setting in LVD Reset Mode



Operation

(1) When the supply voltage rises, the detection voltage set by the option byte is used to release the reset.

(2) The value of the LVIS register is changed.

(3) Waiting for stabilization by software is completed (400 μs or five f_{IL} clock cycles after (2))

(4) At LVD detection (falling), the detection voltage set by the LVIS register

(5) At the LVD reset release (rising), the detection voltage set by the option byte

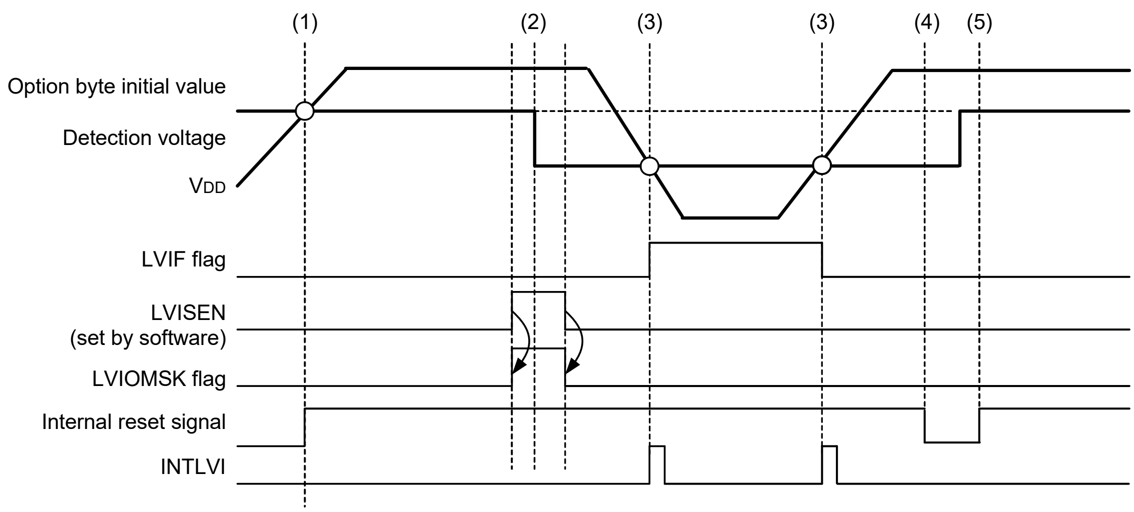
When changing the LVD detection voltage setting, note the following.

Caution The value of the reset release voltage in LVD reset mode is set to the set value in the option byte.

26.5.2 Changing of LVD detection voltage setting in LVD interrupt mode

Figure 26 - 12 shows an Example of Timing for Changing LVD Detection Voltage Setting in LVD Interrupt Mode.

Figure 26 - 12 Example of Timing for Changing LVD Detection Voltage Setting in LVD Interrupt Mode



Operation

(1) When the supply voltage rises, the detection voltage set by the option byte is used to release the reset.

(2) The value of the LVIS register is changed.

(3) At LVD detection (falling and rising), the detection voltage set by the LVIS register

(4) An internal reset is generated.

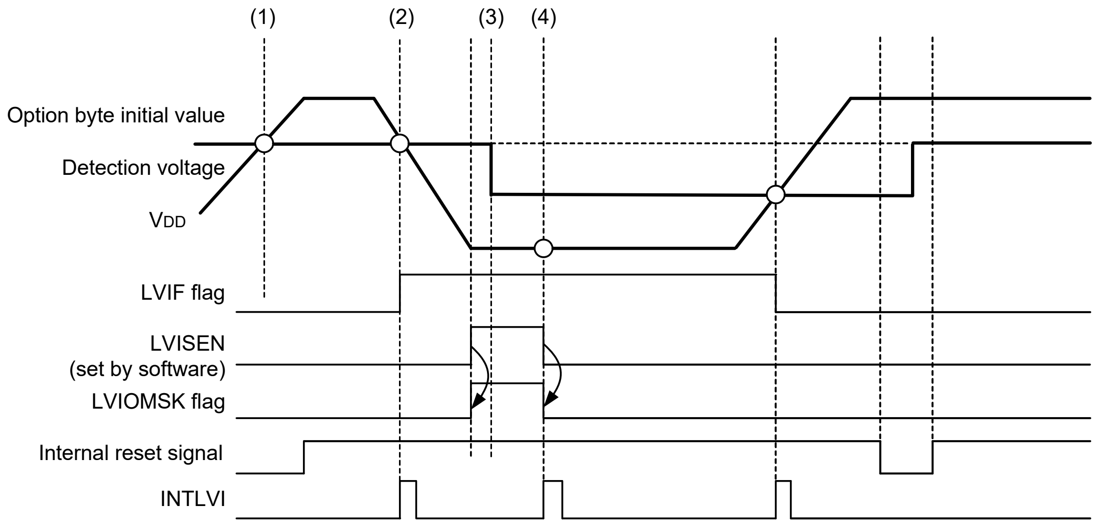
(5) The voltage value is changed to the set value in the option byte again when the internal reset is released.

When changing the LVD detection voltage setting, note the following.

Caution 1. Immediately after all resets are generated, the LVD internal reset retains its reset state until $V_{DD} \geq V_{LVD}$ (set value in the option byte). The LVD internal reset is released when $V_{DD} \geq V_{LVD}$ is detected (set value in the option byte).
After that, an interrupt request signal (INTLVI) is generated when $V_{DD} < V_{LVD}$ or $V_{DD} \geq V_{LVD}$ is detected.

Caution 2. If the LVD set voltage is changed by setting LVISEL4 to LVISEL0 in the LVIS register while $V_{DD} < V_{LVD}$, an LVD interrupt is generated when the masking is released (LVISEN = 0). See Figure 26 - 13.

Figure 26 - 13 Example of Timing for Changing LVD Detection Voltage Using LVDIS When V_{DD} < V_{LVD}



Operation

(1) When the supply voltage rises, the detection voltage set by the option byte is used to release the reset.

(2) At LVD detection (falling), the detection voltage set by the option byte

(3) The value of the LVIS register is changed.

(4) If V_{DD} < V_{LVD} at the same time the masking is released, an interrupt is generated.

26.6 Cautions for Voltage Detector

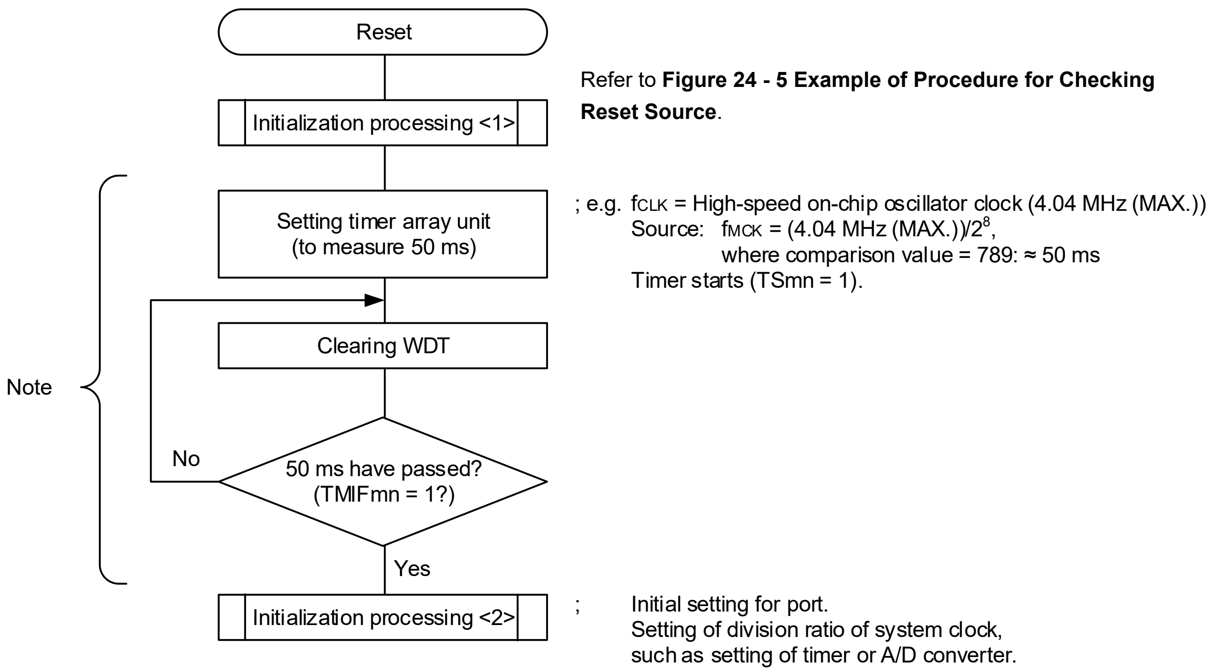
(1) Voltage fluctuation when power is supplied

In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the LVD detection voltage, the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking the following action.

<Action>

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports.

Figure 26 - 14 Example of Software Processing If Supply Voltage Fluctuation is 50 ms or Less in Vicinity of LVD Detection Voltage



Note If reset is generated again during this period, initialization processing <2> is not started.

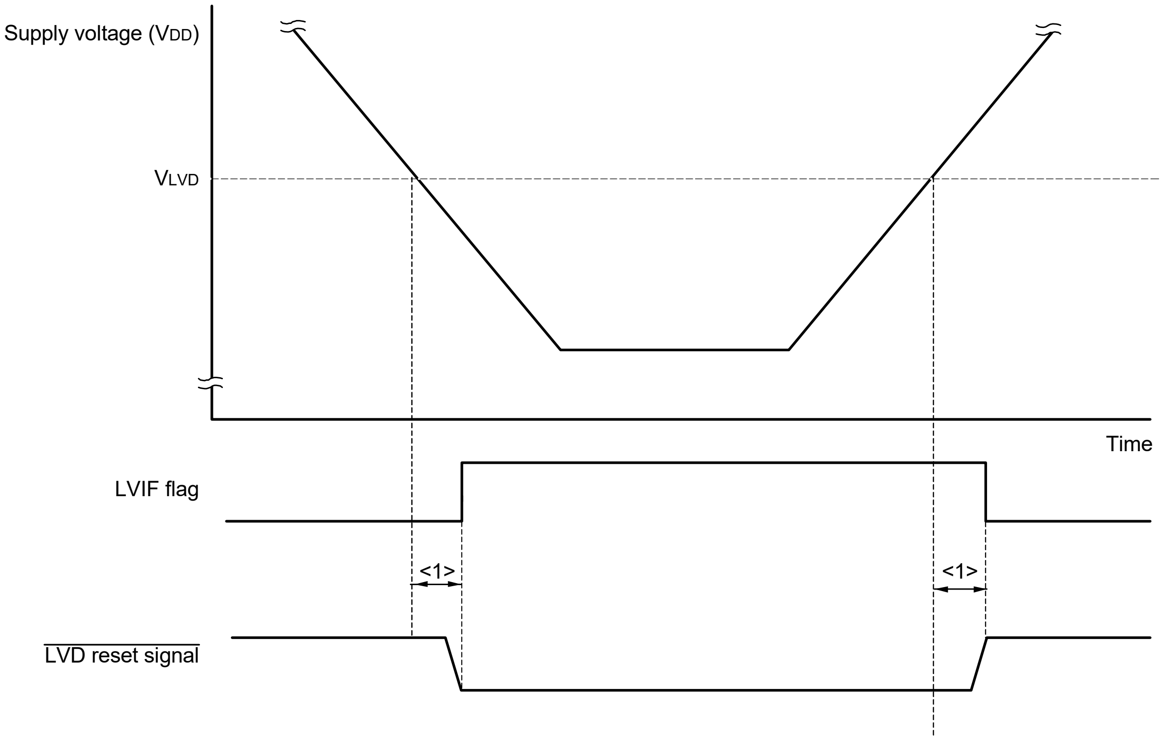
Remark m = 0
n = 0 to 3

(2) Delay from the time LVD reset source is generated until the time LVD reset has been generated or released

There is some delay from the time supply voltage (V_{DD}) < LVD detection voltage (V_{LVD}) until the time LVD reset has been generated.

In the same way, there is also some delay from the time LVD detection voltage (V_{LVD}) ≤ supply voltage (V_{DD}) until the time LVD reset has been released (see **Figure 26 - 15**).

Figure 26 - 15 Delay from the time LVD reset source is generated until the time LVD reset has been generated or released



<1>: Detection delay (300 μs (MAX.))

(3) Power on when LVD is off

Use the external rest input via the $\overline{\text{RESET}}$ pin when the LVD 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 **34.4 AC Characteristics**, and then input a high level to the pin.

(4) Operating voltage fall when LVD is off or LVD interrupt mode is selected

When the operating voltage falls with the LVD is off or with the LVD interrupt mode is selected, this LSI 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 **34.4 AC Characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

CHAPTER 27 SAFETY FUNCTIONS

27.1 Overview of Safety Functions

The following safety functions are provided in the RL78/I1D to comply with the IEC60730 and IEC61508 safety standards.

These functions enable the microcontroller to self-diagnose abnormalities and stop operating if an abnormality is detected.

(1) Flash memory CRC operation function (high-speed CRC, general-purpose CRC)
This detects data errors in the flash memory by performing CRC operations.
Two CRC functions are provided in the RL78/I1D that can be used according to the application or purpose of use.
- High-speed CRC: The CPU can be stopped and a high-speed check executed on its entire code flash memory area during the initialization routine.
- General CRC: This can be used for checking various data in addition to the code flash memory area while the CPU is running.

(2) RAM parity error detection function
This detects parity errors when the RAM is read as data.

(3) RAM guard function
This prevents RAM data from being rewritten when the CPU freezes.

(4) SFR guard function
This prevents SFRs from being rewritten when the CPU freezes.

(5) Invalid memory access detection function
This detects illegal accesses to invalid memory areas (such as areas where no memory is allocated and areas to which access is restricted).

(6) Frequency detection function
This uses the timer array unit to perform a self-check of the CPU/peripheral hardware clock frequency.

(7) A/D test function
This is used to perform a self-check of A/D converter by performing A/D conversion on the positive internal reference voltage, negative reference voltage, analog input channel (ANI), temperature sensor output, and internal reference voltage output.

(8) Digital output signal level detection function for I/O pins
When the I/O pins are output mode, the output level of the pin can be read.

Remark Refer to the IEC60730/60335 self-test library application notes (R01AN1062, R01AN1296) for the RL78 MCU Series, for more information on usage examples of the safety functions required to comply with the IEC60730 and IEC61508 safety standards.

27.2 Registers Used by Safety Functions

The safety functions use the following registers:

Register	Each Function of Safety Function
• Flash memory CRC control register (CRC0CTL) • Flash memory CRC operation result register (PGCRCL)	Flash memory CRC operation function (high-speed CRC)
• CRC input register (CRCIN) • CRC data register (CRCD)	CRC operation function (general-purpose CRC)
• RAM parity error control register (RPECTL)	RAM parity error detection function
• Invalid memory access detection control register (IAWCTL)	RAM guard function
	SFR guard function
	Invalid memory access detection function
• Timer input select register 0 (TIS0)	Frequency detection function
• A/D test register (ADTES)	A/D test function
• Port mode select register (PMS)	Digital output signal level detection function for I/O pins

The content of each register is described in 27.3 Operation of Safety Functions.

27.3 Operation of Safety Functions

27.3.1 Flash memory CRC operation function (high-speed CRC)

The IEC60730 standard mandates the checking of data in the flash memory, and recommends using CRC to do it. The high-speed CRC provided in the RL78/I1D 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, 341 μs@24 MHz with 32-KB 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 CRC, because the general 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.

Reset signal generation clears this register to 00H.

Figure 27 - 1 Format of Flash memory CRC control register (CRC0CTL)

Address: F02F0H After reset:00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CRC0CTL	CRC0EN	0	FEA5	FEA4	FEA3	FEA2	FEA1	FEA0

CRC0EN	Control of high-speed CRC ALU operation
0	Stop the operation.
1	Start the operation according to HALT instruction execution.

FEA5	FEA4	FEA3	FEA2	FEA1	FEA0	High-speed CRC operation range
0	0	0	0	0	0	00000H to 03FFBH (16 K - 4 bytes)
0	0	0	0	0	1	00000H to 07FFBH (32 K - 4 bytes)
Other than the above						Setting prohibited

Remark Input the expected CRC operation result value to be used for comparison in the lowest 4 bytes of the flash memory. Note that the operation range will thereby be reduced by 4 bytes.

27.3.1.2 Flash memory CRC operation result register (PGCRCL)

This register is used to store the high-speed CRC operation results.
The PGCRCL register can be set by a 16-bit memory manipulation instruction.
Reset signal generation clears this register to 0000H.

Figure 27 - 2 Format of Flash memory CRC operation result register (PGCRCL)

Address: F02F2H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
PGCRCL	PGCRC15	PGCRC14	PGCRC13	PGCRC12	PGCRC11	PGCRC10	PGCRC9	PGCRC8

	7	6	5	4	3	2	1	0
	PGCRC7	PGCRC6	PGCRC5	PGCRC4	PGCRC3	PGCRC2	PGCRC1	PGCRC0

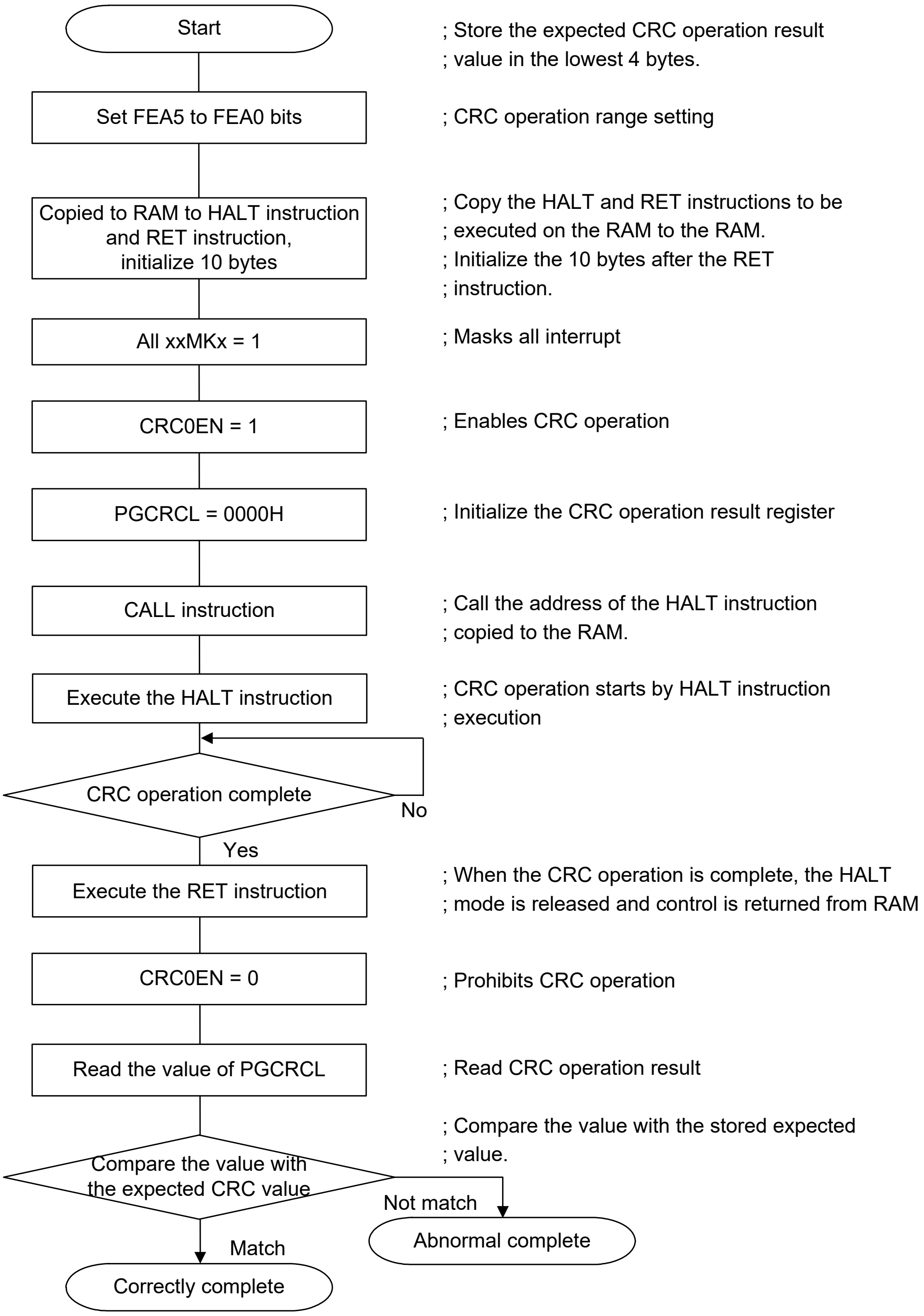
PGCRC15 to 0	High-speed CRC operation results
0000H to FFFFH	Store the high-speed CRC operation results.

Caution The PGCRCL register can only be written if CRC0EN (bit 7 of the CRC0CTL register) = 1.

Figure 27 - 3 shows the Flowchart of Flash Memory CRC Operation Function (High-speed CRC).

<Operation flow>

Figure 27 - 3 Flowchart of Flash Memory CRC Operation Function (High-speed CRC)



Caution 1. The CRC operation is executed only on the code flash.

Caution 2. Store the expected CRC operation value in the area below the operation range in the code flash.

Caution 3. The CRC operation is enabled by executing the HALT instruction in the RAM area. Be sure to execute the HALT instruction in RAM area.

The expected CRC operation value can be calculated by using the integrated development environment CubeSuite+ development environment. Refer to the CubeSuite+ integrated development environment user's manual for details.

27.3.2 CRC operation function (general-purpose CRC)

In order to guarantee safety during operation, the IEC61508 standard mandates the checking of data even while the CPU is operating.

In the RL78/I1D, a general CRC operation can be executed as a peripheral function while the CPU is operating. The general CRC can be used for checking various data in addition to the code flash memory area. The data to be checked can be specified by using software (a user-created program). In HALT mode, the CRC operation function can be used only during DTC transfer.

The general CRC operation can be executed in the main system clock operation mode as well as the subsystem clock operation mode.

The CRC generator polynomial used is "$X^{16} + X^{12} + X^5 + 1$" of CRC-16-CCITT. The data to be input is inverted in bit order and then calculated to allow for LSB-first communication. For example, if the data 12345678H is sent from the LSB, values are written to the CRCIN register in the order of 78H, 56H, 34H, and 12H, enabling a value of 08F6H to be obtained from the CRCD register. This is the result obtained by executing a CRC operation on the bit rows shown below, which consist of the data 12345678H inverted in bit order.

CRCIN setting data	78H	56H	34H	12H	
Bit representation data	0111 1000	0101 0110	0011 0100	0001 0010	
					Bit reverse
Bit reverse data	0001 1110	0110 1010	0010 1100	0100 1000	
					Operation with polynomial
Result data	0110 1111	0001 0000			
					Bit reverse
CRCD data	0000 1000 (08F6H)	1111 0110			←Obtained result

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)

CRCIN register is an 8-bit register that is used to set the CRC operation data of general-purpose CRC.
The possible setting range is 00H to FFH.
The CRCIN register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 27 - 4 Format of CRC input register (CRCIN)

Address:FFFACH After reset:00H R/W

Symbol	7	6	5	4	3	2	1	0
CRCIN								

Bits 7 to 0	Function
00H to FFH	Data input.

27.3.2.2 CRC data register (CRCD)

This register is used to store the general-purpose CRC operation result.

The possible setting range is 0000H to FFFFH.

After 1 clock of CPU/peripheral hardware clock (f_{CLK}) has elapsed from the time CRCIN register is written, the CRC operation result is stored to the CRCD register.

The CRCD register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 27 - 5 Format of CRC data register (CRCD)

Address: F02FAH After reset: 0000H R/W

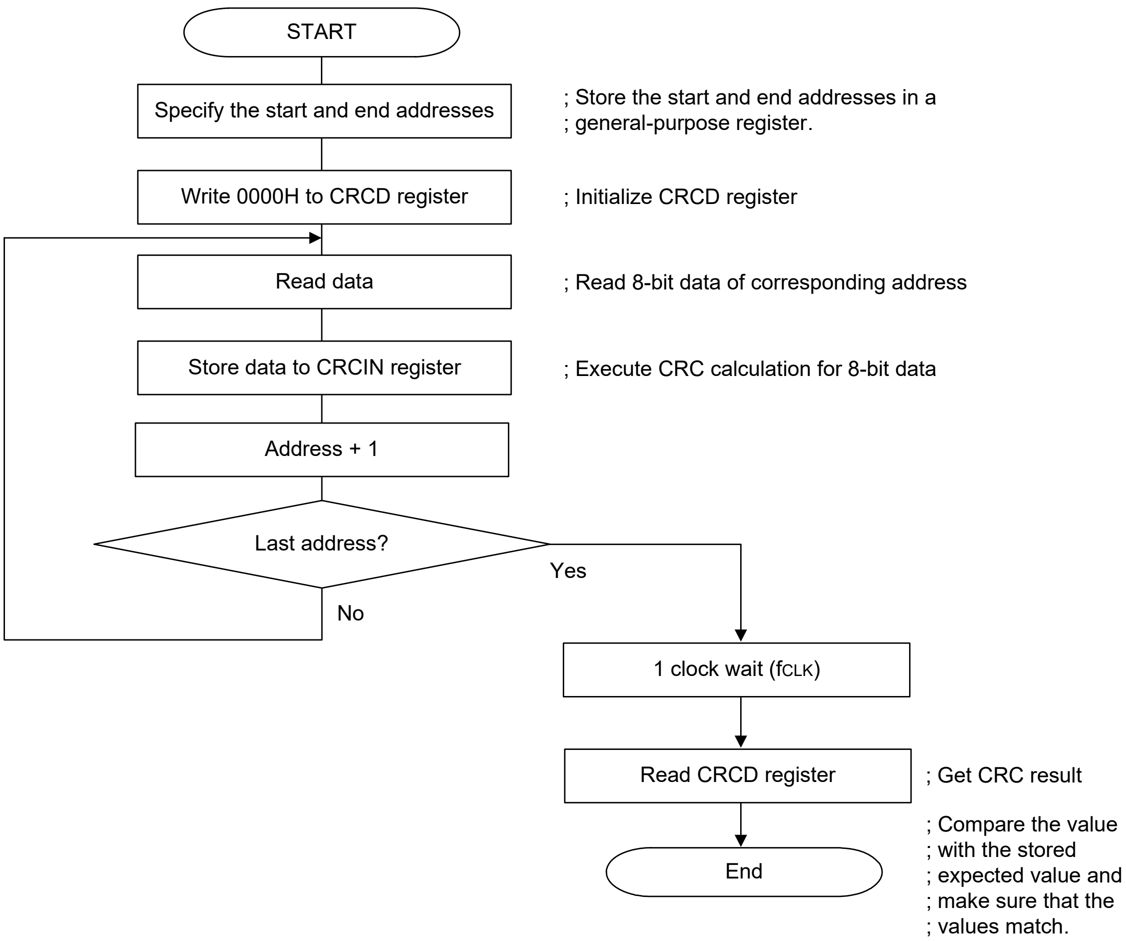
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCD																

Caution 1. Read the value written to CRCD register before writing to CRCIN register.

Caution 2. If writing and storing operation result to CRCD register conflict, the writing is ignored.

<Operation flow>

Figure 27 - 6 CRC Operation Function (General-Purpose CRC)



27.3.3 RAM parity error detection function

The IEC60730 standard mandates the checking of RAM data. A single-bit parity bit is therefore added to all 8-bit data in the RL78/I1D's RAM. By using this RAM parity error detection function, the parity bit is appended when data is written, and the parity is checked when the data is read. This function can also be used to trigger a reset when a parity error occurs.

27.3.3.1 RAM parity error control register (RPECTL)

This register is used to control parity error generation check bit and reset generation due to parity errors.
The RPECTL register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 27 - 7 Format of RAM parity error control register (RPECTL)

Address: F00F5H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
RPECTL	RPERDIS	0	0	0	0	0	0	RPEF

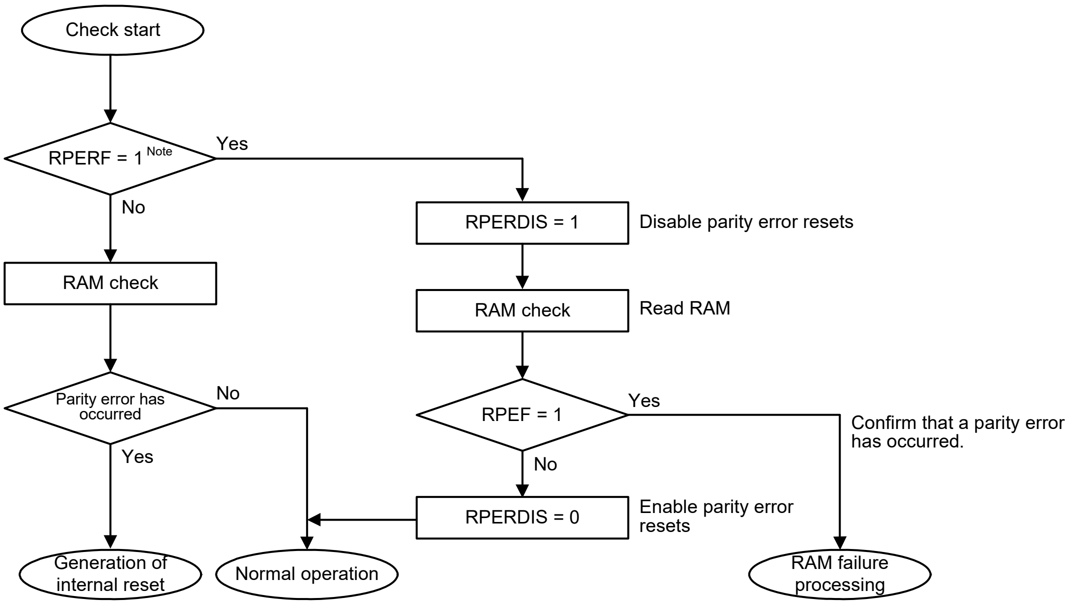
RPERDIS	Parity error reset mask flag
0	Enable parity error resets.
1	Disable parity error resets.

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 is written, and the parity is checked when the data is read. Therefore, while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed before reading data.**
The RL78's CPU executes look-ahead due to the pipeline operation, the CPU might read an uninitialized RAM area that is allocated beyond the RAM used, which causes a RAM parity error. Therefore, while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the RAM area + 10 bytes when instructions are fetched from RAM areas.

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 (1) if a parity error occurs. If the parity error reset is enabled (RPERDIS = 0) while RPEF = 1, a parity error reset occurs when RPERDIS is cleared (0).
Remark 3. The RPECTL flag in the RESF register is set (1) by RAM parity errors and cleared (0) by writing 0 to it or by any reset source. When RPEF = 1, the value is retained even if RAM for which no parity error has occurred is read.
Remark 4. General-purpose registers are not included in the range of RAM parity error detection.

Figure 27 - 8 RAM Parity Error Check Flow



Note See **CHAPTER 24 RESET FUNCTION** for details on how to confirm internal resets due to RAM parity errors.

27.3.4 RAM guard function

In order to guarantee safety during operation, the IEC61508 standard mandates that important data stored in the RAM be protected, even if the CPU freezes.

This RAM guard function is used to protect data in the specified memory space.

If the RAM guard function is specified, writing to the specified RAM space is disabled, but reading from the space can be carried out as usual.

27.3.4.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.

GRAM1 and GRAM0 bits are used in RAM guard function.

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

Reset signal generation clears this register to 00H.

Figure 27 - 9 Format of Invalid memory access detection control register (IAWCTL)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GRAM1	GRAM0	RAM guard space Note
0	0	Disabled. RAM can be written to.
0	1	The 128 bytes starting at the start RAM address
1	0	The 256 bytes starting at the start RAM address
1	1	The 512 bytes starting at the start RAM address

Note The RAM start address differs depending on the size of the RAM provided with the product.

27.3.5 SFR guard function

In order to guarantee safety during operation, the IEC61508 standard mandates that important SFRs be protected from being overwritten, even if the CPU freezes.

This SFR guard function is used to protect data in the control registers used by the port function, interrupt function, clock control function, voltage detection function, and RAM parity error detection function.

If the SFR guard function is specified, writing to the specified SFRs is disabled, but reading from the SFRs can be carried out as usual.

27.3.5.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.

GPORT, GINT and GCSC bits are used in SFR guard function.

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

Reset signal generation clears this register to 00H.

Figure 27 - 10 Format of Invalid memory access detection control register (IAWCTL)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GPORT	Control registers of port function guard
0	Disabled. Control registers of port function can be read or written to.
1	Enabled. Writing to control registers of port function is disabled. Reading is enabled. [Guarded SFR] PMxx, PUxx, PIMxx, POMxx, PMCxx, ADPC, PIORx Note

GINT	Registers of interrupt function guard
0	Disabled. Registers of interrupt function can be read or written to.
1	Enabled. Writing to registers of interrupt function is disabled. Reading is enabled. [Guarded SFR] IFxx, MKxx, PRxx, EGPx, EGNx

GCSC	Control registers of clock control function, voltage detector, and RAM parity error detection function guard
0	Disabled. Control registers of clock control function, voltage detector and RAM parity error detection function can be read or written to.
1	Enabled. Writing to control registers of clock control function, voltage detector and RAM parity error detection function is disabled. Reading is enabled. [Guarded SFR] CMC, CSC, OSTS, CKC, CKSEL, PERx, PRRx, OSMC, LVIM, LVIS, RPECTL, PMMC, MOCODIV, FMCKS

Note Pxx (Port register) is not guarded.

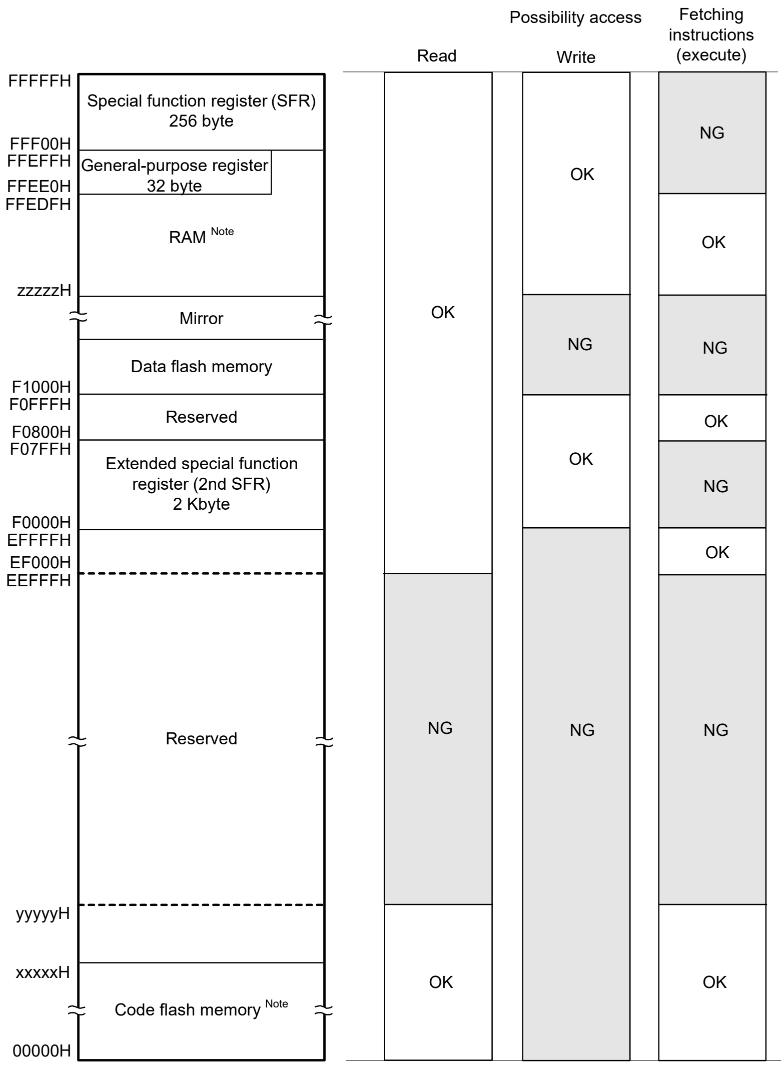
27.3.6 Invalid memory access detection function

The IEC60730 standard mandates checking that the CPU and interrupts are operating correctly.

The illegal memory access detection function triggers a reset if a memory space specified as access-prohibited is accessed.

The illegal memory access detection function applies to the areas indicated by NG in Figure 27 - 11.

Figure 27 - 11 Invalid access detection area



Note The code flash memory, RAM, and lowest detection address of each product are as follows.

Products	Code Flash Memory (00000H to xxxxxH)	RAM (zzzzzH to FFEFFH)	Lowest Detection Address (yyyyyH) when Reading/Fetching (Executing) Instructions
R5F117x8 (x = 6, 7, A)	8192 × 8 bits (00000H to 01FFFH)	768 × 8 bits (FFC00H to FFEFFH)	10000H
R5F117xA (x = 6, 7, A, B, G)	16384 × 8 bits (00000H to 03FFFH)	2048 × 8 bits (FF700H to FFEFFH)	10000H
R5F117xC (x = A, B, G)	32768 × 8 bits (00000H to 07FFFH)	3072 × 8 bits (FF300H to FFEFFH)	10000H

27.3.6.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.
IAWEN bit is used in invalid memory access detection function.
The IAWCTL register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 27 - 12 Format of Invalid memory access detection control register (IAWCTL)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN Note	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

IAWEN Note	Control of invalid memory access detection
0	Disable the detection of invalid memory access.
1	Enable the detection of invalid memory access.

Note Only writing 1 to the IAWEN bit is enabled, not writing 0 to it after setting it to 1.

Remark By specifying WDTON = 1 for the option byte (watchdog timer operation enable), the invalid memory access detection function is enabled even if IAWEN = 0.

27.3.7 Frequency detection function

The IEC60730 standard mandates checking that the oscillation frequency is correct.

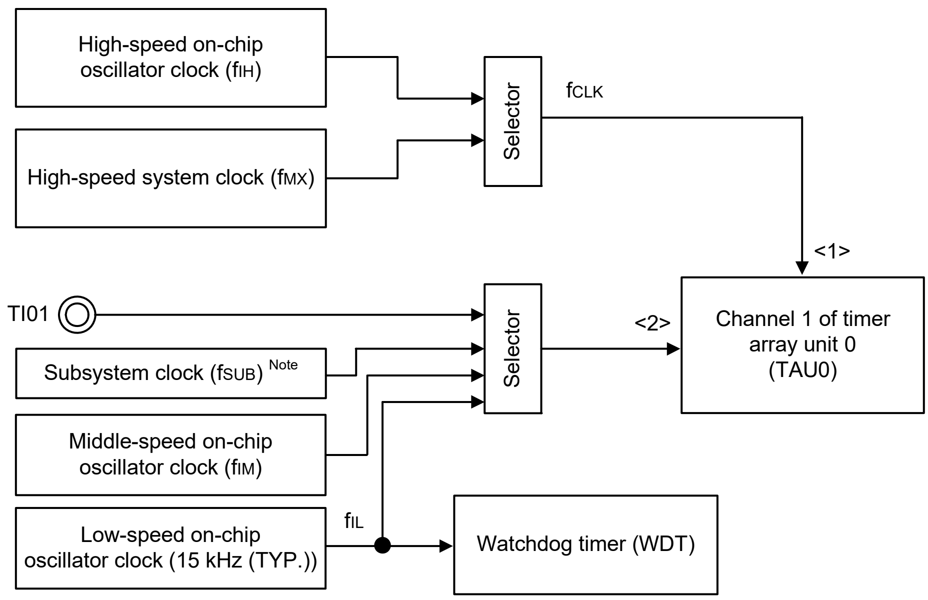
By using the CPU/peripheral hardware clock frequency (f_{CLK}) and measuring the pulse width of the input signal to channel 1 of the timer array unit 0 (TAU0), whether the proportional relationship between the two clock frequencies is correct can be determined. Note that, however, if one or both clock operations are completely stopped, the proportional relationship between the clocks cannot be determined.

<Clocks 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 1 of the timer array unit 0
- Timer input to channel 1 (TI01)
- Low-speed on-chip oscillator clock (f_{IL}: 15 kHz (typ.))
- Middle-speed on-chip oscillator clock (f_{IM})
- Subsystem clock (f_{SUB}) Note

Figure 27 - 13 Configuration of Frequency Detection Function



If pulse interval measurement results in an abnormal value, it can be concluded that the clock frequency is abnormal.

For how to execute pulse interval measurement, see **7.8.4 Operation as input pulse interval measurement**.

Note Can only be selected in the products incorporating the subsystem clock.

27.3.7.1 Timer input select register 0 (TIS0)

The TIS0 register is used to select the timer input of channels 0 and 1 of the timer array unit 0 (TAU0).

The TIS0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 27 - 14 Format of Timer input select register 0 (TIS0)

Address: F0074H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	TIS04	0	TIS02	TIS01	TIS00

TIS04	Selection of timer input used with channel 0
0	Input signal of timer input pin (TI00)
1	Event input signal from ELC

TIS02	TIS01	TIS00	Selection of timer input used with channel 1
0	0	0	Input signal of timer input pin (TI01)
0	0	1	Event input signal from ELC
0	1	0	Input signal of timer input pin (TI01)
0	1	1	Middle-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 above			Setting prohibited

27.3.8 A/D test function

The IEC60730 standard mandates testing the A/D converter. The A/D test function is used to check whether the A/D converter is operating normally by executing A/D conversions of the positive reference voltage and negative reference voltage of the A/D converter, analog input channel (ANI), temperature sensor output voltage, and internal reference voltage. For details on the checking method, refer to the safety function (A/D test) application note (R01AN0955).

The analog multiplexer can be checked using the following procedure.

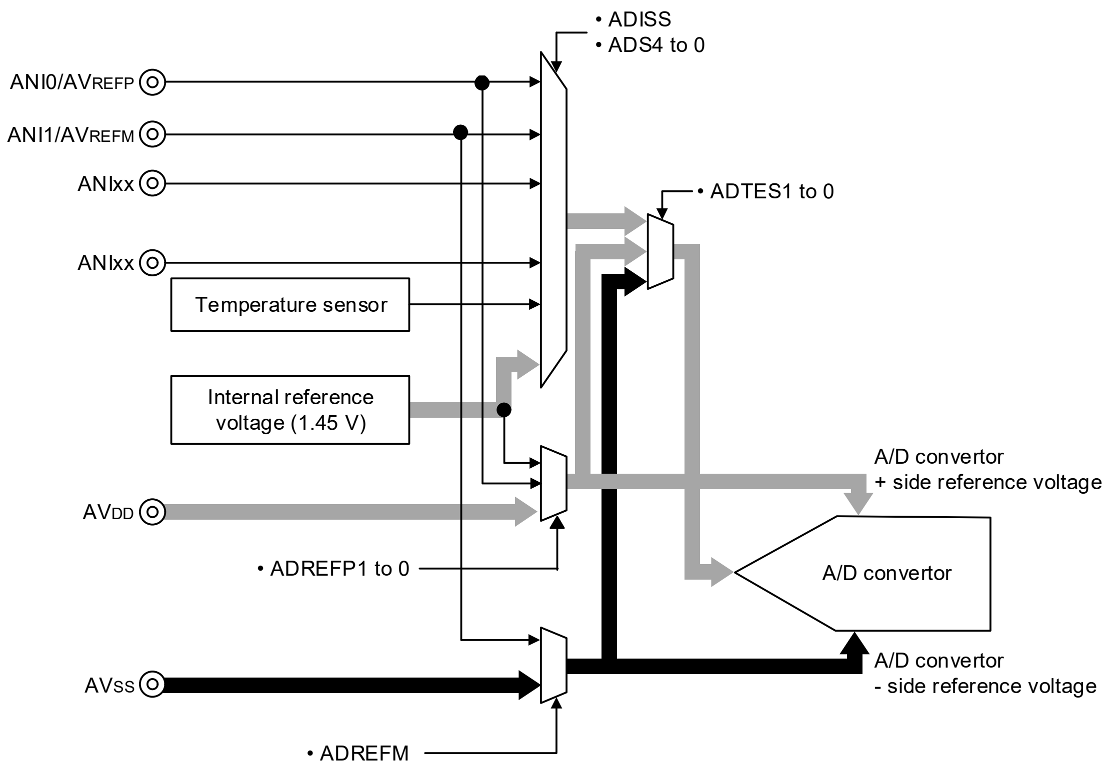
(1) Select the ANIx pin as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 0, 0).
(2) Perform A/D conversion for the ANIx pin (conversion result 1-1).
(3) Select the negative reference voltage of the A/D converter as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 1, 0).
(4) Perform A/D conversion of the negative reference voltage of the A/D converter (conversion result 2-1).
(5) Select the ANIx pin as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 0, 0).
(6) Perform A/D conversion for the ANIx pin (conversion result 1-2).
(7) Select the positive reference voltage of the A/D converter as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 1, 1).
(8) Perform A/D conversion of the positive reference voltage of the A/D converter (conversion result 2-2).
(9) Select the ANIx pin as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 0, 0).
(10) Perform A/D conversion for the ANIx pin (conversion result 1-3).
(11) Make sure that “conversion result 1-1” = “conversion result 1-2” = “conversion result 1-3”.
(12) Make sure that the A/D conversion results of “conversion result 2-1” are all 0 and those of “conversion result 2-2” are all 1.

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 conversion in steps (1) to (10) above, use another method to check the analog multiplexer.

Remark 2. The conversion results might contain an error. Consider an appropriate level of error when comparing the conversion results.

Figure 27 - 15 Configuration of A/D Test Function



27.3.8.1 A/D test register (ADTES)

This register is used to select the A/D converter positive reference voltage, negative reference voltage, analog input channel (ANIxx), temperature sensor output voltage, or internal reference voltage (1.45 V) as the target of A/D conversion.

When using the A/D test function, specify the following settings:

• Select the negative reference voltage as the target of A/D conversion when measuring the zero-scale.

• Select the positive reference voltage as the target of A/D conversion when measuring the full-scale.

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

Reset signal generation clears this register to 00H.

Figure 27 - 16 Format of A/D test register (ADTES)

Address: F0013H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	0	0	ADTES1	ADTES0

ADTES1	ADTES0	A/D conversion target
0	0	ANIxx/temperature sensor output/internal reference voltage (1.45 V) (This is specified using 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 and ADREFP0 bits in the ADM2 register)
Other than the above		Setting prohibited

27.3.8.2 Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted.
Set A/D test register (ADTES) to 00H when measuring the ANIxx/temperature sensor output/internal reference voltage (1.45 V).
The ADS register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 27 - 17 Format of Analog input channel specification register (ADS)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

○ Select mode (ADMD = 0)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P10/ANI0/AVREFP pin
0	0	0	0	0	1	ANI1	P11/ANI1/AVREFM pin
0	0	0	0	1	0	ANI2	P12/ANI2 pin
0	0	0	0	1	1	ANI3	P13/ANI3 pin
0	0	0	1	0	0	ANI4	P14/ANI4 pin
0	0	0	1	0	1	ANI5	P15/ANI5 pin
0	0	0	1	1	0	ANI6	P16/ANI6 pin
0	0	0	1	1	1	ANI7	P17/ANI7 pin
0	0	1	0	0	0	ANI8	P25/ANI8 pin
0	0	1	0	0	1	ANI9	P24/ANI9 pin
0	0	1	0	1	0	ANI10	P23/ANI10 pin
0	0	1	0	1	1	ANI11	P22/ANI11 pin
0	0	1	1	0	0	ANI12	P21/ANI12 pin
0	0	1	1	0	1	ANI13	P20/ANI13 pin
0	1	0	0	0	0	ANI16	P02/ANI16 pin
0	1	0	0	0	1	ANI17	P03/ANI17 pin
0	1	0	0	1	0	ANI18	P04/ANI18 pin
1	0	0	0	0	0	—	Temperature sensor output
1	0	0	0	0	1	—	Internal reference voltage output (1.45 V)
Other than the above						Setting prohibited	

Caution 1. Be sure to clear bits 5 and 6 to 0.

Caution 2. For ports that set to analog input using the PMC register, select input mode using port mode register 0 to 2 (PM0 to PM2).

Caution 3. Do not use the ADS register to set ports that to be set as digital I/O using port mode control register 0, 10, 12, or 14 (PMC0, PMC10, PMC12, PMC14).

Caution 4. Only rewrite the value of the ADISS bit while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 5. When using AVREFP as the positive reference voltage of the A/D converter, do not select ANI0 as an A/D conversion channel.

Caution 6. When using AVREFM as the negative reference voltage of the A/D converter, do not select ANI1 as an A/D conversion channel.

Caution 7. If ADISS is set to 1, the internal reference voltage output (1.45 V) cannot be used for the positive reference voltage. Also, the first conversion result cannot be used after ADISS is set to 1. For details on the setup flow, see 14.7.4 Setup when temperature sensor output voltage/internal reference voltage is selected (example for software trigger mode and one-shot conversion mode).

Caution 8. Do not set ADISS to 1 when entering HALT mode while in STOP mode or while the CPU operates on the subsystem clock. With ADISS = 1, the current value of the A/D converter reference voltage current (IADREF) listed in 34.3.2 Supply current characteristics is added.

27.3.9 Digital output signal level detection function for I/O pins

In the IEC60730, it is required to check that the I/O function correctly operates.

By using the digital output signal level detection function for I/O pins, the digital output level of the pin can be read when the port is set to output mode.

27.3.9.1 Port mode select register (PMS)

This register is used to select the output level from output latch level or pin output level when the pin is output mode in which PMm bit of port mode register (PMm) is 0.

This register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 27 - 18 Format of Port mode select register (PMS)

Address: F007BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PMS	0	0	0	0	0	0	0	PMS0

PMS0	Method for selecting output level to be read when pin is output mode
0	Pmn register value is read.
1	Digital output level of the pin is read.

Caution 1. While the PMS0 bit in the PMS register is set to 1, do not change the value of the port register (Pxx) using a bit manipulation instruction. To change the value of the port register (Pxx), use an 8-bit data manipulation instruction.

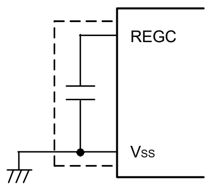
Caution 2. When P60 to P63 are used as general-purpose ports, the pin output level cannot read using PMS0. The read value is 0.

Remark m = 0 to 6, 10 to 12, 13
n = 0 to 7

CHAPTER 28 REGULATOR

28.1 Regulator Overview

The RL78/I1D contains a circuit for operating the device with a constant voltage. At this time, in order to stabilize the regulator output voltage, connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.



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

The regulator output voltage, see **Table 28 - 1**.

Table 28 - 1 Regulator Output Voltage Conditions

Mode	Output Voltage	Condition
LV (low-voltage main) mode	1.8 V	—
LP (Low-power main) mode		
LS (low-speed main) mode		
HS (high-speed main) mode	1.8 V	In STOP mode
		When both the high-speed system clock (f_{SUB}) and the high-speed on-chip oscillator clock (f_{IH}) are stopped during CPU operation with the subsystem clock (f_{XT})
		When both the high-speed system clock (f_{SUB}) and the high-speed on-chip oscillator clock (f_{IH}) are stopped during the HALT mode when the CPU operation with the subsystem clock (f_{XT}) has been set
	2.1 V	Other than above (include during OCD mode) Note

Note When it shifts to the subsystem clock operation or STOP mode during the on-chip debugging, the regulator output voltage is kept at 2.1 V (not decline to 1.8 V).

28.2 Register Controlling Regulator

The following register is used to control the regulator.

• Regulator mode control register (PMMC)

28.2.1 Regulator mode control register (PMMC)

The PMMC register is an 8-bit register used to control the mode of the on-chip regulator.
This register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset generation sets this register to 00H.

Figure 28 - 1 Format of Regulator mode control register (PMMC)

Address: F00F8H After reset: 00H R/W

Symbol	7	<6>	5	4	3	2	1	0
PMMC	0	MCSEL	0	0	0	0	0	0

MCSEL	Control of regulator mode
0	Normal setting
1	Low-power consumption setting

Caution 1. Do not change the flash operation mode select register (FLMODE) when MCSEL is 1.
Caution 2. Do not set MCSEL to 1 in HS (high-speed main) mode and LV (low-voltage main) mode.
Caution 3. In LS (low-speed main) mode, transitions to the STOP mode are prohibited while MCSEL is 1.

CHAPTER 29 OPTION BYTE

29.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory of the RL78/I1D form an option byte area.
Option bytes consist of user option byte (000C0H to 000C2H) and on-chip debug option byte (000C3H).
Upon power application or resetting and starting, an option byte is automatically referenced and a specified function is set. When using the product, be sure to set the following functions by using the option bytes.
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 010C0H to 010C3H. Therefore, set the same values as 000C0H to 000C3H to 010C0H to 010C3H.

Remark The option bytes should always be set regardless of whether each function is used.

29.1.1 User option byte (000C0H to 000C2H/010C0H to 010C2H)

(1) 000C0H/010C0H

○ Setting of watchdog timer operation
- Enabling or disabling of counter operation
- Enabling or disabling of counter operation in the HALT or STOP mode

○ Setting of interval 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 Set the same value as 000C0H to 010C0H when the boot swap operation is used because 000C0H is replaced by 010C0H.

(2) 000C1H/010C1H

○ Setting of LVD operation mode
- Interrupt & reset mode
- Reset mode
- Interrupt mode
- LVD off (external reset input from the $\overline{\text{RESET}}$ pin is used)

○ Setting of LVD detection level (V_{LVDH}, V_{LVDL}, V_{LVD})

Caution 1. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 34.4 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 LSI 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 010C2H).

Caution 2. Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

(3) 000C2H/010C2H

○ Setting of flash operation mode

Make the setting depending on the main system clock frequency (f_{MAIN}) and power supply voltage (V_{DD}) to be used.

• LV (low-voltage main) mode
• LS (low-speed main) mode
• HS (high-speed main) mode

○ Setting of the frequency of the high-speed on-chip oscillator

• Select from 1 MHz to 24 MHz.

Caution Set the same value as 000C2H to 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.

29.1.2 On-chip debug option byte (000C3H/010C3H)

○ 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 Set the same value as 000C3H to 010C3H when the boot swap operation is used because 000C3H is replaced by 010C3H.

29.2 Format of User Option Byte

The format of user option byte is shown below.

Figure 29 - 1 Format of User Option Byte (000C0H/010C0H)

Address: 000C0H/010C0H Note 1

7	6	5	4	3	2	1	0
WDTINT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTINT	Use of interval interrupt of watchdog timer
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% of the overflow time + 1/2 f_{IL} is reached.

WINDOW1	WINDOW0	Watchdog timer window open period Note 2
0	0	Setting prohibited
0	1	50%
1	0	75% Note 3
1	1	100%

WDTON	Operation control of watchdog timer counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time (f_{IL} = 17.25 kHz (MAX.))
0	0	0	$2^6/f_{IL}$ (3.71 ms)
0	0	1	$2^7/f_{IL}$ (7.42 ms)
0	1	0	$2^8/f_{IL}$ (14.84 ms)
0	1	1	$2^9/f_{IL}$ (29.68 ms)
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)
1	0	1	$2^{13}/f_{IL}$ (474.90 ms)
1	1	0	$2^{14}/f_{IL}$ (949.80 ms)
1	1	1	$2^{16}/f_{IL}$ (3799.19 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. Set the same value as 000C0H to 010C0H when the boot swap operation is used because 000C0H is replaced by 010C0H.

Note 2. The window open period is 100% when WDSTBYON = 0, regardless the value of the WINDOW1 and WINDOW0 bits.

(**Note** continues on the next page.)

Note 3. When the window open period is set to 75%, clearing the counter of the watchdog timer (writing ACH to WDTE) must proceed outside the corresponding period from among those listed below, over which clearing of the counter is prohibited (for example, confirming that the interval timer interrupt request flag (WDTIIF) of the watchdog timer is set).

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time (f_{IL} = 17.25 kHz (MAX.))	Period over which clearing the counter is prohibited when the window open period is set to 75%
0	0	0	$2^6/f_{IL}$ (3.71 ms)	1.85 ms to 2.51 ms
0	0	1	$2^7/f_{IL}$ (7.42 ms)	3.71 ms to 5.02 ms
0	1	0	$2^8/f_{IL}$ (14.84 ms)	7.42 ms to 10.04 ms
0	1	1	$2^9/f_{IL}$ (29.68 ms)	14.84 ms to 20.08 ms
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)	56.36 ms to 80.32 ms
1	0	1	$2^{13}/f_{IL}$ (474.90 ms)	237.44 ms to 321.26 ms
1	1	0	$2^{14}/f_{IL}$ (949.80 ms)	474.89 ms to 642.51 ms
1	1	1	$2^{16}/f_{IL}$ (3799.19 ms)	1899.59 ms to 2570.04 ms

Remark f_{IL}: Low-speed on-chip oscillator clock frequency

Figure 29 - 2 Format of User Option Byte (000C1H/010C1H) (1/4)

Address: 000C1H/010C1H Note

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD setting (interrupt & reset mode)

Detection Voltage			Option Byte Setting Value						
VLVDH		VLVDL	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge	Falling edge						LVIMDS1	LVIMDS0
1.77 V	1.73 V	1.63 V	0	0	0	1	0	1	0
1.88 V	1.84 V					0	1		
2.92 V	2.86 V					0	0		
1.98 V	1.94 V	1.84 V		0	1	1	0		
2.09 V	2.04 V					0	1		
3.13 V	3.06 V					0	0		
2.61 V	2.55 V	2.45 V		1	0	1	0		
2.71 V	2.65 V					0	1		
2.92 V	2.86 V	2.75 V		1	1	1	0		
3.02 V	2.96 V					0	1		
—			Settings other than the above are prohibited						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Caution Be sure to set bit 4 to “1”.

Remark 1. For details on the LVD circuit, see **CHAPTER 26 VOLTAGE DETECTOR**.

Remark 2. The detection voltage is a typical value. For details, see **34.6.6 LVD circuit characteristics**.

Figure 29 - 3 Format of User Option Byte (000C1H/010C1H) (2/4)

Address: 000C1H/010C1H Note

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD setting (reset mode)

Detection Voltage		Option Byte Setting Value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
1.67 V	1.63 V	0	0	0	1	1	1	1
1.77 V	1.73 V		0	0	1	0		
1.88 V	1.84 V		0	1	1	1		
1.98 V	1.94 V		0	1	1	0		
2.09 V	2.04 V		0	1	0	1		
2.50 V	2.45 V		1	0	1	1		
2.61 V	2.55 V		1	0	1	0		
2.71 V	2.65 V		1	0	0	1		
2.81 V	2.75 V		1	1	1	1		
2.92 V	2.86 V		1	1	1	0		
3.02 V	2.96 V		1	1	0	1		
3.13 V	3.06 V		0	1	0	0		
—		Settings other than the above are prohibited						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Caution **Be sure to set bit 4 to "1".**

Remark 1. For details on the LVD circuit, see **CHAPTER 26 VOLTAGE DETECTOR**.
Remark 2. The detection voltage is a typical value. For details, see **34.6.6 LVD circuit characteristics**.

Figure 29 - 4 Format of User Option Byte (000C1H/010C1H) (3/4)

Address: 000C1H/010C1H Note

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD setting (interrupt mode)

Detection Voltage		Option Byte Setting Value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
1.67 V	1.63 V	0	0	0	1	1	0	1
1.77 V	1.73 V		0	0	1	0		
1.88 V	1.84 V		0	1	1	1		
1.98 V	1.94 V		0	1	1	0		
2.09 V	2.04 V		0	1	0	1		
2.50 V	2.45 V		1	0	1	1		
2.61 V	2.55 V		1	0	1	0		
2.71 V	2.65 V		1	0	0	1		
2.81 V	2.75 V		1	1	1	1		
2.92 V	2.86 V		1	1	1	0		
3.02 V	2.96 V		1	1	0	1		
3.13 V	3.06 V		0	1	0	0		
—		Settings other than the above are prohibited						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Caution Be sure to set bit 4 to "1".

Remark 1. For details on the LVD circuit, see **CHAPTER 26 VOLTAGE DETECTOR**.

Remark 2. The detection voltage is a typical value. For details, see **34.6.6 LVD circuit characteristics**.

Figure 29 - 5 Format of User Option Byte (000C1H/010C1H) (4/4)

Address: 000C1H/010C1H Note

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD off setting (external reset input from the $\overline{\text{RESET}}$ pin is used)

Detection Voltage		Option Byte Setting Value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
—	—	1	×	×	×	×	×	1
—		Settings other than the above are prohibited						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Caution 1. Be sure to set bit 4 to "1".

Caution 2. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 34.4 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 LSI 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 010C2H).

Remark 1. ×: Don't care

Remark 2. For details on the LVD circuit, see **CHAPTER 26 VOLTAGE DETECTOR**.

Remark 3. The detection voltage is a typical value. For details, see **34.6.6 LVD circuit characteristics**.

Figure 29 - 6 Format of Option Byte (000C2H/010C2H)

Address: 000C2H/010C2H Note

7	6	5	4	3	2	1	0
CMODE1	CMODE0	1	0	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0

CMODE1	CMODE0	Setting of flash operation mode		
			Operating Frequency Range (f_{MAIN})	Operating Voltage Range (V_{DD})
0	0	LV (low-voltage main) mode	1 to 4 MHz	1.6 to 3.6 V
1	0	LS (low-speed main) mode	1 to 8 MHz	1.8 to 3.6 V
1	1	HS (high-speed main) mode	1 to 16 MHz	2.4 to 3.6 V
			1 to 24 MHz	2.7 to 3.6 V
Other than above		Setting prohibited		

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator clock (f_{IH})
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 Set the same value as 000C2H to 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.

Caution 1. Be sure to set bits 5 and 4 to 10B.

Caution 2. The operating frequency range and operating voltage range depend on each operating mode of the flash memory. See 34.4 AC Characteristics for details.

29.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

Figure 29 - 7 Format of On-chip Debug Option Byte (000C3H/010C3H)

Address: 000C3H/010C3H Note

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 debug operation.
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 erases data of flash memory in case of failures in authenticating on-chip debug security ID.

Note Set the same value as 000C3H to 010C3H when the boot swap operation is used because 000C3H is replaced by 010C3H.

Caution Bits 7 and 0 (OCDENSET and OCDERSD) can only be specified a value.
Be sure to set 000010B to bits 6 to 1.

Remark The value on bits 3 to 1 will be written over when the on-chip debug function is in use and thus it will become unstable after the setting.
However, be sure to set the default values (0, 1, and 0) to bits 3 to 1 at setting.

29.4 Setting of Option Byte

The user option byte and on-chip debug option byte can be set using the assembler linker option, in addition to describing in the source. When doing so, the contents set by using the link option take precedence, even if descriptions exist in the source, as mentioned below.

A software description example of the option byte setting is shown below.

```
OPT     CSEG    OPT_BYTE
        DB      36H     ; Does not use interval interrupt of watchdog timer,
                        ; Enables watchdog timer operation,
                        ; Window open period of watchdog timer is 50%,
                        ; Overflow time of watchdog timer is 2^9/fIL,
                        ; Stops watchdog timer operation during HALT/STOP mode
        DB      1AH     ; Select 1.63 V for VLVDL
                        ; Select rising edge 1.77 V, falling edge 1.73 V for VLVDH
                        ; Select the interrupt & reset mode as the LVD operation mode
        DB      2DH     ; Select the LV (low-voltage main) mode as the flash operation mode
                        ; and 1 MHz as the frequency of the high-speed on-chip oscillator clock
        DB      85H     ; Enables on-chip debug operation, does not erase flash memory
                        ; data when security ID authorization fails
```

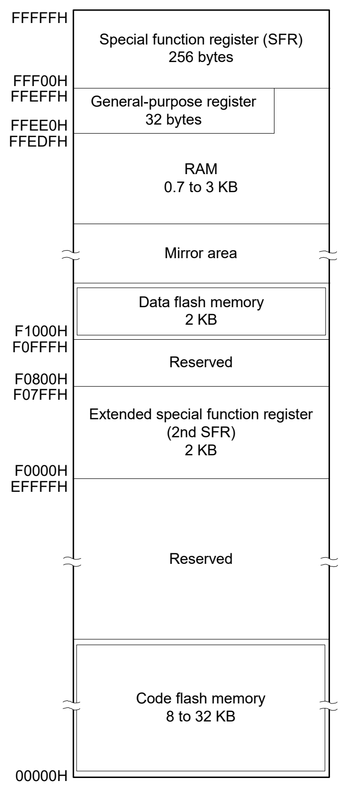
When the boot swap function is used during self-programming, 000C0H to 000C3H is switched to 010C0H to 010C3H. Describe to 010C0H to 010C3H, therefore, the same values as 000C0H to 000C3H as follows.

```
OPT2    CSEG    AT      010C0H
        DB              36H     ; Does not use interval interrupt of watchdog timer,
                                ; Enables watchdog timer operation,
                                ; Window open period of watchdog timer is 50%,
                                ; Overflow time of watchdog timer is 2^9/fIL,
                                ; Stops watchdog timer operation during HALT/STOP mode
        DB              1AH     ; Select 1.63 V for VLVDL
                                ; Select rising edge 1.77 V, falling edge 1.73 V for VLVDH
                                ; Select the interrupt & reset mode as the LVD operation mode
        DB              2DH     ; Select the LV (low main voltage) mode as the flash operation mode
                                ; and 1 MHz as the frequency of the high-speed on-chip oscillator clock
        DB              85H     ; Enables on-chip debug operation, does not erase flash memory
                                ; data when security ID authorization fails
```

Caution To specify the option byte by using assembly language, use OPT_BYTE as the relocation attribute name of the CSEG pseudo instruction. To specify the option byte to 010C0H to 010C3H in order to use the boot swap function, use the relocation attribute AT to specify an absolute address.

CHAPTER 30 FLASH MEMORY

The RL78 microcontroller incorporates the flash memory to which a program can be written, erased, and overwritten while mounted on the board. The flash memory includes the “code flash memory”, in which programs can be executed, and the “data flash memory”, an area for storing data.



The following methods for programming the flash memory are available.
The code flash memory can be rewritten to through serial programming using a flash memory programmer or an external device (UART communication), or through self-programming.

- Serial Programming Using Flash Memory Programmer (see **30.1**)
  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 **30.2**)
  Data can be written to the flash memory on-board through UART communication with an external device (microcontroller or ASIC).
- Self-Programming (see **30.6**)
  The user application can execute self-programming of the code flash memory by using the flash self-programming library.

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 operation mode register (FLMODE). Rewrite the flash memory when the MCSEL bit in the regulator mode control register (PMMC) is 0.

The data flash memory can be rewritten to by using the data flash library during user program execution (background operation). For access and writing to the data flash memory, see **30.8 Data Flash**.

30.1 Serial Programming Using Flash Memory Programmer

The following dedicated flash memory programmer can be used to write data to the internal flash memory of the RL78 microcontroller.

- PG-FP5, PG-FP6
- E1, E2, E2 Lite, E20 on-chip debugging emulator

Data can be written to the flash memory on-board or off-board, by using a dedicated flash memory programmer.

(1) On-board programming

The contents of the flash memory can be rewritten after the RL78 microcontroller has been mounted on the target system. The connectors that connect the dedicated flash memory programmer must be mounted on the target system.

(2) Off-board programming

Data can be written to the flash memory with a dedicated program adapter (FA series) before the RL78 microcontroller is mounted on the target system.

Table 30 - 1 Wiring Between RL78/I1D and Dedicated Flash Memory Programmer

<table>
<tr><th colspan="4" rowspan="2">Pin Configuration of Dedicated Flash Memory Programmer</th><th rowspan="4">Pin Name</th><th colspan="5">Pin No.</th></tr>
<tr><th>20-pin</th><th>24-pin</th><th>30-pin</th><th>32-pin</th><th>48-pin</th></tr>
<tr><th colspan="2">Signal Name</th><th rowspan="2">I/O</th><th rowspan="2">Pin Function</th><th rowspan="2">LSSOP
TSSOP
(4.4 × 6.5)</th><th rowspan="2">HWQFN
(4 × 4)</th><th rowspan="2">LSSOP</th><th rowspan="2">HVQFN
(5 × 5)
LQFP
(7 × 7)</th><th rowspan="2">LFQFP
(7 × 7)</th></tr>
<tr><th>PG-FP5, PG-FP6</th><th>E1, E2, E2 Lite, E20 on-chip debugging emulator</th></tr>
<tr><td>—</td><td>TOOL0</td><td>I/O</td><td>Transmit/ receive signal</td><td rowspan="2">TOOL0/ P40</td><td rowspan="2">2</td><td rowspan="2">23</td><td rowspan="2">4</td><td rowspan="2">31</td><td rowspan="2">45</td></tr>
<tr><td>SI/RxD</td><td>—</td><td>I/O</td><td>Transmit/ receive signal</td></tr>
<tr><td>—</td><td>$\overline{\text{RESET}}$</td><td>Output</td><td rowspan="2">Reset signal</td><td rowspan="2">$\overline{\text{RESET}}$</td><td rowspan="2">3</td><td rowspan="2">24</td><td rowspan="2">5</td><td rowspan="2">32</td><td rowspan="2">3</td></tr>
<tr><td>/RESET</td><td>—</td><td>Output</td></tr>
<tr><td colspan="2">V_{DD} Note 1</td><td>I/O</td><td>V_{DD} voltage generation/ power monitoring</td><td>V_{DD}</td><td>9</td><td>6</td><td>13</td><td>8</td><td>11</td></tr>
<tr><td colspan="2" rowspan="2">GND</td><td rowspan="2">—</td><td rowspan="2">Ground</td><td>V_{SS}</td><td>8</td><td>5</td><td>12</td><td>7</td><td>10</td></tr>
<tr><td>REGC Note 2</td><td>7</td><td>4</td><td>11</td><td>6</td><td>9</td></tr>
<tr><td>FLMD1</td><td>EMV_{DD}</td><td>—</td><td>Driving power for TOOL0 pin</td><td>V_{DD}</td><td>9</td><td>6</td><td>13</td><td>8</td><td>11</td></tr>
</table>

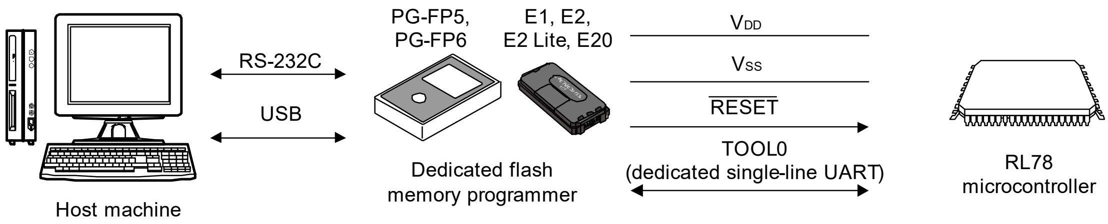
Note 1. The name of the signal for connection in the case of the PG-FP6 is V_{CC}.
Note 2. Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

Remark Pins that are not indicated in the above table can be left open when using the flash memory programmer for flash programming.

30.1.1 Programming Environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 30 - 1 Environment for Writing Program to Flash Memory



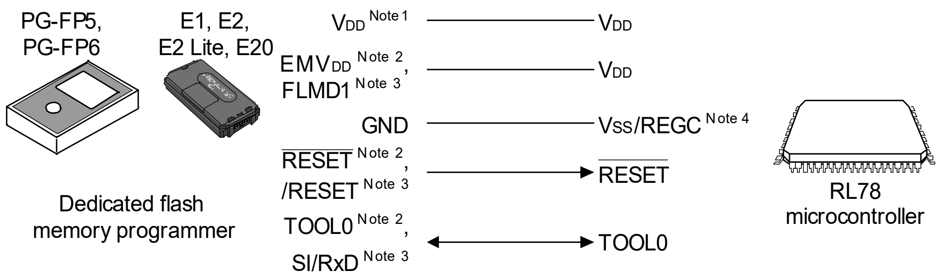
A host machine that controls the dedicated flash memory programmer is necessary.
To interface between the dedicated flash memory programmer and the RL78 microcontroller, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART.

30.1.2 Communication Mode

Communication between the dedicated flash memory programmer and the RL78 microcontroller is established by serial communication using the TOOL0 pin via a dedicated single-line UART of the RL78 microcontroller.

Transfer rate: 1 M, 500 k, 250 k, 115.2 kbps

Figure 30 - 2 Communication with Dedicated Flash Memory Programmer



Note 1. The name of the signal for connection in the case of the PG-FP6 is V_{CC}.
Note 2. When using E1, E2, E2 Lite, E20 on-chip debugging emulator.
Note 3. When using PG-FP5 or PG-FP6.
Note 4. Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

The dedicated flash memory programmer generates the following signals for the RL78 microcontroller. See the manual of PG-FP5, PG-FP6, or E1, E2, E2 Lite, E20 on-chip debugging emulator for details.

Table 30 - 2 Pin Connection

Dedicated Flash Memory Programmer				RL78 microcontroller
Signal Name		I/O	Pin Function	Pin Name Note 1
PG-FP5, PG-FP6	E1, E2, E2 Lite, E20 on-chip debugging emulator			
V_{DD} Note 2		I/O	V_{DD} voltage generation/power monitoring	V_{DD}
GND		—	Ground	V_{SS}, EV_{SS}, REGC Note 3
FLMD1	EMV_{DD}	—	Driving power for TOOL0 pin	V_{DD}
/RESET	—	Output	Reset signal	$\overline{\text{RESET}}$
—	$\overline{\text{RESET}}$	Output		
—	TOOL0	I/O	Transmit/receive signal	TOOL0
SI/RxD	—	I/O	Transmit/receive signal	

Note 1. Pins to be connected differ with the product. For details, see **Table 30 - 1**.
Note 2. The name of the signal for connection in the case of the PG-FP6 is V_{CC}.
Note 3. Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

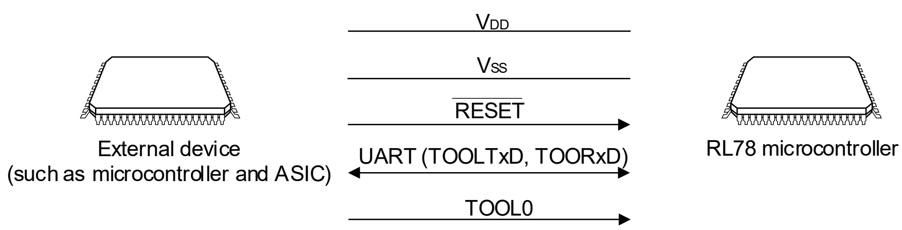
30.2 Serial Programming Using External Device (that Incorporates UART)

On-board data writing to the internal flash memory is possible by using the RL78 microcontroller and an external device (a microcontroller or ASIC) connected to a UART.
On the development of flash memory programmer by user, refer to the **RL78 Microcontrollers (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

30.2.1 Programming Environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 30 - 3 Environment for Writing Program to Flash Memory



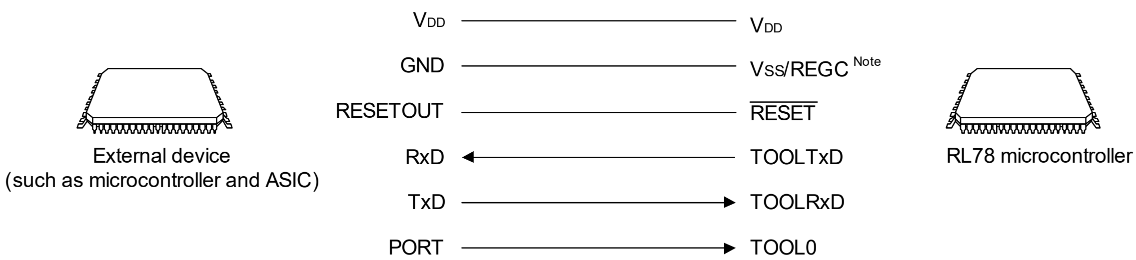
Processing to write data to or delete data from the RL78 microcontroller by using an external device is performed on-board. Off-board writing is not possible.

30.2.2 Communication Mode

Communication between the external device and the RL78 microcontroller is established by serial communication using the TOOLTxD and TOOLRxD pins via the dedicated UART of the RL78 microcontroller.

Transfer rate: 1 M, 500 k, 250 k, 115.2 kbps

Figure 30 - 4 Communication with External Device



Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

The external device generates the following signals for the RL78 microcontroller.

Table 30 - 3 Pin Connection

External Device			RL78 microcontroller
Signal Name	I/O	Pin Function	Pin Name
V_{DD}	I/O	V_{DD} voltage generation/power monitoring	V_{DD}
GND	—	Ground	V_{SS}, REGC Note
RESETOUT	Output	Reset signal output	$\overline{\text{RESET}}$
RxD	Input	Receive signal	TOOLTxD
TxD	Output	Transmit signal	TOOLRxD
PORT	Output	Mode signal	TOOL0

Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

30.3 Connection of Pins on Board

To write the flash memory on-board by using the flash memory programmer, connectors that connect the dedicated flash memory programmer must be provided on the target system. First provide a function that selects the normal operation mode or flash memory programming mode on the board.
When the flash memory programming mode is set, all the pins not used for programming the flash memory are in the same status as immediately after reset. Therefore, if the external device does not recognize the state immediately after reset, the pins must be handled as described below.

Remark For details on flash memory programming mode, refer to **30.4.2 Flash memory programming mode**.

30.3.1 P40/TOOL0 pin

In the flash memory programming mode, connect this pin to the dedicated flash memory programmer via an external 1 kΩ pull-up resistor.
When this pin is used as the port pin, use that by the following method.

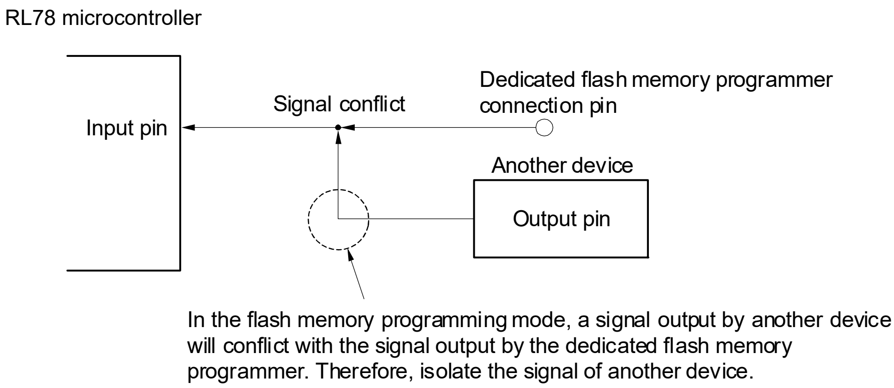
When used as an input pin:	Input of low-level is prohibited for t_{HD} period after external reset release. However, when this pin is used via pull-down resistors, use the 500 kΩ or more resistors.
When used as an output pin:	When this pin is used via pull-down resistors, use the 500 kΩ or more resistors.

Remark 1. t_{HD}: How long to keep the TOOL0 pin at the low level from when the external and internal resets end for setting of the flash memory programming mode (see **34.10 Timing of Entry to Flash Memory Programming Modes**).
Remark 2. The SAU pins are not used for communication between the RL78 microcontroller and dedicated flash memory programmer, because single-line UART (TOOL0 pin) is used.

30.3.2 $\overline{\text{RESET}}$ pin

Signal conflict will occur if the reset signal of the dedicated flash memory programmer and external device are connected to the $\overline{\text{RESET}}$ pin that is connected to the reset signal generator on the board. To prevent this conflict, isolate the connection with the reset signal generator.
The flash memory will not be correctly programmed if the reset signal is input from the user system while the flash memory programming mode is set. Do not input any signal other than the reset signal of the dedicated flash memory programmer and external device.

Figure 30 - 5 Signal Conflict ($\overline{\text{RESET}}$ Pin)



30.3.3 Port pins

Example When the flash memory programming mode is set, all the pins not used for flash memory programming enter the same status as that immediately after reset. If external devices connected to the ports do not recognize the port status immediately after reset, the port pin must be connected to V_{DD}, or V_{SS} via a resistor

30.3.4 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.

30.3.5 X1 and X2 pins

Connect X1 and X2 in the same status as in the normal operation mode.

Remark In the flash memory programming mode, the high-speed on-chip oscillator clock (f_{IH}) is used.

30.3.6 Power supply

To use the supply voltage output of the flash memory programmer, connect the V_{DD} pin to V_{DD}[Note] of the flash memory programmer, and the V_{SS} pin to GND of the flash memory programmer.
To use the on-board supply voltage, connect in compliance with the normal operation mode.
However, when writing to the flash memory by using the flash memory programmer and using the on-board supply voltage, be sure to connect the V_{DD} and V_{SS} pins to V_{DD}[Note] and GND of the flash memory programmer to use the power monitor function with the flash memory programmer.

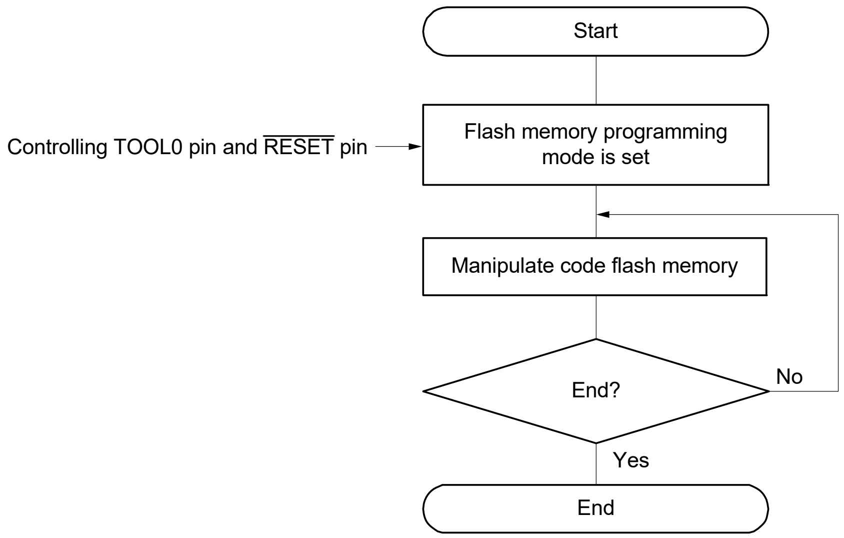
Note The name of the signal for connection in the case of the PG-FP6 is V_{CC}.

30.4 Programming Method

30.4.1 Serial programming procedure

The following figure illustrates a flow for rewriting the code flash memory through serial programming.

Figure 30 - 6 Code Flash Memory Manipulation Procedure



30.4.2 Flash memory programming mode

To rewrite the contents of the code flash memory through serial programming, specify the flash memory programming mode. To enter the mode, set as follows.

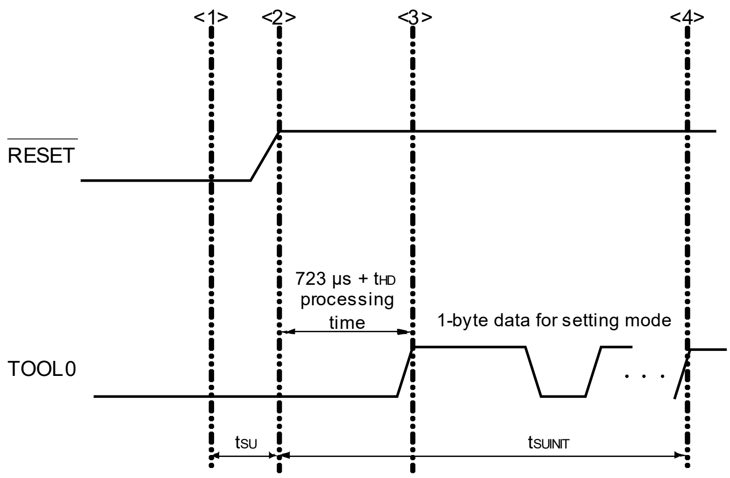
<When serial programming by using the dedicated flash memory programmer>
Connect the RL78 microcontroller to a dedicated flash memory programmer. Communication from the dedicated flash memory programmer is performed to automatically switch to the flash memory programming mode.

<When serial programming by using an external device>
Set the TOOL0 pin to the low level, and then cancel the reset (see **Table 30 - 4**). After that, enter flash memory programming mode according to the procedures <1> to <4> shown in **Figure 30 - 7**. For details, refer to the **RL78 microcontrollers (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

Table 30 - 4 Relationship Between TOOL0 Pin and Operation Mode After Reset Release

TOOL0	Operation Mode
V_{DD}	Normal operation mode
0 V	Flash memory programming mode

Figure 30 - 7 Setting of Flash Memory Programming Mode



<1> The low level is input to the TOOL0 pin.
<2> The external reset ends (POR and LVD reset must end before the external reset ends).
<3> The TOOL0 pin is set to the high level.
<4> Setting of the flash memory programming mode by UART reception and complete the baud rate setting.

Remark t_{SUINIT}: The segment shows that it is necessary to finish specifying the initial communication settings within 100 ms from when the external resets end.
t_{SU}: How long from when the TOOL0 pin is placed at the low level until a pin reset ends.
t_{HD}: How long to keep the TOOL0 pin at the low level from when the external resets end (the flash firmware processing time is excluded).

For details, see **34.10 Timing of Entry to Flash Memory Programming Modes**.

There are two flash memory programming modes: wide voltage mode and full speed mode. The supply voltage value applied to the microcontroller during write operations and the setting information of the user option byte for setting of the flash memory programming mode determine which mode is selected.
When a dedicated flash memory programmer is used for serial programming, setting the voltage on GUI selects the mode automatically.

Table 30 - 5 Programming Modes and Voltages at Which Data Can Be Written, Erased, or Verified

Power Supply Voltage (V_{DD})	User Option Byte Setting for Switching to Flash Memory Programming Mode		Flash Programming Mode
	Flash Operation Mode	Operating Frequency (f_{CLK})	
$2.7\ V \leq V_{DD} \leq 3.6\ V$	Blank state		Full speed mode
	HS (high-speed main) mode	1 MHz to 24 MHz	Full speed mode
	LS (low-speed main) mode	1 MHz to 8 MHz	Wide voltage mode
	LV (low-voltage main) mode	1 MHz to 4 MHz	Wide voltage mode
$2.4\ V \leq V_{DD} < 2.7\ V$	Blank state		Full speed mode
	HS (high-speed main) mode	1 MHz to 16 MHz	Full speed mode
	LS (low-speed main) mode	1 MHz to 8 MHz	Wide voltage mode
	LV (low-voltage main) mode	1 MHz to 4 MHz	Wide voltage mode
$1.8\ V \leq V_{DD} < 2.4\ V$	Blank state		Wide voltage mode
	LS (low-speed main) mode	1 MHz to 8 MHz	Wide voltage mode
	LV (low-voltage main) mode	1 MHz to 4 MHz	Wide voltage mode

Remark 1. Using both the wide voltage mode and full speed mode imposes no restrictions on writing, erasing, or verification.
Remark 2. For details about communication commands, see **30.4.4 Communication commands**.

30.4.3 Selecting communication mode

Communication mode of the RL78 microcontroller as follows.

Table 30 - 6 Communication Modes

Communication Mode	Standard Setting Note 1				Pins Used
	Port	Speed Note 2	Frequency	Multiply Rate	
1-line mode (when flash memory programmer is used, or when external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	—	—	TOOL0
Dedicated UART (when 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. Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

30.4.4 Communication commands

The RL78 microcontroller executes serial programming through the commands listed in **Table 30 - 7**.
The signals sent from the dedicated flash memory programmer or external device to the RL78 microcontroller are called commands, and programming functions corresponding to the commands are executed. For details, refer to the **RL78 microcontroller (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

Table 30 - 7 Flash Memory Control Commands

Classification	Command Name	Function
Verify	Verify	Compares the contents of a specified area of the flash memory with data transmitted from the programmer.
Erase	Block Erase	Erases a specified area in the flash memory.
Blank check	Block Blank Check	Checks if a specified block in the flash memory has been correctly erased
Write	Programming	Writes data to a specified area in the flash memory [Note].
Getting information	Silicon Signature	Gets the RL78 microcontroller information (such as the part number, flash memory configuration, and programming firmware version).
	Checksum	Gets the checksum data for a specified area.
Security	Security Set	Sets security information.
	Security Get	Gets security information.
	Security Release	Release setting of prohibition of writing.
Others	Reset	Used to detect synchronization status of communication.
	Baud Rate Set	Sets baud rate when UART communication mode is selected.

Note Confirm that no data has been written to the write area. Because data cannot be erased after block erase is prohibited, do not write data if the data has not been erased.

Product information (such as product name and firmware version) can be obtained by executing the “Silicon Signature” command.

Tables 30 - 8 and 30 - 9 show signature data list and example of signature data list.

Table 30 - 8 Signature Data List

Field name	Description	Number of transmit data
Device code	The serial number assigned to the device	3 bytes
Device name	Device name (ASCII code)	10 bytes
Code flash memory area last address	Last address of code flash memory area (Sent from lower address. Example. 00000H to 07FFFH (32 KB) → FFH, 7FH, 00H)	3 bytes
Data flash memory area last address	Last address of data flash memory area (Sent from lower address. Example. F1000H to F17FFH (2 KB) → FFH, 17H, 0FH)	3 bytes
Firmware version	Version information of firmware for programming (Sent from upper address. Example. From Ver. 1.23 → 01H, 02H, 03H)	3 bytes

Table 30 - 9 Signature Data List

Field name	Description	Number of transmit data	Data (hexadecimal)
Device code	RL78 protocol A	3 bytes	10 00 06
Device name	R5F117AC	10 bytes	52 = “R” 35 = “5” 46 = “F” 31 = “1” 31 = “1” 37 = “7” 41 = “A” 43 = “C” 20 = “ ” 20 = “ ”
Code flash memory area last address	Code flash memory area 00000H to 07FFFH (32 KB)	3 bytes	FFH 7FH 00H
Data flash memory area last address	Data flash memory area F1000H to F17FFH (2 KB)	3 bytes	FFH 17H 0FH
Firmware version	Ver.1.23	3 bytes	01 02 03

30.5 Processing Time for Each Command When Dedicated Flash Memory Programmer Is in Use (Reference Value)

The following shows the processing time for each command (reference value) when PG-FP5 or PG-FP6 is used as a dedicated flash memory programmer.

Table 30 - 10 Processing Time for Each Command When PG-FP5 or PG-FP6 Is in Use (Reference Value)

PG-FP5 or PG-FP6 Command	Port: TOOL0 (UART)			
	Speed: 1M bps			
	8 Kbytes	12 Kbytes	16 Kbytes	32 Kbytes
Erasing	1	1	1 s	1 s
Writing	1	1.5	1.5 s	1.5 s
Verification	1	1	1.5 s	1.5 s
Writing after erasing	1	1.5	1.5 s	2 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

Mode: Full speed mode (flash operation mode: HS (high speed main) mode)

30.6 Self-Programming

The RL78 microcontroller 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 by using the RL78 microcontroller self-programming library, it can be used to upgrade the program in the field.

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 the flash self-programming library in the state where the IE flag is cleared (0) by the DI instruction. To enable an interrupt, clear (0) the interrupt mask flag to accept in the state where the IE flag is set (1) by the EI instruction, and then execute the flash self-programming library.

Caution 3. The high-speed on-chip oscillator should be kept operating during self-programming. If it is stopped, its clock should be operated (HIOSTOP = 0), and the flash self-programming library should be executed after 65 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. When rewriting the flash memory, do not change the flash operation mode register (FLMODE). Rewrite the flash memory when the MCSEL bit in the regulator mode control register (PMMC) is 0.

Remark For details of the self-programming function, refer to the **RL78 microcontroller Flash Self-Programming Library Type 01 User's Manual (R01US0050)**.

The self-programming function has two flash memory programming modes; wide voltage mode and full speed mode. Specify the mode that corresponds to the flash operation mode specified in bits CMODE1 and CMODE0 in option byte 000C2H.

Specify the full speed mode when the HS (high-speed main) mode is specified. Specify the wide voltage mode when the LS (low-speed main) mode or LV (low-voltage main) mode is specified.

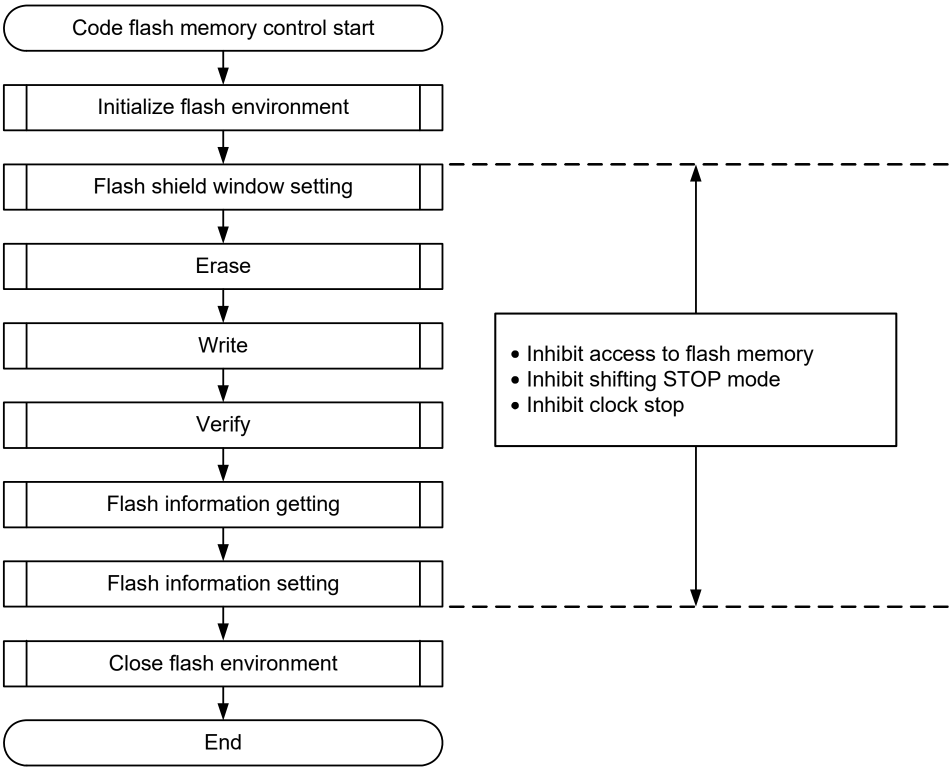
If the argument fsl_flash_voltage_u08 is 00H when the FSL_Init function of the flash self-programming library provided by Renesas Electronics is executed, full speed mode is specified. If the argument is other than 00H, the wide voltage mode is specified.

Remark Using both the wide voltage mode and full speed mode imposes no restrictions on writing, erasing, or verification.

30.6.1 Self-programming procedure

The following figure illustrates a flow for rewriting the code flash memory by using a flash self-programming library.

Figure 30 - 8 Flow of Self-Programming (Rewriting Flash Memory)



30.6.2 Boot swap function

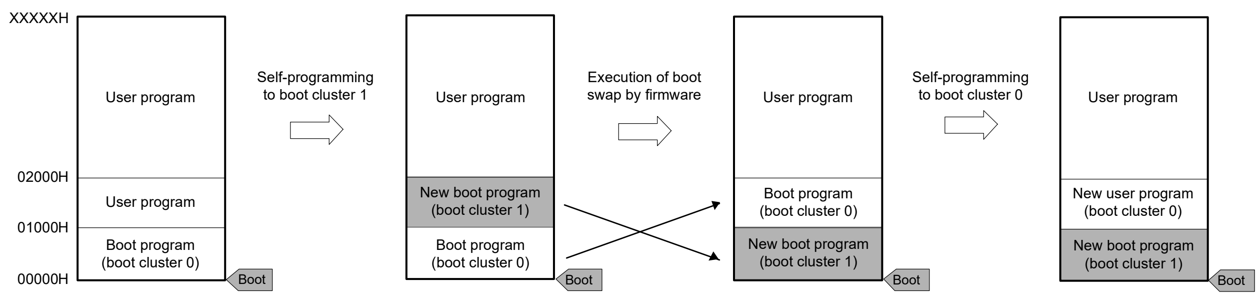
If rewriting the boot area failed by temporary power failure or other reasons, restarting a program by resetting or overwriting is disabled due to data destruction in the boot area.

The boot swap function is used to avoid this problem.

Before erasing boot cluster 0 Note, which is a boot program area, by self-programming, write a new boot program to boot cluster 1 in advance. When the program has been correctly written to boot cluster 1, swap this boot cluster 1 and boot cluster 0 by using the set information function of the firmware of the RL78 microcontroller, so that boot cluster 1 is used as a boot area. After that, erase or write the original boot program area, boot cluster 0. As a result, even if a power failure occurs while the area is being rewritten, the program is executed correctly because it is booted from boot cluster 1 to be swapped when the program is reset and started next.

Note A boot cluster is a 4 KB area and boot clusters 0 and 1 are swapped by the boot swap function.

Figure 30 - 9 Boot Swap Function

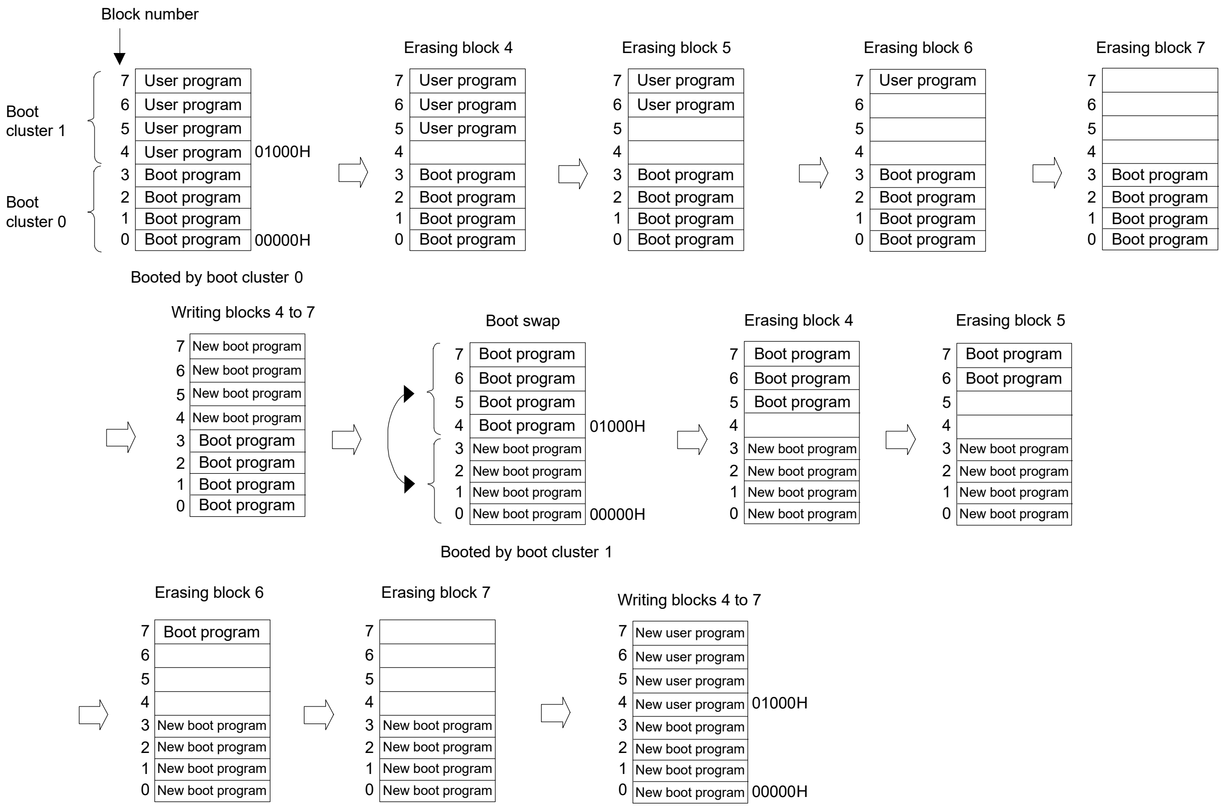


In an example of above figure, it is as follows.

Boot cluster 0: Boot area before boot swap

Boot cluster 1: Boot area after boot swap

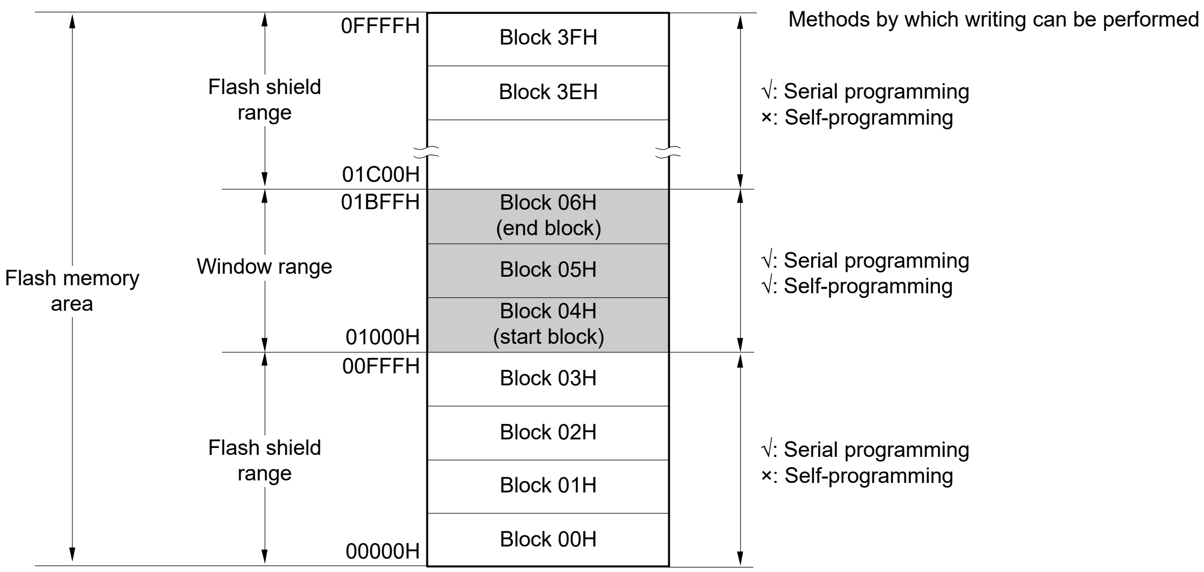
Figure 30 - 10 Example of Executing Boot Swapping



30.6.3 Flash shield window function

The flash shield window function is provided as one of the security functions for self-programming. It disables writing to and erasing areas outside the range specified as a window only during self-programming.
The window range can be set by specifying the start and end blocks. The window range can be set or changed during both serial programming and self-programming.
Writing to and erasing areas outside the window range are disabled during self-programming. During serial programming, however, areas outside the range specified as a window can be written and erased.

Figure 30 - 11 Flash Shield Window Setting Example
(Target Devices: R5F104LE, Start Block: 04H, End Block: 06H)



Caution 1. If the rewrite-prohibited area of the boot cluster 0 overlaps with the flash shield window range, prohibition to rewrite the boot cluster 0 takes priority.

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 30 - 11 Relationship between Flash Shield Window Function Setting/Change Methods and Commands

Programming conditions	Window Range Setting/ Change Methods	Execution Commands	
		Block erase	Write
Self-programming	Specify the starting and ending blocks by the flash self-programming library.	Block erasing is enabled only within the window range.	Writing is enabled only within the range of window range.
Serial programming	Specify the starting and ending blocks on GUI of dedicated flash memory programmer, etc.	Block erasing is enabled also outside the window range.	Writing is enabled also outside the window range.

Remark See **30.7 Security Settings** to prohibit writing/erasing during serial programming.

30.7 Security Settings

The RL78 microcontroller 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 using the Security Set command.

- Disabling block erase
  Execution of the block erase command for a specific block in the flash memory is prohibited during serial programming. However, blocks can be erased by means of self-programming.

- Disabling write
  Execution of the write command for entire blocks in the flash memory is prohibited during serial programming. However, blocks can be written by means of self-programming.
  After the setting of prohibition of writing is specified, releasing the setting by the Security Release command is enabled by a reset.

- Disabling rewriting boot cluster 0
  Execution of the block erase command and write command on boot cluster 0 (00000H to 00FFFH) in the flash memory is prohibited by this setting.

The block erase, write commands, and rewriting boot cluster 0 are enabled by the default setting when the flash memory is shipped. Security can be set by serial programming and self-programming. Each security setting can be used in combination.
Table 30 - 12 shows the relationship between the erase and write commands when the RL78 microcontroller security function is enabled.
After the security settings are specified, releasing the security settings by the Security Release command is enabled by a reset.

Caution The security function of the flash programmer does not support self-programming.

Remark To prohibit writing and erasing during self-programming, use the flash shield window function (see **30.6.3** for detail).

Table 30 - 12 Relationship Between Enabling Security Function and Command

(1) During serial programming

Valid Security	Executed Command	
	Block Erase	Write
Prohibition of block erase	Blocks cannot be erased.	Can be performed.[Note]
Prohibition of writing	Blocks can be erased.	Cannot be performed.
Prohibition of rewriting boot cluster 0	Boot cluster 0 cannot be erased.	Boot cluster 0 cannot be written.

Note Confirm that no data has been written to the write area. Because data cannot be erased after block erase is prohibited, do not write data if the data has not been erased.

(2) During self-programming

Valid Security	Executed Command	
	Block Erase	Write
Prohibition of block erase	Blocks can be erased.	Can be performed.
Prohibition of writing		
Prohibition of rewriting boot cluster 0	Boot cluster 0 cannot be erased.	Boot cluster 0 cannot be written.

Remark To prohibit writing and erasing during self-programming, use the flash shield window function (see **30.6.3** for detail).

Table 30 - 13 Setting Security in Each Programming Mode

(1) During serial programming

Security	Security Setting	How to Disable Security Setting
Prohibition of block erase	Set via GUI of dedicated flash memory programmer, etc.	Cannot be disabled after set.
Prohibition of writing		Set via GUI of dedicated flash memory programmer, etc.
Prohibition of rewriting boot cluster 0		Cannot be disabled after set.

Caution Releasing the setting of prohibition of writing is enabled only when the security is not set as the block erase prohibition and the boot cluster 0 rewrite prohibition with code flash memory area and data flash memory area being blanks.

(2) During self-programming

Security	Security Setting	How to Disable Security Setting
Prohibition of block erase	Set by using flash self-programming library.	Cannot be disabled after set.
Prohibition of writing		Cannot be disabled during self-programming (set via GUI of dedicated flash memory programmer, etc. during serial programming).
Prohibition of rewriting boot cluster 0		Cannot be disabled after set.

30.8 Data Flash

30.8.1 Data flash overview

An overview of the data flash memory is provided below.

- The user program can rewrite the data flash memory by using the flash data library. For details, refer to RL78 Family Flash Data Library User's Manual.
- The data flash memory can also be rewritten to through serial programming using the dedicated flash memory programmer or an external device.
- The data flash can be erased in 1-block (1 KB) units.
- The data flash can be accessed only in 8-bit units.
- The data flash 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).
- Because the data flash memory is an area exclusively used for data, it cannot be used to execute instructions.
- Accessing the data flash memory is not possible while rewriting the code flash memory (during self-programming).
- Manipulating the DFLCTL register is not possible while rewriting the data flash memory.
- Transition to the STOP mode is not possible while rewriting the data flash memory.

Caution 1. The data flash memory is stopped after a reset is canceled. 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 data flash rewrite. If it is stopped, its clock should be operated (HIOSTOP = 0), and the data flash library should be executed after 65 μs have elapsed.

Remark For the flash programming mode, see **30.6 Self-Programming**.

30.8.2 Register controlling data flash memory

30.8.2.1 Data flash control register (DFLCTL)

This register is used to enable or disable accessing to the data flash.
The DFLCTL register is set by a 1-bit or 8-bit memory manipulation instruction.
Reset input sets this register to 00H.

Figure 30 - 12 Format of Data flash control register (DFLCTL)

Address: F0090H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
DFLCTL	0	0	0	0	0	0	0	DFLEN

DFLEN	Data flash access control
0	Disables data flash access
1	Enables data flash access

Caution Manipulating the DFLCTL register is not possible while rewriting the data flash memory.

30.8.3 Procedure for accessing data flash memory

The data flash memory is initially stopped after a reset ends and cannot be accessed (read or programmed). To access the memory, perform the following procedure:

<1> Write 1 to bit 0 (DFLEN) of the data flash control register (DFLCTL).
<2> Wait for the setup to finish for software timer, etc.
The time setup takes differs for each main clock mode.
<Setup time for each main clock mode>
- HS (high-speed main) mode: 5 μs
- LS (low-speed main) mode: 720 ns
- LP (low-power main) mode: 720 ns
- LV (low-voltage main) mode: 10 μs

<3> After the wait, the data flash memory can be accessed.

Caution 1. Accessing the data flash memory is not possible during the setup time.
Caution 2. Transition to the STOP mode is not possible during the setup time. To enter the STOP mode during the setup time, clear DFLEN to 0 and then execute the STOP instruction.
Caution 3. The high-speed on-chip oscillator should be kept operating during data flash rewrite. If it is kept stopped, its clock should be operated (HIOSTOP = 0), and the data flash library should be executed after 65 μs have elapsed.
Caution 4. Once the data flash memory is read while the subsystem clock is selected as the CPU/peripheral hardware clock (CLS = 1), follow the procedure listed as steps (1) to (3) below, in that order, to read the data flash area after switching the CPU/peripheral hardware clock from the subsystem clock to the main system clock.
(1) Make sure the main system clock is selected as the CPU/peripheral hardware clock (CLS = 0).
(2) Read data from any location in the data flash area. The value read at this point is undefined.
(3) Wait for the time listed below according to the operating mode, then read data from the desired parts of the data flash area.
HS (high-speed main) mode: 5 μs
LS (low-speed main) mode: 1 μs
LV (low-voltage main) mode: 10 μs
LP (low-power main) mode: 1 μs

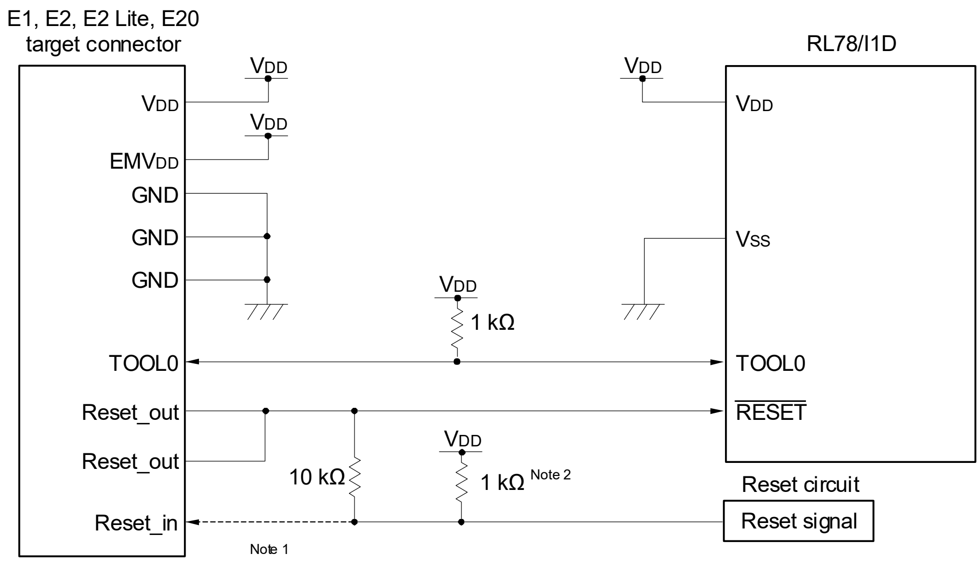
CHAPTER 31 ON-CHIP DEBUG FUNCTION

31.1 Connecting E1, E2, E2 Lite, E20 On-chip Debugging Emulator

The RL78 microcontroller uses the VDD, $\overline{\text{RESET}}$, TOOL0, and VSS pins to communicate with the host machine via an E1, E2, E2 Lite, E20 on-chip debugging emulator. Serial communication is performed by using a single-line UART that uses the TOOL0 pin.

Caution The RL78 microcontroller has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

Figure 31 - 1 Connection Example of E1, E2, E2 Lite, E20 On-chip Debugging Emulator



Note 1. Connecting the dotted line is not necessary during serial programming.

Note 2. If the reset circuit on the target system does not have a buffer and generates a reset signal only with resistors and capacitors, this pull-up resistor is not necessary.

Caution This circuit diagram is assumed that the reset signal outputs from an N-ch O.D. buffer (output resistor: 100 Ω or less)

31.2 On-Chip Debug Security ID

The RL78 microcontroller has an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 29 OPTION BYTE**) and an on-chip debug security ID setting area at 000C4H to 000CDH, to prevent third parties from reading memory content.

When the boot swap function is used, also set a value that is the same as that of 010C3H and 010C4H to 010CDH in advance, because 000C3H, 000C4H to 000CDH and 010C3H, and 010C4H to 010CDH are switched.

Table 31 - 1 On-Chip Debug Security ID

Address	On-Chip Debug Security ID
000C4H to 000CDH	Any ID code of 10 bytes
010C4H to 010CDH	

31.3 Securing of User Resources

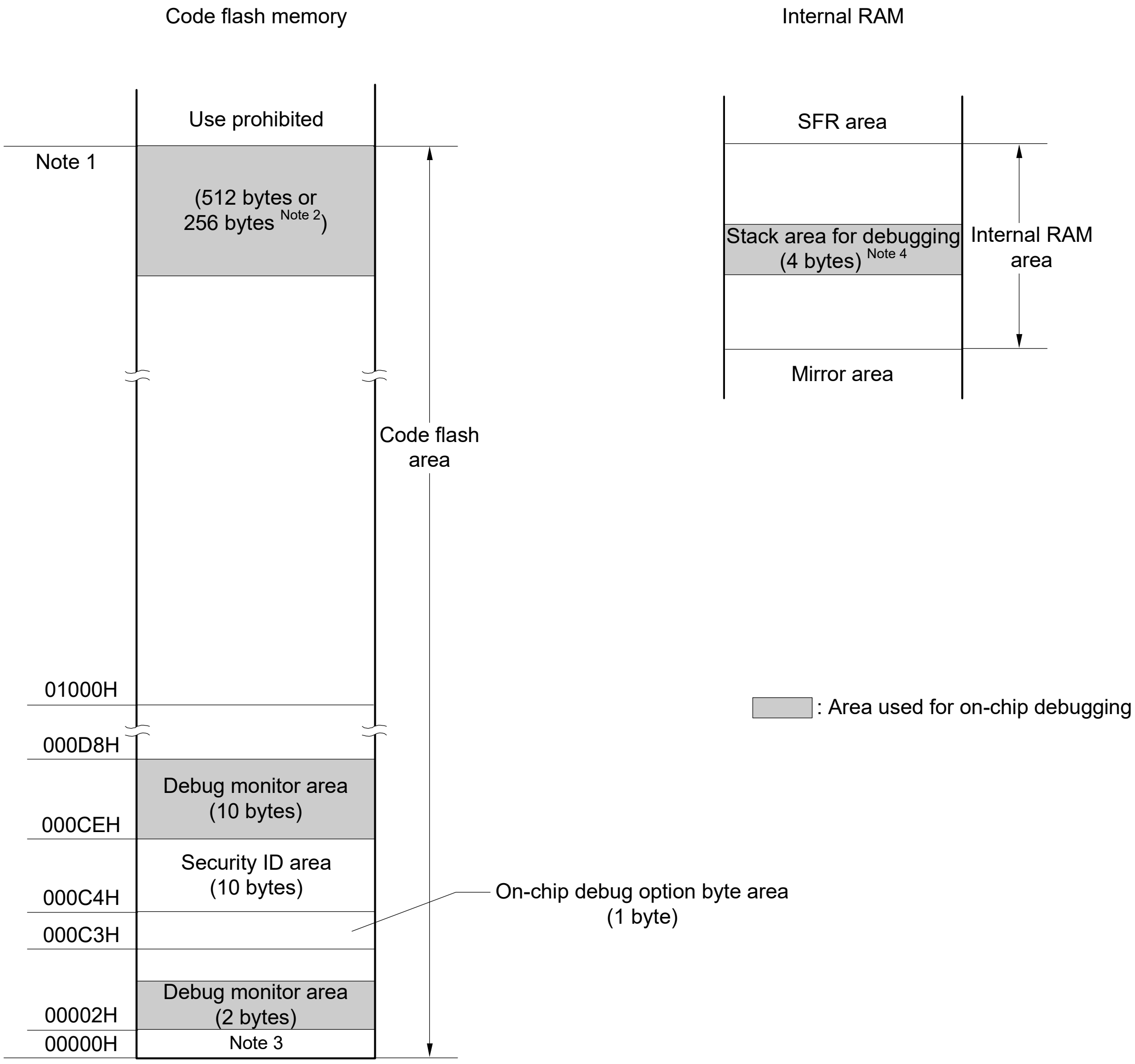
To perform communication between the RL78 microcontroller and E1, E2, E2 Lite, E20 on-chip debugging emulator, as well as each debug function, the securing of memory space must be done beforehand.

If Renesas Electronics assembler or compiler is used, the items can be set by using link options.

(1) Securement of memory space

The shaded portions in Figure 31 - 2 are the areas reserved for placing the debug monitor program, so user programs or data cannot be allocated in these spaces. When using the on-chip debug function, these spaces must be secured so as not to be used by the user program. Moreover, this area must not be rewritten by the user program.

Figure 31 - 2 Memory Spaces Where Debug Monitor Programs Are Allocated



Note 1. Address differs depending on products as follows.

Products (code flash memory capacity)	Address of **Note 1.**
R5F117x8 (x = 6, 7, A)	01FFFH
R5F117xA (x = 6, 7, A, B, G)	03FFFH
R5F117xC (x = A, B, G)	07FFFH

Note 2. When real-time RAM monitor (RRM) function and dynamic memory modification (DMM) function are not used, it is 256 bytes.

Note 3. In debugging, reset vector is rewritten to address allocated to a monitor program.

Note 4. Since this area is allocated immediately before the stack area, the address of this area varies depending on the stack increase and decrease. That is, 4 extra bytes are consumed for the stack area used.
When using self-programming, 12 extra bytes are consumed for the stack area used.

CHAPTER 32 BCD CORRECTION CIRCUIT

32.1 BCD Correction Circuit Function

The result of addition/subtraction of the BCD (binary-coded decimal) code and BCD code can be obtained as BCD code with this circuit.
The decimal correction operation result is obtained by performing addition/subtraction having the A register as the operand and then adding/ subtracting the BCD correction result register (BCDADJ).

32.2 Registers Used by BCD Correction Circuit

The BCD correction circuit uses the following registers.

• BCD correction result register (BCDADJ)

32.2.1 BCD correction result register (BCDADJ)

The BCDADJ register stores correction values for obtaining the add/subtract result as BCD code through add/subtract instructions using the A register as the operand.
The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags.
The BCDADJ register is read by an 8-bit memory manipulation instruction.
Reset input sets this register to undefined.

Figure 32 - 1 Format of BCD correction result register (BCDADJ)

Address: F00FEH After reset: Undefined R

Symbol	7	6	5	4	3	2	1	0
BCDADJ								

32.3 BCD Correction Circuit Operation

The basic operation of the BCD correction circuit is as follows.

(1) Addition: Calculating the result of adding a BCD code value and another BCD code value by using a BCD code value

<1> The BCD code value to which addition is performed is stored in the A register.

<2> By adding the value of the A register and the second operand (value of one more BCD code to be added) as are in binary, the binary operation result is stored in the A register and the correction value is stored in the BCD correction result register (BCDADJ).

<3> Decimal correction is performed by adding in binary the value of the A register (addition result in binary) and the BCDADJ register (correction value), and the correction result is stored in the A register and CY flag.

Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.

An example is shown below.

Examples 1: 99 + 89 = 188

Instruction			A Register	CY Flag	AC Flag	BCDADJ Register
MOV	A, #99H	; <1>	99H	—	—	—
ADD	A, #89H	; <2>	22H	1	1	66H
ADD	A, !BCDADJ	; <3>	88H	1	0	—

Examples 2: 85 + 15 = 100

Instruction			A Register	CY Flag	AC Flag	BCDADJ Register
MOV	A, #85H	; <1>	85H	—	—	—
ADD	A, #15H	; <2>	9AH	0	0	66H
ADD	A, !BCDADJ	; <3>	00H	1	1	—

Examples 3: 80 + 80 = 160

Instruction			A Register	CY Flag	AC Flag	BCDADJ Register
MOV	A, #80H	; <1>	80H	—	—	—
ADD	A, #80H	; <2>	00H	1	0	60H
ADD	A, !BCDADJ	; <3>	60H	1	0	—

(2) Subtraction: Calculating the result of subtracting a BCD code value from another BCD code value by using a BCD code value

<1> The BCD code value from which subtraction is performed is stored in the A register.

<2> By subtracting the value of the second operand (value of BCD code to be subtracted) from the A register as is in binary, the calculation result in binary is stored in the A register, and the correction value is stored in the BCD correction result register (BCDADJ).

<3> Decimal correction is performed by subtracting the value of the BCDADJ register (correction value) from the A register (subtraction result in binary) in binary, and the correction result is stored in the A register and CY flag.

Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.

An example is shown below.

Example: 91 - 52 = 39

Instruction			A Register	CY Flag	AC Flag	BCDADJ Register
MOV	A, #91H	; <1>	91H	—	—	—
SUB	A, #52H	; <2>	3FH	0	1	06H
SUB	A, !BCDADJ	; <3>	39H	0	0	—

CHAPTER 33 INSTRUCTION SET

This chapter lists the instructions in the RL78 microcontroller instruction set. For details of each operation and operation code, refer to the separate document **RL78 Family User's Manual Software (R01US0015)**.

33.1 Conventions Used in Operation List

33.1.1 Operand identifiers and specification methods

Operands are described in the "Operand" column of each instruction in accordance with the description method of the instruction operand identifier (refer to the assembler specifications for details). When there are two or more description methods, select one of them. Alphabetic letters in capitals and the symbols, #, !, !!, $, $!, [ ], and ES: are keywords and are described as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: 16-bit absolute address specification
- !!: 20-bit absolute address specification
- $: 8-bit relative address specification
- $!: 16-bit relative address specification
- [ ]: Indirect address specification
- ES:: Extension address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to describe the #, !, !!, $, $!, [ ], and ES: symbols.
For operand register identifiers, r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in the table below, R0, R1, R2, etc.) can be used for description.

Table 33 - 1 Operand Identifiers and Specification Methods

Identifier	Description Method
r 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 **Tables 3 - 5** to **3 - 7 Special Function Register (SFR) List** for the symbols of the special function registers. The extended special function registers can be described to operand !addr16 as symbols. See **Tables 3 - 8** to **3 - 12 Extended Special Function Register (2nd SFR) List** for the symbols of the extended special function registers.

33.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

Table 33 - 2 Symbols in “Operation” Column

Symbol	Function
A	A register; 8-bit accumulator
X	X register
B	B register
C	C register
D	D register
E	E register
H	H register
L	L register
ES	ES register
CS	CS register
AX	AX register pair; 16-bit accumulator
BC	BC register pair
DE	DE register pair
HL	HL register pair
PC	Program counter
SP	Stack pointer
PSW	Program status word
CY	Carry flag
AC	Auxiliary carry flag
Z	Zero flag
RBS	Register bank select flag
IE	Interrupt request enable flag
()	Memory contents indicated by address or register contents in parentheses
X_H, X_L 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)
$\wedge$	Logical product (AND)
$\vee$	Logical sum (OR)
$\veebar$	Exclusive logical sum (exclusive OR)
—	Inverted data
addr5	16-bit immediate data (even addresses only in 0080H to 00BFH)
addr16	16-bit immediate data
addr20	20-bit immediate data
jdisp8	Signed 8-bit data (displacement value)
jdisp16	Signed 16-bit data (displacement value)

33.1.3 Description of flag operation column

The change of the flag value when the instruction is executed is shown in the “Flag” column using the following symbols.

Table 33 - 3 Symbols in “Flag” Column

Symbol	Change of Flag Value
(Blank)	Unchanged
0	Cleared to 0
1	Set to 1
×	Set/cleared according to the result
R	Previously saved value is restored

33.1.4 PREFIX instruction

Instructions with “ES:” have a PREFIX operation code as a prefix to extend the accessible data area to the 1 MB space (00000H to FFFFFH), by adding the ES register value to the 64 KB space from F0000H to FFFFFH. When a PREFIX operation code is attached as a prefix to the target instruction, only one instruction immediately after the PREFIX operation code is executed as the addresses with the ES register value added.

A interrupt and DTC transfer are not acknowledged between a PREFIX instruction code and the instruction immediately after.

Table 33 - 4 Use Example of PREFIX Operation Code

Instruction	Opcode				
	1	2	3	4	5
MOV !addr16, #byte	CFH	!addr16		#byte	—
MOV ES:!addr16, #byte	11H	CFH	!addr16		#byte
MOV A, [HL]	8BH	—	—	—	—
MOV A, ES: [HL]	11H	8BH	—	—	—

Caution Set the ES register value with MOV ES, A, etc., before executing the PREFIX instruction.

33.2 Operation List

Table 33 - 5 Operation List (1/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	r, #byte	2	1	—	r ← byte			
		PSW, #byte	3	3	—	PSW ← byte	×	×	×
		CS, #byte	3	1	—	CS ← byte			
		ES, #byte	2	1	—	ES ← byte			
		!addr16, #byte	4	1	—	(addr16) ← byte			
		ES:!addr16, #byte	5	2	—	(ES, addr16) ← byte			
		saddr, #byte	3	1	—	(saddr) ← byte			
		sfr, #byte	3	1	—	sfr ← byte			
		[DE+byte], #byte	3	1	—	(DE + byte) ← byte			
		ES:[DE+byte], #byte	4	2	—	((ES, DE) + byte) ← byte			
		[HL+byte], #byte	3	1	—	(HL + byte) ← byte			
		ES:[HL+byte], #byte	4	2	—	((ES, HL) + byte) ← byte			
		[SP+byte], #byte	3	1	—	(SP + byte) ← byte			
		word[B], #byte	4	1	—	(B + word) ← byte			
		ES:word[B], #byte	5	2	—	((ES, B) + word) ← byte			
		word[C], #byte	4	1	—	(C+word) ← byte			
		ES:word[C], #byte	5	2	—	((ES, C) + word) ← byte			
		word[BC], #byte	4	1	—	(BC+word) ← byte			
		ES:word[BC], #byte	5	2	—	((ES, BC) + word) ← byte			
		A, r Note 3	1	1	—	A ← r			
		r, A Note 3	1	1	—	r ← A			
		A, PSW	2	1	—	A ← PSW			
		PSW, A	2	3	—	PSW ← A	×	×	×
		A, CS	2	1	—	A ← CS			
		CS, A	2	1	—	CS ← A			
		A, ES	2	1	—	A ← ES			
		ES, A	2	1	—	ES ← A			
		A, !addr16	3	1	4	A ← (addr16)			
		A, ES:!addr16	4	2	5	A ← (ES, addr16)			
		!addr16, A	3	1	—	(addr16) ← A			
		ES:!addr16, A	4	2	—	(ES, addr16) ← A			
		A, saddr	2	1	—	A ← (saddr)			
		saddr, A	2	1	—	(saddr) ← A			

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 6 Operation List (2/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	A, sfr	2	1	—	A ← sfr			
		sfr, A	2	1	—	sfr ← A			
		A, [DE]	1	1	4	A ← (DE)			
		[DE], A	1	1	—	(DE) ← A			
		A, ES:[DE]	2	2	5	A ← (ES, DE)			
		ES:[DE], A	2	2	—	(ES, DE) ← A			
		A, [HL]	1	1	4	A ← (HL)			
		[HL], A	1	1	—	(HL) ← A			
		A, ES:[HL]	2	2	5	A ← (ES, HL)			
		ES:[HL], A	2	2	—	(ES, HL) ← A			
		A, [DE+byte]	2	1	4	A ← (DE + byte)			
		[DE+byte], A	2	1	—	(DE + byte) ← A			
		A, ES:[DE+byte]	3	2	5	A ← ((ES, DE) + byte)			
		ES:[DE+byte], A	3	2	—	((ES, DE) + byte ← A			
		A, [HL+byte]	2	1	4	A ← (HL + byte)			
		[HL+byte], A	2	1	—	(HL + byte) ← A			
		A, ES:[HL+byte]	3	2	5	A ← ((ES, HL) + byte)			
		ES:[HL+byte], A	3	2	—	((ES, HL) + byte) ← A			
		A, [SP+byte]	2	1	—	A ← (SP + byte)			
		[SP+byte], A	2	1	—	(SP + byte) ← A			
		A, word[B]	3	1	4	A ← (B + word)			
		word[B], A	3	1	—	(B + word) ← A			
		A, ES:word[B]	4	2	5	A ← ((ES, B) + word)			
		ES:word[B], A	4	2	—	((ES, B) + word) ← A			
		A, word[C]	3	1	4	A ← (C + word)			
		word[C], A	3	1	—	(C + word) ← A			
		A, ES:word[C]	4	2	5	A ← ((ES, C) + word)			
		ES:word[C], A	4	2	—	((ES, C) + word) ← A			
		A, word[BC]	3	1	4	A ← (BC + word)			
		word[BC], A	3	1	—	(BC + word) ← A			
		A, ES:word[BC]	4	2	5	A ← ((ES, BC) + word)			
		ES:word[BC], A	4	2	—	((ES, BC) + word) ← A			

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 7 Operation List (3/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	A, [HL+B]	2	1	4	A ← (HL + B)			
		[HL+B], A	2	1	—	(HL + B) ← A			
		A, ES:[HL+B]	3	2	5	A ← ((ES, HL) + B)			
		ES:[HL+B], A	3	2	—	((ES, HL) + B) ← A			
		A, [HL+C]	2	1	4	A ← (HL + C)			
		[HL+C], A	2	1	—	(HL + C) ← A			
		A, ES:[HL+C]	3	2	5	A ← ((ES, HL) + C)			
		ES:[HL+C], A	3	2	—	((ES, HL) + C) ← A			
		X, !addr16	3	1	4	X ← (addr16)			
		X, ES:!addr16	4	2	5	X ← (ES, addr16)			
		X, saddr	2	1	—	X ← (saddr)			
		B, !addr16	3	1	4	B ← (addr16)			
		B, ES:!addr16	4	2	5	B ← (ES, addr16)			
		B, saddr	2	1	—	B ← (saddr)			
		C, !addr16	3	1	4	C ← (addr16)			
		C, ES:!addr16	4	2	5	C ← (ES, addr16)			
		C, saddr	2	1	—	C ← (saddr)			
		ES, saddr	3	1	—	ES ← (saddr)			
	XCH	A, r Note 3	1 (r = X) 2 (other than r = X)	1	—	A ←→ r			
		A, !addr16	4	2	—	A ←→ (addr16)			
		A, ES:!addr16	5	3	—	A ←→ (ES, addr16)			
		A, saddr	3	2	—	A ←→ (saddr)			
		A, sfr	3	2	—	A ←→ sfr			
		A, [DE]	2	2	—	A ←→ (DE)			
		A, ES:[DE]	3	3	—	A ←→ (ES, DE)			
		A, [HL]	2	2	—	A ←→ (HL)			
		A, ES:[HL]	3	3	—	A ←→ (ES, HL)			
		A, [DE+byte]	3	2	—	A ←→ (DE + byte)			
		A, ES:[DE+byte]	4	3	—	A ←→ ((ES, DE) + byte)			
		A, [HL+byte]	3	2	—	A ←→ (HL + byte)			
		A, ES:[HL+byte]	4	3	—	A ←→ ((ES, HL) + byte)			

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 8 Operation List (4/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	XCH	A, [HL+B]	2	2	—	A ←→ (HL + B)			
		A, ES:[HL+B]	3	3	—	A ←→ ((ES, HL) + B)			
		A, [HL+C]	2	2	—	A ←→ (HL + C)			
		A, ES:[HL+C]	3	3	—	A ←→ ((ES, HL) + C)			
	ONEB	A	1	1	—	A ← 01H			
		X	1	1	—	X ← 01H			
		B	1	1	—	B ← 01H			
		C	1	1	—	C ← 01H			
		!addr16	3	1	—	(addr16) ← 01H			
		ES:!addr16	4	2	—	(ES, addr16) ← 01H			
		saddr	2	1	—	(saddr) ← 01H			
	CLRB	A	1	1	—	A ← 00H			
		X	1	1	—	X ← 00H			
		B	1	1	—	B ← 00H			
		C	1	1	—	C ← 00H			
		!addr16	3	1	—	(addr16) ← 00H			
		ES:!addr16	4	2	—	(ES,addr16) ← 00H			
		saddr	2	1	—	(saddr) ← 00H			
	MOVS	[HL+byte], X	3	1	—	(HL + byte) ← X	×		×
		ES:[HL+byte], X	4	2	—	(ES, HL + byte) ← X	×		×
16-bit data transfer	MOVW	rp, #word	3	1	—	rp ← word			
		saddrp, #word	4	1	—	(saddrp) ← word			
		sfrp, #word	4	1	—	sfrp ← word			
		AX, rp Note 3	1	1	—	AX ← rp			
		rp, AX Note 3	1	1	—	rp ← AX			
		AX, !addr16	3	1	4	AX ← (addr16)			
		!addr16, AX	3	1	—	(addr16) ← AX			
		AX, ES:!addr16	4	2	5	AX ← (ES, addr16)			
		ES:!addr16, AX	4	2	—	(ES, addr16) ← AX			
		AX, saddrp	2	1	—	AX ← (saddrp)			
		saddrp, AX	2	1	—	(saddrp) ← AX			
		AX, sfrp	2	1	—	AX ← sfrp			
		sfrp, AX	2	1	—	sfrp ← AX			

Note 1. Number of CPU clocks (fCLK) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (fCLK) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. Except rp = AX

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 9 Operation List (5/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	MOVW	AX, [DE]	1	1	4	AX ← (DE)			
		[DE], AX	1	1	—	(DE) ← AX			
		AX, ES:[DE]	2	2	5	AX ← (ES, DE)			
		ES:[DE], AX	2	2	—	(ES, DE) ← AX			
		AX, [HL]	1	1	4	AX ← (HL)			
		[HL], AX	1	1	—	(HL) ← AX			
		AX, ES:[HL]	2	2	5	AX ← (ES, HL)			
		ES:[HL], AX	2	2	—	(ES, HL) ← AX			
		AX, [DE+byte]	2	1	4	AX ← (DE + byte)			
		[DE+byte], AX	2	1	—	(DE + byte) ← AX			
		AX, ES:[DE+byte]	3	2	5	AX ← ((ES, DE) + byte)			
		ES:[DE+byte], AX	3	2	—	((ES, DE) + byte) ← AX			
		AX, [HL+byte]	2	1	4	AX ← (HL + byte)			
		[HL+byte], AX	2	1	—	(HL + byte) ← AX			
		AX, ES:[HL+byte]	3	2	5	AX ← ((ES, HL) + byte)			
		ES:[HL+byte], AX	3	2	—	((ES, HL) + byte) ← AX			
		AX, [SP+byte]	2	1	—	AX ← (SP + byte)			
		[SP+byte], AX	2	1	—	(SP + byte) ← AX			
		AX, word[B]	3	1	4	AX ← (B + word)			
		word[B], AX	3	1	—	(B + word) ← AX			
		AX, ES:word[B]	4	2	5	AX ← ((ES, B) + word)			
		ES:word[B], AX	4	2	—	((ES, B) + word) ← AX			
		AX, word[C]	3	1	4	AX ← (C + word)			
		word[C], AX	3	1	—	(C + word) ← AX			
		AX, ES:word[C]	4	2	5	AX ← ((ES, C) + word)			
		ES:word[C], AX	4	2	—	((ES, C) + word) ← AX			
		AX, word[BC]	3	1	4	AX ← (BC + word)			
		word[BC], AX	3	1	—	(BC + word) ← AX			
		AX, ES:word[BC]	4	2	5	AX ← ((ES, BC) + word)			
		ES:word[BC], AX	4	2	—	((ES, BC) + word) ← AX			

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 10 Operation List (6/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	MOVW	BC, !addr16	3	1	4	BC ← (addr16)			
		BC, ES:!addr16	4	2	5	BC ← (ES, addr16)			
		DE, !addr16	3	1	4	DE ← (addr16)			
		DE, ES:!addr16	4	2	5	DE ← (ES, addr16)			
		HL, !addr16	3	1	4	HL ← (addr16)			
		HL, ES:!addr16	4	2	5	HL ← (ES, addr16)			
		BC, saddrp	2	1	—	BC ← (saddrp)			
		DE, saddrp	2	1	—	DE ← (saddrp)			
		HL, saddrp	2	1	—	HL ← (saddrp)			
	XCHW	AX, rp Note 3	1	1	—	AX ←→ rp			
	ONEW	AX	1	1	—	AX ← 0001H			
		BC	1	1	—	BC ← 0001H			
	CLRW	AX	1	1	—	AX ← 0000H			
		BC	1	1	—	BC ← 0000H			
8-bit operation	ADD	A, #byte	2	1	—	A, CY ← A + byte	×	×	×
		saddr, #byte	3	2	—	(saddr), CY ← (saddr) + byte	×	×	×
		A, r Note 4	2	1	—	A, CY ← A + r	×	×	×
		r, A	2	1	—	r, CY ← r + A	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16)	×	×	×
		A, saddr	2	1	—	A, C ← 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. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. Except rp = AX

Note 4. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 11 Operation List (7/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	ADDC	A, #byte	2	1	—	A, CY ← A + byte + CY	×	×	×
		saddr, #byte	3	2	—	(saddr), CY ← (saddr) + byte + CY	×	×	×
		A, rv Note 3	2	1	—	A, CY ← A + r + CY	×	×	×
		r, A	2	1	—	r, CY ← r + A + CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16) + CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16) + CY	×	×	×
		A, saddr	2	1	—	A, CY ← A + (saddr) + CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A + (HL) + CY	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A + (ES, HL) + CY	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A + (HL + byte) + CY	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A + ((ES, HL) + byte) + CY	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A + (HL + B) + CY	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A + ((ES, HL) + B) + CY	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A + (HL + C) + CY	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A + ((ES, HL) + C) + CY	×	×	×
	SUB	A, #byte	2	1	—	A, CY ← A - byte	×	×	×
		saddr, #byte	3	2	—	(saddr), CY ← (saddr) - byte	×	×	×
		A, r Note 3	2	1	—	A, CY ← A - r	×	×	×
		r, A	2	1	—	r, CY ← r - A	×	×	×
		A, !addr16	3	1	4	A, CY ← A - (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A - (ES, addr16)	×	×	×
		A, saddr	2	1	—	A, CY ← A - (saddr)	×	×	×
		A, [HL]	1	1	4	A, CY ← A - (HL)	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A - (ES, HL)	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A - (HL + byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A - ((ES, HL) + byte)	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A - (HL + B)	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A - ((ES, HL) + B)	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A - (HL + C)	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A - ((ES, HL) + C)	×	×	×

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 12 Operation List (8/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	SUBC	A, #byte	2	1	—	A, CY ← A - byte - CY	×	×	×
		saddr, #byte	3	2	—	(saddr), CY ← (saddr) - byte - CY	×	×	×
		A, r Note 3	2	1	—	A, CY ← A - r - CY	×	×	×
		r, A	2	1	—	r, CY ← r - A - CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A - (addr16) - CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A - (ES, addr16) - CY	×	×	×
		A, saddr	2	1	—	A, CY ← A - (saddr) - CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A - (HL) - CY	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A - (ES, HL) - CY	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A - (HL + byte) - CY	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A - ((ES, HL) + byte) - CY	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A - (HL + B) - CY	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A - ((ES, HL) + B) - CY	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A - (HL + C) - CY	×	×	×
		A, ES:[HL+C]	3	2	5	A, CY ← A - ((ES:HL) + C) - CY	×	×	×
	AND	A, #byte	2	1	—	A ← A ∧ byte	×		
		saddr, #byte	3	2	—	(saddr) ← (saddr) ∧ byte	×		
		A, r Note 3	2	1	—	A ← A ∧ r	×		
		r, A	2	1	—	R ← r ∧ A	×		
		A, !addr16	3	1	4	A ← A ∧ (addr16)	×		
		A, ES:!addr16	4	2	5	A ← A ∧ (ES:addr16)	×		
		A, saddr	2	1	—	A ← A ∧ (saddr)	×		
		A, [HL]	1	1	4	A ← A ∧ (HL)	×		
		A, ES:[HL]	2	2	5	A ← A ∧ (ES:HL)	×		
		A, [HL+byte]	2	1	4	A ← A ∧ (HL + byte)	×		
		A, ES:[HL+byte]	3	2	5	A ← A ∧ ((ES:HL) + byte)	×		
		A, [HL+B]	2	1	4	A ← A ∧ (HL + B)	×		
		A, ES:[HL+B]	3	2	5	A ← A ∧ ((ES:HL) + B)	×		
		A, [HL+C]	2	1	4	A ← A ∧ (HL + C)	×		
		A, ES:[HL+C]	3	2	5	A ← A ∧ ((ES:HL) + C)	×		

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 13 Operation List (9/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	OR	A, #byte	2	1	—	A ← A ∨ byte	×		
		saddr, #byte	3	2	—	(saddr) ← (saddr) ∨ byte	×		
		A, r Note 3	2	1	—	A ← A ∨ r	×		
		r, A	2	1	—	r ← r ∨ A	×		
		A, !addr16	3	1	4	A ← A ∨ (addr16)	×		
		A, ES:!addr16	4	2	5	A ← A ∨ (ES:addr16)	×		
		A, saddr	2	1	—	A ← A ∨ (saddr)	×		
		A, [HL]	1	1	4	A ← A ∨ (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. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 14 Operation List (10/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	CMP	A, #byte	2	1	—	A - byte	×	×	×
		!addr16, #byte	4	1	4	(addr16) - byte	×	×	×
		ES:!addr16, #byte	5	2	5	(ES:addr16) - byte	×	×	×
		saddr, #byte	3	1	—	(saddr) - byte	×	×	×
		A, r 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. Number of CPU clocks (fCLK) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (fCLK) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. Except r = A

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 15 Operation List (11/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	ADDW	AX, #word	3	1	—	AX, CY ← AX + word	×	×	×
		AX, AX	1	1	—	AX, CY ← AX + AX	×	×	×
		AX, BC	1	1	—	AX, CY ← AX + BC	×	×	×
		AX, DE	1	1	—	AX, CY ← AX + DE	×	×	×
		AX, HL	1	1	—	AX, CY ← AX + HL	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX + (addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX + (ES:addr16)	×	×	×
		AX, saddrp	2	1	—	AX, CY ← AX + (saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX + (HL + byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX + ((ES:HL) + byte)	×	×	×
	SUBW	AX, #word	3	1	—	AX, CY ← AX - word	×	×	×
		AX, BC	1	1	—	AX, CY ← AX - BC	×	×	×
		AX, DE	1	1	—	AX, CY ← AX - DE	×	×	×
		AX, HL	1	1	—	AX, CY ← AX - HL	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX - (addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX - (ES:addr16)	×	×	×
		AX, saddrp	2	1	—	AX, CY ← AX - (saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX - (HL + byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX - ((ES:HL) + byte)	×	×	×
	CMPW	AX, #word	3	1	—	AX - word	×	×	×
		AX, BC	1	1	—	AX - BC	×	×	×
		AX, DE	1	1	—	AX - DE	×	×	×
		AX, HL	1	1	—	AX - HL	×	×	×
		AX, !addr16	3	1	4	AX - (addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX - (ES:addr16)	×	×	×
		AX, saddrp	2	1	—	AX - (saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX - (HL + byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX - ((ES:HL) + byte)	×	×	×

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 16 Operation List (12/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		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. Number of CPU clocks (fCLK) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (fCLK) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Caution Disable interrupts when executing the DIVHU or DIVWU instruction in an interrupt servicing routine. Alternatively, unless they are executed in the RAM area, note that execution of a DIVHU or DIVWU instruction is possible even with interrupts enabled as long as a NOP instruction is added immediately after the DIVHU or DIVWU instruction in the assembly language source code. The following compilers automatically add a NOP instruction immediately after any DIVHU or DIVWU instruction output during the build process.

- **V. 1.71 and later versions of the CA78K0R (Renesas Electronics compiler), for both C and assembly language source code**
- **Service pack 1.40.3 and later versions of the EWRL78 (IAR compiler), for C language source code**
- **GNURL78 (KPIT compiler), for C language source code**

Remark 1. Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Remark 2. MACR indicates the multiplication and accumulation register (MACRH, MACRL).

Table 33 - 17 Operation List (13/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Increment/ decrement	INC	r	1	1	—	r ← r + 1	×	×	
		!addr16	3	2	—	(addr16) ← (addr16) + 1	×	×	
		ES:!addr16	4	3	—	(ES, addr16) ← (ES, addr16) + 1	×	×	
		saddr	2	2	—	(saddr) ← (saddr) + 1	×	×	
		[HL+byte]	3	2	—	(HL + byte) ← (HL + byte) + 1	×	×	
		ES: [HL+byte]	4	3	—	((ES:HL) + byte) ← ((ES:HL) + byte) + 1	×	×	
	DEC	r	1	1	—	r ← r - 1	×	×	
		!addr16	3	2	—	(addr16) ← (addr16) - 1	×	×	
		ES:!addr16	4	3	—	(ES, addr16) ← (ES, addr16) - 1	×	×	
		saddr	2	2	—	(saddr) ← (saddr) - 1	×	×	
		[HL+byte]	3	2	—	(HL + byte) ← (HL + byte) - 1	×	×	
		ES: [HL+byte]	4	3	—	((ES:HL) + byte) ← ((ES:HL) + byte) - 1	×	×	
	INCW	rp	1	1	—	rp ← rp + 1			
		!addr16	3	2	—	(addr16) ← (addr16) + 1			
		ES:!addr16	4	3	—	(ES, addr16) ← (ES, addr16) + 1			
		saddrp	2	2	—	(saddrp) ← (saddrp) + 1			
		[HL+byte]	3	2	—	(HL + byte) ← (HL + byte) + 1			
		ES: [HL+byte]	4	3	—	((ES:HL) + byte) ← ((ES:HL) + byte) + 1			
	DECW	rp	1	1	—	rp ← rp - 1			
		!addr16	3	2	—	(addr16) ← (addr16) - 1			
		ES:!addr16	4	3	—	(ES, addr16) ← (ES, addr16) - 1			
		saddrp	2	2	—	(saddrp) ← (saddrp) - 1			
		[HL+byte]	3	2	—	(HL + byte) ← (HL + byte) - 1			
		ES: [HL+byte]	4	3	—	((ES:HL) + byte) ← ((ES:HL) + byte) - 1			
Shift	SHR	A, cnt	2	1	—	$(CY \leftarrow A_0, A_{m-1} \leftarrow A_m, A_7 \leftarrow 0) \times cnt$			×
	SHRW	AX, cnt	2	1	—	$(CY \leftarrow AX_0, AX_{m-1} \leftarrow AX_m, AX_{15} \leftarrow 0) \times cnt$			×
	SHL	A, cnt	2	1	—	$(CY \leftarrow A_7, A_m \leftarrow A_{m-1}, A_0 \leftarrow 0) \times cnt$			×
		B, cnt	2	1	—	$(CY \leftarrow B_7, B_m \leftarrow B_{m-1}, B_0 \leftarrow 0) \times cnt$			×
		C, cnt	2	1	—	$(CY \leftarrow C_7, C_m \leftarrow C_{m-1}, C_0 \leftarrow 0) \times cnt$			×
	SHLW	AX, cnt	2	1	—	$(CY \leftarrow AX_{15}, AX_m \leftarrow AX_{m-1}, AX_0 \leftarrow 0) \times cnt$			×
		BC, cnt	2	1	—	$(CY \leftarrow BC_{15}, BC_m \leftarrow BC_{m-1}, BC_0 \leftarrow 0) \times cnt$			×
	SAR	A, cnt	2	1	—	$(CY \leftarrow A_0, A_{m-1} \leftarrow A_m, A_7 \leftarrow A_7) \times cnt$			×
	SARW	AX, cnt	2	1	—	$(CY \leftarrow AX_0, AX_{m-1} \leftarrow AX_m, AX_{15} \leftarrow AX_{15}) \times cnt$			×

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark 1. Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Remark 2. cnt indicates the bit shift count.

Table 33 - 18 Operation List (14/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Rotate	ROR	A, 1	2	1	—	(CY, $A_7 \leftarrow A_0$, $A_{m-1} \leftarrow A_m$) × 1			×
	ROL	A, 1	2	1	—	(CY, $A_0 \leftarrow A_7$, $A_{m+1} \leftarrow A_m$) × 1			×
	RORC	A, 1	2	1	—	(CY $\leftarrow A_0$, $A_7 \leftarrow$ CY, $A_{m-1} \leftarrow A_m$) × 1			×
	ROLC	A, 1	2	1	—	(CY $\leftarrow A_7$, $A_0 \leftarrow$ CY, $A_{m+1} \leftarrow A_m$) × 1			×
	ROLWC	AX,1	2	1	—	(CY $\leftarrow AX_{15}$, $AX_0 \leftarrow$ CY, $AX_{m+1} \leftarrow AX_m$) × 1			×
		BC,1	2	1	—	(CY $\leftarrow BC_{15}$, $BC_0 \leftarrow$ CY, $BC_{m+1} \leftarrow BC_m$) × 1			×
Bit manipulate	MOV1	CY, A.bit	2	1	—	CY ← A.bit			×
		A.bit, CY	2	1	—	A.bit ← CY			
		CY, PSW.bit	3	1	—	CY ← PSW.bit			×
		PSW.bit, CY	3	4	—	PSW.bit ← CY	×	×	
		CY, saddr.bit	3	1	—	CY ← (saddr).bit			×
		saddr.bit, CY	3	2	—	(saddr).bit ← CY			
		CY, sfr.bit	3	1	—	CY ← sfr.bit			×
		sfr.bit, CY	3	2	—	sfr.bit ← CY			
		CY,[HL].bit	2	1	4	CY ← (HL).bit			×
		[HL].bit, CY	2	2	—	(HL).bit ← CY			
		CY, ES:[HL].bit	3	2	5	CY ← (ES, HL).bit			×
		ES:[HL].bit, CY	3	3	—	(ES, HL).bit ← CY			
	AND1	CY, A.bit	2	1	—	CY ← CY ∧ A.bit			×
		CY, PSW.bit	3	1	—	CY ← CY ∧ PSW.bit			×
		CY, saddr.bit	3	1	—	CY ← CY ∧ (saddr).bit			×
		CY, sfr.bit	3	1	—	CY ← CY ∧ sfr.bit			×
		CY,[HL].bit	2	1	4	CY ← CY ∧ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∧ (ES, HL).bit			×
	OR1	CY, A.bit	2	1	—	CY ← CY ∨ A.bit			×
		CY, PSW.bit	3	1	—	CY ← CY ∨ PSW.bit			×
		CY, saddr.bit	3	1	—	CY ← CY ∨ (saddr).bit			×
		CY, sfr.bit	3	1	—	CY ← CY ∨ sfr.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ∨ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∨ (ES, HL).bit			×

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 19 Operation List (15/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Bit manipulate	XOR1	CY, A.bit	2	1	—	CY ← CY ⊻ 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. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 20 Operation List (16/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Call/return	CALL	rp	2	3	—	(SP - 2) ← $(PC + 2)_S$, (SP - 3) ← $(PC + 2)_H$, (SP - 4) ← $(PC + 2)_L$, PC ← CS, rp, SP ← SP - 4			
		$!addr20	3	3	—	(SP - 2) ← $(PC + 3)_S$, (SP - 3) ← $(PC + 3)_H$, (SP - 4) ← $(PC + 3)_L$, PC ← PC + 3 + jdisp16, SP ← SP - 4			
		!addr16	3	3	—	(SP - 2) ← $(PC + 3)_S$, (SP - 3) ← $(PC + 3)_H$, (SP - 4) ← $(PC + 3)_L$, PC ← 0000, addr16, SP ← SP - 4			
		!!addr20	4	3	—	(SP - 2) ← $(PC + 4)_S$, (SP - 3) ← $(PC + 4)_H$, (SP - 4) ← $(PC + 4)_L$, PC ← addr20, SP ← SP - 4			
	CALLT	[addr5]	2	5	—	(SP - 2) ← $(PC + 2)_S$, (SP - 3) ← $(PC + 2)_H$, (SP - 4) ← $(PC + 2)_L$, PC_S ← 0000, PC_H ← (0000, addr5 + 1), PC_L ← (0000, addr5), SP ← SP - 4			
	BRK	—	2	5	—	(SP - 1) ← PSW, (SP - 2) ← $(PC + 2)_S$, (SP - 3) ← $(PC + 2)_H$, (SP - 4) ← $(PC + 2)_L$, PC_S ← 0000, PC_H ← (0007FH), PC_L ← (0007EH), SP ← SP - 4, IE ← 0			
	RET	—	1	6	—	PC_L ← (SP), PC_H ← (SP + 1), PC_S ← (SP + 2), SP ← SP + 4			
	RETI	—	2	6	—	PC_L ← (SP), PC_H ← (SP + 1), PC_S ← (SP + 2), PSW ← (SP + 3), SP ← SP + 4	R	R	R
	RETB	—	2	6	—	PC_L ← (SP), PC_H ← (SP + 1), PC_S ← (SP + 2), PSW ← (SP + 3), SP ← SP + 4	R	R	R

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 21 Operation List (17/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Stack manipulate	PUSH	PSW	2	1	—	(SP - 1) ← PSW, (SP - 2) ← 00H, SP ← SP - 2			
		rp	1	1	—	(SP - 1) ← rp_H, (SP - 2) ← rp_L, SP ← SP - 2			
	POP	PSW	2	3	—	PSW ← (SP + 1), SP ← SP + 2	R	R	R
		rp	1	1	—	rp_L ← (SP), rp_H ← (SP + 1), SP ← SP + 2			
	MOVW	SP, #word	4	1	—	SP ← word			
		SP, AX	2	1	—	SP ← AX			
		AX, SP	2	1	—	AX ← SP			
		HL, SP	3	1	—	HL ← SP			
		BC, SP	3	1	—	BC ← SP			
		DE, SP	3	1	—	DE ← SP			
	ADDW	SP, #byte	2	1	—	SP ← SP + byte			
	SUBW	SP, #byte	2	1	—	SP ← SP - byte			
Unconditional branch	BR	AX	2	3	—	PC ← CS, AX			
		$addr20	2	3	—	PC ← PC + 2 + jdisp8			
		$!addr20	3	3	—	PC ← PC + 3 + jdisp16			
		!addr16	3	3	—	PC ← 0000, addr16			
		!!addr20	4	3	—	PC ← addr20			
Conditional branch	BC	$addr20	2	2/4 Note 3	—	PC ← PC + 2 + jdisp8 if CY = 1			
	BNC	$addr20	2	2/4 Note 3	—	PC ← PC + 2 + jdisp8 if CY = 0			
	BZ	$addr20	2	2/4 Note 3	—	PC ← PC + 2 + jdisp8 if Z = 1			
	BNZ	$addr20	2	2/4 Note 3	—	PC ← PC + 2 + jdisp8 if Z = 0			
	BH	$addr20	3	2/4 Note 3	—	PC ← PC + 3 + jdisp8 if (Z ∨ CY) = 0			
	BNH	$addr20	3	2/4 Note 3	—	PC ← PC + 3 + jdisp8 if (Z ∨ CY) = 1			
	BT	saddr.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 1			
		sfr.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
		A.bit, $addr20	3	3/5 Note 3	—	PC ← PC + 3 + jdisp8 if A.bit = 1			
		PSW.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if PSW.bit = 1			
		[HL].bit, $addr20	3	3/5 Note 3	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 1			
		ES:[HL].bit, $addr20	4	4/6 Note 3	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1			

Note 1. Number of CPU clocks (f_{CLK}) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (f_{CLK}) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. This indicates the number of clocks “when condition is not met/when condition is met”.

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

Table 33 - 22 Operation List (18/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Conditional branch	BF	saddr.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 0			
		sfr.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
		A.bit, $addr20	3	3/5 Note 3	—	PC ← PC + 3 + jdisp8 if A.bit = 0			
		PSW.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if PSW.bit = 0			
		[HL].bit, $addr20	3	3/5 Note 3	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 0			
		ES:[HL].bit, $addr20	4	4/6 Note 3	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 0			
	BTCLR	saddr.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 1 then reset (saddr).bit			
		sfr.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit			
		A.bit, $addr20	3	3/5 Note 3	—	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit			
		PSW.bit, $addr20	4	3/5 Note 3	—	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	×	×	×
		[HL].bit, $addr20	3	3/5 Note 3	—	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit			
		ES:[HL].bit, $addr20	4	4/6 Note 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 Note 4	RBn	2	1	—	RBS[1:0] ← n			
	NOP	—	1	1	—	No Operation			
	EI	—	3	4	—	IE ← 1 (Enable Interrupt)			
	DI	—	3	4	—	IE ← 0 (Disable Interrupt)			
	HALT	—	2	3	—	Set HALT Mode			
	STOP	—	2	3	—	Set STOP Mode			

Note 1. Number of CPU clocks (fCLK) when the internal RAM area, SFR area, or extended SFR area is accessed, or when no data is accessed.

Note 2. Number of CPU clocks (fCLK) when the code flash memory is accessed, or when the data flash memory is accessed by an 8-bit instruction.

Note 3. This indicates the number of clocks “when condition is not met/when condition is met”.

Note 4. n indicates the number of register banks (n = 0 to 3)

Remark Number of clock is when program exists in the internal ROM (flash memory) area. If fetching the instruction from the internal RAM area, the number becomes double number plus 3 clocks at a maximum.

CHAPTER 34 ELECTRICAL SPECIFICATIONS

Caution 1. The RL78 microcontrollers have an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

Caution 2. The pins mounted depend on the product. Refer to 2.1 Port Functions to 2.2.1 Functions for each product.

Caution 3. Please contact Renesas Electronics sales office for derating of operation under T_A = +85 to +105°C. Derating is the systematic reduction of load for the sake of improved reliability.

Caution 4. When operating temperature exceeds 85°C, only HS (high-speed main) mode can be used as the flash operation mode. Regulator mode should be used with the normal setting (MCSEL = 0).

34.1 Absolute Maximum Ratings

Absolute Maximum Ratings **(1/2)**

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}, AV_{DD}	V_{DD} = AV_{DD}	-0.3 to + 4.6	V
	AV_{REFP}		0.3 to AV_{DD} + 0.3 Note 2	V
	AV_{SS}		-0.5 to + 0.3	V
	AV_{REFM}		-0.3 to AV_{DD} + 0.3 Note 2 and $AV_{REFM} \leq AV_{REFP}$	V
REGC pin input voltage	V_{IREGC}	REGC	-0.3 to + 2.8 and -0.3 to V_{DD} + 0.3 Note 1	V
Input voltage	V_{I1}	P00 to P04, P30 to P33, P40, P50 to P57, P121 to P124, P130, P137, EXCLK, EXCLKS, $\overline{RESET}$	-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}	P10 to P17, P20 to P25	-0.3 to AV_{DD} + 0.3 Note 2	V
Output voltage	V_{O1}	P00 to P04, P30 to P33, P40, P50 to P57, P60 to P63, P130	-0.3 to V_{DD} + 0.3 Note 2	V
	V_{O2}	P10 to P17, P20 to P25	-0.3 to AV_{DD} + 0.3 Note 2	V
Analog input voltage	V_{AI1}	ANI16 to ANI18	-0.3 to V_{DD} + 0.3 and -0.3 to $AV_{REF}(+)$ + 0.3 Notes 2, 3	V
	V_{AI2}	ANI0 to ANI13	-0.3 to AV_{DD} + 0.3 and -0.3 to $AV_{REF}(+)$ + 0.3 Notes 2, 3	V
	V_{AI3}	Operational amplifier input pin	-0.3 to AV_{DD} + 0.3 Note 2	V

Note 1. Connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF). This value regulates the absolute maximum rating of the REGC pin. Do not use this pin with voltage applied to it.

Note 2. Must be 4.6 V or lower.

Note 3. Do not exceed AV_{REF} (+) + 0.3 V in case of A/D conversion target pin.

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. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

Remark 2. AV_{REF} (+): + side reference voltage of the A/D converter.

Remark 3. V_{SS}: Reference voltage

Absolute Maximum Ratings **(2/2)**

Parameter	Symbols	Conditions		Ratings	Unit
Output current, high	I_{OH1}	Per pin	P00 to P04, P30 to P33, P40, P50 to P57, P130	-40	mA
		Total of all pins -170 mA	P00 to P04, P40, P130	-70	mA
			P30 to P33, P50 to P57	-100	mA
	I_{OH2}	Per pin	P10 to P17, P20 to P25	-0.1	mA
		Total of all pins		-1.4	mA
Output current, low	I_{OL1}	Per pin	P00 to P04, P30 to P33, P40, P50 to P57, P60 to P63, P130	40	mA
		Total of all pins 170 mA	P00 to P04, P40, P130	70	mA
			P30 to P33, P50 to P57, P60 to P63	100	mA
	I_{OL2}	Per pin	P10 to P17, P20 to P25	0.4	mA
		Total of all pins		5.6	mA
Operating ambient temperature	T_A	In normal operation mode		-40 to +105	°C
		In flash memory programming mode			
Storage temperature	T_{stg}			-65 to +150	°C

Caution **Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.**

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

34.2 Oscillator Characteristics

34.2.1 X1, XT1 characteristics

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)
(T_A = +85 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

Resonator	Resonator	Conditions	MIN.	TYP.	MAX.	Unit
X1 clock oscillation frequency (f_X) Note	Ceramic resonator/ crystal resonator	2.7 V ≤ V_{DD} ≤ 3.6 V	1.0		20.0	MHz
		2.4 V ≤ V_{DD} < 2.7 V	1.0		16.0	
		1.8 V ≤ V_{DD} < 2.4 V	1.0		8.0	
		1.6 V ≤ V_{DD} < 1.8 V	1.0		4.0	
XT1 clock oscillation frequency (f_{XT}) Note	Crystal resonator		32	32.768	35	kHz

Note Indicates only permissible oscillator frequency ranges. Refer to **AC Characteristics** for instruction execution time. Request evaluation by the manufacturer of the oscillator circuit mounted on a board to check the oscillator characteristics.

Caution Since the CPU is started by the high-speed on-chip oscillator clock after a reset release, check the X1 clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) by the user. Determine the oscillation stabilization time of the OSTC register and the oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.

Remark When using the X1 oscillator and XT1 oscillator, refer to **6.4 System Clock Oscillator**.

34.2.2 On-chip oscillator characteristics

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)
(T_A = +85 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

Oscillators	Parameters	Conditions		MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator clock frequency Notes 1, 2	f_{IH}			1		24	MHz
High-speed on-chip oscillator clock frequency accuracy		-20 to +85°C	1.8 V ≤ V_{DD} ≤ 3.6 V	-1.0		+1.0	%
			1.6 V ≤ V_{DD} < 1.8 V	-5.0		+5.0	
		-40 to -20°C	1.8 V ≤ V_{DD} ≤ 3.6 V	-1.5		+1.5	%
			1.6 V ≤ V_{DD} < 1.8 V	-5.5		+5.5	
		+85 to +105°C	2.4 V ≤ V_{DD} ≤ 3.6 V	-2.0		+2.0	%
Middle-speed on-chip oscillator oscillation frequency Note 2	f_{IM}			1		4	MHz
Middle-speed on-chip oscillator oscillation frequency accuracy		1.8V ≤ V_{DD} ≤ 3.6V		-12		+12	%
Low-speed on-chip oscillator clock frequency Note 2	f_{IL}				15		kHz
Low-speed on-chip oscillator clock frequency accuracy				-15		+15	%

Note 1. High-speed on-chip oscillator frequency is selected with bits 0 to 3 of the option byte (000C2H) and bits 0 to 2 of the HOCODIV register.

Note 2. This only indicates the oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

34.3 DC Characteristics

34.3.1 Pin characteristics

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)
(T_A = +85 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V) **(1/5)**

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high Note 1	I_{OH1}	Per pin for P00 to P04, P30 to P33, P40, P50 to P57, P130	T_A = -40 to +85°C			-10.0 Note 2	mA
			T_A = +85 to +105°C			-3.0 Note 2	mA
		Total of P00 to P04, P40, P130 (When duty ≤ 70% Note 3)	2.7 V ≤ V_{DD} ≤ 3.6 V			-10.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			-5.0	mA
			1.6 V ≤ V_{DD} < 1.8 V			-2.5	mA
		Total of P30 to P33, P50 to P57 (When duty ≤ 70% Note 3)	2.7 V ≤ V_{DD} ≤ 3.6 V			-19.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			-10.0	mA
			1.6 V ≤ V_{DD} < 1.8 V			-5.0	mA
		Total of all pins (When duty ≤ 70% Note 3)				-29.0	mA
	I_{OH2}	Per pin for P10 to P17, P20 to P25				-0.1 Note 2	mA
		Total of all pins (When duty ≤ 70% Note 3)	1.6 V ≤ V_{DD} ≤ 3.6 V			-1.4	mA

Note 1. Value of current at which the device operation is guaranteed even if the current flows from the V_{DD} pin to an output pin.

Note 2. Do not exceed the total current value.

Note 3. Specification under conditions where the duty factor ≤ 70%.
The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).

- Total output current of pins = (I_{OH} × 0.7)/(n × 0.01)
  <Example> Where n = 80% and I_{OH} = -10.0 mA
  Total output current of pins = (-10.0 × 0.7)/(80 × 0.01) ≈ -8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor.
A current higher than the absolute maximum rating must not flow into one pin.

Caution **P30 and P51 to P56 do not output high level in N-ch open-drain mode.**

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

(T_A = +85 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V) **(2/5)**

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low Note 1	I_{OL1}	Per pin for P00 to P04, P30 to P33, P40, P50 to P57, P130	T_A = -40 to +85°C			20.0 Note 2	mA
			T_A = +85 to +105°C			8.5 Note 2	mA
		Per pin for P60 to P63				15.0 Note 2	mA
		Total of P00 to P04, P40, P130 (When duty ≤ 70% Note 3)	2.7 V ≤ V_{DD} ≤ 3.6 V			15.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			9.0	mA
			1.6 V ≤ V_{DD} < 1.8 V			4.5	mA
		Total of P30 to P33, P50 to P57, P60 to P63 (When duty ≤ 70% Note 3)	2.7 V ≤ V_{DD} ≤ 3.6 V			35.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			20.0	mA
			1.6 V ≤ V_{DD} < 1.8 V			10.0	mA
		Total of all pins (When duty ≤ 70% Note 3)				50.0	mA
	I_{OL2}	Per pin for P10 to P17, P20 to P25				0.4 Note 2	mA
		Total of all pins (When duty ≤ 70% Note 3)	1.6 V ≤ V_{DD} ≤ 3.6 V			5.6	mA

Note 1. Value of current at which the device operation is guaranteed even if the current flows from an output pin to the V_{SS} pin.

Note 2. Do not exceed the total current value.

Note 3. Specification under conditions where the duty factor ≤ 70%.

The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).

- Total output current of pins = (I_{OL} × 0.7)/(n × 0.01)

  <Example> Where n = 80% and I_{OL} = 10.0 mA

  Total output current of pins = (10.0 × 0.7)/(80 × 0.01) ≈ 8.7 mA

  However, the current that is allowed to flow into one pin does not vary depending on the duty factor.

  A current higher than the absolute maximum rating must not flow into one pin.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)
(T_A = +85 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V) **(3/5)**

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P00 to P04, P30 to P33, P40, P50 to P57, P130	Normal input buffer	0.8 V_{DD}		V_{DD}	V
	V_{IH2}	P30, P32, P33, P51, P52, P54 to P57	TTL input buffer 3.3 V ≤ V_{DD} ≤ 3.6 V	2.0		V_{DD}	V
			TTL input buffer 1.6 V ≤ V_{DD} < 3.3 V	1.5		V_{DD}	V
	V_{IH3}	P10 to P17, P20 to P25		0.7 AV_{DD}		AV_{DD}	V
	V_{IH4}	P60 to P63		0.7 V_{DD}		6.0	V
	V_{IH5}	P121 to P124, P137, EXCLK, EXCLKS, $\overline{\text{RESET}}$		0.8 V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	P00 to P04, P30 to P33, P40, P50 to P57, P130	Normal input buffer	0		0.2 V_{DD}	V
	V_{IL2}	P30, P32, P33, P51, P52, P54 to P57	TTL input buffer 3.3 V ≤ V_{DD} ≤ 3.6 V	0		0.5	V
			TTL input buffer 1.6 V ≤ V_{DD} < 3.3 V	0		0.32	V
	V_{IL3}	P10 to P17, P20 to P25		0		0.3 AV_{DD}	V
	V_{IL4}	P60 to P63		0		0.3 V_{DD}	V
	V_{IL5}	P121 to P124, P137, EXCLK, EXCLKS, $\overline{\text{RESET}}$		0		0.2 V_{DD}	V

Caution **The maximum value of V_{IH} of pins P30 and P51 to P56 is V_{DD}, even in the N-ch open-drain mode.**

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)
(T_A = +85 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V) **(4/5)**

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, high	V_{OH1}	P00 to P04, P30 to P33, P40, P50 to P57, P130	2.7 V ≤ V_{DD} ≤ 3.6 V, I_{OH} = -2.0 mA	V_{DD} - 0.6			V
			1.8 V ≤ V_{DD} ≤ 3.6 V Note 3, I_{OH} = -1.5 mA	V_{DD} - 0.5			V
			1.6 V ≤ V_{DD} ≤ 3.6 V Note 1, I_{OH} = -1.0 mA	V_{DD} - 0.5			V
	V_{OH2}	P10 to P17, P20 to P25	1.6 V ≤ AV_{DD} ≤ 3.6 V Note 2, I_{OH} = -100 μA	AV_{DD} - 0.5			V
Output voltage, low	V_{OL1}	P00 to P04, P30 to P33, P40, P50 to P57, P130	2.7 V ≤ V_{DD} ≤ 3.6 V, I_{OL} = 3.0 mA			0.6	V
			2.7 V ≤ V_{DD} ≤ 3.6 V, I_{OL} = 1.5 mA			0.4	V
			1.8 V ≤ V_{DD} ≤ 3.6 V Note 3, I_{OL} = 0.6 mA			0.4	V
			1.6 V ≤ AV_{DD} ≤ 3.6 V Note 1, I_{OL} = 0.3 mA			0.4	V
	V_{OL2}	P10 to P17, P20 to P25	1.6 V ≤ AV_{DD} ≤ 3.6 V Note 2, I_{OL} = 400 μA			0.4	V
	V_{OL3}	P60 to P63	2.7 V ≤ V_{DD} ≤ 3.6 V, I_{OL} = 3.0 mA			0.4	V
			1.8 V ≤ V_{DD} ≤ 3.6 V Note 3, I_{OL} = 2.0 mA			0.4	V
			1.6 V ≤ AV_{DD} ≤ 3.6 V Note 1, I_{OL} = 1.0 mA			0.4	V

Note 1. Only T_A = -40 to +85°C is guaranteed.
Note 2. The condition that 2.4 V ≤ AV_{DD} ≤ 3.6 V is guaranteed when +85°C < T_A ≤ +105°C.
Note 3. The condition that 2.4 V ≤ V_{DD} ≤ 3.6 V is guaranteed when +85°C < T_A ≤ +105°C.

Caution P30 and P51 to P56 do not output high level in N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)
(T_A = +85 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V) **(5/5)**

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P00 to P04, P30 to P33, P40, P50 to P57, P60 to P63, P130, P137	$V_I = V_{DD}$				1	μA
	I_{LIH2}	$\overline{\text{RESET}}$	$V_I = V_{DD}$				1	μA
	I_{LIH3}	P121 to P124 (X1, X2, EXCLK, XT1, XT2, EXCLKS)	$V_I = V_{DD}$	In input port or external clock input			1	μA
				In resonator connection			10	μA
	I_{LIH4}	P10 to P17, P20 to P25	$V_I = AV_{DD}$				1	μA
Input leakage current, low	I_{LIL1}	P00 to P04, P30 to P33, P40, P50 to P57, P60 to P63, P130, P137	$V_I = V_{SS}$				-1	μA
	I_{LIL2}	$\overline{\text{RESET}}$	$V_I = V_{SS}$				-1	μA
	I_{LIL3}	P121 to P124 (X1, X2, EXCLK, XT1, XT2, EXCLKS)	$V_I = V_{SS}$	In input port or external clock input			-1	μA
				In resonator connection			-10	μA
	I_{LIL4}	P10 to P17, P20 to P25	$V_I = AV_{SS}$				-1	μA
On-chip pull-up resistance	R_U	P00 to P04, P30 to P33, P40, P50 to P57, P130	$V_I = V_{SS}$, In input port		10	20	100	kΩ

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

34.3.2 Supply current characteristics

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)
(T_A = +85 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V) **(1/4)**

Parameter	Symbol	Conditions						MIN.	TYP.	MAX.	Unit
Supply current [Note 1]	I_{DD1}	Operating mode	HS (high-speed main) mode	f_{IH} = 24 MHz [Note 3], T_A = -40 to +105°C	Basic operation	V_{DD} = 3.0 V			1.4		mA
			HS (high-speed main) mode	f_{IH} = 24 MHz [Note 3], T_A = -40 to +85°C	Normal operation	V_{DD} = 3.0 V			3.2	6.3	mA
				f_{IH} = 24 MHz [Note 3], T_A = +85 to +105°C	Normal operation	V_{DD} = 3.0 V				6.7	
				f_{IH} = 16 MHz [Note 3], T_A = -40 to +85°C	Normal operation	V_{DD} = 3.0 V			2.4	4.6	
				f_{IH} = 16 MHz [Note 3], T_A = +85 to +105°C	Normal operation	V_{DD} = 3.0 V				4.9	
			LS (low-speed main) mode (MCSEL = 0)	f_{IH} = 8 MHz [Note 3], T_A = -40 to +85°C	Normal operation	V_{DD} = 3.0 V			1.1	2.0	mA
						V_{DD} = 2.0 V			1.1	2.0	
			LS (low-speed main) mode (MCSEL = 1)	f_{IH} = 4 MHz [Note 3], T_A = -40 to +85°C	Normal operation	V_{DD} = 3.0 V			0.72	1.30	mA
						V_{DD} = 2.0 V			0.72	1.30	
				f_{IM} = 4 MHz [Note 7], T_A = -40 to +85°C	Normal operation	V_{DD} = 3.0 V			0.58	1.10	
						V_{DD} = 2.0 V			0.58	1.10	
			LV (low-voltage main) mode	f_{IH} = 3 MHz [Note 3], T_A = -40 to +85°C	Normal operation	V_{DD} = 3.0 V			1.2	1.8	mA
						V_{DD} = 2.0 V			1.2	1.8	
			LP (low-power main) mode [Note 5] (MCSEL = 1)	f_{IH} = 1 MHz [Note 3], T_A = -40 to +85°C	Normal operation	V_{DD} = 3.0 V			290	480	μA
						V_{DD} = 2.0 V			290	480	
				f_{IM} = 1 MHz [Note 5], T_A = -40 to +85°C	Normal operation	V_{DD} = 3.0 V			124	230	
						V_{DD} = 2.0 V			124	230	
			HS (high-speed main) mode	f_{MX} = 20 MHz [Note 2], T_A = -40 to +85°C	Normal operation	V_{DD} = 3.0 V	Square wave input		2.7	5.3	mA
							Resonator connection		2.8	5.5	
				f_{MX} = 20 MHz [Note 2], T_A = +85 to +105°C	Normal operation	V_{DD} = 3.0 V	Square wave input			5.7	
							Resonator connection			5.8	
				f_{MX} = 10 MHz [Note 2], T_A = -40 to +85°C	Normal operation	V_{DD} = 3.0 V	Square wave input		1.8	3.1	
							Resonator connection		1.9	3.2	
				f_{MX} = 10 MHz [Note 2], T_A = +85 to +105°C	Normal operation	V_{DD} = 3.0 V	Square wave input			3.4	
							Resonator connection			3.5	
			LS (low-speed main) mode (MCSEL = 0)	f_{MX} = 8 MHz [Note 2], T_A = -40 to +85°C	Normal operation	V_{DD} = 3.0 V	Square wave input		0.9	1.9	mA
							Resonator connection		1.0	2.0	
				f_{MX} = 8 MHz [Note 2], T_A = -40 to +85°C	Normal operation	V_{DD} = 2.0 V	Square wave input		0.9	1.9	
							Resonator connection		1.0	2.0	
			LS (low-speed main) mode (MCSEL = 1)	f_{MX} = 4 MHz [Note 2], T_A = -40 to +85°C	Normal operation	V_{DD} = 3.0 V	Square wave input		0.6	1.1	mA
							Resonator connection		0.6	1.2	
				f_{MX} = 4 MHz [Note 2], T_A = -40 to +85°C	Normal operation	V_{DD} = 2.0 V	Square wave input		0.6	1.1	
							Resonator connection		0.6	1.2	
			LP (low-power main) mode (MCSEL = 1)	f_{MX} = 1 MHz [Note 2], T_A = -40 to +85°C	Normal operation	V_{DD} = 3.0 V	Square wave input		100	190	μA
							Resonator connection		136	250	
				f_{MX} = 1 MHz [Note 2], T_A = -40 to +85°C	Normal operation	V_{DD} = 2.0 V	Square wave input		100	190	
							Resonator connection		136	250	

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

(TA = -40 to +85°C, 1.6 V ≤ AVDD = VDD ≤ 3.6 V, VSS = AVSS = 0 V)
(TA = +85 to +105°C, 2.4 V ≤ AVDD = VDD ≤ 3.6 V, VSS = AVSS = 0 V) **(2/4)**

Parameter	Symbol	Conditions					MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD1}	Operating mode	Subsystem clock operation	f_{SX} = 32.768 kHz, T_A = -40°C Note 4	Normal operation	Square wave input		3.2	6.1	μA
						Resonator connection		3.3	6.1	
				f_{SX} = 32.768 kHz, T_A = +25°C Note 4	Normal operation	Square wave input		3.4	6.1	
						Resonator connection		3.6	6.1	
				f_{SX} = 32.768 kHz, T_A = +50°C Note 4	Normal operation	Square wave input		3.5	6.7	
						Resonator connection		3.7	6.7	
				f_{SX} = 32.768 kHz, T_A = +70°C Note 4	Normal operation	Square wave input		3.7	7.5	
						Resonator connection		3.9	7.5	
				f_{SX} = 32.768 kHz, T_A = +85°C Note 4	Normal operation	Square wave input		4.0	8.9	
						Resonator connection		4.2	8.9	
				f_{SX} = 32.768 kHz, T_A = +105°C Note 4	Normal operation	Square wave input		4.5	21.0	
						Resonator connection		4.7	21.1	
				f_{IL} = 15 kHz, T_A = -40°C Note 6	Normal operation			1.8	5.9	
				f_{IL} = 15 kHz, T_A = +25°C Note 6	Normal operation			1.9	5.9	
				f_{IL} = 15 kHz, T_A = +85°C Note 6	Normal operation			2.3	8.7	
				f_{IL} = 15 kHz, T_A = +105°C Note 6	Normal operation			3.0	20.9	

<R> **Note 1.** Total current flowing into VDD, including the input leakage current flowing when the level of the input pin is fixed to VDD or VSS. The following points apply in the HS (high-speed main), LS (low-speed main), LV (low-voltage 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 peripheral modules, except for those flowing into the A/D converter, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors, and those flowing while the data flash memory is being rewritten.

In the subsystem clock operation, the currents in both the "TYP." and "MAX." columns do not include the operating currents of the peripheral modules. However, in HALT mode, including the current flowing into the real-time clock 2.

Note 2. When the high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock, low-speed on-chip oscillator clock, and sub clock are stopped.

Note 3. When the high-speed system clock, middle-speed on-chip oscillator clock, low-speed on-chip oscillator clock, and sub clock are stopped.

<R> **Note 4.** When the high-speed system clock, high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock, low-speed on-chip oscillator clock, and sub clock are stopped. When ultra-low-power consumption oscillation is set (AMPHS1, AMPHS0) = (1, 0).

<R> **Note 5.** When the high-speed system clock, high-speed on-chip oscillator clock, sub clock, and low-speed on-chip oscillator clock are stopped.

Note 6. When the high-speed system clock, high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock, and sub clock are stopped.

Note 7. When the high-speed system clock, high-speed on-chip oscillator clock, low-speed on-chip oscillator clock, and sub clock are stopped.

Remark 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)
Remark 2. f_{IH}: High-speed on-chip oscillator clock frequency (24 MHz max.)
Remark 3. f_{IM}: Middle-speed on-chip oscillator clock frequency (4 MHz max.)
Remark 4. f_{IL}: Low-speed on-chip oscillator clock frequency
Remark 5. f_{SX}: Sub clock frequency (XT1 clock oscillation frequency)
Remark 6. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency or low-speed on-chip oscillator clock frequency)
Remark 7. Except subsystem clock operation, temperature condition of the TYP. value is T_A = 25°C.

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)
(T_A = +85 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V) **(3/4)**

Parameter	Symbol	Conditions					MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD2} Note 2	HALT mode	HS (high-speed main) mode	f_{IH} = 24 MHz Note 4, T_A = -40 to +85°C	V_{DD} = 3.0 V			0.37	1.83	mA
				f_{IH} = 24 MHz Note 4, T_A = +85 to +105°C	V_{DD} = 3.0 V				2.85	
				f_{IH} = 16 MHz Note 4, T_A = -40 to +85°C	V_{DD} = 3.0 V			0.36	1.38	
				f_{IH} = 16 MHz Note 4, T_A = +85 to +105°C	V_{DD} = 3.0 V				2.08	
			LS (low-speed main) mode (MCSEL = 0)	f_{IH} = 8 MHz Note 4, T_A = -40 to +85°C	V_{DD} = 3.0 V			250	710	µA
					V_{DD} = 2.0 V			250	710	
			LS (low-speed main) mode (MCSEL = 1)	f_{IH} = 4 MHz Note 4, T_A = -40 to +85°C	V_{DD} = 3.0 V			204	400	µA
					V_{DD} = 2.0 V			204	400	
				f_{IM} = 4 MHz Note 7, T_A = -40 to +85°C	V_{DD} = 3.0 V			40	250	
					V_{DD} = 2.0 V			40	250	
			LV (low-voltage main) mode	f_{IH} = 3 MHz Note 4, T_A = -40 to +85°C	V_{DD} = 3.0 V			425	800	µA
					V_{DD} = 2.0 V			425	800	
			LP (low-power main) mode (MCSEL = 1)	f_{IH} = 1 MHz Note 4, T_A = -40 to +85°C	V_{DD} = 3.0 V			192	400	µA
					V_{DD} = 2.0 V			192	400	
				f_{IM} = 1 MHz Note 7, T_A = -40 to +85°C	V_{DD} = 3.0 V			27	100	
					V_{DD} = 2.0 V			27	100	
			HS (high-speed main) mode	f_{MX} = 20 MHz Note 3, T_A = -40 to +85°C	V_{DD} = 3.0 V	Square wave input		0.20	1.55	mA
						Resonator connection		0.40	1.74	
				f_{MX} = 20 MHz Note 3, T_A = +85 to +105°C	V_{DD} = 3.0 V	Square wave input			2.45	
						Resonator connection			2.57	
				f_{MX} = 10 MHz Note 3, T_A = -40 to +85°C	V_{DD} = 3.0 V	Square wave input		0.15	0.86	
						Resonator connection		0.30	0.93	
				f_{MX} = 10 MHz Note 3, T_A = +85 to +105°C	V_{DD} = 3.0 V	Square wave input			1.28	
						Resonator connection			1.36	
			LS (low-speed main) mode (MCSEL = 0)	f_{MX} = 8 MHz Note 3, T_A = -40 to +85°C	V_{DD} = 3.0 V	Square wave input		68	550	µA
						Resonator connection		120	590	
				f_{MX} = 8 MHz Note 3, T_A = -40 to +85°C	V_{DD} = 2.0 V	Square wave input		68	550	
						Resonator connection		120	590	
			LS (low-speed main) mode (MCSEL = 1)	f_{MX} = 4 MHz Note 3, T_A = -40 to +85°C	V_{DD} = 3.0 V	Square wave input		23	128	µA
						Resonator connection		65	200	
				f_{MX} = 1 MHz Note 3, T_A = -40 to +85°C	V_{DD} = 2.0 V	Square wave input		23	128	
						Resonator connection		65	200	
			LP (low-power main) mode (MCSEL = 1)	f_{MX} = 4 MHz Note 3, T_A = -40 to +85°C	V_{DD} = 3.0 V	Square wave input		10	64	µA
						Resonator connection		48	150	
				f_{MX} = 1 MHz Note 3, T_A = -40 to +85°C	V_{DD} = 2.0 V	Square wave input		10	64	
						Resonator connection		48	150	
			Subsystem clock operation	f_{SX} = 32.768 kHz, T_A = -40°C Note 5		Square wave input		0.24	0.57	µA
						Resonator connection		0.42	0.76	
				f_{SX} = 32.768 kHz, T_A = +25°C Note 5		Square wave input		0.30	0.57	
						Resonator connection		0.54	0.76	
				f_{SX} = 32.768 kHz, T_A = +50°C Note 5		Square wave input		0.35	1.17	
						Resonator connection		0.60	1.36	
				f_{SX} = 32.768 kHz, T_A = +70°C Note 5		Square wave input		0.42	1.97	
						Resonator connection		0.70	2.16	
				f_{SX} = 32.768 kHz, T_A = +85°C Note 5		Square wave input		0.80	3.37	
						Resonator connection		0.95	3.56	
				f_{SX} = 32.768 kHz, T_A = +105°C Note 5		Square wave input		1.80	17.10	
						Resonator connection		2.20	17.50	
				f_{IL} = 15 kHz, T_A = -40°C Note 6				0.40	1.22	µA
				f_{IL} = 15 kHz, T_A = +25°C Note 6				0.47	1.22	
				f_{IL} = 15 kHz, T_A = +85°C Note 6				0.80	3.30	
				f_{IL} = 15 kHz, T_A = +105°C Note 6				2.00	17.30	

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

<R> **Note 1.** Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The following points apply in the HS (high-speed main), LS (low-speed main), LV (low-voltage 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 peripheral modules, except for those flowing into the A/D converter, LVD circuit, I/O port, and on-chip pull-up/pull-down resistors, and those flowing while the data flash memory is being rewritten.

In the subsystem clock operation, the currents in both the "TYP." and "MAX." columns do not include the operating currents of the peripheral modules. However, in HALT mode, including the current flowing into the real-time clock 2.

Note 2. When the HALT instruction is executed in the flash memory.

Note 3. When the high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock, low-speed on-chip oscillator clock, and sub clock are stopped.

Note 4. When the high-speed system clock, middle-speed on-chip oscillator clock, low-speed on-chip oscillator clock, and sub clock are stopped.

<R> **Note 5.** When the high-speed system clock, middle-speed on-chip oscillator clock, low-speed on-chip oscillator clock, and high-speed on-chip oscillator clock are stopped. When RTCLPC = 1 and ultra-low-power consumption oscillation is set (AMPHS1, AMPHS0) = (1, 0).

Note 6. When the high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock, high-speed system clock, and sub clock are stopped.

Note 7. When the high-speed system clock, high-speed on-chip oscillator clock, low-speed on-chip oscillator clock, and sub clock are stopped.

Remark 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

Remark 2. f_{IH}: High-speed on-chip oscillator clock frequency (24 MHz max.)

Remark 3. f_{IM}: Middle-speed on-chip oscillator clock frequency (4 MHz max.)

Remark 4. f_{IL}: Low-speed on-chip oscillator clock frequency

Remark 5. f_{SX}: Sub clock frequency (XT1 clock oscillation frequency)

Remark 6. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency or low-speed on-chip oscillator clock frequency)

Remark 7. Except subsystem clock operation, temperature condition of the TYP. value is T_A = 25°C.

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

(T_A = +85 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V) **(4/4)**

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
<R> Supply current Note 1	I_{DD3}	STOP mode Note 2	T_A = -40°C		0.16	0.51	μA
			T_A = +25°C		0.22	0.51	
			T_A = +50°C		0.27	1.10	
			T_A = +70°C		0.37	1.90	
			T_A = +85°C		0.60	3.30	
			T_A = +105°C		1.50	17.00	

<R> **Note 1.** Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. In the STOP mode, the currents in both the “TYP.” and “MAX.” columns do not include the operating currents of the peripheral modules.

<R> **Note 2.** For the setting of the current values when operating the subsystem clock in STOP mode, see the current values when operating the subsystem clock in HALT mode.

Peripheral Functions (Common to all products)

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)
(T_A = +85 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Low-speed on-chip oscillator operating current	I_{FIL} Note 1				0.20		μA
RTC2 operating current	I_{RTC} Notes 1, 2, 3	f_{SX} = 32.768 kHz			0.02		μA
12-bit interval timer operating current	I_{TMKA} Notes 1, 2, 4	f_{SX} = 32.768 kHz			0.04		μA
8-bit interval timer operating current	I_{TMT} Notes 1, 9	f_{SX} = 32.768 kHz f_{MAIN} stopped (per unit)	8-bit counter mode × 2-channel operation		0.12		μA
			16-bit counter mode operation		0.10		μA
Watchdog timer operating current	I_{WDT} Notes 1, 2, 5	f_{IL} = 15 kHz			0.22		μA
A/D converter operating current	I_{ADC} Notes 6, 10	During maximum-speed conversion	AV_{DD} = 3.0 V		420	720	μA
$AV_{REF(+)}$ current	I_{AVREF} Note 11	AV_{REFP} = 3.0 V, ADREFP1 = 0, ADREFP0 = 1			14.0	25.0	μA
Internal reference voltage (1.45 V) current	I_{ADREF} Notes 1, 12				85.0		μA
Temperature sensor operating current	I_{TMPS} Note 1				85.0		μA
Comparator operating current	I_{CMP} Notes 8, 10	AV_{DD} = 3.6 V, Regulator output voltage = 2.1 V	Comparator high-speed mode Window mode		12.5		μA
			Comparator low-speed mode Window mode		3.0		
			Comparator high-speed mode Standard mode		6.5		
			Comparator low-speed mode Standard mode		1.7		
		AV_{DD} = 3.6 V, Regulator output voltage = 1.8 V	Comparator high-speed mode Window mode		8.0		
			Comparator low-speed mode Window mode		2.2		
			Comparator high-speed mode Standard mode		4.0		
			Comparator low-speed mode Standard mode		1.3		
Operational amplifier operating current	I_{AMP} Notes 10, 13	Low-power consumption mode	One operational amplifier unit operates Note 14		2.5	4.0	μA
			Two operational amplifier units operate Note 14		4.5	8.0	
			Three operational amplifier units operate Note 14		6.5	11.0	
			Four operational amplifier units operate Note 14		8.5	14.0	
		High-speed mode	One operational amplifier unit operates Note 14		140	220	
			Two operational amplifier units operate Note 14		280	410	
			Three operational amplifier units operate Note 14		420	600	
			Four operational amplifier units operate Note 14		560	780	
LVD operating current	I_{LVD} Notes 1, 7				0.10		μA

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

Note 1. Current flowing to V_{DD}.

Note 2. When the high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock, and high-speed system clock are stopped.

Note 3. Current flowing only to the real-time clock 2 (RTC2) (excluding the operating current of the low-speed on-chip oscillator and the XT1 oscillator). The supply current of the RL78 microcontrollers is the sum of the values of either I_{DD1} or I_{DD2}, and I_{RTC}, when the real-time clock 2 operates in operation mode or HALT mode. When the low-speed on-chip oscillator is selected, I_{FIL} should be added. I_{DD2} subsystem clock operation includes the operational current of the real-time clock 2.

Note 4. Current flowing only to the 12-bit interval timer (excluding the operating current of the low-speed on-chip oscillator and the XT1 oscillator). The supply current of the RL78 microcontrollers is the sum of the values of either I_{DD1} or I_{DD2}, and I_{IT}, when the 12-bit interval timer operates in operation mode or HALT mode. When the low-speed on-chip oscillator is selected, I_{FIL} should be added.

Note 5. Current flowing only to the watchdog timer (including the operating current of the low-speed on-chip oscillator).
The supply current of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{WDT} when the watchdog timer is in operation.

Note 6. Current flowing only to the A/D converter. The supply current of the RL78 microcontrollers is the sum of I_{DD1} or I_{DD2} and I_{ADC} when the A/D converter operates in an operation mode or the HALT mode.

Note 7. Current flowing only to the LVD circuit. The supply current of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{LVD} when the LVD circuit is in operation.

Note 8. Current flowing only to the comparator circuit. The supply current of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2}, or I_{DD3} and I_{CMP} when the comparator circuit is in operation.

Note 9. Current flowing only to the 8-bit interval timer (excluding the operating current of the low-speed on-chip oscillator and the XT1 oscillator). The supply current of the RL78 microcontrollers is the sum of the values of either I_{DD1} or I_{DD2}, and I_{IT}, when the 8-bit interval timer operates in operation mode or HALT mode. When the low-speed on-chip oscillator is selected, I_{FIL} should be added.

Note 10. Current flowing to AV_{DD}.

Note 11. Current flowing into AV_{REFP}.

Note 12. Current consumed by generating the internal reference voltage (1.45 V).

Note 13. Current flowing only to the operational amplifier. The current value of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2}, or I_{DD3} and I_{AMP} when the operational amplifier is operating in operating mode, HALT mode, or STOP mode.

Note 14. The values include the operating current of the operational amplifier reference current circuit.

Remark 1. f_{IL}: Low-speed on-chip oscillator clock frequency

Remark 2. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

Remark 3. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 4. Temperature condition of the TYP. value is T_A = 25°C.

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

(T_A = +85 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V) **(2/2)**

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Self-programming operating current	I_{FSP} Notes 1, 3				2.0	12.20	mA
BGO current	I_{BGO} Notes 1, 2				2.0	12.20	mA
SNOOZE operating current	I_{SNOZ} Note 1	ADC operation AV_{REFP} = V_{DD} = 3.0 V T_A = -40 to +85°C	The mode is performed Note 5		0.50	0.60	mA
			The A/D conversion operations are performed Note 1		0.60	0.75	mA
			The A/D conversion operations are performed Note 4		420	720	μA
		ADC operation AV_{REFP} = V_{DD} = 3.0 V T_A = +85 to +105°C	The mode is performed Note 5		0.50	1.10	mA
			The A/D conversion operations are performed Note 1		0.60	1.34	mA
			The A/D conversion operations are performed Note 4		420	720	μA
		Simplified SPI (CSI)/UART operation	T_A = -40 to +85°C		0.70	0.84	mA
			T_A = +85 to +105°C		0.70	1.54	mA

Note 1. Current flowing to V_{DD}.

Note 2. Current flowing during programming of the data flash.

Note 3. Current flowing during self-programming.

Note 4. Current flowing to AV_{DD}.

Note 5. For shift time to the SNOOZE mode, see **23.3.3 SNOOZE mode**.

Remark 1. f_{IL}: Low-speed on-chip oscillator clock frequency

Remark 2. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

Remark 3. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 4. Temperature condition of the TYP. value is T_A = 25°C.

34.4 AC Characteristics

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

(T_A = +85 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V) **(1/2)**

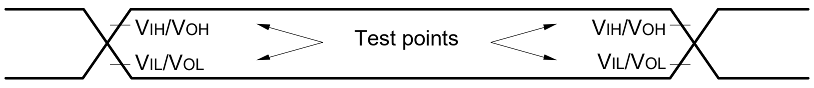
Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation	HS (high-speed main) mode	2.7 V ≤ V_{DD} ≤ 3.6 V	0.04167		1	µs
				2.4 V ≤ V_{DD} < 2.7 V	0.0625		1	µs
			LS (low-speed main) mode	1.8 V ≤ V_{DD} ≤ 3.6 V PMMC. MCSEL = 0	0.125		1	µs
				1.8 V ≤ V_{DD} ≤ 3.6 V PMMC. MCSEL = 1	0.25		1	
			LP (low-power main) mode	1.8 V ≤ V_{DD} ≤ 3.6 V		1		µs
			LV (low-voltage main) mode	1.8 V ≤ V_{DD} ≤ 3.6 V	0.25		1	µs
				1.6 V ≤ V_{DD} < 1.8 V	0.34		1	
		Subsystem clock (f_{SUB}) operation	f_{SX}	1.8 V ≤ V_{DD} ≤ 3.6 V	28.5	30.5	31.3	µs
			f_{IL}	1.8 V ≤ V_{DD} ≤ 3.6 V		66.7		
		In the self-programming mode	HS (high-speed main) mode	2.7 V ≤ V_{DD} ≤ 3.6 V	0.04167		1	µs
				2.4 V ≤ V_{DD} < 2.7 V	0.0625		1	µs
			LS (low-speed main) mode	1.8 V ≤ V_{DD} ≤ 3.6 V	0.125		1	µs
			LV (low-voltage main) mode	1.8 V ≤ V_{DD} ≤ 3.6 V	0.25		1	µs
External system clock frequency	f_{EX}	2.7 V ≤ V_{DD} ≤ 3.6 V			1.0		20.0	MHz
		2.4 V ≤ V_{DD} <2.7 V			1.0		16.0	MHz
		1.8 V ≤ V_{DD} <2.4 V			1		8	MHz
		1.6 V ≤ V_{DD} <1.8 V			1		4	MHz
	f_{EXS}				32		35	kHz
External system clock input high-level width, low-level width	t_{EXH}, t_{EXL}	2.7 V ≤ V_{DD} ≤ 3.6 V			24			ns
		2.4 V ≤ V_{DD} <2.7 V			30			ns
		1.8 V ≤ V_{DD} <2.4 V			60			ns
		1.6 V ≤ V_{DD} <1.8 V			120			ns
	t_{EXHS}, t_{EXLS}				13.7			µs
TI00 to TI03 input high-level width, low-level width	t_{TIH}, t_{TIL}				1/f_{MCK} + 10			ns

Remark f_{MCK}: Timer array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of timer mode register mn (TMRmn). m: Unit number (m = 0), n: Channel number (n = 0 to 3))

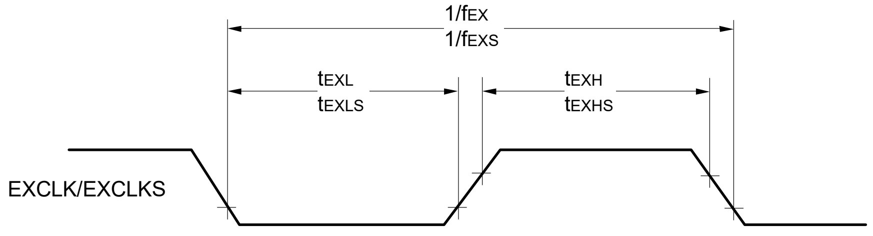
(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)
(T_A = +85 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V) **(2/2)**

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
TO00 to TO03 output frequency	f_{TO}	HS (high-speed main) mode	2.7 V ≤ V_{DD} ≤ 3.6 V			8	MHz
			2.4 V ≤ V_{DD} < 2.7 V			4	
		LS (low-speed main) mode	1.8 V ≤ V_{DD} ≤ 3.6 V			4	
		LP (low-power main) mode	1.8 V ≤ V_{DD} ≤ 3.6 V			0.5	
		LV (low-voltage main) mode	1.6 V ≤ V_{DD} ≤ 3.6 V			2	
PCLBUZ0, PCLBUZ1 output frequency	f_{PCL}	HS (high-speed main) mode	2.7 V ≤ V_{DD} ≤ 3.6 V			8	MHz
			2.4 V ≤ V_{DD} < 2.7 V			4	
		LS (low-speed main) mode	1.8 V ≤ V_{DD} ≤ 3.6 V			4	
		LP (low-power main) mode	1.8 V ≤ V_{DD} ≤ 3.6 V			1	
		LV (low-voltage main) mode	1.8 V ≤ V_{DD} ≤ 3.6 V			4	
			1.6 V ≤ V_{DD} < 1.8 V			2	
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}	INTP0 to INTP6	1.6 V ≤ V_{DD} ≤ 3.6 V	1			μs
Key interrupt input low-level width	t_{KR}	KR0 to KR3	1.8 V ≤ V_{DD} ≤ 3.6 V	250			ns
			1.6 V ≤ V_{DD} < 1.8 V	1			μs
$\overline{\text{RESET}}$ low-level width	t_{RSL}			10			μs

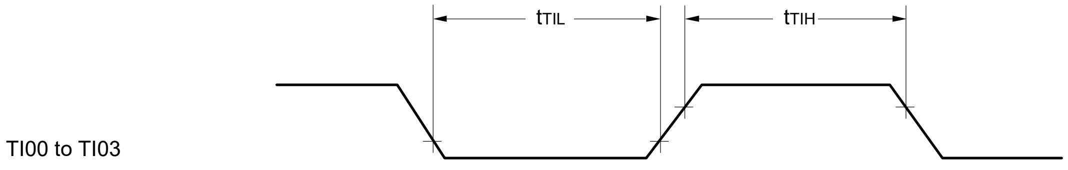
AC Timing Test Points

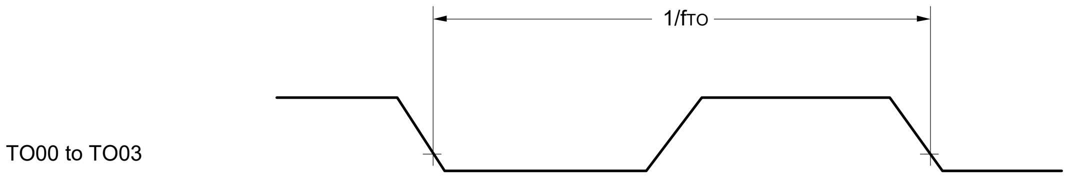


External System Clock Timing

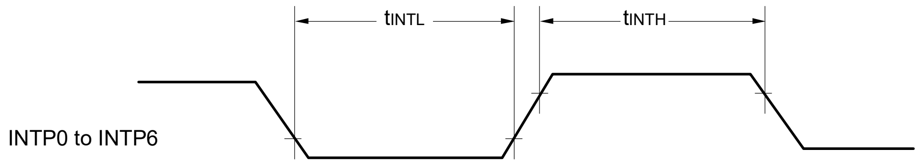


TI/TO Timing

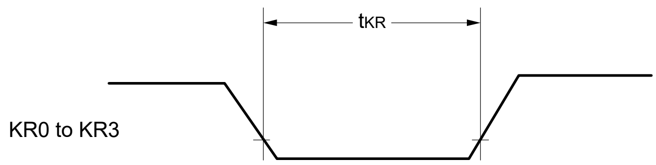




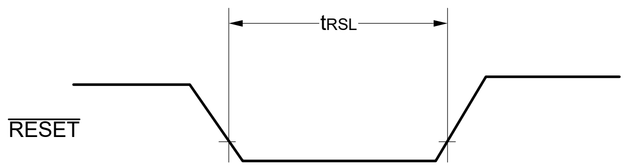
Interrupt Request Input Timing



Key Interrupt Input Timing

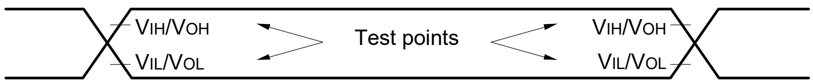


$\overline{\text{RESET}}$ Input Timing



34.5 Peripheral Functions Characteristics

AC Timing Test Points



34.5.1 Serial array unit

(1) During communication at same potential (UART mode)

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LP (Low-power main) mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
Transfer rate Note 1		2.4 V ≤ V_{DD} ≤ 3.6 V			f_{MCK}/6		f_{MCK}/6		f_{MCK}/6		f_{MCK}/6	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 2		4.0		1.3		0.1		0.6	Mbps
		1.8 V ≤ V_{DD} ≤ 3.6 V		—			f_{MCK}/6		f_{MCK}/6		f_{MCK}/6	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 2	—			1.3		0.1		0.6	Mbps
		1.7 V ≤ V_{DD} ≤ 3.6 V		—		—		—			f_{MCK}/6	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 2	—		—		—			0.6	Mbps
		1.6 V ≤ V_{DD} ≤ 3.6 V		—		—		—			f_{MCK}/6	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 2	—		—		—			0.6	Mbps

Note 1. Transfer rate in the SNOOZE mode is 4800 bps only.

Note 2. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:
HS (high-speed main) mode: 24 MHz (2.7 V ≤ V_{DD} ≤ 3.6 V)
16 MHz (2.4 V ≤ V_{DD} ≤ 3.6 V)
LS (low-speed main) mode: 8 MHz (1.8 V ≤ V_{DD} ≤ 3.6 V)
LP (low-power main) mode: 1 MHz (1.8 V ≤ V_{DD} ≤ 3.6 V)
LV (low-voltage main) mode: 4 MHz (1.6 V ≤ V_{DD} ≤ 3.6 V)

Caution Select the normal input buffer for the RxDq pin and the normal output mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg).

(T_A = +85 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

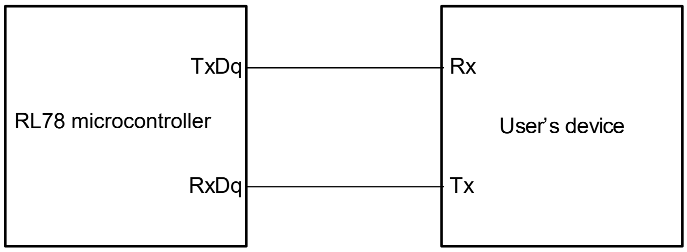
Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
Transfer rate Note 1		2.4 V ≤ V_{DD} ≤ 3.6 V			f_{MCK}/12	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 2		2.0	Mbps

Note 1. Transfer rate in the SNOOZE mode is 4800 bps only.

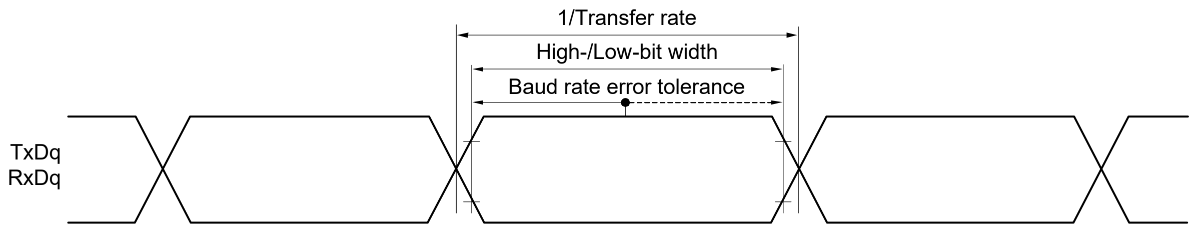
Note 2. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:
HS (high-speed main) mode: 24 MHz (2.7 V ≤ V_{DD} ≤ 3.6 V)
16 MHz (2.4 V ≤ V_{DD} ≤ 3.6 V)

Caution Select the normal input buffer for the RxDq pin and the normal output mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg).

UART mode connection diagram (during communication at same potential)



UART mode bit width (during communication at same potential) (reference)



Remark 1. q: UART number (q = 0), g: PIM and POM number (g = 5)

Remark 2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 01))

(2) During communication at same potential (simplified SPI (CSI) mode) (master mode, SCKp... internal clock output, corresponding CSI00 only)

(T_A = -40 to +85°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LP (Low-power main) mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ f_{CLK}/2	83.3		250		2000		500		ns
SCKp high-/low-level width	t_{KL1}		t_{KCY1}/2 - 10		t_{KCY1}/2 - 50		t_{KCY1}/2 - 50		t_{KCY1}/2 - 50		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}		33		110		110		110		ns
SIp hold time (from SCKp↑) Note 2	t_{KSI1}		10		10		10		10		ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO1}	C = 20 pF Note 4		10		20		20		20	ns

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 4. 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 port input mode register g (PIMg) and port output mode register g (POMg).**

Remark 1. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM numbers (g = 5)

Remark 2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number,
n: Channel number (mn = 00))

(3) During communication at same potential (simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LP (Low-power main) mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ f_{CLK}/4	2.7 V ≤ V_{DD} ≤ 3.6 V	167		500		4000		1000		ns
			2.4 V ≤ V_{DD} ≤ 3.6 V	250								
			1.8 V ≤ V_{DD} ≤ 3.6 V	—								
			1.7 V ≤ V_{DD} ≤ 3.6 V	—		—		—				
			1.6 V ≤ V_{DD} ≤ 3.6 V	—		—		—				
SCKp high-/low-level width	t_{KH1}, t_{KL1}	2.7 V ≤ V_{DD} ≤ 3.6 V		t_{KCY1}/2 - 18		t_{KCY1}/2 - 50		t_{KCY1}/2 - 50		t_{KCY1}/2 - 50		ns
		2.4 V ≤ V_{DD} ≤ 3.6 V		t_{KCY1}/2 - 38								
		1.8 V ≤ V_{DD} ≤ 3.6 V		—								
		1.7 V ≤ V_{DD} ≤ 3.6 V		—		—		—		t_{KCY1}/2 - 100		
		1.6 V ≤ V_{DD} ≤ 3.6 V		—		—		—				
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	2.7 V ≤ V_{DD} ≤ 3.6 V		58		110		110		110		ns
		2.4 V ≤ V_{DD} ≤ 3.6 V		75								
		1.8 V ≤ V_{DD} ≤ 3.6 V		—								
		1.7 V ≤ V_{DD} ≤ 3.6 V		—		—		—		220		
		1.6 V ≤ V_{DD} ≤ 3.6 V		—		—		—				
SIp hold time (from SCKp↑) Note 2	t_{KSI1}	2.4 V ≤ V_{DD} ≤ 3.6 V		19		19		19		19		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V		—								
		1.6 V ≤ V_{DD} ≤ 3.6 V		—		—		—				
Delay time from SCKp↓ to SOp output Note 3	t_{KSO1}	C = 30 pF Note 4	2.4 V ≤ V_{DD} ≤ 3.6 V		33.4		33.4		33.4		33.4	ns
			1.8 V ≤ V_{DD} ≤ 3.6 V		—							
			1.6 V ≤ V_{DD} ≤ 3.6 V		—		—		—			

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes “to SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes “from SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes “from SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 4. 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 port input mode register g (PIMg) and port output mode register g (POMg).

Remark 1. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM numbers (g = 5)

Remark 2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number,
n: Channel number (mn = 00, 01))

(3) During communication at same potential (simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = +85 to +105°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
SCKp cycle time	t_{KCY1}	$t_{KCY1} \geq f_{CLK}/4$	2.7 V ≤ V_{DD} ≤ 3.6 V	250		ns
			2.4 V ≤ V_{DD} ≤ 3.6 V	500		ns
SCKp high-/low-level width	t_{KH1}, t_{KL1}	2.7 V ≤ V_{DD} ≤ 3.6 V		$t_{KCY1}/2 - 36$		ns
		2.4 V ≤ V_{DD} ≤ 3.6 V		$t_{KCY1}/2 - 76$		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	2.7 V ≤ V_{DD} ≤ 3.6 V		66		ns
		2.4 V ≤ V_{DD} ≤ 3.6 V		133		ns
SIp hold time (from SCKp↑) Note 2	t_{KSI1}			38		ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO1}	C = 30 pF Note 4			50	ns

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes “to SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes “from SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes “from SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 4. 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 port input mode register g (PIMg) and port output mode register g (POMg).

Remark 1. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM numbers (g = 5)

Remark 2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 01))

(4) During communication at same potential (simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V) (1/2)

Parameter	Symbol	Conditions		HS (high-speed main) Mode MIN.	HS (high-speed main) Mode MAX.	LS (low-speed main) Mode MIN.	LS (low-speed main) Mode MAX.	LP (Low-power main) mode MIN.	LP (Low-power main) mode MAX.	LV (low-voltage main) Mode MIN.	LV (low-voltage main) Mode MAX.	Unit
SCKp cycle time Note 5	t_{KCY2}	2.7 V ≤ V_{DD} ≤ 3.6 V	f_{MCK} > 16 MHz	$8/f_{MCK}$		—	—	—	—	—	—	ns
			f_{MCK} ≤ 16 MHz	$6/f_{MCK}$		$6/f_{MCK}$		$6/f_{MCK}$		$6/f_{MCK}$		
		2.4 V ≤ V_{DD} ≤ 3.6 V		$6/f_{MCK}$ and 500		$6/f_{MCK}$		$6/f_{MCK}$		$6/f_{MCK}$		
		1.8 V ≤ V_{DD} ≤ 3.6 V		—		$6/f_{MCK}$		$6/f_{MCK}$		$6/f_{MCK}$		
		1.7 V ≤ V_{DD} ≤ 3.6 V		—		—		—				
		1.6 V ≤ V_{DD} ≤ 3.6 V		—		—		—				
SCKp high-/low-level width	t_{KH2}, t_{KL2}	2.7 V ≤ V_{DD} ≤ 3.6 V		$t_{KCY2}/2$ - 8		$t_{KCY2}/2$ - 8		$t_{KCY2}/2$ - 8		$t_{KCY2}/2$ - 8		ns
		2.4 V ≤ V_{DD} ≤ 3.6 V		$t_{KCY2}/2$ - 18		$t_{KCY2}/2$ - 18		$t_{KCY2}/2$ - 18		$t_{KCY2}/2$ - 18		
		1.8 V ≤ V_{DD} ≤ 3.6 V		—								
		1.7 V ≤ V_{DD} ≤ 3.6 V		—		—		—		$t_{KCY2}/2$ - 66		
		1.6 V ≤ V_{DD} ≤ 3.6 V		—		—		—				
SIp setup time (to SCKp↑) Note 1	t_{SIK2}	2.7 V ≤ V_{DD} ≤ 3.6 V		$1/f_{MCK}$ + 20		$1/f_{MCK}$ + 30		$1/f_{MCK}$ + 30		$1/f_{MCK}$ + 30		ns
		2.4 V ≤ V_{DD} ≤ 3.6 V		$1/f_{MCK}$ + 30								
		1.8 V ≤ V_{DD} ≤ 3.6 V		—								
		1.7 V ≤ V_{DD} ≤ 3.6 V		—		—		—		$1/f_{MCK}$ + 40		
		1.6 V ≤ V_{DD} ≤ 3.6 V		—		—		—				
SIp hold time (from SCKp↑) Note 2	t_{KSI2}	2.4 V ≤ V_{DD} ≤ 3.6 V		$1/f_{MCK}$ + 31		$1/f_{MCK}$ + 31		$1/f_{MCK}$ + 31		$1/f_{MCK}$ + 31		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V		—								
		1.7 V ≤ V_{DD} ≤ 3.6 V		—		—		—		$1/f_{MCK}$ + 250		
		1.6 V ≤ V_{DD} ≤ 3.6 V		—		—		—				
Delay time from SCKp↓ to SOp output Note 3	t_{KSO2}	C = 30 pF Note 4	2.7 V ≤ V_{DD} ≤ 3.6 V		$2/f_{MCK}$ + 44		$2/f_{MCK}$ + 110		$2/f_{MCK}$ + 110		$2/f_{MCK}$ + 110	ns
			2.4 V ≤ V_{DD} ≤ 3.6 V		$2/f_{MCK}$ + 75							
			1.8 V ≤ V_{DD} ≤ 3.6 V		—							
			1.7 V ≤ V_{DD} ≤ 3.6 V		—		—		—		$2/f_{MCK}$ + 220	
			1.6 V ≤ V_{DD} ≤ 3.6 V		—		—		—			

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes “to SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes “from SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes “from SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 4. C is the load capacitance of the SOp output lines.

Note 5. The maximum transfer rate when using the SNOOZE mode is 1 Mbps.

Caution **Select the normal input buffer for the SIp pin and the normal output mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg).**

Remark 1. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM numbers (g = 5)

Remark 2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 01))

(4) During communication at same potential (simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

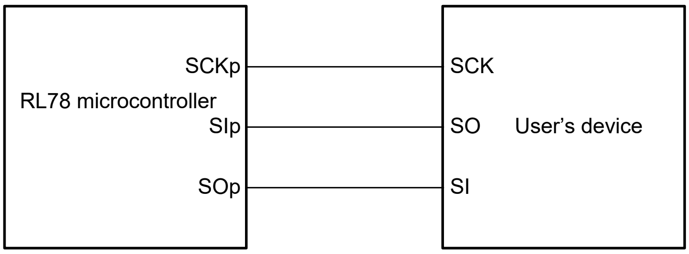
(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V) **(2/2)**

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LP (Low-power main) mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
$\overline{\text{SSI00}}$ setup time	t_{SSIK}	DAPmn = 0	2.7 V ≤ V_{DD} ≤ 3.6 V	120		120		120		120		ns
			2.4 V ≤ V_{DD} < 2.7 V	200		200		200		200		
			1.8 V ≤ V_{DD} < 2.4 V	—								
			1.6 V ≤ V_{DD} < 1.8 V	—		—		—		400		
		DAPmn = 1	2.7 V ≤ V_{DD} ≤ 3.6 V	$1/f_{MCK}$ + 120		$1/f_{MCK}$ + 120		$1/f_{MCK}$ + 120		$1/f_{MCK}$ + 120		ns
			2.4 V ≤ V_{DD} < 2.7 V	$1/f_{MCK}$ + 200		$1/f_{MCK}$ + 200		$1/f_{MCK}$ + 200		$1/f_{MCK}$ + 200		
			1.8 V ≤ V_{DD} < 2.4 V	—								
			1.6 V ≤ V_{DD} < 1.8 V	—		—		—		$1/f_{MCK}$ + 400		
$\overline{\text{SSI00}}$ hold time	t_{KSSI}	DAPmn = 0	2.7 V ≤ V_{DD} ≤ 3.6 V	$1/f_{MCK}$ + 120		$1/f_{MCK}$ + 120		$1/f_{MCK}$ + 120		$1/f_{MCK}$ + 120		ns
			2.4 V ≤ V_{DD} < 2.7 V	$1/f_{MCK}$ + 200		$1/f_{MCK}$ + 200		$1/f_{MCK}$ + 200		$1/f_{MCK}$ + 200		
			1.8 V ≤ V_{DD} < 2.4 V	—								
			1.6 V ≤ V_{DD} < 1.8 V	—		—		—		$1/f_{MCK}$ + 400		
		DAPmn = 1	2.7 V ≤ V_{DD} ≤ 3.6 V	120		120		120		120		ns
			2.4 V ≤ V_{DD} < 2.7 V	200		200		200		200		
			1.8 V ≤ V_{DD} < 2.4 V	—								
			1.6 V ≤ V_{DD} < 1.8 V	—		—		—		400		

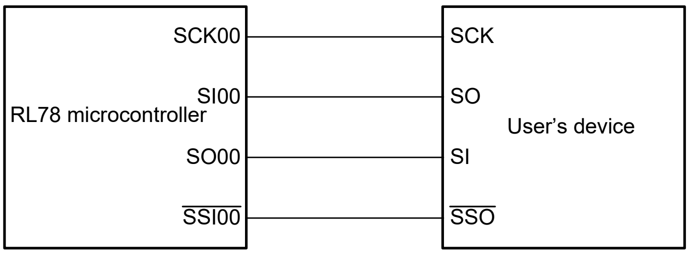
Caution **Select the normal input buffer for the SIp pin and SCKp pin and the normal output mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg).**

Remark p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM numbers (g = 5)

Simplified SPI (CSI) mode connection diagram (during communication at same potential)



Simplified SPI (CSI) mode connection diagram (during communication at same potential)
(Slave Transmission of slave select input function (CSI00))



Remark 1. p: CSI number (p = 00, 01)
Remark 2. m: Unit number, n: Channel number (mn = 00, 01)

(4) During communication at same potential (simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

(T_A = +85 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions		HS (high-speed main) Mode MIN.	HS (high-speed main) Mode MAX.	Unit
SCKp cycle time Note 5	t_{KCY2}	2.7 V ≤ V_{DD} < 3.6 V	f_{MCK} > 16 MHz	16/f_{MCK}		ns
			f_{MCK} ≤ 16 MHz	12/f_{MCK}		ns
		2.4 V ≤ V_{DD} < 2.7 V		12/f_{MCK} and 1000		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	2.7 V ≤ V_{DD} ≤ 3.6 V		t_{KCY2}/2 - 16		ns
		2.4 V ≤ V_{DD} < 2.7 V		t_{KCY2}/2 - 36		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK2}	2.7 V ≤ V_{DD} ≤ 3.6 V		1/f_{MCK} + 40		ns
		2.4 V ≤ V_{DD} < 2.7 V		1/f_{MCK} + 60		ns
SIp hold time (from SCKp↑) Note 2	t_{KSI2}			1/f_{MCK} + 62		ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO2}	C = 30 pF Note 4	2.7 V ≤ V_{DD} ≤ 3.6 V		2/f_{MCK} + 66	ns
			2.4 V ≤ V_{DD} < 2.7 V		2/f_{MCK} + 113	ns

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 4. C is the load capacitance of the SOp output lines.

Note 5. The maximum transfer rate when using the SNOOZE mode is 1 Mbps.

Caution Select the normal input buffer for the SIp pin and the normal output mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg).

Remark 1. p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM numbers (g = 5)

Remark 2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number,
n: Channel number (mn = 00, 01))

(4) During communication at same potential (simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

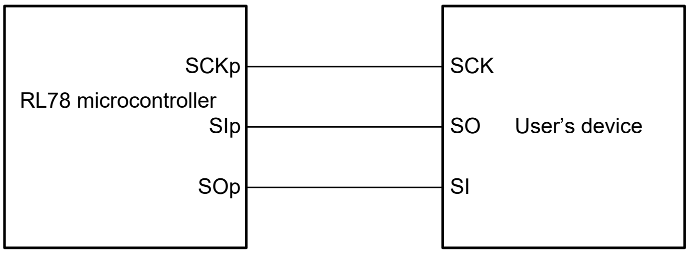
(T_A = +85 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V) **(2/2)**

Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
$\overline{SSI00}$ setup time	t_{SSIK}	DAPmn = 0	2.7 V ≤ V_{DD} ≤ 3.6 V	240		ns
			2.4 V ≤ V_{DD} < 2.7 V	400		ns
		DAPmn = 1	2.7 V ≤ V_{DD} ≤ 3.6 V	$1/f_{MCK}$ + 240		ns
			2.4 V ≤ V_{DD} < 2.7 V	$1/f_{MCK}$ + 400		ns
$\overline{SSI00}$ hold time	t_{KSSI}	DAPmn = 0	2.7 V ≤ V_{DD} ≤ 3.6 V	$1/f_{MCK}$ + 240		ns
			2.4 V ≤ V_{DD} < 2.7 V	$1/f_{MCK}$ + 400		ns
		DAPmn = 1	2.7 V ≤ V_{DD} ≤ 3.6 V	240		ns
			2.4 V ≤ V_{DD} < 2.7 V	400		ns

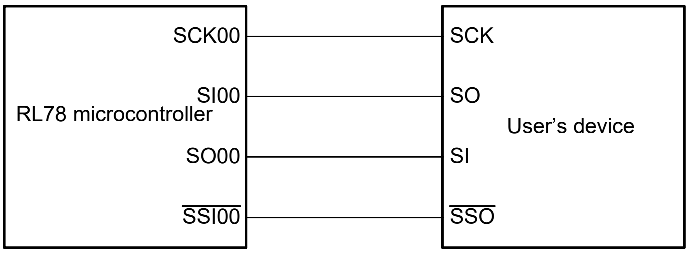
Caution **Select the normal input buffer for the SIp pin and SCKp pin and the normal output mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg).**

Remark p: CSI number (p = 00), m: Unit number (m = 0), n: Channel number (n = 0), g: PIM and POM numbers (g = 5)

Simplified SPI (CSI) mode connection diagram (during communication at same potential)

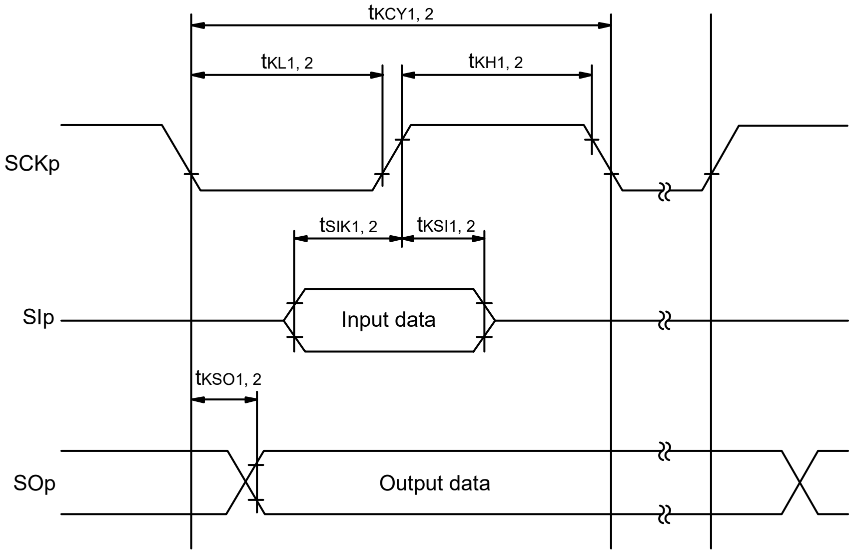


Simplified SPI (CSI) mode connection diagram (during communication at same potential)
(Slave Transmission of slave select input function (CSI00))

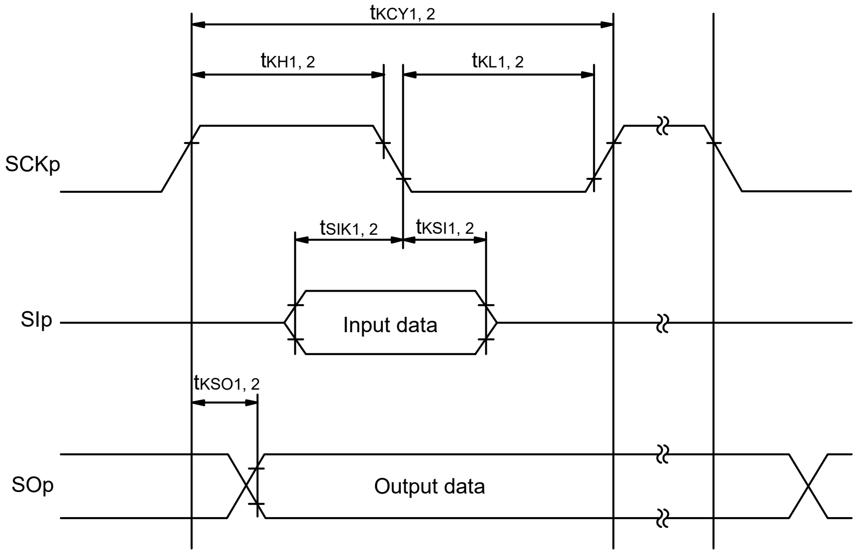


Remark 1. p: CSI number (p = 00, 01)
Remark 2. m: Unit number, n: Channel number (mn = 00, 01)

Simplified SPI (CSI) mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remark 1. p: CSI number (p = 00, 01)

Remark 2. m: Unit number, n: Channel number (mn = 00, 01)

(5) During communication at same potential (simplified I²C mode)

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LP (Low-power main) mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCLr clock frequency	f_{SCL}	2.7 V ≤ V_{DD} ≤ 3.6 V, C_b = 50 pF, R_b = 2.7 kΩ		1000 Note 1		400 Note 1		250 Note 1		400 Note 1	kHz
		1.8 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 3 kΩ		—							
		1.8 V ≤ V_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ		—		300 Note 1		250 Note 1		300 Note 1	
		1.7 V ≤ V_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ		—		—		—		250 Note 1	
		1.6 V ≤ V_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ		—		—		—			
Hold time when SCLr = "L"	t_{LOW}	2.7 V ≤ V_{DD} ≤ 3.6 V, C_b = 50 pF, R_b = 2.7 kΩ	475		1150		1150		1150		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 3 kΩ	—								
		1.8 V ≤ V_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ	—		1550		1550		1550		
		1.7 V ≤ V_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	—		—		—		1850		
		1.6 V ≤ V_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	—		—		—				
Hold time when SCLr = "H"	t_{HIGH}	2.7 V ≤ V_{DD} ≤ 3.6 V, C_b = 50 pF, R_b = 2.7 kΩ	475		1150		1150		1150		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 3 kΩ	—								
		1.8 V ≤ V_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ	—		1550		1550		1550		
		1.7 V ≤ V_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	—		—		—		1850		
		1.6 V ≤ V_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	—		—		—				
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V, C_b = 50 pF, R_b = 2.7 kΩ	$1/f_{MCK}$ + 85 Note 2		$1/f_{MCK}$ + 145 Note 2		$1/f_{MCK}$ + 145 Note 2		$1/f_{MCK}$ + 145 Note 2		ns
		1.8 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 3 kΩ	—								
		1.8 V ≤ V_{DD} < 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		
		1.7 V ≤ V_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	—		—		—		$1/f_{MCK}$ + 290 Note 2		
		1.6 V ≤ V_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	—		—		—				
Data hold time (transmission)	$t_{HD:DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V, C_b = 50 pF, R_b = 2.7 kΩ	0	305	0	305	0	305	0	305	ns
		1.8 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 3 kΩ	—	—		355		355		355	
		1.8 V ≤ V_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ	—	—							
		1.7 V ≤ V_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	—	—	—	—	—	—		405	
		1.6 V ≤ V_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	—	—	—	—	—	—			

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

Note 1. The value must also be equal to or less than $f_{MCK}/4$.

Note 2. Set the f_{MCK} value to keep the hold time of SCLr = "L" and SCLr = "H".

Caution Select the normal input buffer and the N-ch open drain output (V_{DD} tolerance) mode for the SDAr pin and the normal output mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register h (POMh).

(5) During communication at same potential (simplified I²C mode)

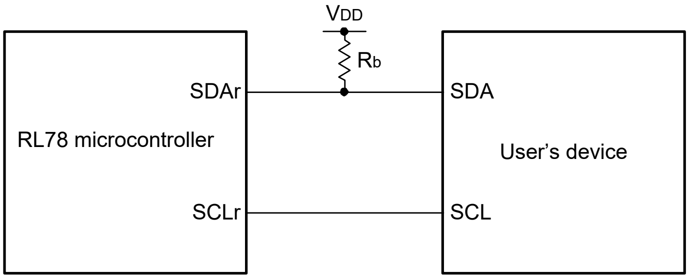
(T_A = +85 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
SCLr clock frequency	f_{SCL}	2.7 V ≤ V_{DD} ≤ 3.6 V, C_b = 50 pF, R_b = 2.7 kΩ		400 Note 1	kHz
		2.4 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 3 kΩ		100 Note 1	kHz
Hold time when SCLr = “L”	t_{LOW}	2.7 V ≤ V_{DD} ≤ 3.6 V, C_b = 50 pF, R_b = 2.7 kΩ	1200		ns
		2.4 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 3 kΩ	4600		ns
Hold time when SCLr = “H”	t_{HIGH}	2.7 V ≤ V_{DD} ≤ 3.6 V, C_b = 50 pF, R_b = 2.7 kΩ	1200		ns
		2.4 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 3 kΩ	4600		ns
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V, C_b = 50 pF, R_b = 2.7 kΩ	$1/f_{MCK}$ + 220 Note 2		ns
		2.4 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 3 kΩ	$1/f_{MCK}$ + 580 Note 2		ns
Data hold time (transmission)	$t_{HD:DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V, C_b = 50 pF, R_b = 2.7 kΩ	0	770	ns
		2.4 V ≤ V_{DD} ≤ 3.6 V, C_b = 100 pF, R_b = 3 kΩ	0	1420	ns

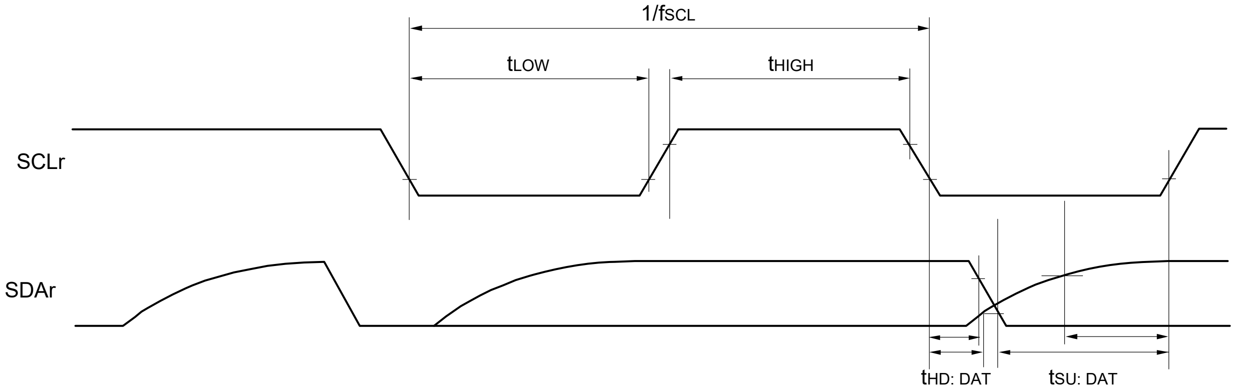
Note 1. The value must also be equal to or less than $f_{MCK}/4$.
Note 2. Set the f_{MCK} value to keep the hold time of SCLr = “L” and SCLr = “H”.

Caution Select the normal input buffer and the N-ch open drain output (V_{DD} tolerance) mode for the SDAr pin and the normal output mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register h (POMh).

Simplified I2C mode connection diagram (during communication at same potential)



Simplified I2C mode serial transfer timing (during communication at same potential)



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), g: PIM number (g = 5), h: POM number (h = 5)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number (m = 0),
n: Channel number (n = 0, 1), mn = 00, 01)

(6) Communication at different potential (1.8 V, 2.5 V) (UART mode) (dedicated baud rate generator output)

(T_A = -40 to +85°C, 1.8 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions		HS (high-speed main) Mode MIN.	HS MAX.	LS (low-speed main) Mode MIN.	LS MAX.	LP (Low-power main) mode MIN.	LP MAX.	LV (low-voltage main) Mode MIN.	LV MAX.	Unit
Transfer rate Notes 1, 2		reception	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V		f_{MCK}/6 Note 1		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 3		4.0		1.3		0.1		0.6	Mbps
			1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V		f_{MCK}/6 Notes 1, 2		f_{MCK}/6 Notes 1, 2		f_{MCK}/6 Notes 1, 2		f_{MCK}/6 Notes 1, 2	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 3		4.0		1.3		0.1		0.6	Mbps

Note 1. Transfer rate in the SNOOZE mode is 4,800 bps only.

Note 2. Use it with V_{DD} ≥ Vb.

Note 3. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:

HS (high-speed main) mode: 24 MHz (2.7 V ≤ V_{DD} ≤ 3.6 V)
16 MHz (2.4 V ≤ V_{DD} ≤ 3.6 V)
LS (low-speed main) mode: 8 MHz (1.8 V ≤ V_{DD} ≤ 3.6 V)
LP (low-power main) mode: 1 MHz (1.8 V ≤ V_{DD} ≤ 3.6 V)
LV (low-voltage main) mode: 4 MHz (1.6 V ≤ V_{DD} ≤ 3.6 V)

Caution **Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (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), g: PIM and POM number (g = 5)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 01)

(6) Communication at different potential (1.8 V, 2.5 V) (UART mode) (dedicated baud rate generator output)

(T_A = -40 to +85°C, 1.8 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V) (2/2)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LP (Low-power main) mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
Transfer rate Note 2		Transmission	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V		Note 1		Note 1		Note 1		Note 1	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 2		1.2 Note 2		1.2 Note 2		1.2 Note 2	Mbps
			1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V		Notes 3, 4		Notes 3, 4		Notes 3, 4		Notes 3, 4	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 5		0.43 Note 5		0.43 Note 5		0.43 Note 5	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 2.7 V ≤ V_{DD} ≤ 3.6 V and 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 2. This value as an example is calculated when the conditions described in the “Conditions” column are met.
Refer to Note 1 above to calculate the maximum transfer rate under conditions of the customer.

Note 3. Use it with V_{DD} ≥ V_b.

Note 4. 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 ≤ V_{DD} < 3.3 V and 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 5. This value as an example is calculated when the conditions described in the “Conditions” column are met.
Refer to Note 4 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 (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.**

(6) Communication at different potential (1.8 V, 2.5 V) (UART mode) (dedicated baud rate generator output)

(T_A = +85 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions			HS (high-speed main) Mode		Unit
					MIN.	MAX.	
Transfer rate Notes 1, 2		Reception	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V			f_{MCK}/12 Note 1	bps
				Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 3		2.0	Mbps
			2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V			f_{MCK}/12 Notes 1, 2	bps
				Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 3		0.66	Mbps

Note 1. Transfer rate in the SNOOZE mode is 4,800 bps only.

Note 2. Use it with V_{DD} ≥ Vb.

Note 3. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:
HS (high-speed main) mode: 24 MHz (2.7 V ≤ V_{DD} ≤ 3.6 V)
16 MHz (2.4 V ≤ V_{DD} ≤ 3.6 V)

Caution **Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (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), g: PIM and POM numbers (g = 5)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 01)

(6) Communication at different potential (1.8 V, 2.5 V) (UART mode) (dedicated baud rate generator output)

(T_A = +85 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V) **(2/2)**

Parameter	Symbol	Conditions			HS (high-speed main) Mode		Unit
					MIN.	MAX.	
Transfer rate Note 2		Transmission	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V			Note 1	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 2	Mbps
			2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V			Notes 3, 4	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 5	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 2.7 V ≤ V_{DD} ≤ 3.6 V and 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 2. This value as an example is calculated when the conditions described in the “Conditions” column are met.
Refer to Note 1 above to calculate the maximum transfer rate under conditions of the customer.

Note 3. Use it with V_{DD} ≥ V_b.

Note 4. 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 2.4 V ≤ V_{DD} < 3.3 V and 1.6 V ≤ V_b ≤ 2.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln (1 - \frac{1.5}{V_b})\} \times 3} \text{ [bps]}$$

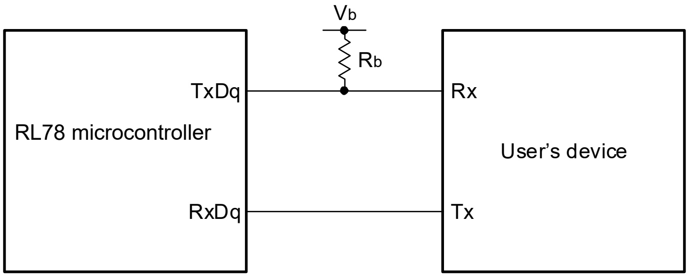
$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln (1 - \frac{1.5}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

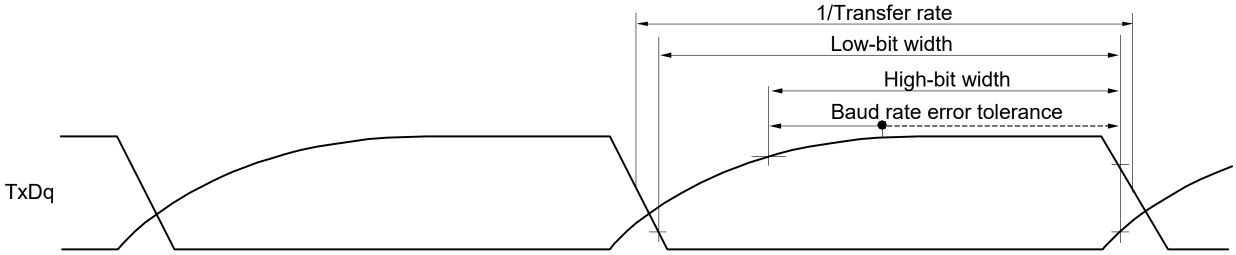
Note 5. This value as an example is calculated when the conditions described in the “Conditions” column are met.
Refer to Note 4 above to calculate the maximum transfer rate under conditions of the customer.

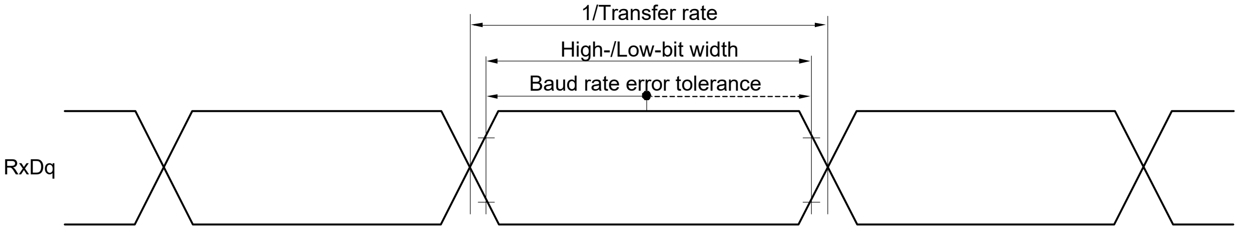
Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

UART mode connection diagram (during communication at different potential)



UART mode bit width (during communication at different potential) (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), g: PIM and POM number (g = 5)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 01))

(7) Communication at different potential (2.5 V) (simplified SPI (CSI) mode) (master mode, SCKp... internal clock output, corresponding CSI00 only)

(T_A = -40 to +85°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LP (Low-power main) mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ f_{CLK}/2	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ	300		1500		1500		1500		ns
SCKp high-level width	t_{KH1}	2.7 V ≤ V_{DD} ≤ 3.6 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		t_{KCY1}/2 - 120		ns
SCKp low-level width	t_{KL1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 1.4 kΩ		t_{KCY1}/2 - 10		t_{KCY1}/2 - 50		t_{KCY1}/2 - 50		t_{KCY1}/2 - 50		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		121		479		479		479		ns
SIp hold time (from SCKp↑) Note 1	t_{KSI1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		10		10		10		10		ns
Delay time from SCKp↓ to SOp output Note 1	t_{KSO1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ			130		130		130		130	ns
SIp setup time (to SCKp↓) Note 2	t_{SIK1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		33		110		110		110		ns
SIp hold time (from SCKp↓) Note 2	t_{KSI1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		10		10		10		10		ns
Delay time from SCKp↑ to SOp output Note 2	t_{KSO1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ			10		10		10		10	ns

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

Note 2. 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 (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (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 number (g = 5)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00))

(8) Communication at different potential (1.8 V, 2.5 V) (simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = -40 to +85°C, 1.8 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V) (1/2)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		LS (low-speed main) Mode		LP (Low-power main) mode		LV (low-voltage main) Mode		Unit
				MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ f_{CLK}/4	2.7V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	500		1150		1150		1150		ns
			1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note, C_b = 30 pF, R_b = 5.5 kΩ	1150		1150		1150		1150		ns
SCKp high-level width	t_{KH1}	2.7 V ≤ V_{DD} ≤ 3.6 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		t_{KCY1}/2 - 170		ns
		1.8 V ≤ V_{DD} < 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		t_{KCY1}/2 - 458		ns
SCKp low-level width	t_{KL1}	2.7 V ≤ V_{DD} ≤ 3.6 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 - 50		t_{KCY1}/2 - 50		t_{KCY1}/2 - 50		ns
		1.8 V ≤ V_{DD} < 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		t_{KCY1}/2 - 50		ns

Note Use it with V_{DD} ≥ V_b.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

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

(8) Communication at different potential (1.8 V, 2.5 V) (simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = -40 to +85°C, 1.8 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V) **(2/2)**

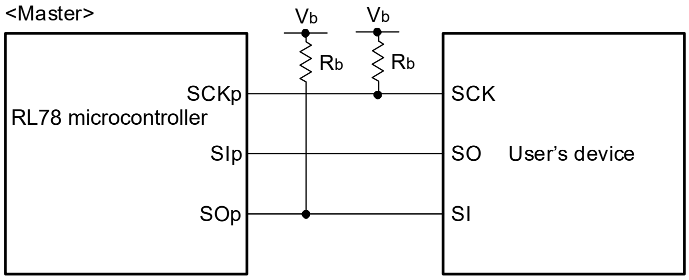
Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LP (Low-power main) mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	177		479		479		479		ns
		1.8 V ≤ V_{DD} ≤ 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ	479		479		479		479		ns
SIp hold time (from SCKp↑) Note 1	t_{KSI1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	19		19		19		19		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ	19		19		19		19		ns
Delay time from SCKp↓ to SOp output Note 1	t_{KSO1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		195		195		195		195	ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ		483		483		483		483	ns
SIp setup time (to SCKp↓) Note 2	t_{SIK1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	44		110		110		110		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ	110		110		110		110		ns
SIp hold time (from SCKp↓) Note 2	t_{KSI1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	19		19		19		19		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ	19		19		19		19		ns
Delay time from SCKp↑ to SOp output Note 2	t_{KSO1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		25		25		25		25	ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ		25		25		25		25	ns

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.
Note 2. When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.
Note 3. Use it with V_{DD} ≥ V_b.

Caution **Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.**

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

Simplified SPI (CSI) mode connection diagram (during communication at different potential)

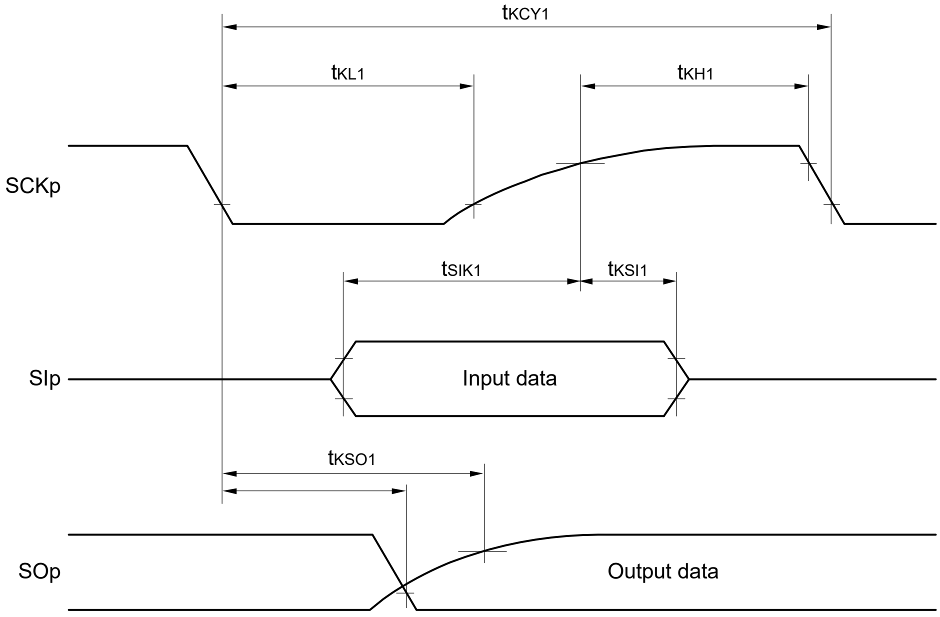


Remark 1. R_b[Ω]: Communication line (SCKp, SOp) pull-up resistance, C_b[F]: Communication line (SCKp, SOp) load capacitance, V_b[V]: Communication line voltage

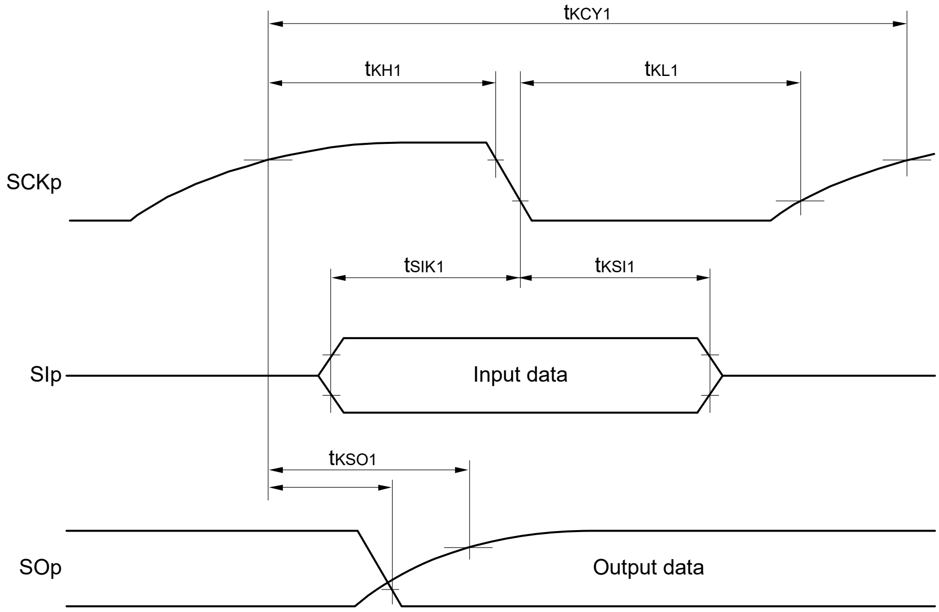
Remark 2. p: CSI number (p = 00, 01), m: Unit number (m = 0), n: Channel number (n = 0, 1), g: PIM and POM numbers (g = 5)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number,
n: Channel number (mn = 00, 01))

Simplified SPI (CSI) mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remark p: CSI number (p = 00, 01), m: Unit number (m = 0), n: Channel number (n = 0, 1), g: PIM and POM numbers (g = 5)

(8) Communication at different potential (1.8 V, 2.5 V) (simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = +85 to 105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
SCKp cycle time	t_{KCY1}	$t_{KCY1} \geq f_{CLK}/4$	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	1000		ns
			2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	2300		ns
SCKp high-level width	t_{KH1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		$t_{KCY1}/2$ - 340		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		$t_{KCY1}/2$ - 916		ns
SCKp low-level width	t_{KL1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		$t_{KCY1}/2$ - 36		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		$t_{KCY1}/2$ - 100		ns

Caution **Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.**

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

(8) Communication at different potential (1.8 V, 2.5 V) (simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = +85 to 105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V) **(2/2)**

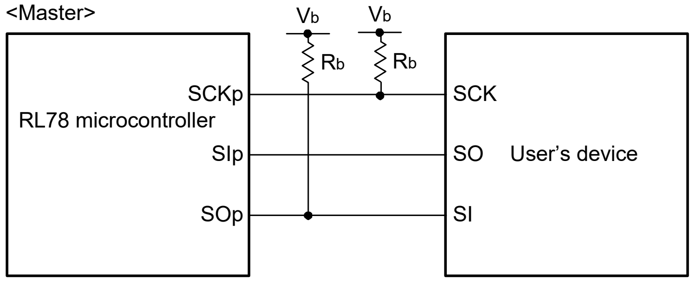
Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	354		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ	958		ns
SIp hold time (from SCKp↑) Note 1	t_{KSI1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	38		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ	38		ns
Delay time from SCKp↓ to SOp output Note 1	t_{KSO1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		390	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ		966	ns
SIp setup time (to SCKp↓) Note 2	t_{SIK1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	88		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ	220		ns
SIp hold time (from SCKp↓) Note 2	t_{KSI1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	38		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ	38		ns
Delay time from SCKp↑ to SOp output Note 2	t_{KSO1}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		50	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ		50	ns

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.
Note 2. When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.
Note 3. Use it with V_{DD} ≥ V_b.

Caution **Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.**

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

Simplified SPI (CSI) mode connection diagram (during communication at different potential)

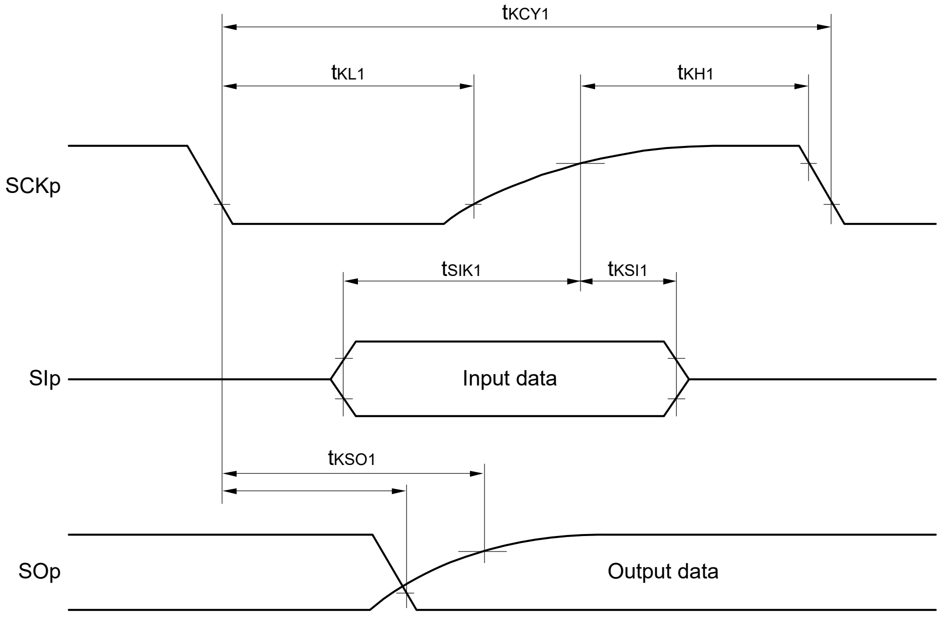


Remark 1. R_b[Ω]: Communication line (SCKp, SOp) pull-up resistance, C_b[F]: Communication line (SCKp, SOp) load capacitance, V_b[V]: Communication line voltage

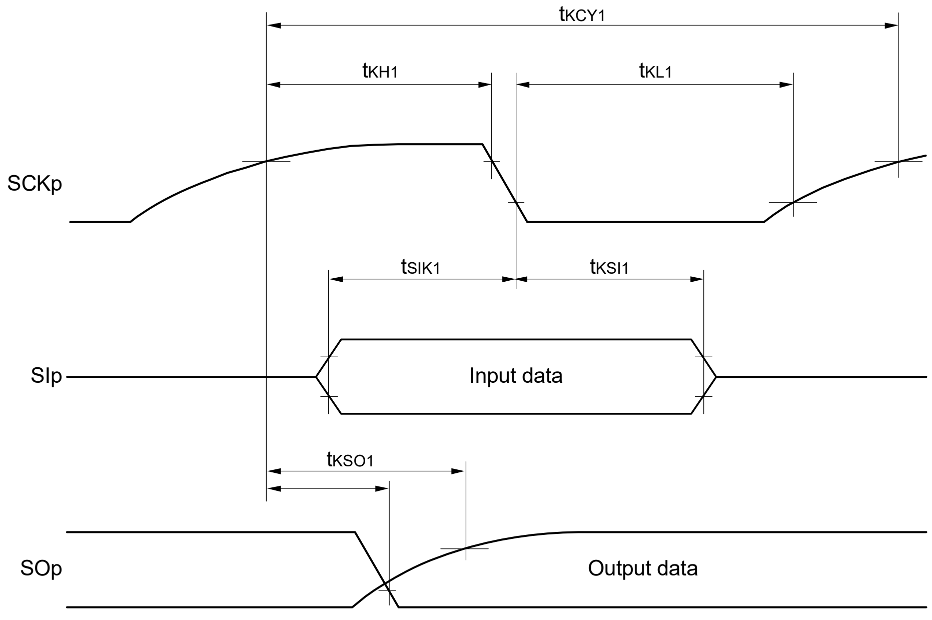
Remark 2. p: CSI number (p = 00, 01), m: Unit number (m = 0), n: Channel number (n = 0, 1), g: PIM and POM numbers (g = 5)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number,
n: Channel number (mn = 00, 01))

Simplified SPI (CSI) mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remark p: CSI number (p = 00, 01), m: Unit number (m = 0), n: Channel number (n = 0, 1), g: PIM and POM numbers (g = 5)

(9) Communication at different potential (1.8 V, 2.5 V) (simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

(T_A = -40 to 85°C, 1.8 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode MIN.	HS (high-speed main) Mode MAX.	LS (low-speed main) Mode MIN.	LS (low-speed main) Mode MAX.	LP (Low-power main) mode MIN.	LP (Low-power main) mode MAX.	LV (low-voltage main) Mode MIN.	LV (low-voltage main) Mode MAX.	Unit
SCKp cycle time Note 1	t_{KCY2}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V	20 MHz < f_{MCK} ≤ 24 MHz	16/f_{MCK}		—		—		—		ns
			16 MHz < f_{MCK} ≤ 20 MHz	14/f_{MCK}		—		—		—		ns
			8 MHz < f_{MCK} ≤ 16 MHz	12/f_{MCK}		—		—		—		ns
			4 MHz < f_{MCK} ≤ 8 MHz	8/f_{MCK}		16/f_{MCK}		—		—		ns
			f_{MCK} ≤ 4 MHz	6/f_{MCK}		10/f_{MCK}		10/f_{MCK}		10/f_{MCK}		ns
		1.8 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2	20 MHz < f_{MCK} ≤ 24 MHz	36/f_{MCK}		—		—		—		ns
			16 MHz < f_{MCK} ≤ 20 MHz	32/f_{MCK}		—		—		—		ns
			8 MHz < f_{MCK} ≤ 16 MHz	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}		10/f_{MCK}		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V		t_{KCY2}/2 - 18		t_{KCY2}/2 - 50		t_{KCY2}/2 - 50		t_{KCY2}/2 - 50		ns
		1.8 V ≤ V_{DD} < 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		t_{KCY2}/2 - 50		ns
SIp setup time (to SCKp↑) Note 3	t_{SIK2}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V		1/f_{MCK} + 20		1/f_{MCK} + 30		1/f_{MCK} + 30		1/f_{MCK} + 30		ns
		1.8 V ≤ V_{DD} < 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		1/f_{MCK} + 30		ns
SIp hold time (from SCKp↑) Note 4	t_{KSI2}			1/f_{MCK} + 31		1/f_{MCK} + 31		1/f_{MCK} + 31		1/f_{MCK} + 31		ns
Delay time from SCKp↓ to SOp output Note 5	t_{KSO2}	2.7 V ≤ V_{DD} ≤ 3.6 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} + 573		2/f_{MCK} + 573		2/f_{MCK} + 573	ns
		1.8 V ≤ V_{DD} < 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		2/f_{MCK} + 573	ns

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

Note 1. Transfer rate in the SNOOZE mode: MAX. 1 Mbps

Note 2. Use it with $V_{DD} \geq V_b$.

Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

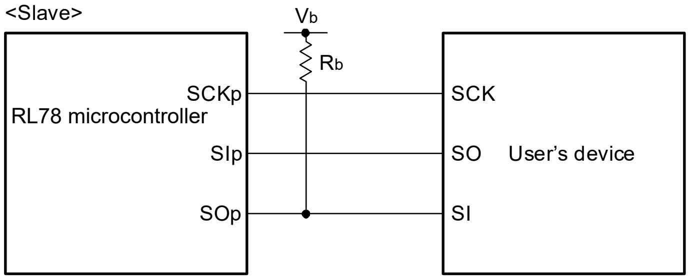
Note 4. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 5. 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 SCKp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

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

Simplified SPI (CSI) mode connection diagram (during communication at different potential)

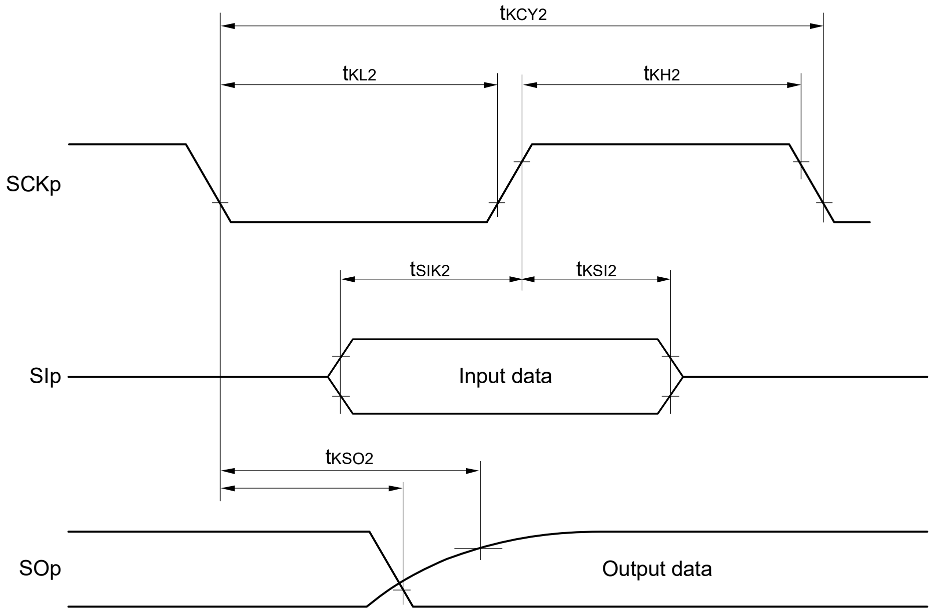


Remark 1. R_b[Ω]: Communication line (SOp) pull-up resistance, C_b[F]: Communication line (SOp) load capacitance, V_b[V]: Communication line voltage

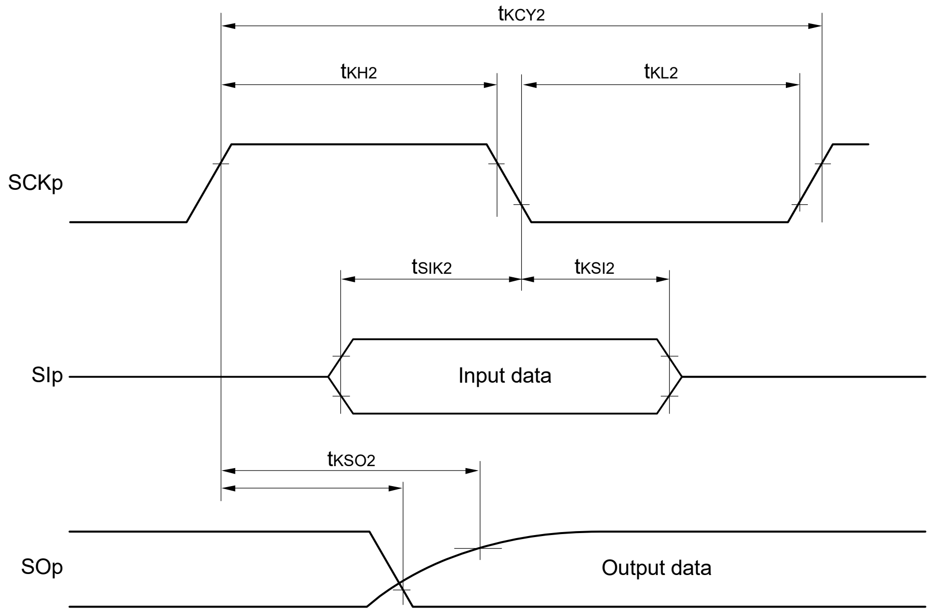
Remark 2. p: CSI number (p = 00, 01), m: Unit number (m = 0), n: Channel number (n = 0, 1), g: PIM and POM numbers (g = 5)

Remark 3. f_{MCK}: Serial array unit operation clock frequency (Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 01))

Simplified SPI (CSI) mode serial transfer timing (slave mode) (during communication at different potential) (When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (slave mode) (during communication at different potential) (When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remark p: CSI number (p = 00, 01), m: Unit number (m = 0), n: Channel number (n = 0, 1), g: PIM and POM numbers (g = 5)

(9) Communication at different potential (1.8 V, 2.5 V) (simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

(T_A = +85 to 105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode MIN.	HS (high-speed main) Mode MAX.	Unit
SCKp cycle time Note 1	t_{KCY2}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V	20 MHz < f_{MCK} ≤ 24 MHz	32/f_{MCK}		ns
			16 MHz < f_{MCK} ≤ 20 MHz	28/f_{MCK}		ns
			8 MHz < f_{MCK} ≤ 16 MHz	24/f_{MCK}		ns
			4 MHz < f_{MCK} ≤ 8 MHz	16/f_{MCK}		ns
			f_{MCK} ≤ 4 MHz	12/f_{MCK}		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2	20 MHz < f_{MCK} ≤ 24 MHz	72/f_{MCK}		ns
			16 MHz < f_{MCK} ≤ 20 MHz	64/f_{MCK}		ns
			8 MHz < f_{MCK} ≤ 16 MHz	52/f_{MCK}		ns
			4 MHz < f_{MCK} ≤ 8 MHz	32/f_{MCK}		ns
			f_{MCK} ≤ 4 MHz	20/f_{MCK}		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V		t_{KCY2}/2 - 36		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2		t_{KCY2}/2 - 100		ns
SIp setup time (to SCKp↑) Note 3	t_{SIK2}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V		1/f_{MCK} + 40		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2		1/f_{MCK} + 60		ns
SIp hold time (from SCKp↑) Note 4	t_{KSI2}			1/f_{MCK} + 62		ns
Delay time from SCKp↓ to SOp output Note 5	t_{KSO2}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V C_b = 30 pF, R_b = 2.7 kΩ			2/f_{MCK} + 428	ns
		2.4 V ≤ V_{DD} < 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} + 1146	ns

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

Note 1. Transfer rate in the SNOOZE mode: MAX. 1 Mbps

Note 2. Use it with $V_{DD} \geq V_b$.

Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

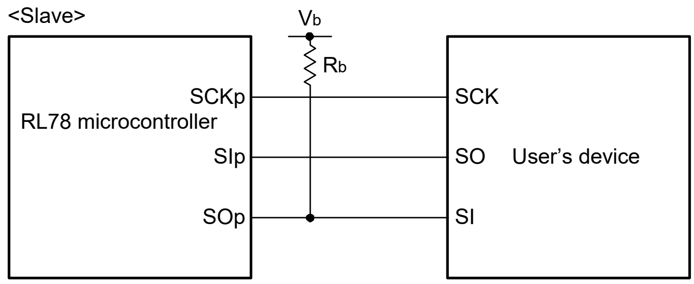
Note 4. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 5. 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 SCKp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

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

Simplified SPI (CSI) mode connection diagram (during communication at different potential)

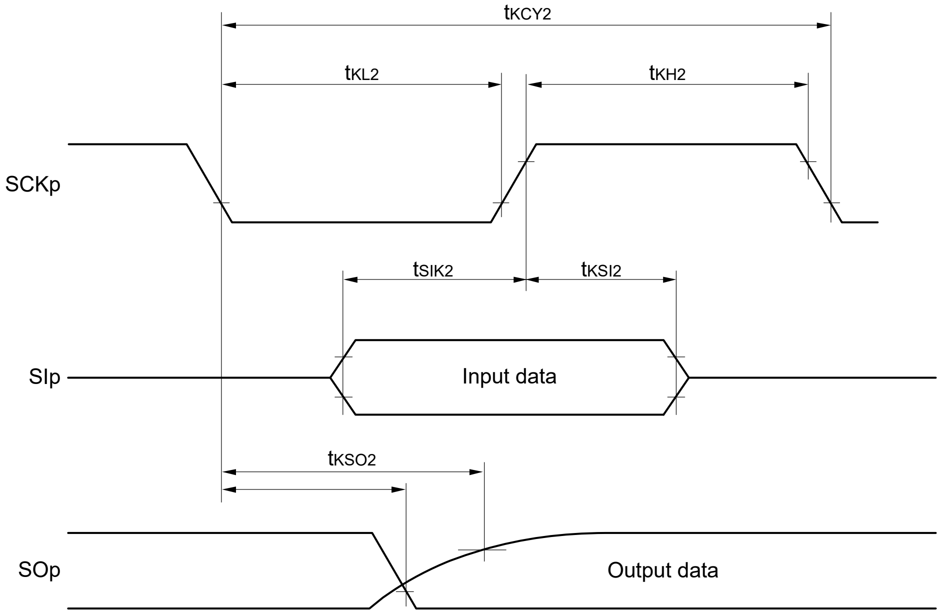


Remark 1. $R_b[\Omega]$: Communication line (SOp) pull-up resistance, $C_b[F]$: Communication line (SOp) load capacitance, $V_b[V]$: Communication line voltage

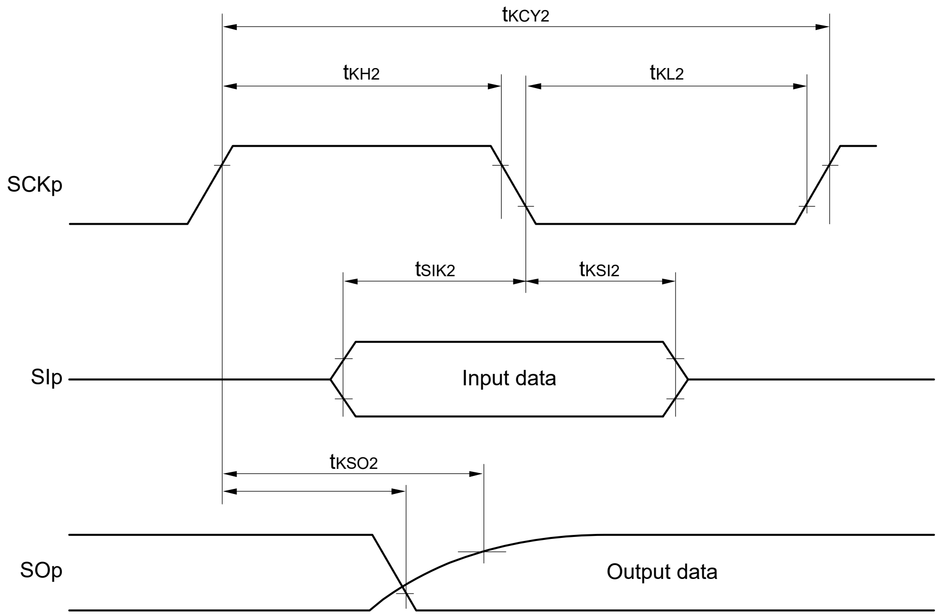
Remark 2. p: CSI number (p = 00, 01), m: Unit number (m = 0), n: Channel number (n = 0, 1), g: PIM and POM numbers (g = 5)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 01))

Simplified SPI (CSI) mode serial transfer timing (slave mode) (during communication at different potential) (When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (slave mode) (during communication at different potential) (When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remark p: CSI number (p = 00, 01), m: Unit number (m = 0), n: Channel number (n = 0, 1), g: PIM and POM numbers (g = 5)

(10) Communication at different potential (1.8 V, 2.5 V) (simplified I2C mode)

(T_A = -40 to 85°C, 1.8 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

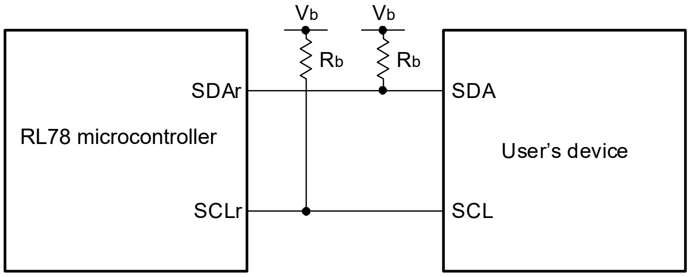
Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LP (Low-power main) mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCLr clock frequency	f_{SCL}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ		1000 Note 1		300 Note 1		250 Note 1		300 Note 1	kHz
		2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ		400 Note 1		300 Note 1		250 Note 1		300 Note 1	kHz
		1.8 V ≤ V_{DD} < 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		250 Note 1		300 Note 1	kHz
Hold time when SCLr = "L"	t_{LOW}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	475		1550		1550		1550		ns
		2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	1150		1550		1550		1550		ns
		1.8 V ≤ V_{DD} < 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		1550		ns
Hold time when SCLr = "H"	t_{HIGH}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	200		610		610		610		ns
		2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	600		610		610		610		ns
		1.8 V ≤ V_{DD} < 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		610		ns
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	$1/f_{MCK}$ + 135 Note 3		$1/f_{MCK}$ + 190 Note 2		$1/f_{MCK}$ + 190 Note 3		$1/f_{MCK}$ + 190 Note 3		ns
		2.7 V ≤ V_{DD} ≤ 3.6 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		$1/f_{MCK}$ + 190 Note 3		ns
		1.8 V ≤ V_{DD} < 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		$1/f_{MCK}$ + 190 Note 3		ns
Data hold time (transmission)	$t_{HD:DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	0	305	0	305	0	305	0	305	ns
		2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	0	355	0	355	0	355	0	355	ns
		1.8 V ≤ V_{DD} < 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	0	405	ns

Note 1. The value must also be equal to or less than f_{MCK}/4.
Note 2. Use it with V_{DD} ≥ V_b.
Note 3. Set the f_{MCK} value to keep the hold time of SCLr = "L" and SCLr = "H".

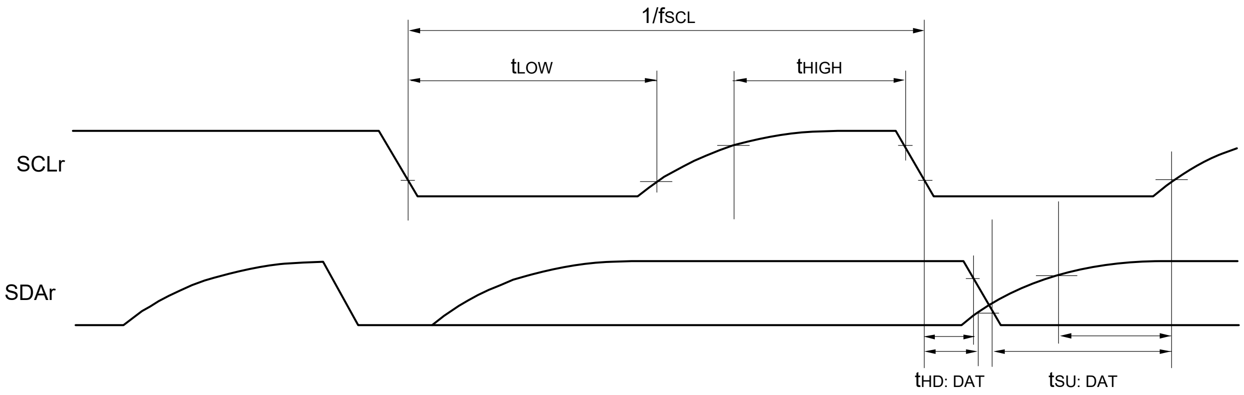
Caution Select the TTL input buffer and the N-ch open drain output (V_{DD} tolerance) mode for the SDAr pin and the N-ch open drain output (V_{DD} tolerance) mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

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

Simplified I²C mode connection diagram (during communication at different potential)



Simplified I²C mode serial transfer timing (during communication at different potential)



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), g: PIM, POM number (g = 5)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01)

(10) Communication at different potential (1.8 V, 2.5 V) (simplified I²C mode)

(T_A = +85 to 105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
SCLr clock frequency	f_{SCL}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ		400 Note 1	kHz
		2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ		100 Note 1	kHz
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ		100 Note 1	kHz
Hold time when SCLr = "L"	t_{LOW}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	1200		ns
		2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	4600		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ	4650		ns
Hold time when SCLr = "H"	t_{HIGH}	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	500		ns
		2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	2400		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ	1830		ns
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	$1/f_{MCK}$ + 340 Note 3		ns
		2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	$1/f_{MCK}$ + 760 Note 3		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ	$1/f_{MCK}$ + 570 Note 3		ns
Data hold time (transmission)	$t_{HD:DAT}$	2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	0	770	ns
		2.7 V ≤ V_{DD} ≤ 3.6 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	0	1420	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 2, C_b = 100 pF, R_b = 5.5 kΩ	0	1215	ns

Note 1. The value must also be equal to or less than $f_{MCK}/4$.

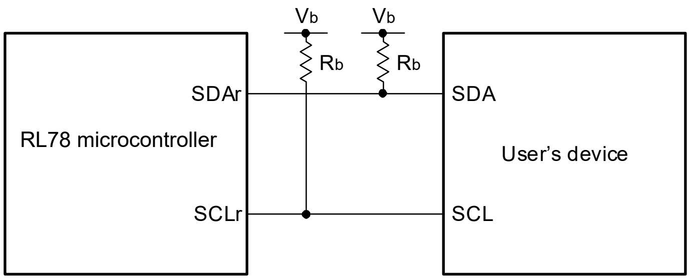
Note 2. Use it with V_{DD} ≥ V_b.

Note 3. Set the f_{MCK} value to keep the hold time of SCLr = "L" and SCLr = "H".

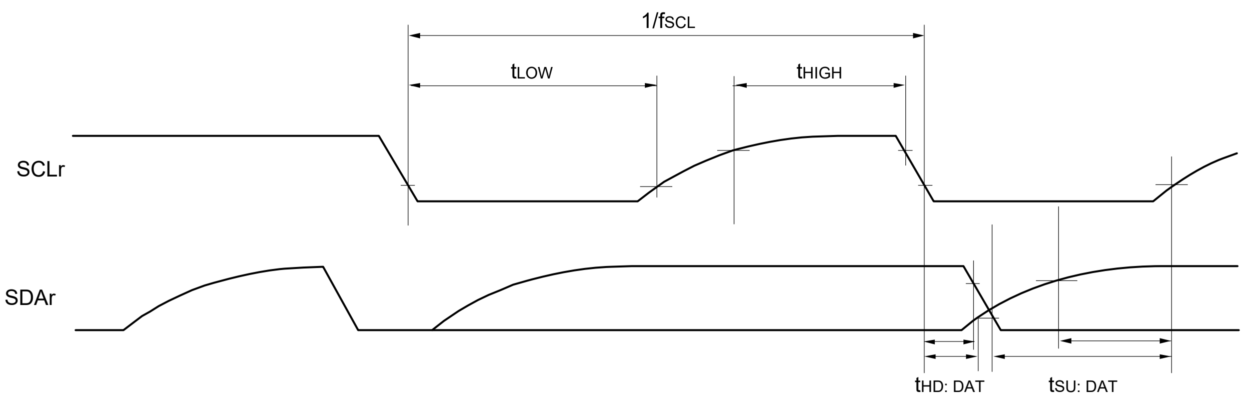
Caution **Select the TTL input buffer and the N-ch open drain output (V_{DD} tolerance) mode for the SDAr pin and the N-ch open drain output (V_{DD} tolerance) mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.**

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

Simplified I2C mode connection diagram (during communication at different potential)



Simplified I2C mode serial transfer timing (during communication at different potential)



Remark 1. Rb[Ω]: Communication line (SDAr, SCLr) pull-up resistance, Cb[F]: Communication line (SDAr, SCLr) load capacitance, Vb[V]: Communication line voltage

Remark 2. r: IIC number (r = 00, 01), g: PIM and POM numbers (g = 5)

Remark 3. fMCK: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number (m = 0),
n: Channel number (n = 0, 1), mn = 00, 01)

34.6 Analog Characteristics

34.6.1 A/D converter characteristics

Classification of A/D converter characteristics

Reference Voltage / Input Channel	Reference voltage (+) = AV_{REFP} Reference voltage (-) = AV_{REFM}	Reference voltage (+) = AV_{DD} Reference voltage (-) = AV_{SS}	Reference voltage (+) = Internal reference voltage Reference voltage (-) = AV_{SS}
High-accuracy channel; ANI0 to ANI13 (input buffer power supply: AV_{DD})	Refer to **34.6.1 (1)**. Refer to **34.6.1 (7)**.	Refer to **34.6.1 (2)**. Refer to **34.6.1 (7)**.	Refer to **34.6.1 (5)**. Refer to **34.6.1 (10)**.
Standard channel; ANI16 to ANI18 (input buffer power supply: V_{DD})	Refer to **34.6.1 (3)**. Refer to **34.6.1 (8)**.	Refer to **34.6.1 (4)**. Refer to **34.6.1 (9)**.	
Internal reference voltage, Temperature sensor output voltage	Refer to **34.6.1 (3)**. Refer to **34.6.1 (8)**.	Refer to **34.6.1 (4)**. Refer to **34.6.1 (9)**.	—

(1) When reference voltage (+) = AVREFP/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (-) = AVREFM/ANI1 (ADREFM = 1), conversion target: ANI2 to ANI13

(TA = -40 to +85°C, 1.6 V ≤ AVREFP ≤ AVDD = VDD ≤ 3.6 V, VSS = 0 V, AVSS = 0 V, Reference voltage (+) = AVREFP, Reference voltage (-) = AVREFM = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES		2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	8		12	bit
			1.8 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	8		10 Note 1	
			1.6 V ≤ AVREFP ≤ AVDD ≤ 3.6 V		8 Note 2		
Overall error Note 3	AINL	12-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±6.0	LSB
		10-bit resolution	1.8 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±5.0	
		8-bit resolution	1.6 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±2.5	
Conversion time	tCONV	ADTYP = 0, 12-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	3.375			μs
		ADTYP = 0, 10-bit resolution Note 1	1.8 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	6.75			
		ADTYP = 0, 8-bit resolution Note 2	1.6 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	13.5			
		ADTYP = 1, 8-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	2.5625			
			1.8 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	5.125			
			1.6 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	10.25			
Zero-scale error Note 3	EZS	12-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±4.5	LSB
		10-bit resolution	1.8 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±4.5	
		8-bit resolution	1.6 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±2.0	
Full-scale error Note 3	EFS	12-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±4.5	LSB
		10-bit resolution	1.8 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±4.5	
		8-bit resolution	1.6 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±2.0	
Integral linearity error Note 3	ILE	12-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±2.0	LSB
		10-bit resolution	1.8 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±1.5	
		8-bit resolution	1.6 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±1.0	
Differential linearity error Note 3	DLE	12-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±1.5	LSB
		10-bit resolution	1.8 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±1.5	
		8-bit resolution	1.6 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±1.0	
Analog input voltage	VAIN			0		AVREFP	V

Note 1. Cannot be used for lower 2 bit of ADCR register

Note 2. Cannot be used for lower 4 bit of ADCR register

Note 3. Excludes quantization error (±1/2 LSB).

Caution Always use AVDD pin with the same potential as the VDD pin.

(2) When reference voltage (+) = AV_{DD} (ADREFP1 = 0, ADREFP0 = 0), reference voltage (-) = AV_{SS} (ADREFM = 0), conversion target: ANI0 to ANI13

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = 0 V, AV_{SS} = 0 V, Reference voltage (+) = AV_{DD}, Reference voltage (-) = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES		2.4 V ≤ AV_{DD} ≤ 3.6 V	8		12	bit
			1.8 V ≤ AV_{DD} ≤ 3.6 V	8		10 Note 1	
			1.6 V ≤ AV_{DD} ≤ 3.6 V		8 Note 2		
Overall error Note 3	AINL	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±7.5	LSB
		10-bit resolution	1.8 V ≤ AV_{DD} ≤ 3.6 V			±5.5	
		8-bit resolution	1.6 V ≤ AV_{DD} ≤ 3.6 V			±3.0	
Conversion time	t_{CONV}	ADTYP = 0, 12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V	3.375			μs
		ADTYP = 0, 10-bit resolution Note 1	1.8 V ≤ AV_{DD} ≤ 3.6 V	6.75			
		ADTYP = 0, 8-bit resolution Note 2	1.6 V ≤ AV_{DD} ≤ 3.6 V	13.5			
		ADTYP = 1, 8-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V	2.5625			
			1.8 V ≤ AV_{DD} ≤ 3.6 V	5.125			
			1.6 V ≤ AV_{DD} ≤ 3.6 V	10.25			
Zero-scale error Note 3	EZS	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±6.0	LSB
		10-bit resolution	1.8 V ≤ AV_{DD} ≤ 3.6 V			±5.0	
		8-bit resolution	1.6 V ≤ AV_{DD} ≤ 3.6 V			±2.5	
Full-scale error Note 3	EFS	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±6.0	LSB
		10-bit resolution	1.8 V ≤ AV_{DD} ≤ 3.6 V			±5.0	
		8-bit resolution	1.6 V ≤ AV_{DD} ≤ 3.6 V			±2.5	
Integral linearity error Note 3	ILE	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±3.0	LSB
		10-bit resolution	1.8 V ≤ AV_{DD} ≤ 3.6 V			±2.0	
		8-bit resolution	1.6 V ≤ AV_{DD} ≤ 3.6 V			±1.5	
Differential linearity error Note 3	DLE	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±2.0	LSB
		10-bit resolution	1.8 V ≤ AV_{DD} ≤ 3.6 V			±2.0	
		8-bit resolution	1.6 V ≤ AV_{DD} ≤ 3.6 V			±1.5	
Analog input voltage	V_{AIN}	ANI0 to ANI6		0		AV_{DD}	V

Note 1. Cannot be used for lower 2 bit of ADCR register

Note 2. Cannot be used for lower 4 bit of ADCR register

Note 3. Excludes quantization error (±1/2 LSB).

Caution Always use AV_{DD} pin with the same potential as the V_{DD} pin.

(3) When reference voltage (+) = AVREFP/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (-) = AVREFM/ANI1 (ADREFM = 1), conversion target: ANI16 to ANI18, internal reference voltage, temperature sensor output voltage

(TA = -40 to +85°C, 1.6 V ≤ VDD ≤ 3.6 V, 1.6 V ≤ AVREFP ≤ AVDD = VDD ≤ 3.6 V, VSS = 0 V, AVSS = 0 V, Reference voltage (+) = AVREFP, Reference voltage (-) = AVREFM = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES		2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	8		12	bit
			1.8 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	8		10 Note 1	
			1.6 V ≤ AVREFP ≤ AVDD ≤ 3.6 V		8 Note 2		
Overall error Note 3	AINL	12-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±7.0	LSB
		10-bit resolution	1.8 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±5.5	
		8-bit resolution	1.6 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±3.0	
Conversion time	tCONV	ADTYP = 0, 12-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	4.125			μs
		ADTYP = 0, 10-bit resolution Note 1	1.8 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	9.5			
		ADTYP = 0, 8-bit resolution Note 2	1.6 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	57.5			
		ADTYP = 1, 8-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	3.3125			
			1.8 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	7.875			
			1.6 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	54.25			
Zero-scale error Note 3	EZS	12-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±5.0	LSB
		10-bit resolution	1.8 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±5.0	
		8-bit resolution	1.6 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±2.5	
Full-scale error Note 3	EFS	12-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±5.0	LSB
		10-bit resolution	1.8 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±5.0	
		8-bit resolution	1.6 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±2.5	
Integral linearity error Note 3	ILE	12-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±3.0	LSB
		10-bit resolution	1.8 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±2.0	
		8-bit resolution	1.6 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±1.5	
Differential linearity error Note 3	DLE	12-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±2.0	LSB
		10-bit resolution	1.8 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±2.0	
		8-bit resolution	1.6 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±1.5	
Analog input voltage	VAIN			0		AVREFP	V
		Internal reference voltage (1.8 V ≤ VDD ≤ 3.6 V)			VBGR Note 4		
		Temperature sensor output voltage (1.8 V ≤ VDD ≤ 3.6 V)			VTMP25 Note 4		

Note 1. Cannot be used for lower 2 bits of ADCR register
Note 2. Cannot be used for lower 4 bits of ADCR register
Note 3. Excludes quantization error (±1/2 LSB).
Note 4. Refer to **34.6.2 Temperature sensor, internal reference voltage output characteristics**.

Caution Always use AVDD pin with the same potential as the VDD pin.

(4) When reference voltage (+) = AVDD (ADREFP1 = 0, ADREFP0 = 0), reference voltage (-) = AVSS (ADREFM = 0), conversion target: ANI16 to ANI18, internal reference voltage, temperature sensor output voltage

(T_A = -40 to +85°C, 1.6 V ≤ AVDD = VDD ≤ 3.6 V, VSS = 0 V, AVSS = 0 V, Reference voltage (+) = AVDD, Reference voltage (-) = AVSS = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES		2.4 V ≤ AVDD ≤ 3.6 V	8		12	bit
			1.8 V ≤ AVDD ≤ 3.6 V	8		10 Note 1	
			1.6 V ≤ AVDD ≤ 3.6 V		8 Note 2		
Overall error Note 3	AINL	12-bit resolution	2.4 V ≤ AVDD ≤ 3.6 V			±8.5	LSB
		10-bit resolution	1.8 V ≤ AVDD ≤ 3.6 V			±6.0	
		8-bit resolution	1.6 V ≤ AVDD ≤ 3.6 V			±3.5	
Conversion time	tCONV	ADTYP = 0, 12-bit resolution	2.4 V ≤ AVDD ≤ 3.6 V	4.125			μs
		ADTYP = 0, 10-bit resolution Note 1	1.8 V ≤ AVDD ≤ 3.6 V	9.5			
		ADTYP = 0, 8-bit resolution Note 2	1.6 V ≤ AVDD ≤ 3.6 V	57.5			
		ADTYP = 1, 8-bit resolution	2.4 V ≤ AVDD ≤ 3.6 V	3.3125			
			1.8 V ≤ AVDD ≤ 3.6 V	7.875			
			1.6 V ≤ AVDD ≤ 3.6 V	54.25			
Zero-scale error Note 3	EZS	12-bit resolution	2.4 V ≤ AVDD ≤ 3.6 V			±8.0	LSB
		10-bit resolution	1.8 V ≤ AVDD ≤ 3.6 V			±5.5	
		8-bit resolution	1.6 V ≤ AVDD ≤ 3.6 V			±3.0	
Full-scale error Note 3	EFS	12-bit resolution	2.4 V ≤ AVDD ≤ 3.6 V			±8.0	LSB
		10-bit resolution	1.8 V ≤ AVDD ≤ 3.6 V			±5.5	
		8-bit resolution	1.6 V ≤ AVDD ≤ 3.6 V			±3.0	
Integral linearity error Note 3	ILE	12-bit resolution	2.4 V ≤ AVDD ≤ 3.6 V			±3.5	LSB
		10-bit resolution	1.8 V ≤ AVDD ≤ 3.6 V			±2.5	
		8-bit resolution	1.6 V ≤ AVDD ≤ 3.6 V			±1.5	
Differential linearity error Note 3	DLE	12-bit resolution	2.4 V ≤ AVDD ≤ 3.6 V			±2.5	LSB
		10-bit resolution	1.8 V ≤ AVDD ≤ 3.6 V			±2.5	
		8-bit resolution	1.6 V ≤ AVDD ≤ 3.6 V			±2.0	
Analog input voltage	VAIN			0		AVDD	V
		Internal reference voltage (1.8 V ≤ VDD ≤ 3.6 V)			VBGR Note 4		
		Temperature sensor output voltage (1.8 V ≤ VDD ≤ 3.6 V)			VTMP25 Note 4		

Note 1. Cannot be used for lower 2 bits of ADCR register

Note 2. Cannot be used for lower 4 bits of ADCR register

Note 3. Excludes quantization error (±1/2 LSB).

Note 4. Refer to **34.6.2 Temperature sensor, internal reference voltage output characteristics**.

Caution Always use AVDD pin with the same potential as the VDD pin.

(5) **When reference voltage (+) = Internal reference voltage (1.45 V) (ADREFP1 = 1, ADREFP0 = 0), reference voltage (-) = AVSS (ADREFM = 0), conversion target: ANI0 to ANI13, ANI16 to ANI18**

(T_A = -40 to +85°C, 1.8 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = 0 V, AV_{SS} = 0 V, Reference voltage (+) = internal reference voltage, Reference voltage (-) = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	RES			8		bit
Conversion time	t_{CONV}	8-bit resolution	16			μs
Zero-scale error Note	EZS	8-bit resolution			±4.0	LSB
Integral linearity error Note	ILE	8-bit resolution			±2.0	LSB
Differential linearity error Note	DLE	8-bit resolution			±2.5	LSB
Analog input voltage	V_{AIN}		0		V_{BGR}	V

Note Excludes quantization error (±1/2 LSB).

Caution Always use AV_{DD} pin with the same potential as the V_{DD} pin.

(6) **When reference voltage (+) = AVREFP/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (-) = AVREFM/ANI1 (ADREFM = 1), conversion target: ANI2 to ANI13**

(T_A = +85 to +105°C, 2.4 V ≤ AV_{REFP} ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = 0 V, AV_{SS} = 0 V, Reference voltage (+) = AV_{REFP}, Reference voltage (-) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES		2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V	8		12	bit
Overall error Note	AINL	12-bit resolution	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±6.0	LSB
Conversion time	t_{CONV}	ADTYP = 0, 12-bit resolution	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V	3.375			μs
Zero-scale error Note	EZS	12-bit resolution	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±4.5	LSB
Full-scale error Note	EFS	12-bit resolution	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±4.5	LSB
Integral linearity error Note	ILE	12-bit resolution	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±2.0	LSB
Differential linearity error Note	DLE	12-bit resolution	2.4 V ≤ AV_{REFP} ≤ AV_{DD} ≤ 3.6 V			±1.5	LSB
Analog input voltage	V_{AIN}			0		AV_{REFP}	V

Note Excludes quantization error (±1/2 LSB).

Caution Always use AV_{DD} pin with the same potential as the V_{DD} pin.

(7) When reference voltage (+) = AV$_{DD}$ (ADREFP1 = 0, ADREFP0 = 0), reference voltage (-) = AV$_{SS}$ (ADREFM = 0), conversion target: ANI0 to ANI13

(T_A = +85 to +105°C, 2.4 V ≤ AV$_{DD}$ = V$_{DD}$ ≤ 3.6 V, V$_{SS}$ = 0 V, AV$_{SS}$ = 0 V, Reference voltage (+) = AV$_{DD}$, Reference voltage (-) = AV$_{SS}$ = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES		2.4 V ≤ AV$_{DD}$ ≤ 3.6 V	8		12	bit
Overall error Note	AINL	12-bit resolution	2.4 V ≤ AV$_{DD}$ ≤ 3.6 V			±7.5	LSB
Conversion time	t_{CONV}	ADTYP = 0, 12-bit resolution	2.4 V ≤ AV$_{DD}$ ≤ 3.6 V	3.375			μs
Zero-scale error Note	E_{ZS}	12-bit resolution	2.4 V ≤ AV$_{DD}$ ≤ 3.6 V			±6.0	LSB
Full-scale error Note	E_{FS}	12-bit resolution	2.4 V ≤ AV$_{DD}$ ≤ 3.6 V			±6.0	LSB
Integral linearity error Note	ILE	12-bit resolution	2.4 V ≤ AV$_{DD}$ ≤ 3.6 V			±3.0	LSB
Differential linearity error Note	DLE	12-bit resolution	2.4 V ≤ AV$_{DD}$ ≤ 3.6 V			±2.0	LSB
Analog input voltage	V_{AIN}			0		AV$_{DD}$	V

Note Excludes quantization error (±1/2 LSB).

Caution Always use AV$_{DD}$ pin with the same potential as the V$_{DD}$ pin.

(8) When reference voltage (+) = AVREFP/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (-) = AVREFM/ANI1 (ADREFM = 1), conversion target ANI16 to ANI18, internal reference voltage, temperature sensor output voltage

(TA = +85 to +105°C, 2.4 V ≤ AVREFP ≤ AVDD = VDD ≤ 3.6 V, VSS = 0 V, AVSS = 0 V, Reference voltage (+) = AVREFP, Reference voltage (-) = AVREFM = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES		2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	8		12	bit
Overall error Note 1	AINL	12-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±7.0	LSB
Conversion time	tCONV	ADTYP = 0, 12-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V	4.125			μs
Zero-scale error Note 1	EZS	12-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±5.0	LSB
Full-scale error Note 1	EFS	12-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±5.0	LSB
Integral linearity error Note 1	ILE	12-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±3.0	LSB
Differential linearity error Note 1	DLE	12-bit resolution	2.4 V ≤ AVREFP ≤ AVDD ≤ 3.6 V			±2.0	LSB
Analog input voltage	VAIN			0		AVREFP	V
		Internal reference voltage (2.4 V ≤ VDD ≤ 3.6 V)		VBGR Note 2			
		Temperature sensor output voltage (2.4 V ≤ VDD ≤ 3.6 V)		VTMP25 Note 2			

Note 1. Excludes quantization error (±1/2 LSB).

Note 2. Refer to **34.6.2 Temperature sensor, internal reference voltage output characteristics**.

Caution Always use AVDD pin with the same potential as the VDD pin.

(9) When reference voltage (+) = AV_{DD} (ADREFP1 = 0, ADREFP0 = 0), reference voltage (-) = AV_{SS} (ADREFM = 0), conversion target: ANI16 to ANI18, internal reference voltage, temperature sensor output voltage

(T_A = +85 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = 0 V, AV_{SS} = 0 V, Reference voltage (+) = AV_{DD}, Reference voltage (-) = AV_{SS} = 0)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	R_{ES}		2.4 V ≤ AV_{DD} ≤ 3.6 V	8		12	bit
Overall error Note 1	AINL	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±8.5	LSB
Conversion time	t_{CONV}	ADTYP = 0, 12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V	4.125			μs
Zero-scale error Note 1	E_{ZS}	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±8.0	LSB
Full-scale error Note 1	E_{FS}	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±8.0	LSB
Integral linearity error Note 1	ILE	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±3.5	LSB
Differential linearity error Note 1	DLE	12-bit resolution	2.4 V ≤ AV_{DD} ≤ 3.6 V			±2.5	LSB
Analog input voltage	V_{AIN}			0		AV_{DD}	V
		Internal reference voltage (2.4 V ≤ V_{DD} ≤ 3.6 V)		V_{BGR} Note 2			
		Temperature sensor output voltage (2.4 V ≤ V_{DD} ≤ 3.6 V)		V_{TMP25} Note 2			

Note 1. Excludes quantization error (±1/2 LSB).

Note 2. Refer to **34.6.2 Temperature sensor, internal reference voltage output characteristics**.

Caution Always use AV_{DD} pin with the same potential as the V_{DD} pin.

(10) When reference voltage (+) = Internal reference voltage (1.45 V) (ADREFP1 = 1, ADREFP0 = 0), reference voltage (-) = AVSS (ADREFM = 0), conversion target: ANI0 to ANI13, ANI16 to ANI18

(TA = +85 to +105°C, 2.4 V ≤ VDD, 2.4 V ≤ AVDD = VDD ≤ 3.6 V, VSS = 0 V, AVSS = 0 V, Reference voltage (+) = internal reference voltage, Reference voltage (-) = AVSS = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	RES			8		bit
Conversion time	tCONV	8-bit resolution	16.0			μs
Zero-scale error Note	EZS	8-bit resolution			±4.0	LSB
Integral linearity error Note	ILE	8-bit resolution			±2.0	LSB
Differential linearity error Note	DLE	8-bit resolution			±2.5	LSB
Analog input voltage	VAIN		0		VBGR	V

Note Excludes quantization error (±1/2 LSB).

Caution Always use AVDD pin with the same potential as the VDD pin.

34.6.2 Temperature sensor, internal reference voltage output characteristics

(TA = -40 to 85°C, 1.8 V ≤ AVDD = VDD ≤ 3.6 V, VSS = AVSS = 0 V)
(TA = +85 to 105°C, 2.4 V ≤ AVDD = VDD ≤ 3.6 V, VSS = AVSS = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Temperature sensor output voltage	VTMPS25	Setting ADS register = 80H, TA = +25°C		1.05		V
Internal reference voltage	VBGR	Setting ADS register = 81H	1.38	1.45	1.50	V
Temperature coefficient	FVTMPS	Temperature sensor output voltage that depends on the temperature		-3.6		mV/°C
Operation stabilization wait time	tAMP	2.4 V ≤ VDD ≤ 3.6 V	5			μs
		1.8 V ≤ VDD < 2.4 V	10			

34.6.3 Comparator

(T_A = -40 to +85°C, 1.6 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)
(T_A = +85 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage range	Ivref0	IVREF0 pin		0		V_{DD} - 1.4 Note	V
	Ivref1	IVREF1 pin		1.4 Note		V_{DD}	V
	Ivcmp	IVCMP0, IVCMP1 pins		-0.3		V_{DD} + 0.3	V
Output delay	td	AV_{DD} = 3.0 V Input slew rate > 50 mV/μs	Comparator high-speed mode, standard mode			1.2	μs
			Comparator high-speed mode, window mode			2.0	μs
			Comparator low-speed mode, standard mode		3.0		μs
			Comparator low-speed mode, window mode		4		μs
Operation stabilization wait time	t_{CMP}			100			μs

Note In window mode, make sure that Vref1 - Vref0 ≥ 0.2 V.

34.6.4 Operational amplifier characteristics

(TA = -40 to +85°C, 1.8 V ≤ AVDD = VDD ≤ 3.6 V, VSS = AVSS = 0 V)
(TA = +85 to +105°C, 2.4 V ≤ AVDD = VDD ≤ 3.6 V, VSS = AVSS = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Common mode input range	Vicm1	Low-power consumption mode		0.2		AVDD - 0.5	V
	Vicm2	High-speed mode		0.3		AVDD - 0.6	V
Output voltage range	Vo1	Low-power consumption mode		0.1		AVDD - 0.1	V
	Vo2	High-speed mode		0.1		AVDD - 0.1	V
Input offset voltage	Vioff			-10		10	mV
Open gain	Av			60	120		dB
Gain-bandwidth (GB) product	GBW1	Low-power consumption mode			0.04		MHz
	GBW2	High-speed mode			1.7		MHz
Phase margin	PM	CL = 20 pF		50			deg
Gain margin	GM	CL = 20 pF		10			dB
Equivalent input noise	Vnoise1	f = 1 kHz	Low-power consumption mode		230		nV/√Hz
	Vnoise2	f = 10 kHz			200		nV/√Hz
	Vnoise3	f = 1 kHz	High-speed mode		90		nV/√Hz
	Vnoise4	f = 2 kHz			70		nV/√Hz
Power supply reduction ratio	PSRR				90		dB
Common mode signal reduction ratio	CMRR				90		dB
Operation stabilization wait time	Tstd1	CL = 20 pF Only operational amplifier is activated Note	Low-power consumption mode	650			μs
	Tstd2		High-speed mode	13			μs
	Tstd3	CL = 20 pF Operational amplifier and reference current circuit are activated simultaneously	Low-power consumption mode	650			μs
	Tstd4		High-speed mode	13			μs
Settling time	Tset1	CL = 20 pF	Low-power consumption mode			750	μs
	Tset2		High-speed mode			13	μs
Slew rate	Tslew1	CL = 20 pF	Low-power consumption mode		0.02		V/μs
	Tslew2		High-speed mode		1.1		V/μs
Load current	Iload1	Low-power consumption mode		-100		100	μA
	Iload2	High-speed mode		-100		100	μA
Load capacitance	CL					20	pF

Note When the operational amplifier reference current circuit is activated in advance.

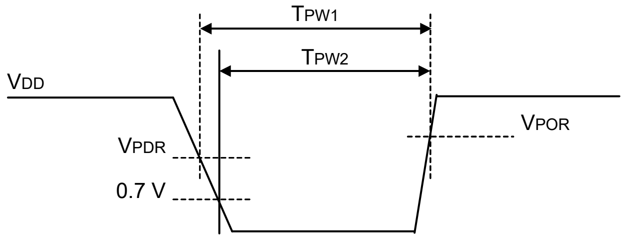
34.6.5 POR circuit characteristics

(T_A = -40 to +105°C, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POR}	The power supply voltage is rising.	T_A = -40 to +85°C	1.47	1.51	1.55	V
			T_A = +85 to +105°C	1.45	1.51	1.57	V
	V_{PDR}	The power supply voltage is falling. Note 1	T_A = -40 to +85°C	1.46	1.50	1.54	V
			T_A = +85 to +105°C	1.44	1.50	1.56	V
Minimum pulse width Note 2	T_{PW1}	Other than STOP/SUB HALT/SUB RUN	T_A = +40 to +105°C	300			μs
	T_{PW2}	STOP/SUB HALT/SUB RUN	T_A = +40 to +105°C	300			μs

Note 1. However, when the operating voltage falls while the LVD is off, enter STOP mode, or enable the reset status using the external reset pin before the voltage falls below the operating voltage range shown in 34.4 AC Characteristics.

Note 2. Minimum time required for a POR reset when V_{DD} exceeds below V_{PDR}. This is also the minimum time required for a POR reset from when V_{DD} exceeds below 0.7 V to when V_{DD} exceeds V_{POR} while STOP mode is entered or the main system clock is stopped through setting bit 0 (HIOSTOP) and bit 7 (MSTOP) in the clock operation status control register (CSC).



34.6.6 LVD circuit characteristics

(1) LVD Detection Voltage of Reset Mode and Interrupt Mode

(T_A = -40 to +85°C, V_{PDR} ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVD2}	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_{LVD3}	The power supply voltage is rising.	2.96	3.02	3.08	V
			The power supply voltage is falling.	2.90	2.96	3.02	V
		V_{LVD4}	The power supply voltage is rising.	2.86	2.92	2.97	V
			The power supply voltage is falling.	2.80	2.86	2.91	V
		V_{LVD5}	The power supply voltage is rising.	2.76	2.81	2.87	V
			The power supply voltage is falling.	2.70	2.75	2.81	V
		V_{LVD6}	The power supply voltage is rising.	2.66	2.71	2.76	V
			The power supply voltage is falling.	2.60	2.65	2.70	V
		V_{LVD7}	The power supply voltage is rising.	2.56	2.61	2.66	V
			The power supply voltage is falling.	2.50	2.55	2.60	V
		V_{LVD8}	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_{LVD9}	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_{LVD10}	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_{LVD11}	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_{LVD12}	The power supply voltage is rising.	1.74	1.77	1.81	V
			The power supply voltage is falling.	1.70	1.73	1.77	V
		V_{LVD13}	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}		300			μs
Detection delay time						300	μs

(T_A = +85 to +105°C, V_{PDR} ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVD2}	The power supply voltage is rising.	3.01	3.13	3.25	V
			The power supply voltage is falling.	2.94	3.06	3.18	V
		V_{LVD3}	The power supply voltage is rising.	2.90	3.02	3.14	V
			The power supply voltage is falling.	2.85	2.96	3.07	V
		V_{LVD4}	The power supply voltage is rising.	2.81	2.92	3.03	V
			The power supply voltage is falling.	2.75	2.86	2.97	V
		V_{LVD5}	The power supply voltage is rising.	2.71	2.81	2.92	V
			The power supply voltage is falling.	2.64	2.75	2.86	V
		V_{LVD6}	The power supply voltage is rising.	2.61	2.71	2.81	V
			The power supply voltage is falling.	2.55	2.65	2.75	V
		V_{LVD7}	The power supply voltage is rising.	2.51	2.61	2.71	V
			The power supply voltage is falling.	2.45	2.55	2.65	V
Minimum pulse width		t_{LW}		300			μs
Detection delay time						300	μs

(2) LVD Detection Voltage of Interrupt & Reset Mode

(T_A = -40 to +85°C, V_{PDR} ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Interrupt and reset mode	V_{LVDA0}	V_{POC0}, V_{POC1}, V_{POC2} = 0, 0, 0, falling reset voltage			1.60	1.63	1.66	V
	V_{LVDA1}		LVIS0, LVIS1 = 1, 0	Rising release reset voltage	1.74	1.77	1.81	V
				Falling interrupt voltage	1.70	1.73	1.77	V
	V_{LVDA2}		LVIS0, LVIS1 = 0, 1	Rising release reset voltage	1.84	1.88	1.91	V
				Falling interrupt voltage	1.80	1.84	1.87	V
	V_{LVDA3}		LVIS0, LVIS1 = 0, 0	Rising release reset voltage	2.86	2.92	2.97	V
				Falling interrupt voltage	2.80	2.86	2.91	V
	V_{LVDB0}	V_{POC0}, V_{POC1}, V_{POC2} = 0, 0, 1, falling reset voltage			1.80	1.84	1.87	V
	V_{LVDB1}		LVIS0, LVIS1 = 1, 0	Rising release reset voltage	1.94	1.98	2.02	V
				Falling interrupt voltage	1.90	1.94	1.98	V
	V_{LVDB2}		LVIS0, LVIS1 = 0, 1	Rising release reset voltage	2.05	2.09	2.13	V
				Falling interrupt voltage	2.00	2.04	2.08	V
	V_{LVDB3}		LVIS0, LVIS1 = 0, 0	Rising release reset voltage	3.07	3.13	3.19	V
				Falling interrupt voltage	3.00	3.06	3.12	V
	V_{LVDC0}	V_{POC0}, V_{POC1}, V_{POC2} = 0, 1, 0, falling reset voltage			2.40	2.45	2.50	V
	V_{LVDC1}		LVIS0, LVIS1 = 1, 0	Rising release reset voltage	2.56	2.61	2.66	V
				Falling interrupt voltage	2.50	2.55	2.60	V
	V_{LVDC2}		LVIS0, LVIS1 = 0, 1	Rising release reset voltage	2.66	2.71	2.76	V
				Falling interrupt voltage	2.60	2.65	2.70	V
	V_{LVDD0}	V_{POC0}, V_{POC1}, V_{POC2} = 0, 1, 1, falling reset voltage			2.70	2.75	2.81	V
	V_{LVDD1}		LVIS0, LVIS1 = 1, 0	Rising release reset voltage	2.86	2.92	2.97	V
				Falling interrupt voltage	2.80	2.86	2.91	V
	V_{LVDD2}		LVIS0, LVIS1 = 0, 1	Rising release reset voltage	2.96	3.02	3.08	V
				Falling interrupt voltage	2.90	2.96	3.02	V

(T_A = +85 to +105°C, V_{PDR} ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Interrupt and reset mode	V_{LVDD0}	V_{POC0}, V_{POC1}, V_{POC2} = 0, 1, 1, falling reset voltage			2.64	2.75	2.86	V
	V_{LVDD1}		LVIS0, LVIS1 = 1, 0	Rising release reset voltage	2.81	2.92	3.03	V
				Falling interrupt voltage	2.75	2.86	2.97	V
	V_{LVDD2}		LVIS0, LVIS1 = 0, 1	Rising release reset voltage	2.90	3.02	3.14	V
				Falling interrupt voltage	2.85	2.96	3.07	V

34.6.7 Power supply voltage rising slope characteristics

(T_A = -40 to +105°C, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Power supply voltage rising slope	S_{VDD}				54	V/ms

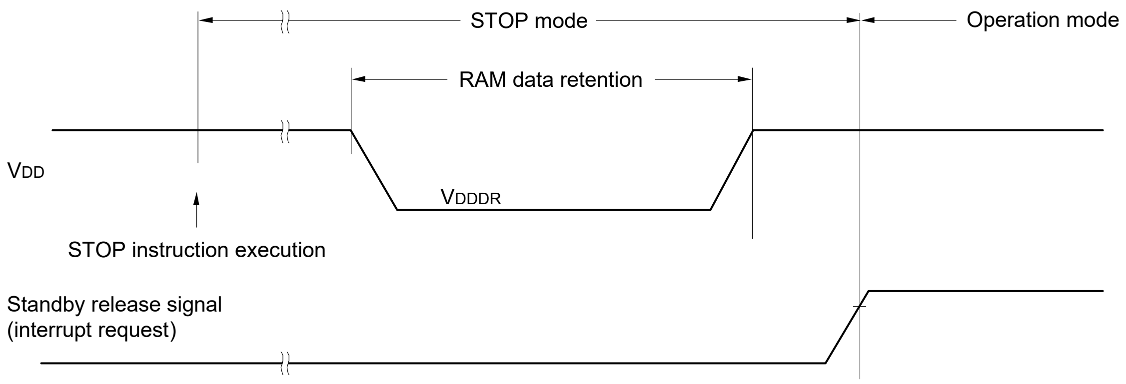
Caution Make sure to keep the internal reset state by the LVD circuit or an external reset until V_{DD} reaches the operating voltage range shown in 34.4 AC Characteristics.

34.7 RAM Data Retention Characteristics

(T_A = -40 to +85°C, 1.8 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)
(T_A = +85 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}	T_A = -40 to +85°C	1.46 Note		3.6	V
		T_A = +85 to +105°C	1.44 Note		3.6	V

Note The value depends on the POR detection voltage. When the voltage drops, the data is retained before a POR reset is effected, but data is not retained when a POR reset is effected.



34.8 Flash Memory Programming Characteristics

(T_A = -40 to +85°C, 1.8 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)
(T_A = +85 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
System clock frequency	f_{CLK}		1		24	MHz
Number of code flash rewrites Notes 1, 2, 3	C_{erwr}	Retained for 20 years T_A = 85°C Note 4	1,000			Times
Number of data flash rewrites Notes 1, 2, 3		Retained for 1 year T_A = 25°C Note 4		1,000,000		
		Retained for 5 years T_A = 85°C Note 4	100,000			
		Retained for 20 years T_A = 85°C Note 4	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. 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.
Note 4. This temperature is the average value at which data are retained.

34.9 Dedicated Flash Memory Programmer Communication (UART)

(T_A = -40 to +85°C, 1.8 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)
(T_A = +85 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

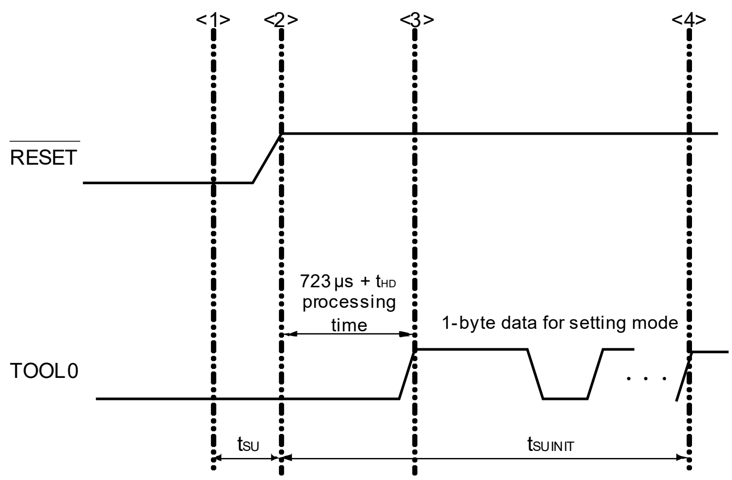
Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		During serial programming	115,200		1,000,000	bps

34.10 Timing of Entry to Flash Memory Programming Modes

(T_A = -40 to +85°C, 1.8 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)
(T_A = +85 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 3.6 V, V_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
How long from when an external reset ends until the initial communication settings are specified Note 1	t_{SUINIT}	POR and LVD reset must end before the external reset ends.			100	ms
How long from when the TOOL0 pin is placed at the low level until an external reset ends Note 1	t_{SU}	POR and LVD reset must end before the external reset ends.	10			μs
How long the TOOL0 pin must be kept at the low level after an external reset ends (excluding the processing time of the firmware to control the flash memory) Notes 1, 2	t_{HD}	POR and LVD reset must end before the external reset ends.	1			ms

Note 1. Deassertion of the POR and LVD reset signals must precede deassertion of the pin reset signal.
Note 2. This excludes the flash firmware processing time (723 μs).



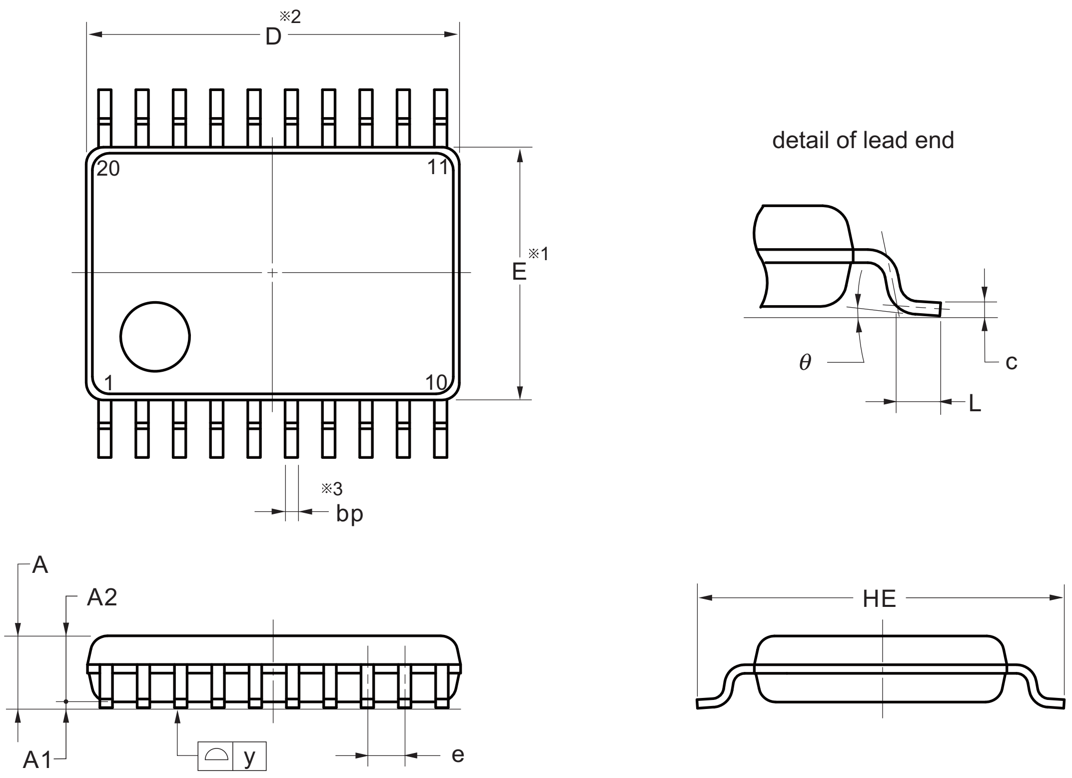
<1> The low level is input to the TOOL0 pin.
<2> The external reset ends (POR and LVD reset must end before the external reset ends).
<3> The TOOL0 pin is set to the high level.
<4> Setting of the flash memory programming mode by UART reception and complete the baud rate setting.

Remark t_{SUINIT}: The segment shows that it is necessary to finish specifying the initial communication settings within 100 ms from when the external resets end.
t_{SU}: How long from when the TOOL0 pin is placed at the low level until a pin reset ends
t_{HD}: How long to keep the TOOL0 pin at the low level from when the external resets end (excluding the processing time of the firmware to control the flash memory)

CHAPTER 35 PACKAGE DRAWINGS

35.1 20-pin package

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LSSOP20-4.4x6.5-0.65	PLSP0020JB-A	P20MA-65-NAA-1	0.1



(UNIT:mm)

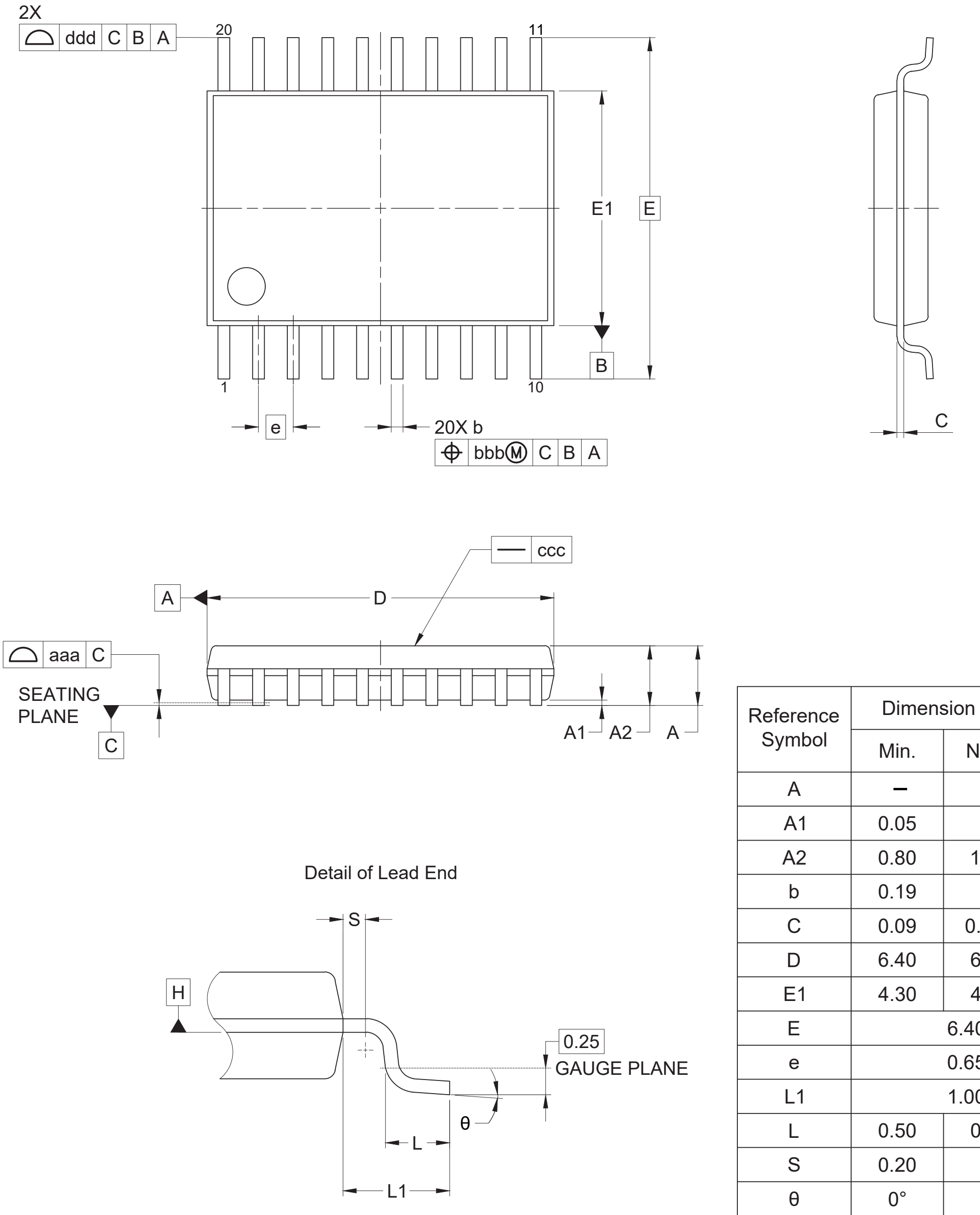
ITEM	DIMENSIONS
D	6.50±0.10
E	4.40±0.10
HE	6.40±0.20
A	1.45 MAX.
A1	0.10±0.10
A2	1.15
e	0.65±0.12
bp	$0.22^{+0.10}_{-0.05}$
c	$0.15^{+0.05}_{-0.02}$
L	0.50±0.20
y	0.10
θ	0° to 10°

NOTE

1.Dimensions "※1" and "※2" do not include mold flash.

2.Dimension "※3" does not include trim offset.

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-TSSOP20-4.40x6.50-0.65	PTSP0020JI-A	0.08

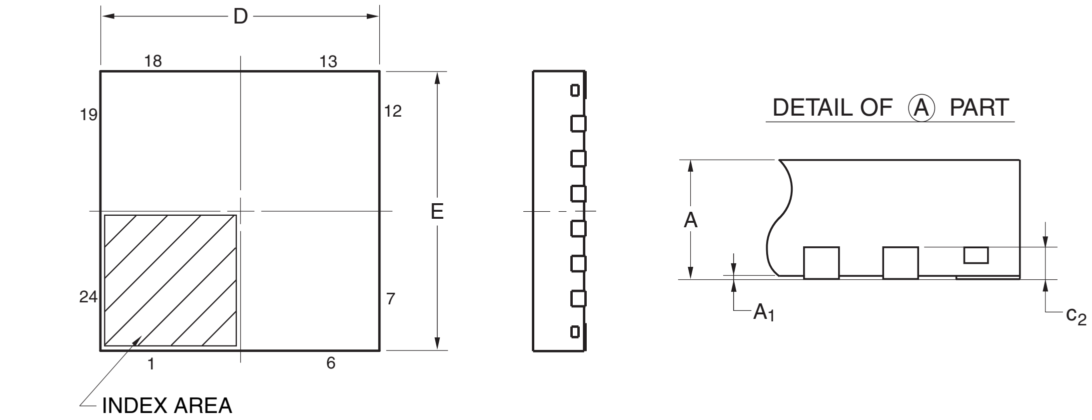


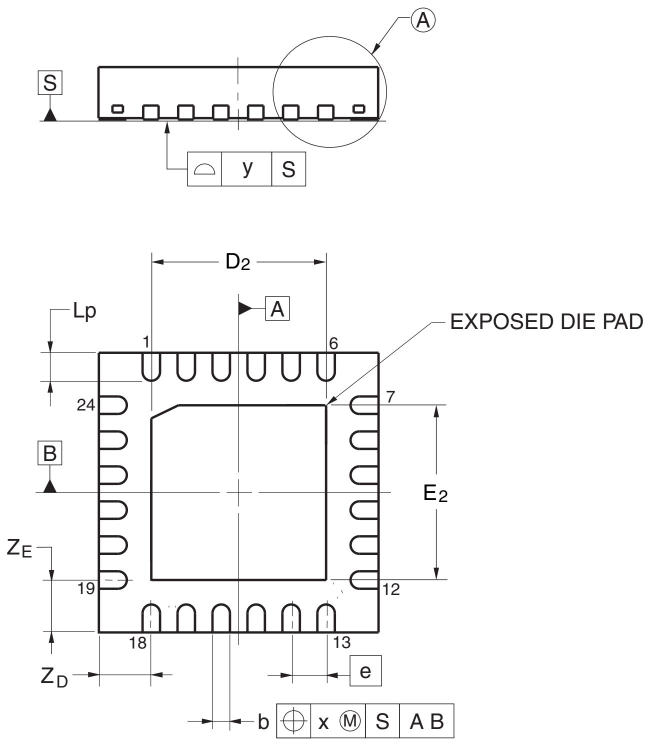
Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	–	–	1.20
A1	0.05	–	0.15
A2	0.80	1.00	1.05
b	0.19	–	0.30
C	0.09	0.127	0.20
D	6.40	6.50	6.60
E1	4.30	4.40	4.50
E	6.40 BSC		
e	0.65 BSC		
L1	1.00 REF		
L	0.50	0.60	0.75
S	0.20	–	–
θ	0°	–	8°
aaa	0.10		
bbb	0.10		
ccc	0.05		
ddd	0.20		

NOTES:
1.DIMENSION 'D' AND 'E1' DOES NOT INCLUDE MOLD FLASH.
2.DIMENSION 'b' DOES NOT INCLUDE TRIM OFFSET.
3.DIMENSION 'D' AND 'E1' TO BE DETERMINED AT DATUM PLANE H.

35.2 24-pin package

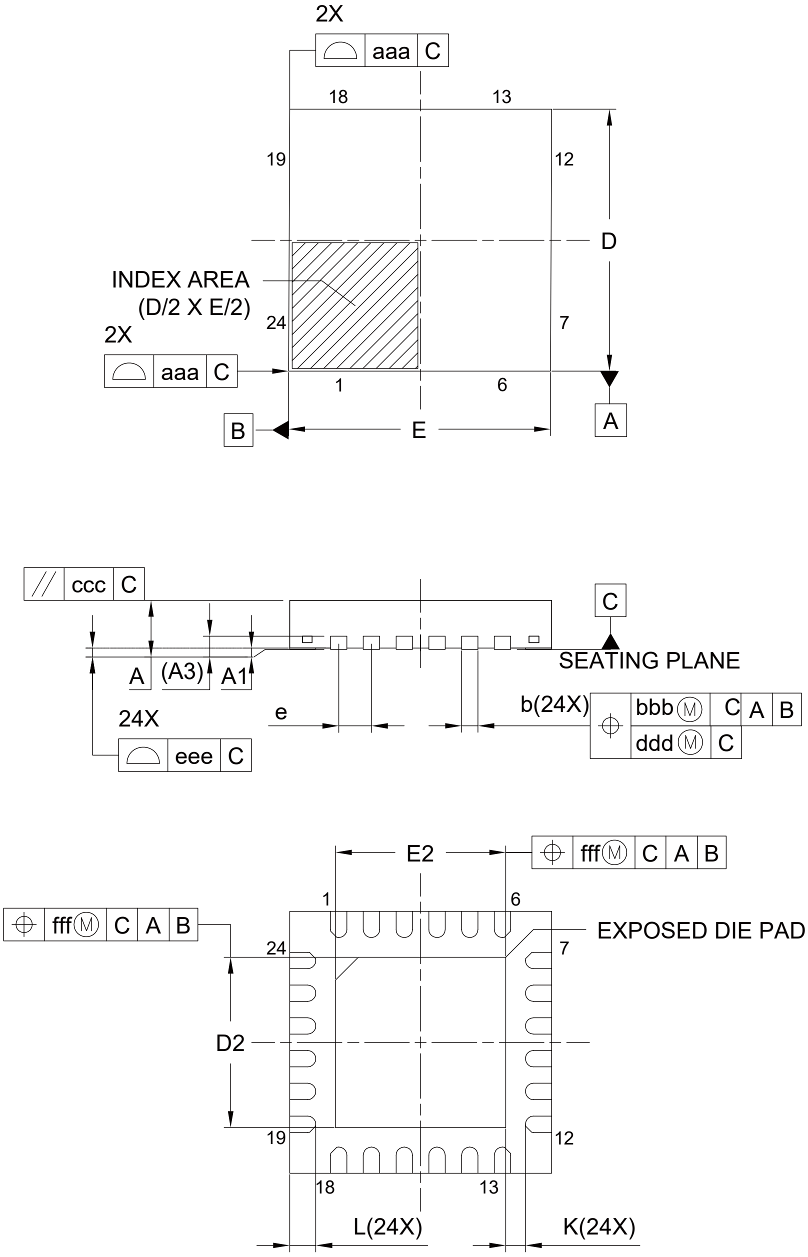
JEITA Package code	RENESAS code	Previous code	MASS(TYP.)[g]
P-HWQFN24-4x4-0.50	PWQN0024KE-A	P24K8-50-CAB-3	0.04





Referance Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	3.95	4.00	4.05
E	3.95	4.00	4.05
A	—	—	0.80
A_1	0.00	—	—
b	0.18	0.25	0.30
e	—	0.50	—
Lp	0.30	0.40	0.50
x	—	—	0.05
y	—	—	0.05
Z_D	—	0.75	—
Z_E	—	0.75	—
c_2	0.15	0.20	0.25
D_2	—	2.50	—
E_2	—	2.50	—

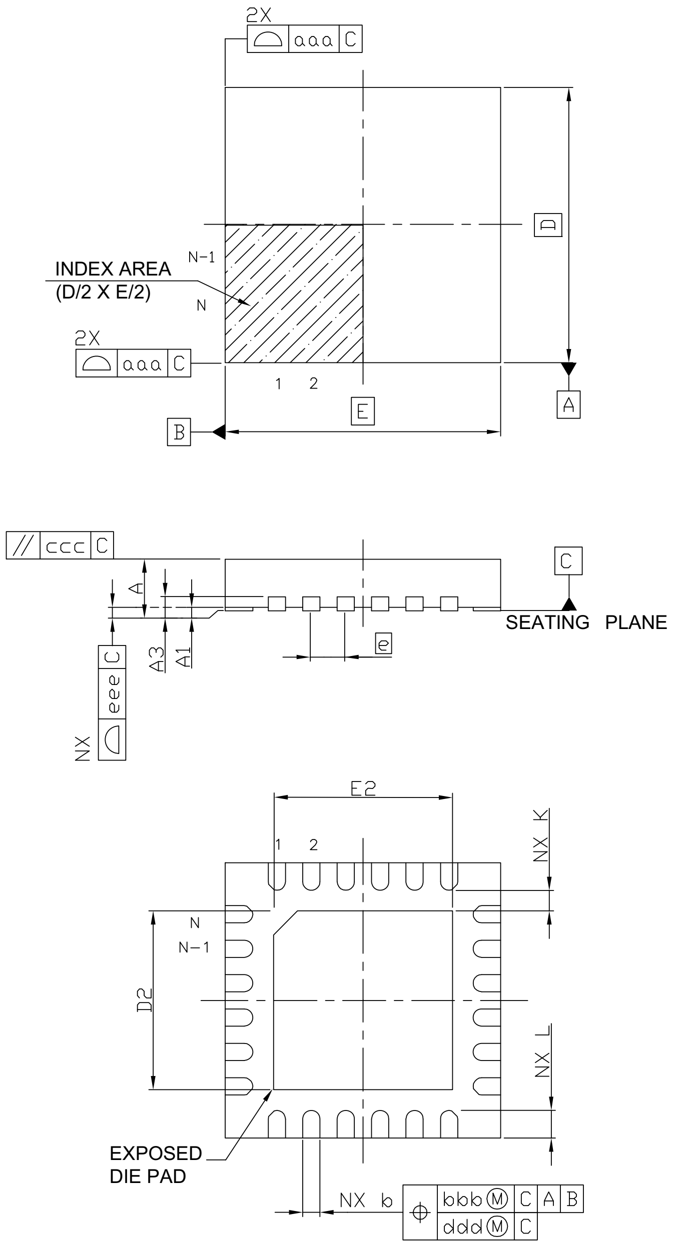
JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN024-4x4-0.50	PWQN0024KF-A	0.04



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	–	0.80	
A1	0.00	0.02	0.05
A3	0.203 REF.		
b	0.18	0.25	0.30
D	4.00 BSC		
E	4.00 BSC		
e	0.50 BSC		
L	0.35	0.40	0.45
K	0.20	–	–
D2	2.55	2.60	2.65
E2	2.55	2.60	2.65
aaa	0.15		
bbb	0.10		
ccc	0.10		
ddd	0.05		
eee	0.08		
fff	0.10		

<R>

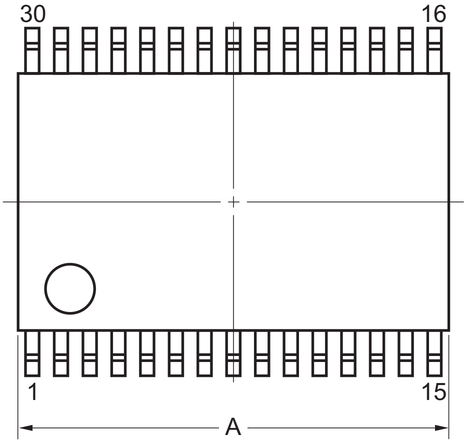
JEITA Package Code	RENESAS Code	MASS(Typ.)[g]
P-HWQFN24-4×4-0.50	PWQN0024KH-A	0.04

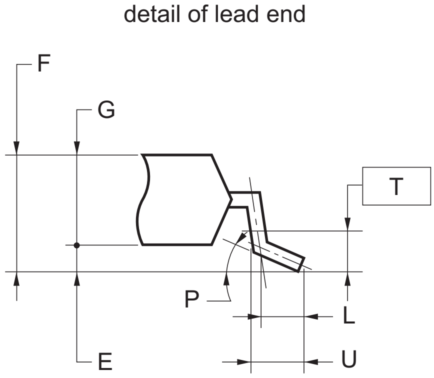


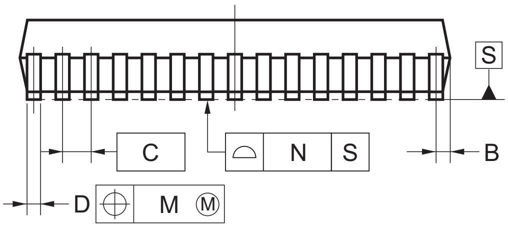
Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A_1	0.00	—	0.05
A_3	0.20 REF.		
b	0.20	0.25	0.30
$\boxed{D}$	—	4.00	—
$\boxed{E}$	—	4.00	—
$\boxed{e}$	—	0.50	—
N	24		
L	0.30	0.40	0.50
K	0.20	—	—
D_2	2.50	2.60	2.70
E_2	2.50	2.60	2.70
aaa	—	—	0.15
bbb	—	—	0.10
ccc	—	—	0.10
ddd	—	—	0.05
eee	—	—	0.08

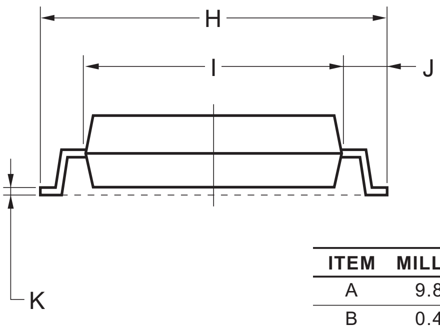
35.3 30-pin package

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LSSOP30-0300-0.65	PLSP0030JB-B	S30MC-65-5A4-3	0.18









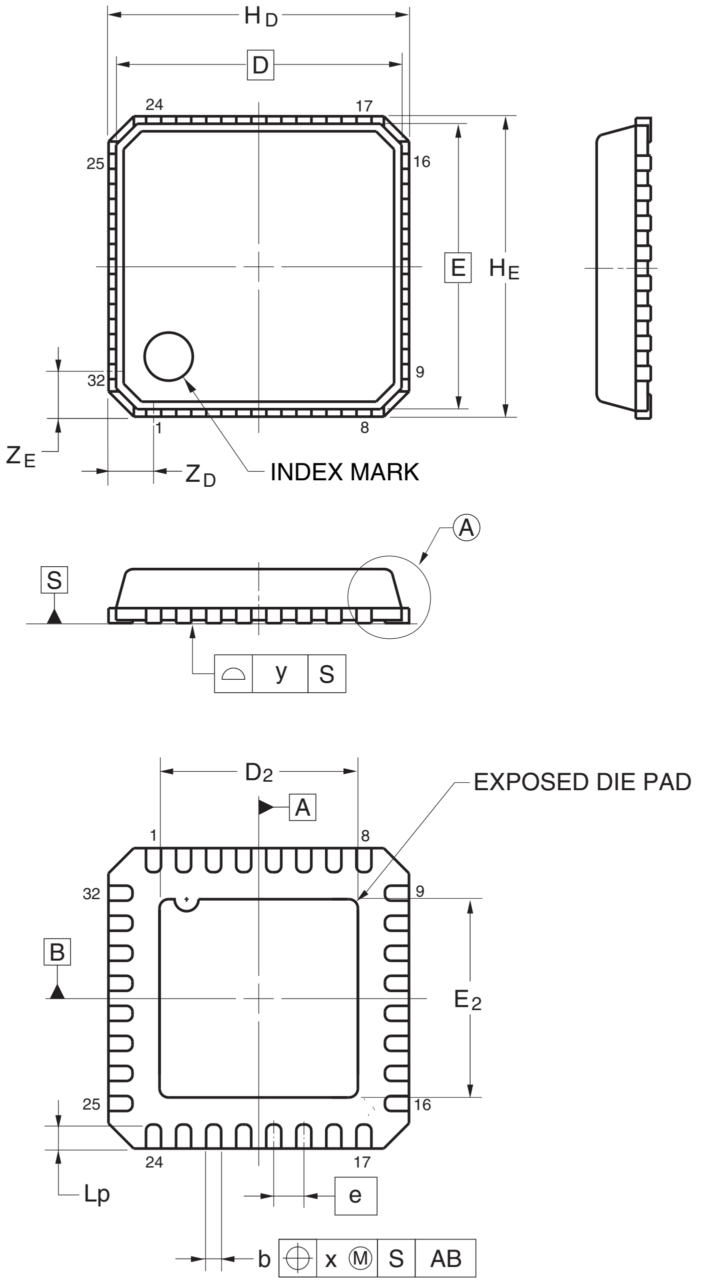
ITEM	MILLIMETERS
A	9.85±0.15
B	0.45 MAX.
C	0.65 (T.P.)
D	$0.24^{+0.08}_{-0.07}$
E	0.1±0.05
F	1.3±0.1
G	1.2
H	8.1±0.2
I	6.1±0.2
J	1.0±0.2
K	0.17±0.03
L	0.5
M	0.13
N	0.10
P	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
T	0.25
U	0.6±0.15

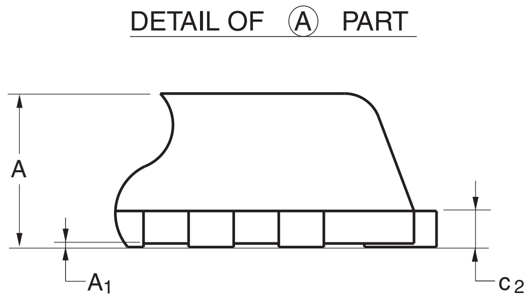
NOTE

Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

35.4 32-pin package

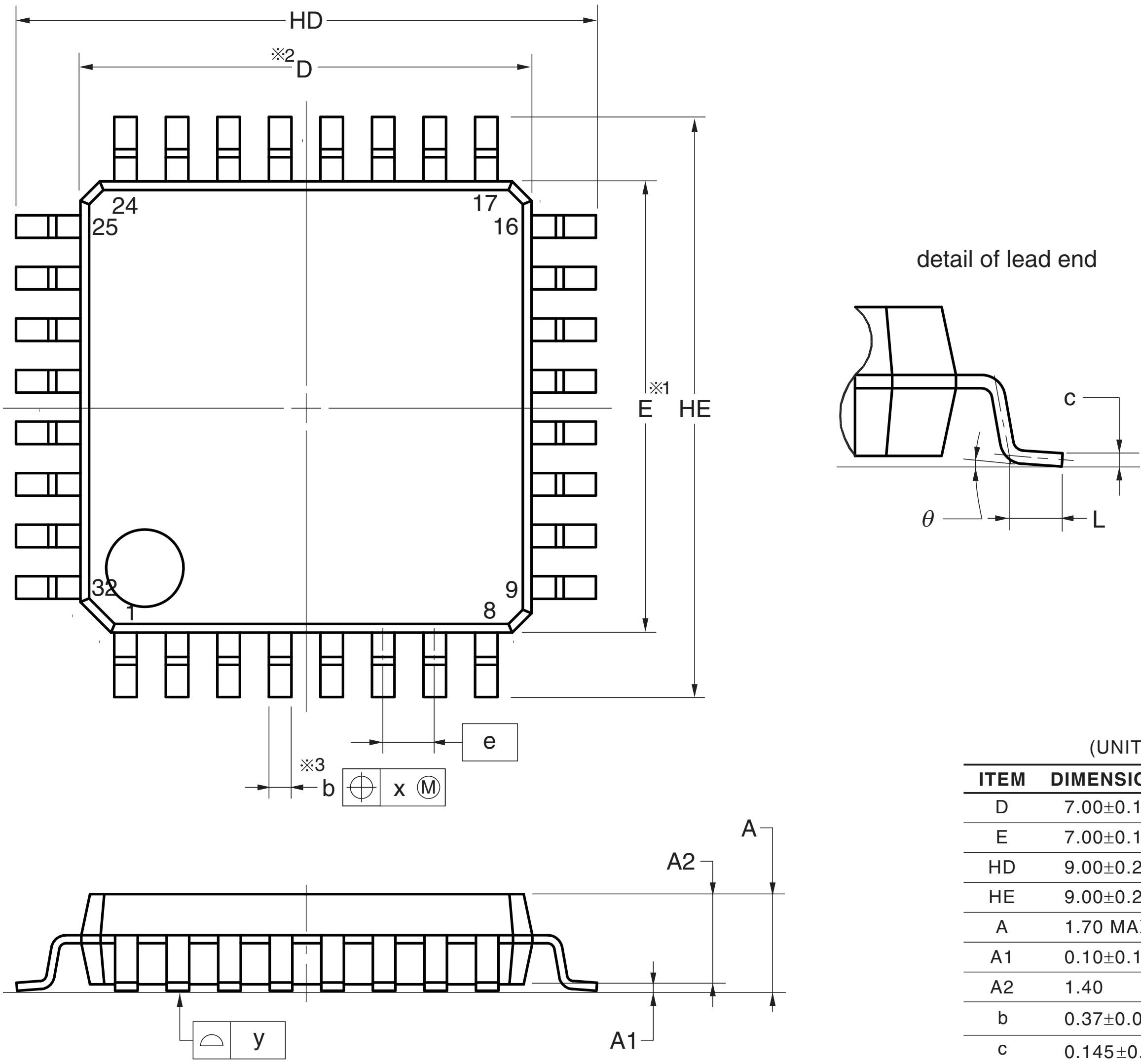
JEITA Package code	RENESAS code	Previous code	MASS(TYP.)[g]
P-HVQFN32-5x5-0.50	PVQN0032KE-A	P32K9-50B-BAH	0.058





Referance Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	4.75	—
E	—	4.75	—
A	—	—	0.90
A_1	0.00	—	—
b	0.20	0.25	0.30
e	—	0.50	—
Lp	0.30	0.40	0.50
x	—	—	0.10
y	—	—	0.05
H_D	4.95	5.00	5.05
H_E	4.95	5.00	5.05
Z_D	—	0.75	—
Z_E	—	0.75	—
c_2	0.19	0.20	0.21
D_2	—	3.30	—
E_2	—	3.30	—

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LQFP32-7x7-0.80	PLQP0032GB-A	P32GA-80-GBT-1	0.2



(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

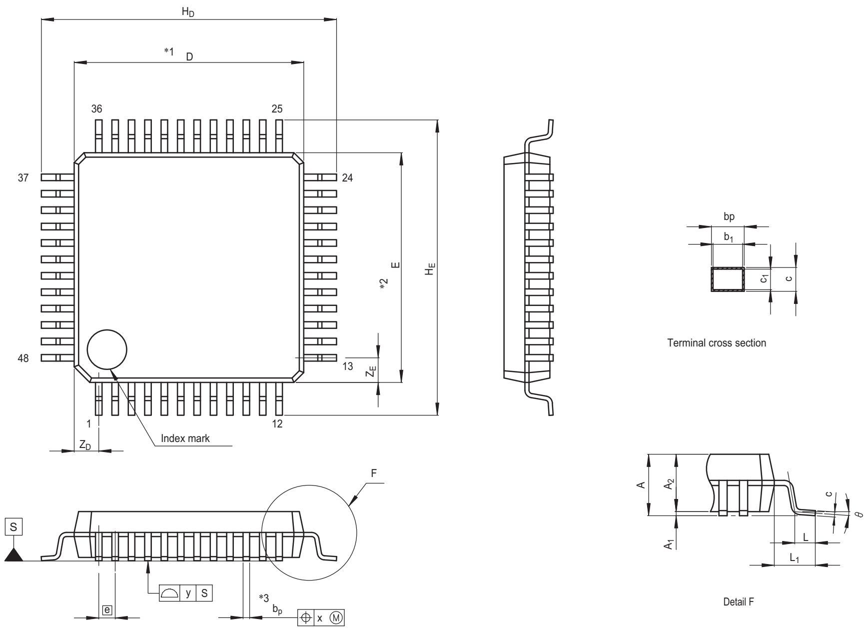
NOTE

1.Dimensions “※1” and “※2” do not include mold flash.

2.Dimension “※3” does not include trim offset.

35.5 48-pin package

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP48-7x7-0.50	PLQP0048KB-A	48P6Q-A	0.2g



NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	6.9	7.0	7.1
E	6.9	7.0	7.1
A_2	—	1.4	—
H_D	8.8	9.0	9.2
H_E	8.8	9.0	9.2
A	—	—	1.7
A_1	0	0.1	0.2
b_p	0.17	0.22	0.27
b_1	—	0.20	—
c	0.09	0.145	0.20
c_1	—	0.125	—
θ	0°	—	8°
e	—	0.5	—
x	—	—	0.08
y	—	—	0.10
Z_D	—	0.75	—
Z_E	—	0.75	—
L	0.35	0.5	0.65
L_1	—	1.0	—

APPENDIX A REVISION HISTORY

A.1 Major Revisions in This Edition

(1/1)

Page	Description	Classification
ALL		
—	“3-wire serial I/O” and “3-wire serial” were modified to “simplified SPI”.	(c)
—	“CSI” was modified to “simplified SPI (CSI)”.	(c)
CHAPTER 1 OUTLINE		
p.3	1.1 Features: Note was added.	(c)
p.5	1.2 Ordering Information: 24-pin RENESAS code was added.	(d)
CHAPTER 4 PORT FUNCTIONS		
p.102	4.4.4 Handling different potential (1.8 V, 2.5 V) by using I/O buffers: Note was added.	(c)
CHAPTER 8 REAL-TIME CLOCK 2		
p.297	Figure 8 - 8 Format of Real-time clock control register 1 (RTCC1) (3/3): The description of the RWAIT bit was modified.	(c)
p.313	Figure 8 - 22 Procedure for Reading Real-time Clock 2: Caution was modified.	(c)
p.315	Figure 8 - 23 Procedure for Writing to Real-time Clock 2: Cautions 1 and 2 were modified.	(c)
CHAPTER 17 SERIAL ARRAY UNIT		
p.463	CHAPTER 17 SERIAL ARRAY UNIT: Note was added.	(c)
p.466	17.1.3 Simplified I^2C (IIC00, IIC01): “Wait detection functions” was modified to “clock stretch detection”.	(a)
p.618	17.8 Operation of Simplified I^2C (IIC00, IIC01) Communication: “Wait detection functions” was modified to “clock stretch detection”.	(a)
CHAPTER 34 ELECTRICAL SPECIFICATIONS		
p.869	34.3.2 Supply current characteristics: Notes 1, 4, and 5 were modified.	(c)
p.871	34.3.2 Supply current characteristics: Notes 1 and 5 were modified.	(c)
p.872	34.3.2 Supply current characteristics: Note 1 was modified and note 2 was deleted.	(c)
CHAPTER 35 PACKAGE DRAWINGS		
p.942	35.2 24-pin package: RENESAS code PWQN0024KH-A 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. Chapter indicates the chapter of each edition.

(1/7)

Edition	Description	Chapter
Rev.2.30	Addition of PG-FP6, addition of E2, E2 Lite, E20	ALL
	Change of description in 1.1 Features	CHAPTER 1 OUTLINE
	Change of Figure 1 - 1 Part Number, Memory Size, and Package of RL78/I1D and addition of note 1	
	Change of table in 1.2 Ordering Information	
	Change of description in 1.3.1 20-pin products	
	Addition of caution in Figure 2 - 9 Pin Block Diagram of Pin Type 7-1-4	CHAPTER 2 PIN FUNCTIONS
	Addition of caution in Figure 2 - 12 Pin Block Diagram of Pin Type 8-1-3	
	Addition of cautions 1 and 2 in Figure 2 - 13 Pin Block Diagram of Pin Type 8-1-4	
	Change of the figure and addition of cautions 1 and 2 in Figure 2 - 14 Pin Block Diagram of Pin Type 8-6-4	
	Change of the figure title and addition of caution in Figure 2 - 15 Pin Block Diagram of Pin Type 12-1-1	
	Change of Table 3 - 10 Extended Special Function Register (2nd SFR) List	CHAPTER 3 CPU ARCHITECTURE
	Change of Figure 3 - 36 Example of CALL, CALLT	
	Change of Table 6 - 5 CPU Clock Transition and SFR Register Setting Examples (5)	CHAPTER 6 CLOCK GENERATOR
	Change of Figure 7 - 20 Format of Timer channel stop register m (TTm)	CHAPTER 7 TIMER ARRAY UNIT
	Change of description in 8.3.3 Subsystem clock supply mode control register (OSMC)	CHAPTER 8 REAL-TIME CLOCK 2
	Change of description in 13.1 Functions of Watchdog Timer	CHAPTER 13 WATCHDOG TIMER
	Change of Table 13 - 3 Setting of Overflow Time of Watchdog Timer	
	Change of description in 13.4.4 Setting watchdog timer interval interrupt	
	Change of Table 13 - 5 Setting of Watchdog Timer Interval Interrupt	
	Change of note 1 in Figure 14 - 5 Timing Chart When A/D Voltage Comparator Is Used	CHAPTER 14 A/D CONVERTER
	Change of Table 14 - 4 A/D Conversion Time Selection (2/4)	
	Change of Figure 14 - 34 Setup when Temperature Sensor Output Voltage/Internal Reference Voltage is Selected	
	Change of Figure 17 - 11 Format of Serial flag clear trigger register mn (SIRmn)	CHAPTER 17 SERIAL ARRAY UNIT
	Change of caution 1 in Figure 17 - 13 Format of Serial status register mn (SSRmn)	
	Change of Figure 17 - 21 Format of Serial standby control register m (SSCm)	
	Change of Figure 17 - 34 Flowchart of Master Transmission (in Continuous Transmission Mode)	
	Change of Figure 17 - 114 Flowchart of UART Transmission (in Continuous Transmission Mode)	
	Change of Figure 23 - 8 When the Interrupt Request Signal is not Generated in the SNOOZE Mode	CHAPTER 23 STANDBY FUNCTION
	Change of Figure 29 - 1 Format of User Option Byte (000C0H/010C0H)	CHAPTER 29 OPTION BYTE

(2/7)

Edition	Description	Chapter
Rev.2.30	Change of Table 30 - 1 Wiring Between RL78/I1D and Dedicated Flash Memory Programmer	CHAPTER 30 FLASH MEMORY
	Addition of note 1 in Table 30 - 1 Wiring Between RL78/I1D and Dedicated Flash Memory Programmer	
	Addition of note 1 in Figure 30 - 2 Communication with Dedicated Flash Memory Programmer	
	Addition of note 2 in Table 30 - 2 Pin Connection	
	Change of Figure 30 - 4 Communication with External Device	
	Addition of note in 30.3.6 Power supply	
	Change of Figure 30 - 7 Setting of Flash Memory Programming Mode	
	Change of the title and description in 30.5 Processing Time for Each Command When Dedicated Flash Memory Programmer Is in Use (Reference Value)	
	Change of table title and header in Table 30 - 10 Processing Time for Each Command When PG-FP5 or PG-FP6 Is in Use (Reference Value)	
	Addition of caution 4 in 30.8.3 Procedure for accessing data flash memory	
	Change of the figure in 34.10 Timing of Entry to Flash Memory Programming Modes	CHAPTER 34 ELECTRICAL SPECIFICATIONS
	Addition of package drawing in 35.1 20-pin package	CHAPTER 35 PACKAGE DRAWINGS
	Addition of package drawing in 35.2 24-pin package	
Rev.2.20	Addition of product name in 1.3.1 20-pin products	CHAPTER 1 OUTLINE
	Addition of product name in 1.3.2 24-pin products	
	Addition of product name in 1.3.3 30-pin products	
	Addition of product name in 1.3.4 32-pin products	
	Change of description and addition of product name in 1.3.4 32-pin products	
	Addition of product name in 1.3.5 48-pin products	
	Change of description in 1.6 Outline of Functions	
	Change of the header of the table in 2.1.1 20-pin Products	CHAPTER 2 PIN FUNCTIONS
	Change of the header of the table in 2.1.2 24-pin Products	
	Change of the header of the table in 2.1.3 30-pin Products	
	Change of the header of the table in 2.1.4 32-pin Products	
	Change of the header of the table in 2.1.5 48-pin Products	
	Change of note 1 in Figure 3 - 2 Memory Map (R5F117xA (x = 6, 7, A, B, G))	CHAPTER 3 CPU ARCHITECTURE
	Change of the value after reset in Table 3 - 9 Extended Special Function Register (2nd SFR) List (2/5)	
	Change of description in 3.3.4 Register indirect addressing	
	Change of description in Figure 4 - 4 Format of Port input mode register	CHAPTER 4 PORT FUNCTIONS
	Change of description in 5.2.1 Flash operating mode select register (FLMODE)	CHAPTER 5 OPERATION STATE CONTROL

(3/7)

Edition	Description	Chapter
Rev.2.20	Change of Figure 6 - 11 Format of Subsystem clock supply mode control register (OSMC)	CHAPTER 6 CLOCK GENERATOR
	Change of note 1, and addition of remark in Figure 6 - 11 Format of Subsystem clock supply mode control register (OSMC)	
	Change of description in 6.4.5 Low-speed on-chip oscillator	
	Addition of caution in 6.6.2 Example of setting X1 oscillation clock	
	Change of description and addition of remark in 6.6.3 Example of setting XT1 oscillation clock	
	Change of description in Table 6 - 7 Changing CPU Clock (1/5)	
	Change of description in Table 6 - 8 Changing CPU Clock (2/5)	
	Change of description in Table 6 - 9 Changing CPU Clock (3/5)	
	Change of description in Table 6 - 10 Changing CPU Clock (4/5)	
	Change of description in Table 6 - 11 Changing CPU Clock (5/5)	
	Change of description in Table 6 - 12 Maximum Time Required for Main System Clock Switchover	
	Change of description in Figure 7 - 13 Format of Timer mode register mn (TMRmn) (1/4)	CHAPTER 7 TIMER ARRAY UNIT
	Change of caution 1 in Figure 7 - 21 Format of Timer input select register 0 (TIS0)	
	Deletion of caution in Figure 7 - 41 TO0n Pin Statuses by Collective Manipulation of TO0n Bit	
	Change of Figure 8 - 4 Format of Subsystem clock supply mode control register (OSMC)	CHAPTER 8 REAL-TIME CLOCK 2
	Change of note 1, and addition of remark in Figure 8 - 4 Format of Subsystem clock supply mode control register (OSMC)	
	Deletion of PER0 in Table 9 - 1 Configuration of Frequency Measurement Circuit	CHAPTER 9 FREQUENCY MEASURE CIRCUIT
	Deletion of PER0 in 9.3 Registers Controlling Frequency Measurement Circuit	
	Change of Figure 9 - 3 Format of Subsystem clock supply mode control register (OSMC)	
	Change of note 1, and addition of remark in Figure 9 - 3 Format of Subsystem clock supply mode control register (OSMC)	
	Change of Figure 9 - 9 Procedure for Setting Frequency Measurement Circuit Using Reference Clock	
	Change of Figure 10 - 4 Format of Subsystem clock supply mode control register (OSMC)	CHAPTER 10 12-BIT INTERVAL TIMER
	Change of note 1, and addition of remark in Figure 10 - 4 Format of Subsystem clock supply mode control register (OSMC)	
	Change of description in 11.4.2 Timer Operation	CHAPTER 11 8-BIT INTERVAL TIMER
	Addition of 11.4.5 Procedure for Setting the 8-bit Interval Timer	
	Change of description in 12.3.2 Registers controlling port functions of pins to be used for clock or buzzer output	CHAPTER 12 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER
	Change of description in 12.5 Cautions of clock output/buzzer output controller	

(4/7)

Edition	Description	Chapter
Rev.2.20	Addition of note in Table 13 - 4 Setting Window Open Period of Watchdog Timer	CHAPTER 13 WATCHDOG TIMER
	Change of Figure 14 - 31 Setting up Software Trigger Mode	CHAPTER 14 A/D CONVERTER
	Change of Figure 14 - 32 Setting up Hardware Trigger No-Wait Mode	
	Change of Figure 14 - 33 Setting up Hardware Trigger Wait Mode	
	Change of Figure 14 - 34 Setup when Temperature Sensor Output Voltage/Internal Reference Voltage is Selected	
	Change of Figure 14 - 39 Flowchart for Setting up SNOOZE Mode	
	Change of description in 14.10 Cautions for A/D Converter	
	Change of Figure 15 - 1 Comparator Block Diagram	CHAPTER 15 COMPARATOR
	Deletion of note 3 in Table 15 - 2 Procedure for Setting Comparator Associated Registers	
	Change of caution 1 in Figure 16 - 2 Format of Operational amplifier mode control register (AMPMC)	CHAPTER 16 OPERATIONAL AMPLIFIER
	Change of Figure 16 - 7 Operational Amplifier State Transitions	
	Deletion of note 3 and remark 2 in Figure 16 - 7 Operational Amplifier State Transitions	
	Addition of caution in Figure 16 - 7 Operational Amplifier State Transitions	
	Change of the flowchart and notes 1 and 2 in 16.4.3 Software trigger mode	
	Change of the flowchart and notes 1 and 3 in 16.4.4 ELC trigger mode	
	Change of the flowchart and notes 1 and 3 in 16.4.5 ELC and A/D Trigger Mode	
	Change of Figure 17 - 1 Block Diagram of Serial Array Unit 0	CHAPTER 17 SERIAL ARRAY UNIT
	Change of note 2 in 17.2.2 Lower 8/9 bits of the serial data register mn (SDRmn)	
	Change of caution 3 in Figure 17 - 10 Format of Serial data register mn (SDRmn)	
	Change of Figure 17 - 20 Examples of Reverse Transmit Data	
	Change of description in 17.5.7 SNOOZE mode function	
	Change of note in Figure 17 - 73 Timing Chart of SNOOZE Mode Operation (once startup) (Type 1: DAPmn = 0, CKPmn = 0)	
	Change of Figure 17 - 74 Flowchart of SNOOZE Mode Operation (once startup)	
	Change of note in Figure 17 - 75 Timing Chart of SNOOZE Mode Operation (continuous startup) (Type 1: DAPmn = 0, CKPmn = 0)	
	Change of Figure 17 - 76 Flowchart of SNOOZE Mode Operation (continuous startup)	
	Change of Figure 17 - 112 Flowchart of UART Transmission (in Single-Transmission Mode)	
	Change of note 2 in Figure 17 - 115 Example of Contents of Registers for UART Reception of UART (UART0) (1/2)	
	Change of description in 17.7.3 SNOOZE mode function	
	Change of Figure 17 - 124 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)	
	Change of Figure 17 - 126 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)	
	Change of the specification of “Activation sources” in Table 19 - 1 DTC Specifications	CHAPTER 19 DATA TRANSFER CONTROLLER (DTC)
	Change of description in 19.5.7 DTC Activation Sources	
	Change of description in 20.3.1 Event output destination select register n (ELSELRn) (n = 00 to 19)	CHAPTER 20 EVENT LINK CONTROLLER (ELC)
	Deletion of note 3 in Table 21 – 1 Interrupt Source List (2/2)	CHAPTER 21 INTERRUPT FUNCTIONS

(5/7)

Edition	Description	Chapter
Rev.2.20	Change of note 2 in Figure 23 - 4 STOP Mode Release by Interrupt Request Generation (1/2)	CHAPTER 23 STANDBY FUNCTION
	Change of description in 23.3.3 SNOOZE mode	
	Addition of note 4 in Figure 23 - 8 When the Interrupt Request Signal is not Generated in the SNOOZE Mode	
	Deletion of caution in 24.1 Timing of Reset Operation	CHAPTER 24 RESET FUNCTION
	Change of note 3 in Figure 25 - 3 Timing of Generation of Internal Reset Signal by Power-on-reset Circuit and Voltage Detector (2/3)	CHAPTER 25 POWER-ON-RESET CIRCUIT
	Change of description in 26.1 Functions of Voltage Detector	CHAPTER 26 VOLTAGE DETECTOR
	Change of description in 26.4.2 When used as interrupt mode	
	Change of description in 29.1.1 User option byte (000C0H to 000C2H/010C0H to 010C2H)	CHAPTER 29 OPTION BYTE
	Addition of note 3 in Figure 29 - 1 Format of User Option Byte (000C0H/010C0H)	
	Change of Figure 29 - 6 Format of Option Byte (000C2H/010C2H)	
	Change of Figure 30 - 4 Communication with External Device	CHAPTER 30 FLASH MEMORY
	Change of conditions in 34.3.2 Supply current characteristics	CHAPTER 34 ELECTRICAL SPECIFICATIONS
	Change of note 1 in 34.3.2 Supply current characteristics	
	Change of conditions and unit in 34.3.2 Supply current characteristics	
	Addition of note 5 in 34.3.2 Supply current characteristics	
	Change of table in 34.8 Flash Memory Programming Characteristics	
	Addition of note 4 in 34.8 Flash Memory Programming Characteristics	
	Change of package drawing in 35.5 48-pin products	CHAPTER 35 PACKAGE DRAWINGS
Rev.2.10	The function name changed from real-time clock to real-time clock 2	ALL
	Change of 1.6 Outline of Functions	CHAPTER 1 OUTLINE
	Addition of description in CPU ARCHITECTURE	CHAPTER 3 CPU ARCHITECTURE
	Change of Figure 3 - 2 Memory Map (R5F117xA (x = 6, 7, A, B, G))	
	Change of Table 3 - 9 Extended Special Function Register (2nd SFR) List (2/5)	
	Change of description in 6.1 Functions of Clock Generator	CHAPTER 6 CLOCK GENERATOR
	Change of Figure 6 - 3 Format of System clock control register (CKC)	
	Change of note 2 in Figure 6 - 3 Format of System clock control register (CKC)	
	Correction of error of caution 4 in Figure 6 - 4 Format of Clock operation status control register (CSC)	
	Change of caution 6 in Figure 6 - 4 Format of Clock operation status control register (CSC)	
	Change of note 4 in Figure 6 - 11 Format of Subsystem clock supply mode control register (OSMC) Change of Table 6 - 14 Maximum Number of Clocks Required for fIH « fIM	
	Change of description in 6.6.7 Conditions before clock oscillation is stopped	
	Change of Table 6 - 16 Conditions Before the Clock Oscillation Is Stopped and Flag Settings	
	Change of description in 6.7 Resonator and Oscillator Constants	
	Change of caution in 7.9.1 Operation as one-shot pulse output function	CHAPTER 7 TIMER ARRAY UNIT

(6/7)

Edition	Description	Chapter
Rev.2.10	Change of note 4 in Figure 8 - 4 Format of Subsystem clock supply mode control register (OSMC)	CHAPTER 8 REAL-TIME CLOCK 2
	Addition of note 1 and 2 of Figure 8 - 8 Format of Real-time clock control register 1 (RTCC1) (3/3)	
	Change of 8.4 Real-time Clock 2 Operation	
	Change of note 4 in Figure 9 - 3 Format of Subsystem clock supply mode control register (OSMC)	CHAPTER 9 FREQUENCY MEASURE CIRCUIT
	Change of note 4 in Figure 10 - 4 Format of Subsystem clock supply mode control register (OSMC)	CHAPTER 10 12-BIT INTERVAL TIMER
	Change of Figure 14 - 5 Timing Chart When A/D Voltage Comparator Is Used	CHAPTER 14 A/D CONVERTER
	Change of Figure 14 - 20 Example of Software Trigger Mode (Select Mode, One-Shot Conversion Mode) Operation Timing	
	Change of Figure 14 - 27 Example of Hardware Trigger Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing	
	Change of Figure 14 - 28 Example of Hardware Trigger Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing	
	Change of Figure 14 - 29 Example of Hardware Trigger Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing	
	Change of Figure 14 - 30 Example of Hardware Trigger Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing	
	Change of note 1 in Figure 15 - 5 Format of Comparator filter control register (COMPFIR)	CHAPTER 15 COMPARATOR
	Change of note 2 in Figure 15 - 5 Format of Comparator filter control register (COMPFIR)	
	Change of note 2 and 3 of Figure 15 - 6 Format of Comparator output control register (COMPOCR)	
	Change of Figure 17 - 8 Format of Serial communication operation setting register mn (SCRmn) (1/2)	CHAPTER 17 SERIAL ARRAY UNIT
	Change of Figure 17 - 9 Format of Serial communication operation setting register mn (SCRmn) (2/2)	
	Change of description in 19.3.2 Control Data Allocation	CHAPTER 19 DATA TRANSFER CONTROLLER (DTC)
	Change of Table 19 - 4 Start Address of Control Data	
	Change of description in 19.3.3 Vector Table	
	Change of Figure 19 - 4 Start Address of Control Data and Vector Table	
	Change of description in 19.4.2 Normal Mode	
	Change of Figure 19 - 16 Example 1 of using normal mode: Consecutively capturing A/D conversion results	
	Change of Table 30 - 1 Wiring Between RL78/I1D and Dedicated Flash Memory Programmer	CHAPTER 30 FLASH MEMORY
	Change of Figure 30 - 2 Communication with Dedicated Flash Memory Programmer	
	Change of Table 30 - 2 Pin Connection	
	Correction of error in 30.8.3 Procedure for accessing data flash memory	
	Change of 34.1 Absolute Maximum Ratings	CHAPTER 34 ELECTRICAL SPECIFICATIONS
	Change of 34.3.1 Pin characteristics	
	Correction of error in 34.3.2 Supply current characteristics	
	Change of note 3 in 34.3.2 Supply current characteristics	

(7/7)

Edition	Description	Chapter
Rev.2.00	Renamed to operation state control	CHAPTER 5 OPERATION STATE CONTROL
	Modification of operating current in table 5 - 2	
	Modification of description	CHAPTER 7 TIMER ARRAY UNIT
	Addition of description in figure 7 - 13	
	Correction of error in 17.7.2 UART reception	CHAPTER 17 SERIAL ARRAY UNIT
	Addition of description in 19.5.3 DTC Pending Instruction	CHAPTER 19 DATA TRANSFER CONTROLLER (DTC)
	Addition of caution in 21.4.4 Interrupt servicing during division instruction	CHAPTER 21 INTERRUPT FUNCTIONS
	Addition of description in 21.4.5 Interrupt request hold	
	Addition of description to note 2 of figure 23 - 4	CHAPTER 23 STANDBY FUNCTION
	Addition of note 3 to figure 23 - 4	
	Addition of description in (1) SNOOZE mode setting and operating statuses	
	Addition of caution to table 33 - 16	CHAPTER 33 INSTRUCTION SET
	Addition of note 7 in 34.3.2 Supply current characteristics	CHAPTER 34 ELECTRICAL SPECIFICATIONS
	Addition of description in 34.3.2 Supply current characteristics	
	Modification of description in 34.3.2 Supply current characteristics	
	Correction of error in 34.3.2 Supply current characteristics	
	Modification of package drawing in 35.2 24-pin products	CHAPTER 35 PACKAGE DRAWINGS
Rev.1.00	First edition issued	All

RL78/I1D User’s Manual: Hardware

Publication Date: Rev.0.50 Dec 20, 2013
Rev.2.40 Mar 20, 2023

Published by: Renesas Electronics Corporation

RL78/I1D

Renesas Electronics Corporation

R01UH0474EJ0240