User's Manual

16

RL78/G11

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.2.60 Mar 2024

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/G11 and design and develop application systems and programs for these devices.
The target products are as follows.

- 10-pin: R5F1051A
- 16-pin: R5F1054A
- 20-pin: R5F1056A
- 24-pin: R5F1057A
- 25-pin: R5F1058A

Purpose

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

Organization

The RL78/G11 manual is separated into two parts: this manual and the software edition (common to the RL78 family).

RL78/G11
User's Manual
Hardware
(This Manual)

- Pin functions
- Internal block functions
- Interrupts
- Other on-chip peripheral functions
- Electrical specifications

RL78 Family
User's Manual
Software

- 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/G11 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/G11 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-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 5.5 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 Mbytes

● General-purpose registers: (8-bit register × 8) × 4 banks

● On-chip RAM: 1.5 Kbytes

Code flash memory

● Code flash memory: 16 Kbytes

● Block size: 1 Kbytes

● On-chip debug function

● Self-programming (with boot swap function/flash shield window function)

Data flash memory

● Data flash memory: 2 Kbytes

● 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 5.5 V

High-speed on-chip oscillator

● Select from 48 MHz, 24 MHz, 16 MHz, 12 MHz, 8 MHz, 6 MHz, 4 MHz, 3 MHz, 2 MHz, and 1 MHz

● High accuracy: ±1.0% (V_{DD} = 1.8 to 5.5 V, T_A = -20 to +85°C)

Middle-speed on-chip oscillator

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

Operating ambient temperature

● T_A = -40 to +85°C (A: Consumer applications)

● 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 14 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 18 types can be linked to the specified peripheral function.

Serial interfaces

● Simplified SPI (CSI^Note 1^): 4 channels

● UART: 2 channels

● I^2C/simplified I^2C: 4 channels

● Multimaster I^2C: 2 channels

Timers

● 16-bit timer (TAU): 4 channels

● TKB: 1 channel

● 12-bit interval timer: 1 channel

● 8-bit interval timer: 2 channels

● Watchdog timer: 1 channel

A/D converter

● 8/10-bit resolution A/D converter (V_{DD} = 1.6 to 5.5 V)

● Analog input: 10 to 11 channels

● Internal reference voltage (1.45 V) and temperature sensor

D/A converter

● 8-bit resolution D/A converter (V_{DD} = 1.6 to 5.5 V)

● Analog input: 2 channels (channel 1: output to the ANO1 pin, channel 0: output to the comparator)

● Output voltage: 0 V to V_{DD}

● Real-time output function

Comparator

● 2 channels

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

PGA

● 1 channel

I/O ports

● I/O port: 17 to 21 (N-ch open drain I/O [V_{DD} withstand voltage[Note 2]/EV_{DD} withstand voltage[Note 3]]: 10 to 14)

● Can be set to N-ch open drain, TTL input buffer, and on-chip pull-up resistor

● Different potential interface: Can connect to a 1.8/2.5/3.0 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

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

Note 2. 16, 20, 24-pin products

Note 3. 25-pin products

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

○ ROM, RAM capacities

Flash ROM	Data flash	RAM	RL78/G11				
			10 pins	16 pins	20 pins	24 pins	25 pins
16 KB	2 KB	1.5 KB	R5F1051A	R5F1054A	R5F1056A	R5F1057A	R5F1058A

Remark 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.

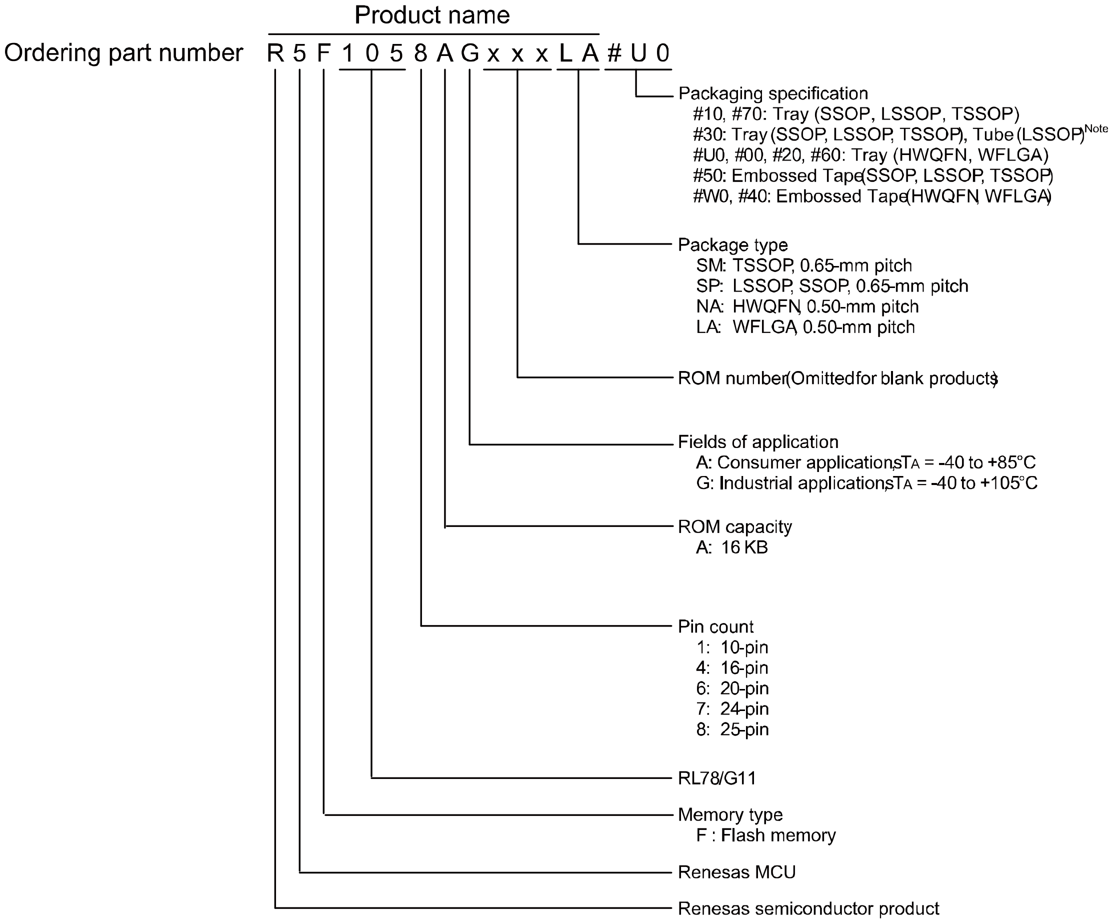
R5F105xA (x = 1, 4, 6, 7, 8): Start address FF900H

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/G11

<R>


Product name
Ordering part number R 5 F 1 0 5 8 A G x x x L A # U 0
Packaging specification
#10, #70: Tray (SSOP, LSSOP, TSSOP)
#30: Tray (SSOP, LSSOP, TSSOP), Tube (LSSOP)Note
#U0, #00, #20, #60: Tray (HWQFN, WFLGA)
#50: Embossed Tape (SSOP, LSSOP, TSSOP)
#W0, #40: Embossed Tape (HWQFN, WFLGA)
Package type
SM: TSSOP, 0.65-mm pitch
SP: LSSOP, SSOP, 0.65-mm pitch
NA: HWQFN, 0.50-mm pitch
LA: WFLGA, 0.50-mm pitch
ROM number (Omitted for blank products)
Fields of application
A: Consumer applications, TA = -40 to +85°C
G: Industrial applications, TA = -40 to +105°C
ROM capacity
A: 16 KB
Pin count
1: 10-pin
4: 16-pin
6: 20-pin
7: 24-pin
8: 25-pin
RL78/G11
Memory type
F : Flash memory
Renesas MCU
Renesas semiconductor product

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

Table 1 - 1 List of Ordering Part Numbers

	Pin Count	Package	Ordering Part Number		RENESAS Code
			Product Name	Packaging Specifications	
<R>	10 pins	10-pin plastic LSSOP (4.4 × 3.6 mm, 0.65-mm pitch)	R5F1051AGSP, R5F1051AASP	#10, #30, #50, #70	PLSP0010JA-A
<R>	16 pins	16-pin plastic SSOP (4.4 × 5.0 mm, 0.65-mm pitch)	R5F1054AGSP, R5F1054AASP	#10, #30, #50, #70	PRSP0016JC-B
<R>		16-pin plastic HWQFN (3 × 3 mm, 0.50-mm pitch)	R5F1054AGNA, R5F1054AANA	#00, #20, #40, #60	PWQN0016KD-A PWQN0016KE-A
	20 pins	20-pin plastic LSSOP (4.4 × 6.5 mm, 0.65-mm pitch)	R5F1056AGSP, R5F1056AASP	#30, #50	PLSP0020JB-A
<R>		20-pin plastic TSSOP (4.4 × 6.5 mm, 0.65-mm pitch)	R5F1056AGSM, R5F1056AASM	#10, #30, #50, #70	PTSP0020JI-A
	24 pins	24-pin plastic HWQFN (4 × 4 mm, 0.50-mm pitch)	R5F1057AGNA, R5F1057AANA	#U0, #W0	PWQN0024KE-A
<R>			R5F1057AGNA, R5F1057AANA	#00, #20, #40, #60	PWQN0024KF-A PWQN0024KH-A
	25 pins	25-pin plastic WFLGA (3 × 3 mm, 0.50-mm pitch)	R5F1058AGLA, R5F1058AALA	#U0, #W0	PWLG0025KA-A

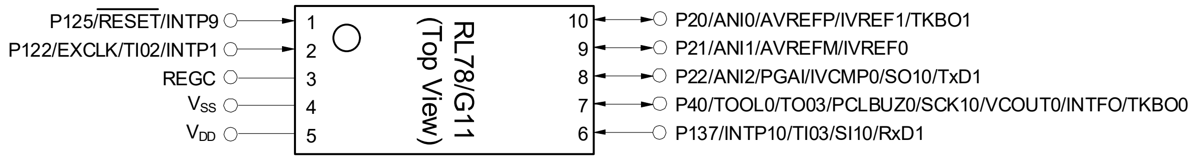
Caution 1. For the fields of application, refer to Figure 1 - 1 Part Number, Memory Size, and Package of RL78/G11.

Caution 2. 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)

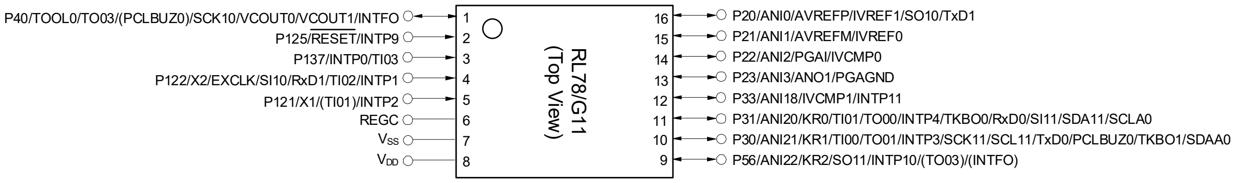
1.3.1 10-pin products

- 10-pin plastic LSSOP (4.4 × 3.6 mm, 0.65-mm pitch)

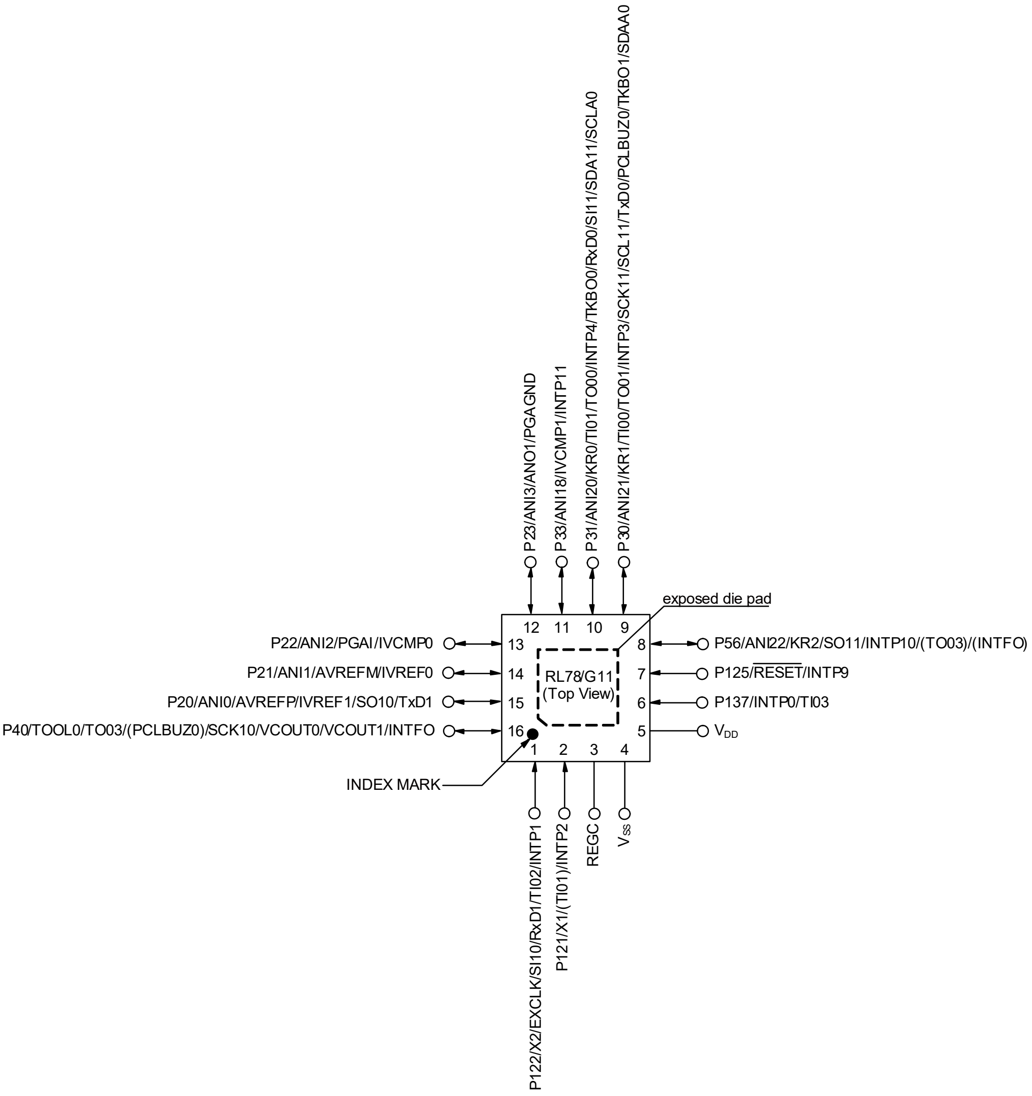

P125/$\overline{\text{RESET}}$/INTP9
P122/EXCLK/TI02/INTP1
REGC
V$_{SS}$
V$_{DD}$
1
2
3
4
5
RL78/G11
(Top View)
10
9
8
7
6
P20/ANI0/AVREFP/IVREF1/TKBO1
P21/ANI1/AVREFM/IVREF0
P22/ANI2/PGAI/IVCMP0/SO10/TxD1
P40/TOOL0/TO03/PCLBUZ0/SCK10/VCOUT0/INTFO/TKBO0
P137/INTP10/TI03/SI10/RxD1

1.3.2 16-pin products

- 16-pin plastic SSOP (4.4 × 5.0 mm, 0.65-mm pitch)

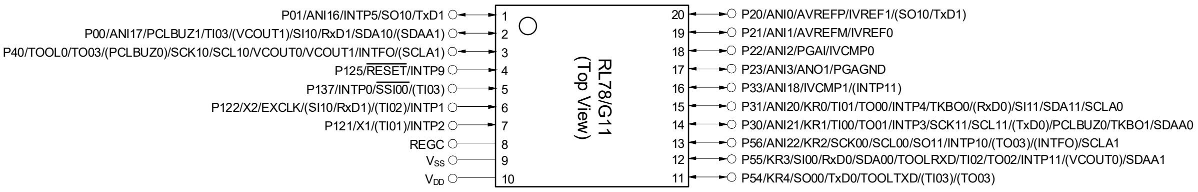

P40/TOOL0/TO03/(PCLBUZ0)/SCK10/VCOUT0/VCOUT1/INTFO
P125/$\overline{\text{RESET}}$/INTP9
P137/INTP0/TI03
P122/X2/EXCLK/SI10/RxD1/TI02/INTP1
P121/X1/(TI01)/INTP2
REGC
V$_{SS}$
V$_{DD}$
1
2
3
4
5
6
7
8
RL78/G11
(Top View)
16
15
14
13
12
11
10
9
P20/ANI0/AVREFP/IVREF1/SO10/TxD1
P21/ANI1/AVREFM/IVREF0
P22/ANI2/PGAI/IVCMP0
P23/ANI3/ANO1/PGAGND
P33/ANI18/IVCMP1/INTP11
P31/ANI20/KR0/TI01/TO00/INTP4/TKBO0/RxD0/SI11/SDA11/SCLA0
P30/ANI21/KR1/TI00/TO01/INTP3/SCK11/SCL11/TxD0/PCLBUZ0/TKBO1/SDAA0
P56/ANI22/KR2/SO11/INTP10/(TO03)/(INTFO)

- 16-pin plastic HWQFN (3 × 3 mm, 0.5-mm pitch)


P23/ANI3/ANO1/PGAGND
P33/ANI18/IVCMP1/INTP11
P31/ANI20/KR0/TI01/TO00/INTP4/TKBO0/RxD0/SI11/SDA11/SCLA0
P30/ANI21/KR1/TI00/TO01/INTP3/SCK11/SCL11/TxD0/PCLBUZ0/TKBO1/SDAA0
exposed die pad
12 11 10 9
P22/ANI2/PGAI/IVCMP0 13
P21/ANI1/AVREFM/IVREF0 14
P20/ANI0/AVREFP/IVREF1/SO10/TxD1 15
P40/TOOL0/TO03/(PCLBUZ0)/SCK10/VCOUT0/VCOUT1/INTFO 16
RL78/G11
(Top View)
8 P56/ANI22/KR2/SO11/INTP10/(TO03)/(INTFO)
7 P125/RESET/INTP9
6 P137/INTP0/TI03
5 VDD
1 2 3 4
INDEX MARK
P122/X2/EXCLK/SI10/RxD1/TI02/INTP1
P121/X1/(TI01)/INTP2
REGC
VSS

1.3.3 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)


P01/ANI16/INTP5/SO10/TxD1 1
P00/ANI17/PCLBUZ1/TI03/(VCOUT1)/SI10/RxD1/SDA10/(SDAA1) 2
P40/TOOL0/TO03/(PCLBUZ0)/SCK10/SCL10/VCOUT0/VCOUT1/INTFO/(SCLA1) 3
P125/RESET/INTP9 4
P137/INTP0/SSI00/(TI03) 5
P122/X2/EXCLK/(SI10/RxD1)/(TI02)/INTP1 6
P121/X1/(TI01)/INTP2 7
REGC 8
VSS 9
VDD 10
RL78/G11
(Top View)
20 P20/ANI0/AVREFP/IVREF1/(SO10/TxD1)
19 P21/ANI1/AVREFM/IVREF0
18 P22/ANI2/PGAI/IVCMP0
17 P23/ANI3/ANO1/PGAGND
16 P33/ANI18/IVCMP1/(INTP11)
15 P31/ANI20/KR0/TI01/TO00/INTP4/TKBO0/(RxD0)/SI11/SDA11/SCLA0
14 P30/ANI21/KR1/TI00/TO01/INTP3/SCK11/SCL11/(TxD0)/PCLBUZ0/TKBO1/SDAA0
13 P56/ANI22/KR2/SCK00/SCL00/SO11/INTP10/(TO03)/(INTFO)/SCLA1
12 P55/KR3/SI00/RxD0/SDA00/TOOLRXD/TI02/TO02/INTP11/(VCOUT0)/SDAA1
11 P54/KR4/SO00/TxD0/TOOLTXD/(TI03)/(TO03)

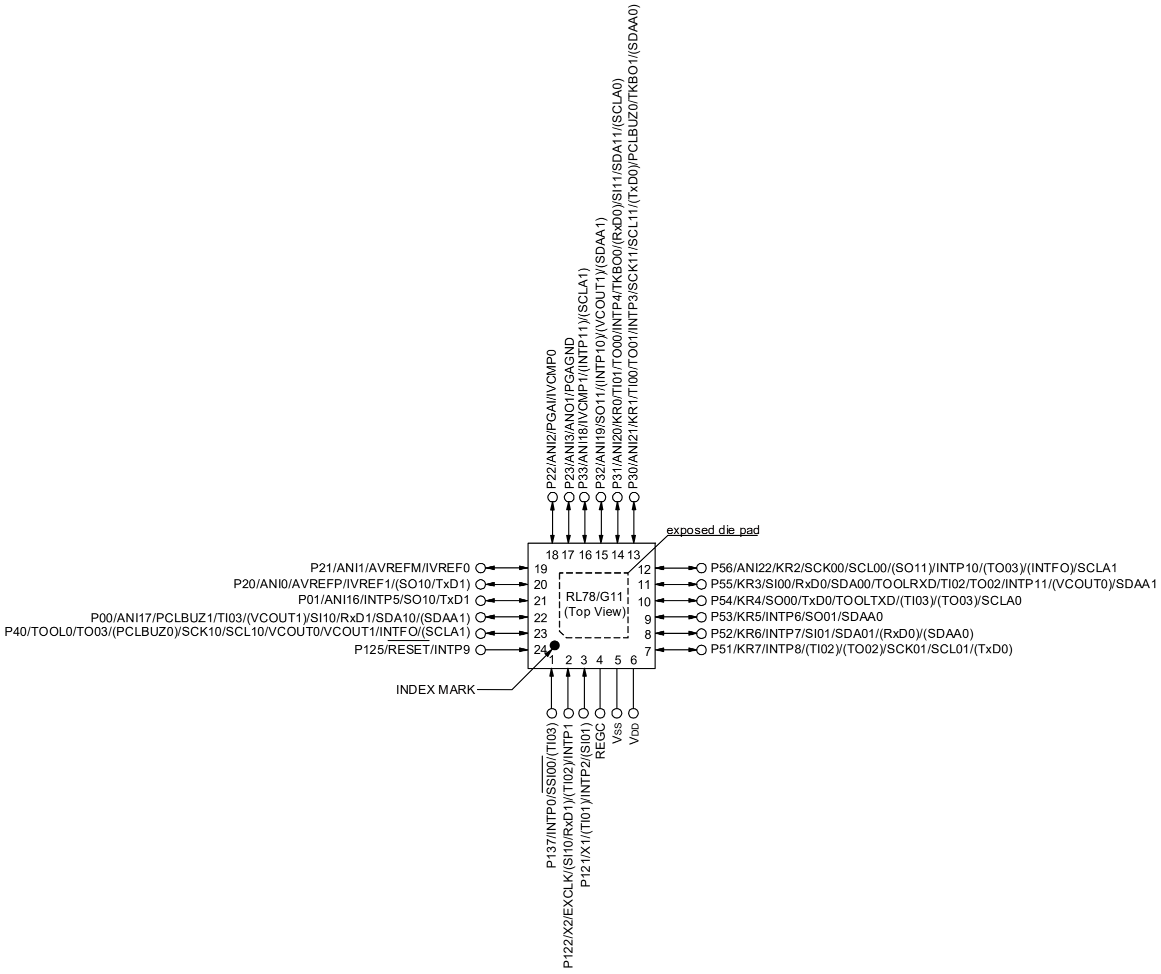
Caution Connect the REGC pin to Vss pin via a capacitor (0.47 to 1 µF).

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 to 3 (PIOR0 to PIOR3).

1.3.4 24-pin products

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


P22/ANI2/PGAI/IVCMP0
P23/ANI3/ANO1/PGAGND
P33/ANI18/IVCMP1/(INTP11)/(SCLA1)
P32/ANI19/SO11/(INTP10)/(VCOUT1)/(SDAA1)
P31/ANI20/KR0/TI01/TO00/INTP4/TKBO0/(RxD0)/SI11/SDA11/(SCLA0)
P30/ANI21/KR1/TI00/TO01/INTP3/SCK11/SCL11/(TxD0)/PCLBUZ0/TKBO1/(SDAA0)
18 17 16 15 14 13
exposed die pad
P21/ANI1/AVREFM/IVREF0 19
P20/ANI0/AVREFP/IVREF1/(SO10/TxD1) 20
P01/ANI16/INTP5/SO10/TxD1 21
P00/ANI17/PCLBUZ1/TI03/(VCOUT1)/SI10/RxD1/SDA10/(SDAA1) 22
P40/TOOL0/TO03/(PCLBUZ0)/SCK10/SCL10/VCOUT0/VCOUT1/INTFO/(SCLA1) 23
P125/RESET/INTP9 24
RL78/G11
(Top View)
12 P56/ANI22/KR2/SCK00/SCL00/(SO11)/INTP10/(TO03)/(INTFO)/SCLA1
11 P55/KR3/SI00/RxD0/SDA00/TOOLRXD/TI02/TO02/INTP11/(VCOUT0)/SDAA1
10 P54/KR4/SO00/TxD0/TOOLTXD/(TI03)/(TO03)/SCLA0
9 P53/KR5/INTP6/SO01/SDAA0
8 P52/KR6/INTP7/SI01/SDA01/(RxD0)/(SDAA0)
7 P51/KR7/INTP8/(TI02)/(TO02)/SCK01/SCL01/(TxD0)
INDEX MARK
1 2 3 4 5 6
P137/INTP0/SSI00/(TI03)
P122/X2/EXCLK/(SI10/RxD1)/(TI02)/INTP1
P121/X1/(TI01)/INTP2/(SI01)
REGC
VSS
VDD

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

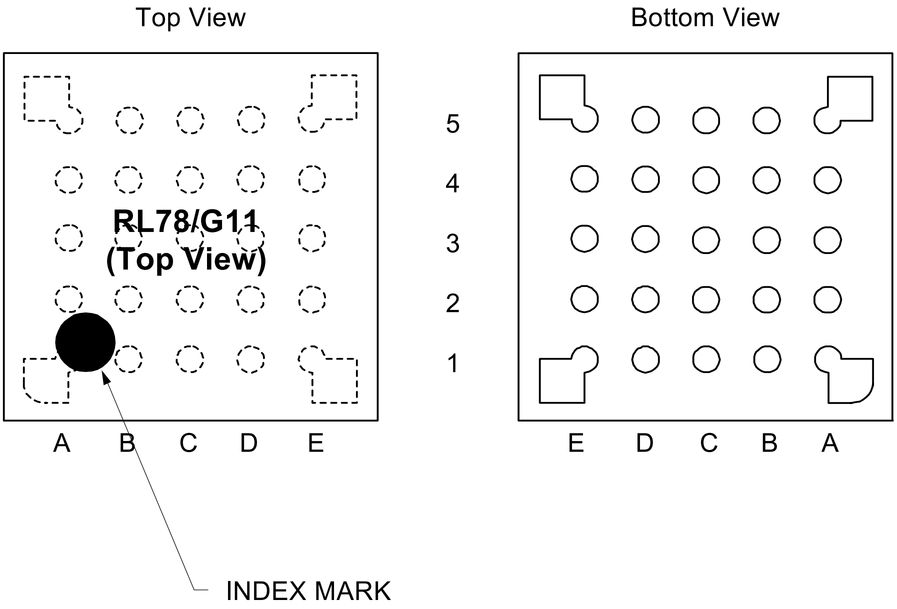
Remark 1. For pin identification, see **1.4 Pin Identification**.

Remark 2. It is recommended to connect an exposed die pad to Vss.

Remark 3. Functions in parentheses in the above figure can be assigned via settings in the peripheral I/O redirection register 0 to 3 (PIOR0 to PIOR3).

1.3.5 25-pin products

- 25-pin plastic WFLGA (3 × 3 mm, 0.5-mm pitch)



	A	B	C	D	E	
5	P40/TOOL0/TO03/(PCLBUZ0)/SCK10/SCL10/VCOUT0/VCOUT1/INTFO/(SCLA1)	P125/$\overline{\text{RESET}}$/INTP9	P01/ANI16/INTP5/SO10/TxD1	P20/ANI0/AVREFP/IVREF1/(SO10/TxD1)	P21/ANI1/AVREFM/IVREF0	5
4	P122/X2/EXCLK/(SI10/RxD1)/(TI02)/INTP1	P137/INTP0/$\overline{\text{SSI00}}$/(TI03)	P00/ANI17/PCLBUZ1/TI03/(VCOUT1)/SI10/RxD1/SDA10/(SDAA1)	P22/ANI2/PGAI/IVCMP0	P23/ANI3/ANO1/PGAGND	4
3	P121/X1/(TI01)/INTP2/(SI01)	VDD	EVDD	P33/ANI18/IVCMP1/(INTP11)/(SCLA1)	P32/ANI19/SO11/(INTP10)/(VCOUT1)/(SDAA1)	3
2	REGC	VSS	P30/ANI21/KR1/TI00/TO01/INTP3/SCK11/SCL11/(TxD0)/PCLBUZ0/TKBO1/(SDAA0)	P31/ANI20/KR0/TI01/TO00/INTP4/TKBO0/(RxD0)/SI11/SDA11/(SCLA0)	P56/ANI22/KR2/SCK00/SCL00/(SO11)/INTP10/(TO03)/(INTFO)/SCLA1	2
1	P51/KR7/INTP8/(TI02)/(TO02)/SCK01/SCL01/(TxD0)	P52/KR6/INTP7/SI01/SDA01/(RxD0)/(SDAA0)	P53/KR5/INTP6/SO01/SDAA0	P54/KR4/SO00/TxD0/TOOLTXD/(TI03)/(TO03)/SCLA0	P55/KR3/SI00/RxD0/SDA00/TOOLRXD/TI02/TO02/INTP11/(VCOUT0)/SDAA1	1
	A	B	C	D	E	

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

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 to 3 (PIOR0 to PIOR3).

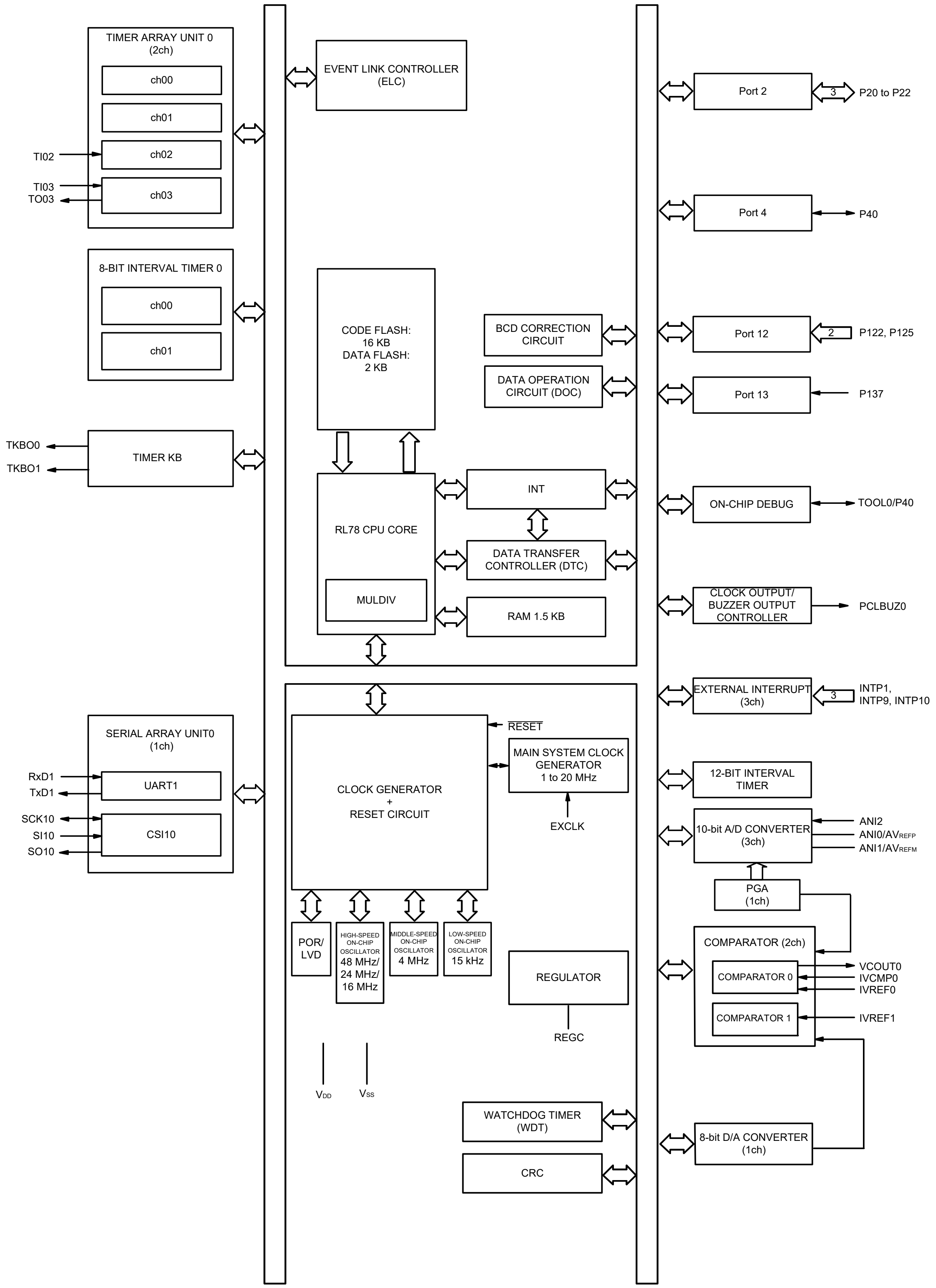
1.4 Pin Identification

ANI0 to ANI3, ANI16 to ANI22	:	Analog input
ANO1	:	Analog output
AV_{REFM}	:	A/D converter reference potential (- side) input
AV_{REFP}	:	A/D converter reference potential (+ side) input
EV_{DD}	:	Power supply
EXCLK	:	External clock input (main system clock)
INTP0 to INTP11	:	External interrupt input
INTFO	:	Interrupt Flag output
IVCMP0, IVCMP1	:	Comparator input
IVREF0, IVREF1	:	Comparator reference input
KR0 to KR7	:	Key return
PGAI, PGAGND	:	PGA Input
P00 to P01	:	Port 0
P20 to P23	:	Port 2
P30 to P33	:	Port 3
P40	:	Port 4
P51 to P56	:	Port 5
P121, P122, P125	:	Port 12
P137	:	Port 13

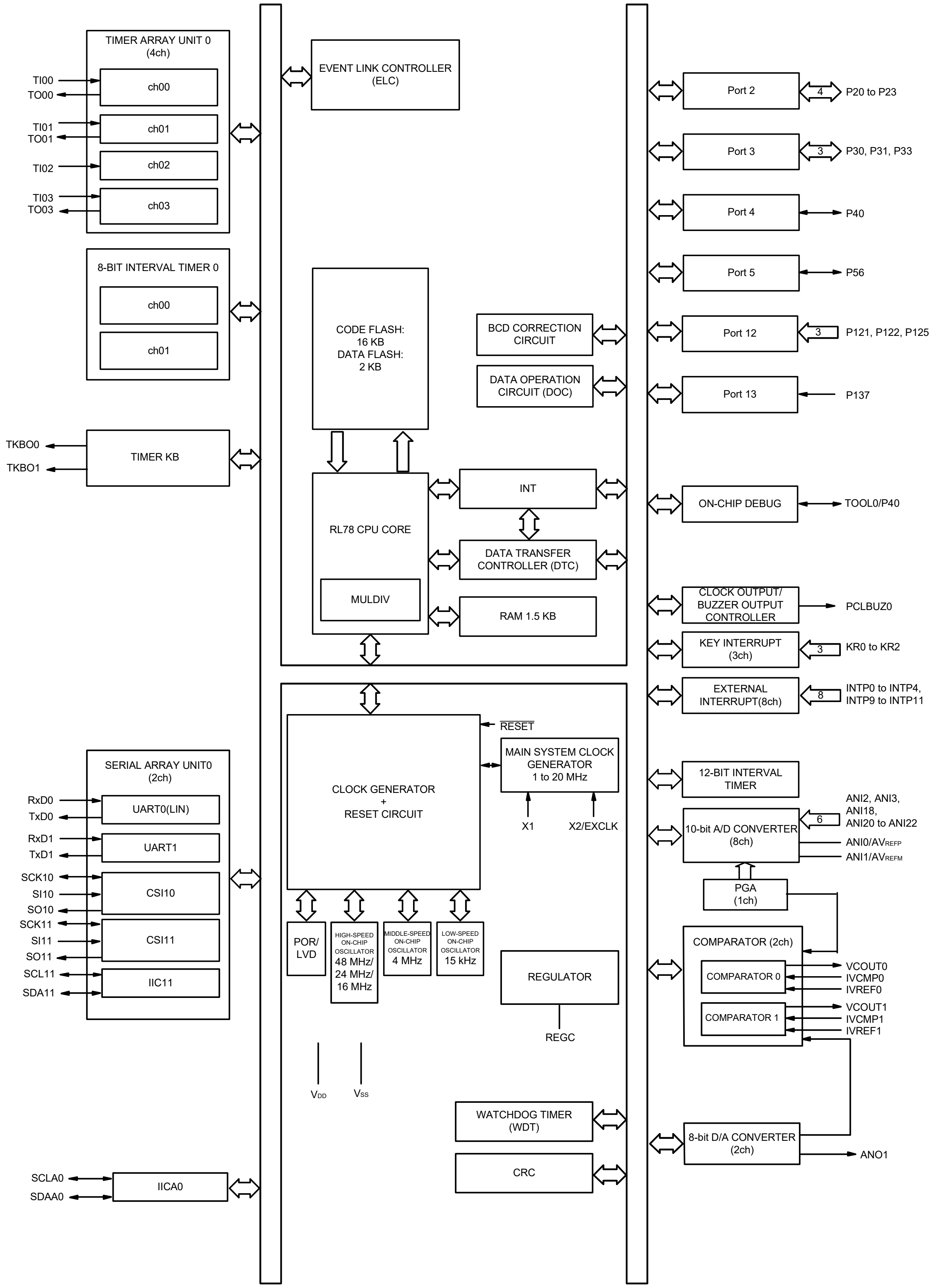
PCLBUZ0, PCLBUZ1	:	Programmable clock output/buzzer output
REGC	:	Regulator capacitance
$\overline{RESET}$	:	Reset
RxD0, RxD1	:	Receive data
SCK00, SCK01, SCK10, SCK11	:	Serial clock input/output
SCLA0, SCLA1	:	Serial clock input/output
SCL00, SCL01, SCL10, SCL11	:	Serial clock output
SDAA0, SDAA1	:	Serial data input/output
SDA00, SDA01, SDA10, SDA11	:	Serial data input/output
SI00, SI01, SI10, SI11	:	Serial data input
SO00, SO01, SO10, SO11	:	Serial data output
$\overline{SSI00}$	:	Serial interface chip select input
TI00 to TI03	:	Timer input
TKBO0, TKBO1	:	TMKB output
TO00 to TO03	:	Timer output
TOOL0	:	Data input/output for tool
TOOLRXD, TOOLTXD	:	Data input/output for external device
TxD0, TxD1	:	Transmit data
VCOUT0, VCOUT1	:	Comparator output
V_{DD}	:	Power supply
Vss	:	Ground
X1, X2	:	Crystal oscillator (main system clock)

1.5 Block Diagram

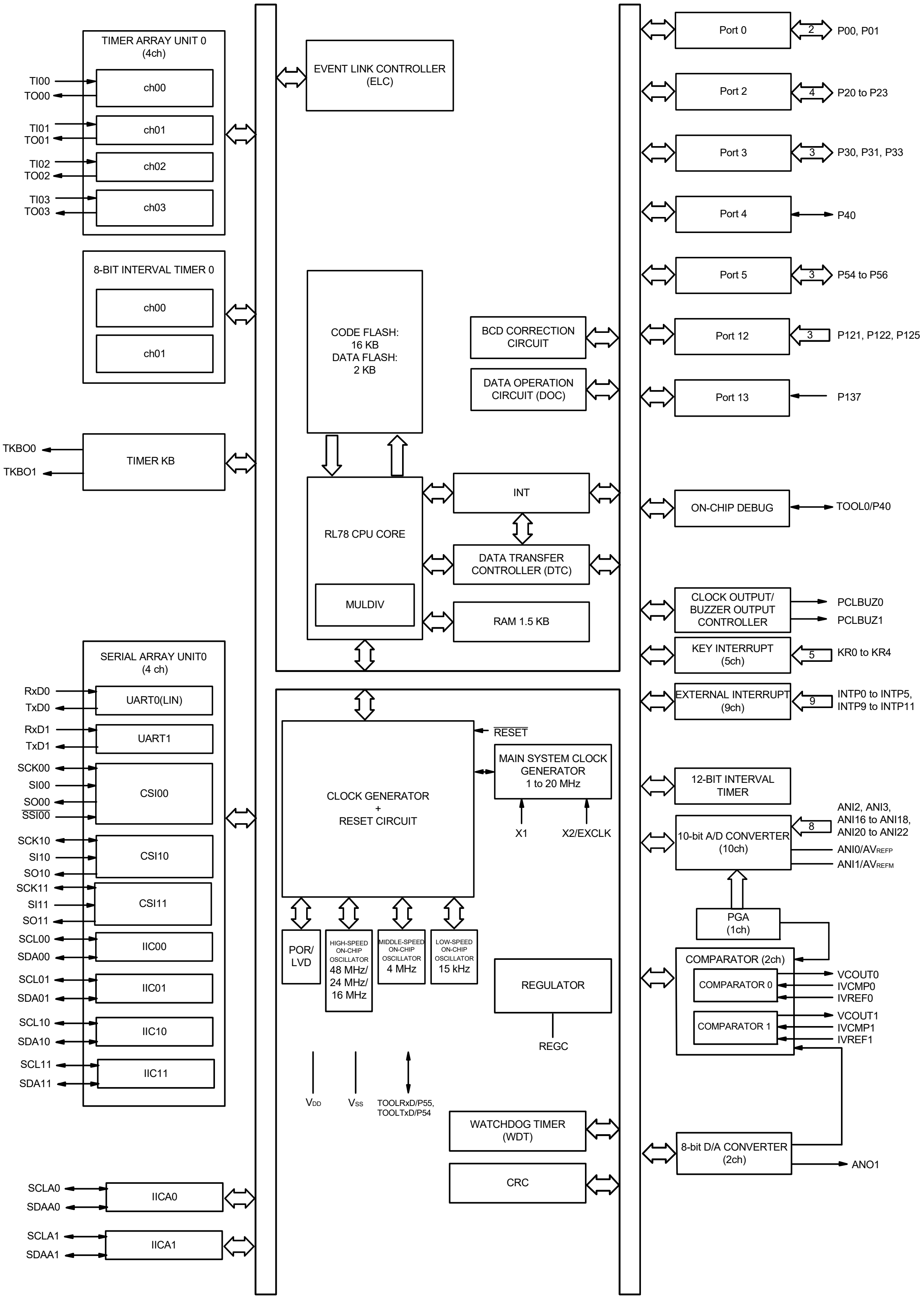
1.5.1 10-pin products



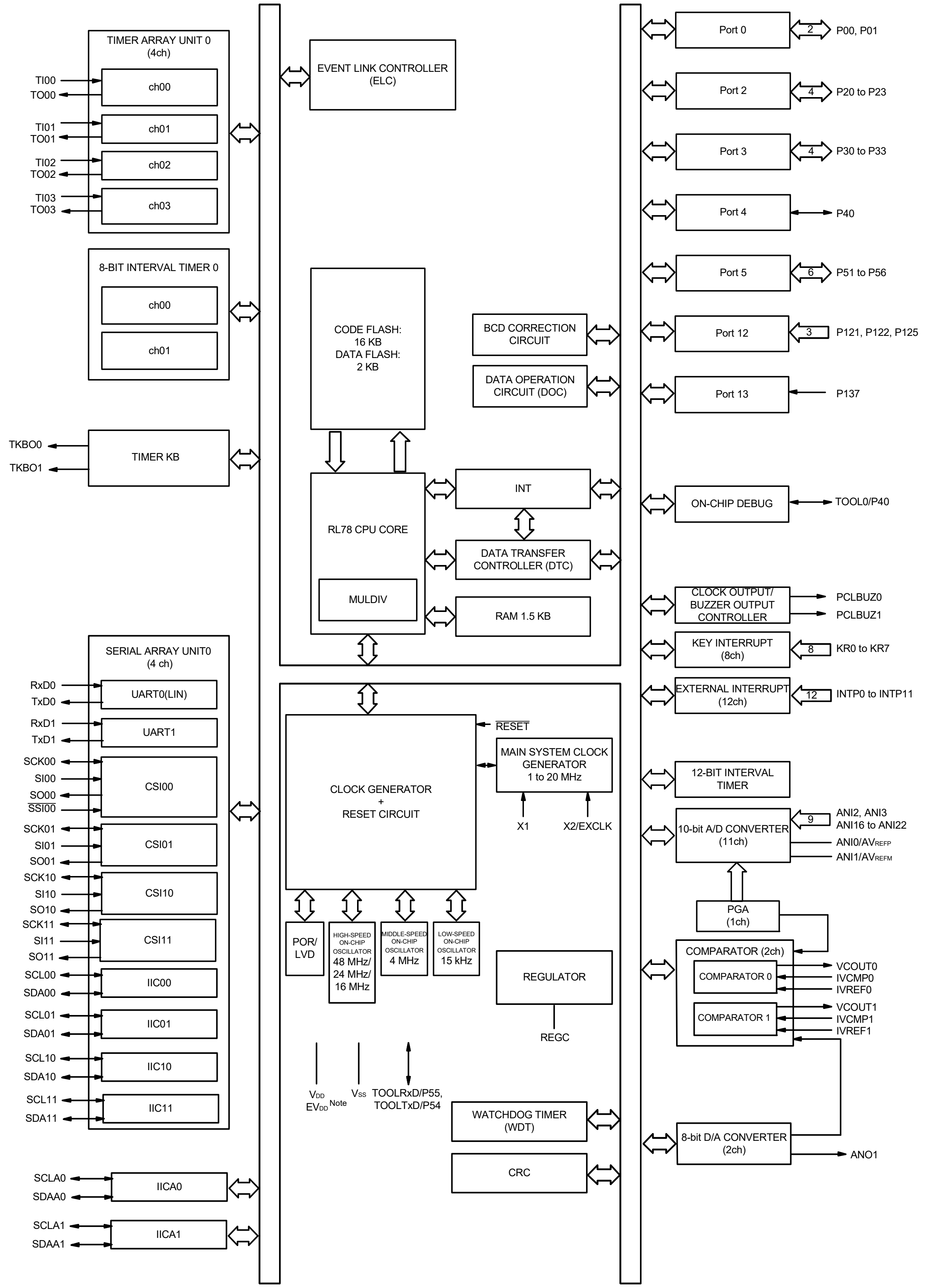
1.5.2 16-pin products



1.5.3 20-pin products



1.5.4 24-pin, 25-pin products



Note 25-pin products

1.6 Outline of Functions

This outline describes the functions at the time when Peripheral I/O redirection register 0 to 3 (PIOR0 to PIOR3) are set to 00H.

(1/2)

Item		10-pin	16-pin	20-pin	24-pin	25-pin
		R5F1051A	R5F1054A	R5F1056A	R5F1057A	R5F1058A
Code flash memory (KB)		16 Kbytes				
Data flash memory (KB)		2 Kbytes				
RAM		1.5 Kbytes				
Address space		1 Mbytes				
Main system clock	High-speed system clock (f_{MX})	X1 (crystal/ceramic) oscillation[Note], external main system clock input (EXCLK) 1 to 20 MHz: V_{DD} = 2.7 to 5.5 V 1 to 16 MHz: V_{DD} = 2.4 to 5.5 V 1 to 8 MHz: V_{DD} = 1.8 to 5.5 V 1 to 4 MHz: V_{DD} = 1.6 to 5.5 V				
	High-speed on-chip oscillator clock (f_{IH}) Max: 24 MHz	HS (High-speed main) mode: 1 to 24 MHz (V_{DD} = 2.7 to 5.5 V), HS (High-speed main) mode: 1 to 16 MHz (V_{DD} = 2.4 to 5.5 V), LS (Low-speed main) mode: 1 to 8 MHz (V_{DD} = 1.8 to 5.5 V), LV (Low-voltage main) mode: 1 to 4 MHz (V_{DD} = 1.6 to 5.5 V), LP (Low-power main) mode: 1 MHz (V_{DD} = 1.8 to 5.5 V)				
	Middle-speed on-chip oscillator clock (f_{IM}) Max: 4 MHz					
Subsystem clock	Low-speed on-chip oscillator clock (f_{IL})	15 kHz (typ.): V_{DD} = 1.6 to 5.5 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)				
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	7	13	17	21	
	CMOS I/O	4	9	13	17	
	CMOS input	3	4			
Timer	16-bit timer	4 channels				
	Watchdog timer	1 channel				
	Timer KB	1 channel				
	12-bit interval timer	1 channel				
	8/16-bit interval timer	2 channels (8 bit)/1 channel (16 bit)				
	Timer output	3	5	6		

Note 16, 20, 24, 25-pin products

Caution 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.

R5F105xA (x = 1, 4, 6, 7, 8): Start address FF900H

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		10-pin	16-pin	20-pin	24-pin	25-pin
		R5F1051A	R5F1054A	R5F1056A	R5F1057A	R5F1058A
Clock output/buzzer output		1		2		
		• 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) • 117 Hz, 234 Hz, 469 Hz, 938 Hz, 1.875 kHz, 3.75 kHz, 7.5 kHz, 15 kHz (subsystem clock: f_{IL} = 15 kHz operation)				
10-bit resolution A/D converter	External	3 channels	8 channels	10 channels	11 channels	
	Internal	1 channel				
8-bit D/A converter		1 channel	2 channels			
Comparator (Window Comparator)		1 channel	2 channels			
PGA		1 channel				
Data Operation Circuit (DOC)		Comparison, addition, and subtraction of 16-bit data				
Serial interface		[10-pin products] • Simplified SPI (CSI): 1 channel/UART: 1 channel [16-pin products] • Simplified SPI (CSI): 2 channels/UART: 2 channels/simplified I^2C: 1 channel [20-pin products] • Simplified SPI (CSI): 3 channels/UART: 2 channels/simplified I^2C: 3 channels [24-pin, 25-pin products] • Simplified SPI (CSI): 4 channels/UART: 2 channels/simplified I^2C: 4 channels				
	I^2C bus	None	1 channel	2 channels		
Data transfer controller (DTC)		13 sources	22 sources	23 sources	24 sources	
Event link controller (ELC)		Event input: 11 Event trigger output: 3	Event input: 16 Event trigger output: 4	Event input: 17 Event trigger output: 4	Event input: 18 Event trigger output: 4	
Vectored interrupt sources	Internal	20	24	25		
	External	3	9	10	13	
Key interrupt		None	3	5	8	
Reset		• Reset by $\overline{RESET}$ pin • Internal reset by watchdog timer • Internal reset by power-on-reset • Internal reset by voltage detector • Internal reset by illegal instruction execution • 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) 1.51 ± 0.06V (T_A = +85 to +105°C) • Power-down-reset: 1.50 ± 0.04 V (T_A = -40 to +85°C) 1.50 ± 0.06V (T_A = +85 to +105°C)				
Voltage detector	Power on	1.67 V to 4.06 V (14 stages)				
	Power down	1.63 V to 3.98 V (14 stages)				
On-chip debug function		Provided (Disable to tracing)				
Power supply voltage		V_{DD} = 1.6 to 5.5 V				
Operating ambient temperature		T_A = -40 to +85°C (Consumer applications) T_A = -40 to +105°C (Industrial applications)				

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) 10-pin products

Power Supply	Corresponding Pins
V_{DD}	P20 to P22, P40, P122, P125, and P137

(2) 16-pin products

Power Supply	Corresponding Pins
V_{DD}	P20 to P23, P30, P31, P33, P40, P56, P121, P122, P125, and P137

(3) 20-pin products

Power Supply	Corresponding Pins
V_{DD}	P00, P01, P20 to P23, P30, P31, P33, P40, P54 to P56, P121, P122, P125 and P137

(4) 24-pin products

Power Supply	Corresponding Pins
V_{DD}	P00, P01, P20 to P23, P30 to P33, P40, P51 to P56, P121, P122, P125 and P137

(5) 25-pin products

Power Supply	Corresponding Pins
V_{DD}	P20 to P23, P121, P122, P125 and P137
EV_{DD}	P00, P01, P30 to P33, P40, P51 to P56

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

2.1.1 10-pin Products

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P20	4-9-3	I/O	Analog input port	ANI0/AVREFP/IVREF1/TKBO1	Port 2. 3-bit I/O port. Input/output can be specified in 1-bit units.
P21	4-9-4			ANI1/AVREFM/IVREF0	
P22	4-18-2			ANI2/PGAI/IVCMP0/SO10/TxD1	
P40	8-1-4	I/O	Input port	TOOL0/TO03/PCLBUZ0/SCK10/ VCOUT0/INTFO/TKBO0	Port 4. 1-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 of P40.
P122	2-2-1	Input	Input port	EXCLK/TI02/INTP1	Port 12. 2-bit input-only port. P125 is also used for the input pin for external reset ($\overline{\text{RESET}}$). To use the pin for external reset, set the PORTSELB bit in the option byte (000C1H) to 1. For P125, use of an on-chip pull-up resistor can be specified by a software setting.
P125	3-1-1			$\overline{\text{RESET}}$/INTP9	
P137	2-1-2	Input	Input port	INTP10/TI03/SI10/RxD1	Port 13. 1-bit input-only port.

2.1.2 16-pin Products

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P20	4-9-3	I/O	Analog input port	P20/ANI0/AVREFP/IVREF1/SO10/TxD1	Port 2. 4-bit I/O port. Input/output can be specified in 1-bit units.
P21	4-9-4			ANI1/AVREFM/IVREF0	
P22	4-18-2			ANI2/PGAI/IVCMP0	
P23	4-8-2			ANI3/ANO1/PGAGND	
P30	8-1-4	I/O	Analog input port	ANI21/KR1/TI00/TO01/INTP3/SCK11/SCL11/TxD0/PCLBUZ0/TKBO1/SDAA0	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 of P30, P31, and P33. Input of P30 and P31 can be set to TTL input buffer. Output of P30 and P31 can be set to N-ch open-drain output.
P31				ANI20/KR0/TI01/TO00/INTP4/TKBO0/RxD0/SI11/SDA11/SCLA0	
P33	7-9-4			ANI18/IVCMP1/INTP11	
P40	8-1-4	I/O	Input port	TOOL0/TO03/(PCLBUZ0)/SCK10/VCOUT0/VCOUT1/INTFO	Port 4. 1-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 of P40.
P56	8-3-4	I/O	Analog input port	ANI22/KR2/SO11/INTP10/(TO03)/(INTFO)	Port 5. 1-bit I/O port. Use of an on-chip pull-up resistor can be specified by a software setting at input of P56. Output of P56 can be set to N-ch open-drain output.
P121	2-2-1	Input	Input port	X1/(TI01)/INTP2	Port 12. 3-bit input-only port. P125 is also used for the input pin for external reset ($\overline{\text{RESET}}$). To use the pin for external reset, set the PORTSELB bit in the option byte (000C1H) to 1. For P125, use of an on-chip pull-up resistor can be specified by a software setting.
P122				X2/EXCLK/SI10/RxD1/TI02/INTP1	
P125	3-1-1			$\overline{\text{RESET}}$/INTP9	
P137	2-1-2	Input	Input port	INTP0/TI03	Port 13. 1-bit input-only port.

2.1.3 20-pin Products

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	8-3-4	I/O	Analog input port	ANI17/PCLBUZ1/TI03/(VCOUT1)/SI10/RxD1/SDA10/(SDAA1)	Port 0. 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 of P00 and P01. Input of P00 can be set to TTL input buffer. Output of P00 and P01 can be set to N-ch open-drain output.
P01	7-3-4			ANI16/INTP5/SO10/TxD1	
P20	4-9-3	I/O	Analog input port	ANI0/AVREFP/IVREF1/(SO10/TXD1)	Port 2. 4-bit I/O port. Input/output can be specified in 1-bit units.
P21	4-9-4			ANI1/AVREFM/IVREF0	
P22	4-18-2			ANI2/PGAI/IVCMP0	
P23	4-8-2			ANI3/ANO1/PGAGND	
P30	8-1-4	I/O	Analog input port	ANI21/KR1/TI00/TO01/INTP3/SCK11/SCL11/(TxD0)/PCLBUZ0/TKBO1/SDAA0	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 of P30, P31 and P33. Input of P30 and P31 can be set to TTL input buffer. Output of P30 and P31 can be set to N-ch open-drain output.
P31				ANI20/KR0/TI01/TO00/INTP4/TKBO0/(RxD0)/SI11/SDA11/SCLA0	
P33	7-9-4			ANI18/IVCMP1/(INTP11)	
P40	8-1-4	I/O	Input port	TOOL0/TO03/(PCLBUZ0)/SCK10/SCL10/VCOUT0/VCOUT1/INTFO/(SCLA1)	Port 4. 1-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 of P40. Input of P40 can be set to TTL input buffer. Output of P40 can be set to N-ch open-drain output.
P54	8-1-4	I/O	Input port	KR4/SO00/TxD0/TOOLTXD/(TI03)/(TO03)	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 of P54 to P56. Input of P55 and P56 can be set to TTL input buffer. Output of P54 to P56 can be set to N-ch open-drain output.
P55				KR3/SI00/RxD0/SDA00/TOOLRXD/TI02/TO02/INTP11/(VCOUT0)/SDAA1	
P56	8-3-4		Analog input port	ANI22/KR2/SCK00/SCL00/SO11/INTP10/(TO03)/(INTFO)/SCLA1	
P121	2-2-1	Input	Input port	X1/(TI01)/INTP2	Port 12. 3-bit input-only port. P125 is also used for the input pin for external reset ($\overline{\text{RESET}}$). To use the pin for external reset, set the PORTSELB bit in the option byte (000C1H) to 1. For P125, use of an on-chip pull-up resistor can be specified by a software setting.
P122				X2/EXCLK/(SI10/RxD1)/(TI02)/INTP1	
P125	3-1-1			$\overline{\text{RESET}}$/INTP9	
P137	2-1-2	Input	Input port	INTP0/$\overline{\text{SSI00}}$/(TI03)	Port 13. 1-bit input-only port.

Remark Functions in parentheses in the table can be assigned via settings in the peripheral I/O redirection register 0 to 3 (PIOR0 to PIOR3).

2.1.4 24-pin Products

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	8-3-4	I/O	Analog input port	ANI17/PCLBUZ1/TI03/(VCOUT1)/SI10/RxD1/SDA10/(SDAA1)	Port 0. 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 of P00 and P01. Input of P00 can be set to TTL input buffer. Output of P00 and P01 can be set to N-ch open-drain output.
P01	7-3-4			ANI16/INTP5/SO10/TxD1	
P20	4-9-3	I/O	Analog input port	ANI0/AVREFP/IVREF1/(SO10/TXD1)	Port 2. 4-bit I/O port. Input/output can be specified in 1-bit units.
P21	4-9-4			ANI1/AVREFM/IVREF0	
P22	4-18-2			ANI2/PGAI/IVCMP0	
P23	4-8-2			ANI3/ANO1/PGAGND	
P30	8-1-4	I/O	Analog input port	ANI21/KR1/TI00/TO01/INTP3/SCK11/SCL11/(TxD0)/PCLBUZ0/TKBO1/(SDAA0)	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 of P30 to P33. Input of P30 to P32 can be set to TTL input buffer. Output of P30 to P33 can be set to N-ch open-drain output.
P31				ANI20/KR0/TI01/TO00/INTP4/TKBO0/(RxD0)/SI11/SDA11/(SCLA0)	
P32				ANI19/SO11/(INTP10)/(VCOUT1)/(SDAA1)	
P33	7-9-4			ANI18/IVCMP1/(INTP11)/(SCLA1)	
P40	8-1-4	I/O	Input port	TOOL0/TO03/(PCLBUZ0)/SCK10/SCL10/VCOUT0/VCOUT1/INTFO/(SCLA1)	Port 4. 1-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 of P40. Input of P40 can be set to TTL input buffer. Output of P40 can be set to N-ch open-drain output.
P51	8-1-4	I/O	Input port	KR7/INTP8/(TI02)/(TO02)/SCK01/SCL01/(TxD0)	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 of P51 to P56. Input of P51 to P56 can be set to TTL input buffer. Output of P51 to P56 can be set to N-ch open-drain output.
P52				KR6/INTP7/SI01/SDA01/(RxD0)/(SDAA0)	
P53				KR5/INTP6/SO01/SDAA0	
P54				KR4/SO00/TxD0/TOOLTXD/(TI03)/(TO03)/SCLA0	
P55				KR3/SI00/RxD0/SDA00/TOOLRXD/TI02/TO02/INTP11/(VCOUT0)/SDAA1	
P56	8-3-4		Analog input port	ANI22/KR2/SCK00/SCL00/(SO11)/INTP10/(TO03)/(INTFO)/SCLA1	
P121	2-2-1	Input	Input port	X1/(TI01)/INTP2/(SI01)	Port 12. 3-bit input-only port. P125 is also used for the input pin for external reset ($\overline{\text{RESET}}$). To use the pin for external reset, set the PORTSELB bit in the option byte (000C1H) to 1. For P125, use of an on-chip pull-up resistor can be specified by a software setting.
P122				X2/EXCLK/(SI10/RxD1)/(TI02)/INTP1	
P125	3-1-1			$\overline{\text{RESET}}$/INTP9	
P137	2-1-2	Input	Input port	INTP0/$\overline{\text{SSI00}}$/(TI03)	Port 13. 1-bit input-only port.

Remark Functions in parentheses in the table can be assigned via settings in the peripheral I/O redirection register 0 to 3 (PIOR0 to PIOR3).

2.1.5 25-pin Products

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	8-3-4	I/O	Analog input port	ANI17/PCLBUZ1/TI03/(VCOUT1)/SI10/RxD1/SDA10/(SDAA1)	Port 0. 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 of P00 and P01. Input of P00 can be set to TTL input buffer. Output of P00 and P01 can be set to N-ch open-drain output.
P01	7-3-4			ANI16/INTP5/SO10/TxD1	
P20	4-9-3	I/O	Analog input port	ANI0/AVREFP/IVREF1/(SO10/TXD1)	Port 2. 4-bit I/O port. Input/output can be specified in 1-bit units.
P21	4-9-4			ANI1/AVREFM/IVREF0	
P22	4-18-2			ANI2/PGAI/IVCMP0	
P23	4-8-2			ANI3/ANO1/PGAGND	
P30	8-1-4	I/O	Analog input port	ANI21/KR1/TI00/TO01/INTP3/SCK11/SCL11/(TxD0)/PCLBUZ0/TKBO1/(SDAA0)	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 of P30 to P33. Input of P30 to P32 can be set to TTL input buffer. Output of P30 to P33 can be set to N-ch open-drain output.
P31				ANI20/KR0/TI01/TO00/INTP4/TKBO0/(RxD0)/SI11/SDA11/(SCLA0)	
P32				ANI19/SO11/(INTP10)/(VCOUT1)/(SDAA1)	
P33	7-9-4			ANI18/IVCMP1/(INTP11)/(SCLA1)	
P40	8-1-4	I/O	Input port	TOOL0/TO03/(PCLBUZ0)/SCK10/SCL10/VCOUT0/VCOUT1/INTFO/(SCLA1)	Port 4. 1-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 of P40. Input of P40 can be set to TTL input buffer. Output of P40 can be set to N-ch open-drain output.
P51	8-1-4	I/O	Input port	KR7/INTP8/(TI02)/(TO02)/SCK01/SCL01/(TxD0)	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 of P51 to P56. Input of P51 to P56 can be set to TTL input buffer. Output of P51 to P56 can be set to N-ch open-drain output.
P52				KR6/INTP7/SI01/SDA01/(RxD0)/(SDAA0)	
P53				KR5/INTP6/SO01/SDAA0	
P54				KR4/SO00/TxD0/TOOLTXD/(TI03)/(TO03)/SCLA0	
P55				KR3/SI00/RxD0/SDA00/TOOLRXD/TI02/TO02/INTP11/(VCOUT0)/SDAA1	
P56	8-3-4		Analog input port	ANI22/KR2/SCK00/SCL00/(SO11)/INTP10/(TO03)/(INTFO)/SCLA1	
P121	2-2-1	Input	Input port	X1/(TI01)/INTP2/(SI01)	Port 12. 3-bit input-only port. P125 is also used for the input pin for external reset ($\overline{\text{RESET}}$). To use the pin for external reset, set the PORTSELB bit in the option byte (000C1H) to 1. For P125, use of an on-chip pull-up resistor can be specified by a software setting.
P122				X2/EXCLK/(SI10/RxD1)/(TI02)/INTP1	
P125	3-1-1			$\overline{\text{RESET}}$/INTP9	
P137	2-1-2	Input	Input port	INTP0/$\overline{\text{SSI00}}$/(TI03)	Port 13. 1-bit input-only port.

Remark Functions in parentheses in the table can be assigned via settings in the peripheral I/O redirection register 0 to 3 (PIOR0 to PIOR3).

2.2 Functions other than port pins

2.2.1 Functions for each product

(1/3)

Function Name	25-pin	24-pin	20-pin	16-pin	10-pin
ANI0	√	√	√	√	√
ANI1	√	√	√	√	√
ANI2	√	√	√	√	√
ANI3	√	√	√	√	—
ANI16	√	√	√	—	—
ANI17	√	√	√	—	—
ANI18	√	√	√	√	—
ANI19	√	√	—	—	—
ANI20	√	√	√	√	—
ANI21	√	√	√	√	—
ANI22	√	√	√	√	—
IVCMP0	√	√	√	√	√
IVCMP1	√	√	√	√	—
IVREF0	√	√	√	√	√
IVREF1	√	√	√	√	√
PGAI	√	√	√	√	√
PGAGND	√	√	√	√	—
KR0	√	√	√	√	—
KR1	√	√	√	√	—
KR2	√	√	√	√	—
KR3	√	√	√	—	—
KR4	√	√	√	—	—
KR5	√	√	—	—	—
KR6	√	√	—	—	—
KR7	√	√	—	—	—
TI00	√	√	√	√	—
TI01	√	√	√	√	—
TI02	√	√	√	√	√
TI03	√	√	√	√	√
TO00	√	√	√	√	—
TO01	√	√	√	√	—
TO02	√	√	√	—	—
TO03	√	√	√	√	√
ANO1	√	√	√	√	—
VCOUT0	√	√	√	√	√
VCOUT1	√	√	√	√	—
TKBO0	√	√	√	√	√
TKBO1	√	√	√	√	√
RXD0	√	√	√	√	—

(2/3)

Function Name	25-pin	24-pin	20-pin	16-pin	10-pin
RXD1	√	√	√	√	√
TXD0	√	√	√	√	—
TXD1	√	√	√	√	√
SCLA0	√	√	√	√	—
SCLA1	√	√	√	—	—
SCL00	√	√	√	—	—
SCL01	√	√	—	—	—
SCL10	√	√	√	—	—
SCL11	√	√	√	√	—
SDAA0	√	√	√	√	—
SDAA1	√	√	√	—	—
SDA00	√	√	√	—	—
SDA01	√	√	—	—	—
SDA10	√	√	√	—	—
SDA11	√	√	√	√	—
SCK00	√	√	√	—	—
SCK01	√	√	—	—	—
SCK10	√	√	√	√	√
SCK11	√	√	√	√	—
SI00	√	√	√	—	—
SI01	√	√	—	—	—
SI10	√	√	√	√	√
SI11	√	√	√	√	—
SO00	√	√	√	—	—
SO01	√	√	—	—	—
SO10	√	√	√	√	√
SO11	√	√	√	√	—
$\overline{\text{SSI00}}$	√	√	√	—	—
INTP0	√	√	√	√	—
INTP1	√	√	√	√	√
INTP2	√	√	√	√	—
INTP3	√	√	√	√	—
INTP4	√	√	√	√	—
INTP5	√	√	√	—	—
INTP6	√	√	—	—	—
INTP7	√	√	—	—	—
INTP8	√	√	—	—	—
INTP9	√	√	√	√	√
INTP10	√	√	√	√	√

(3/3)

Function Name	25-pin	24-pin	20-pin	16-pin	10-pin
INTP11	√	√	√	√	—
INTFO	√	√	√	√	√
PCLBUZ0	√	√	√	√	√
PCLBUZ1	√	√	√	—	—
EXCLK	√	√	√	√	√
X1	√	√	√	√	—
X2	√	√	√	√	—
$\overline{\text{RESET}}$	√	√	√	√	√
REGC	√	√	√	√	√
V_{DD}	√	√	√	√	√
EV_{DD}	√	—	—	—	—
AV_{REFP}	√	√	√	√	√
AV_{REFM}	√	√	√	√	√
V_{SS}	√	√	√	√	√
TOOLRXD	√	√	√	—	—
TOOLTXD	√	√	√	—	—
TOOL0	√	√	√	√	√

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

(1/2)

Function Name	I/O	Function
ANI0 to ANI3	Input	A/D converter analog input (V_{DD} connection)
ANI16 to ANI22	Input	A/D converter analog input (25-pin products: EV_{DD} connection, 24-, 20-, 16-pin products: V_{DD} connection)
IVCMP0, IVCMP1	Input	Comparator analog voltage input
IVREF0, IVREF1	Input	Comparator reference voltage input
PGAI, PGAGND	Input	PGA input
KR0 to KR7	Input	Key interrupt input
TI00 to TI03	Input	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
ANO1	Output	D/A converter output
VCOUT0, VCOUT1	Output	Comparator output
TKBO0, TKBO1	Output	16-bit timer KB0 output
RxD0, RXD1	Input	Serial data input pins of serial interfaces UART0 and UART1
TxD0, TXD1	Output	Serial data output pins of serial interfaces UART0 and UART1
SCLA0, SCLA1	I/O	Serial clock I/O pins of serial interfaces IICA0 and IICA1
SCL00, SCL01, SCL10, SCL11	Output	Serial clock output pins of serial interfaces IIC00, IIC01, IIC10, and IIC11
SDAA0, SDAA1	I/O	Serial data I/O pins of serial interfaces IICA0 and IICA1
SDA00, SDA01, SDA10, SDA11	I/O	Serial data I/O pins of serial interfaces IIC00, IIC01, IIC10, and IIC11
SCK00, SCK01, SCK10, SCK11	I/O	Serial clock output pins of serial interfaces CSI00, CSI01, CSI10, and CSI11
SI00, SI01, SI10, SI11	Input	Serial data input pins of serial interfaces CSI00, CSI01, CSI10, and CSI11
SO00, SO01, SO10, SO11	Output	Serial data output pins of serial interfaces CSI00, CSI01, CSI10, and CSI11
$\overline{\text{SSI00}}$	Input	Chip select input pin of serial interface CSI00
INTP0 to INTP11	Input	External interrupt request input pin
INTFO	Output	Interrupt flag output
PCLBUZ0, PCLBUZ1	Output	Clock output/buzzer output
EXCLK	Input	External clock input for main system clock
X1, X2	—	Resonator connection for main system clock
$\overline{\text{RESET}}$	Input	This is the active-low system reset input.
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.
V_{DD}	—	Power supply for P20 to P23, P121, P122, P125 and P137 pins (25-pin products) Power supply for all pins (24-, 20-, 16-, 10-pin products)
EV_{DD}	—	Power supply for other than P20 to P23, P121, P122, P125 and P137 pins (25-pin product)
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 all pins
TOOLRxD	Input	UART reception pin for the external device connection used during flash memory programming

(2/2)

Function Name	I/O	Function
TOOLTxD	Output	UART transmission pin for the external device connection used during flash memory programming
TOOL0	I/O	Data I/O for flash memory programmer/debugger

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 31.4 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 EV_{DD} to EV_{SS} lines and 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

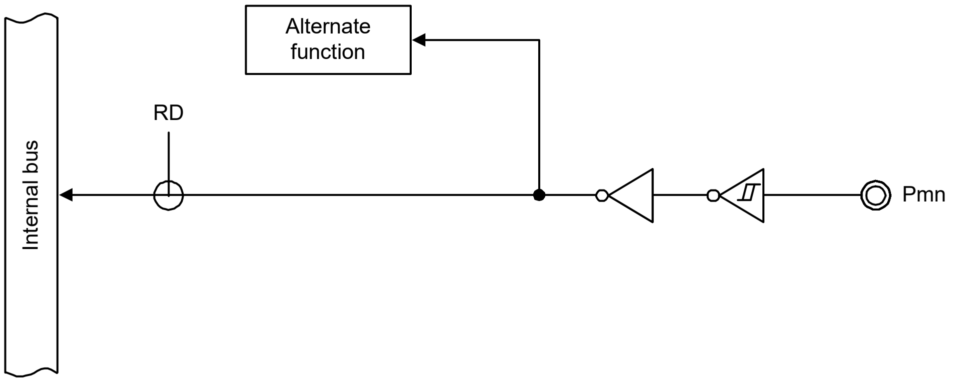
Pin Name	I/O	Recommended Connection of Unused Pins
P00 to P01	I/O	Input: Independently connect to EV$_{DD}$[Note] or V$_{SS}$ via a resistor. Output: Leave open.
P20 to P23		Input: Independently connect to V$_{DD}$ or V$_{SS}$ via a resistor. Output: Leave open.
P30 to P33		Input: Independently connect to EV$_{DD}$[Note] or V$_{SS}$ via a resistor. Output: Leave open.
P40/TOOL0		Input: Independently connect to EV$_{DD}$[Note] via a resistor, or leave open. Output: Leave open.
P51 to P56		Input: Independently connect to EV$_{DD}$[Note] or V$_{SS}$ via a resistor. Output: Leave open.
P121, P122	Input	Independently connect to V$_{DD}$ or V$_{SS}$ via a resistor.
<R> P125/$\overline{\text{RESET}}$	Input	PORTSELB = 0: Independently connect to V$_{DD}$ via a resistor. PORTSELB = 1: Leave open or connect to V$_{DD}$.
P137	Input	Independently connect to V$_{DD}$ or V$_{SS}$ via a resistor.
REGC	—	Connect to V$_{SS}$ via a capacitor (0.47 to 1 μF).

Note This is only possible with 25-pin products. Connect unused pins to V$_{DD}$ in the case of 24, 20, 16, and 10-pin products.

2.4 Pin Block Diagrams

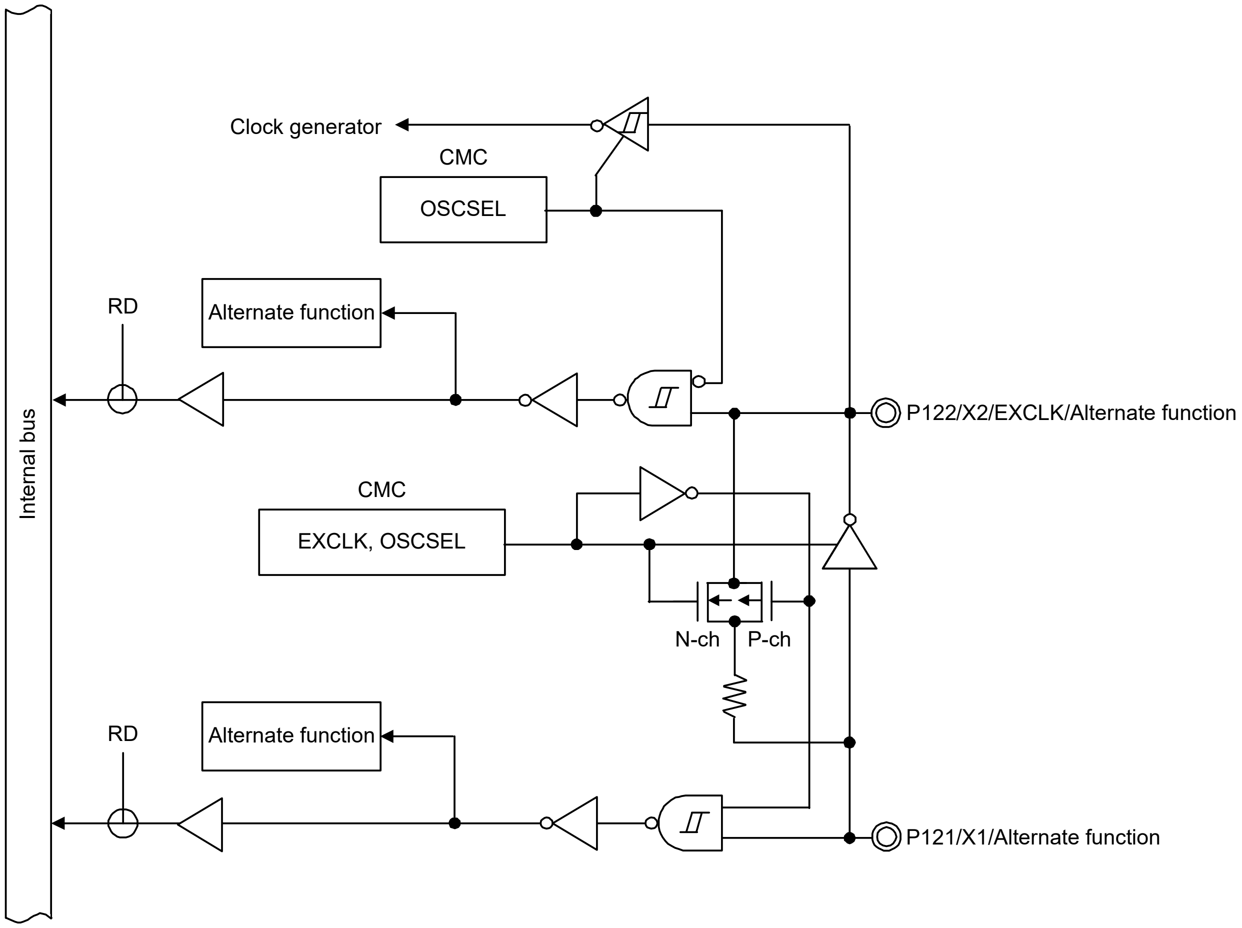
For the pin types listed in 2.1.1 10-pin Products to 2.1.5 25-pin Products, pin block diagrams are shown in Figures 2 - 1 to 2 - 11.

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



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

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



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

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

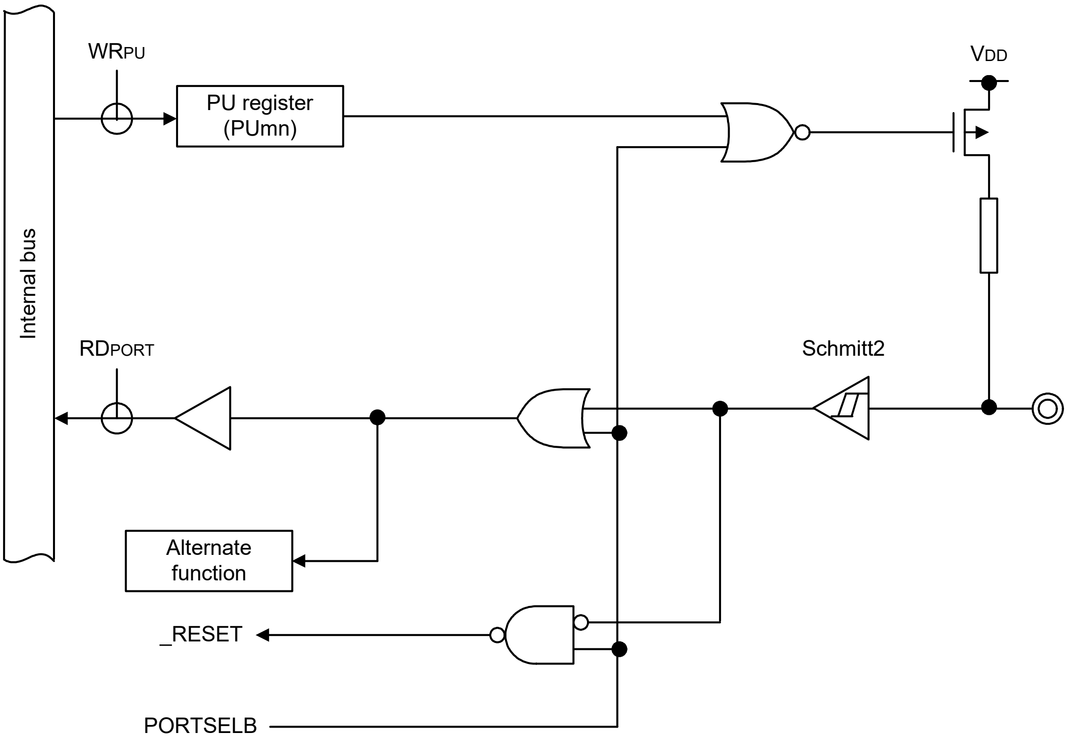


Figure 2 - 4 Pin Block Diagram of Pin Type 4-8-2

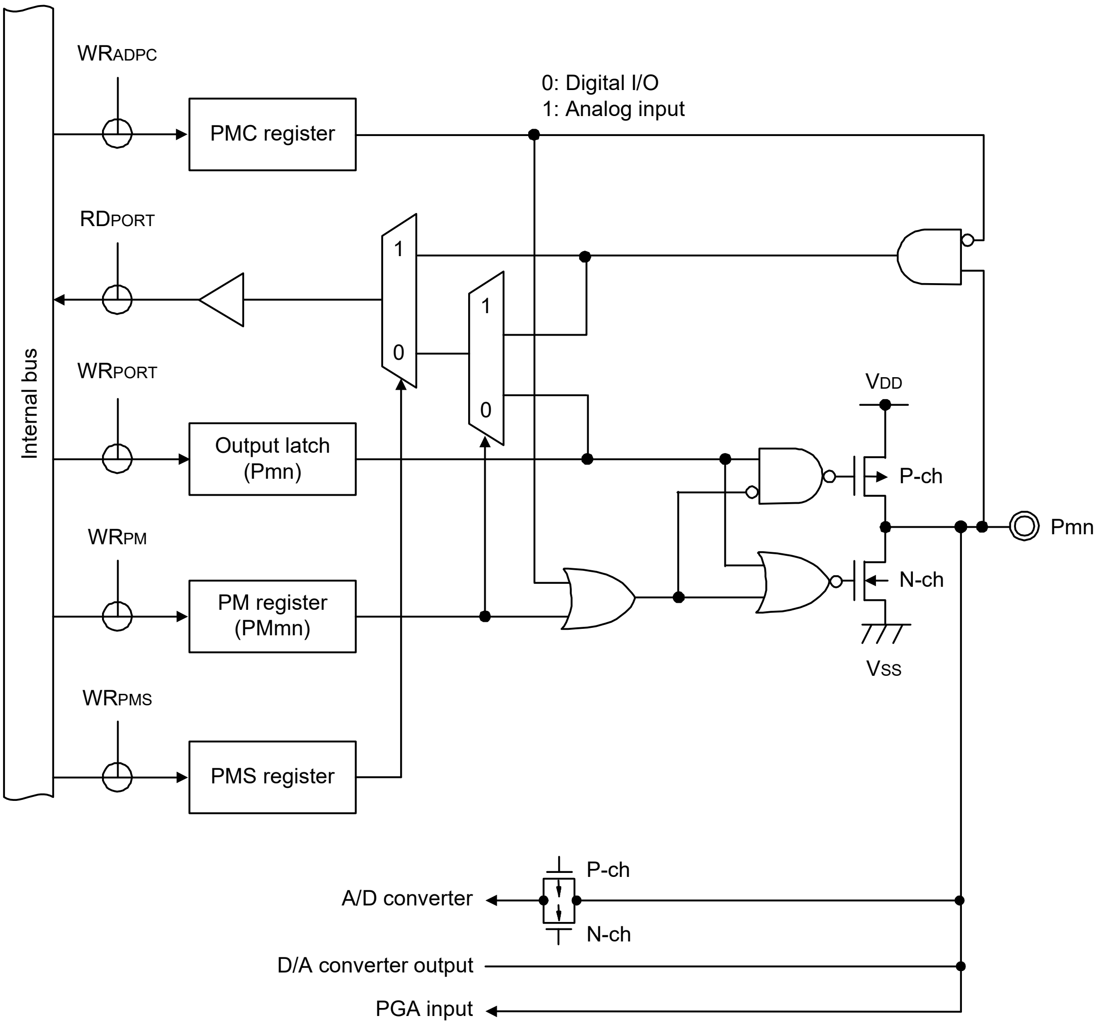
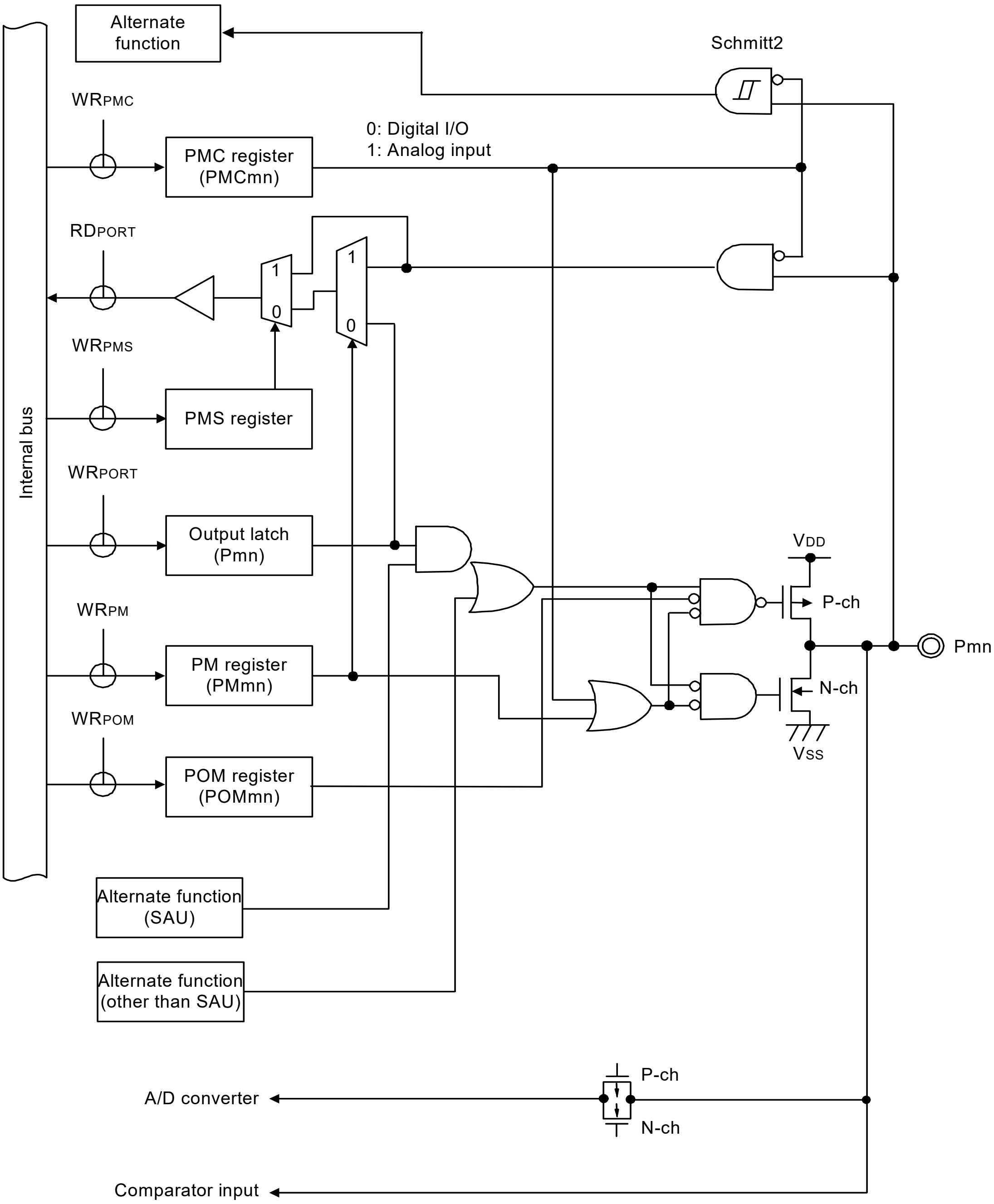


Figure 2 - 5 Pin Block Diagram of Pin Type 4-9-3



Caution **A through current may flow through if the pin is in the intermediate potential, because the input buffer is also turned on when the pin is in N-ch open-drain output mode by port output mode register (POMx).**

Figure 2 - 6 Pin Block Diagram of Pin Type 4-9-4

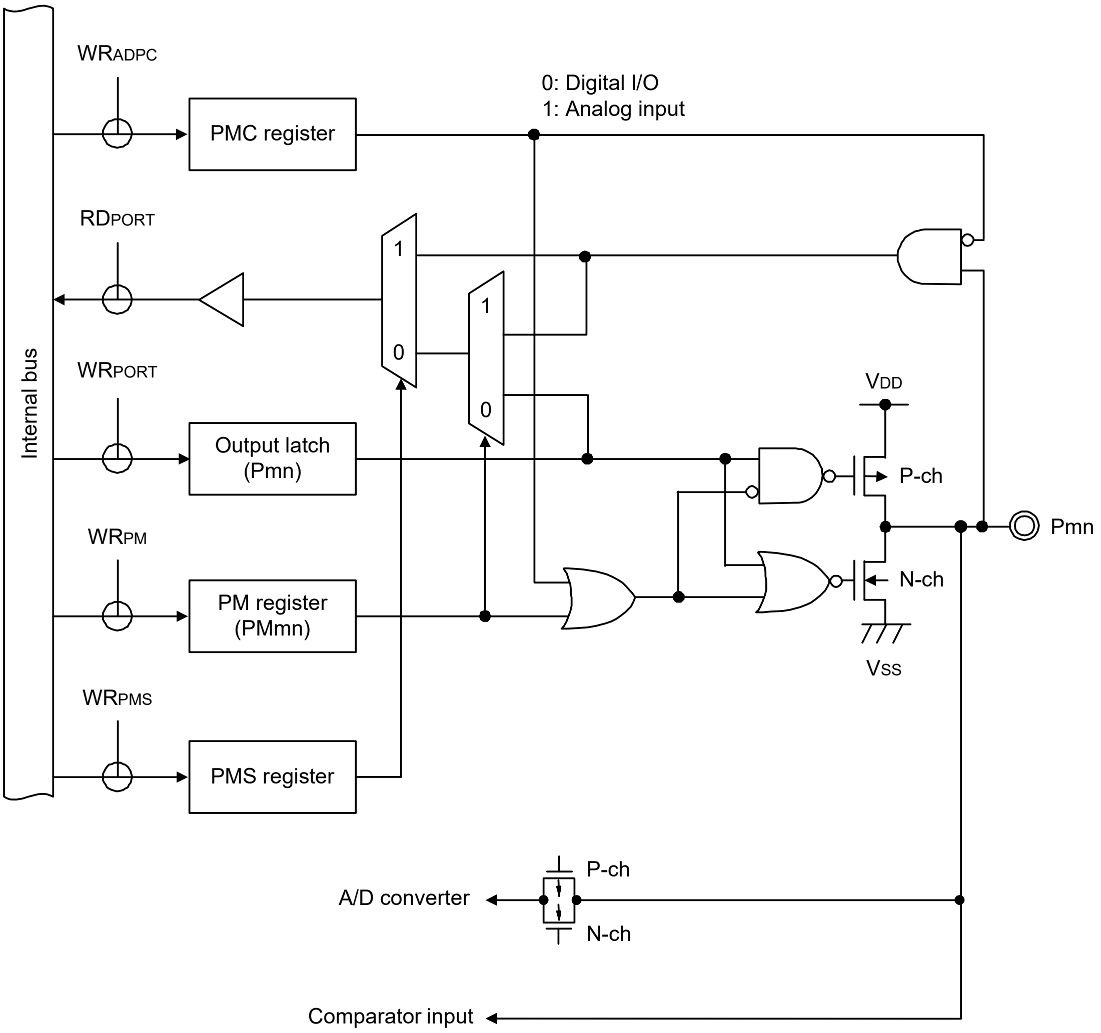


Figure 2 - 7 Pin Block Diagram of Pin Type 4-18-2

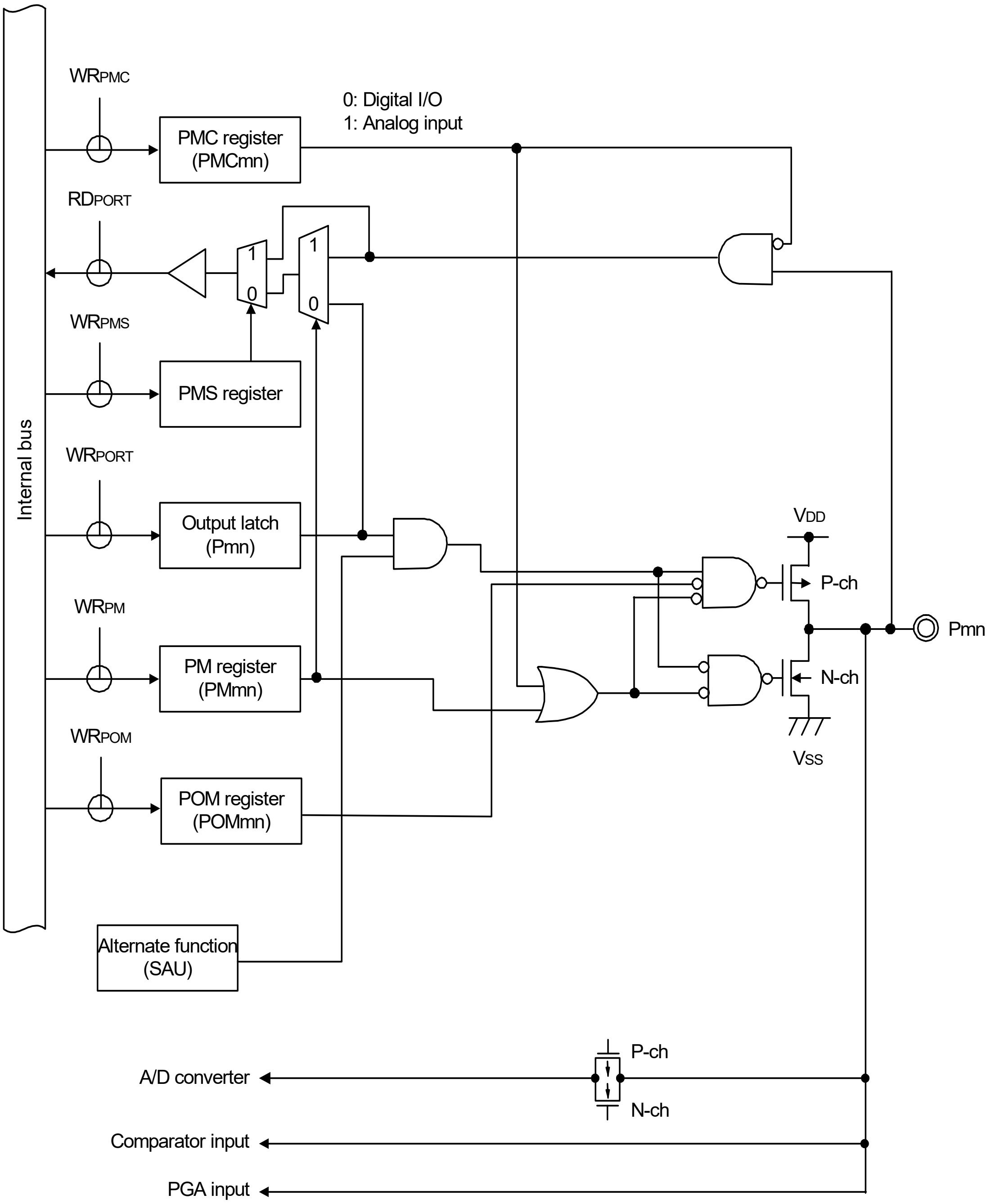
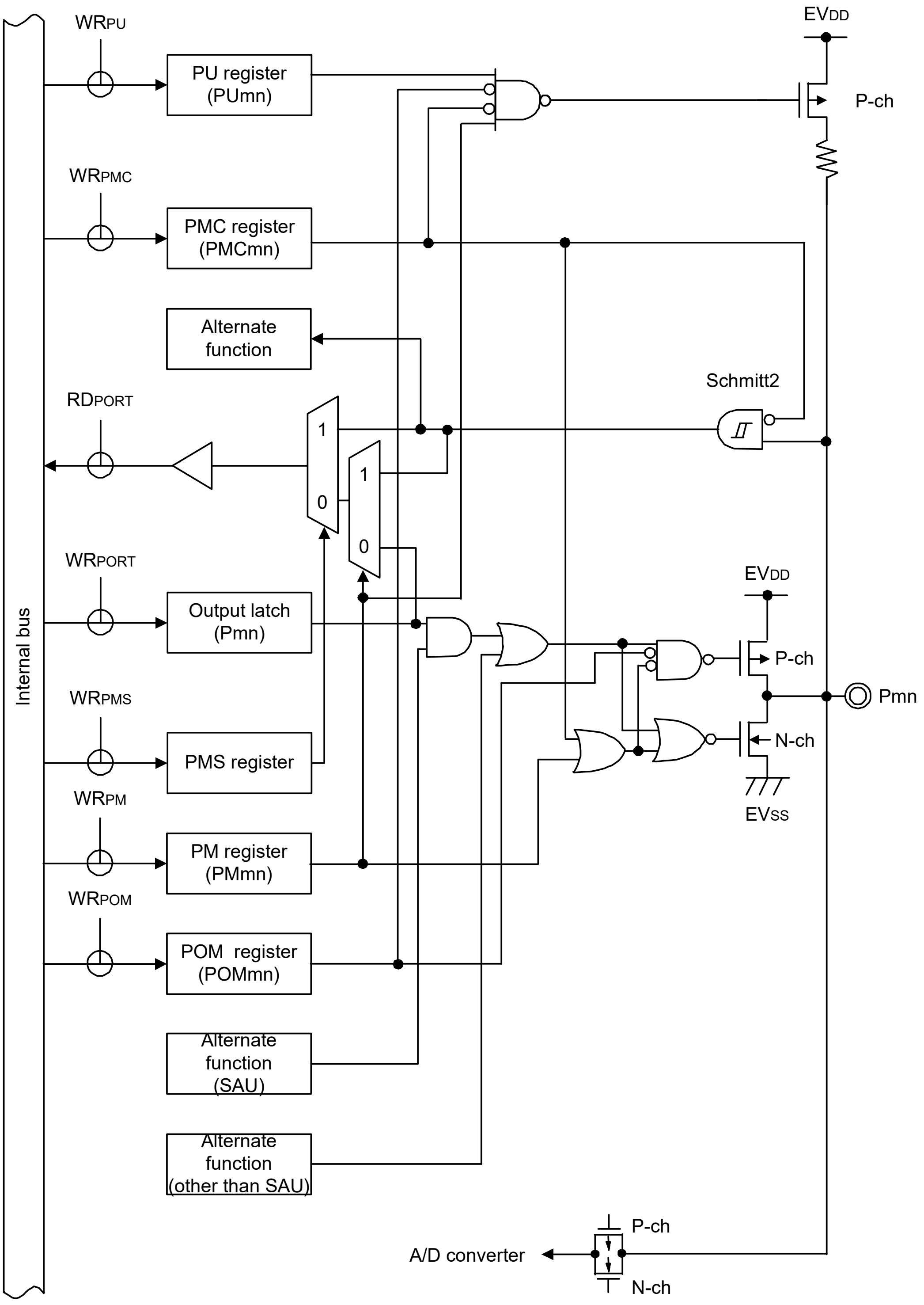


Figure 2 - 8 Pin Block Diagram of Pin Type 7-3-4

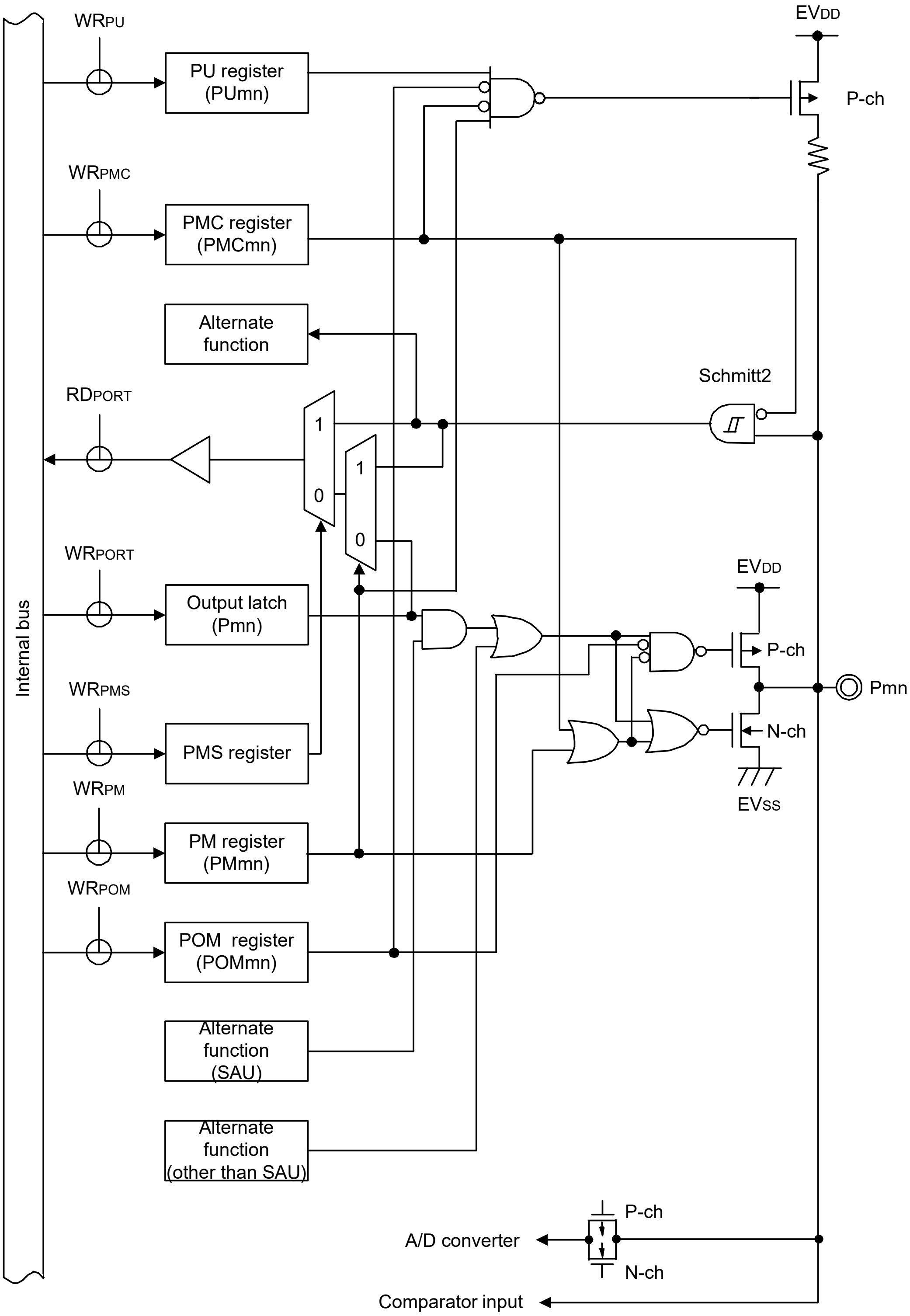


Caution **A through current may flow through if the pin is in the intermediate potential, because the input buffer is also turned on when the pin is in N-ch open-drain output mode by port output mode register (POMx).**

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 7-9-4

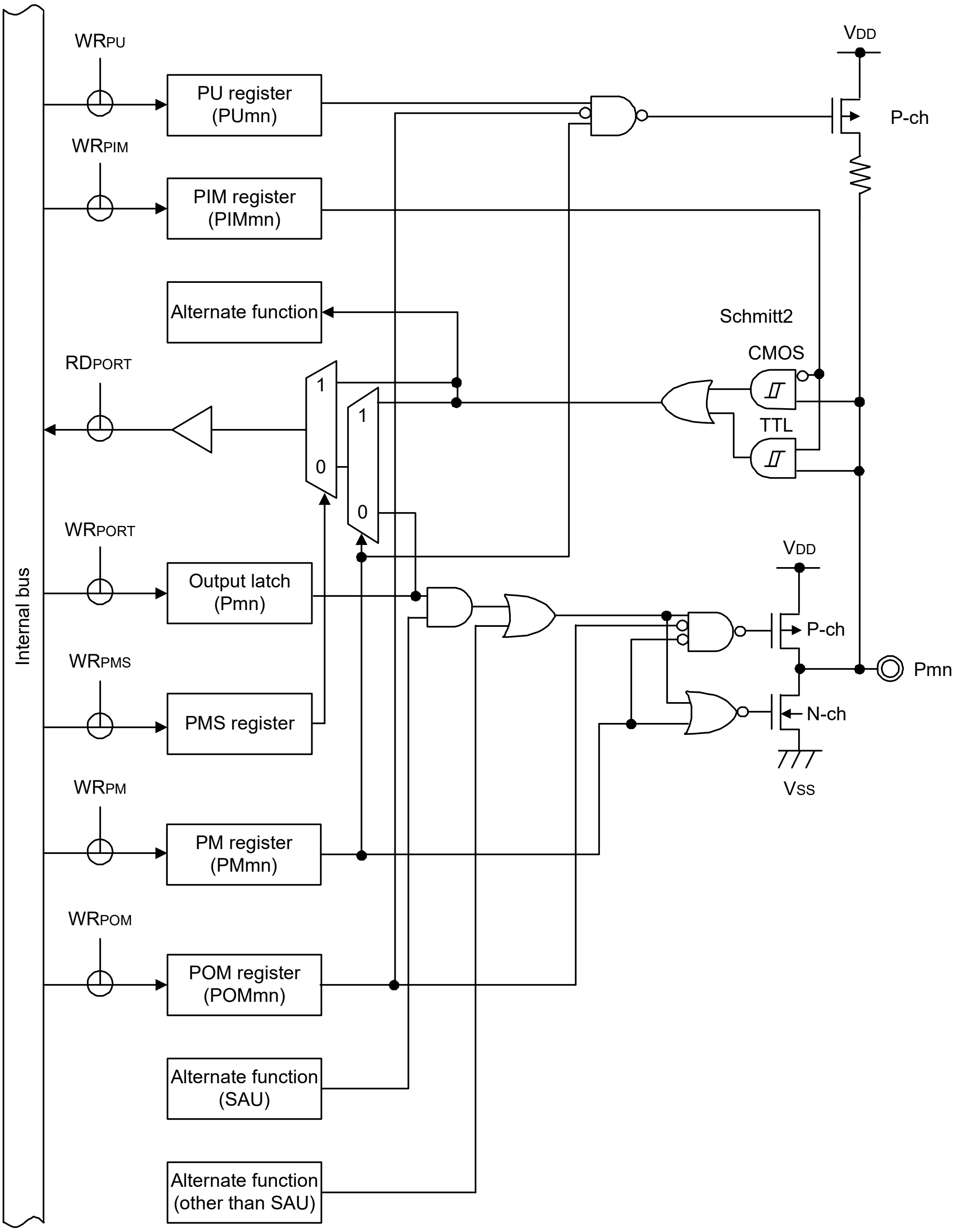


Caution **A through current may flow through if the pin is in the intermediate potential, because the input buffer is also turned on when the pin is in N-ch open-drain output mode by port output mode register (POMx).**

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 8-1-4



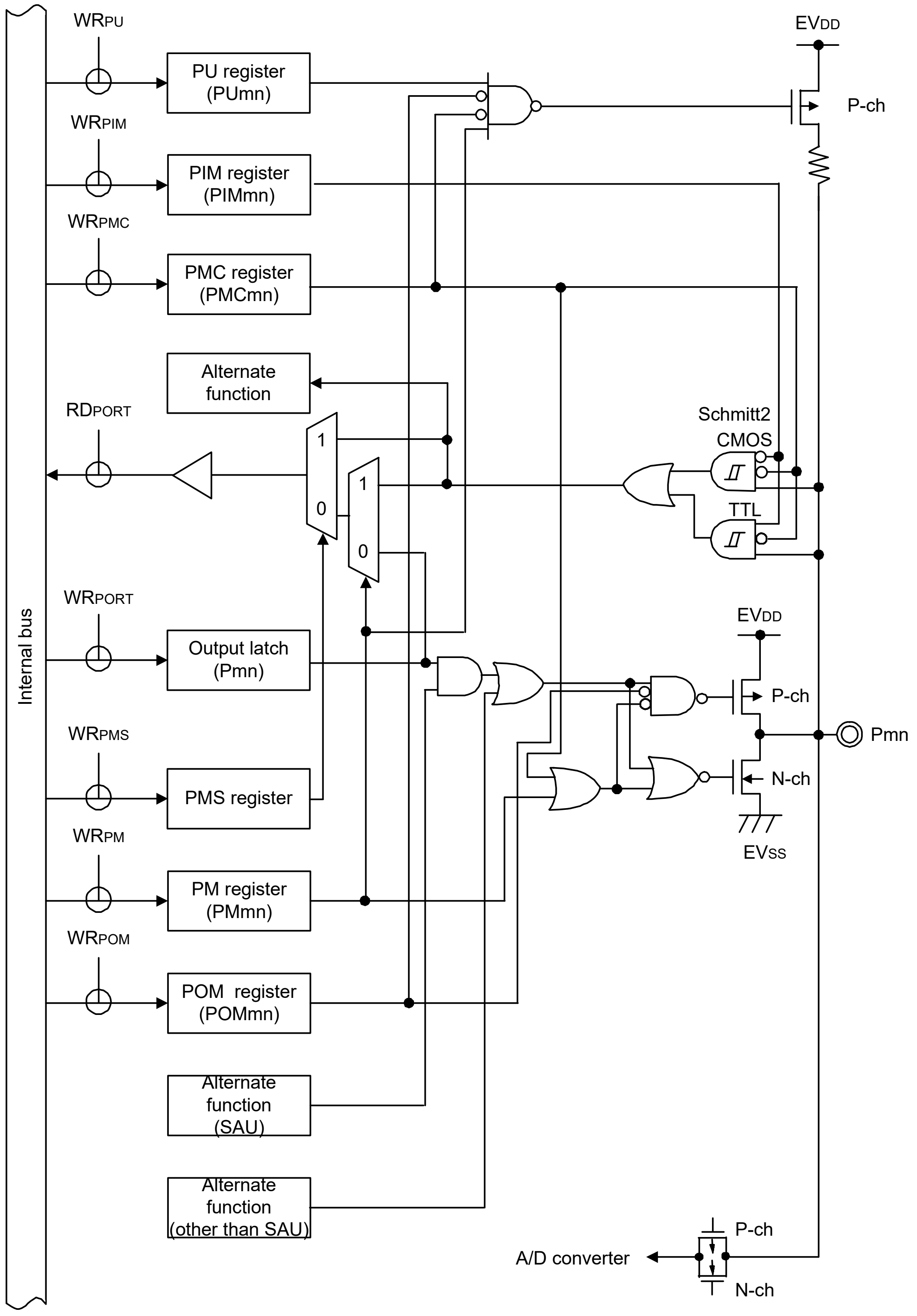
Caution 1. A through current may flow through if the pin is in the intermediate potential, because the input buffer is also turned on when the pin is in N-ch open-drain output mode by port output mode register (POMx).

Caution 2. Because of TTL input buffer structure, if the port input mode register (PIMx) is set in TTL input buffer, a through current may flow through in the case of high level input. It is recommended to input a low level to prevent a through current.

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 8-3-4



Caution 1. **A through current may flow through if the pin is in the intermediate potential, because the input buffer is also turned on when the pin is in N-ch open-drain output mode by port output mode register (POMx).**

Caution 2. **Because of TTL input buffer structure, if the port input mode register (PIMx) is set in TTL input buffer, a through current may flow through in the case of high level input. It is recommended to input a low level to prevent a through current.**

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

Remark 2. SAU: Serial array unit

CHAPTER 3 CPU ARCHITECTURE

The RL78/G11 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)

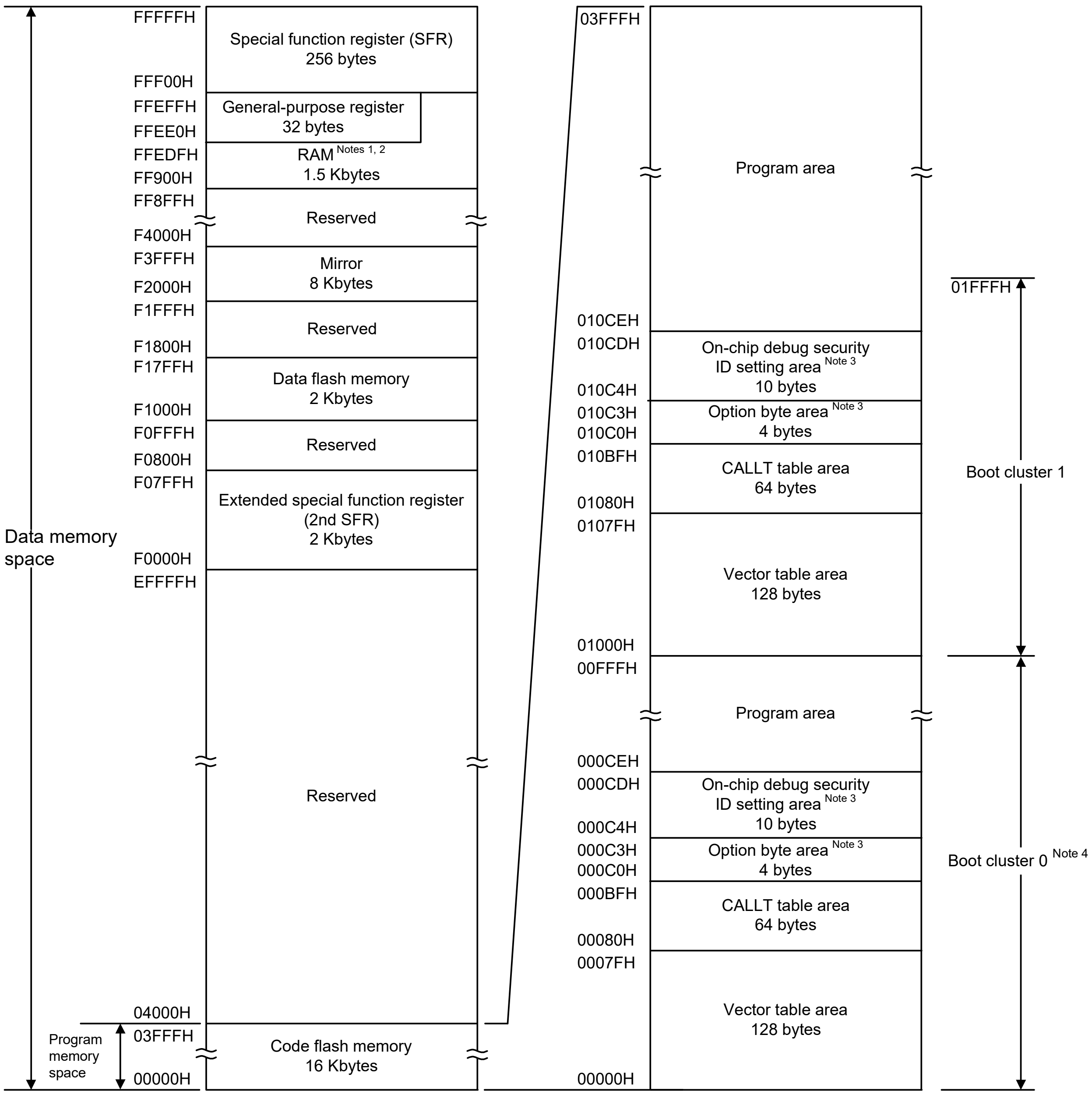
- Data allocation: Little endian

The RL78/G11 does not support OCD trace function.

3.1 Memory Space

Products in the RL78/G11 can access a 1 Mbytes address space. Figures 3 - 1 shows the memory map.

Figure 3 - 1 Memory Map



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 FF900H. 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.

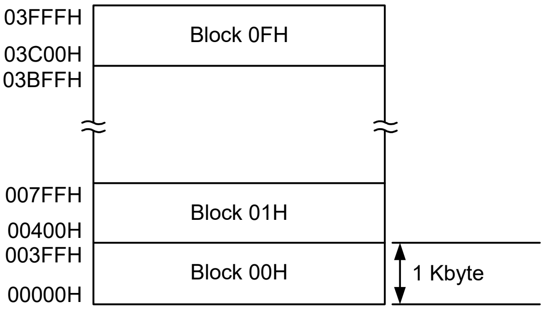
Note 3. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.

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 **31.7 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 28.3.3 RAM parity error detection function.

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



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
00000H to 003FFH	00H	02000H to 023FFH	08H
00400H to 007FFH	01H	02400H to 027FFH	09H
00800H to 00BFFH	02H	02800H to 02BFFH	0AH
00C00H to 00FFFH	03H	02C00H to 02FFFH	0BH
01000H to 013FFH	04H	03000H to 033FFH	0CH
01400H to 017FFH	05H	03400H to 037FFH	0DH
01800H to 01BFFH	06H	03800H to 03BFFH	0EH
01C00H to 01FFFH	07H	03C00H to 03FFFH	0FH

3.1.1 Internal program memory space

The internal program memory space stores the program and table data.
The RL78/G11 products incorporate internal ROM (flash memory) as shown below.

Table 3 - 2 Internal ROM Capacity

Internal ROM	
Structure	Capacity
Flash memory	16384 × 8 bits (00000H to 03FFFH)

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 (1/2)

Vector Table Address	Interrupt Source	25-pin	24-pin	20-pin	16-pin	10-pin
00000H	RESET, POR, LVD, WDT, TRAP, IAW, RPE	√	√	√	√	√
00004H	INTWDTI	√	√	√	√	√
	INTSRO	√	√	√	√	√
00006H	INTLVI	√	√	√	√	√
00008H	INTP0	√	√	√	√	—
0000AH	INTP1	√	√	√	√	√
0000CH	INTP2	√	√	√	√	—
0000EH	INTP3	√	√	√	√	—
00010H	INTP4	√	√	√	√	—
00012H	INTP5	√	√	√	—	—
00014H	INTP6	√	√	—	—	—
00016H	INTST0	√	√	√	√	—
	INTCSI00	√	√	√	—	—
	INTIIC00	√	√	√	—	—
00018H	INTSR0	√	√	√	√	—
	INTCSI01	√	√	—	—	—
	INTIIC01	√	√	—	—	—
0001EH	INTSRE0	√	√	√	√	—
00020H	INTTM00	√	√	√	√	√
00022H	INTST1	√	√	√	√	√
	INTCSI10	√	√	√	√	√
	INTIIC10	√	√	√	—	—
00024H	INTSR1	√	√	√	√	√
	INTCSI11	√	√	√	√	—
	INTIIC11	√	√	√	√	—
00026H	INTSRE1	√	√	√	√	√
00028H	INTIICA0	√	√	√	√	—
0002AH	INTTM01H	√	√	√	√	√
0002CH	INTTM03H	√	√	√	√	√
0002EH	INTTM01	√	√	√	√	√
00030H	INTTM02	√	√	√	√	√
00032H	INTTM03	√	√	√	√	√
00034H	INTAD	√	√	√	√	√
00036H	INTIT	√	√	√	√	√
00038H	INTKR	√	√	√	√	—
0003AH	INTP7	√	√	—	—	—
0003CH	INTP8	√	√	—	—	—
0003EH	INTP9	√	√	√	√	√
00040H	INTP10	√	√	√	√	√
00042H	INTP11	√	√	√	√	—

Table 3 - 3 Vector Table (2/2)

Vector Table Address	Interrupt Source	25-pin	24-pin	20-pin	16-pin	10-pin
00044H	INTCMP0	√	√	√	√	√
00046H	INTCMP1	√	√	√	√	—
00048H	INTDOC	√	√	√	√	√
0004AH	INTIT00	√	√	√	√	√
0004CH	INTIT01	√	√	√	√	√
0004EH	INTTMKB0	√	√	√	√	√
00050H	INTIICA1	√	√	√	—	—
00052H	INTFL	√	√	√	√	√
00054H	INTFO	√	√	√	√	√
0007EH	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 30 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 32 ON-CHIP DEBUG FUNCTION**.

3.1.2 Mirror area

The RL78/G11 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)).

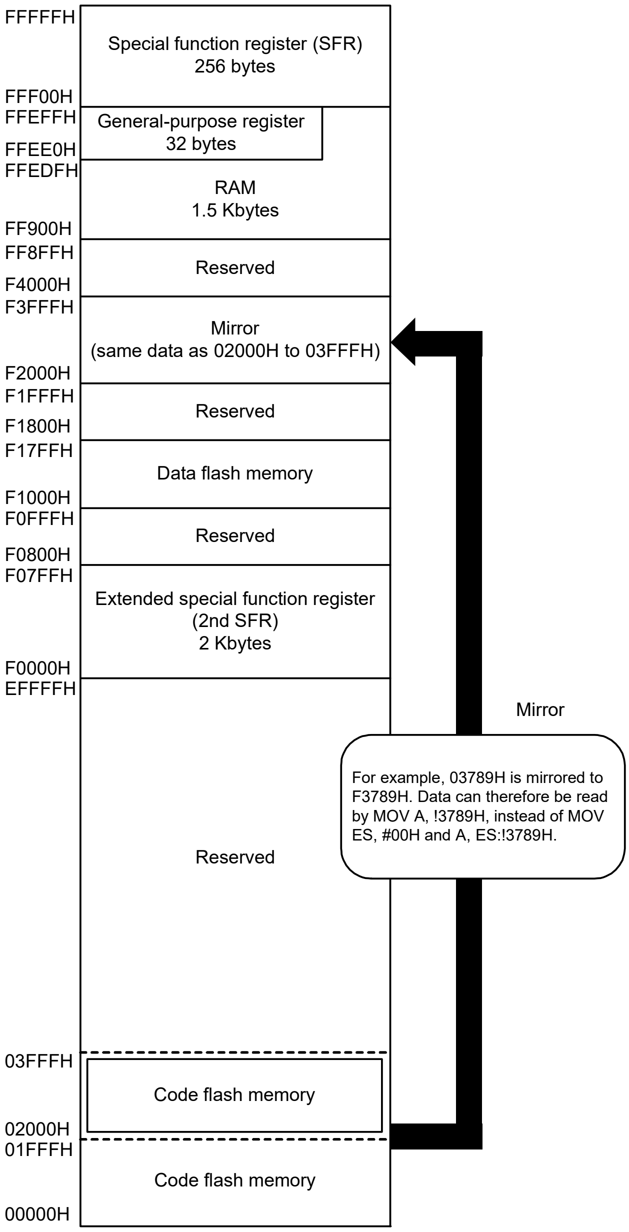
By reading data from F0000H to FFFFFH, an instruction that does not have the ES register as an operand can be used, and thus the contents of the code flash can be read with the shorter code. However, the code flash area is not mirrored to the special function register (SFR), extended special function 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



For example, 03789H is mirrored to F3789H. Data can therefore be read by MOV A, !3789H, instead of MOV ES, #00H and A, ES:!3789H.

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 - 2 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/G11 products incorporate the following RAMs.

Table 3 - 4 Internal RAM Capacity

Internal RAM
1536 × 8 bits (FF900H 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 a part of RAM area starting from FF900H when performing self-programming or rewriting of the data flash memory. For the RAM areas used by the flash library, see Self RAM list of Flash Self-Programming Library for RL78 Family (R20UT2944).

3.1.4 Special function register (SFR) area

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

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

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

On-chip peripheral hardware special function registers (2nd SFRs) are allocated in the area F0000H to F07FFH (see **Table 3 - 6** in **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**).

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

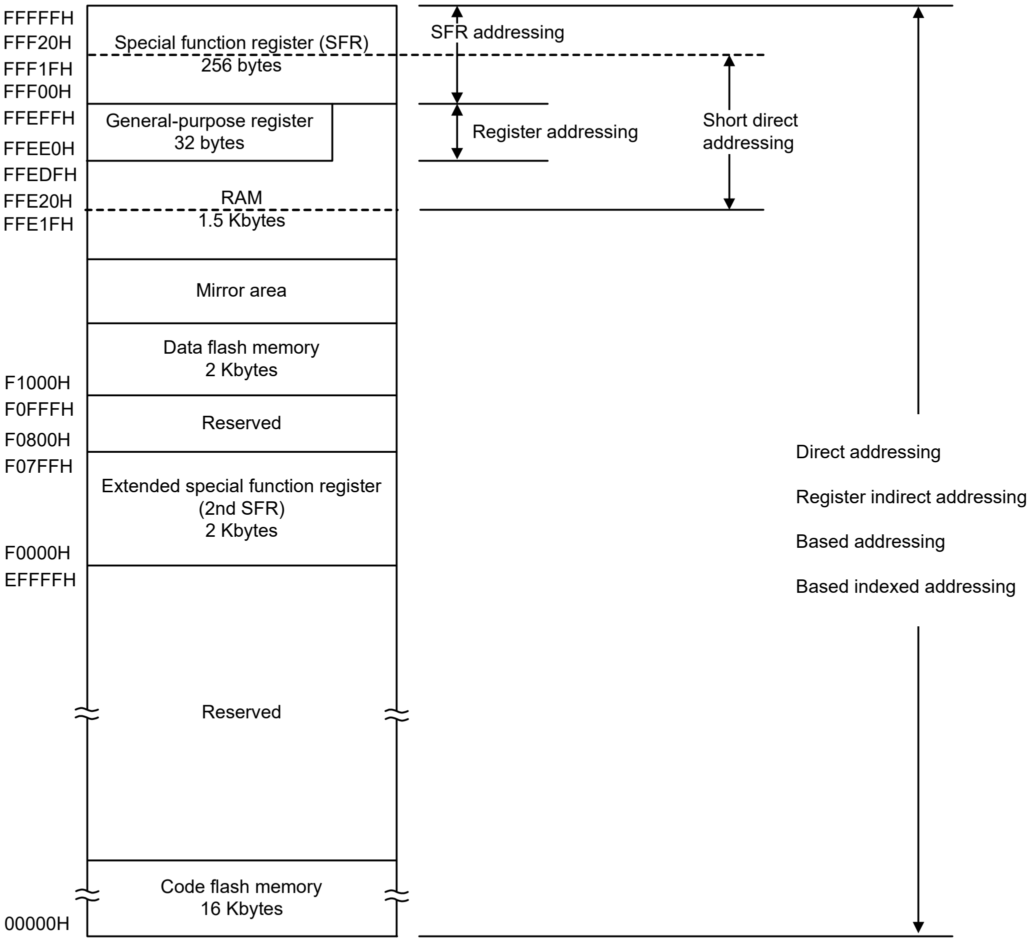
Caution 2. When accessing 8-bit interval timer counter register 0 (TRT0), 8-bit interval timer counter register 1 (TRT1), DOC control register (DOCR), DOC data input register (DODIR), and DOC data setting register (DODSR) allocated in F0500H to F0515H of the extended SFR (2nd SFR), the CPU does not proceed 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/G11, 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 - 3 shows correspondence between data memory and addressing. For details of each addressing, see **3.4 Addressing for Processing Data Addresses**.

Figure 3 - 3 Correspondence Between Data Memory and Addressing



3.2 Processor Registers

The RL78/G11 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).

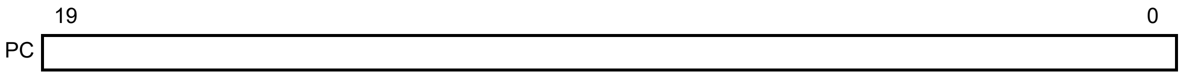
(1) Program counter (PC)

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 00000H and 00001H to the program counter.

Figure 3 - 4 Format of Program Counter



(2) Program status word (PSW)

The program status word is an 8-bit register consisting of various flags set/reset by instruction execution.

Program status word contents are stored in the stack area upon vectored interrupt request is acknowledged or PUSH PSW instruction execution and are restored upon execution of the RETB, RETI and POP PSW instructions. Reset signal generation sets the PSW register to 06H.

Figure 3 - 5 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 **22.3.3**) 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 - 6 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.

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).

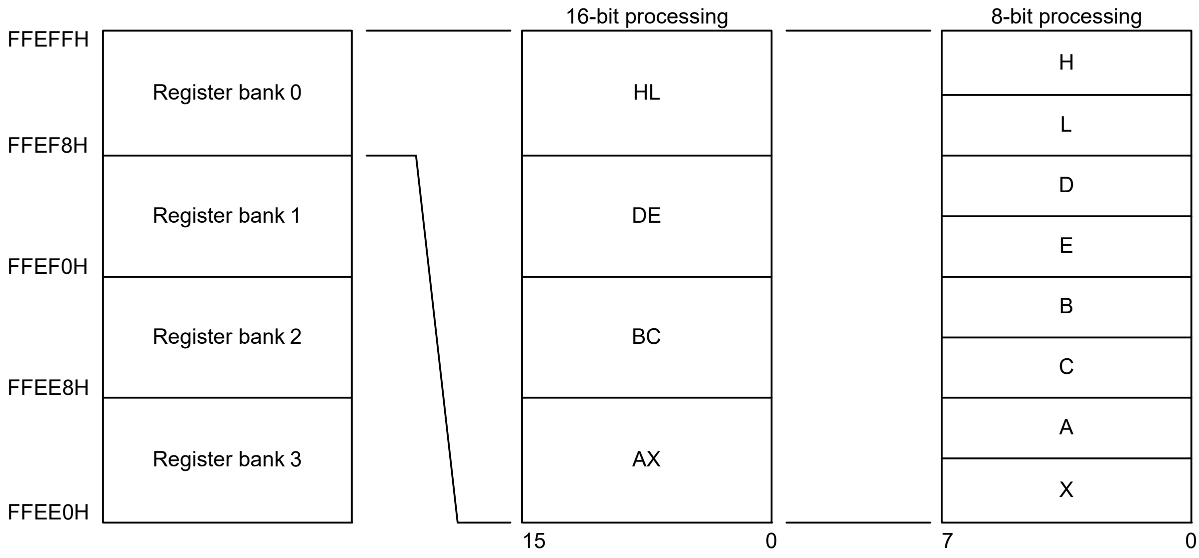
Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

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 - 7 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.

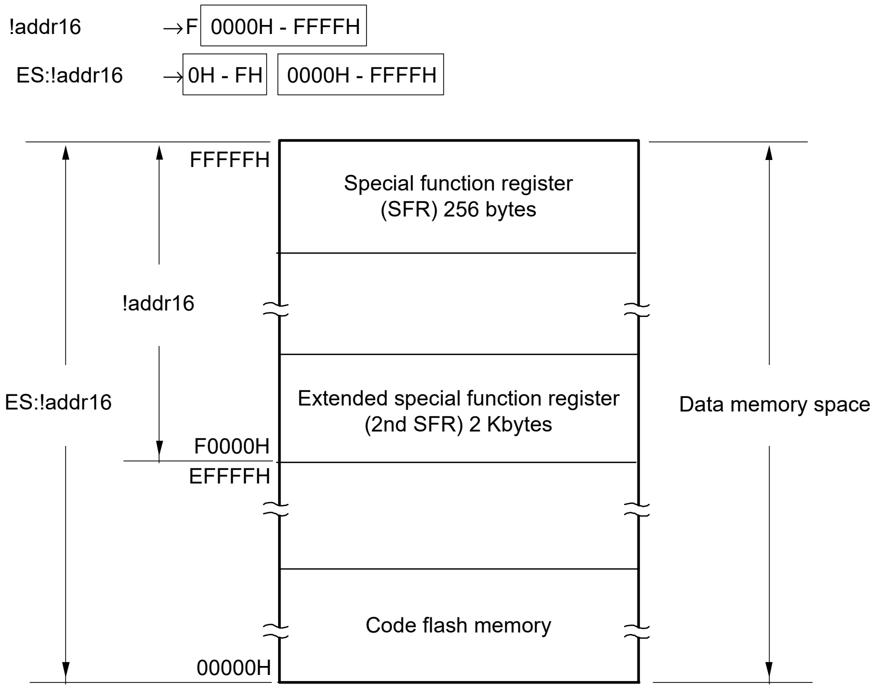
Figure 3 - 8 Configuration of ES and CS Registers

	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 - 9 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.

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

- Symbol
  This item indicates the address of a special function register. It is a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler. When using the assembler, debugger, and simulator, symbols can be written as an instruction operand.
- R/W
  This item indicates whether the corresponding SFR can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulable bit units
  "√" indicates the manipulable bit unit (1, 8, or 16). "—" indicates a bit unit for which manipulation is not possible.
- After reset
  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
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
FFF0CH	Port register 12	P12		R	√	√	—	Undefined
FFF0DH	Port register 13	P13		R	√	√	—	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	10-bit A/D conversion result register	ADCR		R	—	—	√	0000H
FFF1FH	8-bit A/D conversion result register	ADCRH		R	—	√	—	00H
FFF20H	Port mode register 0	PM0		R/W	√	√	—	FFH
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
FFF30H	A/D converter mode register 0	ADM0		R/W	√	√	—	00H
FFF31H	Analog input channel specification register	ADS		R/W	√	√	—	00H
FFF32H	A/D converter mode register 1	ADM1		R/W	√	√	—	00H
FFF34H	Key return control register	KRCTL		R/W	√	√	—	00H
FFF35H	Key return flag register	KRF		R/W	—	√	—	00H
FFF37H	Key return mode register 0	KRM0		R/W	√	√	—	00H
FFF38H	External interrupt rising edge enable register 0	EGP0		R/W	√	√	—	00H
FFF39H	External interrupt falling edge enable register 0	EGN0		R/W	√	√	—	00H
FFF3AH	External interrupt rising edge enable register 1	EGP1		R/W	√	√	—	00H
FFF3BH	External interrupt falling edge enable register 1	EGN1		R/W	√	√	—	00H
FFF3CH	D/A conversion value setting register 0	DACS0		R/W	—	√	—	00H

Table 3 - 5 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	
FFF3DH	D/A conversion value setting register 1	DACS1		R/W	—	√	—	00H
FFF3EH	D/A converter mode register	DAM		R/W	√	√	—	00H
FFF44H	Serial data register 02	TXD1/ SIO10	SDR02	R/W	—	√	√	0000H
FFF45H		—			—	—		
FFF46H	Serial data register 03	RXD1/ SIO11	SDR03	R/W	—	√	√	0000H
FFF47H		—			—	—		
FFF50H	IICA shift register 0	IICA0		R/W	—	√	—	00H
FFF51H	IICA status register 0	IICS0		R	√	√	—	00H
FFF52H	IICA flag register 0	IICF0		R/W	√	√	—	00H
FFF54H	IICA shift register 1	IICA1		R/W	—	√	—	00H
FFF55H	IICA status register 1	IICS1		R	√	√	—	00H
FFF56H	IICA flag register 1	IICF1		R/W	√	√	—	00H
FFF64H	Timer data register 02	TDR02		R/W	—	—	√	0000H
FFF65H								
FFF66H	Timer data register 03	TDR03L	TDR03	R/W	—	√	√	00H
FFF67H		TDR03H			—	√		00H
FFF90H	12-bit interval timer control register	ITMC		R/W	—	—	√	0FFFH
FFF91H								
FFFA0H	Clock operation mode control register	CMC		R/W	—	√	—	00H
FFFA1H	Clock operation status control register	CSC		R/W	√	√	—	C0H
FFFA2H	Oscillation stabilization time counter status register	OSTC		R	√	√	—	00H
FFFA3H	Oscillation stabilization time select register	OSTS		R/W	—	√	—	07H
FFFA4H	System clock control register	CKC		R/W	√	√	—	00H
FFFA5H	Clock output select register 0	CKS0		R/W	√	√	—	00H
FFFA6H	Clock output select register 1	CKS1		R/W	√	√	—	00H
FFFA7H	Subsystem clock select register	CKSEL		R/W	√	√	—	00H
FFFA8H	Reset control flag register	RESF		R	—	√	—	Undefined Note 1
FFFA9H	Voltage detection register	LVIM		R/W	√	√	—	00H Note 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	IF2	R/W	√	√	√	00H
FFFD1H	Interrupt request flag register 2H	IF2H		R/W	√	√		00H
FFFD4H	Interrupt mask flag register 2L	MK2L	MK2	R/W	√	√	√	FFH
FFFD5H	Interrupt mask flag register 2H	MK2H		R/W	√	√		FFH

Table 3 - 5 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	
FFFD8H	Priority specification flag register 02L	PR02L	PR02	R/W	√	√	√	FFH
FFFD9H	Priority specification flag register 02H	PR02H		R/W	√	√		FFH
FFFDCH	Priority specification flag register 12L	PR12L	PR12	R/W	√	√	√	FFH
FFFDDH	Priority specification flag register 12H	PR12H		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.

Reset Source Register		$\overline{\text{RESET}}$ 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
RESF	TRAP	Cleared (0)		Set (1)	Held			Held
	WDTRF			Held	Set (1)	Held		
	RPERF			Held		Set (1)	Held	
	IAWRF			Held			Set (1)	
	LVIRF			Held				Set (1)
LVIM	LVISEN	Cleared (0)						Held
	LVIOMSK	Held						
	LVIF							

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 **Table 3 - 6 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.

Table 3 - 6 gives 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 - 6 Extended Special Function Register (2nd SFR) List (1/7)

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
F003CH	Pull-up resistor option register 12	PU12	R/W	√	√	—	20H
F0040H	Port input mode register 0	PIM0	R/W	√	√	—	00H
F0043H	Port input mode register 3	PIM3	R/W	√	√	—	00H
F0044H	Port input mode register 4	PIM4	R/W	√	√	—	00H
F0045H	Port input mode register 5	PIM5	R/W	√	√	—	00H
F0050H	Port output mode register 0	POM0	R/W	√	√	—	00H
F0053H	Port output mode register 3	POM3	R/W	√	√	—	00H
F0054H	Port output mode register 4	POM4	R/W	√	√	—	00H
F0055H	Port output mode register 5	POM5	R/W	√	√	—	00H
F0060H	Port mode control register 0	PMC0	R/W	√	√	—	FFH
F0062H	Port mode control register 2	PMC2	R/W	√	√	—	FFH
F0063H	Port mode control register 3	PMC3	R/W	√	√	—	FFH
F0065H	Port mode control register 5	PMC5	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
F0075H	Peripheral I/O redirection register 2	PIOR2	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
F0079H	Peripheral I/O redirection register 1	PIOR1	R/W	—	√	—	00H
F007BH	Port mode select register	PMS	R/W	√	√	—	00H
F007CH	Peripheral I/O redirection register 3	PIOR3	R/W	—	√	—	00H
F007DH	Global digital input disable register	GDIDIS	R/W	√	√	—	00H

Table 3 - 6 Extended Special Function Register (2nd SFR) List (2/7)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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
F00F1H	Peripheral reset control register 0	PRR0		R/W	√	√	—	00H
F00F2H	Middle-speed on-chip oscillator frequency select register	MOCODIV		R/W	—	√	—	00H
F00F3H	Operation speed mode control register	OSMC		R/W	√	√	—	00H
F00F5H	RAM parity error control register	RPECTL		R/W	√	√	—	00H
F00F8H	Regulator mode control register	PMMC		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		—			—	—		
F0104H	Serial status register 02	SSR02L	SSR02	R	—	√	√	0000H
F0105H		—			—	—		
F0106H	Serial status register 03	SSR03L	SSR03	R	—	√	√	0000H
F0107H		—			—	—		
F0108H	Serial flag clear trigger register 00	SIR00L	SIR00	R/W	—	√	√	0000H
F0109H		—			—	—		
F010AH	Serial flag clear trigger register 01	SIR01L	SIR01	R/W	—	√	√	0000H
F010BH		—			—	—		
F010CH	Serial flag clear trigger register 02	SIR02L	SIR02	R/W	—	√	√	0000H
F010DH		—			—	—		
F010EH	Serial flag clear trigger register 03	SIR03L	SIR03	R/W	—	√	√	0000H
F010FH		—			—	—		
F0110H	Serial mode register 00	SMR00		R/W	—	—	√	0020H
F0111H								

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 - 6 Extended Special Function Register (2nd SFR) List (3/7)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0112H	Serial mode register 01	SMR01		R/W	—	—	√	0020H
F0113H								
F0114H	Serial mode register 02	SMR02		R/W	—	—	√	0020H
F0115H								
F0116H	Serial mode register 03	SMR03		R/W	—	—	√	0020H
F0117H								
F0118H	Serial communication operation setting register 00	SCR00		R/W	—	—	√	0087H
F0119H								
F011AH	Serial communication operation setting register 01	SCR01		R/W	—	—	√	0087H
F011BH								
F011CH	Serial communication operation setting register 02	SCR02		R/W	—	—	√	0087H
F011DH								
F011EH	Serial communication operation setting register 03	SCR03		R/W	—	—	√	0087H
F011FH								
F0120H	Serial channel enable status register 0	SE0L	SE0	R	√	√	√	0000H
F0121H		—			—	—		
F0122H	Serial channel start register 0	SS0L	SS0	R/W	√	√	√	0000H
F0123H		—			—	—		
F0124H	Serial channel stop register 0	ST0L	ST0	R/W	√	√	√	0000H
F0125H		—			—	—		
F0126H	Serial clock select register 0	SPS0L	SPS0	R/W	—	√	√	0000H
F0127H		—			—	—		
F0128H	Serial output register 0	SO0		R/W	—	—	√	0F0FH
F0129H								
F012AH	Serial output enable register 0	SOE0L	SOE0	R/W	√	√	√	0000H
F012BH		—			—	—		
F0134H	Serial output level register 0	SOL0L	SOL0	R/W	—	√	√	0000H
F0135H		—			—	—		
F0138H	Serial standby control register 0	SSC0L	SSC0	R/W	—	√	√	0000H
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								

Table 3 - 6 Extended Special Function Register (2nd SFR) List (4/7)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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		—			—	—		
F01BCH	Timer output level register 0	TOL0L	TOL0	R/W	—	√	√	0000H
F01BDH		—			—	—		
F01BEH	Timer output mode register 0	TOM0L	TOM0	R/W	—	√	√	0000H
F01BFH		—			—	—		
F0230H	IICA control register 00	IICCTL00		R/W	√	√	—	00H
F0231H	IICA control register 01	IICCTL01		R/W	√	√	—	00H
F0232H	IICA low-level width setting register 0	IICWL0		R/W	—	√	—	FFH
F0233H	IICA high-level width setting register 0	IICWH0		R/W	—	√	—	FFH
F0234H	Slave address register 0	SVA0		R/W	—	√	—	00H
F0238H	IICA control register 10	IICCTL10		R/W	√	√	—	00H
F0239H	IICA control register 11	IICCTL11		R/W	√	√	—	00H
F023AH	IICA low-level width setting register 1	IICWL1		R/W	—	√	—	FFH
F023BH	IICA high-level width setting register 1	IICWH1		R/W	—	√	—	FFH
F023CH	Slave address register 1	SVA1		R/W	—	√	—	00H
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

Table 3 - 6 Extended Special Function Register (2nd SFR) List (5/7)

<table>
<tr><th rowspan="2">Address</th><th rowspan="2">Extended Special Function Register
(2nd SFR) Name</th><th rowspan="2" colspan="2">Symbol</th><th rowspan="2">R/W</th><th colspan="3">Manipulable Bit Range</th><th rowspan="2">After Reset</th></tr>
<tr><th>1-bit</th><th>8-bit</th><th>16-bit</th></tr>
<tr><td>F0244H</td><td>Event output destination select register 04</td><td colspan="2">ELSELR04</td><td>R/W</td><td>—</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F0245H</td><td>Event output destination select register 05</td><td colspan="2">ELSELR05</td><td>R/W</td><td>—</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F0246H</td><td>Event output destination select register 06</td><td colspan="2">ELSELR06</td><td>R/W</td><td>—</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F0247H</td><td>Event output destination select register 07</td><td colspan="2">ELSELR07</td><td>R/W</td><td>—</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F0248H</td><td>Event output destination select register 08</td><td colspan="2">ELSELR08</td><td>R/W</td><td>—</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F0249H</td><td>Event output destination select register 09</td><td colspan="2">ELSELR09</td><td>R/W</td><td>—</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F024AH</td><td>Event output destination select register 10</td><td colspan="2">ELSELR10</td><td>R/W</td><td>—</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F024BH</td><td>Event output destination select register 11</td><td colspan="2">ELSELR11</td><td>R/W</td><td>—</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F024CH</td><td>Event output destination select register 12</td><td colspan="2">ELSELR12</td><td>R/W</td><td>—</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F024DH</td><td>Event output destination select register 13</td><td colspan="2">ELSELR13</td><td>R/W</td><td>—</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F024EH</td><td>Event output destination select register 14</td><td colspan="2">ELSELR14</td><td>R/W</td><td>—</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F024FH</td><td>Event output destination select register 15</td><td colspan="2">ELSELR15</td><td>R/W</td><td>—</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F0250H</td><td>Event output destination select register 16</td><td colspan="2">ELSELR16</td><td>R/W</td><td>—</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F0251H</td><td>Event output destination select register 17</td><td colspan="2">ELSELR17</td><td>R/W</td><td>—</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F02D4H</td><td>Timer clock select register 2</td><td colspan="2">TPS2</td><td>R/W</td><td>—</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F02D5H</td><td>Timer clock select register 3</td><td colspan="2">TPS3</td><td>R/W</td><td>—</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F02E0H</td><td>DTC base address register</td><td colspan="2">DTCBAR</td><td>R/W</td><td>—</td><td>√</td><td>—</td><td>FDH</td></tr>
<tr><td>F02E8H</td><td>DTC activation enable register 0</td><td colspan="2">DTCEN0</td><td>R/W</td><td>√</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F02E9H</td><td>DTC activation enable register 1</td><td colspan="2">DTCEN1</td><td>R/W</td><td>√</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F02EAH</td><td>DTC activation enable register 2</td><td colspan="2">DTCEN2</td><td>R/W</td><td>√</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F02F0H</td><td>Flash memory CRC control register</td><td colspan="2">CRC0CTL</td><td>R/W</td><td>√</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F02F2H</td><td>Flash memory CRC operation result register</td><td colspan="2">PGCRCL</td><td>R/W</td><td>—</td><td>—</td><td>√</td><td>0000H</td></tr>
<tr><td>F02FAH</td><td>CRC data register</td><td colspan="2">CRCD</td><td>R/W</td><td>—</td><td>—</td><td>√</td><td>0000H</td></tr>
<tr><td>F0340H</td><td>Comparator mode setting register</td><td colspan="2">COMPMDR</td><td>R/W</td><td>√</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F0341H</td><td>Comparator filter control register</td><td colspan="2">COMPFIR</td><td>R/W</td><td>√</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F0342H</td><td>Comparator output control register</td><td colspan="2">COMPOCR</td><td>R/W</td><td>√</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F0348H</td><td>Comparator input select register</td><td colspan="2">COMPISEL</td><td>R/W</td><td>√</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F0349H</td><td>PGA control register</td><td colspan="2">PGACTL</td><td>R/W</td><td>√</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F0350H</td><td>8-bit interval timer compare register 00</td><td>TRTCMP00</td><td rowspan="2">TRTCMP0</td><td>R/W</td><td>—</td><td>√</td><td rowspan="2">√</td><td>FFH</td></tr>
<tr><td>F0351H</td><td>8-bit interval timer compare register 01</td><td>TRTCMP01</td><td>R/W</td><td>—</td><td>√</td><td>FFH</td></tr>
<tr><td>F0352H</td><td>8-bit interval timer control register 0</td><td colspan="2">TRTCR0</td><td>R/W</td><td>√</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F0353H</td><td>8-bit interval timer division register 0</td><td colspan="2">TRTMD0</td><td>R/W</td><td>—</td><td>√</td><td>—</td><td>00H</td></tr>
<tr><td>F0400H</td><td rowspan="2">16-bit timer KB compare register 00</td><td rowspan="2" colspan="2">TKBCR0</td><td rowspan="2">R/W</td><td rowspan="2">—</td><td rowspan="2">—</td><td rowspan="2">√</td><td rowspan="2">0000H</td></tr>
<tr><td>F0401H</td></tr>
<tr><td>F0402H</td><td rowspan="2">16-bit timer KB compare register 01</td><td rowspan="2" colspan="2">TKBCR01</td><td rowspan="2">R/W</td><td rowspan="2">—</td><td rowspan="2">—</td><td rowspan="2">√</td><td rowspan="2">0000H</td></tr>
<tr><td>F0403H</td></tr>
<tr><td>F0404H</td><td rowspan="2">16-bit timer KB compare register 02</td><td rowspan="2" colspan="2">TKBCR02</td><td rowspan="2">R/W</td><td rowspan="2">—</td><td rowspan="2">—</td><td rowspan="2">√</td><td rowspan="2">0000H</td></tr>
<tr><td>F0405H</td></tr>
<tr><td>F0406H</td><td rowspan="2">16-bit timer KB compare register 03</td><td rowspan="2" colspan="2">TKBCR03</td><td rowspan="2">R/W</td><td rowspan="2">—</td><td rowspan="2">—</td><td rowspan="2">√</td><td rowspan="2">0000H</td></tr>
<tr><td>F0407H</td></tr>
<tr><td>F0408H</td><td rowspan="2">16-bit timer KB trigger compare register 0</td><td rowspan="2" colspan="2">TKBTGCR0</td><td rowspan="2">R/W</td><td rowspan="2">—</td><td rowspan="2">—</td><td rowspan="2">√</td><td rowspan="2">0000H</td></tr>
<tr><td>F0409H</td></tr>
</table>

Table 3 - 6 Extended Special Function Register (2nd SFR) List (6/7)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F040AH F040BH	16-bit timer KB smooth start initial duty register 00	TKBSIR00	R/W	—	—	√	0000H
F040CH F040DH	16-bit timer KB smooth start initial duty register 01	TKBSIR01	R/W	—	—	√	0000H
F040EH	16-bit timer KB dithering count register 00	TKBDNR00	R/W	—	√	—	00H
F040FH	16-bit timer KB smooth start step width register 00	TKBSSR00	R/W	—	√	—	00H
F0410H	16-bit timer KB dithering count register 01	TKBDNR01	R/W	—	√	—	00H
F0411H	16-bit timer KB smooth start step width register 01	TKBSSR01	R/W	—	√	—	00H
F0412H	16-bit timer KB trigger register 0	TKBTRG0	W	√	√	—	00H
F0413H	16-bit timer KB flag register 0	TKBFLG0	R	√	√	—	00H
F0414H F0415H	16-bit timer KB compare 1L & dithering count register 00	TKBCRLD00	R/W	—	—	√	0000H
F0416H F0417H	16-bit timer KB compare 1L & dithering count register 01	TKBCRLD01	R/W	—	—	√	0000H
F0420H F0421H	16-bit timer counter KB0	TKBCNT0	R	—	—	√	FFH
F0422H F0423H	16-bit timer KB operation control register 00	TKBCTL00	R/W	—	—	√	0000H
F0424H F0425H	16-bit timer KB maximum frequency limit setting register 0	TKBMFR0	R/W	—	—	√	0000H
F0426H	16-bit timer KB output control register 00	TKBIOC00	R/W	√	√	—	00H
F0427H	16-bit timer KB flag clear trigger register 0	TKBCLR0	W	√	√	—	00H
F0428H	16-bit timer KB output control register 01	TKBIOC01	R/W	√	√	—	00H
F0429H	16-bit timer KB operation control register 01	TKBCTL01	R/W	√	√	—	00H
F0430H F0431H	Forced output stop function control register 00	TKBPACTL00	R/W	—	—	√	0000H
F0432H F0433H	Forced output stop function control register 01	TKBPACTL01	R/W	—	—	√	0000H
F0434H	Forced output stop function start trigger register 0	TKBPAHFS0	W	√	√	—	00H
F0435H	Forced output stop function stop trigger register 0	TKBPAHFT0	W	√	√	—	00H
F0436H	Forced output stop function flag register 0	TKBPAFLG0	R	√	√	—	00H
F0437H	Forced output stop function control register 02	TKBPACTL02	R/W	√	√	—	00H
F0440H	Peripheral function switch register 0	PFSEL0	R/W	√	√	—	00H
F0441H	External interrupt edge enable register	INTPEG	R/W	√	√	—	00H
F0448H	Interrupt flag enable register	INTFE	R/W	√	√	—	00H
F0449H	Interrupt flag output control register 0	INTFOCTL0	R/W	√	√	—	00H
F044AH	Interrupt flag output control register 1	INTFOCTL1	R/W	√	√	—	00H

Table 3 - 6 Extended Special Function Register (2nd SFR) List (7/7)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0500H	8-bit interval timer counter register 00	TRT00	TRT0	R	—	√	√	00H
F0501H	8-bit interval timer counter register 01	TRT01		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 Table 3 - 5 Special Function Register (SFR) List.

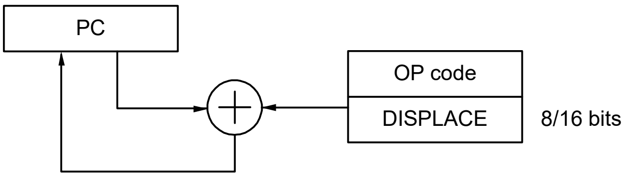
3.3 Instruction Address Addressing

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 - 10 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 - 11 Example of CALL !!addr20/BR !!addr20

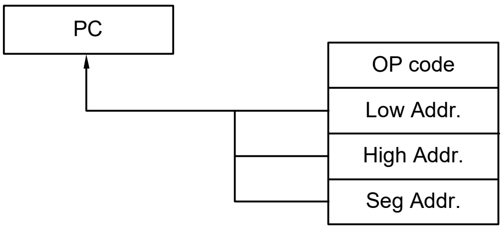
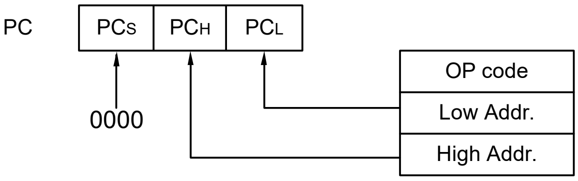


Figure 3 - 12 Example of CALL !addr16/BR !addr16



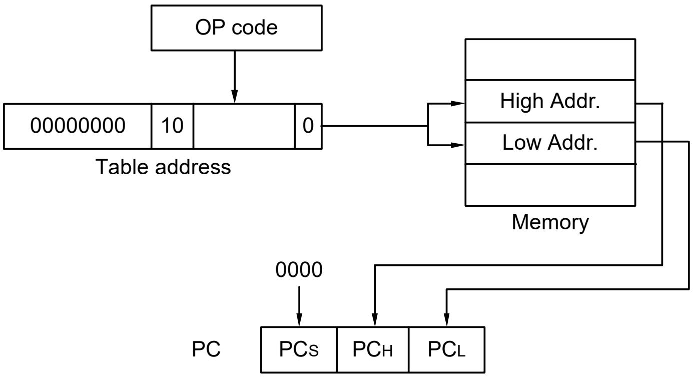
3.3.3 Table indirect addressing

[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 - 13 Outline of Table Indirect Addressing

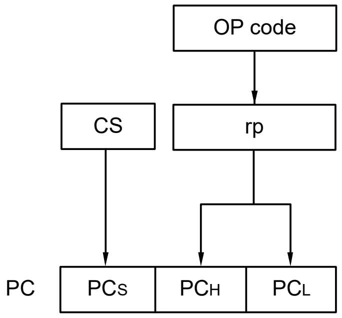


3.3.4 Register indirect addressing

[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 - 14 Outline of Register Indirect Addressing



3.4 Addressing for Processing Data Addresses

3.4.1 Implied addressing

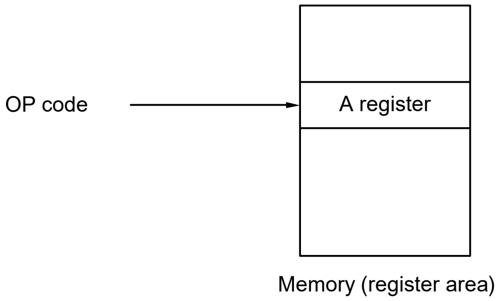
[Function]

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 - 15 Outline of Implied Addressing



3.4.2 Register addressing

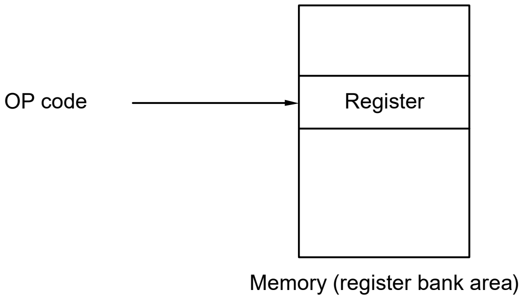
[Function]

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 - 16 Outline of Register Addressing



3.4.3 Direct addressing

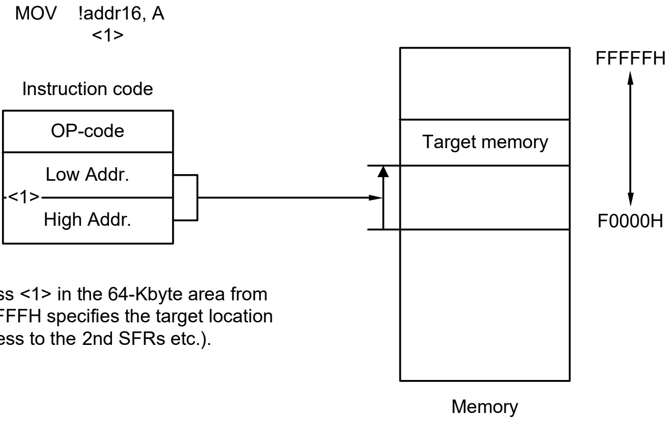
[Function]

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

[Operand format]

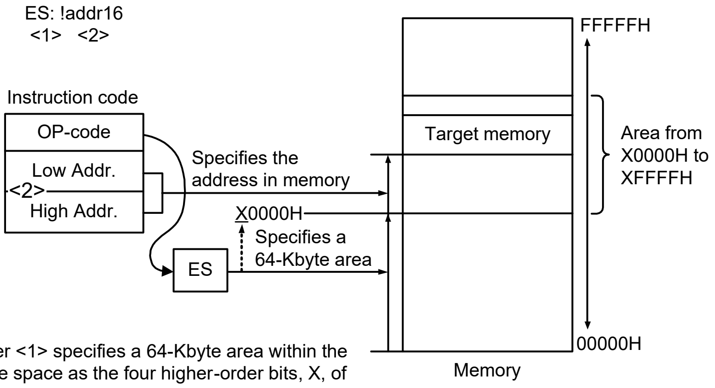
Identifier	Description
!addr16	Label or 16-bit immediate data (only the space from F0000H to FFFFFH is specifiable)
ES:!addr16	Label or 16-bit immediate data (higher 4-bit addresses are specified by the ES register)

Figure 3 - 17 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 - 18 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

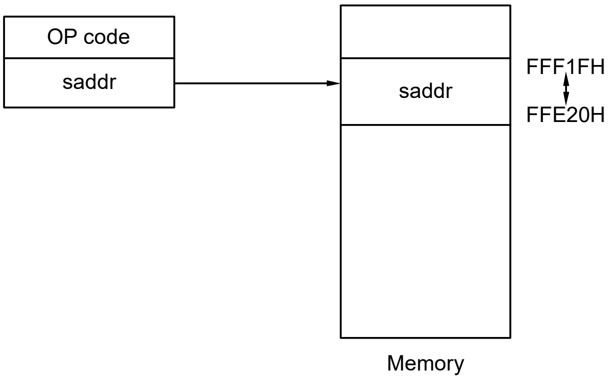
[Function]

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 - 19 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

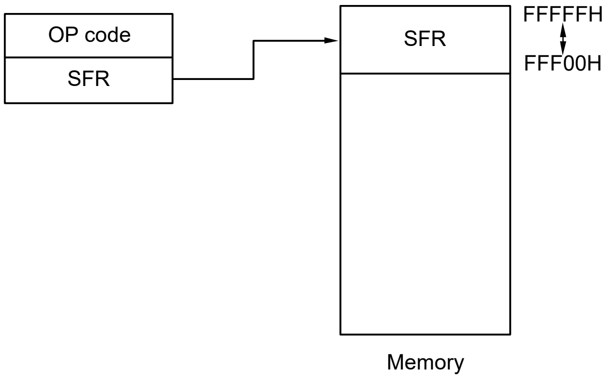
[Function]

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 - 20 Outline of SFR Addressing



3.4.6 Register indirect addressing

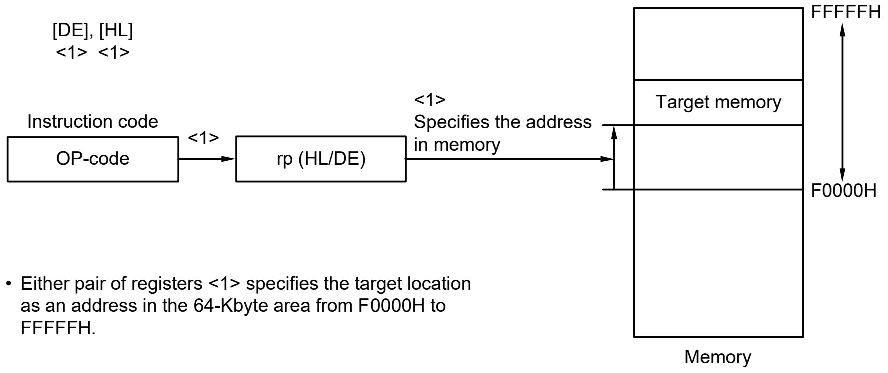
[Function]

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]

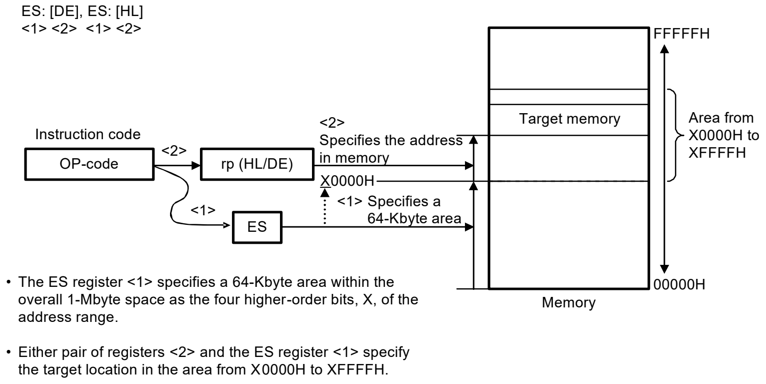
Identifier	Description
—	[DE], [HL] (only the space from F0000H to FFFFFH is specifiable)
—	ES:[DE], ES:[HL] (higher 4-bit addresses are specified by the ES register)

Figure 3 - 21 Example of [DE], [HL]



- Either pair of registers <1> specifies the target location as an address in the 64-Kbyte area from F0000H to FFFFFH.

Figure 3 - 22 Example of ES:[DE], ES:[HL]



- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- Either pair of registers <2> and the ES register <1> specify the target location in the area from X0000H to XFFFFH.

3.4.7 Based addressing

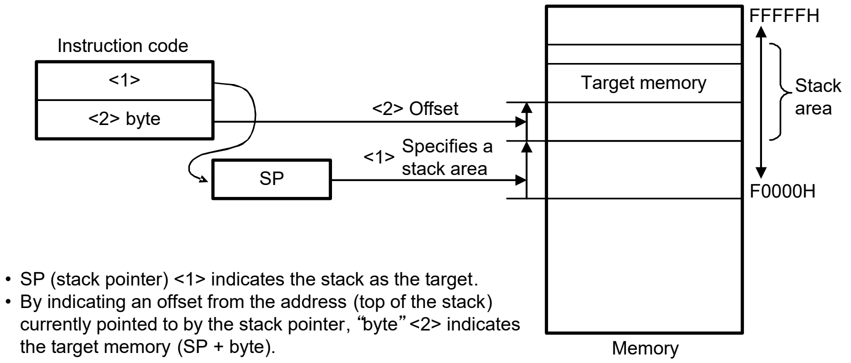
[Function]

Based addressing uses the contents of a register pair specified with the instruction word or 16-bit immediate data as a base address, and 8-bit immediate data or 16-bit immediate data as offset data. The sum of these values is used to specify the target address.

[Operand format]

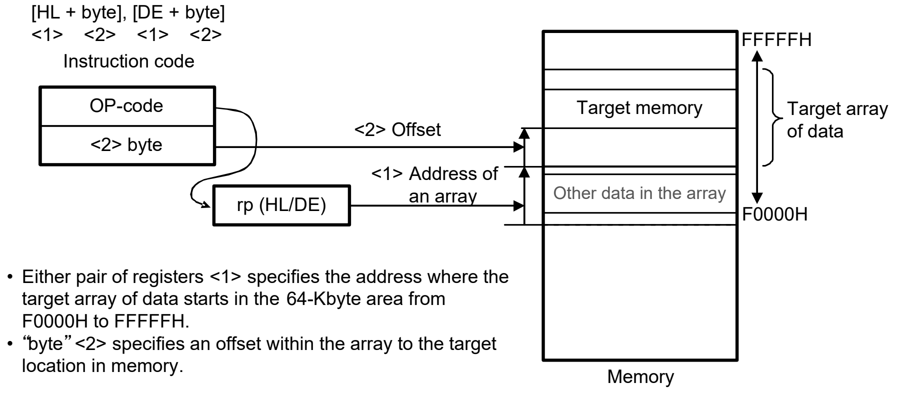
Identifier	Description
—	[HL + byte], [DE + byte], [SP + byte] (only the space from F0000H to FFFFFH is specifiable)
—	word[B], word[C] (only the space from F0000H to FFFFFH is specifiable)
—	word[BC] (only the space from F0000H to FFFFFH is specifiable)
—	ES:[HL + byte], ES:[DE + byte] (higher 4-bit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

Figure 3 - 23 Example of [SP+byte]



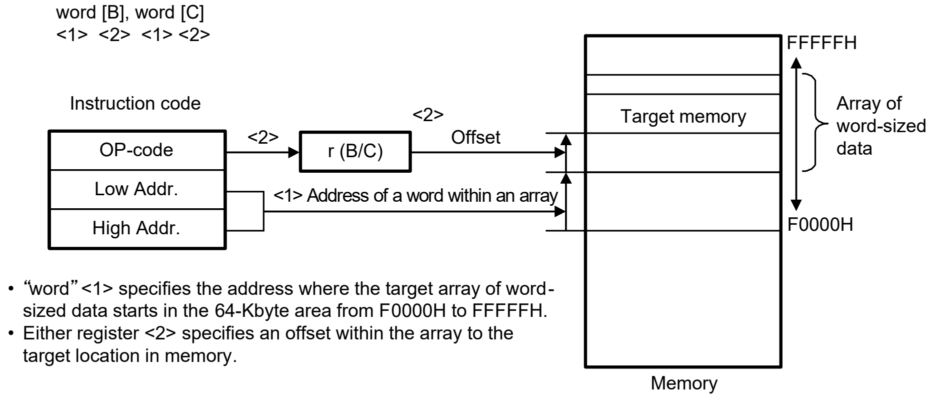
- SP (stack pointer) <1> indicates the stack as the target.
- By indicating an offset from the address (top of the stack) currently pointed to by the stack pointer, "byte" <2> indicates the target memory (SP + byte).

Figure 3 - 24 Example of [HL + byte], [DE + byte]]



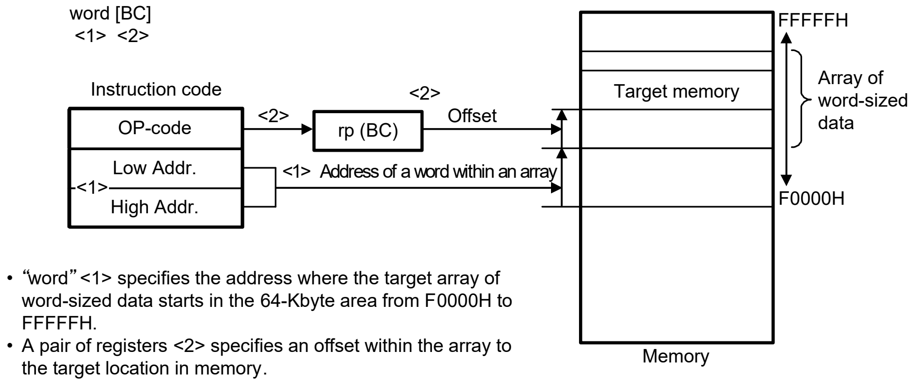
- Either pair of registers <1> specifies the address where the target array of data starts in the 64-Kbyte area from F0000H to FFFFFH.
- "byte" <2> specifies an offset within the array to the target location in memory.

Figure 3 - 25 Example of word [B], word [C]



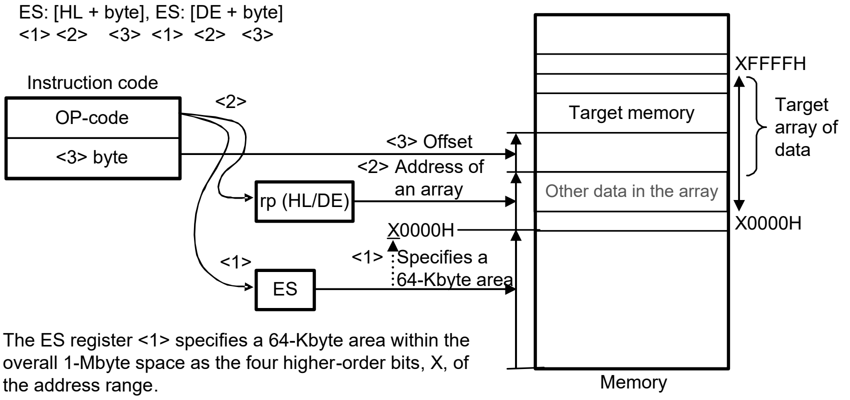
- "word" <1> specifies the address where the target array of word-sized data starts in the 64-Kbyte area from F0000H to FFFFFH.
- Either register <2> specifies an offset within the array to the target location in memory.

Figure 3 - 26 Example of word [BC]



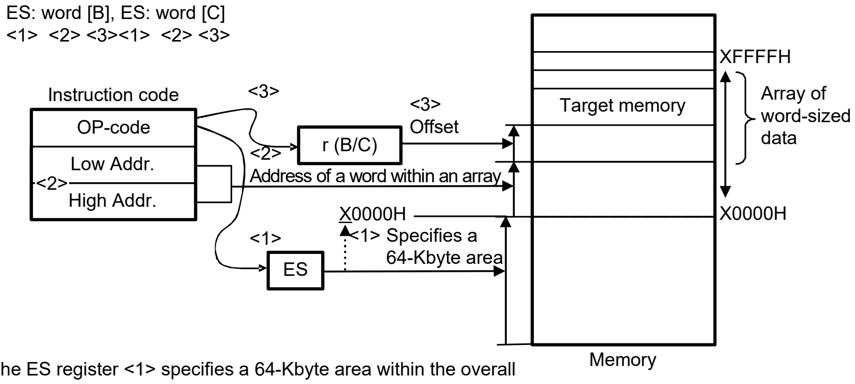
- "word" <1> specifies the address where the target array of word-sized data starts in the 64-Kbyte area from F0000H to FFFFFH.
- A pair of registers <2> specifies an offset within the array to the target location in memory.

Figure 3 - 27 Example of [HL + byte], [DE + byte]]



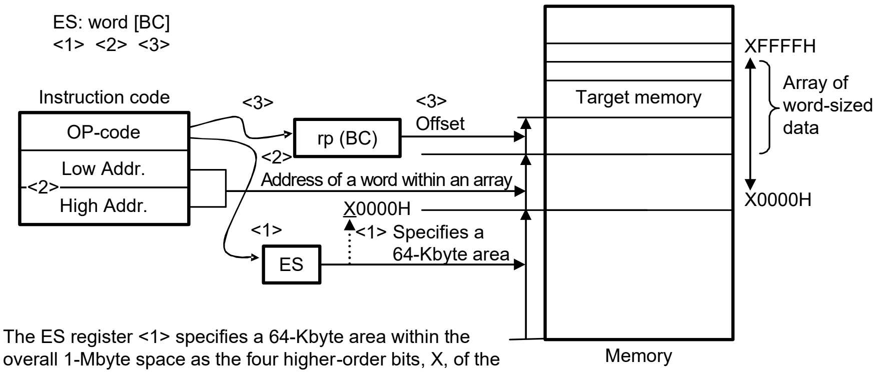
- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- Either pair of registers <2> specifies the address where the target array of data starts in the 64-Kbyte area specified in the ES register <1>.
- "byte" <3> specifies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

Figure 3 - 28 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 the 64-Kbyte area specified in the ES register <1>.
- Either register <3> specifies an offset within the array to the target location in memory.

Figure 3 - 29 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

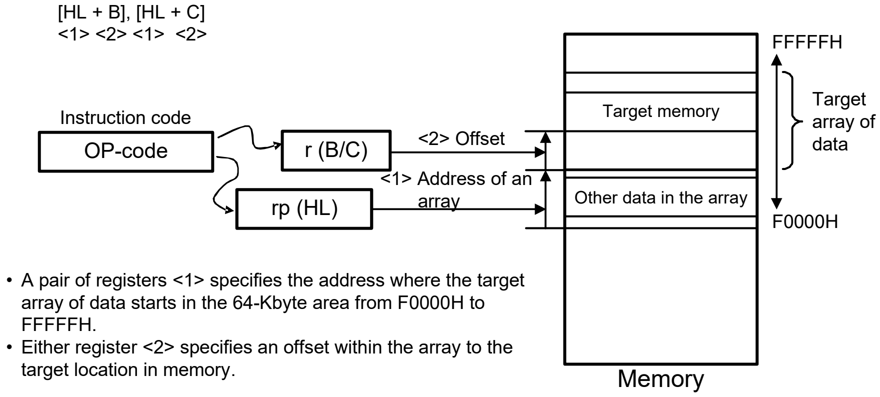
[Function]

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]

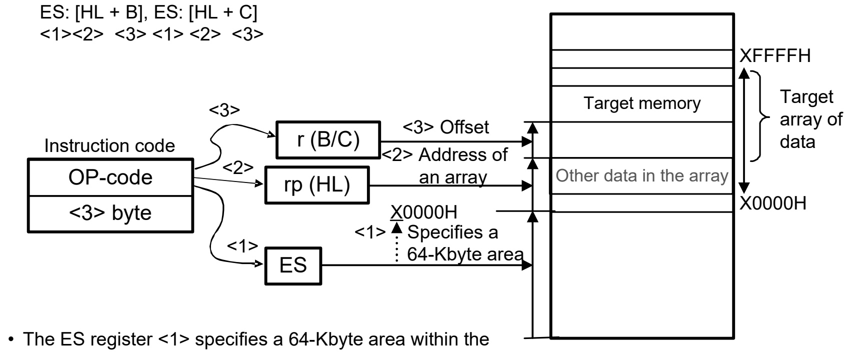
Identifier	Description
—	[HL+B], [HL+C] (only the space from F0000H to FFFFFH is specifiable)
—	ES:[HL+B], ES:[HL+C] (higher 4-bit addresses are specified by the ES register)

Figure 3 - 30 Example of [HL + B], [HL + C]



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

Figure 3 - 31 Example of ES:[HL + B], ES:[HL + C]



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

3.4.9 Stack addressing

[Function]

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

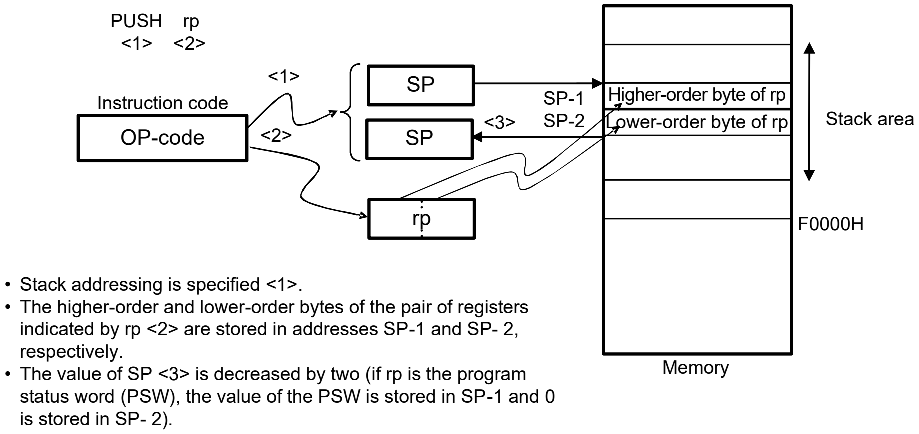
Only the internal RAM area can be set as the stack area.

[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

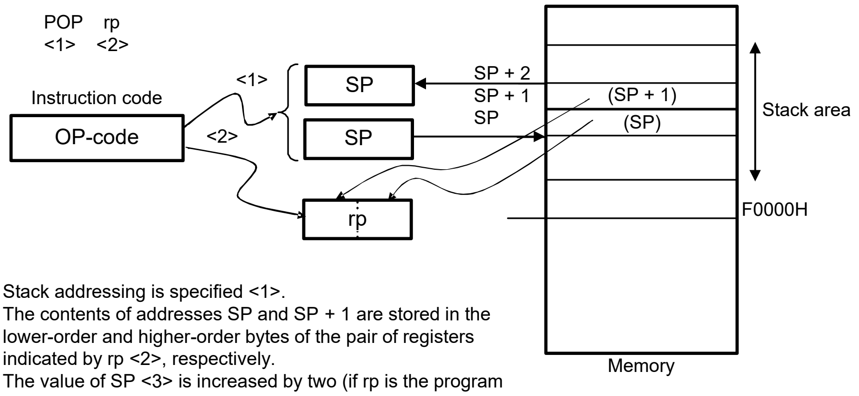
The data to be saved/restored by each stack operation is shown in Figures 3 - 32 to 3 - 37.

Figure 3 - 32 Example of PUSH rp



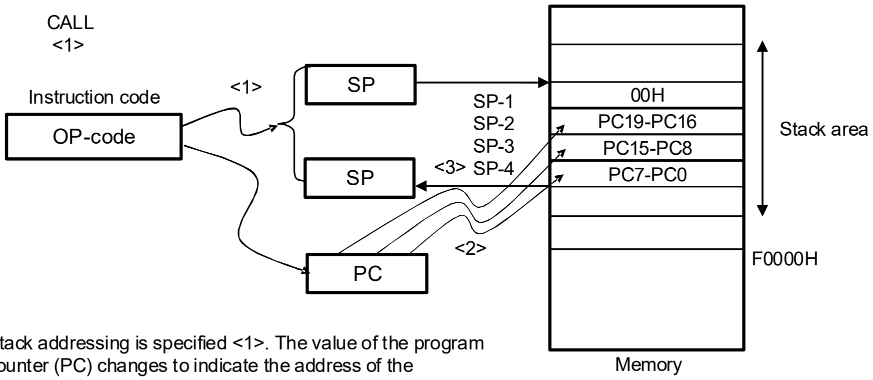
- Stack addressing is specified <1>.
- The higher-order and lower-order bytes of the pair of registers indicated by rp <2> are stored in addresses SP-1 and SP- 2, respectively.
- The value of SP <3> is decreased by two (if rp is the program status word (PSW), the value of the PSW is stored in SP-1 and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

Figure 3 - 33 Example of POP



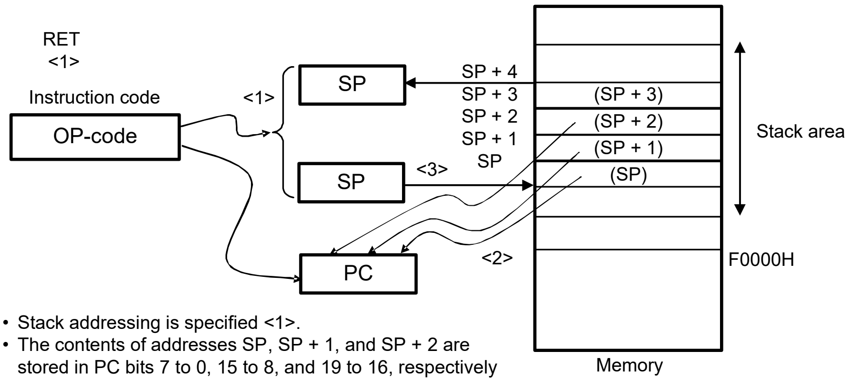
- Stack addressing is specified <1>.
- The contents of addresses SP and SP + 1 are stored in the lower-order and higher-order bytes of the pair of registers indicated by rp <2>, respectively.
- The value of SP <3> is increased by two (if rp is the program status word (PSW), the content of address SP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

Figure 3 - 34 Example of CALL, CALLT



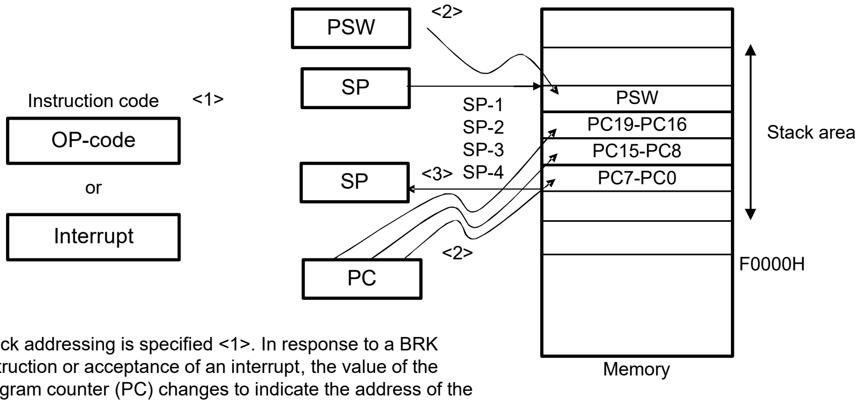
- Stack addressing is specified <1>. The value of the program counter (PC) changes to indicate the address of the instruction following the CALL instruction.
- 00H, the values of PC bits 19 to 16, 15 to 8, and 7 to 0 are stored in addresses SP-1, SP-2, SP-3, and SP-4, respectively <2>.
- The value of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

Figure 3 - 35 Example of RET



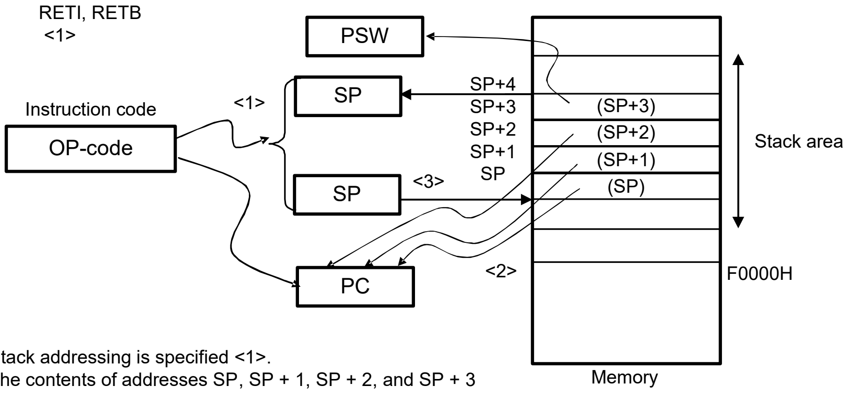
- Stack addressing is specified <1>.
- The contents of addresses SP, SP + 1, and SP + 2 are stored in PC bits 7 to 0, 15 to 8, and 19 to 16, respectively <2>.
- The value of SP <3> is increased by four.

Figure 3 - 36 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 - 37 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/G11 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, PM2 to PM5) Port registers (P0, P2 to P5, P12, P13) Pull-up resistor option registers (PU0, PU3 to PU5, PU12) Port input mode registers (PIM0, PIM3, PIM5) Port output mode registers (POM0, POM3 to POM5) Port mode control registers (PMC0, PMC2, PMC3, PMC5) Peripheral I/O redirection registers (PIOR0 to PIOR3)
Port	• 10-pin products Total: 7 (CMOS I/O: 4, CMOS input: 3) • 16-pin products Total: 13 (CMOS I/O: 9, CMOS input: 4) • 20-pin products Total: 17 (CMOS I/O: 13, CMOS input: 4) • 24-pin products Total: 21 (CMOS I/O: 17, CMOS input: 4) • 25-pin products Total: 21 (CMOS I/O: 17, CMOS input: 4)
Pull-up resistor	• 10-pin products Total: 2 • 16-pin products Total: 6 • 20-pin products Total: 10 • 24-pin products Total: 14 • 25-pin products Total: 14

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 and P01 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).

Input to the P00 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 0 (PIM0).

Output from the P00 and P01 pin can be specified as N-ch open-drain output (V_{DD} tolerance[Note 1]/EV_{DD} tolerance[Note 2]) in 1-bit units using port output mode register 0 (PIM0).

To use P00 and P01 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, external interrupt request input, serial interface data I/O, timer input, comparator output and buzzer output.

Reset signal generation sets port 0 to analog input port.

Note 1. 20, 24-pin products

Note 2. 25-pin products

4.2.2 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).

To use P20 to P23 as digital input/output pins, set them in the digital I/O mode by using port mode control register 2 (PMC2) (can be specified in 1-bit units).

This port can also be used for A/D converter analog input, analog voltage input of comparator, PGA input, serial interface data output, and D/A converter output.

Reset signal generation sets port 2 to analog input port.

4.2.3 Port 3

Port 3 is an I/O port with an output latch. Port 3 can be set to the input mode or output mode in 1-bit units using port mode register 3 (PM3). When the P30 to 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 to P32 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 to P33 pin can be specified as N-ch open-drain output (V_{DD} tolerance[Note 1]/EV_{DD} tolerance[Note 2]) in 1-bit units using port output mode register 3 (POM3).

To use P30 to P33 as digital input/output pins, set them in the digital I/O mode by using port mode control register 3 (PMC3) (can be specified in 1-bit units).

This port can also be used for analog input of A/D converter, external interrupt request input, key interrupt input, clock I/O and data I/O of serial interface, timer I/O, buzzer output, comparator output and analog voltage input of comparator.

Reset signal generation sets port 3 to input port.

Note 1. 20, 24-pin products

Note 2. 25-pin products

4.2.4 Port 4

Port 4 is an I/O port with an output latch. Port 4 can be set to the input mode or output mode in 1-bit units using port mode register 4 (PM4). When the P40 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).

Input to the P40 pin can be specified through a normal input buffer or a TTL input buffer in 1-bit units using port input mode register 4 (PIM4).

Output from the P40 pin can be specified as N-ch open-drain output (V_{DD} tolerance[Note 1]/EV_{DD} tolerance[Note 2]) in 1-bit units using port output mode register 4 (POM4).

This port can also be used for data I/O for a flash memory programmer/debugger, timer output, buzzer output, serial interface clock I/O, comparator output and interrupt flag output.

Reset signal generation sets port 4 to input port.

Note 1. 20, 24-pin products

Note 2. 25-pin products

4.2.5 Port 5

Port 5 is an I/O port with an output latch. Port 5 can be set to the input mode or output mode in 1-bit units using port mode register 5 (PM5). When the P51 to P56 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 to P56 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[Note 1]/EV_{DD} tolerance[Note 2]) in 1-bit units using port output mode register 5 (POM5).

To use P56 as digital input/output pins, set them in the digital I/O mode by using port mode control register 5 (PMC5) (can be specified in 1-bit units).

This port can also be used for analog input of A/D converter, external interrupt request input, key interrupt input, clock I/O and data I/O of serial interface, transmission/reception of programming UART, timer I/O, comparator output and interrupt flag output.

Reset signal generation sets port 5 to input port (other than P56) and analog input port (P56).

Note 1. 16, 20, 24-pin products

Note 2. 25-pin products

4.2.6 Port 12

P121, P122 and P125 are 3-bit input-only port.

Use of an on-chip pull-up resistor can be specified for the P125 pin by using pull-up resistor option register 12 (PU12) (the on-chip pull-up resistor is valid after the $\overline{\text{RESET}}$ pin (PORTSELB = 1) is selected).

This port can also be used for connecting resonator for main system clock, external clock input for main system clock, external interrupt request input, serial interface data input, and timer input.

Caution **After the power is turned on, P125 functions as $\overline{\text{RESET}}$ input. Even if an internal reset signal is released by power-on-reset (POR), the reset status continues as long as the low level is output to this pin. To use P125/KR1/SI01, select the port function (PORTSELB = 0) by the option byte (000C1H) and clear all reset sources.**

4.2.7 Port 13

P137 is a 1-bit input-only port.

This port can also be used for external interrupt request input, chip select input of serial interface and timer 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 x (PIORx)

Caution Which registers and bits are included depends on the product. For registers and bits mounted on each product, see Table 4 - 2. 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 (1/2)

Port		Bit Name						25-pin	24-pin	20-pin	16-pin	10-pin
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register					
Port 0	0	PM00	P00	PU00	PIM00	POM00	PMC00	√	√	√	—	—
	1	PM01	P01	PU01	—	POM01	PMC01	√	√	√	—	—
	2	—	—	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—
Port 2	0	PM20	P20	—	—	—	PMC20	√	√	√	√	√
	1	PM21	P21	—	—	—	PMC21	√	√	√	√	√
	2	PM22	P22	—	—	—	PMC22	√	√	√	√	√
	3	PM23	P23	—	—	—	PMC23	√	√	√	√	—
	4	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—
Port 3	0	PM30	P30	PU30	PIM30	POM30	PMC30	√	√	√	√	—
	1	PM31	P31	PU31	PIM31	POM31	PMC31	√	√	√	√	—
	2	PM32	P32	PU32	PIM32	POM32	PMC32	√	√	—	—	—
	3	PM33	P33	PU33	—	POM33 Note	PMC33	√	√	√	√	—
	4	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—

Note 24, 25-pin products

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

Port		Bit Name						25-pin	24-pin	20-pin	16-pin	10-pin
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register					
Port 4	0	PM40	P40	PU40	PIM40 Note 1	POM40 Note 1	—	√	√	√	√	√
	1	—	—	—	—	—	—	—	—	—	—	—
	2	—	—	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—
Port 5	0	—	—	—	—	—	—	—	—	—	—	—
	1	PM51	P51	PU51	PIM51	POM51	—	√	√	—	—	—
	2	PM52	P52	PU52	PIM52	POM52	—	√	√	—	—	—
	3	PM53	P53	PU53	PIM53	POM53	—	√	√	—	—	—
	4	PM54	P54	PU54	PIM54 Note 2	POM54	—	√	√	√	—	—
	5	PM55	P55	PU55	PIM55	POM55	—	√	√	√	—	—
	6	PM56	P56	PU56	PIM56 Note 1	POM56	PMC56	√	√	√	√	—
	7	—	—	—	—	—	—	—	—	—	—	—
Port 12	0	—	—	—	—	—	—	—	—	—	—	—
	1	—	P121	—	—	—	—	√	√	√	√	—
	2	—	P122	—	—	—	—	√	√	√	√	√
	3	—	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—
	5	—	P125	PU125	—	—	—	√	√	√	√	√
	6	—	—	—	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—	—	—	—
Port 13	0	—	—	—	—	—	—	—	—	—	—	—
	1	—	—	—	—	—	—	—	—	—	—	—
	2	—	—	—	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—	—	—	—
	7	—	P137	—	—	—	—	√	√	√	√	√

Note 1. 20, 24, 25-pin products

Note 2. 24, 25-pin products

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	1	1	1	PM01	PM00	FFF20H	FFH	R/W
PM2	1	1	1	1	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	1	PM56	PM55	PM54	PM53	PM52	PM51	1	FFF25H	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0, 2 to 5; n = 0 to 6)
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 P00, P01, P20 to P23 , P30 to P33, P56 are set to the analog function, if a port is read in input mode, the read value is always 0, not the pin level.
In addition, if the input data is read from P125 when the $\overline{\text{RESET}}$ pin (PORTSELB = 1) is selected, 1 is always read.

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	0	0	0	P01	P00	FFF00H	00H (output latch)	R/W
P2	0	0	0	0	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	0	P56	P55	P54	P53	P52	P51	0	FFF05H	00H (output latch)	R/W
P12	0	0	P125	0	0	P122	P121	0	FFF0CH	Undefined	R
P13	P137	0	0	0	0	0	0	0	FFF0DH	Undefined	R

Pmn	m = 0, 2 to 5, 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

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).

Caution When a port with PIMn register is used to TTL buffer input from a different potential device, set PUmn to 0 and pull-up the port to power supply of the different potential device via an external resistor.

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	0	0	0	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	0	PU56	PU55	PU54	PU53	PU52	PU51	0	F0035H	00H	R/W
PU12	0	0	PU125 Note	0	0	0	0	0	F003CH	20H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0, 3 to 5, 12; n = 0 to 6)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

Note PU125 can be selected when P125/INTP9 (PORTSELB = 0) is selected.
When the $\overline{\text{RESET}}$ pin (PORTSELB = 1) is selected, the on-chip pull-up resistor is enable.

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
PIM0	0	0	0	0	0	0	0	PIM00	F0040H	00H	R/W
PIM3	0	0	0	0	0	PIM32	PIM31	PIM30	F0043H	00H	R/W
PIM4	0	0	0	0	0	0	0	PIM40 Note 1	F0044H	00H	R/W
PIM5	0	PIM56 Note 1	PIM55	PIM54 Note 2	PIM53	PIM52	PIM51	0	F0045H	00H	R/W

PIMmn	Pmn pin input buffer selection (m = 0, 3 to 5; n = 0 to 6)
0	Normal input buffer
1	TTL input buffer

Note 1. 20, 24, 25-pin products

Note 2. 24, 25-pin products

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[Note 1]/EV_{DD} tolerance[Note 2]) mode can be selected during serial communication with an external device of the different potential, and for the SDAA0, SDAA1, SDA00, SDA01, SDA10 and SDA11 pins during multi-master I^{2}C and simplified I^{2}C 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[Note 1]/EV_{DD} tolerance[Note 2]) 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
POM0	0	0	0	0	0	0	POM01	POM00	F0050H	00H	R/W
POM3	0	0	0	0	POM33 Note 3	POM32	POM31	POM30	F0053H	00H	R/W
POM4	0	0	0	0	0	0	0	POM40 Note 4	F0054H	00H	R/W
POM5	0	POM56	POM55	POM54	POM53	POM52	POM51	0	F0055H	00H	R/W

POMmn	Pmn pin output mode selection (m = 0, 2 to 5; n = 0 to 6)
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance[Note 1]/EV_{DD} tolerance[Note 2]) mode

Note 1. 16, 20, 24-pin products
Note 2. 25-pin products
Note 3. 24, 25-pin products
Note 4. 20, 24, 25-pin products

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 pins as 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	1	1	1	PMC01	PMC00	F0060H	FFH	R/W
PMC2	1	1	1	1	PMC23	PMC22	PMC21	PMC20	F0062H	FFH	R/W
PMC3	1	1	1	1	PMC33	PMC32	PMC31	PMC30	F0063H	FFH	R/W
PMC5	1	PMC56	1	1	1	1	1	1	F0065H	FFH	R/W

PMCmn	Pmn pin digital I/O/analog input selection (m = 0, 2, 3, 5; n = 0 to 3, 6)
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 (PIORx)

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 PIORx 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 (PIORx)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIOR0	0	PIOR06	PIOR05	PIOR04	PIOR03	PIOR02	PIOR01	PIOR00	F0077H	00H	R/W
PIOR1	PIOR17	PIOR16	PIOR15	PIOR14	PIOR13	PIOR12	PIOR11	PIOR10	F0079H	00H	R/W
PIOR2	0	0	PIOR25	PIOR24	PIOR23	PIOR22	PIOR21	PIOR20	F0075H	00H	R/W
PIOR3	0	0	0	0	PIOR33	PIOR32	PIOR31	PIOR30	F007CH	00H	R/W

Bit	Function	25-pin		24-pin		20-pin		16-pin	
		Setting value		Setting value		Setting value		Setting value	
		0	1	0	1	0	1	0	1
PIOR06	INTP11	P55	P33	P55	P33	P55	P33	Note	Note
PIOR05	INTP10	P56	P32	P56	P32	P56	Note	Note	Note
PIOR02	VCOUT0	P40	P55	P40	P55	P40	P55	Note	Note
PIOR01	PCLBUZ0	P30	P40	P30	P40	P30	P40	P30	P40
PIOR00	INTFO	P40	P56	P40	P56	P40	P56	P40	P56
PIOR15	TO02	P55	P51	P55	P51	P55	Note	Note	Note
PIOR10	TI01	P31	P121	P31	P121	P31	P121	P31	P121
PIOR25	SO11	P32	P56	P32	P56	Note	Note	—	—
PIOR24	SO10/TxD1	P01	P20	P01	P20	P01	P20	—	—
PIOR23	SI10/RxD1	P00	P122	P00	P122	P00	P122	—	—
PIOR22	SI01	P52	P121	P52	P121	Note	Note	—	—

Bit	Function	Bit	25-pin				24-pin				20-pin				16-pin			
			Setting value				Setting value				Setting value				Setting value			
			00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
PIOR04	PIOR03	VCOUT1	P40	P00	P32	Note	P40	P00	P32	Note	P40	P00	Note	Note	Note	Note	Note	Note
PIOR17	PIOR16	TO03	P40	P56	P54	Note	P40	P56	P54	Note	P40	P56	P54	Note	P40	P56	Note	Note
PIOR14	PIOR13	TI03	P00	P54	P137	Note	P00	P54	P137	Note	P00	P54	P137	Note	Note	Note	Note	Note
PIOR12	PIOR11	TI02	P55	P122	P51	Note	P55	P122	P51	Note	P55	P122	Note	Note	Note	Note	Note	Note
PIOR21	PIOR20	RXD0	P55	P31	P52	Note	P55	P31	P52	Note	P55	P31	Note	Note	Note	Note	Note	Note
		TxD0	P54	P30	P51	Note	P54	P30	P51	Note	P54	P30	Note	Note	Note	Note	Note	Note
PIOR33	PIOR32	SCLA1	P56	P40	P33	Note	P56	P40	P33	Note	P56	P40	Note	Note	—	—	—	—
		SDAA1	P55	P00	P32	Note	P55	P00	P32	Note	P55	P00	Note	Note	—	—	—	—
PIOR31	PIOR30	SCLA0	P54	P31	P54	Note	P54	P31	P54	Note	Note	Note	Note	Note	—	—	—	—
		SDAA0	P53	P30	P52	Note	P53	P30	P52	Note	Note	Note	Note	Note	—	—	—	—

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

4.3.8 Global digital input disable register (GDIDIS)

This register is used to prevent through-current flowing to the input buffers of input ports which use EV_{DD} as the power supply when the EV_{DD} power supply is turned off.
When not all of the I/O ports using EV_{DD} as the power supply are used, low power consumption can be achieved by setting the GDIDIS register (setting the GDIDIS0 bit to 1) to turn off the EV_{DD} power supply.
By setting the GDIDIS0 bit to 1, input to any input buffer using EV_{DD} as the power supply is prohibited, preventing through-current from flowing when the EV_{DD} power supply is turned off.
The GDIDIS register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to 00H.

Remark The GDIDIS register is equipped with 25-pin products.

Figure 4 - 8 Format of Global digital input disable register (GDIDIS)

Address: F007DH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
GDIDIS	0	0	0	0	0	0	0	GDIDIS0

GDIDIS0	Setting of input buffers using EV_{DD} power supply
0	Input to input buffers permitted (default)
1	Input to input buffers prohibited. No through-current flows to the input buffers.

Turn off the EV_{DD} power supply with the following procedure.
1. Prohibit input to input buffers (set GDIDIS0 = 1).
2. Turn off the EV_{DD} power supply.

Turn on again the EV_{DD} power supply with the following procedure.
1. Turn on the EV_{DD} power supply.
2. Permit input to input buffers (set GDIDIS0 = 0).

Caution 1. Do not input an input voltage equal to or greater than EV_{DD} to an input port that uses EV_{DD} as the power supply.

Caution 2. When input to input buffers is prohibited (GDIDIS0 = 1), the value read from the port register (Pxx) of a port that uses EV_{DD} as the power supply is 1. When 1 is set in the port output mode register (POMxx) (N-ch open drain output (EV_{DD} tolerance) mode), the value read from the port register (Pxx) is 0.

Remark Even when input to input buffers is prohibited (GDIDIS0 = 1), peripheral functions which do not use port functions having EV_{DD} as the power supply can be used.

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, 3.0 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, 3.0 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, 3.0 V), set the port input mode registers 0, 3 to 5 (PIM0 and PIM3 to 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, 3.0 V), set the port output mode registers 0, 3 to 5 (POM0, POM3 to POM5) on a bit-by-bit basis to enable normal output (CMOS)/N-ch open drain (V_{DD} tolerance[Note 1]/EV_{DD} tolerance[Note 2]) switching.

Note 1. 16, 20, 24-pin products
Note 2. 25-pin products

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

(1) Setting procedure when using input pins of UART0, UART1, CSI00, CSI01, CSI10, and CSI11 functions for the TTL input buffer

In case of UART0:	P55 (P31, P52)
In case of UART1:	P00
In case of CSI00:	P55, P56
In case of CSI01:	P51, P52
In case of CSI10:	P00, P40
In case of CSI11:	P30, P31

Remark Functions in parentheses can be assigned via settings in the peripheral I/O redirection register 2 (PIOR2).

<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 PIM0 and PIM3 to 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/Simplified SPI (CSI[Note]) mode.

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

(2) Setting procedure when using output pins of UART0, UART1, CSI00, CSI01, CSI10, and CSI11 functions in N-ch open-drain output mode

In case of UART0:	P54 (P30, P51)
In case of UART1:	P01
In case of CSI00:	P54, P56
In case of CSI01:	P51, P53
In case of CSI10:	P01, P40
In case of CSI11:	P30, P32, (P56)

Remark Functions in parentheses can be assigned via settings in the peripheral I/O redirection register 2 (PIOR2).

<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 POM0 and POM3 to POM5 register to 1 to set the N-ch open drain output (V_{DD} tolerance[Note 1]/EV_{DD} tolerance[Note 2]) mode.
<5> Enable the operation of the serial array unit and set the mode to the UART/Simplified SPI (CSI) mode.
<6> Set the corresponding bit of the PM0 and PM3 to PM5 registers to the output mode. At this time, the output data is high level, so the pin is in the Hi-Z state.

Note 1. 16, 20, 24-pin products
Note 2. 25-pin products

(3) Setting procedure when using I/O pins of IIC00, IIC01, IIC10, and IIC11 functions with a different potential (1.8 V, 2.5 V, 3.0 V)

In case of simplified IIC00: P55, P56
In case of simplified IIC01: P51, P52
In case of simplified IIC10: P00, P40
In case of simplified IIC01: P30, P31

<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 POM0, POM3 to POM5 registers to 1 to set the N-ch open drain output (V_{DD} tolerance[Note 1]/EV_{DD} tolerance[Note 2]) mode.

<5> Set the corresponding bit of the PIM0, PIM3 to 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 PM0, PM3 to 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.

Note 1. 16, 20, 24-pin products
Note 2. 25-pin products

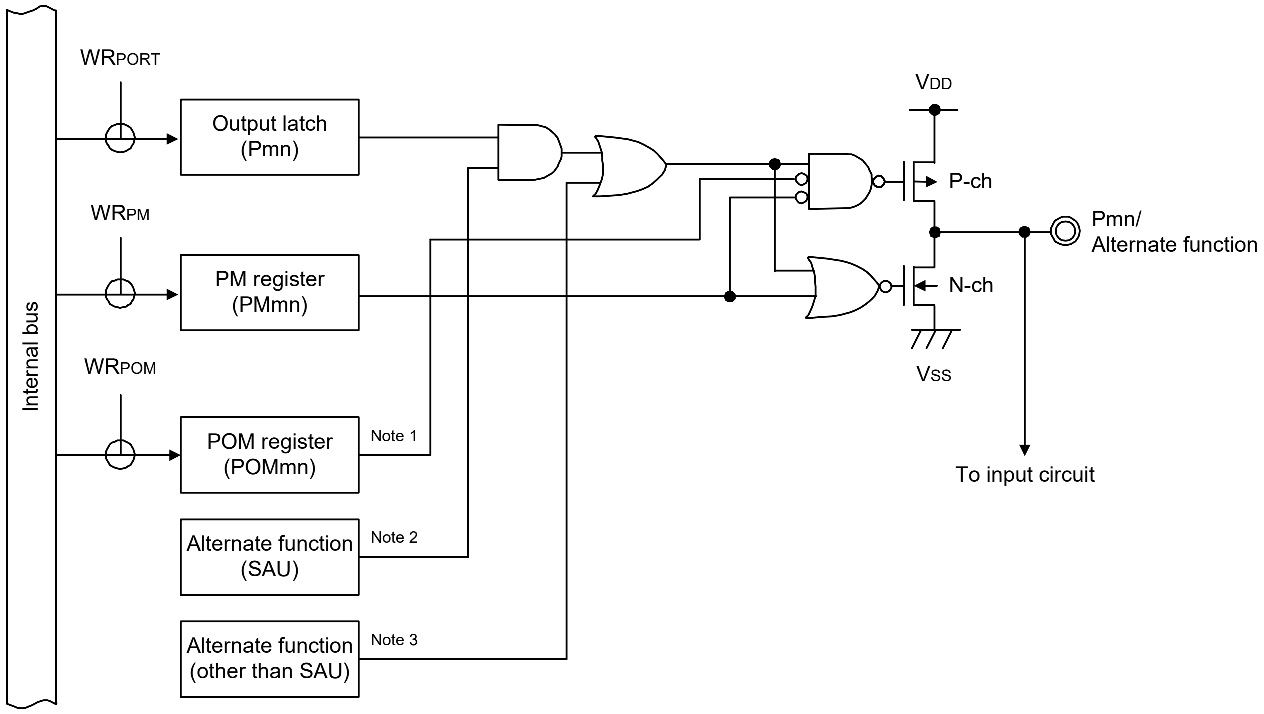
4.5 Register Settings When Using Alternate Function

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 - 9 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, 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 - 3.

Figure 4 - 9 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 - 3 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 to 3 (PIOR0 to PIOR3). 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) SDAAn = 0, SCLAn = 0 (setting when IICA is not used)
When IICA is not used, set the IICEn bit in IICA control register n0 (IICCTLn0) to 0 (operation stopped). This is the same setting as the initial state.

(5) PCLBUZn = 0 (setting when clock/buzzer output is not used)
When the clock/buzzer output is not used, set the PCLOEn bit in clock output select register n (CKSn) to 0 (output disabled). This is the same setting as the initial state.

(6) TKBO0 = 0 (setting when 16-bit timer KB0 is not used)
When 16-bit time KB0 is not used, set TKBTOEn0,1 bits of 16-bit timer KB output control registers n1 (TKBIOCn1) to 0 (stop output). This is the same setting as 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 - 4 to 4 - 10. The registers used to control the port functions should be set as shown in Tables 4 - 4 to 4 - 10. See the following remark for legends used in Tables 4 - 4 to 4 - 10.

Remark —: Not supported

×: Don't care[Note]

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 to 3 (PIOR0 to PIOR3).

Note This includes cases where there are no multiplexed pin functions.

Table 4 - 4 Setting Examples of Registers When Using P00 and P01 Pin Function

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		25 -pin	24 -pin	20 -pin	16 -pin	10 -pin
	Function Name	I/O						SAU Output Function	Other than SAU					
P00	P00	Input	—	×	0	1	×	×	×	√	√	√	—	—
		Output	—	0	0	0	0/1	SDA10 = 1	PCLBUZ1 = 0 (VCOUT1) = 0 (SDAA1) = 0					
		N ch-OD output	—	1	0	0	0/1							
	ANI17	Analog input	—	×	1	1	×	×	×	√	√	√	—	—
	PCLBUZ1	Output	—	0	0	0	0	SDA10 = 1	(VCOUT1) = 0 (SDAA1) = 0	√	√	√	—	—
	TI03	Input	—	×	0	1	×	×	×	√	√	√	—	—
	(VCOUT1)	Output	PIOR0 [4:3] = 01	0	0	0	0	SDA10 = 1	PCLBUZ1 = 0 (SDAA1) = 0	√	√	√	—	—
	SI10/RxD1	Input	—	×	0	1	×	×	×	√	√	√	—	—
	SDA10	I/O	—	1	0	0	1	—	PCLBUZ1 = 0 (VCOUT1) = 0 (SDAA1) = 0	√	√	√	—	—
	(SDAA1)	I/O	PIOR3 [3:2] = 01	1	0	0	0	SDA10 = 1	PCLBUZ1 = 0 (VCOUT1) = 0	√	√	√	—	—
P01	P01	Input	—	×	0	1	×	×	—	√	√	√	—	—
		Output	—	0	0	0	0/1	SO10/TxD1 = 1	—					
		N ch-OD output	—	1	0	0	0/1							
	ANI16	Analog input	—	×	1	1	×	×	—	√	√	√	—	—
	INTP5	Input	—	×	0	1	×	×	—	√	√	√	—	—
	SO10/ TxD1	Output	—	0/1	0	0	1	—	—	√	√	√	—	—

Table 4 - 5 Setting Examples of Registers When Using P20 to P23 Pin Function

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		25 -pin	24 -pin	20 -pin	16 -pin	10 -pin
	Function Name	I/O						SAU Output Function	Other than SAU					
P20	P20	Input	—	—	0	1	×	×	×	√	√	√	√	√
		Output	—	—	0	0	0/1	(SO10/TxD1) = 1 Note 1 SO10/TxD1 = 1 Note 2	TKBO1 = 0 Note 3					
	ANI0	Analog input	—	—	1	1	×	×	×	√	√	√	√	√
	AVREFP	Analog input	—	—	1	1	×	×	×	√	√	√	√	√
	IVREF1	Analog input	—	—	1	1	×	×	×	√	√	√	√	√
	(SO10/TxD1)	Output	PIOR2 [4] = 1	—	0	0	1	—	—	√	√	√	—	—
	SO10/TxD1	Output	—	—	0	0	1	—	—	—	—	—	√	—
	TKBO1	Output	—	—	0	0	0	—	—	—	—	—	—	√
P21	P21	Input	—	—	0	1	×	—	—	√	√	√	√	√
		Output	—	—	0	0	0/1	—	—					
	ANI1	Analog input	—	—	1	1	×	—	—	√	√	√	√	√
	AVREFM	Analog input	—	—	1	1	×	—	—	√	√	√	√	√
	IVREF0	Analog input	—	—	1	1	×	—	—	√	√	√	√	√
P22	P22	Input	—	—	0	1	×	×	—	√	√	√	√	√
		Output	—	—	0	0	0/1	SO10/TxD1 = 1 Note 3	—					
	ANI2	Analog input	—	—	1	1	×	×	—	√	√	√	√	√
	PGAI	Analog input	—	—	1	1	×	×	—	√	√	√	√	√
	IVCMP0	Analog input	—	—	1	1	×	×	—	√	√	√	√	√
	SO10/TxD1	Output	—	—	0	0	1	—	—	—	—	—	—	√
P23	P23	Input	—	—	0	1	×	—	—	√	√	√	√	—
		Output	—	—	0	0	0/1	—	—					
	ANI3	Analog input	—	—	1	1	×	—	—	√	√	√	√	—
	ANO1	Analog input	—	—	1	1	×	—	—	√	√	√	√	—
	PGAGND	Analog input	—	—	1	1	×	—	—	√	√	√	√	—

Note 1. 20, 24, 25-pin products only

Note 2. 16-pin products only

Note 3. 10-pin products only

Table 4 - 6 Setting Examples of Registers When Using P30 to P33 Pin Function (1/3)

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		25 -pin	24 -pin	20 -pin	16 -pin	10 -pin
	Function Name	I/O						SAU Output Function	Other than SAU					
P30	P30	Input	—	×	0	1	×	×	×	√	√	√	√	—
		Output	—	0	0	0	0/1	SCK11/ SCL11 = 1 (TxD0) = 1 Note 1 TxD0 = 1 Note 2	TO01 = 0 PCLBUZ0 = 0 TKBO1 = 0 (SDAA0) = 0 Note 3 SDAA0 = 0 Note 4					
		N ch-OD output	—	1	0	0	0/1							
	ANI21	Analog input	—	×	1	1	×	×	×	√	√	√	√	—
	KR1	Input	—	×	0	1	×	×	×	√	√	√	√	—
	TI00	Input	—	×	0	1	×	×	×	√	√	√	√	—
	TO01	Output	—	0	0	0	0	SCK11/ SCL11 = 1 (TxD0) = 1 Note 1 TxD0 = 1 Note 2	PCLBUZ0 = 0 TKBO1 = 0 (SDAA0) = 0 Note 3 SDAA0 = 0 Note 4	√	√	√	√	—
	INTP3	Input	—	×	0	1	×	×	×	√	√	√	√	—
	SCK11	Input	—	×	0	1	×	×	×	√	√	√	√	—
		Output	—	0/1	0	0	1	(TxD0) = 1 Note 1 TxD0 = 1 Note 2	TO01 = 0 PCLBUZ0 = 0 TKBO1 = 0 (SDAA0) = 0 Note 3 SDAA0 = 0 Note 4	√	√	√	√	—
	SCL11	Output	—	0/1	0	0	1	(TxD0) = 1 Note 1 TxD0 = 1 Note 2	TO01 = 0 PCLBUZ0 = 0 TKBO1 = 0 (SDAA0) = 0 Note 3 SDAA0 = 0 Note 4	√	√	√	√	—
	(TxD0)	Output	PIOR2 [1:0] = 01	0/1	0	0	1	SCK11/ SCL11 = 1	TO01 = 0 PCLBUZ0 = 0 TKBO1 = 0 (SDAA0) = 0 Note 3 SDAA0 = 0 Note 5	√	√	√	—	—
	TxD0	Output	—	0/1	0	0	1	SCK11/ SCL11 = 1	TO01 = 0 PCLBUZ0 = 0 TKBO1 = 0 SDAA0 = 0	—	—	—	√	—
	PCLBUZ0	Output	—	0	0	0	0	SCK11/ SCL11 = 1 (TxD0) = 1 Note 1 TxD0 = 1 Note 2	TO01 = 0 TKBO1 = 0 (SDAA0) = 0 Note 3 SDAA0 = 0 Note 4	√	√	√	√	—

Table 4 - 6 Setting Examples of Registers When Using P30 to P33 Pin Function (2/3)

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		25 -pin	24 -pin	20 -pin	16 -pin	10 -pin
	Function Name	I/O						SAU Output Function	Other than SAU					
P30	TKBO1	Output	—	0	0	0	0	SCK11/ SCL11 = 1 (TxD0) = 1[Note 1] TxD0 = 1[Note 2]	TO01 = 0 PCLBUZ0 = 0 (SDAA0) = 0 [Note 3] SDAA0 = 0[Note 4]	√	√	√	√	—
	(SDAA0)	I/O	PIOR3 [1:0] = 01	1	0	0	0	SCK11/ SCL11 = 1 (TxD0) = 1	TO01 = 0 PCLBUZ0 = 0 TKBO1 = 0	√	√	—	—	—
	SDAA0	I/O	—	1	0	0	0	SCK11/ SCL11 = 1 (TxD0) = 1[Note 5] TxD0 = 1[Note 2]	TO01 = 0 PCLBUZ0 = 0 TKBO1 = 0	—	—	√	√	—
P31	P31	Input	—	×	0	1	×	×	×	√	√	√	√	—
		Output	—	0	0	0	0/1	SDA11 = 1	TO00 = 0 TKBO0 = 0 (SCLA0) = 0[Note 3] SCLA0 = 0[Note 4]					
		N ch-OD output	—	1	0	0	0/1							
	ANI20	Analog input	—	×	1	1	×	×	×	√	√	√	√	—
	KR0	Input	—	×	0	1	×	×	×	√	√	√	√	—
	TI01	Input	—	×	0	1	×	×	×	√	√	√	√	—
	TO00	Output	—	0	0	0	0	SDA11 = 1	TKBO0 = 0 (SCLA0) = 0[Note 3] SCLA0 = 0[Note 4]	√	√	√	√	—
	INTP4	Input	—	×	0	1	×	×	×	√	√	√	√	—
	TKBO0	Output	—	0	0	0	0	SDA11 = 1	TO00 = 0 (SCLA0) = 0[Note 3] SCLA0 = 0[Note 4]	√	√	√	√	—
	(RxD0)	Input	PIOR2 [1:0] = 01	×	0	1	×	×	×	√	√	√	—	—
	RxD0	Input	—	×	0	1	×	×	×	—	—	—	√	—
	SI11	Input	—	×	0	1	×	×	×	√	√	√	√	—
	SDA11	I/O	—	1	0	0	1	—	TO00 = 0 TKBO0 = 0 (SCLA0) = 0[Note 3] SCLA0 = 0[Note 4]	√	√	√	√	—
	(SCLA0)	I/O	PIOR3 [1:0] = 01	1	0	0	0	SDA11 = 1	TO00 = 0 TKBO0 = 0	√	√	—	—	—
	SCLA0	I/O	—	1	0	0	0	SDA11 = 1	TO00 = 0 TKBO0 = 0	—	—	√	√	—

Table 4 - 6 Setting Examples of Registers When Using P30 to P33 Pin Function (3/3)

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		25 -pin	24 -pin	20 -pin	16 -pin	10 -pin
	Function Name	I/O						SAU Output Function	Other than SAU					
P32	P32	Input	—	×	0	1	×	×	×	√	√	—	—	—
		Output	—	0	0	0	0/1	SO11 = 1	(VCOUT1) = 0 (SDAA1) = 0					
		N ch-OD output	—	1	0	0	0/1							
	ANI19	Analog input	—	×	1	1	×	×	×	√	√	—	—	—
	SO11	Output	—	0/1	0	0	1	—	(VCOUT1) = 0 (SDAA1) = 0	√	√	—	—	—
	(INTP10)	Input	PIOR0[5] = 1	×	0	1	×	×	×	√	√	—	—	—
	(VCOUT1)	Output	PIOR0 [4:3] = 10	0	0	0	0	SO11 = 1	(SDAA1) = 0	√	√	—	—	—
	(SDAA1)	I/O	PIOR3 [3:2] = 10	1	0	0	0	SO11 = 1	(VCOUT1) = 0	√	√	—	—	—
P33	P33	Input	—	×	0	1	×	—	×	√	√	√	√	—
		Output	—	0	0	0	0/1	—	(SCLA1) = 0$^{\text{Note 3}}$					
		N ch-OD output	—	1	0	0	0/1							
	ANI18	Analog input	—	×	1	1	×	—	×	√	√	√	√	—
	IVCMP1	Analog input	—	×	1	1	×	—	×	√	√	√	√	—
	(INTP11)	Input	PIOR0[6] = 1	×	0	1	×	—	×	√	√	√	—	—
	INTP11	Input	—	×	0	1	×	—	×	—	—	—	√	—
	(SCLA1)	I/O	PIOR3 [3:2] = 10	1	0	0	0	—	—	√	√	—	—	—

Note 1. 20, 24, 25-pin products only

Note 2. 16-pin products only

Note 3. 24, 25-pin products only

Note 4. 16, 20-pin products only

Note 5. 20-pin products only

Table 4 - 7 Setting Examples of Registers When Using P40 Pin Function (1/2)

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		25 -pin	24 -pin	20 -pin	16 -pin	10 -pin
	Function Name	I/O						SAU Output Function	Other than SAU					
P40	P40	Input	—	×	0	1	×	×	×	√	√	√	√	√
		Output	—	0	0	0	0/1	SCK10 = 1 SCL10 = 1 Note 1	TO03 = 0 (PCLBUZ0) = 0 Note 2 PCLBUZ0 = 0 Note 3 VCOUT0 = 0 VCOUT1 = 0$^{\text{Note 2}}$ INTFO = 0 (SCLA1) = 0$^{\text{Note 1}}$ TKBO0 = 0$^{\text{Note 3}}$					
		N ch-OD output	—	1	0	0	0/1							
	TO03	Output	—	0	0	0	0	SCK10 = 1 SCL10 = 1 Note 1	(PCLBUZ0) = 0 Note 2 PCLBUZ0 = 0 Note 3 VCOUT0 = 0 VCOUT1 = 0$^{\text{Note 2}}$ INTFO = 0 (SCLA1) = 0$^{\text{Note 1}}$ TKBO0 = 0$^{\text{Note 3}}$	√	√	√	√	√
	(PCLBUZ0)	Output	PIOR0[1] = 1	0	0	0	0	SCK10 = 1 SCL10 = 1 Note 1	TO03 = 0 VCOUT0 = 0 VCOUT1 = 0 INTFO = 0 (SCLA1) = 0$^{\text{Note 1}}$	√	√	√	√	—
	PCLBUZ0	Output	—	0	0	0	0	SCK10 = 1	TO03 = 0 VCOUT0 = 0 INTFO = 0 TKBO0 = 0	—	—	—	—	√
	SCK10	Input	—	×	0	1	×	×	×	√	√	√	√	√
		Output	—	0/1	0	0	1	×	TO03 = 0 (PCLBUZ0) = 0 Note 2 PCLBUZ0 = 0 Note 3 VCOUT0 = 0 VCOUT1 = 0$^{\text{Note 2}}$ INTFO = 0 (SCLA1) = 0$^{\text{Note 1}}$ TKBO0 = 0$^{\text{Note 3}}$	√	√	√	√	√
	SCL10	Output	—	0/1	0	0	1	×	TO03 = 0 (PCLBUZ0) = 0 VCOUT0 = 0 VCOUT1 = 0 INTFO = 0 (SCLA1) = 0	√	√	√	—	—

Table 4 - 7 Setting Examples of Registers When Using P40 Pin Function (2/2)

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		25 -pin	24 -pin	20 -pin	16 -pin	10 -pin
	Function Name	I/O						SAU Output Function	Other than SAU					
P40	VCOUT0	Output	—	0	0	0	0	SCK10 = 1 SCL10 = 1 Note 1	TO03 = 0 (PCLBUZ0) = 0 Note 2 PCLBUZ0 = 0 Note 3 VCOUT1 = 0 Note 2 INTFO = 0 (SCLA1) = 0 Note 1 TKBO0 = 0Note 3	√	√	√	√	√
	VCOUT1	Output	—	0	0	0	0	SCK10 = 1 SCL10 = 1 Note 1	TO03 = 0 (PCLBUZ0) = 0 VCOUT0 = 0 INTFO = 0 (SCLA1) = 0 Note 1	√	√	√	√	—
	INTFO	Output	—	0	0	0	0	SCK10 = 1 SCL10 = 1 Note 1	TO03 = 0 (PCLBUZ0) = 0 Note 2 PCLBUZ0 = 0 Note 3 VCOUT0 = 0 VCOUT1 = 0 Note 2 (SCLA1) = 0 Note 1 TKBO0 = 0Note 3	√	√	√	√	√
	(SCLA1)	I/O	PIOR3 [3:2] = 01	1	0	0	0	SCK10/ SCL10=1	TO03 = 0 (PCLBUZ0) = 0 VCOUT0 = 0 VCOUT1 = 0 INTFO = 0	√	√	√	—	—
	TKBO0	Output	—	0	0	0	0	SCK10 = 1	TO03 = 0 PCLBUZ0 = 0 VCOUT0 = 0 INTFO = 0	—	—	—	—	√

Note 1. 20, 24, 25-pin products only

Note 2. 16, 20, 24, 25-pin products only

Note 3. 10-pin products only

Table 4 - 8 Setting Examples of Registers When Using P51 to P56 Pin Function (1/3)

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		25 -pin	24 -pin	20 -pin	16 -pin	10 -pin
	Function Name	I/O						SAU Output Function	Other than SAU					
P51	P51	Input	—	×	0	1	×	×	×	√	√	—	—	—
		Output	—	0	0	0	0/1	SCK10/ SCL10 = 1 (TxD0) = 1	(TO02) = 0					
		N ch-OD output	—	1	0	0	0/1							
	KR7	Input	—	×	0	1	×	×	×	√	√	—	—	—
	INTP8	Input	—	×	0	1	×	×	×	√	√	—	—	—
	(TI02)	Input	PIOR1[2:1] = 10$^{\text{Note 1}}$	×	0	1	×	×	×	√	√	—	—	—
	(TO02)	Output	PIOR1[5] = 1$^{\text{Note 1}}$	0	0	0	0	SCK10/ SCL10 = 1 (TxD0) = 1	—	√	√	—	—	—
	SCK01	Input	—	×	0	1	×	×	×	√	√	—	—	—
		Output	—	0/1	0	0	1	(TxD0) = 1	(TO02) = 0	√	√	—	—	—
	SCL01	Output	—	0/1	0	0	1	(TxD0) = 1	(TO02) = 0	√	√	—	—	—
	(TxD0)	Output	PIOR2[1:0] = 10	0/1	0	0	1	SCK10/ SCL10=1	(TO02) = 0	√	√	—	—	—
P52	P52	Input	—	×	0	1	×	×	×	√	√	—	—	—
		Output	—	0	0	0	0/1	SDA01 = 1	(SDAA0) = 0					
		N ch-OD output	—	1	0	0	0/1							
	KR6	Input	—	×	0	1	×	×	×	√	√	—	—	—
	INTP7	Input	—	×	0	1	×	×	×	√	√	—	—	—
	SI01	Input	—	×	0	1	×	×	×	√	√	—	—	—
	SDA01	I/O	—	1	0	0	1	—	(SDAA0) = 0	√	√	—	—	—
	(RxD0)	Input	PIOR2[1:0] = 10	×	0	1	×	×	×	√	√	—	—	—
	(SDAA0)	I/O	PIOR3[1:0] = 10	1	0	0	0	SDA01 = 1	—	√	√	—	—	—
P53	P53	Input	—	×	0	1	×	×	×	√	√	—	—	—
		Output	—	0	0	0	0/1	SO01 = 1	SDAA0 = 0					
		N ch-OD output	—	1	0	0	0/1							
	KR5	Input	—	×	0	1	×	×	×	√	√	—	—	—
	INTP6	Input	—	×	0	1	×	×	×	√	√	—	—	—
	SO01	Output	—	0/1	0	0	1	—	SDAA0 = 0	√	√	—	—	—
	SDAA0	I/O	—	1	0	0	0	SO01 = 1	—	√	√	—	—	—

Table 4 - 8 Setting Examples of Registers When Using P51 to P56 Pin Function (2/3)

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		25 -pin	24 -pin	20 -pin	16 -pin	10 -pin
	Function Name	I/O						SAU Output Function	Other than SAU					
P54	P54	Input	—	×	0	1	×	×	×	√	√	√	—	—
		Output	—	0	0	0	0/1	SO00/ TxD0 = 1	(TO03) = 0 SCLA0 = 0Note 1					
		N ch-OD output	—	1	0	0	0/1							
	KR4	Input	—	×	0	1	×	×	×	√	√	√	—	—
	SO00/TxD0	Output	—	0/1	0	0	1	—	(TO03) = 0 SCLA0 = 0Note 1	√	√	√	—	—
	(TI03)	Input	PIOR1[4:3] = 01	×	0	1	×	×	×	√	√	√	—	—
	(TO03)	Output	PIOR1[7:6] = 10	0	0	0	0	SO00/ TxD0 = 1	SCLA0 = 0Note 1	√	√	√	—	—
	SCLA0	I/O	—	1	0	0	0	SO00/ TxD0 = 1	(TO03) = 0	√	√	—	—	—
P55	P55	Input	—	×	0	1	×	×	×	√	√	√	—	—
		Output	—	0	0	0	0/1	SDA00 = 1	TO02 = 0 (VCOUT0) = 0 SDAA1 = 0					
		N ch-OD output	—	1	0	0	0/1							
	KR3	Input	—	×	0	1	×	×	×	√	√	√	—	—
	SI00/RxD0	Input	—	×	0	1	×	×	×	√	√	√	—	—
	SDA00	I/O	—	1	0	0	1	—	TO02 = 0 (VCOUT0) = 0 SDAA1 = 0	√	√	√	—	—
	TI02	Input	—	×	0	1	×	×	×	√	√	√	—	—
	TO02	Output	—	0	0	0	0	SDA00 = 1	(VCOUT0) = 0 SDAA1 = 0	√	√	√	—	—
	INTP11	Input	—	×	0	1	×	×	×	√	√	√	—	—
	(VCOUT0)	Output	PIOR0[2] = 1	0	0	0	0	SDA00 = 1	TO02 = 0 SDAA1 = 0	√	√	√	—	—
	SDAA1	I/O	—	1	0	0	0	SDA00 = 1	TO02 = 0 (VCOUT0) = 0	√	√	√	—	—

Table 4 - 8 Setting Examples of Registers When Using P51 to P56 Pin Function (3/3)

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		25 -pin	24 -pin	20 -pin	16 -pin	10 -pin
	Function Name	I/O						SAU Output Function	Other than SAU					
P56	P56	Input	—	×	0	1	×	×	×	√	√	√	√	—
		Output	—	0	0	0	0/1	SCK00/ SCL00 = 1[Note 3] (SO11) = 1[Note 1] SO11 = 1[Note 2]	(TO03) = 0 (INTFO) = 0 SCLA1 = 0[Note 3]					
		N ch-OD output	—	1	0	0	0/1							
	ANI22	Analog input	—	×	1	1	×	×	×	√	√	√	√	—
	KR2	Input	—	×	0	1	×	×	×	√	√	√	√	—
	SCK00	Input	—	×	0	1	×	×	×	√	√	√	—	—
		Output	—	0/1	0	0	1	(SO11) = 1[Note 1] SO11 = 1[Note 4]	(TO03) = 0 (INTFO) = 0 SCLA1 = 0	√	√	√	—	—
	SCL00	Output	—	0/1	0	0	1	(SO11) = 1[Note 1] SO11 = 1[Note 4]	(TO03) = 0 (INTFO) = 0 SCLA1 = 0	√	√	√	—	—
	(SO11)	Output	PIOR2[5] = 1	0/1	0	0	1	SCK00/ SCL00 = 1	(TO03) = 0 (INTFO) = 0 SCLA1 = 0	√	√	—	—	—
	SO11	Output	—	0/1	0	0	1	SCK00/ SCL00 = 1[Note 4]	(TO03) = 0 (INTFO) = 0 SCLA1 = 0[Note 4]	—	—	√	√	—
	INTP10	Input	—	×	0	1	×	×	×	√	√	√	√	—
	(TO03)	Output	PIOR1[7:6] = 01	0	0	0	0	SCK00/ SCL00 = 1[Note 3] (SO11) = 1[Note 1] SO11 = 1[Note 2]	(INTFO) = 0 SCLA1 = 0[Note 3]	√	√	√	√	—
	(INTFO)	Output	PIOR0[0] = 1	0	0	0	0	SCK00/ SCL00 = 1[Note 3] (SO11) = 1[Note 1] SO11 = 1[Note 2]	(TO03) = 0 SCLA1 = 0[Note 3]	√	√	√	√	—
	SCLA1	I/O	—	1	0	0	0	SCK00/ SCL00 = 1 (SO11) = 1[Note 1] SO11 = 1[Note 4]	(TO03) = 0 (INTFO) = 0	√	√	√	—	—

Note 1. 24, 25-pin products only

Note 2. 16, 20-pin products only

Note 3. 20, 24, 25-pin products only

Note 4. 20-pin products only

Table 4 - 9 Setting Examples of Registers When Using P121, P122 and P125 Pin Function

Pin Name	Used Function		PIORx	Pxx	Other	25 -pin	24 -pin	20 -pin	16 -pin	10 -pin
	Function Name	I/O								
P121	P121	Input	—	×	EXCLK, OSCSEL = 00/10/11	√	√	√	√	—
	X1	—	—	—	EXCLK, OSCSEL = 01	√	√	√	√	—
	(TI01)	Input	PIOR1[0] = 1	×	EXCLK, OSCSEL = 00/10/11	√	√	√	√	—
	INTP2	Input	—	×	EXCLK, OSCSEL = 00/10/11	√	√	√	√	—
	(SI01)	Input	PIOR2[2] = 1	×	EXCLK, OSCSEL = 00/10/11	√	√	—	—	—
P122	P122	Input	—	×	EXCLK, OSCSEL = 00/10	√	√	√	√	√
	X2	—	—	—	EXCLK, OSCSEL = 01	√	√	√	√	√
	EXCLK	Input	—	—	EXCLK, OSCSEL = 11	√	√	√	√	√
	SI10/RxD0	Input	—	×	EXCLK, OSCSEL = 00/10	—	—	—	√	—
	(SI10/RxD1)	Input	PIOR2[3] = 1	×	EXCLK, OSCSEL = 00/10	√	√	√	—	—
	TI02	Input	—	×	EXCLK, OSCSEL = 00/10	—	—	—	√	√
	(TI02)	Input	PIOR1[2:1] = 01	×	EXCLK, OSCSEL = 00/10	√	√	√	—	—
	INTP1	Input	—	×	EXCLK, OSCSEL = 00/10	√	√	√	√	√
P125	P125	Input	—	×	PORTSELB = 0	√	√	√	√	√
	$\overline{\text{RESET}}$	—	—	—	PORTSELB = 1	√	√	√	√	√
	INTP9	Input	—	×	PORTSELB = 0	√	√	√	√	√

Table 4 - 10 Setting Examples of Registers When Using P137 Pin Function

Pin Name	Used Function		PIORx	POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		25 -pin	24 -pin	20 -pin	16 -pin	10 -pin
	Function Name	I/O						SAU Output Function	Other than SAU					
P137	P137	Input	—	—	—	—	×	—	—	√	√	√	√	√
	INTP0	Input	—	—	—	—	×	—	—	√	√	√	√	√
	$\overline{\text{SSI00}}$	Input	—	—	—	—	×	—	—	√	√	√	—	—
	$\overline{\text{TI03}}$	Input	—	—	—	—	×	—	—	—	—	—	√	√
	(TI03)	Input	PIOR1 [4:3] = 10	—	—	—	×	—	—	√	√	√	—	—
	SI10/RxD1	Input	—	—	—	—	×	—	—	—	—	—	—	√

4.6 Cautions When Using Port Function

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

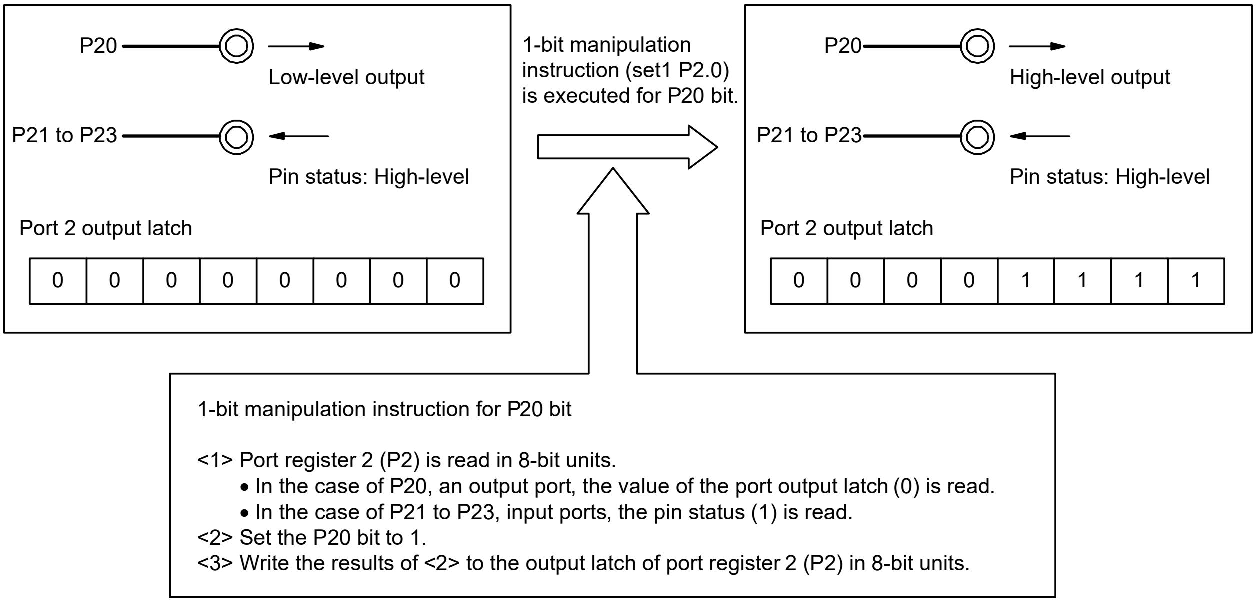
When a 1-bit manipulation instruction is executed on a port that provides both input and output functions, the output latch value of an input port that is not subject to manipulation may be written in addition to the targeted bit.
Therefore, it is recommended to rewrite the output latch when switching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

<Example> When P20 is an output port, P21 to P23 are input ports (all pin statuses are high level), and the port latch value of port 2 is 00H, if the output of output port P20 is changed from low level to high level via a 1-bit manipulation instruction, the output latch value of port 2 is 0FH.

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/G11.

<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 P20, which is an output port, is read, while the pin statuses of P21 to P23, which are input ports, are read. If the pin statuses of P21 to P23 are high level at this time, the read value is 0EH.
The value is changed to 0FH by the manipulation in <2>.
0FH is written to the output latch by the manipulation in <3>.

Figure 4 - 10 1-bit Manipulation Instruction (P20)



1-bit manipulation instruction for P20 bit

<1> Port register 2 (P2) is read in 8-bit units.
- In the case of P20, an output port, the value of the port output latch (0) is read.
- In the case of P21 to P23, input ports, the pin status (1) is read.

<2> Set the P20 bit to 1.
<3> Write the results of <2> to the output latch of port register 2 (P2) in 8-bit units.

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 to 3 (PIOR0 to PIOR3). 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.

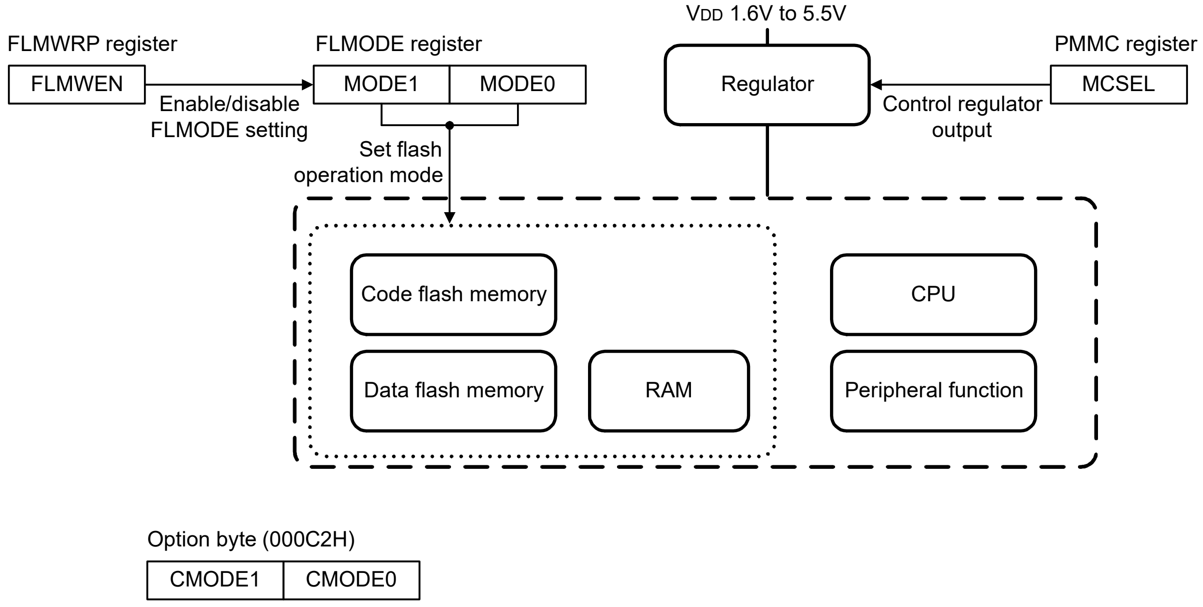
5.1 Configuration of Operation State Control

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)	2.7 V to 5.5 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.
		2.4 V to 5.5 V	1 MHz to 16 MHz		
LS (low-speed main) mode	Normal setting (MCSEL = 0)	1.8 V to 5.5 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 5.5 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 5.5 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 5.5 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	CFLSTOP

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} ≤ 5.5 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} ≤ 5.5 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} ≤ 5.5 V in LS mode.)

CFLSTOP	Control of the code flash memory operation
0	Operates
1	Stops

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 (low-speed on-chip oscillator (f_{IL})).

Caution 3. Do not change the values of the MODE1 and MODE0 bits while the setting of the CFLSTOP bit is 1.

Caution 4. Do not change the value of the MODE1 and MODE0 bits using the DTC.

Caution 5. 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 6. 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 7. 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 *1
LS (low-speed main) mode → LV (low-voltage main) mode	99 cycles *1, *2
LP (low-power main) mode → LS (low-speed main) mode	10 cycles *1
LS (low-speed main) mode → LP (low-power main) mode	10 cycles *1
LV (low-voltage main) mode → LS (low-speed main) mode	20 cycles *1
HS (high-speed main) mode → LS (low-speed main) mode	30 cycles *1

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

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

Caution 10.Since the operation of the code flash memory stops as soon as the CFLSTOP bit is set to 1, set the CFLSTOP bit from program code that has been transferred to RAM.

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 FRQSEL4 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	FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0

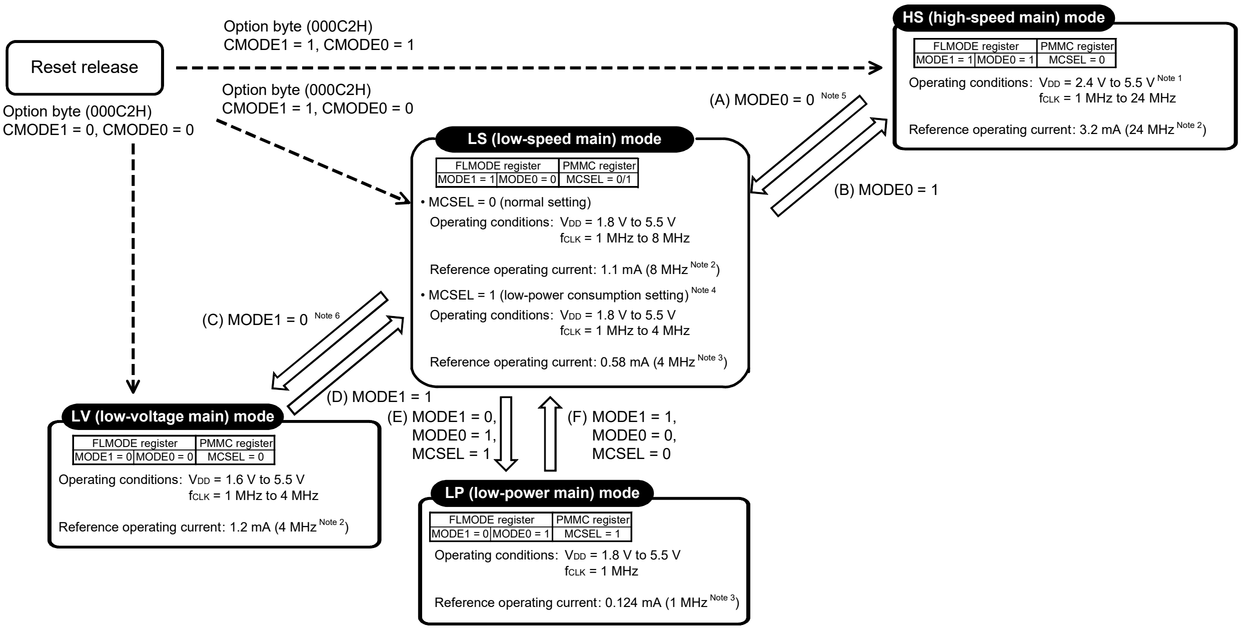
CMODE1	CMODE0	Setting of flash operation mode		
			Operating Frequency Range (f_{MAIN})	Operating Voltage Range (V_{DD})
0	0	LV (low-voltage main) mode	1 to 4 MHz	1.6 to 5.5 V
1	0	LS (low-speed main) mode	1 to 8 MHz	1.8 to 5.5 V
1	1	HS (high-speed main) mode	1 to 16 MHz	2.4 to 5.5 V
			1 to 24 MHz	2.7 to 5.5 V
Other than above		Setting prohibited		

FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator clock	
					f_{HOCO}	f_{IH}
1	0	0	0	0	48 MHz	24 MHz
0	0	0	0	0	24 MHz	24 MHz
0	1	0	0	1	16 MHz	16 MHz
0	0	0	0	1	12 MHz	12 MHz
0	1	0	1	0	8 MHz	8 MHz
0	0	0	1	0	6 MHz	6 MHz
0	1	0	1	1	4 MHz	4 MHz
0	0	0	1	1	3 MHz	3 MHz
0	1	1	0	0	2 MHz	2 MHz
0	1	1	0	1	1 MHz	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


Reset release
Option byte (000C2H)
CMODE1 = 1, CMODE0 = 1
Option byte (000C2H)
CMODE1 = 1, CMODE0 = 0
Option byte (000C2H)
CMODE1 = 0, CMODE0 = 0
HS (high-speed main) mode
FLMODE register | PMMC register
MODE1 = 1 | MODE0 = 1 | MCSEL = 0
Operating conditions: VDD = 2.4 V to 5.5 V Note 1
fCLK = 1 MHz to 24 MHz
Reference operating current: 3.2 mA (24 MHz Note 2)
(A) MODE0 = 0 Note 5
(B) MODE0 = 1
LS (low-speed main) mode
FLMODE register | PMMC register
MODE1 = 1 | MODE0 = 0 | MCSEL = 0/1
• MCSEL = 0 (normal setting)
Operating conditions: VDD = 1.8 V to 5.5 V
fCLK = 1 MHz to 8 MHz
Reference operating current: 1.1 mA (8 MHz Note 2)
• MCSEL = 1 (low-power consumption setting) Note 4
Operating conditions: VDD = 1.8 V to 5.5 V
fCLK = 1 MHz to 4 MHz
Reference operating current: 0.58 mA (4 MHz Note 3)
(C) MODE1 = 0 Note 6
(D) MODE1 = 1
(E) MODE1 = 0,
MODE0 = 1,
MCSEL = 1
(F) MODE1 = 1,
MODE0 = 0,
MCSEL = 0
LV (low-voltage main) mode
FLMODE register | PMMC register
MODE1 = 0 | MODE0 = 0 | MCSEL = 0
Operating conditions: VDD = 1.6 V to 5.5 V
fCLK = 1 MHz to 4 MHz
Reference operating current: 1.2 mA (4 MHz Note 2)
LP (low-power main) mode
FLMODE register | PMMC register
MODE1 = 0 | MODE0 = 1 | MCSEL = 1
Operating conditions: VDD = 1.8 V to 5.5 V
fCLK = 1 MHz
Reference operating current: 0.124 mA (1 MHz Note 3)

Note 1. The operating frequency and operating voltage range are as follows.
1 MHz ≤ f_{CLK} ≤ 16 MHz (2.4 V ≤ V_{DD} ≤ 5.5 V)
1 MHz ≤ f_{CLK} ≤ 24 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)

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.

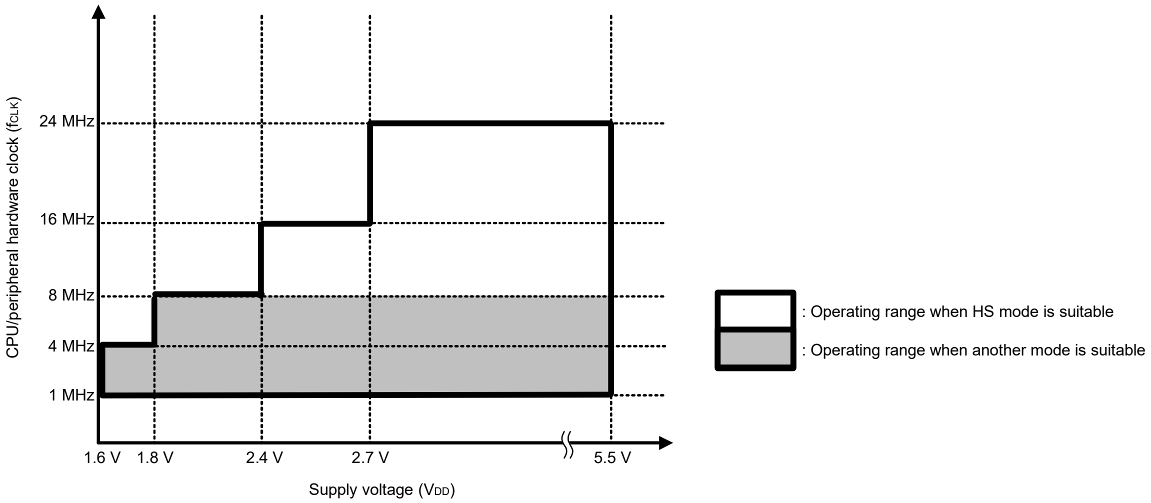
HS mode can be operated immediately after a reset release. Also, this mode can be entered from LS (low-speed main) mode. When entering HS mode, make sure that the supply voltage is 2.4 V ≤ V_{DD} ≤ 5.5 V in LS mode and the operating frequency is 1 MHz ≤ f_{CLK} ≤ 8 MHz.

Operating in HS mode is suitable when the power supply voltage and operating frequency meet any of the following conditions.

- The power supply voltage is 2.4 V ≤ V_{DD} ≤ 5.5 V and the operating frequency is 1 MHz ≤ f_{CLK} ≤ 16 MH
- The power supply voltage is 2.7 V ≤ V_{DD} ≤ 5.5 V and the operating frequency is 1 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} ≤ 5.5 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} ≤ 5.5 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)

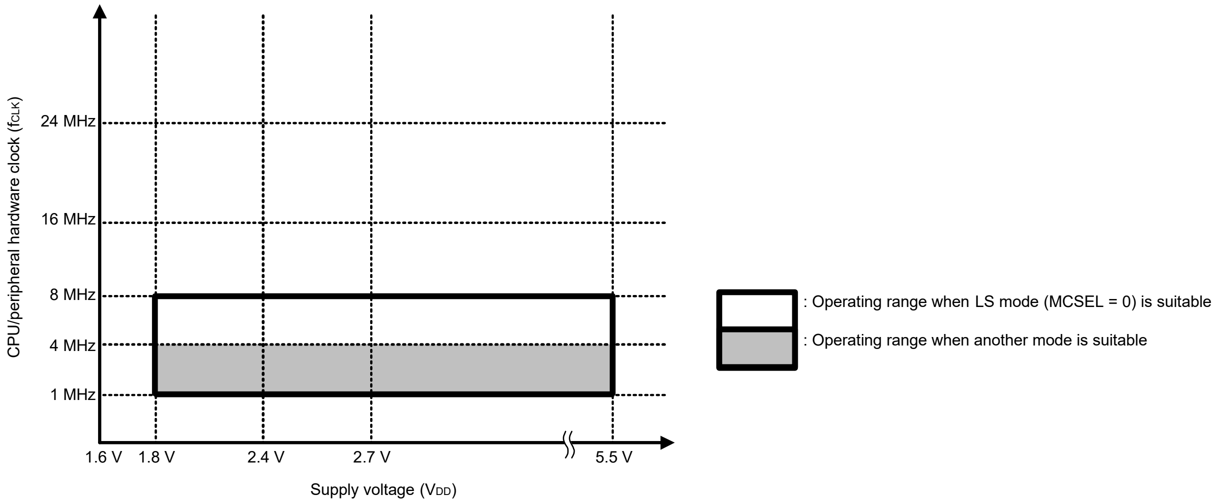
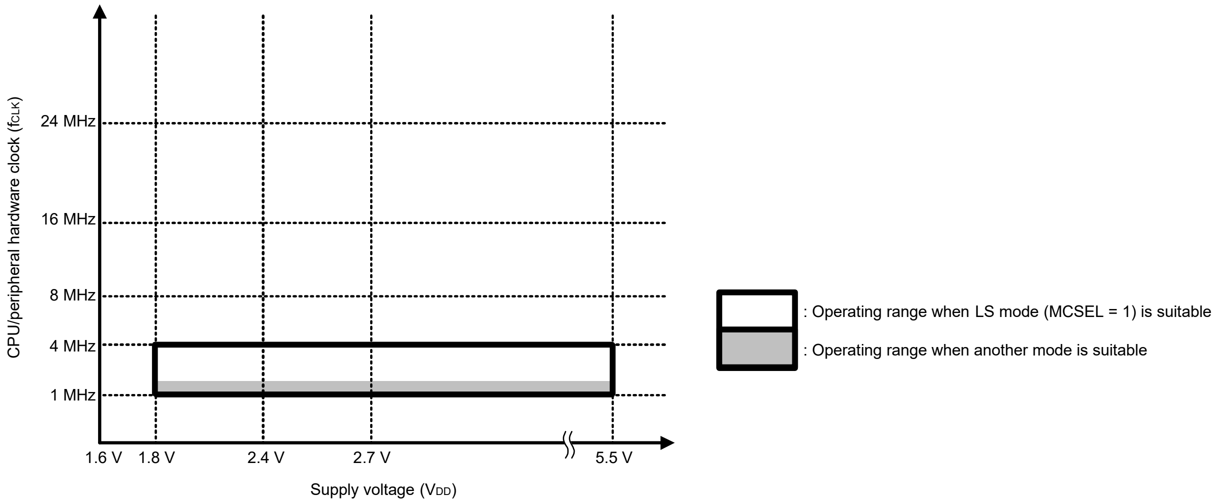


Figure 5 - 9 Operating Range in LS Mode (MCSEL = 1)



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

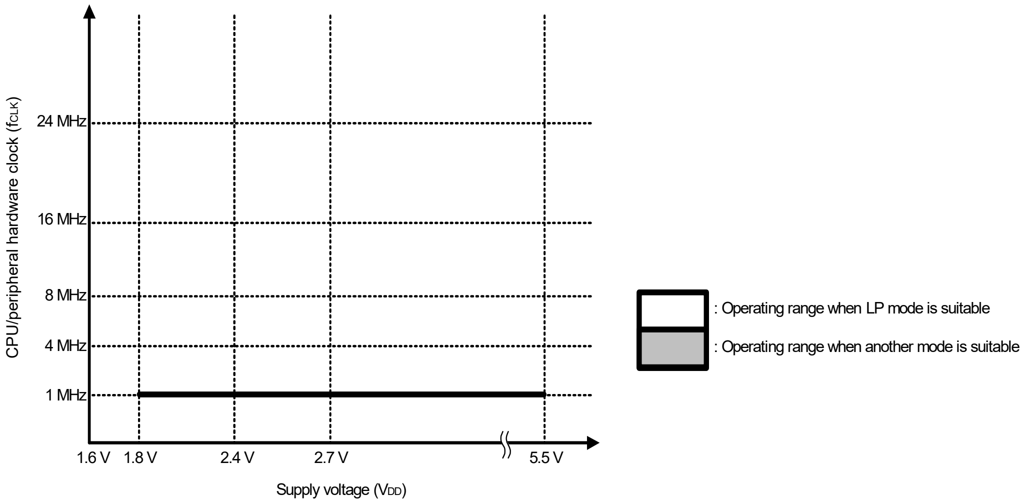
5.5.3 Details of LP (low-power main) mode

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} ≤ 5.5 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.

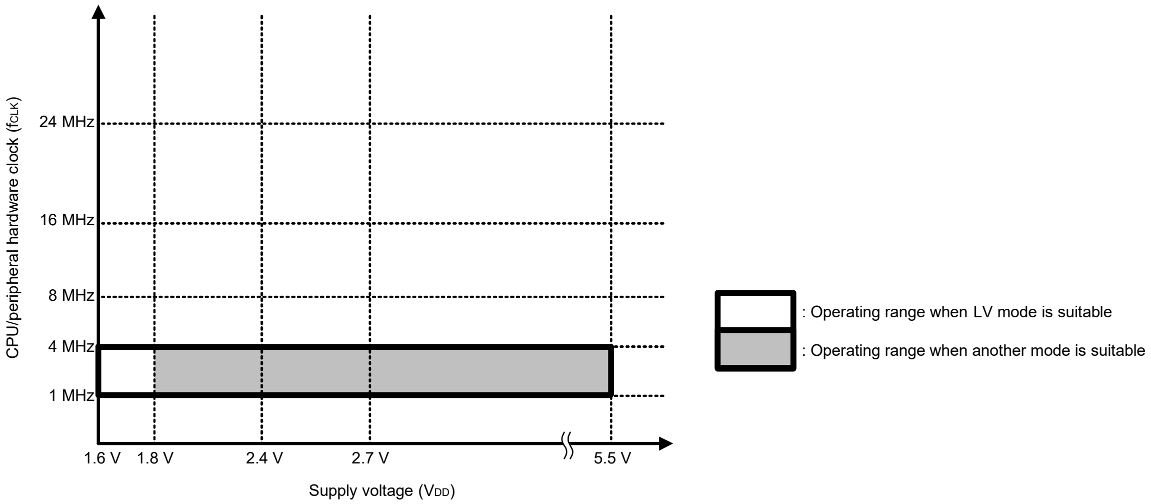
5.5.4 Details on LV (low-voltage main) mode

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} ≤ 5.5 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

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 of the high-speed on-chip oscillator (f_{HOCO}) can be selected from among 48, 24, 16, 12, 8, 6, 4, 3, 2, and 1 MHz (TYP.) by using the option byte (000C2H).

If f_{HOCO} is 48 MHz, f_{IH} is 24 MHz. If f_{HOCO} is no greater than 24 MHz, f_{IH} is generated by using f_{HOCO} without change instead of by frequency-dividing it. 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	48
2.7 V ≤ V_{DD} ≤ 5.5 V	√	√	√	√	√	√	√	√	√	√
2.4 V ≤ V_{DD} ≤ 5.5 V	√	√	√	√	√	√	√	√	—	—
1.8 V ≤ V_{DD} ≤ 5.5 V	√	√	√	√	√	√	—	—	—	—
1.6 V ≤ V_{DD} ≤ 5.5 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 30 OPTION BYTE**).

(2) Subsystem clock

<1> 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 operation speed 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_{IL}:	Low-speed on-chip oscillator frequency

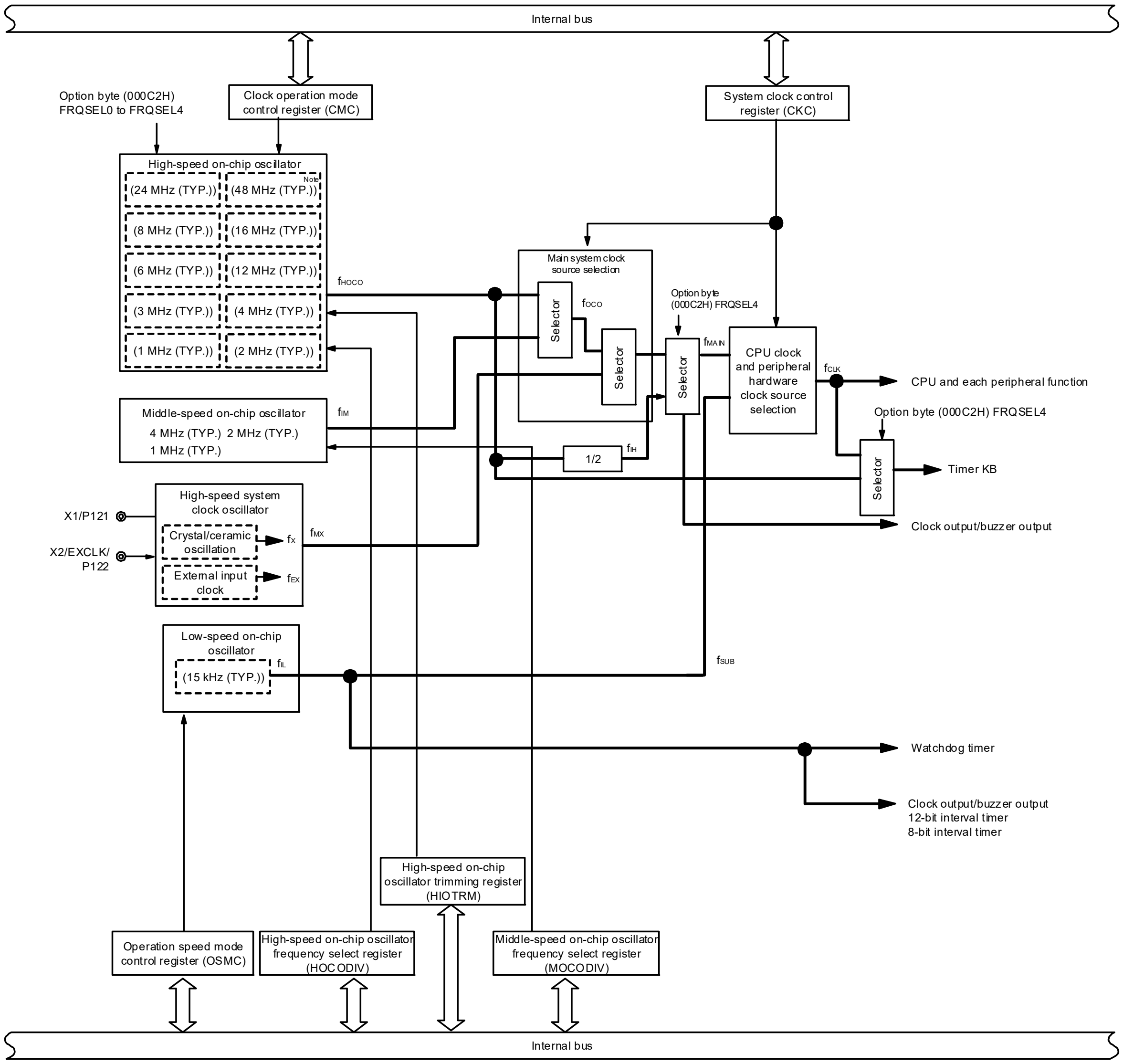
6.2 Configuration of Clock Generator

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) Operation speed 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)
Oscillators	X1 oscillator High-speed on-chip oscillator Middle-speed on-chip oscillator Low-speed on-chip oscillator

Figure 6 - 1 Block Diagram of Clock Generator



Note Two times frequency for TMKB.

Remark

fX:	X1 clock oscillation frequency
fHOCO:	High-speed on-chip oscillator clock frequency (48 MHz max.)
fIH:	High-speed on-chip oscillator clock frequency (24 MHz max.)
fIM:	Middle-speed on-chip oscillator clock frequency (4 MHz max.)
fEX:	External main system clock frequency
fMX:	High-speed system clock frequency
fMAIN:	Main system clock frequency
fXT:	XT1 clock oscillation frequency
fSUB:	Subsystem clock frequency
fCLK:	CPU/peripheral hardware clock frequency
fIL:	Low-speed on-chip oscillator clock frequency
fOCO:	Main on-chip oscillator clock frequency (fIH or fIM)

6.3 Registers Controlling Clock Generator

The following registers are used to control the clock generator.

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

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

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 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 bit while f_{IH} is selected as f_{CLK} after a reset ends (before f_{CLK} is switched to f_{MX}).

Caution 5. Although the maximum system clock frequency is 24 MHz, the maximum frequency of the X1 oscillator is 20 MHz.

Remark fx: 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.

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 12-bit interval timer, clock output/buzzer output, 8-bit interval timer, and watchdog timer) is also changed at the same time. Consequently, stop each peripheral function when changing the CPU/peripheral hardware clock.

Caution 3. If the subsystem clock is used as the peripheral hardware clock, the operations of the A/D converter and IICA are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 35 or CHAPTER 36 ELECTRICAL SPECIFICATIONS.

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

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})

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, and middle-speed on-chip oscillator 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	1	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	

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

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

Caution 5. 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		
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 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.

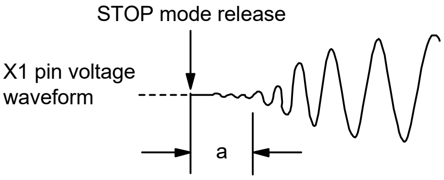
Caution 1. After the above time has elapsed, the bits are set to 1 in order from the MOST8 bit and remain 1.

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 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 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

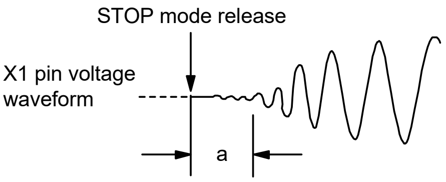
Caution 1. Change the setting of the OSTS register before setting the MSTOP bit of the clock operation status control register (CSC) to 0.

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 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 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 low-speed on-chip oscillator clock
0	Do not select low-speed on-chip oscillator clock
1	Select low-speed on-chip oscillator clock

Caution 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.

- A/D converter
- D/A converter
- PGA0
- Serial array unit 0
- IICA0, 1
- Timer KB0
- Timer array unit 0
- Comparator
- DTC
- 12-bit interval timer
- 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	0	IICA1EN	ADCEN	IICA0EN	0	SAU0EN	0	TAU0EN

IICA1EN	Control of IICA1 input clock supply
0	Stops input clock supply. • SFR used by IICA1 cannot be written. The read value is 0H. However, the SFR is not initialized. [Note 1]
1	Enables input clock supply. • SFR used by IICA1 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 2]
1	Enables input clock supply. • SFR used by the A/D converter can be read and written.

IICA0EN	Control of IICA0 input clock supply
0	Stops input clock supply. • SFR used by IICA0 cannot be written. The read value is 0H. However, the SFR is not initialized. [Note 3]
1	Enables input clock supply. • SFR used by IICA0 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 4]
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 5]
1	Enables input clock supply. • SFR used by timer array unit 0 can be read and written.

Note 1. To initialize IICA1EN and the SFR used by IICA1EN, use bit 6 (IICA1RES) of PRR0.

Note 2. To initialize the A/D converter and the SFR used by the A/D converter, use bit 5 (ADCRES) of PRR0.

Note 3. To initialize IICA0EN and the SFR used by IICA0EN, use bit 4 (IICA0RES) of PRR0.

Note 4. To initialize serial array unit 0 and the SFR used by serial array unit 0, use bit 2 (SAU0RES) of PRR0.

Note 5. 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, and 7

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.

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	DACEN	0	CMPEN	0	DTCEN	PGA0EN	0	0

DACEN	Control of D/A converter input clock supply
0	Stops input clock supply. • SFR used by the D/A 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 D/A converter can be read and written.

CMPEN	Control of comparator input clock supply
0	Stops input clock supply. • SFR used by the comparator cannot be written. The read value is 0H. However, the SFR is not initialized. [Note 2]
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.

PGA0EN	Control of comparator input clock supply
0	Stops input clock supply. • SFR used by the PGA cannot be written. The read value is 0H. However, the SFR is not initialized. [Note 3]
1	Enables input clock supply. • SFR used by comparator can be read and written.

Note 1. To initialize DACEN and the SFR used by DACEN, use bit 7 (DACRES) of PRR1.

Note 2. To initialize the comparator and the SFR used by the comparator, use bit 5 (CMPRES) of PRR1.

Note 3. To initialize PGA0EN and the SFR used by PGA0EN, use bit 2 (PGA0RES) of PRR1.

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

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	0	DOCEN	0	0	0	0	TKB0EN

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.

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.

TKB0EN	Control of timer KB0 input clock supply
0	Stops input clock supply. • SFR used by the timer KB0 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 KB0 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.

Note 3. To initialize TKB0EN and the SFR used by TKB0EN, use bit 0 (TKB0RES) of PRR2.

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

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.

6.3.8 Operation speed mode control register (OSMC)

The OSMC register can be used to select the operation clock of 12-bit interval timer, 8-bit interval timer, and clock output/buzzer output controller.

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

Reset input sets this register to undefined.

Figure 6 - 11 Format of Operation speed mode control register (OSMC)

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

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

WUTMMCK0	Selection of operation clock for 12-bit interval timer[Note 2], 8-bit interval timer, and clock output/buzzer output controller
0	Do not select low-speed on-chip oscillator clock
1	Select low-speed on-chip oscillator clock

Note 1. Be sure to set bits 0 to 3, 5 to 7 to 0.

Note 2. Operation clock of 12-bit interval timer is selected by timer clock select register 3 (TPS3) and OSMC.WUTMMCK0 (see **8.3.4 Timer clock select register 3 (TPS3)**).

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 FRQSEL4, 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			
			FRQSEL4 = 0		FRQSEL4 = 1	
			FRQSEL3 = 0	FRQSEL3 = 1	FRQSEL3 = 0	FRQSEL3 = 1
0	0	0	f_{IH} = 24 MHz	Setting prohibited	f_{IH} = 24 MHz f_{HOCO} = 48 MHz	Setting prohibited
0	0	1	f_{IH} = 12 MHz	f_{IH} = 16 MHz	f_{IH} = 12 MHz f_{HOCO} = 24 MHz	Setting prohibited
0	1	0	f_{IH} = 6 MHz	f_{IH} = 8 MHz	f_{IH} = 6 MHz f_{HOCO} = 12 MHz	Setting prohibited
0	1	1	f_{IH} = 3 MHz	f_{IH} = 4 MHz	f_{IH} = 3 MHz f_{HOCO} = 6 MHz	Setting prohibited
1	0	0	Setting prohibited	f_{IH} = 2 MHz	Setting prohibited	Setting prohibited
1	0	1	Setting prohibited	f_{IH} = 1 MHz	Setting prohibited	Setting prohibited
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 5.5 V
1	0	LS (low-speed main) mode	1 to 8 MHz	1.8 to 5.5 V
1	1	HS (high-speed main) mode	1 to 16 MHz	2.4 to 5.5 V
			1 to 24 MHz	2.7 to 5.5 V
Other than above		Setting prohibited		

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

6.4 System Clock Oscillator

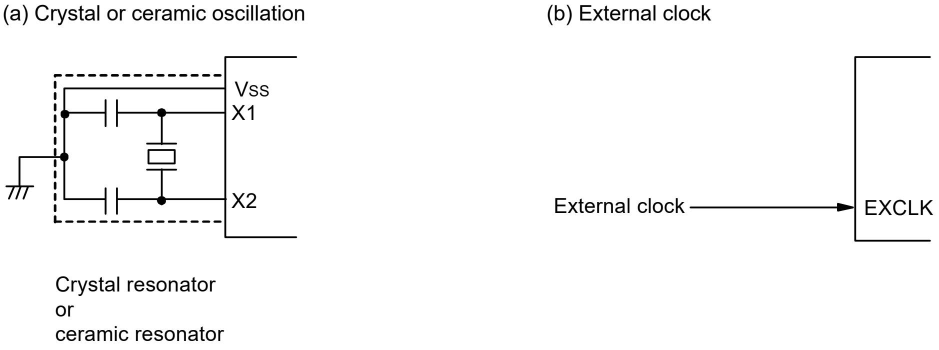
6.4.1 X1 oscillator

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 - 15 shows an example of the external circuit of the X1 oscillator.

Figure 6 - 15 Example of External Circuit of X1 Oscillator

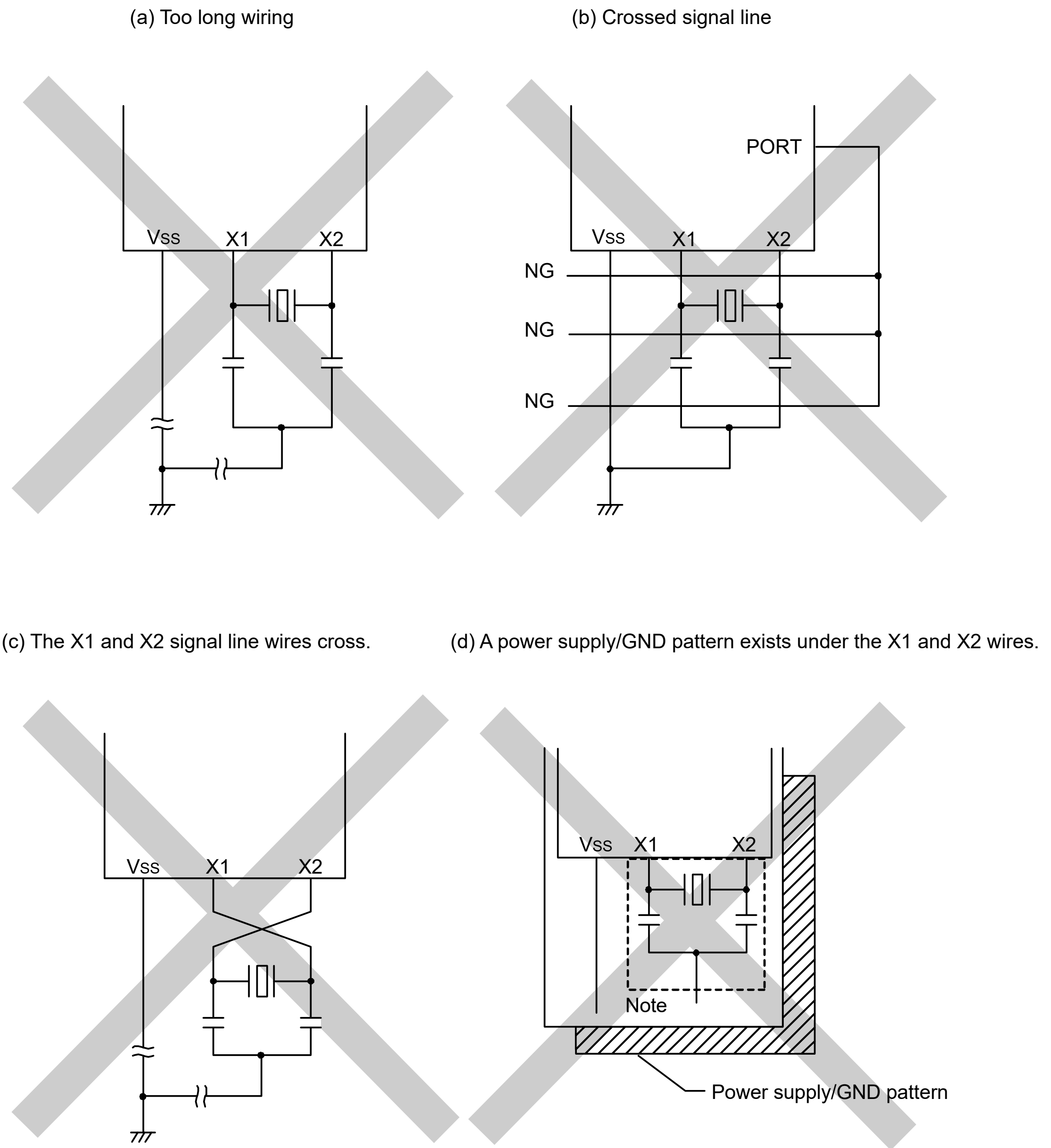


Caution When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the Figures 6 - 15 to avoid an adverse effect from wiring capacitance.

- **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.**

Figure 6 - 16 shows examples of incorrect resonator connection.

Figure 6 - 16 Examples of Incorrect Resonator Connection (1/2)



(a) Too long wiring

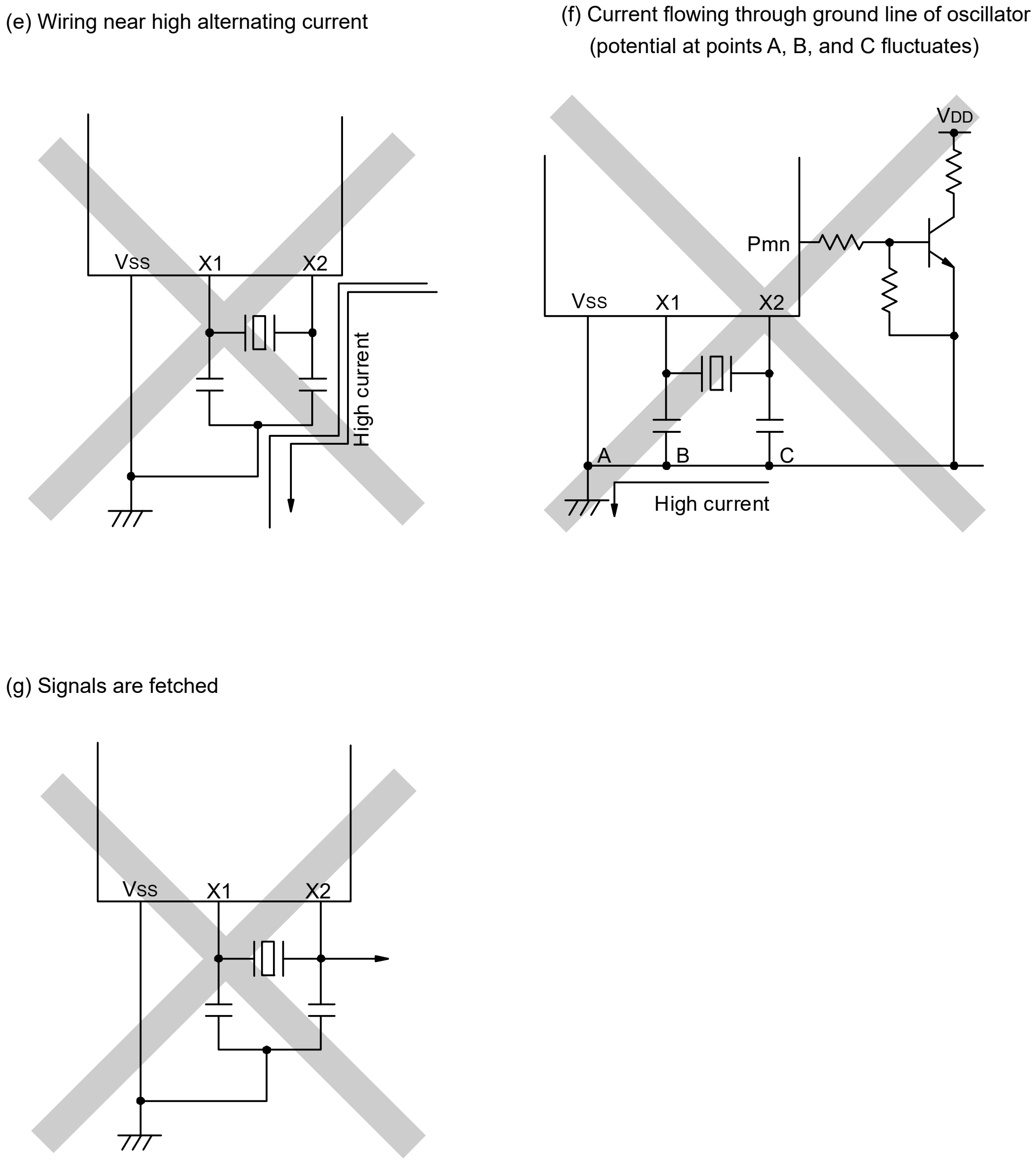
(b) Crossed signal line

(c) The X1 and X2 signal line wires cross.

(d) A power supply/GND pattern exists under the X1 and X2 wires.

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

Figure 6 - 16 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

6.4.2 High-speed on-chip oscillator

The high-speed on-chip oscillator is incorporated in the RL78/G11. 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.

6.4.3 Middle-speed on-chip oscillator

The middle-speed on-chip oscillator is incorporated in the RL78/G11. Oscillation can be controlled by bit 1 (MIOEN) of the clock operation status control register (CSC).

6.4.4 Low-speed on-chip oscillator

The low-speed on-chip oscillator is incorporated in the RL78/G11.

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 operation speed 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

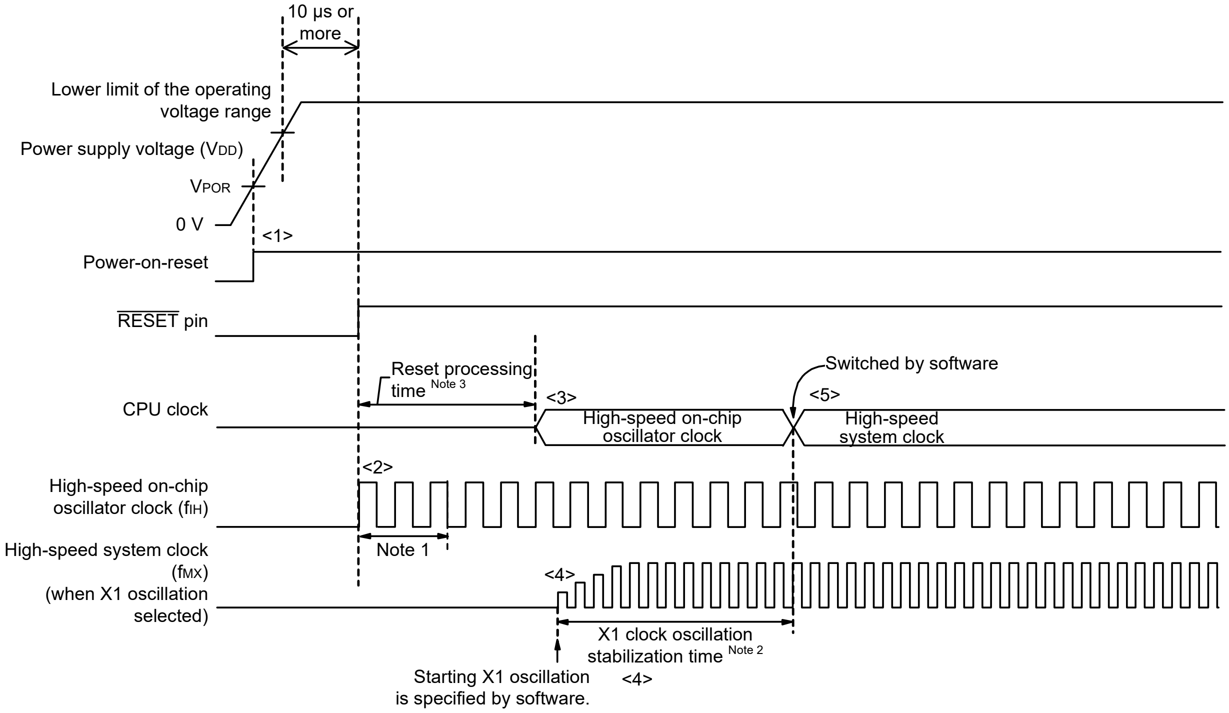
The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode (see **Figure 6 - 1**).

• 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}
• Low-speed on-chip oscillator clock	f_{IL}
• CPU/peripheral hardware clock	f_{CLK}

The CPU starts operation when the high-speed on-chip oscillator starts outputting after a reset release in the RL78/G11.

When the power supply voltage is turned on, the clock generator operation is shown in Figure 6 - 17.

Figure 6 - 17 Clock Generator Operation When Power Supply Voltage Is Turned On


10 μs or more
Lower limit of the operating voltage range
Power supply voltage (VDD)
VPOR
0 V
<1>
Power-on-reset
RESET pin
Reset processing time Note 3
<3>
Switched by software
<5>
CPU clock
High-speed on-chip oscillator clock
High-speed system clock
<2>
High-speed on-chip oscillator clock (fIH)
Note 1
High-speed system clock (fMX) (when X1 oscillation selected)
<4>
X1 clock oscillation stabilization time Note 2
<4>
Starting X1 oscillation is specified by software.

<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 **35.4 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 clock via software (see **6.6.2 Example of setting X1 oscillation clock**).

<5> When switching the CPU clock to the X1 clock, wait for the clock oscillation to stabilize, and then set switching via software (see **6.6.2 Example of setting X1 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 26 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 48, 24, 16, 12, 8, 6, 4, 3, 2, and 1 MHz by using FRQSEL0 to FRQSEL4 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: 00C2H

	7	6	5	4	3	2	1	0
Option byte (000C2H)	CMODE1 0/1	CMODE0 0/1	 1	FRQSEL4 0/1	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 5.5 V @ 1 MHz to 4 MHz
1	0	LS (low-speed main) mode	V_{DD} = 1.8 V to 5.5 V @ 1 MHz to 8 MHz
1	1	HS (high-speed main) mode	V_{DD} = 2.4 V to 5.5 V @ 1 MHz to 16 MHz V_{DD} = 2.7 V to 5.5 V @ 1 MHz to 24 MHz
Other than above		Setting prohibited	

FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator	
					f_{HOCO}	f_{IH}
1	0	0	0	0	48 MHz	24 MHz
0	0	0	0	0	24 MHz	24 MHz
0	1	0	0	1	16 MHz	16 MHz
0	0	0	0	1	12 MHz	12 MHz
0	1	0	1	0	8 MHz	8 MHz
0	0	0	1	0	6 MHz	6 MHz
0	1	0	1	1	4 MHz	4 MHz
0	0	0	1	1	3 MHz	3 MHz
0	1	1	0	0	2 MHz	2 MHz
0	1	1	0	1	1 MHz	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			
			FRQSEL4 = 0		FRQSEL4 = 1	
			FRQSEL3 = 0	FRQSEL3 = 1	FRQSEL3 = 0	FRQSEL3 = 1
0	0	0	f_{IH} = 24 MHz	Setting prohibited	f_{IH} = 24 MHz f_{HOCO} = 48 MHz	Setting prohibited
0	0	1	f_{IH} = 12 MHz	f_{IH} = 16 MHz	f_{IH} = 12 MHz f_{HOCO} = 24 MHz	Setting prohibited
0	1	0	f_{IH} = 6 MHz	f_{IH} = 8 MHz	f_{IH} = 6 MHz f_{HOCO} = 12 MHz	Setting prohibited
0	1	1	f_{IH} = 3 MHz	f_{IH} = 4 MHz	f_{IH} = 3 MHz f_{HOCO} = 6 MHz	Setting prohibited
1	0	0	Setting prohibited	f_{IH} = 2 MHz	Setting prohibited	Setting prohibited
1	0	1	Setting prohibited	f_{IH} = 1 MHz	Setting prohibited	Setting prohibited
Other than above			Setting prohibited			

6.6.2 Example of setting X1 oscillation clock

After a reset release, the CPU/peripheral hardware clock (fCLK) 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 fCLK 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	0	0	0	0	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	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 5.5 V
1	0	LS (low-speed main) mode	1 to 8 MHz	1.8 to 5.5 V
1	1	HS (high-speed main) mode	1 to 16 MHz	2.4 to 5.5 V
			1 to 24 MHz	2.7 to 5.5 V
Other than above		Setting prohibited		

6.6.3 CPU clock status transition diagram

Figure 6 - 18 shows the CPU clock status transition diagram of this product.

Figure 6 - 18 CPU Clock Status Transition Diagram

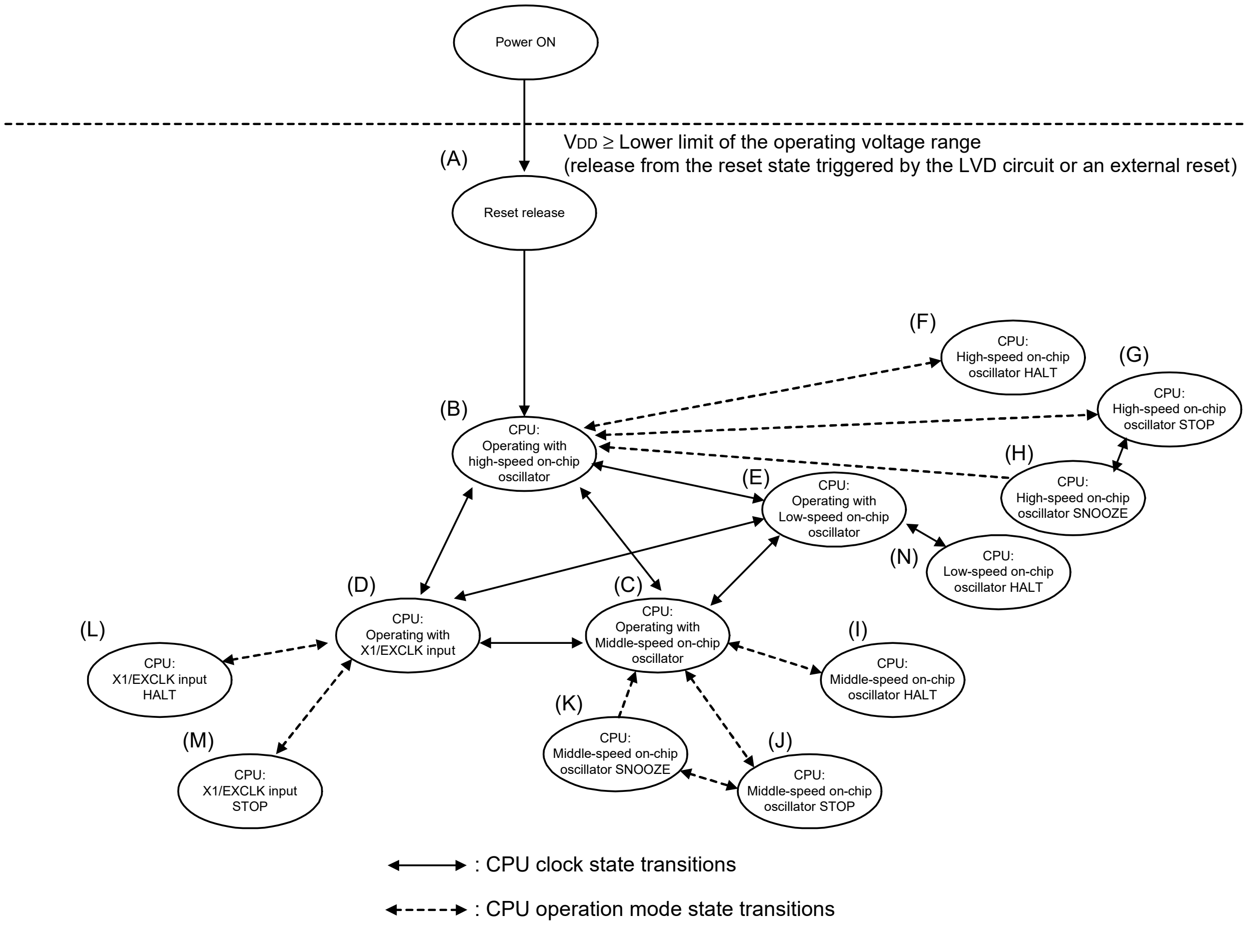

Power ON
V_{DD} ≥ Lower limit of the operating voltage range
(release from the reset state triggered by the LVD circuit or an external reset)
(A)
Reset release
(B)
CPU: Operating with high-speed on-chip oscillator
(F)
CPU: High-speed on-chip oscillator HALT
(G)
CPU: High-speed on-chip oscillator STOP
(H)
CPU: High-speed on-chip oscillator SNOOZE
(E)
CPU: Operating with Low-speed on-chip oscillator
(N)
CPU: Low-speed on-chip oscillator HALT
(D)
CPU: Operating with X1/EXCLK input
(C)
CPU: Operating with Middle-speed on-chip oscillator
(L)
CPU: X1/EXCLK input HALT
(M)
CPU: X1/EXCLK input STOP
(I)
CPU: Middle-speed on-chip oscillator HALT
(K)
CPU: Middle-speed on-chip oscillator SNOOZE
(J)
CPU: Middle-speed on-chip oscillator STOP
: CPU clock state transitions
: CPU operation mode state transitions

Table 6 - 3 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)

(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)

(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 (N) in Table 6 - 3 correspond to (A) to (N) in Figure 6 - 18.

Table 6 - 3 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)

(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

Unnecessary if these registers are already set

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 35 or CHAPTER 36 ELECTRICAL SPECIFICATIONS).**

Remark (A) to (N) in Table 6 - 3 correspond to (A) to (N) in Figure 6 - 18.

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

(5) Changing to low-speed on-chip oscillator 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	CKSEL	Oscillation accuracy stabilization time	CKC Register
	SELLOSC		CSS
Changing to low-speed on-chip oscillator	1	210 μs	1

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

Remark 1. ×: don't care

Remark 2. (A) to (N) in Table 6 - 3 correspond to (A) to (N) in Figure 6 - 18.

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

(6) Changing from CPU operation mode (B), (C), (D), and (E) to HALT mode (F), (I), (L) and (N)
Target state transition: (B) → (F), (C) → (I), (D) → (L), (E) → (N)

Mode After Change	Setting
HALT mode	Executing HALT instruction

(7) Changing from CPU operation mode (B), (C), and (D) to STOP mode (G), (J), and (M)
Target state transition: (B) → (G), (C) → (J), (D) → (M)

(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

(8) Changing from STOP mode (G) and (J), to SNOOZE mode (H) and (K)

For details about the setting for switching from the STOP mode to the SNOOZE mode, see **13.5.7 SNOOZE mode function**, **13.7.3 SNOOZE mode function**, and **16.8 SNOOZE Mode Function**.

Remark (A) to (N) in Table 6 - 3 correspond to (A) to (N) in Figure 6 - 18.

6.6.4 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 - 4 Changing CPU Clock (1/2)

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 • MIOEN = 1	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	
	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	
	Low-speed on-chip oscillator clock	Selection of low-speed on-chip oscillator • SELLOSC = 1	
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	—
	Low-speed on-chip oscillator clock	Selection of low-speed on-chip oscillator • SELLOSC = 1	X1 oscillation can be stopped (MSTOP = 1) after checking that the CPU clock is changed.

Table 6 - 4 Changing CPU Clock (2/2)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
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	—
	Low-speed on-chip oscillator clock	Selection of low-speed on-chip oscillator • SELLOSC = 1	External main system clock input can be disabled (MSTOP = 1) after checking that the CPU clock is changed.
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	Operating current can be reduced by stopping the low-speed on-chip oscillator clock (SELLOSC = 0) 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	
	Low-speed on-chip oscillator clock	Transition not possible	—

6.6.5 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 - 5** to **6 - 8**).

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 - 5 Maximum Time Required for Main System Clock Switchover

Clock A	Switching directions	Clock B	Remark
f_{OCO}	↔	f_{MX}	See **Table 6 - 6**
f_{IH}	↔	f_{IM}	See **Table 6 - 7**
f_{MAIN}	↔	f_{SUB}	See **Table 6 - 8**

Table 6 - 6 Maximum Number of Clocks Required for f_{OCO} ↔ 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 - 7 Maximum Number of Clocks Required for f_{IH} ↔ 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 - 8 Maximum Number of Clocks Required for f_{MAIN} ↔ f_{SUB}

Set Value Before Switchover	Set Value After Switchover	
CSS	CSS	
	0 ($f_{CLK} = f_{MAIN}$)	1 ($f_{CLK} = f_{SUB}$)
0 ($f_{CLK} = f_{MAIN}$)		1 + 2 f_{MAIN}/f_{SUB} clocks
1 ($f_{CLK} = f_{SUB}$)	3 clocks	

Remark 1. The number of clocks listed in Tables 6 - 6 to 6 - 8 is the number of CPU clocks before switchover.

Remark 2. Calculate the number of clocks in Tables 6 - 6 to 6 - 8 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

6.6.6 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 - 9 Conditions Before the Clock Oscillation Is Stopped and Flag Settings

Clock	Conditions Before Clock Oscillation Is Stopped (External Clock Input Disabled)	Flag Settings of SFR Register
High-speed on-chip oscillator clock	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		
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

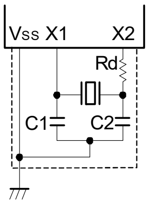
6.7 Resonator and Oscillator Constants

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 - 19 Example of X1 oscillation External Circuit



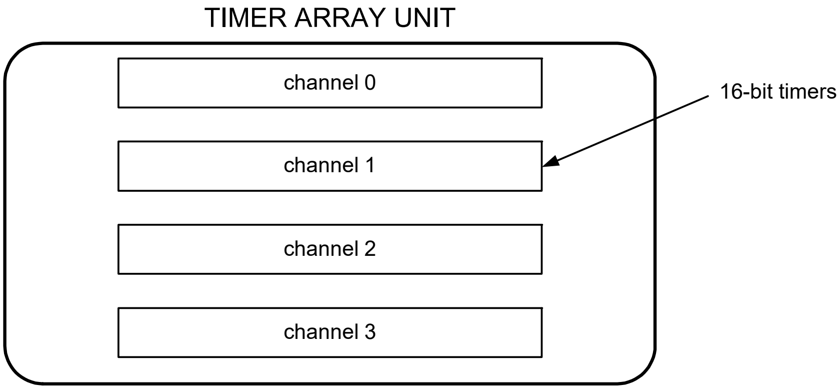
CHAPTER 7 TIMER ARRAY UNIT

The number of timer array unit channels differs depending on the product.

Channel	10-pin	16, 20, 24, 25-pin
Channel 0	—	√
Channel 1	—	√
Channel 2	√	√
Channel 3	√	√

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**) • Two-channel input with one-shot pulse output function (→ refer to **7.9.2**) • PWM output (→ refer to **7.9.3**) • Multiple PWM output (→ refer to **7.9.4**)

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

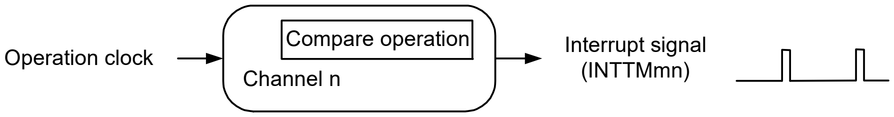
Timer array unit has the following functions.

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.

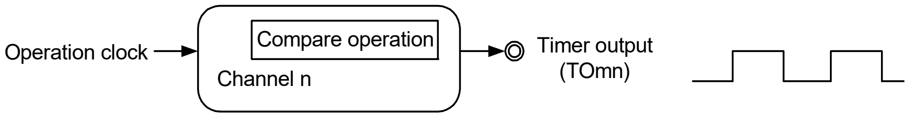
(1) Interval timer

Each timer of a unit can be used as a reference timer that generates an interrupt (INTTMmn) at fixed intervals.



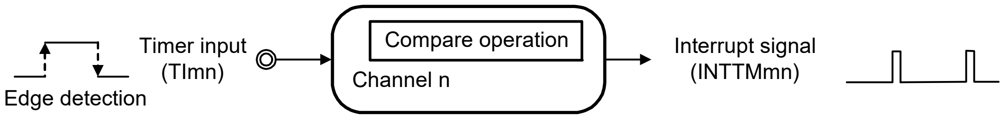
(2) Square wave output

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).



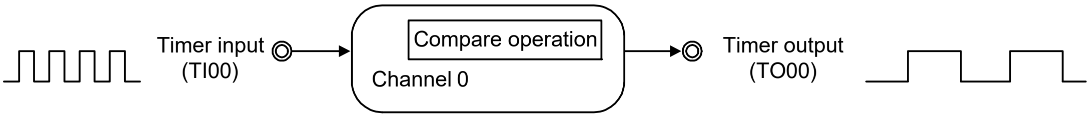
(3) External event counter

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.



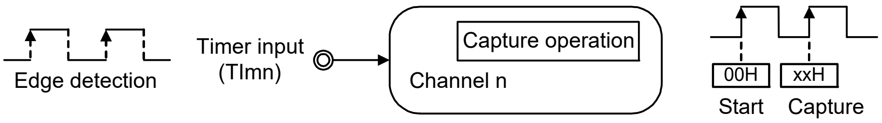
(4) Divider function (channel 0 only)

A clock input from a timer input pin (TI00) is divided and output from an output pin (TOm0).



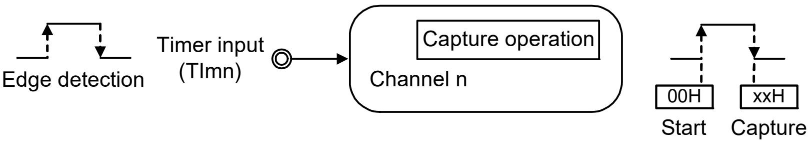
(5) Input pulse interval measurement

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.



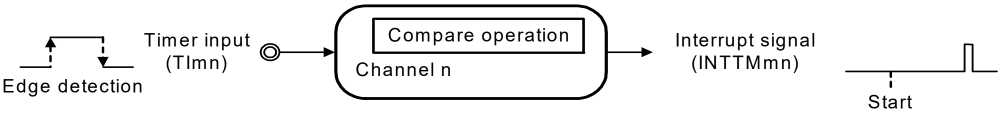
(6) Measurement of high-/low-level width of input signal

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.



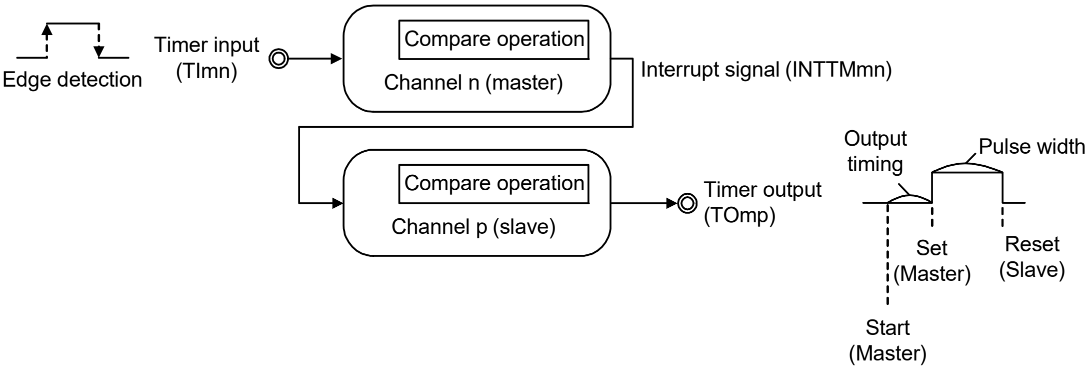
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

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.

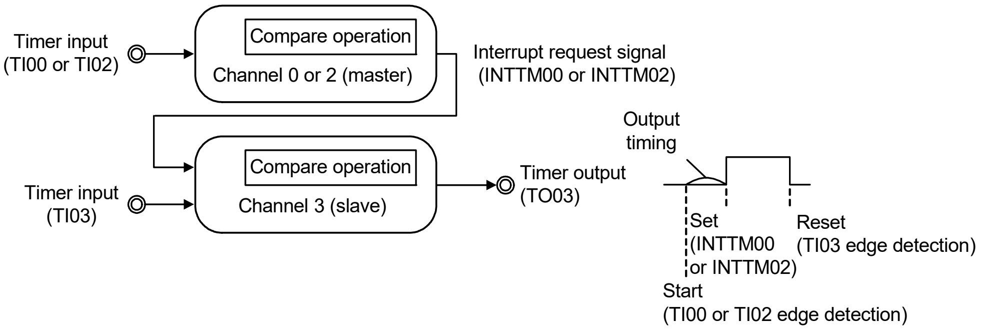


Caution For details about the rules of simultaneous channel operation function, see 7.4.1 Basic rules of simultaneous channel operation function.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2),
p: Slave channel number (p = 3)

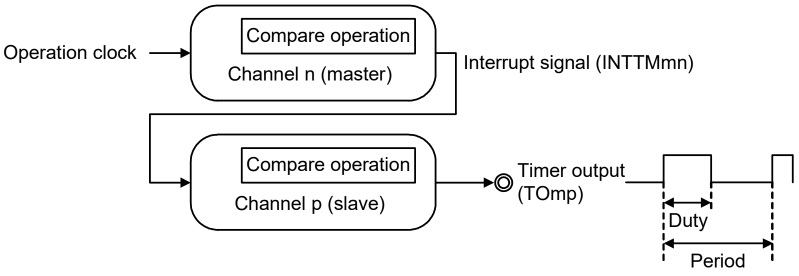
(2) Two-channel input with one-shot pulse output function

Two channels are used as a set to generate any one-shot pulse by setting or resetting the timer output pin (TO03) at a valid edge of the timer input pin (TI00 or TI02, TI03) input.



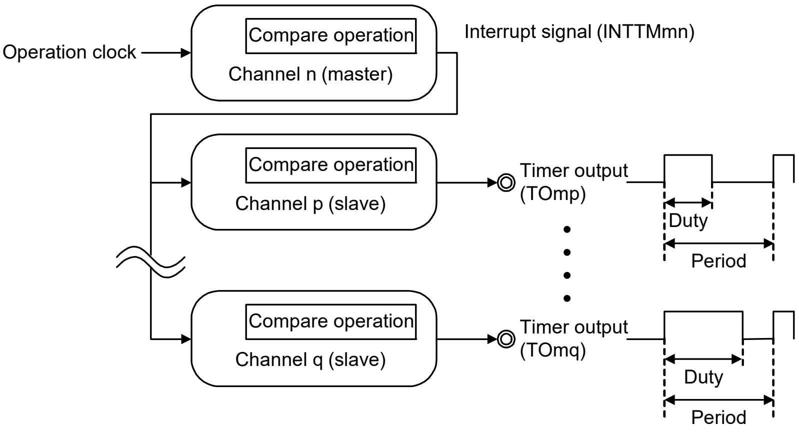
(3) PWM (Pulse Width Modulation) output

Two channels are used as a set to generate a pulse with a specified period and a specified duty factor.



(4) 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 ($n < p < q \leq 3$)

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 Count Register mn (TCRmn) Read Value in Various Operation Modes** for details.

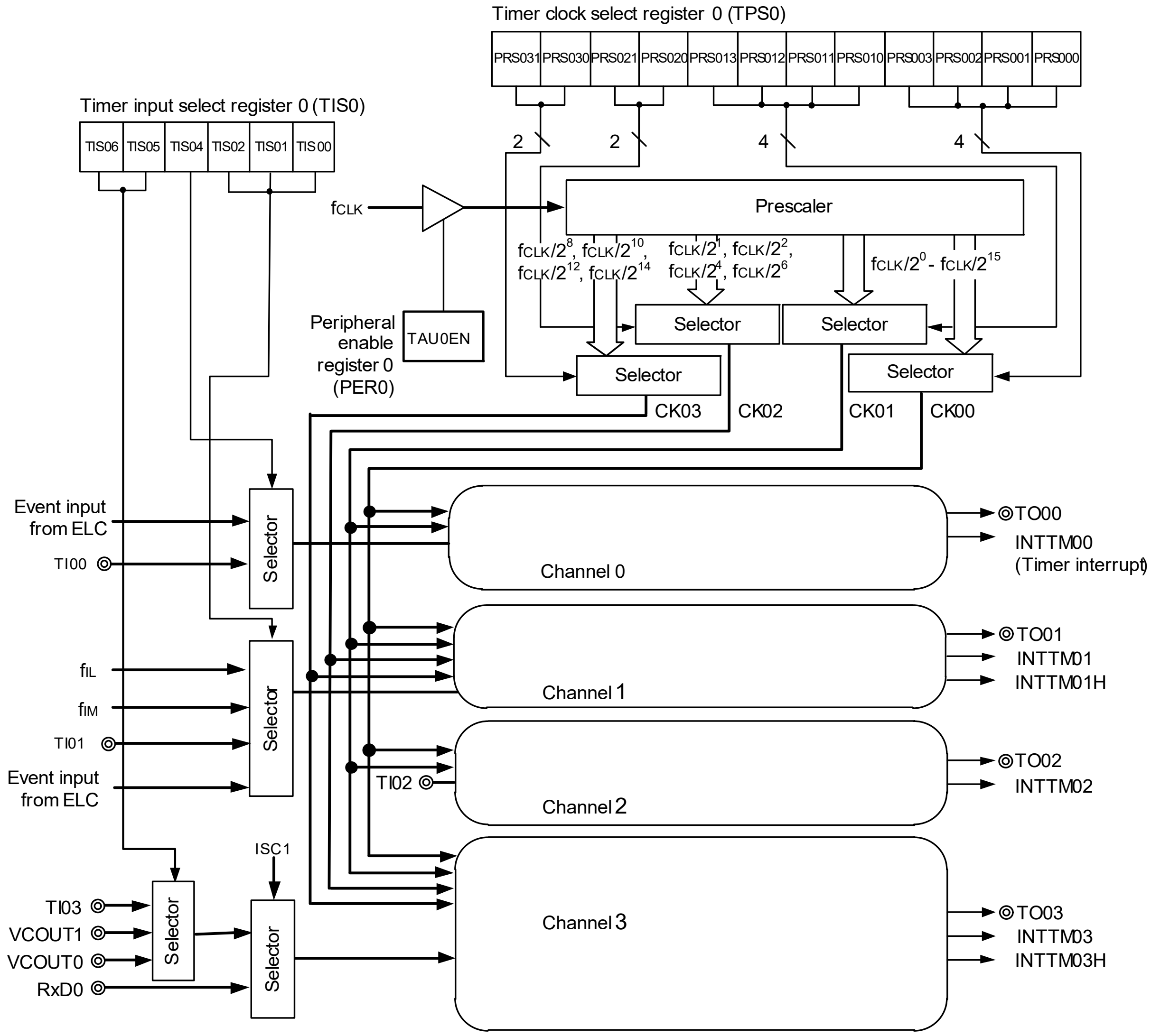
Note 2. The port mode control register (PMCxx), port mode registers (PMxx) and port registers (Pxx) to be set differ depending on the product. For details, see **4.5 Register Settings When Using Alternate Function**.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

The timer array unit has four input and output pins (TI00 to TI03 and TO00 to TO03).
When timer input and timer output are shared by the same pin, either only timer input or only timer output can be used.

Figures 7 - 1 to 7 - 5 show the block diagrams of the timer array unit.

Figure 7 - 1 Entire Configuration of Timer Array Unit 0



Remark 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

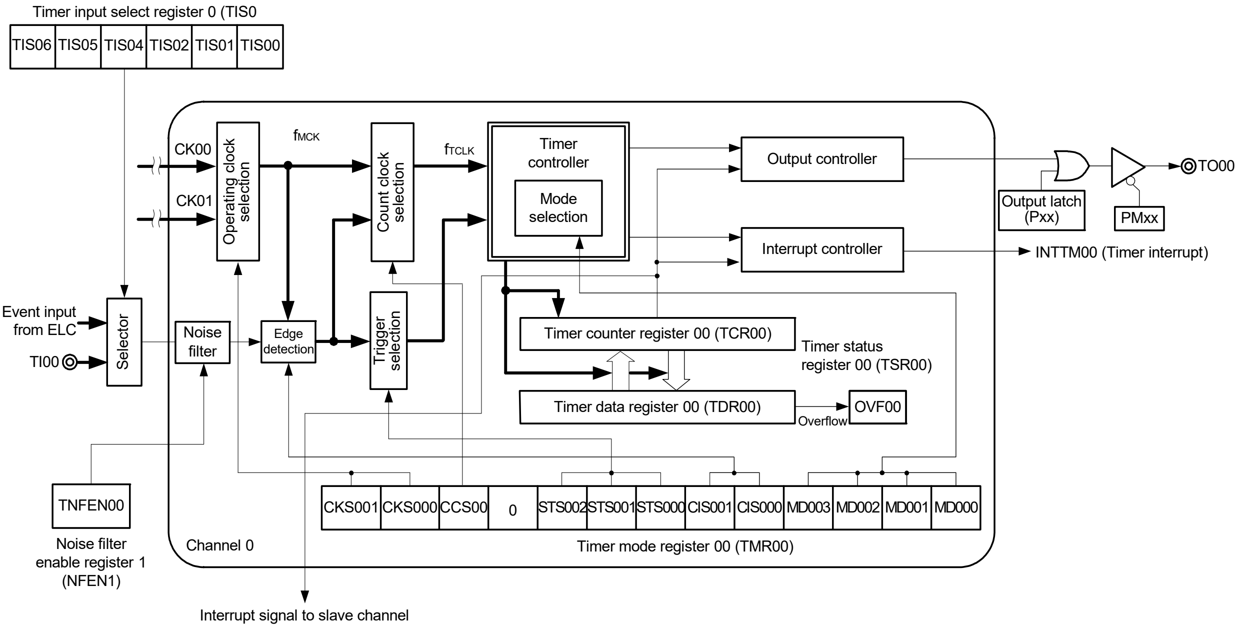


Figure 7 - 3 Internal Block Diagram of Channel 1 of Timer Array Unit 0

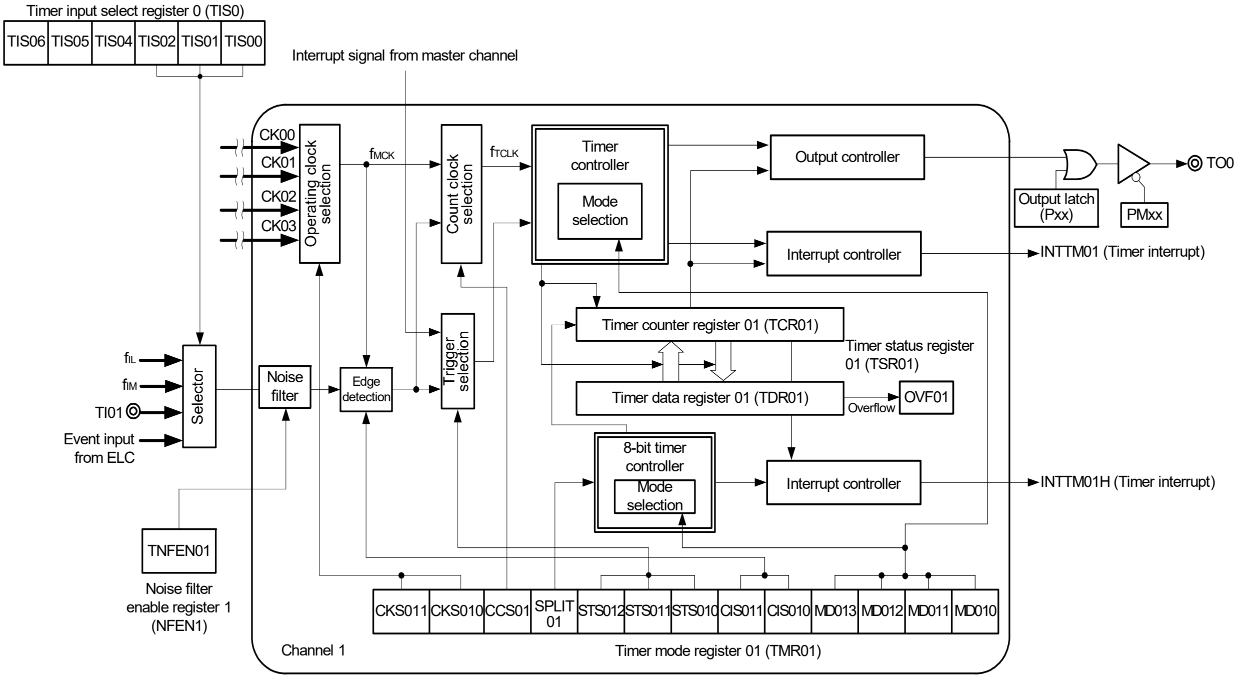


Figure 7 - 4 Internal Block Diagram of Channel 2 of Timer Array Unit 0

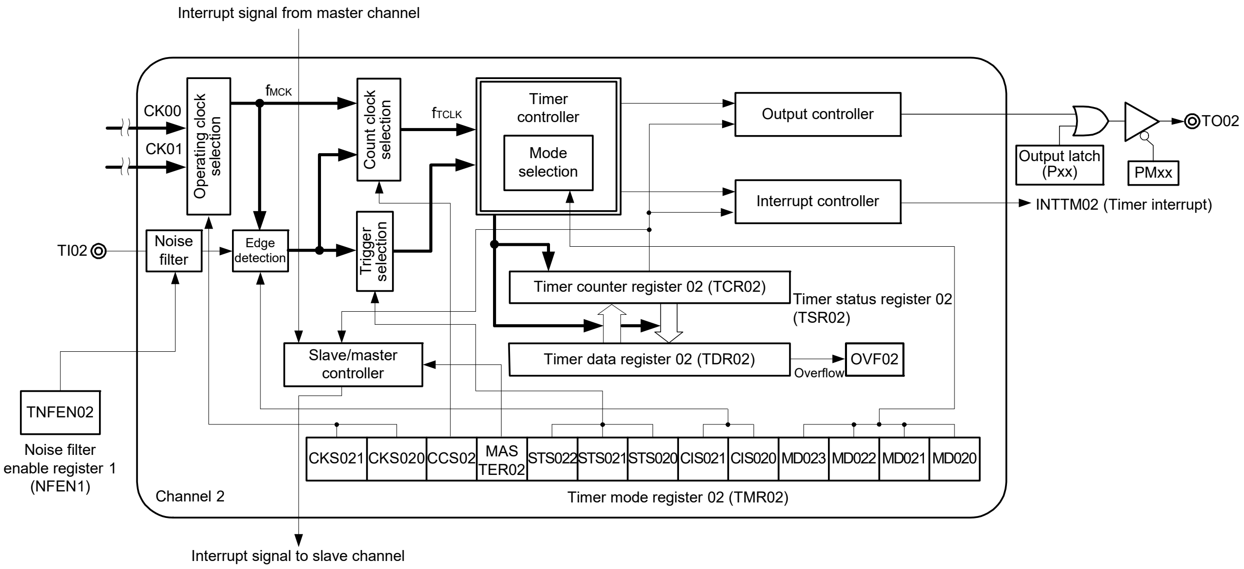
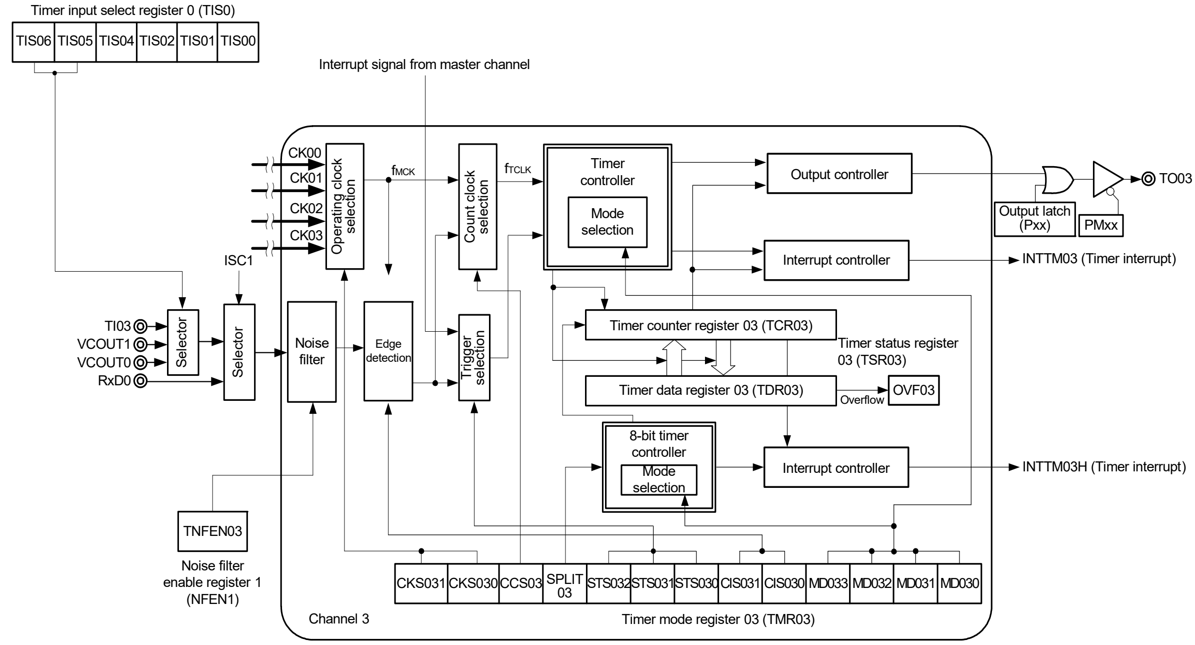


Figure 7 - 5 Internal Block Diagram of Channel 3 of Timer Array Unit 0



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 - 2 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

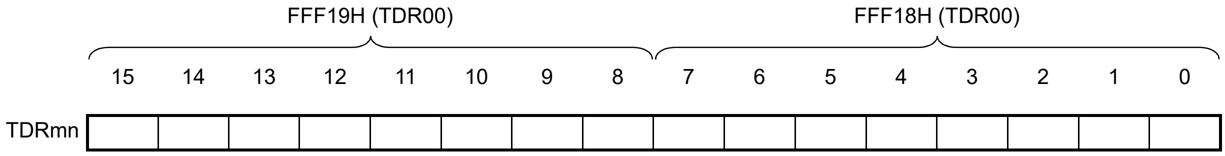
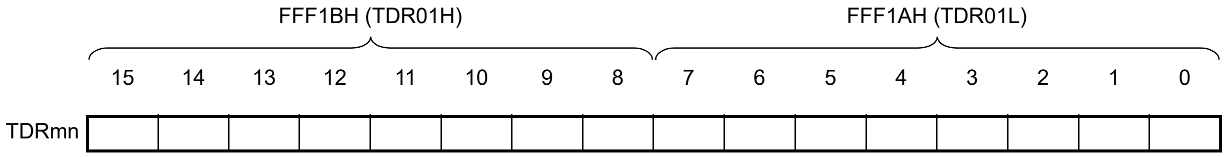


Figure 7 - 8 Format of Timer data register mn (TDRmn) (n = 1, 3)

Address: FFF1AH, FFF1BH (TDR01), FFF66H, FFF67H (TDR03) After reset: 0000H R/W



(i) When timer data register mn (TDRmn) is used as compare register

Counting down is started from the value set to the TDRmn register. When the count value reaches 0000H, an interrupt signal (INTTMmn) is generated. The TDRmn register holds its value until it is rewritten.

Caution The TDRmn register does not perform a capture operation even if a capture trigger is input, when it is set to the compare function.

(ii) When timer data register mn (TDRmn) is used as capture register

The count value of timer count register mn (TCRmn) is captured to the TDRmn register when the capture trigger is input.

A valid edge of the TImn pin can be selected as the capture trigger. This selection is made by timer mode register mn (TMRmn).

Remark m: Unit number (m = 0), 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	0	IICA1EN	ADCEN	IICA0EN	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 registers 0, 3, 5 (PMC0, PMC3, PMC5), port mode registers 0, 3, 4, 5 (PM0, PM3, PM4, PM5), and port registers 0, 3, 4, 5, 12, 13 (P0, P3, P4, P5, P12, P13)).

- **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 and 7

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	IICA1RES	ADCRES	IICA0RES	0	SAU0RES	0	TAU0RES

TAU0RES	Reset control of timer array unit 0
0	Timer array unit is released from the reset state.
1	Timer array unit is in the 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 - 11 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 - 3 can be achieved by using the interval timer function.

Table 7 - 3 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 (fMCK), 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 - 12 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 - 12 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).
1	1	0	INTTMmn of the master channel is used as a start trigger, and the valid edge of the TImp pin input of the slave channel is used as a end trigger (capture trigger).
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 - 12 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 - 12 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)	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

MDmn3	MDmn2	MDmn1	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))	MDmn0	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 - 4** 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 - 13 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 - 4 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 - 14 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.

<table>
<tr><th>TEmn</th><th>Indication of operation enable/stop status of channel n</th></tr>
<tr><td>0</td><td>Operation is stopped.</td></tr>
<tr><td>1</td><td>Operation is enabled.</td></tr>
<tr><td colspan="2">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.</td></tr>
</table>

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 - 15 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 - 5** 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 - 5** 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 - 5** 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 - 16 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. This bit is the trigger to stop operation of the lower 8-bit timer for TTm1 and TTm3 when channel 1 or 3 is in the 8-bit timer mode.

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 - 17 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	TIS06	TIS05	TIS04	0	TIS02	TIS01	TIS00

TIS06	TIS05	Selection of timer input used with channel 3
0	0	Input signal of timer input pin (TI03)
0	1	VCOUT1 output signal of comparator 1 Note 1
1	0	VCOUT0 output signal of comparator 0 Note 2
1	1	Setting prohibited

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})
Other than above			Setting prohibited

Note 1. The VCOUT1 signal cannot be output to the external pin.

Note 2. The VCOUT0 signal cannot be output to the external pin.

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_{MCK} is selected for the f_{IL} (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 - 18 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	TOEm 3	TOEm 2	TOEm 1	TOEm 0

TOE mn	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 - 19 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 - 20 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	TOLm 3	TOLm 2	TOLm 1	0

TOL mn	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 - 21 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 Input switch control register (ISC)

The ISC1 bit controls the TI03 pin input in channel 3 of the timer array unit.

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 7 - 22 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	ISC1	ISC0

ISC1	Setting of the TI03 pin input in channel 3 of the timer array unit
0	Signal specified in the TIS06 and TIS05 bits in the TIS0 register [Note]
1	RXD0 pin input signal

Note See **Figure 7 - 17 Format of Timer input select register 0 (TIS0)**.

Caution Be sure to clear bits 6 to 2 to 0.

7.3.15 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 - 23 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.

7.3.16 Registers that control port functions of timer input/output pins

Using the timer array unit requires setting of the registers that control the port functions for the port pins with which the timer array unit pin functions for the target channel are multiplexed (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**.

Using a port pin which is multiplexed with a timer output pin function (e.g. P30/TO01) for timer output requires setting the corresponding bits in the port mode control register (PMCxx), port mode register (PMxx), and port register (Pxx) to 0.

Example: When P30/TO01 is to be used 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.

Using a port pin which is multiplexed with a timer input pin function (e.g. P30/TI00) for timer input requires setting the corresponding bit in the port mode register (PMxx) to 1, and the corresponding bit in the port mode control register (PMCxx) to 0. At this time, the value of the corresponding bit in the port register (Pxx) may be 0 or 1.

Example: When P30/TI00 is to be used 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.

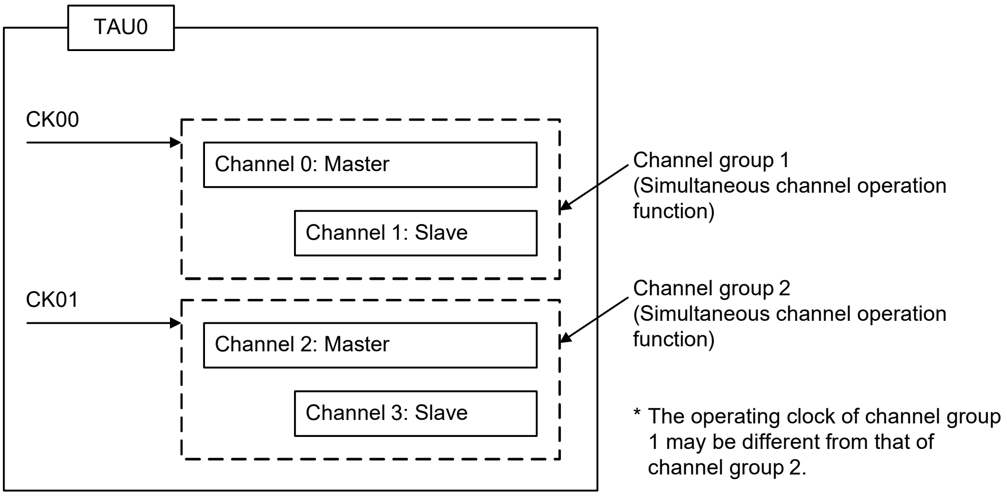
(13) CKm2/CKm3 cannot be selected while channels are operating simultaneously, because the operating clocks of master channels and slave channels have to be synchronized.

(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.

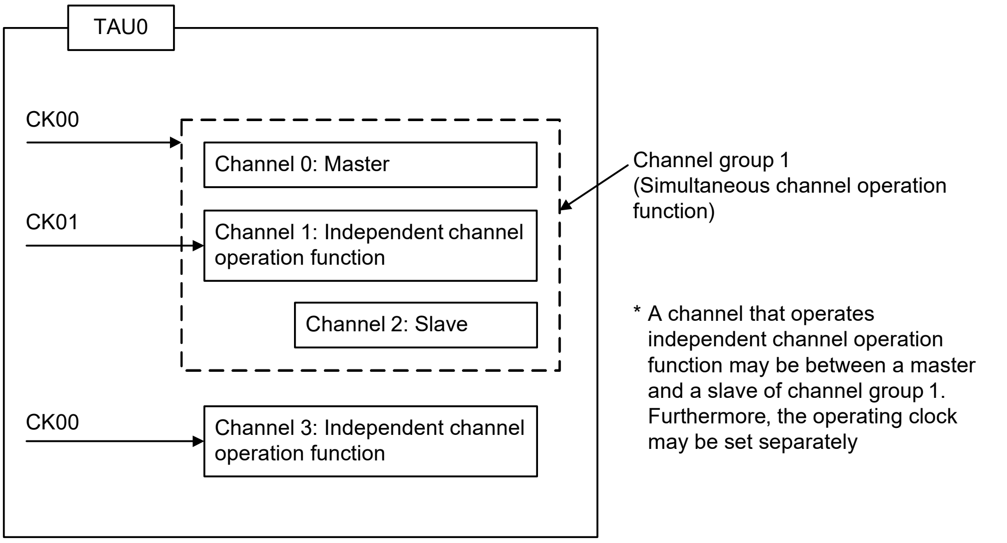
The rules of the simultaneous channel operation function are applied in a channel group (a master channel and slave channels forming one simultaneous channel operation function).
If two or more channel groups that do not operate in combination are specified, the basic rules of the simultaneous channel 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


TAU0
CK00
CK01
Channel 0: Master
Channel 1: Slave
Channel 2: Master
Channel 3: Slave
Channel group 1
(Simultaneous channel operation function)
Channel group 2
(Simultaneous channel operation function)
* The operating clock of channel group 1 may be different from that of channel group 2.

Example 2


TAU0
CK00
CK01
CK00
Channel 0: Master
Channel 1: Independent channel operation function
Channel 2: Slave
Channel 3: Independent channel operation function
Channel group 1
(Simultaneous channel operation function)
* A channel that operates independent channel operation function may be between a master and a slave of channel group 1. Furthermore, the operating clock may be set separately

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 (f_{TCLK})

The count clock (f_{TCLK}) of the timer array unit can be selected between following by CCSmn bit of timer mode register mn (TMRmn).

- Operation clock (f_{MCK}) specified by the CKSmn0 and CKSmn1 bits
- Valid edge of input signal input from the TImn pin

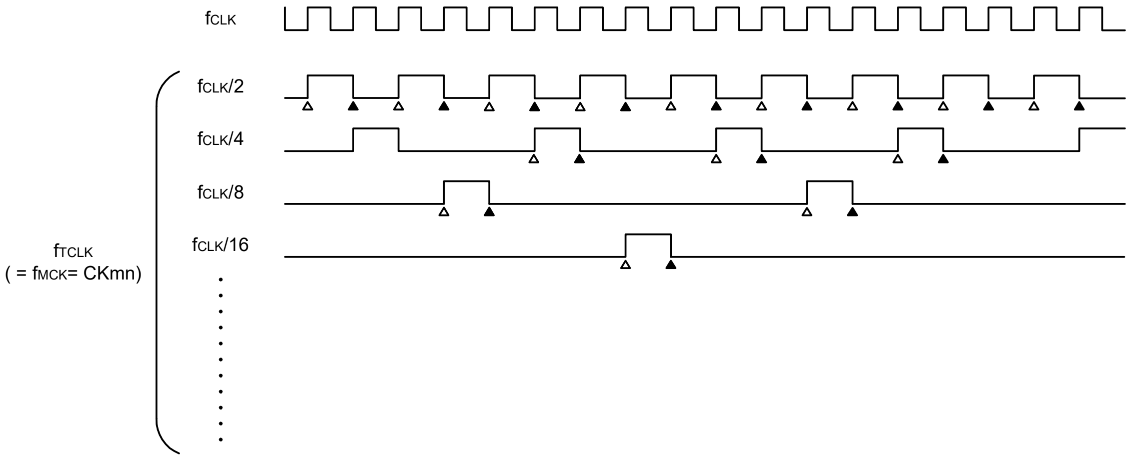
Because the timer array unit is designed to operate in synchronization with f_{CLK}, the timings of the count clock (f_{TCLK}) are shown below.

(1) When operation clock (f_{MCK}) specified by the CKSmn0 and CKSmn1 bits is selected (CCSmn = 0)

The count clock (f_{TCLK}) is between f_{CLK} to f_{CLK} /2^{15} by setting of timer clock select register m (TPSm). When a divided f_{CLK} is selected, however, the clock selected in TPSmn register, but a signal which becomes high level for one period of f_{CLK} from its rising edge. When a f_{CLK} is selected, fixed to high level.

Counting of timer count register mn (TCRmn) delayed by one period of f_{CLK} from rising edge of the count clock, because of synchronization with f_{CLK}. But, this is described as “counting at rising edge of the count clock”, as a matter of convenience.

Figure 7 - 24 Timing of f_{CLK} and count clock (f_{TCLK}) (When CCSmn = 0)



Remark 1. △ : Rising edge of the count clock

▲ : Synchronization, increment/decrement of counter

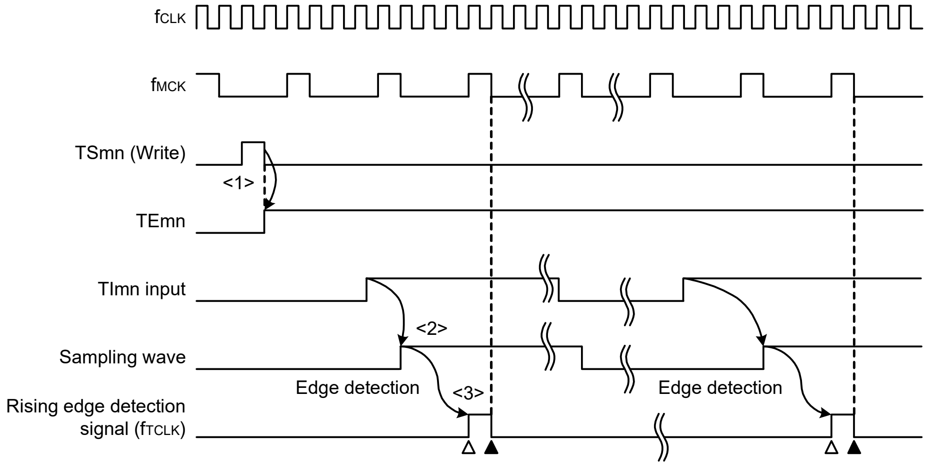
Remark 2. f_{CLK}: CPU/peripheral hardware clock

(2) When valid edge of input signal via the TImn pin is selected (CCSmn = 1)

The count clock (fTCLK) becomes the signal that detects valid edge of input signal via the TImn pin and synchronizes next rising fMCK. The count clock (fTCLK) is delayed for 1 to 2 period of fMCK 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 fCLK from rising edge of the count clock, because of synchronization with fCLK. But, this is described as “counting at valid edge of input signal via the TImn pin”, as a matter of convenience.

Figure 7 - 25 Timing of fCLK and count clock (fTCLK) (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 fMCK.
<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. fCLK: CPU/peripheral hardware clock
fMCK: 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 - 25.

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 - 5.

Table 7 - 5 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.

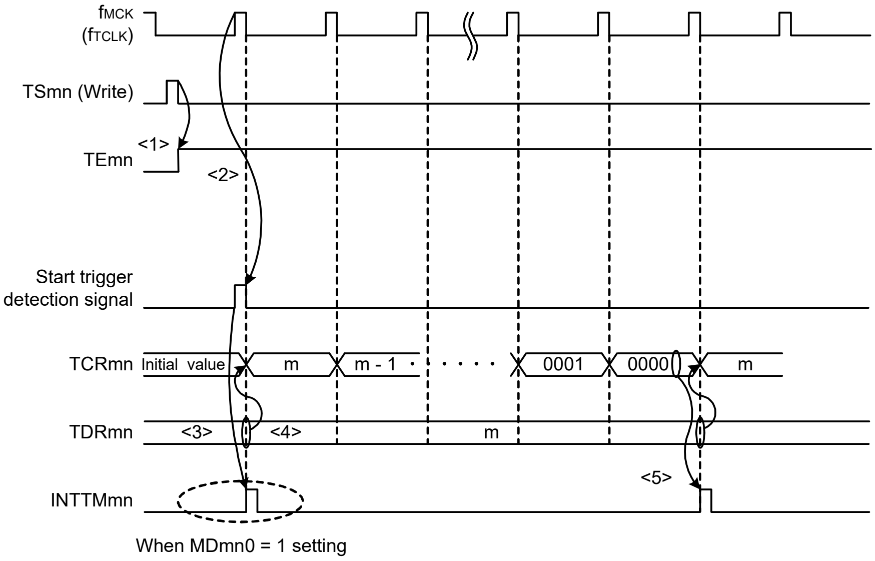
<2> A start trigger is generated at the first count clock after operation is enabled.

<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 - 26 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

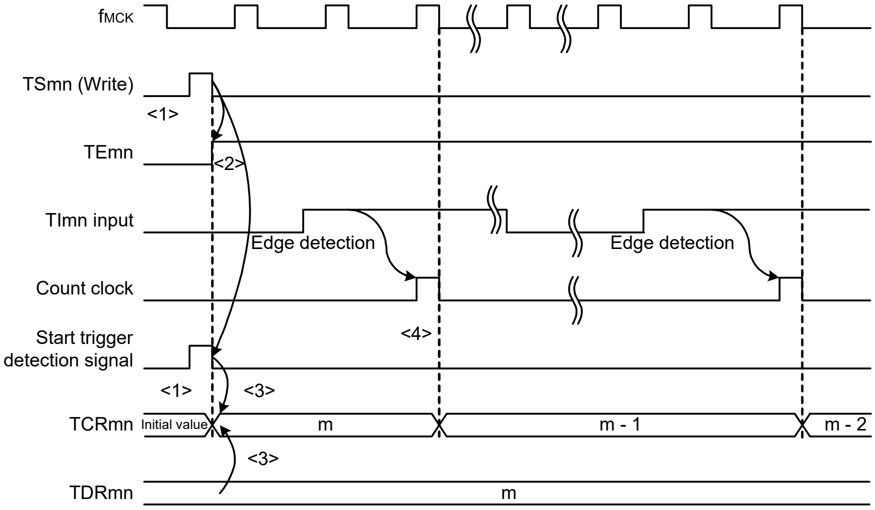
<1> Timer count register mn (TCRmn) holds its initial value while operation is stopped (TEmn = 0).

<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 - 27 Operation Timing (In Event Counter Mode)



Remark Figure 7 - 27 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).

(3) Operation of capture mode (input pulse interval measurement)

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.

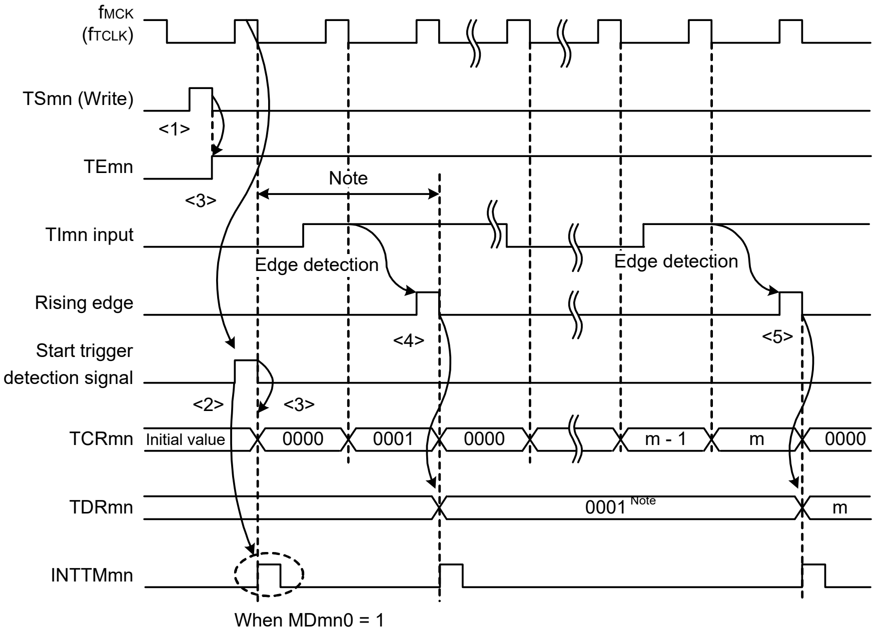
<2> Timer count register mn (TCRmn) holds the initial value until count clock generation.

<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 - 28 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 - 28 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.

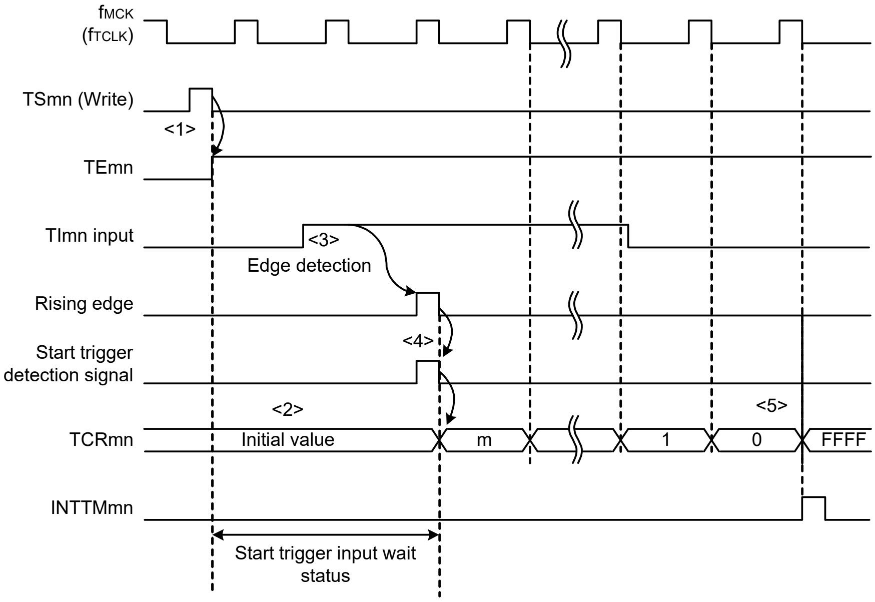
<2> Timer count register mn (TCRmn) holds the initial value until start trigger generation.

<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 - 29 Operation Timing (In One-count Mode)



Remark Figure 7 - 29 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).

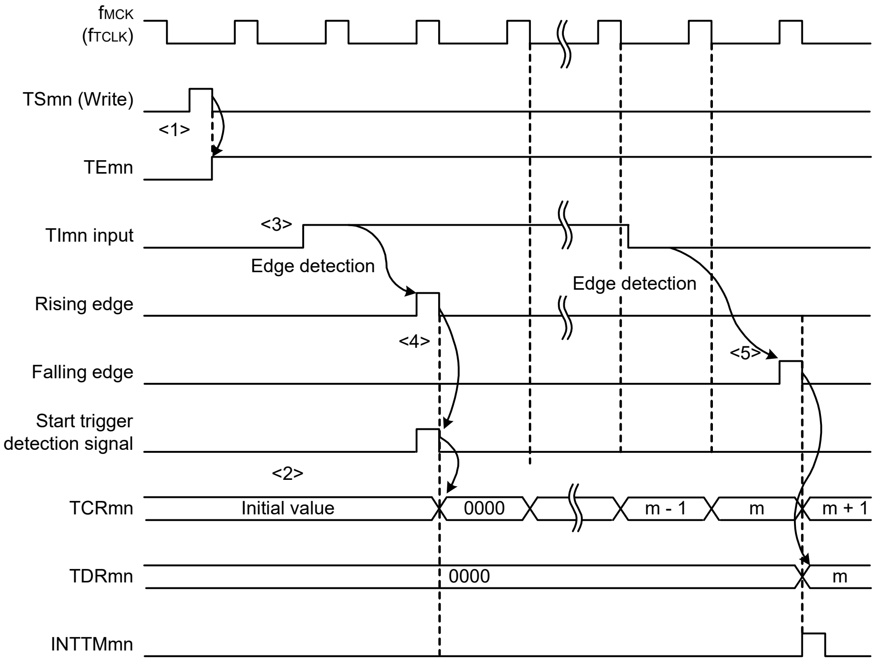
<2> Timer count register mn (TCRmn) holds the initial value until start trigger generation.

<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 - 30 Operation Timing (In Capture & One-count Mode: High-level Width Measurement)

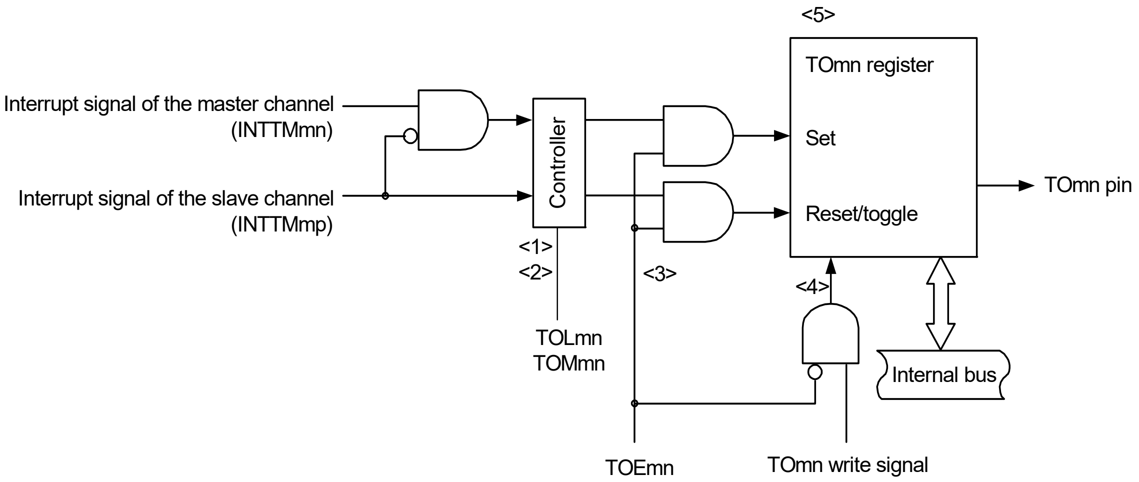


Remark Figure 7 - 30 shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 fMCK cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the 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 - 31 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.

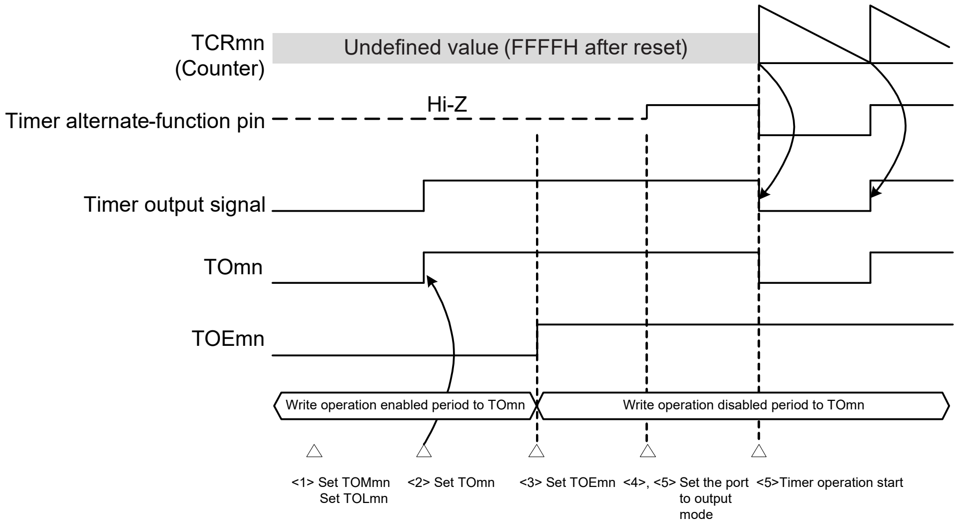
<5> The TOm register can always be read, and the TOmn pin output level can be checked.

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 - 32 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 **7.3.16 Registers that control port functions of timer input/output pins**).

<5> The port I/O setting is set to output (see **7.3.16 Registers that control port functions of timer input/output pins**).

<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)

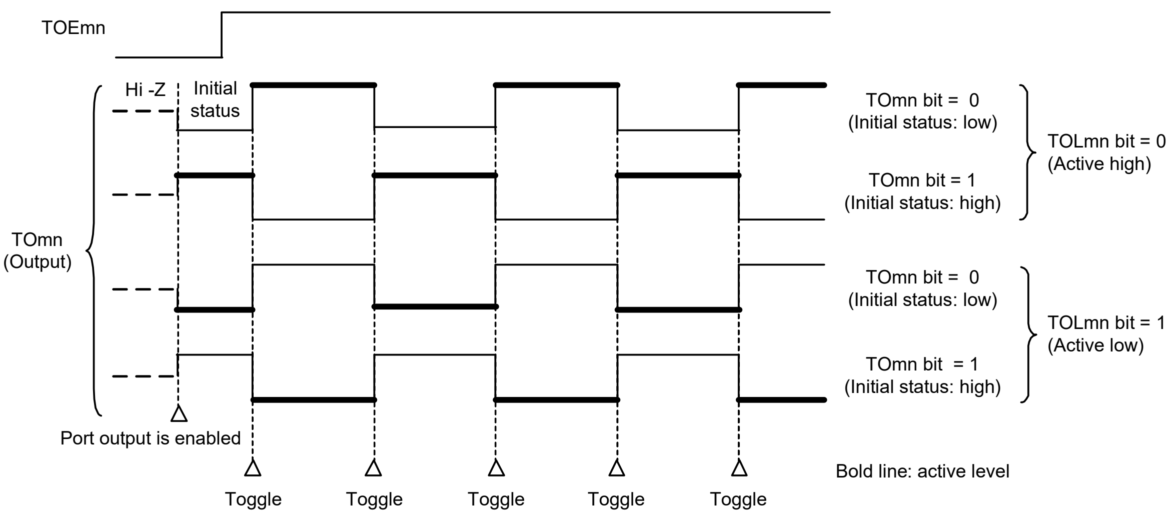
(2) Default level of TOmn pin and output level after timer operation start

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.

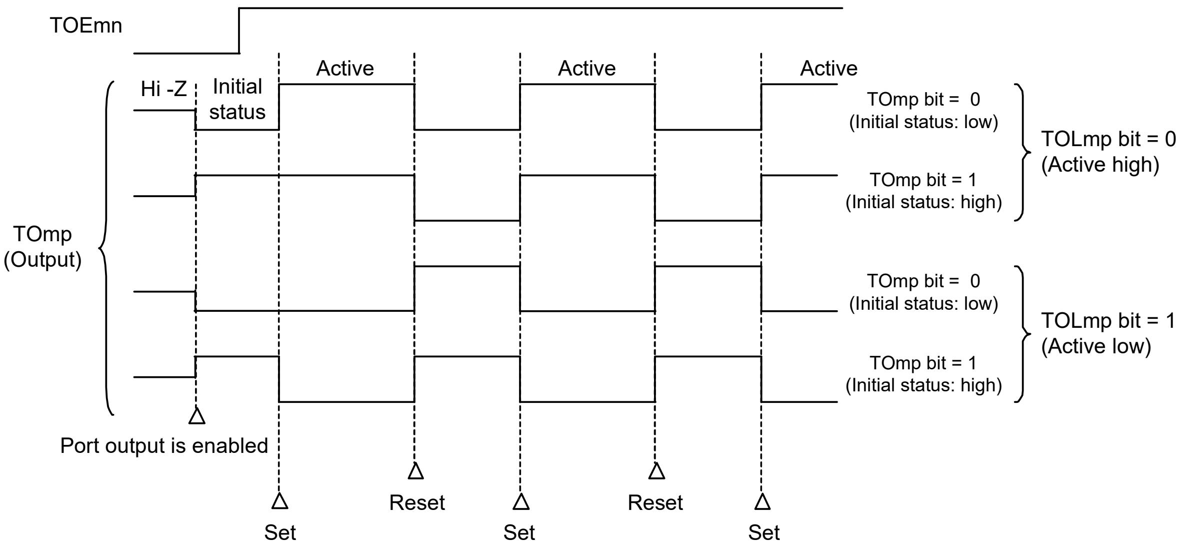
Figure 7 - 33 TOmn Pin Output Status at Toggle Output (TOMmn = 0)


TOEmn
Hi -Z
Initial status
TOmn (Output)
Port output is enabled
Toggle
Toggle
Toggle
Toggle
Toggle
TOmn bit = 0 (Initial status: low)
TOmn bit = 1 (Initial status: high)
TOLmn bit = 0 (Active high)
TOmn bit = 0 (Initial status: low)
TOmn bit = 1 (Initial status: high)
TOLmn bit = 1 (Active low)
Bold line: active level

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 - 34 TOmn Pin Output Status at PWM Output (TOMmn = 1)


TOEmn
Hi -Z
Initial status
Active
Active
Active
TOmp (Output)
TOmp bit = 0 (Initial status: low)
TOmp bit = 1 (Initial status: high)
TOLmp bit = 0 (Active high)
TOmp bit = 0 (Initial status: low)
TOmp bit = 1 (Initial status: high)
TOLmp bit = 1 (Active low)
Port output is enabled
Set
Reset
Set
Reset
Set

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)

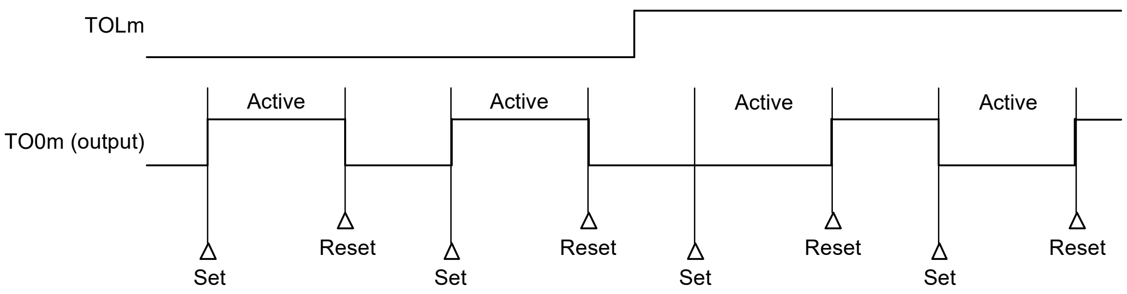
(3) Operation of TOmn pin in slave channel output mode (TOMmn = 1)

(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 - 35 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.

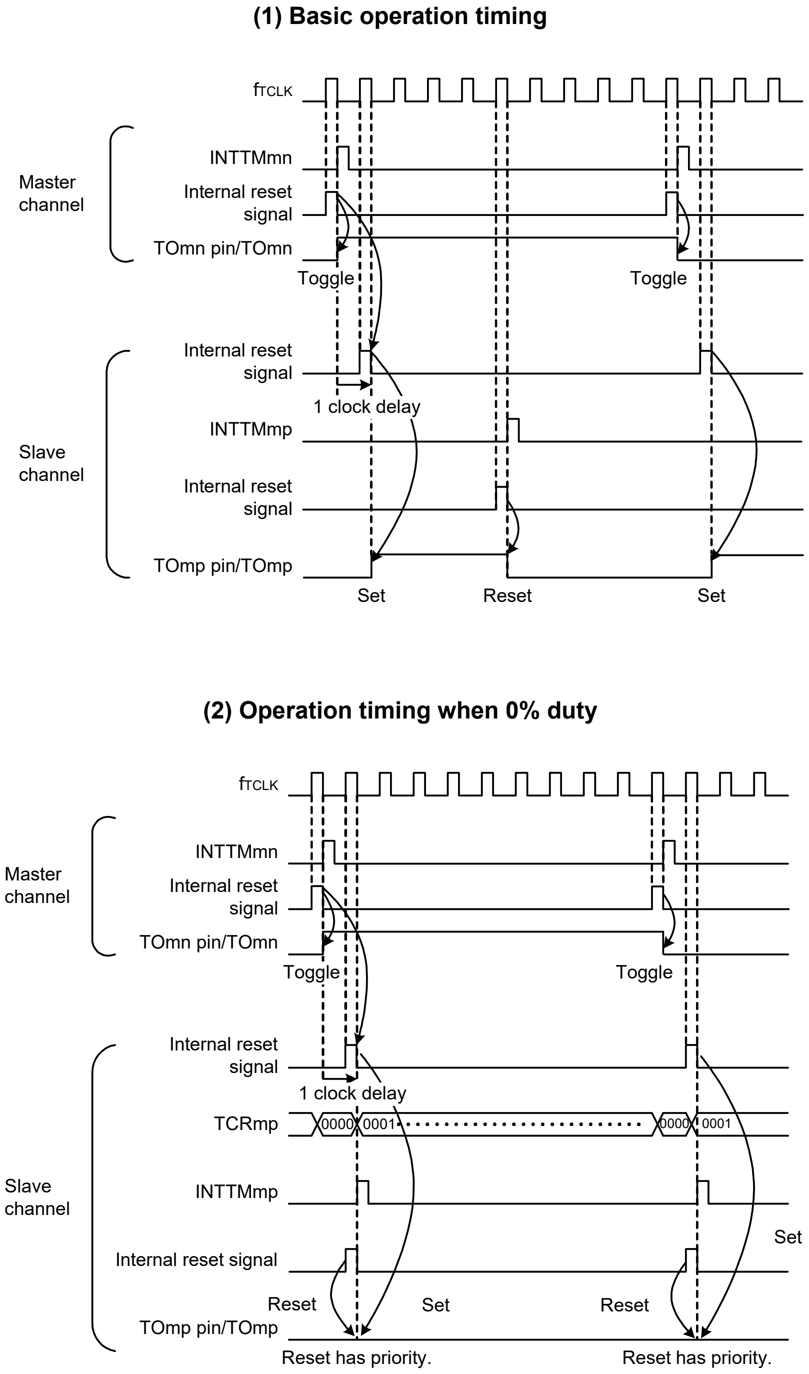
If the set condition and reset condition are generated at the same time, a higher priority is given to the latter.

Figure 7 - 36 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 - 36 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 - 37 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

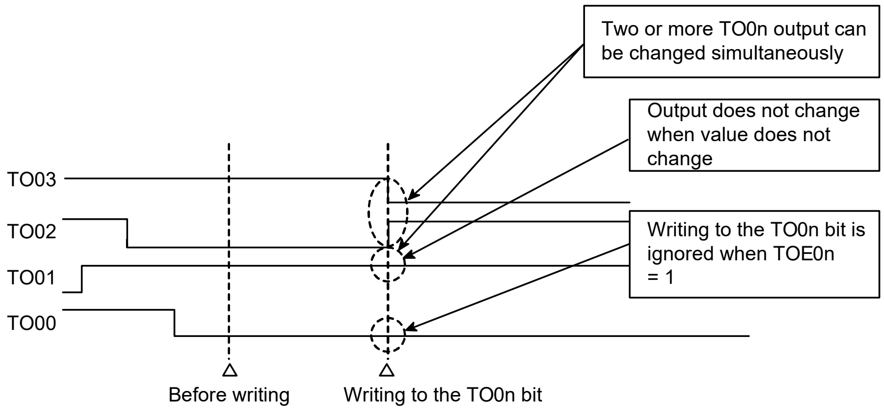
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
												○	○	○	×

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 - 38 TO0n Pin Statuses by Collective Manipulation of TO0n Bit


Two or more TO0n output can be changed simultaneously
Output does not change when value does not change
Writing to the TO0n bit is ignored when TOE0n = 1
TO03
TO02
TO01
TO00
Before writing
Writing to the 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.

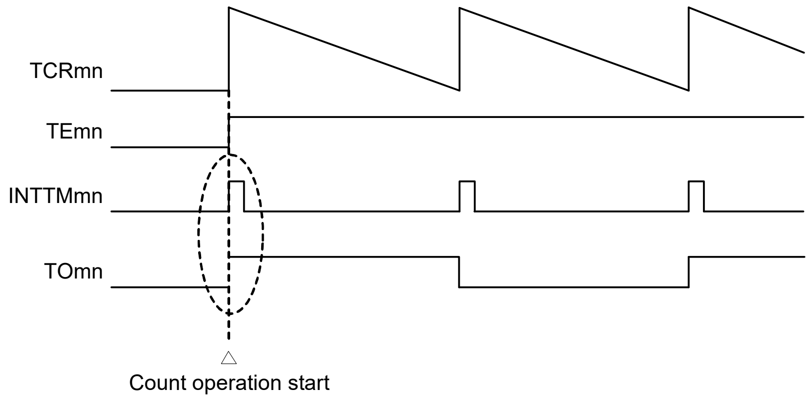
When MDmn0 is set to 1, the count operation start timing can be known by the timer interrupt (INTTMmn) generation.

In the other modes, neither timer interrupt at count operation start nor TOmn output is controlled.

Figure 7 - 39 shows operation examples when the interval timer mode (TOEmn = 1, TOMmn = 0) is set.

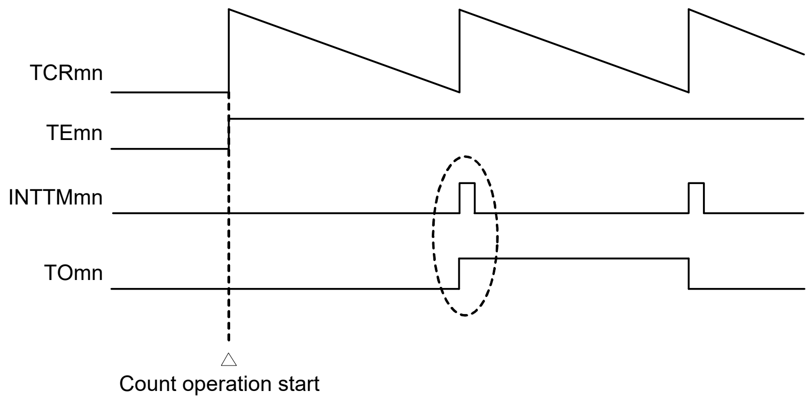
Figure 7 - 39 Operation examples of timer interrupt at count operation start and TOmn output

(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.

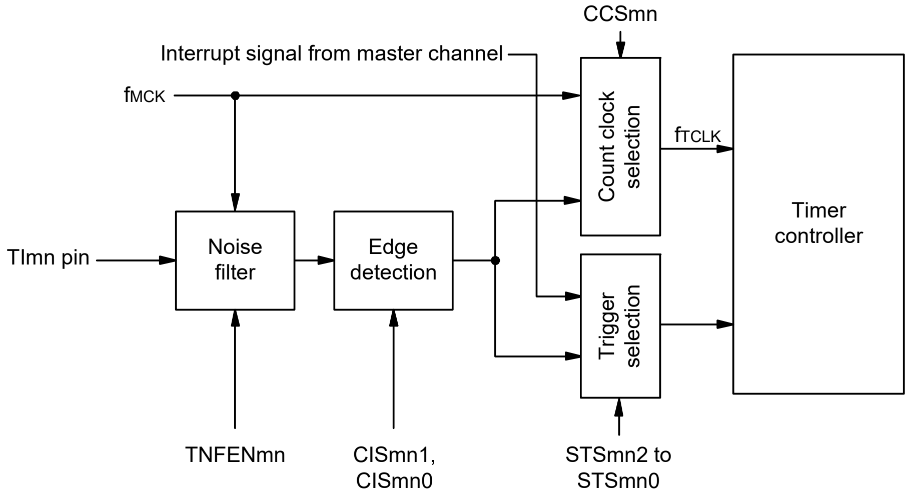
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

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.

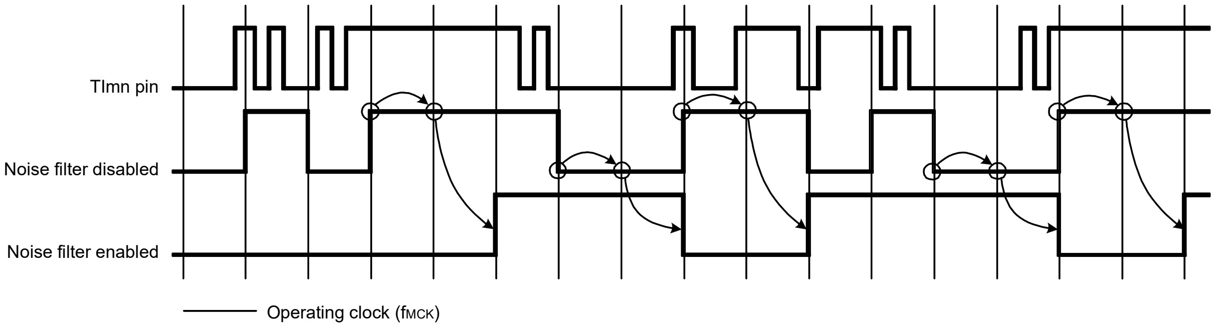
Figure 7 - 40 Input Circuit Configuration



7.7.2 Noise filter

When the noise filter is disabled, the input signal is only synchronized with the operating clock (fMCK) for channel n. When the noise filter is enabled, after synchronization with the operating clock (fMCK) for channel n, whether the signal keeps the same value for two clock cycles is detected. The following shows differences in waveforms output from the noise filter between when the noise filter is enabled and disabled.

Figure 7 - 41 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 35.4 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.

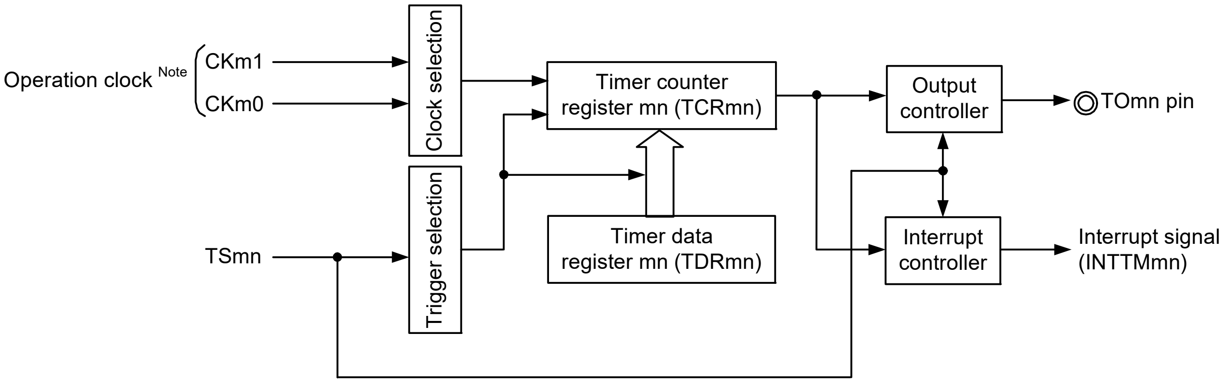
After that, the TCRmn register count down in synchronization with the count clock.

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.

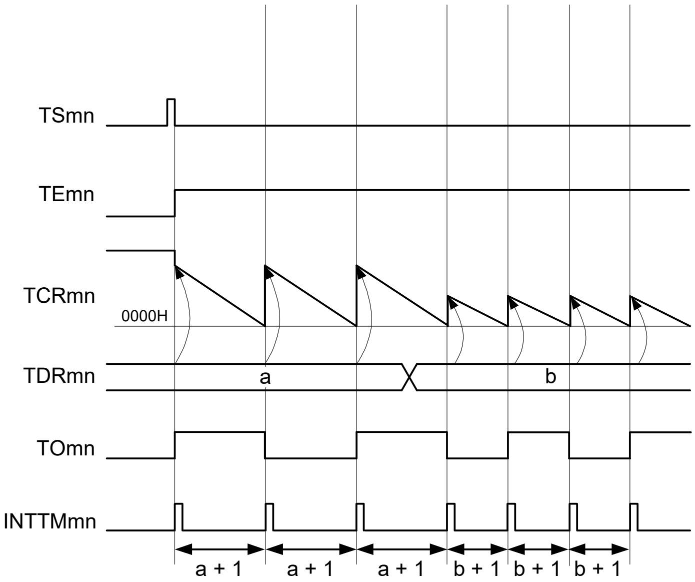
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 7 - 42 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 - 43 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 - 44 Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output

(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 - 45 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.	The TOmn pin goes into Hi-Z output state.
	Sets the TOmn bit and determines default level of the TOmn output.	The TOmn default setting level is output when the port mode register is in the output mode and the port register is 0.
	Sets the TOEmn bit to 1 and enables operation of TOmn.	TOmn does not change because channel stops operating.
	Clears the port register and port mode register to 0.	The TOmn pin outputs the TOmn set level.
Operation start	(Sets the TOEmn bit to 1 only if using TOmn output and resuming operation.). Sets the TSmn (TSHm1, TSHm3) bit to 1. The TSmn (TSHm1, TSHm3) bit automatically returns to 0 because it is a trigger bit.	TEmn (TEHm1, TEHm3) = 1, and count operation starts. Value of the TDRmn register is loaded to timer count register mn (TCRmn). 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 - 45 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. When holding the TOmn pin output level is not necessary Setting not required.	The TOmn pin output level is held by port function.
	The TAUmEN bit of the PER0 register is cleared to 0.	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.

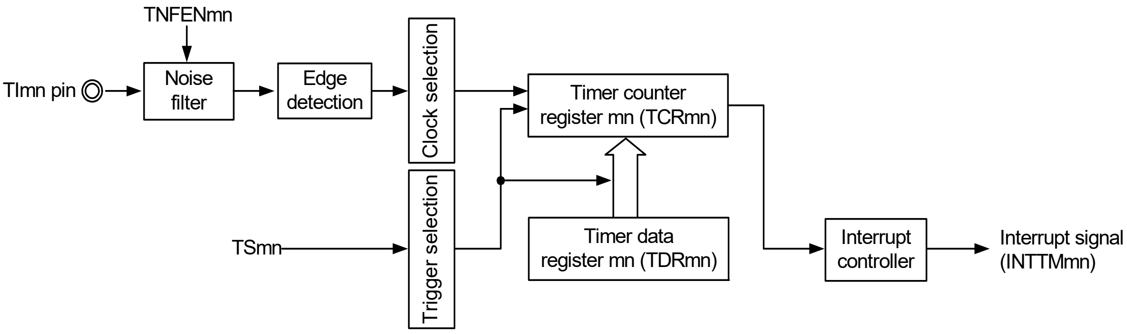
The TCRmn register counts down each time the valid input edge of the TImn pin has been detected. When TCRmn = 0000H, the TCRmn register loads the value of the TDRmn register again, and outputs INTTMmn.

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.

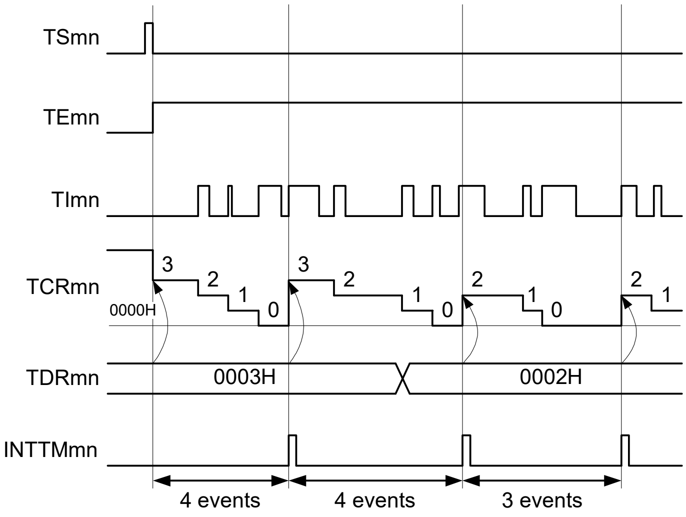
The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid during the next count period.

Figure 7 - 46 Block Diagram of Operation as External Event Counter



Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 7 - 47 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 - 48 Example of Set Contents of Registers in External Event Counter Mode

(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: Outputs 0 from TOmn
	0	

(c) Timer output enable register m (TOEm)

	Bit n	
TOEm	TOEmn	0: Stops the TOmn output operation by counting operation.
	0	

(d) Timer output level register m (TOLm)

	Bit n	
TOLm	TOLmn	0: Cleared to 0 when TOMmn = 0 (master channel output mode).
	0	

(e) Timer output mode register m (TOMm)

	Bit n	
TOMm	TOMmn	0: Sets master channel output mode.
	0	

Note		
	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 - 49 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

The timer array unit can be used as a frequency divider that divides a clock input to the TImn pin and outputs the result from the TOmn pin.

The divided clock frequency output from TOmn can be calculated by the following expression.

- When rising edge/falling edge is selected:
  Divided clock frequency = Input clock frequency/{(Set value of TDRmn + 1) × 2}
- When both edges are selected:
  Divided clock frequency ≅ Input clock frequency/(Set value of TDRmn + 1)

Timer count register mn (TCRmn) operates as a down counter in the interval timer mode.

After the channel start trigger bit (TSmn) of timer channel start register 0 (TS0) is set to 1, the TCRmn register loads the value of timer data register mn (TDRmn) when the TImn valid edge is detected.

If the MDmn0 bit of timer mode register mn (TMRmn) is 0 at this time, INTTMmn is not output and TOmn is not toggled. If the MDmn0 bit of timer mode register mn (TMRmn) is 1, INTTMmn is output and TOmn is toggled.

After that, the TCRmn register counts down at the valid edge of the TImn pin. When TCRmn = 0000H, it toggles TOmn. At the same time, the TCRmn register loads the value of the TDRmn register again, and continues counting.

If detection of both the edges of the TImn pin is selected, the duty factor error of the input clock affects the divided clock period of the TOmn output.

The period of the TOmn output clock includes a sampling error of one period of the operation clock.

Clock period of TOmn output = Ideal TOmn output clock period ± Operation clock period (error)

The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid during the next count period.

Figure 7 - 50 Block Diagram of Operation as Frequency Divider

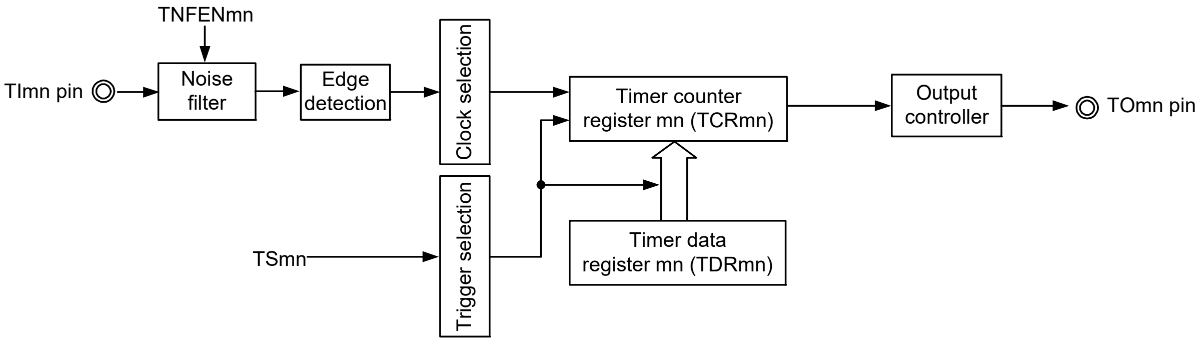
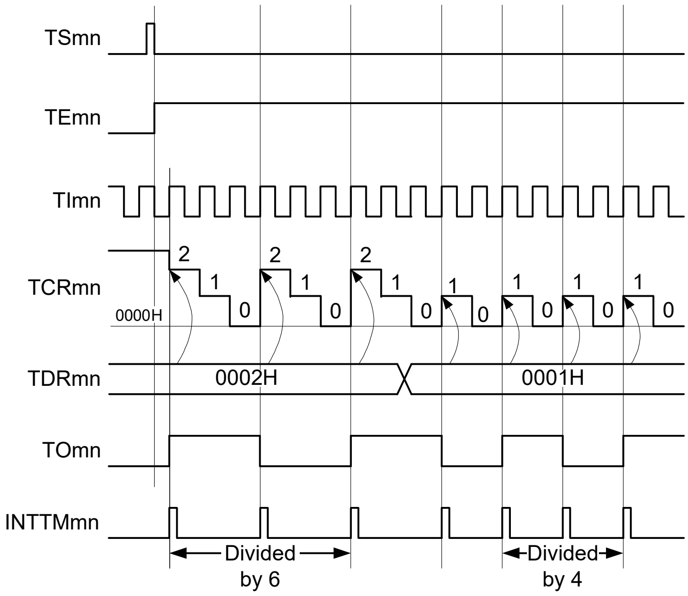


Figure 7 - 51 Example of Basic Timing of Operation as Frequency Divider (MDmn0 = 1)



Remark TSmn: Bit n of timer channel start register 0 (TS0)

TEmn: Bit n of timer channel enable status register 0 (TE0)

TImn: TImn pin input signal

TCRmn: Timer count register mn (TCRmn)

TDRmn: Timer data register mn (TDRmn)

TOmn: TOmn pin output signal

Figure 7 - 52 Example of Set Contents of Registers During Operation as Frequency Divider

(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		STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	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 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: 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 TImn pin input valid edge.

Operation clock (fMCK) selection
00B: Selects CKm0 as operation clock of channel 0.
10B: Selects CKm1 as operation clock of channel 0.

(b) Timer output register 0 (TO0)

	Bit 0
TO0	TOmn
	0

0: Outputs 0 from TOmn.
1: Outputs 1 from TOmn.

(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 (TOMmn = 0)

(e) Timer output mode register 0 (TOM0)

	Bit 0
TOM0	TOMmn
	0

0: Sets master channel output mode.

Figure 7 - 53 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 CK00 to CK03.	
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 and selects the detection edge). Sets interval (period) value to timer data register mn (TDRmn).	Channel stops operating. (Clock is supplied and some power is consumed.)
	Clears the TOMmn bit of timer output mode register 0 (TOM0) to 0 (master channel output mode). Clears the TOLmn bit to 0.	The TOmn pin goes into Hi-Z output state.
	Sets the TOmn bit and determines default level of the TOmn output. →	The TOmn default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOEmn bit to 1 and enables operation of TOmn. →	TOmn does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmn pin outputs the TOmn set level.
Operation start	Sets the TOEmn bit to 1 (only when operation is resumed). → Sets the TSmn bit to 1. The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and count operation starts. Value of the TDRmn register is loaded to timer count register mn (TCRmn). 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 TO0 and TOE0 registers can be changed. Set values of the TMRmn register, TOMmn, and TOLmn bits cannot be changed.	Counter (TCRmn) counts down. When count value reaches 0000H, the value of the TDRmn register is loaded to the TCRmn register again, and the count operation is continued. By detecting TCRmn = 0000H, INTTMmn is generated and TOmn performs toggle operation. After that, the above operation is repeated.
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops. The TOmn output is not initialized but holds current status.
	The TOEmn bit is cleared to 0 and value is set to the TOmn bit. →	The TOmn pin outputs the TOmn set level.
TAU stop	To hold the TOmn pin output level Clears the TOmn bit to 0 after the value to be held is set to the port register. → When holding the TOmn pin output level is not necessary Setting not required.	The TOmn pin output level is held by port function.
	The TAU0EN 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 0 is stopped All circuits are initialized and SFR of each channel is also initialized. (The TOmn bit is cleared to 0 and the TOmn pin is set to port mode).

Operation is resumed.

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.

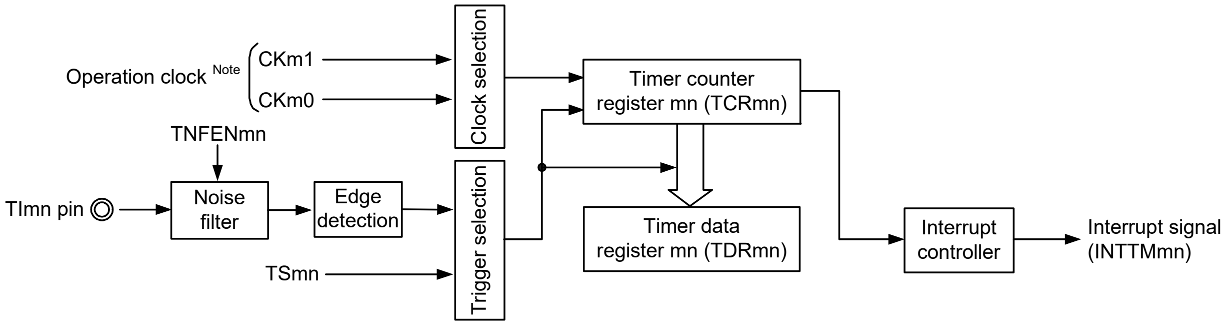
When the TImn pin input valid edge is detected, the count value of the TCRmn register is transferred (captured) to timer data register mn (TDRmn) and, at the same time, the TCRmn register is cleared to 0000H, and the INTTMmn is output. If the counter overflows at this time, the OVF bit of timer status 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.

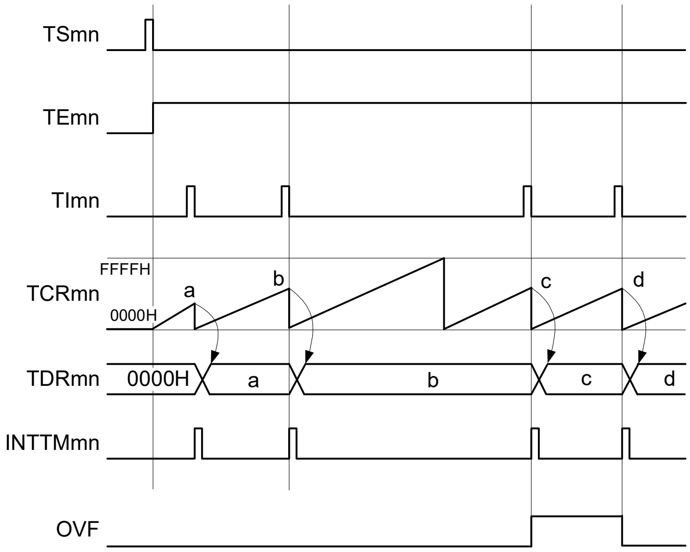
Figure 7 - 54 Block Diagram of Operation as Input Pulse Interval Measurement



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 - 55 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 - 56 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 (fMCK).

Operation clock (fMCK) 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 - 57 Operation Procedure When Input Pulse Interval 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).	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.

Operation is resumed.

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.

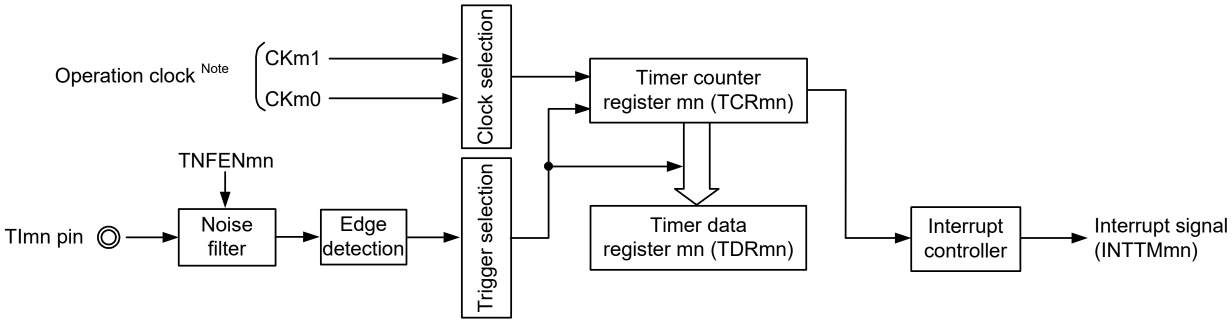
Signal width of TImn input = Period of count clock × ((10000H × TSRmn: OVF) + (Capture value of TDRmn + 1))

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.

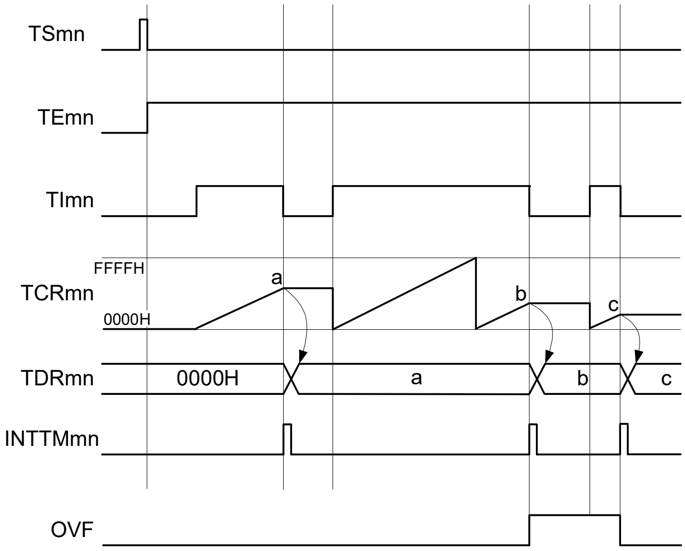
CISmn1, CISmn0 of TMRmn register = 10B: Low-level width is measured.
CISmn1, CISmn0 of TMRmn register = 11B: High-level width is measured.

Figure 7 - 58 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 - 59 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 - 60 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 (fMCK).

Operation clock (fMCK) 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 - 61 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)

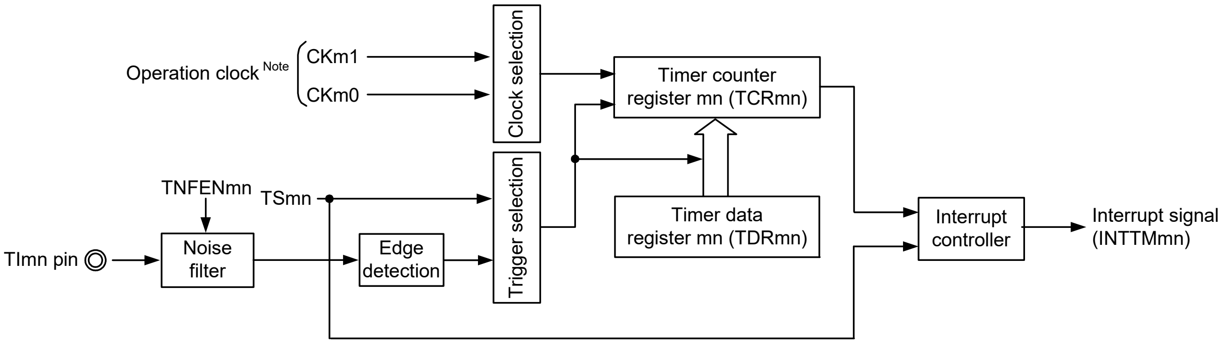
Timer count register mn (TCRmn) operates as a down counter in the one-count mode.

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.

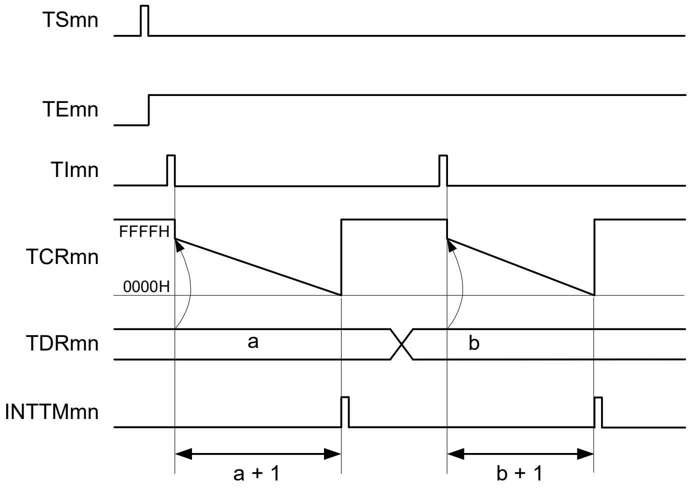
Figure 7 - 62 Block Diagram of Operation as Delay Counter



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 - 63 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 - 64 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.

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 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.

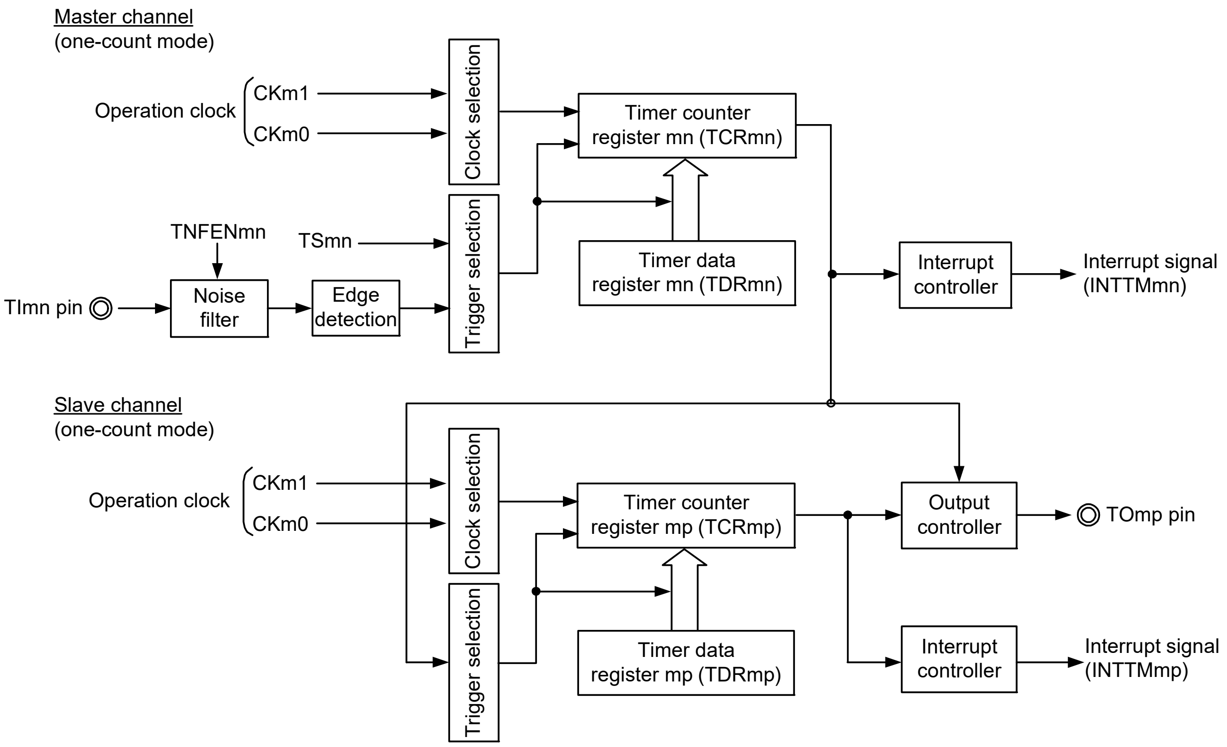
The slave channel operates in the one-count mode and counts the pulse width. The TCRmp register of the slave channel starts operation using INTTMmn of the master channel as a start trigger, and loads the value of the TDRmp register. The TCRmp register counts down from the value of The TDRmp register 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.

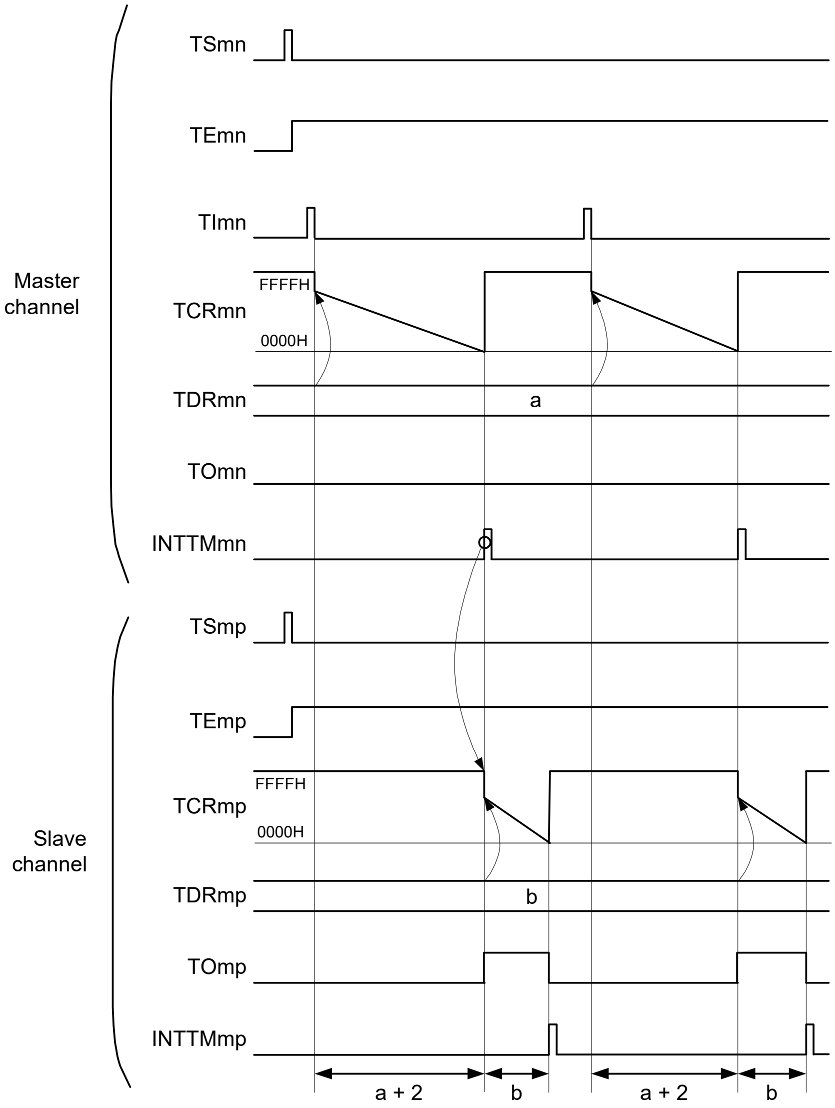
Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n = 0: p = 1, 2, 3, n = 2: p = 3)

Figure 7 - 66 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 - 67 Example of Basic Timing of Operation as One-Shot Pulse Output Function



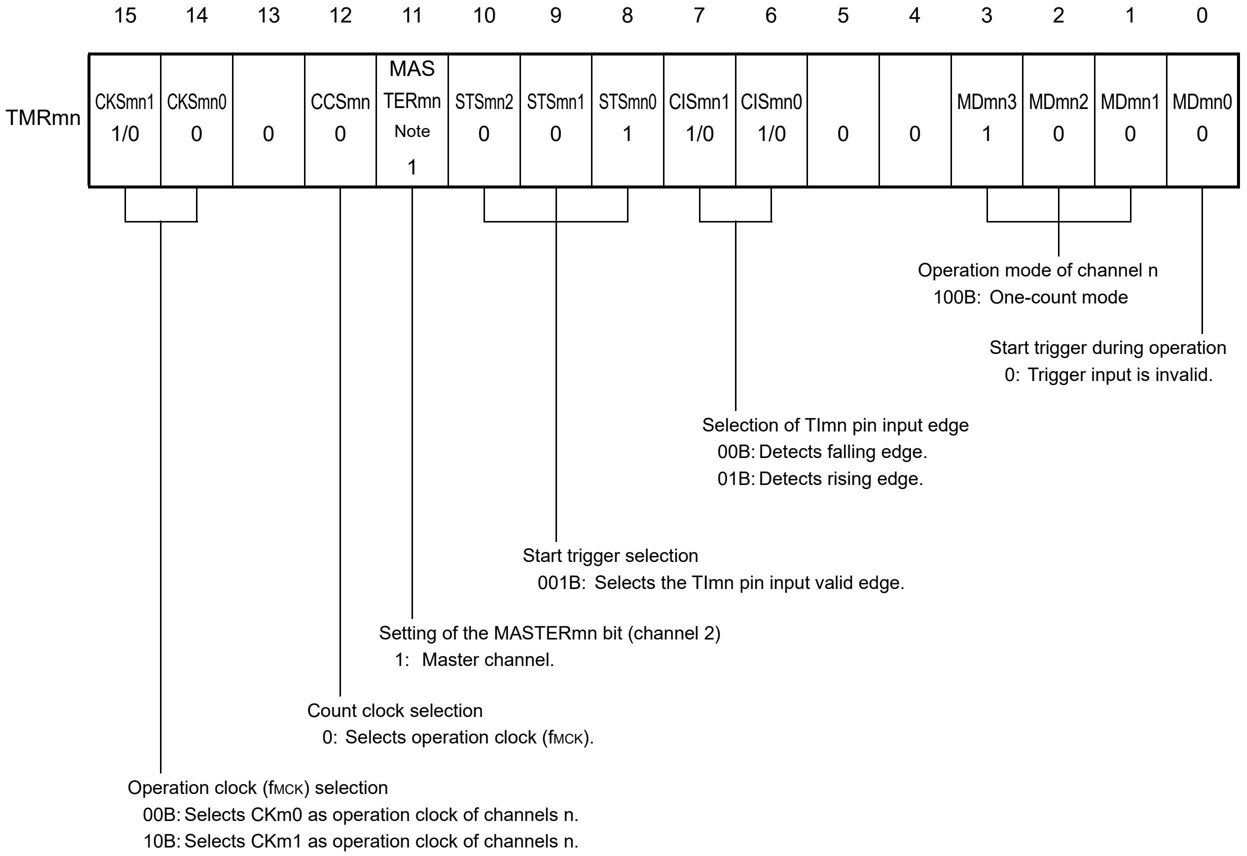
Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n = 0: p = 1, 2, 3, n = 2: p = 3)

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 - 68 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	CKSmn0		CCSmn	MASTERmn Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	0	1	0	0	1	1/0	1/0	0	0	1	0	0	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.

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

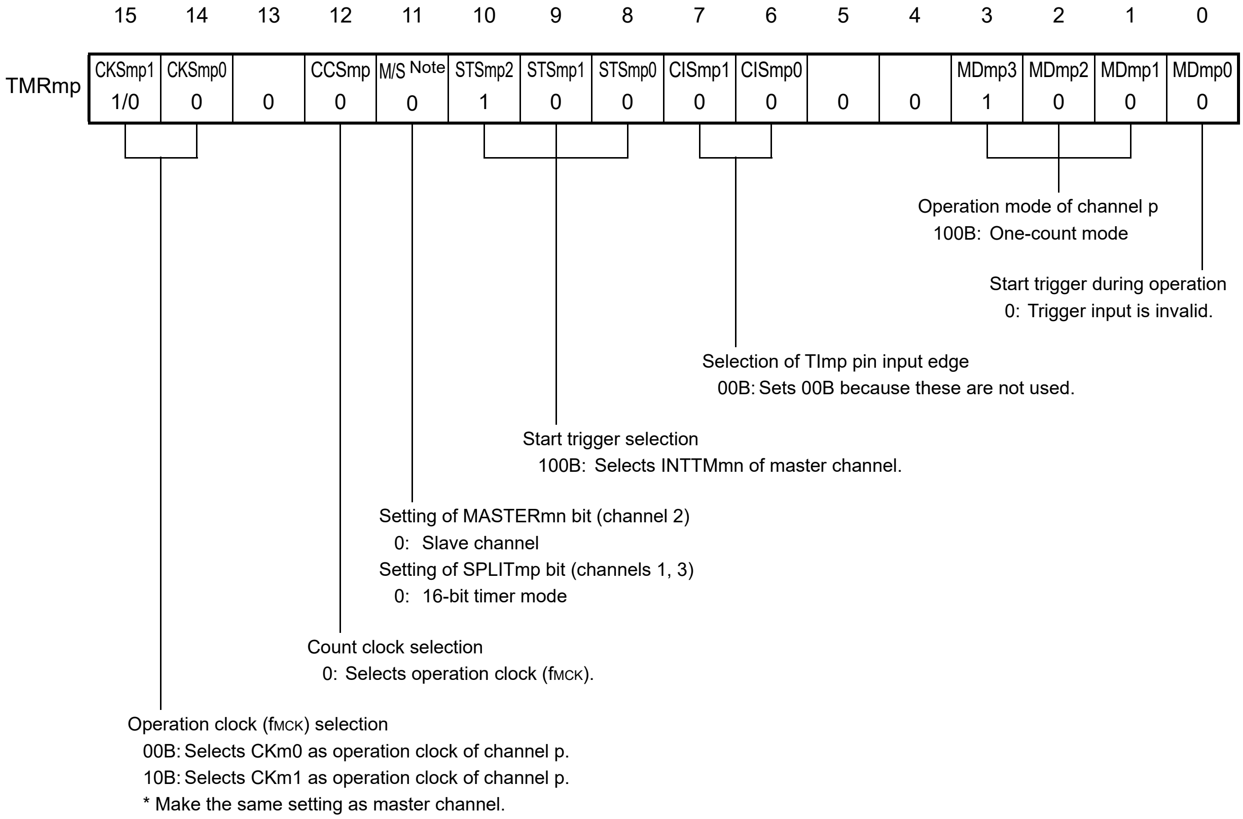
0: Sets master channel output mode.

Note TMRm2: MASTERmn = 1
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)

Figure 7 - 69 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 1/0	CKSmp0 0	0	CCSmp 0	M/S Note 0	STSmp2 1	STSmp1 0	STSmp0 0	CISmp1 0	CISmp0 0	0	0	MDmp3 1	MDmp2 0	MDmp1 0	MDmp0 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 (fMCK).

Operation clock (fMCK) 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 - 70 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.	The TOmp pin goes into Hi-Z output state.
	Sets the TOmp bit and determines default level of the TOmp output. →	The TOmp default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOEmp bit to 1 and enables operation of TOmp. →	TOmp does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmp pin outputs the TOmp set level.

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

Figure 7 - 70 Operation Procedure of One-Shot Pulse Output Function (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets the TOEmp bit (slave) to 1 (only when operation is resumed). The TSmn (master) and TSmp (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	The TEmn and TEmp bits are set to 1 and the master channel enters the 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.)

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 Two-channel input with one-shot pulse output function

TBy using signal input to two pins (TImn and TImp), a one-shot pulse having any delay pulse width can be generated.

Delay time = {Set value of TDRmn (master) + 2}   count clock period
One-shot pulse active-level width =
count clock period   ((10000H + TSRmp:OVF) + (capture value of TDRmp (slave) + 1)

Caution The TImn and TImp pin inputs are each sampled using the operating clock (f_{MCK}) selected with the CKSmn1 bit of the timer mode register (TMRmn), so an error of one cycle of the operating clock (f_{MCK}) per pin occurs.

The master channel should be operated in the one-count mode to start counting the delays (output delay time) upon detection of a valid edge of the master channel TImn pin input used as the start trigger. Upon detection of a start trigger (valid edge of TImn pin input), the master channel loads the value of timer data register mn (TDRmn) to the timer count register mn (TCRmn), and performs counting down in synchronization with the count clock (f_{TCLK}).When TCRmn = 0000H, the master channel outputs INTTMmn and outputs the active level from the TOmp pin. It stops counting until the next start trigger is detected.

The slave channel should be operated in the capture mode to set the one-shot pulse to the inactive level upon detection of a valid edge of the slave channel TImn pin input, or timer input signal selected by the TIS0 register used as the end trigger. Upon detection of an end trigger (valid edge of TImp pin input), the slave channel transfers (captures) the count value of the TCRmp register to the TDRmp register, and clears it to 0000H. Simultaneously, the slave channel outputs INTTMmp and the inactive level from the TOmp pin. Here, if the counter overflow has occurred, the OVF bit in the timer status register mn (TSRmp) is set; if not, the OVF bit is cleared. After this, the same steps are repeated.

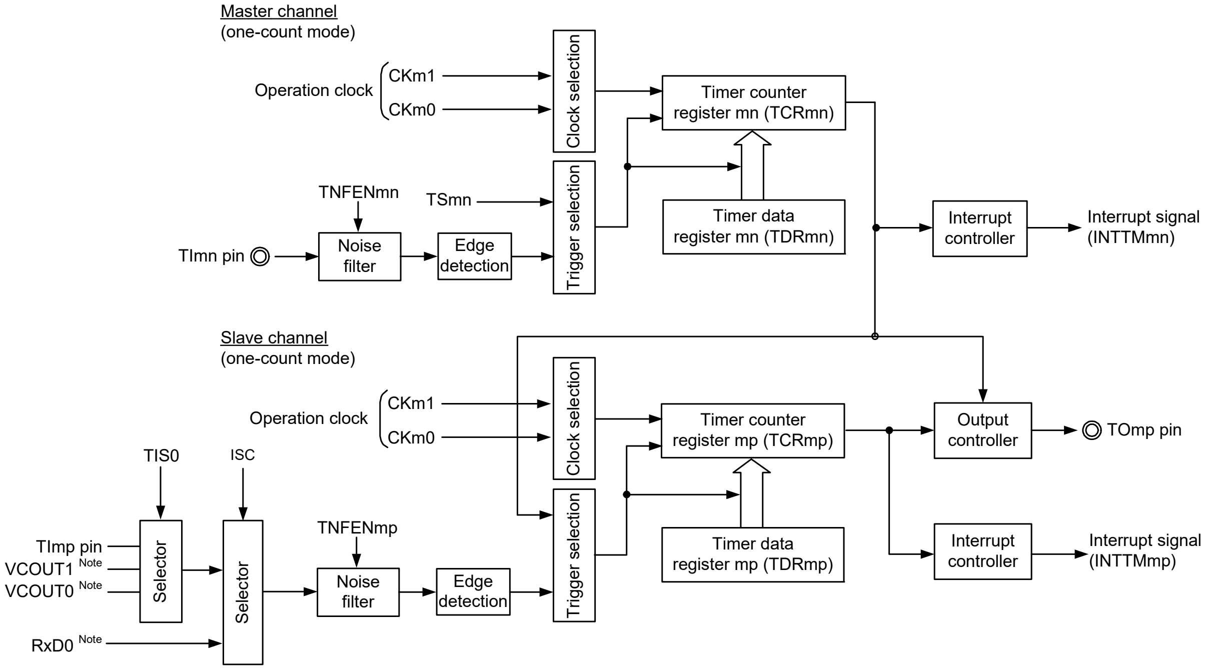
When the count value is captured to the TDRmp register, the OVF bit in the TSRmp register is updated depending on the overflow status during the active level period, which allows the overflow status of the captured value to be checked.

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 TSRmp register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.

Instead of using the TImn pin input, the software operation (TSmn = 1) can be used as a start trigger for the master channel.

Remark m: Unit number (m = 0)
n: Channel number (n = 0, 2)
p: Slave channel number (p = 3)

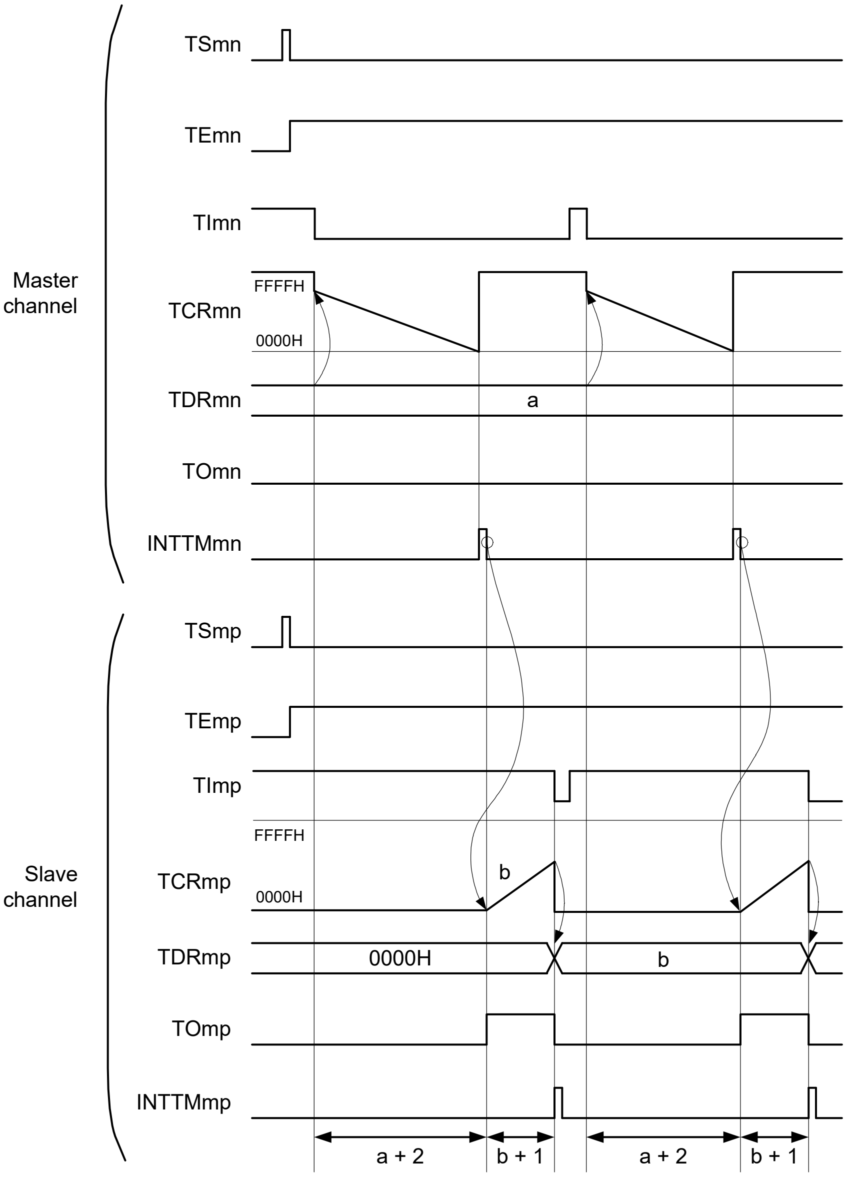
Figure 7 - 71 Block Diagram of Operation for Two-channel Input with One-shot Pulse Output Function



Note Only in channel 3.

Remark m: Unit number (m = 0)
n: Channel number (n = 0, 2)
p: Slave channel number (p = 3)

Figure 7 - 72 Example of Basic Timing of Operation for Two-channel Input with One-shot Pulse Output Function

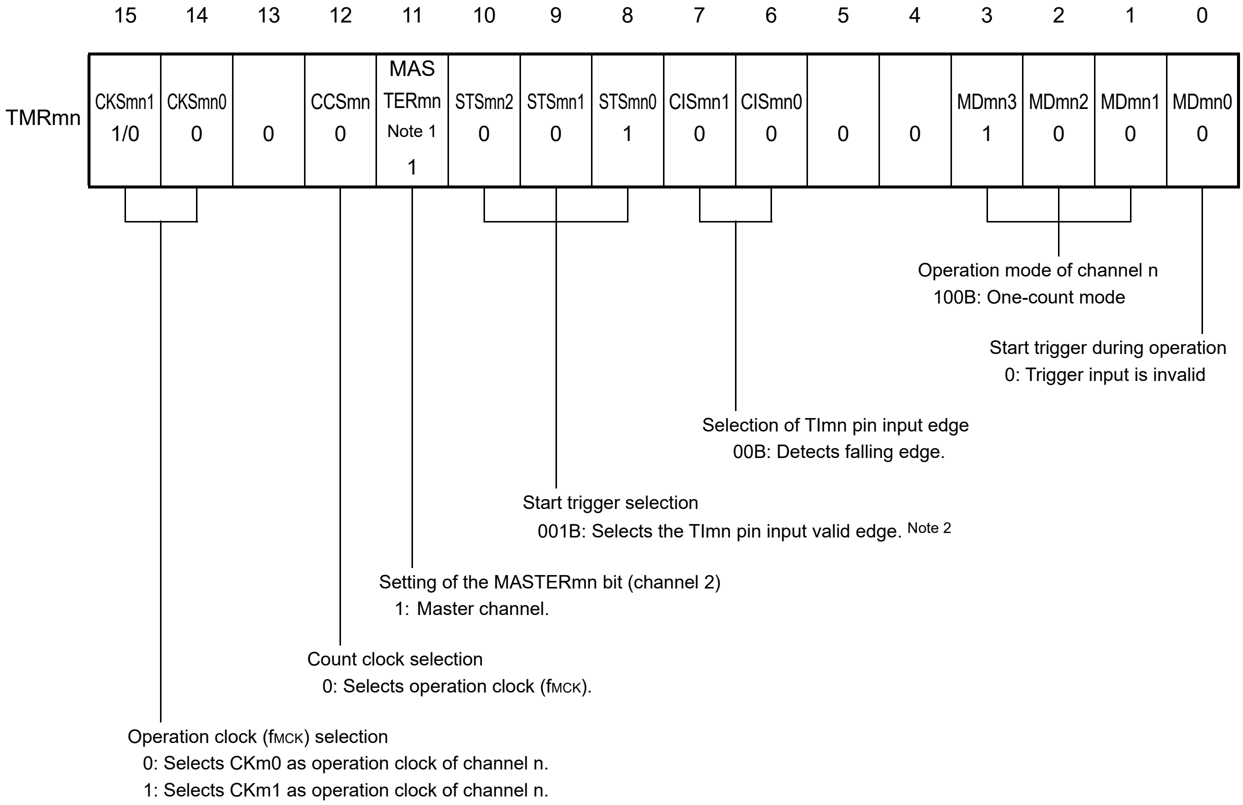


Remark 1. m: Unit number (m = 0)
n: Channel number (n = 0, 2)
p: Slave channel number (p = 3)

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 - 73 Example of Set Contents of Registers for Two-channel Input with One-shot Pulse Output Function (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	MAS TERmn Note 1 1	STSmn2 0	STSmn1 0	STSmn0 1	CISmn1 0	CISmn0 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.

Start trigger selection
001B: Selects the TImn pin input valid edge. Note 2

Setting of the MASTERmn bit (channel 2)
1: Master channel.

Count clock selection
0: Selects operation clock (fMCK).

Operation clock (fMCK) selection
0: Selects CKm0 as operation clock of channel n.
1: 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: Setting is invalid because master channel output mode is set (TOMmn = 0).0)

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn 0

0: Sets master channel output mode.

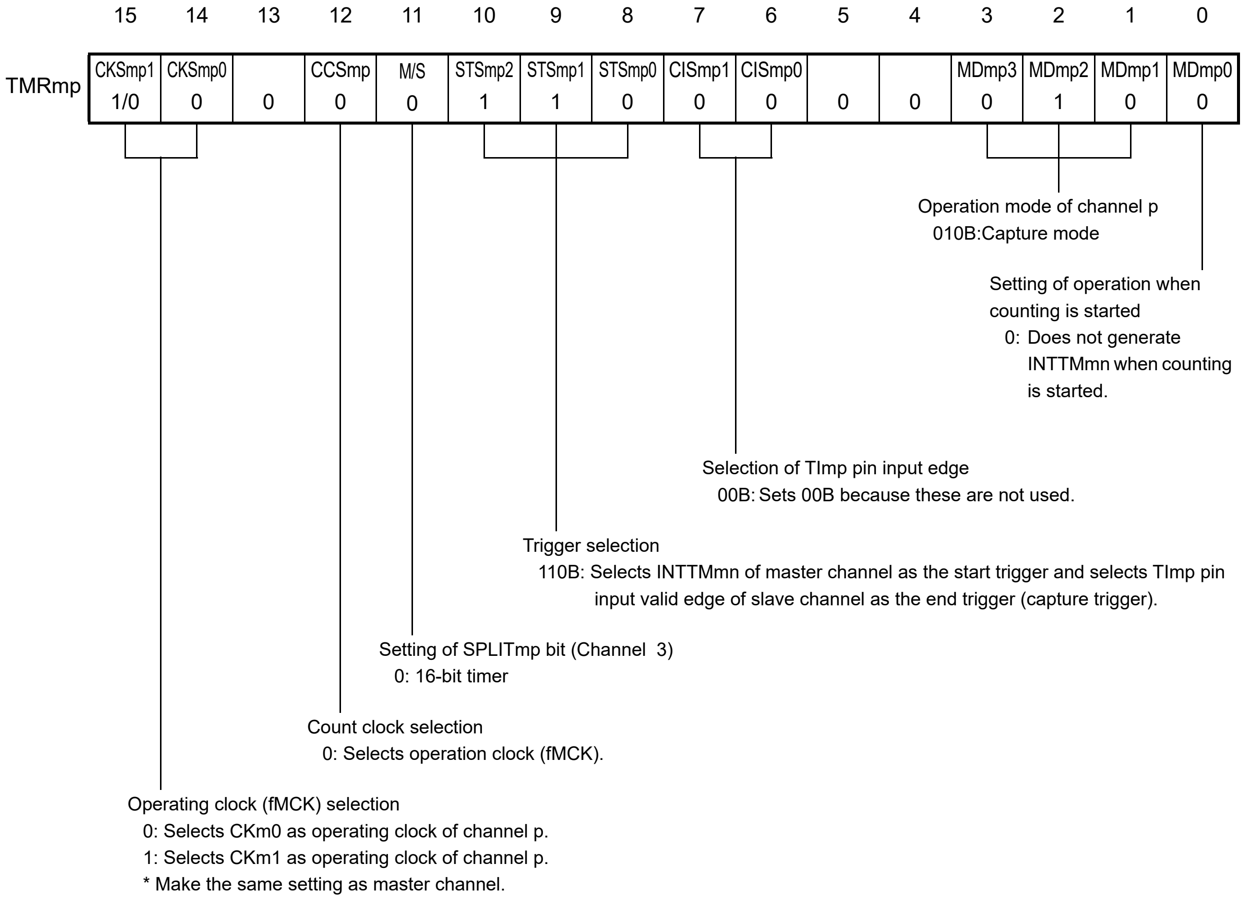
Note 1. TMRm2: MASTERmn = 1
TMRm0: Fixed to 0

Note 2. A software operation (TSmn = 1) can be used as a start trigger, instead of using the TImn pin input.

Remark m: Unit number (m = 0),
n: Channel number (n = 0, 2)

Figure 7 - 74 Example of Set Contents of Registers for Two-channel Input with One-shot Pulse Output Function (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 1/0	CKSmp0 0	0	CCSmp 0	M/S 0	STSmp2 1	STSmp1 1	STSmp0 0	CISmp1 0	CISmp0 0	0	0	MDmp3 0	MDmp2 1	MDmp1 0	MDmp0 0

Operation mode of channel p
010B:Capture mode

Setting of operation when counting is started
0: Does not generate INTTMmn when counting is started.

Selection of TImp pin input edge
00B: Sets 00B because these are not used.

Trigger selection
110B: Selects INTTMmn of master channel as the start trigger and selects TImp pin input valid edge of slave channel as the end trigger (capture trigger).

Setting of SPLITmp bit (Channel 3)
0: 16-bit timer

Count clock selection
0: Selects operation clock (fMCK).

Operating clock (fMCK) selection
0: Selects CKm0 as operating clock of channel p.
1: Selects CKm1 as operating 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. (the level set in the TOmp bit is output from the TOmp pin). 1: Enables the TOmp output operation by counting operation. (output from the TO0p pin is toggled).

(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.

Remark m: Unit number (m = 0)
n: Channel number (n = 0, 2)
p: Slave channel number (p = 3)

Figure 7 - 75 Procedure for Two-channel Input with One-shot Pulse Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to SFR of the TAU is disabled.)
	Sets the TAU0EN bit of peripheral enable registers 0 (PER0) to 1 (when the TAUmEN bit is 0, read/write operation is disabled). →	Power-on status. Each channel stops operating. (Clock supply is started and writing to SFR of the TAU is enabled.)
	Sets timer clock select register m (TPSm). Determines operating clock (CK00 and CK01) for each channel.	
Channel default setting	Sets noise filter enable register 1 (NFEN1). Sets timer mode register mn, p (TMRmn, TMRmp) (determines operation mode for each channel and selects the detection edge).	Channel stops operating.
	Sets master channel Sets delay (output delay time) to timer data register mn (TDRmn) (for the access procedure to the TDRmnH and TDRmnL registers, see **7.2.2 Timer data register mn (TDRmn)**). Clears the target bit of timer output mode register 0 (TOM0) to 0 (master channel output mode). Clears the target bit of the TOm0 register to 0. Clears the target bit of the timer output enable register 0 (TOEm) to 0. Sets slave channel. Sets the target bit of timer output mode register m (TOMm) to 1 (slave channel output mode). Sets the target bit of the TOLm register. 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. (output mode is set) →	The TOmp pin goes into Hi-Z state. (The port mode register is set to input mode.) TOmp does not change because channel stops operating. (The TOmp pin is not affected even if the TO0p bit is modified). The level set in the TOmp bit is output from the TOmp pin.

Remark m: Unit number (m = 0),
n: Channel number (n = 0, 2)
p: Slave channel number (p = 3)

Figure 7 - 75 Procedure for Two-channel Input with One-shot Pulse Output Function (2/2)

	Software Operation	Hardware Status
Operation start	Sets the TOEmp bit of the slave channel to 1 to enable TOmp operation (only when operation is resumed). Sets the target bits of the TSm register (master and slave) to 1 at the same time. → The target bits of the TSm register automatically return to 0 because they are trigger bits.	The target bits of the TEm register are set to 1 and the master channel enters the TImn pin input valid edge detection wait status.
	Count operation starts on detection of the next start triggers: → - The TImn pin input valid edge is detected. - The TSmn bit is set to 1 by software.	Value of the TDRmn register is loaded to the timer count register mn (TCRmn) of the master channel, and count down operation starts.
During operation	Changes master channel setting. The TCRmn register can always be read (for the access procedure to the TCRmnH and TCRmnL registers, see **7.2.1 Timer count register mn (TCRmn)**). The set values of only the CISmn1 and CIS0nm bits of the TMRmn register can be changed. The set values in the target bits of the TDRmn, TOm, TOEm, TOMm, and TOLm registers cannot be changed. Changes slave channel setting. The TDRmp register can always be read. The TCRmp register can always be read. The TSRmp register can always be read. The set values of only the CISmp1 and CISmp0 bits of the TMRmp register can be changed. The set values in the target bits of the TOmp, TOEmp, TOMm, and TOLm registers can be changed.	The master channel counter (TCRmn) performs count down operation. When the count value reaches TCRmn = 0000H, INTTMmn is generated, and the counter stops at TCRmn = FFFFH until the next start trigger is detected (the TImn pin input valid edge is detected or TSmn bit is set to 1). The slave channel, triggered by INTTMmn of the master channel, clears the timer counter register mp (TCRmp) to 0000H. The counter (TCRmp) starts counting up from 0000H, and when the TImn pin input valid edge is detected, the count value is transferred to the timer data register mp (TDRmp) (capture) and TCRmp register is cleared to 0000H. At this time, INTTMmn is generated, which sets the TOmp output level to inactive After that, the above operation is repeated.
Operation stop	Sets the target bits of the TT0 register (master and slave) to 1 at the same time. → The target bits of the TT0 register automatically return to 0 because they are trigger bits.	The target bits of the TEm register are cleared to 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..
	Clears the TOEmp bit of slave channel to 0 and sets a value to the TOmp bit. →	The level set in the TOmp bit is output from the TOmp pin.
TAU stop	To hold the TOmp pin output level Clears the TOmp bit to 0 after the value to be held (output latch) is set to the port register. →	The TOmp pin output level is held by port function.
	Clears the TAUmEN bit of the PER0 register to 0. →	Power-off status Clock supply is stopped and SFR of the TAU is initialized.

Operation is resumed.

Remark m: Unit number (m = 0)
n: Channel number (n = 0, 2)
p: Slave channel number (p = 3)

7.9.3 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}

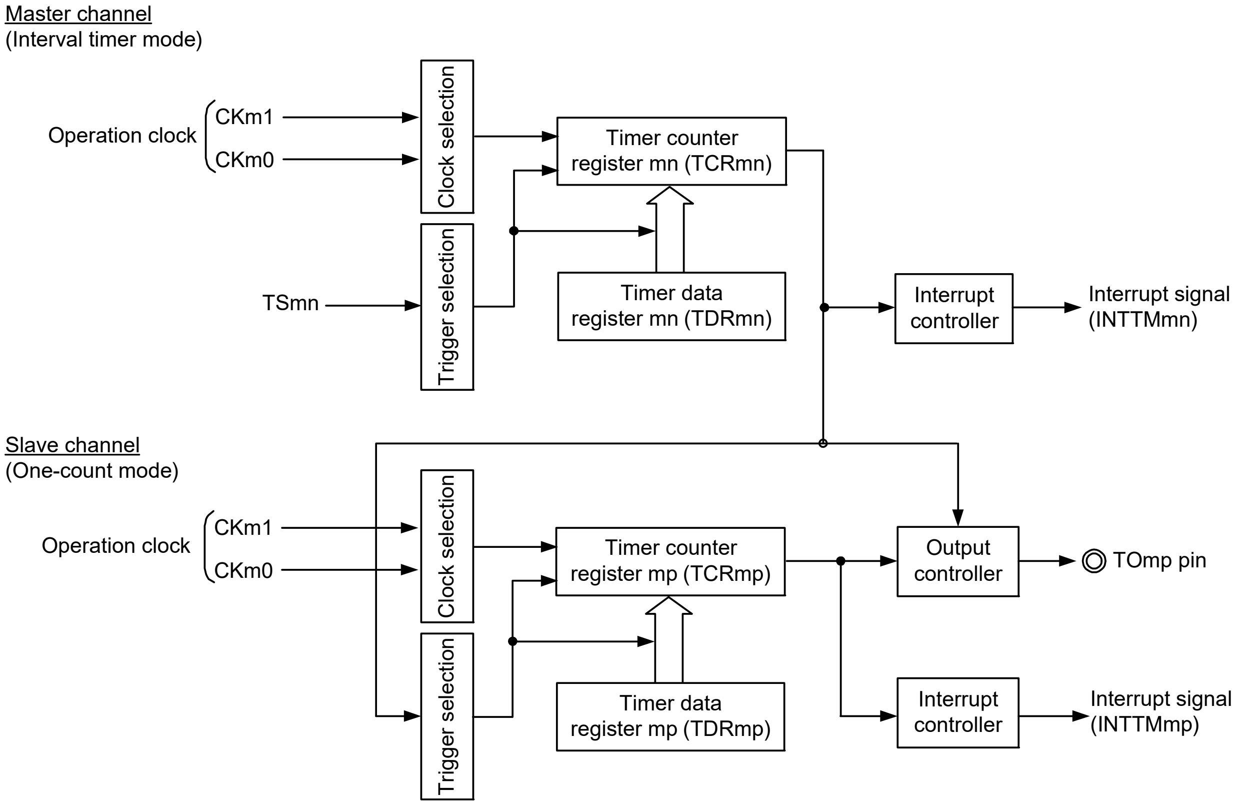
Remark The duty factor exceeds 100% if the set value of TDRmp (slave) > (set value of TDRmn (master) + 1), it summarizes to 100% output.

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.

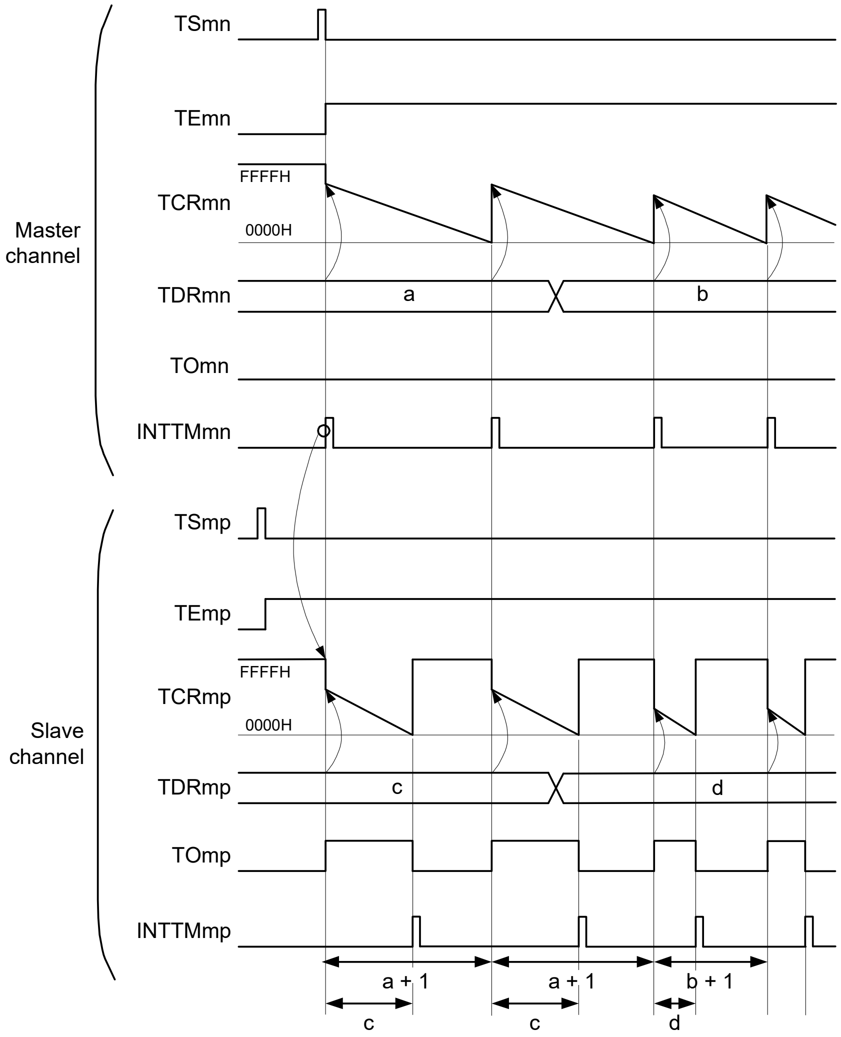
Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)
p: Slave channel number (n = 0: p = 1, 2, 3, n = 2: p = 3)

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

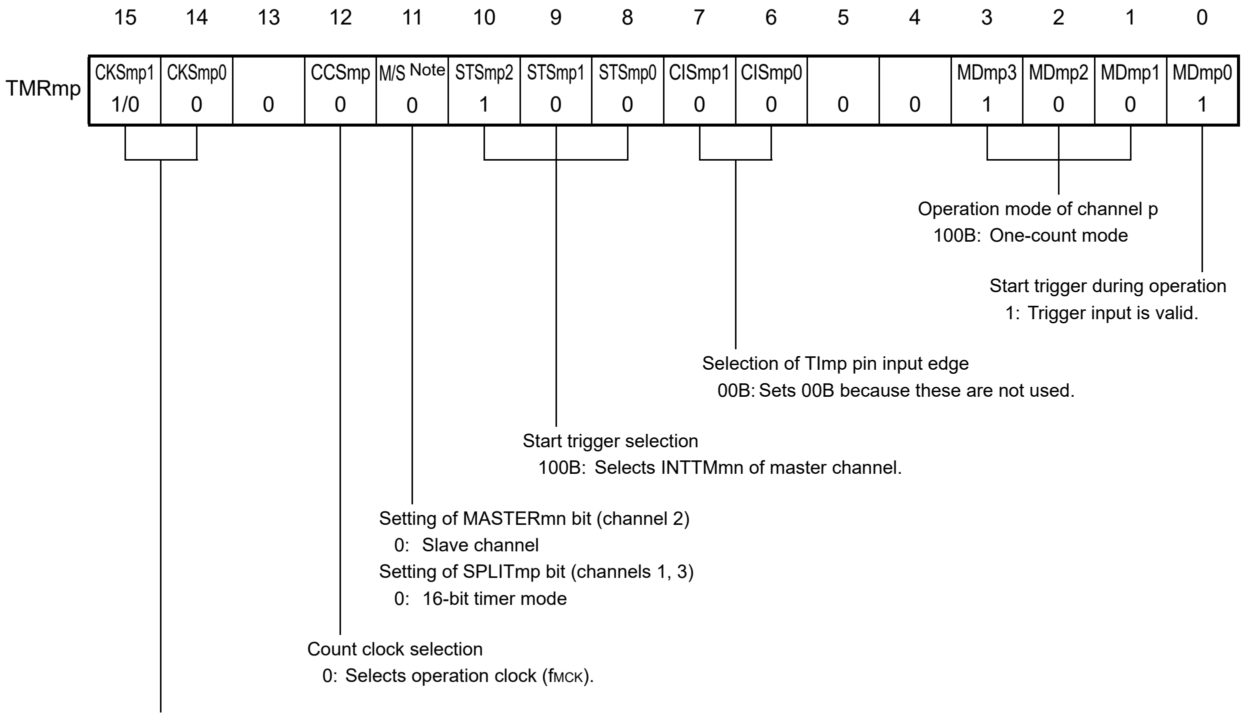
0: Sets master channel output mode.

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)



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 - 80 Operation Procedure When PWM Function Is Used (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets the TOEmp bit (slave) to 1 (only when operation is resumed). The TSmn (master) and TSmp (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	TEmn = 1, TEmp = 1 When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	Set values of the TMRmn and TMRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. Set values of the TDRmn and TDRmp registers can be changed after INTTMmn of the master channel is generated. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used.	The counter of the master channel loads the TDRmn register value to timer count register mn (TCRmn), and counts down. When the count value reaches TCRmn = 0000H, INTTMmn output is generated. At the same time, the value of the TDRmn register is loaded to the TCRmn register, and the counter starts counting down again. At the slave channel, the value of the TDRmp register is loaded to the TCRmp register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. The TTmn and TTmp bits automatically return to 0 because they are trigger bits.	TEmn, TEmp = 0, and count operation stops. The TCRmn and TCRmp registers hold count value and stop. The TOmp output is not initialized but holds current status.
	The TOEmp bit of slave channel is cleared to 0 and value is set to the TOmp bit.	The TOmp pin outputs the TOmp set level.
TAU stop	To hold the TOmp pin output level Clears the TOmp bit to 0 after the value to be held is set to the port register. When holding the TOmp pin output level is not necessary Setting not required.	The TOmp pin output level is held by port function.
	The TAUmEN bit of the PER0 register is cleared to 0. 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.)

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)

7.9.4 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

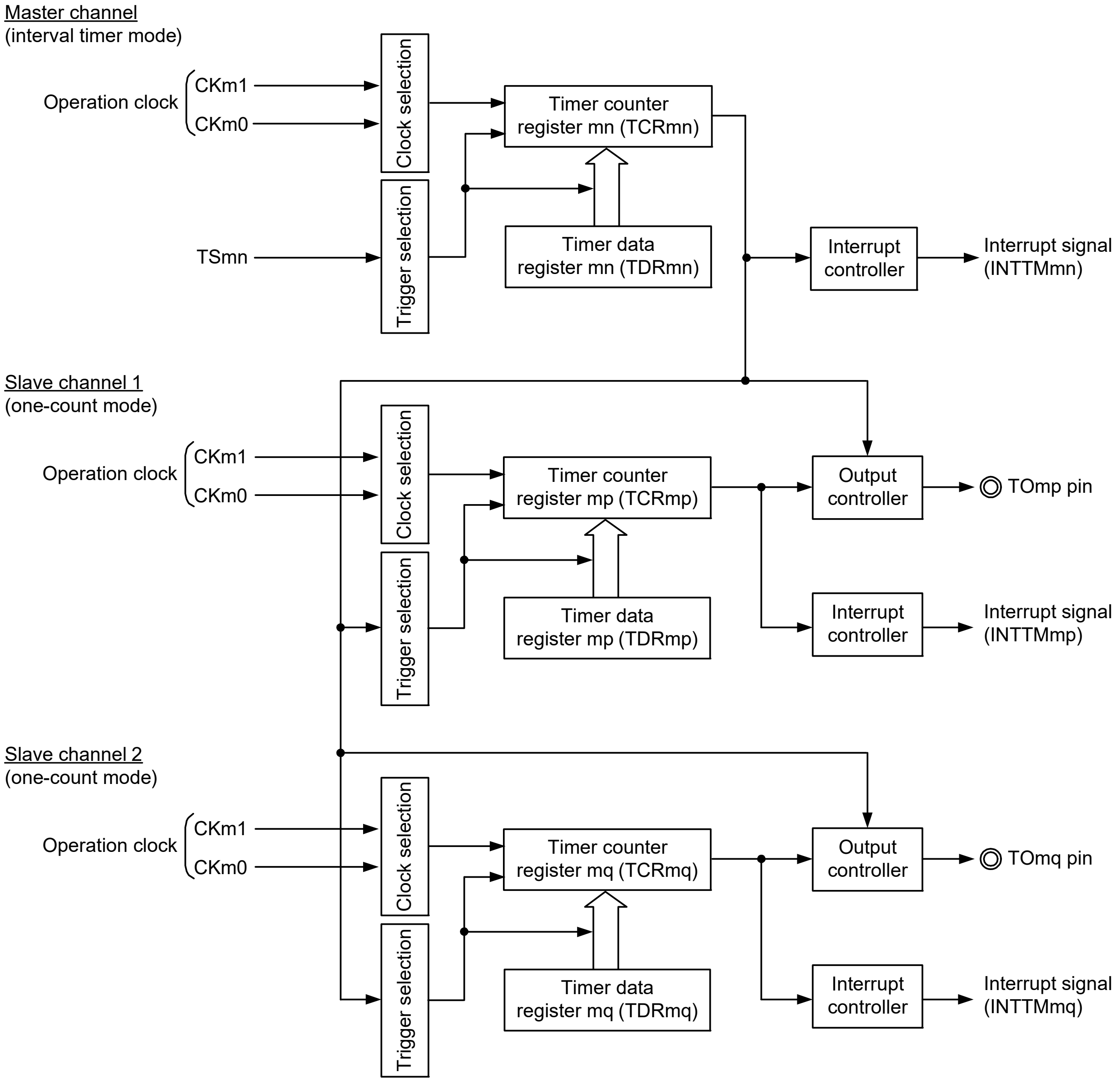
Remark Although the duty factor exceeds 100% if the set value of TDRmp (slave 1) > {set value of TDRmn (master) + 1} or if the {set value of TDRmq (slave 2)} > {set value of TDRmn (master) + 1}, it is summarized into 100% output.

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).**

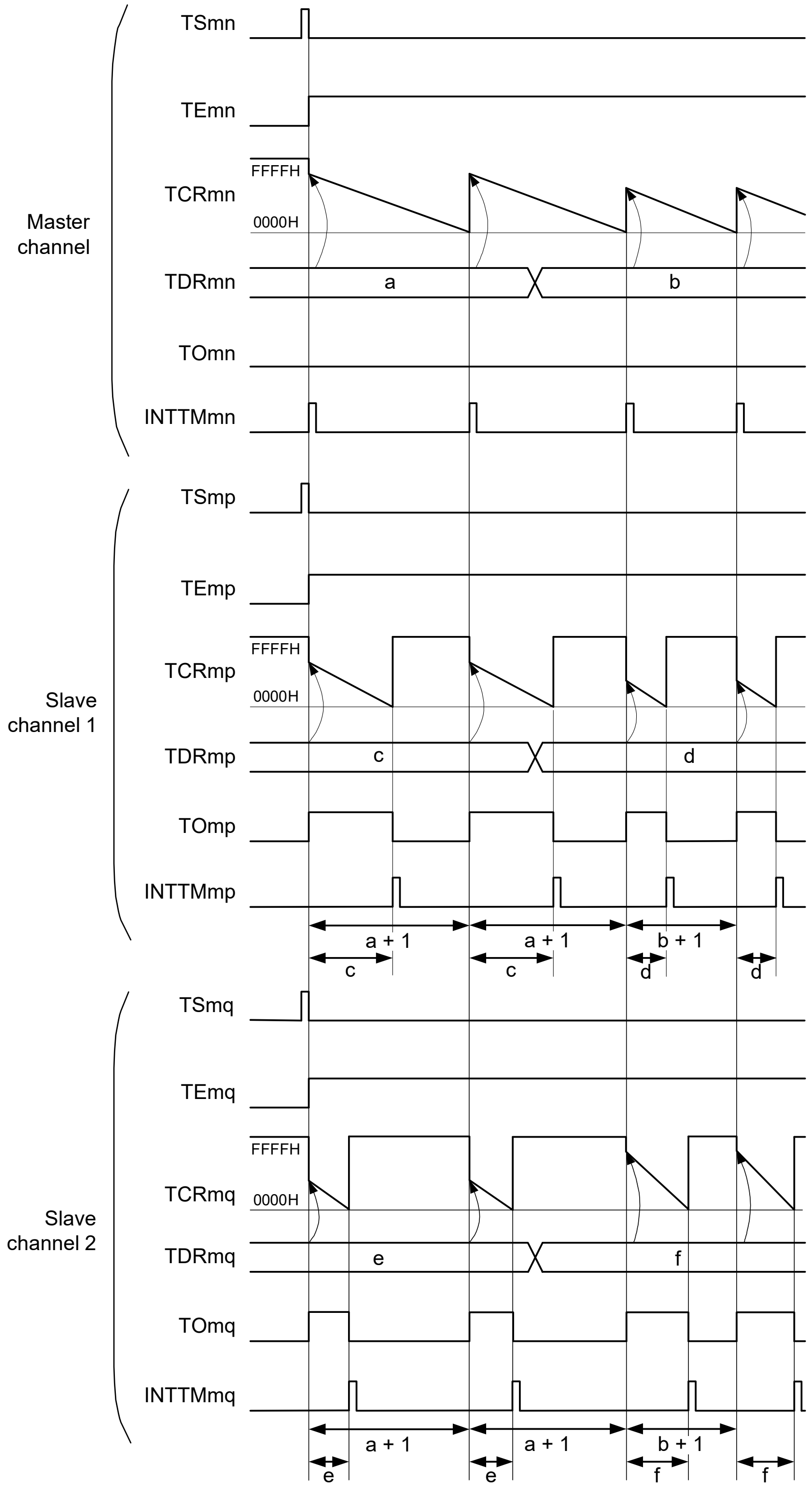
Remark m: Unit number (m = 0), n: Channel number (n = 0)
p: Slave channel number 1, q: Slave channel number 2
$n < p < q \leq 3$ (Where p and q are integers greater than n)

Figure 7 - 81 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 - 82 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
$n < p < q \leq 3$ (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 - 83 Example of Set Contents of Registers
When Multiple PWM Output Function (Master Channel) Is Used**

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	MAS TERmn 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 (fMCK).

Operation clock (fMCK) 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 - 84 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 1/0	CKSmp0 0	 0	CCSmp 0	M/S Note 0	STSmp2 1	STSmp1 0	STSmp0 0	CISmp1 0	CISmp0 0	 0	 0	MDmp3 1	MDmp2 0	MDmp1 0	MDmp0 1

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmq	CKSmq1 1/0	CKSmq0 0	 0	CCSmq 0	M/S Note 0	STSmq2 1	STSmq1 0	STSmq0 0	CISmq1 0	CISmq0 0	 0	 0	MDmq3 1	MDmq2 0	MDmq1 0	MDmq0 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 1/0	TOmp 1/0

0: Outputs 0 from TOmp or TOmq.
1: Outputs 1 from TOmp or TOmq.

(c) Timer output enable register m (TOEm)

	Bit q	Bit p
TOEm	TOEmq 1/0	TOEmp 1/0

0: Stops the TOmp or TOmq output operation by counting operation.
1: Enables the TOmp or TOmq output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit q	Bit p
TOLm	TOLmq 1/0	TOLmp 1/0

0: Positive logic output (active-high)
1: Negative logic output (active-low)

(e) Timer output mode register m (TOMm)

	Bit q	Bit p
TOMm	TOMmq 1	TOMmp 1

1: Sets the slave channel output mode.

Note 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
$n < p < q \leq 3$ (Where p and q are integers greater than n)

Figure 7 - 85 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.	The TOmp and TOmq pins go into Hi-Z output state.
	Sets the TOmp and TOmq bits and determines default level of the TOmp and TOmq outputs. →	The TOmp and TOmq default setting levels are output when the port mode register is in output mode and the port register is 0.
	Sets the TOEmp and TOEmq bits to 1 and enables operation of TOmp and TOmq. →	TOmp and TOmq do not change because channels stop operating.
	Clears the port register and port mode register to 0. →	The TOmp and TOmq pins output the TOmp and TOmq set levels.

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

Figure 7 - 85 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 ≤ 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 12-BIT INTERVAL TIMER

8.1 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.

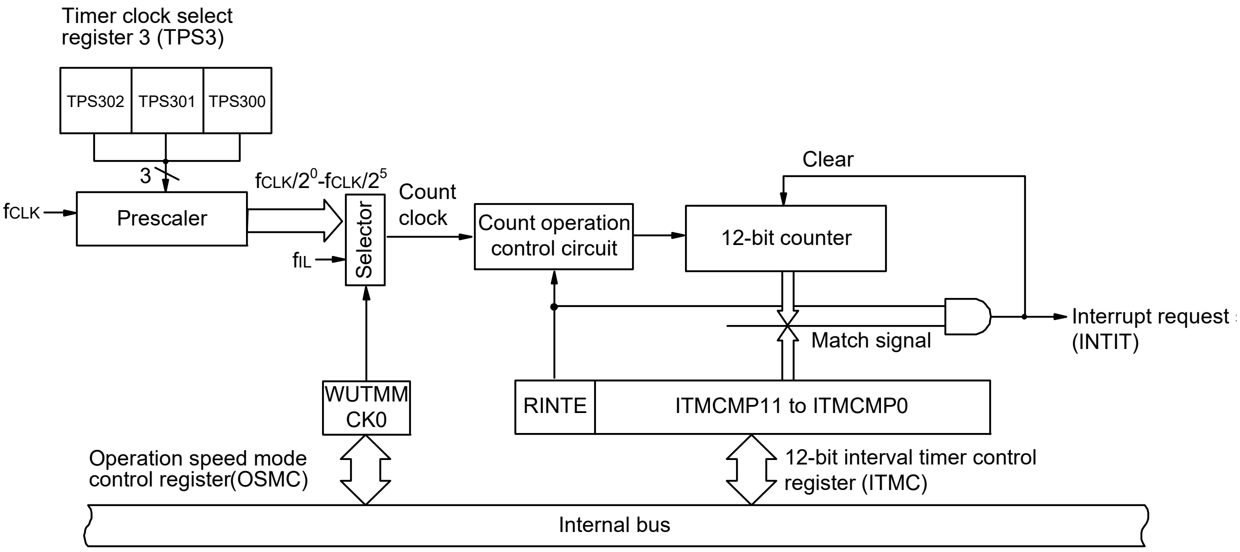
8.2 Configuration of 12-bit Interval Timer

The 12-bit interval timer includes the following hardware.

Table 8 - 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)
	Operation speed mode control register (OSMC)
	Timer clock select register 3 (TPS3)
	12-bit interval timer control register (ITMC)

Figure 8 - 1 Block Diagram of 12-bit Interval Timer


Timer clock select register 3 (TPS3)
TPS302
TPS301
TPS300
3
fCLK
Prescaler
fCLK/2^0-fCLK/2^5
fIL
Selector
Count clock
Count operation control circuit
12-bit counter
Clear
Match signal
Interrupt request (INTIT)
WUTMMCK0
RINTE
ITMCMP11 to ITMCMP0
Operation speed mode control register(OSMC)
12-bit interval timer control register (ITMC)
Internal bus

8.3 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)
- Operation speed mode control register (OSMC)
- Timer clock select register 3 (TPS3)
- 12-bit interval timer control register (ITMC)

8.3.1 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 8 - 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	0	DOCEN	0	0	0	0	TKB0EN

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 (except timer clock select register 3 (TPS3)).

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).

8.3.2 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 8 - 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	TKB0RES

TMKARES	Reset control of 12-bit interval timer
0	12-bit interval timer reset release
1	12-bit interval timer reset state

8.3.3 Operation speed mode control register (OSMC)

The WUTMMCK0 bit can be used to select the 12-bit interval timer operation clock.

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

Reset input sets this register to undefined.

Figure 8 - 4 Format of operation speed mode control register (OSMC)

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

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

WUTMMCK0	Selection of operation clock for 12-bit interval timer Note 2, 8-bit interval timer, and clock output/buzzer output controller
0	Low-speed on-chip oscillator clock is not selected.
1	Low-speed on-chip oscillator clock is selected.

Note 1. Be sure to set bits 0 to 3, 5, and 6 to 0.

Note 2. The operating clock for the 12-bit interval timer can be selected by the timer clock select register 3 (TPS3) and OSMC.WUTMMCK0 (see **8.3.4 Timer clock select register 3 (TPS3)**).

8.3.4 Timer clock select register 3 (TPS3)

This register is an 8-bit register to select the count clock.
The value of this register can be changed while the 12-bit interval timer is stopped.

The TPS3 register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 8 - 5 Format of timer clock select register 3 (TPS3)

Address: F02D5H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TPS3	0	0	0	0	0	TPS302	TPS301	TPS300

TPS 302	TPS 301	TPS 300	Selection of count clock Note 1					
				f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	f_{CLK} = 24 MHz
0	0	0	f_{IL} Note 2	15 MHz				
0	0	1	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz	24 MHz
0	1	0	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	12 MHz
0	1	1	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	6 MHz
1	0	0	$f_{CLK}/2^3$	250 kHz	625 MHz	1.25 MHz	2.5 MHz	3 MHz
1	0	1	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 Mhz	1.5 MHz
1	1	0	$f_{CLK}/2^5$	62.5 kHz	156 kHz	313 kHz	625 kHz	750 kHz
Other than above			Setting prohibited					

Note 1. Stop the 12-bit interval timer (by setting RINTE to 0) before changing the count clock.
Note 2. f_{IL} is selected when OSMC.WUTMMCK0 is 1 (when OSMC.WUTMMCK0 is 0, the clock is not supplied).

Caution Be sure to clear bits 7 to 3 to 0.

8.3.5 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 8 - 6 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

<table>
<tr><th>ITCMP11 to ITCMP0</th><th>Specification of the 12-bit interval timer compare value</th></tr>
<tr><td>001H</td><td rowspan="3">These bits generate a fixed-cycle interrupt (count clock cycles x (ITCMP setting + 1)).</td></tr>
<tr><td>•
•
•</td></tr>
<tr><td>FFFH</td></tr>
<tr><td>000H</td><td>Setting prohibit</td></tr>
<tr><td colspan="2">Example interrupt cycles when 001H or FFFH is specified for ITCMP11 to ITCMP0
• ITCMP11 to ITCMP0 = 001H, count clock: when f_{SUB} = 15 kHz
1/15 [kHz] × (1 + 1) ≅ 0.133 [ms] = 13.3 [μs]
• ITCMP11 to ITCMP0 = FFFH, count clock: when f_{SUB} = 15 kHz
1/15 [kHz] × (4095 + 1) ≅ 273 [ms]</td></tr>
</table>

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.

8.4 12-bit Interval Timer Operation

8.4.1 12-bit interval timer operation timing

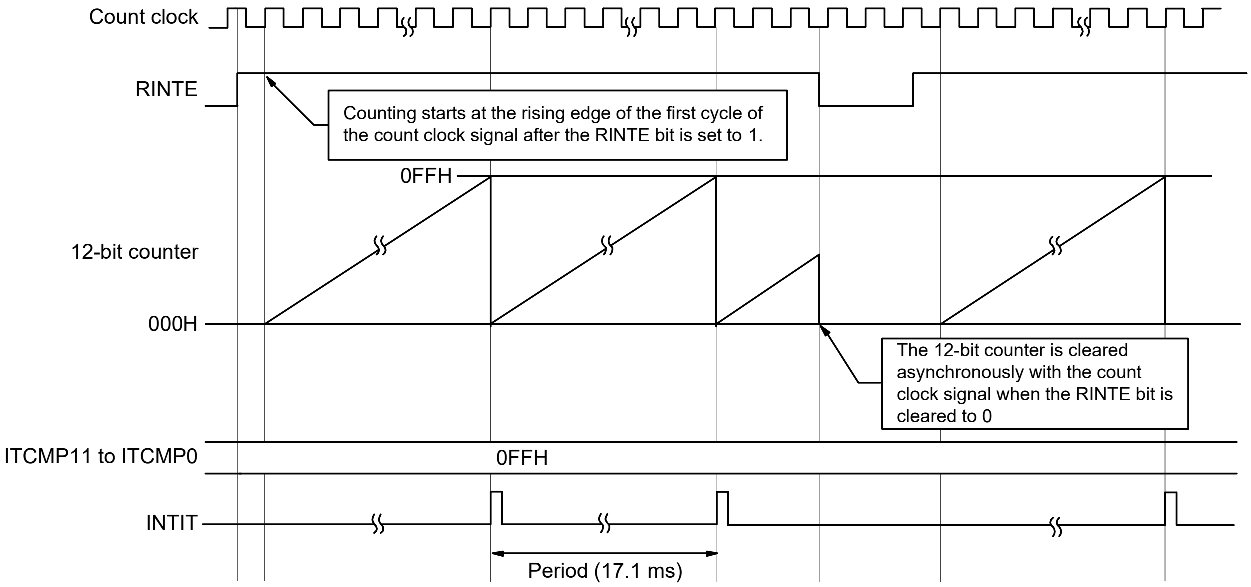
The count value specified for the ITCMP11 to ITCMP0 bits is used as an interval to operate an 12-bit interval timer that repeatedly generates interrupt requests (INTIT).

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 8 - 7 12-bit Interval Timer Operation Timing (ITCMP11 to ITCMP0 = 0FFH, count clock: fIL = 15 kHz)

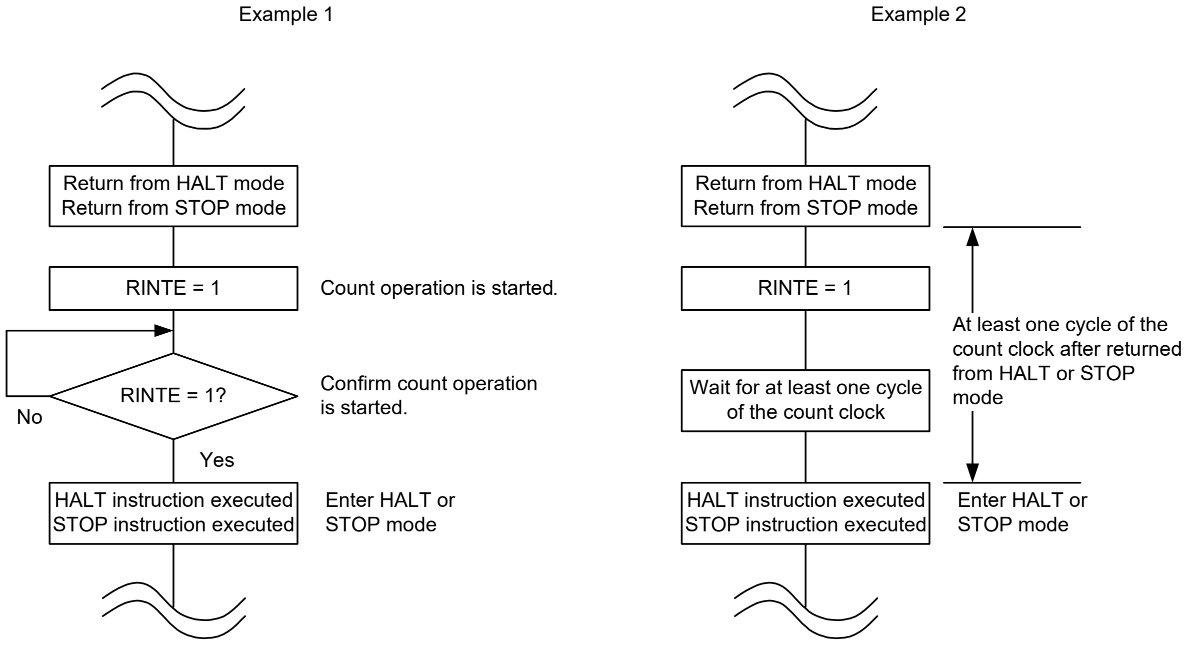

Count clock
RINTE
Counting starts at the rising edge of the first cycle of the count clock signal after the RINTE bit is set to 1.
0FFH
12-bit counter
000H
The 12-bit counter is cleared asynchronously with the count clock signal when the RINTE bit is cleared to 0
ITCMP11 to ITCMP0
0FFH
INTIT
Period (17.1 ms)

8.4.2 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 8 - 8**).
- 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 8 - 8**).

Figure 8 - 8 Procedure of entering to HALT or STOP mode after setting RINTE to 1


Example 1
Return from HALT mode
Return from STOP mode
RINTE = 1
Count operation is started.
RINTE = 1?
No
Yes
Confirm count operation is started.
HALT instruction executed
STOP instruction executed
Enter HALT or STOP mode
Example 2
Return from HALT mode
Return from STOP mode
RINTE = 1
Wait for at least one cycle of the count clock
At least one cycle of the count clock after returned from HALT or STOP mode
HALT instruction executed
STOP instruction executed
Enter HALT or STOP mode

CHAPTER 9 8-BIT INTERVAL TIMER

The 8-bit interval timer has two 8-bit timers (channel 0 and channel 1) which operate independently. The two timers can be connected to operate as a 16-bit timer.

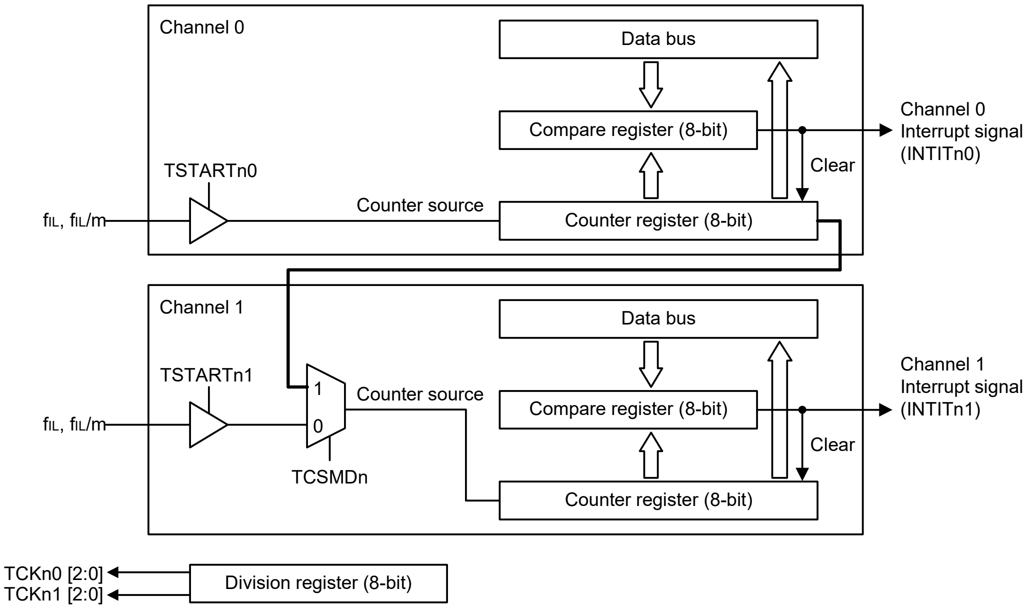
9.1 Overview

The 8-bit interval timer is an 8-bit timer that operates using the f_{IL} clock that is asynchronous with the CPU.
Table 9 - 1 lists the 8-Bit Interval Timer Specifications and Figure 9 - 1 shows the 8-Bit Interval Timer Block Diagram.

Table 9 - 1 8-Bit Interval Timer Specifications

Item	Description
Count source (operating clock)	• 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 9 - 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

9.2 I/O Pins

The 8-bit interval timer does not have an I/O pin.

9.3 Registers

Table 9 - 2 lists the 8-Bit Interval Timer Register Configuration.

Table 9 - 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

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

9.3.1 8-bit interval timer counter register ni (TRTni) (n = 0, 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 9 - 2 Format of 8-bit interval timer counter register ni (TRTni)

Address: F0500H (TRT00), F0501H (TRT01) 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 **9.4.4 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.

9.3.2 8-bit interval timer counter register n (TRTn) (n = 0)

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 9 - 3 Format of 8-bit interval timer counter register n (TRTn)

Address: F0500H (TRT0) After reset: 0000H R Notes 1, 2

	F0501H (TRT01)								F0500H (TRT00)							
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 **9.4.4 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.

9.3.3 8-bit interval timer compare register ni (TRTCMPni) (n = 0, 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 (TRT0) to 00H.

Refer to **9.4.4 Timing for Updating Compare Register Values** for the timing of rewriting the compare value.

Figure 9 - 4 Format of 8-bit interval timer compare register ni (TRTCMPni)

Address: F0350H (TRTCMP00), F0351H (TRTCMP01) 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.

9.3.4 8-bit interval timer compare register n (TRTCMPn) (n = 0)

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 **9.4.4 Timing for Updating Compare Register Values** for the timing of rewriting the compare value.

Figure 9 - 5 Format of 8-bit interval timer compare register n (TRTCMPn)

Address: F0350H (TRTCMP0) After reset: FFFFH R/W Note 2

F0351H (TRTCMP01) F0350H (TRTCMP00)

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.

9.3.5 8-bit interval timer control register n (TRTCRn) (n = 0)

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 9 - 6 Format of 8-bit interval timer control register n (TRTCRn)

Address: F0352H (TRTCR0) 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 **9.4 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 **9.4 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 **9.4 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_{IL}) have elapsed. Refer to **9.5.3 8-Bit Interval Timer Setting Procedure** for details.

Note 2. Refer to **9.5.1 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.

9.3.6 8-bit interval timer division register n (TRTMDn) (n = 0)

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 9 - 7 Format of 8-bit interval timer division register n (TRTMDn)

Address: F0353H (TRTMD0) 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	fIL
0	0	1	ffIL/2
0	1	0	fIL/4
0	1	1	fIL/8
1	0	0	fIL/16
1	0	1	fIL/32
1	1	0	fIL/64
1	1	1	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 **9.4 Operation** for details.			

TCKn0			8-bit interval timer 0 division select Notes 1, 2, 3
Bit 2	Bit 1	Bit 0	
0	0	0	fIL
0	0	1	fIL/2
0	1	0	fIL/4
0	1	1	fIL/8
1	0	0	fIL/16
1	0	1	fIL/32
1	1	0	fIL/64
1	1	1	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 **9.4 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.

9.4 Operation

9.4.1 Count Mode

The following two modes are supported: 8-bit counter mode and 16-bit counter mode. Table 9 - 3 lists the Registers and Settings Used in 8-Bit Counter Mode and Table 9 - 4 lists the Registers and Settings Used in 16-Bit Counter Mode.

Table 9 - 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

Table 9 - 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.

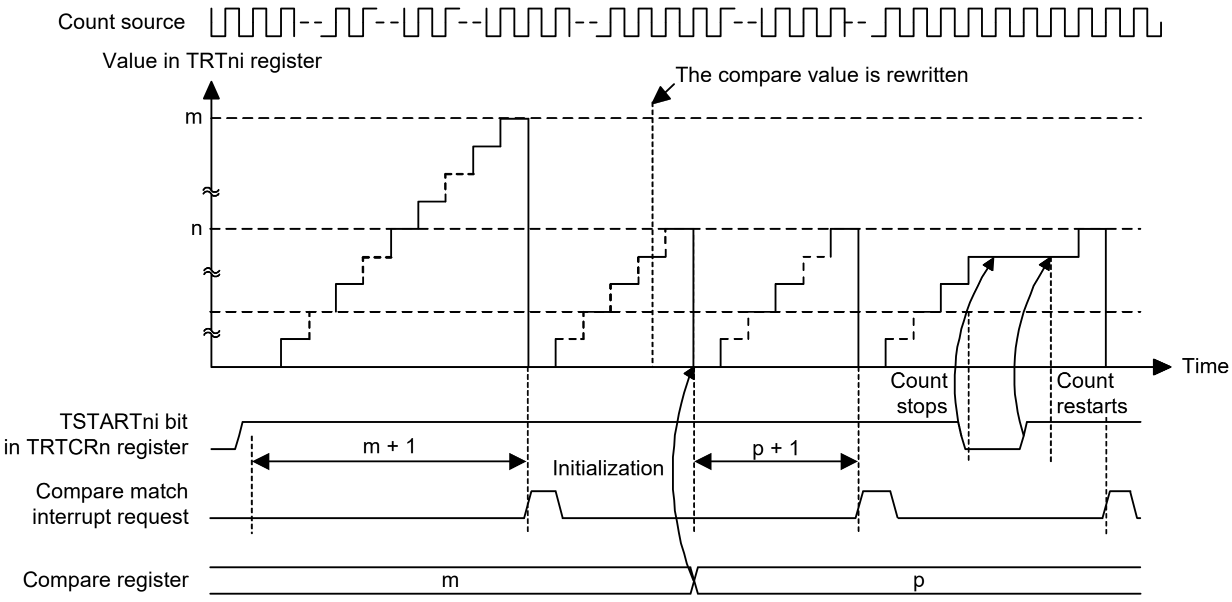
Remark n = 0

9.4.2 Timer Operation

The counter is incremented by the count source selected by the TCKni (n = 0, i = 0, 1) bit in the division register (TRTMDn). The count value is decremented each time the count source is input. After the count value is set to the compare value, the value is compared and matched when the next count source is input, and then an interrupt is generated. The interrupt request is output with a single pulse that is synchronized with the count source. 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 continues retaining the count value immediately before operation is stopped. To clear the count value, set the compare value in the TRTCMPni register again. After the TRTCMPni register is written, the count value is cleared after two cycles of the count source.

Figure 9 - 8 Example of Timer Operation



Remark n = 0 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 (f_{IL}) is selected
  Maximum: Two cycles of the count source
  Minimum: One cycle of the count source
- When the count source ($f_{IL}/2^m$) is selected
  Maximum: One cycle of the count source
  Minimum: One cycle of the selected clock (f_{IL})

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 9 - 5 lists the Interrupt Sources in 8-Bit/16-Bit Count Mode.

Table 9 - 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

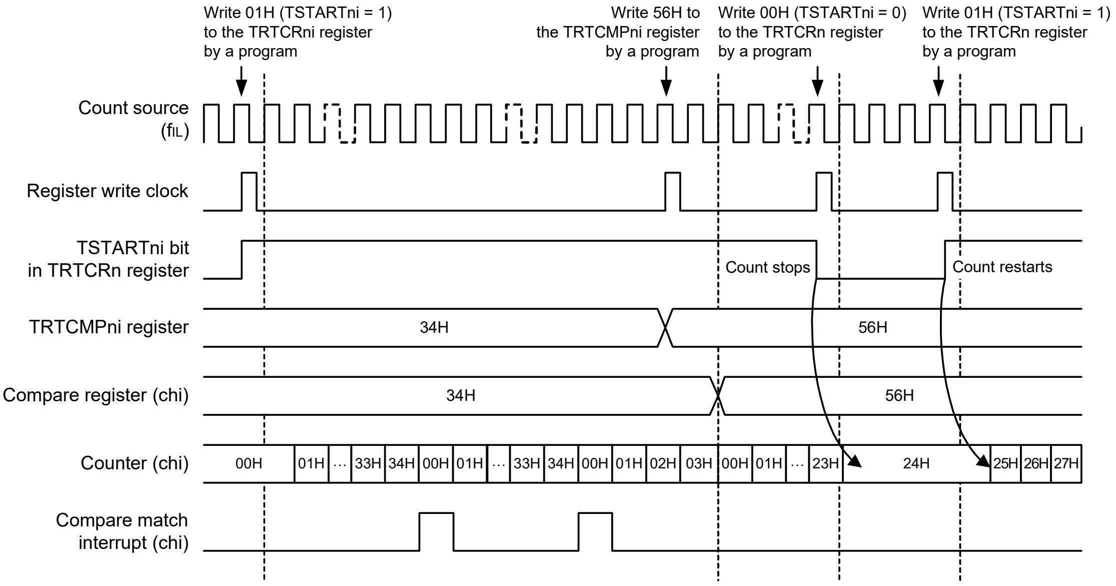
9.4.3 Start/Stop Timing

9.4.3.1 When Count Source (fIL) is Selected

After 1 is written to the TSTARTni (n = 0, i = 0, 1) bit in the TRTCRn register, the count is started by the next low-speed on-chip oscillator clock (fIL), and then the counter is incremented from 00H to 01H by the next count source (fIL). Likewise, after 0 is written to the TSTARTni bit, the count is stopped after the counter is incremented by the low-speed on-chip oscillator clock (fIL).

Figure 9 - 9 shows the timing for starting/stopping count operation, and Figure 9 - 10 shows the timing of count stop → compare setting (count clearing) → count start. Figure 9 - 9 and Figure 9 - 10 show the update timing in 8-bit counter mode, but operation is performed at the same timing even in 16-bit counter mode.

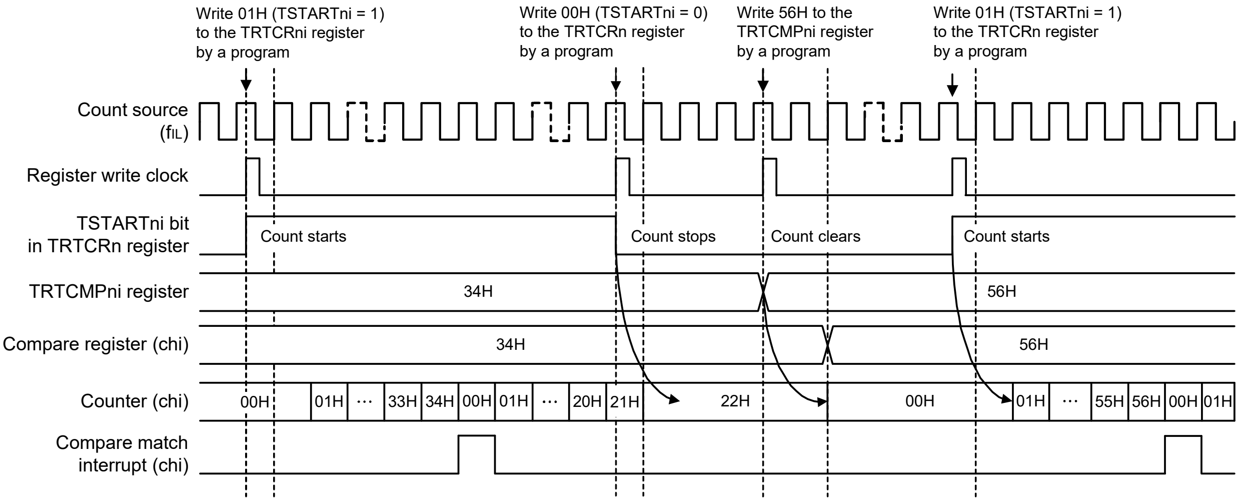
Figure 9 - 9 Example of Count Start/Stop Operation (fIL Selected)


Write 01H (TSTARTni = 1) to the TRTCRni register by a program
Write 56H to the TRTCMPni register by a program
Write 00H (TSTARTni = 0) to the TRTCRn register by a program
Write 01H (TSTARTni = 1) to the TRTCRn register by a program
Count source (fIL)
Register write clock
TSTARTni bit in TRTCRn register
Count stops
Count restarts
TRTCMPni register
34H
56H
Compare register (chi)
34H
56H
Counter (chi)
00H
01H
...
33H
34H
00H
01H
...
33H
34H
00H
01H
02H
03H
00H
01H
...
23H
24H
25H
26H
27H
Compare match interrupt (chi)

The TCSMDn bit in the TRTCRn register is set to 0 (8-bit counter operation)

Remark n = 0
i = 0, 1

Figure 9 - 10 Example of Count Stop → Count Clearing → Count Start Operation (f_{IL} Selected)



The TCSMDn bit in the TRTCRn register is set to 0 (8-bit counter operation)

Remark n = 0
i = 0, 1

9.4.3.2 When Count Source ($f_{IL}/2^m$) is Selected

After 1 is written to the TSTARTni (n = 0, i = 0, 1) bit in the TRTCRn register, the count is started with the next low-speed on-chip oscillator clock (f_{IL}), and then the counter is incremented from 00H to 01H by the next count source ($f_{IL}/2^m$). Likewise, after 0 is written to the TSTARTni bit, the count is stopped with the low-speed on-chip oscillator clock (f_{IL}).

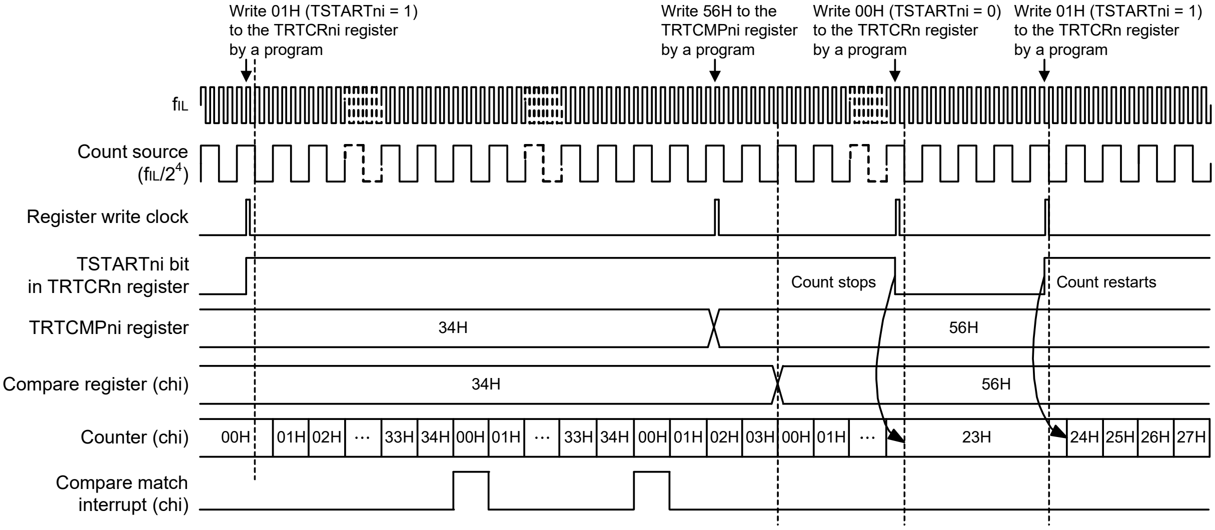
However, the first period to count 00H when the timer starts counting is shorter than one cycle of the count source as below, depending on the timing for writing to the TSTARTni bit and the timing of the next count source.

Minimum: One cycle of the low-speed on-chip oscillator clock (f_{IL})

Maximum: One cycle of the count source

Figure 9 - 11 shows the timing for starting/stopping count operation, and Figure 9 - 12 shows the timing of count stop → compare setting (count clearing) → count start. Figure 9 - 11 and Figure 9 - 12 show the update timing in 8-bit counter mode, but operation is performed at the same timing even in 16-bit counter mode.

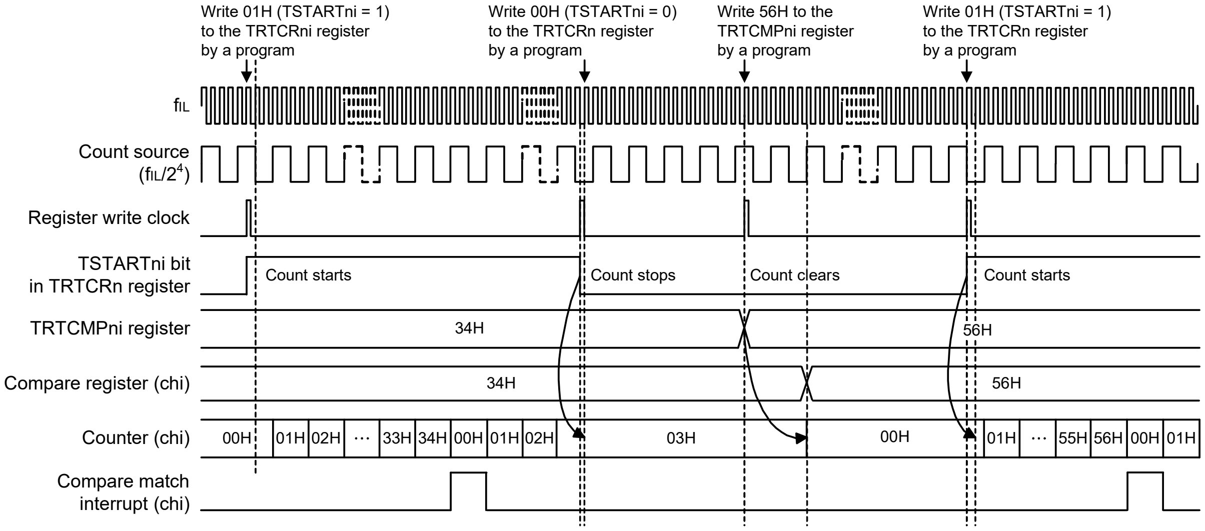
Figure 9 - 11 Example of Count Start/Stop Operation ($f_{IL}/2^m$ Selected)


Write 01H (TSTARTni = 1) to the TRTCRni register by a program
Write 56H to the TRTCMPni register by a program
Write 00H (TSTARTni = 0) to the TRTCRn register by a program
Write 01H (TSTARTni = 1) to the TRTCRn register by a program
fIL
Count source (fIL/2^4)
Register write clock
TSTARTni bit in TRTCRn register
Count stops
Count restarts
TRTCMPni register
34H
56H
Compare register (chi)
34H
56H
Counter (chi)
00H
01H
02H
···
33H
34H
00H
01H
···
33H
34H
00H
01H
02H
03H
00H
01H
···
23H
24H
25H
26H
27H
Compare match interrupt (chi)

The TCSMDn bit in the TRTCRn register is set to 0 (8-bit counter operation)

Remark n = 0
i = 0, 1

Figure 9 - 12 Example of Count Stop → Count Clearing → Count Start Operation ($f_{IL}/2^m$ Selected)



The TCSMDn bit in the TRTCRn register is set to 0 (8-bit counter operation)

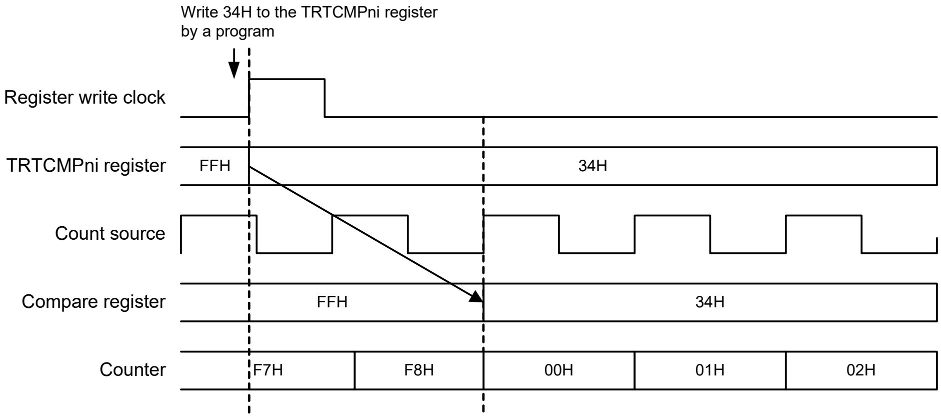
Remark n = 0
i = 0, 1

9.4.4 Timing for Updating Compare Register Values

The timing for updating the value of the TRTCMPni (n = 0, 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 9 - 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 9 - 13 Timing of Compare Value Rewrite Operation

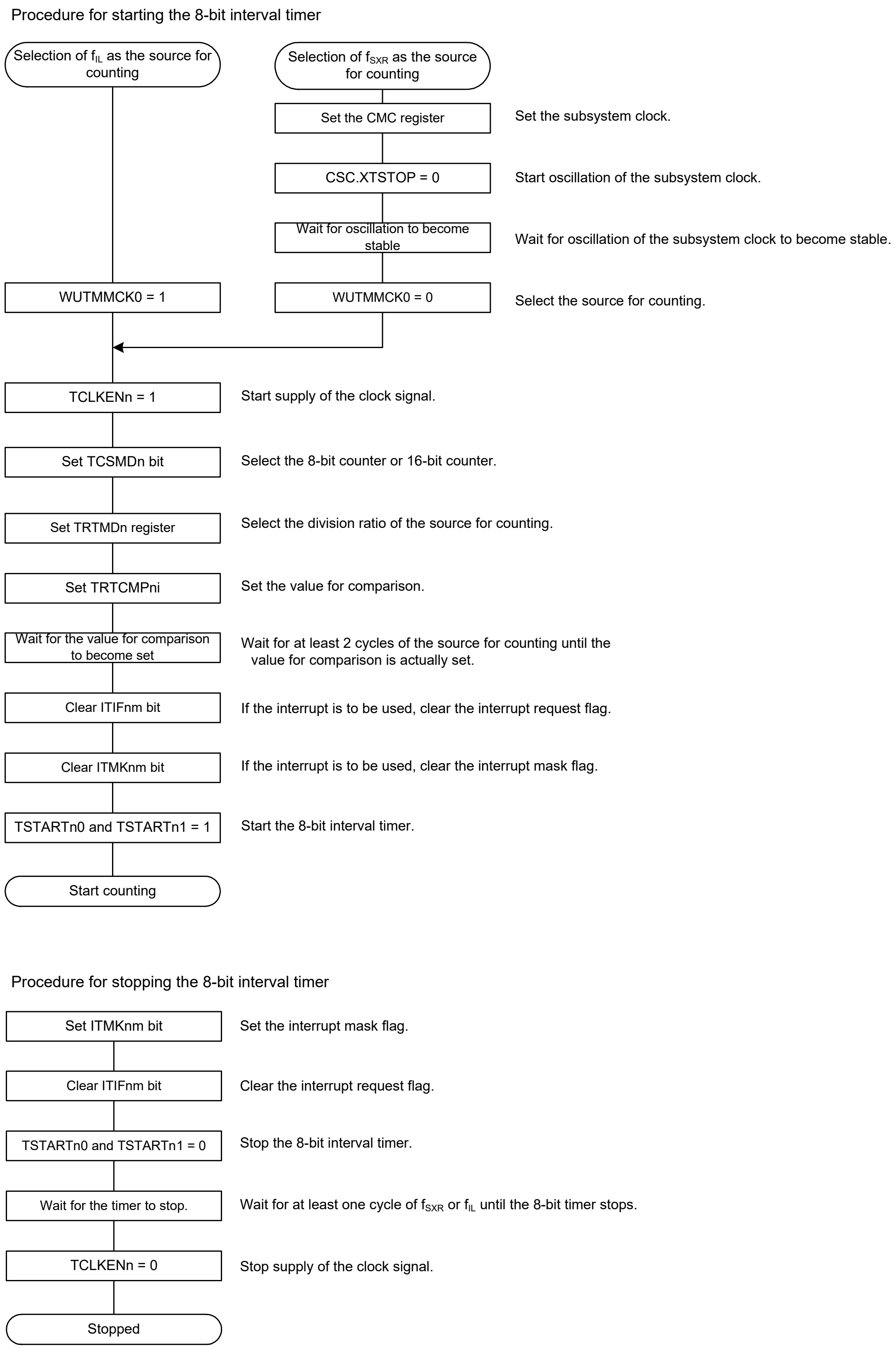


Remark n = 0
i = 0, 1

9.4.5 Procedure for Setting the 8-bit Interval Timer

The procedure for setting the 8-bit interval timer is shown below.

Figure 9 - 14 Procedure for Setting the 8-bit Interval Timer


Selection of f_{IL} as the source for counting
Selection of f_{SXR} as the source for counting
Set the CMC register
CSC.XTSTOP = 0
Wait for oscillation to become stable
WUTMMCK0 = 1
WUTMMCK0 = 0
TCLKENn = 1
Set TCSMDn bit
Set TRTMDn register
Set TRTCMPni
Wait for the value for comparison to become set
Clear ITIFnm bit
Clear ITMKnm bit
TSTARTn0 and TSTARTn1 = 1
Start counting
Set ITMKnm bit
Clear ITIFnm bit
TSTARTn0 and TSTARTn1 = 0
Wait for the timer to stop.
TCLKENn = 0
Stopped

Procedure for starting the 8-bit interval timer

Set the subsystem clock.

Start oscillation of the subsystem clock.

Wait for oscillation of the subsystem clock to become stable.

Select the source for counting.

Start supply of the clock signal.

Select the 8-bit counter or 16-bit counter.

Select the division ratio of the source for counting.

Set the value for comparison.

Wait for at least 2 cycles of the source for counting until the value for comparison is actually set.

If the interrupt is to be used, clear the interrupt request flag.

If the interrupt is to be used, clear the interrupt mask flag.

Start the 8-bit interval timer.

Procedure for stopping the 8-bit interval timer

Set the interrupt mask flag.

Clear the interrupt request flag.

Stop the 8-bit interval timer.

Wait for at least one cycle of f_{SXR} or f_{IL} until the 8-bit timer stops.

Stop supply of the clock signal.

9.5 Notes on 8-Bit Interval Timer

9.5.1 Changing Settings of Operating Mode

The settings of bits TCSMDn and TCKni (n = 0, 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_{IL} to elapse before accessing the registers (TRTCRn and TRTMDn) associated with the 8-bit interval timer.

9.5.2 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.

9.5.3 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_{IL} to elapse before setting the TCLKENn bit to 0.

CHAPTER 10 16-BIT TIMER KB0

16-bit timer KB0 is a timer that can generate PWM output which is suitable for power sources and lighting control.

10.1 Functions of 16-bit Timer KB0

16-bit timers KB0 is dedicated PWM output timer and have two outputs each, enabling the generation of up to two PWM outputs. Complementary PWM output can also be generated to control a half-bridge circuit (2 outputs). Also, by linking with a comparator, INTP10, or INTP11, PWM output can be stopped urgently.
16-bit timer KB0 is provided with the following functions.

(1) PWM output
•The duty and the cycle of PWM output can be changed during timer operation.
•The default level while the timer is stopped and the active level while the timer is operating can be set to high level and low level, respectively.

(2) Trigger output (ELC event generation signal output)
Can be output to the ELC event generation source using the 16-bit timer KB trigger compare register 0 (TKBTGCR0).

(3) Timer restart function (by interlocking with the comparator and INTP)
Timer output can be restarted directly (not via the CPU) when a trigger source occurs (comparator 0 and 1 output, INTP10, INTP11). By using this function, critical conduction mode PFC control can be implemented, for example.

(4) Forced output stop function 1 (by interlocking with the comparator)
Timer output can be fixed to Hi-Z, high, or low level directly (not via the CPU) and asynchronously with the operation clock f_{KBKC} of the 16-bit timer KBn circuit when a trigger source occurs (comparator 0 and 1 output). The forced output stop status is canceled synchronously with the operation clock f_{KBKC} of the 16-bit timer KBn circuits by setting the stop trigger of forced output stop function 1.

(5) Forced output stop function 2 (by interlocking with the comparator and INTP)
Timer output can be fixed to high or low level directly (not via the CPU) and asynchronously with the operation clock f_{KBKC} of the 16-bit timer KBn circuit when a trigger source occurs (comparator 0 and 1 output, INTP 10, INTP11). The forced output stop status is canceled at the beginning of the next counter cycle after the trigger source occurs or after the trigger source signal changes to inactive level.

(6) PWM output dithering function
The “set duty + 1” waveform in each 16-period cycle can be output in the range of periods 0 to 15. By using this function, PWM that is 16 times the count clock can be output as the average resolution of 16 timer KBn cycles.

(7) PWM output smooth start function
It is possible to make a soft-start that automatically increases duty after PWM output starts until it reaches to the configured duty value.
It is possible to configure initial duty and duty plus one incremental period.

(8) Maximum frequency limit function

When using the timer restart function, if a trigger occurs earlier than the set maximum frequency, restart can be suspended until the set maximum frequency.

(9) Interleave PFC output mode

With the timer restart function, it is possible to use external factors to automatically alternate restart output between two outputs. It is possible to make interleaved PFC control with Critical Conduction Mode.

Remark Critical Conduction Mode is a PFC control method that activates a switching FET by detecting zero level of inductor current.

10.2 Configuration of 16-bit Timers KB0

16-bit timers KB0 include the following hardware.

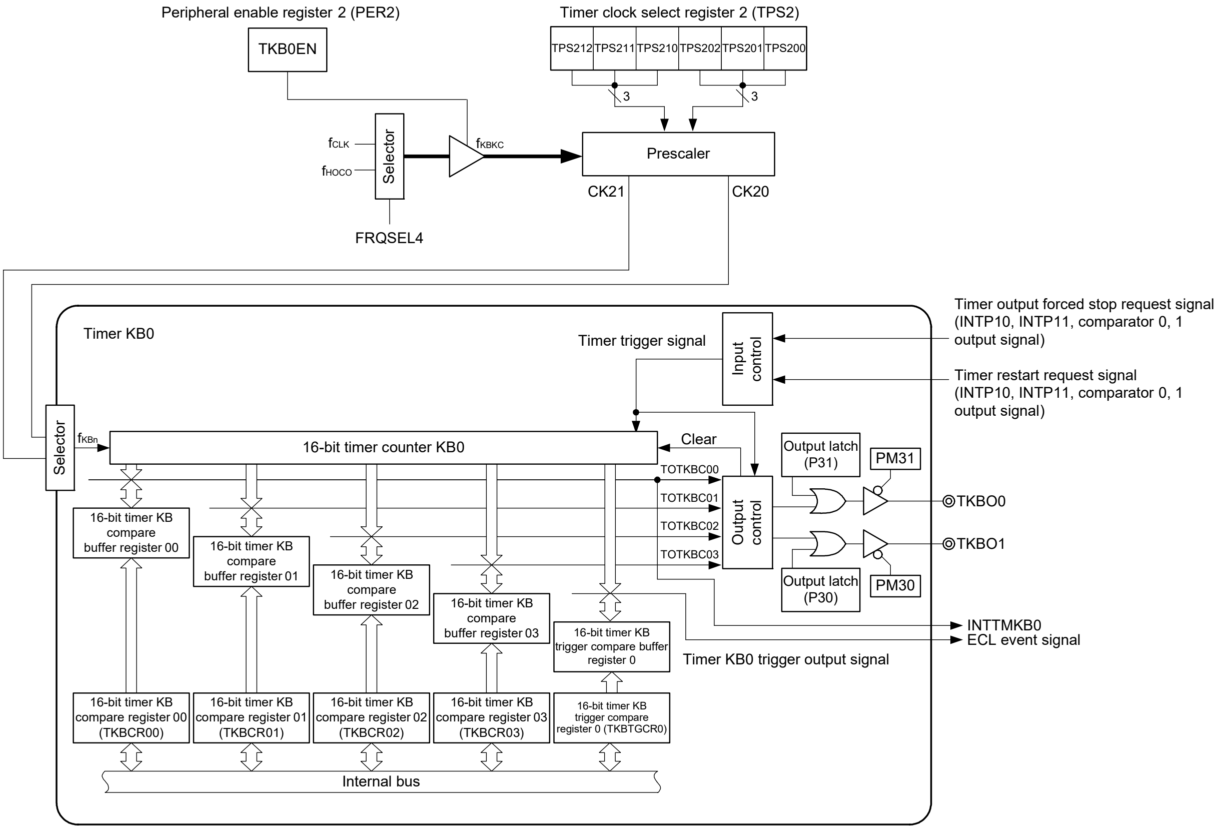
Table 10 - 1 Configuration of 16-bit Timers KB0

Item	Configuration
Timer/counter	16-bit timer counter KBn (TKBCNTn)
Registers	16-bit timer KB compare registers n0 to n3 (TKBCRn0 to TKBCRn3) 16-bit timer KB trigger compare register n (TKBTGCRn)
Timer output	TKBO0, TKBO1
Control registers	Peripheral enable register 2 (PER2) Peripheral reset control register 2 (PRR2) Timer clock select register 2 (TPS2) 16-bit timer KB operation control register n0 (TKBCTLn0) 16-bit timer KB operation control register n1 (TKBCTLn1) 16-bit timer KB output control register n0 (TKBIOCn0) 16-bit timer KB output control register n1 (TKBIOCn1) 16-bit timer KB flag register n (TKBFLGn) 16-bit timer KB trigger register n (TKBTRGn) 16-bit timer KB flag clear trigger register n (TKBCLRn) 16-bit timer KB dithering count registers n0, n1 (TKBDNRn0, TKBDNRn1) 16-bit timer KB compare 1L & dithering count register n0 (TKBCRLDn0) 16-bit timer KB compare 3L & dithering count register n1 (TKBCRLDn1) 16-bit timer KB smooth start initial duty registers n0, n1 (TKBSIRn0, TKBSIRn1) 16-bit timer KB smooth start step width registers n0, n1 (TKBSSRn0, TKBSSRn1) 16-bit timer KB maximum frequency limit setting register n (TKBMFRn) Peripheral function switch register 0 (PFSEL0) External interrupt edge enable register (INTPEG) Port mode control register 3 (PMC3) Port mode register 3 (PM3) Port register 3 (P3)

Remark n = 0

Figure 10 - 1 shows a block diagram.

Figure 10 - 1 Block Diagram of 16-bit Timer KB0


Peripheral enable register 2 (PER2)
TKB0EN
Timer clock select register 2 (TPS2)
TPS212
TPS211
TPS210
TPS202
TPS201
TPS200
3
3
fCLK
fHOCO
Selector
FRQSEL4
fKBKC
Prescaler
CK21
CK20
Timer KB0
Timer trigger signal
Input control
Timer output forced stop request signal (INTP10, INTP11, comparator 0, 1 output signal)
Timer restart request signal (INTP10, INTP11, comparator 0, 1 output signal)
Selector
fKBn
16-bit timer counter KB0
Clear
TOTKBC00
TOTKBC01
TOTKBC02
TOTKBC03
Output control
Output latch (P31)
PM31
TKBO0
TKBO1
Output latch (P30)
PM30
INTTMKB0
ECL event signal
Timer KB0 trigger output signal
16-bit timer KB compare buffer register 00
16-bit timer KB compare buffer register 01
16-bit timer KB compare buffer register 02
16-bit timer KB compare buffer register 03
16-bit timer KB trigger compare buffer register 0
16-bit timer KB compare register 00 (TKBCR00)
16-bit timer KB compare register 01 (TKBCR01)
16-bit timer KB compare register 02 (TKBCR02)
16-bit timer KB compare register 03 (TKBCR03)
16-bit timer KB trigger compare register 0 (TKBTGCR0)
Internal bus

Remark 1. fKBKC: Operation clock of whole 16-bit timer KBn and KC0 circuit

fKBn: Count clock of 16-bit timer KBn

Remark 2. n = 0

10.2.1 16-bit timer KB compare registers n0 to n3 (TKBCRn0 to TKBCRn3)

TKBCRnm can be refreshed (writing the same value) and its value can be rewritten while the timer is counting (TKBCEn = 1). When the value of TKBCRnm is rewritten while the timer is operating, that value is latched, transferred to TKBCRnm at the following timing, and the value of TKBCRnm is changed.

- When starting count operation of counter (TKBCEn = 0)
- When a batch overwrite trigger (TKBRDTn = 1) or an external trigger (TKBTSEn = 1) occurs

This register can be read or written in 16-bit units.

Reset signal generation clears this register to 0000H.

Figure 10 - 2 Format of 16-bit Timer KB Compare Register nm (TKBCRnm)

Address: F0400H (TKBCR00), 0402H (TKBCR01), 0404H (TKBCR02), F0406H (TKBCR03)
After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TKBCRnm																

Remark n = 0, m = 0 to 3

10.2.2 16-bit timer KB trigger compare register n (TKBTGCRn)

TKBTGCRn can be refreshed (writing the same value) and its value can be rewritten while the timer is counting (TKBCEn = 1). When the value of TKBTGCRn is rewritten while the timer is operating, that value is latched, transferred to TKBTGCRn at the following timing, and the value of TKBTGCRn is changed.

- When starting count operation of counter (TKBCEn = 0)
- When a batch overwrite trigger (TKBRDTn = 1) or an external trigger (TKBTSEn = 1) occurs

Periodic signals from this register can be used as a hardware trigger for A/D conversion.

This register can be read or written in 16-bit units.

Reset signal generation clears this register to 0000H.

Figure 10 - 3 Format of 16-bit Timer KB Trigger Compare Register n (TKBTGCRn)

Address: F0408H (TKBTGCR0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TKBTGCRn																

Remark n = 0

10.3 Registers Controlling 16-bit Timers KB0

16-bit timers KB0 are controlled by the following registers.

- Peripheral enable register 2 (PER2)
- Peripheral reset control register 2 (PRR2)
- Timer clock select register 2 (TPS2)
- 16-bit timer KB operation control registers n0, n1 (TKBCTLn0, TKBCTLn1)
- 16-bit timer KB output control registers n0, n1 (TKBIOCn0, TKBIOCn1)
- 16-bit timer KB flag register n (TKBFLGn)
- 16-bit timer KB trigger register n (TKBTRGn)
- 16-bit timer KB flag clear trigger register n (TKBCLRn)
- 16-bit timer KB dithering count registers n0, n1 (TKBDNRn0, TKBDNRn1)
- 16-bit timer KB compare 1L & dithering count register n0 (TKBCRLDn0)
- 16-bit timer KB compare 3L & dithering count register n1 (TKBCRLDn1)
- 16-bit timer KB smooth start initial duty registers n0, n1 (TKBSIRn0, TKBSIRn1)
- 16-bit timer KB smooth start step width registers n0, n1 (TKBSSRn0, TKBSSRn1)
- 16-bit timer KB maximum frequency limit setting register n (TKBMFRn)
- Peripheral function switch register 0 (PFSEL0)
- External interrupt edge enable register (INTPEG)
- Port mode control register 3 (PMC3)
- Port mode register 3 (PM3)
- Port register 3 (P3)

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 timers KB0 are used, be sure to set bits 0 (TKB0EN) of this register to 1.

The PER2 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10 - 4 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	0	DOCEN	0	0	0	0	TKB0EN

TKB0EN	Control of timer KB0 input clock
0	Stops supply of input clock. • SFR used by timer KB0 cannot be written.
1	Supplies input clock. • SFR used by timer KB0 can be read/written.

Caution When setting timer KB0, be sure to set the TKB0EN bit to 1 first. If TKB0EN = 0, writing to a control register of timer KB0 is ignored, and all read values are default values (except for timer clock select register 2 (TPS2), port mode control register 3 (PMC3), port mode register 3 (PM3), and port register 3 (P3)).

10.3.2 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 16-bit timer KB0, be sure to set bit 0 (TKB0RES) 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 - 5 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	TKB0RES

TKB0RES	Reset control of 16-bit timer KB0
0	16-bit timer KB0 is released from the reset state.
1	16-bit timer KB0 is in the reset state.

10.3.3 Timer clock select register 2 (TPS2)

The TPS2 register is a 16-bit register that is used to select two types of operation clocks (CK20, CK21) that are commonly supplied to timers KB0 from external prescaler. CK21 is selected by using bits 6 to 4 of the TPS2 register, and CK20 is selected by using bits 2 to 0.

Rewriting of the TPS2 register during timer operation is possible only in the following cases.

If the TPS200 to TPS202 bits can be rewritten (n = 0):

All timers for which CK20 is selected as the operation clock (TKBCKSn = 0) are stopped (TKBCEn = 0).

If the TPS210 to TPS212 bits can be rewritten (n = 0):

All timers for which CK21 is selected as the operation clock (TKBCKSn = 1) are stopped (TKBCEn = 0).

The TPS2 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10 - 6 Format of Timer Clock Select Register 2 (TPS2)

Address: F02D4H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TPS2	0	TPS 212	TPS 211	TPS 210	0	TPS 202	TPS 201	TPS 200

- Option byte (000C2H/010C2H) FRQSEL4 = 0

TPS 2k2	TPS 2k1	TPS 2k0	Selection of operation clock (CK2k) (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	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz	24 MHz
0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz	12 MHz
0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz	6 MHz
0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz	3 MHz
1	0	0	$f_{CLK}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz	1.5 MHz
1	0	1	$f_{CLK}/2^5$	62.5 kHz	156.2 kHz	312.5 kHz	625 kHz	750 kHz
Other than the above			Setting prohibited					

- Option byte (000C2H/010C2H) FRQSEL4 = 1

TPS2k2	TPS2k1	TPS2k0	Selection of operation clock (CK2k)Note (k = 0, 1)				
				f_{HOCO} = 6 MHz	f_{HOCO} = 12 MHz	f_{HOCO} = 24 MHz	f_{HOCO} = 48 MHz
0	0	0	f_{HOCO}	6 MHz	12 MHz	24 MHz	48 MHz
0	0	1	$f_{HOCO}/2$	3 MHz	6 MHz	12 MHz	24 MHz
0	1	0	$f_{HOCO}/2^2$	1.5 MHz	3 MHz	6 MHz	12 MHz
0	1	1	$f_{HOCO}/2^3$	750 kHz	1.5 MHz	3 MHz	6 MHz
1	0	0	$f_{HOCO}/2^4$	375 kHz	750 kHz	1.5 MHz	3 MHz
1	0	1	$f_{HOCO}/2^5$	187.5 kHz	375 kHz	750 kHz	1.5 MHz
Other than the above			Setting prohibited				

Caution 1. When changing the clock selected for fCLK (by changing the system clock control register (CKC) value), stop timers KB0 (TKBCEn = 0).

Caution 2. Be sure to clear bits 7 and 3 to "0".

Remark f_{CLK}: CPU/peripheral hardware clock frequency
f_{HOCO}: High-speed on-chip oscillator clock frequency

10.3.4 16-bit timer KB operation control register n0 (TKBCTLn0)

TKBCTLn0 is a register that controls smooth start function, dithering function, maximum frequency limit function, Interleave PFC output mode, compare register batch overwrite function set by external trigger and timer KB0 restart trigger.

TKBCTLn0 can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 10 - 7 Format of 16-bit Timer KB Operation Control Register n0 (TKBCTLn0) (1/2)

Address: F0422H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
TKBCTLn0	0	0	TKBSSEn1	TKBDIEn1	0	0	TKBSSEn0	TKBDIEn0
	7	6	5	4	3	2	1	0
	TKBMFEn	0	TKBIRSn1	TKBIRSn0	0	TKBTSEn	TKBSTSn1	TKBSTSn0

TKBSSEnp	Control of TKBOp PWM output smooth start function
0	Does not use PWM output smooth start function.
1	Use PWM output smooth start function.

TKBDIEnp	Control of TKBOp PWM output dithering function
0	Does not use PWM output dithering function.
1	Use PWM output dithering function.

TKBMFEn	Control of TKBO0 and TKBO1 maximum frequency limit function
0	Does not use the maximum frequency limit function.
1	Use the maximum frequency limit function.

TKBISRn1	TKBISRn0	Configuration of acceptable range of INTP11 input that immediately outputs TKBO1 in interleaved PFC output mode.
0	0	T/2 to T/2+T/64
0	1	T/2 to T/2+T/32
1	0	T/2 to T/2+T/16
1	1	T/2 to T/2+ T/8

TKBTSEn	Control of compare register batch overwrite function set by external trigger
0	Does not use compare register batch overwrite function set by external trigger.
1	Use compare register batch overwrite function set by external trigger.

Remark 1. n = 0, p = 0, 1

Remark 2. T is the period of the last restart

Figure 10 - 7 Format of 16-bit Timer KB Operation Control Register n0 (TKBCTLn0) (2/2)

TKBSTSn1	TKBSTSn0	Selection of timer KBn count start trigger
0	0	Does not use trigger input.
0	1	External interrupt signal (INTP10)
1	0	External interrupt signal (INTP11)
1	1	Comparator detection signal specified by CTRGSEL1 and CTRGSEL0 bits of the PFSEL0 register[Note]

Note When comparator detection signal is used as a count start trigger, set the CiOE bit of the comparator output control register (COMPOCR) to 1. For detail, see 19.5 Caution for Using Timer KB Simultaneous Operation Function.

Caution 1. During timer operation, setting the other bits of the TKBCTLn0 register is prohibited. However, the TKBCTLn0 register can be refreshed (the same value is written).

Caution 2. Be sure to clear bits 15, 14, 11, 10, 6, and 3 to “0”.

Caution 3. For setting of INTP10/INTP11, see CHAPTER 19 COMPARATOR.

Remark i = 0, 1, n = 0, p = 0, 1

10.3.5 16-bit timer KB operation control register n1 (TKBCTLn1)

TKBCTLn1 is a register that controls the count operation and sets the count clock of 16-bit timer.

TKBCTLn1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10 - 8 Format of 16-bit Timer KB Operation Control Register n1 (TKBCTLn1)

Address: F0429H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
TKBCTLn1	TKBCE0	0	0	TKBCKS0	0	0	TKBMD01	TKBMD00

TKBCEn	Timer KBn operation control
0	Stops timer operation (counter is set to FFFF).
1	Enables timer count operation.

TKBCKS0	Timer KB0 clock selection
0	CK20 clock selected by TPS202 to TPS200 bits
1	CK21 clock selected by TPS212 to TPS210 bits

TKBMDn1	TKBMDn0	Timer KBn operating mode selection
0	0	Standalone mode (uses master)
1	1	Interleave PFC output mode
Other than the above		Setting prohibited

Caution 1. During timer operation, setting the other bits of the TKBCTLn1 register is prohibited. However, the TKBCTLn1 register can be refreshed (the same value is written).

Caution 2. In TKBCTLn1, be sure to clear bits 6, 5, 3, and 2 to “0”.

Remark n = 0

10.3.6 16-bit timer KB output control register n0 (TKBIOCn0)

TKBIOCn0 is a register that setting the default level/active level in 16-bit timer KBn output (TKBOp).

TKBIOCn0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10 - 9 Format of 16-bit Timer KB Output Control Register n0 (TKBIOCn0)

Address: F0426H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	<1>	<0>
TKBIOCn0	0	0	0	0	TKBTOLn1	TKBTOLn0	TKBTODn1	TKBTODn0

TKBTOLnp	Active level setting of timer output TKBOp
0	High level
1	Low level

TKBTODnp	Default level setting of timer output TKBOp
0	Low level
1	High level

Caution 1. During timer operation, setting the other bits of the TKBIOCn0 register is prohibited. However, the TKBIOCn0 register can be refreshed (the same value is written).

Caution 2. Be sure to clear bits 7 to 4 to “0”.

Caution 3. Actual TKBOp pin output is set not only by TKBOp output but by the port mode registers (PMxx) and port registers (Pxx) for the shared ports.

Remark n = 0, p = 0, 1

10.3.7 16-bit timer KB output control register n1 (TKBIOCn1)

TKBIOCn1 is a register that controls disable/enable timer control in 16-bit timer KBn output (TKBOp).

TKBIOCn1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10 - 10 Format of 16-bit Timer KB Output Control Register n1 (TKBIOCn1)

Address: F0428H After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
TKBIOCn1	0	0	0	0	0	0	TKBTOEn1	TKBTOEn0

TKBTOEnp	Timer output TKBOp output enable/disable
0	Disables timer output (low-level output when TKBTODnp = 0, and high-level output when TKBTODnp = 1.)
1	Enables timer output

Caution 1. The TKBIOCn1 register can be overwritten while the timer is operating.

Caution 2. Be sure to clear bits 7 to 2 to “0”.

Caution 3. ctual TKBOp pin output is set not only by TKBOp output but by the port mode registers (PMxx) and port registers (Pxx) for the shared ports.

Remark n = 0, p = 0, 1

10.3.8 16-bit timer KB flag register n (TKBFLGn)

TKBFLGn is a register with status flags for 16-bit timer KBn.

TKBFLGn can be read by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10 - 11 Format of 16-bit Timer KB Flag Register n (TKBFLGn)

Address: F0413H After reset: 00H R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
TKBFLGn	TKBSSFn1	TKBSSFn0	TKBSEFn1	TKBSEFn0	TKBIRFn	TKBIEFn	TKBMFFn	TKBRSFn

TKBSSFnp	Status flag for PWM output smooth start function of TKBOp pin
0	During stop in PWM output smooth start function
1	Executing in PWM output smooth start function

TKBSEFnp	Error flag for PWM output smooth start function of TKBOp pin
0	No error, or completion of clearing by TKBCLSEnp
1	Error (TKBRDTnp = 1 occurred during PWM output smooth start execute (TKBSSFnp = 1))

TKBIRFn	Undetected INTP11 trigger error flag for interleave PFC mode
0	No error, or completion of clearing by TKBCLIRn
1	Error (Undetected INTP11 trigger is in judgment range set by 0 to T/2 and TKBIRSn1 and TKBIRSn0)

TKBIEFn	Multiplex detection INTP11 trigger error flag for interleave PFC mode
0	No error, or completion of clearing by TKBCLIEn
1	Error (Another INTP11 trigger was detected during the TKBO1 active output)

TKBMFFn	Status flag for maximum frequency limit function
0	Maximum frequency limit function is not occurred, or completion of clearing by TKBCLMFn
1	Maximum frequency limit function is occurred

TKBRSFn	Pending status flag for batch overwrite trigger
0	Batch overwrite enabled status or completion of batch overwrite caused by to batch overwrite trigger
1	On hold (waiting for completion) status of batch overwrite due to writing on batch overwrite trigger bit TKBRDTn.

Remark 1. n = 0, p = 0, 1

Remark 2. T is the period of the last restart

10.3.9 16-bit timer KB trigger register n (TKBTRGn)

TKBTRGn is a trigger register used for batch overwriting of the compare register for 16-bit timer KBn.

TKBTRGn can be written by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10 - 12 Format of 16-bit Timer KB Trigger Register n (TKBTRGn)

Address: F0412H After reset: 00H W

Symbol	7	6	5	4	3	2	1	<0>
TKBFLGn	0	0	0	0	0	0	0	TKBRDTn

TKBRDTn	Trigger for batch overwrite request of compare register
0	Invalid setting
1	Batch overwrite request of compare register

Remark n = 0

10.3.10 16-bit timer KB flag clear trigger register n (TKBCLRn)

TKBCLRn is a register used to clear flags in the 16-bit timer KB flag register n (TKBFLGn).

TKBCLRn can be written by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10 - 13 Format of 16-bit Timer KB Flag Clear Trigger Register n (TKBCLRn)

Address: F0427H After reset: 00H W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	0
TKBCLRn	0	0	TKBCLSEn1	TKBCLSEn0	TKBCLIRn	TKBCLIEn	TKBCLMFn	0

TKBCLSEnp	Trigger for clearing error flag for PWM output smooth start function of TKBOp pin
0	Invalid setting
1	Clear the TKBSEFnp flag to “0”.

TKBCLIRn	Trigger for clearing undetected INTP11 trigger error flag for interleave PFC mode
0	Invalid setting
1	Clear the TKBIRFn flag to “0”.

TKBCLIEn	Trigger for clearing multiplex detection INTP11 trigger error flag for interleave PFC mode
0	Invalid setting
1	Clear the TKBIEFn flag to “0”.

TKBCLMFn	Trigger for clearing status flag for maximum frequency limit function
0	Invalid setting
1	Clear the TKBMFFn flag to “0”.

Remark n = 0, p = 0, 1

10.3.11 16-bit timer KB dithering count registers n0, n1 (TKBDNRn0, TKBDNRn1)

TKBDNRnp is a register that is used by the PWM dithering function for TKBOp output.

When the values in this register of the higher 4 bits are N (N = 0H to FH), the active period for N times during each 16-period cycle of PWM output is output to one count clock extended.

Table 10 - 2 shows the relation between the TKBDNRnp setting and the active period for N repetitions of one count clock extended.

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

Reset signal generation clears this register to 00H.

Figure 10 - 14 Format of 16-bit Timer KB Dithering Count Register np (TKBDNRnp)

Address: F040EH (TKBDNR00), F0410H (TKBDNR01) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TKBDNRnp					0	0	0	0

Caution Be sure to clear bits 3 to 0 to "0".

Remark n = 0, p = 0, 1

Table 10 - 2 16-bit Timer KB Dithering Count Register np (TKBDNRnp) Setting

Period Repetitions (N)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0																
1	▨															
2	▨								▨							
3	▨				▨				▨							
4	▨				▨				▨				▨			
5	▨		▨		▨				▨				▨			
6	▨		▨		▨				▨		▨		▨			
7	▨		▨		▨		▨		▨		▨		▨			
8	▨		▨		▨		▨		▨		▨		▨		▨	
9	▨	▨	▨		▨		▨		▨		▨		▨		▨	
10	▨	▨	▨		▨		▨		▨	▨	▨		▨		▨	
11	▨	▨	▨		▨	▨	▨		▨	▨	▨		▨		▨	
12	▨	▨	▨		▨	▨	▨		▨	▨	▨		▨	▨	▨	
13	▨	▨	▨	▨	▨	▨	▨		▨	▨	▨		▨	▨	▨	
14	▨	▨	▨	▨	▨	▨	▨		▨	▨	▨	▨	▨	▨	▨	
15	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	

Remark 1. □ Cell period: Inactive output via settings in the TKBCRn1 and TKBCRn3 registers

▨ Cell period: Inactive output via "settings + 1" in the TKBCRn1 and TKBCRn3 registers

Remark 2. n = 0, p = 0, 1

10.3.12 16-bit timer KB compare 1L & dithering count register n0 (TKBCRLDn0)

TKBCRLDn0 is a register that stores the “lower 8 bits of TKBCRn1 register” values in its higher 8 bits and the “TKBDNRn0 register” values in its lower 8 bits.

TKBCRLDn0 can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 10 - 15 Format of 16-bit Timer KB Compare 1L & Dithering Count Register n0 (TKBCRLDn0)

Address: F0414H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TKBCRLDn0													0	0	0	0

Lower 8 bits of TKBCRn1 register TKBDNRn0 register

Caution Be sure to clear bits 3 to 0 to “0”.

Remark n = 0

10.3.13 16-bit timer KB compare 3L & dithering count register n1 (TKBCRLDn1)

TKBCRLDn1 is a register that stores the “lower 8 bits of TKBCRn3 register” values in its higher 8 bits and the “TKBDNRn1 register” values in its lower 8 bits.

TKBCRLDn1 can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 10 - 16 Format of 16-bit Timer KB Compare 3L & Dithering Count Register n1 (TKBCRLDn1)

Address: F0416H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TKBCRLDn1													0	0	0	0

Lower 8 bits of TKBCRn3 register TKBDNRn1 register

Caution Be sure to clear bits 3 to 0 to “0”.

Remark n = 0

10.3.14 16-bit timer KB smooth start initial duty registers n0, n1 (TKBSIRn0, TKBSIRn1)

TKBSIRnp is a register that sets the default duty for the PWM output smooth start function for TKBOp output.
TKBSIRnp can be set by a 16-bit memory manipulation instruction.
Reset signal generation clears this register to 0000H.

Figure 10 - 17 Format of 16-bit Timer KB Smooth Start Initial Duty Register np (TKBSIRnp)

Address: F040AH (TKBSIR00), F040CH (TKBSIR01) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TKBSIRnp																

Remark n = 0, p = 0, 1

10.3.15 16-bit timer KB smooth start step width registers n0, n1 (TKBSSRn0, TKBSSRn1)

TKBSSRnp is a register that is used by the PWM output smooth start function for TKBOp output.
When the value of this register is N (N = 0000B to 1111B), output of a PWM with the active output period is continued for N + 1 times by setting TKBSIRnp. Afterward, output continues with the (active period + 1 clock) waveform for N + 1 cycles, then with the (active period + 2 clock) waveform for N + 1 cycles, and so on. Finally, when TKBCRn1 and TKBCRn3 have the same duty, the PWM output smooth start function is cleared and normal PWM output is set.
TKBSSRnp can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 10 - 18 Format of 16-bit Timer KB Smooth Start Step Width Register np (TKBSSRnp)

Address: F040FH (TKBSSR00), F0411H (TKBSSR01) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TKBSSRnp	0	0	0	0				

Caution Be sure to clear bits 7 to 4 to “0”.

Remark n = 0, p = 0, 1

10.3.16 16-bit timer KB maximum frequency limit setting register n (TKBMFRn)

TKBMFRn is a register that sets the minimum period for the timer restart of external trigger.

When the counter (TKBCNTn) value is smaller than this TKBMFRn value, if trigger input is detected, the trigger is held pending, and the counter (TKBCNTn) is cleared (restart) after counting to the value set to TKBMFRn.

TKBMFRn can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 10 - 19 Format of 16-bit Timer KB Maximum Frequency Limit Setting Register n (TKBMFRn)

Address: F0424H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TKBMFRn																

Remark n = 0

10.3.17 Peripheral function switch register 0 (PFSEL0)

PFSEL0 selects I/O settings about 16-bit timers KB0 and peripheral function.
PFSEL0 can be written by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Remark When INTP10 and INTP11 are used with forced output stop function 2, select none for the noise filter to shorten the response interval from input of a trigger to termination of output.

Figure 10 - 20 Format of Peripheral Function Switch Register 0 (PFSEL0)

Address: F0440H After reset: 00H R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
PFSEL0	CTRGSEL1	CTRGSEL0	INTPINV1	INTPINV0	PNFEN1	PNFEN0	TMRSTEN1	TMRSTEN0

CTRGSEL1	CTRGSEL0	Timer KB counter start trigger source[Note 1]
0	0	Use comparator 0 detection
0	1	Use comparator 1 detection
1	0	Use both of comparator 0 and comparator 1 detections
1	1	Setting prohibited (output signal is fixed to low level)

INTPINV1	Invert setting of INTP11 signal
0	Do not invert INTP11 signal
1	Invert INTP11 signal

INTPINV0	Invert setting of INTP10 signal
0	Do not invert INTP10 signal
1	Invert INTP10 signal

PNFEN1	Noise filter setting of external interrupt INTP11
0	Noise filter enable
1	Noise filter disable

PNFEN0	Noise filter setting of external interrupt INTP10
0	Noise filter enable
1	Noise filter disable

TMRSTEN1	Switch of external interrupt INTP11[Note 2]
0	External interrupt function is selected (stop mode release enabled, timer restart disabled)
1	Timer restart function is selected (stop mode release disabled, timer restart enabled).

TMRSTEN0	Switch of external interrupt INTP10[Note 2]
0	External interrupt function is selected (stop mode release enabled, timer restart disabled)
1	Timer restart function/forced output stop function 2 is selected (stop mode release disabled, timer restart enabled).

Note 1. When comparator i detection signal is used as a counter start trigger source, set the CiOE bit of the comparator output control register (COMPOCR) to 1. For detail, see **19.5 Caution for Using Timer KB Simultaneous Operation Function**.

Note 2. When INTP10 or INTP11 is used as a trigger of the timer KB forced output stop function 2 or timer restart function, see **19.5 Caution for Using Timer KB Simultaneous Operation Function**.

Remark See **Figure 19 - 1 Comparator Block Diagram**.
i = 0, 1

10.3.18 External interrupt edge enable register (INTPEG)

INTPEG sets the valid edges of external interrupts (INTP10 and INTP11) which trigger the timer restart function.

INTPEG can be written by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10 - 21 Format of External Interrupt Edge Enable Register (INTPEG)

Address: F0441H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
INTPEG	0	0	0	0	INTPEGP11	INTPEGN11	INTPEGP10	INTPEGN10

INTPEGP11	INTPEGN11	INTP11 pin valid edge selection
0	0	Edge detection disabled (disables output of timer restart signal (output signal = fixed to low level))
0	1	Falling edge (enables output of timer restart signal)
1	0	Rising edge (enables output of timer restart signal)
1	1	Both rising and falling edges (enables output of timer restart signal)

INTPEGP10	INTPEGN10	INTP10 pin valid edge selection
0	0	Edge detection disabled (disables output of timer restart signal (output signal = fixed to low level))
0	1	Falling edge (enables output of timer restart signal)
1	0	Rising edge (enables output of timer restart signal)
1	1	Both rising and falling edges (enables output of timer restart signal)

10.3.19 Registers that control port functions of 16-bit timer KB input/output pins

Using 16-bit timer KB requires setting of the registers that control the port functions for the port pins with which the 16-bit timer KB pin functions for the target channel are multiplexed (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**.

Using a port pin which is multiplexed with a 16-bit timer KB output pin function (e.g. P30/TKBO1) for 16-bit timer KB output requires setting the corresponding bits in the port mode control register (PMCxx), port mode register (PMxx), and port register (Pxx) to 0.

Example: When P30/TKBO1 is to be used 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.

10.4 Operation of 16-bit Timers KB0

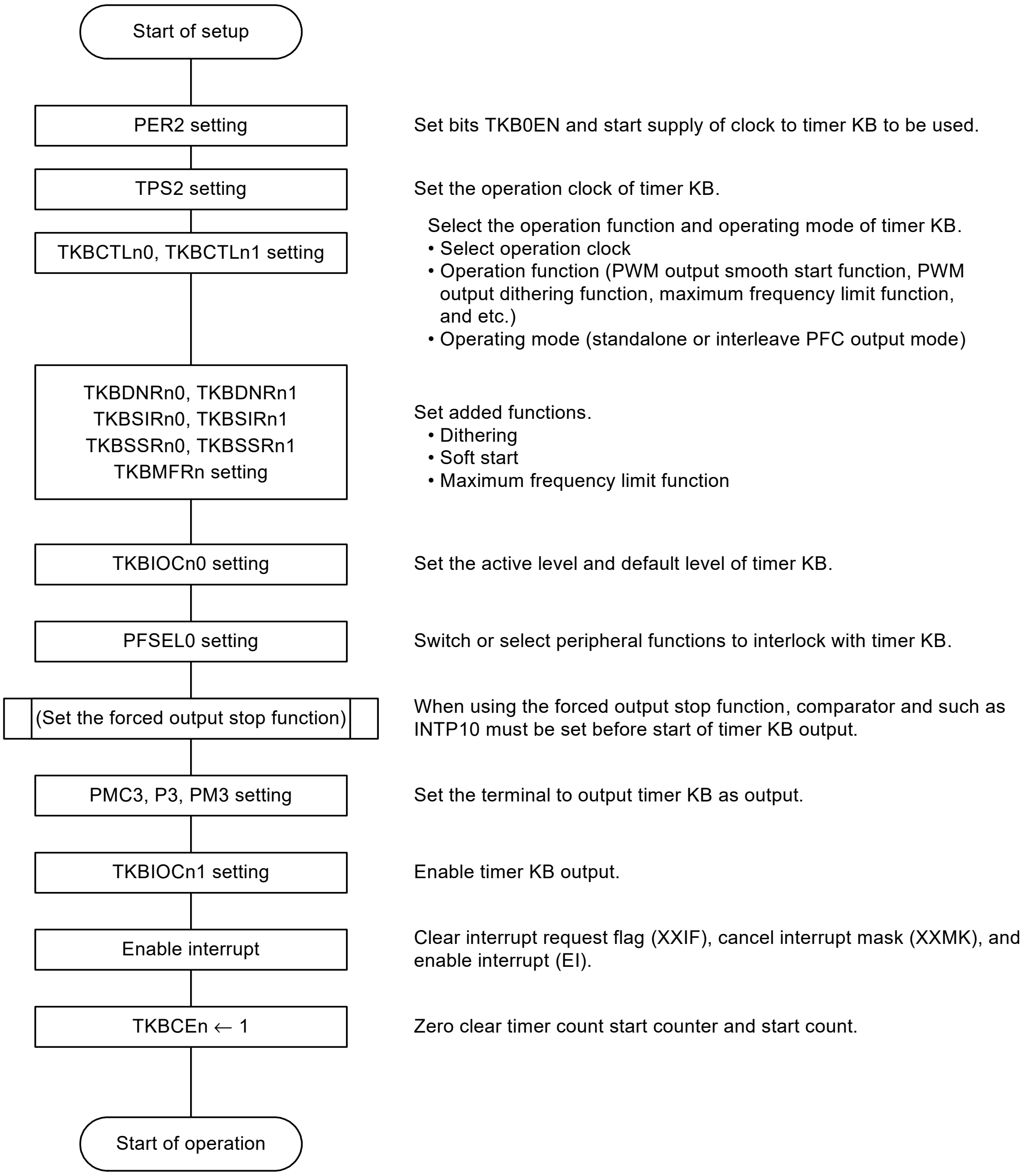
Operation specifications of 16-bit timers KB0 described below.

- Counter basic operation (See **10.4.1**.)
- Default level and active level (See **10.4.2**.)
- Stop/restart operation (See **10.4.3**.)
- Batch overwrite (See **10.4.4**.)

There are 3 different operation modes for 16-bit timers KB0

- Standalone mode (period controlled by TKBCRn0) (See **10.4.5**.)
- Standalone mode (period controlled by external trigger input) (See **10.4.6**.)
- Interleave PFC output mode (See **10.4.7**.)

Figure 10 - 22 Timer KB Operation Setting Example (Operation Start Flow)



Set bits TKB0EN and start supply of clock to timer KB to be used.

Set the operation clock of timer KB.

Select the operation function and operating mode of timer KB.
- Select operation clock
- Operation function (PWM output smooth start function, PWM output dithering function, maximum frequency limit function, and etc.)
- Operating mode (standalone or interleave PFC output mode)

Set added functions.
- Dithering
- Soft start
- Maximum frequency limit function

Set the active level and default level of timer KB.

Switch or select peripheral functions to interlock with timer KB.

When using the forced output stop function, comparator and such as INTP10 must be set before start of timer KB output.

Set the terminal to output timer KB as output.

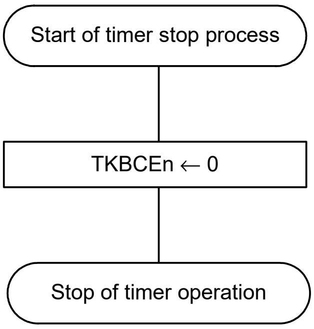
Enable timer KB output.

Clear interrupt request flag (XXIF), cancel interrupt mask (XXMK), and enable interrupt (EI).

Zero clear timer count start counter and start count.

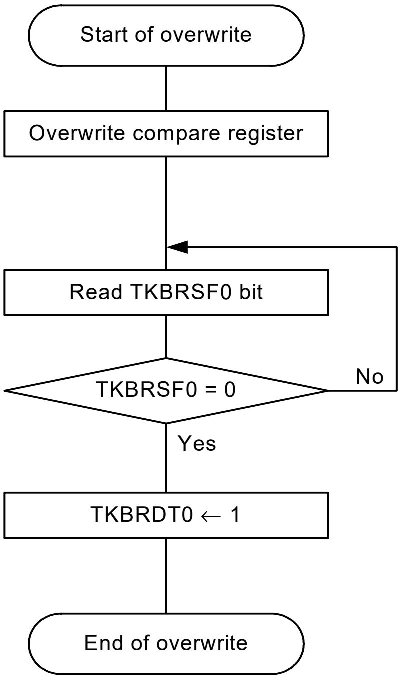
Remark n = 0

Figure 10 - 23 Timer KB Operation Setting Example (Operation Stop Flow)



Timer count stops and timer KB output becomes fixed output at the set default level.

Figure 10 - 24 Timer KB Operation Setting Example (Compare Register Batch Overwrite Flow)



TKBCNTn, TKBCRn0 to TKBCRn3, TKBTGCRn (n = 0)
Overwrite TKBDNRn0, TKBDNRn1, TKBCRLDn0, and TKBCRLDn1 registers.

Read TKBRSF0 of TKBFLG0 register.
Accept batch overwrite request when TKBRSF0 = 0.

Compare register batch overwrite request

After accepting overwrite request, TKBRSF0 bit remains “1” until setting is actually changed.

Remark The batch overwrite function is used to change the timer counter operation setting while timer KB is operating. The set value is reflected to the operation from the next restart.

10.4.1 Counter basic operation

(1) Count start operation

In any mode, the 16-bit counter of timer KB starts its counting from initial value of FFFFH. It increments the counter from FFFFH to 0000H, 0001H, 0002H, 0003H and so on.

(2) Clear operation

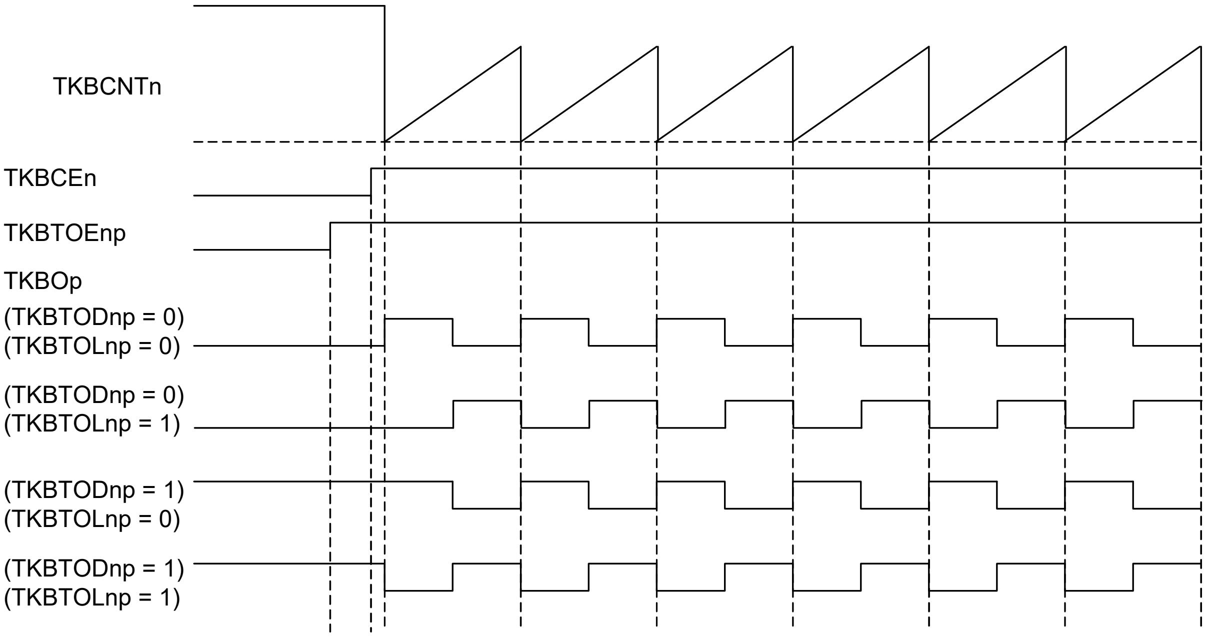
The 16-bit counter is reset to 0000H when the 16-bit counter value matches with the value defined in TKBCRn0 or an external trigger is in effect if the period is determined by external triggers. INTTMKBn interruption occurs when the counter is cleared at the time when it matches with the value defined in TKBCRn0, but it does not occur when the counter is cleared by an external trigger.

10.4.2 Default level and active level

(1) Basic operation

Default level and active level settings are available for timer KB output by 16-bit timer KB output control register n0 (TKBIOCn0).

Figure 10 - 25 Figure of Timing of Default and Active Level (Basic Operation)



When TKBTOEnp is switched from “0” to “1”, PWM waveform is output according to the generation of TKBOp set condition/reset condition and TKBTOLnp setting.
When TKBTOEnp is switched from “1” to “0”, default level is output for TKBOp according to TKBTODnp setting.

(2) TKBTOEnp switched from “0” to “1”

When TKBTOEnp is changed from 0 to 1 before the value of counter TKBCNTn matches with the value of compare register TKBCRnp, while the timer counter is in operation, the timer output generated becomes the PWM waveform in accordance with the TKBTOLnp setting at the timing when it matches.

If TKBTOEnp is changed from 0 to 1 after the value of counter TKBCNTn matches with the value of compare register TKBCRnp, the timer output remains its initial setting level until the next timing of match occurs.

Figure 10 - 26 Figure of Timing of Default and Active Level
(TKBTOEnp = 0 Switched to 1 Before Matching Counter and Compare Register (TKBCRn1 to TKBCRn3))

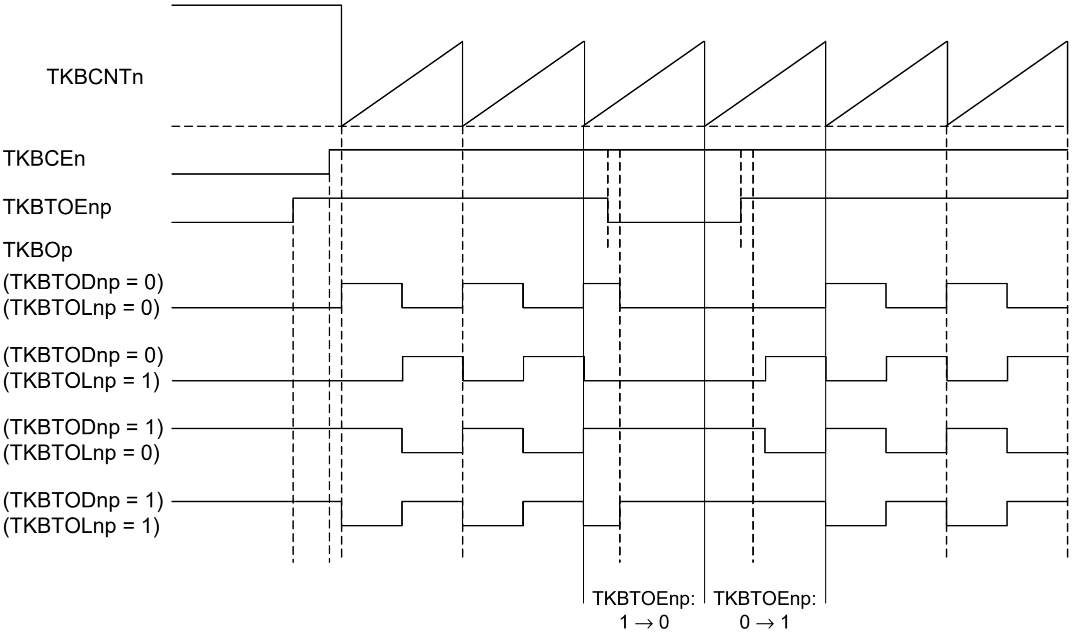
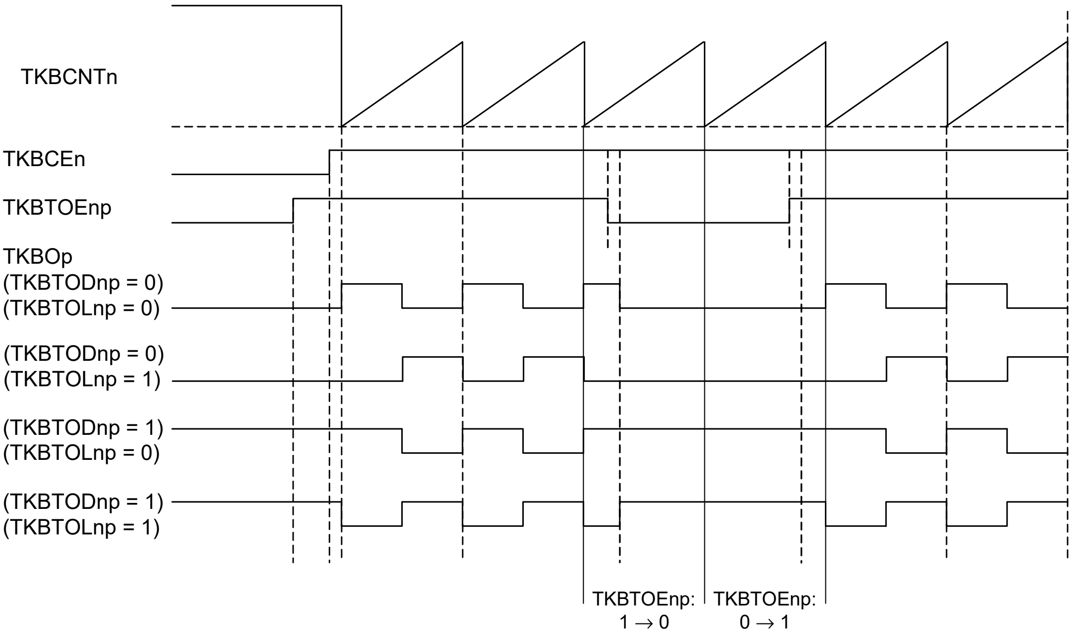


Figure 10 - 27 Figure of Timing of Default and Active Level
(TKBTOEnp = 0 Switched to 1 After Matching Counter and Compare Register (TKBCRn1 to TKBCRn3))

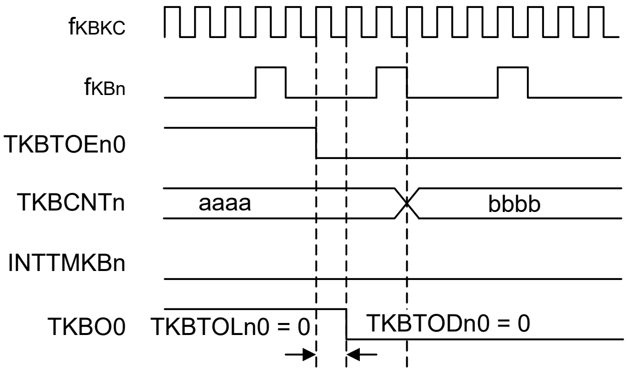


(3) TKBTOEnp switched from “1” to “0”

(a) Basic timing

TKBOp is default level set by TKBTODnp after 1 fKBKC when TKBTOEnp is switched from “1” to “0”.

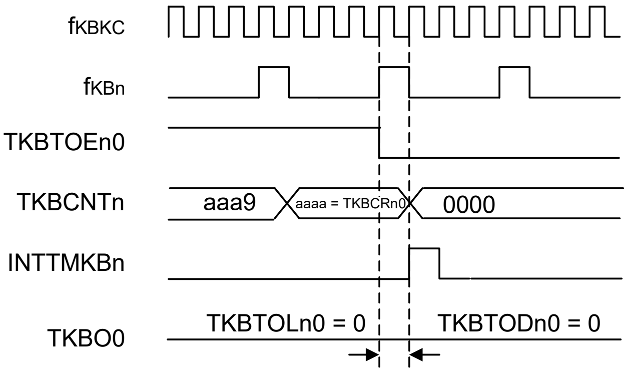
Figure 10 - 28 Figure of Timing of Default and Active Level (TKBTOEn0 Switched from “1” to “0”)



(b) When the setting due to the matched value of TKBCRn0 and the event that TKBTOEnp is cleared occur at the same instant:

When TKBTOEnp set timing (high to low) is simultaneous with the matching between TKBCNTn and TKBCRnm, the change of TKBTOEnp is given priority to become default level set for TKBTODnp.

Figure 10 - 29 Figure of Timing for Default and Active Level (TKBO0 Set Timing (Low to High) Is Simultaneous with the Matching Between TKBCNTn and TKBCRnm)

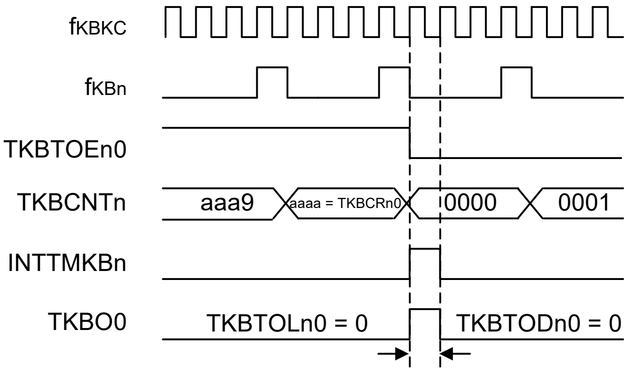


(c) When the operation of TKBTOEnp is simultaneous with generation of timer count clock

TKBOp is set by the matching of TKBCNTn = TKBCRnm in case when the operation of TKBTOEnp is simultaneous with generation of timer count clock.

After 1 fKBKC, TKBOp is default level which is set with TKBTODnp.

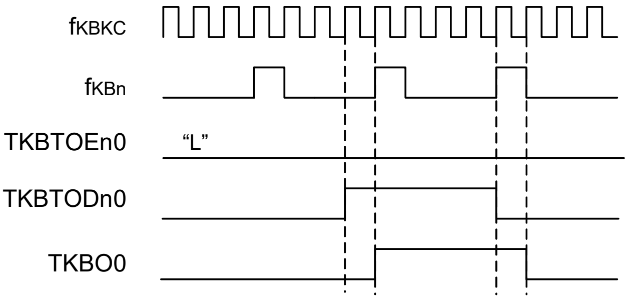
Figure 10 - 30 Figure of Timing of Default and Active Level (Operation of TKBTOEn0 Is Simultaneous with generation of Timer Count Clock.)



(4) Change TKBTODnp at TKBTOEnp = 0

When TKBTODnp being changed at TKBTOEnp = 0, after 1 fKBKC, TKBOp is default level which is set with TKBTODnp.

Figure 10 - 31 Figure of Timing of Default and Active Level (Change TKBTODn0 at TKBTOEn0 = 0)



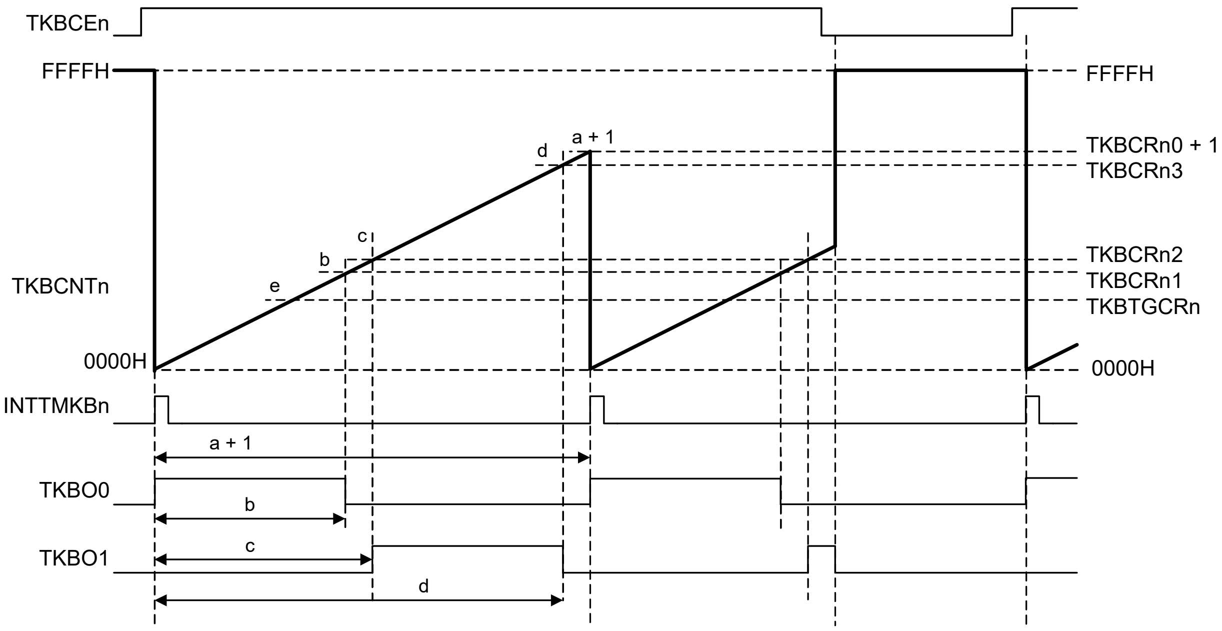
10.4.3 Stop/restart operation

Stop and start of operation of 16-bit timer KB is available by controlling TKBCEn.
16-bit timer KB is reset and stop operation by changing TKBCEn from “1” to “0”.

Counter TKBCNTn is reset to FFFFH and stop operation then.
TKBOp output outputs default level set by TKBTODnp.

16-bit timer KB starts operation by changing TKBCEn from “0” to “1”.
Counter TKBCNTn maintains FFFFH when TKBCEn = 0 and start up counting operation by changing TKBCEn from “0” to “1”.

Figure 10 - 32 Figure of Timing of Stop Operation (TKBTOLnp = 0, TKBTODnp = 0)

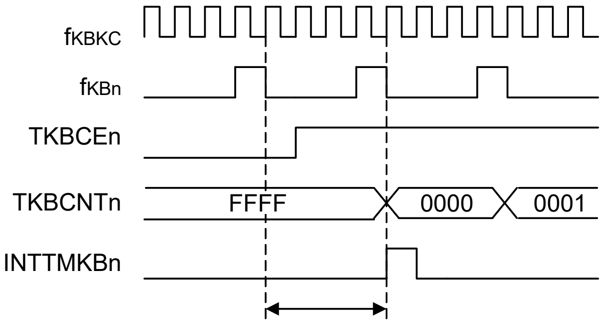


(1) Count operation start timing

When TKBCEn is switched from “0” to “1” counting operation starts after the progress of the minimum 1 fKBKC to the maximum 1 fKBn.

INTTMKBn is output at counting operation start timing.

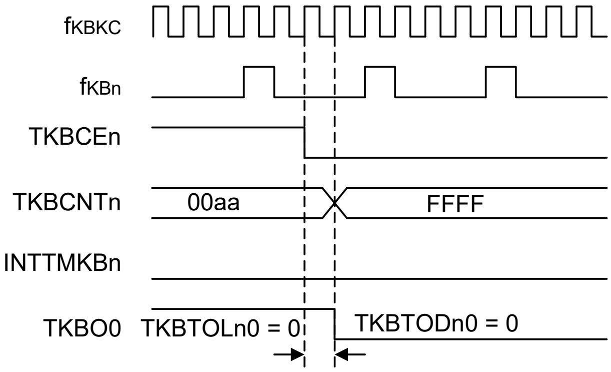
Figure 10 - 33 Figure of Timing of Start Operation (TKBCEn Switched from “0” to “1”)



(2) Count operation stop timing

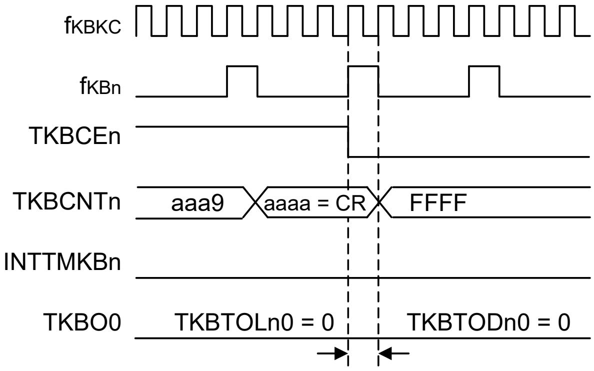
When TKBCEn is switched from “1” to “0” counting operation is stopped after the progress of minimum 1 fKBKC. TKBCNTn is reset to FFFFH and TKBOp is default level set by TKBTODnp.

Figure 10 - 34 Figure of Timing of Stop Operation (TKBCEn Switched from “1” to “0”)



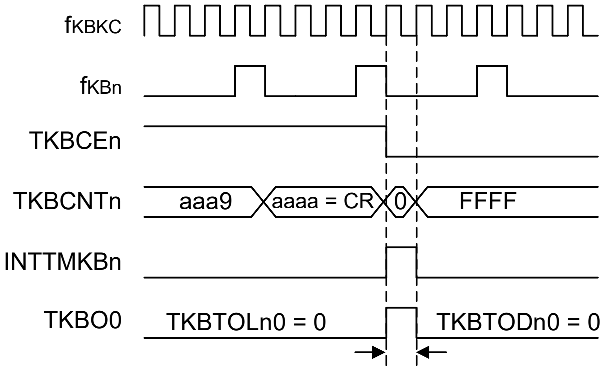
When TKBCEn is switched from “1” to “0” counting operation is stopped after the progress of minimum 1 fKBKC. Before the generation of 1 fKBn, INTTMKBn is not output even matching of TKBCNTn = TKBCRn0 being generated.

Figure 10 - 35 Figure of Timing of Stop Operation
(Operation of TKBCEn Is Before the Generation of Timer Count Clock.)



TKBOp is set and INTTMKBn being output via the matching of TKBCNTn = TKBCRn0 generated in case when the operation of TKBCEn is simultaneous with the generation of 1 fKBKC.
TKBCNTn is reset to FFFFH after the progress of 1 fKBKC and TKBOp is default level set by TKBTODnp.

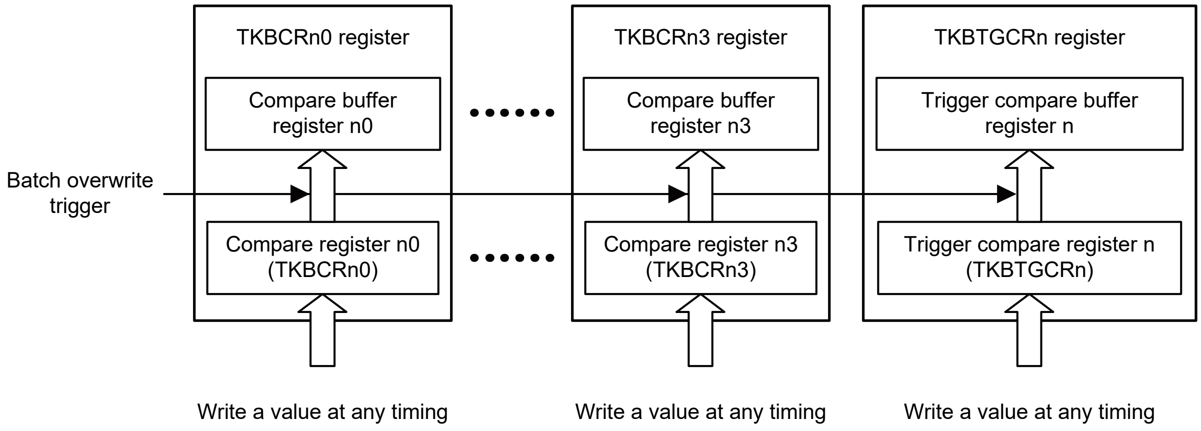
Figure 10 - 36 Figure of Timing of Stop Operation
(Operation of TKBCEn Is Simultaneous with the Generation of Timer Count Clock.)



10.4.4 Batch overwrite operation

TMKB compare register np (TKBCRnp) for timer KB has, as shown in Figure 10 - 37, two stages.
Therefore, its value does not become effective immediately even if any value is set to TKBCRnp by a program. The value set to TKBCRnp at any timing is transferred at once to buffer registers at the time when the counter starts running or when transfer trigger occurs, and it is actually used for any comparison operation. This enables multiple compare registers to be set with each value at different timing.

Figure 10 - 37 Compare Register Batch Overwrite Function


TKBCRn0 register
Compare buffer register n0
Compare register n0 (TKBCRn0)
TKBCRn3 register
Compare buffer register n3
Compare register n3 (TKBCRn3)
TKBTGCRn register
Trigger compare buffer register n
Trigger compare register n (TKBTGCRn)
Batch overwrite trigger
Write a value at any timing
Write a value at any timing
Write a value at any timing

Remark As shown above, TMKB compare registers np (TKBCRnp) have two stage structure and they are treated as a single register except when values are written to them.

(1) Timing ob batch overwrite

There are three cases when the compare registers are written all together. Among these, (c) can be controlled by configuration of the register.

(a) When starting count operation of timer KB
(b) Count value of the 16-bit counter and the value that is set to TMKB compare register n0 (TKBCRn0) matches.
(c) An external trigger occurs, while batch overwrite with an external trigger is permitted.

10.4.5 Standalone mode (period controlled by TKBCRn0)

(1) Outline of functions

In standalone operation mode, the period is defined by setting value of TKBCRn0, then TKBO0 is generated by TKBCRn0 and TKBCRn1, and then TKBO1 is generated by TKBCRn2 and TKBCRn3.

Duty can be set within range of 0% to 100% and the period and Duty can be calculated using the following formula.

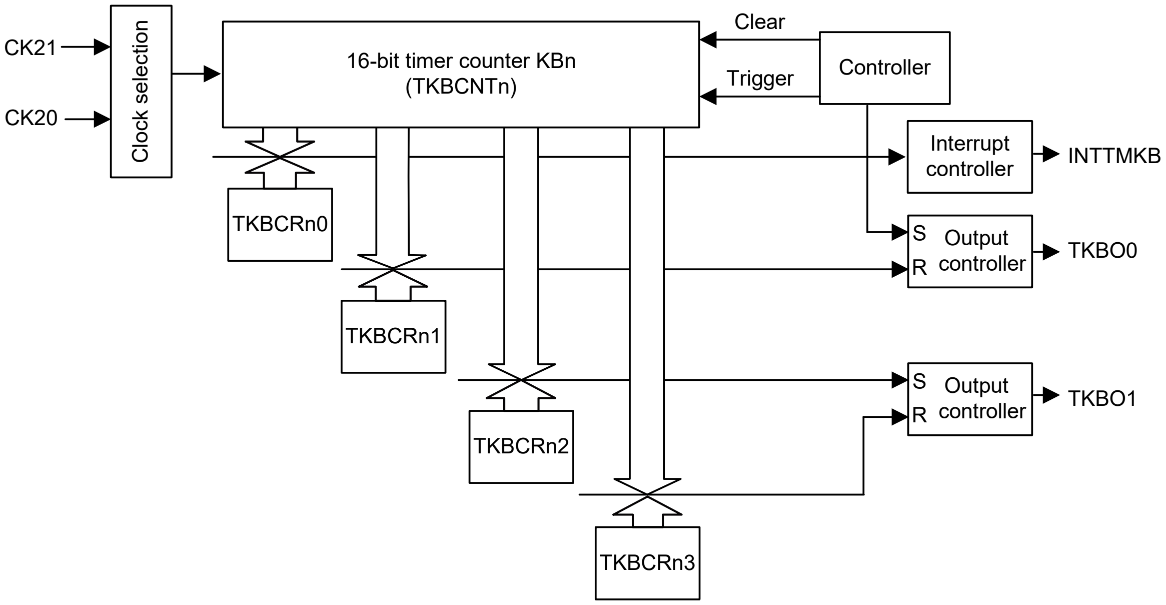
[Calculation formula for TKBO0 output]
Pulse period = (TKBCRn0 setting + 1) × Counter clock period
Duty [%] = (TKBCRn1 setting/(TKBCRn0 setting + 1)) × 100
0% Output: TKBCRn1 setting = 0000H
100% Output: TKBCRn1 setting ≥ TKBCRn0 setting + 1

[Calculation formula for TKBO1 output]
Duty [%] = ((TKBCRn3 setting – TKBCRn2 setting)/(TKBCRn0 setting + 1)) × 100
0% Output: TKBCRn3 setting = TKBCRn2 setting
100% Output: TKBCRn2 setting = 0000H, TKBCRn3 setting ≥ TKBCRn0 setting + 1

Caution It should always be: TKBCRn2 setting ≤ TKBCRn3 setting.

Figure 10 - 38 shows the configuration of standalone mode.

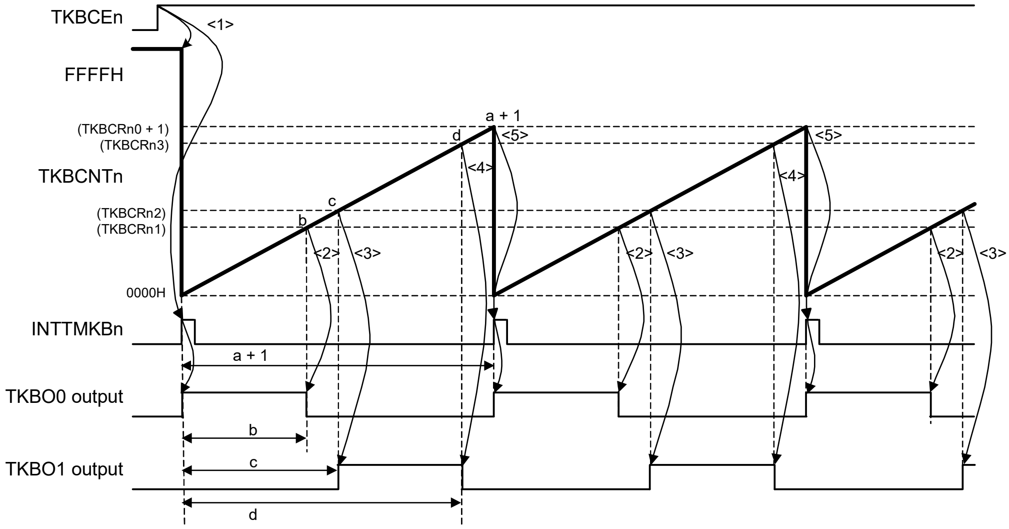
Figure 10 - 38 Configuration of Standalone Mode (Period Controlled by TKBCRn0)



(2) Outline of operation

Figure 10 - 39 shows the timing sample for standalone mode.

Figure 10 - 39 Timing Sample for Standalone Mode (Period Controlled by TKBCRn0)
(at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



This section describes an example about the standalone operation (periodic control by TKBCR0). The following descriptions are linked with <1> to <5> in Figure 10 - 39.

<1> When TKBCEn is set with a value of 1, the 16-bit timer counter KBn (TKBCNTn) changes from FFFFH to 0000H in synchronizing with the count clock, then it starts counting up. At the same time, INTTMKBn output is generated and TKBO0 output changes from its initial value specified with TKB0TOD0 bit of TKB0IOC0 register to its active value (high level in this example) specified with TKB0TOL0 bit (TKBO1 output hold its initial value specified with TKB0TOD1 bit).

<2> When TKBCNTn is counted up and its value matches with the value specified in TMKB compare register n1 (TKBCRn1), TKBO0 output becomes inactive level.

<3> When TKBCNTn is counted up and its value matches with the value specified in TMKB compare register n2 (TKBCRn2), TKBO1 output becomes active level.

<4> When TKBCNTn is counted up and its value matches with the value specified in TMKB compare register n3 (TKBCRn3), TKBO1 output becomes inactive level.

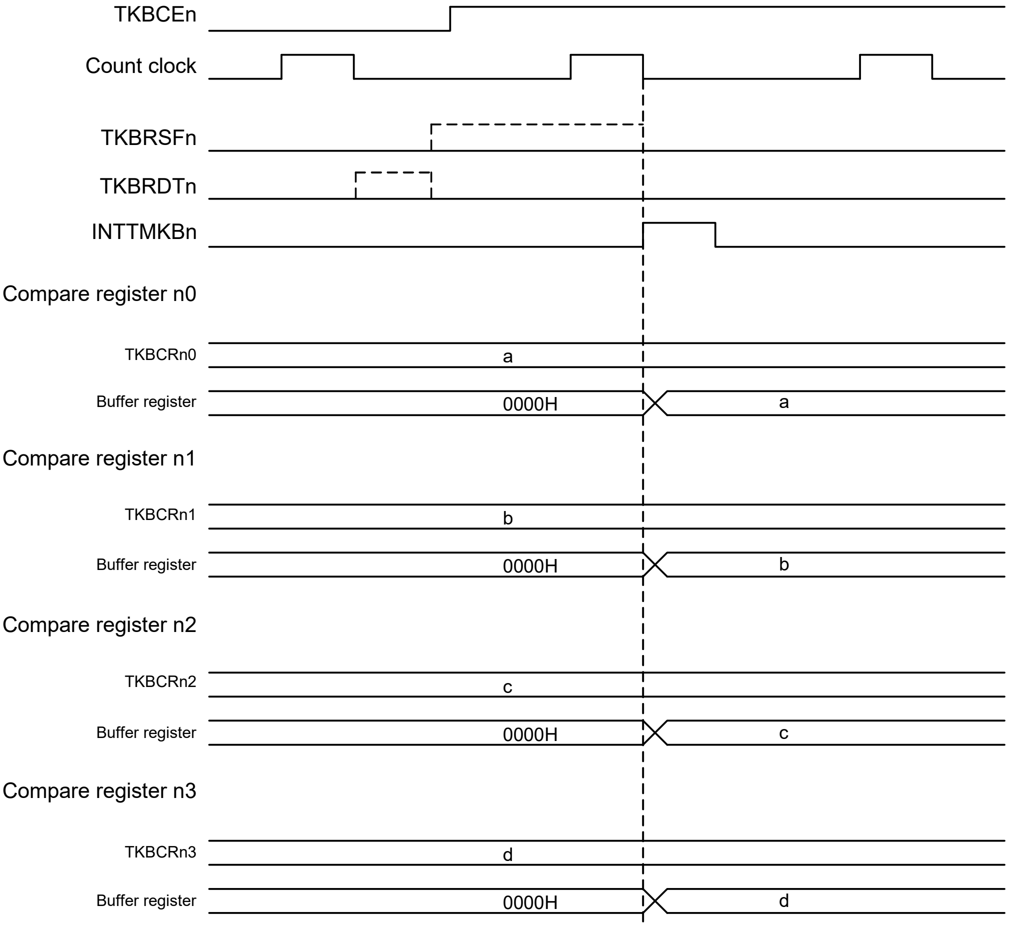
<5> When TKBCNTn is counted up and its value matches with the value specified in TMKB compare register n0 (TKBCRn0), INTTMKBn output is generated at the next count clock and TKBO0 output becomes active level. TKBCNTn starts its upward counting from 0000H.

<6> Repeats <2> through <5>.

(3) Operation of batch overwrite (at starting the counting operation)

Compare register of the timers KB0 have function which updates internal buffer register simultaneously at the starting of counter operation caused by count clock which is generated after overwriting “1” to TKBCEn bit. Batch overwrite is generated without writing “1” on TKBRDTn bit only in case of counting operation start timing (see **Figure 10 - 40**).

Figure 10 - 40 Batch Overwrite Function: Figure of Buffer Updating Timing at Counting Operation Start



Remark When TKBCEn = 0, TKBRSFn is set to “1” at writing “1” to TKBRDTn.
TKBRSFn is cleared to “0” at counting operation start timing (counter start trigger generated).

(4) Batch overwrite function: Update buffer during counting operation

Compare register of the timers KB0 has a function which updates internal buffer register simultaneously at the next counter clear (TKBCNTn and TKBCRn0 matched), identifying the writing “1” to TKBRDTn bit as batch overwriting trigger. TKBRSFn is provided as flag to indicate from writing of “1” to TKBRDTn bit until the completion of batch overwrite (see **Figure 10 - 41**).

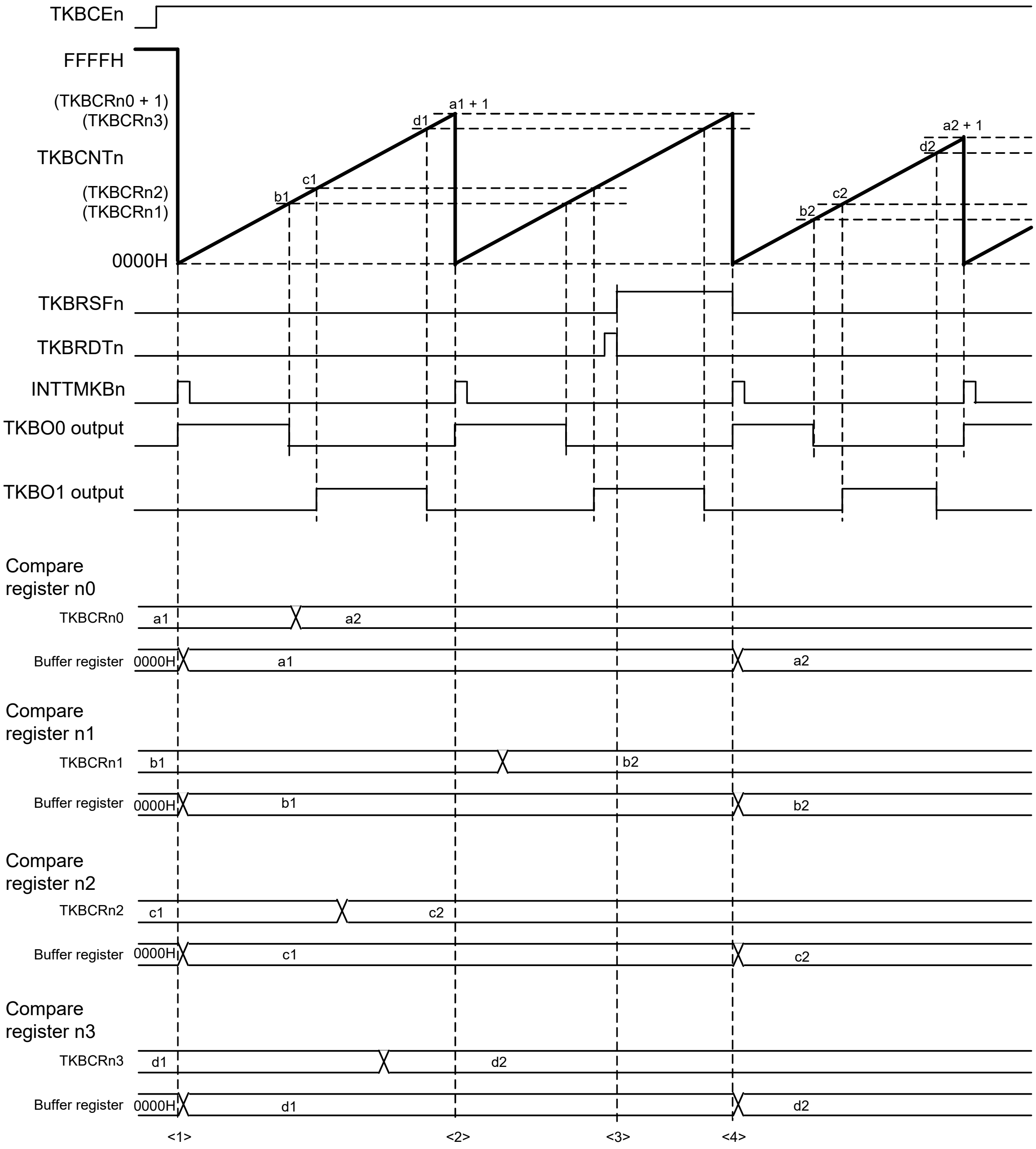
<1> Compare register setting is transferred to buffer register at the timing when TKBCEn is set from “0” to “1” and TKBCNTn starts counting operation.

<2> Overwriting is not generated if writing of “1” to TKBRDTn is not implemented even counter clear is generated after TKBCRn0 to TKBCRn3 and TKBTGCRn registers are overwritten.

<3> Batch overwrite pending flag (TKBRSFn) is “1” by writing “1” to TKBRDTn.

<4> Compare register setting is transferred to buffer register by counter clear generated at TKBRSFn = 1. TKBRSFn is “0” simultaneously.

Figure 10 - 41 Batch Overwrite Function: Figure of the Timing of Buffer Updating During Counting Operation



(5) Sample of register setting details at standalone mode (period controlled by TKBCRn0)

bit No.	15	14	13	12	11	10	9	8
TKBCTLn0	-	-	TKBSSEn1	TKBDIEn1	-	-	TKBSSEn0	TKBDIEn0
Setting	0	0	1/0	1/0	0	0	1/0	1/0

bit No.	7	6	5	4	3	2	1	0
TKBCTLn0	TKBMFEn	-	TKBIRSn1	TKBIRSn0	-	TKBTSEn	TKBSTSn1	TKBSTSn0
Setting	0	0	0	0	0	0	0	0

bit No.	7	6	5	4	3	2	1	0
TKBCTLn1	TKBCEn	-	-	TKBCKSn	-	-	TKBMDn1	TKBMDn0
Setting	1	0	0	1/0	0	0	0	0

bit No.	7	6	5	4	3	2	1	0
TKBIOCn0	-	-	-	-	TKBTOLn	TKBTOLn0	TKBTODn1	TKBTODn0
Setting	0	0	0	0	1/0	1/0	1/0	1/0

bit No.	7	6	5	4	3	2	1	0
TKBIOCn1	-	-	-	-	-	-	TKBTOEn1	TKBTOEn0
Setting	0	0	0	0	0	0	1/0	1/0

Register	Setting range
TKBCRn0	0000H to FFFFH
TKBCRn1	0000H to FFFFH
TKBCRn2	0000H to FFFFH
TKBCRn3	0000H to FFFFH
TKBTGCRn	0000H to FFFFH
TKBSIRn0	0000H to FFFFH
TKBSIRn1	0000H to FFFFH
TKBSSRn0	00H to 0FH
TKBSSRn1	00H to 0FH
TKBDNRn0	00H to F0H
TKBDNRn1	00H to F0H
TKBMFRn	0000H

□ : Setting is fixed for this mode: ▨ : Setting is not needed (default setting)

10.4.6 Standalone mode (period controlled by external trigger input)

(1) Outline of functions

By standalone mode, period can be controlled not only by TKBCRn0 but also by external trigger input.

Input signals selected by TKBSTSn1 and TKBSTSn0 bits of 16-bit timer KB operation control register are used to detect external trigger input (Timer restart function). By using this function, critical conduction mode PFC control can be implemented, for example.

When the external trigger input is detected, counter TKBCNTn is cleared with 0000H and TKBO0/TKBO1 output is respectively set to active level and inactive level. When setting value of TKBCRn0 and the counter (TKBCNTn) match is generated before external trigger input detection, counter is cleared to 0000H and the operation is continued.

For the formula to calculate TKBO0/TKBO1 output in case external trigger input not yet detected and the period is controlled by TKBCRn0, see **10.4.5 Standalone mode (period controlled by TKBCRn0)**.

Calculation formula for TKBO0/TKBO1 output in case of period to be controlled by external trigger input detection is as follows:

[Calculation formula for TKBO0 output]

Pulse period = (Counter value at external trigger input detection + 1) × Count clock period

Duty [%] = (Setting value of TKBCRn1/(Counter value of external trigger input detection + 1)) × 100

0% output: TKBCRn1 setting = 0000H

100% output: TKBCRn1 setting ≥ Counter value at external trigger input detection + 1

[Calculation formula for TKBO1 output]

Pulse period = (Counter value at external trigger input detection + 1) × Count clock period

Duty [%] = ((Setting value of TKBCRn3 – Setting value of TKBCRn2)/(Counter value of external trigger input detection +1)) × 100

0% output: TKBCRn3 setting = TKBCRn2 setting

100% output: TKBCRn2 setting = 0000H, TKBCRn3 setting ≥ Counter value at external trigger input detection + 1

Caution It should always be: TKBCRn2 setting ≤ KBCRn3 setting.

Figure 10 - 42 shows the configuration of standalone mode (period controlled by external trigger input).

Figure 10 - 42 Configuration of Standalone Mode (Period Controlled by External Trigger Input)

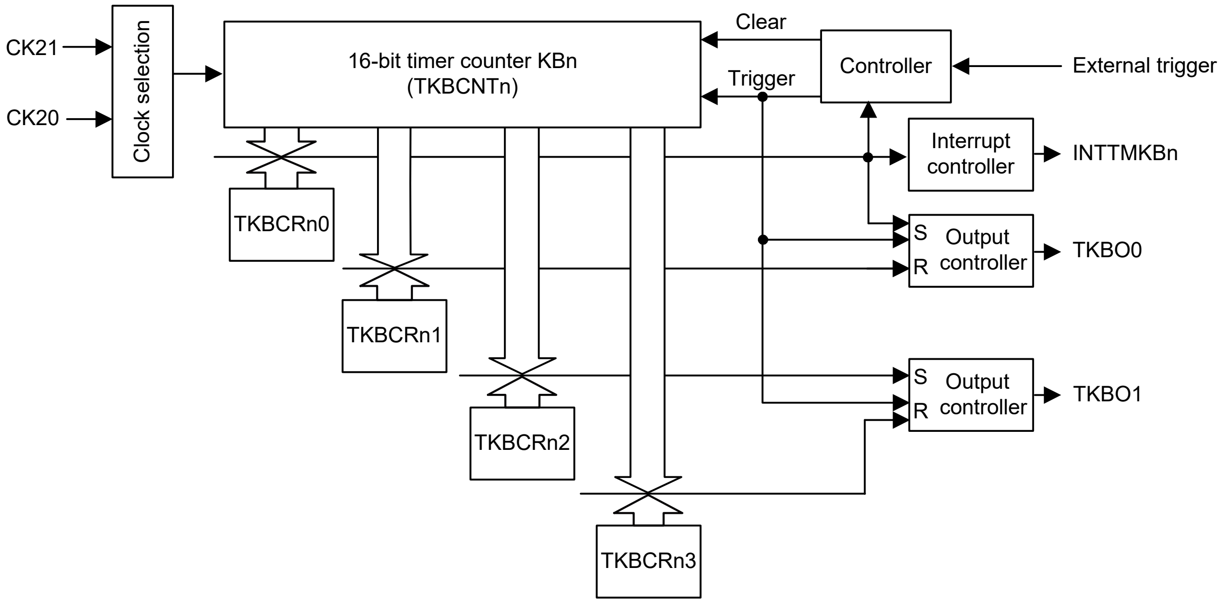


Table 10 - 3 External Trigger Assignment List of Standalone Operation

(Period Controlled by External Trigger Input)

	Timer KB0
Comparator 0	√
Comparator 1	√
INTP10	√
INTP11	√

(2) Batch overwrite function (period controlled by external trigger input, buffer updating during counting operation (TKBTSEn bit set to 1))

In standalone mode of period controlled by external trigger input, counter clear and compare register batch overwrite is implemented at the timing when external trigger input is detected after writing “1” to TKBRDTn bit and through setting TKBTSEn bit of TKBCTLn0 register in “1”.

Same as in counter clear, batch overwriting is implemented as well in case when TKBCRn0 and counter (TKBCNTn) being matched before the detection of external trigger input after writing “1” to TKBRDTn bit.

Factor of external trigger input is selected at TKBSTSn1 and TKBSTSn0 of TKBCTLn0 register. Figure 10 - 43 shows an example of the timing of the batch overwrite operation with TKBTSEn bit set to “1”.

<1> Compare register setting is transferred to buffer register at the timing when TKBCEn is set from “0” to “1” and TKBCNTn starts counting operation.

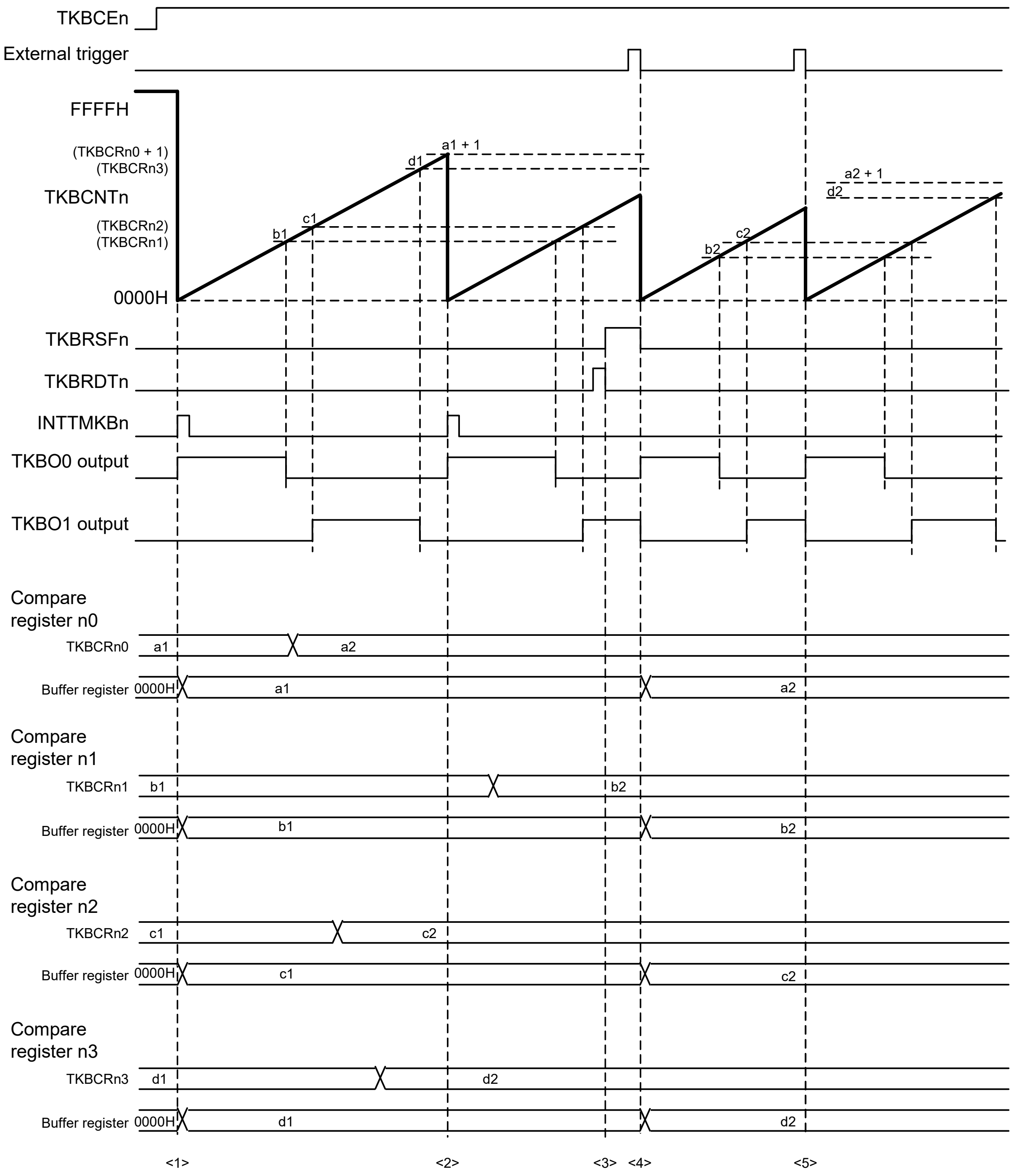
<2> Overwriting is not generated if writing of “1” to TKBRDTn is not implemented even counter clear is generated after TKBCRn0 to TKBCRn3 and TKBTGCRn registers are overwritten.

<3> Batch overwrite pending flag (TKBRSFn) is “1” by writing “1” to TKBRDTn.

<4> When a counter clear is generated by an external trigger input while TKBTSEn bit is set to “1” and TKBRSFn bit is “1”, the setting value in the compare register is transferred to the buffer register. At the same time, TKBRSFn bit becomes “0”.

<5> Even if the counter clear event is generated by the external trigger input, the batch overwrite does not occur unless a value “1” is written in TKBRDTn bit.

Figure 10 - 43 Batch Overwrite Function: Figure of Standalone for External Trigger Input Factor and the Timing of Buffer Updating During Counting Operation (TKBTSEn Bit Set to 1)



(3) Batch overwrite function (period controlled by external trigger input, buffer updating during counting operation (TKBTSEn bit clear to 0))

This is an example of the case where TKBTSEn bit in TKBCTLn0 register is set to "0" during standalone operation under the periodic control by external trigger input. In this case, the counter is cleared when a value "1" is written in TKBRDTn bit and the external trigger input is detected while batch overwrite suspension flag (TKBRSFn bit) is "1".

An external trigger factor is selected with TKBSTSn1 and TKBSTSn0 bit in TKBCTLn0 register. Figure 10 - 44 shows an example of the batch overwrite operation timing when TKBTSEn bit is set to "0".

<1> Compare register setting is transferred to buffer register at the timing when TKBCEn is set from "0" to "1" and TKBCNTn starts counting operation.

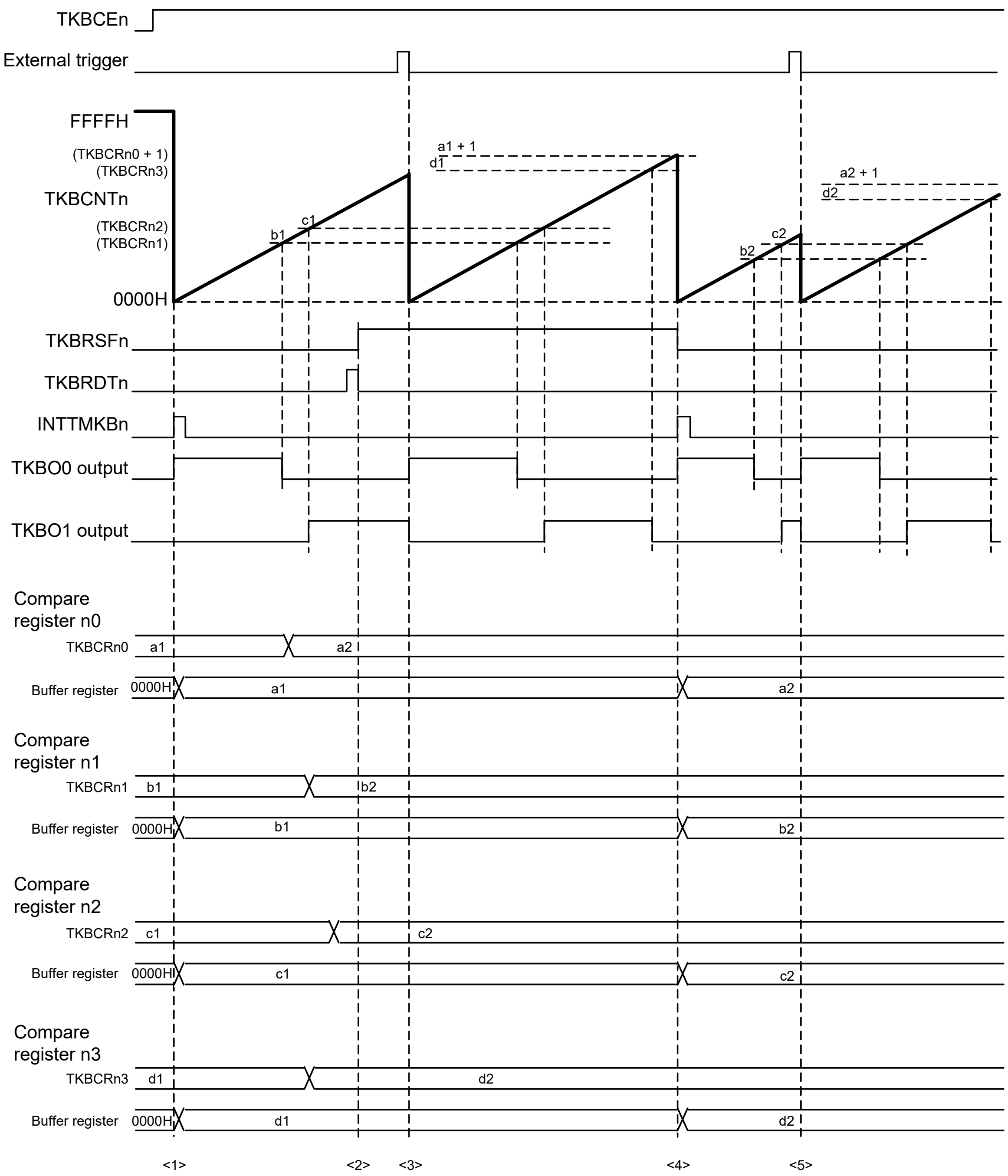
<2> After rewritten TKBCRn0 to TKBCRn3 and TKBTGCRn register, batch overwrite pending flag (TKBRSFn) is "1" by writing "1" to TKBRDTn.

<3> Even if the counter clear event is generated by the external trigger input, the batch overwrite does not occur unless TKBTSEn bit is "1".

<4> When the counter clear event (TKBCNTn matches with TKBCRn0) occurs under the status of TKBRSFn bit is "1", the value set to the compare register is transferred to the buffer register. At the same time, TKBRSFn bit becomes "0".

<5> Even if the counter clear event is generated by the external trigger input, the batch overwrite does not occur unless TKBTSEn and TKBRSFn bits are both "1".

Figure 10 - 44 Batch Overwrite Function: Figure of standalone for External Trigger Input Factor and the Timing of Buffer Updating during Counting Operation (TKBTSEn bit clear to 0)



(4) Sample of register setting details at standalone mode (period controlled by external trigger input)

	15	14	13	12	11	10	9	8
TKBCTLn0	-	-	TKBSSEn1	TKBDIEn1	-	-	TKBSSEn0	TKBDIEn0
	0	0	0	0	0	0	0	0

	7	6	5	4	3	2	1	0
	TKBMFEn	-	TKBIRSn10	TKBIRSn00	-	TKBTSEn	TKBSTSn1	TKBSTSn0
	1/0	0	0	0	0	1/0	1/0	1/0

	7	6	5	4	3	2	1	0
TKBCTLn1	TKBCEn	-	-	TKBCKSn	-	-	TKBMDn1	TKBMDn0
	1/0	0	0	1/0	0	0	0	0

	7	6	5	4	3	2	1	0
TKBIOCn0	-	-	-	-	TKBTOLn1	TKBTOLn0	TKBTODn1	TKBTODn0
	0	0	0	0	1/0	1/0	1/0	1/0

	7	6	5	4	3	2	1	0
TKBIOCn1	-	-	-	-	-	-	TKBTOEn1	TKBTOEn0
	0	0	0	0	0	0	1/0	1/0

TKBCRn0	0000H to FFFFH
TKBCRn1	0000H to FFFFH
TKBCRn2	0000H to FFFFH
TKBCRn3	0000H to FFFFH
TKBTGCRn	0000H to FFFFH
TKBSIRn0	0000H
TKBSIRn1	0000H
TKBSSRn0	00H
TKBSSRn1	00H
TKBDNRn0	00H
TKBDNRn1	00H
TKBMFRn	0000H to FFFFH

□ : Setting is fixed for this mode: ▨ : Setting is not needed (default setting)

10.4.7 Interleave PFC (Power Factor Correction) output mode

This is the mode that can generate a signal as interleave output that controls PFC circuit which regulates the harmonic current of the power source.
As interleaved PFC circuit can regulate peak input current at greater extent than single PFC circuit, it can make parts smaller and implement high powered power source units.
Interleaved PFC control requires two inputs for zero current detection and two PWM outputs for switching.
TMKB implements the interleaved PFC control by combining external interrupt input INTP10 and timer output TKBO0, and interrupt input INTP11 and timer output TKBO1.
TKBO0 generates pulse output based on the signal input of INTP10, and TKBO1 generates pulse output based on the signal input of INTP11.
In this case, it controls TKBO1 output to be 180 degree of phase shift based on the output timing of TKBO0.

Remark Single PFC control can be implemented in standalone mode (periodic control by external input trigger).
For more detail, see **10.4.6 Standalone mode (period controlled by external trigger input)**.

Timer KBn restart period is set by TKBCRn0 for cases in which external input INTP10 not being detected.
Active width for TKBO0 output is set by TKBCRn1.
Active width for TKBO1 output is set by TKBCRn3.
Therefore, TKBCRn2 is not used for this function.

Remark Interleave PFC (Power Factor Correction) output mode does not use TKBCRn2.

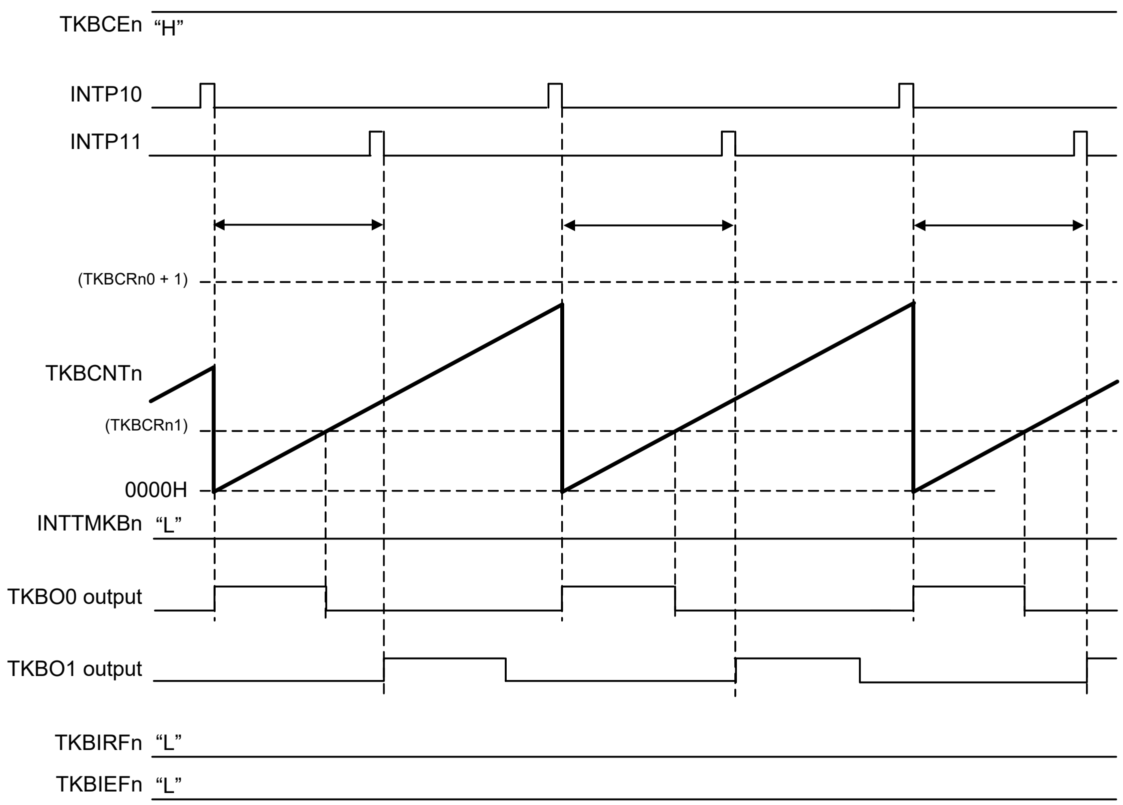
The setting value for TKBTOLn0 bit and TKBTODn0 bit, and TKBTOLn1 bit and TKBTODn1 bit must be the same value. This makes that when the default level is low (high) level, the active level becomes high (low) level.

[Calculation formula for TKBO0 output & TKBO1 output]
Pulse period (MAX)[Note 1] = (TKBCRn0 setting + 1) × Count clock period
Active width for TKBO0 output = TKBCRn1 setting × Count clock period
Active width for TKBO1 output = TKBCRn3 setting × Count clock period
Width of the cycle-to-cycle phase difference during TKBO1 output [Note 2] = INT[(width of the previous period – 1) [Note 3]/2 + 1] × Count clock period

Note 1. This is the timer KBn restart period in case when external interrupt input INTP10 not being detected.
Note 2. Except when condition No. 7 holds.
Note 3. When condition No. 1 holds, this is determined by the setting of TKBCRn0.

Figure 10 - 45 shows overview of the basic operation of the interleave PFC mode. In the basic operation of the interleave PFC mode, TKBCNTn is incremented from 0000H by using INTP10 as a trigger. In this case, TKBO0 becomes active level, then becomes inactive level when it matches with the setting value of TKBCR1 register.
TKBO1 becomes active level by being triggered by INTP11 which has a phase shifted from the one of INTP10, and becomes inactive level when it matches with the setting value of TKBCRn3 register.
Another INTP10 comes in before TKBCNTn matches with the setting value of TKBCRn0 register, then the above operation is repeated.

Figure 10 - 45 Operation Outline of Basic Operation for Interleave PFC Mode (at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



(1) Output condition of TKBO1 at interleave PFC

There are output conditions for TKBO1 output which are controlled according to the table below.

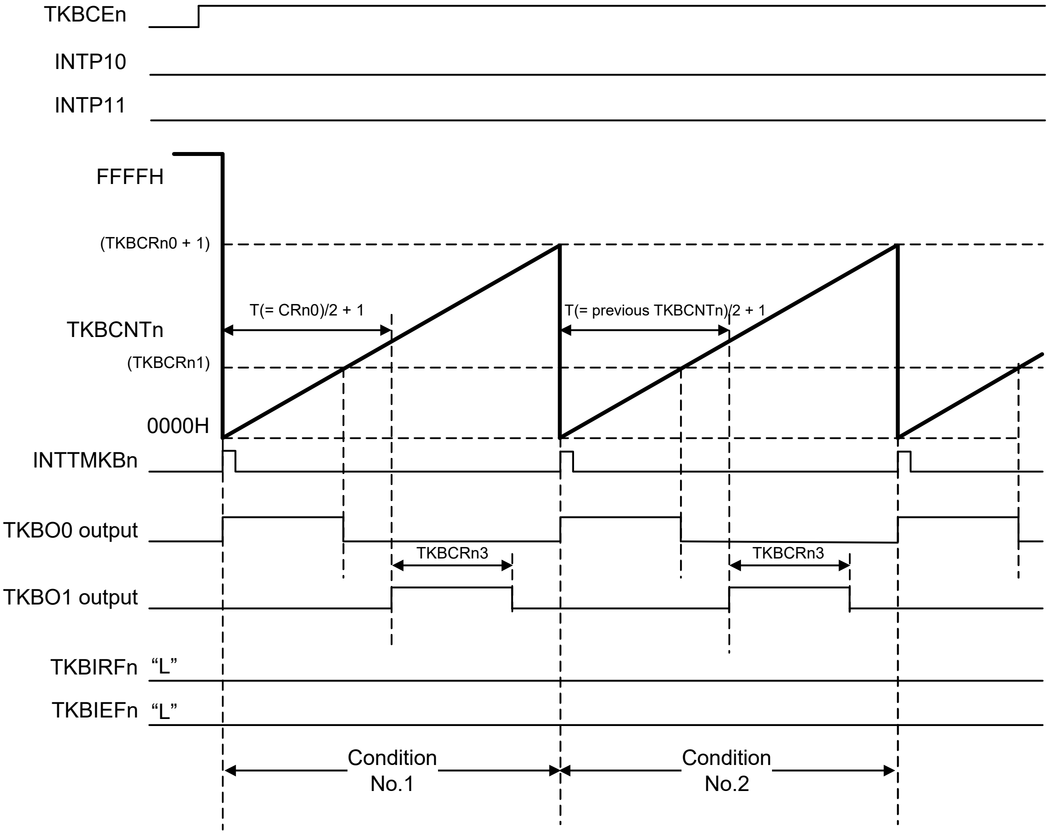
Condition No.	Judgment Status 1 (INTP10 Input)	Judgment Status 2 (Matching with CR0/INTP11 Input)	Judgment Status 3 (Period Width)	Output Status
1	First period (Generate a wave form setting T for CR0)	–	–	Output by T/2
2	INTP10 input not detected	Matching of CNTn and CRn0 (Ignore INTP11 input detection)	Subsequent period is over T/2	Output by T/2
3	↑	↑	Succeeding period is below T/2	Maintain the status
4	Subsequent period of No.3	–	–	Output by T/2
5	INTP10 input detected (for the first time)Note 1	–	–	Output by T/2
6	INTP10 input detected (from the second time)Note 2	INTP11 detected (within the range from previous TKBO1 falling edge to T/2)	–	Output by T/2
7	INTP10 input detected (from the second time)Note 2	INTP11 detected (T/2 to T/2+T/(TKBIRSn1 and TKBIRSn0 setting) range)	–	Output by Trigger Input
8	INTP10 input detected (from the second time)Note 2	INTP11 detected after the range (T/2+T/(TKBIRSn1 and TKBIRSn0 setting))	–	Maintain the status
9	Subsequent period of No.8	–	–	Output by T/2
10	INTP10 input detected	–	Succeeding period is below T/2	Maintain the status
11	Subsequent period of No.10	–	–	Output by T/2

Note 1. INTP10 input detected (for the first time) means that the previous period wasn't cleared for INTP10 input being detected.

Note 2. INTP10 input detected (from the second time) means that the previous period being cleared for INTP10 input being detected.

Please see the following figures of wave form corresponding to each “Condition No.”

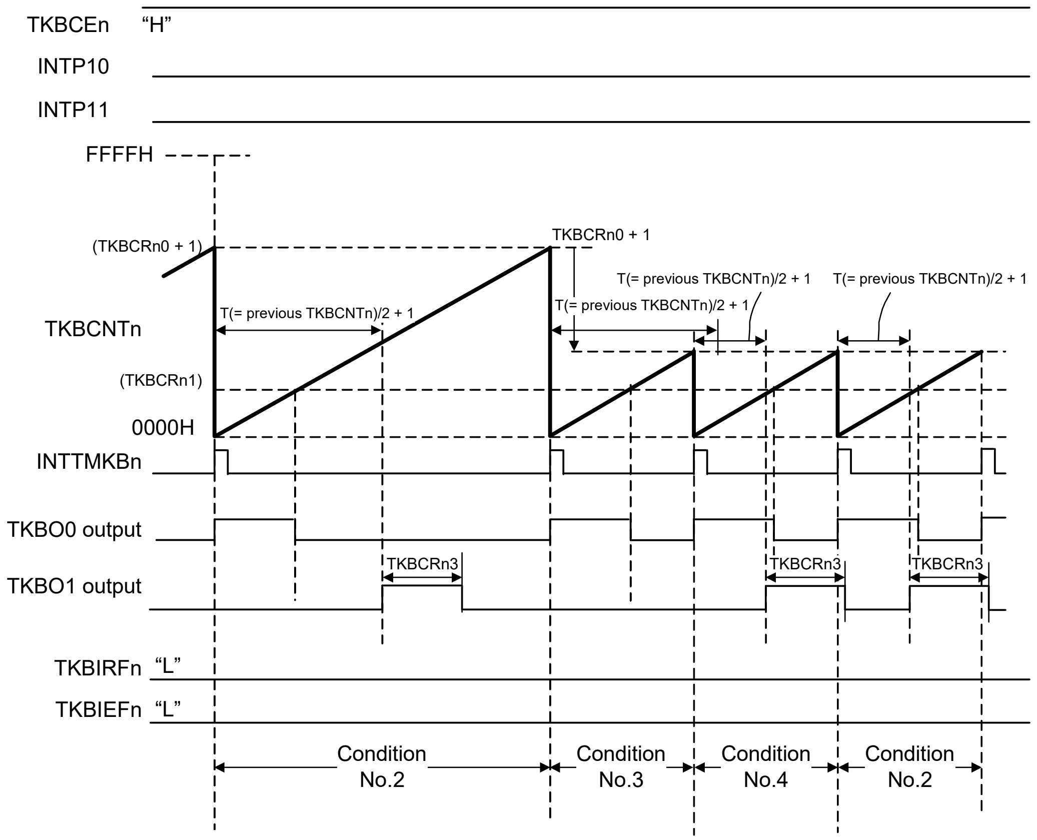
Figure 10 - 46 Figure of Timing of Interleave PFC Mode (Operation for Conditions No. 1 and No. 2) (at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



Condition No.1 Only for the first period after TKBCEn = 1 setting, TKBO1 with setting width of TKBCRn3 is output setting "T" as TKBCRn0.

Condition No.2 In the second period, TKBO1 with setting width of TKBCRn3 is output at T/2 of previous period.

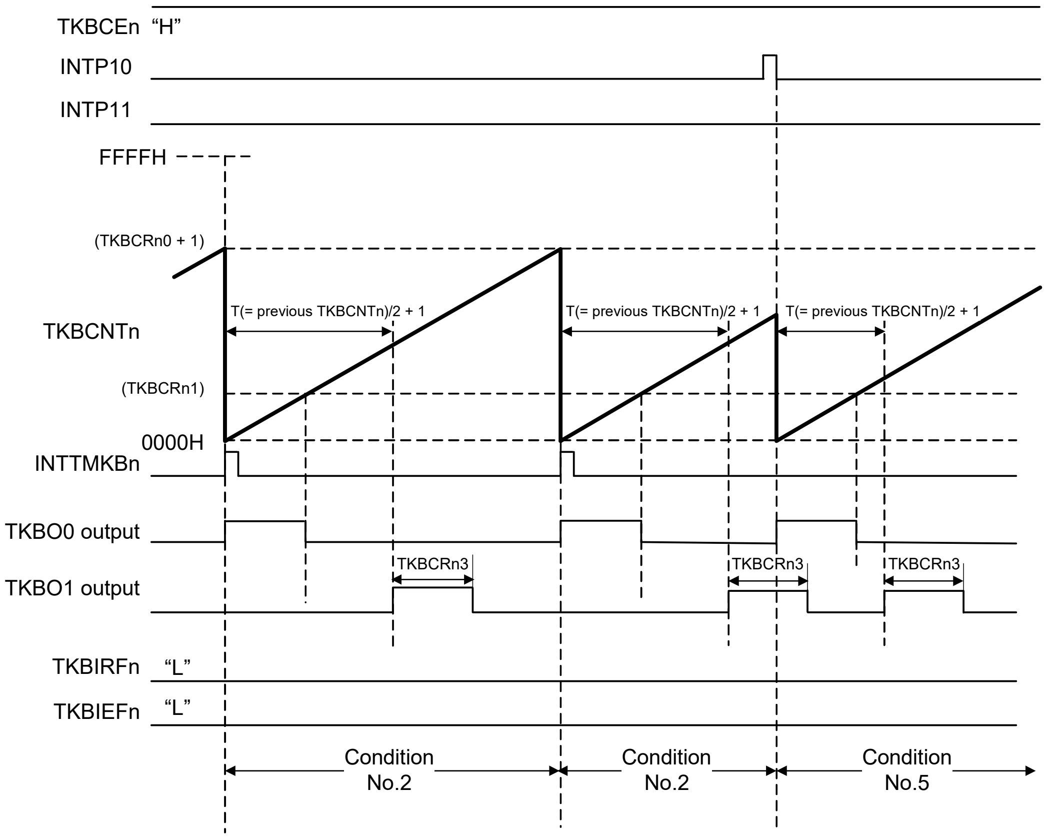
Figure 10 - 47 Figure of Timing of Interleave PFC Mode (Below T/2s No. 3 and No. 4)
(at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



Condition No.3 TKBO1 keeps the status and T/2 of the previous period not ensured.

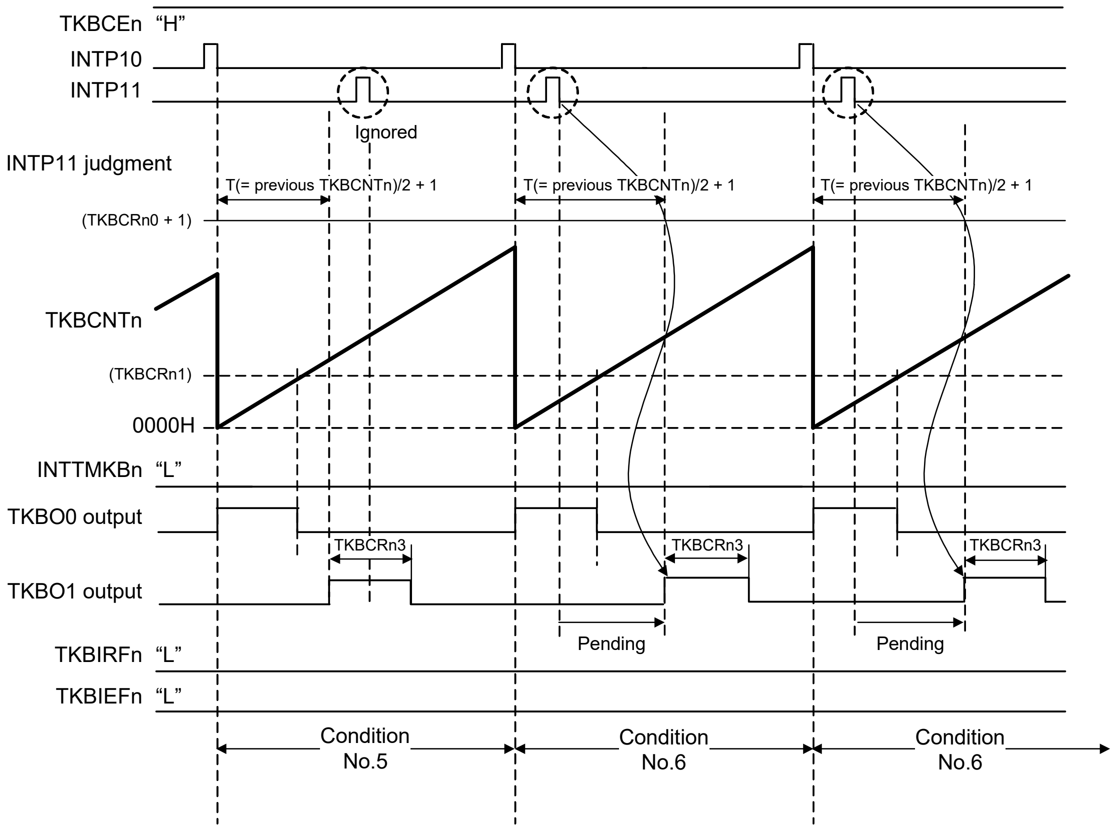
Condition No.4 TKBO1 with setting width of TKBCRn3 is output at T/2 of previous period.

Figure 10 - 48 Figure of Timing of Interleave PFC Mode (Operation for Condition No. 5)
(at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



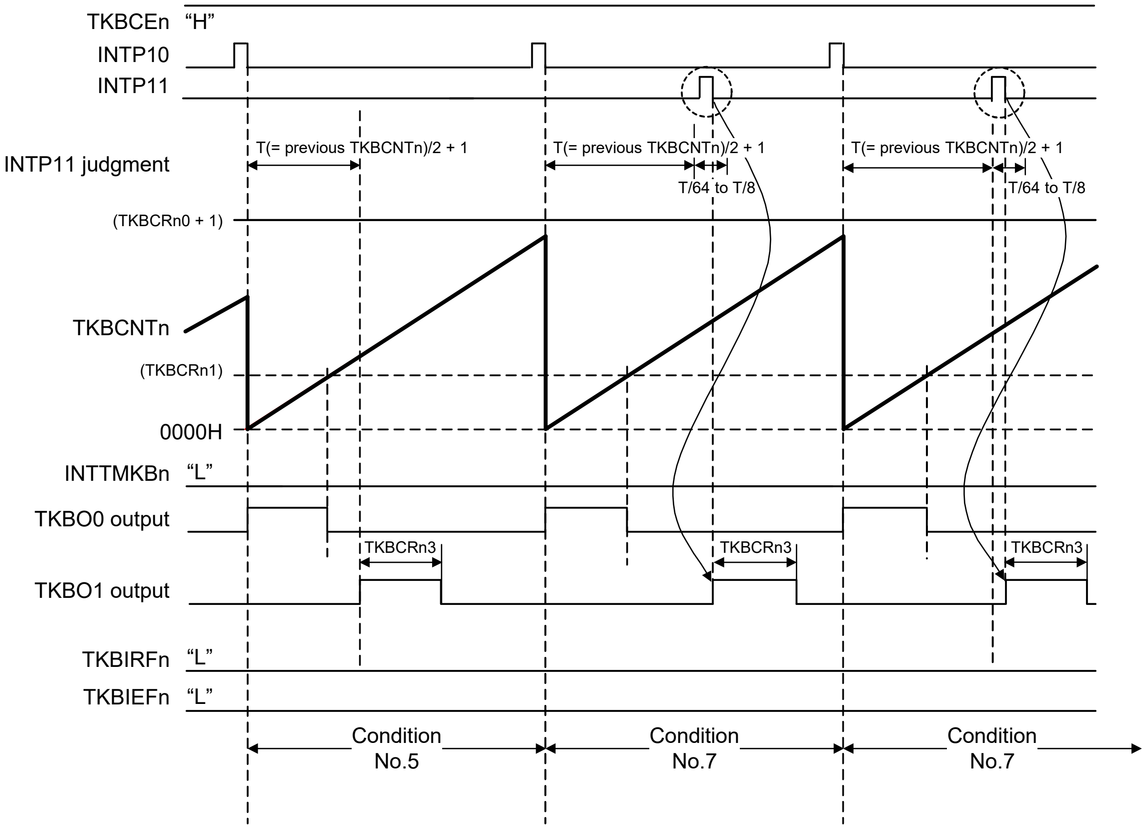
Condition No.5 INTP10 which was first detected after setting TKBCEn = 1 outputs TKBO1 with setting width of TKBCRn3.

Figure 10 - 49 Figure of Timing of Interleave PFC Mode (Operation for Conditions No. 6) (at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



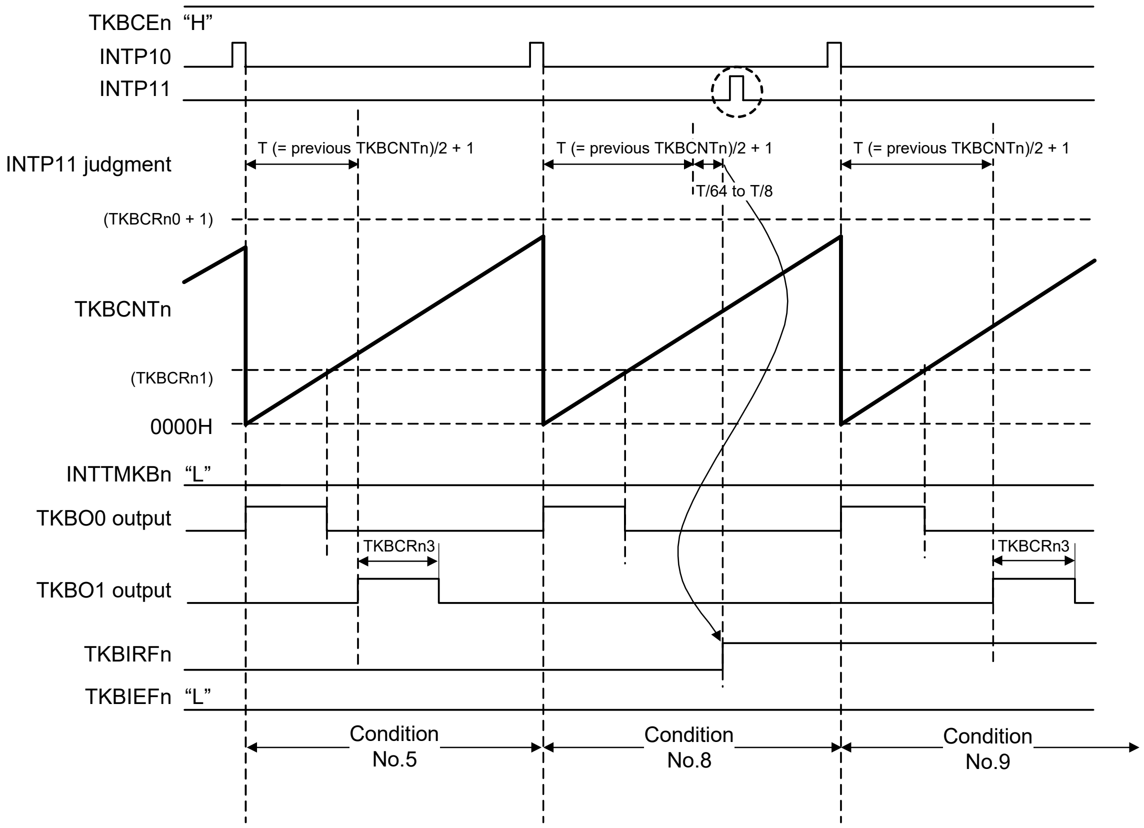
Condition No.6 TKBO1 with setting width of TKBCRn3 is output at T/2 of previous period as INTP11 input is below T/2 of the previous period.

Figure 10 - 50 Figure of Timing of Interleave PFC Output Mode (Operation for Conditions No. 7) (at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



Condition No.7 After the detection of INTP10 when INTP11 is detected over T/2 of the previous period and within T/2 + T/m (m stands for 8/16/32/64; set by TKBIRSn1, TKBIRSn0), TKBO1 is output by setting width of TKBCRn3.

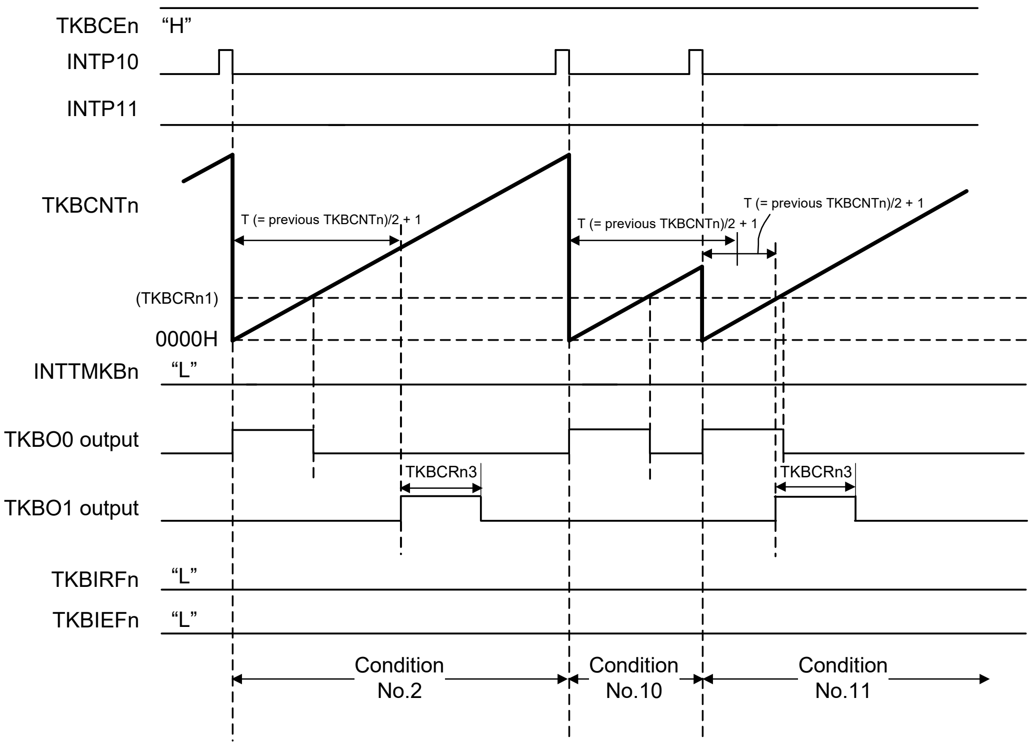
Figure 10 - 51 Figure of Timing of Interleave PFC Output Mode (Operation for Conditions No. 8 to 9) (at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



Condition No.8 INTP11 wasn't detected within T/2 + T/m (m stands for 8/16/32/64; set at TKBIRSn1 & TKBIRSn0) of the previous period and TKBO1 maintains the status. Then TKBIRFn is set by "1".

Condition No.9 TKBO1 with setting width of TKBCRn3 is output at T/2 of previous period.

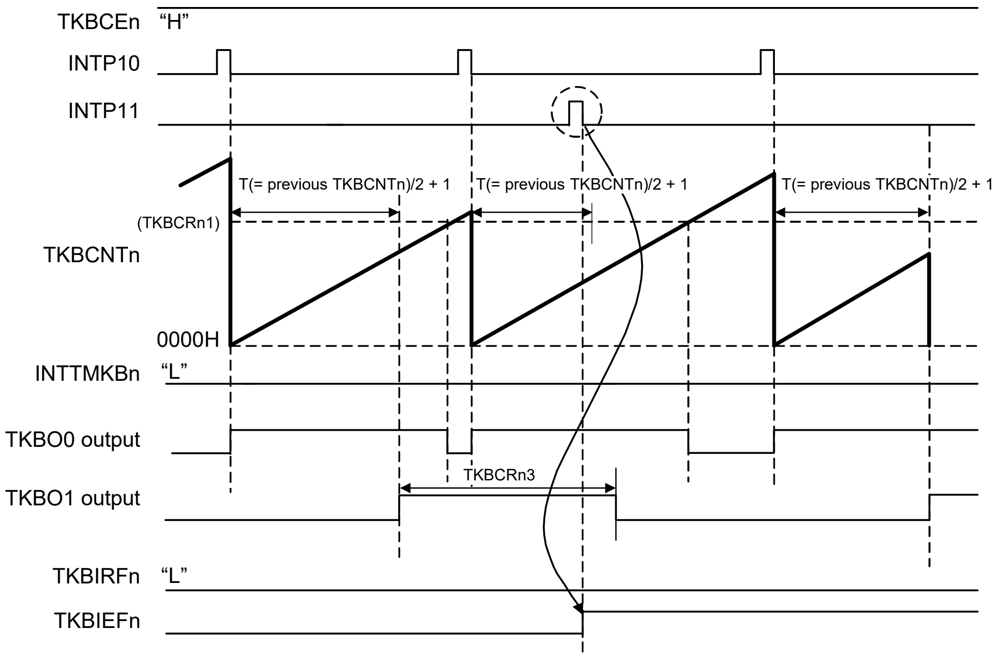
Figure 10 - 52 Figure of Timing of Interleave PFC Output Mode (Operation for Conditions No. 10 and No. 11) (at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))



Condition No.10 TKBO1 keeps the status and T/2 of the previous period not ensured.

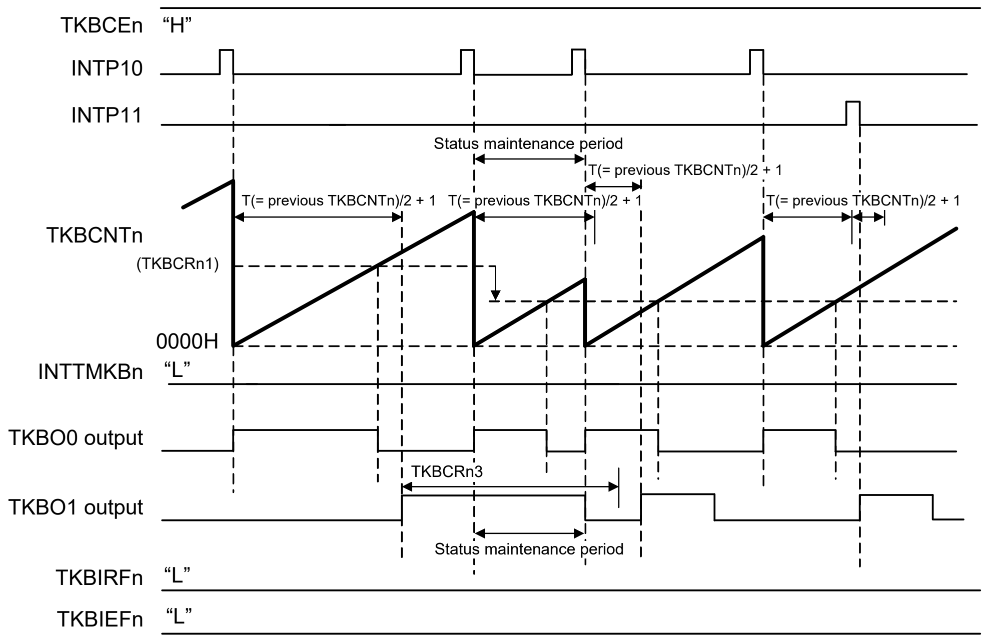
Condition No.11 TKBO1 with setting width of TKBCRn3 is output at T/2 of previous period.

Figure 10 - 53 Figure of Timing of Interleave PFC Output Mode (In Case When INTP11 Input Was Detected During TKBO1 Output)



When INTP11 input is detected during TKBO1 output of the previous period, this trigger is ignored. This is when TKBIEFn is set by "1".

Figure 10 - 54 Figure of Timing of Interleave PFC Output Mode (Output of TKBO1 Is at the Width of the Previous Output Width and Exceeds Period of Status Maintenance)


TKBCEn "H"
INTP10
INTP11
Status maintenance period
T(= previous TKBCNTn)/2 + 1
T(= previous TKBCNTn)/2 + 1
T(= previous TKBCNTn)/2 + 1
T(= previous TKBCNTn)/2 + 1
TKBCNTn
(TKBCRn1)
0000H
INTTMKBn "L"
TKBO0 output
TKBCRn3
TKBO1 output
Status maintenance period
TKBIRFn "L"
TKBIEFn "L"

When TKBO1 output of the previous output width is long which exceeds status maintenance period, it is default output compulsively at the starting timing of the subsequent period following the completion of the status maintenance period.

(2) List of register setting at interleave PFC output mode

bit No.	15	14	13	12	11	10	9	8
TKBCTLn0	-	-	TKBSSEn1	TKBDIEn1	-	-	TKBSSEn0	TKBDIEn0
Setting	0	0	0	0	0	0	0	0

bit No.	7	6	5	4	3	2	1	0
TKBCTLn0	TKBMFEn	-	TKBIRSn1	TKBIRSn0	-	TKBTSEn	TKBSTSn1	TKBSTSn0
Setting	1/0	0	1/0	1/0	0	1	0	0

bit No.	7	6	5	4	3	2	1	0
TKBCTLn1	TKBCEn	-	-	TKBCKSn	-	-	TKBMDn1	TKBMDn0
Setting	1	0	0	1/0	0	0	1	1

bit No.	7	6	5	4	3	2	1	0
TKBIOCn0	-	-	-	-	TKBTOLn	TKBTOLn0	TKBTODn1	TKBTODn0
Setting	0	0	0	0	1/0	1/0	1/0	1/0

bit No.	7	6	5	4	3	2	1	0
TKBIOCn1	-	-	-	-	-	-	TKBTOEn1	TKBTOEn0
Setting	0	0	0	0	0	0	1/0	1/0

Register	Setting range
TKBCRn0	0000H to FFFFH
TKBCRn1	0000H to FFFFH
TKBCRn2	0000H
TKBCRn3	0000H to FFFFH
TKBTGCRn	0000H to FFFFH
TKBSIRn0	0000H
TKBSIRn1	0000H
TKBSSRn0	00H
TKBSSRn1	00H
TKBDNRn0	00H
TKBDNRn1	00H
TKBMFRn	0000H to FFFFH

□ : Setting is fixed for this mode: ▨ : Setting is not needed (default setting)

10.5 Option Function of 16-bit Timers KB0

Option function can be added to timers KB0.

The following table shows available option for each operation modes for timer KB0.

Operation Mode		Standalone Mode		Interleave PFC Output Mode
Period Controlling Method for Operation Mode		Period Controlled by TKBCR0m	Period Controlled by Trigger	Period Controlled by INTP10/TKBCR0m
Optional Function	Trigger Signal Output Function	√	√	√
	PWM Output Dithering Function	√	–	–
	PWM Output Smooth Start Function	√	–	–
	Maximum Frequency Limit Function	–	√	√

Remark 1. m = 0 to 3

Remark 2. For details of the operation specifications, see **10.4.2 Default level and active level** and **10.4.3 Stop/restart operation**.

10.5.1 Trigger Output Function

Timer KBn trigger output signal can be generated by setting 16-bit timer KB trigger compare register n (TKBTGCRn). This trigger output signal can be used as an ELC event input signal (ELSELR17).
Timer KBn trigger output signal is output by detecting the match between TKBCNTn and TKBTGCRn which makes trigger output available at any timing corresponding to set period of TKBCRnm. Output width of timer KBn trigger output signal is the width of 1 clock of timer clock. Trigger output timing from PWM output period start can be calculated by following formula;

Trigger output timing = TKBTGCRn setting × Count clock period

Caution Timer KBn trigger output signal is not output when TKBCRn0 < TKBTGCRn.

Figure 10 - 55 Trigger Output Function for Standalone Mode (Period Controlled by TKB0CR0)

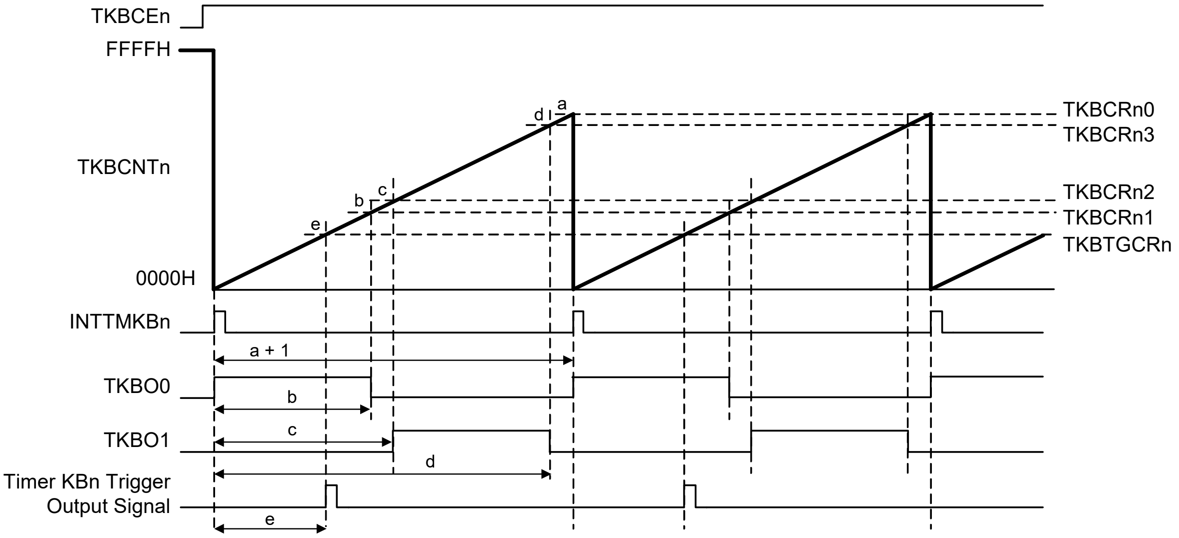
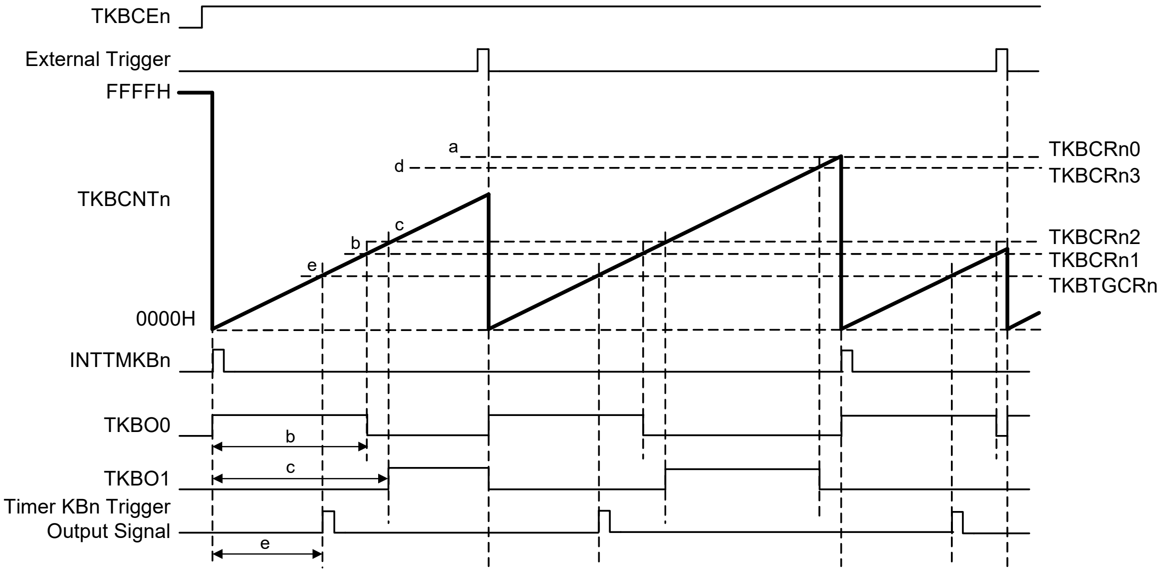


Figure 10 - 56 Trigger Output Function for Standalone Mode (Period Controlled by External Trigger Input)



Remark n = 0

10.5.2 PWM output dithering function

16-bit timer KB is available for high resolution PWM output using PWM output dithering function.

Having 16 periods of PWM period as standard, 16 times higher PWM output is available for average resolution through extension of active period by 1 count clock at n period (n = 0 to 15) during 16 periods.

The period extending active period during 16 periods by 1 count clock is defined by TKBDNRnp.

The relationship between TKBDNRnp and the period extending active period for 1 count clock is as follows:

Figure 10 - 57 Figure of Relationship Between TKBDNRnp and the Period which Extends Active Period for 1 Count Clock

Period Repetitions (N)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0																
1	▨															
2	▨								▨							
3	▨				▨				▨							
4	▨				▨				▨				▨			
5	▨		▨		▨				▨				▨			
6	▨		▨		▨				▨		▨		▨			
7	▨		▨		▨		▨		▨		▨		▨			
8	▨		▨		▨		▨		▨		▨		▨		▨	
9	▨	▨	▨		▨		▨		▨		▨		▨		▨	
10	▨	▨	▨		▨		▨		▨	▨	▨		▨		▨	
11	▨	▨	▨		▨	▨	▨		▨	▨	▨		▨		▨	
12	▨	▨	▨		▨	▨	▨		▨	▨	▨		▨	▨	▨	
13	▨	▨	▨	▨	▨	▨	▨		▨	▨	▨		▨	▨	▨	
14	▨	▨	▨	▨	▨	▨	▨		▨	▨	▨	▨	▨	▨	▨	
15	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	

Remark 1. □ Cell period: Reset output waveform via settings for TKBCRn1 and TKBCRn3 registers

▨ Cell period: Reset output waveform via settings +1 for TKBCRn1 and TKBCRn3 registers

Remark 2. n = 0

p = 0, 1

Figure 10 - 58 Figure of Waveform at Dithering Operation

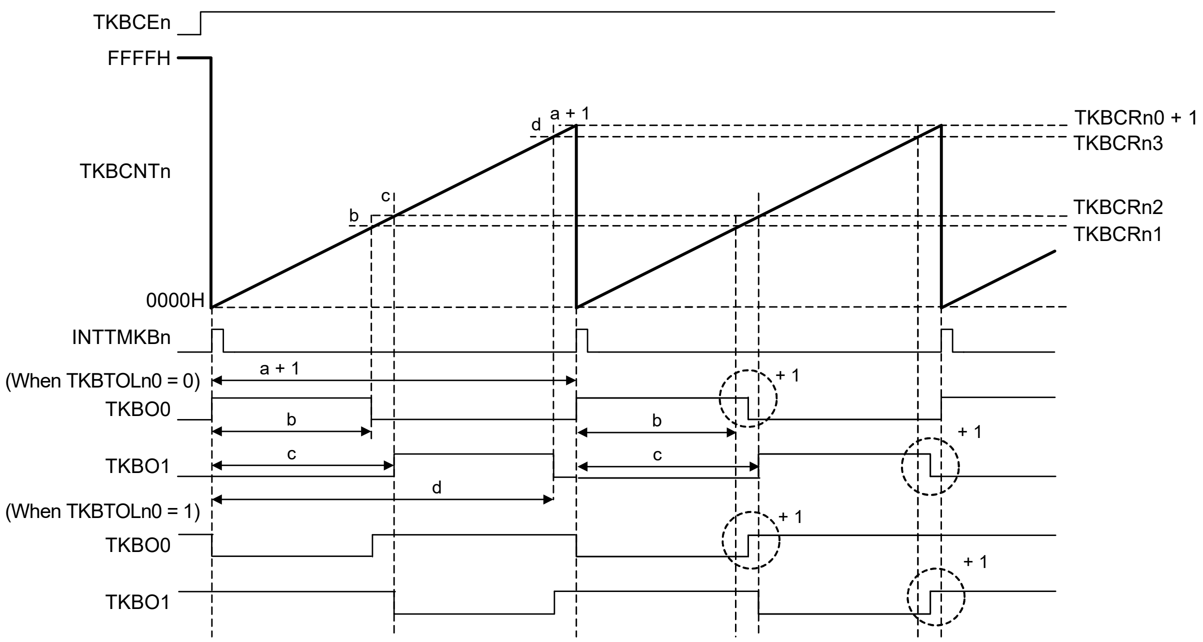


Figure 10 - 59 Figure of Waveform at Dithering Operation
(When TKBCRn1 = TKBCRn0 (100% Nearest Neighbor), TKBCRn2 = TKBCRn3 (0% Nearest Neighbor)

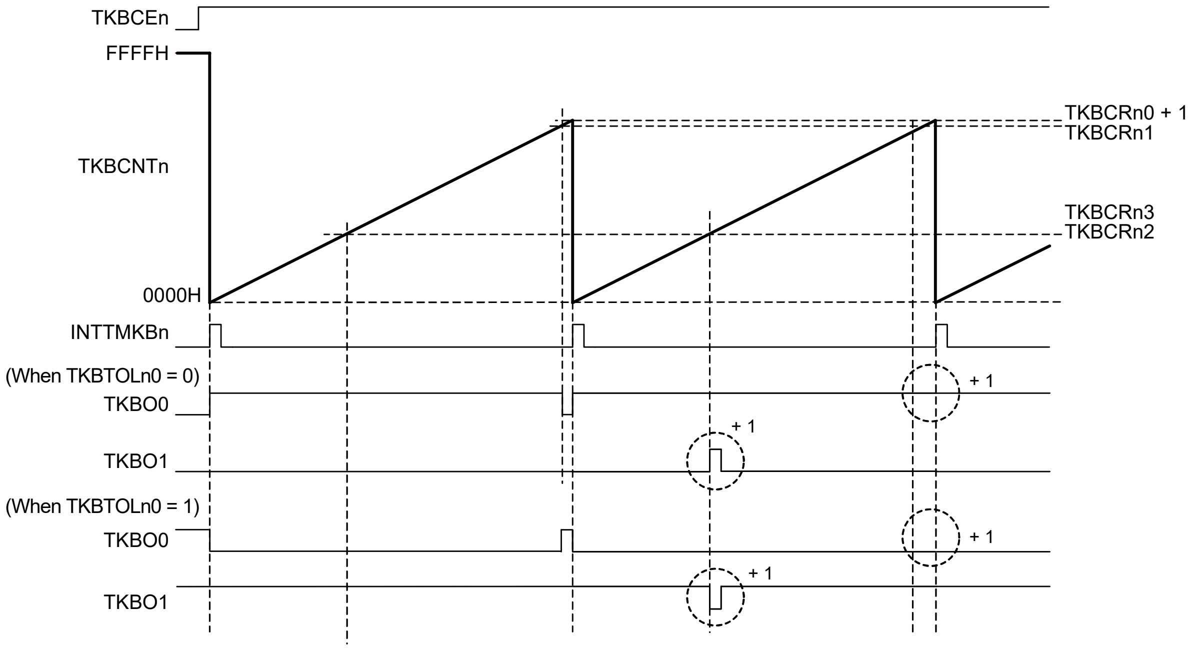
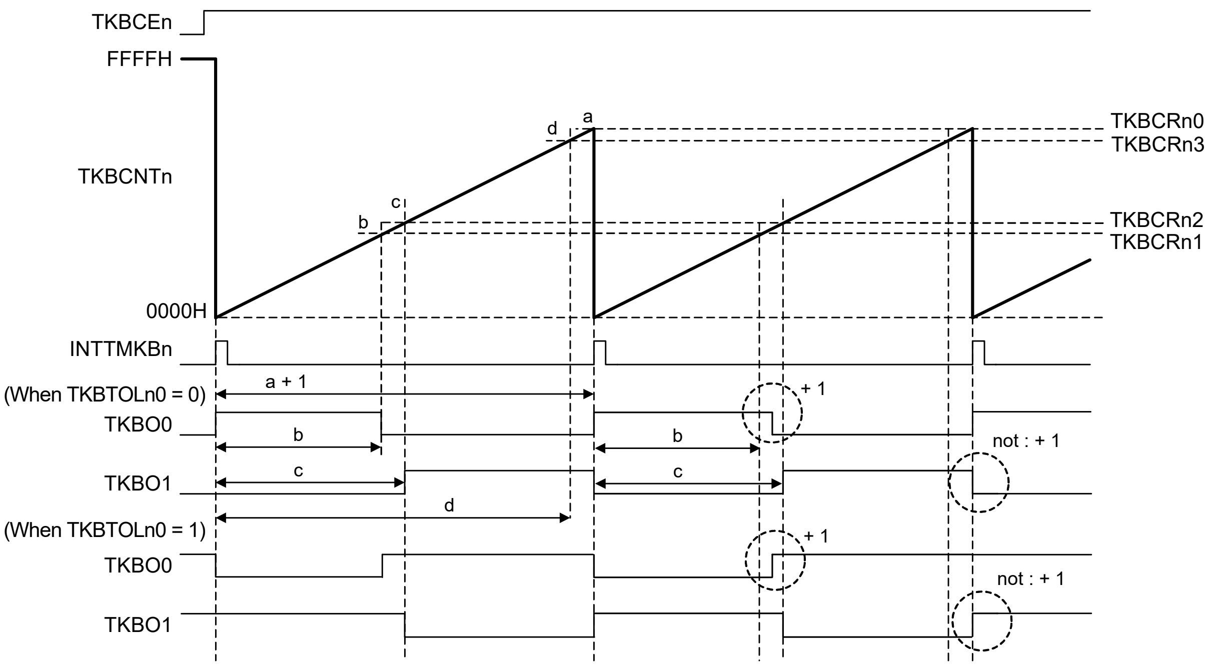


Figure 10 - 60 Figure of Waveform at Dithering Operation (When TKBCRn3 = TKBCRn0+1)



(1) Availability for operation mode

This shows enable or disable status under each mode that is specified by TKBCTLn0 register (TKBSTSn1 and TKBSTSn0 bit) and TKBCTLn1 register (TKBMDn1 and TKBMDn0 bit).

Operation Mode	TKBMDn1, TKBMDn0	TKBSTSn1, TKBSTSn0	Setting Availability
Standalone mode (period controlled by TKBCRn0)	00	00	√
Standalone mode (period controlled by external trigger input)	00	01/10/11	-
Interleave PFC output mode	11	-	-

PWM output dithering function is available when external trigger input isn't used and the period being controlled by TKBCRn0.
TKBDNRn0/TKBDNRn1 control PWM output dithering function of relative TKBO0/TKBO1.

Caution 1.[Overwrite during the operation (TKBCEn = 1) of TKBDNRn0/TKBDNRn1 register]
Regarding TKBDNRn0/TKBDNRn1 owns buffer, overwrite during the operation (TKBCEn = 1) is available.
At this time, batch overwriting is available via writing “1” to TKBRDTn bit.

Caution 2. [Access by TKBCRLDn0/TKBCRLDn1 register]
TKBCRLDn0 is a 16-bit register mapping lower 8 bits of TKBCRn1 and TKBDNRn0.
TKBCRLDn1 is a 16-bit register mapping lower 8 bits of TKBCRn3 and TKBDNRn1.
Value of TKBDNRn0/TKBDNRn1 is changed even in case that they have accessed TKBCRLDn0/TKBCRLDn1 register.
Value of TKBCRn1/TKBCRn3 is changed even in case that they have accessed TKBCRLDn0/TKBCRLDn1 register.
Only the lower 8 bits of TKBCRn1/TKBCRn3 are changed when TKBCRLDn0/TKBCRLDn1 register is accessed.

Caution 3. [To Combine PWM Output Smooth Start Function with PWM Output Dithering Function]
PWM output dithering function is invalid during the execution of PWM output smooth start function (TKBSSFnp = 1).
PWM output dithering function is valid when PWM output smooth start function is stopped (TKBSSFnp = 0).

10.5.3 PWM output smooth start function

Timer KB0 own PMW output smooth start function corresponding to rush current control and over-voltage prevention. PWM output smooth start function begins at timer start timing. PWM output smooth start function is initiated by the timer start timing. The process that a user has performed with software in the past can be easily accomplished with the optional function of the hardware. It generates PWM waveform setting the default duty register (TKBSIRnp) of 16-bit timer KB smooth start as 1 period active period. After outputting PWM waveform of the same active period adding 1 to the value of repetition assigned by 16-bit timer KB smooth start step width register (TKBSSRnp), outputs the same TKBSSRnp + 1 period waveform again, executing “active period + 1”. After repeating the action, PWM output smooth start function is cancelled when the same active period defined by TKBCRn1 and TKBCRn3 is reached.

16-bit timer KB smooth start default duty register should be set according to following condition;

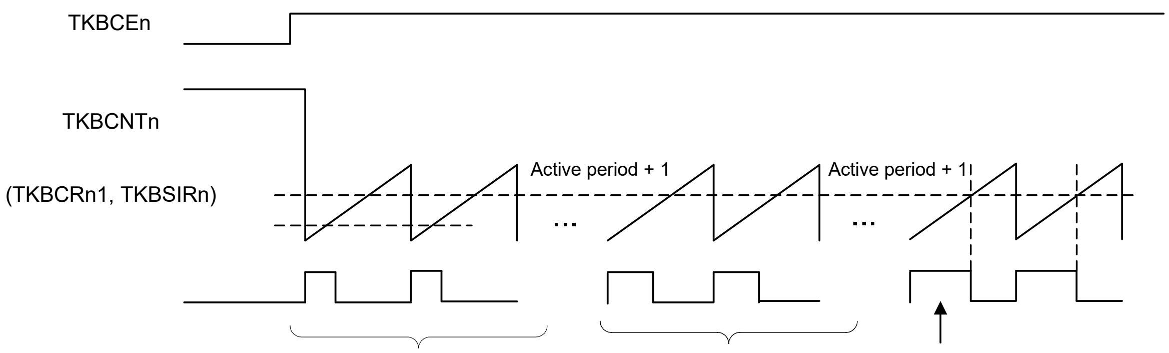
$\mathrm{0000H} \le \mathrm{TKBSIRn0} < \mathrm{TKBCRn1} \le \mathrm{TKBCRn0} + 1$

$\mathrm{TKBCRn2} \le \mathrm{TKBSIRn1} < \mathrm{TKBCRn3} \le \mathrm{TKBCRn0} + 1$

It should be set according to following condition when synchronous start/clear mode is applied;

$\mathrm{TKBCRn0} \le \mathrm{TKBSIRn0} < \mathrm{TKBCRn1} \le \mathrm{TKBCR00} + 1$ of Master

Figure 10 - 61 PWM Output Smooth Start Function



After detecting TKBCEn = 0 → 1, the n + 1 period (TKBSSRn0) is the value of default register (TKBSIRn0).

After n + 1 period, 1 is added to active time to be n + 1 period and repeated by the same value.

Continuation to be canceled when the period reaches TKBCRn1.

(1) Operation mode available for PMW output smooth start function

Operation Mode	TKBMDn1, TKBMDn0	TKBSTSn1, TKBSTSn0	Setting Available
Standalone mode (period controlled by TKBCRn0)	00B	00B	√
Standalone mode (period controlled by external trigger input)	00B	01B/10B/11B	–
Interleave PFC output mode	11B	–	–

(2) Overwrite during the operation (TKBCEn = 1) of TKBSIRn0/TKBSIRn1/TKBSSRn0/TKBSSRn1 registers

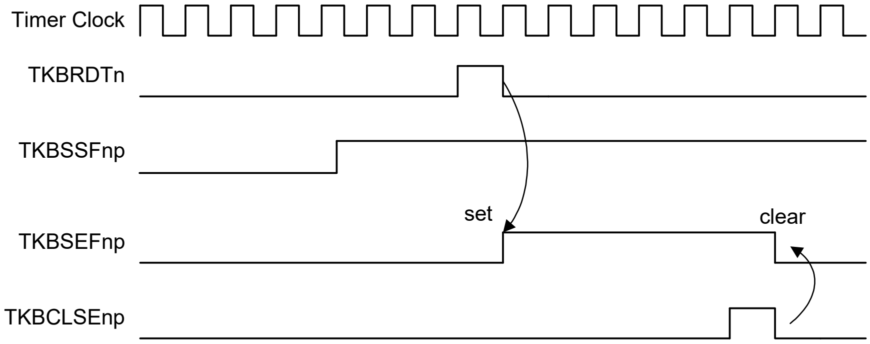
Overwrite during the operation (TKBCEn = 1) is available for TKBSIRn0/TKBSIRn1/TKBSSRn0/TKBSSRn1. TKBSIRn0/TKBSIRn1/TKBSSRn0/TKBSSRn1 own buffer and batch overwriting is available via writing "1" to TKBRDTn bit. In TKBSIRn0/TKBSIRn1, the buffer value at starting PWM output smooth start function is duty default, and in TKBSSRn0/TKBSSRn1, it is comparison value of internal 4-bit counter.

The internal 4-bit counter is incremented upward using the period of TKBCNTn as a count clock and it becomes 0H when it matches with TKBSSRn0/TKBSSRn1, then continues its counting operation.

(3) Overwrite during the operation (TKBCEn = 1) of TKBCRn0/TKBCRn1/TKBCRn2/TKBCRn3/TKBSIRn0/TKBSIRn1/TKBSSRn0/TKBSSRn1 registers

When TKBRDTn is set as "1" during the period of PWM output soft-start (TKBSSFn0 = 1 and TKBSSFn1 = 1), batch overwrite is masked and TKBSEFnp flag is set. In order to perform batch overwrite, clear TKBSEFnp and confirm TKBSSFnp becomes "0", then set "1" to TKBRDTn.

Figure 10 - 62 Overwrite During the Smooth Start Function Operation (TKBSSFnp = 1) of TKBCRn0/TKBCRn1/TKBCRn2/TKBCRn3/TKBSIRn0/TKBSIRn1/TKBSSRn0/TKBSSRn1 Registers



(4) To Combine PWM output smooth start function with PWM output dithering function

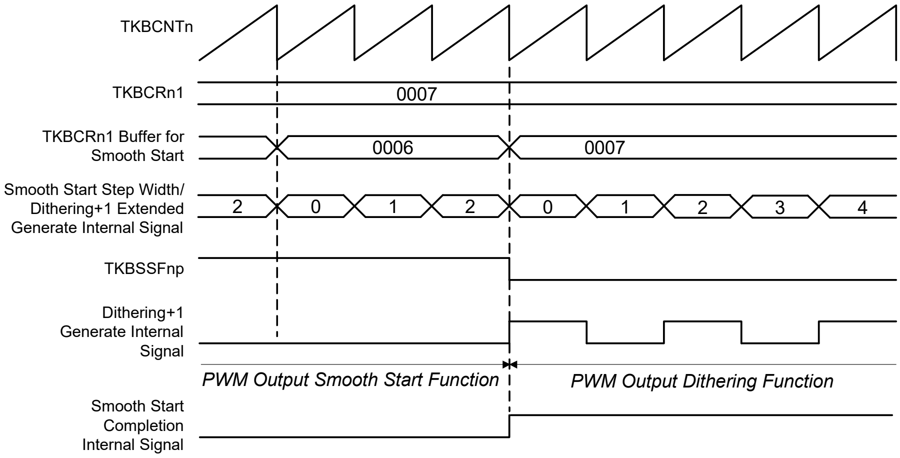
PWM output dithering function is invalid during the execution of PWM output smooth start function (TKBSSFnp = 1).

PWM output dithering function will be valid when PWM output smooth start function is stopped (TKBSSFnp = 0).

(5) Completion of PWM output smooth start function and operation of TKBSSFnp

Figure 10 - 63 shows when TKBCRn1 is 0007H, TKBDNRnp is 70H and TKBSSRnp is 02H. At the timing that TKBCRn1 = 0007H and the value of TKBCRn1 buffer for internal soft-start matches, TKBSSFnp is cleared then dithering function begins.

Figure 10 - 63 Completion of PWM Output Smooth Start Function and Operation of TKBSSFnp


TKBCNTn
TKBCRn1
0007
TKBCRn1 Buffer for Smooth Start
0006
0007
Smooth Start Step Width/ Dithering+1 Extended Generate Internal Signal
2
0
1
2
0
1
2
3
4
TKBSSFnp
Dithering+1 Generate Internal Signal
PWM Output Smooth Start Function
PWM Output Dithering Function
Smooth Start Completion Internal Signal

10.5.4 Maximum frequency limit function

Timers KB0 is a function that regulates the minimum period of the counter clear (maximum frequency) in the periodic control by external trigger or interleaved PFC output mode.

When this function is used, if external trigger input which performs the counter clear occurs while the counter value is less than the setting value of maximum frequency limit register (TKBMFRn), it performs the counter clear after it continues counting until it reaches the setting value of TKBMFRn.

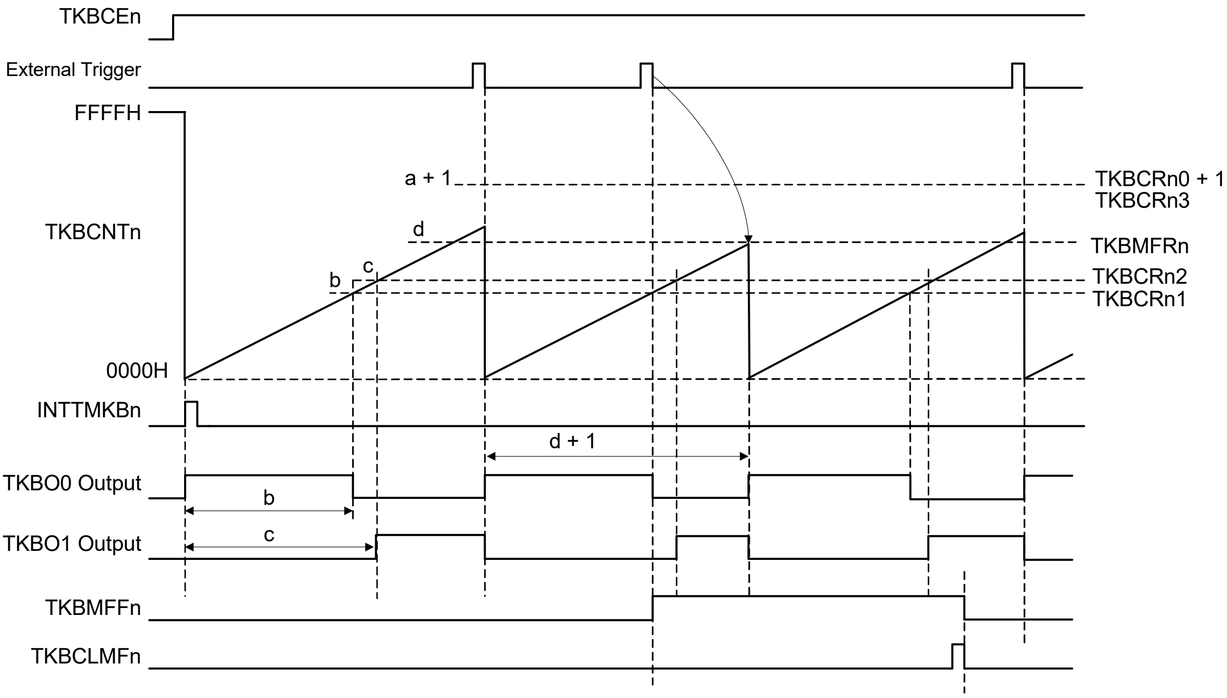
(1) Formula for maximum frequency limit (= 1/Minimum period)

Minimum period (= 1/Maximum frequency limit) = (TKBMFRn setting + 1) × Count clock period

Caution The following condition need to be satisfied: TKBMFRn setting ≤ TKBCRn0 setting

When counter value is smaller than TKBMFRn at the timing for external trigger input detection, “1” is set for TKBMFFn flag. TKBMFFn flag is cleared to “0” by writing “1” to TKBCLMFn bit.

Figure 10 - 64 Maximum Frequency Limit Function



Remark Period controlled by external trigger input.

(2) Operation Mode Available for Maximum Frequency Limit Function

Operation Mode	TKBMDn1, TKBMDn0	TKBSTSn1, TKBSTSn0	Setting Available
Standalone mode (period controlled by TKBCRn0)	00B	00B	–
Standalone mode (period controlled by external trigger input)	00B	01B/10B/11B	√
Interleave PFC output mode	11B	–	√

Remark Available when the period is controlled by external trigger input.

10.6 Forced Output Stop Function

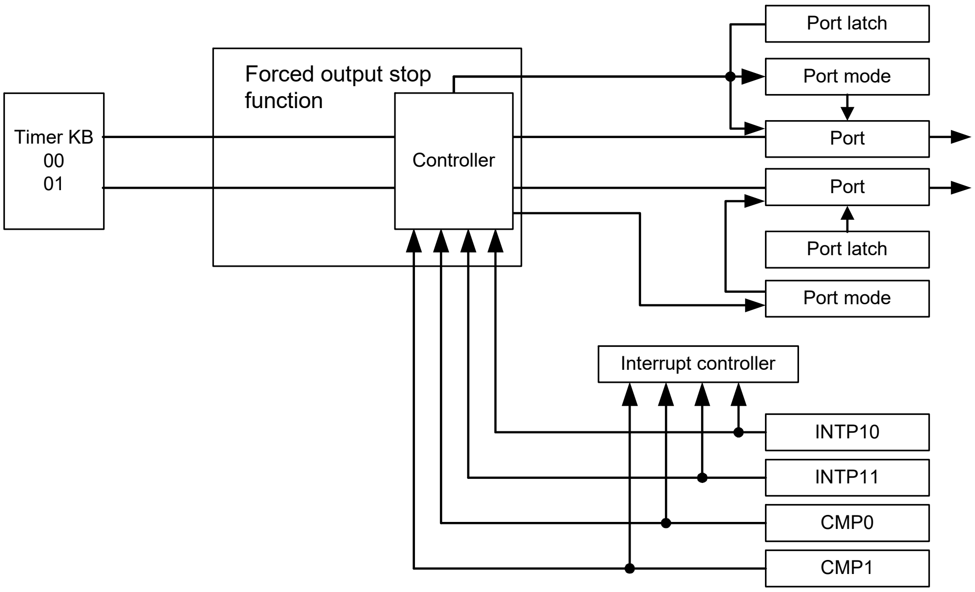
Forced output stop function is a function to protect power supply, etc.

If any abnormal situation that occurs in a power circuit configured outside of a micro-computer leads to over-voltage of over-current, making voltage or current sense signal into INTP10 and INTP11/comparator can protect the circuit by maintaining the timer output high impedance or fixed output state without being intermediated by a CPU's program control.

With this function, abnormality is identified only when input signal edge have been detected. Fixed level without edge is not recognized as abnormality.

The following figure shows the system structure of forced output stop function.

Figure 10 - 65 System Structure of Forced Output Stop Function



10.6.1 Forced output stop function 1 and 2

There are two ways of controls in forced output stop function. Forced output stop function 1 can select fixed level output or high-impedance output, and forced output stop function 2 can only set fixed level output. Then the difference of the control method is shown.

(1) Selectable Output Levels for Forced Output Stop Function 1 & 2

Selectable Output Levels	Forced Output Stop	
	Function 1	Function 2
High-impedance output	√	–
Low-level fixed output	√	√
High-level fixed output	√	√

(2) Start/cancel of Forced Output Stop Function 1 & 2

Function/Operation Details (Start Forced Output Stop)	Forced Output Stop	
	Function 1	Function 2
Start forced output stop following the detection of the rising edge of comparator output.	√	√
Start forced output stop following the rising or falling edge detection by external interruption input (INTP10 and INTP11).	–	√
Start forced output stop by software bit (TKBPAHTSnp) setting.	√	–

Function/Operation Details (Cancel Forced Output Stop)	Forced Output Stop	
	Function 1	Function 2
Cancel forced output stop by software bit (TKBPAHTTnp) setting.	√	–
Cancel forced output stop by synchronization with TMKB period after the software bit (TKBPAHTTnp) setting.	√	–
Forced output stop cancelled at the next counter period after the beginning of forced output stop.	–	√
Forced output stop cancelled at the next counter period after the detection of rising or falling edge by trigger signal of forced output stop.	–	√

Remark n = 0 to 2, p = 0, 1

(3) Trigger Signal Selectable and the Conditions for Trigger Bit Available for Forced Output Stop Function 1 & 2

Selectable Trigger Signals	Forced Output Stop	
	Function 1	Function 2
Comparator 0 and 1	√	√
External interrupt input (INTP10 and INTP11)	–	√

Trigger Bits Available	Forced Output Stop	
	Function 1	Function 2
TKBPAHTSnp (Trigger bit which starts forced output stop of TKBOp output)	√	–
TKBPAHTTnp (Trigger bit which cancels forced output stop of TKBOp output)	√	–

Remark n = 0, p = 0, 1

Table 10 - 4 External Trigger Assignment List of Forced Output Stop Function 1

	TKBO0	TKBO1
Comparator 0	√	√
Comparator 1	√	√
INTP10	−	−
INTP11	−	−

Table 10 - 5 External Trigger Assignment List of Forced Output Stop Function 2

	TKBO0	TKBO1
Comparator 0	√	√
Comparator 1	√	√
INTP10	√	√
INTP11	√	√

Caution For setting of INTP10/INTP11, see CHAPTER 19 COMPARATOR.

10.6.2 Configuration of forced output stop function

Forced output stop function includes the following hardware.

Table 10 - 6 Configuration of Forced Output Stop Function

Item	Configuration
Control registers	Peripheral enable register 2 (PER2) Forced output stop function control register n0 (TKBPACTLn0) Forced output stop function control register n1 (TKBPACTLn1) Forced output stop function control register n2 (TKBPACTLn2) Forced output stop function flag register n (TKBPAFLGn) Forced output stop function 1 start trigger register n (TKBPAHFSn) Forced output stop function cancel 1 trigger register n (TKBPAHFTn)

10.6.3 Registers controlling forced output stop function

Forced output stop function is controlled by the following registers.

- Forced output stop function control register n0 (TKBPACTLn0)
- Forced output stop function control register n1 (TKBPACTLn1)
- Forced output stop function control register n2 (TKBPACTLn2)
- Forced output stop function flag register n (TKBPAFLGn)
- Forced output stop function start 1 trigger register n (TKBPAHFSn)
- Forced output stop function cancel 1 trigger register n (TKBPAHFTn)

10.6.3.1 Forced output stop function control registers n0, n1 (TKBPACTLn0, TKBPACTLn1)

TKBPACTLnp is a register that selects the signal to be used as the trigger to control the forced output stop function of the TKBOp pin, and to select the pin for setting forced output stop mode.

TKBPACTLnp can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 10 - 66 Format of Forced Output Stop Function Control Register 0p (TKBPACTL0p) (1/2)

Address: F0430H (TKBPACTL00), F0432H (TKBPACTL01) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
TKBPACTL0p	TKBPAFXS0p3	TKBPAFXS0p2	TKBPAFXS0p1	TKBPAFXS0p0	0	0	0	TKBPAFCM0p
	7	6	5	4	3	2	1	0
	0	0	TKBPAHZS0p1	TKBPAHZS0p0	TKBPAHCM0p1	TKBPAHCM0p0	TKBPAMD0p1	TKBPAMD0p0

TKBPAFXS0p3	External interruption trigger selection for forced output stop function 2
0	INTP11 can not be used as a trigger.
1	INTP11 can be used as a trigger.[Note 1]

TKBPAFXS0p2	External interruption trigger selection for forced output stop function 2
0	INTP10 can not be used as a trigger.
1	INTP10 can be used as a trigger.[Note 1]

TKBPAFXS0p1	Comparator trigger selection for forced output stop function 2
0	Comparator 1 can not be used as a trigger.
1	Comparator 1 can be used as a trigger.[Note 3]

TKBPAFXS0p0	Comparator trigger selection for forced output stop function 2
0	Comparator 0 can not be used as a trigger.
1	Comparator 0 can be used as a trigger.[Note 3]

TKBPAFCM0p	Operation mode selection for forced output stop function 2
0	Forced output stop function 2 starts with trigger input, and forced output stop function 2 is cleared at the next counter period.[Note 2]
1	Forced output stop function 2 starts with trigger input, and forced output stop function 2 is cleared at the next counter period following detection of the reverse edge of the trigger.[Note 2]

TKBPAHZS0p1	Comparator trigger selection for forced output stop function 1
0	Comparator 1 can not be used as a trigger.
1	Comparator 1 can be used as a trigger.[Note 3]

TKBPAHZS0p0	Comparator trigger selection for forced output stop function 1
0	Comparator 0 can not be used as a trigger.
1	Comparator 0 can be used as a trigger.[Note 3]

Figure 10 - 66 Format of Forced Output Stop Function Control Register 0p (TKBPACTL0p) (2/2)

TKBPAHCM0p1	TKBPAHCM0p0	Clear condition selection for forced output stop function 1
0	0	Forced output stop function 1 starts with trigger input, and forced output stop function 1 is cleared when forced output stop function release trigger (TKBPAHTT0p) = 1 is written, regardless of the trigger signal level.
0	1	Forced output stop function 1 starts with trigger input, and when the trigger signal is in its active period, writing “forced output stop function release trigger (TKBPAHTT0p) = 1” is invalid. Forced output stop function 1 is cleared when forced output stop function release trigger (TKBPAHTT0p) = 1 is written while the trigger signal is in its inactive period.
1	0	Forced output stop function 1 starts with trigger input, and forced output stop function 1 is cleared at the next counter period after forced output stop function release trigger (TKBPAHTT0p) = 1 is written, regardless of the trigger signal level. Note 2
1	1	Forced output stop function 1 starts with trigger input, and when the trigger signal is in its active period, writing “forced output stop function release trigger (TKBPAHTT0p) = 1” is invalid. Forced output stop function 1 is cleared at the next counter period after forced output stop function release trigger (TKBPAHTT0p) = 1 is written when the trigger signal is in its inactive period. Note 2

TKBPAMD0p1	TKBPAMD0p0	Output status selection when executing forced output stop function	
		Forced output stop function 1	Forced output stop function 2
0	0	Hi-Z output	Output fixed at low level
0	1	Hi-Z output	Output fixed at high level
1	0	Output fixed at low level	Output fixed at low level
1	1	Output fixed at high level	Output fixed at high level

Note 1. When INTP10 and INTP11 is used as the forced output stop function 2, see **19.5 Caution for Using Timer KB Simultaneous Operation Function**.

Note 2. When timer KB is stopped (TKBCEn = 0) without waiting for the next counter period, the forced output stop function is kept on until timer KB is restarted (TKBCEn = 1).

Note 3. When comparator i is used as the forced output stop function 1 and 2, set the CiOE bit of the comparator output control register (COMPOCR) to 1.

Caution 1. During timer operation, setting the other bits of the TKBPACTL0p register is prohibited. However, the TKBPACTL0p register can be refreshed (the same value is written).

Caution 2. Be sure to clear bits 11 to 9, 7 and 6 to “0”.

Remark i = 0, 1, n = 0, p = 0, 1

10.6.3.2 Forced output stop function control register n2 (TKBPACTLn2)

TKBPACTLn2 is a register that enables or disables the forced output stop function of the TKBOp pin.

TKBPACTLn2 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10 - 67 Format of Forced Output Stop Function Control Register n2 (TKBPACTLn2)

Address: F0437H After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
TKBPACTLn2	0	0	0	0	0	0	TKBPACEn1	TKBPACEn0

TKBPACEnp	Input control of trigger signal used for forced output stop function of the TKBOp pin.
0	Disable operation of forced output stop function
1	Enable operation of forced output stop function

Caution 1. The TKBPACTLn2 register can be overwritten while the timer is operating.

Caution 2. Be sure to clear bits 7 to 2 to “0”.

Remark n = 0, p = 0, 1

10.6.3.3 Forced output stop function flag register (TKBPAFLGn)

TKBPAFLGn is a register with status flags for forced output stop function of the TKBOp pin.

TKBPAFLGn can be read by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10 - 68 Format of Forced Output Stop Function Flag Register (TKBPAFLGn)

Address: F0436H After reset: 00H R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
TKBPAFLGn	TKBPAFSFn1	TKBPAHSFn1	TKBPAFSFn0	TKBPAHSFn0	TKBPAFIFn1	TKBPAHIFn1	TKBPAFIFn0	TKBPAHIFn0

TKBPAFSFnp	Status flag of forced output stop function 2 for TKBOp pin
0	Forced output stop function clear status
1	Forced output stop function status

TKBPAHSFnp	Status flag of forced output stop function 1 for TKBOp pin
0	Forced output stop function clear status
1	Forced output stop function status

TKBPAFIFnp	Input monitor bit of forced output stop function 2 for TKBOp pin
0	Forced output stop function 2 trigger signal is at low level (inactive)
1	Forced output stop function 2 trigger signal is at high level (active)

TKBPAHIFnp	Input monitor bit of forced output stop function 1 for TKBOp pin
0	Forced output stop function 1 trigger signal is at low level (inactive)
1	Forced output stop function 1 trigger signal is at high level (active)

Caution **The timing to cancel the forced output stop function 1 depends on the setting. For details, see 10.7.2 Software cancel operation for forced output stop function 1.**

Remark n = 0, p = 0, 1

10.6.3.4 Forced output stop function 1 start trigger register n (TKBPAHFSn)

TKBPAHFSn is the start trigger register used by forced output stop function 1 of the TKBOp pin.

TKBPAHFSn can be written by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10 - 69 Format of Forced Output Stop Function 1 Start Trigger Register n (TKBPAHFSn)

Address: F0434H After reset: 00H W

Symbol	7	6	5	4	3	2	<1>	<0>
TKBPAHFSn	0	0	0	0	0	0	TKBPAHTSn1	TKBPAHTSn0

TKBPAHTSnp	Start trigger of forced output stop function 1 for TKBOp output
0	Invalid setting
1	Starts forced output stop function 1 for TKBOp output

Caution 1. The TKBPAHFSn register can be overwritten while the timer is operating.

Caution 2. Be sure to clear bits 7 to 2 to “0”.

Caution 3. When TKBPAHFSn register is read, 0 is read.

Remark n = 0, p = 0, 1

10.6.3.5 Forced output stop function cancel 1 trigger register n (TKBPAHFTn)

TKBPAHFTn is the cancel trigger register used by forced output stop function 1 of the TKBOp pin.

TKBPAHFTn can be written by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10 - 70 Format of Forced Output Stop Function Cancel 1 Trigger Register n (TKBPAHFTn)

Address:F0435H After reset: 00H W

Symbol	7	6	5	4	3	2	<1>	<0>
TKBPAHFTn	0	0	0	0	0	0	TKBPAHTTn1	TKBPAHTTn0

TKBPAHTTnp	Start trigger of forced output stop function 1 for TKBOp output
0	Invalid setting
1	Starts forced output stop function 1 for TKBOp output

Caution 1. The TKBPAHFTn register can be overwritten while the timer is operating.

Caution 2. Be sure to clear bits 7 to 2 to “0”.

Caution 3. When TKBPAHFSn register is read, 0 is read.

Caution 4. The timing to cancel the forced output stop function 1 depends on the setting. For details, see 10.7.2 Software cancel operation for forced output stop function 1.

Remark n = 0, p = 0, 1

10.7 Operation of Forced Output Stop Function 1

Timer output can be fixed to Hi-Z, high, or low level directly (not via the CPU) and asynchronously with the operation clock fKBKC of the 16-bit timer KBn circuit when a trigger source occurs (comparator 0 and 1 output). The forced output stop status is canceled synchronously with the operation clock fKBKC of the 16-bit timer KBn circuits by setting the stop trigger of forced output stop function 1.

10.7.1 Summary for forced output stop function 1

In this function, comparator output signal and software trigger is used as trigger signal for forced output stop function 1.

The output level selectable at forced output stop is controlled byTKBPAMDnp0, TKBPAMDnp1 bit of TKBPACTLnp register.

The following table shows the relationship of forced output stop function 1 of output p terminal (TKBOp) for timer KBn.

Table 10 - 7 The Relationship of Forced Output Stop Function 1 of the TKBOp Pin

TKBPAMDnp1	TKBPAMDnp0	Output Level Selection at Forced Output Stop Function 1 Execution
0	0	Hi-Z Output
0	1	Hi-Z Output
1	0	Low level fixed output
1	1	High-level fixed output

The selection for comparator output being used is controlled by TKBPAHZSnp2 to TKBPAHZSnp0 bit for forced output stop function control register np (TKBPACTLnp).

The following table shows trigger selection for forced output stop function 1 of output p terminal (TKBOp) for timer KBn.

Table 10 - 8 The Trigger Selection for Forced Output Stop Function 1 of the TKBOp Pin

Bit	Selectable Trigger Signals
TKBPAHZSnp0	Comparator 0
TKBPAHZSnp1	Comparator 1

Remark n = 0, p = 0, 1

10.7.2 Software cancel operation for forced output stop function 1

The table below shows the start trigger (TKBPAHTSnp bit for TKBPAHFSn register) setting to start forced output stop function 1.

Table 10 - 9 Operation of Start Trigger (TKBPAHTSnp Bit) of Forced Output Stop Function 1

TKBPAHTSnp	Start of Forced Output Stop Function by Software
0	Invalid setting
1	Writing “1” initiates the fixed output control of high-impedance/low-level/high-level for TKBOp output (the same function with rising edge detection of trigger signal by forced output stop function 1).

The table below shows the cancel trigger (TKBPAHTTnp bit for TKBPAHFTn register) setting to cancel forced output stop function 1.

Table 10 - 10 Operation of Cancel Trigger (TKBPAHTTnp Bit) at Forced Output Stop Function 1

TKBPACTLnp Register		Cancel of Forced Output Stop Function 1 by Software
TKBPAHCMnp1	TKBPAHCMnp0	
0	0	Forced output stop function 1 starts with trigger input, and forced output stop function 1 is cleared when forced output stop function release trigger (TKBPAHTTnp) = 1 is written, regardless of the trigger signal level.
0	1	Forced output stop function 1 starts with trigger input, and when the trigger signal is in its active period, writing “forced output stop function release trigger (TKBPAHTTnp) = 1” is invalid. Forced output stop function 1 is cleared when forced output stop function release trigger (TKBPAHTTnp) = 1 is written while the trigger signal is in its inactive period.
1	0	Forced output stop function 1 starts with trigger input, and forced output stop function 1 is cleared at the next counter period after forced output stop function release trigger (TKBPAHTTnp) = 1 is written, regardless of the trigger signal level. Note
1	1	Forced output stop function 1 starts with trigger input, and when the trigger signal is in its active period, writing “forced output stop function release trigger (TKBPAHTTnp) = 1” is invalid. Forced output stop function 1 is cleared at the next counter period after forced output stop function release trigger (TKBPAHTTnp) = 1 is written when the trigger signal is in its inactive period. Note

Note When timer KB is stopped (TKBCEn = 0) without waiting for the next counter period, the forced output stop function is kept on until timer KB is restarted (TKBCEn = 1).

Remark n = 0, p = 0, 1

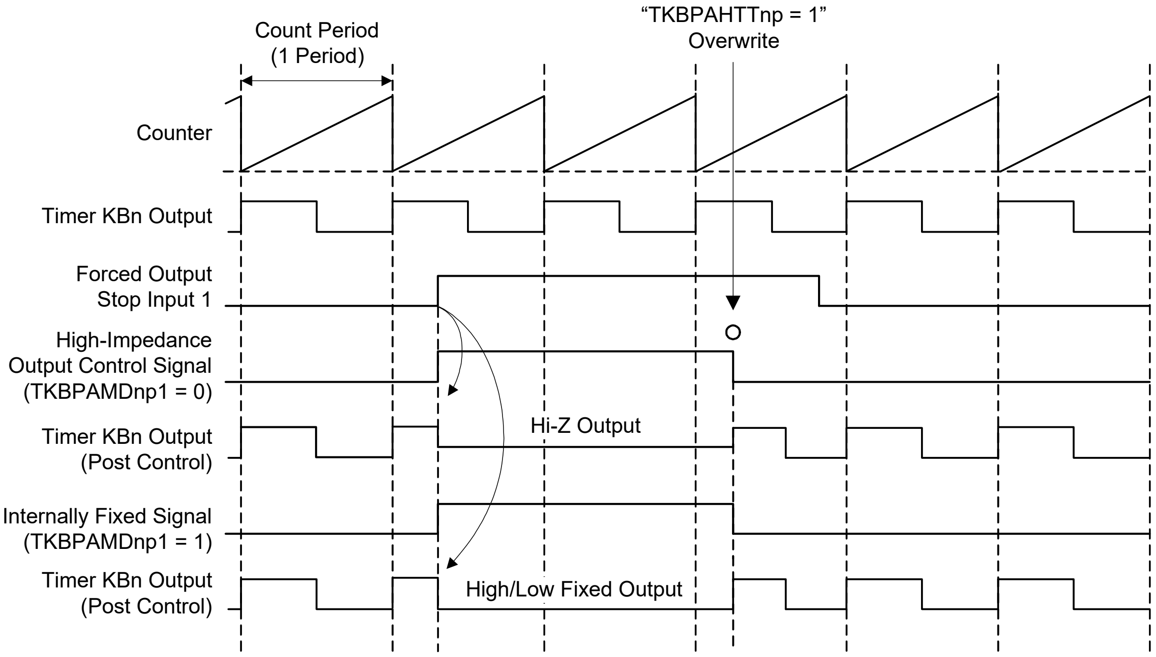
10.7.3 Basic operation of forced output stop function 1

This shows the operations of forced output function 1 with different setting of TKBPAHCMnp1 and TKBPAHCMnp0 registers.

The trigger signal that initiates the forced output stop function 1 (forced output stop input 1) is an OR output of the trigger signal selected by TKBPAHZSnp0 to TKBPAHZSnp2 bits of forced output stop function control register np (TKBPACTLnp) and TKBPAHTSnp bit of the trigger register n (TKBPAHFSn) that initiates forced output stop function 1.

(1) Forced output stop function 1 at TKBPAHCMnp1, TKBPAHCMnp0 = 0, 0

Figure 10 - 71 Forced Output Stop Function at TKBPAHCMnp1, TKBPAHCMnp0 = 0, 0



(a) TKBPAMDnp1 = 0 (Hi-Z output)

High-impedance output is realized via the detection of rising edge of forced output stop input 1.

Regardless of input level of forced output stop input 1, it returns to timer output via writing of "1" to cancel trigger (TKBPAHTTnp bit).

High-level period of high-impedance output control signal is the period for forced output stop 1 (high-impedance output).

(b) TKBPAMDnp1 = 1 (fixed output)

Fixed output is realized in low-level/high-level according to TKBPAMDnp0 setting at the detection of rising edge of forced output stop input 1.

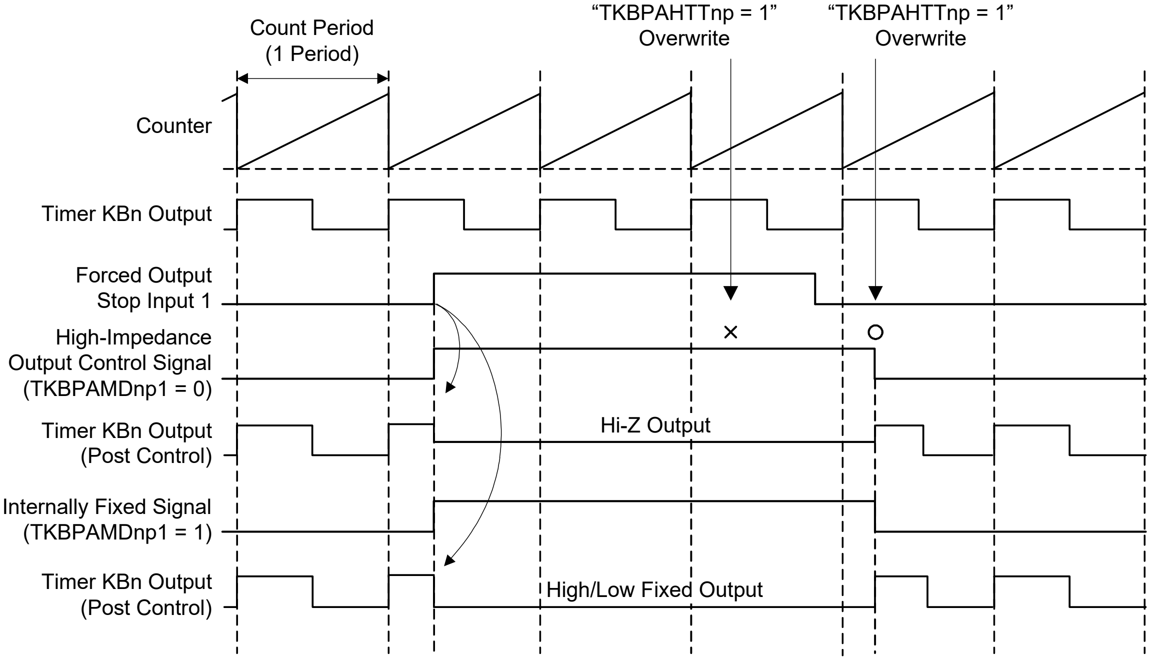
Regardless of input level of forced output stop input 1, output level fixing is cancelled and returned to timer output via writing of "1" to cancel trigger (TKBPAHTTnp bit).

High-level period of internal fixed signal is the period for forced output stop 1 (low-level/high-level output fixing).

Remark n = 0, p = 0, 1

(2) Forced output stop function 1 at TKBPAHCMnp1, TKBPAHCMnp0 = 0, 1

Figure 10 - 72 Forced Output Stop Function 1 at TKBPAHCMnp1, TKBPAHCMnp0 = 0, 1



(a) TKBPAMDnp1 = 0 (Hi-Z output)

High-impedance output is realized via the detection of rising edge of forced output stop input 1.

During the active level (high-level) period of forced output stop input 1, writing "1" to cancel trigger (TKBPAHTTnp bit) is invalid.

After forced output stop input 1 turned into inactive level (low-level), it returns to timer output via writing of "1" to cancel trigger (TKBPAHTTnp bit).

High-level period of high-impedance output control signal is the period for forced output stop 1 (high-impedance output).

(b) TKBPAMDnp1 = 1 (fixed output)

Fixed output is realized in low-level/high-level according to TKBPAMDnp0 setting at the detection of rising edge of forced output stop input 1.

During the active level (high-level) period of forced output stop input 1, writing "1" to cancel trigger (TKBPAHTTnp bit) is invalid.

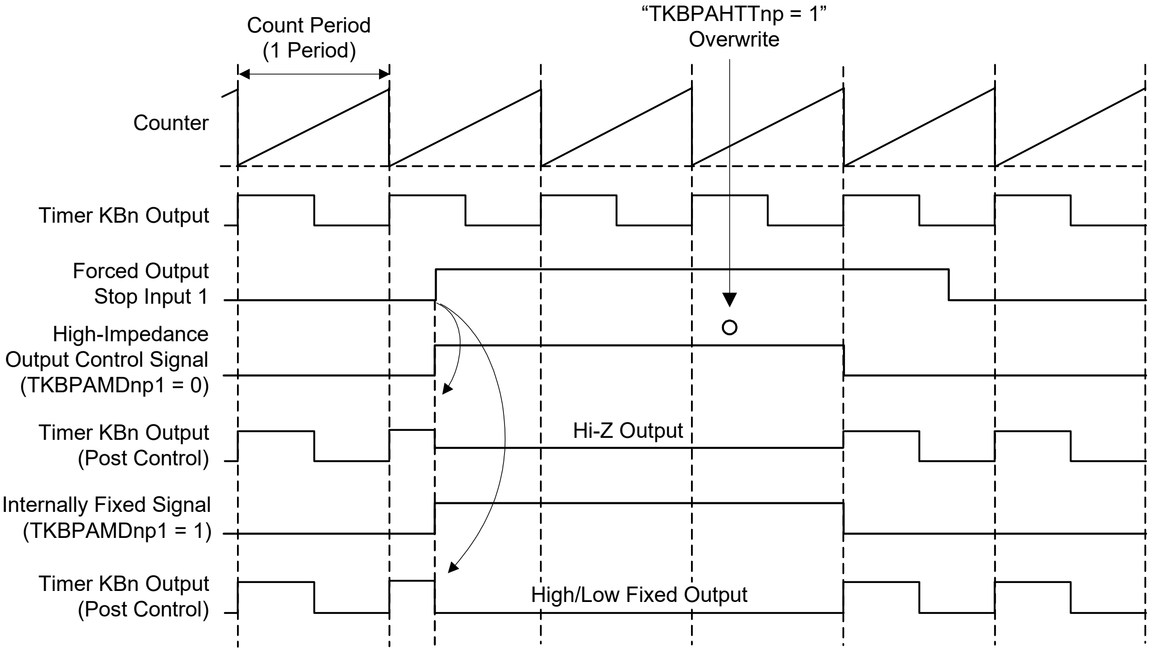
After the forced output stop input 1 turned into inactive level (low-level), output level fixing is cancelled and returned to timer output via writing of "1" to cancel trigger (TKBPAHTTnp bit).

High-level period of internal fixed signal is the period for forced output stop 1 (low-level/high-level output fixing).

Remark n = 0, p = 0, 1

(3) Forced output stop function 1 at TKBPAHCMnp1, TKBPAHCMnp0 = 1, 0

Figure 10 - 73 Forced Output Stop Function 1 at TKBPAHCMnp1, TKBPAHCMnp0 = 1, 0



(a) TKBPAMDnp1 = 0 (Hi-Z output)

High-impedance output is realized via the detection of rising edge of forced output stop input 1.

Regardless of the input level of forced output stop input 1, it returns to timer output after writing of "1" to cancel trigger (TKBPAHTTnp bit), in the next period.

High-level period of high-impedance output control signal is the period for forced output stop 1 (high-impedance output).

(b) TKBPAMDnp1 = 1 (fixed output)

Fixed output is realized in low-level/high-level according to TKBPAMDnp0 setting at the detection of rising edge of forced output stop input 1.

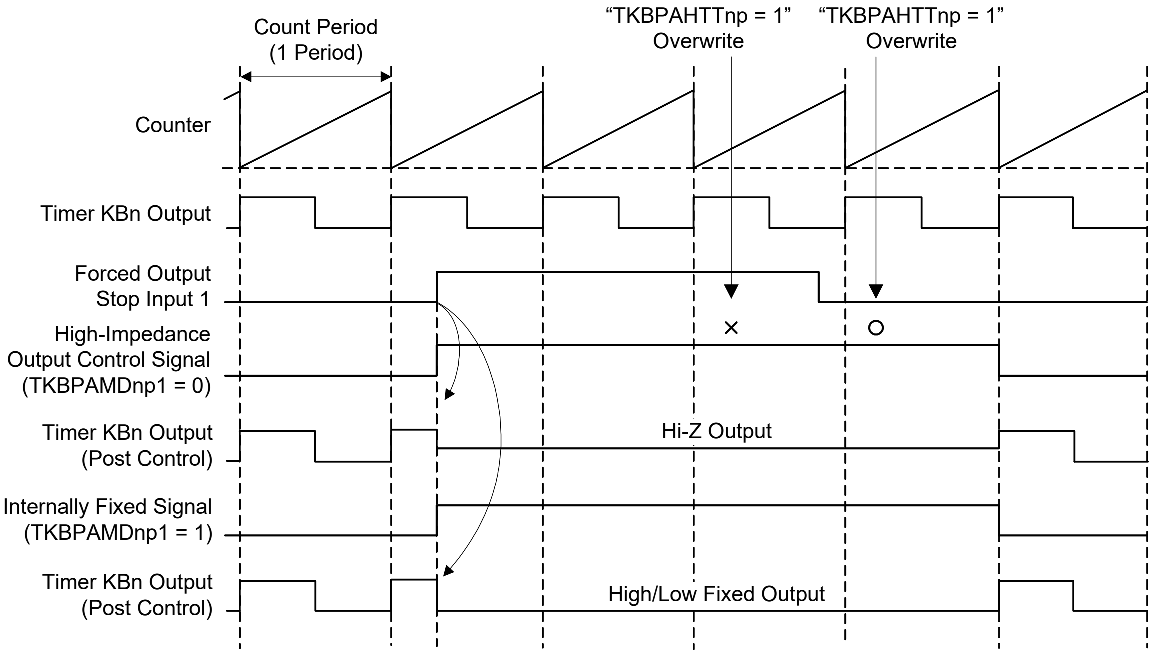
Regardless of input level of forced output stop input 1, output level fixing is cancelled and returned to timer output after writing of "1" to cancel trigger (TKBPAHTTnp bit) in the next counter period.

High-level period of internal fixed signal is the period for forced output stop 1 (low-level/high-level output fixing).

Remark n = 0, p = 0, 1

(4) Forced output stop function 1 at TKBPAHCMnp1, TKBPAHCMnp0 = 1, 1

Figure 10 - 74 Forced Output Stop Function 1 at TKBPAHCMnp1, TKBPAHCMnp0 = 1, 1



(a) TKBPAMDnp1 = 0 (Hi-Z output)

High-impedance output is realized via the detection of rising edge of forced output stop input 1.

During the active level (high-level) period of forced output stop input 1, writing "1" to cancel trigger (TKBPAHTTnp bit) is invalid.

It returns to timer output after writing of "1" to cancel trigger (TKBPAHTTnp bit) during the inactive level (low-level) of forced output stop input 1, in the next period.

High-level period of high-impedance output control signal is the period for forced output stop 1 (high-impedance output).

(b) TKBPAMDnp1 = 1 (fixed output)

Fixed output is realized in low-level/high-level according to TKBPAMDnp0 setting at the detection of rising edge of forced output stop input 1.

During the active level (high-level) period of forced output stop input 1, writing "1" to cancel trigger (TKBPAHTTnp bit) is invalid.

During the inactive level (low-level) period of forced output stop input 1, output level fixing is cancelled and returned to timer output in the next counter period after writing "1" to cancel trigger (TKBPAHTTnp bit).

High-level period of internal fixed signal is the period for forced output stop 1 (low-level/high-level output fixing).

Remark n = 0, p = 0, 1

10.8 Operation of Forced Output Stop Function 2

Timer output can be fixed to high or low level directly (not via the CPU) and asynchronously with the operation clock fKBKC of the 16-bit timer KBn circuit when a trigger source occurs (comparator 0 and 1 output, INTP10 and INTP11). The forced output stop status is canceled at the beginning of the next counter cycle after the trigger source occurs or after the trigger source signal changes to inactive level.

10.8.1 Summary for forced output stop function 2

In this function, comparator output signal and external interrupt (INTP10 and INTP11) used as trigger signal for forced output stop function 2.

The output level selectable at forced output stop is controlled byTKBPAMDnp0, TKBPAMDnp1 bit of TKBPACTLnp register.

The following table shows the relationship of forced output stop function 2 of output p terminal (TKBOp) for timer KBn.

Table 10 - 11 The Relationship of Forced Output Stop Function 2 of the TKBOp Pin

TKBPAMDnp1	TKBPAMDnp0	Output Level Selection at Forced Output Stop Function 2 Execution
0	0	Low level fixed output
0	1	High-level fixed output
1	0	Low level fixed output
1	1	High-level fixed output

The selection for comparator output being used is controlled by TKBPAFXSnp3 to TKBPAFXSnp0 bit for forced output stop function control register np (TKBPACTLnp).

The following table shows trigger selection for forced output stop function 2 of output p terminal (TKBOp) for timer KBn.

Table 10 - 12 The Trigger Selection for Forced Output Stop Function 2 of the TKBOp Pin

Bit	Selectable Trigger Signals
TKBPAFXSnp0	Comparator 0
TKBPAFXSnp1	Comparator 1
TKBPAFXSnp2	INTP10
TKBPAFXSnp3	INTP11

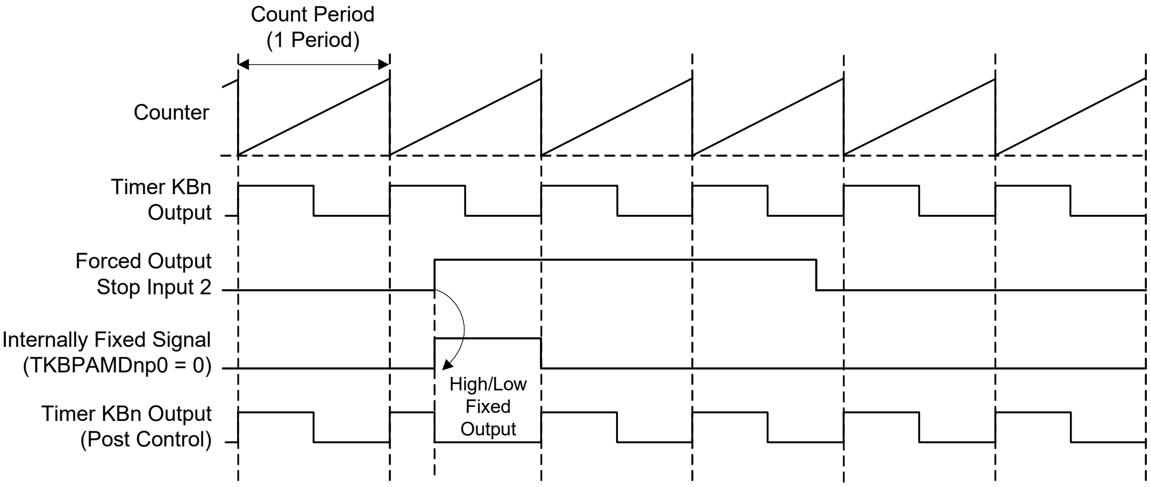
Remark n = 0, p = 0, 1

10.8.2 Basic operation of forced output stop function 2

This shows the operations of forced output function 2 with different setting of TKBPAFCMnp bits.
The trigger signal that initiates the forced output stop function 2 (forced output stop input 2) is the trigger signal selected by TKBPAFXSnp0 to TKBPAFXSnp3 bits of forced output stop function control register np.

(1) Forced output stop function 2 at TKBPAFCMnp = 0

Figure 10 - 75 Forced Output Stop Function 2 at TKBPAFCMnp = 0

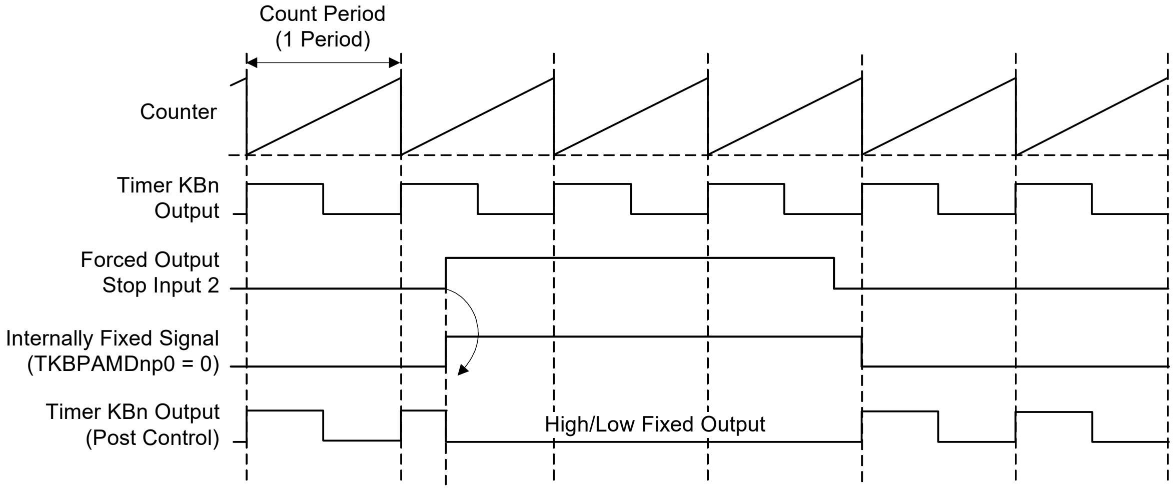


Fixed output is realized in low-level/high-level according to TKBPAMDnp0 setting at the detection of rising edge of forced output stop input 2.
Regardless of the input level of the forced output stop input 2, the fixing of output level is canceled at the next counter cycle and returned to timer output.
High-level period of internal fixed signal is the period for forced output stop 2 (low-level/high-level output fixing).

Remark n = 0, p = 0, 1

(2) Forced output stop function 2 at TKBPAFCMnp = 1

Figure 10 - 76 Forced Output Stop Function 2 at TKBPAFCMnp = 1



Fixed output is realized in low-level/high-level according to TKBPAMDnp0 setting at the detection of rising edge of forced output stop input 2.

After the forced output stop input 2 becomes reverse edge, the fixing of output level is canceled at the next counter cycle and returned to timer output.

High-level period of internal fixed signal is the period for forced output stop 2 (low-level/high-level output fixing).

Remark n = 0, p = 0, 1

CHAPTER 11 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

11.1 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.

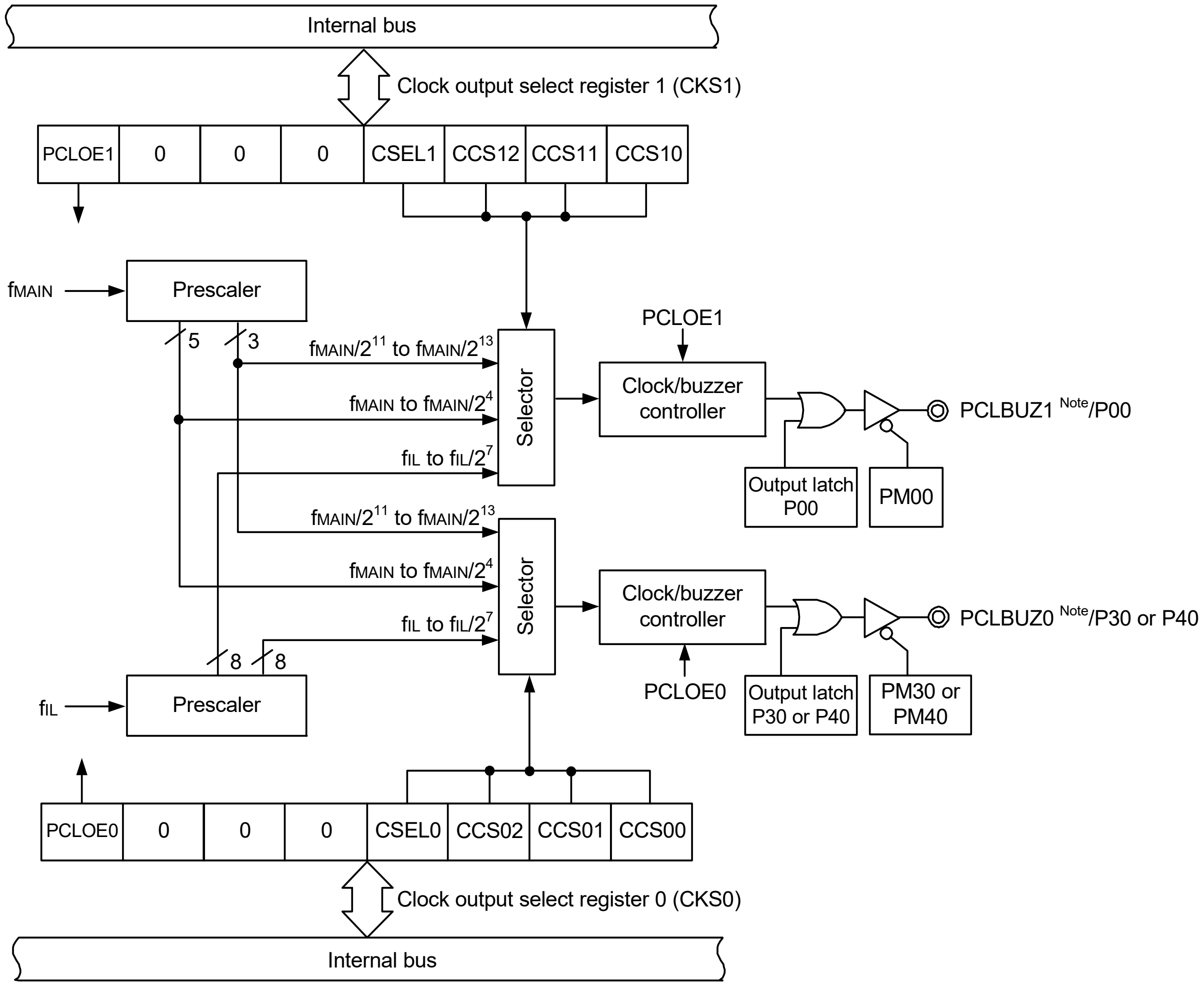
Two output pins, PCLBUZ0 and PCLBUZ1, are available.

The PCLBUZn pin outputs a clock selected by clock output select register n (CKSn).

Figure 11 - 1 shows the Block Diagram of Clock Output/Buzzer Output Controller.

Remark n = 0, 1

Figure 11 - 1 Block Diagram of Clock Output/Buzzer Output Controller



Note For output frequencies available from PCLBUZ0 and PCLBUZ1, refer to **35.4, 36.4 AC Characteristics**.

11.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

Table 11 - 1 Configuration of Clock Output/Buzzer Output Controller

Item	Configuration
Control registers	Clock output select registers n (CKSn) Operation speed mode control register (OSMC) Port mode registers 0, 3, 4 (PM0, PM3, PM4) Port registers 0, 3, 4 (P0, P3, P4) Port mode control registers 0, 3 (PMC0, PMC3) Peripheral I/O redirection register 0 (PIOR0)

11.3 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)
- Operation speed mode control register (OSMC)
- Port mode registers 0, 3, 4 (PM0, PM3, PM4)
- Port registers 0, 3, 4 (P0, P3, P4)
- Port mode control registers 0, 3 (PMC0, PMC3)
- Peripheral I/O redirection register 0 (PIOR0)

11.3.1 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 11 - 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	12 MHz
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_{IL}	15 kHz			
1	0	0	1	$f_{IL}/2$	7.5 kHz			
1	0	1	0	$f_{IL}/2^2$	3.75 kHz			
1	0	1	1	$f_{IL}/2^3$	1.875 kHz			
1	1	0	0	$f_{IL}/2^4$	938 Hz			
1	1	0	1	$f_{IL}/2^5$	469 Hz			
1	1	1	0	$f_{IL}/2^6$	234 Hz			
1	1	1	1	$f_{IL}/2^7$	117 Hz			

Note Use the output clock within a range of 16 MHz. See **35.4, 36.4 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_{IL}: Low-speed on-chip oscillator clock frequency

11.3.2 Registers that control port functions of clock output/buzzer output pins

Using the clock output/buzzer output requires setting of the registers that control the port functions for the port pins with which the clock output/buzzer output pin functions for the target channel are multiplexed (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)**.

Using a port pin which is multiplexed with a clock output/buzzer output pin function (e.g. P30/PCLBUZ0) for clock output/buzzer output requires setting the corresponding bits in the port mode register (PMxx), port mode control register (PMCxx), and port register (Pxx) to 0.

Example: When P30/PCLBUZ0 is to be used for clock output/buzzer 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.

11.4 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).

11.4.1 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.

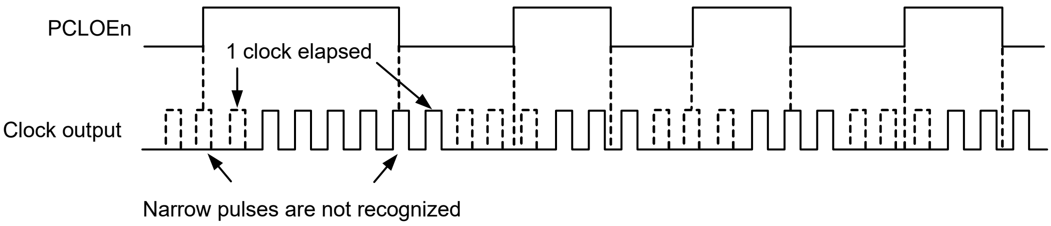
<2> Select the output frequency with bits 0 to 3 (CCSn0 to CCSn2, CSELn) of the clock output select register (CKSn) of the PCLBUZn pin (output in disabled status).

<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 11 - 3 shows enabling or stopping output using the PCLOEn bit and the timing of outputting the clock.

Remark 2. n = 0, 1

Figure 11 - 3 Timing of Outputting Clock from PCLBUZn Pin



11.5 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 12 WATCHDOG TIMER

12.1 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 25 RESET FUNCTION**.
When 75% of the overflow time + 1/2 f_{IL} is reached, an interval interrupt can be generated.

12.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 12 - 1 Configuration of Watchdog Timer

Item	Configuration
Counter	Internal counter (17 bits)
Control register	Watchdog timer enable register (WDTE)

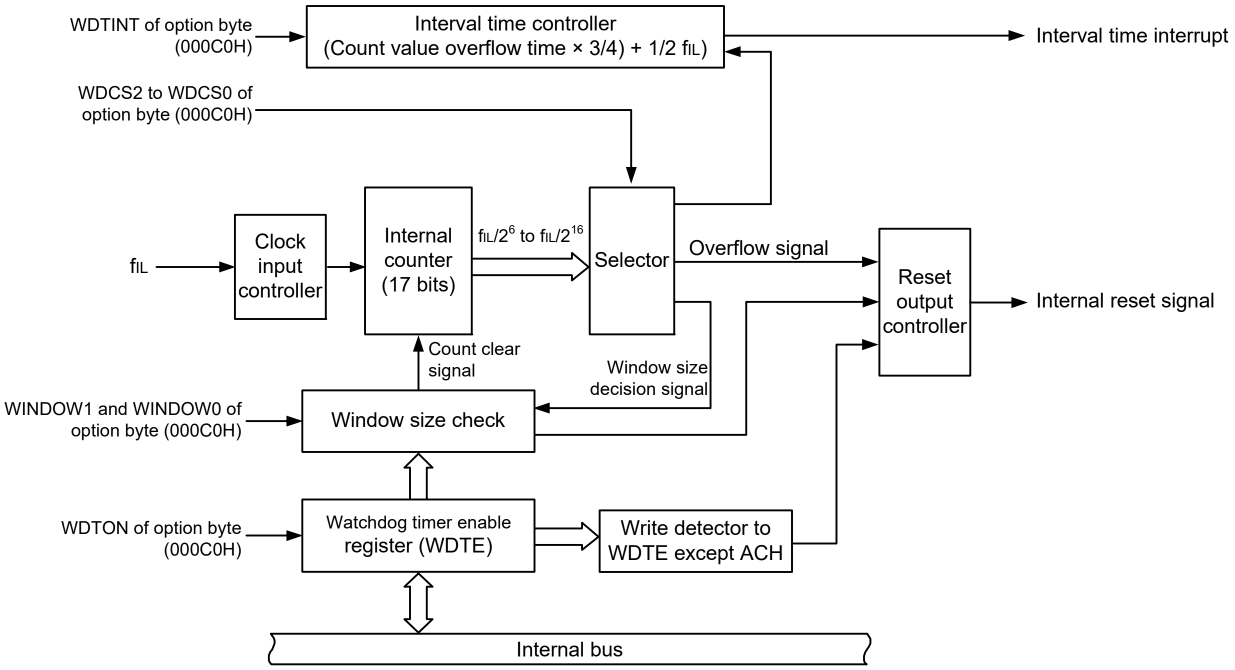
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the option byte.

Table 12 - 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 30 OPTION BYTE**.

Figure 12 - 1 Block Diagram of Watchdog Timer


WDTINT of option byte (000C0H)
Interval time controller (Count value overflow time × 3/4) + 1/2 fIL)
Interval time interrupt
WDCS2 to WDCS0 of option byte (000C0H)
fIL
Clock input controller
Internal counter (17 bits)
fIL/2^6 to fIL/2^16
Selector
Overflow signal
Reset output controller
Internal reset signal
Count clear signal
Window size decision signal
WINDOW1 and WINDOW0 of option byte (000C0H)
Window size check
WDTON of option byte (000C0H)
Watchdog timer enable register (WDTE)
Write detector to WDTE except ACH
Internal bus

Remark fIL: Low-speed on-chip oscillator clock

12.3 Register Controlling Watchdog Timer

The watchdog timer is controlled by the watchdog timer enable register (WDTE).

12.3.1 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 12 - 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)).

12.4 Operation of Watchdog Timer

12.4.1 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 30**).

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 **12.4.2** and **CHAPTER 30**).
- Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (000C0H) (for details, see **12.4.3** and **CHAPTER 30**).

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.

12.4.2 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 12 - 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

<R> **Note** Using the watchdog timer under the following conditions may lead to the generation of an interval interrupt (INTWDTI) after one cycle of the watchdog timer clock once the watchdog timer counter has been cleared.

Usage conditions that may lead to the generation of an interval interrupt:

- The overflow time of the watchdog timer is set to $2^{13}/f_{IL}$, $2^{14}/f_{IL}$, or $2^{16}/f_{IL}$.
- The interval interrupt is in use (the setting of the WDTINT bit of the relevant option byte is 1).
- ACH is written to the WDTE register (FFFABH) when the watchdog timer counter has reached or exceeded 75% of the overflow time.

This interrupt can be masked by clearing the watchdog timer counter through steps 1 to 5 below.

1. Set the WDTIMK bit of the interrupt mask flag register 0 (MK0L) to 1 before clearing the watchdog timer counter.
2. Clear the watchdog timer counter.
3. Wait for at least 80 μs.
4. Clear the WDTIIF bit of the interrupt request flag register (IF0L) to 0.
5. Clear the WDTIMK bit of the interrupt mask flag register 0 (MK0L) to 0.

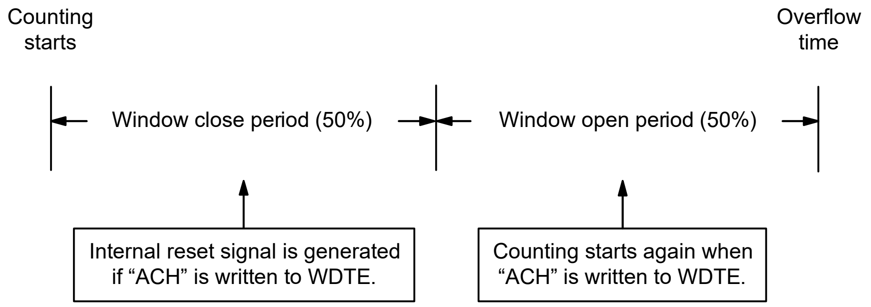
Remark f_{IL}: Low-speed on-chip oscillator clock frequency

12.4.3 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%


Counting starts
Overflow time
Window close period (50%)
Window open period (50%)
Internal reset signal is generated if “ACH” is written to WDTE.
Counting starts again when “ACH” is written to WDTE.

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 12 - 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

12.4.4 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 12 - 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 13 SERIAL ARRAY UNIT

The serial array unit has two serial channels. All channels can achieve UART, and only channel 0 can achieve Simplified SPI (CSI[Note]) and simplified I^2C.

Function assignment of each channel supported by the RL78/G11 is as shown below.

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.

- 10-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART
0	0	—	—
	1		
	2	CSI10	UART1
	3	—	

- 16-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	—	UART0 (LIN-bus supported)	—
	1			
	2	CSI10	UART1	—
	3	CSI11		IIC11

- 20-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 (LIN-bus supported)	IIC00
	1	—		—
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11

- 24, 25-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 (LIN-bus supported)	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11

When “UART0” is used for channels 0 and 1, CSI00 and CSI01 cannot be used, but CSI10, UART1, or IIC10 of channels 2 or 3 can be used.

Caution Most of the following descriptions in this chapter use the unit and channel configuration of the 24-pin products as an example.

13.1 Functions of Serial Array Unit

Each serial interface supported by the RL78/G11 has the following features.

13.1.1 Simplified SPI (CSI00, CSI01, CSI10, CSI11)

Data is transmitted or received in synchronization with the serial clock (SCK) output from the master channel. Simplified SPI communication is clocked communication performed by using three communication lines: one for the serial clock (SCK), one for transmitting serial data (SO), one for receiving serial data (SI).
For details about the settings, see **13.5 Operation of Simplified SPI (CSI00, CSI01, CSI10, CSI11) 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

Simplified SPIs (CSIs) of the following channels 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 SPIs (CSIs) can be specified.
- CSI00

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics. For details, see **CHAPTER 35**, **CHAPTER 36 ELECTRICAL SPECIFICATIONS**.

13.1.2 UART (UART0, UART1)

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel).

The LIN-bus can be implemented by using timer array unit with an external interrupt (INTP0).

For details about the settings, see **13.7 Operation of UART (UART0, UART1) Communication**.

[Data transmission/reception]

- Data length of 7, 8, or 9 bits[Note]
- 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

The LIN-bus is accepted in UART0 (0 and 1 channels).

[LIN-bus functions]

• Wakeup signal detection	Using the external interrupt (INTP0) and timer array unit
• Break field (BF) detection	
• Sync field measurement, baud rate calculation	

Note Only the following UART can be specified for the 9-bit data length.

- UART0

13.1.3 Simplified I^{2}C (IIC00, IIC01, IIC10, IIC11)

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This simplified I^{2}C 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 **13.9 Operation of Simplified I^{2}C (IIC00, IIC01, IIC10, IIC11) Communication**.

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function Note and ACK detection function
- Data length of 8 bits (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Manual generation of start condition and stop condition

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- ACK error or overrun error

* [Functions not supported by simplified I^{2}C]

- Slave transmission, slave reception
- Arbitration loss detection function
- Clock stretch detection functions

Note When receiving the last data, ACK will not be output if 0 is written to the SOEmn bit (serial output enable register m (SOEm)) and serial communication data output is stopped. See the processing flow in **13.9.3 (2)** for details.

Remark To use an I^{2}C bus of full function, see **CHAPTER 14 SERIAL INTERFACE IICA**.

13.2 Configuration of Serial Array Unit

The serial array unit includes the following hardware.

Table 13 - 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, SCK10, SCK11 pins (for Simplified SPI) and SCL00, SCL01, SCL10, SCL11 pins (for simplified I^{2}C)
Serial data input	SI00, SI01, SI10, SI11 pins (for Simplified SPI), RxD0 pin (for LIN-bus supported UART), RxD1 (for UART)
Serial data output	SO00, SO01, SO10, SO11 pins (for Simplified SPI), TxD0 pin (for LIN-bus supported UART), TxD1 pin (for UART)
Serial data I/O	SDA00, SDA01, SDA10, SDA11 pins (for simplified I^{2}C)
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 registers 0, 3 to 5 (PIM0, PIM3 to PIM5) • Port output mode registers 0, 3 to 5 (POM0, POM3 to POM5) • Port mode registers 0, 2 to 5 (PM0, PM2 to PM5) • Port registers 0, 2 to 5 (P0, P2 to P5)

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

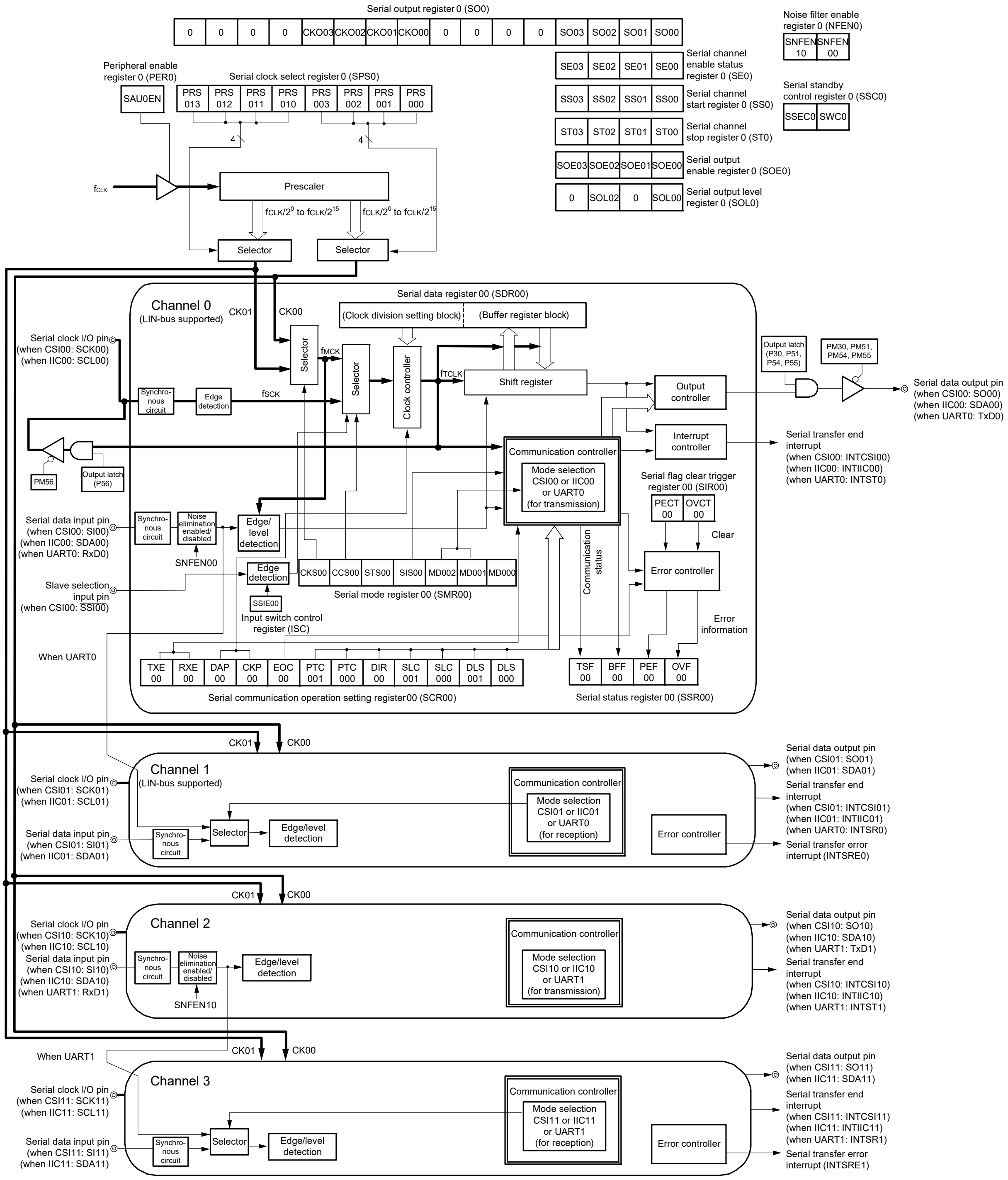
Note The lower 8 bits of serial data register mn (SDRmn) can be read or written as the following SFRs, depending on the communication mode.

- 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 to 3), p: CSI number (p = 00, 01, 10, 11), q: UART number (q = 0, 1), r: IIC number (r = 00, 01, 10, 11)

Figure 13 - 1 shows the Block Diagram of Serial Array Unit 0.

Figure 13 - 1 Block Diagram of Serial Array Unit 0



13.2.1 Shift register

This is a 9-bit register that converts parallel data into serial data or vice versa.

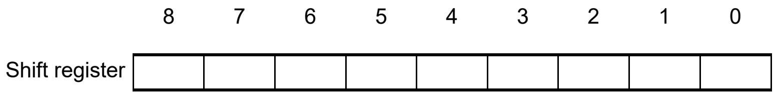
In case of the UART communication of nine bits of data, nine bits (bits 0 to 8) are used [Note].

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).



Note Only the following UARTs can be specified for the 9-bit data length.
- UART0

13.2.2 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

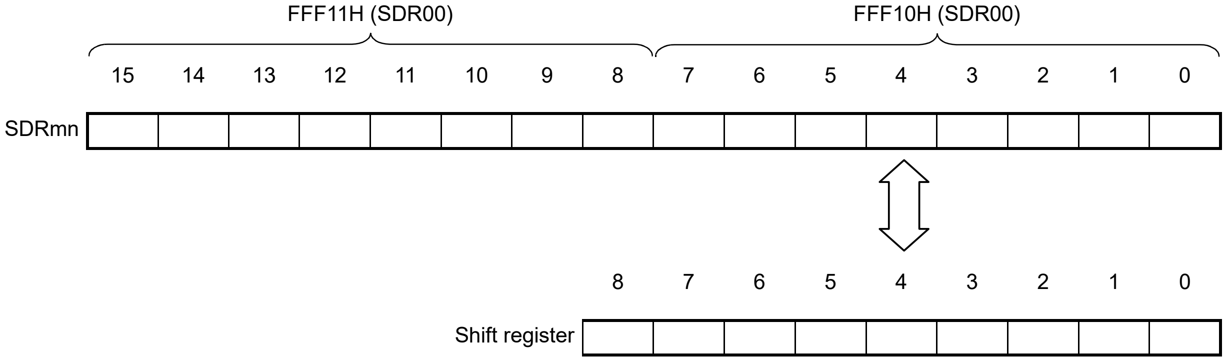
Note 2. When operation is stopped (SEmn = 0), do not rewrite SDRmn[7:0] by an 8-bit memory manipulation instruction (SDRmn[15:9] are all cleared to 0).

Remark 1. After data is received, 0 is stored in bits 0 to 8 in bit portions that exceed the data length.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11),
q: UART number (q = 0, 1), r: IIC number (r = 00, 01, 10, 11)

Figure 13 - 2 Format of Serial data register mn (SDRmn) (mn = 00, 01)

Address: FFF10H, FFF11H (SDR00)Note 1, FFF12H, FFF13H (SDR01)Note 2 After reset: 0000H R/W



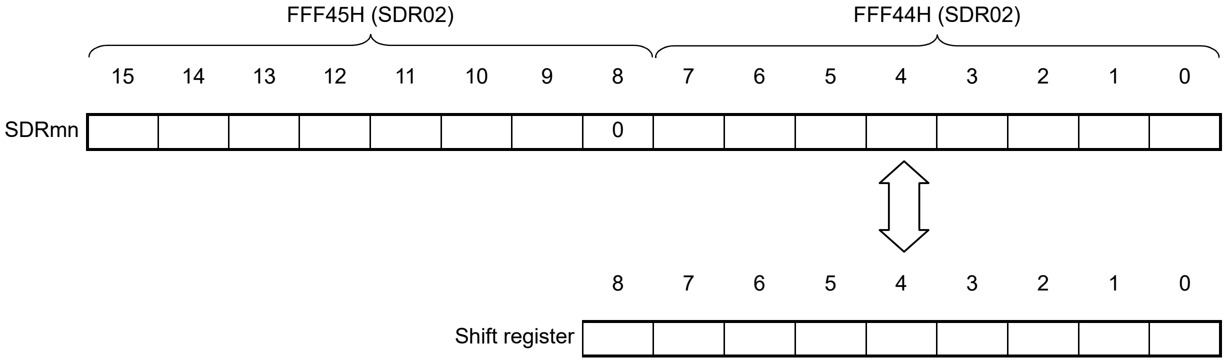
Note 1. 20, 24, 25-pin products

Note 2. 24, 25-pin products

Remark For the function of the higher 7 bits of the SDRmn register, see **13.3 Registers Controlling Serial Array Unit**.

Figure 13 - 3 Format of Serial data register mn (SDRmn) (mn = 02, 03)

Address: FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03)Note After reset: 0000H R/W



Note 16, 20, 24, 25-pin products

Caution **Be sure to clear bit 8 to "0".**

Remark For the function of the higher 7 bits of the SDRmn register, see **13.3 Registers Controlling Serial Array Unit**.

13.3 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 registers 0, 3 to 5 (PIM0, PIM3 to PIM5)
- Port output mode registers 0, 3 to 5 (POM0, POM3 to POM5)
- Port mode registers 0, 2 to 5 (PM0, PM2 to PM5)
- Port registers 0, 2 to 5 (P0, P2 to P5)

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

13.3.1 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 13 - 4 Format of Peripheral enable register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	3	<2>	1	<0>
PER0	0	IICA1EN	ADCEN	IICA0EN	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 registers 0, 3 to 5 (PIM0, PIM3 to PIM5), port output mode registers 0, 3 to 5 (POM, POM3 to POM5), port mode registers 0, 2 to 5 (PM0, PM2 to PM5), and port registers 0, 2 to 5 (P0, P2 to P5).

- **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, and 7

13.3.2 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 13 - 5 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	IICA1RES	ADCRES	IICA0RES	0	SAU0RES	0	TAU0RES

SAU0RES	Reset control of serial array unit
0	Serial array unit is released from the reset state.
1	Serial array unit is in the reset state.

13.3.3 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 13 - 6 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	Selection 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

13.3.4 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 I^2C), and an interrupt source. This register is also used to invert the level of the receive data only in the UART mode.
Rewriting the SMRmn register is prohibited when the register is in operation (when SEmn = 1). However, the MDmn0 bit can be rewritten during operation.
The SMRmn register can be set by a 16-bit memory manipulation instruction.
Reset signal generation sets the SMRmn register to 0020H.

Figure 13 - 7 Format of Serial mode register mn (SMRmn) (1/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03), 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

CKS mn	Selection of operation clock (f_{MCK}) of channel n
0	Operation clock CKm0 set by the SPSm register
1	Operation clock CKm1 set by the SPSm register
Operation clock (f_{MCK}) is used by the edge detector. In addition, depending on the setting of the CCSmn bit and the higher 7 bits of the SDRmn register, a transfer clock (f_{TCLK}) is generated.	

CCS mn	Selection of transfer clock (f_{TCLK}) of channel n
0	Divided operation clock f_{MCK} specified by the CKSmn bit
1	Clock input f_{SCK} from the 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.	

STS mn Note	Selection of start trigger source
0	Only software trigger is valid (selected for simplified SPI (CSI), UART transmission, and simplified I^2C).
1	Valid edge of the RxDq pin (selected for UART reception)
Transfer is started when the above source is satisfied after 1 is set to the SSm register.	

Note The SMR01, SMR03 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 and SMR02 registers) to 0. Be sure to set bit 5 to 1.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11), q: UART number (q = 0, 1), r: IIC number (r = 00, 01, 10, 11)

Figure 13 - 7 Format of Serial mode register mn (SMRmn) (2/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03) 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^{2}C 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, SMR03 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 and SMR02 registers) to 0. Be sure to set bit 5 to 1.**

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11), q: UART number (q = 0, 1), r: IIC number (r = 00, 01, 10, 11)

13.3.5 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.

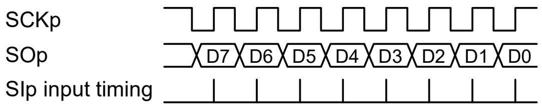
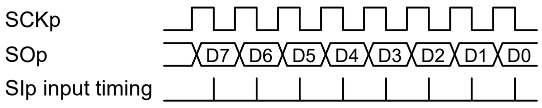
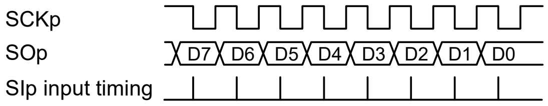
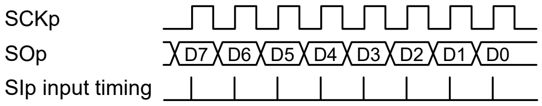
Reset signal generation sets the SCRmn register to 0087H.

Figure 13 - 8 Format of Serial communication operation setting register mn (SCRmn) (1/2)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03) 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

TXE mn	RXE mn	Setting of operation mode of channel n
0	0	Disable communication.
0	1	Reception only
1	0	Transmission only
1	1	Transmission/reception

DAP mn	CKP mn	Selection of data and clock phase in Simplified SPI (CSI) mode	Type
0	0	 SCKp SOp SIp input timing D7 D6 D5 D4 D3 D2 D1 D0	1
0	1	 SCKp SOp SIp input timing D7 D6 D5 D4 D3 D2 D1 D0	2
1	0	 SCKp SOp SIp input timing D7 D6 D5 D4 D3 D2 D1 D0	3
1	1	 SCKp SOp SIp input timing D7 D6 D5 D4 D3 D2 D1 D0	4
Be sure to set DAPmn, CKPmn = 0, 0 in the UART mode and simplified I²C mode.			

EOC mn	Mask control of error interrupt signal (INTSREx (x = 0, 1))
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 I²C mode, and during UART transmission Note 2.	

Note 1. The SCR00, SCR02 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 and SCR03 registers to 0). Be sure to set bit 2 to 1.**

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11)

Figure 13 - 8 Format of Serial communication operation setting register mn (SCRmn) (2/2)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03) 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^2C 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^2C 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, 02 only)
1	1	Setting prohibited
When the transfer end interrupt is selected, the interrupt is generated when all stop bits have been completely transferred. Set 1 bit (SLCmn1, SLCmn0 = 0, 1) during UART reception and in the simplified I^2C mode. Set no stop bit (SLCmn1, SLCmn0 = 0, 0) in the 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^2C mode.		

Note 1. The SCR00, SCR02 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 and SCR03 registers to 0). Be sure to set bit 2 to 1.**

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11)

13.3.6 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 13 - 9 Format of Serial data register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00)Note 1, FFF12H, FFF13H (SDR01)Note 2 After reset: 0000H R/W

	FFF11H (SDR00)								FFF10H (SDR00)							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn																

Address: FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03)Note 3 After reset: 0000H R/W

	FFF45H (SDR02)								FFF44H (SDR02)							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn								0								

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	$f_{MCK}/256$

Note 1. 20, 24, 25-pin products

Note 2. 24, 25-pin products

Note 3. 16, 20, 24, 25-pin products

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 I^{2}C is used. Set SDRmn[15:9] to 0000001B or greater.

Caution 3. When operation is stopped (SEmn = 0), do not rewrite SDRmn[7:0] by an 8-bit memory manipulation instruction (SDRmn[15:9] are all cleared to 0).

Remark 1. For the function of the lower 8/9 bits of the SDRmn register, see **13.2 Configuration of Serial Array Unit**.
Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

13.3.7 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 13 - 10 Format of Serial flag clear trigger register mn (SIRmn)

Address: F0108H, F0109H (SIR00) to F010EH, F010FH (SIR03) 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	FEC Tmn Note	PEC Tmn	OVC Tmn

FEC Tmn Note	Clear trigger of framing error flag of channel n
0	Not cleared
1	Clears the FEFmn bit of the SSRmn register to 0.

PEC Tmn	Clear trigger of parity error flag of channel n
0	Not cleared
1	Clears the PEFmn bit of the SSRmn register to 0.

OVC Tmn	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, SIR03 register only.

Caution Be sure to clear bits 15 to 3 (or bits 15 to 2 for the SIR00 and SIR02 registers) to 0.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)
Remark 2. When the SIRmn register is read, 0000H is always read.

13.3.8 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 13 - 11 Format of Serial status register mn (SSRmn) (1/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03) 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

<table>
<tr><th>TSF
mn</th><th>Communication status indication flag of channel n</th></tr>
<tr><td>0</td><td>Communication is stopped or suspended.</td></tr>
<tr><td>1</td><td>Communication is in progress.</td></tr>
<tr><td colspan="2">&lt;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.</td></tr>
</table>

<table>
<tr><th>BFF
mn</th><th>Buffer register status indication flag of channel n</th></tr>
<tr><td>0</td><td>Valid data is not stored in the SDRmn register.</td></tr>
<tr><td>1</td><td>Valid data is stored in the SDRmn register.</td></tr>
<tr><td colspan="2">&lt;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.</td></tr>
</table>

Note The SSR01, SSR03 register only.

Caution **When the simplified SPI (CSI) is performing reception operations in the SNOOZE mode (SWCm = 1), the BFFmn flag will not change.**

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Figure 13 - 11 Format of Serial status register mn (SSRmn) (2/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03) 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

FEF mn 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.	

PEF mn	Parity error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception) or ACK is not detected (during I²C 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 I²C transmission (ACK is not detected).	

OVF mn	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, SSR03 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 the simplified SPI (CSI) is performing reception operations in the SNOOZE mode (SWCm = 1), the OVFmn flag will not change.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

13.3.9 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 13 - 12 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	SS03	SS02	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 4 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 to 3)

Remark 2. When the SSm register is read, 0000H is always read.

13.3.10 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 13 - 13 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	ST03	ST02	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 4 of the ST0 register to 0.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

Remark 2. When the STm register is read, 0000H is always read.

13.3.11 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 13 - 14 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	SE03	SE02	SE01	SE00

SEmn	Indication of operation enable/stop status of channel n
0	Operation stops
1	Operation is enabled.

Caution Be sure to clear bits 15 to 4 of the SE0 register to 0.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

13.3.12 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 13 - 15 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	SOE 03	SOE 02	SOE 01	SOE 00

SOE mn	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 4 of the SOE0 register to 0.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

13.3.13 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 0F0FH.

Figure 13 - 16 Format of Serial output register m (SOm)

Address: F0128H, F0129H After reset: 0F0FH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	CKO 03	CKO 02	CKO 01	CKO 00	0	0	0	0	SO 03	SO 02	SO 01	SO 00

CKO mn	Serial clock output of channel n
0	Serial clock output value is 0.
1	Serial clock output value is 1.

SO mn	Serial data output of channel n
0	Serial data output value is 0.
1	Serial data output value is 1.

Caution **Be sure to clear bits 15 to 12 and 7 to 4 of the SO0 register to 0. For 20-pin products, be sure to set bit 9 of the SO0 register to 1.**

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3)

13.3.14 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 simplifies I^{2}C 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 13 - 17 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	SOL02	0	SOL00

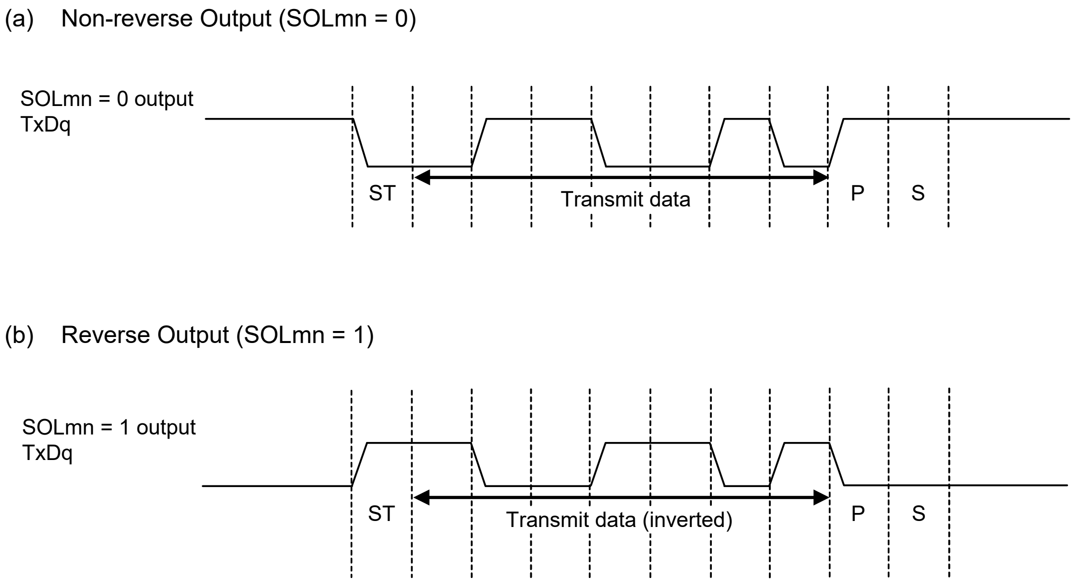
SOLmn	Selects inversion of the level of the transmit data of channel n in UART mode
0	Communication data is output as is.
1	Communication data is inverted and output.

Caution Be sure to clear bits 15 to 3 and 1 of the SOL0 register to 0.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2)

Figure 13 - 18 shows examples in which the level of transmit data is reversed during UART transmission.

Figure 13 - 18 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, 2)

13.3.15 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 13 - 19 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

<table>
<tr><th>SSECm</th><th>Selection of whether to enable or disable the generation of communication error interrupts in the SNOOZE mode</th></tr>
<tr><td>0</td><td>Enable the generation of error interrupts (INTSRE0).</td></tr>
<tr><td>1</td><td>Disable the generation of error interrupts (INTSRE0).</td></tr>
<tr><td colspan="2">• 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.</td></tr>
</table>

<table>
<tr><th>SWCm</th><th>Setting of the SNOOZE mode</th></tr>
<tr><td>0</td><td>Do not use the SNOOZE mode function.</td></tr>
<tr><td>1</td><td>Use the SNOOZE mode function.</td></tr>
<tr><td colspan="2">• When there is a hardware trigger signal in the STOP mode, the STOP mode is exited, and CSI00/UART0 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.</td></tr>
</table>

Caution Setting SSECm, SWCm = 1, 0 is prohibited.

Figure 13 - 20 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.

13.3.16 Input switch control register (ISC)

The ISC1 and ISC0 bits of the ISC register are used to realize a LIN-bus communication operation by UART0 in coordination with an external interrupt and the timer array unit.

When bit 0 is set to 1, the input signal of the serial data input (RxD0) pin is selected as an external interrupt (INTP0) that can be used to detect a wakeup signal.

When bit 1 is set to 1, the input signal of the serial data input (RxD0) pin is selected as a timer input, so that wake up signal can be detected, the low width of the break field, and the pulse width of the sync field can be measured by the timer.

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 13 - 21 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	ISC1	ISC0

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.

ISC1	Switching channel 3 input of timer array unit 0
0	Uses the input signal specified by TIS06 and TIS05 bits of TIS0 register as a timer input (normal operation).
1	Input signal of the RxD0 pin is used as timer input (detects the wakeup signal and measures the low width of the break field and the pulse width of the sync field).

ISC0	Switching external interrupt (INTP0) input
0	Uses the input signal of the INTP0 pin as an external interrupt (normal operation).
1	Uses the input signal of the RxD0 pin as an external interrupt (wakeup signal detection).

Caution Be sure to clear bits 6 to 2 to 0.

13.3.17 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^{2}C 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 13 - 22 Format of Noise filter enable register 0 (NFEN0)

Address: F0070H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN0	0	0	0	0	0	SNFEN10	0	SNFEN00

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.	

SNFEN10	Use of noise filter of RxD1 pin
0	Noise filter OFF
1	Noise filter ON
Set the SNFEN10 bit to 1 to use the RxD1 pin. Clear the SNFEN10 bit to 0 to use the other than RxD1 pin.	

Caution Be sure to clear bits 7 to 3, 1, and 0 of 10 and 16-pin products to 0. Be sure to clear bits 7 to 3 and 1 of 20, 24, and 25-pin products to 0.

13.3.18 Registers that control port functions of serial input/output pins

Using the serial array unit requires setting of the registers that control the port functions for the port pins with which the serial array unit pin functions for the target channel are multiplexed (port mode register (PMxx), port register (Pxx), port input mode register (PIMxx), port output mode register (POMxx), and port mode control register (PMCxx)).

For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.2 Port registers (Pxx)**, **4.3.4 Port input mode registers (PIMxx)**, **4.3.5 Port output mode registers (POMxx)**, and **4.3.6 Port mode control registers (PMCxx)**.

Using a port pin which is multiplexed with a serial data output or serial clock output pin function (e.g. P01/ANI16/INTP5/SO10/TxD1) for serial data output or serial clock output requires setting the corresponding bit in the port mode register (PMxx) to 0, and the corresponding bit in the port register (Pxx) to 1.

Using a the port pin in N-ch open-drain output (V_{DD} tolerance[Note 1]/EV_{DD} tolerance[Note 2]) mode requires setting the corresponding bit in the port output mode register (POMxx) to 1. When connecting an external device operating at a different voltage (1.8 V, 2.5 V, or 3 V), see **4.4.4 Handling different potential (1.8 V, 2.5 V, 3.0 V) by using I/O buffers**.

Example When P01/ANI16/INTP5/SO10/TxD1 is to be used for serial data output
Set the PMC01 bit of port mode control register 0 to 0.
Set the PM01 bit of port mode register 0 to 0.
Set the P01 bit of port register 0 to 1.

Using a port pin which is multiplexed with a serial data input or serial clock input pin function (e.g. P00/ANI17/PCLBUZ1/TI03/(VCOUT1)/SI10/RxD1/SDA10/(SDAA1)) for serial data input or serial clock input requires setting the corresponding bit in the port mode register (PMxx) to 1. At this time, the value of the corresponding bit in the port register (Pxx) may be 0 or 1.

Using a TTL input buffer requires setting the corresponding bit in the port input mode register (PIMxx) to 1. When connecting an external device operating at a different voltage (1.8 V, 2.5 V or 3 V), see **4.4.4 Handling different potential (1.8 V, 2.5 V, 3.0 V) by using I/O buffers**.

Example When P00/ANI17/PCLBUZ1/TI03/(VCOUT1)/SI10/RxD1/SDA10/(SDAA1) is to be used for serial data input
Set the PMC00 bit of port mode control register 0 to 1.
Set the PM00 bit of port mode register 0 to 1.
Set the P00 bit of port register 0 to 0 or 1.

Note 1. 10, 16, 20, 24-pin products

Note 2. 25-pin products

13.4 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.

13.4.1 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 13 - 23 Peripheral Enable Register 0 (PER0) Setting When Stopping the Operation by Units

(a) Peripheral enable register 0 (PER0)... Set only the bit of SAUm to be stopped to 0.

	7	6	5	4	3	2	1	0
PER0		IICA1EN	ADCEN	IICA0EN		SAU0EN		TAU0EN
	0	×	×	×	0	0/1	0	×

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 registers 0, 3 to 5 (PIM0, PIM3 to PIM5)**
- **Port output mode registers 0, 3 to 5 (POM0, POM3 to POM5)**
- **Port mode registers 0, 2 to 5 (PM0, PM2 to PM5)**
- **Port registers 0, 2 to 5 (P0, P2 to P5)**

Caution 2. Be sure to clear the following bits to 0.
Bits 1, 3, and 7

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

13.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

Figure 13 - 24 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	STm3 0/1	STm2 0/1	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	SEm3 0/1	SEm2 0/1	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	SOEm3 0/1	SOEm2 0/1	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	SOm3 0/1	SOm2 0/1	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 to 3)

Remark 2. ▨: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

13.5 Operation of Simplified SPI (CSI00, CSI01, CSI10, CSI11) Communication

This is a clocked communication function that uses three lines: serial clock (SCK) and serial data (SI and SO) lines.

[Data transmission/reception]

- Data length of 7 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

Simplified SPIs (CSIs) of the following channels 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 SPIs (CSIs) can be specified.

- CSI00

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics. For details, see **CHAPTER 35**, **CHAPTER 36 ELECTRICAL SPECIFICATIONS**.

The channels supporting simplified SPI (CSI00, CSI01, CSI10, CSI11) are channels 0 to 3 of SAU0.

- 10-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART
0	0	—	—
	1	—	
	2	CSI10	UART1
	3	—	

- 16-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	—	UART0 (LIN-bus supported)	—
	1	—		—
	2	CSI10	UART1	—
	3	CSI11		IIC11

- 20-pin products

Unit	Channel	Used as Smplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00 (supporting slave select input function)	UART0 (LIN-bus supported)	IIC00
	1	—		—
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11

- 24, 25-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 (LIN-bus supported)	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11

Simplified SPI (CSI00, CSI01, CSI10, CSI11) performs the following seven types of communication operations.

• Master transmission	(See **13.5.1**.)
• Master reception	(See **13.5.2**.)
• Master transmission/reception	(See **13.5.3**.)
• Slave transmission	(See **13.5.4**.)
• Slave reception	(See **13.5.5**.)
• Slave transmission/reception	(See **13.5.6**.)
• SNOOZE mode function (CSI00 only)	(See **13.5.7**.)

13.5.1 Master transmission

Master transmission is that the RL78 microcontroller outputs a transfer clock and transmits data to another device.

Simplified SPI	CSI00	CSI01	CSI10	CSI11
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0
Pins used	SCK00, SO00	SCK01, SO01	SCK10, SO10	SCK11, SO11
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	None			
Transfer data length	7 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 35**, **CHAPTER 36 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), mn = 00 to 03

(1) Register setting

Figure 13 - 25 Example of Contents of Registers for Master Transmission of Simplified SPI (CSI00, CSI01, CSI10, CSI11)

(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 **13.3 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)							

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	CKOm3 0/1	CKOm2 0/1	CKOm1 0/1	CKOm0 0/1	0	0	0	0	SOm3 0/1	SOm2 0/1	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	SOEm3 0/1	SOEm2 0/1	SOEm1 0/1	SOEm0 0/1

(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

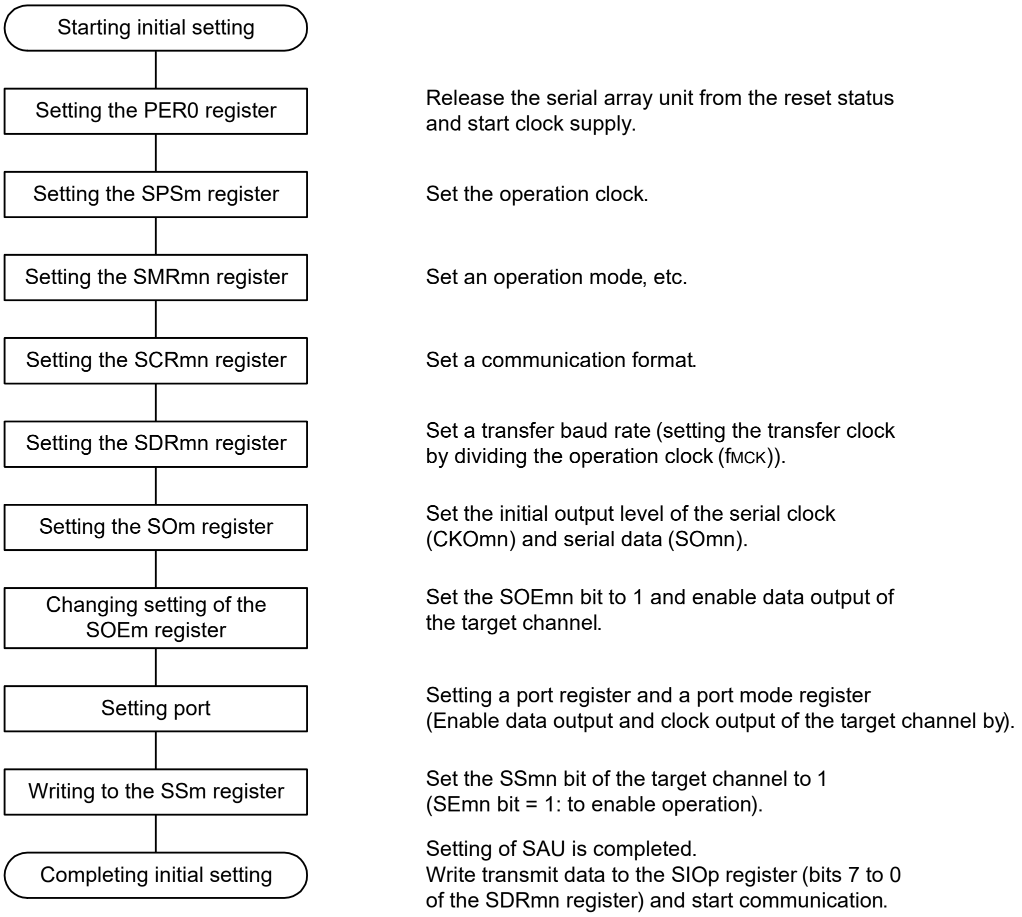
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 0/1	SSm1 0/1	SSm0 0/1

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11), mn = 00 to 03

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 13 - 26 Initial Setting Procedure for Master Transmission


Starting initial setting
Setting the PER0 register
Setting the SPSm register
Setting the SMRmn register
Setting the SCRmn register
Setting the SDRmn register
Setting the SOm register
Changing setting of the SOEm register
Setting port
Writing to the SSm register
Completing initial setting

Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).

Set the initial output level of the serial clock (CKOmn) and serial data (SOmn).

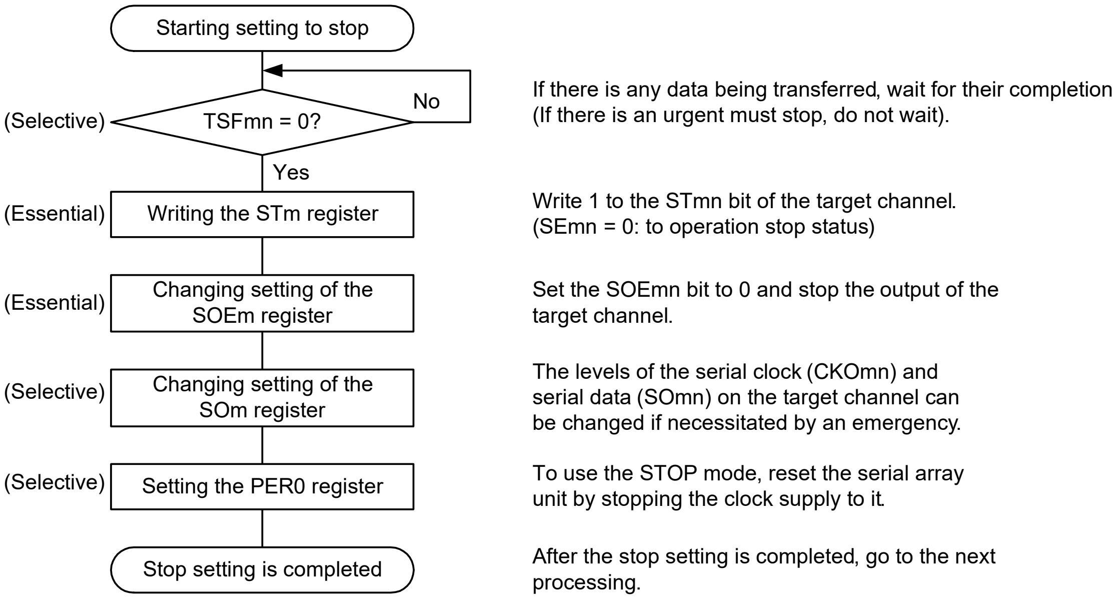
Set the SOEmn bit to 1 and enable data output of the target channel.

Setting a port register and a port mode register (Enable data output and clock output of the target channel by).

Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation).

Setting of SAU is completed.
Write transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and start communication.

Figure 13 - 27 Procedure for Stopping Master Transmission


Starting setting to stop
(Selective)
TSFmn = 0?
No
Yes
(Essential)
Writing the STm register
(Essential)
Changing setting of the SOEm register
(Selective)
Changing setting of the SOm register
(Selective)
Setting the PER0 register
Stop setting is completed

If there is any data being transferred, wait for their completion (If there is an urgent must stop, do not wait).

Write 1 to the STmn bit of the target channel. (SEmn = 0: to operation stop status)

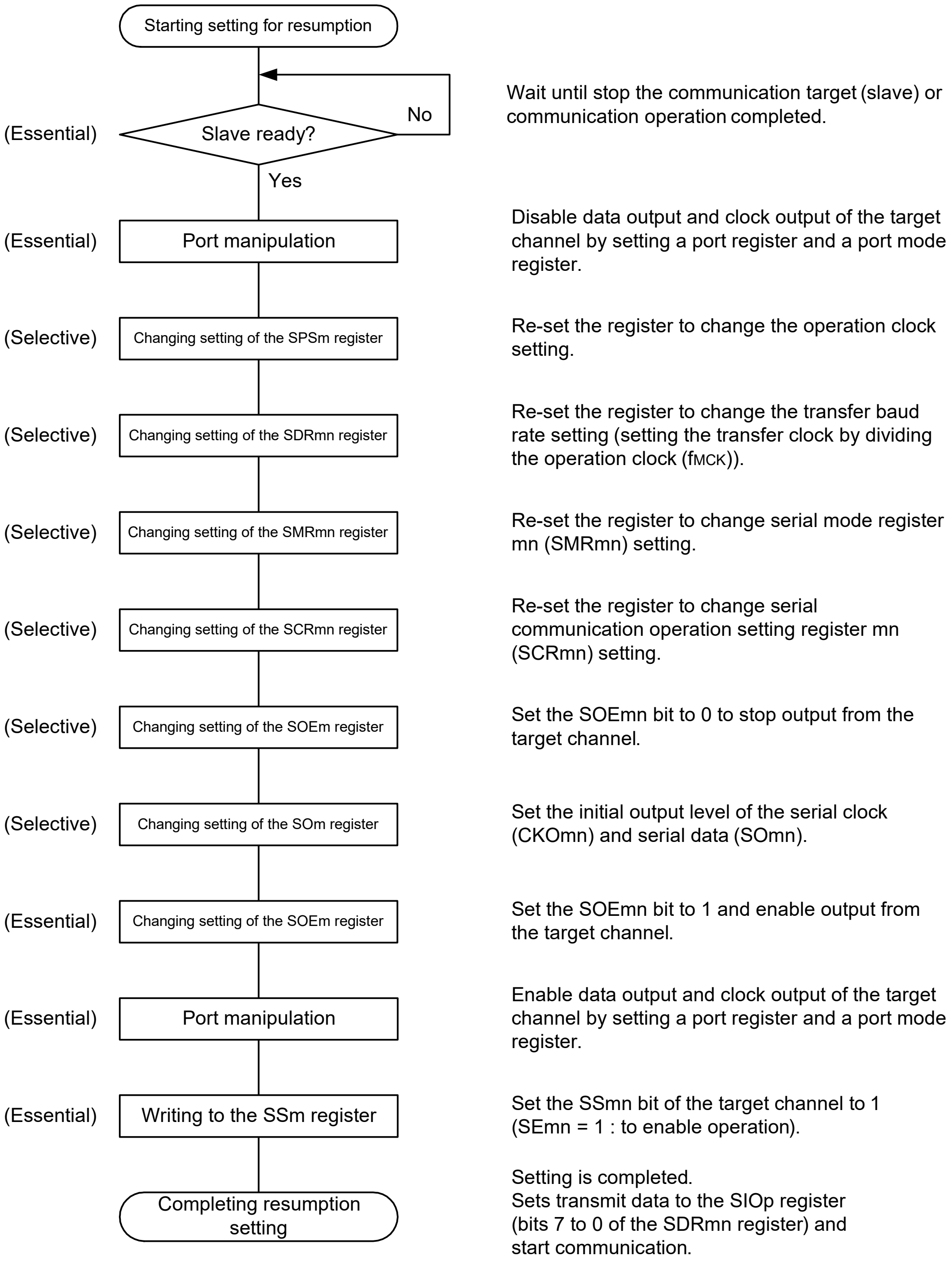
Set the SOEmn bit to 0 and stop the output of the target channel.

The levels of the serial clock (CKOmn) and serial data (SOmn) on the target channel can be changed if necessitated by an emergency.

To use the STOP mode, reset the serial array unit by stopping the clock supply to it.

After the stop setting is completed, go to the next processing.

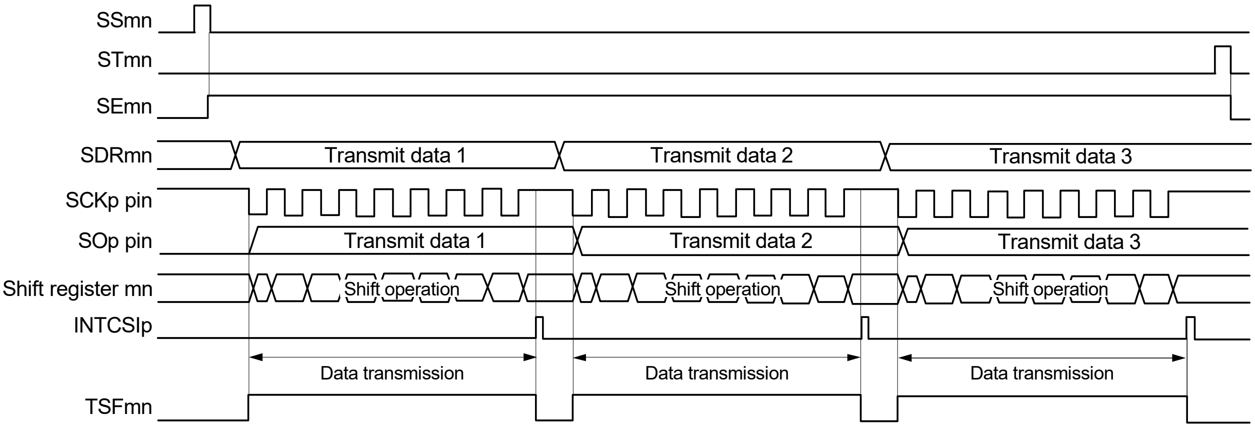
Figure 13 - 28 Procedure for Resuming Master Transmission


Starting setting for resumption
(Essential)
Slave ready?
No
Yes
Wait until stop the communication target (slave) or communication operation completed.
(Essential)
Port manipulation
Disable data output and clock output of the target channel by setting a port register and a port mode register.
(Selective)
Changing setting of the SPSm register
Re-set the register to change the operation clock setting.
(Selective)
Changing setting of the SDRmn register
Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (fMCK)).
(Selective)
Changing setting of the SMRmn register
Re-set the register to change serial mode register mn (SMRmn) setting.
(Selective)
Changing setting of the SCRmn register
Re-set the register to change serial communication operation setting register mn (SCRmn) setting.
(Selective)
Changing setting of the SOEm register
Set the SOEmn bit to 0 to stop output from the target channel.
(Selective)
Changing setting of the SOm register
Set the initial output level of the serial clock (CKOmn) and serial data (SOmn).
(Essential)
Changing setting of the SOEm register
Set the SOEmn bit to 1 and enable output from the target channel.
(Essential)
Port manipulation
Enable data output and clock output of the target channel by setting a port register and a port mode register.
(Essential)
Writing to the SSm register
Set the SSmn bit of the target channel to 1 (SEmn = 1 : to enable operation).
Completing resumption setting
Setting is completed.
Sets transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and start communication.

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.

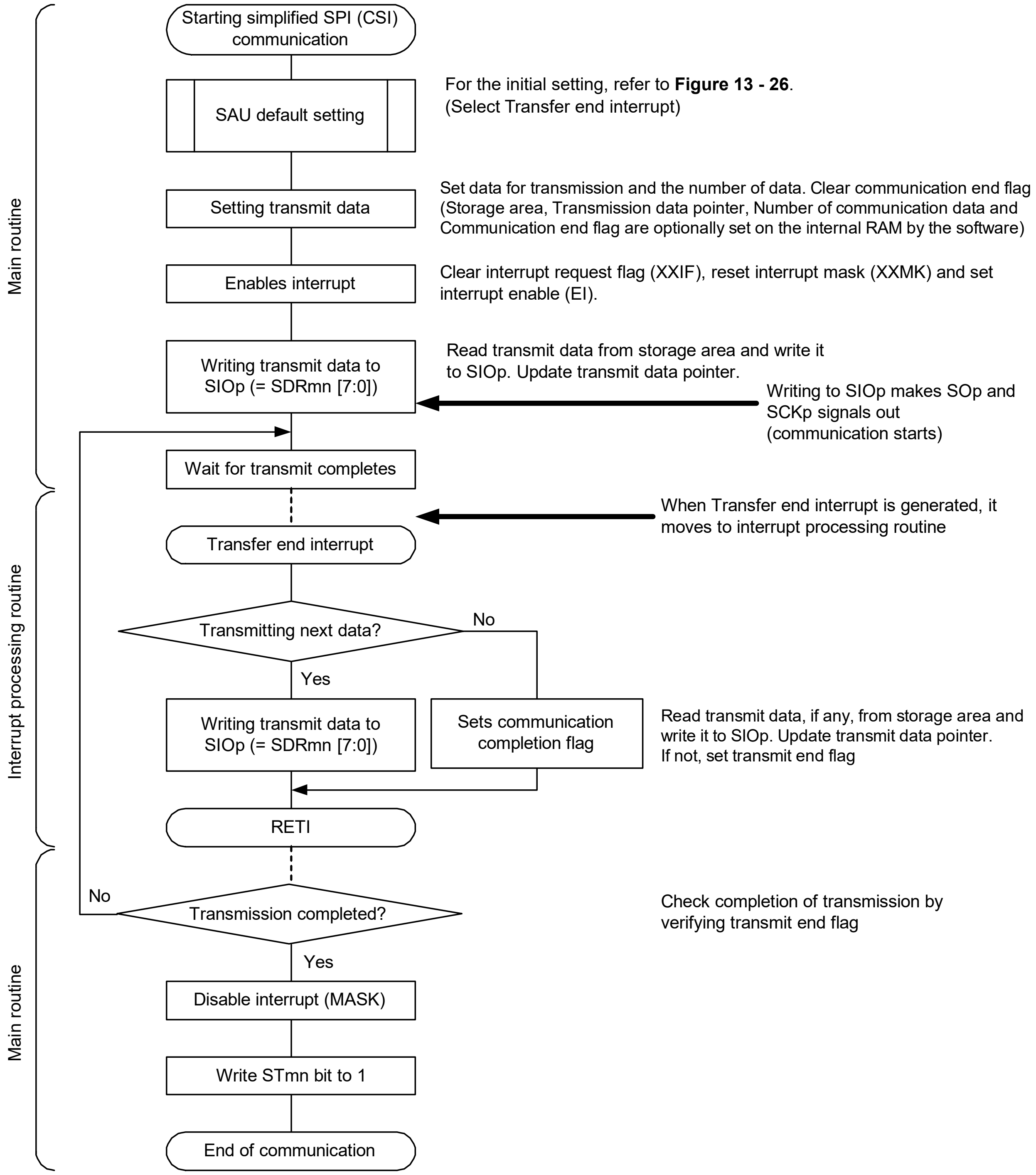
(3) Processing flow (in single-transmission mode)

Figure 13 - 29 Timing Chart of Master Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



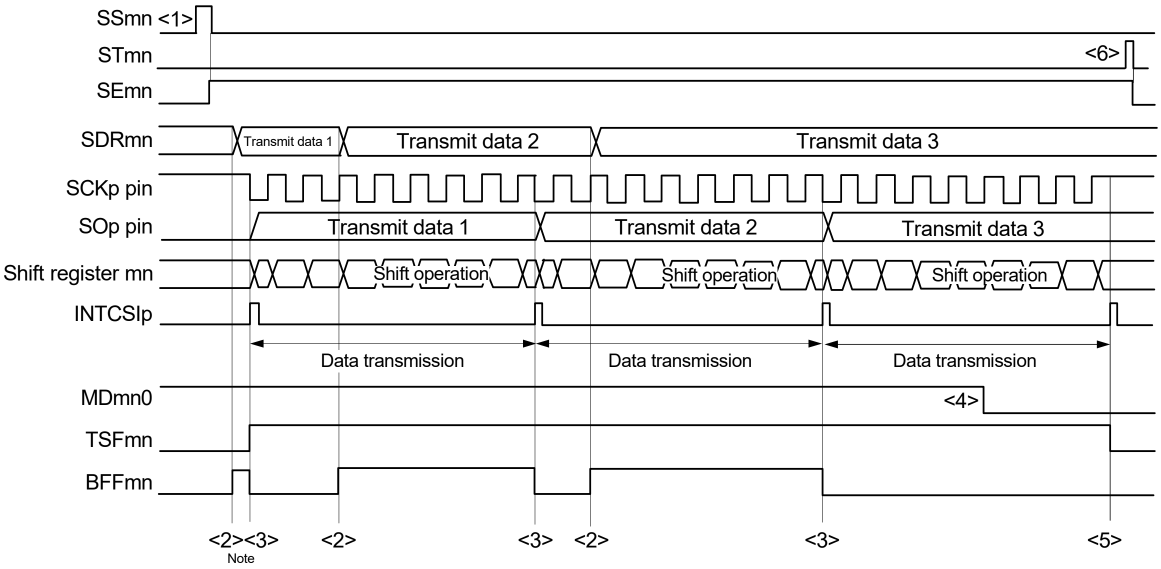
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11), mn = 00 to 03

Figure 13 - 30 Flowchart of Master Transmission (in Single-Transmission Mode)


Main routine
Starting simplified SPI (CSI) communication
SAU default setting
For the initial setting, refer to Figure 13 - 26.
(Select Transfer end interrupt)
Setting transmit data
Set data for transmission and the number of data. Clear communication end flag (Storage area, Transmission data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI).
Writing transmit data to SIOp (= SDRmn [7:0])
Read transmit data from storage area and write it to SIOp. Update transmit data pointer.
Writing to SIOp makes SOp and SCKp signals out (communication starts)
Wait for transmit completes
Interrupt processing routine
When Transfer end interrupt is generated, it moves to interrupt processing routine
Transfer end interrupt
Transmitting next data?
No
Yes
Writing transmit data to SIOp (= SDRmn [7:0])
Sets communication completion flag
Read transmit data, if any, from storage area and write it to SIOp. Update transmit data pointer.
If not, set transmit end flag
RETI
Main routine
No
Transmission completed?
Check completion of transmission by verifying transmit end flag
Yes
Disable interrupt (MASK)
Write STmn bit to 1
End of communication

(4) Processing flow (in continuous transmission mode)

Figure 13 - 31 Timing Chart of Master Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)

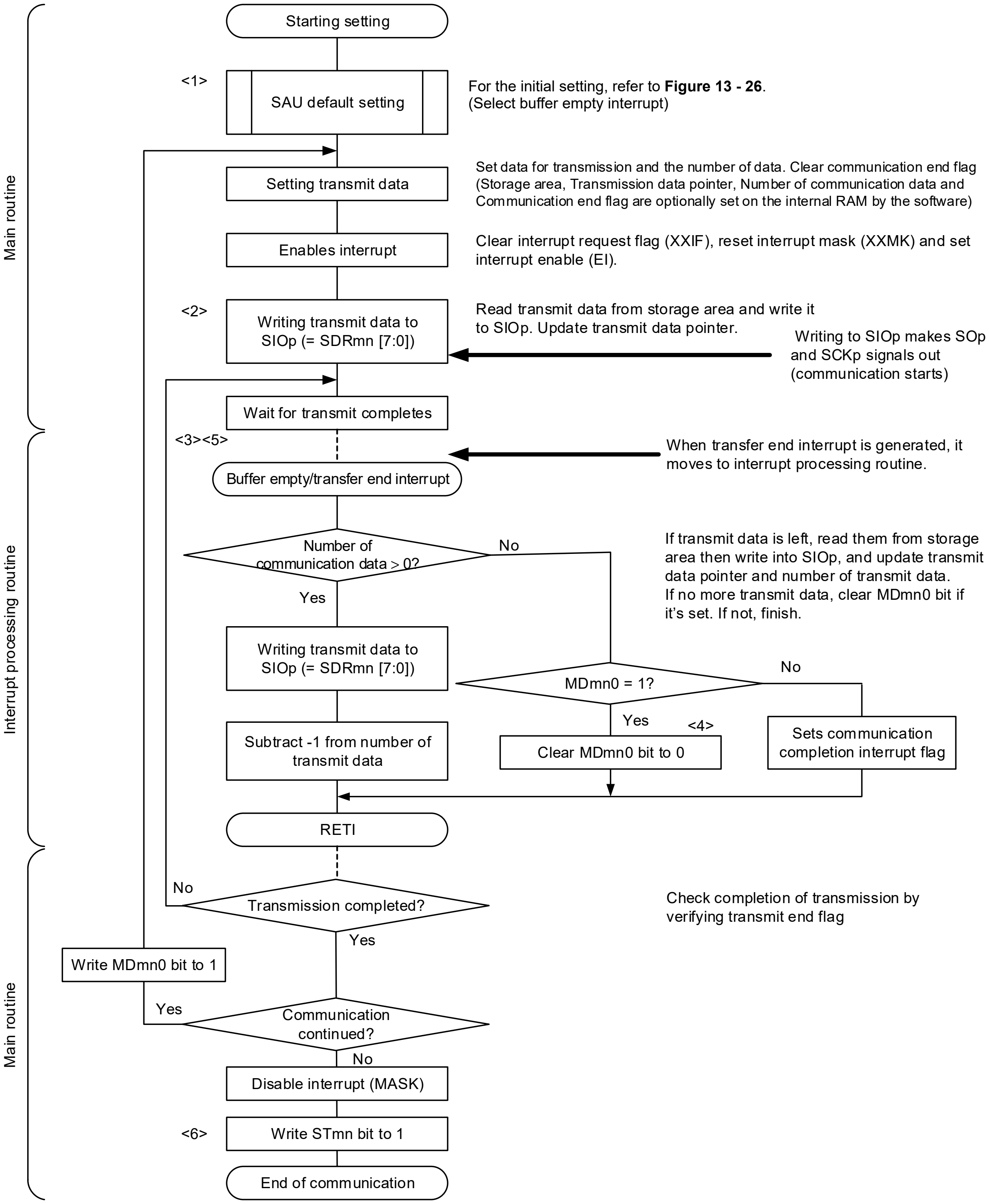


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution **The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.**

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11), mn = 00 to 03

Figure 13 - 32 Flowchart of Master Transmission (in Continuous Transmission Mode)


Main routine
Starting setting
<1>
SAU default setting
For the initial setting, refer to **Figure 13 - 26**.
(Select buffer empty interrupt)
Setting transmit data
Set data for transmission and the number of data. Clear communication end flag
(Storage area, Transmission data pointer, Number of communication data and
Communication end flag are optionally set on the internal RAM by the software)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set
interrupt enable (EI).
<2>
Writing transmit data to SIOp (= SDRmn [7:0])
Read transmit data from storage area and write it
to SIOp. Update transmit data pointer.
Writing to SIOp makes SOp
and SCKp signals out
(communication starts)
Wait for transmit completes
<3><5>
Interrupt processing routine
Buffer empty/transfer end interrupt
When transfer end interrupt is generated, it
moves to interrupt processing routine.
Number of communication data > 0?
No
Yes
If transmit data is left, read them from storage
area then write into SIOp, and update transmit
data pointer and number of transmit data.
If no more transmit data, clear MDmn0 bit if
it's set. If not, finish.
Writing transmit data to SIOp (= SDRmn [7:0])
MDmn0 = 1?
No
Yes
<4>
Subtract -1 from number of transmit data
Clear MDmn0 bit to 0
Sets communication completion interrupt flag
RETI
Main routine
No
Transmission completed?
Check completion of transmission by
verifying transmit end flag
Yes
Write MDmn0 bit to 1
Yes
Communication continued?
No
Disable interrupt (MASK)
<6>
Write STmn bit to 1
End of communication

Remark <1> to <6> in the figure correspond to <1> to <6> in Figure 13 - 31 Timing Chart of Master Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0).

13.5.2 Master reception

Master reception is that the RL78 microcontroller outputs a transfer clock and receives data from other device.

Simplified SPI	CSI00	CSI01	CSI10	CSI11
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0
Pins used	SCK00, SI00	SCK01, SI01	SCK10, SI10	SCK11, SI11
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	Overrun error detection flag (OVFmn) only			
Transfer data length	7 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 35**, **CHAPTER 36 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11), mn = 00 to 03

(1) Register setting

Figure 13 - 33 Example of Contents of Registers for Master Reception of simplified SPI (CSI00, CSI01, CSI10, CSI11)

(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 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 **13.3 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	Receive data (Write FFH as dummy data.)							

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	CKOm3 0/1	CKOm2 0/1	CKOm1 0/1	CKOm0 0/1	0	0	0	0	SOm3 ×	SOm2 ×	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	SOEm3 ×	SOEm2 ×	SOEm1 ×	SOEm0 ×

(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

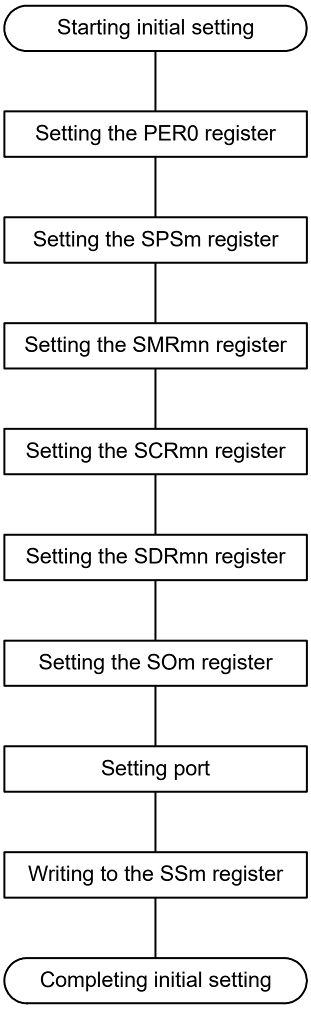
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 0/1	SSm1 0/1	SSm0 0/1

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11), mn = 00 to 03

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 13 - 34 Initial Setting Procedure for Master Reception


Starting initial setting
Setting the PER0 register
Setting the SPSm register
Setting the SMRmn register
Setting the SCRmn register
Setting the SDRmn register
Setting the SOm register
Setting port
Writing to the SSm register
Completing initial setting

Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate (setting the transfer clock by dividing the operation clock (fMCK)).

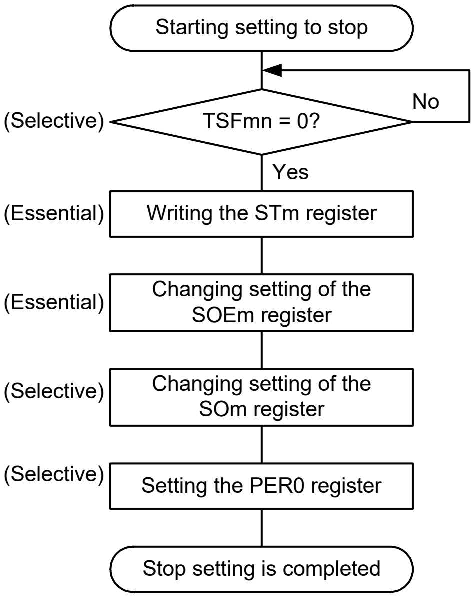
Set the initial output level of the serial clock (CKOmn).

Enable clock output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation).

Initial setting is completed.
Set dummy data to the SIOp register (bits 7 to 0 of the SDRmn register) and start communication.

Figure 13 - 35 Procedure for Stopping Master Reception


Starting setting to stop
(Selective)
TSFmn = 0?
No
Yes
(Essential)
Writing the STm register
(Essential)
Changing setting of the SOEm register
(Selective)
Changing setting of the SOm register
(Selective)
Setting the PER0 register
Stop setting is completed

If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait.)

Write 1 to the STmn bit of the target channel.
(SEmn = 0: to operation stop status)

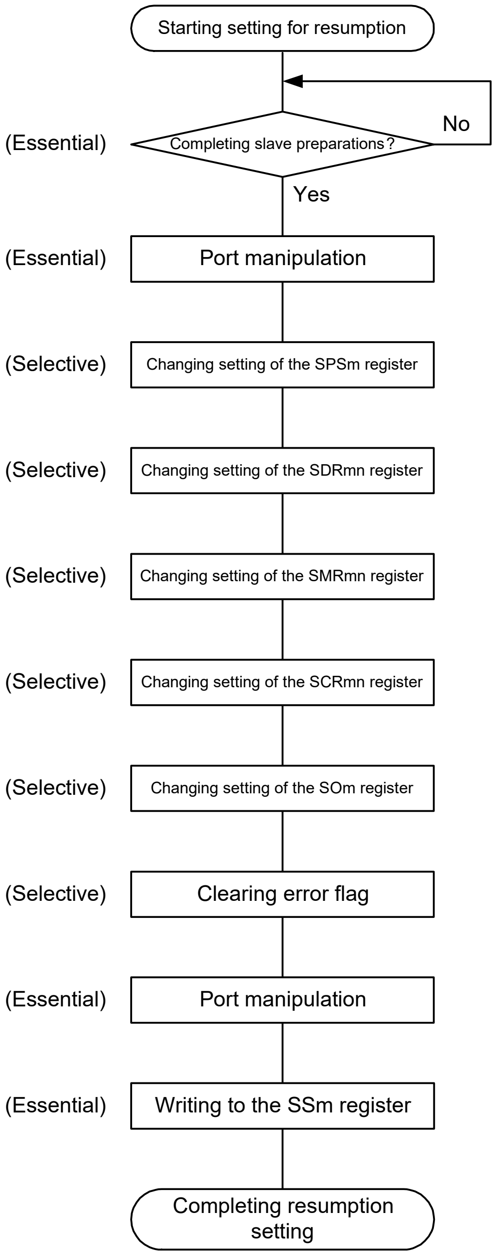
Set the SOEmn bit to 0 and stop the output of the target channel.

The levels of the serial clock (CKOmn) on the target channel can be changed if necessitated by an emergency.

Reset the serial array unit by stopping the clock supply to it.

After the stop setting is completed, go to the next processing.

Figure 13 - 36 Procedure for Resuming Master Reception


Starting setting for resumption
(Essential)
Completing slave preparations?
No
Yes
(Essential)
Port manipulation
(Selective)
Changing setting of the SPSm register
(Selective)
Changing setting of the SDRmn register
(Selective)
Changing setting of the SMRmn register
(Selective)
Changing setting of the SCRmn register
(Selective)
Changing setting of the SOm register
(Selective)
Clearing error flag
(Essential)
Port manipulation
(Essential)
Writing to the SSm register
Completing resumption setting

Wait until stop the communication target (slave) or communication operation completed.

Disable clock output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change serial communication operation setting register mn (SCRmn) setting.

Set the initial output level of the serial clock (CKOmn).

If the OVF flag remains set, clear this using serial flag clear trigger register mn (SIRmn).

Enable clock output of the target channel by setting a port register and a port mode register.

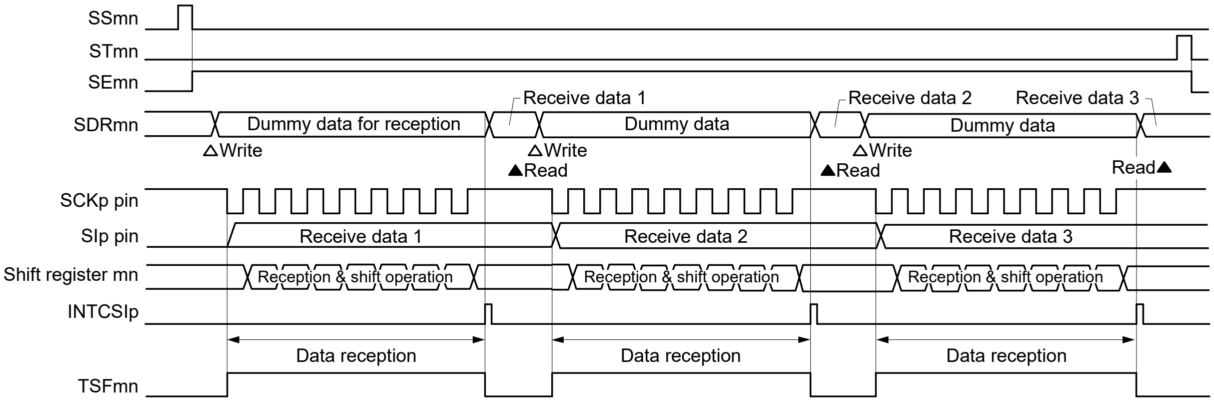
Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation).

Setting is completed.
Sets dummy data to the SIOp register (bits 7 to 0 of the SDRmn register) and start communication.

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.

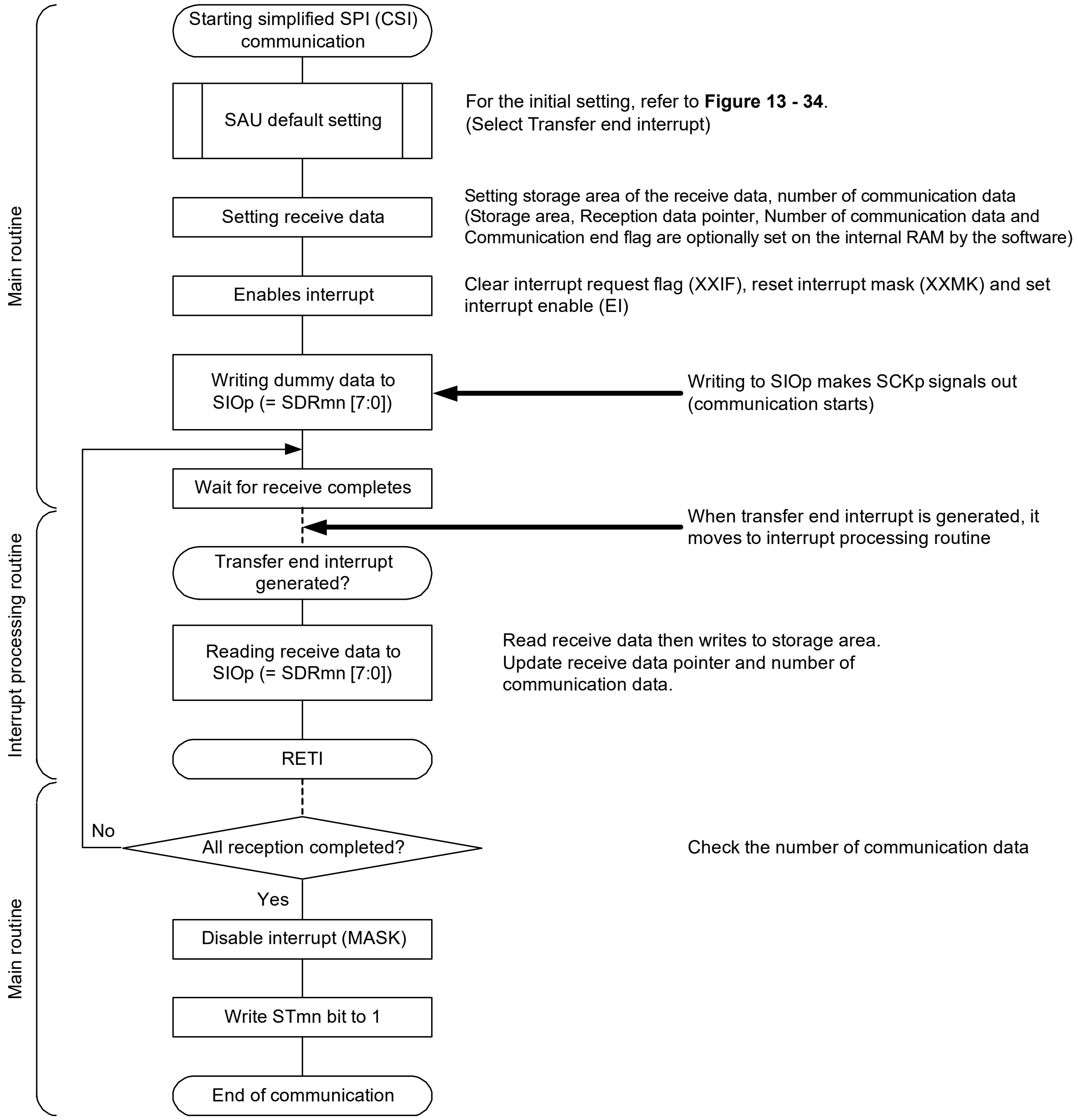
(3) Processing flow (in single-reception mode)

Figure 13 - 37 Timing Chart of Master Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



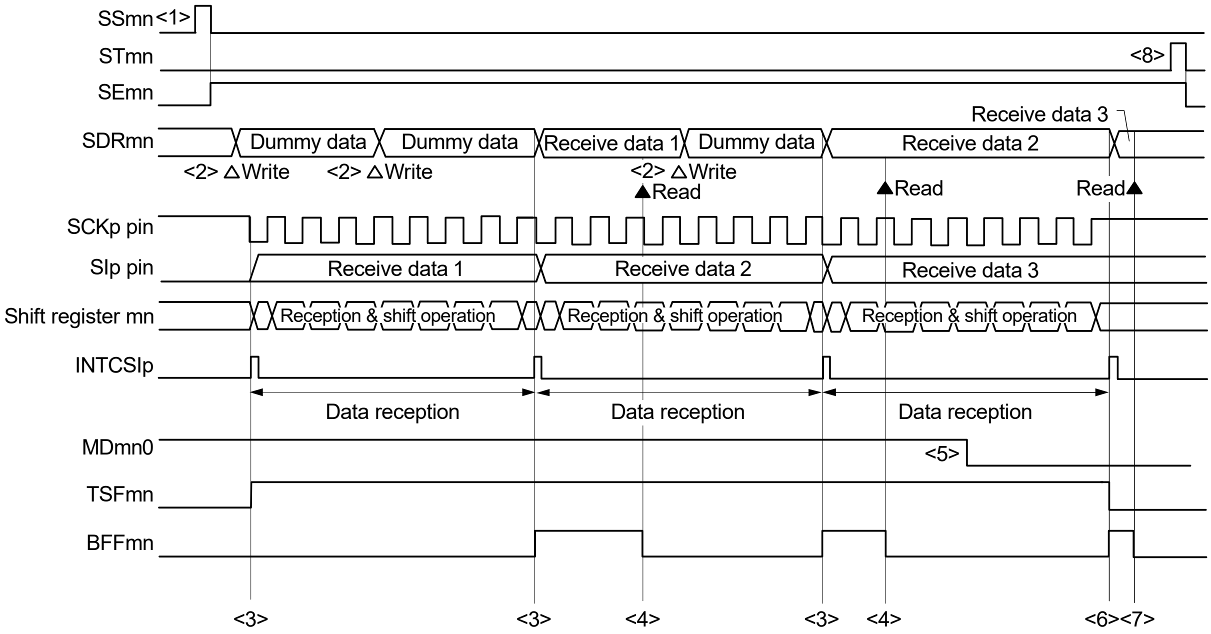
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11), mn = 00 to 03

Figure 13 - 38 Flowchart of Master Reception (in Single-Reception Mode)


Main routine
Starting simplified SPI (CSI) communication
SAU default setting
For the initial setting, refer to Figure 13 - 34.
(Select Transfer end interrupt)
Setting receive data
Setting storage area of the receive data, number of communication data
(Storage area, Reception data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI)
Writing dummy data to SIOp (= SDRmn [7:0])
Writing to SIOp makes SCKp signals out (communication starts)
Wait for receive completes
Interrupt processing routine
When transfer end interrupt is generated, it moves to interrupt processing routine
Transfer end interrupt generated?
Reading receive data to SIOp (= SDRmn [7:0])
Read receive data then writes to storage area. Update receive data pointer and number of communication data.
RETI
Main routine
No
All reception completed?
Check the number of communication data
Yes
Disable interrupt (MASK)
Write STmn bit to 1
End of communication

(4) Processing flow (in continuous reception mode)

Figure 13 - 39 Timing Chart of Master Reception (in Continuous Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

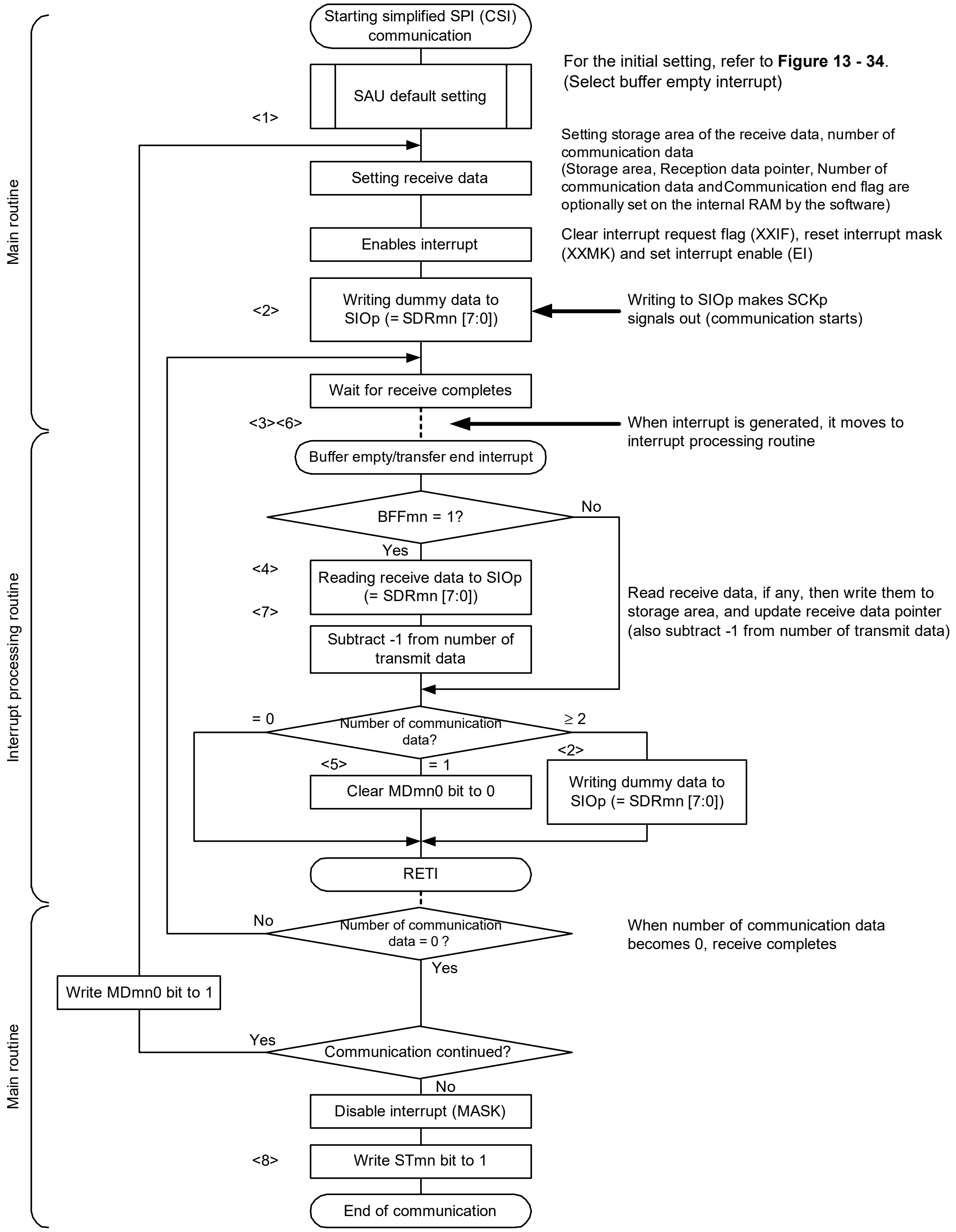


Caution **The MDmn0 bit can be rewritten even during operation.**
However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in Figure 13 - 40 Flowchart of Master Reception (in Continuous Reception Mode).

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11), mn = 00 to 03

Figure 13 - 40 Flowchart of Master Reception (in Continuous Reception Mode)


Main routine
Interrupt processing routine
Main routine
Starting simplified SPI (CSI) communication
SAU default setting
For the initial setting, refer to Figure 13 - 34.
(Select buffer empty interrupt)
<1>
Setting receive data
Setting storage area of the receive data, number of communication data
(Storage area, Reception data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI)
<2>
Writing dummy data to SIOp (= SDRmn [7:0])
Writing to SIOp makes SCKp signals out (communication starts)
Wait for receive completes
<3><6>
When interrupt is generated, it moves to interrupt processing routine
Buffer empty/transfer end interrupt
BFFmn = 1?
No
Yes
<4>
Reading receive data to SIOp (= SDRmn [7:0])
<7>
Subtract -1 from number of transmit data
Read receive data, if any, then write them to storage area, and update receive data pointer (also subtract -1 from number of transmit data)
= 0
Number of communication data?
≥ 2
<2>
<5>
= 1
Clear MDmn0 bit to 0
Writing dummy data to SIOp (= SDRmn [7:0])
RETI
No
Number of communication data = 0 ?
When number of communication data becomes 0, receive completes
Yes
Write MDmn0 bit to 1
Yes
Communication continued?
No
Disable interrupt (MASK)
<8>
Write STmn bit to 1
End of communication

Remark <1> to <8> in the figure correspond to <1> to <8> in Figure 13 - 39 Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

13.5.3 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	CSI00	CSI01	CSI10	CSI11
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01	SCK10, SI10, SO10	SCK11, SI11, SO11
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	Overrun error detection flag (OVFmn) only			
Transfer data length	7 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 35**, **CHAPTER 36 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11), mn = 00 to 03

(1) Register setting

Figure 13 - 41 Example of Contents of Registers for Master Transmission/Reception of Simplified SPI (CSI00, CSI01, CSI10, CSI11)

(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 **13.3 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							
									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	CKOm3 0/1	CKOm2 0/1	CKOm1 0/1	CKOm0 0/1	0	0	0	0	SOm3 0/1	SOm2 0/1	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	SOEm3 0/1	SOEm2 0/1	SOEm1 0/1	SOEm0 0/1

(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

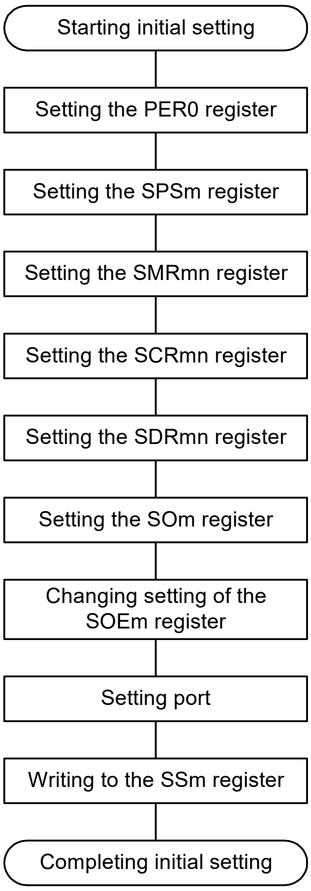
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 0/1	SSm1 0/1	SSm0 0/1

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11), mn = 00 to 03

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 13 - 42 Initial Setting Procedure for Master Transmission/Reception


Starting initial setting
Setting the PER0 register
Setting the SPSm register
Setting the SMRmn register
Setting the SCRmn register
Setting the SDRmn register
Setting the SOm register
Changing setting of the SOEm register
Setting port
Writing to the SSm register
Completing initial setting

Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).

Set the initial output level of the serial clock (CKOmn) and serial data (SOmn).

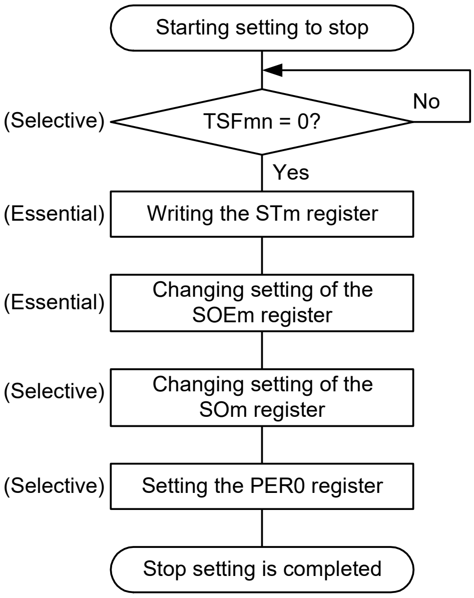
Set the SOEmn bit to 1 and enable data output of the target channel.

Enable data output and clock output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation).

Initial setting is completed.
Set transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and start.

Figure 13 - 43 Procedure for Stopping Master Transmission/Reception


Starting setting to stop
(Selective)
TSFmn = 0?
No
Yes
(Essential)
Writing the STm register
(Essential)
Changing setting of the SOEm register
(Selective)
Changing setting of the SOm register
(Selective)
Setting the PER0 register
Stop setting is completed

If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait.)

Write 1 to the STmn bit of the target channel.
(SEmn = 0: to operation stop status)

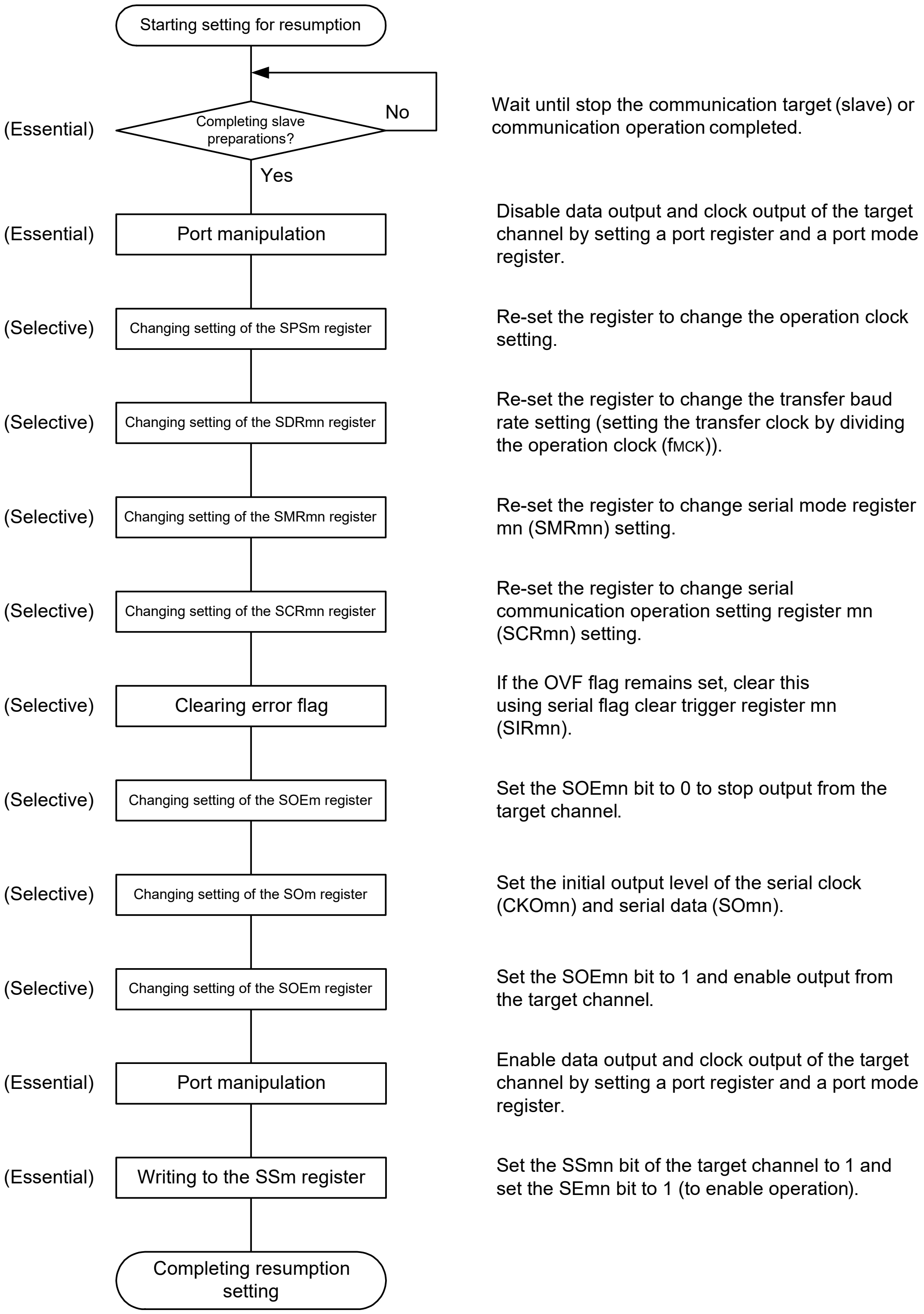
Set the SOEmn bit to 0 and stop the output of the target channel.

The levels of the serial clock (CKOmn) and serial data (SOmn) on the target channel can be changed if necessitated by an emergency.

Reset the serial array unit by stopping the clock supply to it.

After the stop setting is completed, go to the next processing.

Figure 13 - 44 Procedure for Resuming Master Transmission/Reception


Starting setting for resumption
(Essential)
Completing slave preparations?
No
Yes
(Essential)
Port manipulation
(Selective)
Changing setting of the SPSm register
(Selective)
Changing setting of the SDRmn register
(Selective)
Changing setting of the SMRmn register
(Selective)
Changing setting of the SCRmn register
(Selective)
Clearing error flag
(Selective)
Changing setting of the SOEm register
(Selective)
Changing setting of the SOm register
(Selective)
Changing setting of the SOEm register
(Essential)
Port manipulation
(Essential)
Writing to the SSm register
Completing resumption setting

Wait until stop the communication target (slave) or communication operation completed.

Disable data output and clock output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change serial communication operation setting register mn (SCRmn) setting.

If the OVF flag remains set, clear this using serial flag clear trigger register mn (SIRmn).

Set the SOEmn bit to 0 to stop output from the target channel.

Set the initial output level of the serial clock (CKOmn) and serial data (SOmn).

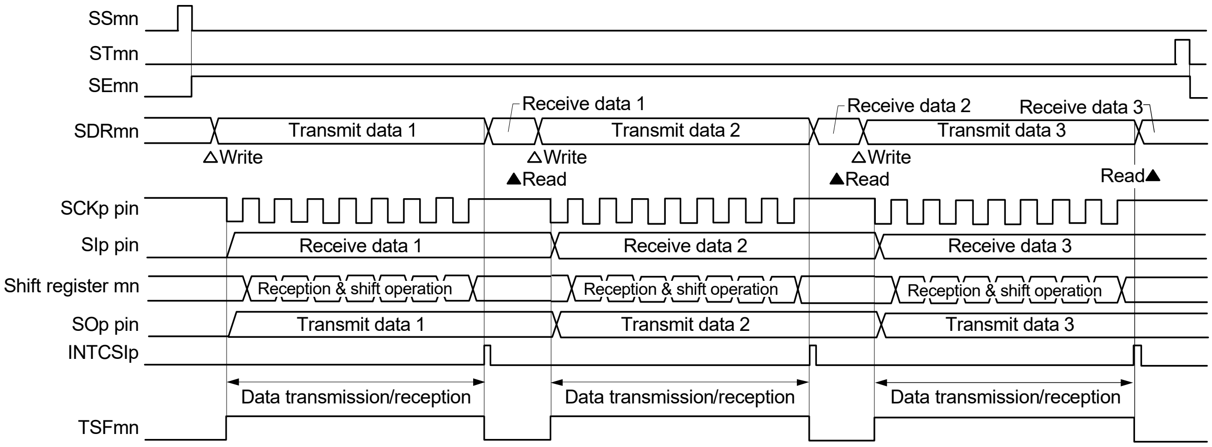
Set the SOEmn bit to 1 and enable output from the target channel.

Enable data output and clock output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 and set the SEmn bit to 1 (to enable operation).

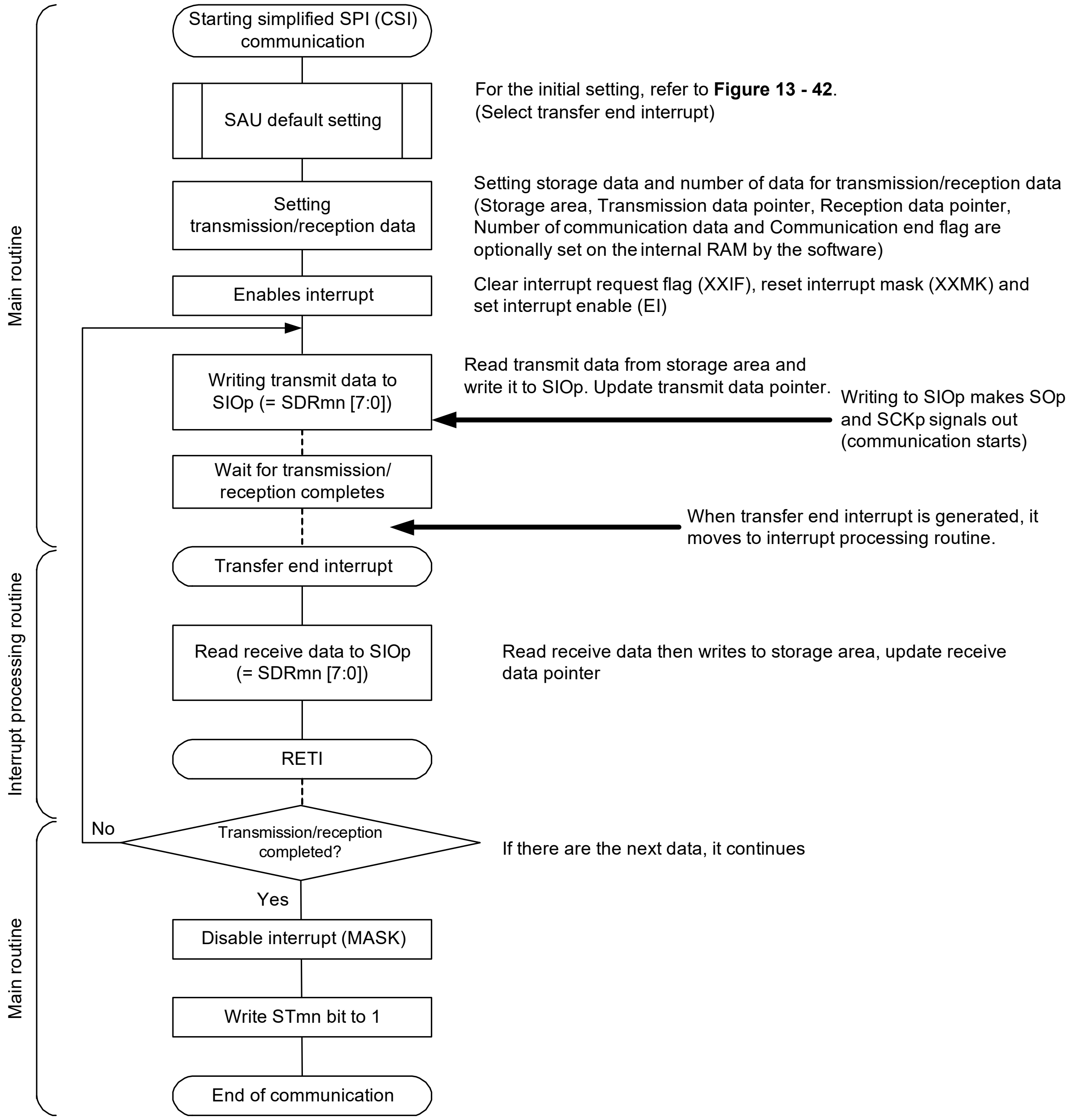
(3) Processing flow (in single-transmission/reception mode)

Figure 13 - 45 Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



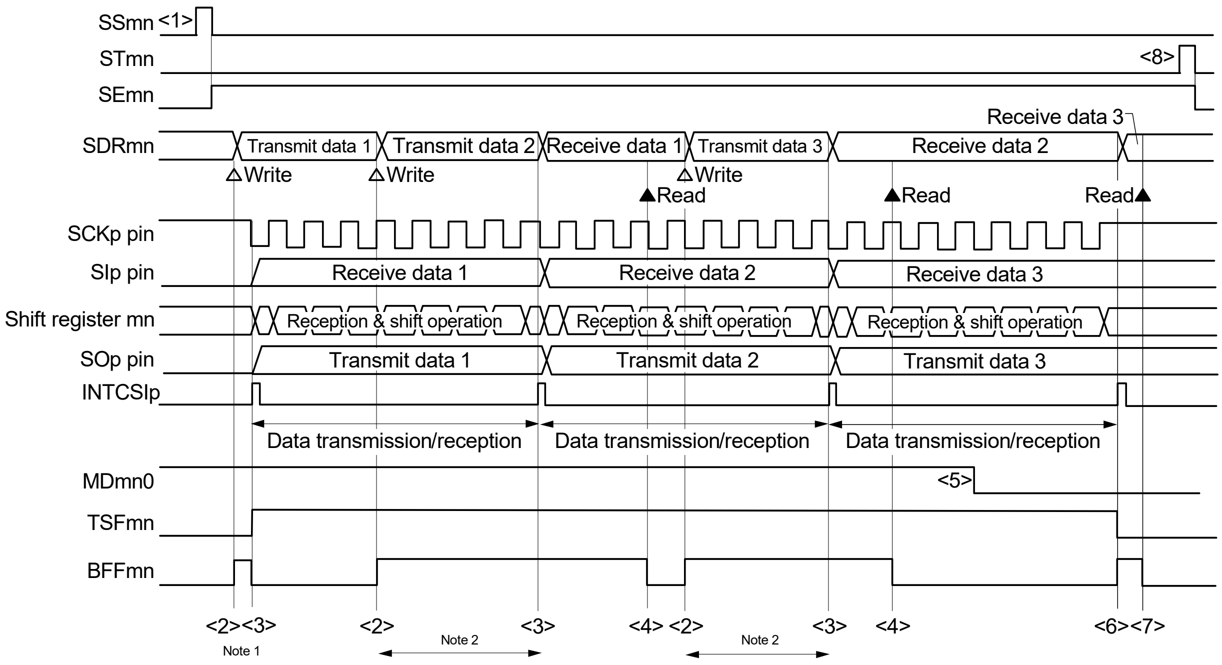
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11), mn = 00 to 03

Figure 13 - 46 Flowchart of Master Transmission/Reception (in Single- Transmission/Reception Mode)


Main routine
Starting simplified SPI (CSI) communication
SAU default setting
For the initial setting, refer to **Figure 13 - 42**.
(Select transfer end interrupt)
Setting transmission/reception data
Setting storage data and number of data for transmission/reception data (Storage area, Transmission data pointer, Reception data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI)
Writing transmit data to SIOp (= SDRmn [7:0])
Read transmit data from storage area and write it to SIOp. Update transmit data pointer.
Writing to SIOp makes SOp and SCKp signals out (communication starts)
Wait for transmission/ reception completes
When transfer end interrupt is generated, it moves to interrupt processing routine.
Interrupt processing routine
Transfer end interrupt
Read receive data to SIOp (= SDRmn [7:0])
Read receive data then writes to storage area, update receive data pointer
RETI
Main routine
No
Transmission/reception completed?
If there are the next data, it continues
Yes
Disable interrupt (MASK)
Write STmn bit to 1
End of communication

(4) Processing flow (in continuous transmission/reception mode)

Figure 13 - 47 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.

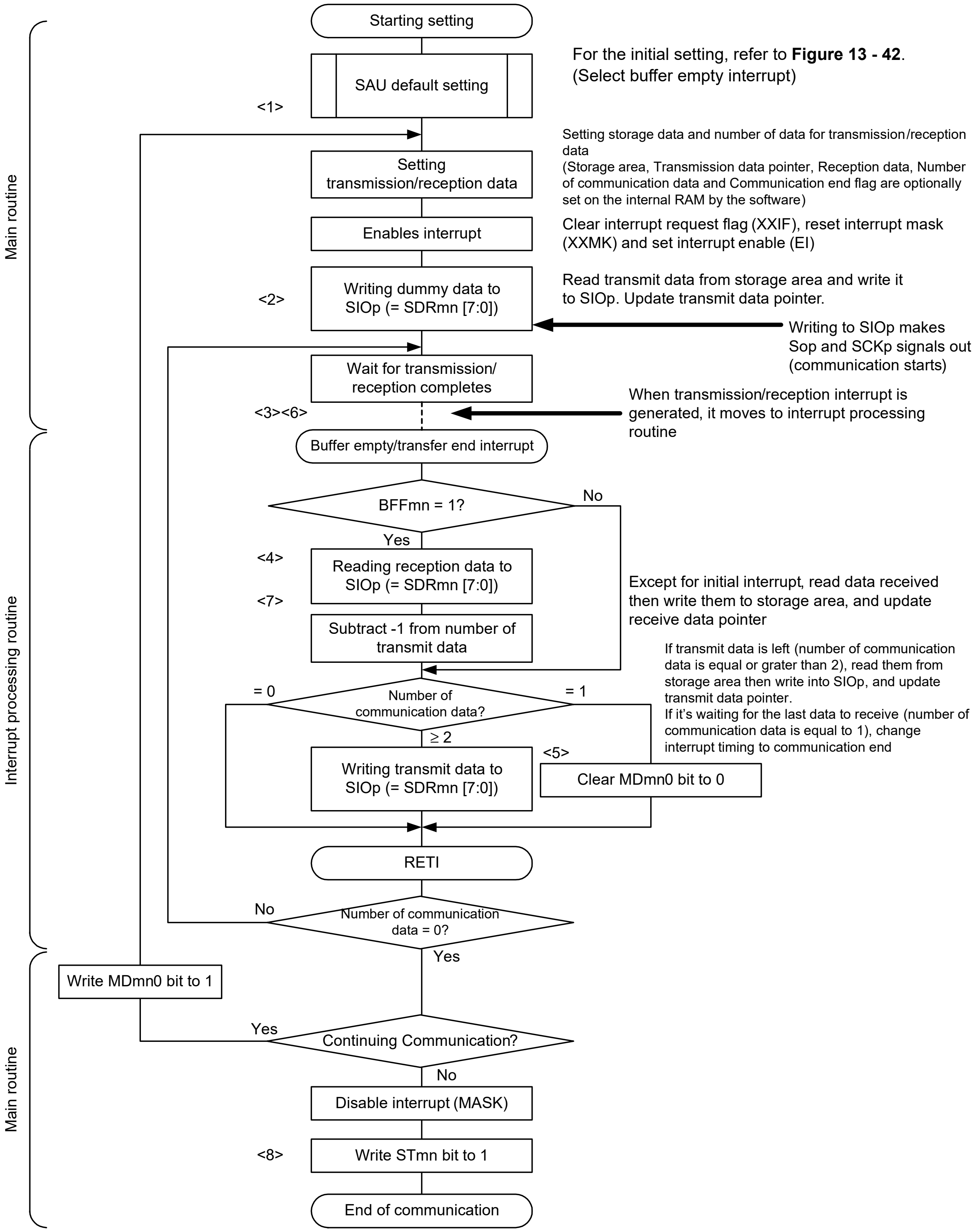
Note 2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

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 13 - 48 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode).

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11), mn = 00 to 03

Figure 13 - 48 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)


Main routine
Interrupt processing routine
Main routine
Starting setting
SAU default setting
For the initial setting, refer to Figure 13 - 42.
(Select buffer empty interrupt)
<1>
Setting transmission/reception data
Setting storage data and number of data for transmission/reception data
(Storage area, Transmission data pointer, Reception data, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI)
<2>
Writing dummy data to SIOp (= SDRmn [7:0])
Read transmit data from storage area and write it to SIOp. Update transmit data pointer.
Writing to SIOp makes Sop and SCKp signals out (communication starts)
Wait for transmission/ reception completes
<3><6>
When transmission/reception interrupt is generated, it moves to interrupt processing routine
Buffer empty/transfer end interrupt
BFFmn = 1?
No
Yes
<4>
Reading reception data to SIOp (= SDRmn [7:0])
Except for initial interrupt, read data received then write them to storage area, and update receive data pointer
<7>
Subtract -1 from number of transmit data
If transmit data is left (number of communication data is equal or grater than 2), read them from storage area then write into SIOp, and update transmit data pointer.
If it's waiting for the last data to receive (number of communication data is equal to 1), change interrupt timing to communication end
= 0
Number of communication data?
= 1
≥ 2
Writing transmit data to SIOp (= SDRmn [7:0])
<5>
Clear MDmn0 bit to 0
RETI
No
Number of communication data = 0?
Yes
Write MDmn0 bit to 1
Yes
Continuing Communication?
No
Disable interrupt (MASK)
<8>
Write STmn bit to 1
End of communication

Remark <1> to <8> in the figure correspond to <1> to <8> in Figure 13 - 47 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

13.5.4 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	CSI10	CSI11
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0
Pins used	SCK00, SO00	SCK01, SO01	SCK10, SO10	SCK11, SO11
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	Overrun error detection flag (OVFmn) only			
Transfer data length	7 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, SCK01, SCK10, and SCK11 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 35**, **CHAPTER 36 ELECTRICAL SPECIFICATIONS**).

Remark 1. f_{MCK}: Operation clock frequency of target channel

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 3), mn = 00 to 03

(1) Register setting

Figure 13 - 49 Example of Contents of Registers for Slave Transmission of Simplified SPI (CSI00, CSI01, CSI10, CSI11)

(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 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 **13.3 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	0	0	0	0	CKOm3 ×	CKOm2 ×	CKOm1 ×	CKOm0 ×	0	0	0	0	SOm3 0/1	SOm2 0/1	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	SOEm3 0/1	SOEm2 0/1	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	SSm3 0/1	SSm2 0/1	SSm1 0/1	SSm0 0/1

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11), mn = 00 to 03

Remark 2. ▣: Setting is fixed in the simplified SPI (CSI) slave transmission mode,

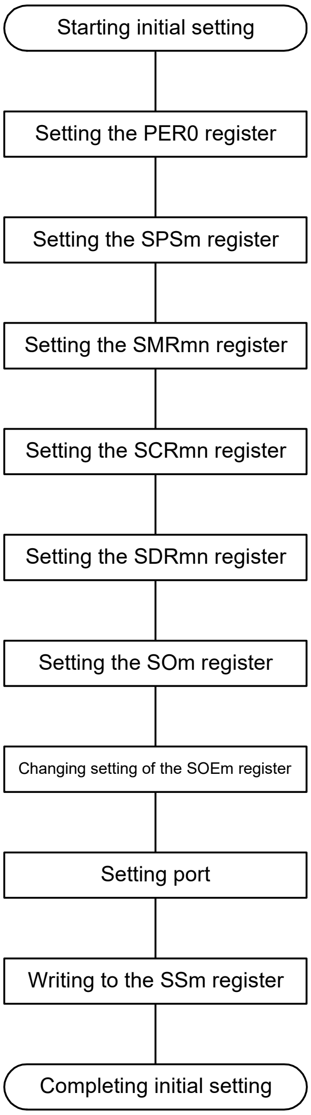
▨: Setting disabled (set to the initial value)

×: 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 13 - 50 Initial Setting Procedure for Slave Transmission


Starting initial setting
Setting the PER0 register
Setting the SPSm register
Setting the SMRmn register
Setting the SCRmn register
Setting the SDRmn register
Setting the SOm register
Changing setting of the SOEm register
Setting port
Writing to the SSm register
Completing initial setting

Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set bits 15 to 9 to 0000000B for baud rate setting.

Set the initial output level of the serial data (SOmn).

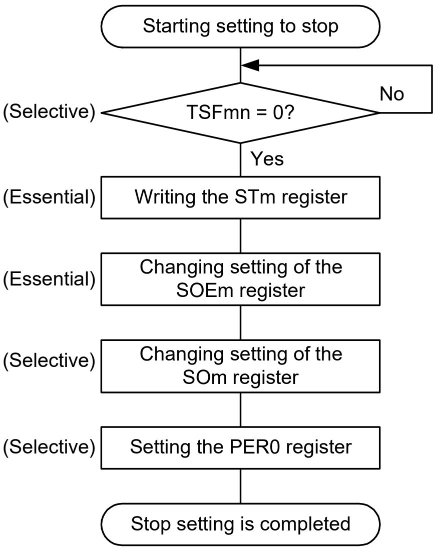
Set the SOEmn bit to 1 and enable data output of the target channel.

Enable data output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation).

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.

Figure 13 - 51 Procedure for Stopping Slave Transmission


Starting setting to stop
(Selective)
TSFmn = 0?
No
Yes
(Essential)
Writing the STm register
(Essential)
Changing setting of the SOEm register
(Selective)
Changing setting of the SOm register
(Selective)
Setting the PER0 register
Stop setting is completed

If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait.)

Write 1 to the STmn bit of the target channel (SEmn = 0: to operation stop status).

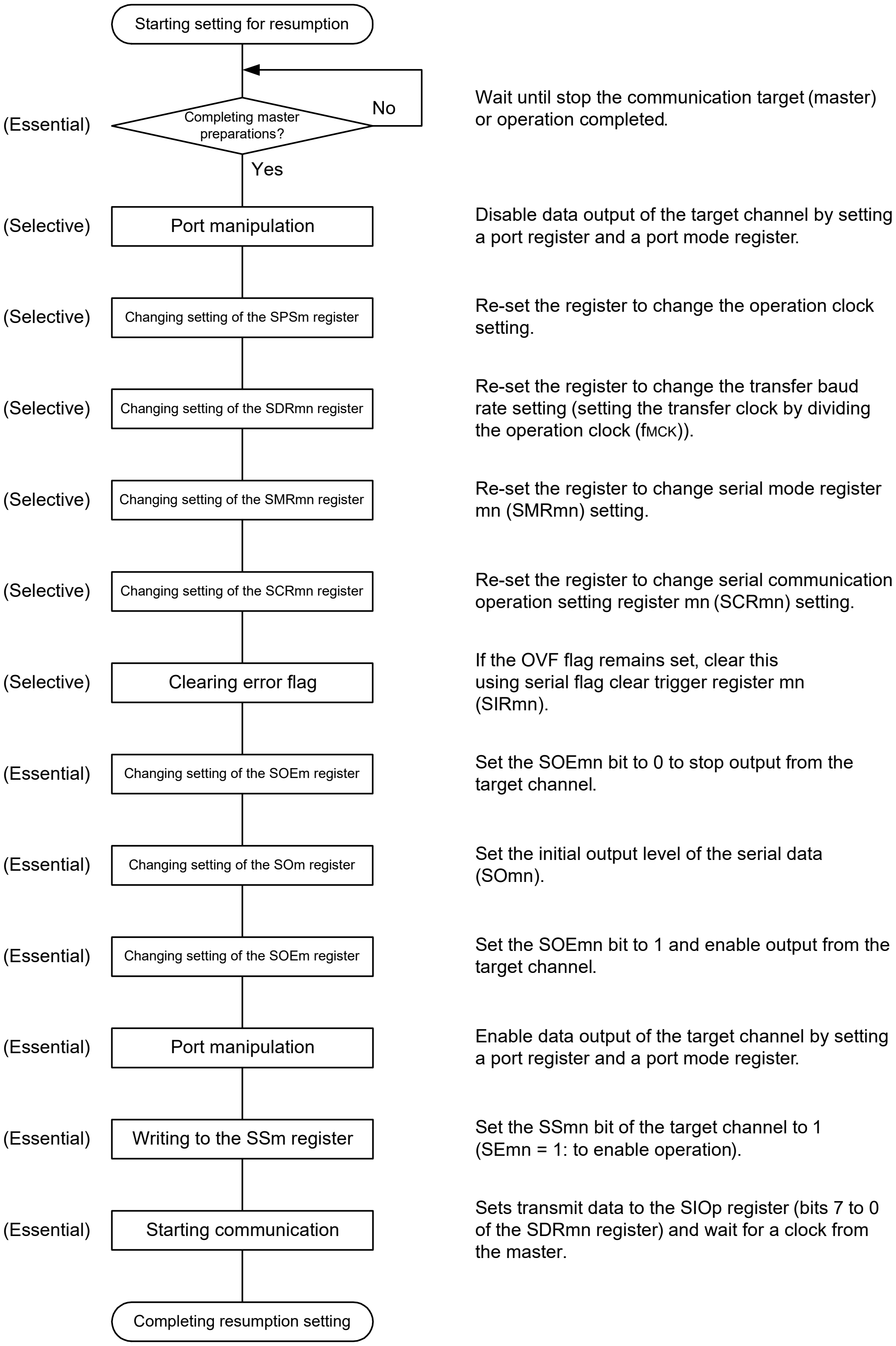
Set the SOEmn bit to 0 and stop the output of the target channel.

The levels of the serial data (SOmn) on the target channel can be changed if necessitated by an emergency.

Reset the serial array unit by stopping the clock supply to it.

After the stop setting is completed, go to the next processing.

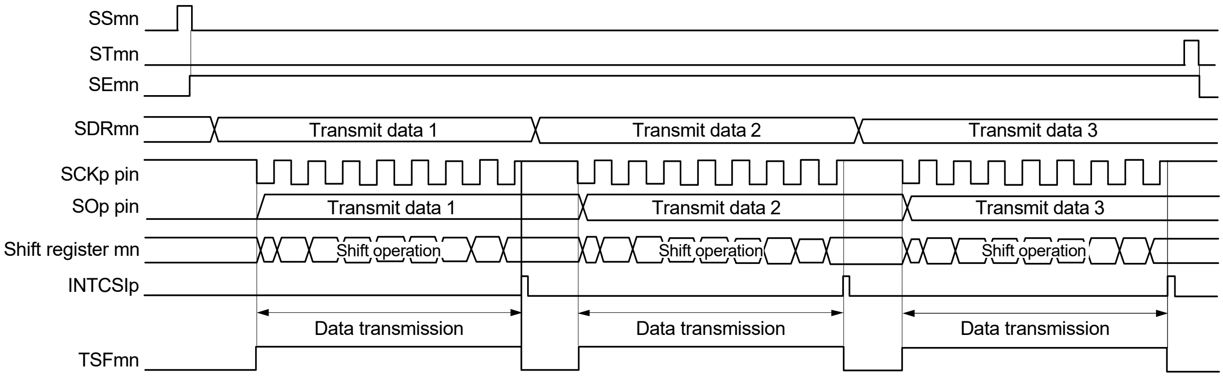
Figure 13 - 52 Procedure for Resuming Slave Transmission


Starting setting for resumption
(Essential)
Completing master preparations?
No
Yes
Wait until stop the communication target (master) or operation completed.
(Selective)
Port manipulation
Disable data output of the target channel by setting a port register and a port mode register.
(Selective)
Changing setting of the SPSm register
Re-set the register to change the operation clock setting.
(Selective)
Changing setting of the SDRmn register
Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (fMCK)).
(Selective)
Changing setting of the SMRmn register
Re-set the register to change serial mode register mn (SMRmn) setting.
(Selective)
Changing setting of the SCRmn register
Re-set the register to change serial communication operation setting register mn (SCRmn) setting.
(Selective)
Clearing error flag
If the OVF flag remains set, clear this using serial flag clear trigger register mn (SIRmn).
(Essential)
Changing setting of the SOEm register
Set the SOEmn bit to 0 to stop output from the target channel.
(Essential)
Changing setting of the SOm register
Set the initial output level of the serial data (SOmn).
(Essential)
Changing setting of the SOEm register
Set the SOEmn bit to 1 and enable output from the target channel.
(Essential)
Port manipulation
Enable data output of the target channel by setting a port register and a port mode register.
(Essential)
Writing to the SSm register
Set the SSmn bit of the target channel to 1 (SEmn = 1: to enable operation).
(Essential)
Starting communication
Sets transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and wait for a clock from the master.
Completing resumption setting

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.

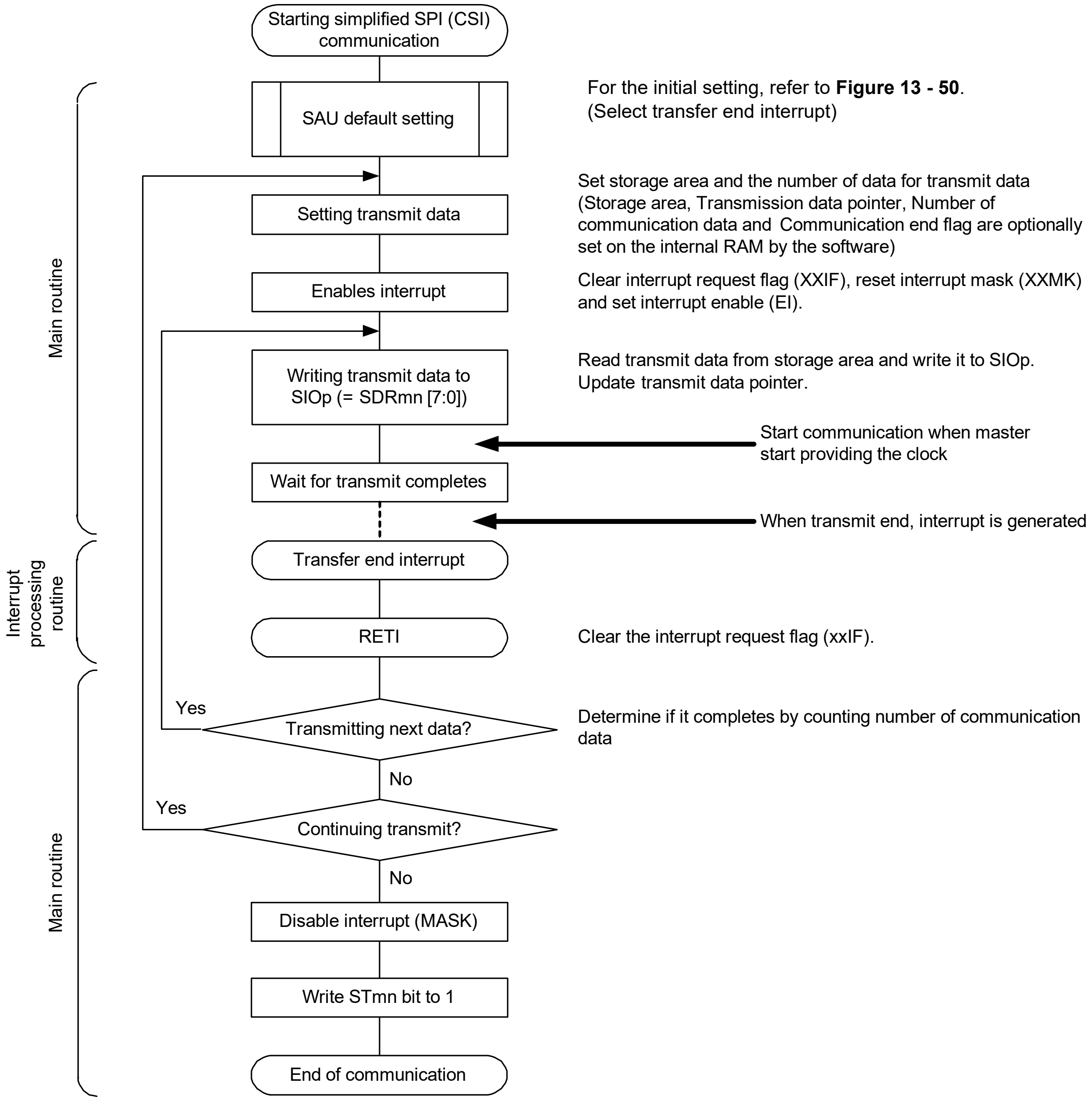
(3) Processing flow (in single-transmission mode)

Figure 13 - 53 Timing Chart of Slave Transmission (in Single-Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)



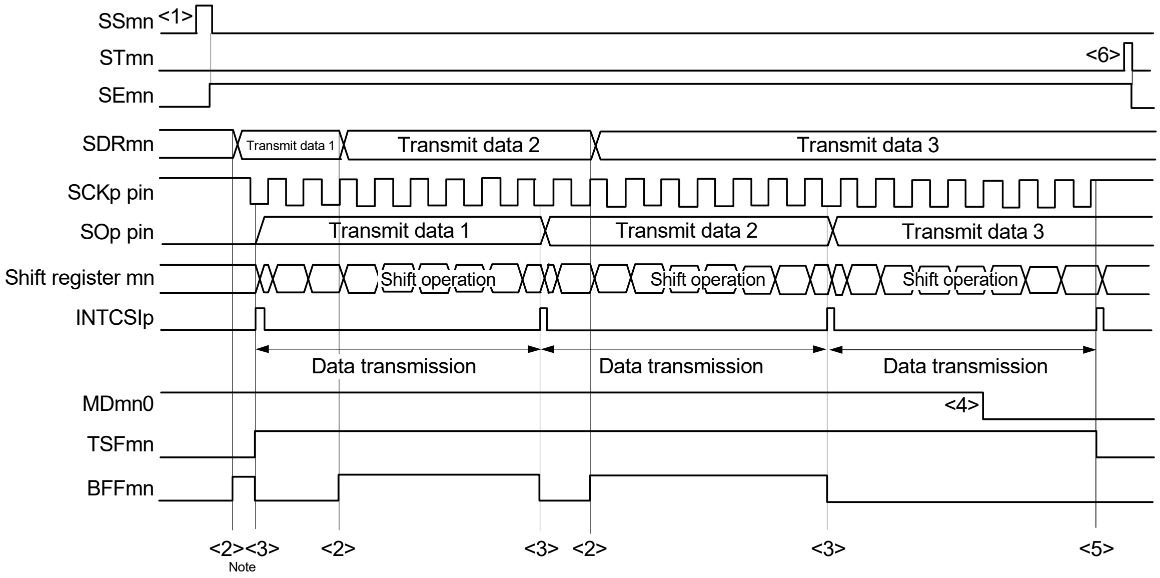
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11), mn = 00 to 03

Figure 13 - 54 Flowchart of Slave Transmission (in Single-Transmission Mode)


Starting simplified SPI (CSI) communication
Main routine
SAU default setting
For the initial setting, refer to **Figure 13 - 50**.
(Select transfer end interrupt)
Setting transmit data
Set storage area and the number of data for transmit data (Storage area, Transmission data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI).
Writing transmit data to SIOp (= SDRmn [7:0])
Read transmit data from storage area and write it to SIOp. Update transmit data pointer.
Start communication when master start providing the clock
Wait for transmit completes
When transmit end, interrupt is generated
Interrupt processing routine
Transfer end interrupt
RETI
Clear the interrupt request flag (xxIF).
Yes
Transmitting next data?
Determine if it completes by counting number of communication data
No
Yes
Continuing transmit?
No
Main routine
Disable interrupt (MASK)
Write STmn bit to 1
End of communication

(4) Processing flow (in continuous transmission mode)

Figure 13 - 55 Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)

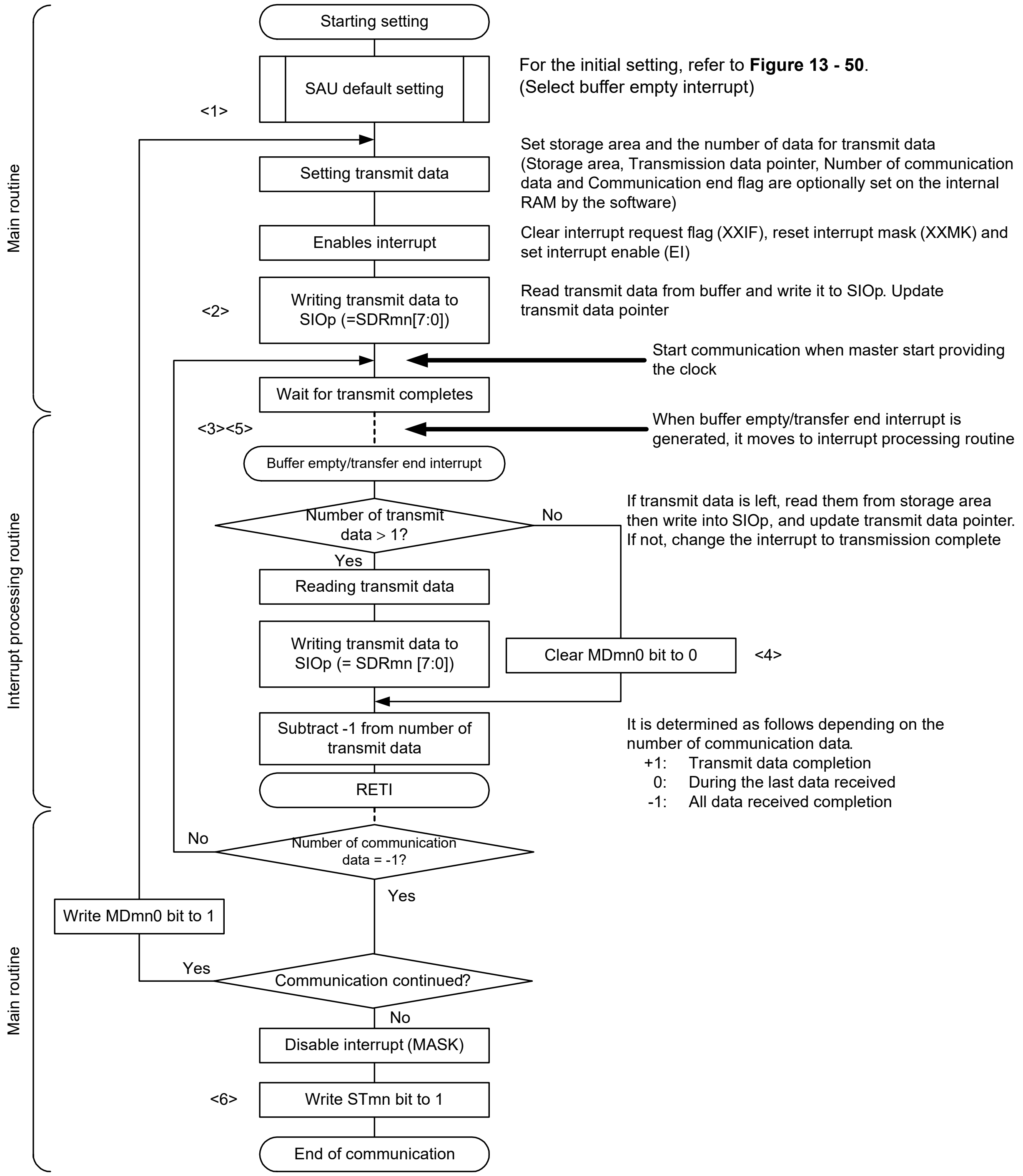


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11), mn = 00 to 03

Figure 13 - 56 Flowchart of Slave Transmission (in Continuous Transmission Mode)



For the initial setting, refer to **Figure 13 - 50**.
(Select buffer empty interrupt)

Set storage area and the number of data for transmit data (Storage area, Transmission data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)

Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI)

Read transmit data from buffer and write it to SIOp. Update transmit data pointer

Start communication when master start providing the clock

When buffer empty/transfer end interrupt is generated, it moves to interrupt processing routine

If transmit data is left, read them from storage area then write into SIOp, and update transmit data pointer. If not, change the interrupt to transmission complete

It is determined as follows depending on the number of communication data.

+1:	Transmit data completion
0:	During the last data received
-1:	All data received completion

Remark <1> to <6> in the figure correspond to <1> to <6> in Figure 13 - 55 Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0).

13.5.5 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	CSI10	CSI11
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0
Pins used	SCK00, SI00	SCK01, SI01	SCK10, SI10	SCK11, SI11
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error detection flag	Overrun error detection flag (OVFmn) only			
Transfer data length	7 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, SCK01, SCK10, and SCK11 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 35**, **CHAPTER 36 ELECTRICAL SPECIFICATIONS**).

Remark 1. f_{MCK}: Operation clock frequency of target channel

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 3), mn = 00 to 03

(1) Register setting

Figure 13 - 57 Example of Contents of Registers for Slave Reception of Simplified SPI (CSI00, CSI01, CSI10, CSI11)

(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 **13.3 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	CKOm3 ×	CKOm2 ×	CKOm1 ×	CKOm0 ×	0	0	0	0	SOm3 ×	SOm2 ×	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	SOEm3 ×	SOEm2 ×	SOEm1 ×	SOEm0 ×

(f) Serial channel start register m (SSm)... Sets only the bits of the target channel to 1.

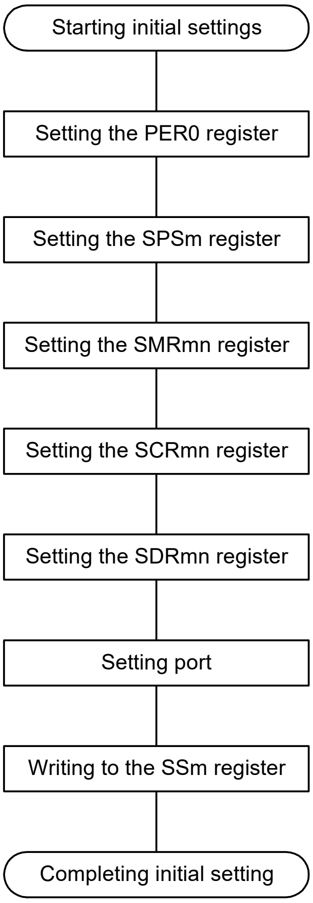
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 0/1	SSm1 0/1	SSm0 0/1

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11), mn = 00 to 03

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 13 - 58 Initial Setting Procedure for Slave Reception


Starting initial settings
Setting the PER0 register
Setting the SPSm register
Setting the SMRmn register
Setting the SCRmn register
Setting the SDRmn register
Setting port
Writing to the SSm register
Completing initial setting

Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

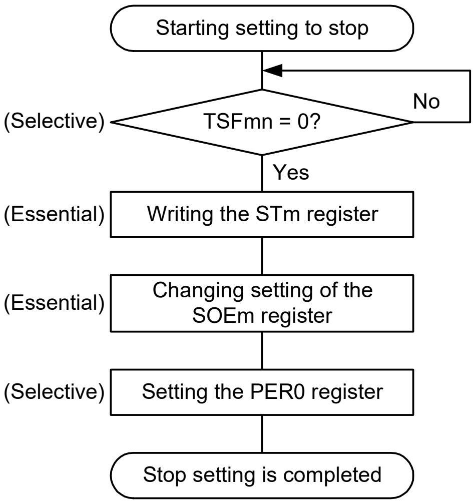
Set a communication format.

Set bits 15 to 9 to 0000000B for baud rate setting.

Enable data input and clock input of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation). Wait for a clock from the master.

Figure 13 - 59 Procedure for Stopping Slave Reception



If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait.)

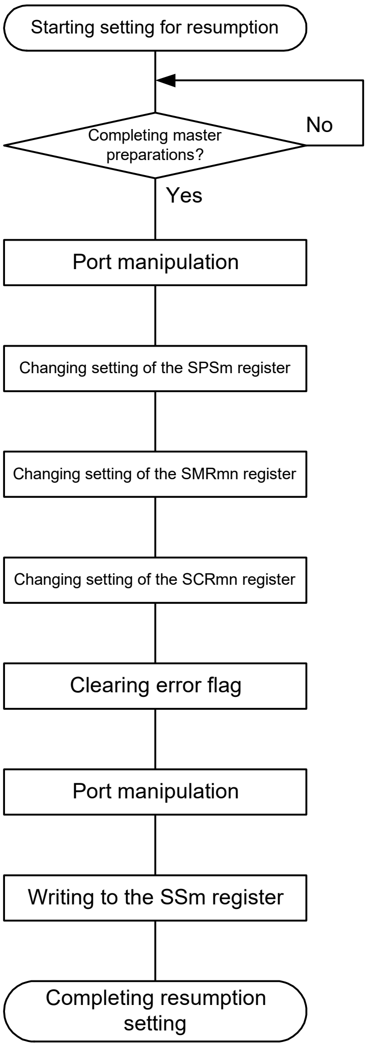
Write 1 to the STmn bit of the target channel.
(SEmn = 0: to operation stop status)

Set the SOEmn bit to 0 and stop the output of the target channel.

Reset the serial array unit by stopping the clock supply to it.

After the stop setting is completed, go to the next processing.

Figure 13 - 60 Procedure for Resuming Slave Reception


Starting setting for resumption
Completing master preparations?
No
Yes
Port manipulation
Changing setting of the SPSm register
Changing setting of the SMRmn register
Changing setting of the SCRmn register
Clearing error flag
Port manipulation
Writing to the SSm register
Completing resumption setting

(Essential) Wait until stop the communication target (master) or operation completed.

(Essential) Disable clock output of the target channel by setting a port register and a port mode register.

(Selective) Re-set the register to change the operation clock setting.

(Selective) Re-set the register to change serial mode register mn (SMRmn) setting.

(Selective) Re-set the register to change serial communication operation setting register mn (SCRmn) setting.

(Selective) If the OVF flag remains set, clear this using serial flag clear trigger register mn (SIRmn).

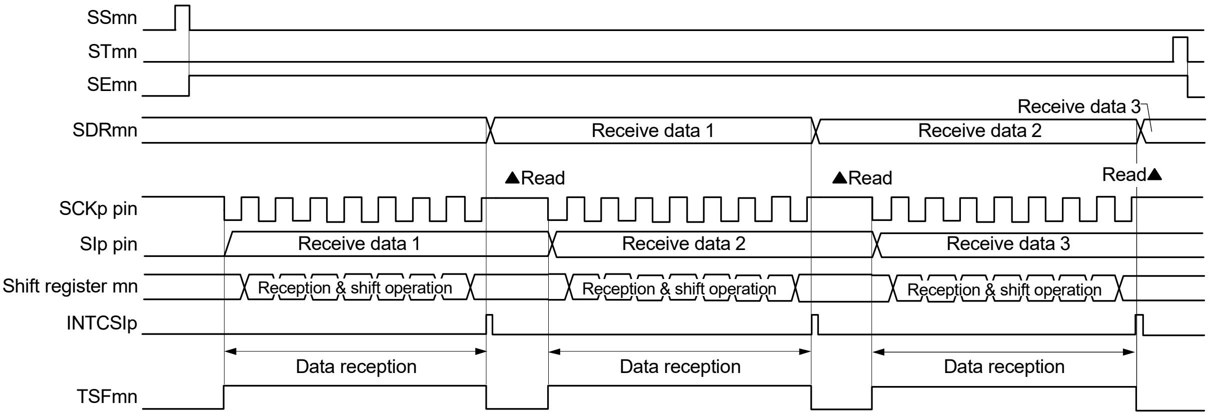
(Essential) Enable clock output of the target channel by setting a port register and a port mode register.

(Essential) Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation). Wait for a clock from the master.

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.

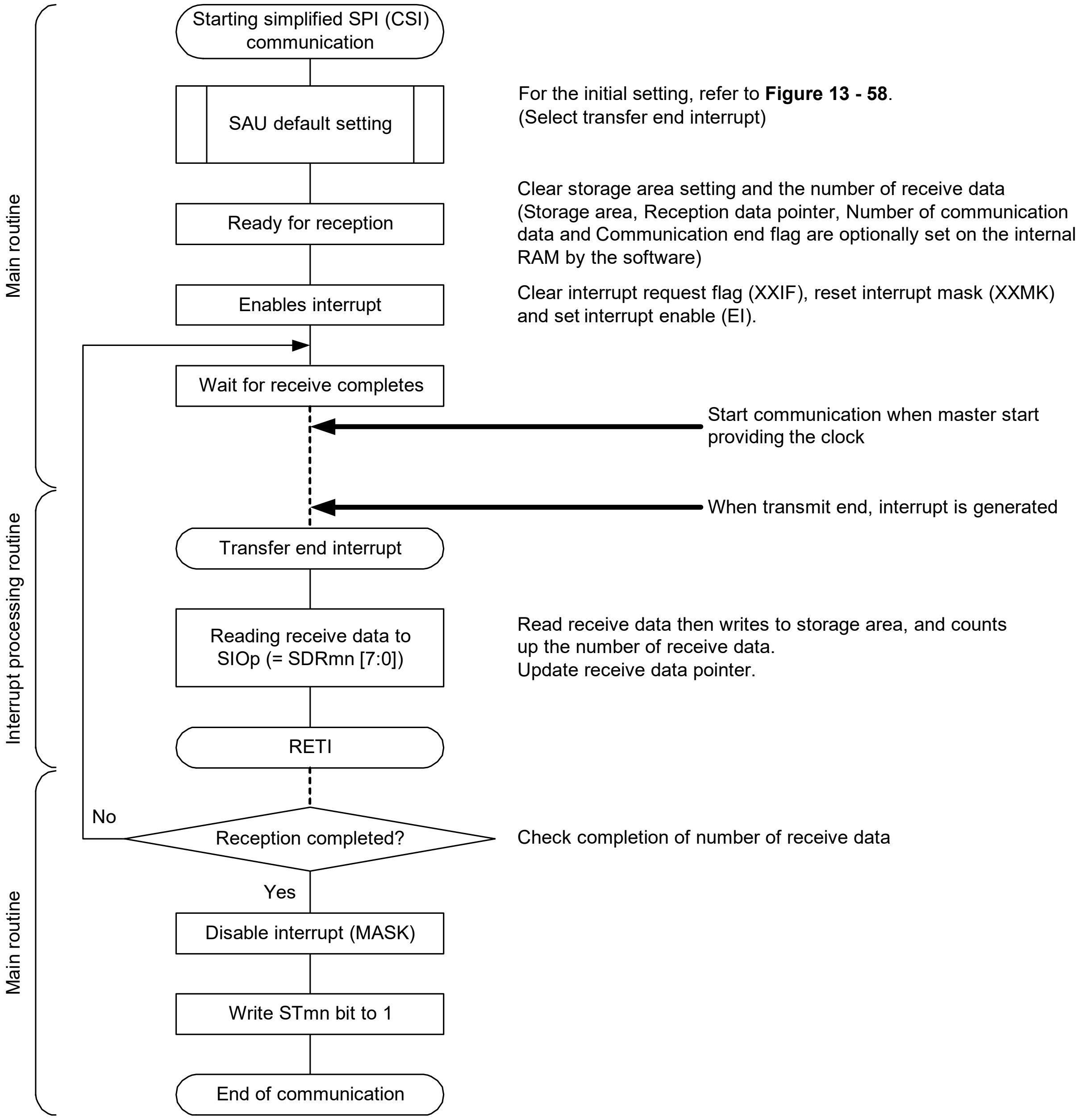
(3) Processing flow (in single-reception mode)

Figure 13 - 61 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 to 3), p: CSI number (p = 00, 01, 10, 11), mn = 00 to 03

Figure 13 - 62 Flowchart of Slave Reception (in Single-Reception Mode)


Main routine
Starting simplified SPI (CSI) communication
SAU default setting
For the initial setting, refer to **Figure 13 - 58**. (Select transfer end interrupt)
Ready for reception
Clear storage area setting and the number of receive data (Storage area, Reception data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI).
Wait for receive completes
Start communication when master start providing the clock
When transmit end, interrupt is generated
Interrupt processing routine
Transfer end interrupt
Reading receive data to SIOp (= SDRmn [7:0])
Read receive data then writes to storage area, and counts up the number of receive data.
Update receive data pointer.
RETI
Main routine
No
Reception completed?
Check completion of number of receive data
Yes
Disable interrupt (MASK)
Write STmn bit to 1
End of communication

13.5.6 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	CSI11	CSI11
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01	SCK10, SI10, SO10	SCK11, SI11, SO11
Interrupt	INTCSI00	INTCSI01	INTCSI10	INTCSI11
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	Overrun error detection flag (OVFmn) only			
Transfer data length	7 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, SCK01, SCK10, and SCK11 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 35**, **CHAPTER 36 ELECTRICAL SPECIFICATIONS**).

Remark 1. f_{MCK}: Operation clock frequency of target channel

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 3), mn = 00 to 03

(1) Register setting

Figure 13 - 63 Example of Contents of Registers for Slave Transmission/Reception of Simplified SPI (CSI00, CSI01, CSI10, CSI11)

(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 **13.3 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							

(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	CKOm3 ×	CKOm2 ×	CKOm1 ×	CKOm0 ×	0	0	0	0	SOm3 0/1	SOm2 0/1	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	SOEm3 0/1	SOEm2 0/1	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	SSm3 0/1	SSm2 0/1	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 to 3), p: CSI number (p = 00, 01, 10, 11), mn = 00 to 03

Remark 2. ▣: Setting is fixed in the simplified SPI (CSI) master transmission/reception mode

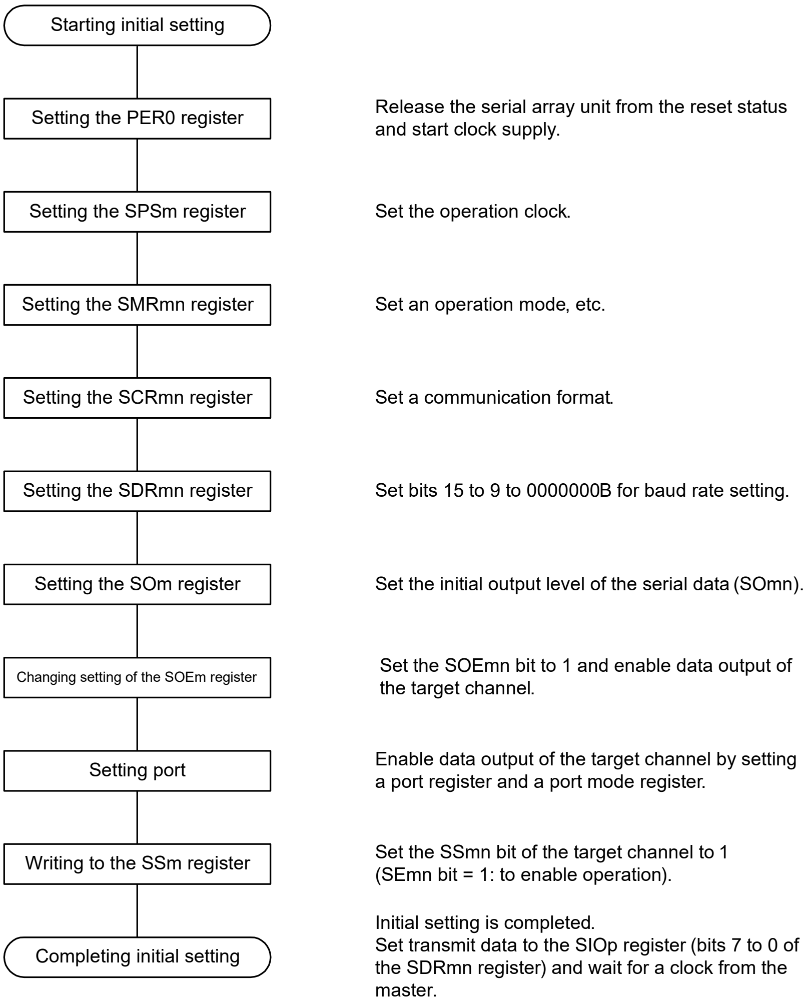
▨: Setting disabled (set to the initial value)

×: 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 13 - 64 Initial Setting Procedure for Slave Transmission/Reception


Starting initial setting
Setting the PER0 register
Setting the SPSm register
Setting the SMRmn register
Setting the SCRmn register
Setting the SDRmn register
Setting the SOm register
Changing setting of the SOEm register
Setting port
Writing to the SSm register
Completing initial setting

Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set bits 15 to 9 to 0000000B for baud rate setting.

Set the initial output level of the serial data (SOmn).

Set the SOEmn bit to 1 and enable data output of the target channel.

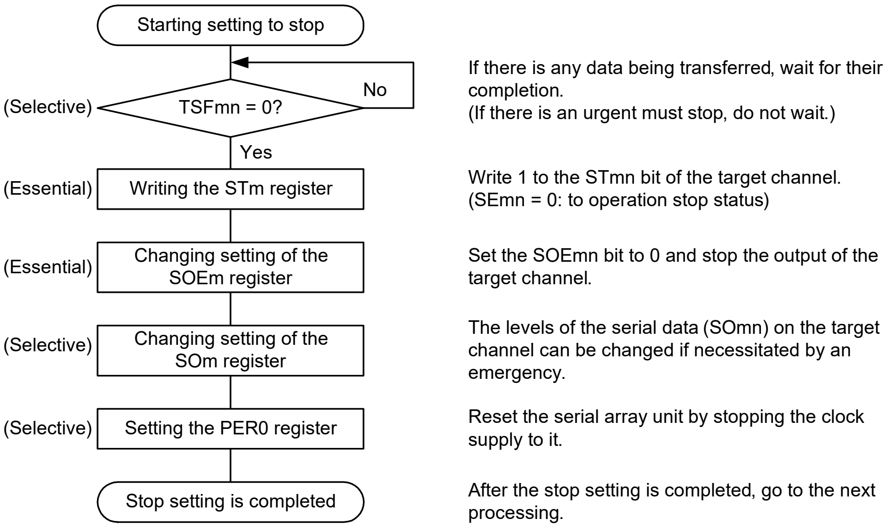
Enable data output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation).

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.

Figure 13 - 65 Procedure for Stopping Slave Transmission/Reception


Starting setting to stop
(Selective)
TSFmn = 0?
No
Yes
(Essential)
Writing the STm register
(Essential)
Changing setting of the SOEm register
(Selective)
Changing setting of the SOm register
(Selective)
Setting the PER0 register
Stop setting is completed

If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait.)

Write 1 to the STmn bit of the target channel.
(SEmn = 0: to operation stop status)

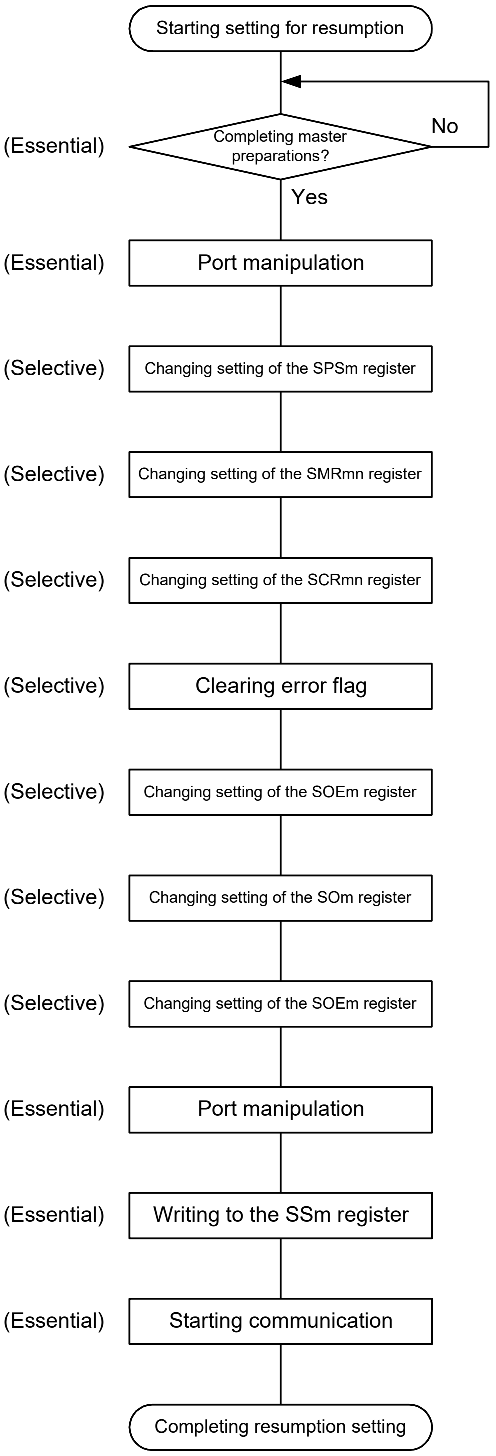
Set the SOEmn bit to 0 and stop the output of the target channel.

The levels of the serial data (SOmn) on the target channel can be changed if necessitated by an emergency.

Reset the serial array unit by stopping the clock supply to it.

After the stop setting is completed, go to the next processing.

Figure 13 - 66 Procedure for Resuming Slave Transmission/Reception


Starting setting for resumption
(Essential)
Completing master preparations?
No
Yes
(Essential)
Port manipulation
(Selective)
Changing setting of the SPSm register
(Selective)
Changing setting of the SMRmn register
(Selective)
Changing setting of the SCRmn register
(Selective)
Clearing error flag
(Selective)
Changing setting of the SOEm register
(Selective)
Changing setting of the SOm register
(Selective)
Changing setting of the SOEm register
(Essential)
Port manipulation
(Essential)
Writing to the SSm register
(Essential)
Starting communication
Completing resumption setting

Wait until stop the communication target (master) or operation completed.

Disable data output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change serial communication operation setting register mn (SCRmn) setting.

If the OVF flag remains set, clear this using serial flag clear trigger register mn (SIRmn).

Set the SOEmn bit to 0 to stop output from the target channel.

Set the initial output level of the serial data (SOmn).

Set the SOEmn bit to 1 and enable output from the target channel.

Enable data output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 (SEmn = 1: to enable operation).

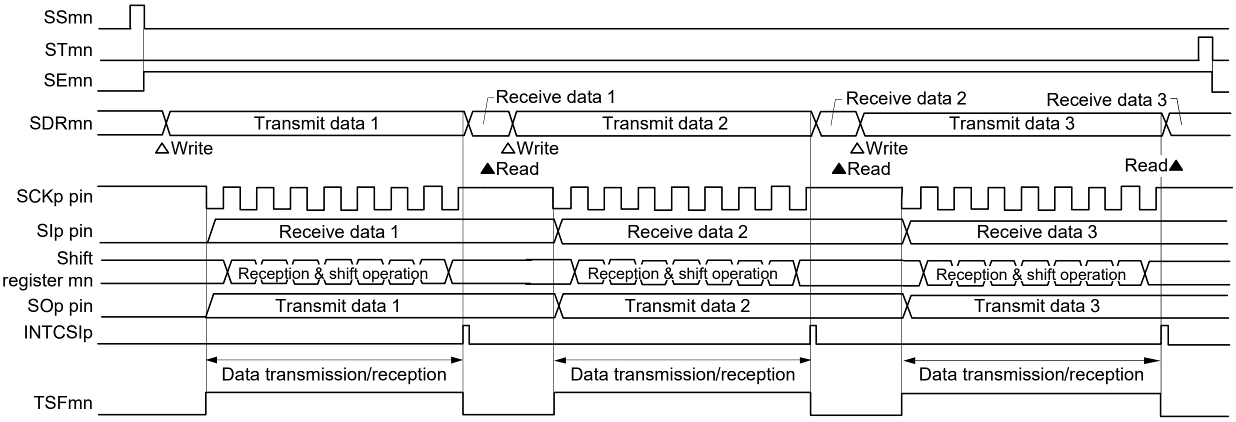
Sets transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and wait for a clock from the master.

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.

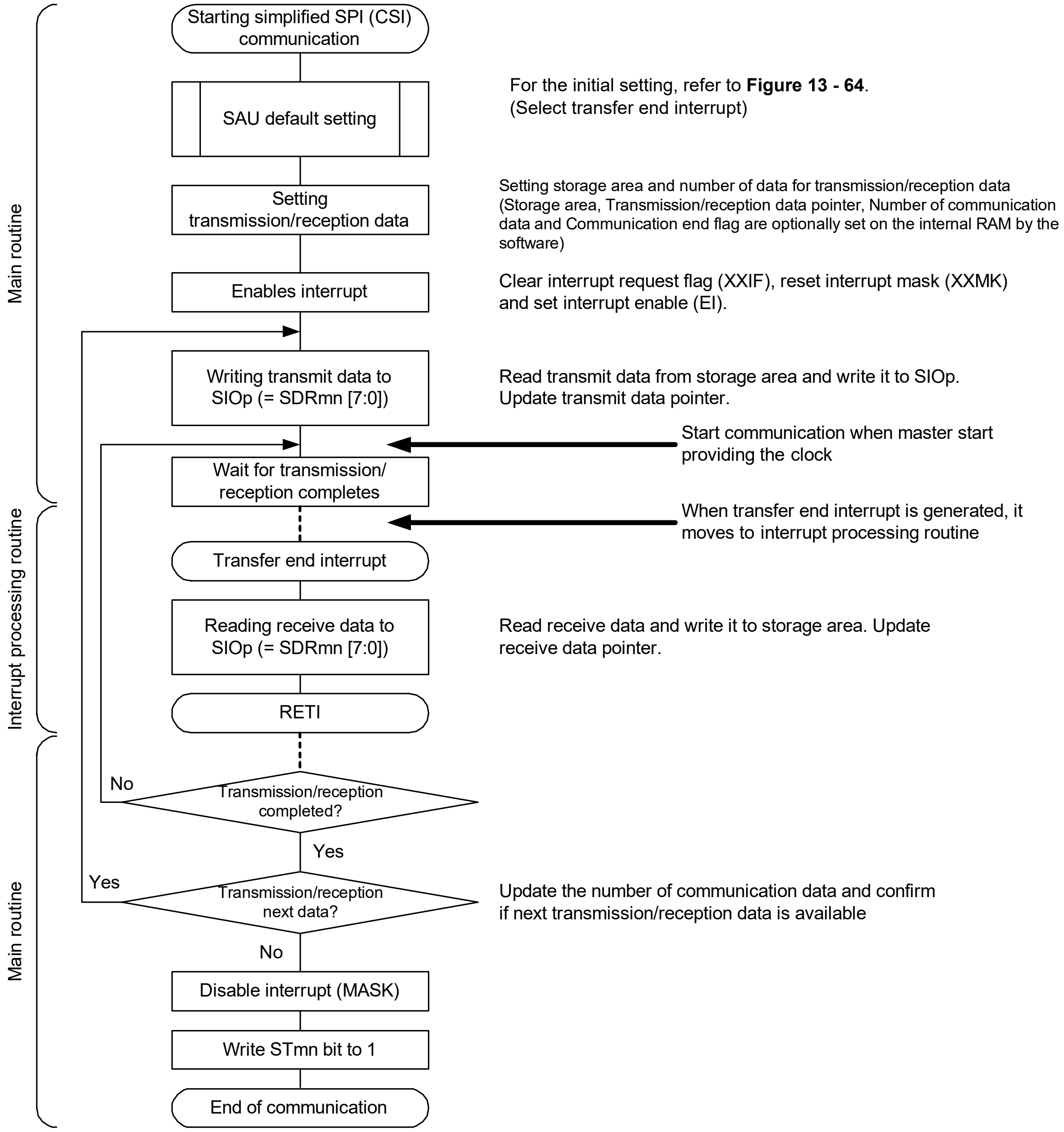
(3) Processing flow (in single-transmission/reception mode)

Figure 13 - 67 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 to 3), p: CSI number (p = 00, 01, 10, 11), mn = 00 to 03

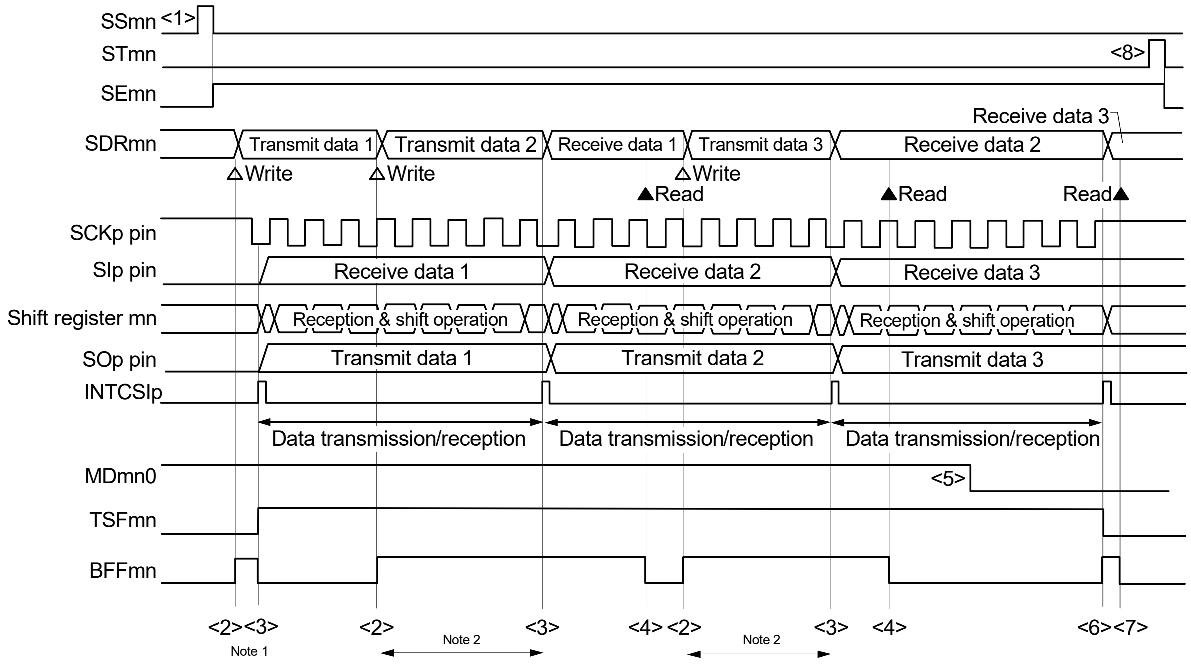
Figure 13 - 68 Flowchart of Slave Transmission/Reception (in Single- Transmission/Reception Mode)


Main routine
Starting simplified SPI (CSI) communication
SAU default setting
For the initial setting, refer to **Figure 13 - 64**.
(Select transfer end interrupt)
Setting transmission/reception data
Setting storage area and number of data for transmission/reception data (Storage area, Transmission/reception data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI).
Writing transmit data to SIOp (= SDRmn [7:0])
Read transmit data from storage area and write it to SIOp. Update transmit data pointer.
Wait for transmission/ reception completes
Start communication when master start providing the clock
When transfer end interrupt is generated, it moves to interrupt processing routine
Interrupt processing routine
Transfer end interrupt
Reading receive data to SIOp (= SDRmn [7:0])
Read receive data and write it to storage area. Update receive data pointer.
RETI
Main routine
No
Transmission/reception completed?
Yes
Yes
Transmission/reception next data?
Update the number of communication data and confirm if next transmission/reception data is available
No
Disable interrupt (MASK)
Write STmn bit to 1
End of communication

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 13 - 69 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.

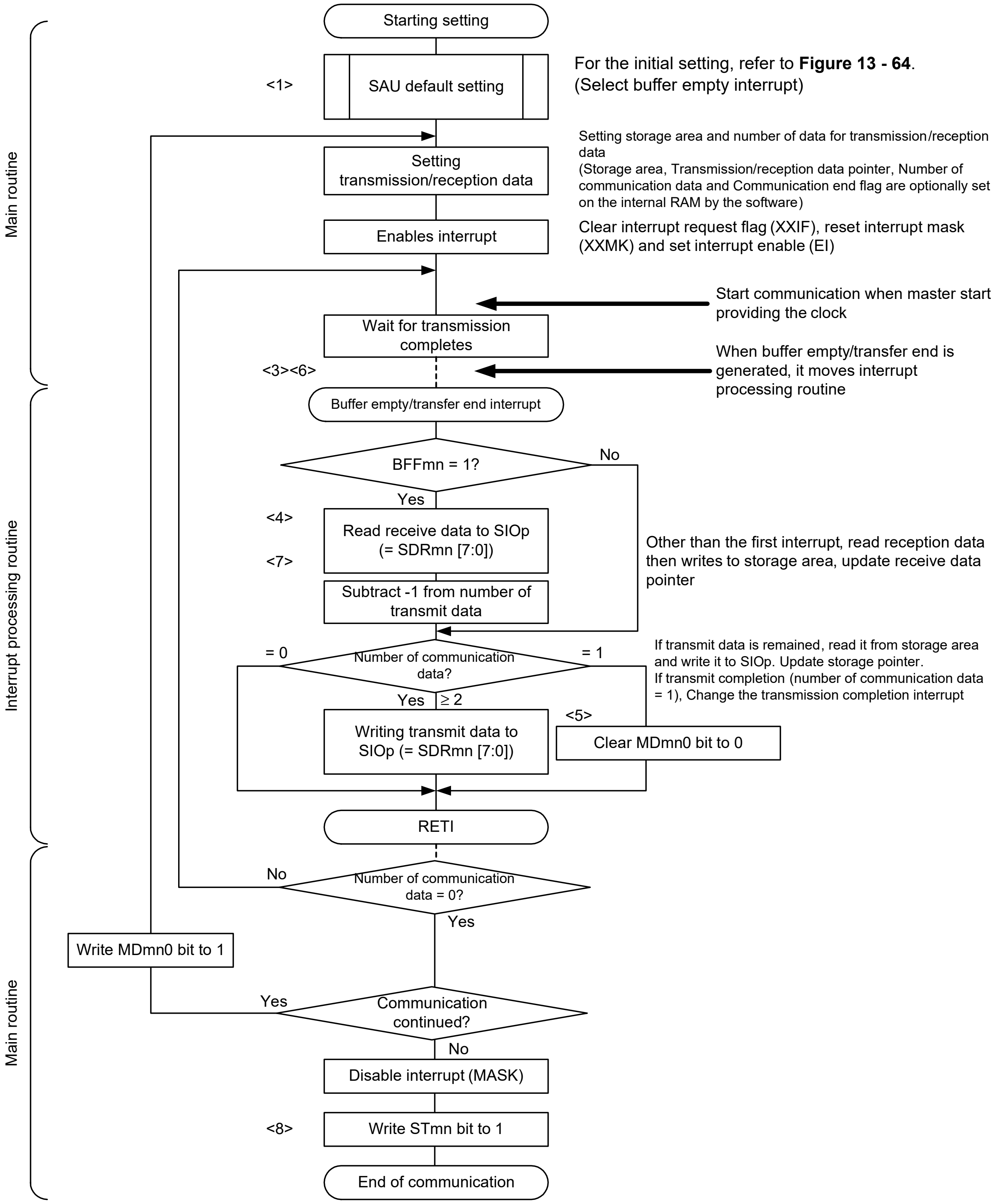
Note 2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

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 13 - 70 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode).

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01, 10, 11), mn = 00 to 03

Figure 13 - 70 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)


Main routine
Interrupt processing routine
Main routine
Starting setting
<1>
SAU default setting
For the initial setting, refer to Figure 13 - 64.
(Select buffer empty interrupt)
Setting transmission/reception data
Setting storage area and number of data for transmission/reception data
(Storage area, Transmission/reception data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI)
Start communication when master start providing the clock
Wait for transmission completes
When buffer empty/transfer end is generated, it moves interrupt processing routine
<3><6>
Buffer empty/transfer end interrupt
BFFmn = 1?
No
Yes
<4>
Read receive data to SIOp (= SDRmn [7:0])
<7>
Subtract -1 from number of transmit data
Other than the first interrupt, read reception data then writes to storage area, update receive data pointer
= 0
Number of communication data?
= 1
If transmit data is remained, read it from storage area and write it to SIOp. Update storage pointer.
If transmit completion (number of communication data = 1), Change the transmission completion interrupt
Yes
≥ 2
Writing transmit data to SIOp (= SDRmn [7:0])
<5>
Clear MDmn0 bit to 0
RETI
No
Number of communication data = 0?
Yes
Write MDmn0 bit to 1
Yes
Communication continued?
No
Disable interrupt (MASK)
<8>
Write STmn bit to 1
End of communication

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 13 - 69 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

13.5.7 SNOOZE mode function

SNOOZE mode makes simplified SPI (CSI) operate reception by SCKp pin input detection while the STOP mode. Normally simplified SPI (CSI) stops communication in the STOP mode. But, using the SNOOZE mode makes reception simplified SPI (CSI) operate unless the CPU operation by detecting SCKp pin input. Only the following channels can be set to the SNOOZE mode.

- CSI00

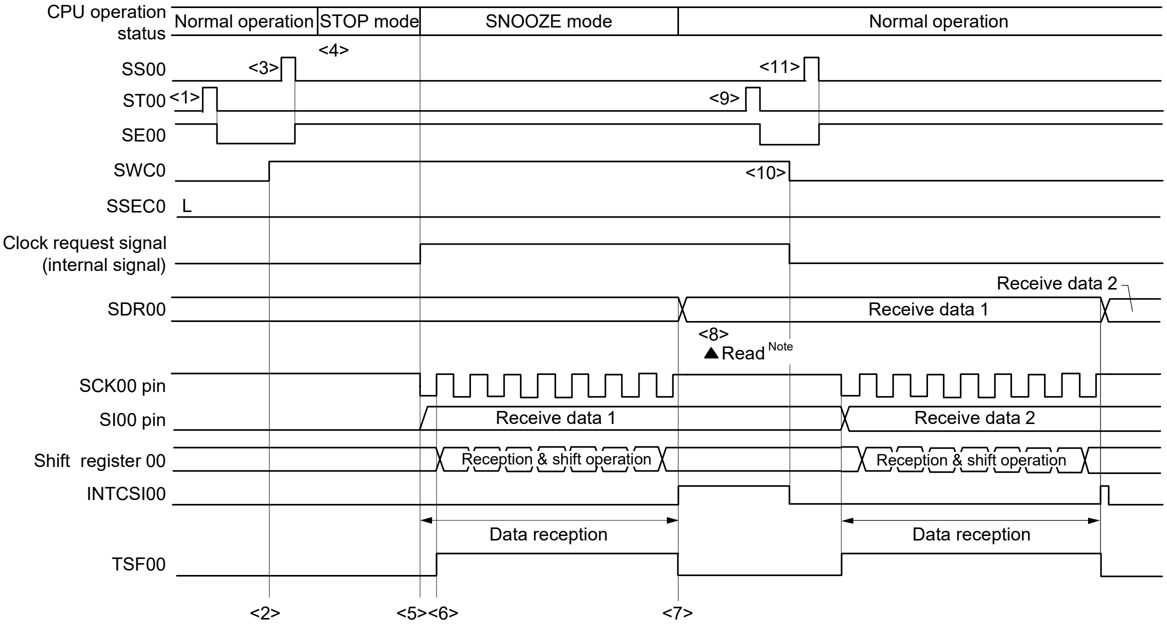
When using the simplified SPI (CSI) in SNOOZE mode, make the following setting before switching to the STOP mode (see **Figure 13 - 72** and **Figure 13 - 74 Flowchart of SNOOZE Mode Operation**).

- 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 (f_{IH}) or middle-speed on-chip oscillator clock (f_{IM}) is selected for f_{CLK}.
Caution 2. The maximum transfer rate when using CSIp in the SNOOZE mode is 1 Mbps.

(1) SNOOZE mode operation (once startup)

Figure 13 - 71 Timing Chart of SNOOZE Mode Operation (once startup) (Type 1: DAPmn = 0, CKPmn = 0)

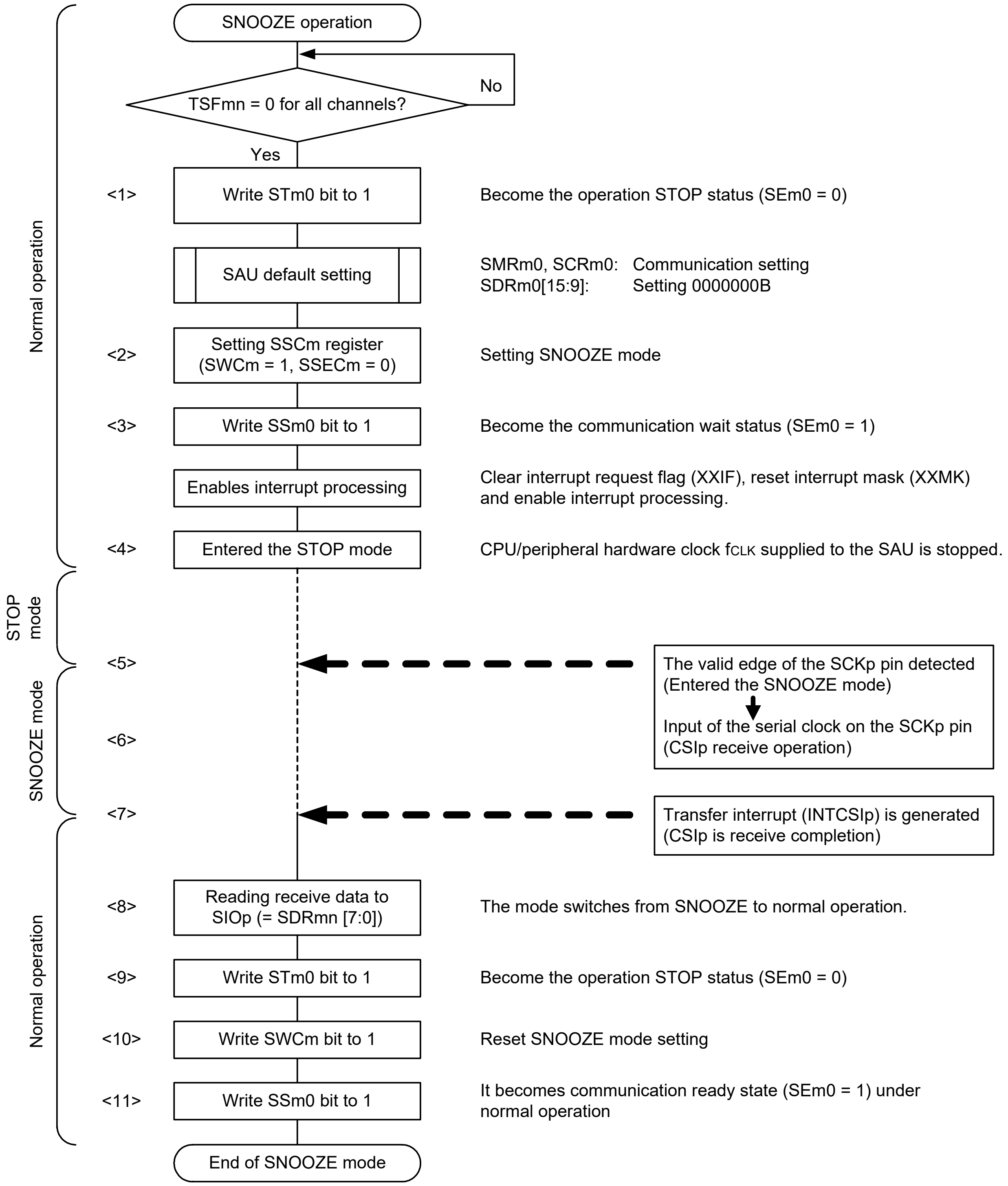


Note Only read received data while SWCm = 1 and before the next valid edge of the SCKp pin input is detected.

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 13 - 72 Flowchart of SNOOZE Mode Operation (once startup).
Remark 2. m = 0; p = 00

Figure 13 - 72 Flowchart of SNOOZE Mode Operation (once startup)

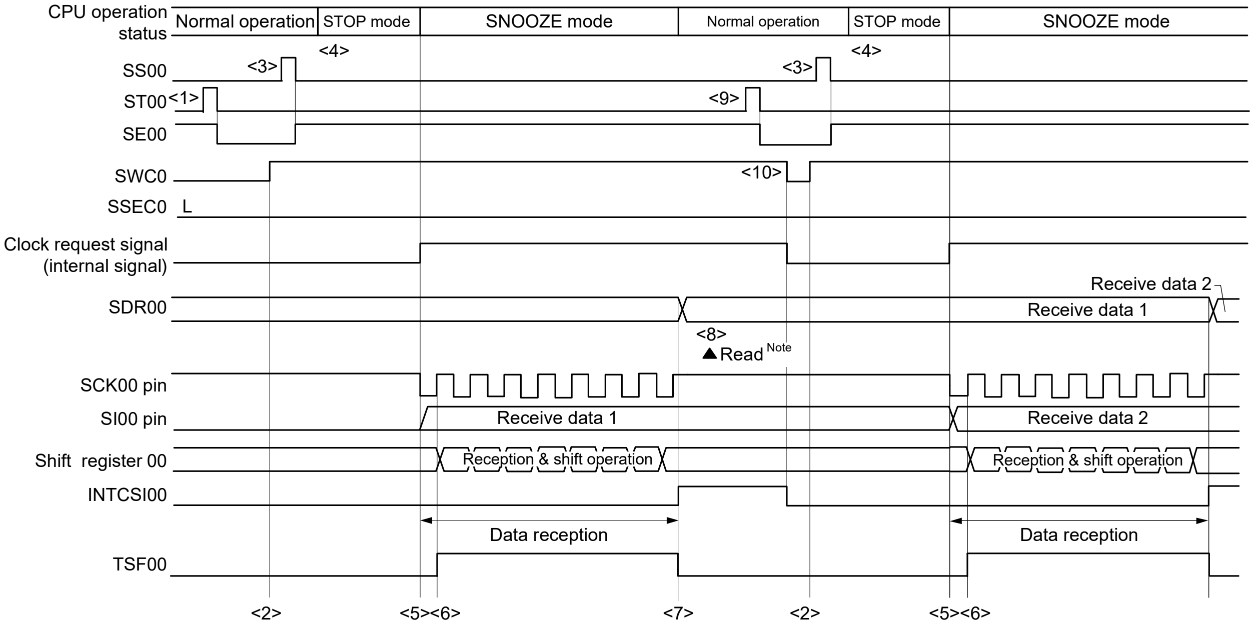

Normal operation
SNOOZE operation
TSFmn = 0 for all channels?
No
Yes
<1>
Write STm0 bit to 1
Become the operation STOP status (SEm0 = 0)
SAU default setting
SMRm0, SCRm0: Communication setting
SDRm0[15:9]: Setting 0000000B
<2>
Setting SSCm register
(SWCm = 1, SSECm = 0)
Setting SNOOZE mode
<3>
Write SSm0 bit to 1
Become the communication wait status (SEm0 = 1)
Enables interrupt processing
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK)
and enable interrupt processing.
<4>
Entered the STOP mode
CPU/peripheral hardware clock fCLK supplied to the SAU is stopped.
STOP mode
<5>
The valid edge of the SCKp pin detected
(Entered the SNOOZE mode)
Input of the serial clock on the SCKp pin
(CSIp receive operation)
SNOOZE mode
<6>
<7>
Transfer interrupt (INTCSIp) is generated
(CSIp is receive completion)
Normal operation
<8>
Reading receive data to
SIOp (= SDRmn [7:0])
The mode switches from SNOOZE to normal operation.
<9>
Write STm0 bit to 1
Become the operation STOP status (SEm0 = 0)
<10>
Write SWCm bit to 1
Reset SNOOZE mode setting
<11>
Write SSm0 bit to 1
It becomes communication ready state (SEm0 = 1) under
normal operation
End of SNOOZE mode

Remark 1. <1> to <11> in the figure correspond to <1> to <11> in Figure 13 - 71 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 13 - 73 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.

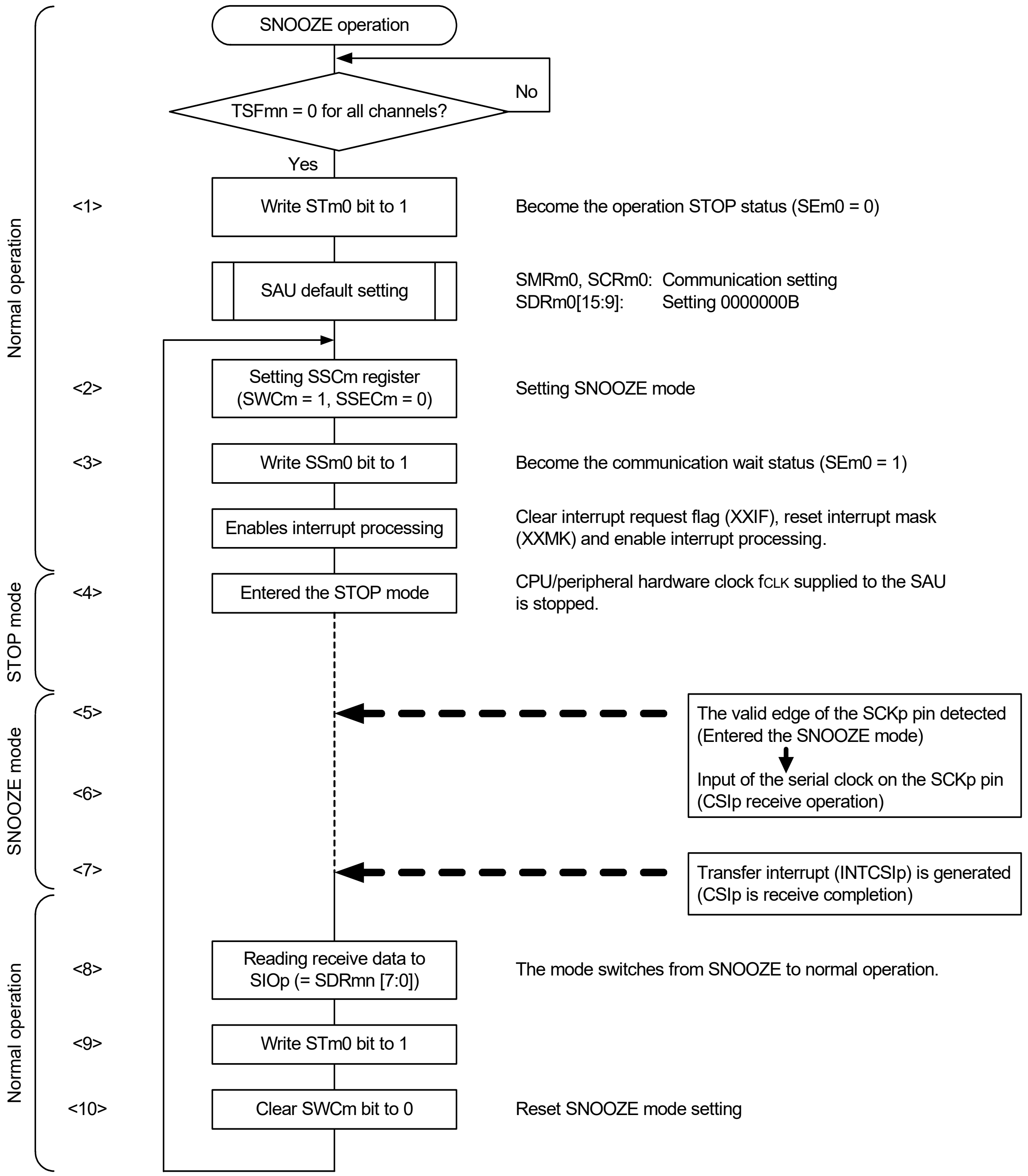
Caution 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (clear the SEm0 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE release).

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 13 - 74 Flowchart of SNOOZE Mode Operation (continuous startup).

Remark 2. m = 0; p = 00

Figure 13 - 74 Flowchart of SNOOZE Mode Operation (continuous startup)


Normal operation
STOP mode
SNOOZE mode
Normal operation
SNOOZE operation
TSFmn = 0 for all channels?
No
Yes
<1>
Write STm0 bit to 1
Become the operation STOP status (SEm0 = 0)
SAU default setting
SMRm0, SCRm0: Communication setting
SDRm0[15:9]: Setting 0000000B
<2>
Setting SSCm register (SWCm = 1, SSECm = 0)
Setting SNOOZE mode
<3>
Write SSm0 bit to 1
Become the communication wait status (SEm0 = 1)
Enables interrupt processing
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and enable interrupt processing.
<4>
Entered the STOP mode
CPU/peripheral hardware clock fCLK supplied to the SAU is stopped.
<5>
<6>
The valid edge of the SCKp pin detected (Entered the SNOOZE mode)
Input of the serial clock on the SCKp pin (CSIp receive operation)
<7>
Transfer interrupt (INTCSIp) is generated (CSIp is receive completion)
<8>
Reading receive data to SIOp (= SDRmn [7:0])
The mode switches from SNOOZE to normal operation.
<9>
Write STm0 bit to 1
<10>
Clear SWCm bit to 0
Reset SNOOZE mode setting

Remark 1. <1> to <10> in the figure correspond to <1> to <10> in Figure 13 - 73 Timing Chart of SNOOZE Mode Operation (continuous startup) (Type 1: DAPmn = 0, CKPmn = 0).

Remark 2. m = 0; p = 00

13.5.8 Calculating transfer clock frequency

The transfer clock frequency for simplified SPI (CSI00, CSI01, CSI10 and CSI11) communication can be calculated by the following expressions.

(1) Master

(Transfer clock frequency) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [Hz]

(2) Slave

(Transfer clock frequency) = {Frequency of serial clock (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 13 - 2 Selection of Operation Clock For Simplified SPI

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 = 0 to 3), mn = 00 to 03

13.5.9 Procedure for processing errors that occurred during simplified SPI (CSI00, CSI01, CSI10, CSI11) communication

The procedure for processing errors that occurred during simplified SPI (CSI00, CSI01, CSI10 and CSI11) communication is described in Figure 13 - 75.

Figure 13 - 75 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 to 3), mn = 00 to 03

13.6 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 35**, **CHAPTER 36 ELECTRICAL SPECIFICATIONS**.

- 20-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 (LIN-bus supported)	IIC00
	1	—		—
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11

- 24, 25-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 (LIN-bus supported)	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11

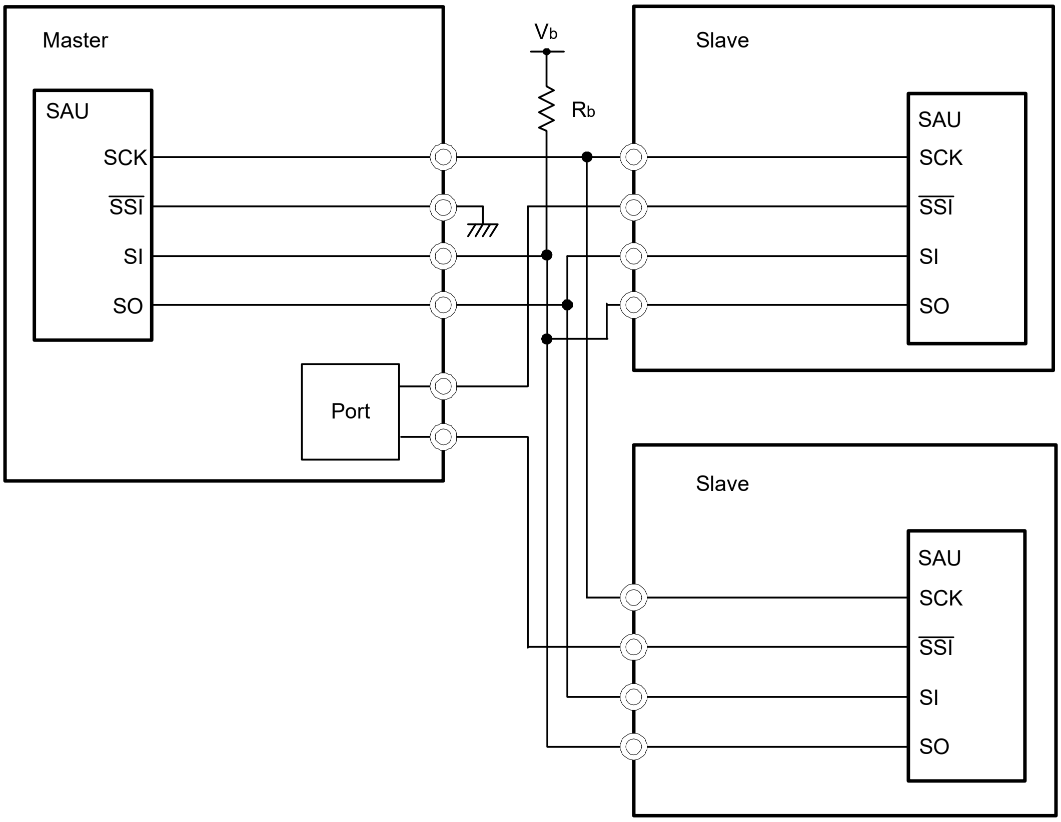
Slave select input function performs the following three types of communication operations.

• Slave transmission	(See **13.6.1**.)
• Slave reception	(See **13.6.2**.)
• Slave transmission/reception	(See **13.6.3**.)

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.

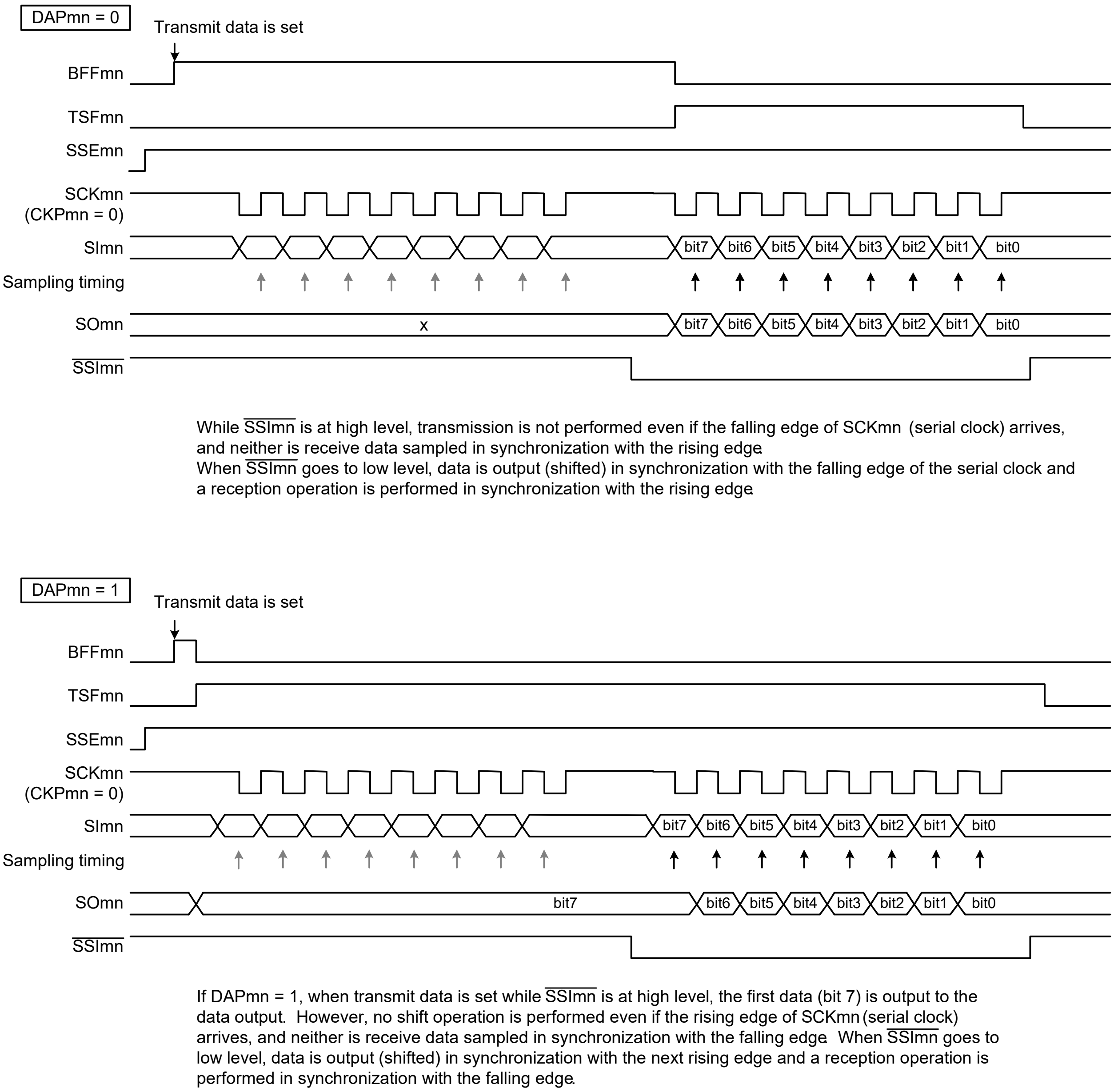
Figure 13 - 76 Example of Slave Select Input Function Configuration



Caution Make sure that $V_{DD} \geq V_b$ in the case of 10, 16, 20, and 24-pin products, and that $EV_{DD} \geq V_b$ in the case of 25-pin products.
Select the N-ch open-drain output (V_{DD} tolerance[Note 1]/EV_{DD} tolerance[Note 2]) mode for the SO00 pin.

Note 1. 10, 16, 20, 24-pin products
Note 2. 25-pin products

Figure 13 - 77 Slave Select Input Function Timing Diagram



While $\overline{\text{SSImn}}$ 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 $\overline{\text{SSImn}}$ goes to low level, data is output (shifted) in synchronization with the falling edge of the serial clock and a reception operation is performed in synchronization with the rising edge.

If DAPmn = 1, when transmit data is set while $\overline{\text{SSImn}}$ is at high level, the first data (bit 7) is output to the data output. However, no shift operation is performed even if the rising edge of SCKmn (serial clock) arrives, and neither is receive data sampled in synchronization with the falling edge. When $\overline{\text{SSImn}}$ goes to low level, data is output (shifted) in synchronization with the next rising edge and a reception operation is performed in synchronization with the falling edge.

Remark m: Unit number (m = 0), n: Channel number (n = 0)

13.6.1 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 35**, **CHAPTER 36 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 13 - 78 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 **13.3 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	0	0	0	0	CKOm3 ×	CKO2 ×	CKOm1 ×	CKOm0 ×	0	0	0	0	SOm3 ×	SOm2 ×	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	SOEm3 ×	SOEm2 ×	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 13 - 78 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													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	×	×	×	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						ISC1	ISC0
	0/1	0	0	0	0	0	0/1	0/1

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 is fixed in the simplified SPI (CSI) slave transmission mode,

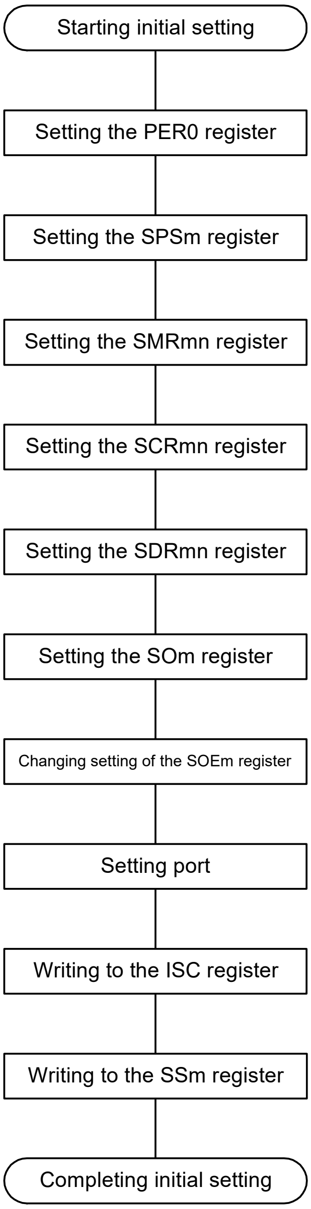
▨: Setting disabled (set to the initial value)

×: 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 13 - 79 Initial Setting Procedure for Slave Transmission


Starting initial setting
Setting the PER0 register
Setting the SPSm register
Setting the SMRmn register
Setting the SCRmn register
Setting the SDRmn register
Setting the SOm register
Changing setting of the SOEm register
Setting port
Writing to the ISC register
Writing to the SSm register
Completing initial setting

Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set bits 15 to 9 to 0000000B for baud rate setting.

Set the initial output level of the serial data (SOmn).

Set the SOEmn bit to 1 and enable data output of the target channel.

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.

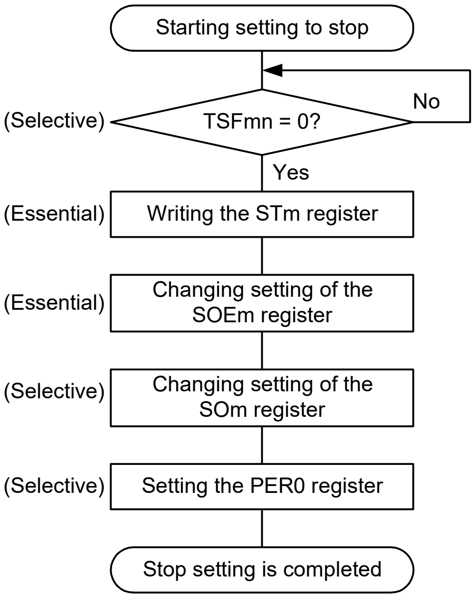
Set the SSIE00 bit to 1 and enable slave select function operation of channel 0.

Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation).

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.

Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 13 - 80 Procedure for Stopping Slave Transmission


Starting setting to stop
(Selective)
TSFmn = 0?
No
Yes
(Essential)
Writing the STm register
(Essential)
Changing setting of the SOEm register
(Selective)
Changing setting of the SOm register
(Selective)
Setting the PER0 register
Stop setting is completed

If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait.)

Write 1 to the STmn bit of the target channel (SEmn = 0: to operation stop status).

Set the SOEmn bit to 0 and stop the output of the target channel.

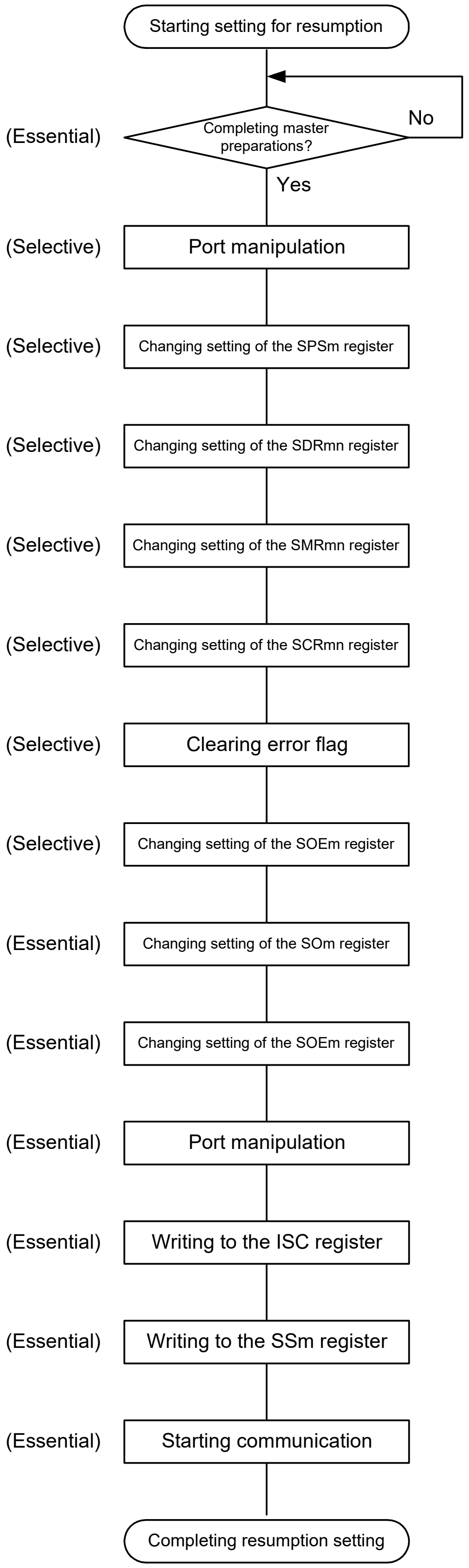
The levels of the serial data (SOmn) on the target channel can be changed if necessitated by an emergency.

Reset the serial array unit by stopping the clock supply to it.

After the stop setting is completed, go to the next processing.

Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 13 - 81 Procedure for Resuming Slave Transmission


Starting setting for resumption
(Essential)
Completing master preparations?
No
Yes
(Selective)
Port manipulation
(Selective)
Changing setting of the SPSm register
(Selective)
Changing setting of the SDRmn register
(Selective)
Changing setting of the SMRmn register
(Selective)
Changing setting of the SCRmn register
(Selective)
Clearing error flag
(Selective)
Changing setting of the SOEm register
(Essential)
Changing setting of the SOm register
(Essential)
Changing setting of the SOEm register
(Essential)
Port manipulation
(Essential)
Writing to the ISC register
(Essential)
Writing to the SSm register
(Essential)
Starting communication
Completing resumption setting

Wait until stop the communication target (master) or operation completed.

Disable data output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (f_{MCK})).

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change serial communication operation setting register mn (SCRmn) setting.

If the OVF flag remains set, clear this using serial flag clear trigger register mn (SIRmn).

Set the SOEmn bit to 0 to stop output from the target channel.

Set the initial output level of the serial data (SOmn).

Set the SOEmn bit to 1 and enable output from the target channel.

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.

Set the SSIE00 bit to 1 and enable slave select function operation of channel 0.

Set the SSmn bit of the target channel to 1 (SEmn = 1: to enable operation).

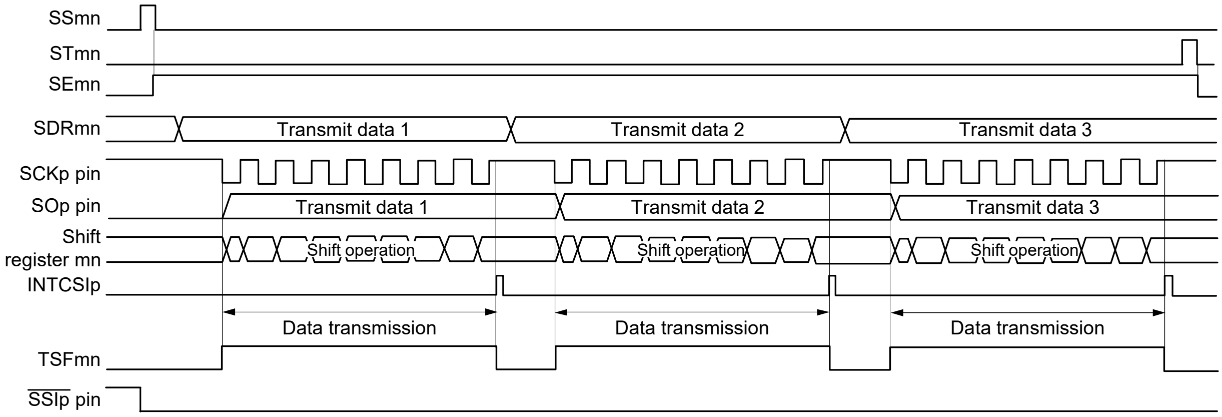
Sets transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and wait for a clock from the master.

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)

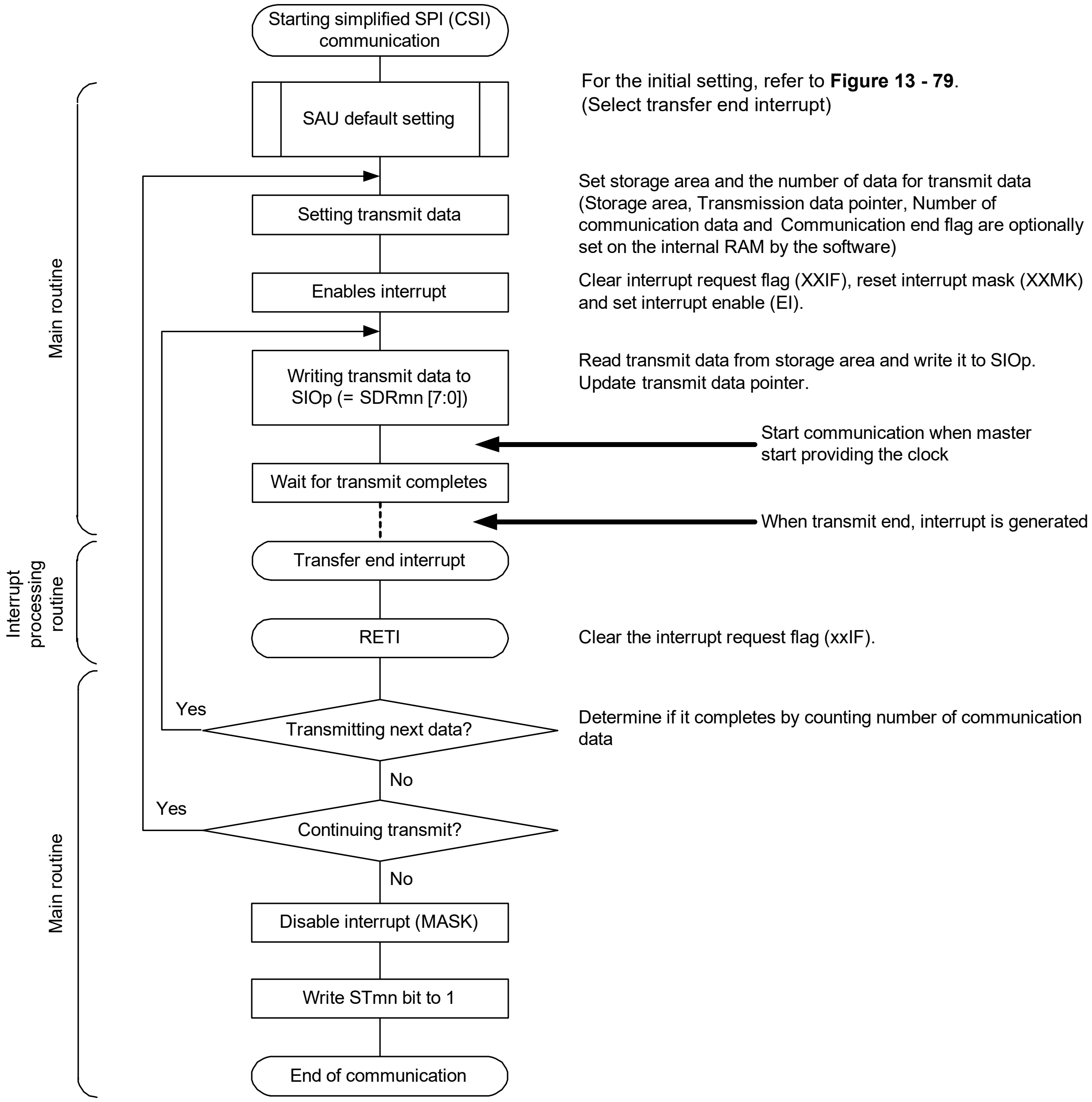
(3) Processing flow (in single-transmission mode)

Figure 13 - 82 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)

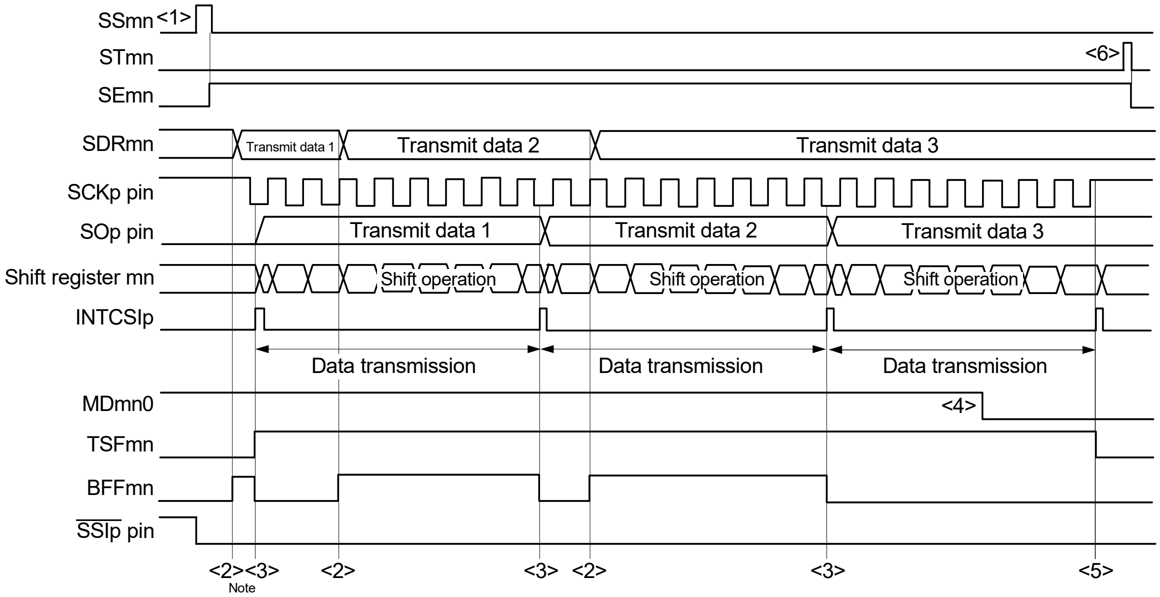
Figure 13 - 83 Flowchart of Slave Transmission (in Single-Transmission Mode)


Starting simplified SPI (CSI) communication
SAU default setting
For the initial setting, refer to **Figure 13 - 79**.
(Select transfer end interrupt)
Setting transmit data
Set storage area and the number of data for transmit data (Storage area, Transmission data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI).
Writing transmit data to SIOp (= SDRmn [7:0])
Read transmit data from storage area and write it to SIOp.
Update transmit data pointer.
Start communication when master start providing the clock
Wait for transmit completes
When transmit end, interrupt is generated
Transfer end interrupt
RETI
Clear the interrupt request flag (xxIF).
Transmitting next data?
Yes
No
Determine if it completes by counting number of communication data
Continuing transmit?
Yes
No
Disable interrupt (MASK)
Write STmn bit to 1
End of communication
Main routine
Interrupt processing routine
Main routine

Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

(4) Processing flow (in continuous transmission mode)

Figure 13 - 84 Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)

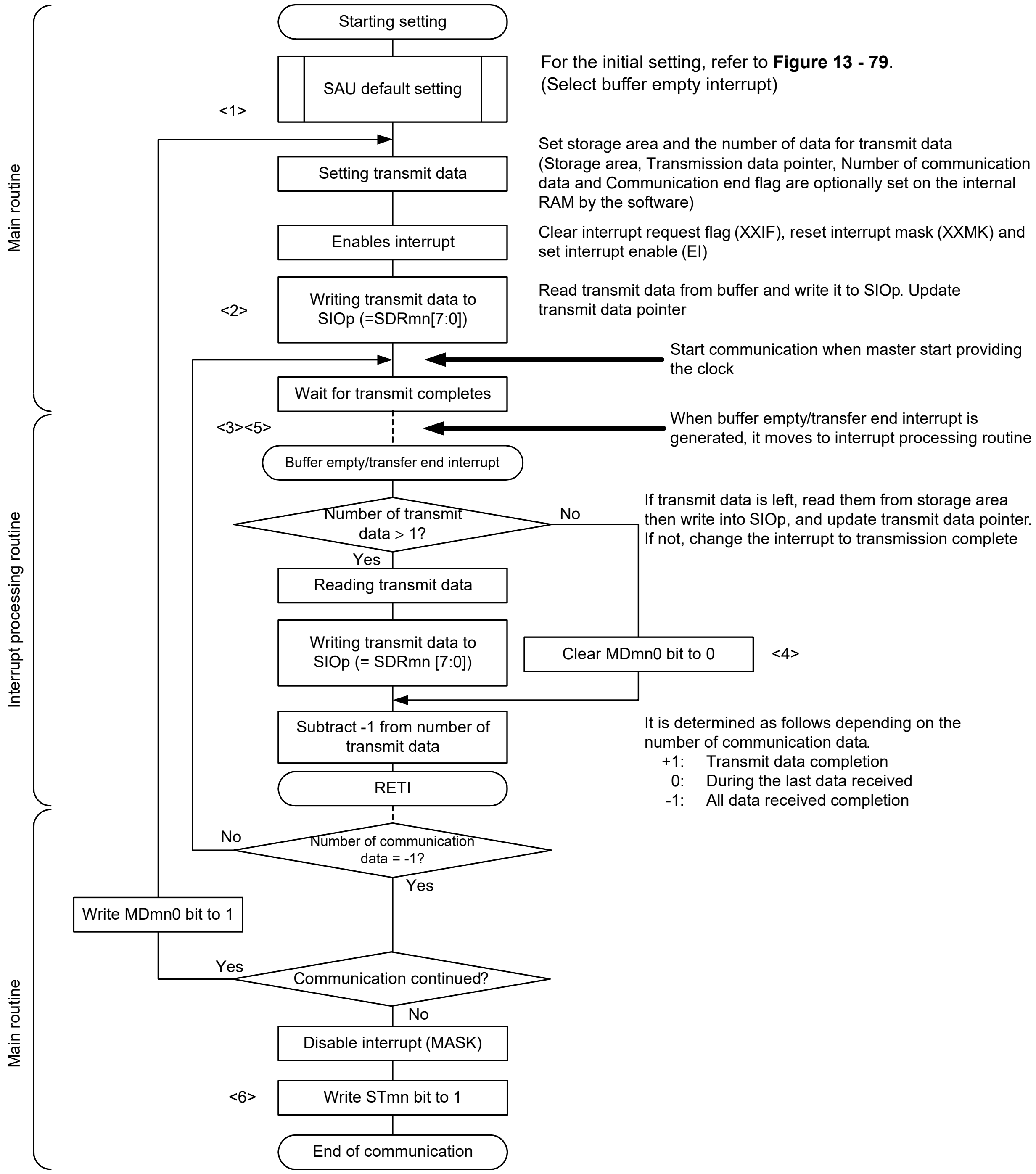


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

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 13 - 85 Flowchart of Slave Transmission (in Continuous Transmission Mode)


Main routine
Interrupt processing routine
Main routine
Starting setting
SAU default setting
<1>
For the initial setting, refer to **Figure 13 - 79**.
(Select buffer empty interrupt)
Setting transmit data
Set storage area and the number of data for transmit data (Storage area, Transmission data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI)
<2>
Writing transmit data to SIOp (=SDRmn[7:0])
Read transmit data from buffer and write it to SIOp. Update transmit data pointer
Start communication when master start providing the clock
Wait for transmit completes
<3><5>
When buffer empty/transfer end interrupt is generated, it moves to interrupt processing routine
Buffer empty/transfer end interrupt
Number of transmit data > 1?
No
Yes
If transmit data is left, read them from storage area then write into SIOp, and update transmit data pointer. If not, change the interrupt to transmission complete
Reading transmit data
Writing transmit data to SIOp (= SDRmn [7:0])
Clear MDmn0 bit to 0
<4>
Subtract -1 from number of transmit data
It is determined as follows depending on the number of communication data.
+1: Transmit data completion
0: During the last data received
-1: All data received completion
RETI
No
Number of communication data = -1?
Yes
Write MDmn0 bit to 1
Yes
Communication continued?
No
Disable interrupt (MASK)
<6>
Write STmn bit to 1
End of communication

Remark 1. <1> to <6> in the figure correspond to <1> to <6> in Figure 13 - 84 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)

13.6.2 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 35**, **CHAPTER 36 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 13 - 86 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 (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 **13.3 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	CKOm3 ×	CKOm2 ×	CKOm1 ×	CKOm0 ×	0	0	0	0	SOm3 ×	SOm2 ×	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	SOEm3 ×	SOEm2 ×	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 13 - 86 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	SSm3 ×	SSm2 ×	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	ISC1 0/1	ISC0 0/1

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 is fixed in the simplified SPI (CSI) slave reception mode,

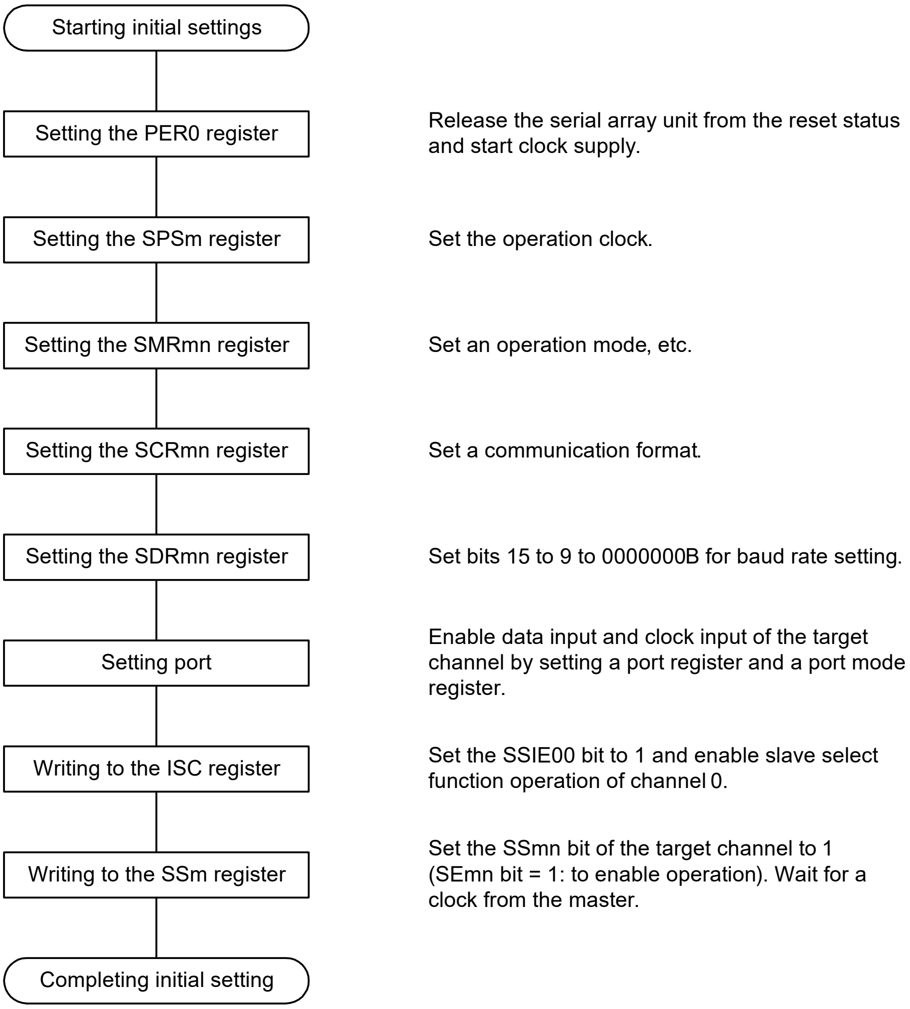
▨: Setting disabled (set to the initial value)

×: 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

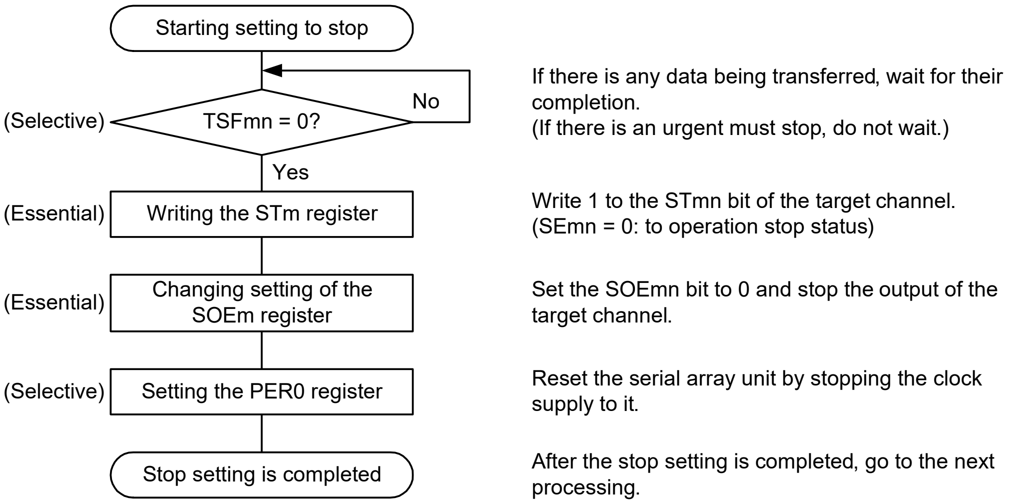
(2) Operation procedure

Figure 13 - 87 Initial Setting Procedure for Slave Reception



Starting initial settings
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 port
Enable data input and clock input of the target channel by setting a port register and a port mode register.
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). Wait for a clock from the master.
Completing initial setting

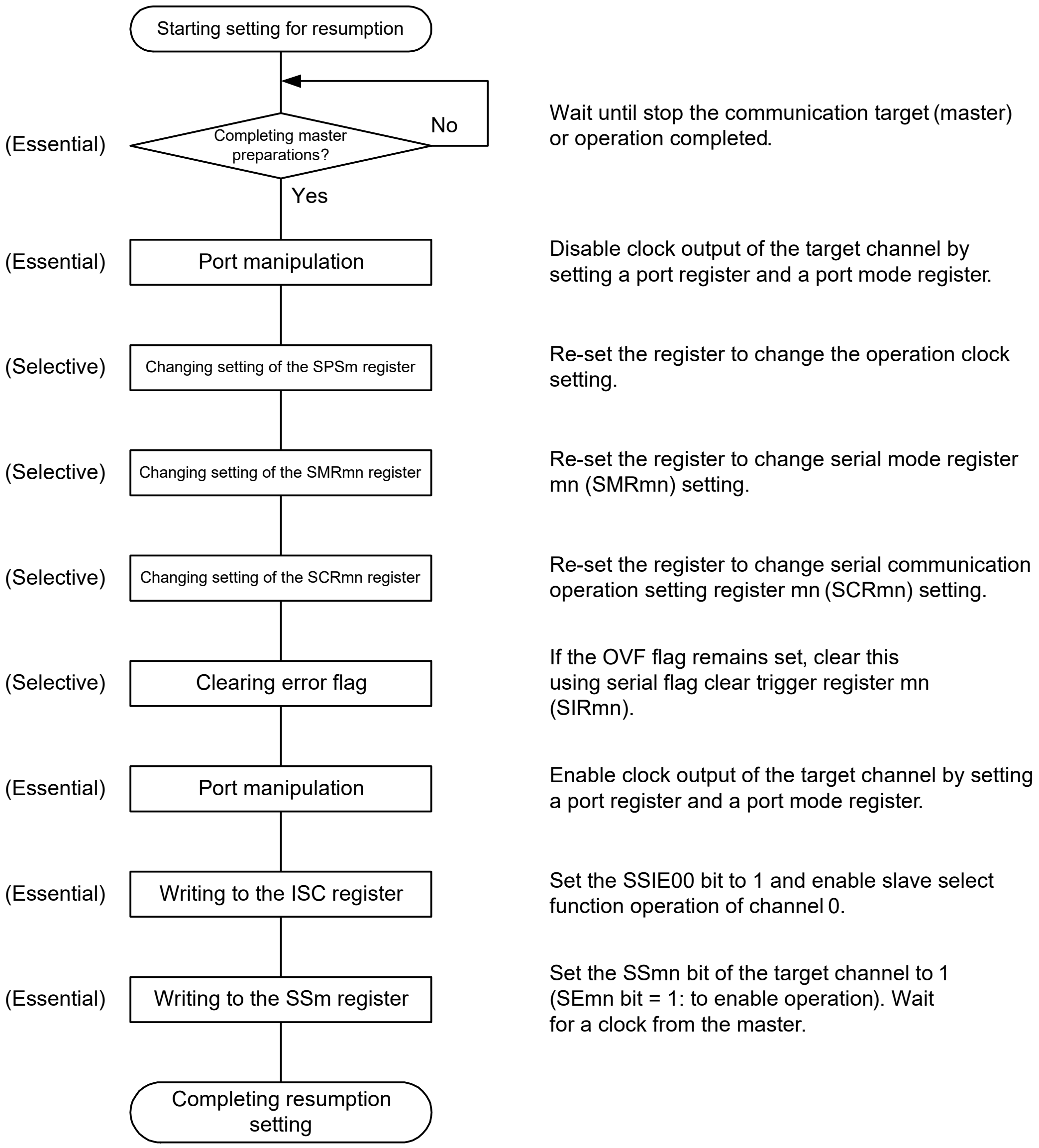
Figure 13 - 88 Procedure for Stopping Slave Reception



Starting setting to stop
(Selective)
TSFmn = 0?
No
Yes
If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait.)
(Essential)
Writing the STm register
Write 1 to the STmn bit of the target channel.
(SEmn = 0: to operation stop status)
(Essential)
Changing setting of the SOEm register
Set the SOEmn bit to 0 and stop the output of the target channel.
(Selective)
Setting the PER0 register
Reset the serial array unit by stopping the clock supply to it.
Stop setting is completed
After the stop setting is completed, go to the next processing.

Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

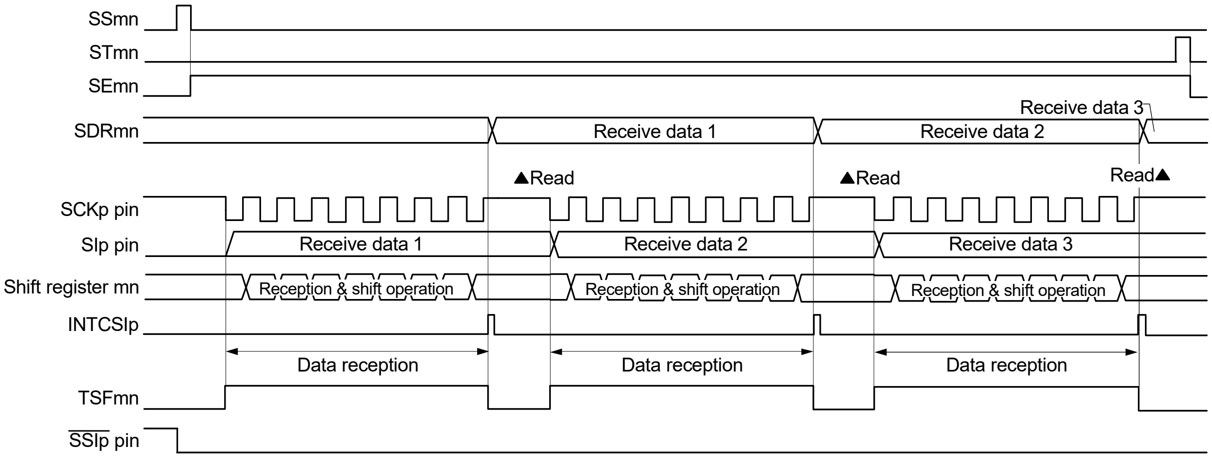
Figure 13 - 89 Procedure for Resuming Slave Reception


Starting setting for resumption
(Essential)
Completing master preparations?
No
Yes
Wait until stop the communication target (master) or operation completed.
(Essential)
Port manipulation
Disable clock output of the target channel by setting a port register and a port mode register.
(Selective)
Changing setting of the SPSm register
Re-set the register to change the operation clock setting.
(Selective)
Changing setting of the SMRmn register
Re-set the register to change serial mode register mn (SMRmn) setting.
(Selective)
Changing setting of the SCRmn register
Re-set the register to change serial communication operation setting register mn (SCRmn) setting.
(Selective)
Clearing error flag
If the OVF flag remains set, clear this using serial flag clear trigger register mn (SIRmn).
(Essential)
Port manipulation
Enable clock output of the target channel by setting a port register and a port mode register.
(Essential)
Writing to the ISC register
Set the SSIE00 bit to 1 and enable slave select function operation of channel 0.
(Essential)
Writing to the SSm register
Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation). Wait for a clock from the master.
Completing resumption setting

Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

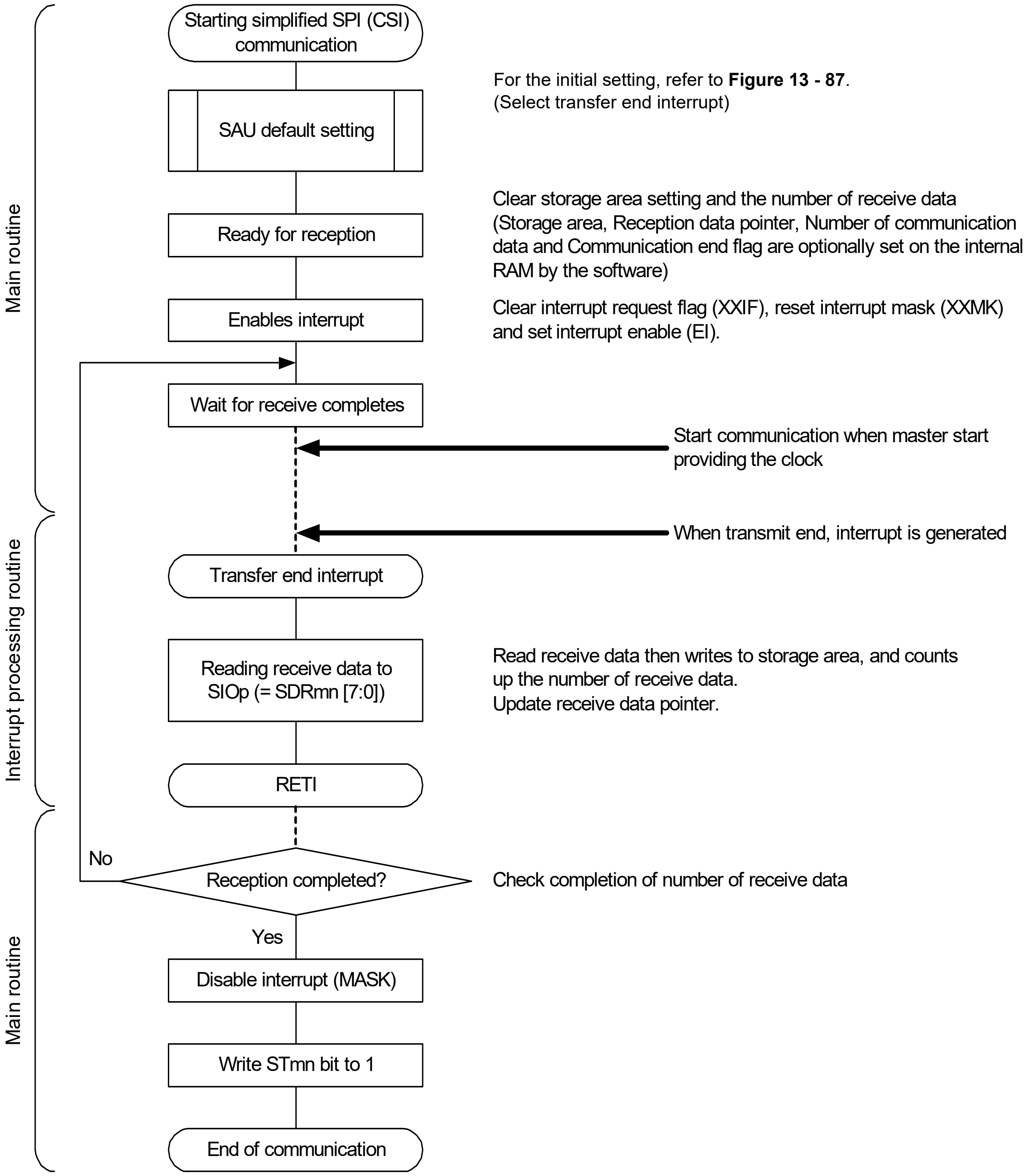
(3) Processing flow (in single-reception mode)

Figure 13 - 90 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 13 - 91 Flowchart of Slave Reception (in Single-Reception Mode)


Main routine
Starting simplified SPI (CSI) communication
SAU default setting
For the initial setting, refer to **Figure 13 - 87**.
(Select transfer end interrupt)
Ready for reception
Clear storage area setting and the number of receive data (Storage area, Reception data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI).
Wait for receive completes
Start communication when master start providing the clock
Interrupt processing routine
When transmit end, interrupt is generated
Transfer end interrupt
Reading receive data to SIOp (= SDRmn [7:0])
Read receive data then writes to storage area, and counts up the number of receive data.
Update receive data pointer.
RETI
Main routine
No
Reception completed?
Check completion of number of receive data
Yes
Disable interrupt (MASK)
Write STmn bit to 1
End of communication

13.6.3 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 35**, **CHAPTER 36 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 13 - 92 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 (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 **13.3 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							

(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	CKOm3 ×	CKOm2 ×	CKOm1 ×	CKOm0 ×	0	0	0	0	SOm3 ×	SOm2 ×	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	SOEm3 ×	SOEm2 ×	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 13 - 92 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	SSm3 ×	SSm2 ×	SSm1 ×	SSm0 ×

(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	ISC1 0/1	ISC0 0/1

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 is fixed in the simplified SPI (CSI) slave transmission/reception mode

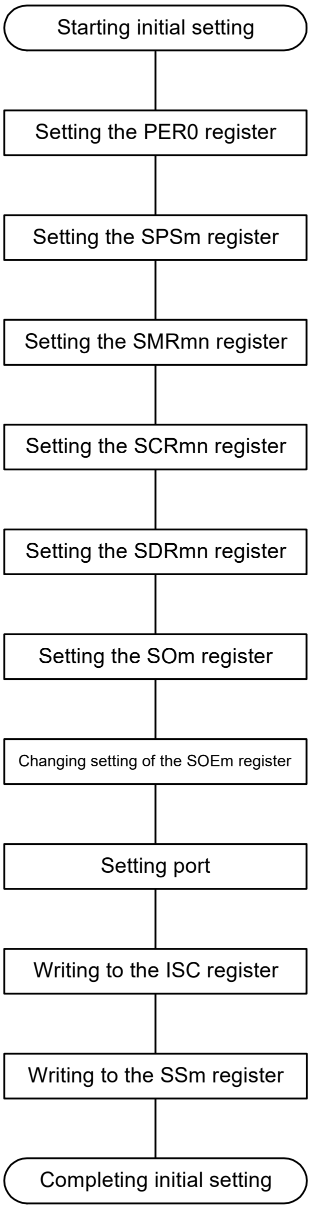
▨: Setting disabled (set to the initial value)

×: 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

(2) Operation procedure

Figure 13 - 93 Initial Setting Procedure for Slave Transmission/Reception


Starting initial setting
Setting the PER0 register
Setting the SPSm register
Setting the SMRmn register
Setting the SCRmn register
Setting the SDRmn register
Setting the SOm register
Changing setting of the SOEm register
Setting port
Writing to the ISC register
Writing to the SSm register
Completing initial setting

Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set bits 15 to 9 to 0000000B for baud rate setting.

Set the initial output level of the serial data (SOmn).

Set the SOEmn bit to 1 and enable data output of the target channel.

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.

Set the SSIE00 bit to 1 and enable slave select function operation of channel 0.

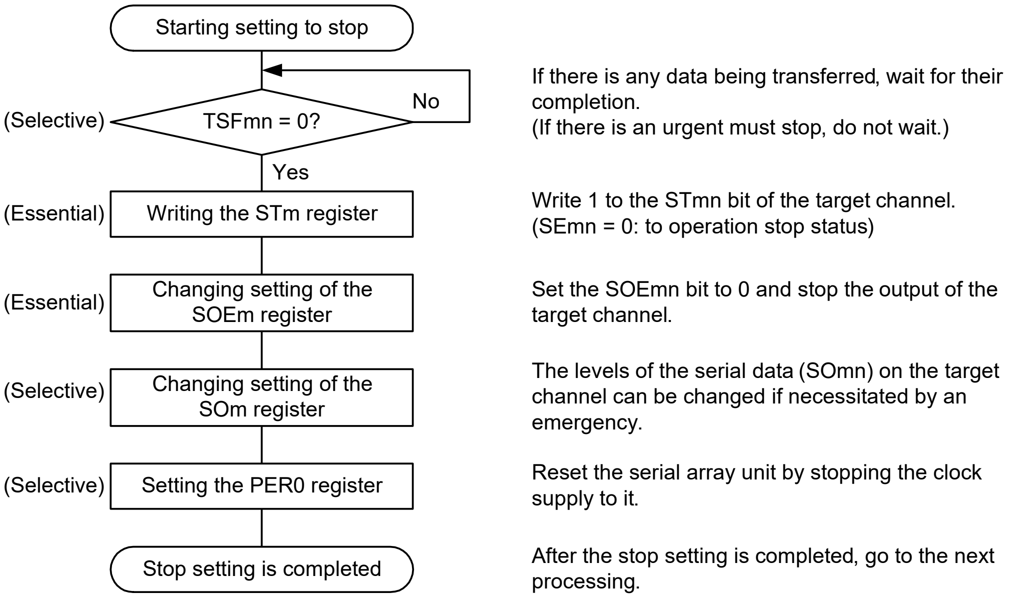
Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation).

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 13 - 94 Procedure for Stopping Slave Transmission/Reception


Starting setting to stop
(Selective)
TSFmn = 0?
No
Yes
(Essential)
Writing the STm register
(Essential)
Changing setting of the SOEm register
(Selective)
Changing setting of the SOm register
(Selective)
Setting the PER0 register
Stop setting is completed

If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait.)

Write 1 to the STmn bit of the target channel.
(SEmn = 0: to operation stop status)

Set the SOEmn bit to 0 and stop the output of the target channel.

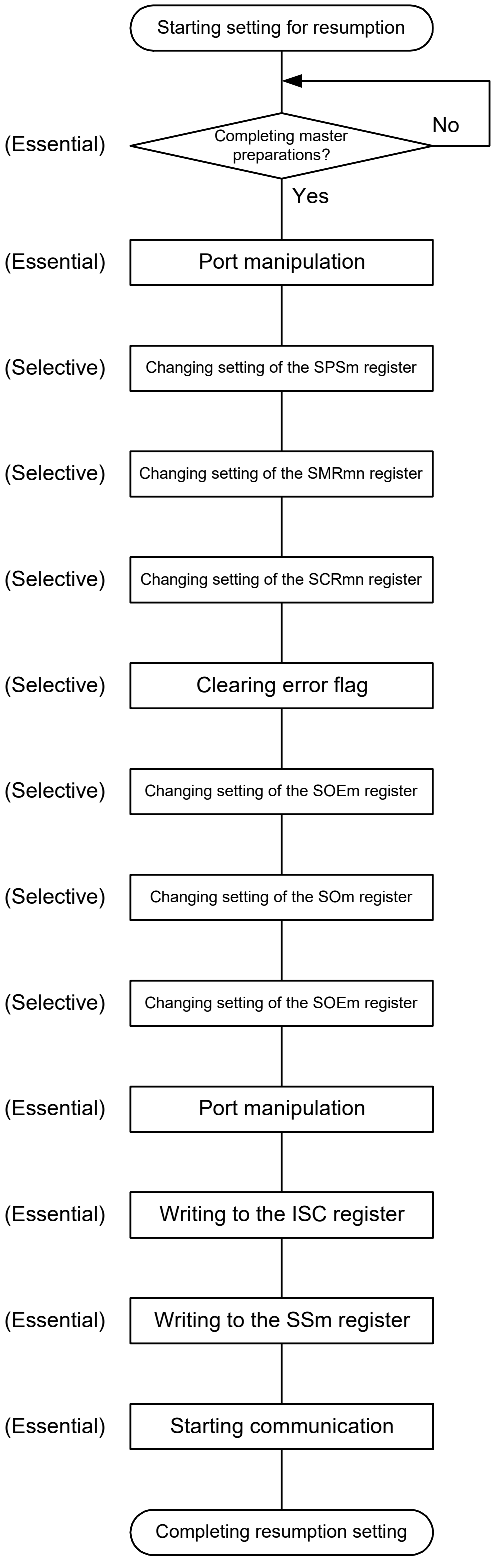
The levels of the serial data (SOmn) on the target channel can be changed if necessitated by an emergency.

Reset the serial array unit by stopping the clock supply to it.

After the stop setting is completed, go to the next processing.

Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 13 - 95 Procedure for Resuming Slave Transmission/Reception


Starting setting for resumption
(Essential)
Completing master preparations?
No
Yes
(Essential)
Port manipulation
(Selective)
Changing setting of the SPSm register
(Selective)
Changing setting of the SMRmn register
(Selective)
Changing setting of the SCRmn register
(Selective)
Clearing error flag
(Selective)
Changing setting of the SOEm register
(Selective)
Changing setting of the SOm register
(Selective)
Changing setting of the SOEm register
(Essential)
Port manipulation
(Essential)
Writing to the ISC register
(Essential)
Writing to the SSm register
(Essential)
Starting communication
Completing resumption setting

Wait until stop the communication target (master) or operation completed.

Disable data output of the target channel by setting a port register and a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change serial communication operation setting register mn (SCRmn) setting.

If the OVF flag remains set, clear this using serial flag clear trigger register mn (SIRmn).

Set the SOEmn bit to 0 to stop output from the target channel.

Set the initial output level of the serial data (SOmn).

Set the SOEmn bit to 1 and enable output from the target channel.

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.

Set the SSIE00 bit to 1 and enable slave select function operation of channel 0.

Set the SSmn bit of the target channel to 1 (SEmn = 1: to enable operation).

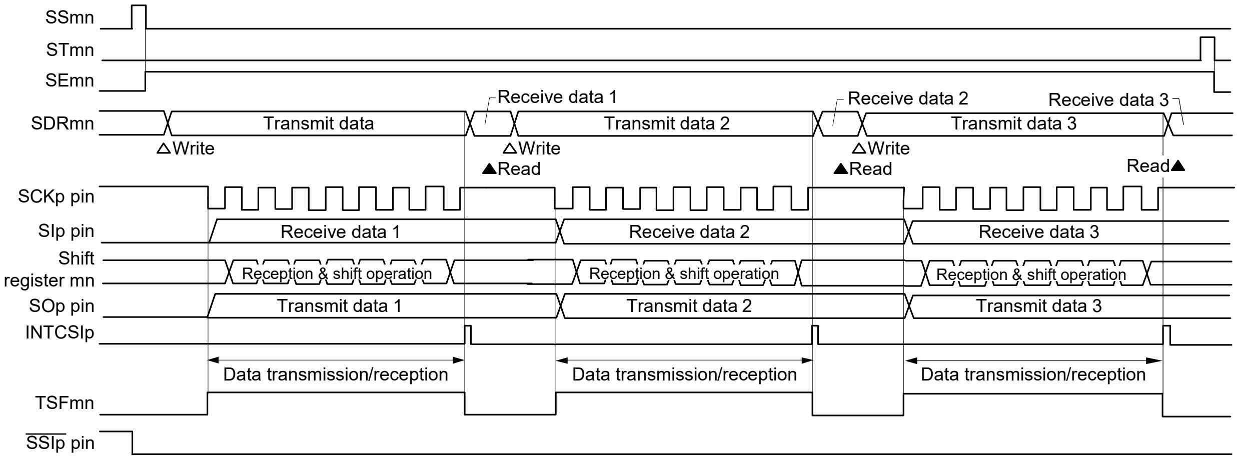
Sets transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and wait for a clock from the master.

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.

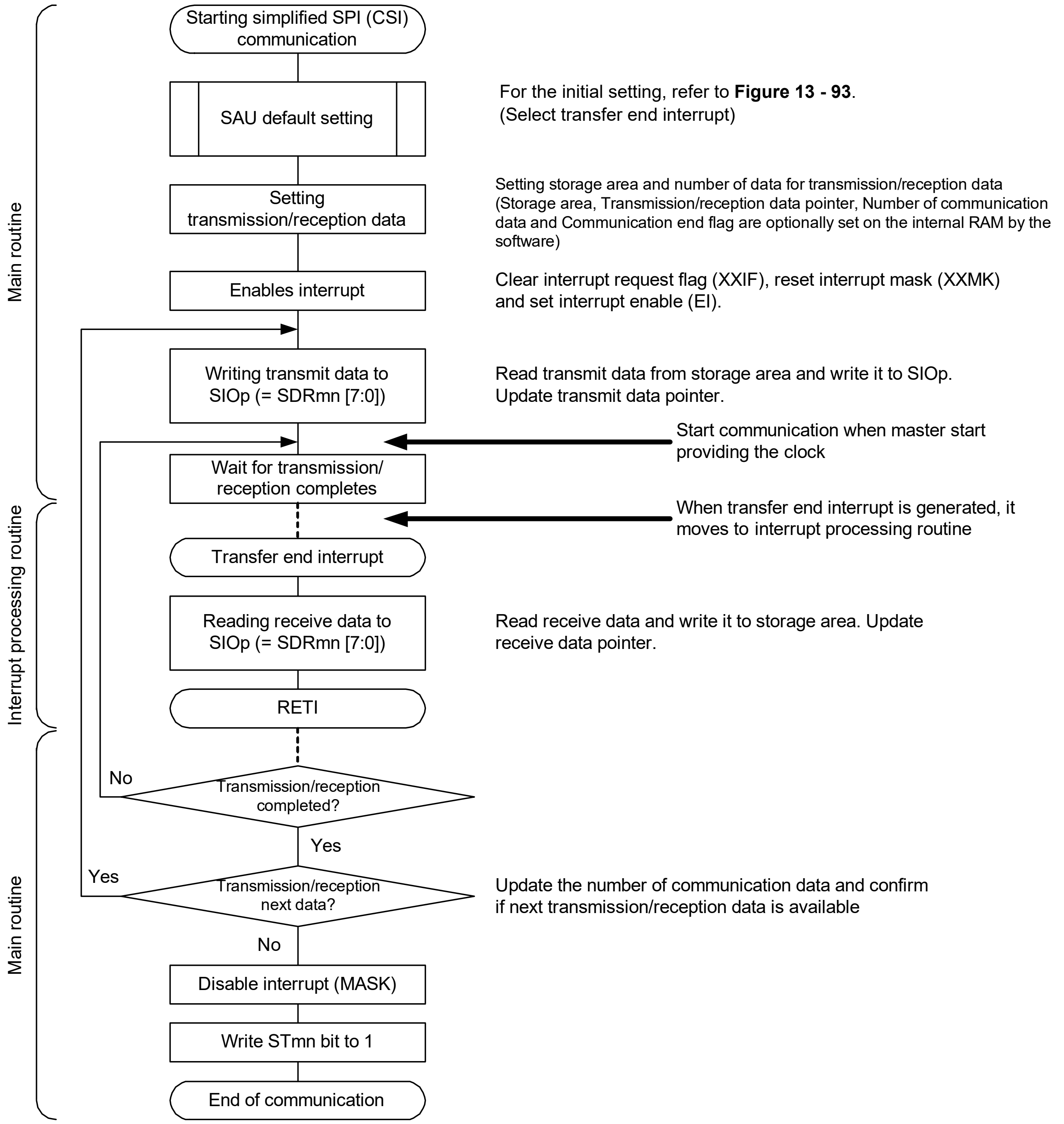
(3) Processing flow (in single-transmission/reception mode)

Figure 13 - 96 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 13 - 97 Flowchart of Slave Transmission/Reception (in Single- Transmission/Reception Mode)

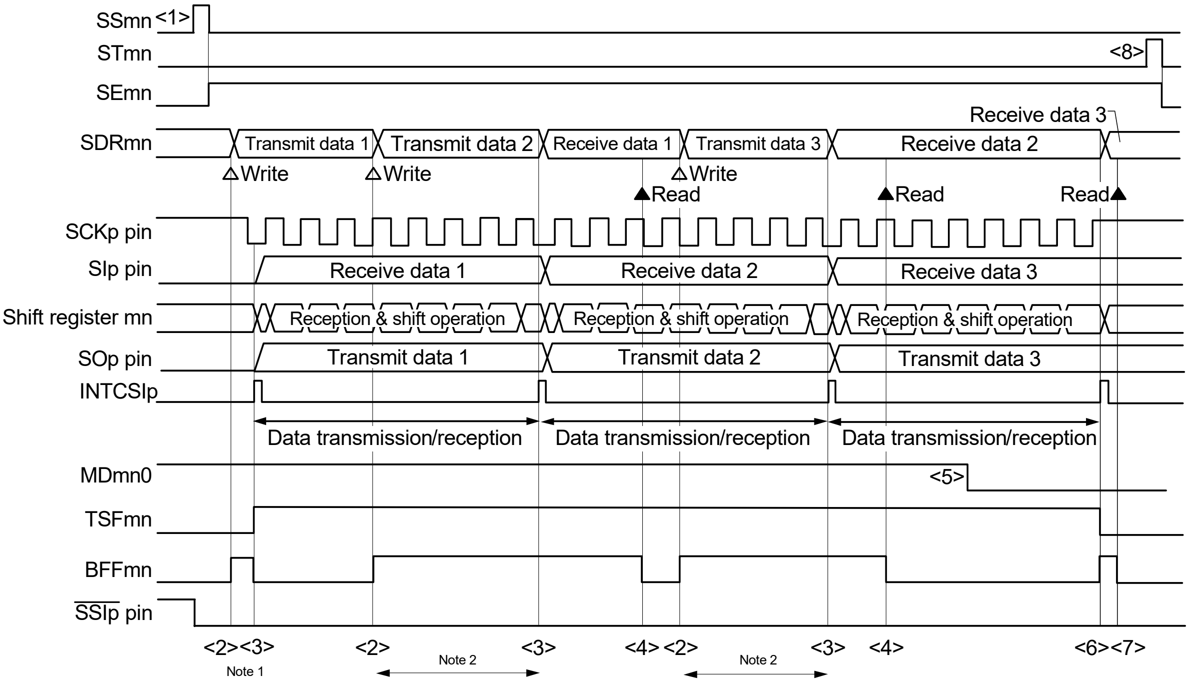

Main routine
Interrupt processing routine
Main routine
Starting simplified SPI (CSI) communication
SAU default setting
For the initial setting, refer to **Figure 13 - 93**.
(Select transfer end interrupt)
Setting transmission/reception data
Setting storage area and number of data for transmission/reception data (Storage area, Transmission/reception data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI).
Writing transmit data to SIOp (= SDRmn [7:0])
Read transmit data from storage area and write it to SIOp. Update transmit data pointer.
Start communication when master start providing the clock
Wait for transmission/ reception completes
When transfer end interrupt is generated, it moves to interrupt processing routine
Transfer end interrupt
Reading receive data to SIOp (= SDRmn [7:0])
Read receive data and write it to storage area. Update receive data pointer.
RETI
No
Transmission/reception completed?
Yes
Yes
Transmission/reception next data?
Update the number of communication data and confirm if next transmission/reception data is available
No
Disable interrupt (MASK)
Write STmn bit to 1
End of communication

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 13 - 98 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.

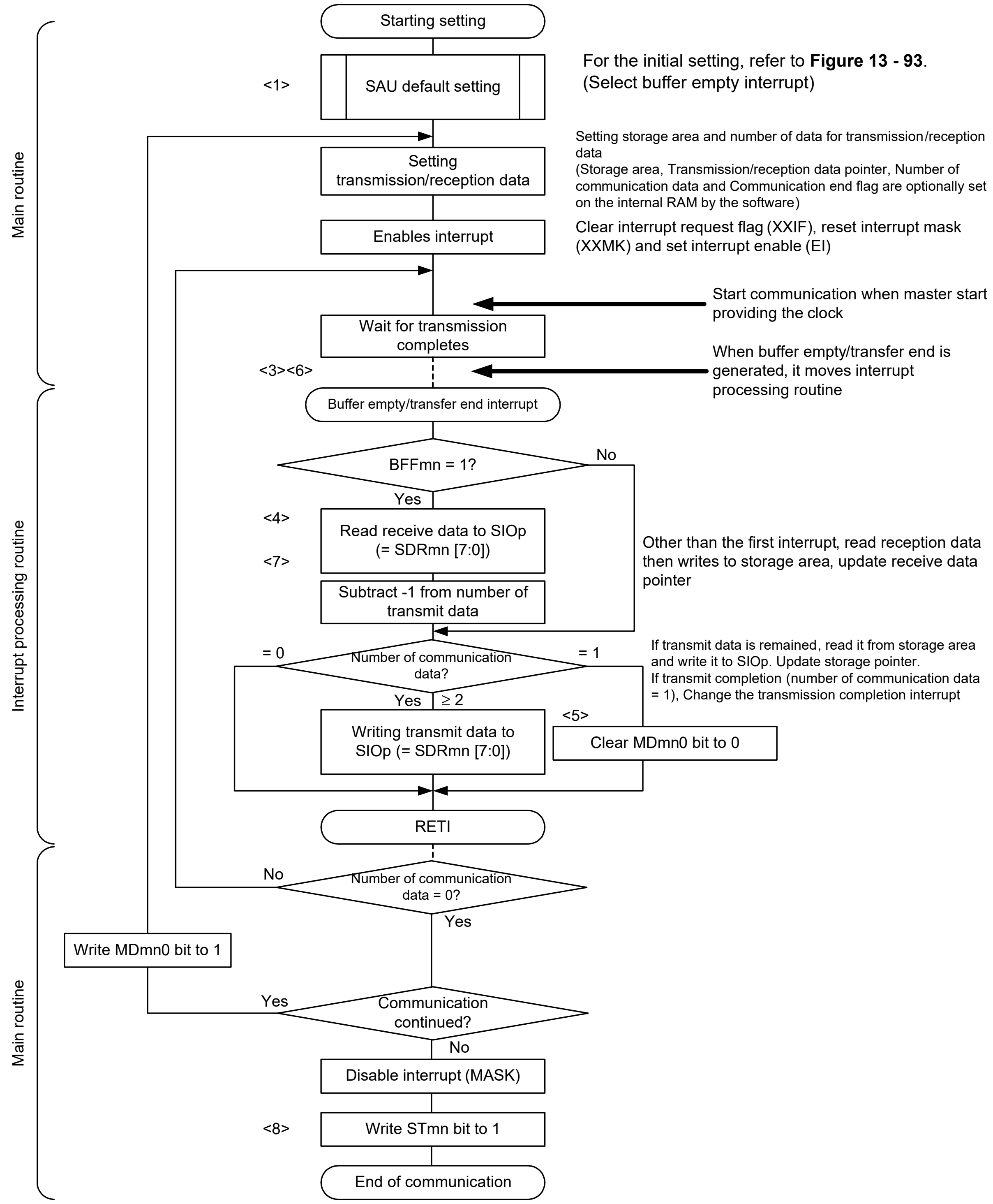
Note 2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

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 13 - 99 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 13 - 99 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)


Main routine
Interrupt processing routine
Main routine
Starting setting
<1>
SAU default setting
For the initial setting, refer to **Figure 13 - 93**.
(Select buffer empty interrupt)
Setting transmission/reception data
Setting storage area and number of data for transmission/reception data
(Storage area, Transmission/reception data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software)
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI)
Start communication when master start providing the clock
Wait for transmission completes
When buffer empty/transfer end is generated, it moves interrupt processing routine
<3><6>
Buffer empty/transfer end interrupt
BFFmn = 1?
No
Yes
<4>
<7>
Read receive data to SIOp (= SDRmn [7:0])
Subtract -1 from number of transmit data
Other than the first interrupt, read reception data then writes to storage area, update receive data pointer
= 0
Number of communication data?
= 1
If transmit data is remained, read it from storage area and write it to SIOp. Update storage pointer.
If transmit completion (number of communication data = 1), Change the transmission completion interrupt
Yes
≥ 2
<5>
Writing transmit data to SIOp (= SDRmn [7:0])
Clear MDmn0 bit to 0
RETI
No
Number of communication data = 0?
Yes
Write MDmn0 bit to 1
Yes
Communication continued?
No
Disable interrupt (MASK)
<8>
Write STmn bit to 1
End of communication

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 13 - 98 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)

13.6.4 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 13 - 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)

13.6.5 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 13 - 100.

Figure 13 - 100 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)

13.7 Operation of UART (UART0, UART1) Communication

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel).
The LIN-bus can be implemented by using timer array unit 0 (channel 3) with an external interrupt (INTP0).

[Data transmission/reception]

- Data length of 7, 8, or 9 bits[Note]
- 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, UARTs 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.

- UART0

The LIN-bus is accepted in UART0 (0 and 1 channels).

[LIN-bus functions]

• Wakeup signal detection • Break field (BF) detection • Sync field measurement, baud rate calculation	} Using the external interrupt (INTP0) and timer array unit

Note Only the following UART can be specified for the 9-bit data length.

- UART0

UART0 uses channels 0 and 1 of SAU0.
UART1 uses channels 2 and 3 of SAU0.

- 10-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART
0	0	—	—
	1	—	
	2	CSI10	UART1
	3	—	

- 16-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^{2}C
0	0	—	UART0 (LIN-bus supported)	—
	1	—		—
	2	CSI10	UART1	—
	3	CSI11		IIC11

- 20-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 function)	UART0 (LIN-bus supported)	IIC00
	1	—		—
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11

- 24, 25-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 function)	UART0 (LIN-bus supported)	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11

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. At this time, however, channel 2 or 3 of the same unit can be used for a function other than UART0, such as UART1, CSI10, and IIC10.

Caution When using a serial array unit for UART, both the transmitter side (even-numbered channel) and the receiver side (odd-numbered channel) can only be used for UART.

UART performs the following four types of communication operations.

• UART transmission	(See **13.7.1**.)
• UART reception	(See **13.7.2**.)
• LIN transmission (UART0 only)	(See **13.8.1**.)
• LIN reception (UART0 only)	(See **13.8.2**.)

13.7.1 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	UART1
Target channel	Channel 0 of SAU0	Channel 2 of SAU0
Pins used	TxD0	TxD1
Interrupt	INTST0	INTST1
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	None	
Transfer data length	7, 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 35**, **CHAPTER 36 ELECTRICAL SPECIFICATIONS**). For UART transmission, use the high-speed system 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, 2), mn = 00, 02

(1) Register setting

Figure 13 - 101 Example of Contents of Registers for UART Transmission of UART (UART0, UART1) (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 (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	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							
									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	SOLm2 0/1	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.
Only UART0 can be specified for the 9-bit data length.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 2), q: UART number (q = 0, 1), mn = 00, 02

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 13 - 101 Example of Contents of Registers for UART Transmission of UART (UART0, UART1) (2/2)

(e) Serial output register m (SOm)... Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	CKOm3 ×	CKOm2 ×	CKOm1 ×	CKOm0 ×	0	0	0	0	SOm3 ×	SOm2 0/1 Note	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	SOEm3 ×	SOEm2 0/1	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	SSm3 ×	SSm2 0/1	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, 2), q: UART number (q = 0, 1), mn = 00, 02

Remark 2. □: Setting is fixed in the UART transmission mode,

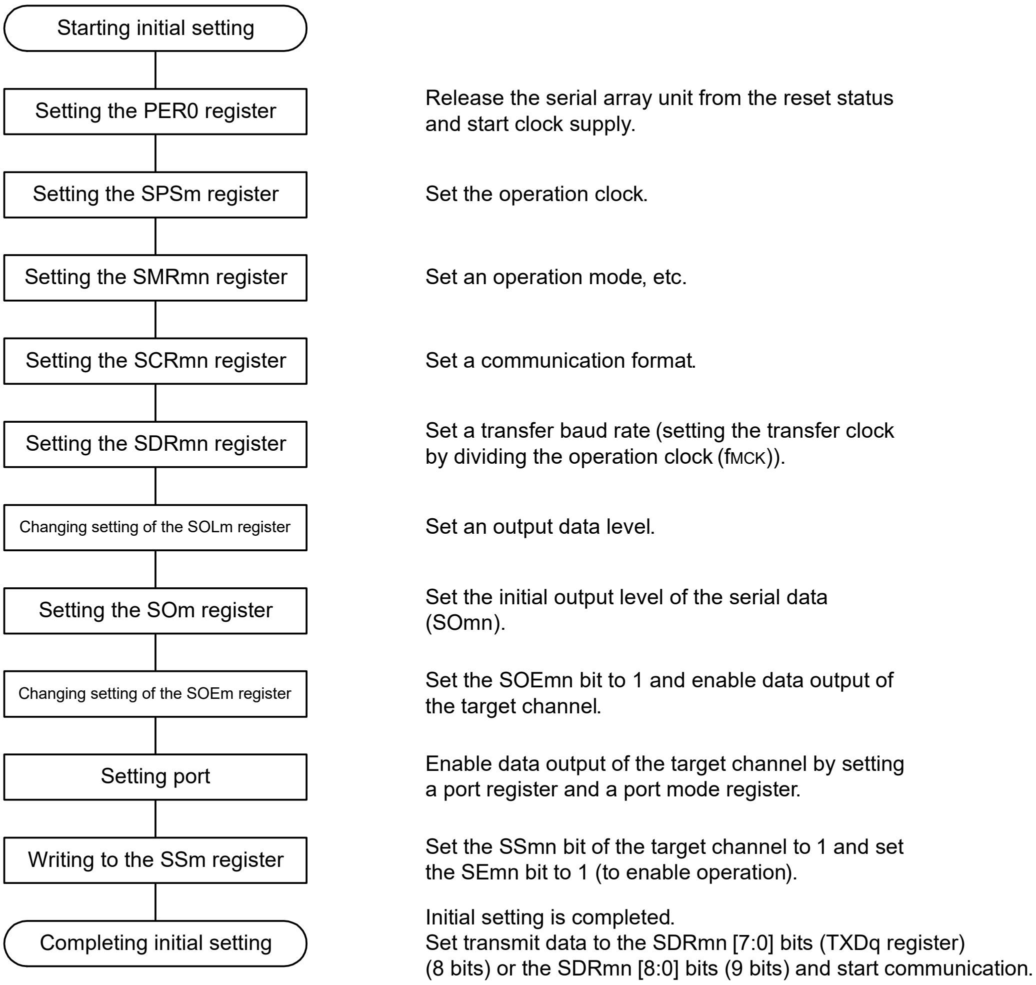
▨: Setting disabled (set to the initial value)

×: 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

(2) Operation procedure

Figure 13 - 102 Initial Setting Procedure for UART Transmission


Starting initial setting
Setting the PER0 register
Setting the SPSm register
Setting the SMRmn register
Setting the SCRmn register
Setting the SDRmn register
Changing setting of the SOLm register
Setting the SOm register
Changing setting of the SOEm register
Setting port
Writing to the SSm register
Completing initial setting

Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).

Set an output data level.

Set the initial output level of the serial data (SOmn).

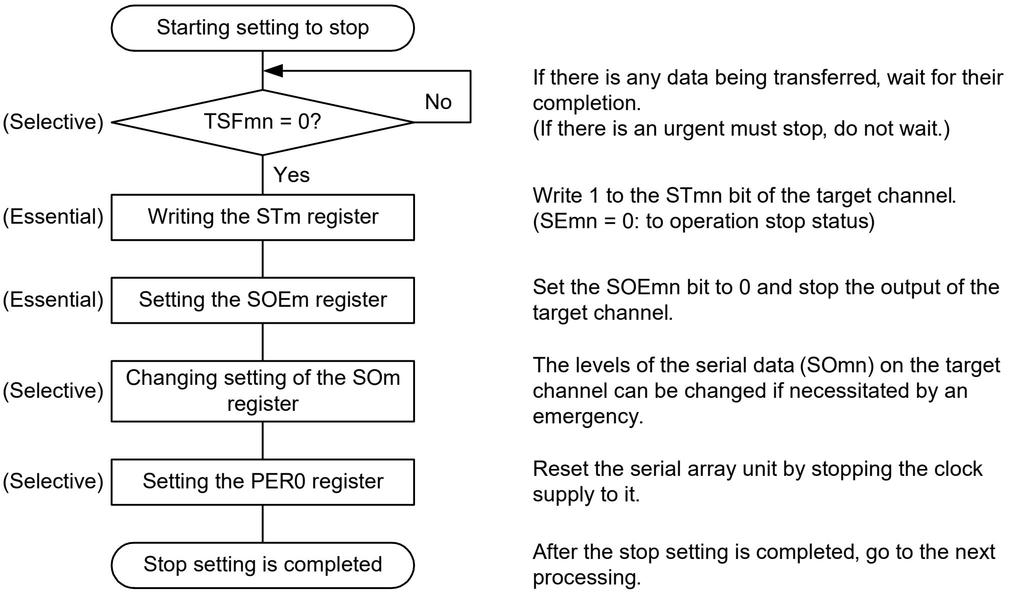
Set the SOEmn bit to 1 and enable data output of the target channel.

Enable data output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 and set the SEmn bit to 1 (to enable operation).

Initial setting is completed.
Set transmit data to the SDRmn [7:0] bits (TXDq register) (8 bits) or the SDRmn [8:0] bits (9 bits) and start communication.

Figure 13 - 103 Procedure for Stopping UART Transmission



If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait.)

Write 1 to the STmn bit of the target channel.
(SEmn = 0: to operation stop status)

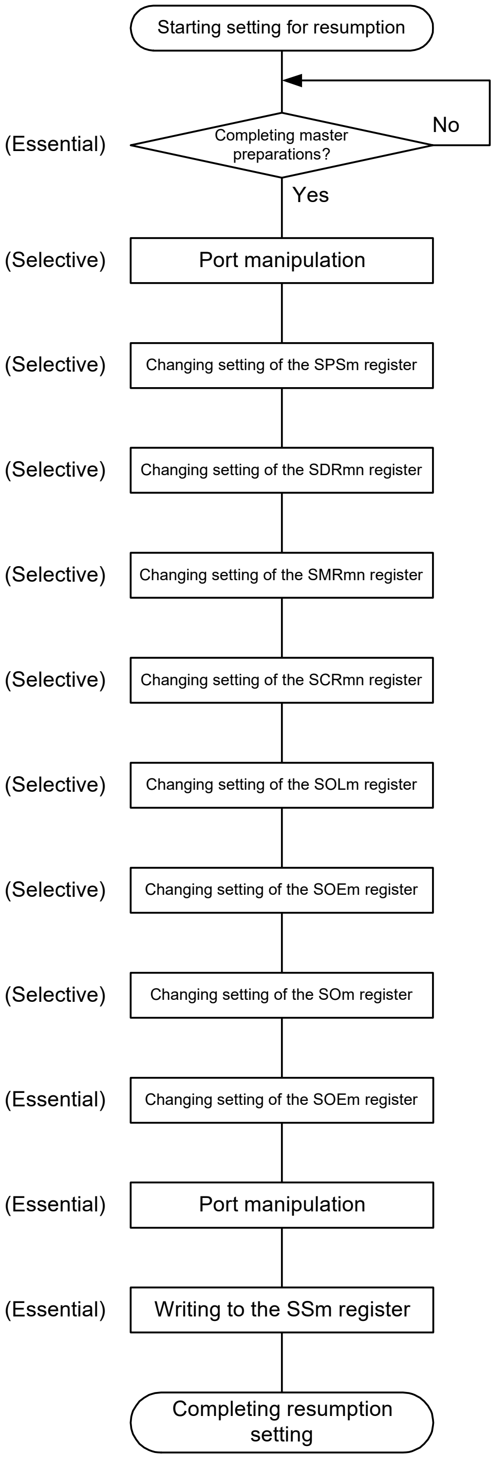
Set the SOEmn bit to 0 and stop the output of the target channel.

The levels of the serial data (SOmn) on the target channel can be changed if necessitated by an emergency.

Reset the serial array unit by stopping the clock supply to it.

After the stop setting is completed, go to the next processing.

Figure 13 - 104 Procedure for Resuming UART Transmission


Starting setting for resumption
(Essential)
Completing master preparations?
No
Yes
(Selective)
Port manipulation
(Selective)
Changing setting of the SPSm register
(Selective)
Changing setting of the SDRmn register
(Selective)
Changing setting of the SMRmn register
(Selective)
Changing setting of the SCRmn register
(Selective)
Changing setting of the SOLm register
(Selective)
Changing setting of the SOEm register
(Selective)
Changing setting of the SOm register
(Essential)
Changing setting of the SOEm register
(Essential)
Port manipulation
(Essential)
Writing to the SSm register
Completing resumption setting

Wait until stop the communication target or communication operation completed

Disable data output of the target channel by setting a port mode register.

Re-set the register to change the operation clock setting.

Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (fMCK)).

Re-set the register to change serial mode register mn (SMRmn) setting.

Re-set the register to change the serial communication operation setting register mn (SCRmn) setting.

Re-set the register to change serial output level register m (SOLm) setting.

Clear the SOEmn bit to 0 and stop output.

Set the initial output level of the serial data (SOmn).

Set the SOEmn bit to 1 and enable output.

Enable data output of the target channel by setting a port register and a port mode register.

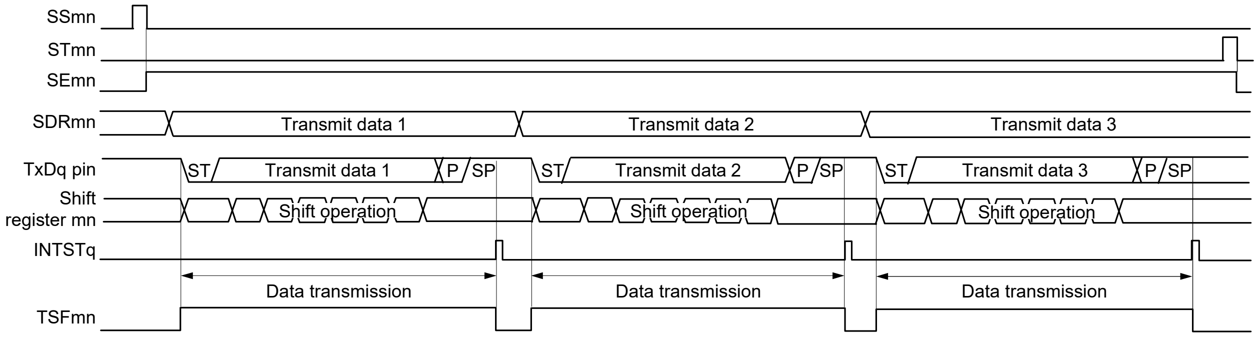
Set the SSmn bit of the target channel to 1 and set the SEmn bit to 1 (to enable operation).

Setting is completed.
Set transmit data to the SDRmn [7:0] bits (TXDq register) (8 bits) or the SDRmn [8:0] bits (9 bits) and start communication.

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.

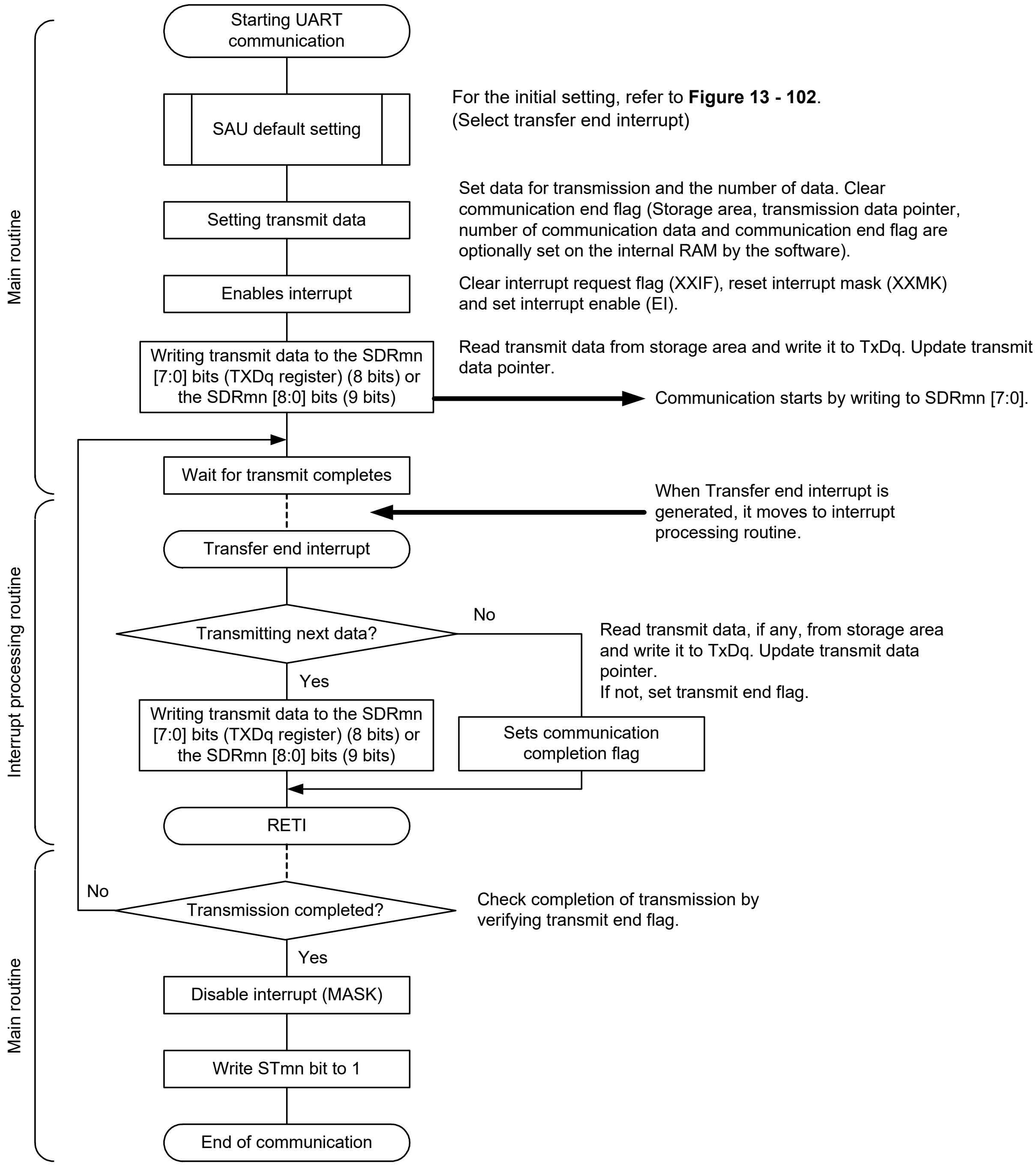
(3) Processing flow (in single-transmission mode)

Figure 13 - 105 Timing Chart of UART Transmission (in Single-Transmission Mode)



Remark m: Unit number (m = 0), n: Channel number (n = 0, 2), q: UART number (q = 0, 1), mn = 00, 02

Figure 13 - 106 Flowchart of UART Transmission (in Single-Transmission Mode)


Main routine
Interrupt processing routine
Main routine
Starting UART communication
SAU default setting
Setting transmit data
Enables interrupt
Writing transmit data to the SDRmn [7:0] bits (TXDq register) (8 bits) or the SDRmn [8:0] bits (9 bits)
Wait for transmit completes
Transfer end interrupt
Transmitting next data?
No
Yes
Writing transmit data to the SDRmn [7:0] bits (TXDq register) (8 bits) or the SDRmn [8:0] bits (9 bits)
Sets communication completion flag
RETI
No
Transmission completed?
Yes
Disable interrupt (MASK)
Write STmn bit to 1
End of communication

For the initial setting, refer to **Figure 13 - 102**.
(Select transfer end interrupt)

Set data for transmission and the number of data. Clear communication end flag (Storage area, transmission data pointer, number of communication data and communication end flag are optionally set on the internal RAM by the software).

Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI).

Read transmit data from storage area and write it to TxDq. Update transmit data pointer.

Communication starts by writing to SDRmn [7:0].

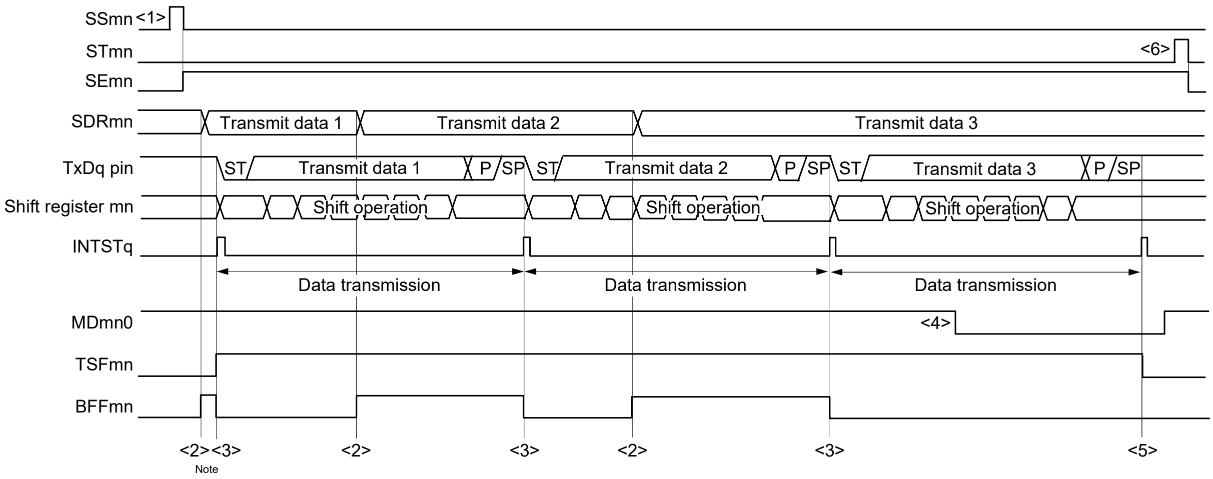
When Transfer end interrupt is generated, it moves to interrupt processing routine.

Read transmit data, if any, from storage area and write it to TxDq. Update transmit data pointer.
If not, set transmit end flag.

Check completion of transmission by verifying transmit end flag.

(4) Processing flow (in continuous transmission mode)

Figure 13 - 107 Timing Chart of UART Transmission (in Continuous Transmission Mode)

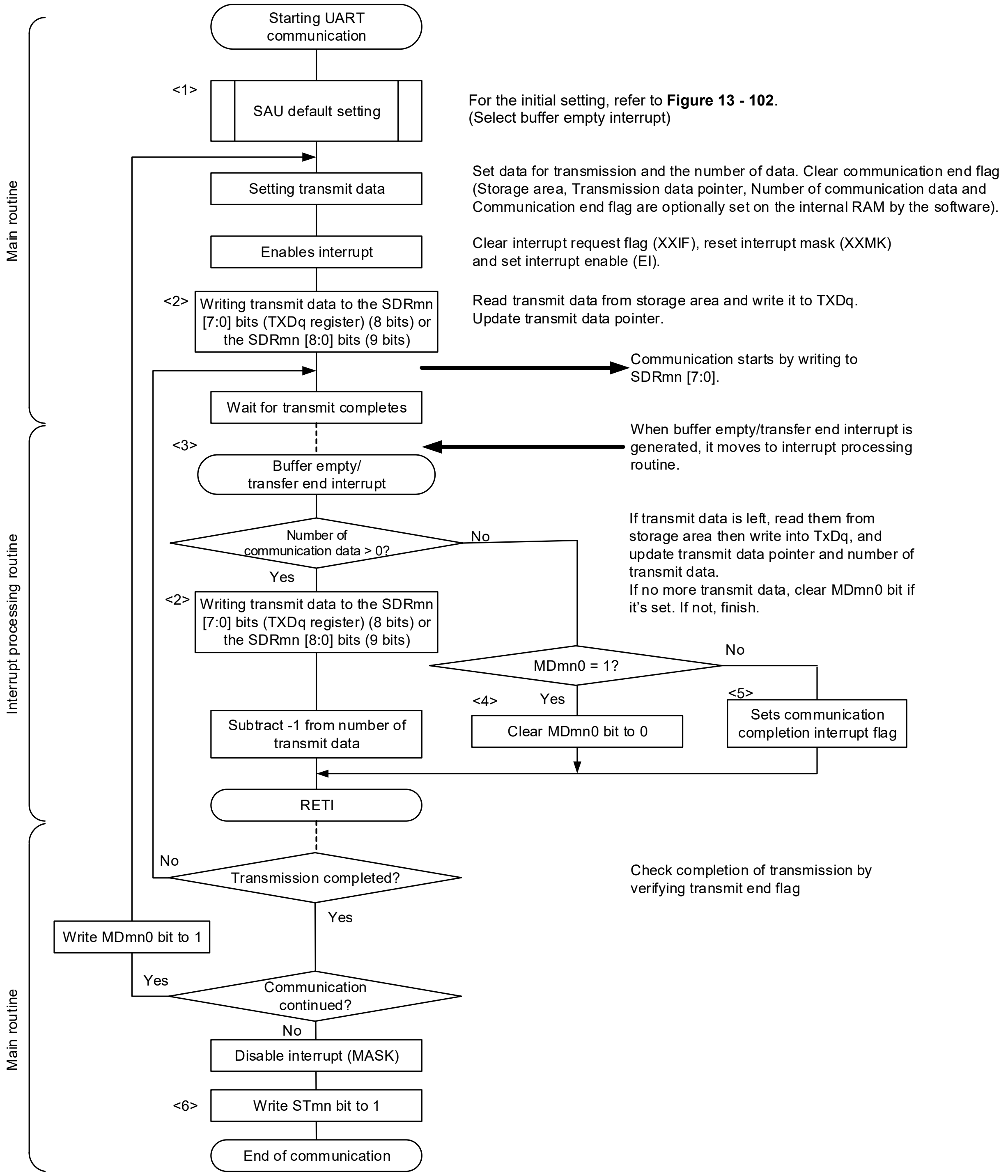


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

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, 2), q: UART number (q = 0, 1), mn = 00, 02

Figure 13 - 108 Flowchart of UART Transmission (in Continuous Transmission Mode)


Main routine
Starting UART communication
<1>
SAU default setting
For the initial setting, refer to Figure 13 - 102.
(Select buffer empty interrupt)
Setting transmit data
Set data for transmission and the number of data. Clear communication end flag (Storage area, Transmission data pointer, Number of communication data and Communication end flag are optionally set on the internal RAM by the software).
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (EI).
<2>
Writing transmit data to the SDRmn [7:0] bits (TXDq register) (8 bits) or the SDRmn [8:0] bits (9 bits)
Read transmit data from storage area and write it to TXDq.
Update transmit data pointer.
Communication starts by writing to SDRmn [7:0].
Wait for transmit completes
Interrupt processing routine
<3>
Buffer empty/ transfer end interrupt
When buffer empty/transfer end interrupt is generated, it moves to interrupt processing routine.
Number of communication data > 0?
No
Yes
If transmit data is left, read them from storage area then write into TxDq, and update transmit data pointer and number of transmit data.
If no more transmit data, clear MDmn0 bit if it's set. If not, finish.
<2>
Writing transmit data to the SDRmn [7:0] bits (TXDq register) (8 bits) or the SDRmn [8:0] bits (9 bits)
MDmn0 = 1?
No
<4>
Yes
<5>
Subtract -1 from number of transmit data
Clear MDmn0 bit to 0
Sets communication completion interrupt flag
RETI
Main routine
No
Transmission completed?
Check completion of transmission by verifying transmit end flag
Yes
Write MDmn0 bit to 1
Yes
Communication continued?
No
Disable interrupt (MASK)
<6>
Write STmn bit to 1
End of communication

Remark <1> to <6> in the figure correspond to <1> to <6> in Figure 13 - 107 Timing Chart of UART Transmission (in Continuous Transmission Mode).

13.7.2 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	UART1
Target channel	Channel 1 of SAU0	Channel 3 of SAU0
Pins used	RxD0	RxD1
Interrupt	INTSR0	INTSR1
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error interrupt	INTSRE0	INTSRE1
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 35**, **CHAPTER 36 ELECTRICAL SPECIFICATIONS**). For UART reception, use the high-speed system 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, 3), mn = 01, 03

(1) Register setting

Figure 13 - 109 Example of Contents of Registers for UART Reception of UART (UART0, UART1) (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	14	13	12	11	10	9	8	7	6	5	4	3	2	1	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 reception data.
Only UART0 can be specified for the 9-bit data length.

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, 3), mn = 01, 03,
r: Channel number (r = n - 1),
q: UART number (q = 0, 1)

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 13 - 109 Example of Contents of Registers for UART Reception of UART (UART0, UART1) (2/2)

(e) Serial output register m (SOm)... The register that not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	CKOm3 ×	CKOm2 ×	CKOm1 ×	CKOm0 ×	0	0	0	0	SOm3 ×	SOm2 ×	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	SOEm3 ×	SOEm2 ×	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	SSm3 0/1	SSm2 ×	SSm1 0/1	SSm0 ×

Remark 1. m: Unit number (m = 0)

Remark 2. □: Setting is fixed in the UART reception mode,

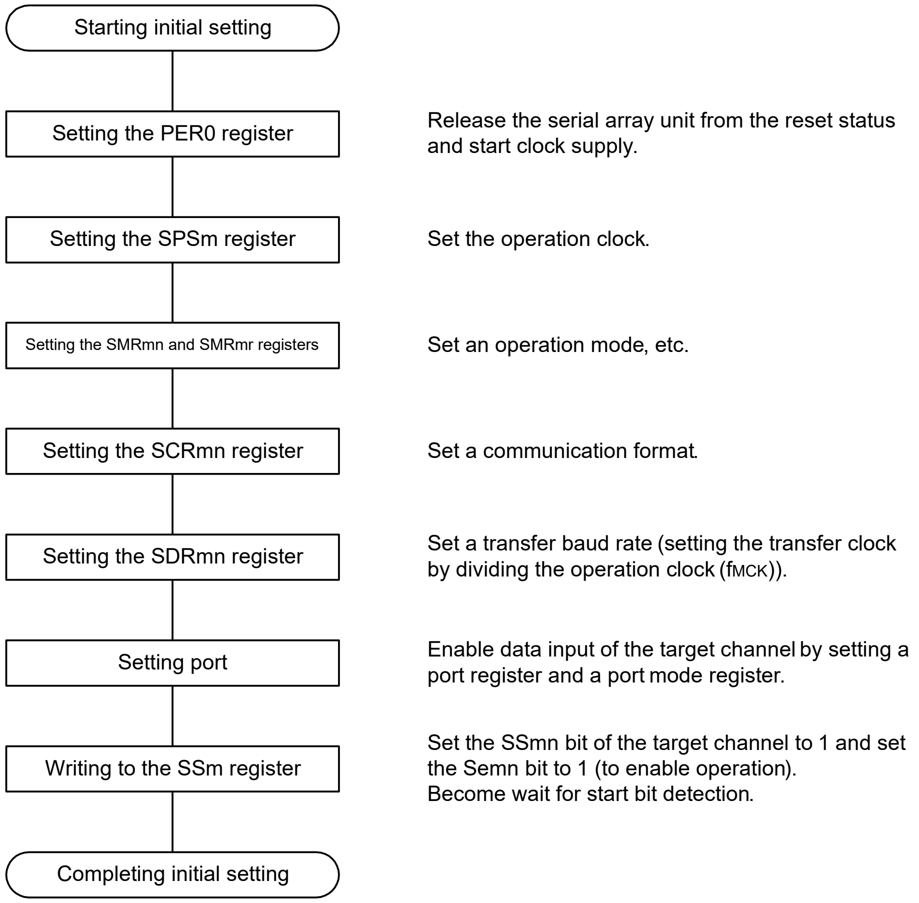
▨: Setting disabled (set to the initial value)

×: 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

(2) Operation procedure

Figure 13 - 110 Initial Setting Procedure for UART Reception


Starting initial setting
Setting the PER0 register
Setting the SPSm register
Setting the SMRmn and SMRmr registers
Setting the SCRmn register
Setting the SDRmn register
Setting port
Writing to the SSm register
Completing initial setting

Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

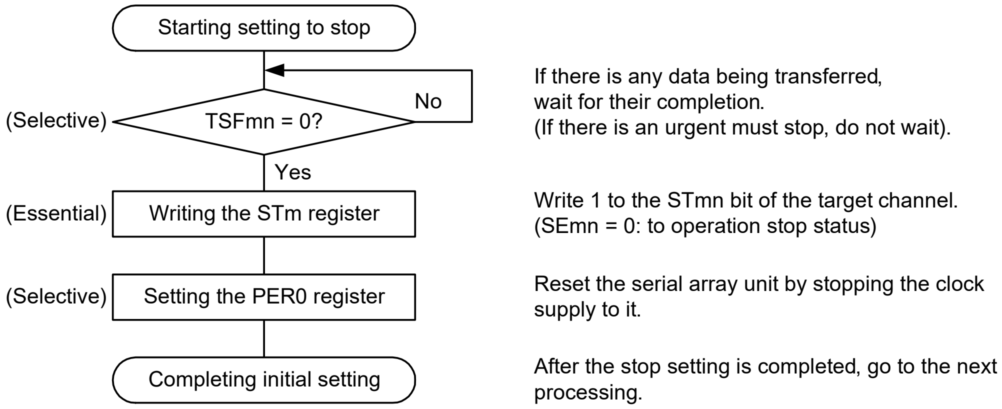
Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).

Enable data input of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 and set the Semn bit to 1 (to enable operation).
Become wait for start bit detection.

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 13 - 111 Procedure for Stopping UART Reception



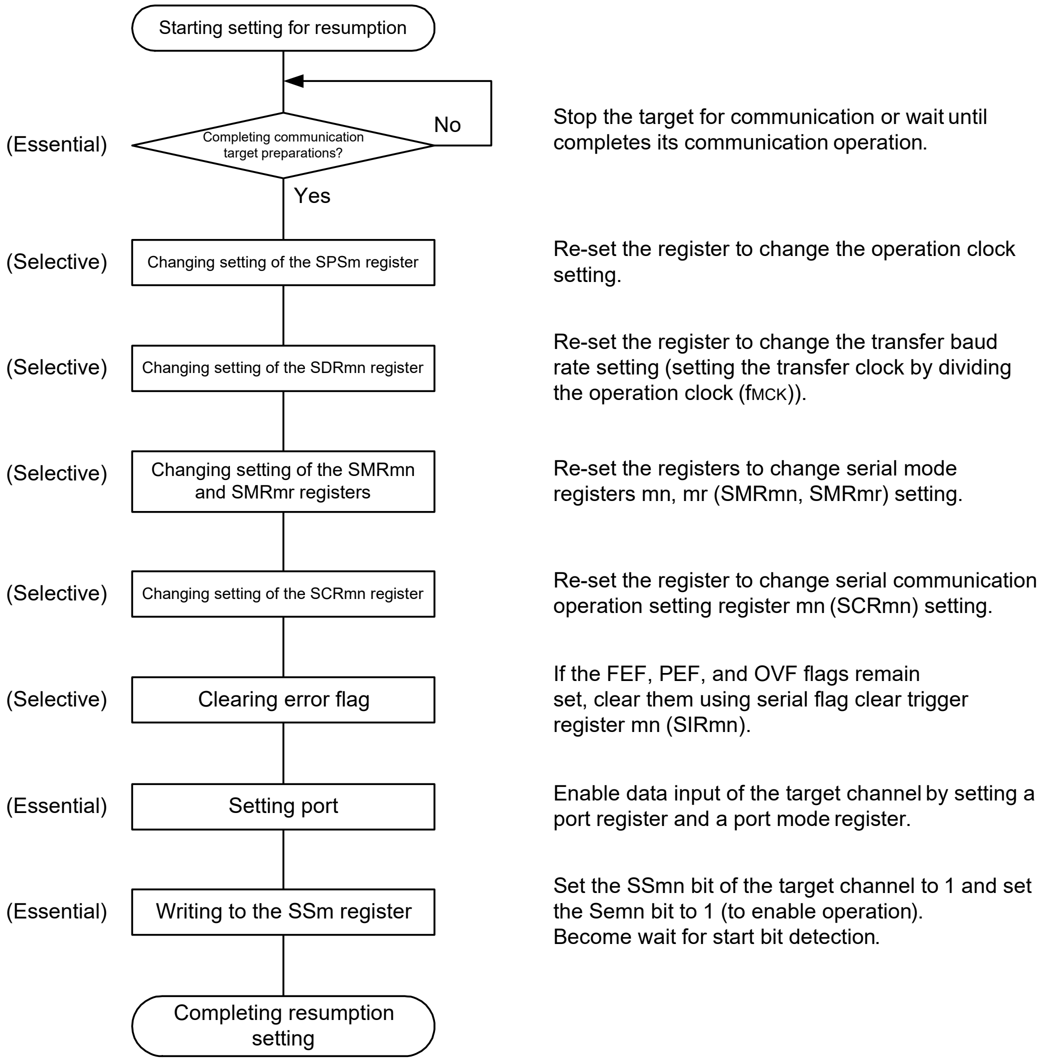
If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait).

Write 1 to the STmn bit of the target channel.
(SEmn = 0: to operation stop status)

Reset the serial array unit by stopping the clock supply to it.

After the stop setting is completed, go to the next processing.

Figure 13 - 112 Procedure for Resuming UART Reception

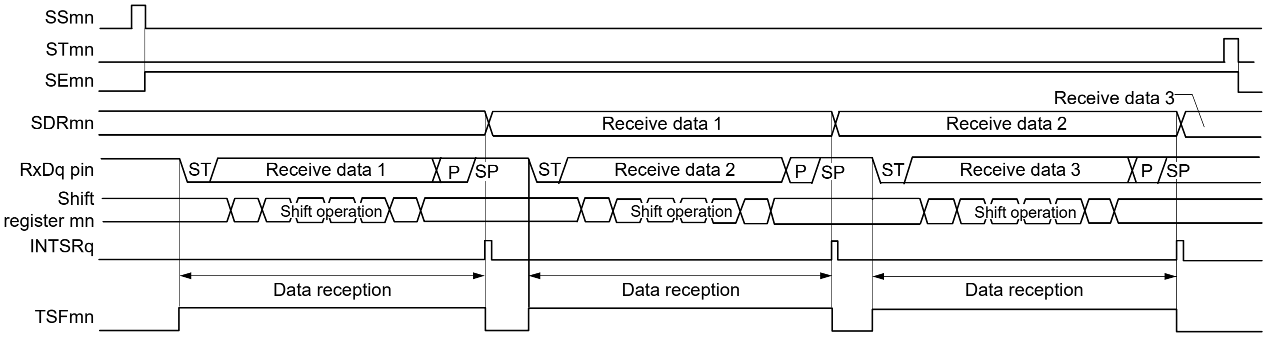

Starting setting for resumption
(Essential)
Completing communication target preparations?
No
Yes
Stop the target for communication or wait until completes its communication operation.
(Selective)
Changing setting of the SPSm register
Re-set the register to change the operation clock setting.
(Selective)
Changing setting of the SDRmn register
Re-set the register to change the transfer baud rate setting (setting the transfer clock by dividing the operation clock (fMCK)).
(Selective)
Changing setting of the SMRmn and SMRmr registers
Re-set the registers to change serial mode registers mn, mr (SMRmn, SMRmr) setting.
(Selective)
Changing setting of the SCRmn register
Re-set the register to change serial communication operation setting register mn (SCRmn) setting.
(Selective)
Clearing error flag
If the FEF, PEF, and OVF flags remain set, clear them using serial flag clear trigger register mn (SIRmn).
(Essential)
Setting port
Enable data input of the target channel by setting a port register and a port mode register.
(Essential)
Writing to the SSm register
Set the SSmn bit of the target channel to 1 and set the Semn bit to 1 (to enable operation).
Become wait for start bit detection.
Completing resumption setting

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.

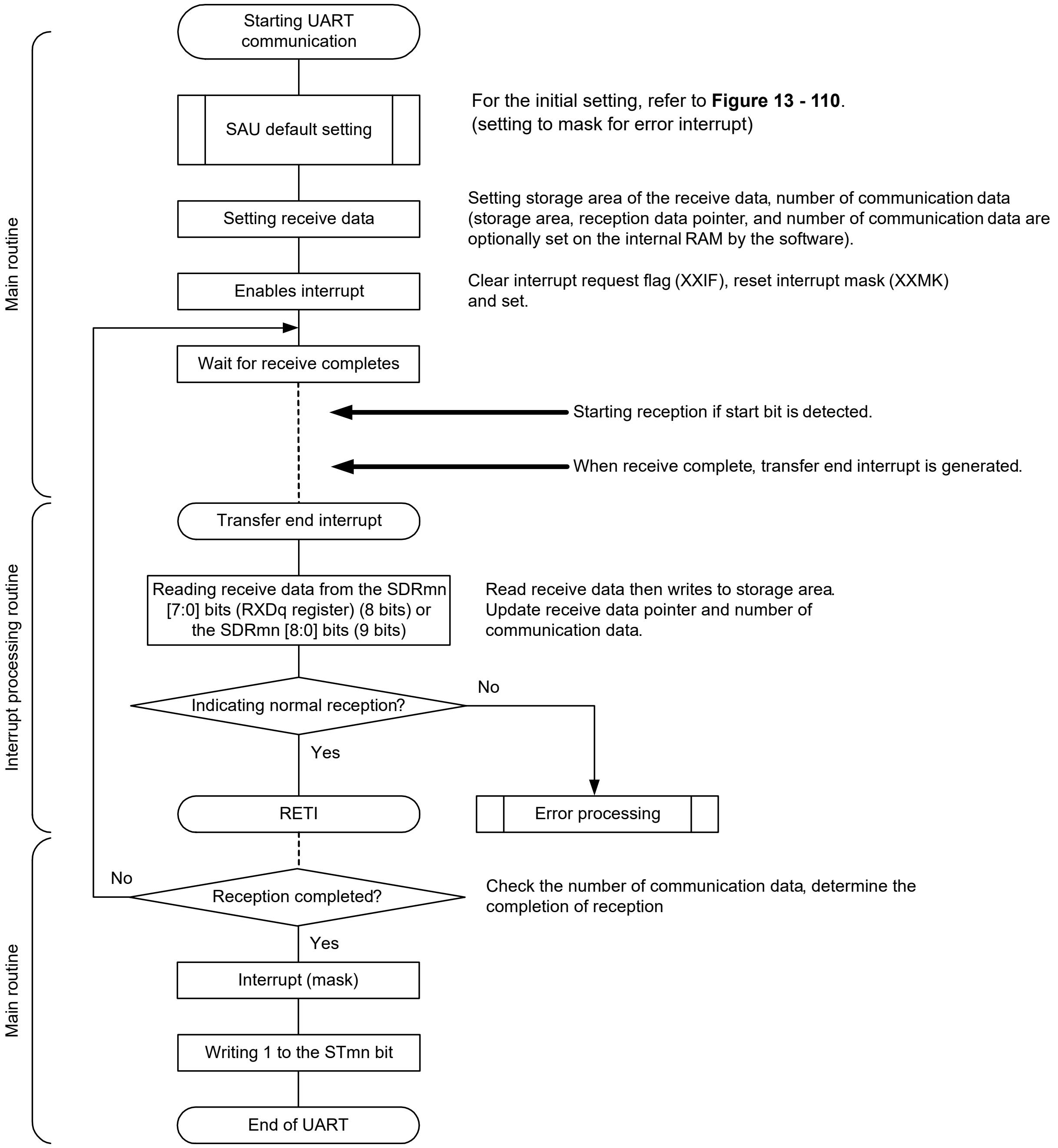
(3) Processing flow

Figure 13 - 113 Timing Chart of UART Reception



Remark m: Unit number (m = 0), n: Channel number (n = 1, 3), mn = 01, 03,
r: Channel number (r = n - 1), q: UART number (q = 0, 1)

Figure 13 - 114 Flowchart of UART Reception


Main routine
Starting UART communication
SAU default setting
For the initial setting, refer to Figure 13 - 110.
(setting to mask for error interrupt)
Setting receive data
Setting storage area of the receive data, number of communication data (storage area, reception data pointer, and number of communication data are optionally set on the internal RAM by the software).
Enables interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set.
Wait for receive completes
Starting reception if start bit is detected.
When receive complete, transfer end interrupt is generated.
Interrupt processing routine
Transfer end interrupt
Reading receive data from the SDRmn [7:0] bits (RXDq register) (8 bits) or the SDRmn [8:0] bits (9 bits)
Read receive data then writes to storage area. Update receive data pointer and number of communication data.
Indicating normal reception?
No
Yes
RETI
Error processing
Main routine
No
Reception completed?
Check the number of communication data, determine the completion of reception
Yes
Interrupt (mask)
Writing 1 to the STmn bit
End of UART

13.7.3 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 13 - 117** and **13 - 119 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 13 - 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. The CPU shifts from the STOP mode to the SNOOZE mode on detecting the valid edge of the RxDq signal. Note, however, that transfer through the UART channel may not start and the CPU may remain in the SNOOZE mode if an input pulse on the RxDq pin is too short to be detected as a start bit.

In such cases, data may not be received correctly, and this may lead to a framing error or parity error in the next UART transfer.

Caution 6. When using UART reception to transition from STOP mode to SNOOZE mode, use the high-speed on-chip oscillator.

Table 13 - 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.

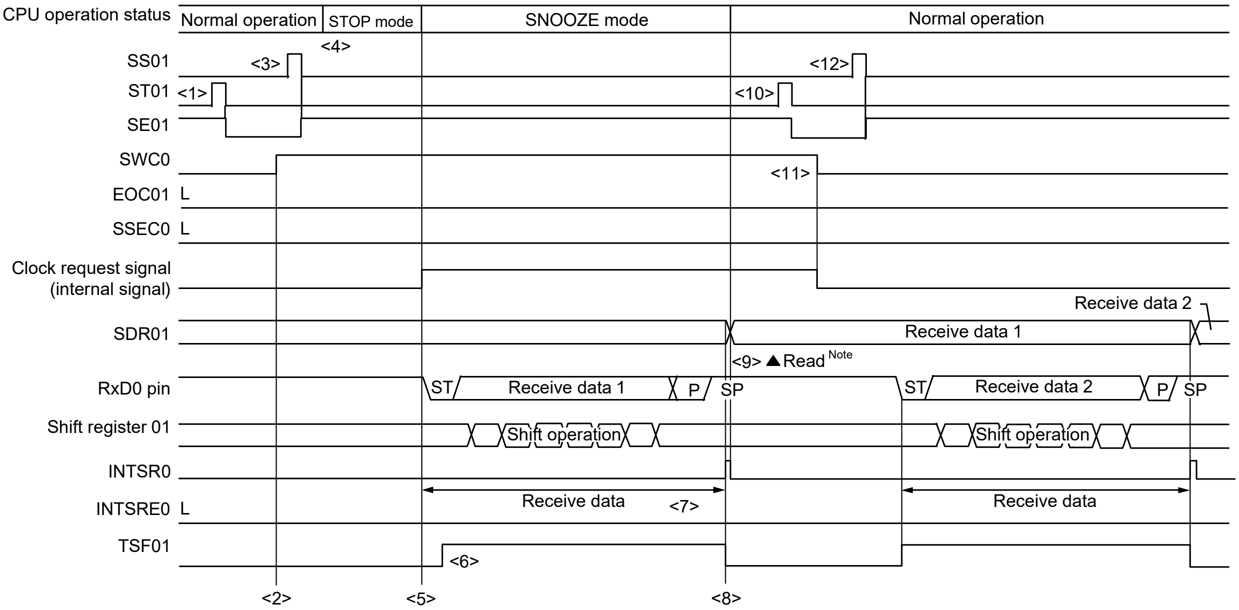
- In the case of f_{IH} ± 1.5%, perform (Maximum permissible value - 0.5%) and (Minimum permissible value + 0.5%) to the values in the above table.
- In the case of f_{IH} ± 2.0%, perform (Maximum permissible value - 1.0%) and (Minimum permissible value + 1.0%) to the values in the above table.

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 13 - 115 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)



Note Read the received data when SWCm is 1.

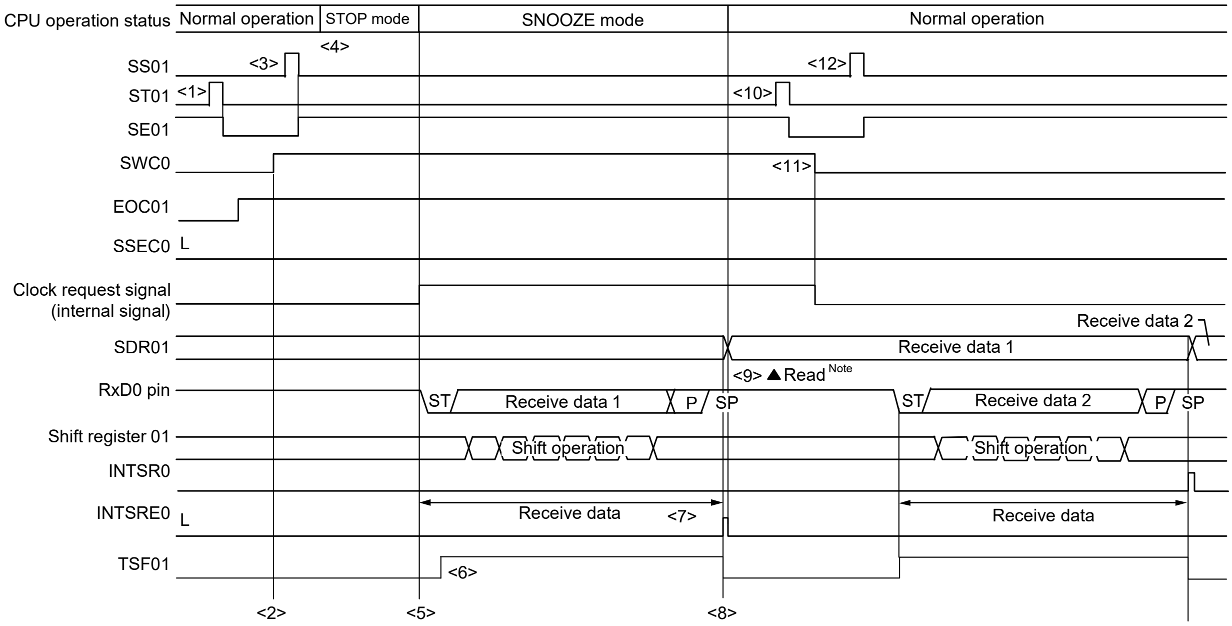
Caution Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit, and stop the operation).
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

Remark 1. <1> to <12> in the figure correspond to <1> to <12> in Figure 13 - 117 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0).

Remark 2. m = 0; q = 0, 1

(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 13 - 116 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)



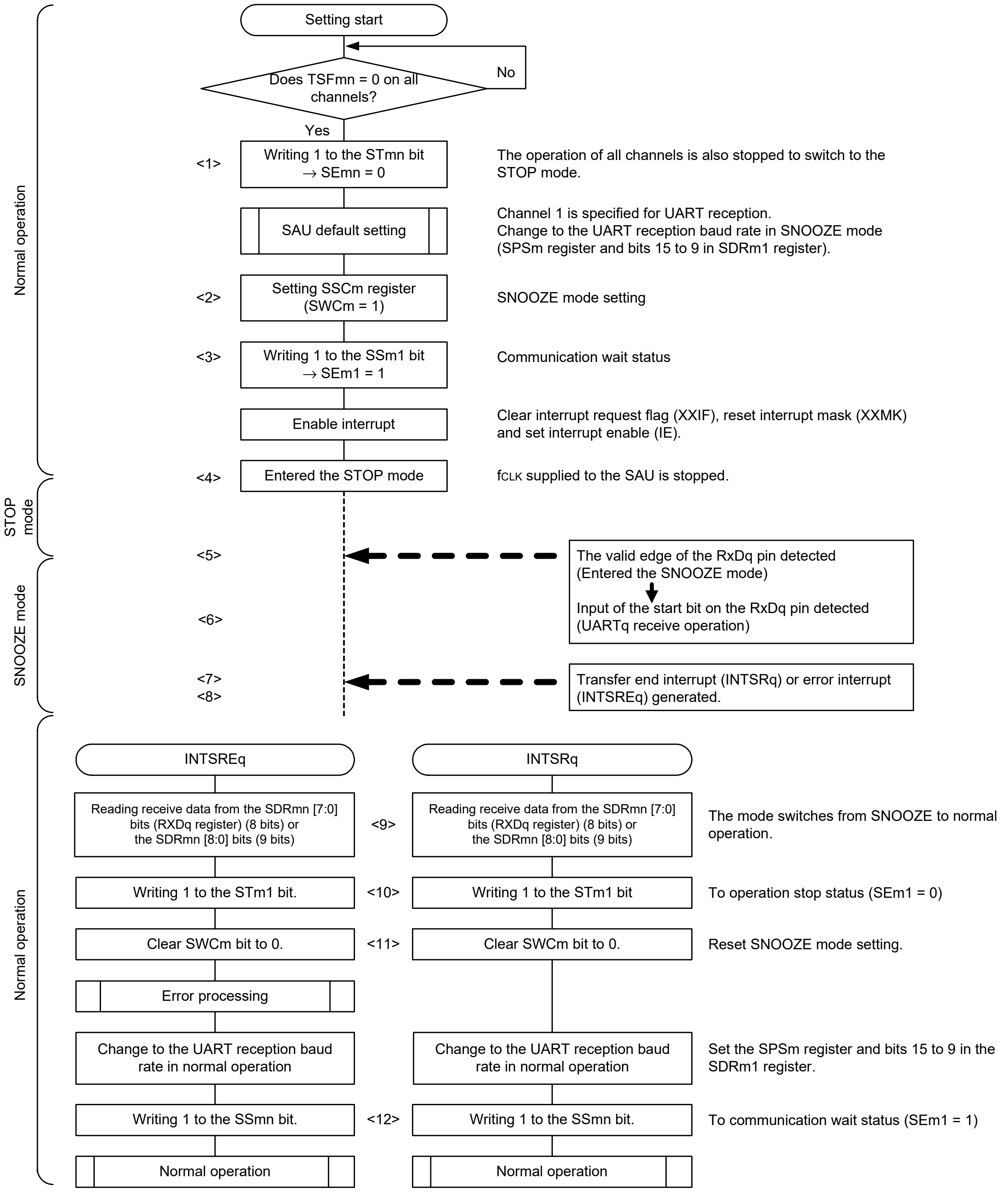
Note Read the received data when SWCm = 1.

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 13 - 117 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0).

Remark 2. m = 0; q = 0, 1

Figure 13 - 117 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)


Normal operation
STOP mode
SNOOZE mode
Normal operation
Setting start
Does TSFmn = 0 on all channels?
No
Yes
<1>
Writing 1 to the STmn bit → SEmn = 0
The operation of all channels is also stopped to switch to the STOP mode.
SAU default setting
Channel 1 is specified for UART reception.
Change to the UART reception baud rate in SNOOZE mode (SPSm register and bits 15 to 9 in SDRm1 register).
<2>
Setting SSCm register (SWCm = 1)
SNOOZE mode setting
<3>
Writing 1 to the SSm1 bit → SEm1 = 1
Communication wait status
Enable interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt enable (IE).
<4>
Entered the STOP mode
fCLK supplied to the SAU is stopped.
<5>
The valid edge of the RxDq pin detected (Entered the SNOOZE mode)
<6>
Input of the start bit on the RxDq pin detected (UARTq receive operation)
<7>
<8>
Transfer end interrupt (INTSRq) or error interrupt (INTSREq) generated.
INTSREq
INTSRq
Reading receive data from the SDRmn [7:0] bits (RXDq register) (8 bits) or the SDRmn [8:0] bits (9 bits)
<9>
Reading receive data from the SDRmn [7:0] bits (RXDq register) (8 bits) or the SDRmn [8:0] bits (9 bits)
The mode switches from SNOOZE to normal operation.
Writing 1 to the STm1 bit.
<10>
Writing 1 to the STm1 bit
To operation stop status (SEm1 = 0)
Clear SWCm bit to 0.
<11>
Clear SWCm bit to 0.
Reset SNOOZE mode setting.
Error processing
Change to the UART reception baud rate in normal operation
Change to the UART reception baud rate in normal operation
Set the SPSm register and bits 15 to 9 in the SDRm1 register.
Writing 1 to the SSmn bit.
<12>
Writing 1 to the SSmn bit.
To communication wait status (SEm1 = 1)
Normal operation
Normal operation

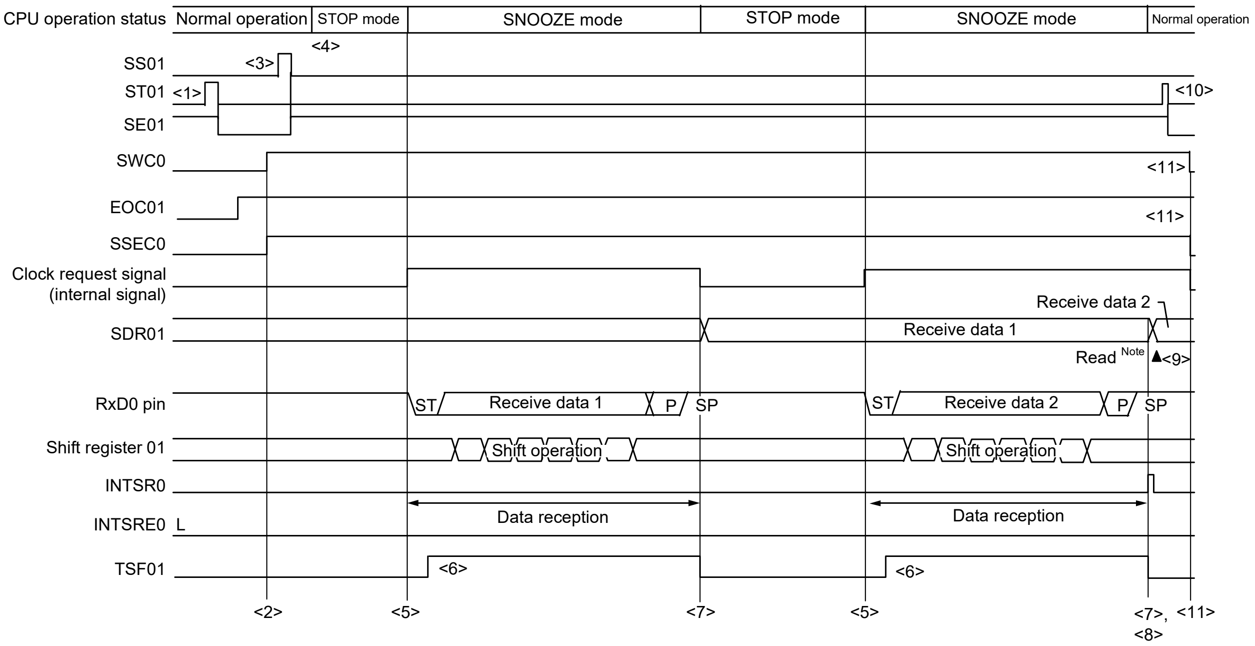
Remark 1. <1> to <12> in the figure correspond to <1> to <12> in Figure 13 - 115 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1) and Figure 13 - 116 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0).

Remark 2. m = 0; q = 0, 1

(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 13 - 118 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).

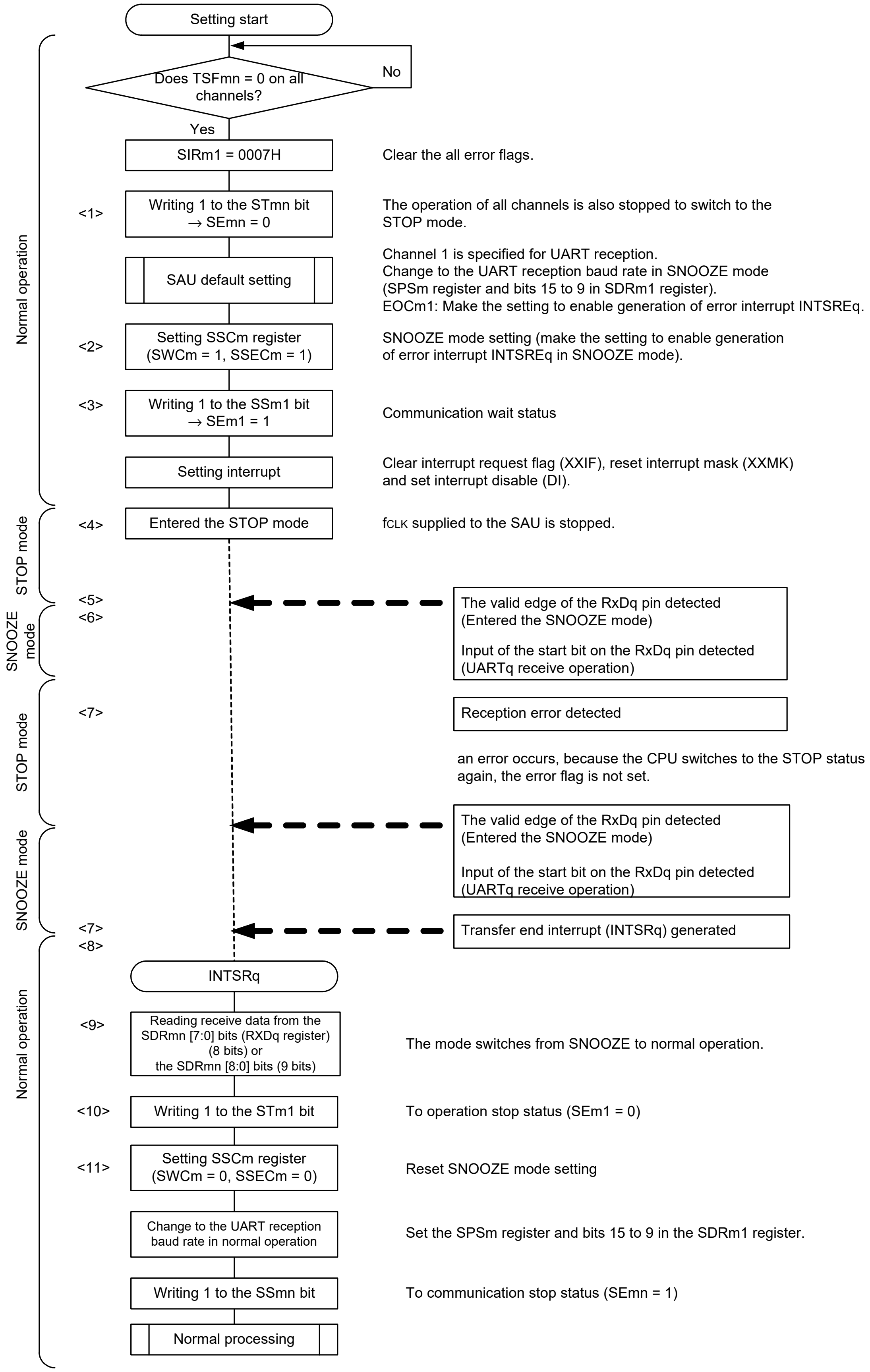
And after completion the receive operation, also clearing SWCm bit to 0 (SNOOZE mode release).

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 13 - 119 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1).

Remark 2. m = 0; q = 0, 1

Figure 13 - 119 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)


Setting start
Does TSFmn = 0 on all channels?
No
Yes
SIRm1 = 0007H
Clear the all error flags.
<1>
Writing 1 to the STmn bit → SEmn = 0
The operation of all channels is also stopped to switch to the STOP mode.
SAU default setting
Channel 1 is specified for UART reception.
Change to the UART reception baud rate in SNOOZE mode (SPSm register and bits 15 to 9 in SDRm1 register).
EOCm1: Make the setting to enable generation of error interrupt INTSREq.
<2>
Setting SSCm register (SWCm = 1, SSECm = 1)
SNOOZE mode setting (make the setting to enable generation of error interrupt INTSREq in SNOOZE mode).
<3>
Writing 1 to the SSm1 bit → SEm1 = 1
Communication wait status
Setting interrupt
Clear interrupt request flag (XXIF), reset interrupt mask (XXMK) and set interrupt disable (DI).
Normal operation
<4>
Entered the STOP mode
fCLK supplied to the SAU is stopped.
STOP mode
<5>
<6>
The valid edge of the RxDq pin detected (Entered the SNOOZE mode)
Input of the start bit on the RxDq pin detected (UARTq receive operation)
SNOOZE mode
<7>
Reception error detected
an error occurs, because the CPU switches to the STOP status again, the error flag is not set.
STOP mode
The valid edge of the RxDq pin detected (Entered the SNOOZE mode)
Input of the start bit on the RxDq pin detected (UARTq receive operation)
SNOOZE mode
<7>
<8>
Transfer end interrupt (INTSRq) generated
INTSRq
<9>
Reading receive data from the SDRmn [7:0] bits (RXDq register) (8 bits) or the SDRmn [8:0] bits (9 bits)
The mode switches from SNOOZE to normal operation.
<10>
Writing 1 to the STm1 bit
To operation stop status (SEm1 = 0)
<11>
Setting SSCm register (SWCm = 0, SSECm = 0)
Reset SNOOZE mode setting
Change to the UART reception baud rate in normal operation
Set the SPSm register and bits 15 to 9 in the SDRm1 register.
Writing 1 to the SSmn bit
To communication stop status (SEmn = 1)
Normal processing
Normal operation

(**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 13 - 118 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1).

Remark 2. m = 0; q = 0, 1

13.7.4 Calculating baud rate

(1) Baud rate calculation expression

The baud rate for UART (UART0, UART1) communication can be calculated by the following expressions.

(Baud rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [bps]

Caution Setting serial data register mn (SDRmn) SDRmn[15:9] = (0000000B, 0000001B) is prohibited.

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 = 0 to 3), mn = 00 to 03

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 13 - 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 = 0 to 3), mn = 00 to 03

(2) Baud rate error during transmission

The baud rate error of UART (UART0, UART1) communication during transmission can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

(Baud rate error) = (Calculated baud rate value) ÷ (Target baud rate) × 100 - 100 [%]

Here is an example of setting a UART baud rate at f_{CLK} = 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, 2), mn = 00, 02

(3) Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UART0, UART1) communication can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

$$\text{(Maximum receivable baud rate)} = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times Brate$$

$$\text{(Minimum receivable baud rate)} = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times Brate$$

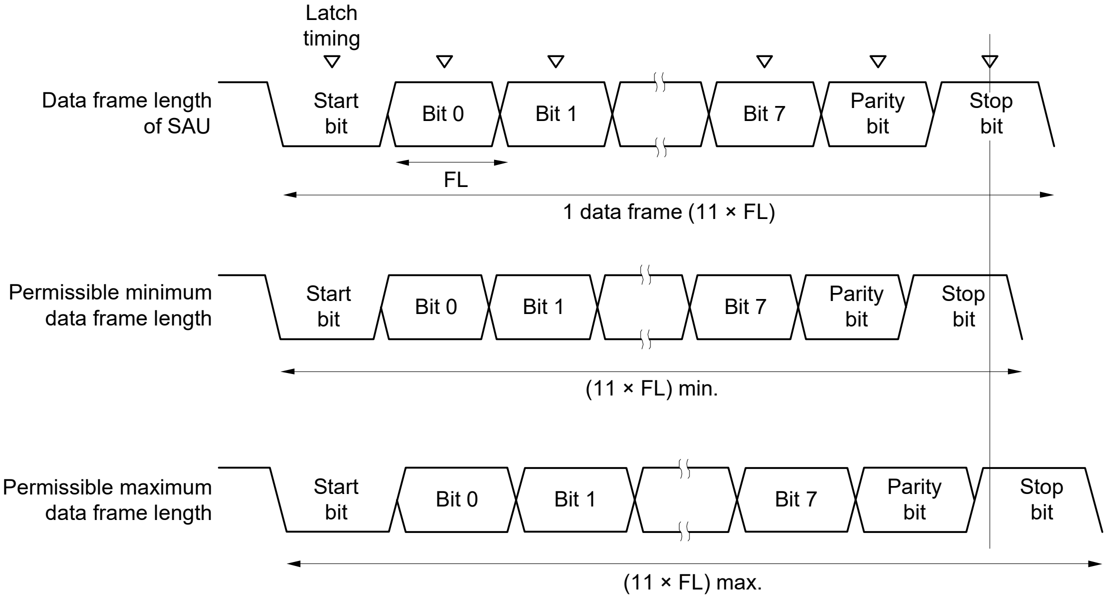
Brate: Calculated baud rate value at the reception side (See **13.7.4 (1) Baud rate calculation expression**.)

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), n: Channel number (n = 1, 3), mn = 01, 03

Figure 13 - 120 Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in Figure 13 - 120, 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.

13.7.5 Procedure for processing errors that occurred during UART (UART0, UART1) communication

The procedure for processing errors that occurred during UART (UART0, UART1) communication is described in Figures 13 - 121 and 13 - 122.

Figure 13 - 121 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 13 - 122 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 to 3), mn = 00 to 03

13.8 LIN Communication Operation

13.8.1 LIN transmission

Of UART transmission, UART0 support LIN communication.

For LIN transmission, channel 0 of is used.

UART	UART0	UART1
Support of LIN communication	Supported	Not supported
Target channel	Channel 0 of SAU0	—
Pins used	TxD0	—
Interrupt	INTST0	
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	None	
Transfer data length	8 bits	
Transfer rate [Note]	Max. f_{MCK}/6 [bps] (SDR00 [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]	
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)	
Parity bit	No parity bit	
Stop bit	Appending 1 bit	
Data direction	MSB or LSB first	

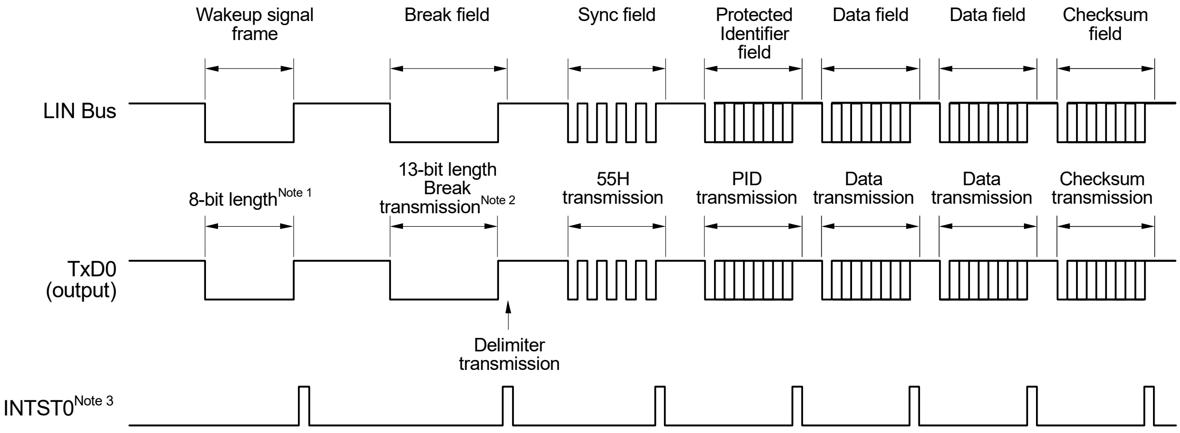
Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 35**, **CHAPTER 36 ELECTRICAL SPECIFICATIONS**). In addition, LIN communication is usually 2.4/9.6/19.2 kbps is often used.

Remark f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol designed to reduce the cost of an automobile network. Communication of LIN is single-master communication and up to 15 slaves can be connected to one master. The slaves are used to control switches, actuators, and sensors, which are connected to the master via LIN.
Usually, the master is connected to a network such as CAN (Controller Area Network). A LIN bus is a single-wire bus to which nodes are connected via transceiver conforming to ISO9141.
According to the protocol of LIN, the master transmits a frame by attaching baud rate information to it. A slave receives this frame and corrects a baud rate error from the master. If the baud rate error of a slave is within ±15%, communication can be established.

Figure 13 - 123 outlines a transmission operation of LIN.

Figure 13 - 123 Transmission Operation of LIN



Note 1. Set the baud rate in accordance with the wakeup signal regulations and transmit data of 80H.

Note 2. A break field is defined to have a width of 13 bits and output a low level. Where the baud rate for main transfer is N [bps], therefore, the baud rate of the break field is calculated as follows.

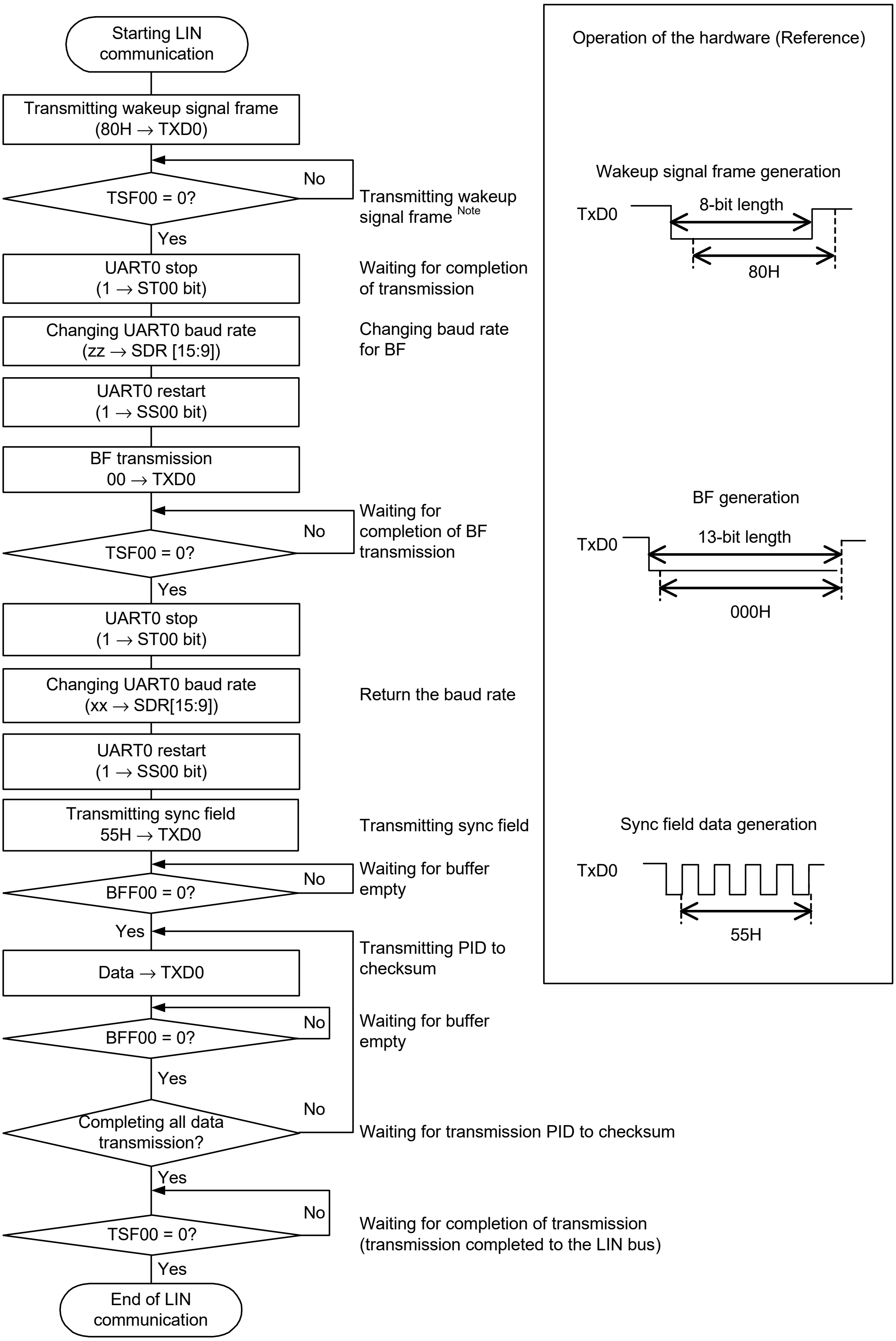
$$\boxed{\text{(Baud rate of break field)} = 9/13 \times N}$$

By transmitting data of 00H at this baud rate, a break field is generated.

Note 3. INTST0 is output upon completion of transmission. INTST0 is also output at BF transmission.

Remark The interval between fields is controlled by software.

Figure 13 - 124 Flowchart for LIN Transmission


Starting LIN communication
Transmitting wakeup signal frame (80H → TXD0)
TSF00 = 0?
No
Yes
Transmitting wakeup signal frame Note
UART0 stop (1 → ST00 bit)
Waiting for completion of transmission
Changing UART0 baud rate (zz → SDR [15:9])
Changing baud rate for BF
UART0 restart (1 → SS00 bit)
BF transmission 00 → TXD0
TSF00 = 0?
No
Yes
Waiting for completion of BF transmission
UART0 stop (1 → ST00 bit)
Changing UART0 baud rate (xx → SDR[15:9])
Return the baud rate
UART0 restart (1 → SS00 bit)
Transmitting sync field 55H → TXD0
Transmitting sync field
BFF00 = 0?
No
Yes
Waiting for buffer empty
Data → TXD0
Transmitting PID to checksum
BFF00 = 0?
No
Yes
Waiting for buffer empty
Completing all data transmission?
No
Yes
Waiting for transmission PID to checksum
TSF00 = 0?
No
Yes
Waiting for completion of transmission (transmission completed to the LIN bus)
End of LIN communication
Operation of the hardware (Reference)
Wakeup signal frame generation
TxD0
8-bit length
80H
BF generation
TxD0
13-bit length
000H
Sync field data generation
TxD0
55H

Note When LIN-bus start from sleep status only.

Remark Default setting of the UART is complete, and the flow from the transmission enable status.

13.8.2 LIN reception

Of UART reception, UART0 support LIN communication.

For LIN reception, channel 1 is used.

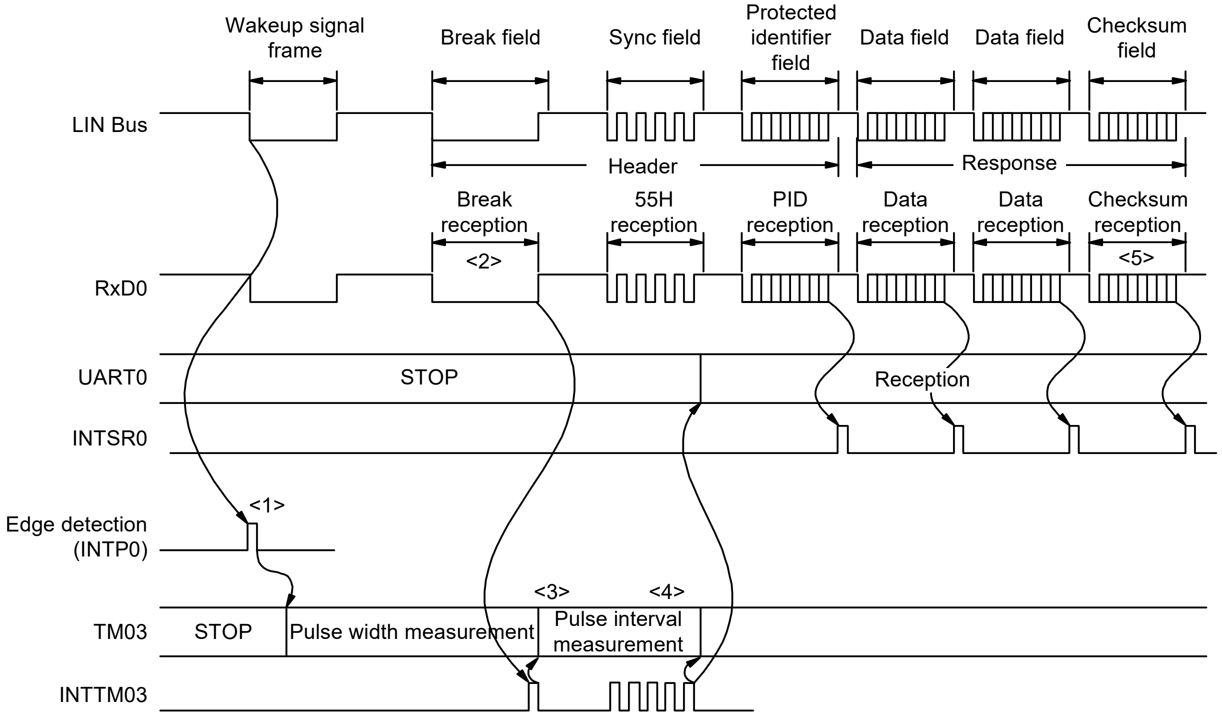
UART	UART0	UART1
Support of LIN communication	Supported	Not supported
Target channel	Channel 1 of SAU0	—
Pins used	RxD0	—
Interrupt	INTSR0	—
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error interrupt	INTSRE0	—
Error detection flag	• Framing error detection flag (FEF01) • Overrun error detection flag (OVF01)	
Transfer data length	8 bits	
Transfer rate Note	Max. $f_{MCK}/6$ [bps] (SDR01 [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]	
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)	
Parity bit	No parity bit (The parity bit is not checked.)	
Stop bit	Appending 1 bit	
Data direction	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 35**, **CHAPTER 36 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

Figure 13 - 125 outlines a reception operation of LIN.

Figure 13 - 125 Reception Operation of LIN



Here is the flow of signal processing.

<1> The wakeup signal is detected by detecting an interrupt edge (INTP0) on a pin. When the wakeup signal is detected, change TM03 to pulse width measurement upon detection of the wakeup signal to measure the low level width of the BF signal. Then wait for BF signal reception.

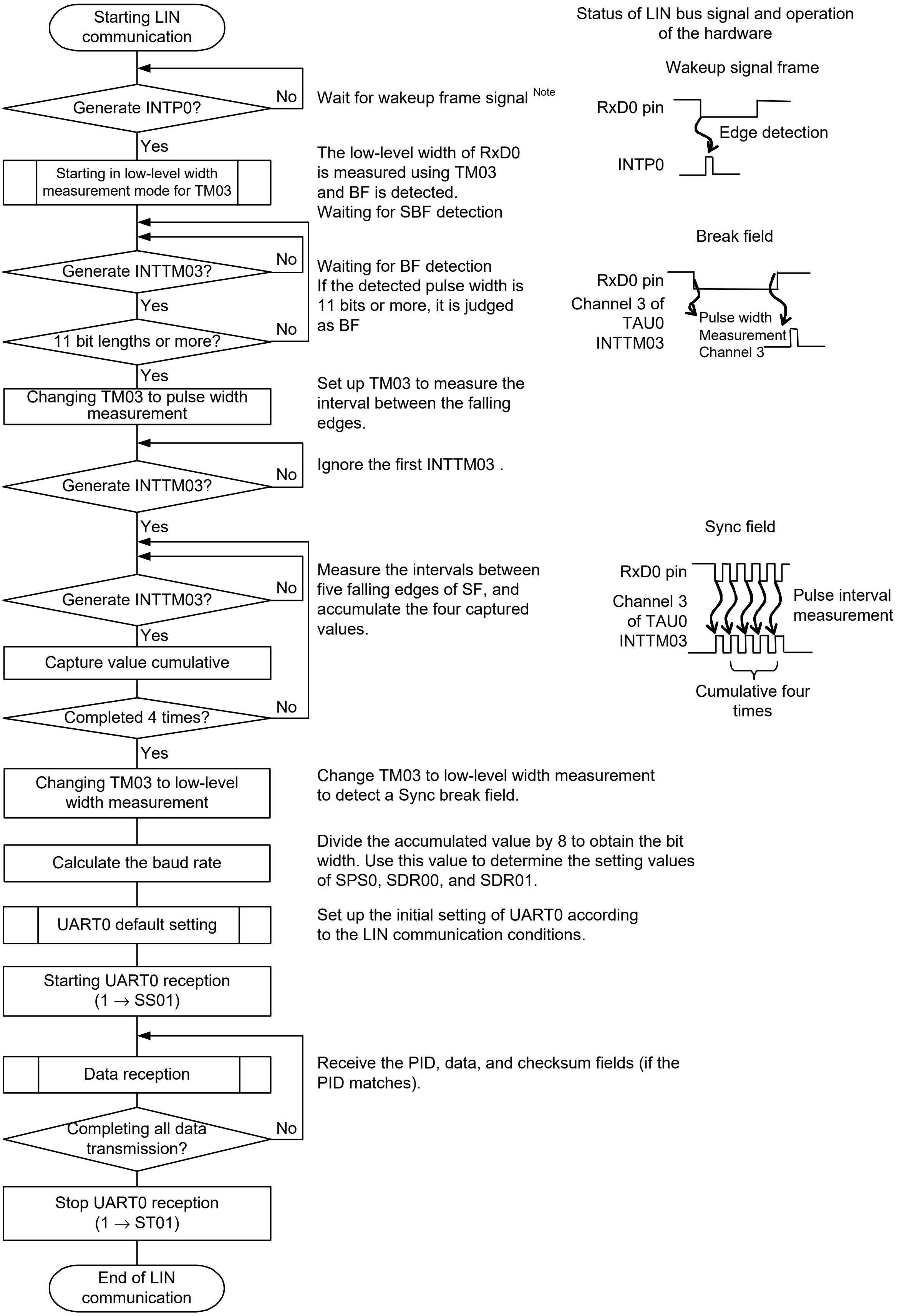
<2> TM03 starts measuring the low-level width upon detection of the falling edge of the BF signal, and then captures the data upon detection of the rising edge of the BF signal. The captured data is used to judge whether it is the BF signal.

<3> When the BF signal has been received normally, change TM03 to pulse interval measurement and measure the interval between the falling edges of the RxD0 signal in the Sync field four times (see **7.8.4 Operation as input pulse interval measurement**).

<4> Calculate a baud rate error from the bit interval of sync field (SF). Stop UART0 once and adjust (re-set) the baud rate.

<5> The checksum field should be distinguished by software. In addition, processing to initialize UART0 after the checksum field is received and to wait for reception of BF should also be performed by software.

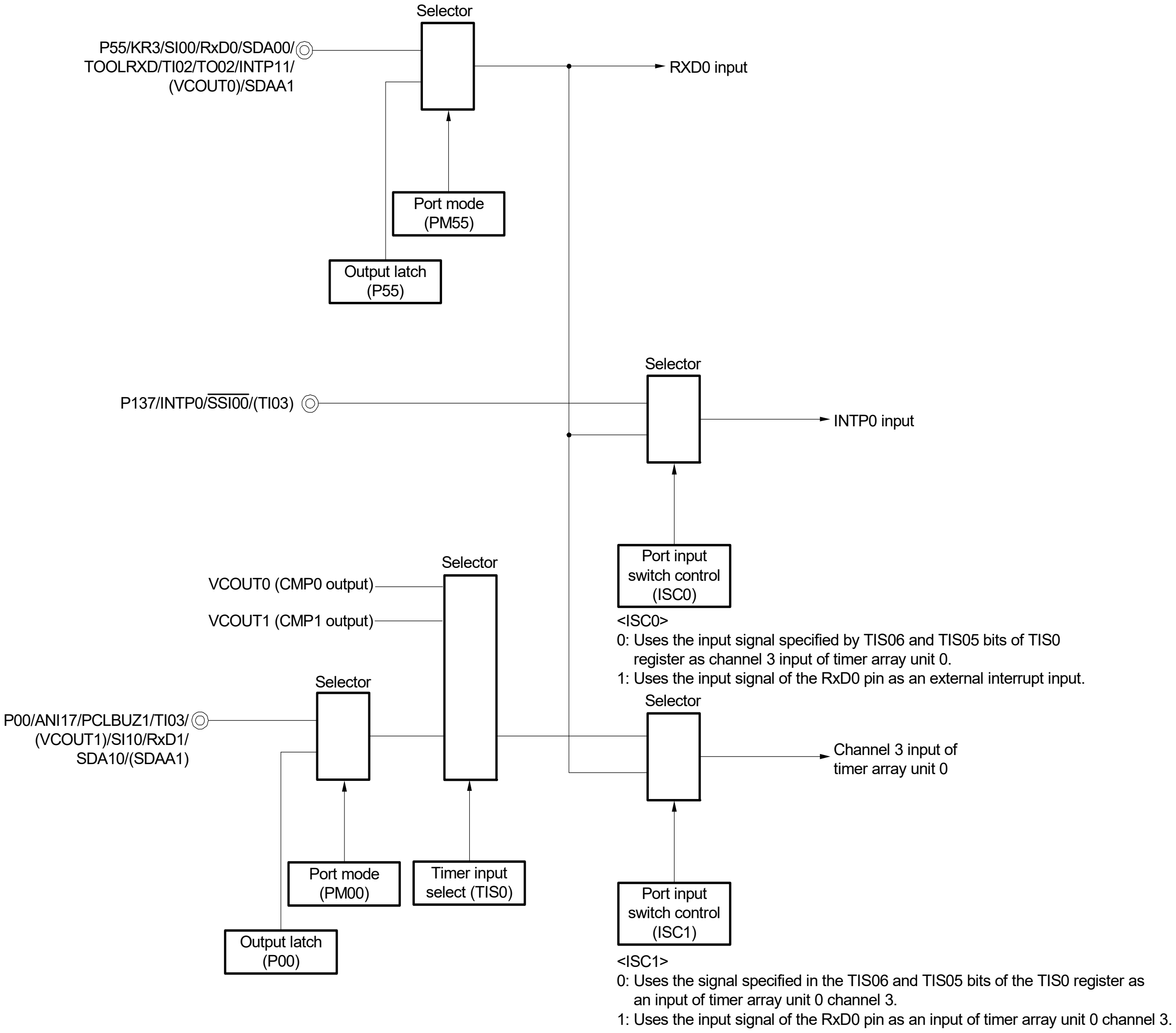
Figure 13 - 126 Flowchart for LIN Reception


Starting LIN communication
Generate INTP0?
No
Yes
Wait for wakeup frame signal Note
Status of LIN bus signal and operation of the hardware
Wakeup signal frame
RxD0 pin
Edge detection
INTP0
Starting in low-level width measurement mode for TM03
The low-level width of RxD0 is measured using TM03 and BF is detected. Waiting for SBF detection
Break field
Generate INTTM03?
No
Yes
11 bit lengths or more?
No
Yes
Waiting for BF detection If the detected pulse width is 11 bits or more, it is judged as BF
RxD0 pin
Channel 3 of TAU0 INTTM03
Pulse width Measurement Channel 3
Changing TM03 to pulse width measurement
Set up TM03 to measure the interval between the falling edges.
Generate INTTM03?
No
Yes
Ignore the first INTTM03 .
Sync field
Generate INTTM03?
No
Yes
Measure the intervals between five falling edges of SF, and accumulate the four captured values.
RxD0 pin
Channel 3 of TAU0 INTTM03
Pulse interval measurement
Cumulative four times
Capture value cumulative
Completed 4 times?
No
Yes
Changing TM03 to low-level width measurement
Change TM03 to low-level width measurement to detect a Sync break field.
Calculate the baud rate
Divide the accumulated value by 8 to obtain the bit width. Use this value to determine the setting values of SPS0, SDR00, and SDR01.
UART0 default setting
Set up the initial setting of UART0 according to the LIN communication conditions.
Starting UART0 reception (1 → SS01)
Data reception
Receive the PID, data, and checksum fields (if the PID matches).
Completing all data transmission?
No
Stop UART0 reception (1 → ST01)
End of LIN communication

Note Required in the sleep status only.

Figure 13 - 127 shows the configuration of a port that manipulates reception of LIN.
The wakeup signal transmitted from the master of LIN is received by detecting an edge of an external interrupt (INTP0). The length of the sync field transmitted from the master can be measured by using the external event capture operation of the timer array unit 0 to calculate a baud-rate error.
By controlling switch of port input (ISC0/ISC1), the input source of port input (RxD0) for reception can be input to the external interrupt pin (INTP0) and timer array unit

Figure 13 - 127 Port Configuration for Manipulating Reception of LIN



<ISC0>
0: Uses the input signal specified by TIS06 and TIS05 bits of TIS0 register as channel 3 input of timer array unit 0.
1: Uses the input signal of the RxD0 pin as an external interrupt input.

<ISC1>
0: Uses the signal specified in the TIS06 and TIS05 bits of the TIS0 register as an input of timer array unit 0 channel 3.
1: Uses the input signal of the RxD0 pin as an input of timer array unit 0 channel 3.

Remark ISC0, ISC1: Bits 0 and 1 of the input switch control register (ISC) (See **Figure 13 - 21**.)

The peripheral functions used for the LIN communication operation are as follows.

<Peripheral functions used>

- External interrupt (INTP0); Wakeup signal detection
  Usage: To detect an edge of the wakeup signal and the start of communication
- Channel 3 of timer array unit; Baud rate error detection, break field (BF) detection.
  Usage: To detect the length of the sync field (SF) and divide it by the number of bits in order to detect an error
  (The interval of the edge input to RxD0 is measured in the capture mode.)
  Measured the low-level width, determine whether break field (BF).
- Channels 0 and 1 (UART0) of serial array unit 0 (SAU0)

13.9 Operation of Simplified I²C (IIC00, IIC01, IIC10, IIC11) Communication

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This communication function is designed to execute single communication with devices such as EEPROM, flash memory, and A/D converter, and therefore, can be used only by the master.
Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.

[Data transmission/reception]
- Master transmission, master reception (only master function with a single master)
- ACK output function Note and ACK detection function
- Data length of 8 bits
  (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Generation of start condition and stop condition for software

[Interrupt function]
- Transfer end interrupt

[Error detection flag]
- Overrun error
- ACK error

* [Functions not supported by simplified I²C]
- Slave transmission, slave reception
- Multi-master function (arbitration loss detection function)
- Clock stretch detection function

Note When receiving the last data, ACK will not be output if 0 is written to the SOEmn (SOEm register) bit and serial communication data output is stopped. See the processing flow in **13.9.3 (2)** for details.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), mn = 00 to 03

The channel supporting simplified I^2C (IIC00, IIC01, IIC10, IIC11) is channels 0 to 3 of SAU0.

- 16-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	—	UART0 (LIN-bus supported)	—
	1	—		—
	2	CSI10	UART1	—
	3	CSI11		IIC11

- 20-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 (LIN-bus supported)	IIC00
	1	—		—
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11

- 24, 32-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 (LIN-bus supported)	IIC00
	1	CSI01		IIC01
	2	CSI10	UART1	IIC10
	3	CSI11		IIC11

Simplified I^2C (IIC00, IIC01, IIC10, IIC11) performs the following four types of communication operations.

• Address field transmission	(See **13.9.1**.)
• Data transmission	(See **13.9.2**.)
• Data reception	(See **13.9.3**.)
• Stop condition generation	(See **13.9.4**.)

13.9.1 Address field transmission

Address field transmission is a transmission operation that first executes in I²C 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 I²C	IIC00	IIC01	IIC10	IIC11
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0
Pins used	SCL00, SDA00 Note 1	SCL01, SDA01 Note 1	SCL10, SDA10 Note 1	SCL11, SDA11 Note 1
Interrupt	INTIIC00	INTIIC01	INTIIC10	INTIIC11
	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 I²C. • 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			

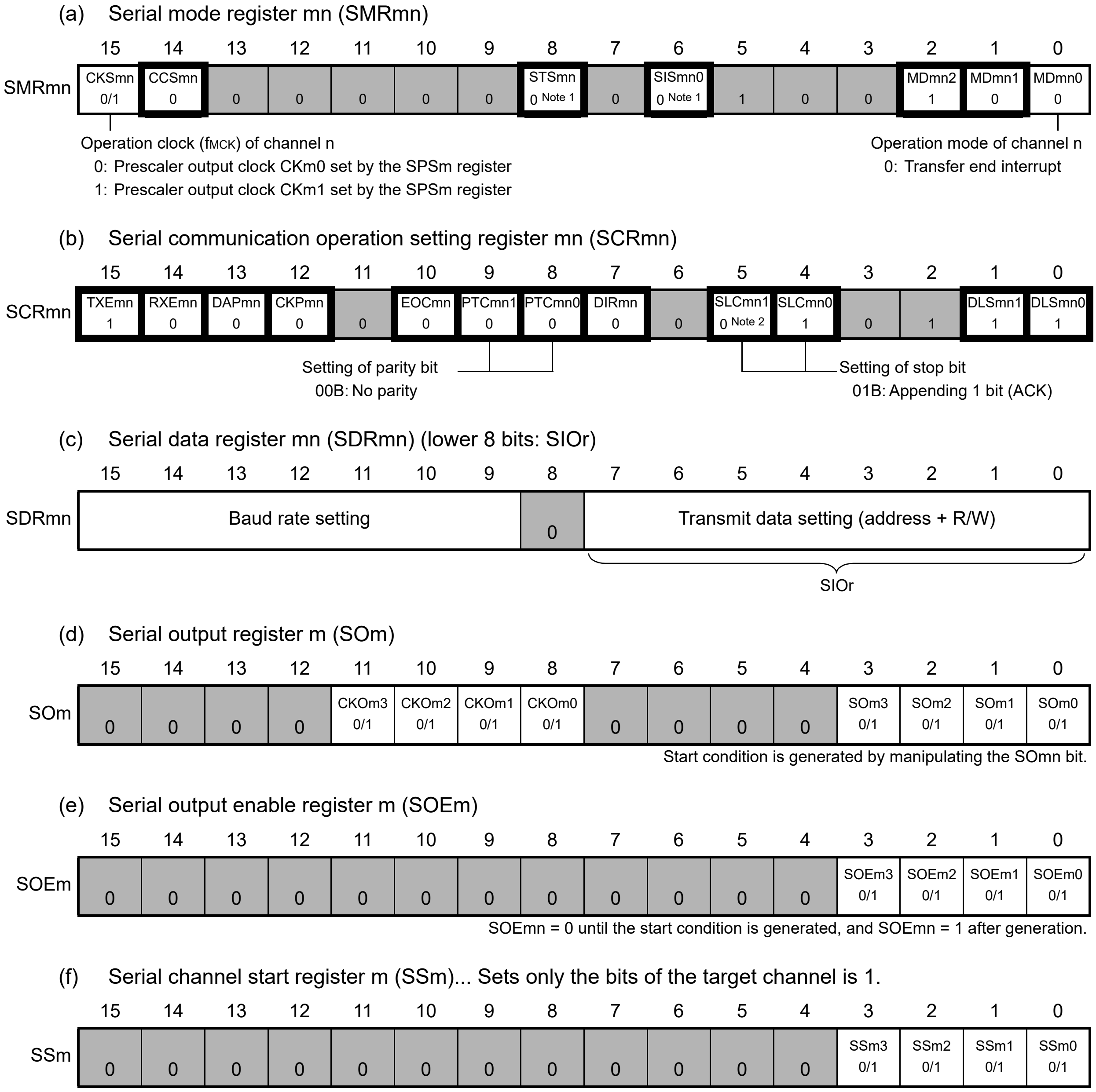
Note 1. To perform communication via simplified I²C, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) with the port output mode register (POMxx). For details, see **4.3 Registers Controlling Port Function** and **4.5 Register Settings When Using Alternate Function**.
When IIC00 is communicating 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, 3.0 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 35**, **CHAPTER 36 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), mn = 00 to 03

(1) Register setting

Figure 13 - 128 Example of Contents of Registers for Address Field Transmission of Simplified I²C (IIC00, IIC01, IIC10, IIC11)



(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)							

$\underbrace{\text{bits 7 to 0}}_{\text{SIOr}}$

(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	CKOm3 0/1	CKOm2 0/1	CKOm1 0/1	CKOm0 0/1	0	0	0	0	SOm3 0/1	SOm2 0/1	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	SOEm3 0/1	SOEm2 0/1	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	SSm3 0/1	SSm2 0/1	SSm1 0/1	SSm0 0/1

Note 1. Only provided for the SMR01 and SMR03 registers

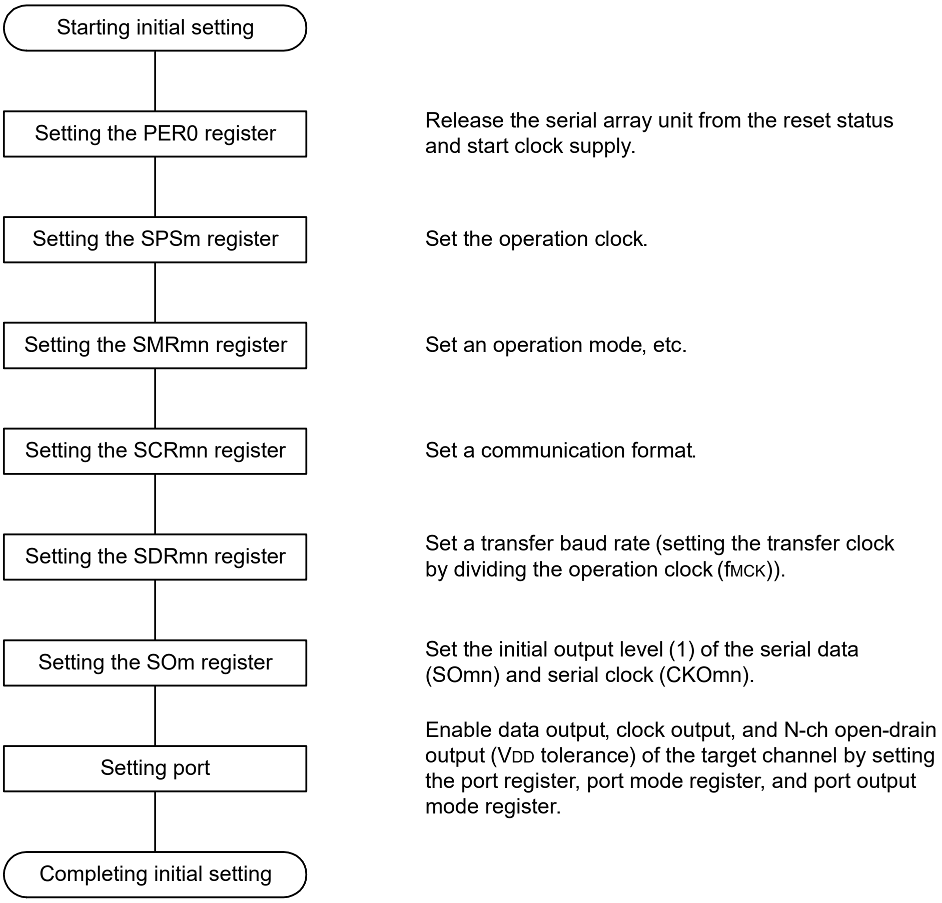
Note 2. Only provided for the SCR00 and SCR02 registers

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11), mn = 00 to 03

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

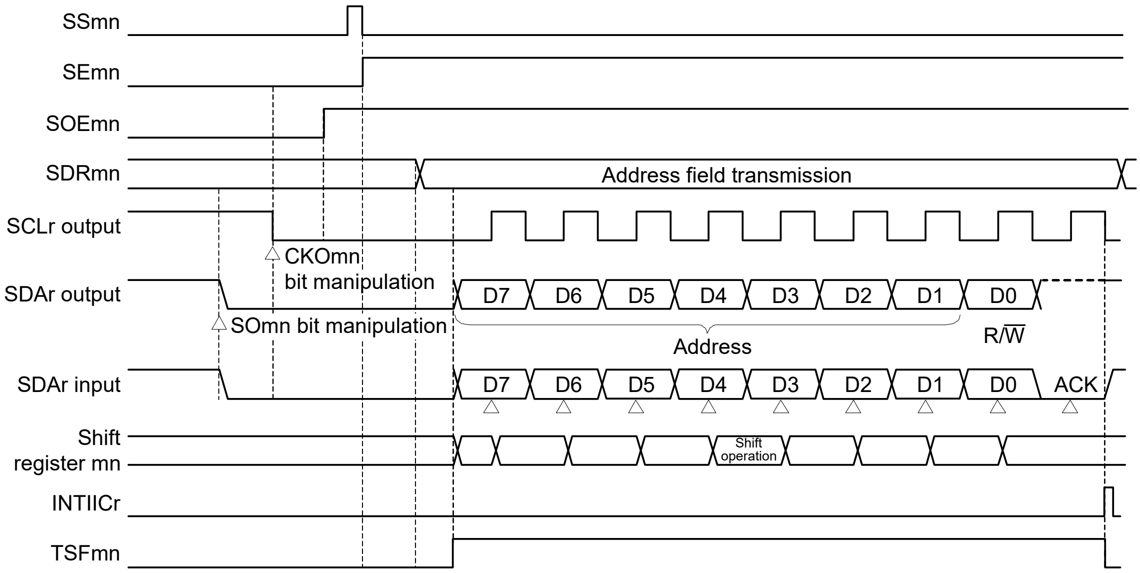
Figure 13 - 129 Initial Setting Procedure for Address Field Transmission


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 a transfer baud rate (setting the transfer clock by dividing the operation clock (fMCK)).
Setting the SOm register
Set the initial output level (1) of the serial data (SOmn) and serial clock (CKOmn).
Setting port
Enable data output, clock output, and N-ch open-drain output (VDD tolerance) of the target channel by setting the port register, port mode register, and port output mode register.
Completing initial setting

Remark At the end of the initial setting, the simplified I^{2}C (IIC00, IIC01, IIC10, IIC11) must be set so that output is disabled and operations are stopped.

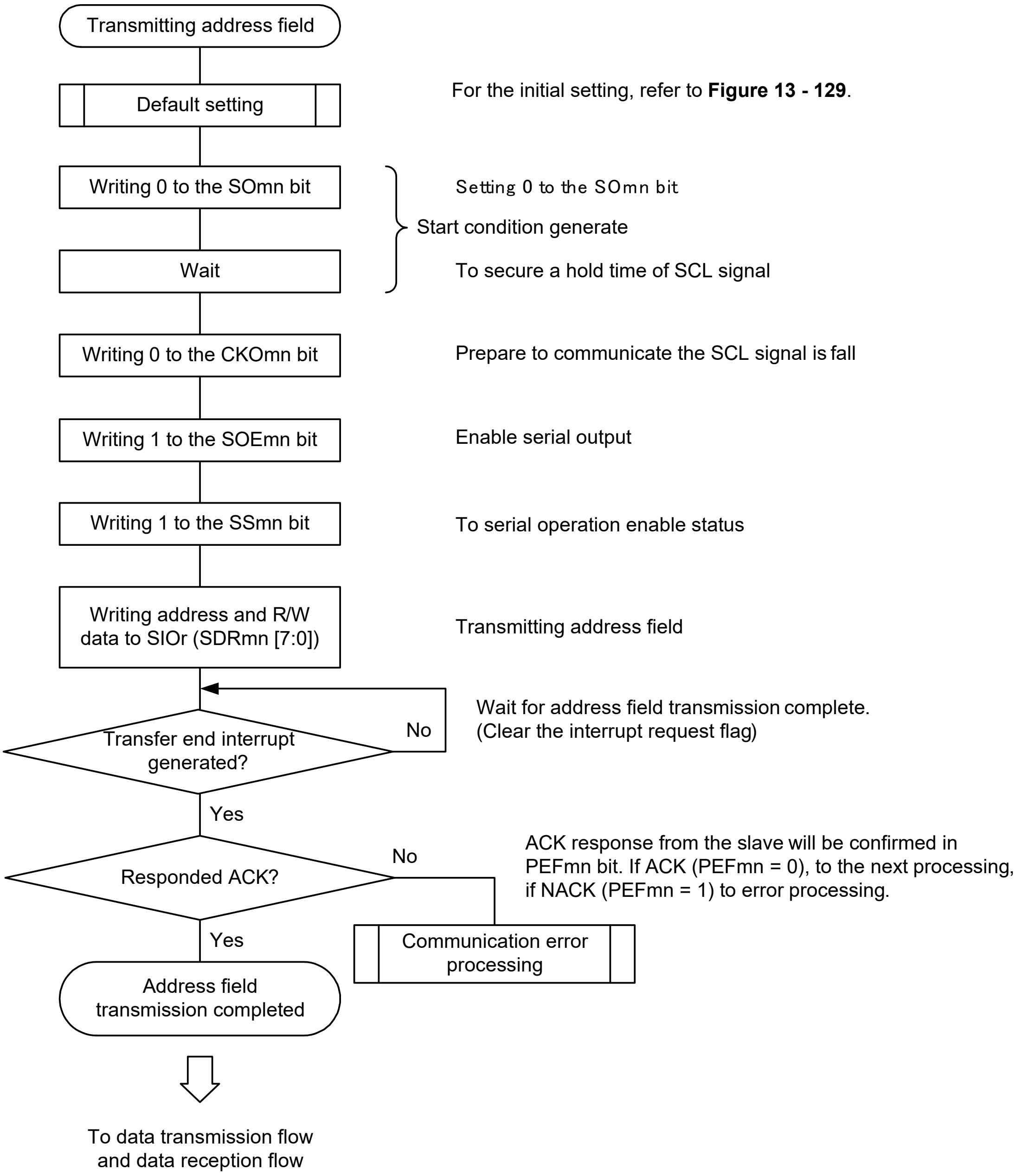
(3) Processing flow

Figure 13 - 130 Timing Chart of Address Field Transmission



Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11), mn = 00 to 03

Figure 13 - 131 Flowchart of Address Field Transmission


Transmitting address field
Default setting
Writing 0 to the SOmn bit
Wait
Writing 0 to the CKOmn bit
Writing 1 to the SOEmn bit
Writing 1 to the SSmn bit
Writing address and R/W data to SIOr (SDRmn [7:0])
Transfer end interrupt generated?
No
Yes
Responded ACK?
No
Yes
Communication error processing
Address field transmission completed
To data transmission flow and data reception flow

For the initial setting, refer to **Figure 13 - 129**.

Setting 0 to the SOmn bit

Start condition generate

To secure a hold time of SCL signal

Prepare to communicate the SCL signal is fall

Enable serial output

To serial operation enable status

Transmitting address field

Wait for address field transmission complete.
(Clear the interrupt request flag)

ACK response from the slave will be confirmed in PEFmn bit. If ACK (PEFmn = 0), to the next processing, if NACK (PEFmn = 1) to error processing.

13.9.2 Data transmission

Data transmission is an operation to transmit data to the target for transfer (slave) after transmission of an address field. After all data are transmitted to the slave, a stop condition is generated and the bus is released.

Simplified I²C	IIC00	IIC01	IIC10	IIC11
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0
Pins used	SCL00, SDA00 Note 1	SCL01, SDA01 Note 1	SCL10, SDA10 Note 1	SCL11, SDA11 Note 1
Interrupt	INTIIC00	INTIIC01	INTIIC10	INTIIC11
	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 I²C. • Max. 1 MHz (fast mode plus) • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)			
Data level	Non-reverse output (default: high level)			
Parity bit	No parity bit			
Stop bit	Appending 1 bit (for ACK reception timing)			
Data direction	MSB first			

Note 1. To perform communication via simplified I²C, set the N-ch open-drain output (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, 3.0 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 35**, **CHAPTER 36 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), mn = 00 to 03

(1) Register setting

Figure 13 - 132 Example of Contents of Registers for Data Transmission of Simplified I²C (IIC00, IIC01, IIC10, IIC11)

(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	14	13	12	11	10	9	8	7	6	5	4	3	2	1	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	CKOm3 0/1 Note 4	CKOm2 0/1 Note 4	CKOm1 0/1 Note 4	CKOm0 0/1 Note 4	0	0	0	0	SOm3 0/1 Note 4	SOm2 0/1 Note 4	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	SOEm3 1	SOEm2 1	SOEm1 1	SOEm0 1

(f) Serial channel start register m (SSm)... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 0/1	SSm1 0/1	SSm0 0/1

Note 1. Only provided for the SMR01 and SMR03 registers

Note 2. Only provided for the SCR00 and SCR02 registers

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 to 3), r: IIC number (r = 00, 01, 10, 11), mn = 00 to 03

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 13 - 133 Timing Chart of Data Transmission

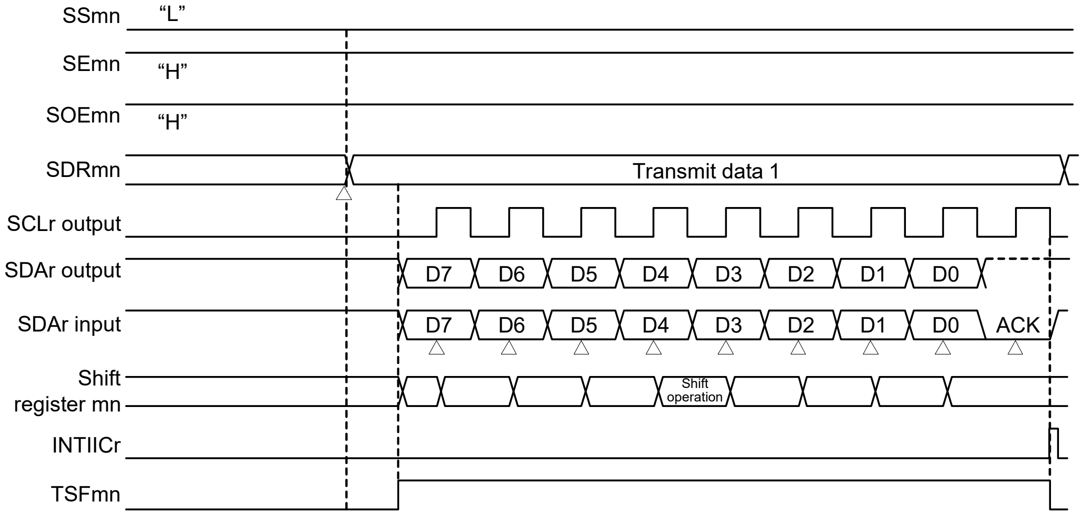
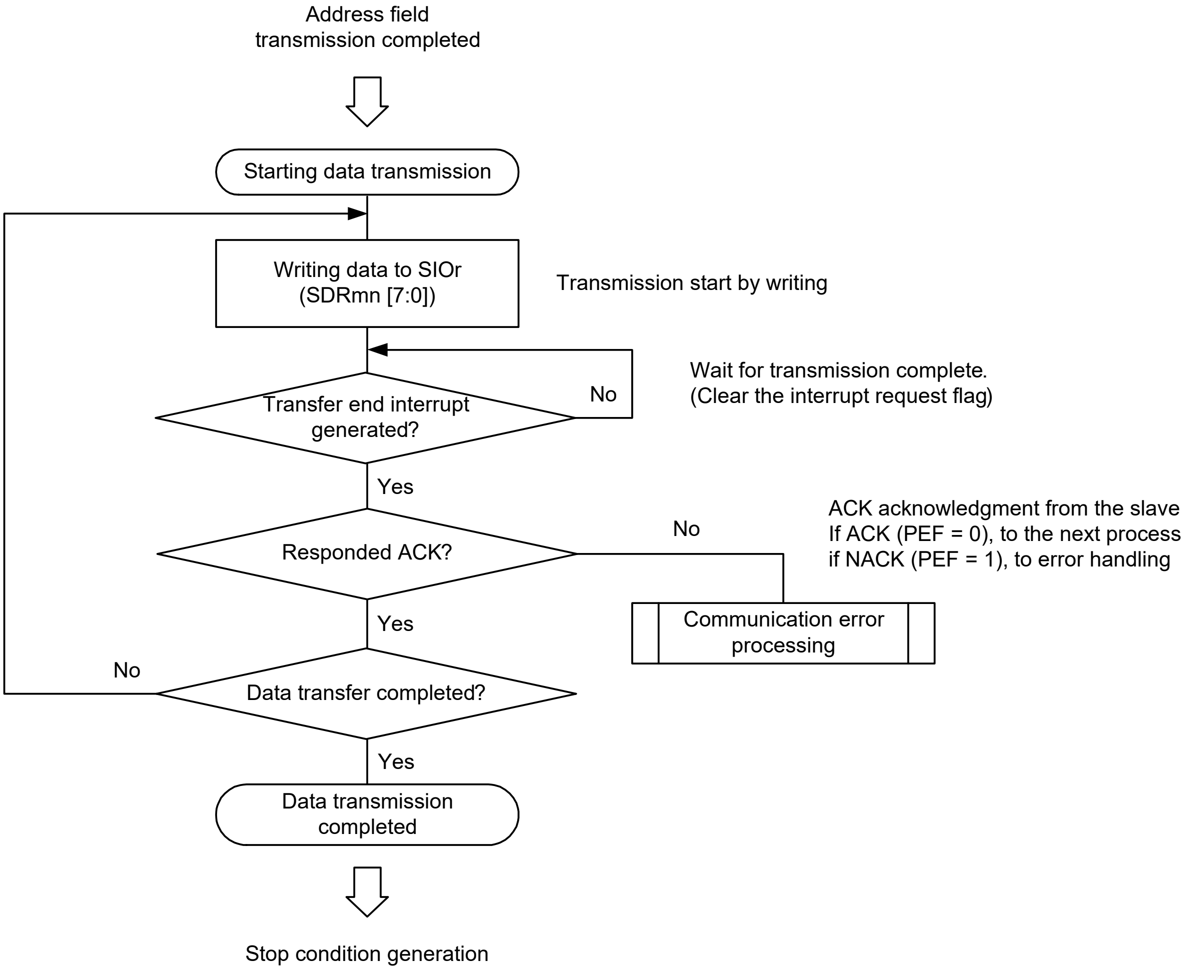


Figure 13 - 134 Flowchart of Data Transmission


Address field transmission completed
Starting data transmission
Writing data to SIOr (SDRmn [7:0])
Transmission start by writing
Transfer end interrupt generated?
No
Wait for transmission complete.
(Clear the interrupt request flag)
Yes
Responded ACK?
No
ACK acknowledgment from the slave
If ACK (PEF = 0), to the next process
if NACK (PEF = 1), to error handling
Communication error processing
Yes
No
Data transfer completed?
Yes
Data transmission completed
Stop condition generation

13.9.3 Data reception

Data reception is an operation to receive data to the target for transfer (slave) after transmission of an address field. After all data are received to the slave, a stop condition is generated and the bus is released.

Simplified I^{2}C	IIC00	IIC01	IIC10	IIC11
Target channel	Channel 0 of SAU0	Channel 1 of SAU0	Channel 2 of SAU0	Channel 3 of SAU0
Pins used	SCL00, SDA00 Note 1	SCL01, SDA01 Note 1	SCL10, SDA10 Note 1	SCL11, SDA11 Note 1
Interrupt	INTIIC00	INTIIC01	INTIIC10	INTIIC11
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error detection flag	Overrun error detection flag (OVFmn) only			
Transfer data length	8 bits			
Transfer rate 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 (ACK transmission)			
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, 3.0 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 35**, **CHAPTER 36 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), mn = 00 to 03

(1) Register setting

Figure 13 - 135 Example of Contents of Registers for Data Reception of Simplified I2C (IIC00, IIC01, 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	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting Note 3							0	Dummy transmit data setting (FFH)							
									SIOr							

(d) Serial output register m (SOm)... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	CKOm3 0/1 Note 4	CKOm2 0/1 Note 4	CKOm1 0/1 Note 4	CKOm0 0/1 Note 4	0	0	0	0	SOm3 0/1 Note 4	SOm2 0/1 Note 4	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	SOEm3 0/1	SOEm2 0/1	SOEm1 0/1	SOEm0 0/1

(f) Serial channel start register m (SSm)... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 0/1	SSm1 0/1	SSm0 0/1

Note 1. Only provided for the SMR01 and SMR03 registers

Note 2. Only provided for the SCR00 and SCR02 registers

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 to 3), r: IIC number (r = 00, 01, 10, 11), mn = 00 to 03

Remark 2. ▣: Setting is fixed in the IIC mode,

■: Setting disabled (set to the initial value)

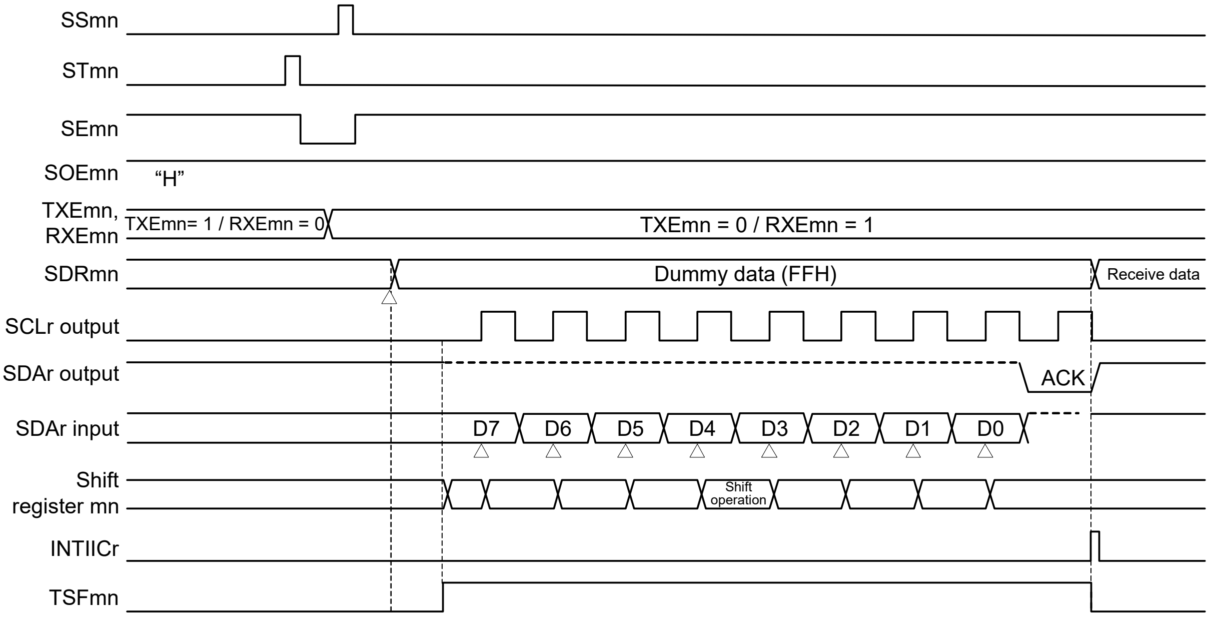
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

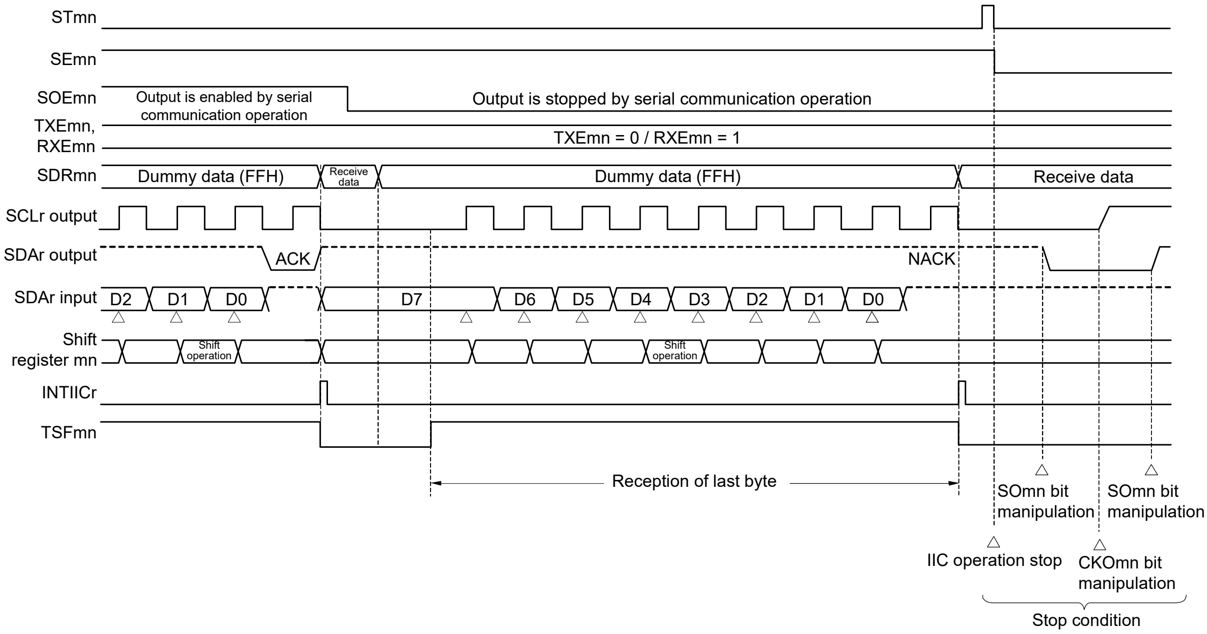
(2) Processing flow

Figure 13 - 136 Timing Chart of Data Reception

(a) When starting data reception

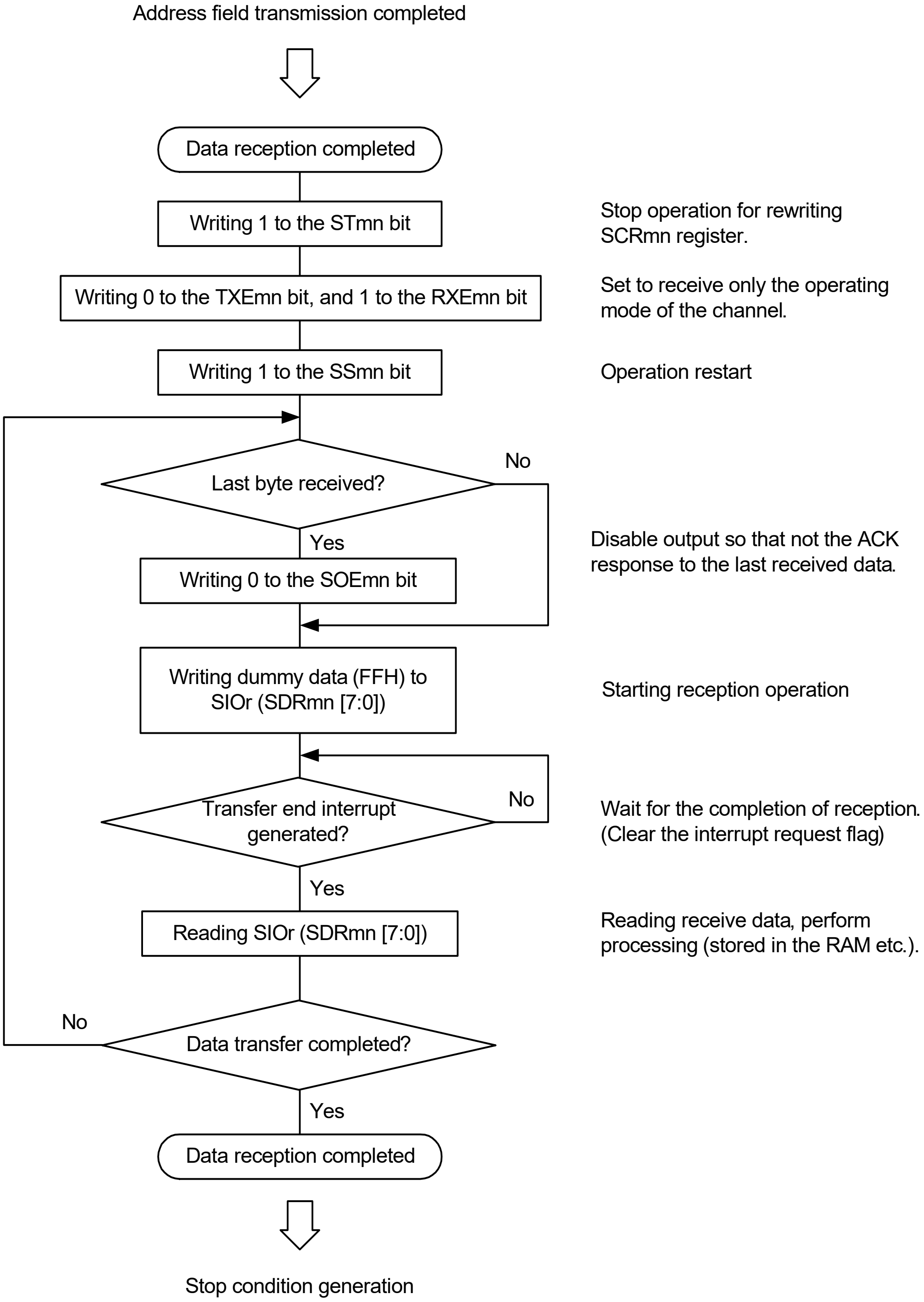


(b) When receiving last data



Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), r: IIC number (r = 00, 01, 10, 11), mn = 00 to 03

Figure 13 - 137 Flowchart of Data Reception


Address field transmission completed
Data reception completed
Writing 1 to the STmn bit
Stop operation for rewriting SCRmn register.
Writing 0 to the TXEmn bit, and 1 to the RXEmn bit
Set to receive only the operating mode of the channel.
Writing 1 to the SSmn bit
Operation restart
Last byte received?
No
Yes
Writing 0 to the SOEmn bit
Disable output so that not the ACK response to the last received data.
Writing dummy data (FFH) to SIOr (SDRmn [7:0])
Starting reception operation
Transfer end interrupt generated?
No
Yes
Wait for the completion of reception. (Clear the interrupt request flag)
Reading SIOr (SDRmn [7:0])
Reading receive data, perform processing (stored in the RAM etc.).
No
Data transfer completed?
Yes
Data reception completed
Stop condition generation

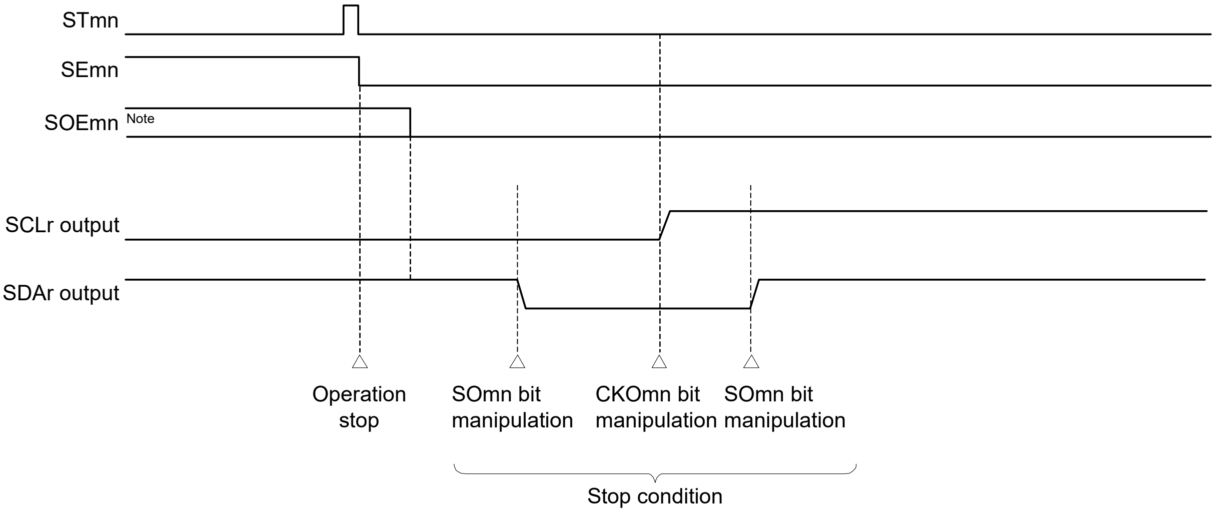
Caution **ACK is not output when the last data is received (NACK). Communication is then completed by setting 1 to the STmn bit of serial channel stop register m (STm) to stop operation and generating a stop condition.**

13.9.4 Stop condition generation

After all data are transmitted to or received from the target slave, a stop condition is generated and the bus is released.

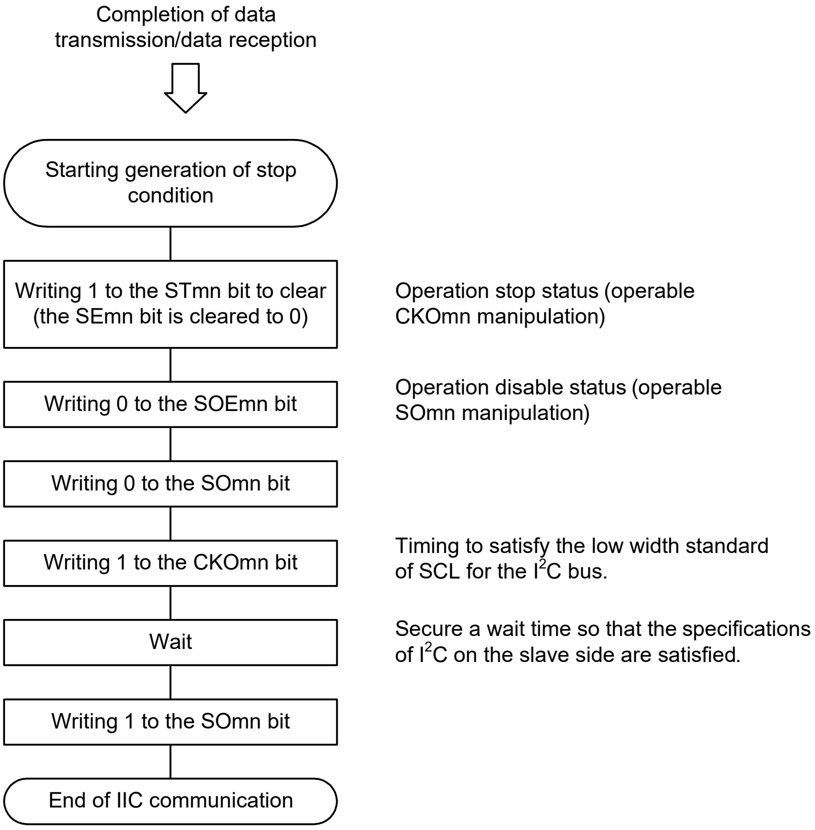
(1) Processing flow

Figure 13 - 138 Timing Chart of Stop Condition Generation



Note During a receive operation, the SOEmn bit of serial output enable register m (SOEm) is cleared to 0 before receiving the last data.

Figure 13 - 139 Flowchart of Stop Condition Generation


Completion of data transmission/data reception
Starting generation of stop condition
Writing 1 to the STmn bit to clear (the SEmn bit is cleared to 0)
Operation stop status (operable CKOmn manipulation)
Writing 0 to the SOEmn bit
Operation disable status (operable SOmn manipulation)
Writing 0 to the SOmn bit
Writing 1 to the CKOmn bit
Timing to satisfy the low width standard of SCL for the I2C bus.
Wait
Secure a wait time so that the specifications of I2C on the slave side are satisfied.
Writing 1 to the SOmn bit
End of IIC communication

13.9.5 Calculating transfer rate

The transfer rate for simplified I^{2}C (IIC00, IIC01, 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^{2}C is 50%. The I^{2}C 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^{2}C bus specifications. Make sure that the SDRmn[15:9] value satisfies the I^{2}C 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 to 3), mn = 00 to 03

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 13 - 6 Selection of Operation Clock For Simplified I²C

SMRmn Register	SPSm Register								Operation Clock (f_{MCK}) Note	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 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 to 3), mn = 00 to 03

Here is an example of setting an I²C transfer rate where f_{MCK} = f_{CLK} = 24 MHz.

I²C 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%.

13.9.6 Procedure for processing errors that occurred during simplified I^{2}C (IIC00, IIC01, IIC10, IIC11) communication

The procedure for processing errors that occurred during simplified I^{2}C (IIC00, IIC01, IIC10, IIC11) communication is described in **Figures 13 - 140** and **13 - 141**.

Figure 13 - 140 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 13 - 141 Processing Procedure in Case of ACK error in Simplified I^{2}C 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 to 3), r: IIC number (r = 00, 01, 10, 11), mn = 00 to 03

CHAPTER 14 SERIAL INTERFACE IICA

The number of serial interfaces IICA differs depending on the product.

Channel	10-pin	16-pin	20, 24, 25-pin
IICA0	—	√	√
IICA1	—	—	√

14.1 Functions of Serial Interface IICA

Serial interface IICA has the following three modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

(2) I^2C bus mode (multimaster supported)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCLAn) line and a serial data bus (SDAAn) line.

This mode complies with the I^2C bus format and the master device can generated “start condition”, “address”, “transfer direction specification”, “data”, and “stop condition” data to the slave device, via the serial data bus. The slave device automatically detects these received status and data by hardware. This function can simplify the part of application program that controls the I^2C bus.

Since the SCLAn and SDAAn pins are used for open drain outputs, serial interface IICA requires pull-up resistors for the serial clock line and the serial data bus line.

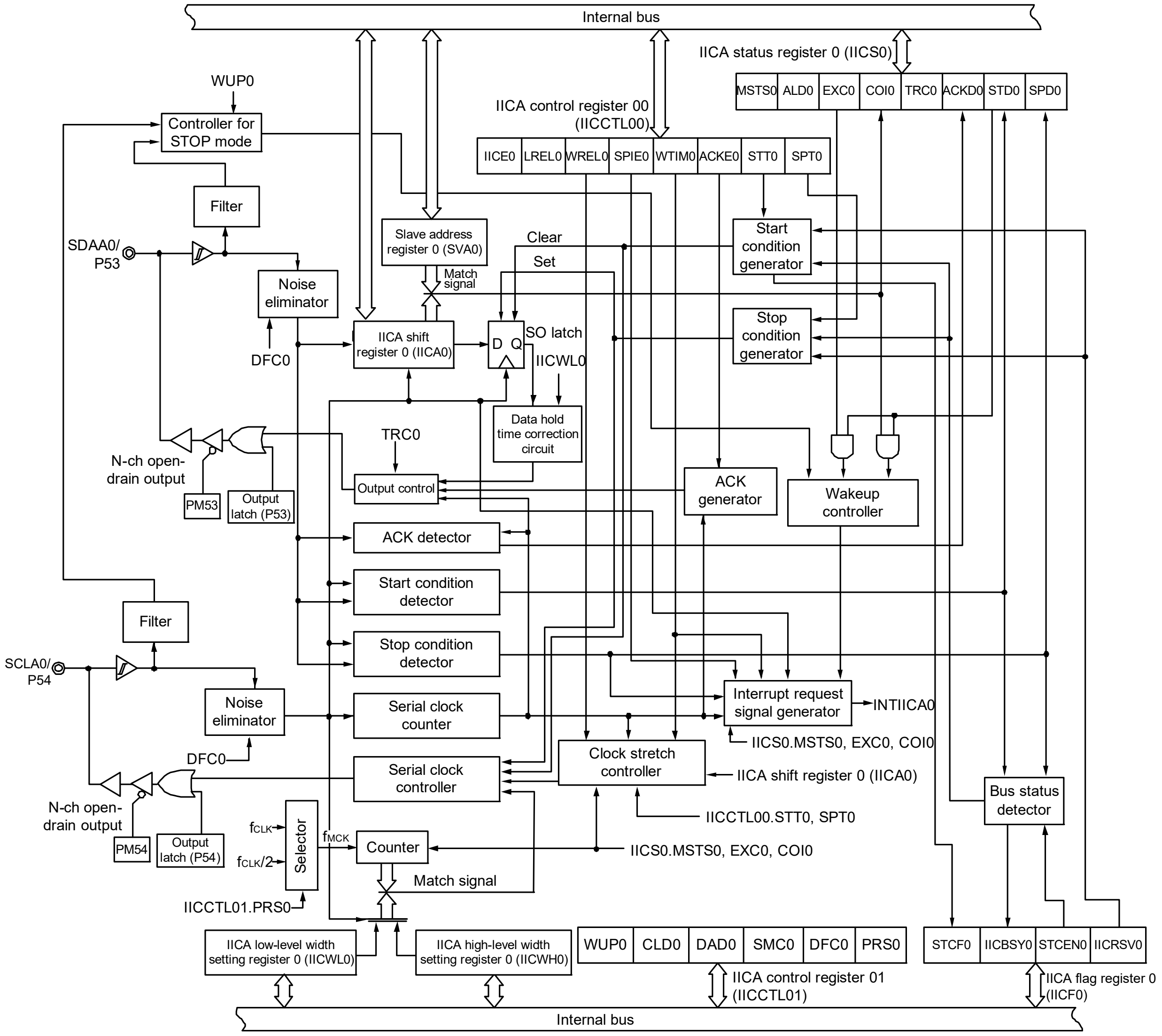
(3) Wakeup mode

The STOP mode can be released by generating an interrupt request signal (INTIICAn) when an extension code from the master device or a local address has been received while in STOP mode. This can be set by using the WUPn bit of IICA control register n1 (IICCTLn1).

Figure 14 - 1 shows a block diagram of serial interface IICA

Remark n = 0, 1

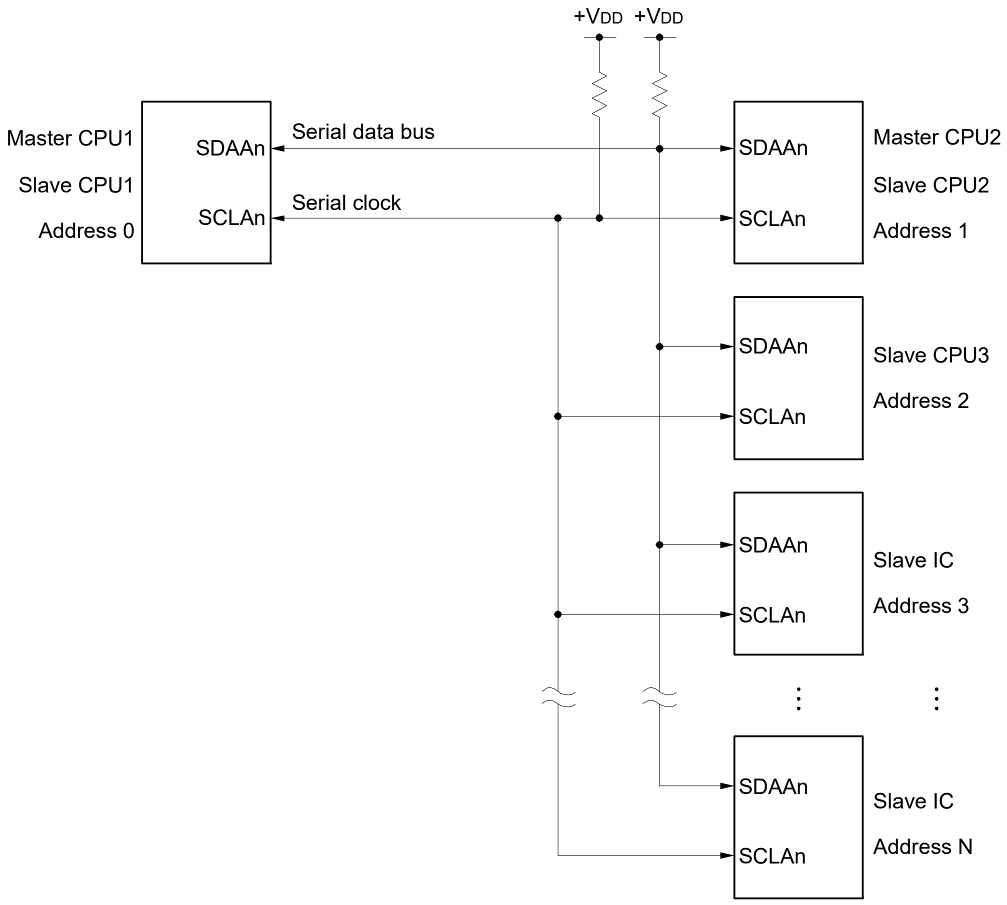
Figure 14 - 1 Block Diagram of Serial Interface IICA



Remark The IICA pins shown in the above diagram are those when PIOR31 is 0 and PIOR30 is 0 in the 24 and 25-pin products.

Figure 14 - 2 shows a serial bus configuration example.

Figure 14 - 2 Serial Bus Configuration Example Using I^{2}C Bus



Remark n = 0, 1

14.2 Configuration of Serial Interface IICA

Serial interface IICA includes the following hardware.

Table 14 - 1 Configuration of Serial Interface IICA

Item	Configuration
Registers	IICA shift register n (IICAn) Slave address register n (SVAn)
Control registers	Peripheral enable register 0 (PER0) Peripheral reset control register 0 (PRR0) IICA control register n0 (IICCTLn0) IICA status register n (IICSn) IICA flag register n (IICFn) IICA control register n1 (IICCTLn1) IICA low-level width setting register n (IICWLn) IICA high-level width setting register n (IICWHn) Port mode registers 0, 3 to 5 (PM0, PM3 to PM5) Port registers 0, 3 to 5 (P0, P3 to P5) Port output mode registers 0, 3 to 5 (POM0, POM3 to POM5) Port mode control registers 0, 3, 5 (PMC0, PMC3, PMC5)

Remark n = 0, 1

(1) IICA shift register n (IICAn)

The IICAn register is used to convert 8-bit serial data to 8-bit parallel data and vice versa in synchronization with the serial clock. The IICAn register can be used for both transmission and reception.

The actual transmit and receive operations can be controlled by writing and reading operations to the IICAn register.

Cancel the clock stretch state and start data transfer by writing data to the IICAn register during the clock stretch period.

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

Reset signal generation clears IICAn to 00H.

Figure 14 - 3 Format of IICA shift register n (IICAn)

Address: FFF50H (IICA0), FFF54H (IICA1) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IICAn								

Caution 1. Do not write data to the IICAn register during data transfer.

Caution 2. Write or read the IICAn register only during the clock stretch period. Accessing the IICAn register in a communication state other than during the clock stretch period is prohibited. When the device serves as the master, however, the IICAn register can be written only once after the communication trigger bit (STTn) is set to 1.

Caution 3. When communication is reserved, write data to the IICAn register after the interrupt triggered by a stop condition is detected.

Remark n = 0, 1

(2) Slave address register n (SVAn)

This register stores seven bits of local addresses {A6, A5, A4, A3, A2, A1, A0} when in slave mode.

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

However, rewriting to this register is prohibited while STDn = 1 (while the start condition is detected).

Reset signal generation clears the SVAn register to 00H.

Figure 14 - 4 Format of Slave address register n (SVAn)

Address: F0234H (SVA0), F023CH (SVA1) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SVAn	A6	A5	A4	A3	A2	A1	A0	0 Note

Note Bit 0 is fixed to 0.

(3) SO latch

The SO latch is used to retain the SDAAn pin's output level.

(4) Wakeup controller

This circuit generates an interrupt request (INTIICAn) when the address received by this register matches the address value set to the slave address register n (SVAn) or when an extension code is received.

(5) Serial clock counter

This counter counts the serial clocks that are output or input during transmit/receive operations and is used to verify that 8-bit data was transmitted or received.

(6) Interrupt request signal generator

This circuit controls the generation of interrupt request signals (INTIICAn).

An I^{2}C interrupt request is generated by the following two triggers.

- Falling edge of eighth or ninth clock of the serial clock (set by the WTIMn bit)
- Interrupt request generated when a stop condition is detected (set by the SPIEn bit)

Remark WTIMn bit: Bit 3 of IICA control register n0 (IICCTLn0)
SPIEn bit: Bit 4 of IICA control register n0 (IICCTLn0)

(7) Serial clock controller

In master mode, this circuit generates the clock output via the SCLAn pin from a sampling clock.

(8) Clock stretch controller

This circuit controls the clock stretch timing.

Remark n = 0, 1

(9) ACK generator, stop condition detector, start condition detector, and ACK detector

These circuits generate and detect each status.

(10) Data hold time correction circuit

This circuit generates the hold time for data corresponding to the falling edge of the serial clock.

(11) Start condition generator

This circuit generates a start condition when the STTn bit is set to 1.

However, in the communication reservation disabled status (IICRSVn bit = 1), when the bus is not released (IICBSYn bit = 1), start condition requests are ignored and the STCFn bit is set to 1.

(12) Stop condition generator

This circuit generates a stop condition when the SPTn bit is set to 1.

(13) Bus status detector

This circuit detects whether or not the bus is released by detecting start conditions and stop conditions.

However, as the bus status cannot be detected immediately following operation, the initial status is set by the STCENn bit.

Remark 1. STTn bit: Bit 1 of IICA control register n0 (IICCTLn0)
SPTn bit: Bit 0 of IICA control register n0 (IICCTLn0)
IICRSVn bit: Bit 0 of IICA flag register n (IICFn)
IICBSYn bit: Bit 6 of IICA flag register n (IICFn)
STCFn bit: Bit 7 of IICA flag register n (IICFn)
STCENn bit: Bit 1 of IICA flag register n (IICFn)

Remark 2. n = 0, 1

14.3 Registers Controlling Serial Interface IICA

Serial interface IICA is controlled by the following nine registers.

- Peripheral enable register 0 (PER0)
- Peripheral reset control register 0 (PRR0)
- IICA control register n0 (IICCTLn0)
- IICA flag register n (IICFn)
- IICA status register n (IICSn)
- IICA control register n1 (IICCTLn1)
- IICA low-level width setting register n (IICWLn)
- IICA high-level width setting register n (IICWHn)
- Port mode registers 0, 3 to 5 (PM0, PM3 to PM5)
- Port registers 0, 3 to 5 (P0, P3 to P5)
- Port output mode registers 0, 3 to 5 (POM0, POM3 to POM5)
- Port mode control registers 0, 3, 5 (PMC0, PMC3, PMC5)

Remark n = 0, 1

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 serial interface IICAn is used, be sure to set bits 6, 4 (IICA1EN, IICA0EN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 14 - 5 Format of Peripheral enable register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	3	<2>	1	<0>
PER0	0	IICA1EN	ADCEN	IICA0EN	0	SAU0EN	0	TAU0EN

IICAnEN	Control of serial interface IICAn input clock supply
0	Stops input clock supply. • SFR used by serial interface IICAn cannot be written. • Serial interface IICAn is in the reset status.
1	Enables input clock supply. • SFR used by serial interface IICAn can be read/written.

Caution 1. When setting serial interface IICA, be sure to set the following registers first while the IICAnEN bit is set to 1. If IICAnEN = 0, the control registers of serial interface IICA are set to their initial values, and writing to them is ignored (except for port mode registers 0, 3 to 5 (PM0, PM3 to PM5), port registers 0, 3 to 5 (P0, P3 to P5), port output mode registers 0, 3 to 5 (POM0, POM3 to POM5), and port mode control registers 0, 3, 5 (PMC0, PMC3, PMC5)).

- **IICA control register n0 (IICCTLn0)**
- **IICA flag register n (IICFn)**
- **IICA status register n (IICSn)**
- **IICA control register n1 (IICCTLn1)**
- **IICA low-level width setting register n (IICWLn)**
- **IICA high-level width setting register n (IICWHn)**

Caution 2. Be sure to clear the bits 7, 3, 1 to 0.

Remark n = 0, 1

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 IICA1 and IICA0, be sure to set bits 6 and 4 (IICA1RES and IICA0RES) 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 - 6 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	IICA1RES	ADCRES	IICA0RES	0	SAU0RES	0	TAU0RES

IICA1RES	Reset control of IICA1
0	IICA1 is released from the reset state.
1	IICA1 is in the reset state.

IICA0RES	Reset control of IICA0
0	IICA0 is released from the reset state.
1	IICA0 is in the reset state.

14.3.3 IICA control register n0 (IICCTLn0)

This register is used to enable/stop I^{2}C operations, set clock stretch timing, and set other I^{2}C operations.

The IICCTLn0 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, set the SPIEn, WTIMn, and ACKEn bits while IICEn = 0 or during the clock stretch period. These bits can be set at the same time when the IICEn bit is set from “0” to “1”.

Reset signal generation clears this register to 00H.

Remark n = 0, 1

Figure 14 - 7 Format of IICA control register n0 (IICCTLn0) (1/4)

Address: F0230H (IICCTL00), F0238H (IICCTL10) After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICCTLn0	IICEn	LRELn	WRELn	SPIEn	WTIMn	ACKEn	STTn	SPTn

IICEn	I^{2}C operation enable
0	Stop operation. Reset the IICA status register n (IICSn) [Note 1]. Stop internal operation.
1	Enable operation.
Be sure to set this bit (1) while the SCLAn and SDAAn lines are at high level.	
Condition for clearing (IICEn = 0)	Condition for setting (IICEn = 1)
• Cleared by instruction • Reset	• Set by instruction

LRELn Notes 2, 3	Exit from communications
0	Normal operation
1	This exits from the current communications and sets standby mode. This setting is automatically cleared to 0 after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCLAn and SDAAn lines are set to high impedance. The following flags of IICA control register n0 (IICCTLn0) and the IICA status register n (IICSn) are cleared to 0. • STTn • SPTn • MSTSn • EXCn • COIn • TRCn • ACKDn • STDn
The standby mode following exit from communications remains in effect until the following communications entry conditions are met. • After a stop condition is detected, restart is in master mode. • An address match or extension code reception occurs after the start condition.	
Condition for clearing (LRELn = 0)	Condition for setting (LRELn = 1)
• Automatically cleared after execution • Reset	• Set by instruction

WRELn Notes 2, 3	Clock stretch cancellation
0	Do not cancel clock stretch
1	Cancel clock stretch. This setting is automatically cleared after clock stretch is canceled.
When the WRELn bit is set (clock stretch canceled) during the clock stretch period at the ninth clock pulse in the transmission status (TRCn = 1), the SDAAn line goes into the high impedance state (TRCn = 0).	
Condition for clearing (WRELn = 0)	Condition for setting (WRELn = 1)
• Automatically cleared after execution • Reset	• Set by instruction

Note 1. The IICA shift register n (IICAn), the STCFn and IICBSYn bits of the IICA flag register n (IICFn), and the CLDn and DADn bits of IICA control register n1 (IICCTLn1) are reset.

Note 2. The signal of this bit is invalid while IICEn is 0.

Note 3. When the LRELn and WRELn bits are read, 0 is always read.

Caution **If the operation of I^{2}C is enabled (IICEn = 1) when the SCLAn line is high level, the SDAAn line is low level, and the digital filter is turned on (DFCn bit of IICCTLn1 register = 1), a start condition will be inadvertently detected immediately. In this case, set (1) the LRELn bit by using a 1-bit memory manipulation instruction immediately after enabling operation of I^{2}C (IICEn = 1).**

Remark n = 0, 1

Figure 14 - 7 Format of IICA control register n0 (IICCTLn0) (2/4)

SPIEn Note 1	Enable/disable generation of interrupt request when stop condition is detected
0	Disable
1	Enable
If the WUPn bit of IICA control register n1 (IICCTLn1) is 1, no stop condition interrupt will be generated even if SPIEn = 1.	
Condition for clearing (SPIEn = 0)	Condition for setting (SPIEn = 1)
• Cleared by instruction • Reset	• Set by instruction

WTIMn Note 1	Control of clock stretch and interrupt request generation
0	Interrupt request is generated at the eighth clock's falling edge. Master mode: After output of eight clocks, clock output is set to low level and clock stretch is set. Slave mode: After input of eight clocks, the clock is set to low level and clock stretch is set for master device.
1	Interrupt request is generated at the ninth clock's falling edge. Master mode: After output of nine clocks, clock output is set to low level and clock stretch is set. Slave mode: After input of nine clocks, the clock is set to low level and clock stretch is set for master device.
An interrupt is generated at the falling edge of the ninth clock during address transfer independently of the setting of this bit. The setting of this bit is valid when the address transfer is completed. When in master mode, a clock stretch is inserted at the falling edge of the ninth clock during address transfers. For a slave device that has received a local address, a clock stretch is inserted at the falling edge of the ninth clock after an acknowledge (ACK) is issued. However, when the slave device has received an extension code, a clock stretch is inserted at the falling edge of the eighth clock.	
Condition for clearing (WTIMn = 0)	Condition for setting (WTIMn = 1)
• Cleared by instruction • Reset	• Set by instruction

ACKEn Notes 1, 2	Acknowledgment control
0	Disable acknowledgment.
1	Enable acknowledgment. During the ninth clock period, the SDAAn line is set to low level.
Condition for clearing (ACKEn = 0)	Condition for setting (ACKEn = 1)
• Cleared by instruction • Reset	• Set by instruction

Note 1. The signal of this bit is invalid while IICEn is 0. Set this bit during that period.

Note 2. The set value is invalid during address transfer and if the code is not an extension code.
When the device serves as a slave and the addresses match, an acknowledgment is generated regardless of the set value.

Remark n = 0, 1

Figure 14 - 7 Format of IICA control register n0 (IICCTLn0) (3/4)

STTn Notes 1, 2	Start condition trigger
0	Do not generate a start condition.
1	When bus is released (in standby state, when IICBSYn = 0): If this bit is set (1), a start condition is generated (startup as the master). When a third party is communicating: • When communication reservation function is enabled (IICRSVn = 0) Functions as the start condition reservation flag. When set to 1, automatically generates a start condition after the bus is released. • When communication reservation function is disabled (IICRSVn = 1) Even if this bit is set (1), the STTn bit is cleared and the STTn clear flag (STCFn) is set (1). No start condition is generated. In the clock stretch state (when master device): Generates a restart condition after releasing the clock stretch.

Cautions concerning set timing

- For master reception: Cannot be set to 1 during transfer. Can be set to 1 only in the clock stretch period when the ACKEn bit has been cleared to 0 and slave has been notified of final reception.
- For master transmission: A start condition cannot be generated normally during the acknowledge period. Set to 1 during the clock stretch period that follows output of the ninth clock.
- Cannot be set to 1 at the same time as stop condition trigger (SPTn).
- Once STTn is set (1), setting it again (1) before the clear condition is met is not allowed.

Condition for clearing (STTn = 0)	Condition for setting (STTn = 1)
• Cleared by setting the STTn bit to 1 while communication reservation is prohibited. • Cleared by loss in arbitration • Cleared after start condition is generated by master device • Cleared by LRELn = 1 (exit from communications) • When IICEn = 0 (operation stop) • Reset	• Set by instruction

Note 1. The signal of this bit is invalid while IICEn is 0.
Note 2. The STTn bit is always read as 0.

Remark 1. Bit 1 (STTn) becomes 0 when it is read after data setting.
Remark 2. IICRSVn: Bit 0 of IICA flag register n (IICFn)
STCFn: Bit 7 of IICA flag register n (IICFn)
Remark 3. n = 0, 1

Figure 14 - 7 Format of IICA control register n0 (IICCTLn0) (4/4)

<table>
<tr><td>SPTn Note</td><td colspan="2">Stop condition trigger</td></tr>
<tr><td>0</td><td colspan="2">Stop condition is not generated.</td></tr>
<tr><td>1</td><td colspan="2">Stop condition is generated (termination of master device’s transfer).</td></tr>
<tr><td colspan="3">Cautions concerning set timing
• For master reception: Cannot be set to 1 during transfer. Can be set to 1 only in the clock stretch period when the ACKEn bit has been cleared to 0 and slave has been notified of final reception.
• For master transmission: A stop condition cannot be generated normally during the acknowledge period. Therefore, set it during the clock stretch period that follows output of the ninth clock.
• Cannot be set to 1 at the same time as start condition trigger (STTn).
• The SPTn bit can be set to 1 only when in master mode.
• When the WTIMn bit has been cleared to 0, if the SPTn bit is set to 1 during the clock stretch period that follows output of eight clocks, note that a stop condition will be generated during the high-level period of the ninth clock. The WTIMn bit should be changed from 0 to 1 during the clock stretch period following the output of eight clocks, and the SPTn bit should be set to 1 during the clock stretch period that follows the output of the ninth clock.
• Once SPTn is set (1), setting it again (1) before the clear condition is met is not allowed.</td></tr>
<tr><td colspan="2">Condition for clearing (SPTn = 0)</td><td>Condition for setting (SPTn = 1)</td></tr>
<tr><td colspan="2">• Cleared by loss in arbitration
• Automatically cleared after stop condition is detected
• Cleared by LRELn = 1 (exit from communications)
• When IICEn = 0 (operation stop)
• Reset</td><td>• Set by instruction</td></tr>
</table>

Note When the SPTn register is read, 0 is always read.

Caution **When bit 3 (TRCn) of the IICA status register n (IICSn) is set to 1 (transmission status), bit 5 (WRELn) of IICA control register n0 (IICCTLn0) is set to 1 during the ninth clock and clock stretch is canceled, after which the TRCn bit is cleared (reception status) and the SDAAn line is set to high impedance. Release the clock stretch performed while the TRCn bit is 1 (transmission status) by writing to the IICA shift register n.**

Remark n = 0, 1

14.3.4 IICA status register n (IICSn)

This register indicates the status of I^{2}C.

The IICSn register is read by a 1-bit or 8-bit memory manipulation instruction only when STTn = 1 and during the clock stretch period.

Reset signal generation clears this register to 00H.

Caution Reading the IICSn register while the address match wakeup function is enabled (WUPn = 1) in STOP mode is prohibited. When the WUPn bit is changed from 1 to 0 (wakeup operation is stopped), regardless of the INTIICAn interrupt request, the change in status is not reflected until the next start condition or stop condition is detected. To use the wakeup function, therefore, enable (SPIEn = 1) the interrupt generated by detecting a stop condition and read the IICSn register after the interrupt has been detected.

Remark STTn: bit 1 of IICA control register n0 (IICCTLn0)
WUPn: bit 7 of IICA control register n1 (IICCTLn1)

Figure 14 - 8 Format of IICA status register n (IICSn) (1/3)

Address: FFF51H (IICS0), FFF55H (IICS1) After reset: 00H R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICSn	MSTSn	ALDn	EXCn	COIn	TRCn	ACKDn	STDn	SPDn

MSTSn	Master status check flag
0	Slave device status or communication standby status
1	Master device communication status

Condition for clearing (MSTSn = 0)	Condition for setting (MSTSn = 1)
• When a stop condition is detected • When ALDn = 1 (arbitration loss) • Cleared by LRELn = 1 (exit from communications) • When the IICEn bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is generated

ALDn	Detection of arbitration loss
0	This status means either that there was no arbitration or that the arbitration result was a “win”.
1	This status indicates the arbitration result was a “loss”. The MSTSn bit is cleared.

Condition for clearing (ALDn = 0)	Condition for setting (ALDn = 1)
• Automatically cleared after the IICSn register is read Note • When the IICEn bit changes from 1 to 0 (operation stop) • Reset	• When the arbitration result is a “loss”.

Note This register is also cleared when a 1-bit memory manipulation instruction is executed for bits other than the IICSn register. Therefore, when using the ALDn bit, read the data of this bit before the data of the other bits.

Remark 1. LRELn: Bit 6 of IICA control register n0 (IICCTLn0)
IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

Remark 2. n = 0, 1

Figure 14 - 8 Format of IICA status register n (IICSn) (2/3)

EXCn	Detection of extension code reception	
0	Extension code was not received.	
1	Extension code was received.	
Condition for clearing (EXCn = 0)		Condition for setting (EXCn = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LRELn = 1 (exit from communications) • When the IICEn bit changes from 1 to 0 (operation stop) • Reset		• When the higher four bits of the received address data is either “0000” or “1111” (set at the rising edge of the eighth clock).

COIn	Detection of matching addresses	
0	Addresses do not match.	
1	Addresses match.	
Condition for clearing (COIn = 0)		Condition for setting (COIn = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LRELn = 1 (exit from communications) • When the IICEn bit changes from 1 to 0 (operation stop) • Reset		• When the received address matches the local address (slave address register n (SVAn)) (set at the rising edge of the eighth clock).

TRCn	Detection of transmit/receive status	
0	Receive status (other than transmit status). The SDAAn line is set for high impedance.	
1	Transmit status. The value in the SOn latch is enabled for output to the SDAAn line (valid starting at the falling edge of the first byte’s ninth clock).	
Condition for clearing (TRCn = 0)		Condition for setting (TRCn = 1)
<Both master and slave> • When a stop condition is detected • Cleared by LRELn = 1 (exit from communications) • When the IICEn bit changes from 1 to 0 (operation stop) • Cleared by WRELn = 1 Note (clock stretch cancel) • When the ALDn bit changes from 0 to 1 (arbitration loss) • Reset • When not used for communication (MSTSn, EXCn, COIn = 0) <Master> • When “1” is output to the first byte’s LSB (transfer direction specification bit) <Slave> • When a start condition is detected • When “0” is input to the first byte’s LSB (transfer direction specification bit)		<Master> • When a start condition is generated • When 0 (master transmission) is output to the LSB (transfer direction specification bit) of the first byte (during address transfer) <Slave> • When 1 (slave transmission) is input to the LSB (transfer direction specification bit) of the first byte from the master (during address transfer)

Note When bit 3 (TRCn) of the IICA status register n (IICSn) is set to 1 (transmission status), bit 5 (WRELn) of IICA control register n0 (IICCTLn0) is set to 1 during the ninth clock and clock stretch is canceled, after which the TRCn bit is cleared (reception status) and the SDAAn line is set to high impedance. Release the clock stretch performed while the TRCn bit is 1 (transmission status) by writing to the IICA shift register n.

Remark 1. LRELn: Bit 6 of IICA control register n0 (IICCTLn0)
IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

Remark 2. n = 0, 1

Figure 14 - 8 Format of IICA status register n (IICSn) (3/3)

ACKDn	Detection of acknowledge (ACK)
0	Acknowledge was not detected.
1	Acknowledge was detected.
Condition for clearing (ACKDn = 0)	Condition for setting (ACKDn = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock • Cleared by LRELn = 1 (exit from communications) • When the IICEn bit changes from 1 to 0 (operation stop) • Reset	• After the SDAAn line is set to low level at the rising edge of SCLAn line's ninth clock

STDn	Detection of start condition
0	Start condition was not detected.
1	Start condition was detected. This indicates that the address transfer period is in effect.
Condition for clearing (STDn = 0)	Condition for setting (STDn = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock following address transfer • Cleared by LRELn = 1 (exit from communications) • When the IICEn bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is detected

SPDn	Detection of stop condition
0	Stop condition was not detected.
1	Stop condition was detected. The master device's communication is terminated and the bus is released.
Condition for clearing (SPDn = 0)	Condition for setting (SPDn = 1)
• At the rising edge of the address transfer byte's first clock following setting of this bit and detection of a start condition • When the WUPn bit changes from 1 to 0 • When the IICEn bit changes from 1 to 0 (operation stop) • Reset	• When a stop condition is detected

Remark 1. LRELn: Bit 6 of IICA control register n0 (IICCTLn0)
IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

Remark 2. n = 0, 1

14.3.5 IICA flag register n (IICFn)

This register sets the operation mode of I^{2}C and indicates the status of the I^{2}C bus.

The IICFn register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the STTn clear flag (STCFn) and I^{2}C bus status flag (IICBSYn) bits are read-only.

The IICRSVn bit can be used to enable/disable the communication reservation function.

The STCENn bit can be used to set the initial value of the IICBSYn bit.

The IICRSVn and STCENn bits can be written only when the operation of I^{2}C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) = 0). When operation is enabled, the IICFn register can be read.

Reset signal generation clears this register to 00H.

Figure 14 - 9 Format of IICA flag register n (IICFn)

Address: FFF52H (IICF0), FFF56H (IICF1) After reset: 00H R/W Note

Symbol	<7>	<6>	5	4	3	2	<1>	<0>
IICFn	STCFn	IICBSYn	0	0	0	0	STCENn	IICRSVn

STCFn	STTn clear flag
0	Generate start condition
1	Start condition generation unsuccessful: clear the STTn flag

Condition for clearing (STCFn = 0)	Condition for setting (STCFn = 1)
• Cleared by STTn = 1 • When IICEn = 0 (operation stop) • Reset	• Generating start condition unsuccessful and the STTn bit cleared to 0 when communication reservation is disabled (IICRSVn = 1).

IICBSYn	I2C bus status flag
0	Bus release status (communication initial status when STCENn = 1)
1	Bus communication status (communication initial status when STCENn = 0)

Condition for clearing (IICBSYn = 0)	Condition for setting (IICBSYn = 1)
• Detection of stop condition • When IICEn = 0 (operation stop) • Reset	• Detection of start condition • Setting of the IICEn bit when STCENn = 0

STCENn	Initial start enable trigger
0	After operation is enabled (IICEn = 1), enable generation of a start condition upon detection of a stop condition.
1	After operation is enabled (IICEn = 1), enable generation of a start condition without detecting a stop condition.

Condition for clearing (STCENn = 0)	Condition for setting (STCENn = 1)
• Cleared by instruction • Detection of start condition • Reset	• Set by instruction

IICRSVn	Communication reservation function disable bit
0	Enable communication reservation
1	Disable communication reservation

Condition for clearing (IICRSVn = 0)	Condition for setting (IICRSVn = 1)
• Cleared by instruction • Reset	• Set by instruction

Note Bits 6 and 7 are read-only.

Caution 1. Write to the STCENn bit only when the operation is stopped (IICEn = 0).

Caution 2. As the bus release status (IICBSYn = 0) is recognized regardless of the actual bus status when STCENn = 1, when generating the first start condition (STTn = 1), it is necessary to verify that no third party communications are in progress in order to prevent such communications from being destroyed.

Caution 3. Write to IICRSVn only when the operation is stopped (IICEn = 0).

Remark 1. STTn: Bit 1 of IICA control register n0 (IICCTLn0)

Remark 2. IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

Remark 3. n = 0, 1

14.3.6 IICA control register n1 (IICCTLn1)

This register is used to set the operation mode of I^{2}C and detect the statuses of the SCLAn and SDAAn pins.

The IICCTLn1 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the CLDn and DADn bits are read-only.

Set the IICCTLn1 register, except the WUPn bit, while operation of I^{2}C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) is 0).

Reset signal generation clears this register to 00H.

Figure 14 - 10 Format of IICA control register n1 (IICCTLn1) (1/2)

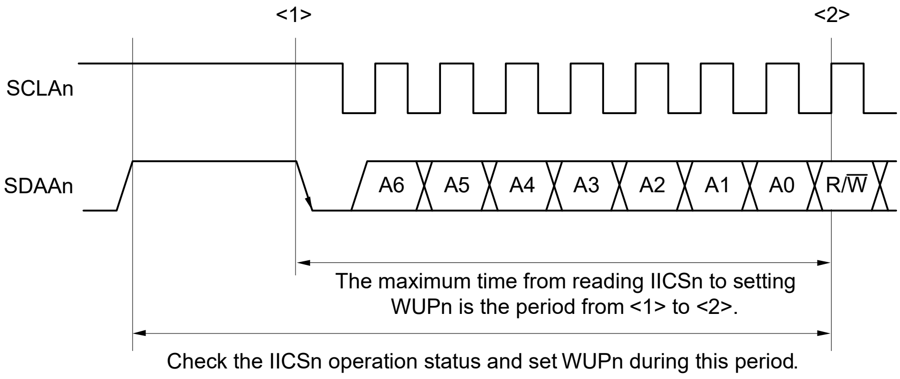
Address: F0231H (IICCTL01), F0239H (IICCTL11) After reset: 00H R/W [Note 1]

Symbol	<7>	6	<5>	<4>	<3>	<2>	1	<0>
IICCTLn1	WUPn	0	CLDn	DADn	SMCn	DFCn	0	PRSn

<table>
<tr><th>WUPn</th><th colspan="2">Control of address match wakeup</th></tr>
<tr><td>0</td><td colspan="2">Stops operation of address match wakeup function in STOP mode.</td></tr>
<tr><td>1</td><td colspan="2">Enables operation of address match wakeup function in STOP mode.</td></tr>
<tr><td colspan="3">To shift to STOP mode when WUPn = 1, execute the STOP instruction at least three f_{MCK} clocks after setting (1) the WUPn bit (see <b>Figure 14 - 22 Flow When Setting WUPn = 1</b>).
Clear (0) the WUPn bit after the address has matched or an extension code has been received. The subsequent communication can be entered by the clearing (0) WUPn bit. (The clock stretch must be released and transmit data must be written after the WUPn bit has been cleared (0).)
The interrupt timing when the address has matched or when an extension code has been received, while WUPn = 1, is identical to the interrupt timing when WUPn = 0. (A delay of the difference of sampling by the clock will occur.)
Furthermore, when WUPn = 1, a stop condition interrupt is not generated even if the SPIEn bit is set to 1.</td></tr>
<tr><td colspan="2">Condition for clearing (WUPn = 0)</td><td>Condition for setting (WUPn = 1)</td></tr>
<tr><td colspan="2">• Cleared by instruction (after address match or extension code reception)</td><td>• Set by instruction (when the MSTSn, EXCn, and COIn bits are “0”, and the STDn bit also “0” (communication not entered)) [Note 2]</td></tr>
</table>

Note 1. Bits 4 and 5 are read-only.

Note 2. The status of the IICA status register n (IICSn) must be checked and the WUPn bit must be set during the period shown below.



The maximum time from reading IICSn to setting WUPn is the period from <1> to <2>.

Check the IICSn operation status and set WUPn during this period.

Remark n = 0, 1

Figure 14 - 10 Format of IICA control register n1 (IICCTLn1) (2/2)

CLDn	Detection of SCLAn pin level (valid only when IICEn = 1)
0	The SCLAn pin was detected at low level.
1	The SCLAn pin was detected at high level.
Condition for clearing (CLDn = 0)	Condition for setting (CLDn = 1)
• When the SCLAn pin is at low level • When IICEn = 0 (operation stop) • Reset	• When the SCLAn pin is at high level

DADn	Detection of SDAAn pin level (valid only when IICEn = 1)
0	The SDAAn pin was detected at low level.
1	The SDAAn pin was detected at high level.
Condition for clearing (DADn = 0)	Condition for setting (DADn = 1)
• When the SDAAn pin is at low level • When IICEn = 0 (operation stop) • Reset	• When the SDAAn pin is at high level

SMCn	Operation mode switching
0	Operates in standard mode (fastest transfer rate: 100 kbps).
1	Operates in fast mode (fastest transfer rate: 400 kbps) or fast mode plus (fastest transfer rate: 1 Mbps).

DFCn	Digital filter operation control
0	Digital filter off.
1	Digital filter on.
Use the digital filter only in fast mode and fast mode plus. The digital filter is used for noise elimination. The transfer clock does not vary, regardless of the DFCn bit being set (1) or cleared (0).	

PRSn	Operation clock (f_{MCK}) control
0	Selects f_{CLK} (1 MHz $\leq f_{CLK} \leq$ 20 MHz)
1	Selects $f_{CLK}/2$ (20 MHz $\leq f_{CLK}$)

Caution 1. The maximum operating frequency of the IICA operating clock (f_{MCK}) is 20 MHz (Max.). Only when f_{CLK} exceeds 20 MHz, set bit 0 (PRSn) of IICA control register n1 (IICCTLn1) to 1.

Caution 2. Note the minimum f_{CLK} operating frequency when setting the transfer clock. The minimum f_{CLK} operating frequency for serial interface IICA is determined according to the mode.

Fast mode:	**f_{CLK} = 3.5 MHz (MIN.)**
Fast mode plus:	**f_{CLK} = 10 MHz (MIN.)**
Normal mode:	**f_{CLK} = 1 MHz (MIN.)**

Caution 3. The fast mode plus is only available in the products for “A: Consumer applications (T_A = -40°C to +85°C)” and “D: Industrial applications (T_A = -40°C to +85°C)”.

Remark 1. IICEn: Bit 7 of IICA control register n0 (IICCTLn0)

Remark 2. n = 0, 1

14.3.7 IICA low-level width setting register n (IICWLn)

This register is used to set the low-level width (t_{LOW}) of the SCLAn pin signal that is output by serial interface IICA and to control the SDAAn pin signal.

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

Set the IICWLn register while operation of I^2C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) is 0).

Reset signal generation sets this register to FFH.

For details about setting the IICWLn register, see **14.4.2 Setting transfer clock by using IICWLn and IICWHn registers**.

The data hold time is one-quarter of the time set by the IICWLn register.

Figure 14 - 11 Format of IICA low-level width setting register n (IICWLn)

Address: F0232H (IICWL0), F023AH (IICWL1) After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWLn								

14.3.8 IICA high-level width setting register n (IICWHn)

This register is used to set the high-level width of the SCLAn pin signal that is output by serial interface IICA and to control the SDAAn pin signal.

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

Set the IICWHn register while operation of I^2C is disabled (bit 7 (IICEn) of IICA control register n0 (IICCTLn0) is 0).

Reset signal generation sets this register to FFH.

Figure 14 - 12 Format of IICA high-level width setting register n (IICWHn)

Address: F0233H (IICWH0), F023BH (IICWH1) After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWHn								

Remark 1. For setting procedures of the transfer clock on master side and of the IICWLn and IICWHn registers on slave side, see **14.4.2 (1)** and **14.4.2 (2)**, respectively.

Remark 2. n = 0, 1

14.3.9 Registers that control port functions of IICA input/output pins

Using the serial interface IICA requires setting of the registers that control the port functions for the port pins with which the IICA pin functions for the target channel are multiplexed (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**.

Using a port pin which is multiplexed with an IICA pin function (e.g. P30/SDAA0) for IICA requires setting the corresponding bits in the port mode control register (PMCxx), port mode register (PMxx), and port register (Pxx) to 0.

Example: When P30/SDAA0 is to be used for IICA
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.

14.4 I2C Bus Mode Functions

14.4.1 Pin configuration

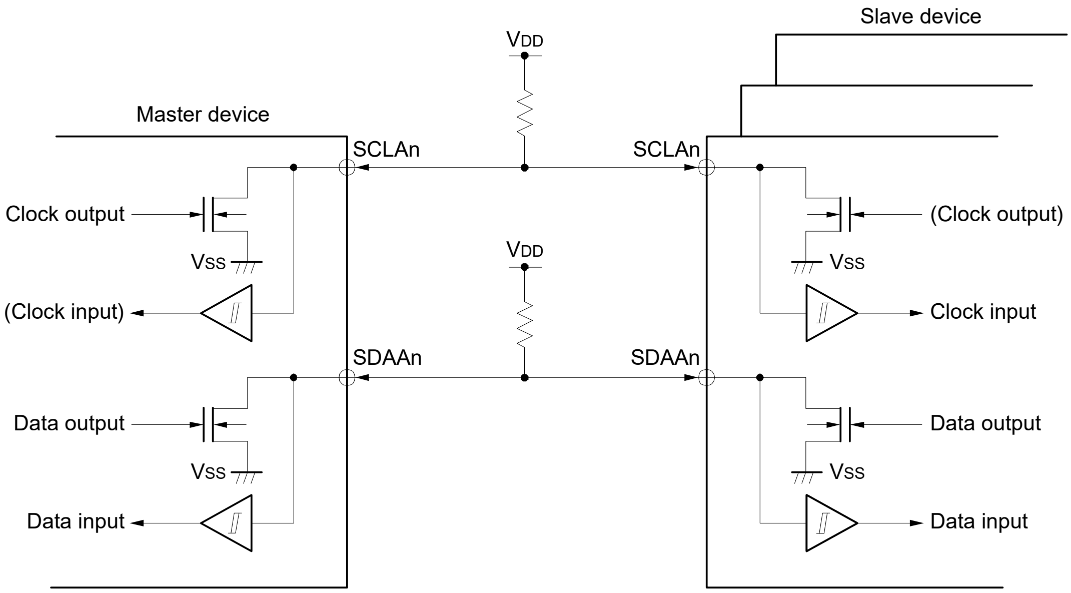
The serial clock pin (SCLAn) and the serial data bus pin (SDAAn) are configured as follows.

(1) SCLAn...... This pin is used for serial clock input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

(2) SDAAn ..... This pin is used for serial data input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open-drain outputs, an external pull-up resistor is required.

Figure 14 - 13 Pin Configuration Diagram



Remark n = 0, 1

14.4.2 Setting transfer clock by using IICWLn and IICWHn registers

(1) Setting transfer clock on master side

$$\boxed{\text{Transfer clock} = \frac{f_{MCK}}{\text{IICWL} + \text{IICWH} + f_{MCK}\ (t_R + t_F)}}$$

At this time, the optimal setting values of the IICWLn and IICWHn registers are as follows.
(The fractional parts of all setting values are rounded up.)

- When the fast mode

$$\text{IICWLn} = \frac{0.52}{\text{Transfer clock}} \times f_{MCK}$$

$$\text{IICWHn} = \left(\frac{0.48}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

- When the standard mode

$$\text{IICWLn} = \frac{0.47}{\text{Transfer clock}} \times f_{MCK}$$

$$\text{IICWHn} = \left(\frac{0.53}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

- When the fast mode plus

$$\text{IICWLn} = \frac{0.50}{\text{Transfer clock}} \times f_{MCK}$$

$$\text{IICWHn} = \left(\frac{0.50}{\text{Transfer clock}} - t_R - t_F\right) \times f_{MCK}$$

(2) Setting IICWLn and IICWHn registers on slave side
(The fractional parts of all setting values are truncated.)

- When the fast mode

$\text{IICWLn} = 1.3\ \mu\text{s} \times f_{MCK}$
$\text{IICWHn} = (1.2\ \mu\text{s} - t_R - t_F) \times f_{MCK}$

- When the standard mode

$\text{IICWLn} = 4.7\ \mu\text{s} \times f_{MCK}$
$\text{IICWHn} = (5.3\ \mu\text{s} - t_R - t_F) \times f_{MCK}$

- When the fast mode plus

$\text{IICWLn} = 0.50\ \mu\text{s} \times f_{MCK}$
$\text{IICWHn} = (0.50\ \mu\text{s} - t_R - t_F) \times f_{MCK}$

Caution 1.The maximum operating frequency of the IICA operating clock (f_{MCK}) is 20 MHz (Max.). Only when f_{CLK} exceeds 20 MHz, set bit 0 (PRSn) of IICA control register n1 (IICCTLn1) to 1.

Caution 2. Note the minimum f_{CLK} operating frequency when setting the transfer clock. The minimum f_{CLK} operating frequency for serial interface IICA is determined according to the mode.

Fast mode:	**f_{CLK} = 3.5 MHz (MIN.)**
Fast mode plus:	**f_{CLK} = 10 MHz (MIN.)**
Normal mode:	**f_{CLK} = 1 MHz (MIN.)**

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

Remark 1. Calculate the rise time (t_R) and fall time (t_F) of the SDAAn and SCLAn signals separately, because they differ depending on the pull-up resistance and wire load.

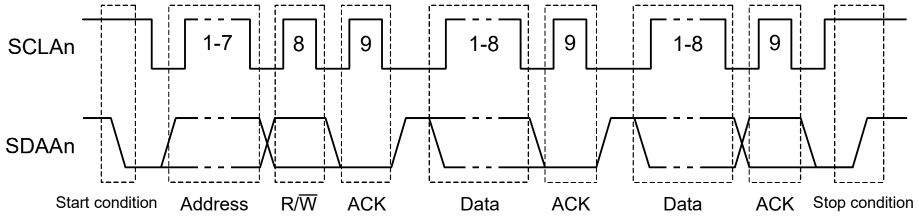
Remark 2. IICWLn: IICA low-level width setting register n
IICWHn: IICA high-level width setting register n
t_F: SDAAn and SCLAn signal falling times
t_R: SDAAn and SCLAn signal rising times
f_{MCK}: IICA operating clock frequency

Remark 3. n = 0, 1

14.5 I2C Bus Definitions and Control Methods

The I2C bus's serial data communication format and the signals used by the I2C bus are described below. Figure 14 - 14 shows the transfer timing for the "start condition", "address", "data", and "stop condition" output via the I2C bus's serial data bus.

Figure 14 - 14 I2C Bus Serial Data Transfer Timing



The master device generates the start condition, slave address, and stop condition.

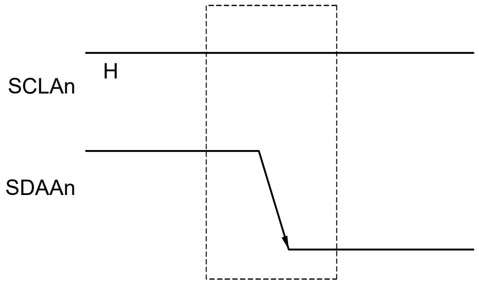
The acknowledge (ACK) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCLAn) is continuously output by the master device. However, in the slave device, the SCLAn pin low level period can be extended and a clock stretch can be inserted.

14.5.1 Start conditions

A start condition is met when the SCLAn pin is at high level and the SDAAn pin changes from high level to low level. The start conditions for the SCLAn pin and SDAAn pin are signals that the master device generates to the slave device when starting a serial transfer. When the device is used as a slave, start conditions can be detected.

Figure 14 - 15 Start Conditions



A start condition is output when bit 1 (STTn) of IICA control register n0 (IICCTLn0) is set (1) after a stop condition has been detected (SPDn: Bit 0 of the IICA status register n (IICSn) = 1). When a start condition is detected, bit 1 (STDn) of the IICSn register is set (1).

Remark n = 0, 1

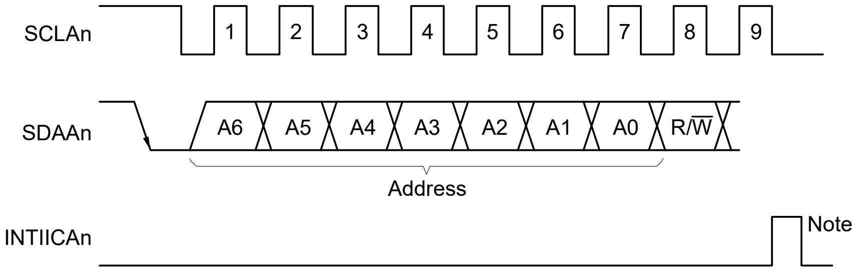
14.5.2 Addresses

The address is defined by the 7 bits of data that follow the start condition.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via the bus lines. Therefore, each slave device connected via the bus lines must have a unique address.

The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data values stored in the slave address register n (SVAn). If the address data matches the SVAn register values, the slave device is selected and communicates with the master device until the master device generates a start condition or stop condition.

Figure 14 - 16 Address



Note INTIICAn is not issued if data other than a local address or extension code is received during slave device operation.

Addresses are output when a total of 8 bits consisting of the slave address and the transfer direction described in **14.5.3 Transfer direction specification** are written to the IICA shift register n (IICAn). The received addresses are written to the IICAn register.

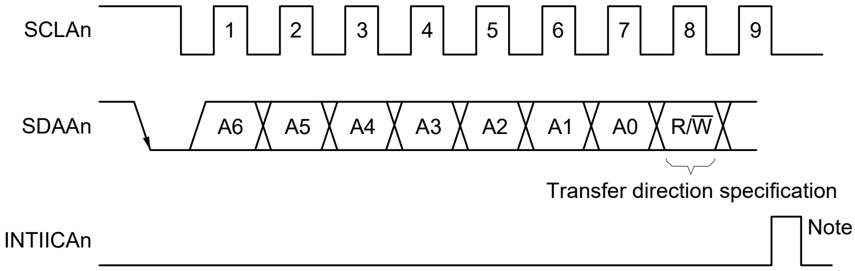
The slave address is assigned to the higher 7 bits of the IICAn register.

14.5.3 Transfer direction specification

In addition to the 7-bit address data, the master device sends 1 bit that specifies the transfer direction.

When this transfer direction specification bit has a value of “0”, it indicates that the master device is transmitting data to a slave device. When the transfer direction specification bit has a value of “1”, it indicates that the master device is receiving data from a slave device.

Figure 14 - 17 Transfer Direction Specification



Note INTIICAn is not issued if data other than a local address or extension code is received during slave device operation.

Remark n = 0, 1

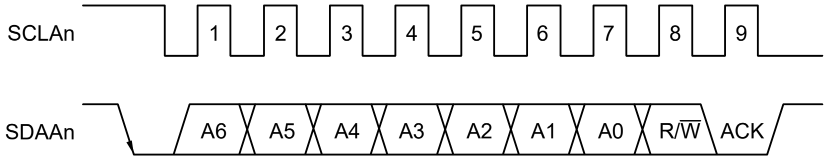
14.5.4 Acknowledge (ACK)

ACK is used to check the status of serial data at the transmission and reception sides.
The reception side returns ACK each time it has received 8-bit data.
The transmission side usually receives ACK after transmitting 8-bit data. When ACK is returned from the reception side, it is assumed that reception has been correctly performed and processing is continued. Whether ACK has been detected can be checked by using bit 2 (ACKDn) of the IICA status register n (IICSn).
When the master receives the last data item, it does not return ACK and instead generates a stop condition. If a slave does not return ACK after receiving data, the master outputs a stop condition or restart condition and stops transmission. If ACK is not returned, the possible causes are as follows.

<1> Reception was not performed normally.
<2> The final data item was received.
<3> The reception side specified by the address does not exist.

To generate ACK, the reception side makes the SDAAn line low at the ninth clock (indicating normal reception).
Automatic generation of ACK is enabled by setting bit 2 (ACKEn) of IICA control register n0 (IICCTLn0) to 1. Bit 3 (TRCn) of the IICSn register is set by the data of the eighth bit that follows 7-bit address information. Usually, set the ACKEn bit to 1 for reception (TRCn = 0).
If a slave can receive no more data during reception (TRCn = 0) or does not require the next data item, then the slave must inform the master, by clearing the ACKEn bit to 0, that it will not receive any more data.
When the master does not require the next data item during reception (TRCn = 0), it must clear the ACKEn bit to 0 so that ACK is not generated. In this way, the master informs a slave at the transmission side that it does not require any more data (transmission will be stopped).

Figure 14 - 18 ACK



When the local address is received, ACK is automatically generated, regardless of the value of the ACKEn bit.
When an address other than that of the local address is received, ACK is not generated (NACK).
When an extension code is received, ACK is generated if the ACKEn bit is set to 1 in advance.
How ACK is generated when data is received differs as follows depending on the setting of the clock stretch timing.

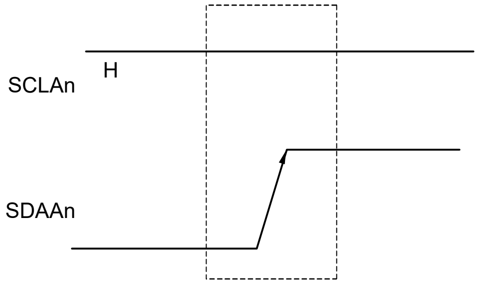
- When the falling edge of the eighth clock is set (bit 3 (WTIMn) of IICCTLn0 register = 0):
  By setting the ACKEn bit to 1 before releasing the clock stretch state, ACK is generated at the falling edge of the eighth clock of the SCLAn pin.
- When the falling edge of the ninth clock is set (bit 3 (WTIMn) of IICCTLn0 register = 1):
  ACK is generated by setting the ACKEn bit to 1 in advance.

Remark n = 0, 1

14.5.5 Stop condition

When the SCLAn pin is at high level, changing the SDAAn pin from low level to high level generates a stop condition.
A stop condition is a signal that the master device generates to the slave device when serial transfer has been completed. When the device is used as a slave, stop conditions can be detected.

Figure 14 - 19 Stop Condition



A stop condition is generated when bit 0 (SPTn) of IICA control register n0 (IICCTLn0) is set to 1. When the stop condition is detected, bit 0 (SPDn) of the IICA status register n (IICSn) is set to 1 and INTIICAn is generated when bit 4 (SPIEn) of the IICCTLn0 register is set to 1.

Remark n = 0, 1

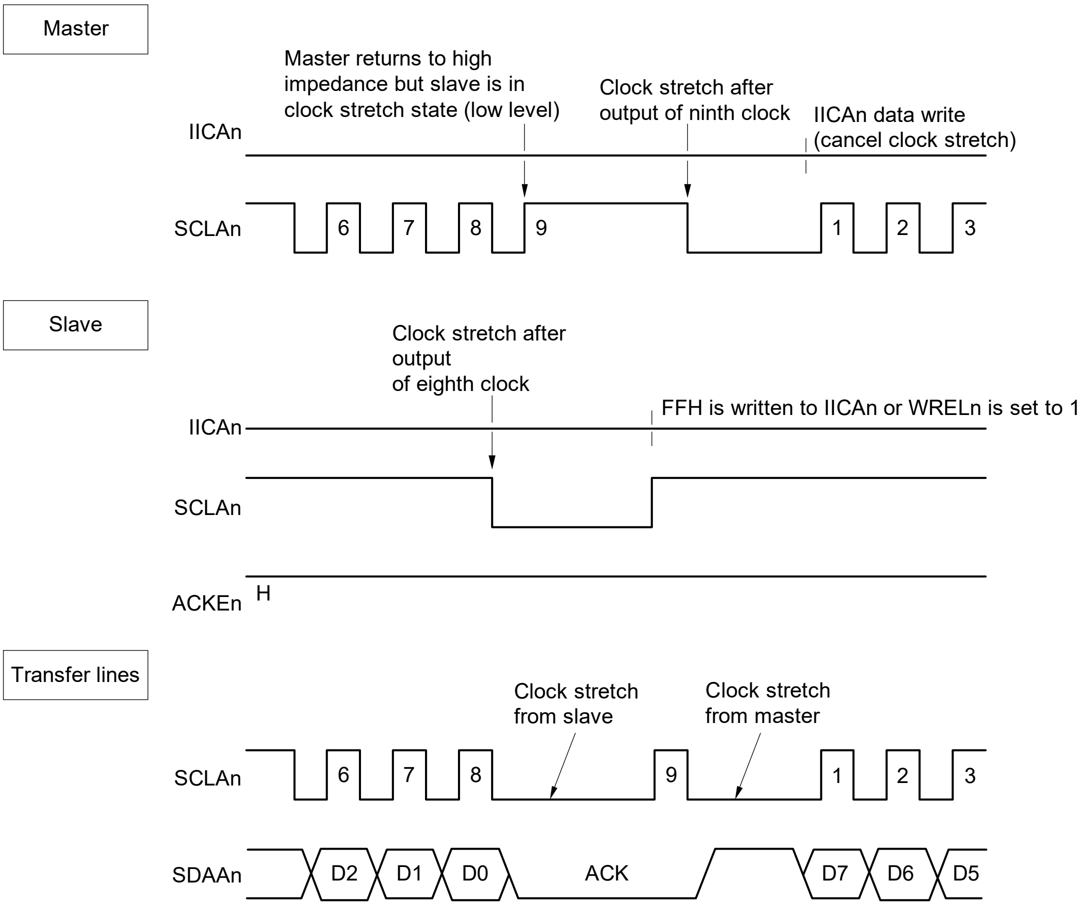
14.5.6 Clock stretch

The clock stretch is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (i.e., is in a clock stretch state).

Setting the SCLAn pin to low level notifies the communication partner of the clock stretch state. When clock stretch state has been canceled for both the master and slave devices, the next data transfer can begin.

Figure 14 - 20 Clock stretch (1/2)

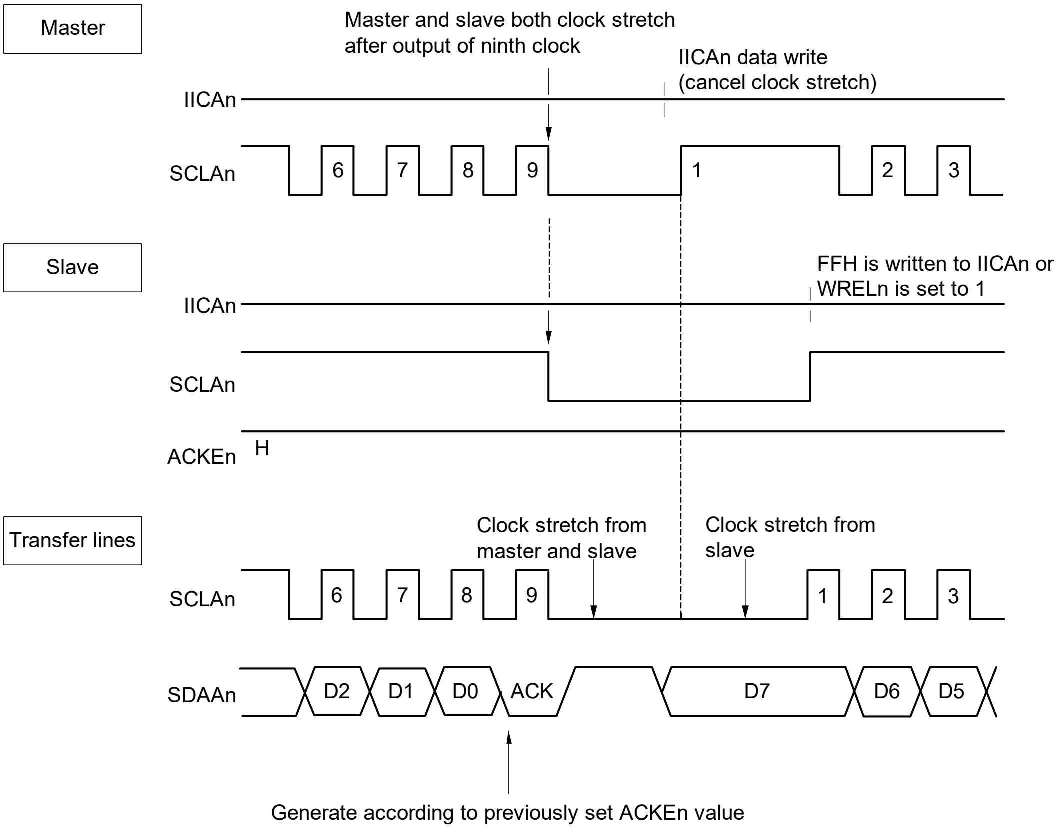
(1) When the master is at the falling edge of the ninth clock and the slave is at the falling edge of the eighth clock and clock stretching occurs
(master transmits, slave receives, and ACKEn = 1)


Master
Master returns to high impedance but slave is in clock stretch state (low level)
Clock stretch after output of ninth clock
IICAn data write (cancel clock stretch)
IICAn
SCLAn
6
7
8
9
1
2
3
Slave
Clock stretch after output of eighth clock
FFH is written to IICAn or WRELn is set to 1
IICAn
SCLAn
ACKEn
H
Transfer lines
Clock stretch from slave
Clock stretch from master
SCLAn
6
7
8
9
1
2
3
SDAAn
D2
D1
D0
ACK
D7
D6
D5

Remark n = 0, 1

Figure 14 - 20 Clock stretch (2/2)

(2) When master and slave are both at the falling edge of the ninth clock and clock stretching occurs (master transmits, slave receives, and ACKEn = 1)


Master
Master and slave both clock stretch after output of ninth clock
IICAn data write (cancel clock stretch)
IICAn
SCLAn
6
7
8
9
1
2
3
Slave
FFH is written to IICAn or WRELn is set to 1
IICAn
SCLAn
ACKEn
H
Transfer lines
Clock stretch from master and slave
Clock stretch from slave
SCLAn
6
7
8
9
1
2
3
SDAAn
D2
D1
D0
ACK
D7
D6
D5
Generate according to previously set ACKEn value

Remark ACKEn: Bit 2 of IICA control register n0 (IICCTLn0)
WRELn: Bit 5 of IICA control register n0 (IICCTLn0)

A clock stretch may be automatically generated depending on the setting of bit 3 (WTIMn) of IICA control register n0 (IICCTLn0).
Normally, the receiving side cancels the clock stretch state when bit 5 (WRELn) of the IICCTLn0 register is set to 1 or when FFH is written to the IICA shift register n (IICAn), and the transmitting side cancels the clock stretch state when data is written to the IICAn register.
The master device can also cancel the clock stretch state via either of the following methods.

- By setting bit 1 (STTn) of the IICCTLn0 register to 1
- By setting bit 0 (SPTn) of the IICCTLn0 register to 1

Remark n = 0, 1

14.5.7 Canceling clock stretch

The I^{2}C usually cancels a clock stretch state by the following processing.

- Writing data to the IICA shift register n (IICAn)
- Setting bit 5 (WRELn) of IICA control register n0 (IICCTLn0) (canceling clock stretch)
- Setting bit 1 (STTn) of the IICCTLn0 register (generating start condition) Note
- Setting bit 0 (SPTn) of the IICCTLn0 register (generating stop condition) Note

Note Master only

When the above clock stretch canceling processing is executed, the I^{2}C cancels the clock stretch state and communication is resumed.
To cancel a clock stretch state and transmit data (including addresses), write the data to the IICAn register.
To receive data after canceling a clock stretch state, or to complete data transmission, set bit 5 (WRELn) of the IICCTLn0 register to 1.
To generate a restart condition after canceling a clock stretch state, set bit 1 (STTn) of the IICCTLn0 register to 1.
To generate a stop condition after canceling a clock stretch state, set bit 0 (SPTn) of the IICCTLn0 register to 1.
Execute the canceling processing only once for one clock stretch state.
If, for example, data is written to the IICAn register after canceling a clock stretch state by setting the WRELn bit to 1, an incorrect value may be output to SDAAn line because the timing for changing the SDAAn line conflicts with the timing for writing the IICAn register.
In addition to the above, communication is stopped if the IICEn bit is cleared to 0 when communication has been aborted, so that the clock stretch state can be canceled.
If the I^{2}C bus has deadlocked due to noise, processing is saved from communication by setting bit 6 (LRELn) of the IICCTLn0 register, so that the clock stretch state can be canceled.

Caution If a processing to cancel a clock stretch state is executed when WUPn = 1, the clock stretch state will not be canceled.

Remark n = 0, 1

14.5.8 Interrupt request (INTIICAn) generation timing and clock stretch control

The setting of bit 3 (WTIMn) of IICA control register n0 (IICCTLn0) determines the timing by which INTIICAn is generated and the corresponding clock stretch control, as shown in Table 14 - 2.

Table 14 - 2 INTIICAn Generation Timing and Clock Stretch Control

WTIMn	During Slave Device Operation			During Master Device Operation		
	Address	Data Reception	Data Transmission	Address	Data Reception	Data Transmission
0	9 Notes 1, 2	8 Note 2	8 Note 2	9	8	8
1	9 Notes 1, 2	9 Note 2	9 Note 2	9	9	9

Note 1. The slave device's INTIICAn signal and clock stretch period occurs at the falling edge of the ninth clock only when there is a match with the address set to the slave address register n (SVAn).
At this point, ACK is generated regardless of the value set to the IICCTLn0 register's bit 2 (ACKEn). For a slave device that has received an extension code, INTIICAn occurs at the falling edge of the eighth clock.
However, if the address does not match after restart, INTIICAn is generated at the falling edge of the 9th clock, but clock stretch does not occur.

Note 2. If the received address does not match the contents of the slave address register n (SVAn) and extension code is not received, neither INTIICAn nor a clock stretch occurs.

Remark The numbers in the table indicate the number of the serial clock's clock signals. Interrupt requests and clock stretch control are both synchronized with the falling edge of these clock signals.

(1) During address transmission/reception

- Slave device operation: Interrupt and clock stretch timing are determined depending on the conditions described in Notes 1 and 2 above, regardless of the WTIMn bit.
- Master device operation: Interrupt and clock stretch timing occur at the falling edge of the ninth clock regardless of the WTIMn bit.

(2) During data reception

- Master/slave device operation: Interrupt and clock stretch timing are determined according to the WTIMn bit.

(3) During data transmission

- Master/slave device operation: Interrupt and clock stretch timing are determined according to the WTIMn bit.

Remark n = 0, 1

(4) Clock stretch cancellation method

The four clock stretch cancellation methods are as follows.

- Writing data to the IICA shift register n (IICAn)
- Setting bit 5 (WRELn) of IICA control register n0 (IICCTLn0) (canceling clock stretch)
- Setting bit 1 (STTn) of IICCTLn0 register (generating start condition) Note
- Setting bit 0 (SPTn) of IICCTLn0 register (generating stop condition) Note

Note Master only.

When clock stretching timing is set to the falling edge of the eighth clock (WTIMn = 0), the presence/absence of ACK generation must be determined prior to clock stretch cancellation.

(5) Stop condition detection

INTIICAn is generated when a stop condition is detected (only when SPIEn = 1).

14.5.9 Address match detection method

In I^{2}C bus mode, the master device can select a particular slave device by transmitting the corresponding slave address.

Address match can be detected automatically by hardware. An interrupt request (INTIICAn) occurs when the address set to the slave address register n (SVAn) matches the slave address sent by the master device, or when an extension code has been received.

14.5.10 Error detection

In I^{2}C bus mode, the status of the serial data bus (SDAAn) during data transmission is captured by the IICA shift register n (IICAn) of the transmitting device, so the IICA data prior to transmission can be compared with the transmitted IICA data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

Remark n = 0, 1

14.5.11 Extension code

(1) When the higher 4 bits of the receive address are either “0000” or “1111”, the extension code reception flag (EXCn) is set to 1 for extension code reception and an interrupt request (INTIICAn) is issued at the falling edge of the eighth clock. The local address stored in the slave address register n (SVAn) is not affected.

(2) The settings below are specified if 11110xx0 is transferred from the master by using a 10-bit address transfer when the SVAn register is set to 11110xx0. Note that INTIICAn occurs at the falling edge of the eighth clock.

- Higher four bits of data match: EXCn = 1
- Seven bits of data match: COIn = 1

Remark EXCn: Bit 5 of IICA status register n (IICSn)
COIn: Bit 4 of IICA status register n (IICSn)

(3) Since the processing after the interrupt request occurs differs according to the data that follows the extension code, such processing is performed by software.
If the extension code is received while a slave device is operating, then the slave device is participating in communication even if its address does not match.
For example, after the extension code is received, if you do not wish to operate the target device as a slave device, set bit 6 (LRELn) of IICA control register n0 (IICCTLn0) to 1 to set the standby mode for the next communication operation.

Table 14 - 3 Bit Definitions of Major Extension Codes

Slave Address	R/W Bit	Description
0000 000	0	General call address
1111 0xx	0	10-bit slave address specification (during address authentication)
1111 0xx	1	10-bit slave address specification (after address match, when read command is issued)

Remark 1. See the I²C bus specifications issued by NXP Semiconductors for details of extension codes other than those described above.

Remark 2. n = 0, 1

14.5.12 Arbitration

When several master devices simultaneously generate a start condition (when the STTn bit is set to 1 before the STDn bit is set to 1), communication among the master devices is performed as the number of clocks are adjusted until the data differs. This kind of operation is called arbitration.

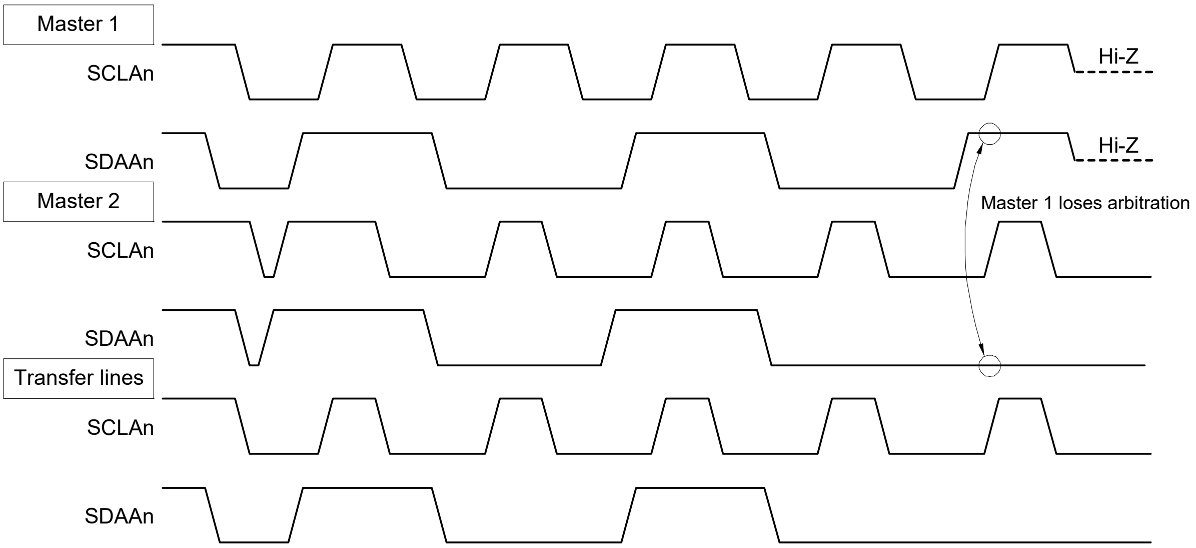
When one of the master devices loses in arbitration, an arbitration loss flag (ALDn) in the IICA status register n (IICSn) is set (1) via the timing by which the arbitration loss occurred, and the SCLAn and SDAAn lines are both set to high impedance, which releases the bus.

The arbitration loss is detected based on the timing of the next interrupt request (the eighth or ninth clock, when a stop condition is detected, etc.) and the ALDn = 1 setting that has been made by software.

For details of interrupt request timing, see **14.5.8 Interrupt request (INTIICAn) generation timing and clock stretch control**.

Remark STDn: Bit 1 of IICA status register n (IICSn)
STTn: Bit 1 of IICA control register n0 (IICCTLn0)

Figure 14 - 21 Arbitration Timing Example



Remark n = 0, 1

Table 14 - 4 Status During Arbitration and Interrupt Request Generation Timing

Status During Arbitration	Interrupt Request Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer [Note 1]
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During ACK transfer period after data transmission	
When restart condition is detected during data transfer	
When stop condition is detected during data transfer	When stop condition is generated (when SPIEn = 1) [Note 2]
When data is at low level while attempting to generate a restart condition	At falling edge of eighth or ninth clock following byte transfer [Note 1]
When stop condition is detected while attempting to generate a restart condition	When stop condition is generated (when SPIEn = 1) [Note 2]
When data is at low level while attempting to generate a stop condition	At falling edge of eighth or ninth clock following byte transfer [Note 1]
When SCLAn is at low level while attempting to generate a restart condition	

Note 1. When the WTIMn bit (bit 3 of IICA control register n0 (IICCTLn0)) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIMn = 0 and the extension code's slave address is received, an interrupt request occurs at the falling edge of the eighth clock.

Note 2. When there is a chance that arbitration will occur, set SPIEn = 1 for master device operation.

Remark 1. SPIEn: Bit 4 of IICA control register n0 (IICCTLn0)

Remark 2. n = 0, 1

14.5.13 Wakeup function

The I^{2}C bus slave function is a function that generates an interrupt request signal (INTIICAn) when a local address and extension code have been received.

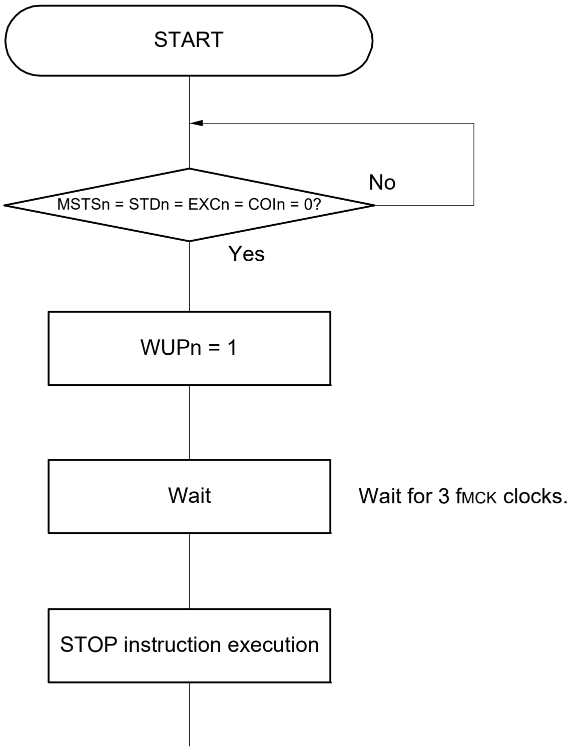
This function makes processing more efficient by preventing unnecessary INTIICAn signal from occurring when addresses do not match.

When a start condition is detected, wakeup standby mode is set. This wakeup standby mode is in effect while addresses are transmitted due to the possibility that an arbitration loss may change the master device (which has generated a start condition) to a slave device.

To use the wakeup function in the STOP mode, set the WUPn bit to 1. Addresses can be received regardless of the operation clock. An interrupt request signal (INTIICAn) is also generated when a local address and extension code have been received. Operation returns to normal operation by using an instruction to clear (0) the WUPn bit after this interrupt has been generated.

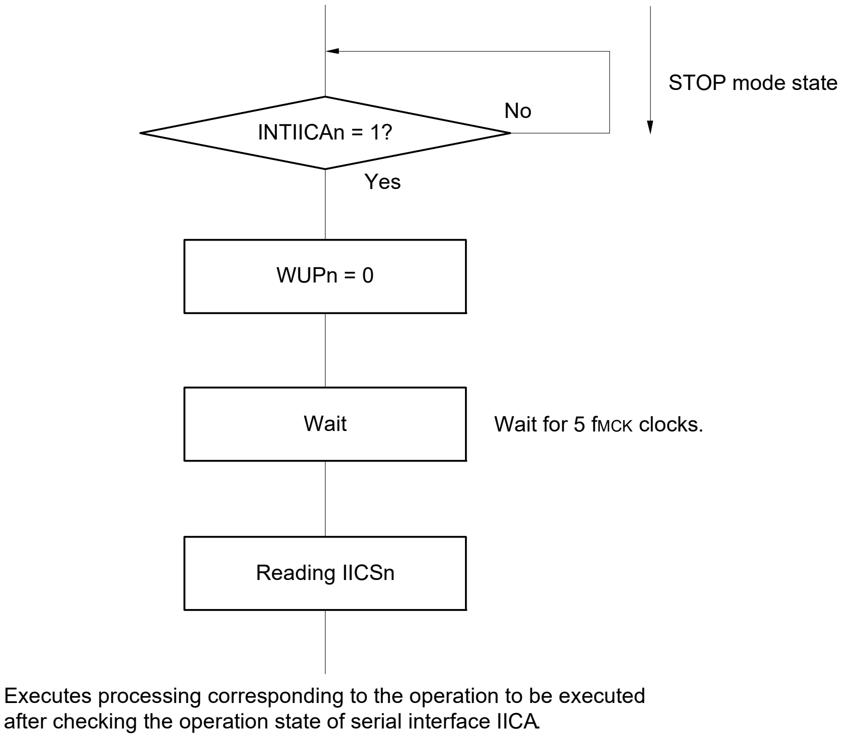
Figure 14 - 22 shows the flow for setting WUPn = 1 and Figure 14 - 23 shows the flow for setting WUPn = 0 upon an address match.

Figure 14 - 22 Flow When Setting WUPn = 1



Remark n = 0, 1

Figure 14 - 23 Flow When Setting WUPn = 0 upon Address Match (Including Extension Code Reception)

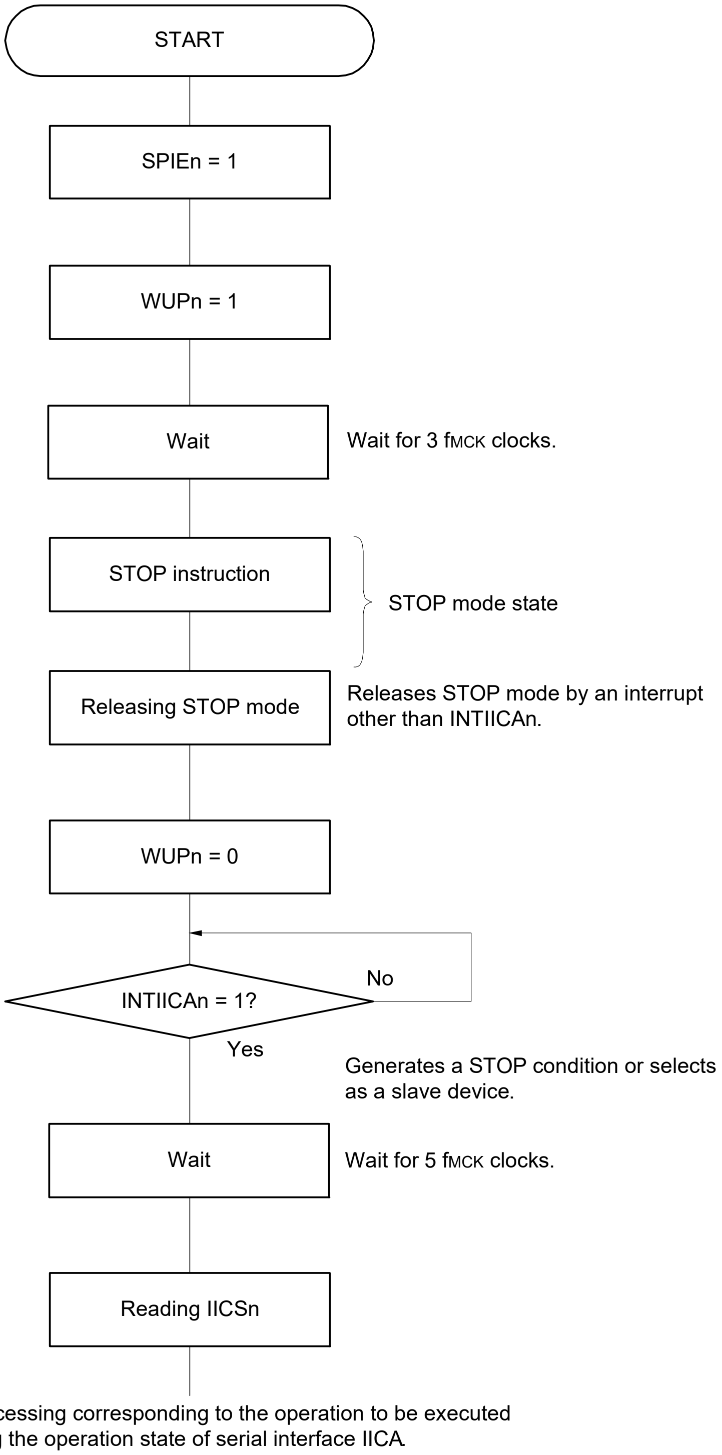

STOP mode state
INTIICAn = 1?
No
Yes
WUPn = 0
Wait
Wait for 5 fMCK clocks.
Reading IICSn
Executes processing corresponding to the operation to be executed
after checking the operation state of serial interface IICA.

Use the following flows to perform the processing to release the STOP mode other than by an interrupt request (INTIICAn) generated from serial interface IICA.

- When operating as the master device for the next IIC communication: Flow shown in Figure 14 - 24
- When operating as a slave device for the next IIC communication:
  When the INTIICAn interrupt is used to return from the mode:
    Same as the flow in Figure 14 - 23
  When an interrupt other than the INTIICAn interrupt is used to return from the mode:
    Continue operation while WUPn = 1 until an INTIICAn interrupt is generated.

Remark n = 0, 1

Figure 14 - 24 When Operating as Master Device after Releasing STOP Mode other than by INTIICAn


START
SPIEn = 1
WUPn = 1
Wait
Wait for 3 fMCK clocks.
STOP instruction
STOP mode state
Releasing STOP mode
Releases STOP mode by an interrupt other than INTIICAn.
WUPn = 0
INTIICAn = 1?
No
Yes
Generates a STOP condition or selects as a slave device.
Wait
Wait for 5 fMCK clocks.
Reading IICSn
Executes processing corresponding to the operation to be executed after checking the operation state of serial interface IICA.

Remark n = 0, 1

14.5.14 Communication reservation

(1) When communication reservation function is enabled (bit 0 (IICRSVn) of IICA flag register n (IICFn) = 0)

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled (ACK is not returned and the bus was released by setting bit 6 (LRELn) of IICA control register n0 (IICCTLn0) to 1 and saving communication).

If bit 1 (STTn) of the IICCTLn0 register is set to 1 while the bus is not used (after a stop condition is detected), a start condition is automatically generated and wait state is set.

If an address is written to the IICA shift register n (IICAn) after bit 4 (SPIEn) of the IICCTLn0 register was set to 1, and it was detected by generation of an interrupt request signal (INTIICAn) that the bus was released (detection of the stop condition), then the device automatically starts communication as the master. Data written to the IICAn register before the stop condition is detected is invalid.

When the STTn bit has been set to 1, the operation mode (as start condition or as communication reservation) is determined according to the bus status.

- If the bus has been released..a start condition is generated
- If the bus has not been released (standby mode)...........communication reservation

Check whether the communication reservation operates or not by using the MSTSn bit (bit 7 of the IICA status register n (IICSn)) after the STTn bit is set to 1 and the wait time elapses.

Use software to secure the wait time calculated by the following expression.

Wait time from setting STTn = 1 to checking the MSTSn flag:

$(\text{IICWLn setting value} + \text{IICWHn setting value} + 4)/f_{MCK} + t_F \times 2$

Remark 1. IICWLn: IICA low-level width setting register n

IICWHn: IICA high-level width setting register n

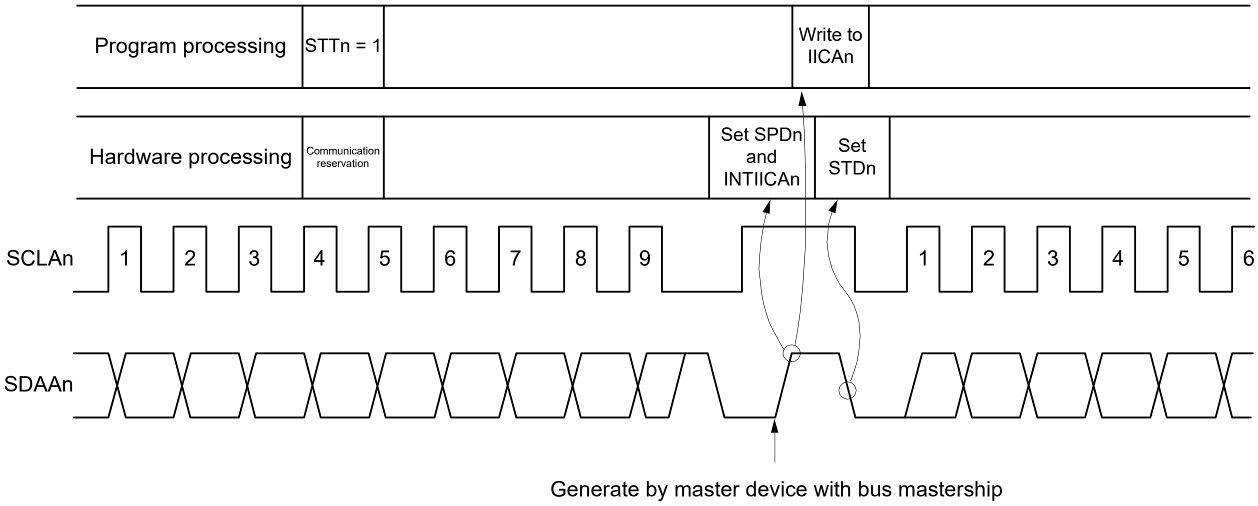
t_F: SDAAn and SCLAn signal falling times

f_{MCK}: IICA operating clock frequency

Remark 2. n = 0, 1

Figure 14 - 25 shows the Communication Reservation Timing.

Figure 14 - 25 Communication Reservation Timing



Remark IICAn: IICA shift register n

STTn: Bit 1 of IICA control register n0 (IICCTLn0)

STDn: Bit 1 of IICA status register n (IICSn)

SPDn: Bit 0 of IICA status register n (IICSn)

Communication reservations are accepted via the timing shown in Figure 14 - 26. After bit 1 (STDn) of the IICA status register n (IICSn) is set to 1, a communication reservation can be made by setting bit 1 (STTn) of IICA control register n0 (IICCTLn0) to 1 before a stop condition is detected.

Figure 14 - 26 Timing for Accepting Communication Reservations

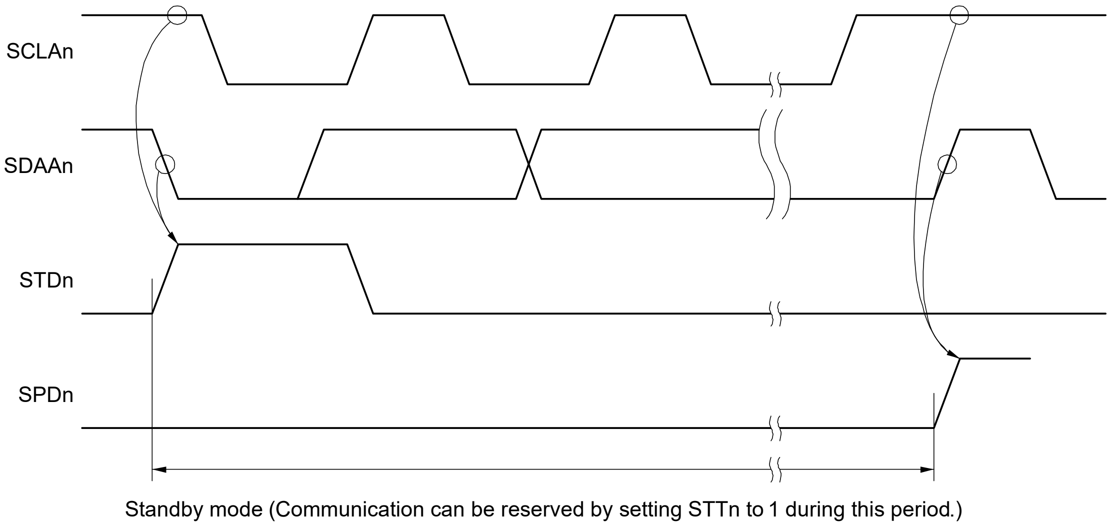
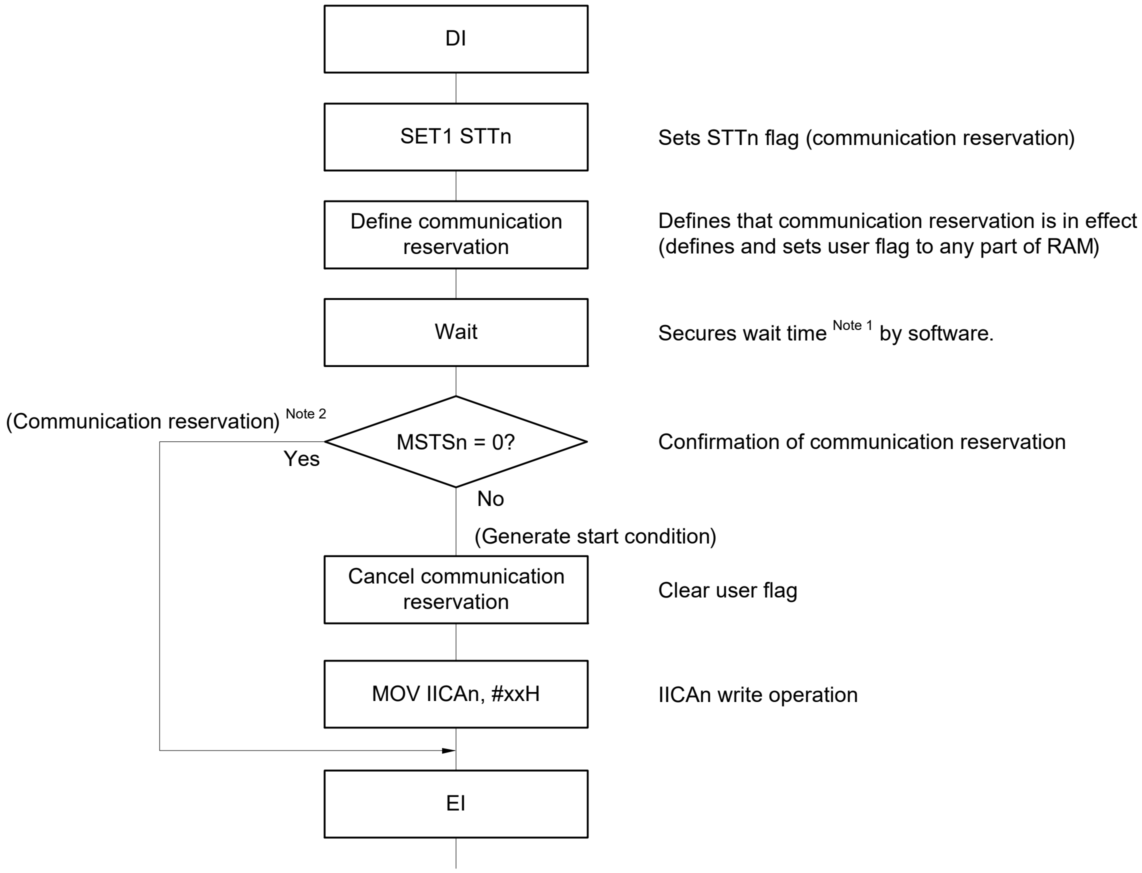

SCLAn
SDAAn
STDn
SPDn
Standby mode (Communication can be reserved by setting STTn to 1 during this period.)

Figure 14 - 27 shows the Communication Reservation Protocol.

Remark n = 0, 1

Figure 14 - 27 Communication Reservation Protocol


DI
SET1 STTn
Sets STTn flag (communication reservation)
Define communication reservation
Defines that communication reservation is in effect (defines and sets user flag to any part of RAM)
Wait
Secures wait time Note 1 by software.
(Communication reservation) Note 2
MSTSn = 0?
Confirmation of communication reservation
Yes
No
(Generate start condition)
Cancel communication reservation
Clear user flag
MOV IICAn, #xxH
IICAn write operation
EI

Note 1. The wait time is calculated as follows.
(IICWLn setting value + IICWHn setting value + 4)/f_{MCK} + $t_F \times 2$

Note 2. The communication reservation operation executes a write to the IICA shift register n (IICAn) when a stop condition interrupt request occurs.

Remark1.	STTn:	Bit 1 of IICA control register n0 (IICCTLn0)
	MSTSn:	Bit 7 of IICA status register n (IICSn)
	IICAn:	IICA shift register n
	IICWLn:	IICA low-level width setting register n
	IICWHn:	IICA high-level width setting register n
	t_F:	SDAAn and SCLAn signal falling times
	f_{MCK}:	IICA operating clock frequency

Remark2. n = 0, 1

(2) When communication reservation function is disabled (bit 0 (IICRSVn) of IICA flag register n (IICFn) = 1)

When bit 1 (STTn) of IICA control register n0 (IICCTLn0) is set to 1 when the bus is not used in a communication during bus communication, this request is rejected and a start condition is not generated. The following two statuses are included in the status where bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled (ACK is not returned and the bus was released by setting bit 6 (LRELn) of the IICCTLn0 register to 1 and saving communication)

To confirm whether the start condition was generated or request was rejected, check STCFn (bit 7 of the IICFn register). It takes up to five f_{MCK} clocks until the STCFn bit is set to 1 after setting STTn = 1. Therefore, secure this time by software.

Remark n = 0, 1

14.5.15 Cautions

(1) When STCENn = 0

Immediately after I^{2}C operation is enabled (IICEn = 1), the bus communication status (IICBSYn = 1) is recognized regardless of the actual bus status. When changing from a mode in which no stop condition has been detected to a master device communication mode, first generate a stop condition to release the bus, then perform master device communication.

When using multiple masters, it is not possible to perform master device communication when the bus has not been released (when a stop condition has not been detected).

Use the following sequence for generating a stop condition.

<1> Set IICA control register n1 (IICCTLn1).
<2> Set bit 7 (IICEn) of IICA control register n0 (IICCTLn0) to 1.
<3> Set bit 0 (SPTn) of the IICCTLn0 register to 1.

(2) When STCENn = 1

Immediately after I^{2}C operation is enabled (IICEn = 1), the bus released status (IICBSYn = 0) is recognized regardless of the actual bus status. To generate the first start condition (STTn = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

(3) If other I^{2}C communications are already in progress

If I^{2}C operation is enabled and the device participates in communication already in progress when the SDAAn pin is low and the SCLAn pin is high, the macro of I^{2}C recognizes that the SDAAn pin has gone low (detects a start condition). If the value on the bus at this time can be recognized as an extension code, ACK is returned, but this interferes with other I^{2}C communications. To avoid this, start I^{2}C in the following sequence.

<1> Clear bit 4 (SPIEn) of the IICCTLn0 register to 0 to disable generation of an interrupt request signal (INTIICAn) when the stop condition is detected.
<2> Set bit 7 (IICEn) of the IICCTLn0 register to 1 to enable the operation of I^{2}C.
<3> Wait for detection of the start condition.
<4> Set bit 6 (LRELn) of the IICCTLn0 register to 1 before ACK is returned (4 to 72 f_{MCK} clocks after setting the IICEn bit to 1), to forcibly disable detection.

(4) Setting the STTn and SPTn bits (bits 1 and 0 of the IICCTLn0 register) again after they are set and before they are cleared to 0 is prohibited.

(5) When transmission is reserved, set the SPIEn bit (bit 4 of the IICCTLn0 register) to 1 so that an interrupt request is generated when the stop condition is detected. Transfer is started when communication data is written to the IICA shift register n (IICAn) after the interrupt request is generated. Unless the interrupt is generated when the stop condition is detected, the device stops in the wait state because the interrupt request is not generated when communication is started. However, it is not necessary to set the SPIEn bit to 1 when the MSTSn bit (bit 7 of the IICA status register n (IICSn)) is detected by software.

Remark n = 0, 1

14.5.16 Communication operations

The following shows three operation procedures with the flowchart.

(1) Master operation in single master system

The flowchart when using the RL78/G11 as the master in a single master system is shown below.

This flowchart is broadly divided into the initial settings and communication processing. Execute the initial settings at startup. If communication with the slave is required, prepare the communication and then execute communication processing.

(2) Master operation in multimaster system

In the I^{2}C bus multimaster system, whether the bus is released or used cannot be judged by the I^{2}C bus specifications when the bus takes part in a communication. Here, when data and clock are at a high level for a certain period (1 frame), the RL78/G11 takes part in a communication with bus released state.

This flowchart is broadly divided into the initial settings, communication waiting, and communication processing. The processing when the RL78/G11 looses in arbitration and is specified as the slave is omitted here, and only the processing as the master is shown. Execute the initial settings at startup to take part in a communication. Then, wait for the communication request as the master or wait for the specification as the slave. The actual communication is performed in the communication processing, and it supports the transmission/reception with the slave and the arbitration with other masters.

(3) Slave operation

An example of when the RL78/G11 is used as the I^{2}C bus slave is shown below.

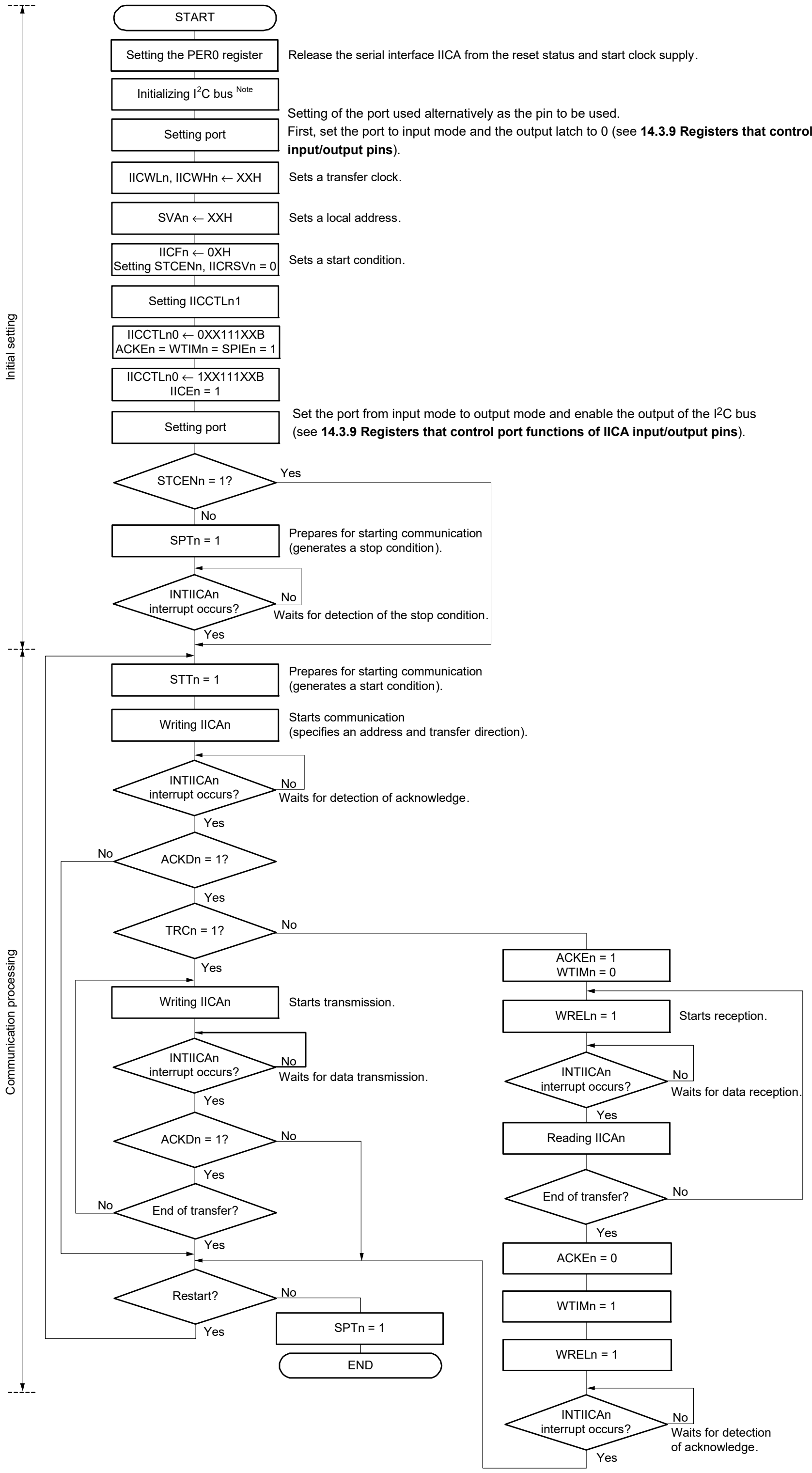
When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIICAn interrupt occurrence (communication waiting). When an INTIICAn interrupt occurs, the communication status is judged and its result is passed as a flag over to the main processing.

By checking the flags, necessary communication processing is performed.

Remark n = 0, 1

(1) Master operation in single master system

Figure 14 - 28 Master Operation in Single-Master System


Initial setting
Communication processing
START
Setting the PER0 register
Release the serial interface IICA from the reset status and start clock supply.
Initializing I2C bus Note
Setting port
Setting of the port used alternatively as the pin to be used.
First, set the port to input mode and the output latch to 0 (see 14.3.9 Registers that control port functions of IICA input/output pins).
IICWLn, IICWHn ← XXH
Sets a transfer clock.
SVAn ← XXH
Sets a local address.
IICFn ← 0XH
Setting STCENn, IICRSVn = 0
Sets a start condition.
Setting IICCTLn1
IICCTLn0 ← 0XX111XXB
ACKEn = WTIMn = SPIEn = 1
IICCTLn0 ← 1XX111XXB
IICEn = 1
Setting port
Set the port from input mode to output mode and enable the output of the I2C bus
(see 14.3.9 Registers that control port functions of IICA input/output pins).
STCENn = 1?
Yes
No
SPTn = 1
Prepares for starting communication
(generates a stop condition).
INTIICAn
interrupt occurs?
No
Waits for detection of the stop condition.
Yes
STTn = 1
Prepares for starting communication
(generates a start condition).
Writing IICAn
Starts communication
(specifies an address and transfer direction).
INTIICAn
interrupt occurs?
No
Waits for detection of acknowledge.
Yes
No
ACKDn = 1?
Yes
TRCn = 1?
No
Yes
ACKEn = 1
WTIMn = 0
Writing IICAn
Starts transmission.
WRELn = 1
Starts reception.
INTIICAn
interrupt occurs?
No
Waits for data transmission.
Yes
INTIICAn
interrupt occurs?
No
Waits for data reception.
Yes
ACKDn = 1?
No
Yes
Reading IICAn
No
End of transfer?
Yes
End of transfer?
No
Yes
ACKEn = 0
Restart?
No
Yes
SPTn = 1
END
WTIMn = 1
WRELn = 1
INTIICAn
interrupt occurs?
No
Waits for detection
of acknowledge.
Yes

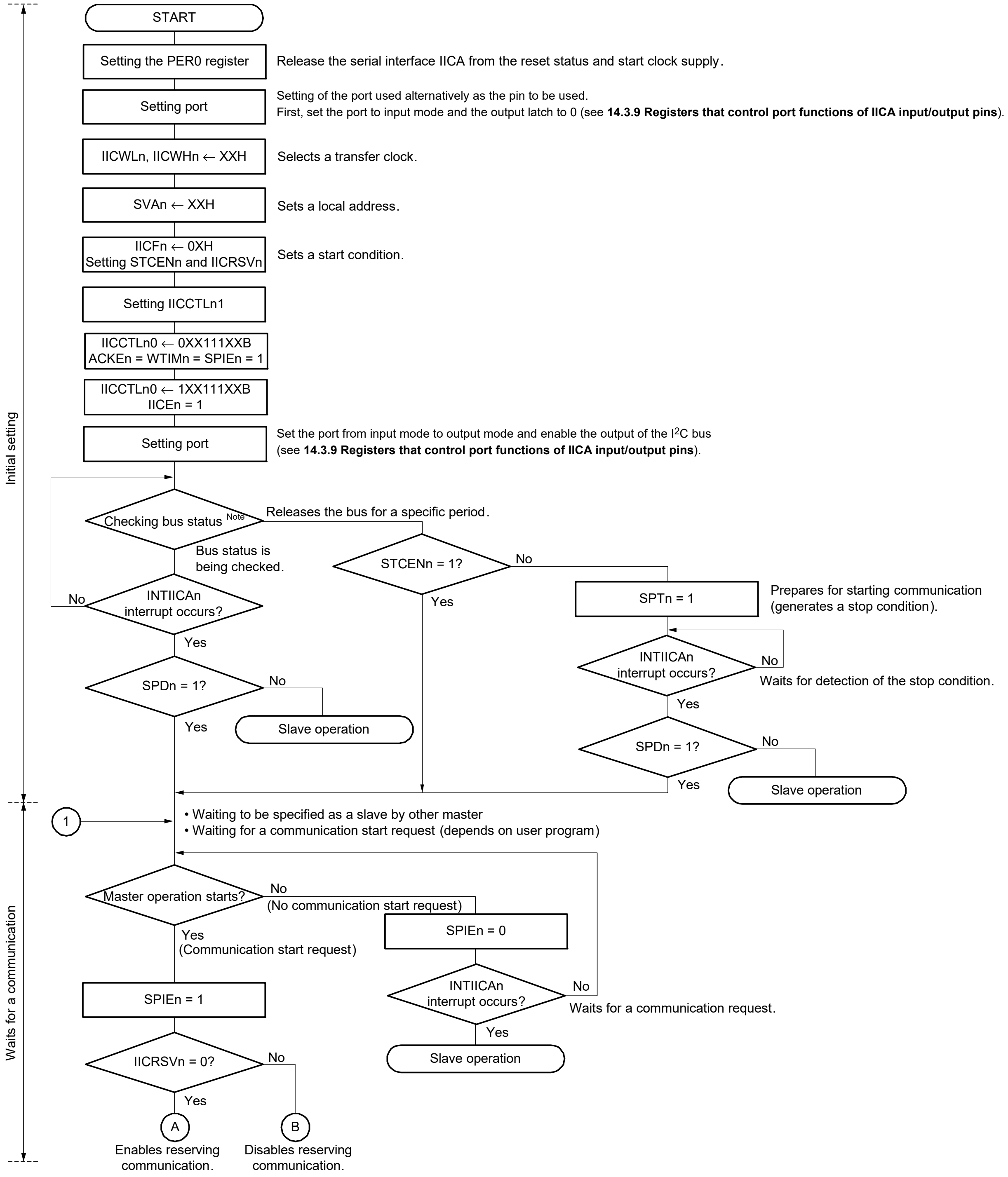
Note Release (SCLAn and SDAAn pins = high level) the I2C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDAAn pin, for example, set the SCLAn pin in the output port mode, and output a clock pulse from the output port until the SDAAn pin is constantly at high level.

Remark1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

Remark2. n = 0, 1

(2) Master operation in multimaster system

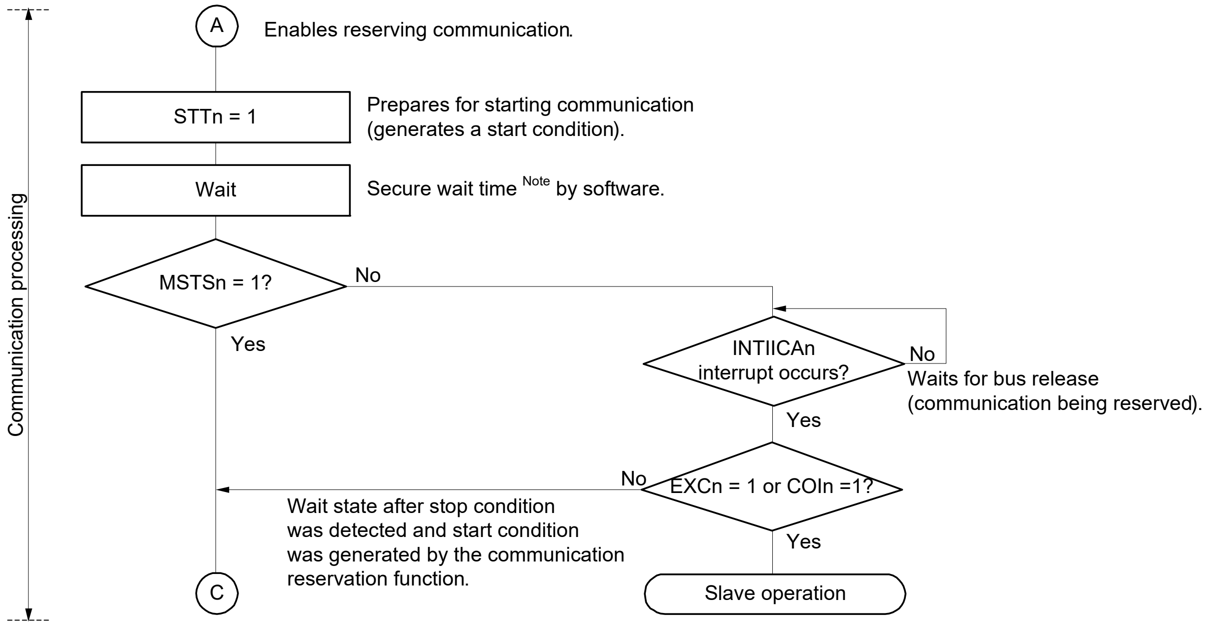
Figure 14 - 29 Master Operation in Multi-Master System (1/3)


Initial setting
START
Setting the PER0 register
Release the serial interface IICA from the reset status and start clock supply.
Setting port
Setting of the port used alternatively as the pin to be used.
First, set the port to input mode and the output latch to 0 (see 14.3.9 Registers that control port functions of IICA input/output pins).
IICWLn, IICWHn ← XXH
Selects a transfer clock.
SVAn ← XXH
Sets a local address.
IICFn ← 0XH
Setting STCENn and IICRSVn
Sets a start condition.
Setting IICCTLn1
IICCTLn0 ← 0XX111XXB
ACKEn = WTIMn = SPIEn = 1
IICCTLn0 ← 1XX111XXB
IICEn = 1
Setting port
Set the port from input mode to output mode and enable the output of the I2C bus
(see 14.3.9 Registers that control port functions of IICA input/output pins).
Checking bus status Note
Releases the bus for a specific period.
Bus status is being checked.
STCENn = 1?
No
Yes
SPTn = 1
Prepares for starting communication (generates a stop condition).
No
INTIICAn interrupt occurs?
Yes
INTIICAn interrupt occurs?
No
Waits for detection of the stop condition.
Yes
SPDn = 1?
No
Yes
Slave operation
SPDn = 1?
No
Yes
Slave operation
Waits for a communication
1
• Waiting to be specified as a slave by other master
• Waiting for a communication start request (depends on user program)
Master operation starts?
No
(No communication start request)
Yes
(Communication start request)
SPIEn = 0
INTIICAn interrupt occurs?
No
Waits for a communication request.
Yes
Slave operation
SPIEn = 1
IICRSVn = 0?
No
Yes
A
B
Enables reserving communication.
Disables reserving communication.

Note Confirm that the bus is released (CLDn bit = 1, DADn bit = 1) for a specific period (for example, for a period of one frame). If the SDAAn pin is constantly at low level, decide whether to release the I2C bus (SCLAn and SDAAn pins = high level) in conformance with the specifications of the product that is communicating.

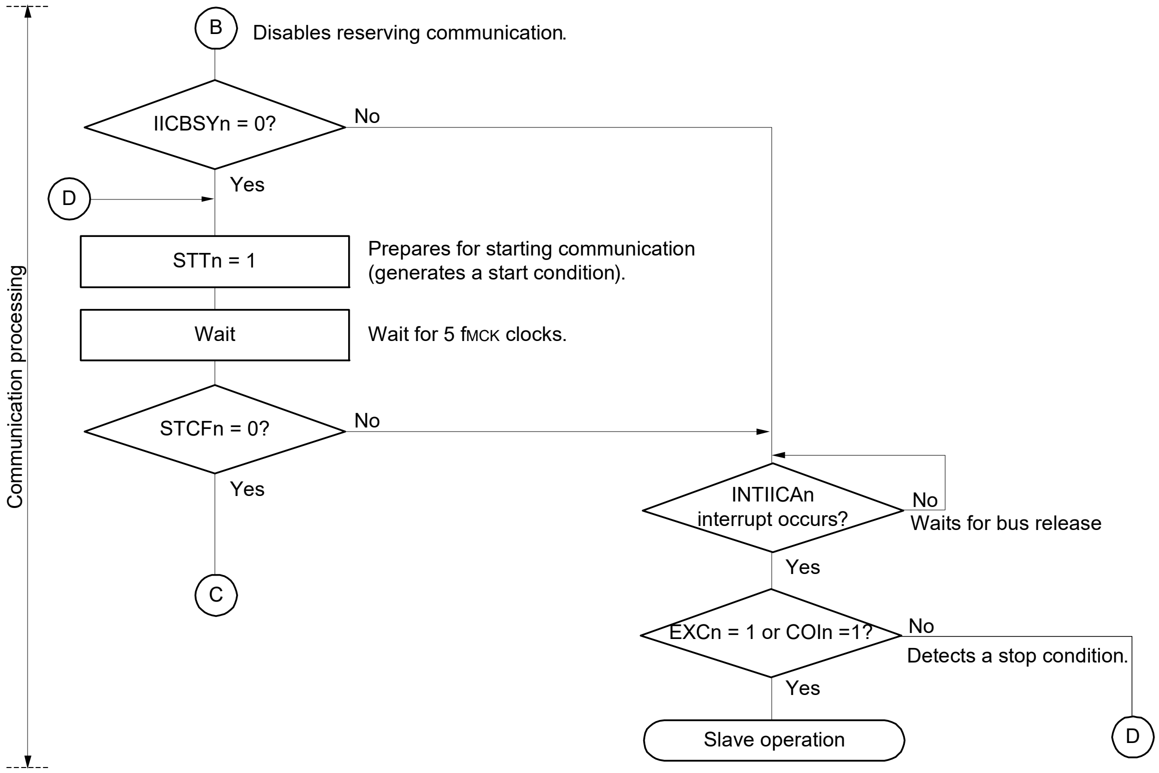
Remark n = 0, 1

Figure 14 - 29 Master Operation in Multi-Master System (2/3)


Communication processing
A
Enables reserving communication.
STTn = 1
Prepares for starting communication (generates a start condition).
Wait
Secure wait time Note by software.
MSTSn = 1?
No
Yes
INTIICAn interrupt occurs?
No
Waits for bus release (communication being reserved).
Yes
EXCn = 1 or COIn =1?
No
Wait state after stop condition was detected and start condition was generated by the communication reservation function.
Yes
C
Slave operation

Note The wait time is calculated as follows.

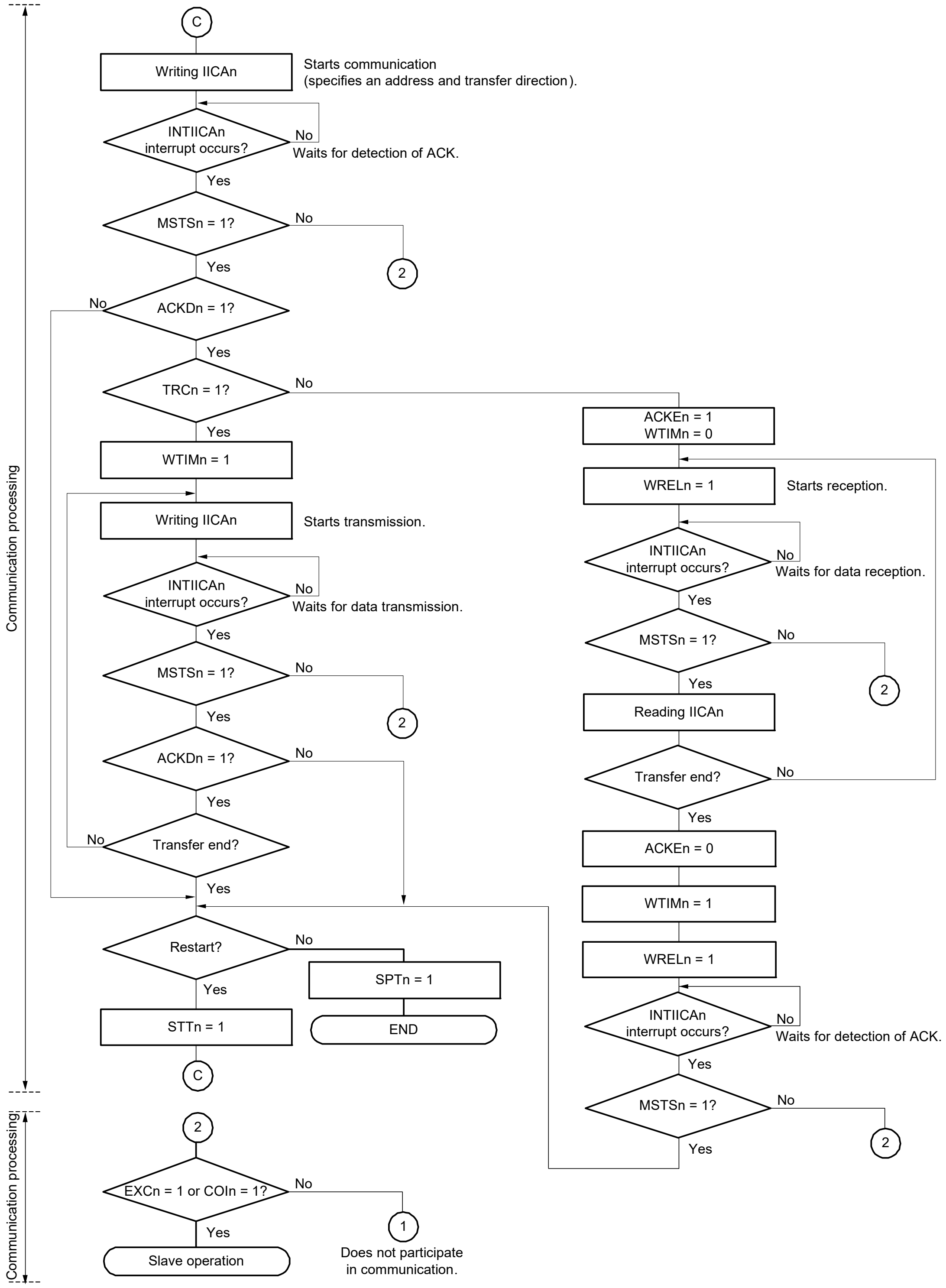
(IICWLn setting value + IICWHn setting value + 4)/f_{MCK} + t_F × 2



Remark1.	IICWLn:	IICA low-level width setting register n
	IICWHn:	IICA high-level width setting register n
	t_F:	SDAAn and SCLAn signal falling times
	f_{MCK}:	IICA operating clock frequency

Remark2. n = 0, 1

Figure 14 - 29 Master Operation in Multi-Master System (3/3)



Remark 1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

Remark 2. To use the device as a master in a multi-master system, read the MSTSn bit each time interrupt INTIICAn has occurred to check the arbitration result.

Remark 3. To use the device as a slave in a multi-master system, check the status by using the IICA status register n (IICSn) and IICA flag register n (IICFn) each time interrupt INTIICAn has occurred, and determine the processing to be performed next.

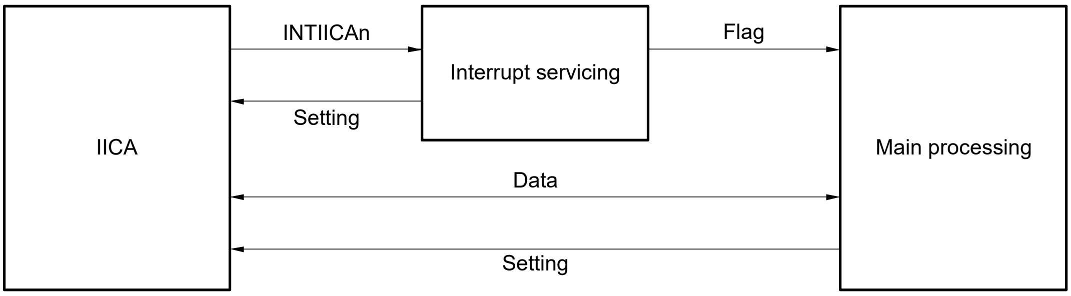
Remark 4. n = 0, 1

(3) Slave operation

The processing procedure of the slave operation is as follows.

Basically, the slave operation is event-driven. Therefore, processing by the INTIICAn interrupt (processing that must substantially change the operation status such as detection of a stop condition during communication) is necessary.

In the following explanation, it is assumed that the extension code is not supported for data communication. It is also assumed that the INTIICAn interrupt servicing only performs status transition processing, and that actual data communication is performed by the main processing.



Therefore, data communication processing is performed by preparing the following three flags and passing them to the main processing instead of INTIICAn.

<1> Communication mode flag

This flag indicates the following two communication statuses.

•Clear mode: Status in which data communication is not performed

•Communication mode: Status in which data communication is performed (from valid address detection to stop condition detection, no detection of ACK from master, address mismatch)

<2> Ready flag

This flag indicates that data communication is enabled. Its function is the same as the INTIICAn interrupt for ordinary data communication. This flag is set by interrupt servicing and cleared by the main processing. Clear this flag by interrupt servicing when communication is started. However, the ready flag is not set by interrupt servicing when the first data is transmitted. Therefore, the first data is transmitted without the flag being cleared (an address match is interpreted as a request for the next data).

<3> Communication direction flag

This flag indicates the direction of communication. Its value is the same as the TRCn bit.

Remark n = 0, 1

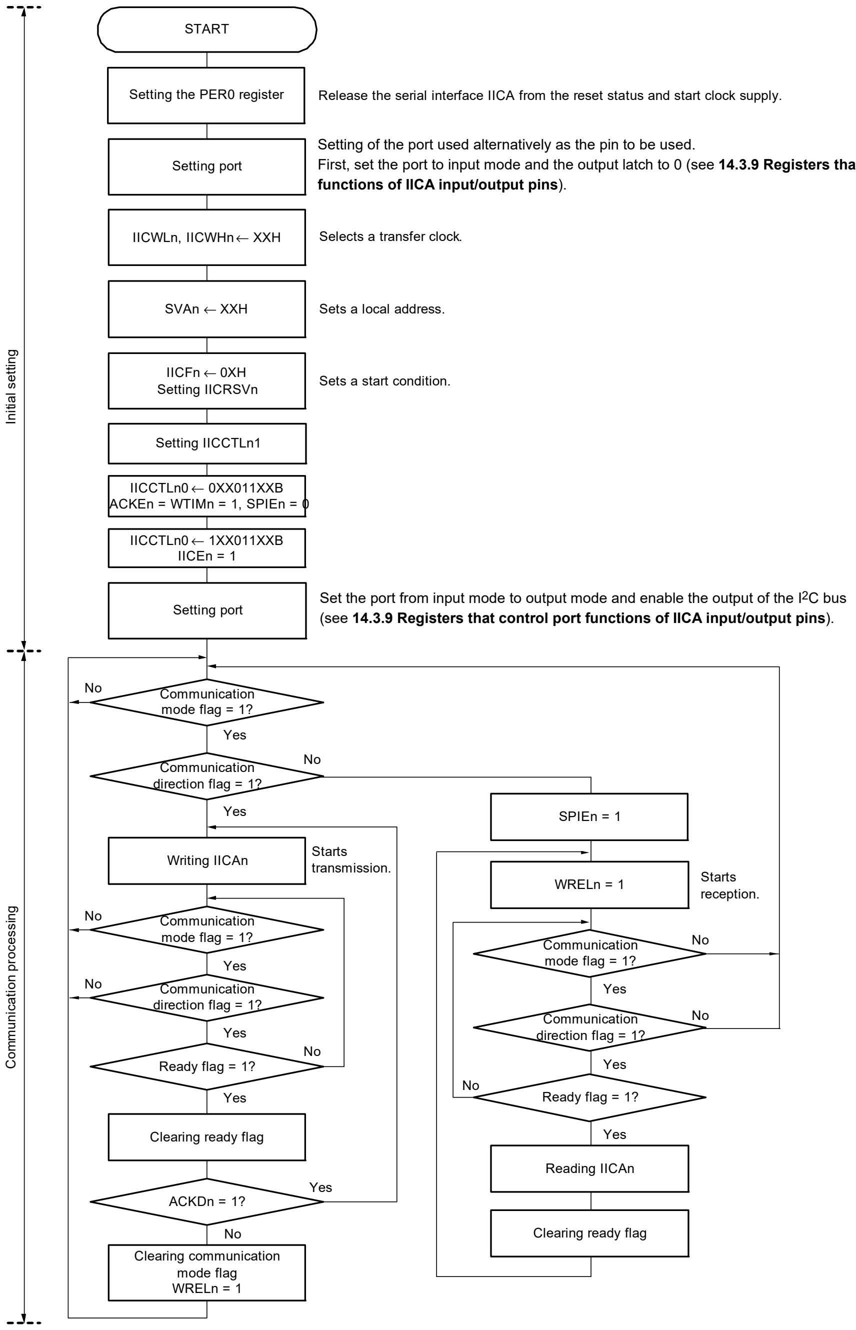
The main processing of the slave operation is explained next.

Start serial interface IICA and wait until communication is enabled. When communication is enabled, execute communication by using the communication mode flag and ready flag (processing of the stop condition and start condition is performed by an interrupt. Here, check the status by using the flags).

The transmission operation is repeated until the master no longer returns ACK. If ACK is not returned from the master, communication is completed.

For reception, the necessary amount of data is received. When communication is completed, ACK is not returned as the next data. After that, the master generates a stop condition or restart condition. Exit from the communication status occurs in this way.

Figure 14 - 30 Slave Operation Flowchart (1)



Release the serial interface IICA from the reset status and start clock supply.

Setting of the port used alternatively as the pin to be used.
First, set the port to input mode and the output latch to 0 (see **14.3.9 Registers that control port functions of IICA input/output pins**).

Selects a transfer clock.

Sets a local address.

Sets a start condition.

Set the port from input mode to output mode and enable the output of the I2C bus (see **14.3.9 Registers that control port functions of IICA input/output pins**).

Remark1. Conform to the specifications of the product that is in communication, regarding the transmission and reception formats.
Remark2. n = 0, 1

An example of the processing procedure of the slave with the INTIICAn interrupt is explained below (processing is performed assuming that no extension code is used). The INTIICAn interrupt checks the status, and the following operations are performed.

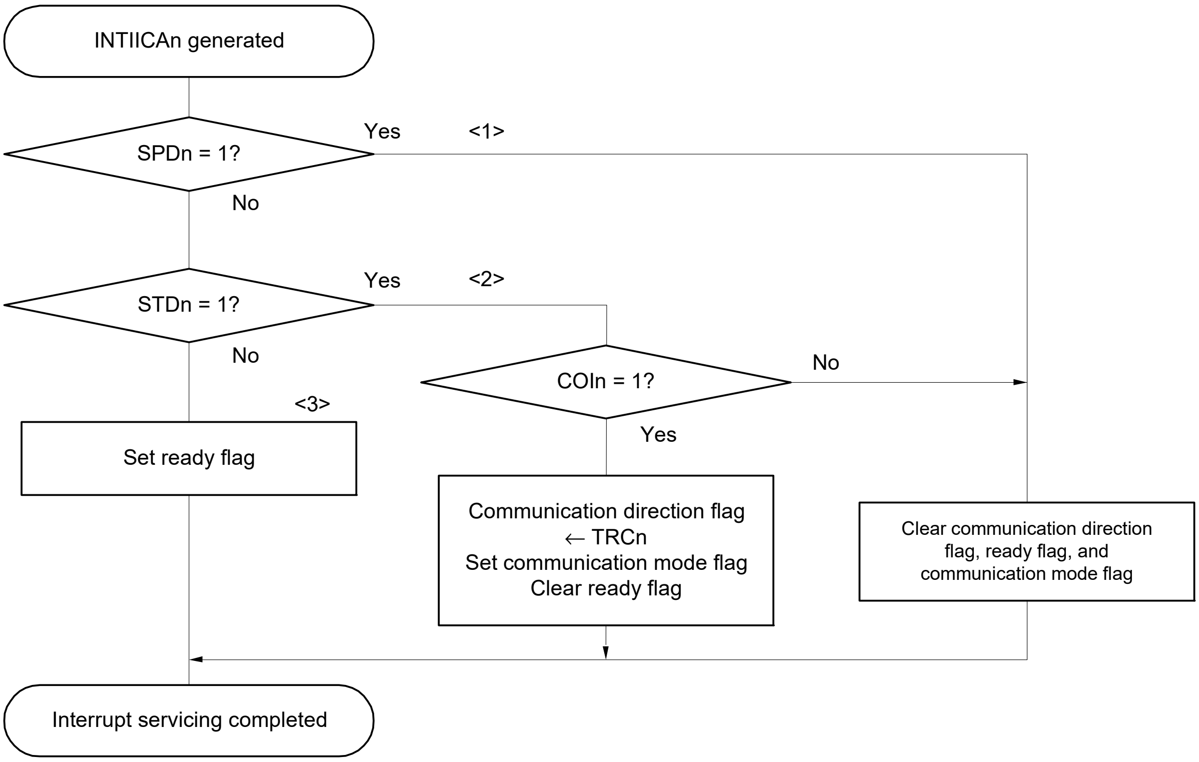
<1> Communication is stopped if the stop condition is issued.

<2> If the start condition is issued, the address is checked and communication is completed if the address does not match. If the address matches, the communication mode is set, wait is cancelled, and processing returns from the interrupt (the ready flag is cleared).

<3> For data transmit/receive, only the ready flag is set. Processing returns from the interrupt with the I^{2}C bus remaining in the wait state.

Remark <1> to <3> above correspond to <1> to <3> in Figure 14 - 31 Slave Operation Flowchart (2).

Figure 14 - 31 Slave Operation Flowchart (2)


INTIICAn generated
SPDn = 1?
Yes
<1>
No
STDn = 1?
Yes
<2>
No
COIn = 1?
No
Yes
<3>
Set ready flag
Communication direction flag ← TRCn
Set communication mode flag
Clear ready flag
Clear communication direction flag, ready flag, and communication mode flag
Interrupt servicing completed

Remark n = 0, 1

14.5.17 Timing of I^{2}C interrupt request (INTIICAn) occurrence

The timing of transmitting or receiving data and generation of interrupt request signal INTIICAn, and the value of the IICA status register n (IICSn) when the INTIICAn signal is generated are shown below.

Remark 1.

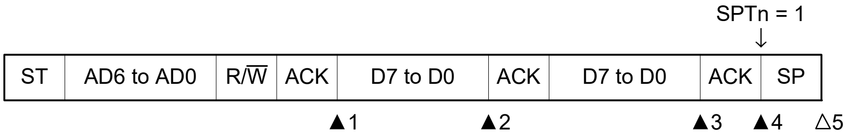
ST:	Start condition
AD6 to AD0:	Address
R/$\overline{W}$:	Transfer direction specification
ACK:	Acknowledge
D7 to D0:	Data
SP:	Stop condition

Remark 2. n = 0, 1

(1) Master device operation

(a) Start ~ Address ~ Data ~ Data ~ Stop (transmission/reception)

(i) When WTIMn = 0



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×000B

▲3: IICSn = 1000×000B (Sets the WTIMn bit to 1) Note

▲4: IICSn = 1000××00B (Sets the SPTn bit to 1)

△5: IICSn = 00000001B

Note To generate a stop condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

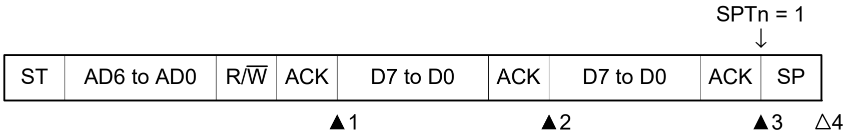
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don’t care

Remark n = 0, 1

(ii) When WTIMn = 1



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×100B

▲3: IICSn = 1000××00B (Sets the SPTn bit to 1)

△4: IICSn = 00000001B

Remark ▲: Always generated

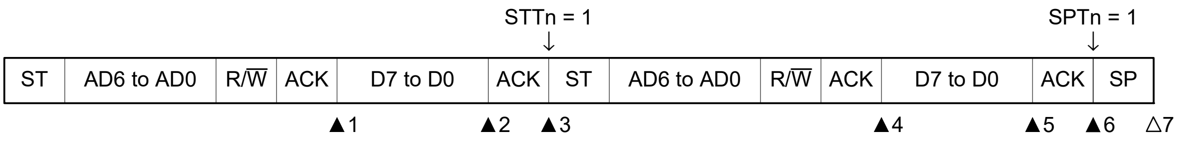
△: Generated only when SPIEn = 1

×: Don’t care

Remark n = 0, 1

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

(i) When WTIMn = 0



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×000B (Sets the WTIMn bit to 1) [Note 1]

▲3: IICSn = 1000××00B (Clears the WTIMn bit to 0 [Note 2], sets the STTn bit to 1)

▲4: IICSn = 1000×110B

▲5: IICSn = 1000×000B (Sets the WTIMn bit to 1) [Note 3]

▲6: IICSn = 1000××00B (Sets the SPTn bit to 1)

△7: IICSn = 00000001B

Note 1. To generate a start condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

Note 2. Clear the WTIMn bit to 0 to restore the original setting.

Note 3. To generate a stop condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

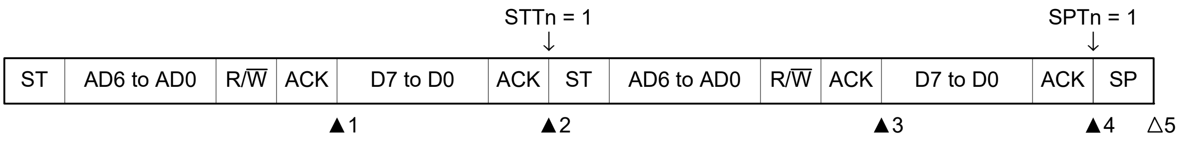
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0, 1

(ii) When WTIMn = 1



▲1: IICSn = 1000×110B

▲2: IICSn = 1000××00B (Sets the STTn bit to 1)

▲3: IICSn = 1000×110B

▲4: IICSn = 1000××00B (Sets the SPTn bit to 1)

△5: IICSn = 00000001B

Remark ▲: Always generated

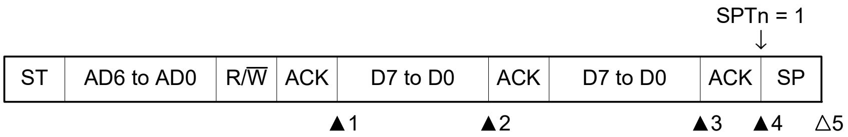
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0, 1

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

(i) When WTIMn = 0



▲1: IICSn = 1010×110B

▲2: IICSn = 1010×000B

▲3: IICSn = 1010×000B (Sets the WTIMn bit to 1) Note

▲4: IICSn = 1010××00B (Sets the SPTn bit to 1)

△5: IICSn = 00000001B

Note To generate a stop condition, set the WTIMn bit to 1 and change the timing for generating the INTIICAn interrupt request signal.

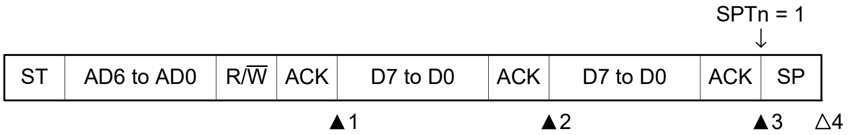
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0, 1

(ii) When WTIMn = 1



▲1: IICSn = 1010×110B

▲2: IICSn = 1010×100B

▲3: IICSn = 1010××00B (Sets the SPTn bit to 1)

△4: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

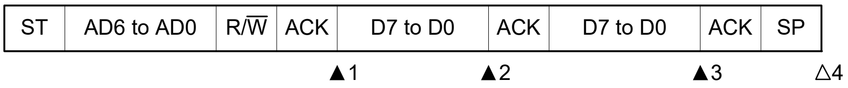
×: Don't care

Remark n = 0, 1

(2) Slave device operation (slave address data reception)

(a) Start ~ Address ~ Data ~ Data ~ Stop

(i) When WTIMn = 0



▲1: IICSn = 0001×110B

▲2: IICSn = 0001×000B

▲3: IICSn = 0001×000B

△4: IICSn = 00000001B

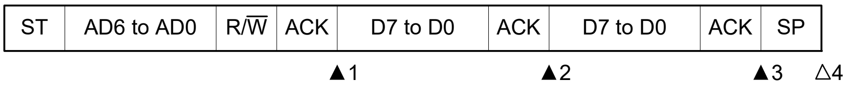
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0, 1

(ii) When WTIMn = 1



▲1: IICSn = 0001×110B

▲2: IICSn = 0001×100B

▲3: IICSn = 0001××00B

△4: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0, 1

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, matches with SVAn)

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	SP	
				▲1	▲2					▲3	▲4		△5

▲1: IICSn = 0001×110B

▲2: IICSn = 0001×000B

▲3: IICSn = 0001×110B

▲4: IICSn = 0001×000B

△5: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don’t care

Remark n = 0, 1

(ii) When WTIMn = 1 (after restart, matches with SVAn)

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	SP	
				▲1		▲2				▲3		▲4	△5

▲1: IICSn = 0001×110B

▲2: IICSn = 0001××00B

▲3: IICSn = 0001×110B

▲4: IICSn = 0001××00B

△5: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don’t care

Remark n = 0, 1

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, does not match address (= extension code))

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	SP
			▲1		▲2				▲3		▲4	△5

▲1: IICSn = 0001×110B

▲2: IICSn = 0001×000B

▲3: IICSn = 0010×010B

▲4: IICSn = 0010×000B

△5: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0, 1

(ii) When WTIMn = 1 (after restart, does not match address (= extension code))

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	SP
			▲1		▲2				▲3	▲4	▲5	△6

▲1: IICSn = 0001×110B

▲2: IICSn = 0001××00B

▲3: IICSn = 0010×010B

▲4: IICSn = 0010×110B

▲5: IICSn = 0010××00B

△6: IICSn = 00000001B

Remark ▲: Always generated

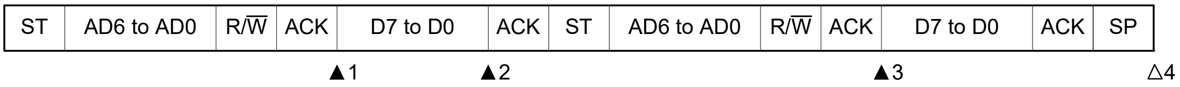
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0, 1

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, does not match address (= not extension code))



▲1: IICSn = 0001×110B

▲2: IICSn = 0001×000B

▲3: IICSn = 00000110B

△4: IICSn = 00000001B

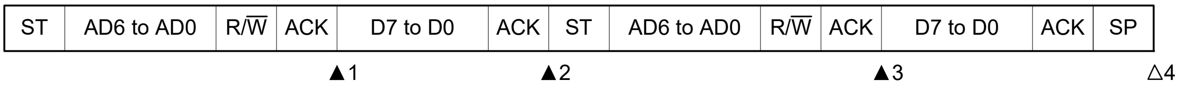
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0, 1

(ii) When WTIMn = 1 (after restart, does not match address (= not extension code))



▲1: IICSn = 0001×110B

▲2: IICSn = 0001××00B

▲3: IICSn = 00000110B

△4: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

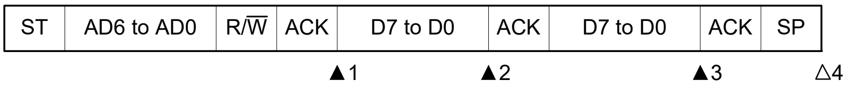
Remark n = 0, 1

(3) Slave device operation (when receiving extension code)

The device is always participating in communication when it receives an extension code.

(a) Start ~ Code ~ Data ~ Data ~ Stop

(i) When WTIMn = 0



▲1: IICSn = 0010×010B

▲2: IICSn = 0010×000B

▲3: IICSn = 0010×000B

△4: IICSn = 00000001B

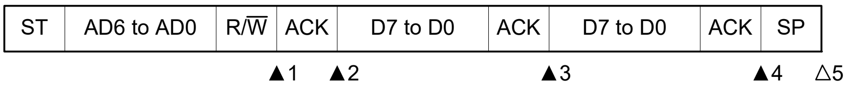
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don’t care

Remark n = 0, 1

(ii) When WTIMn = 1



▲1: IICSn = 0010×010B

▲2: IICSn = 0010×110B

▲3: IICSn = 0010×100B

▲4: IICSn = 0010××00B

△5: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don’t care

Remark n = 0, 1

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, matches SVAn)

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	SP	
			▲1		▲2					▲3	▲4		△5

▲1: IICSn = 0010×010B

▲2: IICSn = 0010×000B

▲3: IICSn = 0001×110B

▲4: IICSn = 0001×000B

△5: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0, 1

(ii) When WTIMn = 1 (after restart, matches SVAn)

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	SP	
			▲1	▲2		▲3				▲4		▲5	△6

▲1: IICSn = 0010×010B

▲2: IICSn = 0010×110B

▲3: IICSn = 0010××00B

▲4: IICSn = 0001×110B

▲5: IICSn = 0001××00B

△6: IICSn = 00000001B

Remark ▲: Always generated

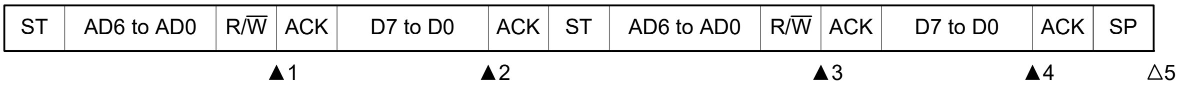
△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0, 1

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, extension code reception)



▲1: IICSn = 0010×010B

▲2: IICSn = 0010×000B

▲3: IICSn = 0010×010B

▲4: IICSn = 0010×000B

△5: IICSn = 00000001B

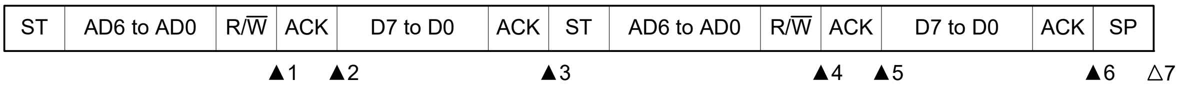
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don’t care

Remark n = 0, 1

(ii) When WTIMn = 1 (after restart, extension code reception)



▲1: IICSn = 0010×010B

▲2: IICSn = 0010×110B

▲3: IICSn = 0010××00B

▲4: IICSn = 0010×010B

▲5: IICSn = 0010×110B

▲6: IICSn = 0010××00B

△7: IICSn = 00000001B

Remark ▲: Always generated

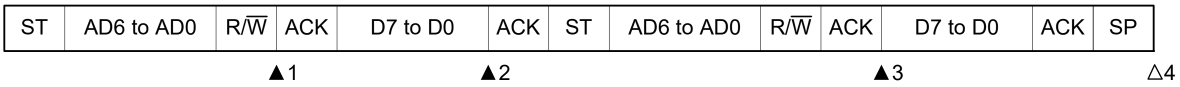
△: Generated only when SPIEn = 1

×: Don’t care

Remark n = 0, 1

(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIMn = 0 (after restart, does not match address (= not extension code))



▲1: IICSn = 0010 × 010B

▲2: IICSn = 0010 × 000B

▲3: IICSn = 00000 × 10B

△4: IICSn = 00000001B

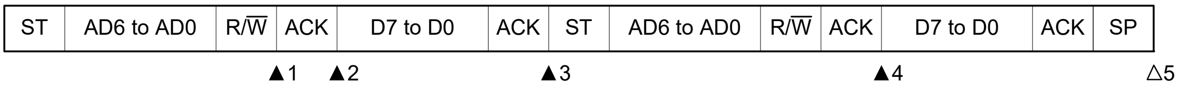
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don’t care

Remark n = 0, 1

(ii) When WTIMn = 1 (after restart, does not match address (= not extension code))



▲1: IICSn = 0010 × 010B

▲2: IICSn = 0010 × 110B

▲3: IICSn = 0010 × × 00B

▲4: IICSn = 00000 × 10B

△5: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don’t care

Remark n = 0, 1

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop

ST	AD6 to AD0	$R/\overline{W}$	ACK	D7 to D0	ACK	D7 to D0	ACK	SP	
									△1

△1: IICSn = 00000001B

Remark △: Generated only when SPIEn = 1

Remark n = 0, 1

(5) Arbitration loss operation (operation as slave after arbitration loss)

When the device is used as a master in a multi-master system, read the MSTSn bit each time interrupt request signal INTIICAn has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data

(i) When WTIMn = 0

ST	AD6 to AD0	$R/\overline{W}$	ACK	D7 to D0	ACK	D7 to D0	ACK	SP	
			▲1	▲2		▲3			△4

▲1: IICSn = 0101×110B

▲2: IICSn = 0001×000B

▲3: IICSn = 0001×000B

△4: IICSn = 00000001B

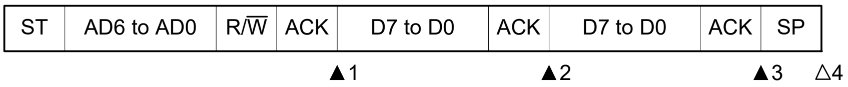
Remark ▲: Always generated

△: Generated only when SPIEn = 1

× : Don’t care

Remark n = 0, 1

(ii) When WTIMn = 1



▲1: IICSn = 0101×110B

▲2: IICSn = 0001×100B

▲3: IICSn = 0001××00B

△4: IICSn = 00000001B

Remark ▲: Always generated

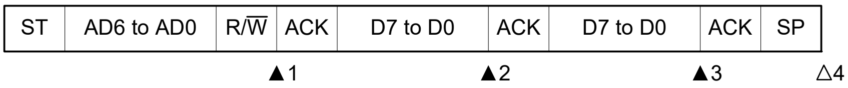
△: Generated only when SPIEn = 1

×: Don’t care

Remark n = 0, 1

(b) When arbitration loss occurs during transmission of extension code

(i) When WTIMn = 0



▲1: IICSn = 0110×010B

▲2: IICSn = 0010×000B

▲3: IICSn = 0010×000B

△4: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don’t care

Remark n = 0, 1

(ii) When WTIMn = 1

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	D7 to D0	ACK	SP
		▲1	▲2		▲3		▲4	△5

▲1: IICSn = 0110×010B

▲2: IICSn = 0010×110B

▲3: IICSn = 0010×100B

▲4: IICSn = 0010××00B

△5: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

Remark n = 0, 1

(6) Operation when arbitration loss occurs (no communication after arbitration loss)

When the device is used as a master in a multi-master system, read the MSTSn bit each time interrupt request signal INTIICAn has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data (when WTIMn = 1)

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	D7 to D0	ACK	SP
			▲1					△2

▲1: IICSn = 01000110B

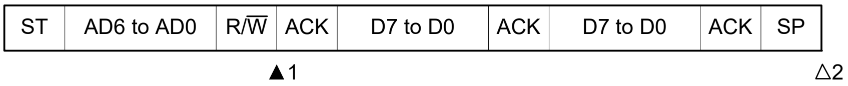
△2: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

Remark n = 0, 1

(b) When arbitration loss occurs during transmission of extension code



▲1: IICSn = 0110×010B

Sets LRELn = 1 by software

△2: IICSn = 00000001B

Remark ▲: Always generated

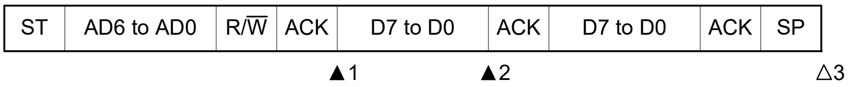
△: Generated only when SPIEn = 1

×: Don’t care

Remark n = 0, 1

(c) When arbitration loss occurs during transmission of data

(i) When WTIMn = 0



▲1: IICSn = 10001110B

▲2: IICSn = 01000000B

△3: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

Remark n = 0, 1

(ii) When WTIMn = 1

ST	AD6 to AD0	$R/\overline{W}$	ACK	D7 to D0	ACK	D7 to D0	ACK	SP	
			▲1		▲2				△3

▲1: IICSn = 10001110B

▲2: IICSn = 01000100B

△3: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

Remark n = 0, 1

(d) When loss occurs due to restart condition during data transfer

(i) Not extension code (Example: unmatches with SVAn)

ST	AD6 to AD0	$R/\overline{W}$	ACK	D7 to Dm	ST	AD6 to AD0	$R/\overline{W}$	ACK	D7 to D0	ACK	SP	
			▲1					▲2				△3

▲1: IICSn = 1000×110B

▲2: IICSn = 01000110B

△3: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

m = 6 to 0

Remark n = 0, 1

(ii) Extension code

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to Dm	ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to D0	ACK	SP
			▲1					▲2			△3

▲1: IICSn = 1000×110B

▲2: IICSn = 01100010B

Sets LRELn = 1 by software

△3: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don't care

m = 6 to 0

Remark n = 0, 1

(e) When loss occurs due to stop condition during data transfer

ST	AD6 to AD0	R/$\overline{W}$	ACK	D7 to Dm	SP
			▲1		△2

▲1: IICSn = 10000110B

△2: IICSn = 01000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

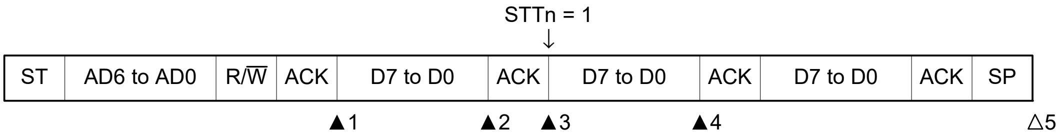
×: Don't care

m = 6 to 0

Remark n = 0, 1

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

(i) When WTIMn = 0



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×000B (Sets the WTIMn bit to 1)

▲3: IICSn = 1000×100B (Clears the WTIMn bit to 0)

▲4: IICSn = 01000000B

△5: IICSn = 00000001B

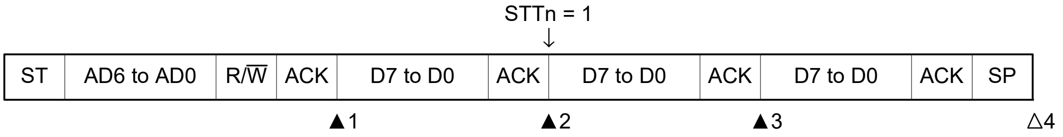
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don’t care

Remark n = 0, 1

(ii) When WTIMn = 1



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×100B (Sets the STTn bit to 1)

▲3: IICSn = 01000100B

△4: IICSn = 00000001B

Remark ▲: Always generated

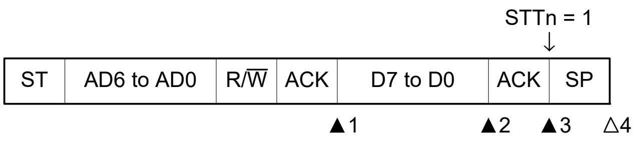
△: Generated only when SPIEn = 1

×: Don’t care

Remark n = 0, 1

(g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

(i) When WTIMn = 0



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×000B (Sets the WTIMn bit to 1)

▲3: IICSn = 1000××00B (Sets the STTn bit to 1)

△4: IICSn = 01000001B

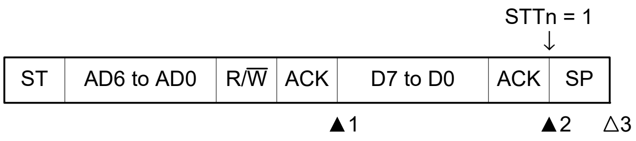
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don’t care

Remark n = 0, 1

(ii) When WTIMn = 1



▲1: IICSn = 1000×110B

▲2: IICSn = 1000××00B (Sets the STTn bit to 1)

△3: IICSn = 01000001B

Remark ▲: Always generated

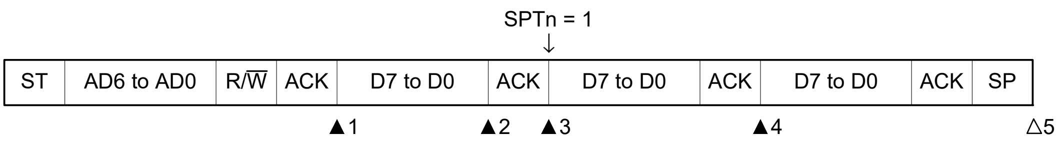
△: Generated only when SPIEn = 1

×: Don’t care

Remark n = 0, 1

(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

(i) When WTIMn = 0



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×000B (Sets the WTIMn bit to 1)

▲3: IICSn = 1000×100B (Clears the WTIMn bit to 0)

▲4: IICSn = 01000100B

△5: IICSn = 00000001B

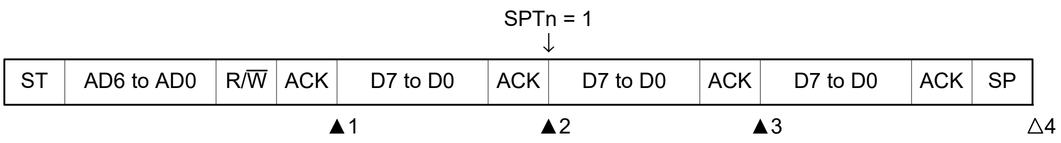
Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don’t care

Remark n = 0, 1

(ii) When WTIMn = 1



▲1: IICSn = 1000×110B

▲2: IICSn = 1000×100B (Sets the SPTn bit to 1)

▲3: IICSn = 01000100B

△4: IICSn = 00000001B

Remark ▲: Always generated

△: Generated only when SPIEn = 1

×: Don’t care

Remark n = 0, 1

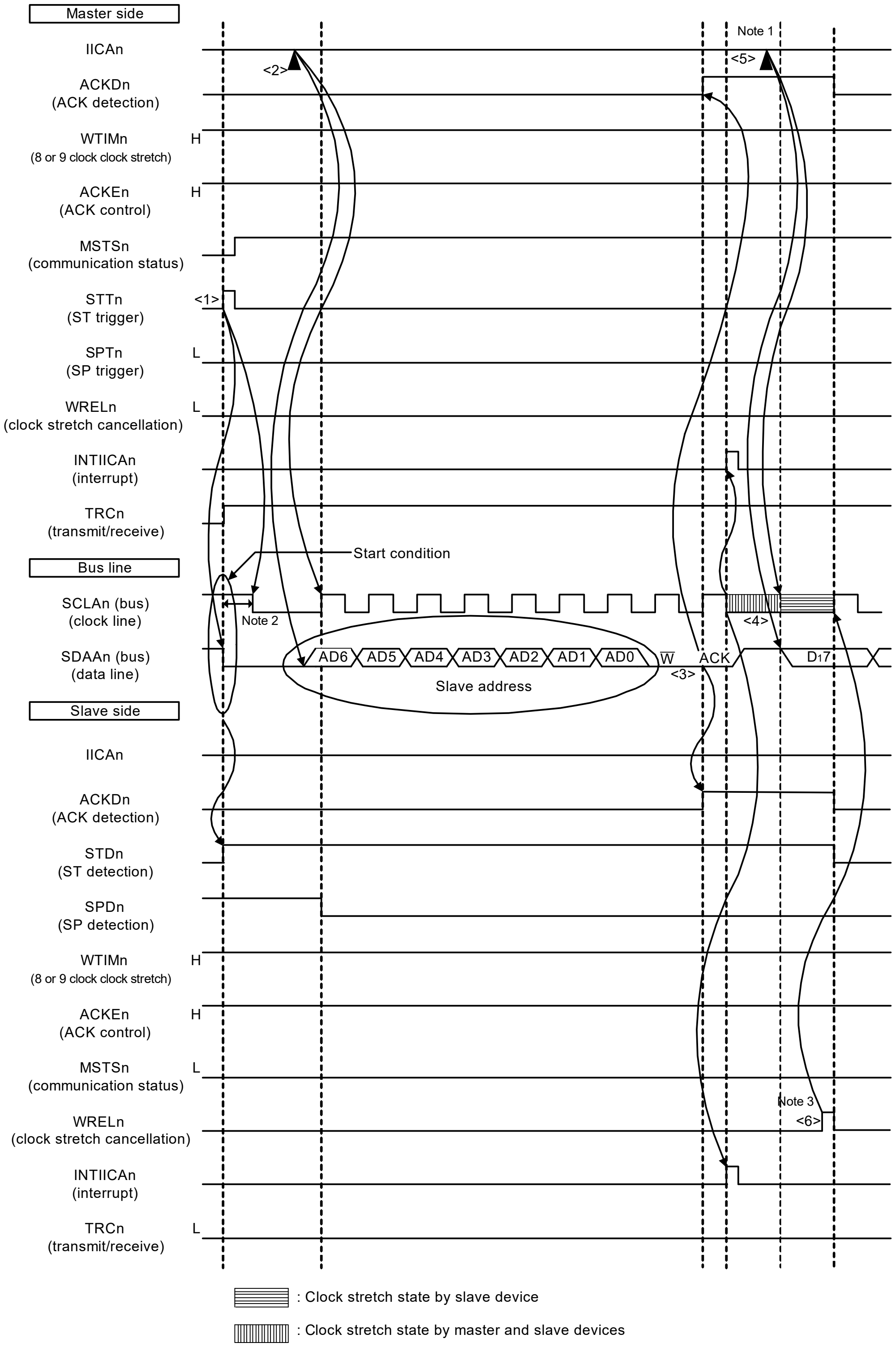
14.6 Timing Charts

When using the I^{2}C bus mode, the master device outputs an address via the serial bus to select one of several slave devices as its communication partner.
After outputting the slave address, the master device transmits the TRCn bit (bit 3 of the IICA status register n (IICSn)), which specifies the data transfer direction, and then starts serial communication with the slave device.
Figures 14 - 32 to 14 - 33 show timing charts of the data communication.
The IICA shift register n (IICAn)'s shift operation is synchronized with the falling edge of the serial clock (SCLAn).
The transmit data is transferred to the SO latch and is output (MSB first) via the SDAAn pin.
Data input via the SDAAn pin is captured into IICAn at the rising edge of SCLAn.

Remark n = 0, 1

Figure 14 - 32 Example of Master to Slave Communication (When 9-Clock Clock Stretch Is Selected for Master, 9-Clock Clock Stretch Is Selected for Slave) (1/4)

(1) Start condition ~ address ~ data



Note 1. Write data to IICAn, not setting the WRELn bit, in order to cancel a clock stretch state during transmission by a master device.

Note 2. Make sure that the time between the fall of the SDAAn pin signal and the fall of the SCLAn pin signal is at least 4.0 µs when specifying standard mode and at least 0.6 µs when specifying fast mode.

Note 3. For releasing clock stretch state during reception of a slave device, write “FFH” to IICAn or set the WRELn bit.

Remark n = 0, 1

The meanings of <1> to <6> in (1) Start condition ~ address ~ data in Figure 14 - 32 are explained below.

<1> The start condition trigger is set by the master device (STTn = 1) and a start condition (i.e. SCLAn = 1 changes SDAAn from 1 to 0) is generated once the bus data line goes low (SDAAn). When the start condition is subsequently detected, the master device enters the master device communication status (MSTSn = 1). The master device is ready to communicate once the bus clock line goes low (SCLAn = 0) after the hold time has elapsed.

<2> The master device writes the address + W (transmission) to the IICA shift register n (IICAn) and transmits the slave address.

<3> In the slave device if the address received matches the address (SVAn value) of a slave device Note, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a clock stretch status (SCLAn = 0) and issues an interrupt (INTIICAn: address match) Note.

<5> The master device writes the data to transmit to the IICAn register and releases the clock stretch status that it set by the master device.

<6> If the slave device releases the clock stretch status (WRELn = 1), the master device starts transferring data to the slave device.

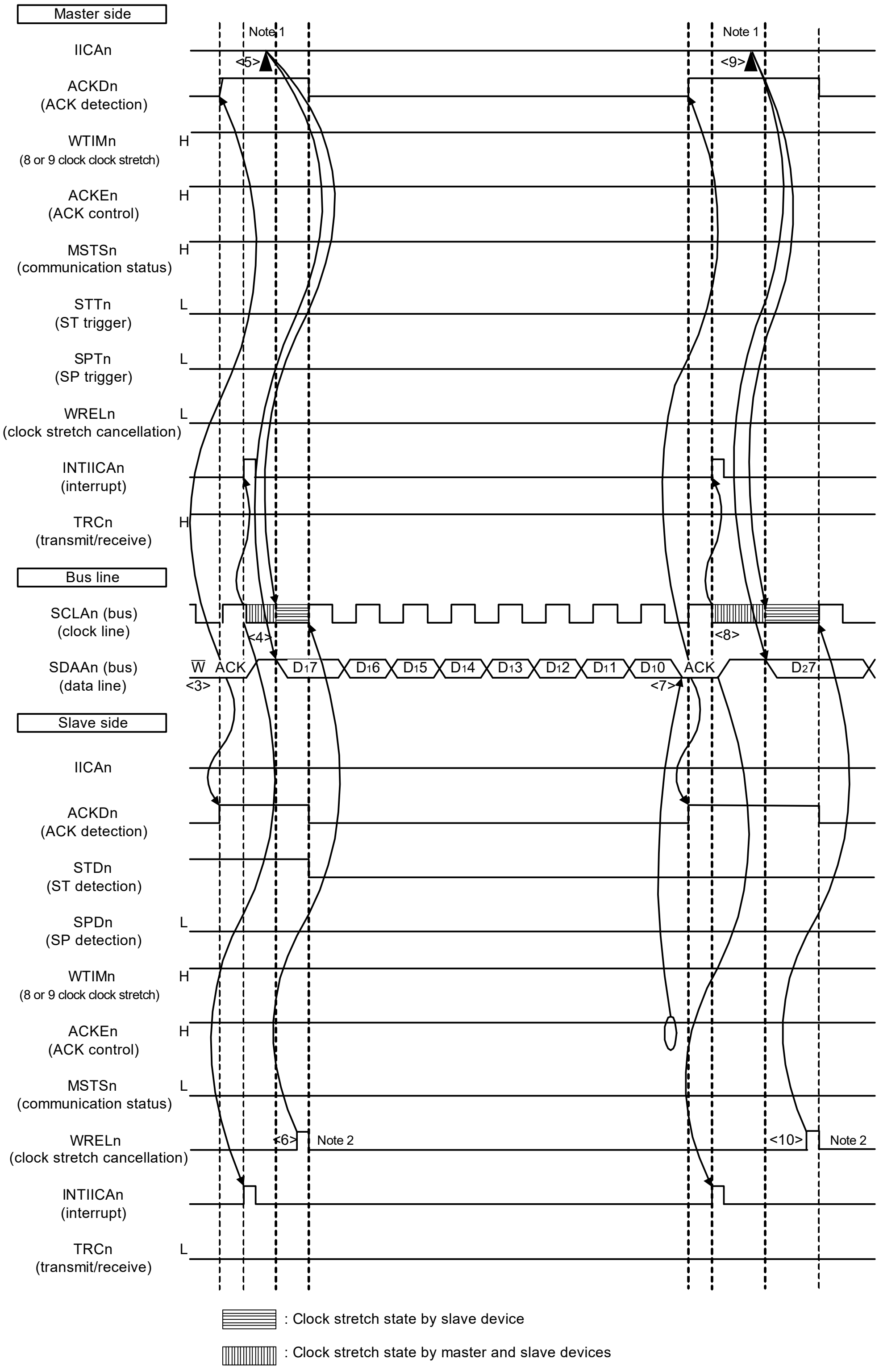
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set a clock stretch status. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <15> in Figures 14 - 32 (1/4) to 14 - 32 (4/4) represent the entire procedure for communicating data using the I^{2}C bus. Figure 14 - 32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 14 - 32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 14 - 32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Remark 2. n = 0, 1

Figure 14 - 32 Example of Master to Slave Communication (When 9-Clock Clock Stretch Is Selected for Master, 9-Clock Clock Stretch Is Selected for Slave) (2/4)

(2) Address ~ data ~ data



▤ : Clock stretch state by slave device

▥ : Clock stretch state by master and slave devices

Note 1. Write data to IICAn, not setting the WRELn bit, in order to cancel a clock stretch state during transmission by a master device.

Note 2. For releasing clock stretch state during reception of a slave device, write “FFH” to IICAn or set the WRELn bit.

Remark n = 0, 1

The meanings of <3> to <10> in (2) Address ~ data ~ data in Figure 14 - 32 are explained below.

<3> In the slave device if the address received matches the address (SVAn value) of a slave device Note, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a clock stretch status (SCLAn = 0) and issues an interrupt (INTIICAn: address match) Note.

<5> The master device writes the data to transmit to the IICA shift register n (IICAn) and releases the clock stretch status that it set by the master device.

<6> If the slave device releases the clock stretch status (WRELn = 1), the master device starts transferring data to the slave device.

<7> After data transfer is completed, because of ACKEn = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a clock stretch status (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).

<9> The master device writes the data to transmit to the IICAn register and releases the clock stretch status that it set by the master device.

<10>The slave device reads the received data and releases the clock stretch status (WRELn = 1). The master device then starts transferring data to the slave device.

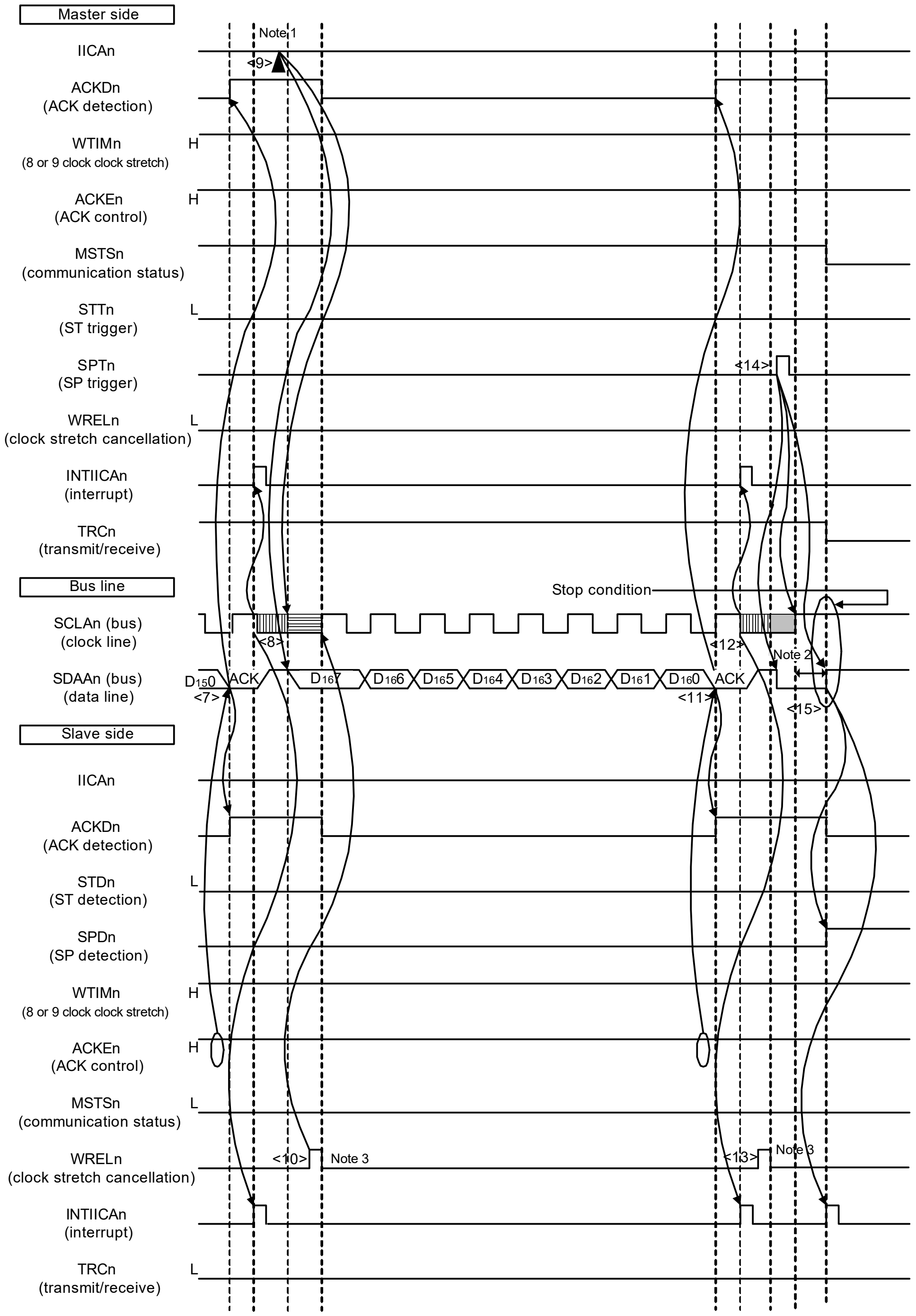
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set a clock stretch status. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <15> in Figures 14 - 32 (1/4) to Figures 14 - 32 (4/4) represent the entire procedure for communicating data using the I^{2}C bus. Figure 14 - 32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 14 - 32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 14 - 32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Remark 2. n = 0, 1

Figure 14 - 32 Example of Master to Slave Communication (When 9-Clock Clock Stretch Is Selected for Master, 9-Clock Clock Stretch Is Selected for Slave) (3/4)

(3) Data ~ data ~ stop condition



▒ : Clock stretch state by master device

☰ : Clock stretch state by slave device

▥ : Clock stretch state by master and slave devices

Note 1. Write data to IICAn, not setting the WRELn bit, in order to cancel a clock stretch state during transmission by a master device.

Note 2. Make sure that the time between the rise of the SCLAn pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0 µs when specifying standard mode and at least 0.6 µs when specifying fast mode.

Note 3. For releasing clock stretch state during reception of a slave device, write “FFH” to IICAn or set the WRELn bit.

Remark n = 0, 1

The meanings of <7> to <15> in (3) Data ~ data ~ stop condition in Figure 14 - 32 are explained below.

<7> After data transfer is completed, because of ACKEn = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a clock stretch status (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).

<9> The master device writes the data to transmit to the IICA shift register n (IICAn) and releases the clock stretch status that it set by the master device.

<10>The slave device reads the received data and releases the clock stretch status (WRELn = 1). The master device then starts transferring data to the slave device.

<11>When data transfer is complete, the slave device (ACKEn =1) sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<12>The master device and slave device set a clock stretch status (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).

<13>The slave device reads the received data and releases the clock stretch status (WRELn = 1).

<14> By the master device setting a stop condition trigger (SPTn = 1), the bus data line is cleared (SDAAn = 0) and the bus clock line is set (SCLAn = 1). After the stop condition setup time has elapsed, by setting the bus data line (SDAAn = 1), the stop condition is then generated (i.e. SCLAn =1 changes SDAAn from 0 to 1).

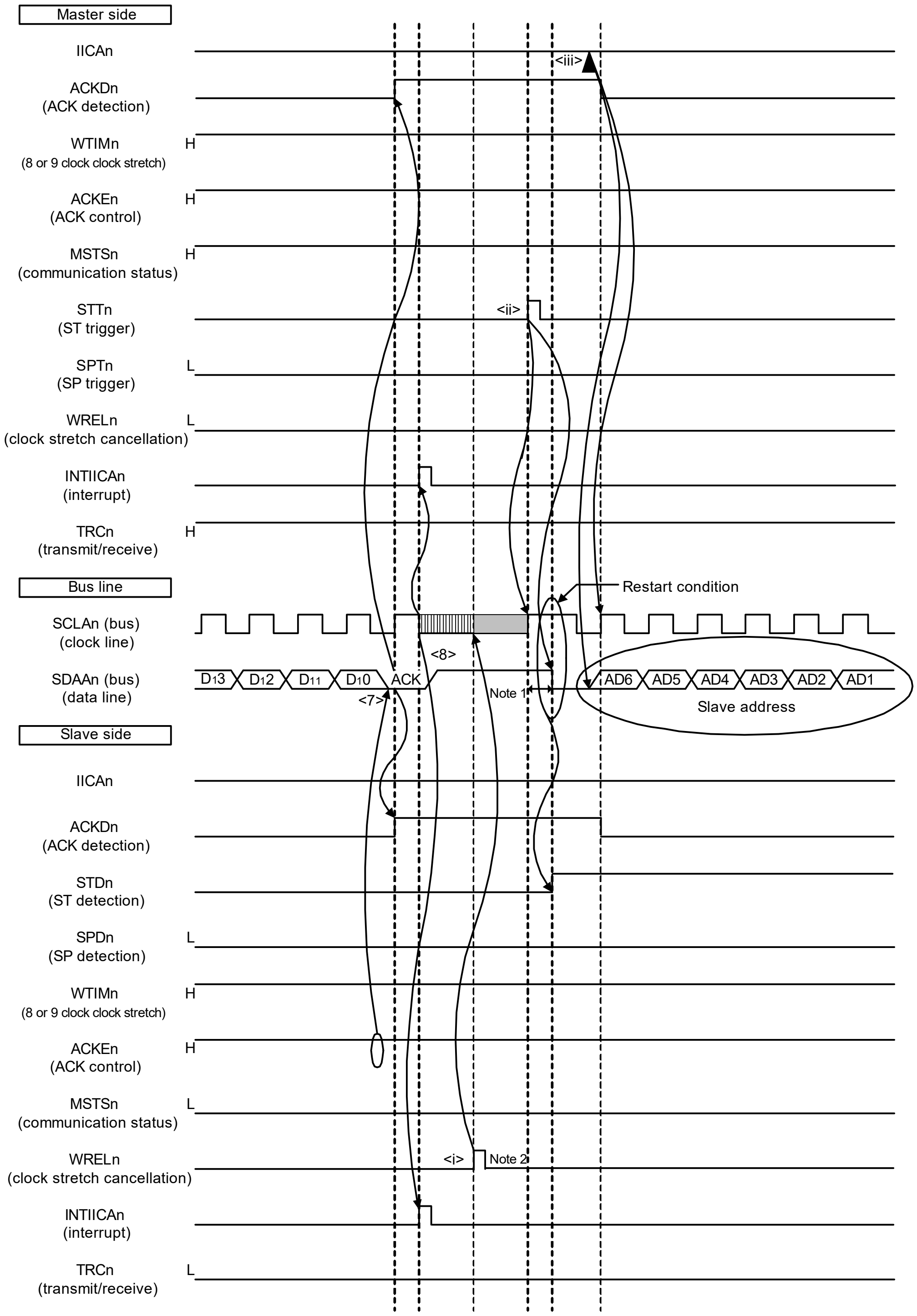
<15> When a stop condition is generated, the slave device detects the stop condition and issues an interrupt (INTIICAn: stop condition).

Remark 1. <1> to <15> in Figures 14 - 32 represent the entire procedure for communicating data using the I^{2}C bus. Figure 14 - 32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 14 - 32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 14 - 32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Remark 2. n = 0, 1

Figure 14 - 32 Example of Master to Slave Communication (When 9-Clock Clock Stretch Is Selected for Master, 9-Clock Clock Stretch Is Selected for Slave) (4/4)

(4) Data ~ restart condition ~ address



■ : Clock stretch state by master device

▤ : Clock stretch state by slave device

▥ : Clock stretch state by master and slave devices

Note 1. Make sure that the time between the rise of the SCLAn pin signal and the generation of the start condition after a restart condition has been issued is at least 4.7 µs when specifying standard mode and at least 0.6 µs when specifying fast mode.

Note 2. For releasing clock stretch state during reception of a slave device, write “FFH” to IICAn or set the WRELn bit.

Remark n = 0, 1

The following describes the operations in Figure 14 - 32 (4) Data ~ restart condition ~ address. After the operations in steps <7> and <8>, the operations in steps <i> to <iii> are performed. These steps return the processing to step <iii>, the data transmission step.

<7> After data transfer is completed, because of ACKEn = 1, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a clock stretch status (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).

<i> The slave device reads the received data and releases the clock stretch status (WRELn = 1).

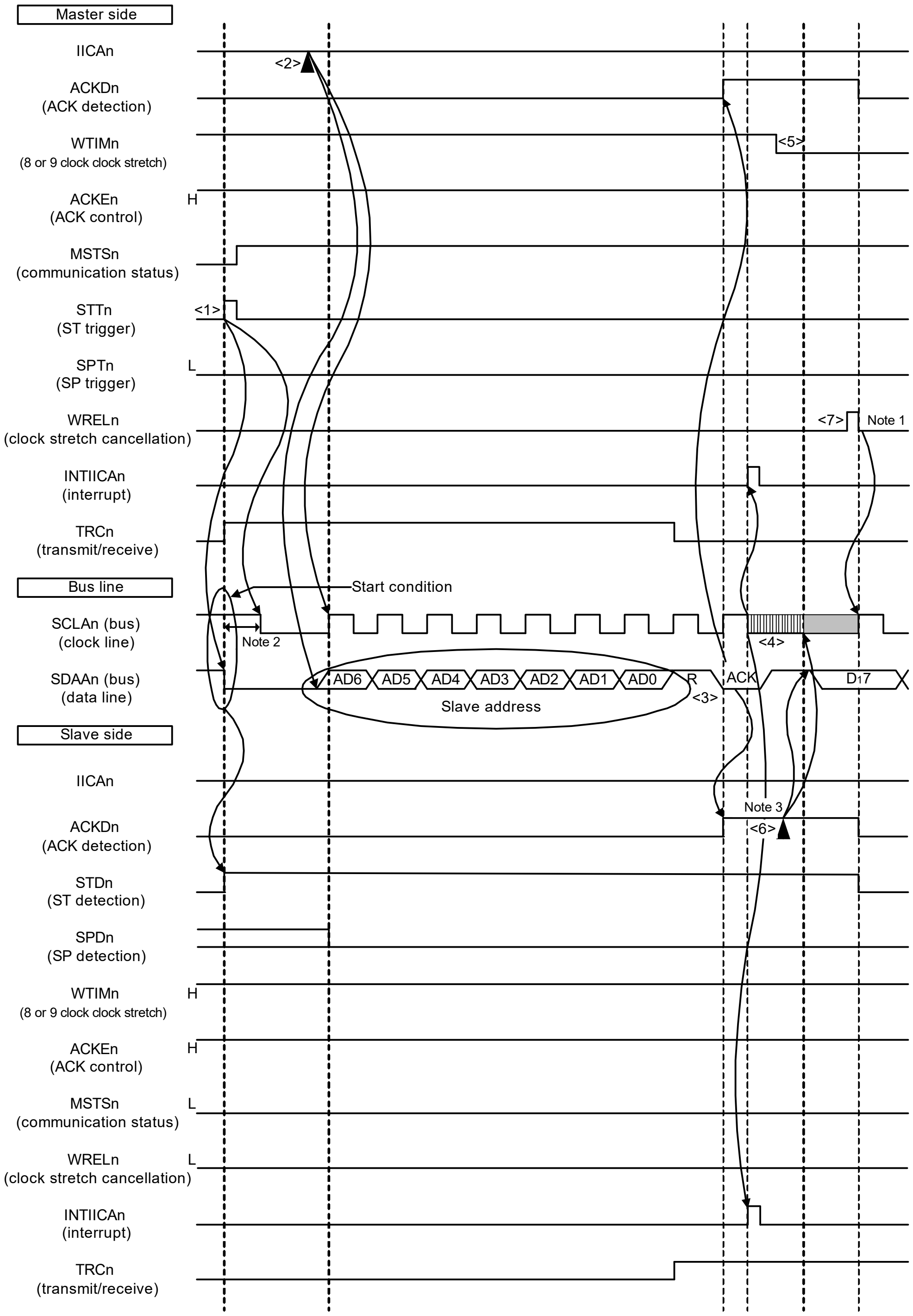
<ii> The start condition trigger is set again by the master device (STTn = 1) and a start condition (i.e. SCLAn =1 changes SDAAn from 1 to 0) is generated once the bus clock line goes high (SCLAn = 1) and the bus data line goes low (SDAAn = 0) after the restart condition setup time has elapsed. When the start condition is subsequently detected, the master device is ready to communicate once the bus clock line goes low (SCLAn = 0) after the hold time has elapsed.

<iii> The master device writing the address + R/W (transmission) to the IICA shift register (IICAn) enables the slave address to be transmitted.

Remark n = 0, 1

Figure 14 - 33 Example of Slave to Master Communication
(When 8-Clock Clock Stretch Is Selected for Master, 9-Clock Clock Stretch Is Selected for Slave) (1/3)

(1) Start condition ~ address ~ data



▒ : Clock stretch state by master device
☰ : Clock stretch state by slave device
▥ : Clock stretch state by master and slave devices

Note 1. For releasing clock stretch state during reception of a master device, write “FFH” to IICAn or set the WRELn bit.

Note 2. Make sure that the time between the fall of the SDAAn pin signal and the fall of the SCLAn pin signal is at least 4.0 µs when specifying standard mode and at least 0.6 µs when specifying fast mode.

Note 3. Write data to IICAn, not setting the WRELn bit, in order to cancel a clock stretch state during transmission by a slave device.

Remark n = 0, 1

The meanings of <1> to <7> in (1) Start condition ~ address ~ data in Figure 14 - 33 are explained below.

<1> The start condition trigger is set by the master device (STTn = 1) and a start condition (i.e. SCLAn =1 changes SDAAn from 1 to 0) is generated once the bus data line goes low (SDAAn). When the start condition is subsequently detected, the master device enters the master device communication status (MSTSn = 1). The master device is ready to communicate once the bus clock line goes low (SCLAn = 0) after the hold time has elapsed.

<2> The master device writes the address + R (reception) to the IICA shift register n (IICAn) and transmits the slave address.

<3> In the slave device if the address received matches the address (SVAn value) of a slave device [Note], that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a clock stretch status (SCLAn = 0) and issues an interrupt (INTIICAn: address match) [Note].

<5> The timing at which the master device sets the clock stretch status changes to the 8th clock (WTIMn = 0).

<6> The slave device writes the data to transmit to the IICAn register and releases the clock stretch status that it set by the slave device.

<7> The master device releases the clock stretch status (WRELn = 1) and starts transferring data from the slave device to the master device.

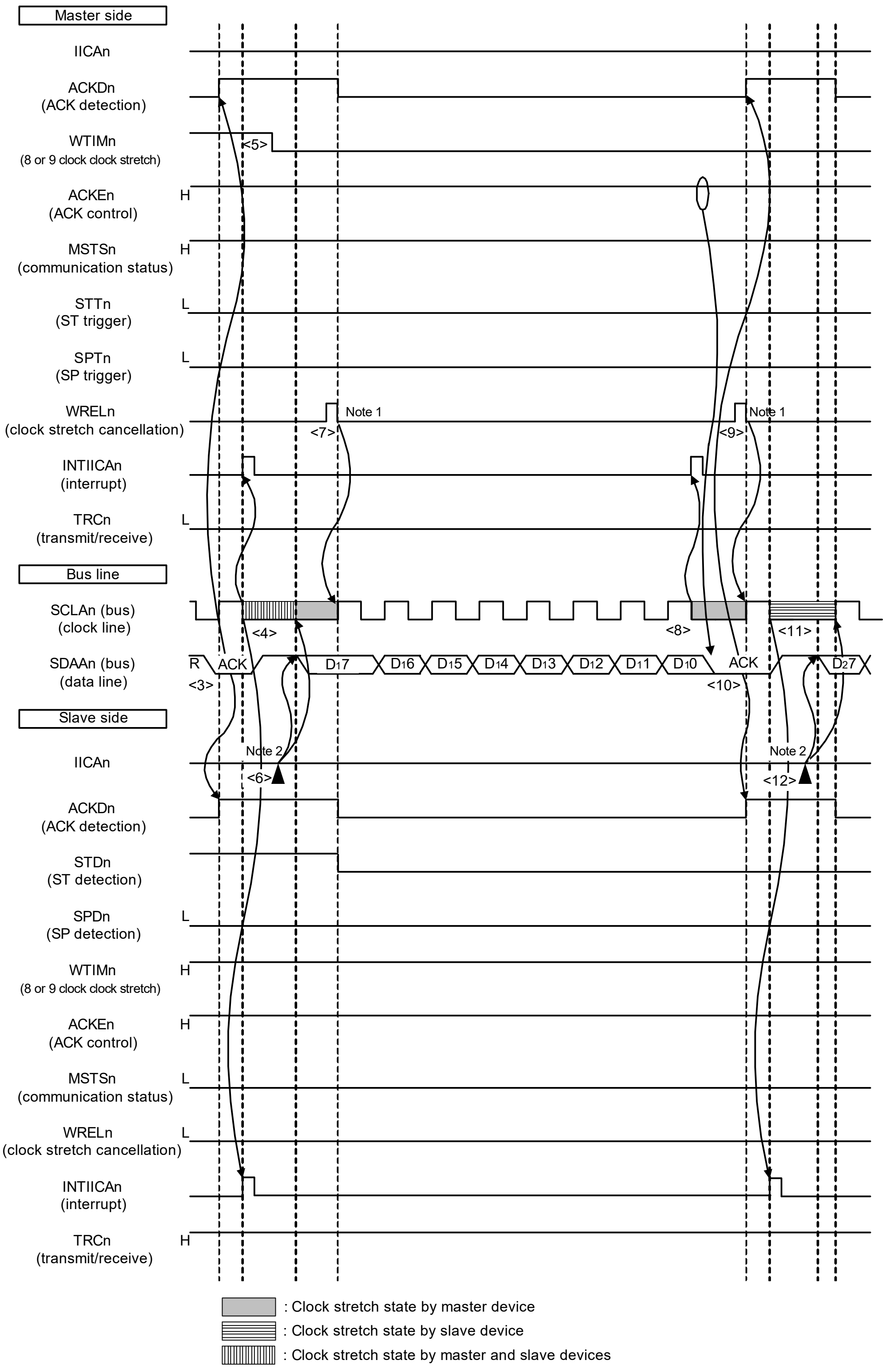
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set a clock stretch status. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <19> in Figures 14 - 33 represent the entire procedure for communicating data using the I^{2}C bus. Figure 14 - 33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 14 - 33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 14 - 33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Remark 2. n = 0, 1

Figure 14 - 33 Example of Slave to Master Communication (When 8-Clock Clock Stretch Is Selected for Master, 9-Clock Clock Stretch Is Selected for Slave) (2/3)

(2) Address ~ data ~ data



Note 1. For releasing clock stretch state during reception of a master device, write “FFH” to IICAn or set the WRELn bit.

Note 2. Write data to IICAn, not setting the WRELn bit, in order to cancel a clock stretch state during transmission by a slave device.

Remark n = 0, 1

The meanings of <3> to <12> in (2) Address ~ data ~ data in Figure 14 - 33 are explained below.

<3> In the slave device if the address received matches the address (SVAn value) of a slave device Note, that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKDn = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICAn: end of address transmission) at the falling edge of the 9th clock. The slave device whose address matched the transmitted slave address sets a clock stretch status (SCLAn = 0) and issues an interrupt (INTIICAn: address match) Note.

<5> The master device changes the timing of the clock stretch status to the 8th clock (WTIMn = 0).

<6> The slave device writes the data to transmit to the IICA shift register n (IICAn) and releases the clock stretch status that it set by the slave device.

<7> The master device releases the clock stretch status (WRELn = 1) and starts transferring data from the slave device to the master device.

<8> The master device sets a clock stretch status (SCLAn = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICAn: end of transfer). Because of ACKEn = 1 in the master device, the master device then sends an ACK by hardware to the slave device.

<9> The master device reads the received data and releases the clock stretch status (WRELn = 1).

<10>The ACK is detected by the slave device (ACKDn = 1) at the rising edge of the 9th clock.

<11>The slave device set a clock stretch status (SCLAn = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICAn: end of transfer).

<12>By the slave device writing the data to transmit to the IICAn register, the clock stretch status set by the slave device is released. The slave device then starts transferring data to the master device.

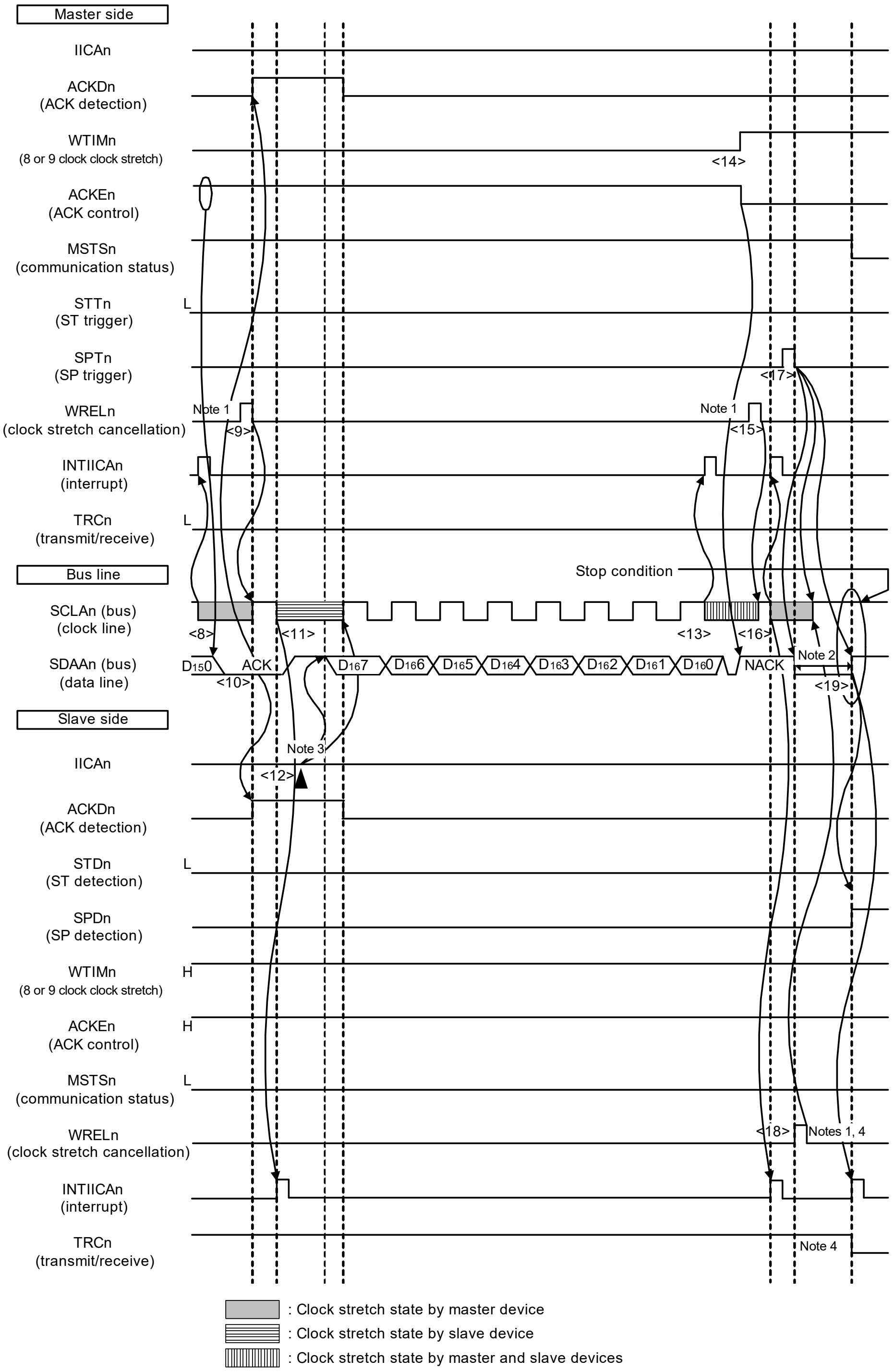
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDAAn = 1). The slave device also does not issue the INTIICAn interrupt (address match) and does not set a clock stretch status. The master device, however, issues the INTIICAn interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark 1. <1> to <19> in Figures 14 - 33 represent the entire procedure for communicating data using the I^{2}C bus. Figure 14 - 33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 14 - 33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 14 - 33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Remark 2. n = 0, 1

Figure 14 - 33 Example of Slave to Master Communication (When 8-Clock and 9-Clock Clock Stretch Is Selected for Master, 9-Clock Clock Stretch Is Selected for Slave) (3/3)

(3) Data ~ data ~ stop condition



Note 1. To cancel a clock stretch state, write “FFH” to IICAn or set the WRELn bit.

Note 2. Make sure that the time between the rise of the SCLAn pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0 µs when specifying standard mode and at least 0.6 µs when specifying fast mode.

Note 3. Write data to IICAn, not setting the WRELn bit, in order to cancel a clock stretch state during transmission by a slave device.

Note 4. If a clock stretch state during transmission by a slave device is canceled by setting the WRELn bit, the TRCn bit will be cleared.

Remark n = 0, 1

The meanings of <8> to <19> in (3) Data ~ data ~ stop condition in Figure 14 - 33 are explained below.

<8> The master device sets a clock stretch status (SCLAn = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICAn: end of transfer). Because of ACKEn = 0 in the master device, the master device then sends an ACK by hardware to the slave device.

<9> The master device reads the received data and releases the clock stretch status (WRELn = 1).

<10>The ACK is detected by the slave device (ACKDn = 1) at the rising edge of the 9th clock.

<11>The slave device set a clock stretch status (SCLAn = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICAn: end of transfer).

<12>By the slave device writing the data to transmit to the IICA register, the clock stretch status set by the slave device is released. The slave device then starts transferring data to the master device.

<13>The master device issues an interrupt (INTIICAn: end of transfer) at the falling edge of the 8th clock, and sets a clock stretch status (SCLAn = 0). Because ACK control (ACKEn = 1) is performed, the bus data line is at the low level (SDAAn = 0) at this stage.

<14>The master device sets NACK as the response (ACKEn = 0) and changes the timing at which it sets the clock stretch status to the 9th clock (WTIMn = 1).

<15>If the master device releases the clock stretch status (WRELn = 1), the slave device detects the NACK (ACKDn = 0) at the rising edge of the 9th clock.

<16>The master device and slave device set a clock stretch status (SCLAn = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICAn: end of transfer).

<17> When the master device issues a stop condition (SPTn = 1), the bus data line is cleared (SDAAn = 0) and the master device releases the clock stretch status. The master device then waits until the bus clock line is set (SCLAn = 1).

<18> The slave device acknowledges the NACK, halts transmission, and releases the clock stretch status (WRELn = 1) to end communication. Once the slave device releases the clock stretch status, the bus clock line is set (SCLAn = 1).

<19> Once the master device recognizes that the bus clock line is set (SCLAn = 1) and after the stop condition setup time has elapsed, the master device sets the bus data line (SDAAn = 1) and issues a stop condition (i.e. SCLAn =1 changes SDAAn from 0 to 1). The slave device detects the generated stop condition and slave device issue an interrupt (INTIICAn: stop condition).

Remark 1. <1> to <19> in Figures 14 - 33 represent the entire procedure for communicating data using the I^{2}C bus. Figure 14 - 33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 14 - 33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 14 - 33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Remark 2. n = 0, 1

CHAPTER 15 DATA OPERATION CIRCUIT (DOC)

15.1 Overview

The data operation circuit (DOC) is used to compare, add, and subtract 16-bit data.

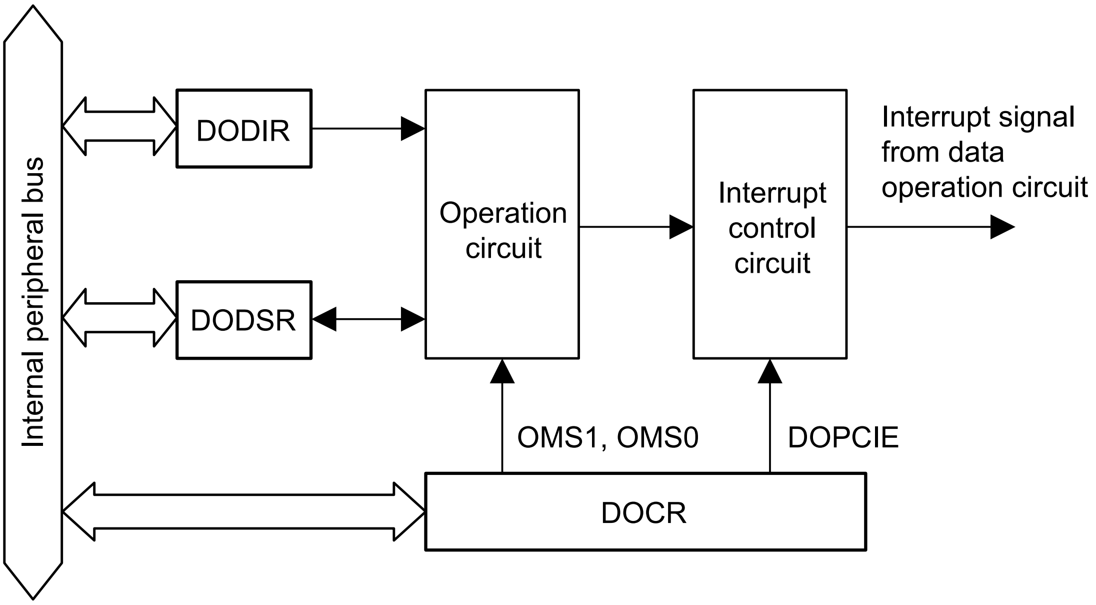
Table 15 - 1 lists the data operation circuit specifications and Figure 15 - 1 shows a block diagram of the data operation circuit.

16-bit data is compared and an interrupt can be generated when a selected condition applies.

Table 15 - 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 15 - 1 DOC Block Diagram



DOCR: DOC control register
DODIR: DOC data input register
DODSR: DOC data setting register

15.2 Registers Controlling Data Operation Circuit

Table 15 - 2 lists the registers used to control the data operation circuit.

Table 15 - 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

15.2.1 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 15 - 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	0	DOCEN	0	0	0	0	TKB0EN

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 1 to 4, 6

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.

15.2.2 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 15 - 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	TKB0RES

DOCRES	Reset control of data operation circuit
0	The data operation circuit is released from the reset state.
1	The data operation circuit is in the reset state.

15.2.3 DOC control register (DOCR)

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

Figure 15 - 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	Writing 0 has no effect.
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.

15.2.4 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 15 - 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																

15.2.5 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 15 - 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																

15.3 Operation

15.3.1 Data Comparison Mode

Figure 15 - 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.

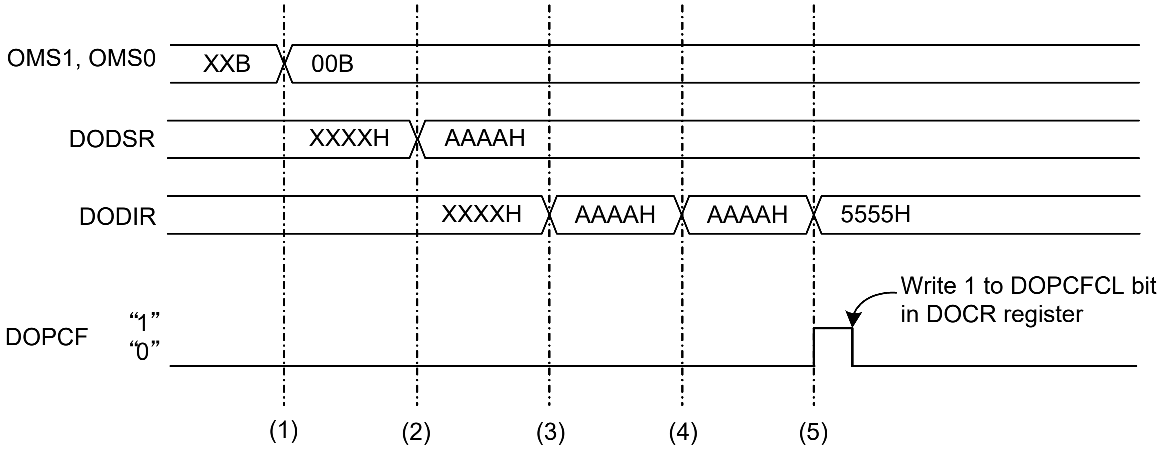
(3) 16-bit data for comparison is written to the DODIR register.

(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 15 - 7 Example of Operation in Data Comparison Mode



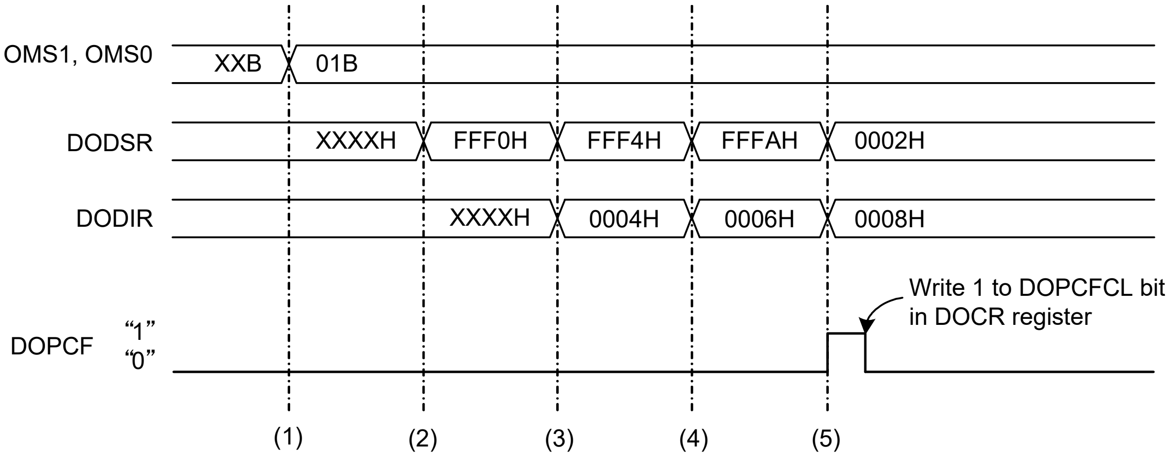
15.3.2 Data Addition Mode

Figure 15 - 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 15 - 8 Example of Operation in Data Addition Mode



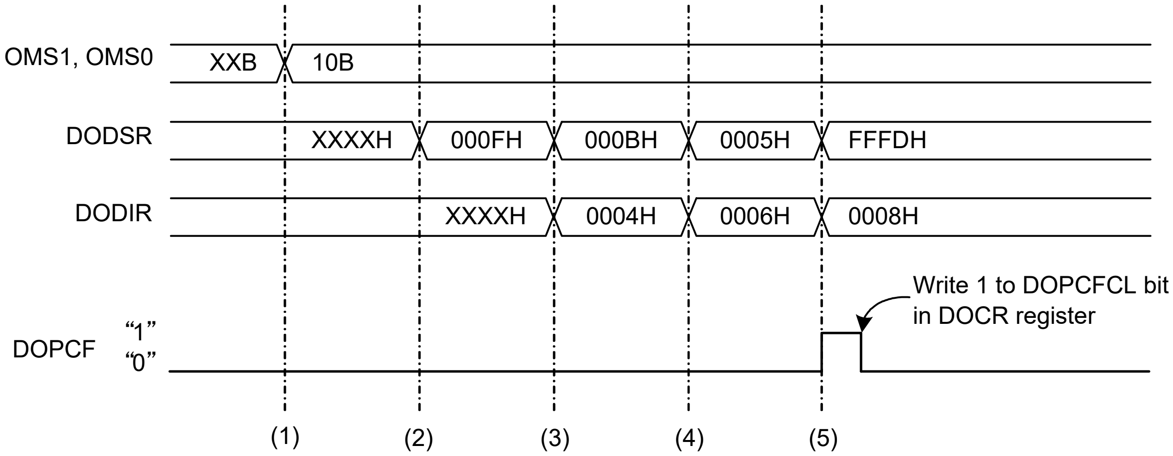
15.3.3 Data Subtraction Mode

Figure 15 - 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 15 - 9 Example of Operation in Data Subtraction Mode



15.4 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 15 - 3 describes the interrupt request.

Table 15 - 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 16 A/D CONVERTER

The number of analog input channels of the A/D converter differs, depending on the product.

	10-pin	16-pin	20-pin	24, 25-pin
Analog input channels	3 ch (ANI0 to ANI2)	8 ch (ANI0 to ANI3, ANI18, ANI20 to ANI22)	10 ch (ANI0 to ANI3, ANI16 to ANI18, ANI20 to ANI22)	11 ch (ANI0 to ANI3, ANI16 to ANI22)

16.1 Function of A/D Converter

The A/D converter is a converter that converts analog input signals into digital values, and is configured to control analog inputs, including up to 11 channels of A/D converter analog inputs (ANI0 to ANI3 and ANI16 to ANI22). 10-bit or 8-bit resolution can be selected by the ADTYP bit of the A/D converter mode register 2 (ADM2).
The A/D converter has the following function.

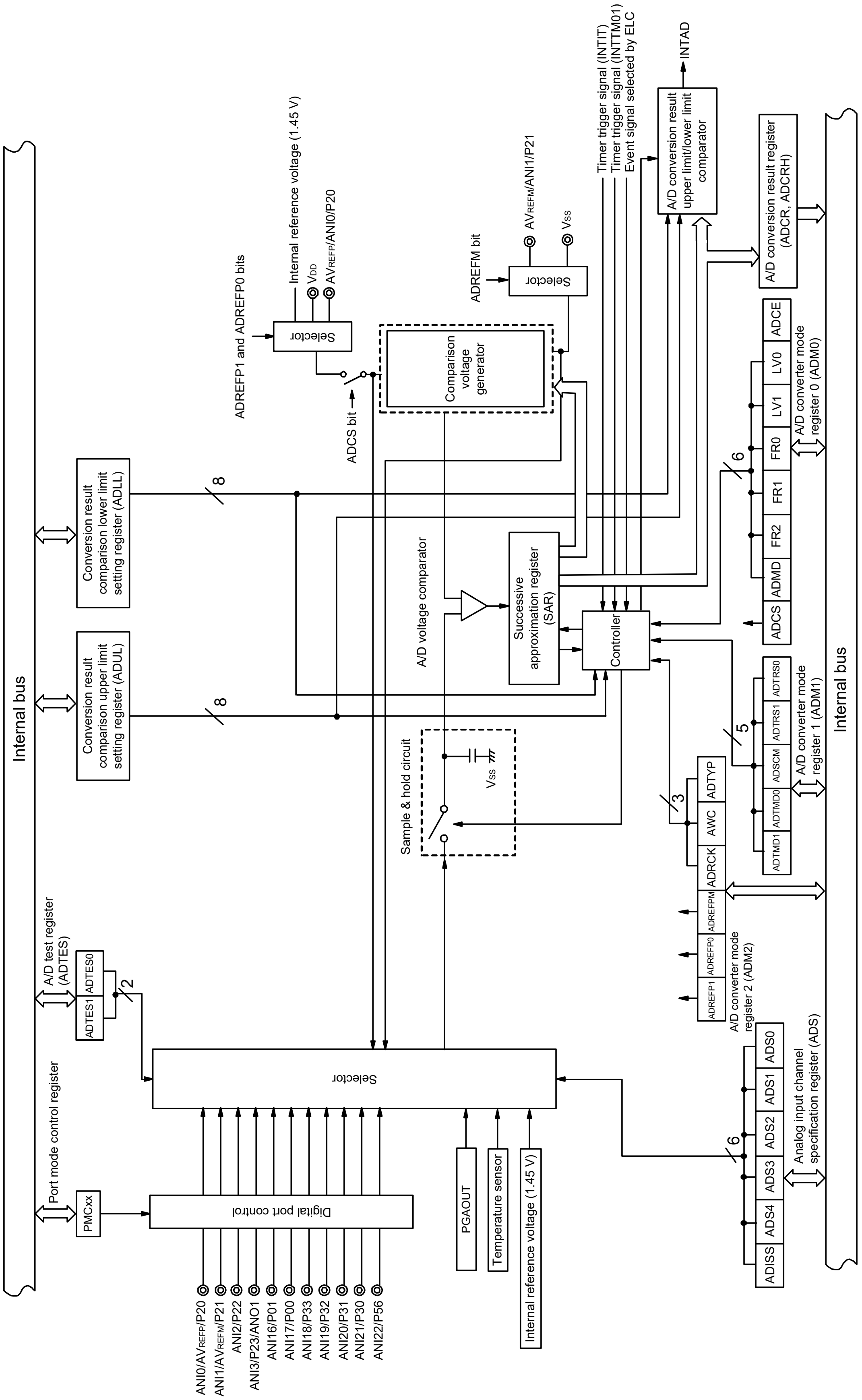
- 10-bit or 8-bit resolution A/D conversion

  10-bit or 8-bit resolution A/D conversion is carried out repeatedly for one analog input channel selected from ANI0 to ANI3 and ANI16 to ANI22. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated (when in the select mode).

Various A/D conversion modes can be specified by using the mode combinations below.

Trigger mode	Software trigger	Conversion is started by software.
	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 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. Four consecutive channels can be selected from ANI0 to ANI3 as analog input channels.
Conversion operation mode	One-shot conversion mode	A/D conversion is performed on the selected channel once.
	Sequential conversion mode	A/D conversion is sequentially performed on the selected channels until it is stopped by software.
Operation voltage mode	Standard 1 or standard 2 mode	Conversion is done in the operation voltage range of 2.7 V ≤ V_{DD} ≤ 5.5 V.
	Low voltage 1 or low voltage 2 mode	Conversion is done in the operation voltage range of 1.6 V ≤ V_{DD} ≤ 5.5 V. Select this mode for conversion at a low voltage. Because the operation voltage is low, it is internally boosted during conversion.
Sampling time selection	Sampling clock cycles: 7 f_{AD}	The sampling time in standard 1 or low voltage 1 mode is seven cycles of the conversion clock (f_{AD}). Select this mode when the output impedance of the analog input source is high and the sampling time should be long.
	Sampling clock cycles: 5 f_{AD}	The sampling time in standard 2 or low voltage 2 mode is five cycles of the conversion clock (f_{AD}). Select this mode when enough sampling time is ensured (for example, when the output impedance of the analog input source is low).

Figure 16 - 1 Block Diagram of A/D Converter


Internal bus
Conversion result comparison upper limit setting register (ADUL)
Conversion result comparison lower limit setting register (ADLL)
8
8
A/D test register (ADTES)
ADTES1
ADTES0
2
Port mode control register
PMCxx
Digital port control
Selector
ANI0/AVREFP/P20
ANI1/AVREFM/P21
ANI2/P22
ANI3/P23/ANO1
ANI16/P01
ANI17/P00
ANI18/P33
ANI19/P32
ANI20/P31
ANI21/P30
ANI22/P56
PGAOUT
Temperature sensor
Internal reference voltage (1.45 V)
Sample & hold circuit
VSS
A/D voltage comparator
Comparison voltage generator
ADCS bit
ADREFP1 and ADREFP0 bits
Selector
Internal reference voltage (1.45 V)
VDD
AVREFP/ANI0/P20
ADREFM bit
Selector
AVREFM/ANI1/P21
VSS
Successive approximation register (SAR)
Controller
Timer trigger signal (INTIT)
Timer trigger signal (INTTM01)
Event signal selected by ELC
A/D conversion result upper limit/lower limit comparator
INTAD
A/D conversion result register (ADCR, ADCRH)
ADCS
ADMD
FR2
FR1
FR0
LV1
LV0
ADCE
6
A/D converter mode register 0 (ADM0)
ADTMD1
ADTMD0
ADSCM
ADTRS1
ADTRS0
5
A/D converter mode register 1 (ADM1)
ADREFP1
ADREFP0
ADREFM
ADRCK
AWC
ADTYP
3
A/D converter mode register 2 (ADM2)
ADISS
ADS4
ADS3
ADS2
ADS1
ADS0
6
Analog input channel specification register (ADS)
Internal bus

Remark Analog input pin for Figure 16 - 1 when a 25 -pin product is used.

16.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

(1) ANI0 to ANI3 and ANI16 to ANI22 pins

These are the analog input pins of the 11 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 the voltage generated from the voltage tap of the comparison voltage generator with the analog input voltage. If the analog input voltage is found to be greater than the reference voltage (1/2 AVREF) 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 AVREF), the MSB bit of the SAR is reset.

After that, bit 8 of the SAR register is automatically set, and the next comparison is made. The voltage tap of the comparison voltage generator is selected by the value of bit 9, to which the result has been already set.

Bit 9 = 0: (1/4 AVREF)
Bit 9 = 1: (3/4 AVREF)

The voltage tap of the comparison voltage generator and the analog input voltage are compared and bit 8 of the SAR register is manipulated according to the result of the comparison.

Analog input voltage ≥ Voltage tap of comparison voltage generator: Bit 8 = 1
Analog input voltage ≤ Voltage tap of comparison voltage generator: Bit 8 = 0

Comparison is continued like this to bit 0 of the SAR register.

When performing A/D conversion at a resolution of 8 bits, the comparison continues until bit 2 of the SAR register.

Remark AVREF: The + side reference voltage of the A/D converter. This can be selected from AVREFP, the internal reference voltage (1.45 V), and VDD.

(4) Comparison voltage generator

The comparison voltage generator generates the comparison voltage input from an analog input pin.

(5) Successive approximation register (SAR)

The SAR register is a register that sets voltage tap data whose values from the comparison voltage generator match the voltage values of the analog input pins, 1 bit at a time starting from the most significant bit (MSB).

If data is set in the SAR register all the way to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register (conversion results) are held in the A/D conversion result register (ADCR). When all the specified A/D conversion operations have ended, an A/D conversion end interrupt request signal (INTAD) is generated.

(6) 10-bit A/D conversion result register (ADCR)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCR register holds the A/D conversion result in its higher 10 bits (the lower 6 bits are fixed to 0).

(7) 8-bit A/D conversion result register (ADCRH)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCRH register stores the higher 8 bits of the A/D conversion result.

(8) Controller

This circuit controls the conversion time of an input analog signal that is to be converted into a digital signal, as well as starting and stopping of the conversion operation. When A/D conversion has been completed, this controller generates INTAD 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 ANI3 and ANI16 to ANI22 are converted to digital signals based on the voltage applied between AVREFP and the – side reference voltage (AVREFM/VSS).

In addition to AVREFP, it is possible to select VDD 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 VSS as the – side reference voltage of the A/D converter.

16.3 Registers Controlling A/D Converter

The A/D converter is controlled by 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)
- 10-bit A/D conversion result register (ADCR)
- 8-bit A/D conversion result register (ADCRH)
- Analog input channel specification register (ADS)
- Conversion result comparison upper limit setting register (ADUL)
- Conversion result comparison lower limit setting register (ADLL)
- A/D test register (ADTES)
- Port mode control registers 0, 2, 3, and 5 (PMC0, PMC2, PMC3, PMC5)
- Port mode registers 0, 2, 3, and 5 (PM0, PM2, PM3, PM5)

16.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When 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 16 - 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	0	IICA1EN	ADCEN	IICA0EN	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, the values of the A/D converter control registers are cleared to their initial values and writing to them is ignored (except for port mode registers 0, 2, 3 and 5 (PM0, PM2, PM3, PM5), port mode control registers 0, 2, 3 and 5 (PMC0, PMC2, PMC3, PMC5)).

- **A/D converter mode register 0 (ADM0)**
- **A/D converter mode register 1 (ADM1)**
- **A/D converter mode register 2 (ADM2)**
- **10-bit A/D conversion result register (ADCR)**
- **8-bit A/D conversion result register (ADCRH)**
- **Analog input channel specification register (ADS)**
- **Conversion result comparison upper limit setting register (ADUL)**
- **Conversion result comparison lower limit setting register (ADLL)**
- **A/D test register (ADTES)**

Caution 2. Be sure to clear the following bits to 0.

bits 1, 3, and 7

16.3.2 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 5 (ADCRES) 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 16 - 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	IICA1RES	ADCRES	IICA0RES	0	SAU0RES	0	TAU0RES

ADCRES	Reset control of serial array unit
0	A/D converter is released from the reset state.
1	A/D converter is in the reset state.

16.3.3 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 16 - 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 **Table 16 - 3 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 1 μs from the start of operation for the operation to stabilize. Therefore, when the ADCS bit is set to 1 after 1 μs or more has elapsed from the time ADCE bit is set to 1, the conversion result at that time has priority over the first conversion result. Otherwise, ignore data of the first conversion.

Caution 1. Change the ADMD, FR2 to FR0, LV1, and LV0 bits while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 2. Do not set the ADCS bit to 1 and the ADCE bit to 0 at the same time.

Caution 3. Do not change the ADCS and ADCE 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 16.7 A/D Converter Setup Flowchart.

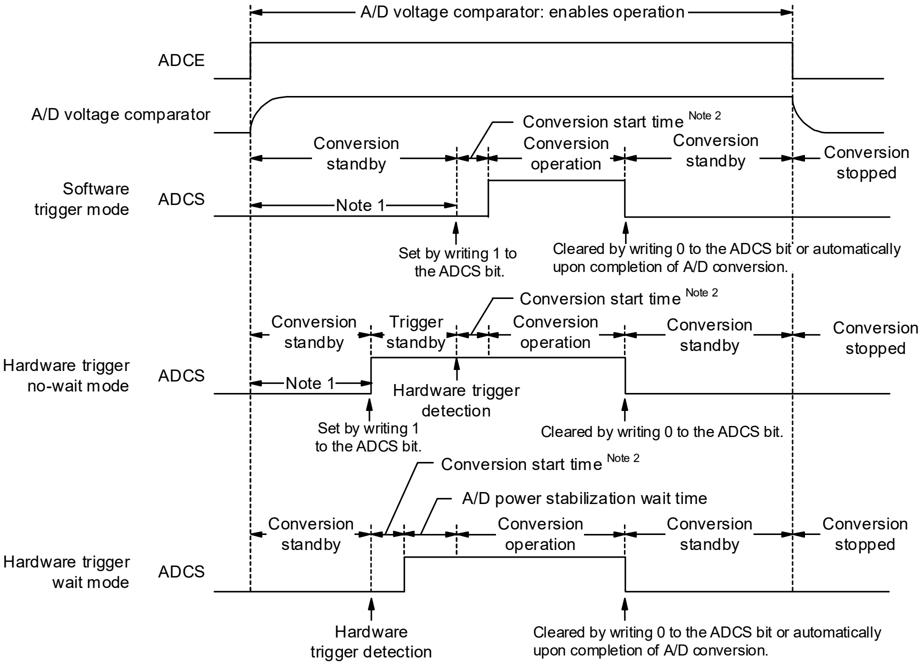
Table 16 - 1 Settings of ADCS and ADCE Bits

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 16 - 2 Setting and Clearing Conditions for ADCS Bit

A/D Conversion Mode			Set Conditions	Clear Conditions
Software trigger	Select mode	Sequential conversion mode	When 1 is written to ADCS	When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • 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 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 16 - 5 Timing Chart When A/D Voltage Comparator Is Used


A/D voltage comparator: enables operation
ADCE
A/D voltage comparator
Conversion start time Note 2
Conversion standby
Conversion operation
Conversion standby
Conversion stopped
Software trigger mode
ADCS
Note 1
Set by writing 1 to the ADCS bit.
Cleared by writing 0 to the ADCS bit or automatically upon completion of A/D conversion.
Conversion start time Note 2
Conversion standby
Trigger standby
Conversion operation
Conversion standby
Conversion stopped
Hardware trigger no-wait mode
ADCS
Note 1
Hardware trigger detection
Set by writing 1 to the ADCS bit.
Cleared by writing 0 to the ADCS bit.
Conversion start time Note 2
A/D power stabilization wait time
Conversion standby
Conversion operation
Conversion standby
Conversion stopped
Hardware trigger wait mode
ADCS
Hardware trigger detection
Cleared by writing 0 to the ADCS bit or automatically upon completion of A/D conversion.

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 1 μs or longer to stabilize the internal circuit.

Note 2. The following time is the maximum amount of time necessary to start conversion.

ADM0			Conversion Clock (f_{AD})	Conversion Start Time (Number of f_{CLK} Clocks)	
FR2	FR1	FR0		Software trigger mode/ Hardware trigger no wait mode	Hardware trigger wait mode
0	0	0	f_{CLK}/64	63	1
0	0	1	f_{CLK}/32	31	
0	1	0	f_{CLK}/16	15	
0	1	1	f_{CLK}/8	7	
1	0	0	f_{CLK}/6	5	
1	0	1	f_{CLK}/5	4	
1	1	0	f_{CLK}/4	3	
1	1	1	f_{CLK}/2	1	

However, for the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 2, and 3 in scan mode, the conversion start time and stabilization wait time for A/D power supply do not occur after a hardware trigger is detected.

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 16 - 3 A/D Conversion Time Selection (1/4)

(1) When there is no A/D power supply stabilization wait time Normal mode 1, 2 (software trigger mode/hardware trigger no-wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of Conversion Clock Note	Conversion Time	Conversion Time at 10-Bit Resolution				
									2.7 V ≤ V_{DD} ≤ 5.5 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	Normal 1	f_{CLK}/64	19 f_{AD} (number of sampling clock: 7 f_{AD})	1216/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	76 µs	50.667 µs
0	0	1				f_{CLK}/32		608/f_{CLK}			76 µs	38 µs	25.333 µs
0	1	0				f_{CLK}/16		304/f_{CLK}		76 µs	38 µs	19 µs	12.667 µs
0	1	1				f_{CLK}/8		152/f_{CLK}		38 µs	19 µs	9.5 µs	6.333 µs
1	0	0				f_{CLK}/6		114/f_{CLK}		28.5 µs	14.25 µs	7.125 µs	4.75 µs
1	0	1				f_{CLK}/5		95/f_{CLK}	95 µs	23.75 µs	11.875 µs	5.938 µs	3.958 µs
1	1	0				f_{CLK}/4		76/f_{CLK}	76 µs	19 µs	9.5 µs	4.75 µs	3.167 µs
1	1	1				f_{CLK}/2		38/f_{CLK}	38 µs	9.5 µs	4.75 µs	2.375 µs	Setting prohibited
0	0	0	0	1	Normal 2	f_{CLK}/64	17 f_{AD} (number of sampling clock: 5 f_{AD})	1088/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	68 µs	45.333 µs
0	0	1				f_{CLK}/32		544/f_{CLK}			68 µs	34 µs	22.667 µs
0	1	0				f_{CLK}/16		272/f_{CLK}		68 µs	34 µs	17 µs	11.333 µs
0	1	1				f_{CLK}/8		136/f_{CLK}		34 µs	17 µs	8.5 µs	5.667 µs
1	0	0				f_{CLK}/6		102/f_{CLK}		25.5 µs	12.75 µs	6.375 µs	4.25 µs
1	0	1				f_{CLK}/5		85/f_{CLK}	85 µs	21.25 µs	10.625 µs	5.3125 µs	3.542 µs
1	1	0				f_{CLK}/4		68/f_{CLK}	68 µs	17 µs	8.5 µs	4.25 µs	2.833 µs
1	1	1				f_{CLK}/2		34/f_{CLK}	34 µs	8.5 µs	4.25 µs	2.125 µs	Setting prohibited

Note These are the numbers of clock cycles when conversion is with 10-bit resolution. When 8-bit resolution is selected, the values are shorter by two cycles of the conversion clock (f_{AD}).

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 35.6.1 or 36.6.1 A/D converter characteristics A/D converter characteristics.

Caution 2. Rewrite the FR2 to FR0, LV1 and LV0 bits to other than the same data while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 16 - 3 A/D Conversion Time Selection (2/4)

(2) When there is no A/D power supply stabilization wait time Low-voltage mode 1, 2 (software trigger mode/hardware trigger no-wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (fAD)	Number of Conversion Clock Note 4	Conversion Time	Conversion Time at 10-Bit Resolution				
									1.6 V ≤ VDD ≤ 5.5 V		Note 1	Note 2	Note 3
FR2	FR1	FR0	LV1	LV0					fCLK = 1 MHz	fCLK = 4 MHz	fCLK = 8 MHz	fCLK = 16 MHz	fCLK = 24 MHz
0	0	0	1	0	Low-voltage 1	fCLK/64	19 fAD (number of sampling clock: 7 fAD)	1216/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	76 µs	50.667 µs
0	0	1				fCLK/32		608/fCLK			76 µs	38 µs	25.333 µs
0	1	0				fCLK/16		304/fCLK		76 µs	38 µs	19 µs	12.667 µs
0	1	1				fCLK/8		152/fCLK		38 µs	19 µs	9.5 µs	6.333 µs
1	0	0				fCLK/6		114/fCLK		28.5 µs	14.25 µs	7.125 µs	4.75 µs
1	0	1				fCLK/5		95/fCLK	95 µs	23.75 µs	11.875 µs	5.938 µs	3.958 µs
1	1	0				fCLK/4		76/fCLK	76 µs	19 µs	9.5 µs	4.75 µs	3.167 µs
1	1	1				fCLK/2		38/fCLK	38 µs	9.5 µs	4.75 µs	2.375 µs	Setting prohibited
0	0	0	1	1	Low-voltage 2	fCLK/64	17 fAD (number of sampling clock: 5 fAD)	1088/fCLK	Setting prohibited	Setting prohibited	Setting prohibited	68 µs	45.333 µs
0	0	1				fCLK/32		544/fCLK			68 µs	34 µs	22.667 µs
0	1	0				fCLK/16		272/fCLK		68 µs	34 µs	17 µs	11.333 µs
0	1	1				fCLK/8		136/fCLK		34 µs	17 µs	8.5 µs	5.667 µs
1	0	0				fCLK/6		102/fCLK		25.5 µs	12.75 µs	6.375 µs	4.25 µs
1	0	1				fCLK/5		85/fCLK	85 µs	21.25 µs	10.625 µs	5.3125 µs	3.542 µs
1	1	0				fCLK/4		68/fCLK	68 µs	17 µs	8.5 µs	4.25 µs	2.833 µs
1	1	1				fCLK/2		34/fCLK	34 µs	8.5 µs	4.25 µs	2.125 µs	Setting prohibited

Note 1. 1.8 V ≤ VDD ≤ 5.5 V

Note 2. 2.4 V ≤ VDD ≤ 5.5 V

Note 3. 2.7 V ≤ VDD ≤ 5.5 V

Note 4. These are the numbers of clock cycles when conversion is with 10-bit resolution. When 8-bit resolution is selected, the values are shorter by two cycles of the conversion clock (fAD).

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (tCONV) described in 35.6.1 or 36.6.1 A/D converter characteristics.

Caution 2. Rewrite the FR2 to FR0, LV1 and LV0 bits to other than the same data while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

Remark fCLK: CPU/peripheral hardware clock frequency

Table 16 - 3 A/D Conversion Time Selection (3/4)

(3) When there is A/D power supply stabilization wait time Normal mode 1, 2 (hardware trigger wait mode Note 1)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of A/D Power Supply Stabilization Wait Clock	Number of Conversion Clock Note 2	A/D Power Supply Stabilization Wait Time + Conversion Time	A/D Power Supply Stabilization Wait Time + Conversion Time at 10-Bit Resolution				
										2.7 V ≤ V_{DD} ≤ 5.5 V				
FR 2	FR 1	FR 0	LV 1	LV 0						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	Normal 1	f_{CLK}/64	8 f_{AD}	19 f_{AD} (number of sampling clock: 7 f_{AD})	1728/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	108 µs	72 µs
0	0	1				f_{CLK}/32			864/f_{CLK}			108 µs	54 µs	36 µs
0	1	0				f_{CLK}/16			432/f_{CLK}		108 µs	54 µs	27 µs	18 µs
0	1	1				f_{CLK}/8			216/f_{CLK}		54 µs	27 µs	13.5 µs	9 µs
1	0	0				f_{CLK}/6			162/f_{CLK}		40.5 µs	20.25 µs	10.125 µs	6.75 µs
1	0	1				f_{CLK}/5			135/f_{CLK}	135 µs	33.75 µs	16.875 µs	8.4375 µs	5.625 µs
1	1	0				f_{CLK}/4			108/f_{CLK}	108 µs	27 µs	13.5 µs	6.75 µs	4.5 µs
1	1	1				f_{CLK}/2			54/f_{CLK}	54 µs	13.5 µs	6.75 µs	3.375 µs	2.25 µs
0	0	0	0	1	Normal 2	f_{CLK}/64	8 f_{AD}	17 f_{AD} (number of sampling clock: 5 f_{AD})	1600/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	100 µs	66.667 µs
0	0	1				f_{CLK}/32			800/f_{CLK}			100 µs	50 µs	33.333 µs
0	1	0				f_{CLK}/16			400/f_{CLK}		100 µs	50 µs	25 µs	16.667 µs
0	1	1				f_{CLK}/8			200/f_{CLK}		50 µs	25 µs	12.5 µs	8.333 µs
1	0	0				f_{CLK}/6			150/f_{CLK}		37.5 µs	18.75 µs	9.375 µs	6.25 µs
1	0	1				f_{CLK}/5			125/f_{CLK}	125 µs	31.25 µs	15.625 µs	7.8125 µs	5.208 µs
1	1	0				f_{CLK}/4			100/f_{CLK}	100 µs	25 µs	12.5 µs	6.25 µs	4.167 µs
1	1	1				f_{CLK}/2			50/f_{CLK}	50 µs	12.5 µs	6.25 µs	3.125 µs	2.083 µs

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 16 - 3**).

Note 2. These are the numbers of clock cycles when conversion is with 10-bit resolution. When 8-bit resolution is selected, the values are shorter by two cycles of the conversion clock (f_{AD}).

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 35.6.1 or 36.6.1 A/D converter characteristics.
Note that the conversion time (t_{CONV}) does not include the A/D power supply stabilization wait time.

Caution 2. Rewrite the FR2 to FR0, LV1 and LV0 bits to other than the same data while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

Caution 4. When hardware trigger wait mode, specify the conversion time, including the A/D power supply stabilization wait time from the hardware trigger detection.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 16 - 3 A/D Conversion Time Selection (4/4)

(4) When there is A/D power supply stabilization wait time Low-voltage mode 1, 2 (hardware trigger wait mode Note 1)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of A/D Power Supply Stabilization Wait Clock	Number of Conversion Clock Note 5	A/D Power Supply Stabilization Wait Time + Conversion Time	A/D Power Supply Stabilization Wait Time + Conversion Time at 10-Bit Resolution				
										1.6 V ≤ V_{DD} ≤ 5.5 V		Note 2	Note 3	Note 4
FR 2	FR 1	FR 0	LV 1	LV 0						f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 24 MHz
0	0	0	1	0	Low-voltage1	$f_{CLK}/64$	2 f_{AD}	19 f_{AD} (number of sampling clock: 7 f_{AD})	1344/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	84 µs	56 µs
0	0	1				$f_{CLK}/32$			672/f_{CLK}			84 µs	42 µs	28 µs
0	1	0				$f_{CLK}/16$			336/f_{CLK}		84 µs	42 µs	21 µs	14 µs
0	1	1				$f_{CLK}/8$			168/f_{CLK}		42 µs	21 µs	10.5 µs	7 µs
1	0	0				$f_{CLK}/6$			126/f_{CLK}		31.25 µs	15.75 µs	7.875 µs	5.25 µs
1	0	1				$f_{CLK}/5$			105/f_{CLK}	105 µs	26.25 µs	13.125 µs	6.5625 µs	4.375 µs
1	1	0				$f_{CLK}/4$			84/f_{CLK}	84 µs	21 µs	10.5 µs	5.25 µs	3.5 µs
1	1	1				$f_{CLK}/2$			42/f_{CLK}	42 µs	10.5 µs	5.25 µs	2.625 µs	1.75 µs
0	0	0	1	1	Low-voltage2	$f_{CLK}/64$	2 f_{AD}	17 f_{AD} (number of sampling clock: 5 f_{AD})	1216/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	76 µs	50.667 µs
0	0	1				$f_{CLK}/32$			608/f_{CLK}			76 µs	38 µs	25.333 µs
0	1	0				$f_{CLK}/16$			304/f_{CLK}		76 µs	38 µs	19 µs	12.667 µs
0	1	1				$f_{CLK}/8$			152/f_{CLK}		38 µs	19 µs	9.5 µs	6.333 µs
1	0	0				$f_{CLK}/6$			114/f_{CLK}		28.5 µs	14.25 µs	7.125 µs	4.75 µs
1	0	1				$f_{CLK}/5$			95/f_{CLK}	96 µs	23.75 µs	11.875 µs	5.938 µs	3.958 µs
1	1	0				$f_{CLK}/4$			76/f_{CLK}	76 µs	19 µs	9.5 µs	4.75 µs	3.167 µs
1	1	1				$f_{CLK}/2$			38/f_{CLK}	38 µs	9.5 µs	4.75 µs	2.375 µs	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 16 - 3**).

Note 2. 1.8 V ≤ V_{DD} ≤ 5.5 V

Note 3. 2.4 V ≤ V_{DD} ≤ 5.5 V

Note 4. 2.7 V ≤ V_{DD} ≤ 5.5 V

Note 5. These are the numbers of clock cycles when conversion is with 10-bit resolution. When 8-bit resolution is selected, the values are shorter by two cycles of the conversion clock (f_{AD}).

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 35.6.1 or 36.6.1 A/D converter characteristics.
Note that the conversion time (t_{CONV}) does not include the A/D power supply stabilization wait time.

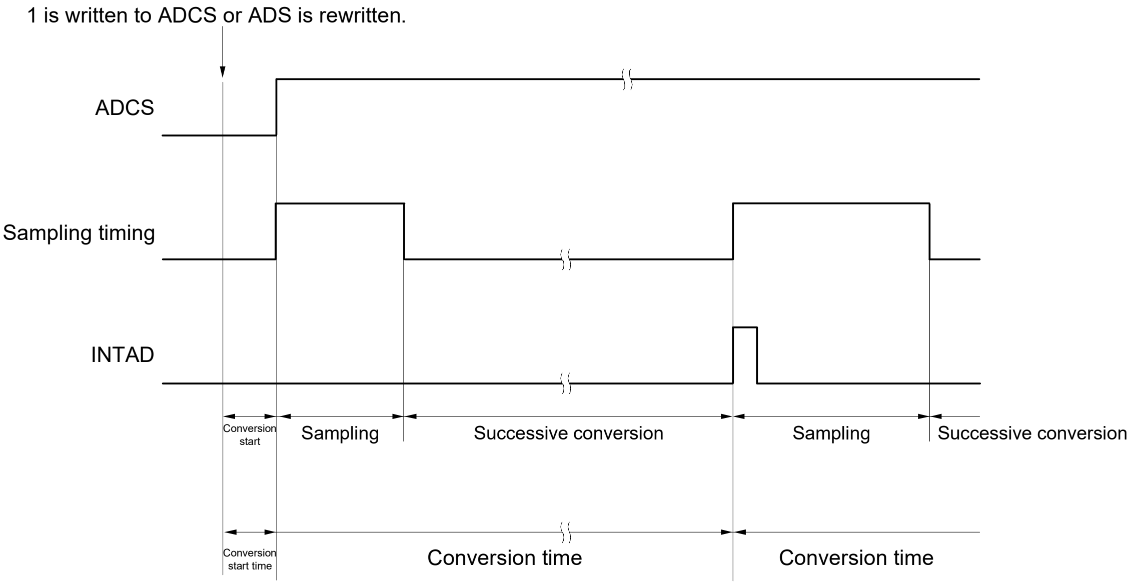
Caution 2. Rewrite the FR2 to FR0, LV1 and LV0 bits to other than the same data while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

Caution 4. When hardware trigger wait mode, specify the conversion time, including the A/D power supply stabilization wait time from the hardware trigger detection.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Figure 16 - 6 A/D Converter Sampling and A/D Conversion Timing (Example for Software Trigger Mode)



16.3.4 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 16 - 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	0	Software trigger mode
0	1	
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	Setting prohibited
1	1	12-bit interval timer interrupt signal (INTIT)

Caution 1. Rewrite the value of the ADM1 register while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 2. To complete A/D conversion, specify at least the following time as the hardware trigger interval:
Hardware trigger no wait mode: 2 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 mode, input of the next INTIT will not be recognized as a valid hardware trigger for up to four f_{CLK} cycles after the first INTIT is input.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

16.3.5 A/D converter mode register 2 (ADM2)

This register is used to select the + side or - side reference voltage of the A/D converter, 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 16 - 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 VDD
0	1	Supplied from P20/AVREFP/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) Reference voltage stabilization wait time (A)
(4) Set ADCE = 1
(5) Reference voltage stabilization wait time (B)
When ADREFP1 and ADREFP0 are set to 1 and 0, the setting is changed to A = 10 µs, B = 1 µs.
When ADREFP1 and ADREFP0 are set to 0 and 0 or 0 and 1, A needs no wait and B = 1 µs.

- 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.

ADREFM	Selection of the - side reference voltage of the A/D converter
0	Supplied from VSS
1	Supplied from P21/AVREFM/ANI1

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. When entering STOP mode or HALT mode while the CPU is operating on the subsystem clock, do not set ADREFP1 to 1. When selecting internal reference voltage (ADREFP1, ADREFP0 = 1, 0), the current value of A/D converter reference voltage current (IADREF) shown in 35.3.2 or 36.3.2 Supply current characteristics is added.

Caution 3. When using AVREFP and AVREFM, specify ANI0 and ANI1 as the analog input channels and specify input mode by using the port mode register.

Figure 16 - 8 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

ADRCK	Checking the upper limit and lower limit conversion result values
0	The interrupt signal (INTAD) is output when the ADLL register ≤ the ADCR register ≤ the ADUL register (AREA1).
1	The interrupt signal (INTAD) is output when the ADCR register < the ADLL register (AREA2) or the ADUL register < the ADCR register (AREA3).
Figure 16 - 9 shows the generation range of the interrupt 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).

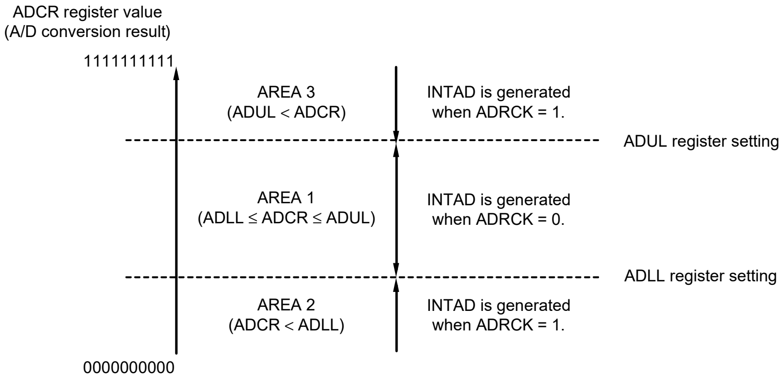
- The SNOOZE mode function can only be specified when the high-speed on-chip oscillator clock (f_{IH}) or middle-speed on-chip oscillator clock (f_{IM}) is selected for the CPU/peripheral hardware clock (f_{CLK}). If any other clock is selected, specifying this mode is prohibited.
- Using the SNOOZE 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 + conversion start time + 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 mode.

ADTYP	Selection of the A/D conversion resolution
0	10-bit resolution
1	8-bit resolution

Note Refer to "Transition time from STOP mode to SNOOZE mode" in **24.3.3 SNOOZE mode**.

Caution Only rewrite the value of the ADM2 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).

Figure 16 - 9 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.

16.3.6 10-bit A/D conversion result register (ADCR)

This register is a 16-bit register that stores the A/D conversion result in the select mode. The lower 6 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR). The higher 8 bits of the conversion result are stored in FFF1FH and the lower 2 bits are stored in the higher 2 bits of FFF1EH [Note].

The ADCR register can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Note If the A/D conversion result is outside the range specified by using the A/D conversion comparison function (the value specified by the ADRCK bit of the ADM2 register and ADUL/ADLL registers; see **Figure 16 - 9**), the result is not stored.

Figure 16 - 10 Format of 10-bit A/D conversion result register (ADCR)

Address: FFF1FH, FFF1EH After reset: 0000H R

Symbol	FFF1FH								FFF1EH							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCR											0	0	0	0	0	0

Caution 1. When 8-bit resolution A/D conversion is selected (when the ADTYP bit of A/D converter mode register 2 (ADM2) is 1) and the ADCR register is read, 0 is read from the lower two bits (bits 7 and 6 of the ADCR register).

Caution 2. When the ADCR register is accessed in 16-bit units, the higher 10 bits of the conversion result are read in order starting at bit 15.

16.3.7 8-bit A/D conversion result register (ADCRH)

This register is an 8-bit register that stores the A/D conversion result. The higher 8 bits of 10-bit resolution are stored [Note].

The ADCRH register can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Note If the A/D conversion result is outside the range specified by using the A/D conversion comparison function (the value specified by the ADRCK bit of the ADM2 register and ADUL/ADLL registers; see **Figure 16 - 9**), the result is not stored.

Figure 16 - 11 Format of 8-bit A/D conversion result register (ADCRH)

Address: FFF1FH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ADCRH								

Caution When writing to the A/D converter mode register 0 (ADM0), Analog input channel specification register (ADS), 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.

16.3.8 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 16 - 12 Format of Analog input channel specification register (ADS) (1/2)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

- Select mode (ADMD = 0)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	0	0	ANI0	P20/ANI0/AVREFP pin
0	0	0	0	0	1	ANI1	P21/ANI1/AVREFM pin
0	0	0	0	1	0	ANI2	P22/ANI2 pin
0	0	0	0	1	1	ANI3	P23/ANI3 pin
0	1	0	0	0	0	ANI16	P01/ANI16 pin
0	1	0	0	0	1	ANI17	P00/ANI17 pin
0	1	0	0	1	0	ANI18	P33/ANI18 pin
0	1	0	0	1	1	ANI19	P32/ANI19 pin
0	1	0	1	0	0	ANI20	P31/ANI20 pin
0	1	0	1	0	1	ANI21	P30/ANI21
0	1	0	1	1	0	ANI22	P56/ANA22
0	1	0	1	1	1	—	PGAOUT (PGA output)
1	0	0	0	0	0	—	Temperature sensor output voltage Note
1	0	0	0	0	1	—	Internal reference voltage (1.45 V)
Other than the above						Setting prohibited	

Note If the internal reference voltage (1.45 V) is selected as the reference voltage of comparator 0 or comparator 1, the temperature sensor output cannot be selected.

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

Figure 16 - 12 Format of Analog input channel specification register (ADS) (2/2)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

- Scan mode (ADMD = 1)

ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel			
					Scan 0	Scan 1	Scan 2	Scan 3
0	0	0	0	0	ANI0	ANI1	ANI2	ANI3
Other than the above					Setting prohibited			

Caution 1. Be sure to clear bits 5 and 6 to 0.

Caution 2. Set a channel to be set the analog input by PMCx registers in the input mode by using port mode registers 0, 2, 3, and 5 (PM0, PM2, PM3, PM5).

Caution 3. Do not set the pin that is set by Port mode control registers 0, 2, 3, and 5 (PMC0, PMC2, PMC3, PMC5) as digital I/O by the ADS register.

Caution 4. Rewrite the value of the ADISS bit while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 5. If using AVREFP as the + side reference voltage of the A/D converter, do not select ANI0 as an A/D conversion channel.

Caution 6. If using AVREFM as the − side reference voltage of the A/D converter, do not select ANI1 as an A/D conversion channel.

Caution 7. If the ADISS bit is set to 1, the internal reference voltage (1.45 V) cannot be used for the + side reference voltage. After the ADISS bit is set to 1, the initial conversion result cannot be used.
For the setting flow, see 16.7.4 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. When the ADISS bit is set to 1, the A/D converter reference voltage current (IADREF) indicated in 35.3.2 or 36.3.2 Supply current characteristics will be added.

16.3.9 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 16 - 9**).

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

Reset signal generation sets this register to FFH.

Caution 1.When 10-bit resolution A/D conversion is selected, the higher eight bits of the 10-bit A/D conversion result register (ADCR) are compared with the ADUL and ADLL registers.

Caution 2. Only write new values to the ADUL and ADLL registers while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The setting of the ADUL and ADLL registers must be greater than that of the ADLL register.

Figure 16 - 13 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

16.3.10 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 16 - 9**).

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

Reset signal generation clears this register to 00H.

Figure 16 - 14 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 10-bit resolution A/D conversion is selected, the higher eight bits of the 10-bit A/D conversion result register (ADCR) are compared with the ADUL and ADLL registers.

Caution 2. Only write new values to the ADUL and ADLL registers while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The setting of the ADUL and ADLL registers must be greater than that of the ADLL register.

16.3.11 A/D test register (ADTES)

This register is used to select the + side reference voltage or - side reference voltage for the 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 this register to test the converter, 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 16 - 15 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	The - side reference voltage (selected by the ADREFM bit of the ADM2 register)
1	1	The + side reference voltage (selected by the ADREFP1 or ADREFP0 bit of the ADM2 register)
Other than the above		Setting prohibited

Caution For details on the A/D test function, refer to CHAPTER 28 SAFETY FUNCTIONS.

16.3.12 Registers that control port functions of analog input pins

Set the registers that control the port functions for the port pins with which the analog input pin functions of the A/D converter are multiplexed (port mode registers (PMxx), port mode control registers (PMCxx)). For details, see **4.3.1 Port mode registers (PMxx)** and **4.3.6 Port mode control registers (PMCxx)**.

Using the ANI0 to ANI3 and ANI16 to ANA22 pins for analog input to the A/D converter requires setting the corresponding bits in the port mode register (PMxx) and port mode control register (PMCxx) to 1.

16.4 A/D Converter Conversion Operations

The A/D converter conversion operations are described below.

<1> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<2> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the sampled voltage is held until the A/D conversion operation has ended.

<3> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) 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 8 of the SAR register is automatically set to 1, and the operation proceeds to the next comparison.

The series resistor string voltage tap is selected according to the preset value of bit 9, as described below.

- Bit 9 = 1: (3/4) AVREF
- Bit 9 = 0: (1/4) AVREF

The voltage tap and sampled voltage are compared and bit 8 of the SAR register is manipulated as follows.

- Sampled voltage ≥ Voltage tap: Bit 8 = 1
- Sampled voltage < Voltage tap: Bit 8 = 0

<6> Comparison is continued in this way up to bit 0 of the SAR register.

<7> Upon completion of the comparison of 10 bits, an effective digital result value remains in the SAR register, and the result value is transferred to the A/D conversion result register (ADCR, ADCRH) and then latched Note 1.

At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.

<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 16 - 9**), 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.

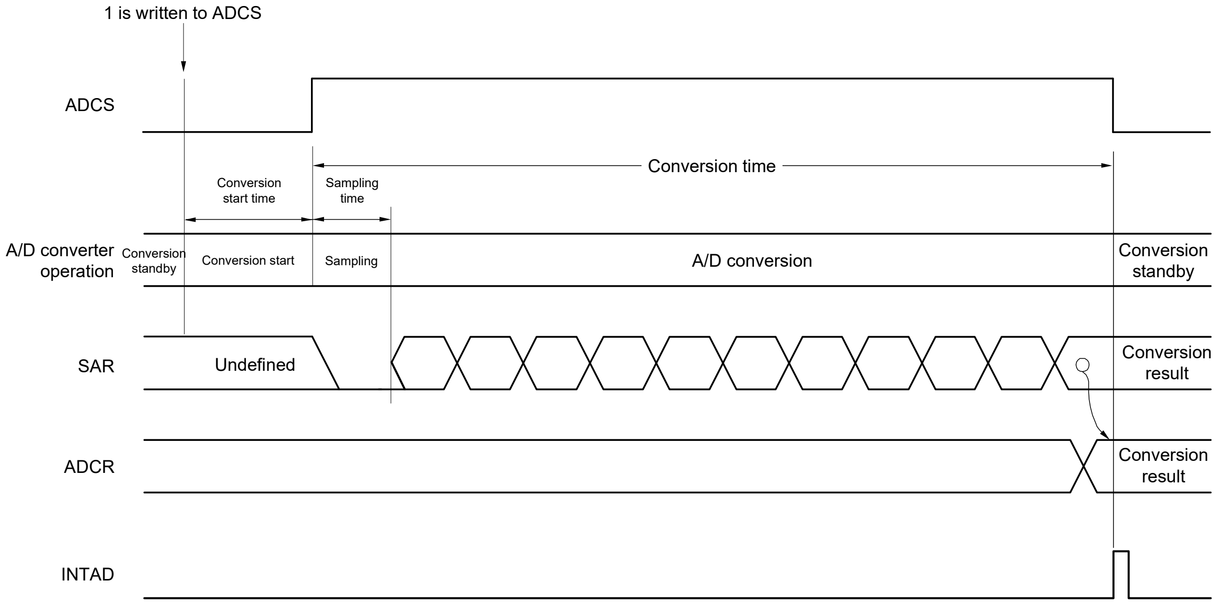
Note 2. While in the sequential conversion mode, the ADCS flag is not automatically cleared to 0. This flag is not automatically cleared to 0 while in the one-shot conversion mode of the hardware trigger no-wait mode, either. Instead, 1 is retained.

Remark 1. Two types of the A/D conversion result registers are available.

- ADCR register (16 bits): Store 10-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 VDD.

Figure 16 - 16 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.

Writing 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.

16.5 Input Voltage and Conversion Results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI3 and ANI16 to ANI22) and the theoretical A/D conversion result (stored in the 10-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$\text{SAR} = \text{INT}\left(\frac{V_{\text{AIN}}}{AV_{\text{REF}}} \times 1024 + 0.5\right)$$

$$\text{ADCR} = \text{SAR} \times 64$$

or

$$\left(\frac{\text{ADCR}}{64} - 0.5\right) \times \frac{AV_{\text{REF}}}{1024} \leq V_{\text{AIN}} < \left(\frac{\text{ADCR}}{64} + 0.5\right) \times \frac{AV_{\text{REF}}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses

VAIN: Analog input voltage

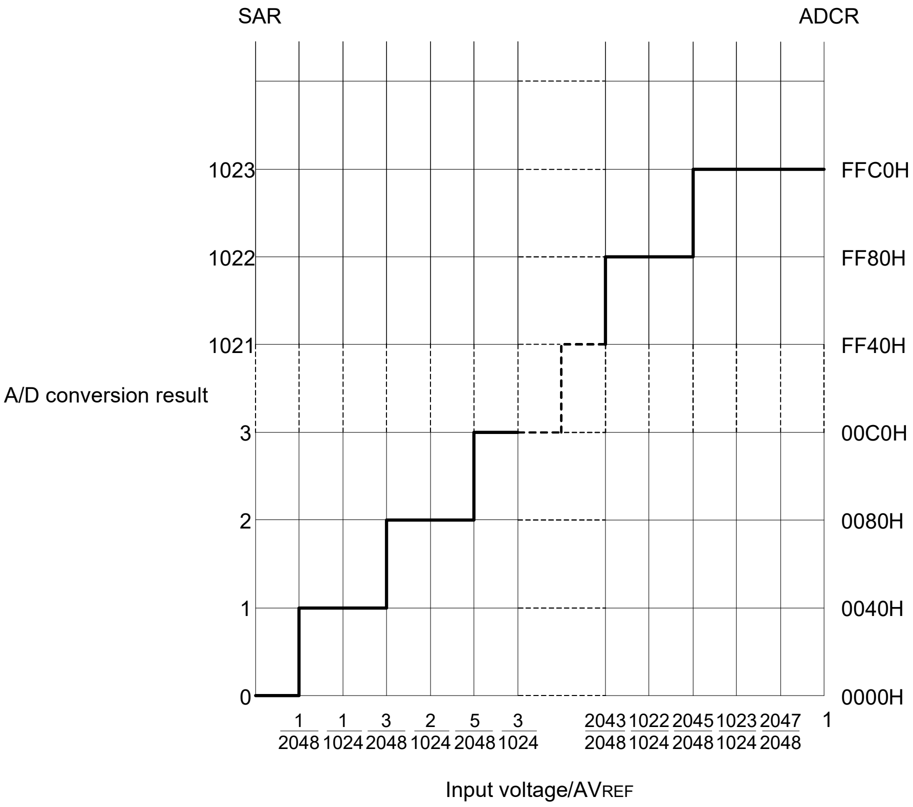
AVREF: AVREF pin voltage

ADCR: A/D conversion result register (ADCR) value

SAR: Successive approximation register

Figure 16 - 17 shows the Relationship Between Analog Input Voltage and A/D Conversion Result.

Figure 16 - 17 Relationship Between Analog Input Voltage and A/D Conversion Result



Remark AVREF: The + side reference voltage of the A/D converter. This can be selected from AVREFP, the internal reference voltage (1.45 V), and VDD.

16.6 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 **16.7 A/D Converter Setup Flowchart**.

16.6.1 Software trigger mode (select mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform the A/D conversion of the analog input specified by the analog input channel specification register (ADS).

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts.

<4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

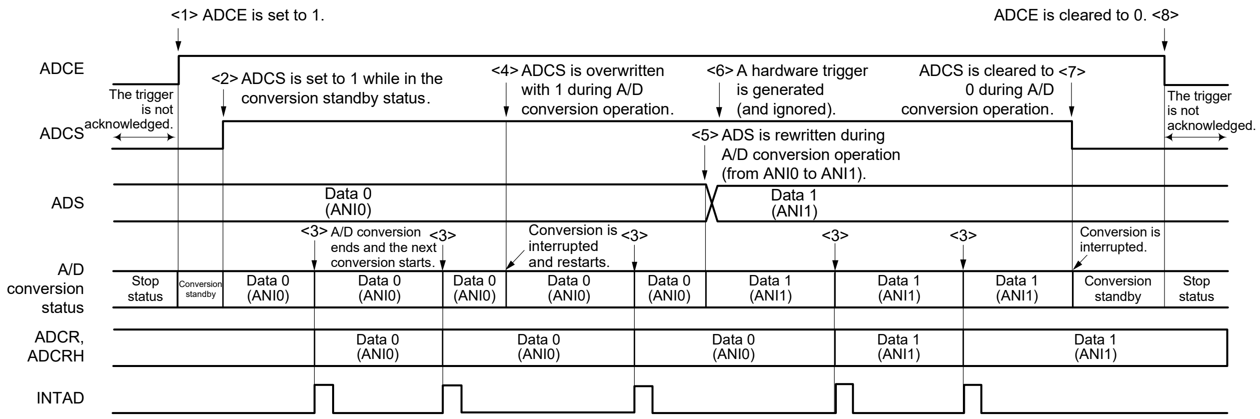
<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<6> Even if a hardware trigger is input during conversion operation, A/D conversion does not start.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 16 - 18 Example of Software Trigger Mode (Select Mode, Sequential Conversion Mode) Operation Timing



<1> ADCE is set to 1.
ADCE is cleared to 0. <8>
ADCE
The trigger is not acknowledged.
<2> ADCS is set to 1 while in the conversion standby status.
<4> ADCS is overwritten with 1 during A/D conversion operation.
<6> A hardware trigger is generated (and ignored).
ADCS is cleared to 0 during A/D conversion operation. <7>
The trigger is not acknowledged.
ADCS
<5> ADS is rewritten during A/D conversion operation (from ANI0 to ANI1).
ADS
Data 0 (ANI0)
Data 1 (ANI1)
<3> A/D conversion ends and the next conversion starts.
<3>
Conversion is interrupted and restarts.
<3>
<3>
<3>
Conversion is interrupted.
A/D conversion status
Stop status
Conversion standby
Data 0 (ANI0)
Data 0 (ANI0)
Data 0 (ANI0)
Data 0 (ANI0)
Data 0 (ANI0)
Data 1 (ANI1)
Data 1 (ANI1)
Data 1 (ANI1)
Conversion standby
Stop status
ADCR, ADCRH
Data 0 (ANI0)
Data 0 (ANI0)
Data 0 (ANI0)
Data 1 (ANI1)
Data 1 (ANI1)
INTAD

16.6.2 Software trigger mode (select mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 µs), the ADCS bit of the ADM0 register is set to 1 to perform the A/D conversion of the analog input specified by the analog input channel specification register (ADS).

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the system enters the A/D conversion standby status.

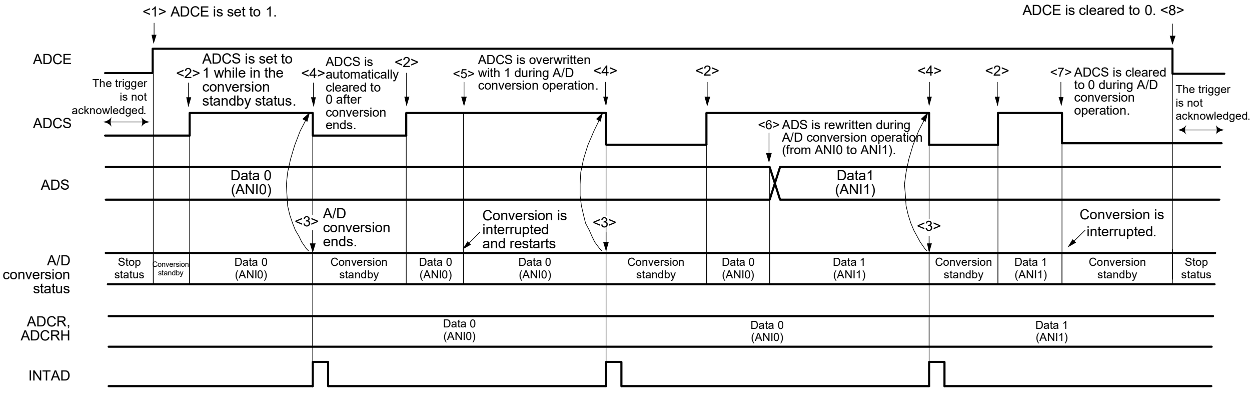
<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby status.

Figure 16 - 19 Example of Software Trigger Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



<1> ADCE is set to 1.
ADCE is cleared to 0. <8>
ADCE
ADCS
ADS
A/D conversion status
ADCR, ADCRH
INTAD
The trigger is not acknowledged.
<2> ADCS is set to 1 while in the conversion standby status.
<4> ADCS is automatically cleared to 0 after conversion ends.
<2>
<5> ADCS is overwritten with 1 during A/D conversion operation.
<4>
<2>
<4>
<2>
<7> ADCS is cleared to 0 during A/D conversion operation.
The trigger is not acknowledged.
<6> ADS is rewritten during A/D conversion operation (from ANI0 to ANI1).
Data 0 (ANI0)
Data1 (ANI1)
<3> A/D conversion ends.
Conversion is interrupted and restarts
<3>
<3>
Conversion is interrupted.
Stop status
Conversion standby
Data 0 (ANI0)
Conversion standby
Data 0 (ANI0)
Data 0 (ANI0)
Conversion standby
Data 0 (ANI0)
Data 1 (ANI1)
Conversion standby
Data 1 (ANI1)
Conversion standby
Stop status
Data 0 (ANI0)
Data 0 (ANI0)
Data 1 (ANI1)

16.6.3 Software trigger mode (scan mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts (until all four channels are finished).

<4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

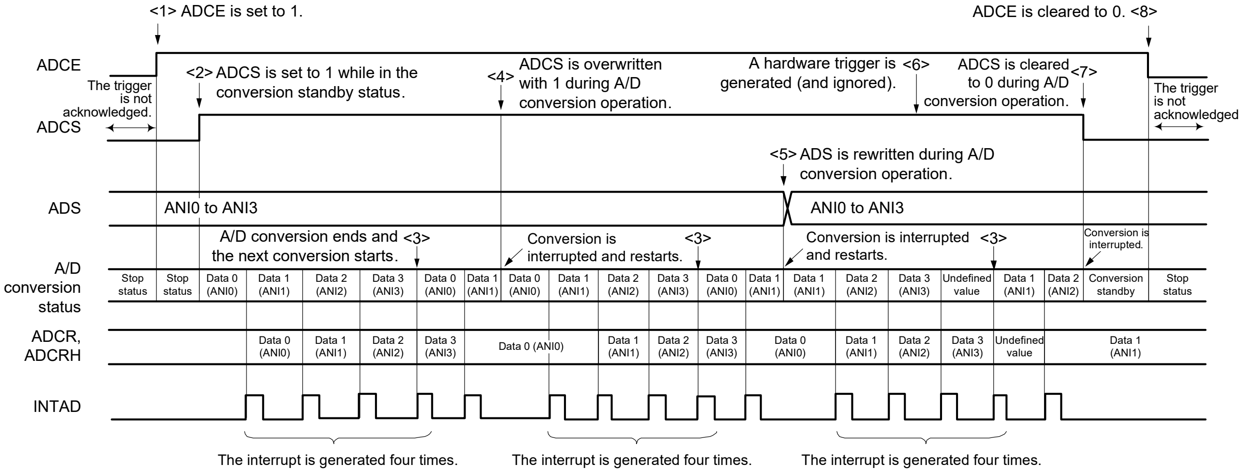
<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<6> Even if a hardware trigger is input during conversion operation, A/D conversion does not start.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 16 - 20 Example of Software Trigger Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



<1> ADCE is set to 1.
ADCE is cleared to 0. <8>
ADCE
The trigger is not acknowledged.
ADCS
<2> ADCS is set to 1 while in the conversion standby status.
<4>
ADCS is overwritten with 1 during A/D conversion operation.
A hardware trigger is generated (and ignored).
<6>
ADCS is cleared to 0 during A/D conversion operation.
<7>
The trigger is not acknowledged
<5> ADS is rewritten during A/D conversion operation.
ADS
ANI0 to ANI3
ANI0 to ANI3
A/D conversion ends and the next conversion starts.
<3>
Conversion is interrupted and restarts.
<3>
Conversion is interrupted and restarts.
<3>
Conversion is interrupted.
A/D conversion status
Stop status
Stop status
Data 0 (ANI0)
Data 1 (ANI1)
Data 2 (ANI2)
Data 3 (ANI3)
Data 0 (ANI0)
Data 1 (ANI1)
Data 0 (ANI0)
Data 1 (ANI1)
Data 2 (ANI2)
Data 3 (ANI3)
Data 0 (ANI0)
Data 1 (ANI1)
Data 1 (ANI1)
Data 2 (ANI2)
Data 3 (ANI3)
Undefined value
Data 1 (ANI1)
Data 2 (ANI2)
Conversion standby
Stop status
ADCR, ADCRH
Data 0 (ANI0)
Data 1 (ANI1)
Data 2 (ANI2)
Data 3 (ANI3)
Data 0 (ANI0)
Data 1 (ANI1)
Data 2 (ANI2)
Data 3 (ANI3)
Data 0 (ANI0)
Data 1 (ANI1)
Data 2 (ANI2)
Data 3 (ANI3)
Undefined value
Data 1 (ANI1)
INTAD
The interrupt is generated four times.
The interrupt is generated four times.
The interrupt is generated four times.

16.6.4 Software trigger mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion of the four channels ends, the ADCS bit is automatically cleared to 0, and the system enters the A/D conversion standby status.

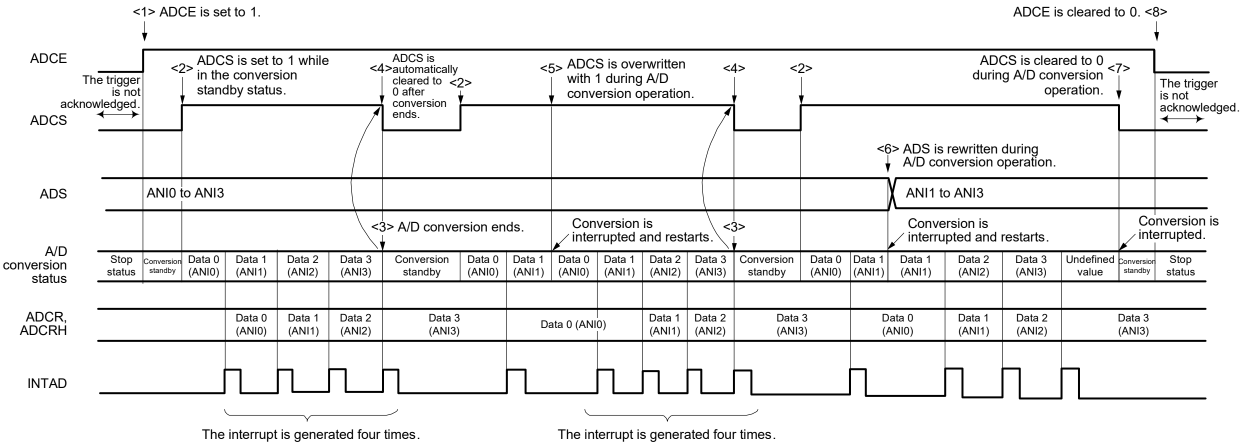
<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby status.

Figure 16 - 21 Example of Software Trigger Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing



16.6.5 Hardware trigger no-wait mode (select mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).

<4> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts.

<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

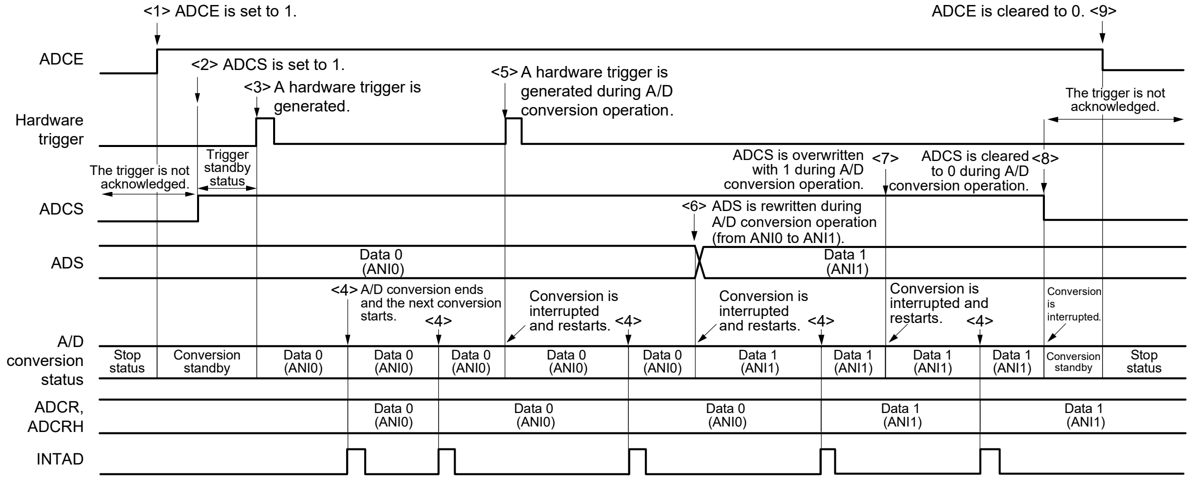
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<9> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 16 - 22 Example of Hardware Trigger No-Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing


<1> ADCE is set to 1.
ADCE is cleared to 0. <9>
ADCE
<2> ADCS is set to 1.
<3> A hardware trigger is generated.
<5> A hardware trigger is generated during A/D conversion operation.
The trigger is not acknowledged.
Hardware trigger
The trigger is not acknowledged.
Trigger standby status
ADCS is overwritten with 1 during A/D conversion operation. <7>
ADCS is cleared to 0 during A/D conversion operation. <8>
ADCS
<6> ADS is rewritten during A/D conversion operation (from ANI0 to ANI1).
ADS
Data 0 (ANI0)
Data 1 (ANI1)
<4> A/D conversion ends and the next conversion starts.
<4>
Conversion is interrupted and restarts.
<4>
Conversion is interrupted and restarts.
<4>
Conversion is interrupted and restarts.
<4>
Conversion is interrupted.
A/D conversion status
Stop status
Conversion standby
Data 0 (ANI0)
Data 0 (ANI0)
Data 0 (ANI0)
Data 0 (ANI0)
Data 0 (ANI0)
Data 1 (ANI1)
Data 1 (ANI1)
Data 1 (ANI1)
Data 1 (ANI1)
Conversion standby
Stop status
ADCR, ADCRH
Data 0 (ANI0)
Data 0 (ANI0)
Data 0 (ANI0)
Data 1 (ANI1)
Data 1 (ANI1)
INTAD

16.6.6 Hardware trigger no-wait mode (select mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).

<4> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<5> After A/D conversion ends, the ADCS bit remains set to 1, and the system enters the A/D conversion standby status.

<6> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

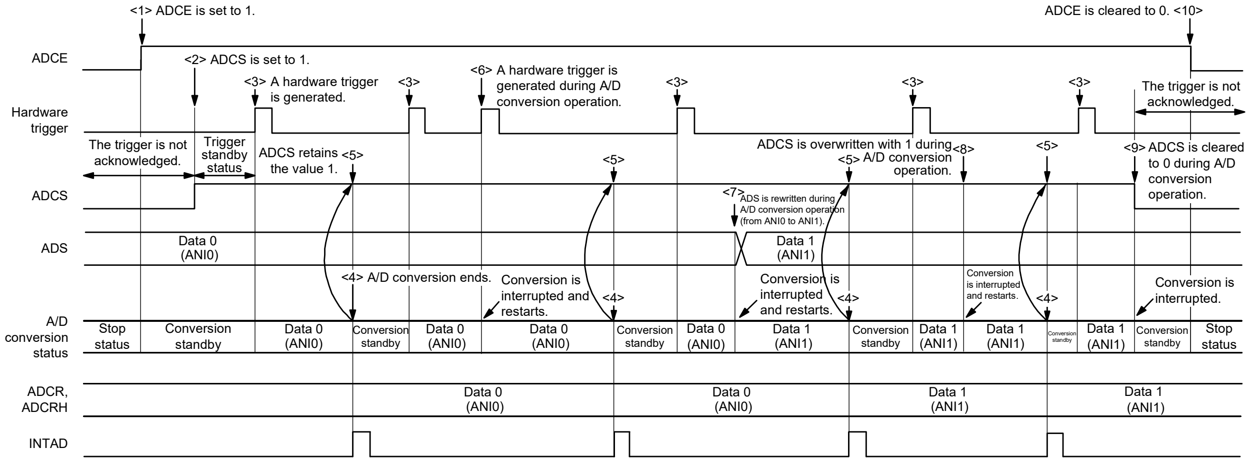
<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<9> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<10>When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 16 - 23 Example of Hardware Trigger No-Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



<1> ADCE is set to 1.
ADCE is cleared to 0. <10>
ADCE
<2> ADCS is set to 1.
<3> A hardware trigger is generated.
<3>
<6> A hardware trigger is generated during A/D conversion operation.
<3>
<3>
<3>
The trigger is not acknowledged.
Hardware trigger
The trigger is not acknowledged.
Trigger standby status
ADCS retains the value 1.
<5>
<5>
ADCS is overwritten with 1 during A/D conversion operation.
<5>
<8>
<5>
<9> ADCS is cleared to 0 during A/D conversion operation.
ADCS
<7> ADS is rewritten during A/D conversion operation (from ANI0 to ANI1).
ADS
Data 0 (ANI0)
Data 1 (ANI1)
<4> A/D conversion ends.
Conversion is interrupted and restarts.
<4>
Conversion is interrupted and restarts.
<4>
Conversion is interrupted and restarts.
<4>
Conversion is interrupted.
A/D conversion status
Stop status
Conversion standby
Data 0 (ANI0)
Conversion standby
Data 0 (ANI0)
Data 0 (ANI0)
Conversion standby
Data 0 (ANI0)
Data 1 (ANI1)
Conversion standby
Data 1 (ANI1)
Data 1 (ANI1)
Conversion standby
Data 1 (ANI1)
Conversion standby
Stop status
ADCR, ADCRH
Data 0 (ANI0)
Data 0 (ANI0)
Data 1 (ANI1)
Data 1 (ANI1)
INTAD

16.6.7 Hardware trigger no-wait mode (scan mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<4> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts.

<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

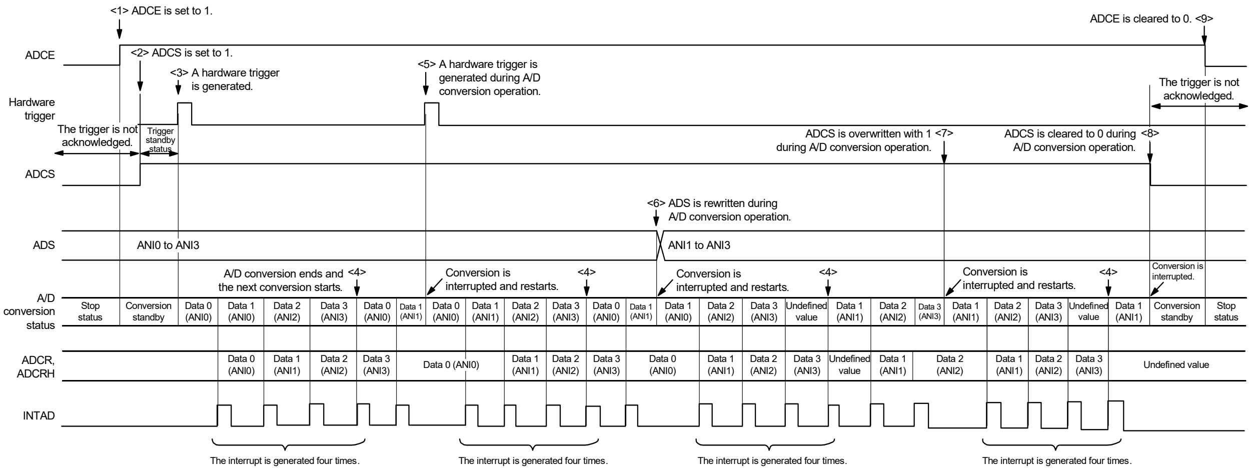
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<9> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 16 - 24 Example of Hardware Trigger No-Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



16.6.8 Hardware trigger no-wait mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<4> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<5> After A/D conversion of the four channels ends, the ADCS bit remains set to 1, and the system enters the A/D conversion standby status.

<6> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

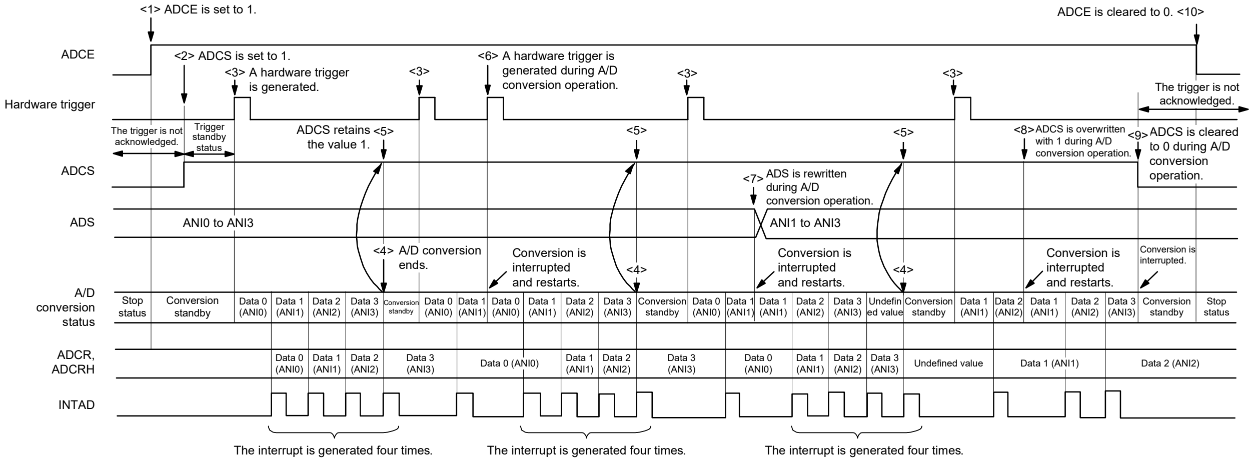
<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<9> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<10>When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 16 - 25 Example of Hardware Trigger No-Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing



16.6.9 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.)

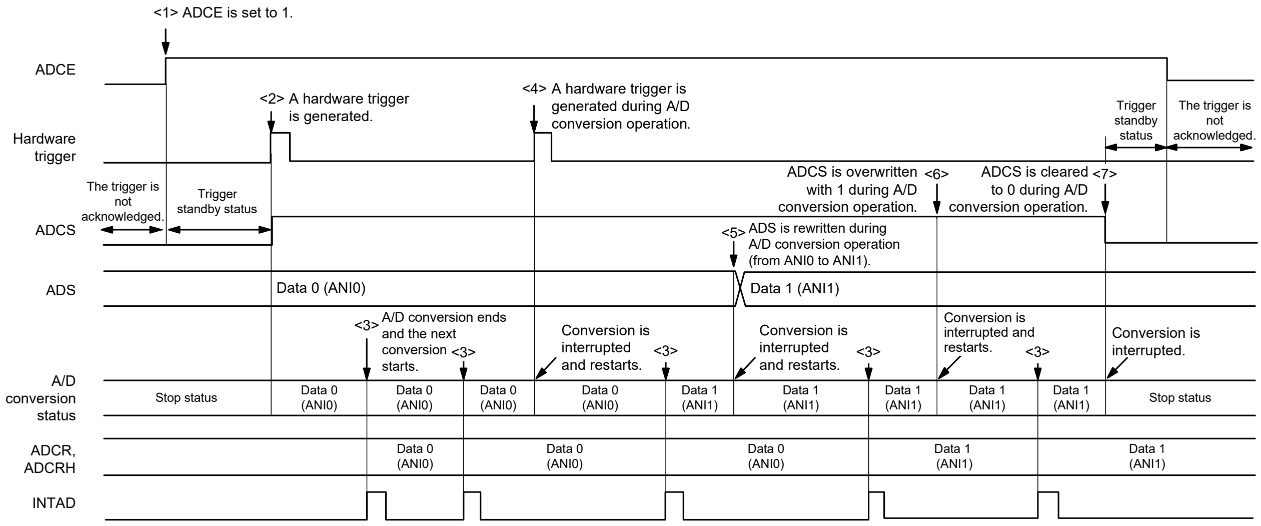
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<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 16 - 26 Example of Hardware Trigger Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



<1> ADCE is set to 1.
ADCE
Hardware trigger
<2> A hardware trigger is generated.
<4> A hardware trigger is generated during A/D conversion operation.
Trigger standby status
The trigger is not acknowledged.
ADCS is overwritten with 1 during A/D conversion operation. <6>
ADCS is cleared to 0 during A/D conversion operation. <7>
The trigger is not acknowledged.
Trigger standby status
ADCS
<5> ADS is rewritten during A/D conversion operation (from ANI0 to ANI1).
ADS
Data 0 (ANI0)
Data 1 (ANI1)
<3> A/D conversion ends and the next conversion starts.
<3>
Conversion is interrupted and restarts.
<3>
Conversion is interrupted and restarts.
<3>
Conversion is interrupted and restarts.
<3>
Conversion is interrupted.
A/D conversion status
Stop status
Data 0 (ANI0)
Data 0 (ANI0)
Data 0 (ANI0)
Data 0 (ANI0)
Data 1 (ANI1)
Data 1 (ANI1)
Data 1 (ANI1)
Data 1 (ANI1)
Data 1 (ANI1)
Stop status
ADCR, ADCRH
Data 0 (ANI0)
Data 0 (ANI0)
Data 0 (ANI0)
Data 1 (ANI1)
Data 1 (ANI1)
INTAD

16.6.10 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.

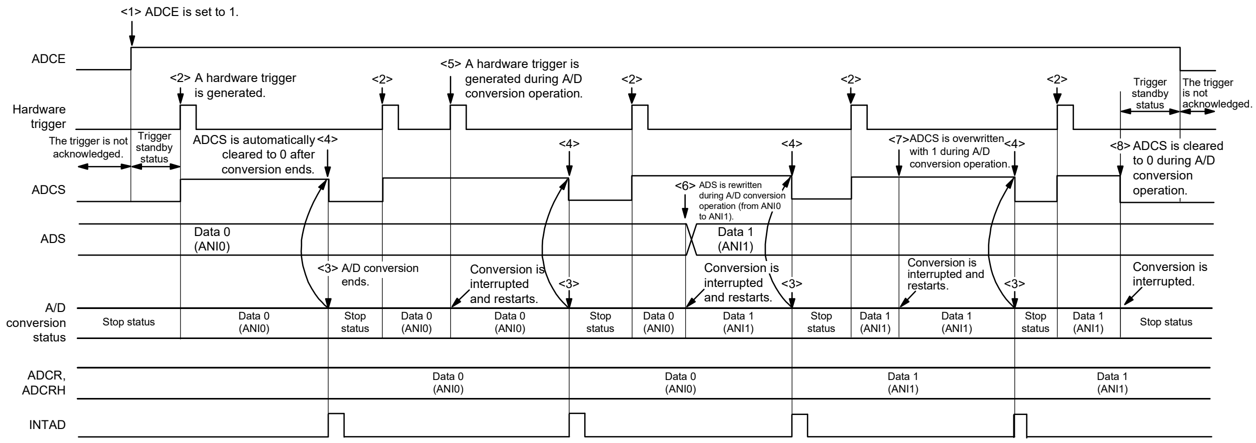
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<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 16 - 27 Example of Hardware Trigger Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing


<1> ADCE is set to 1.
ADCE
<2> A hardware trigger is generated.
<2>
<5> A hardware trigger is generated during A/D conversion operation.
<2>
<2>
<2>
Trigger standby status
The trigger is not acknowledged.
Hardware trigger
The trigger is not acknowledged.
Trigger standby status
ADCS is automatically cleared to 0 after conversion ends.
<4>
<4>
<4>
<7>ADCS is overwritten with 1 during A/D conversion operation.
<4>
<8> ADCS is cleared to 0 during A/D conversion operation.
ADCS
<6> ADS is rewritten during A/D conversion operation (from ANI0 to ANI1).
ADS
Data 0 (ANI0)
Data 1 (ANI1)
<3> A/D conversion ends.
Conversion is interrupted and restarts.
<3>
Conversion is interrupted and restarts.
<3>
Conversion is interrupted and restarts.
<3>
Conversion is interrupted.
A/D conversion status
Stop status
Data 0 (ANI0)
Stop status
Data 0 (ANI0)
Data 0 (ANI0)
Stop status
Data 0 (ANI0)
Data 1 (ANI1)
Stop status
Data 1 (ANI1)
Data 1 (ANI1)
Stop status
Data 1 (ANI1)
Stop status
ADCR, ADCRH
Data 0 (ANI0)
Data 0 (ANI0)
Data 1 (ANI1)
Data 1 (ANI1)
INTAD

16.6.11 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.

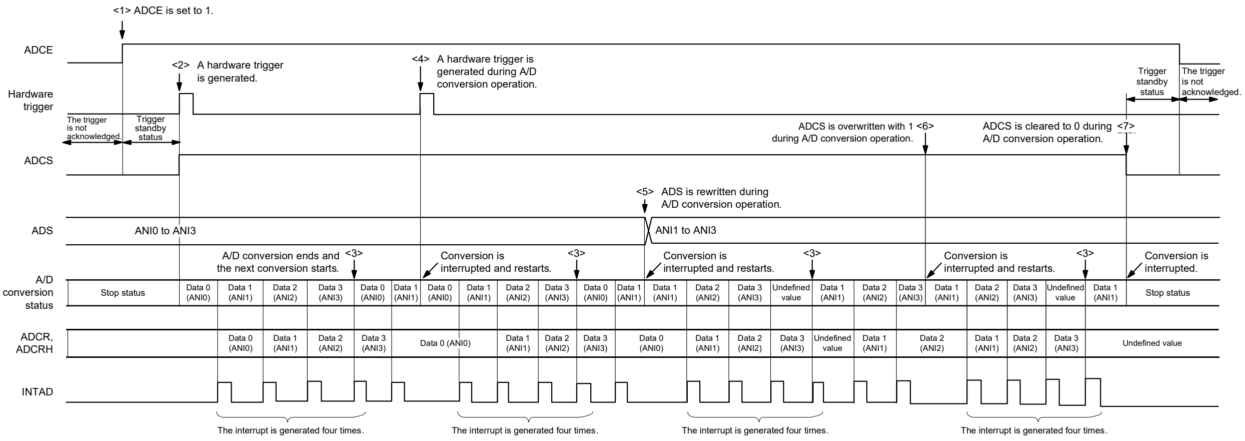
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<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 16 - 28 Example of Hardware Trigger Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



16.6.12 Hardware trigger wait mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input. A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the stop status.

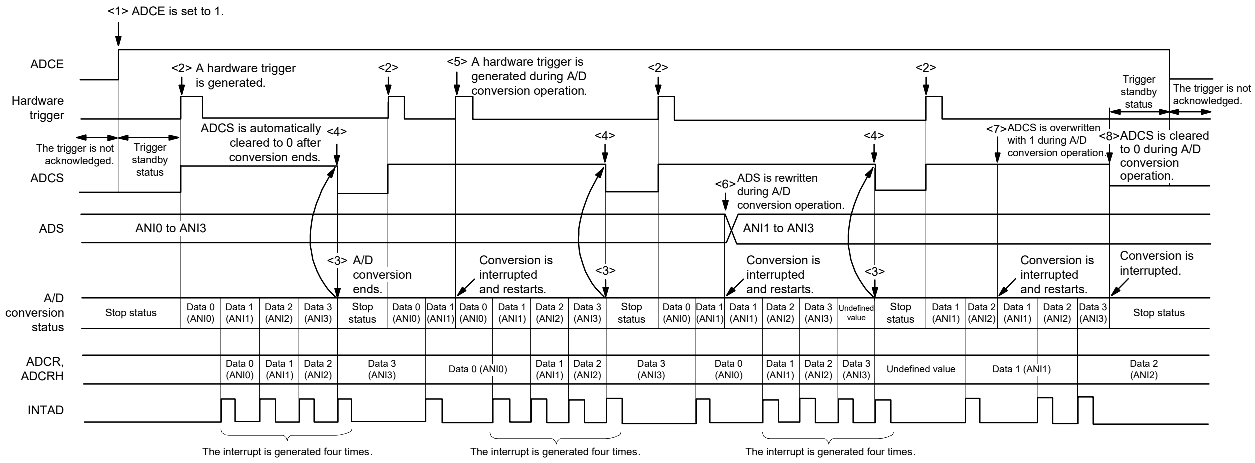
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 16 - 29 Example of Hardware Trigger Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing

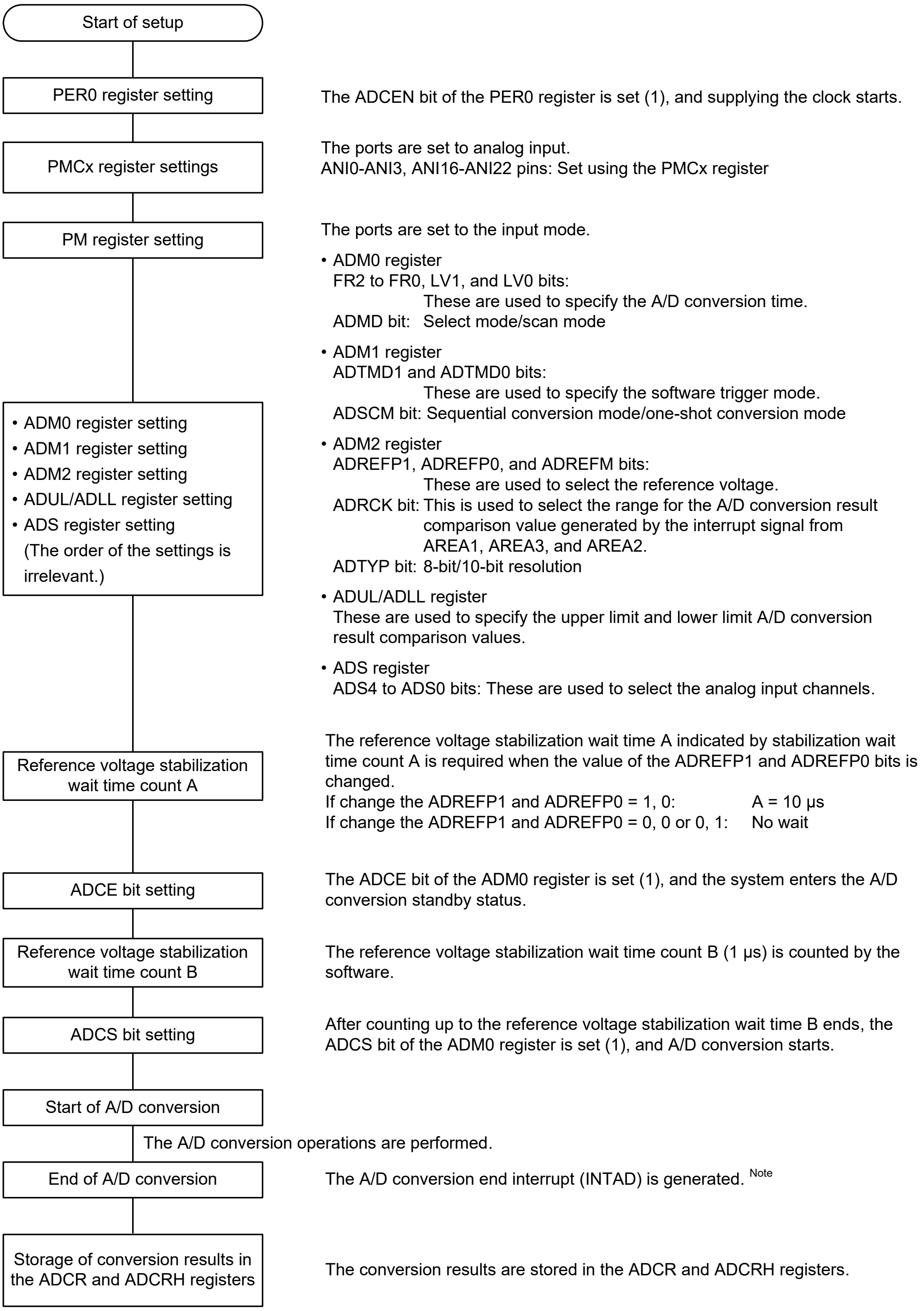


16.7 A/D Converter Setup Flowchart

The A/D converter setup flowchart in each operation mode is described below.

16.7.1 Setting up software trigger mode

Figure 16 - 30 Setting up Software Trigger Mode



The ADCEN bit of the PER0 register is set (1), and supplying the clock starts.

The ports are set to analog input.
ANI0-ANI3, ANI16-ANI22 pins: Set using the PMCx register

The ports are set to the input mode.

- 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.
  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/10-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.

The reference voltage stabilization wait time A indicated by 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: No wait

The ADCE bit of the ADM0 register is set (1), and the system enters the A/D conversion standby status.

The reference voltage stabilization wait time count B (1 μs) is counted by the software.

After counting up to the reference voltage stabilization wait time B ends, the ADCS bit of the ADM0 register is set (1), and A/D conversion starts.

The A/D conversion operations are performed.

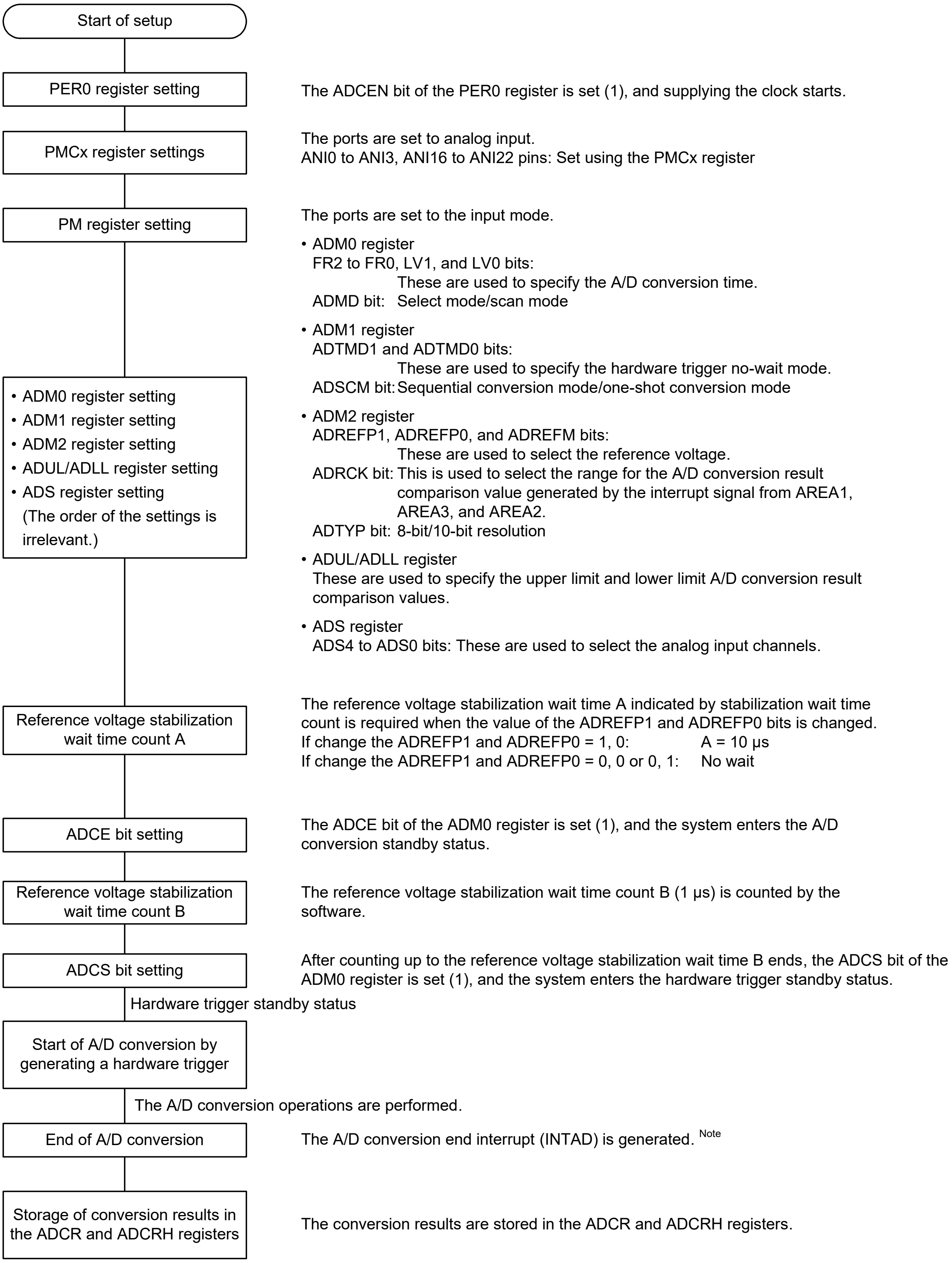
The A/D conversion end interrupt (INTAD) is generated. Note

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 interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH register.

16.7.2 Setting up hardware trigger no-wait mode

Figure 16 - 31 Setting up Hardware Trigger No-Wait Mode


Start of setup
PER0 register setting
PMCx register settings
PM register setting
• ADM0 register setting
• ADM1 register setting
• ADM2 register setting
• ADUL/ADLL register setting
• ADS register setting
(The order of the settings is irrelevant.)
Reference voltage stabilization wait time count A
ADCE bit setting
Reference voltage stabilization wait time count B
ADCS bit setting
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
Storage of conversion results in the ADCR and ADCRH registers

The ADCEN bit of the PER0 register is set (1), and supplying the clock starts.

The ports are set to analog input.
ANI0 to ANI3, ANI16 to ANI22 pins: Set using the PMCx register

The ports are set to the input mode.

- 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.
  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/10-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.

The reference voltage stabilization wait time A indicated by stabilization wait time count 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: No wait

The ADCE bit of the ADM0 register is set (1), and the system enters the A/D conversion standby status.

The reference voltage stabilization wait time count B (1 μs) is counted by the software.

After counting up to the reference voltage stabilization wait time B ends, the ADCS bit of the ADM0 register is set (1), and the system enters the hardware trigger standby status.

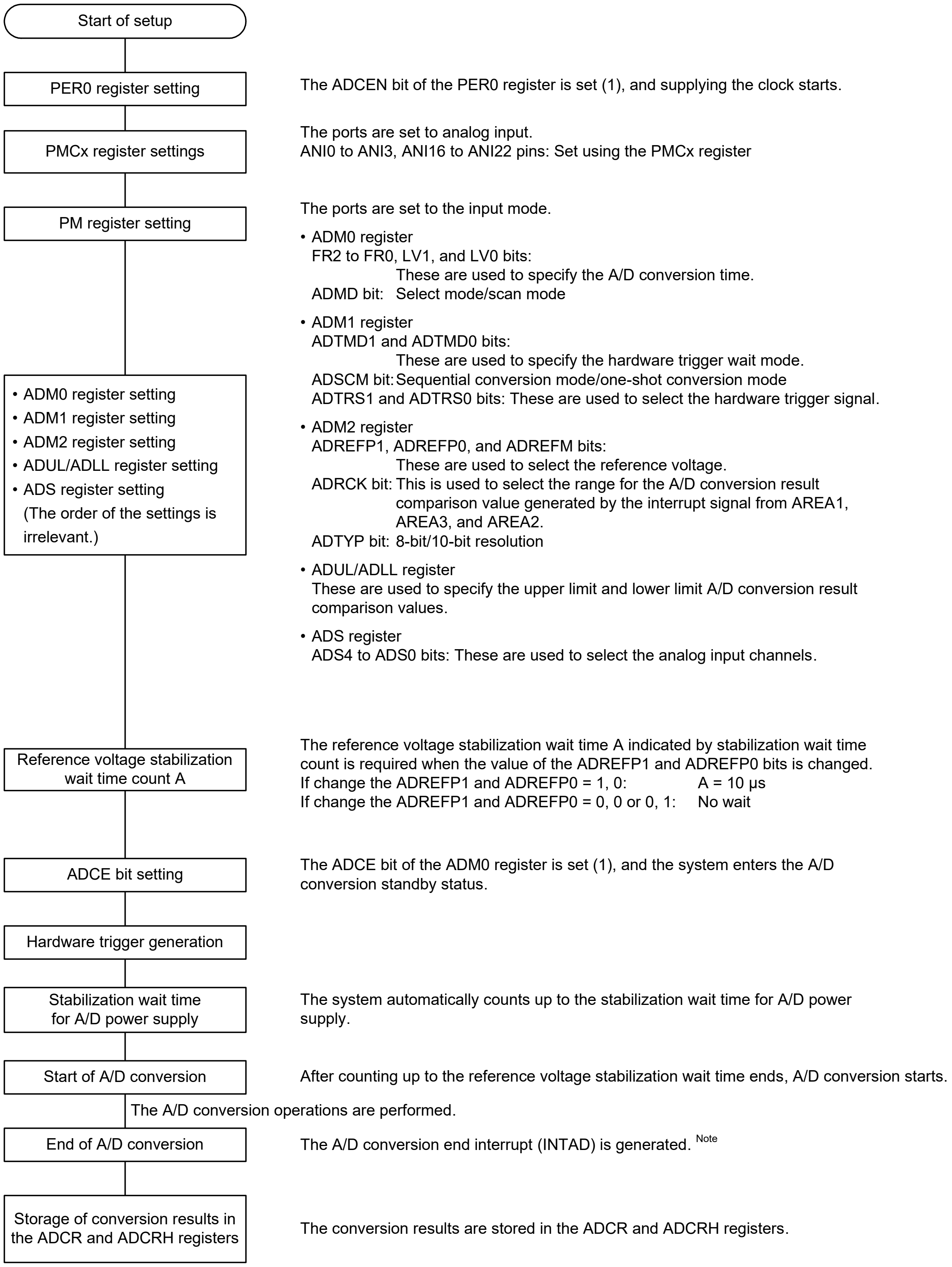
The A/D conversion end interrupt (INTAD) is generated. [Note]

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 interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH register.

16.7.3 Setting up hardware trigger wait mode

Figure 16 - 32 Setting up Hardware Trigger Wait Mode


Start of setup
PER0 register setting
PMCx register settings
PM register setting
• ADM0 register setting
• ADM1 register setting
• ADM2 register setting
• ADUL/ADLL register setting
• ADS register setting
(The order of the settings is irrelevant.)
Reference voltage stabilization wait time count A
ADCE bit setting
Hardware trigger generation
Stabilization wait time for A/D power supply
Start of A/D conversion
End of A/D conversion
Storage of conversion results in the ADCR and ADCRH registers

The ADCEN bit of the PER0 register is set (1), and supplying the clock starts.

The ports are set to analog input.
ANI0 to ANI3, ANI16 to ANI22 pins: Set using the PMCx register

The ports are set to the input mode.

- 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 wait mode.
  ADSCM bit: Sequential conversion mode/one-shot conversion mode
  ADTRS1 and ADTRS0 bits: These are used to select the hardware trigger signal.
- ADM2 register
  ADREFP1, ADREFP0, and ADREFM bits:
  These are used to select the reference voltage.
  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/10-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.

The reference voltage stabilization wait time A indicated by stabilization wait time count 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: No wait

The ADCE bit of the ADM0 register is set (1), and the system enters the A/D conversion standby status.

The system automatically counts up to the stabilization wait time for A/D power supply.

After counting up to the reference voltage stabilization wait time ends, A/D conversion starts.

The A/D conversion operations are performed.

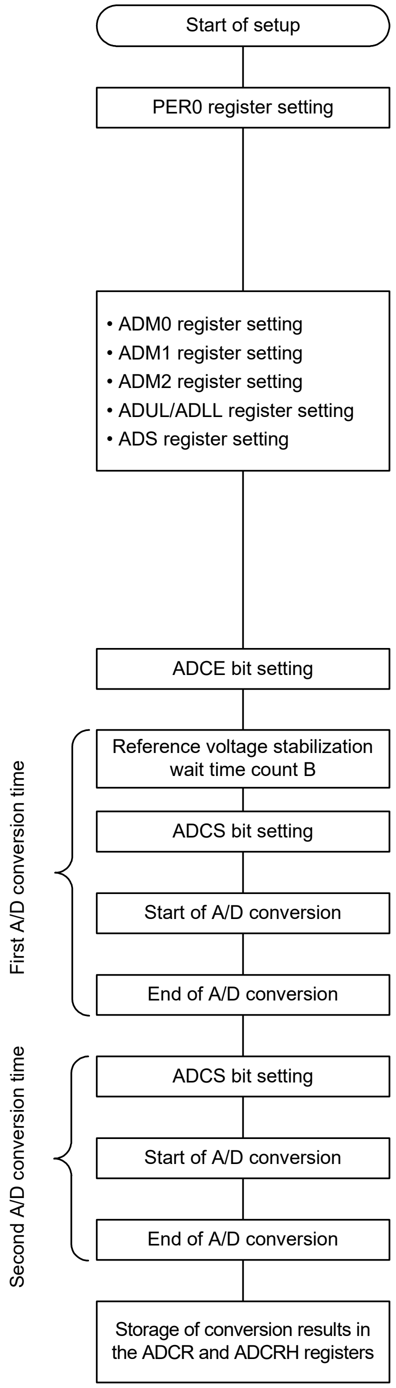
The A/D conversion end interrupt (INTAD) is generated. Note

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 interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH register.

16.7.4 Setup when temperature sensor output voltage/internal reference voltage is selected (example for software trigger mode and one-shot conversion mode)

Figure 16 - 33 Setup when temperature sensor output voltage/internal reference voltage is selected



The ADCEN bit of the PER0 register is set (1), and supplying the clock starts.

- ADM0 register
  FR2 to FR0, LV1, and LV0 bits: These are used to specify the A/D conversion time.
  ADMD bit: This is used to specify the select mode.
- ADM1 register
  ADTMD1 and ADTMD0 bits: These are used to specify the software trigger mode.
  ADSCM bit: One-shot conversion mode
- ADM2 register
  ADREFP1, ADREFP0, and ADREFM bits: These are used to select the reference voltage.
  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/10-bit resolution
- ADUL/ADLL register
  These are used to specify the upper limit and lower limit A/D conversion result comparison values.
- ADS register
  ADISS and ADS4 to ADS0 bits: These are used to select the temperature sensor output voltage or internal reference voltage.

The ADCE bit of the ADM0 register is set (1), and the system enters the A/D conversion standby status.

The reference voltage stabilization wait time count B (1 μs) is counted by the software.

After counting up to the reference voltage stabilization wait time B ends, the ADCS bit of the ADM0 register is set (1), and A/D conversion starts

The A/D conversion end interrupt (INTAD) will be generated.
After ADISS is set (1), the initial conversion result cannot be used.

The ADCS bit of the ADM0 register is set (1), and A/D conversion starts.

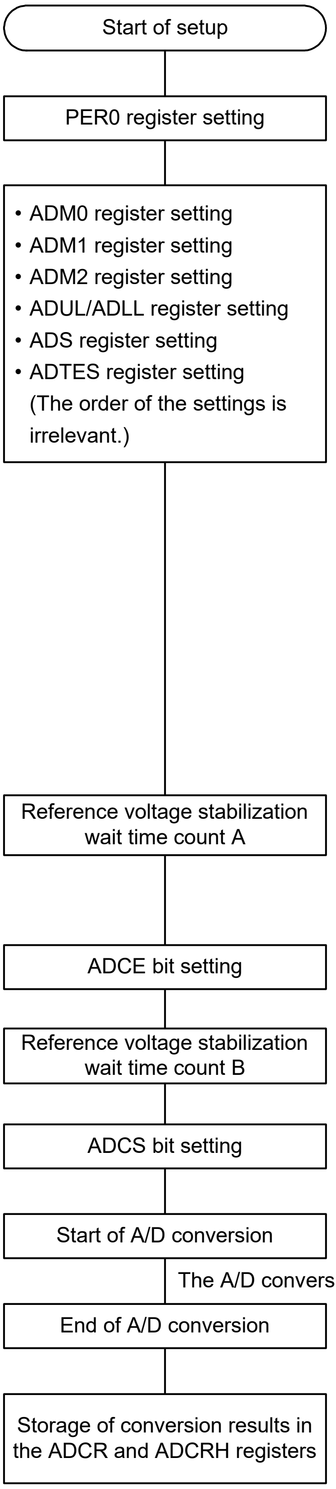
The A/D conversion end interrupt (INTAD) is generated. [Note]

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 interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH register.

16.7.5 Setting up test mode

Figure 16 - 34 Setting up Test Mode


Start of setup
PER0 register setting
• ADM0 register setting
• ADM1 register setting
• ADM2 register setting
• ADUL/ADLL register setting
• ADS register setting
• ADTES register setting
(The order of the settings is irrelevant.)
Reference voltage stabilization wait time count A
ADCE bit setting
Reference voltage stabilization wait time count B
ADCS bit setting
Start of A/D conversion
End of A/D conversion
Storage of conversion results in the ADCR and ADCRH registers

The ADCEN bit of the PER0 register is set (1), and supplying the clock starts.

- ADM0 register
  FR2 to FR0, LV1, and LV0 bits: These are used to specify the A/D conversion time.
  ADMD bit: This is used to specify the select mode
- ADM1 register
  ADTMD1 and ADTMD0 bits: These are used to specify the software trigger mode.
  ADSCM bit: This is used to specify the one-shot conversion mode.
- ADM2 register
  ADREFP1, ADREFP0, and ADREFM bits: These are used to select for the reference voltage.
  ADRCK bit: This is used to set the range for the A/D conversion result comparison value generated by the interrupt signal to AREA2.
  ADTYP bit: This is used to specify 10-bit resolution.
- ADUL/ADLL register
  These set ADUL to FFH and ADLL to 00H (initial values).
- ADS register
  ADS4 to ADS0 bits: These are used to set to ANI0.
- ADTES register
  ADTES1, ADTES0 bits: AV$_{\text{REFM}}$/AV$_{\text{REFP}}$

The reference voltage stabilization wait time A indicated by stabilization wait time count 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: No wait

The ADCE bit of the ADM0 register is set (1), and the system enters the A/D conversion standby status.

The reference voltage stabilization wait time count B (1 μs) is counted by the software.

After counting up to the reference voltage stabilization wait time B ends, the ADCS bit of the ADM0 register is set (1), and A/D conversion starts.

The A/D conversion operations are performed.

The A/D conversion end interrupt (INTAD) is generated. Note

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 interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH register.

Caution For the procedure for testing the A/D converter, see 28.3.8 A/D test function.

16.8 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.

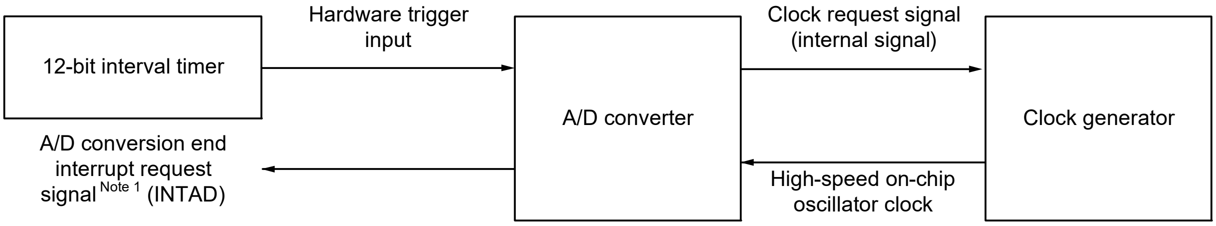
If the A/D conversion result range is specified using the ADUL and ADLL registers, A/D conversion results can be judged at a certain interval of time in SNOOZE mode. Using this function enables power supply voltage monitoring and input key judgment based on A/D inputs.

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 That 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 16 - 35 Block Diagram When Using SNOOZE Mode Function


12-bit interval timer
Hardware trigger input
A/D converter
Clock request signal (internal signal)
Clock generator
A/D conversion end interrupt request signal Note 1 (INTAD)
High-speed on-chip oscillator clock

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 **16.7.3 Setting up hardware trigger wait mode** Note 2.) At this time, bit 2 (AWC) of A/D converter mode register 2 (ADM2) is set to 1. After the initial settings are specified, bit 0 (ADCE) of A/D converter mode register 0 (ADM0) is set to 1.

If a hardware trigger is input after switching to the STOP mode, the high-speed on-chip oscillator clock is supplied to the A/D converter. After supplying this clock, the system automatically counts up to the 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 The hardware trigger is event selected by ELC, INTIT.
Specify the hardware trigger by using the 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.

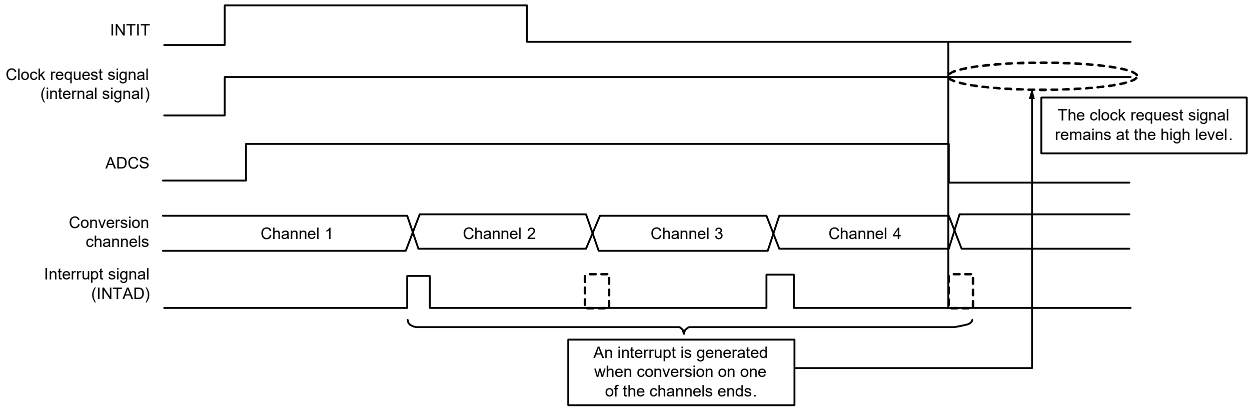
- While in the select mode

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 the A/D converter mode register 2 (ADM2). If the AWC bit is left set to 1, A/D conversion will not start normally in the subsequent SNOOZE or normal operation mode.

- While in the scan mode

If even one A/D conversion end interrupt request signal (INTAD) is generated during A/D conversion of the four channels, the 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 16 - 36 Operation Example When Interrupt Is Generated After A/D Conversion Ends (While in Scan Mode)


INTIT
Clock request signal (internal signal)
ADCS
Conversion channels
Channel 1
Channel 2
Channel 3
Channel 4
Interrupt signal (INTAD)
The clock request signal remains at the high level.
An interrupt is generated when conversion on one of the channels ends.

(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 16 - 37 Operation Example When No Interrupt Is Generated After A/D Conversion Ends (While in Scan Mode)

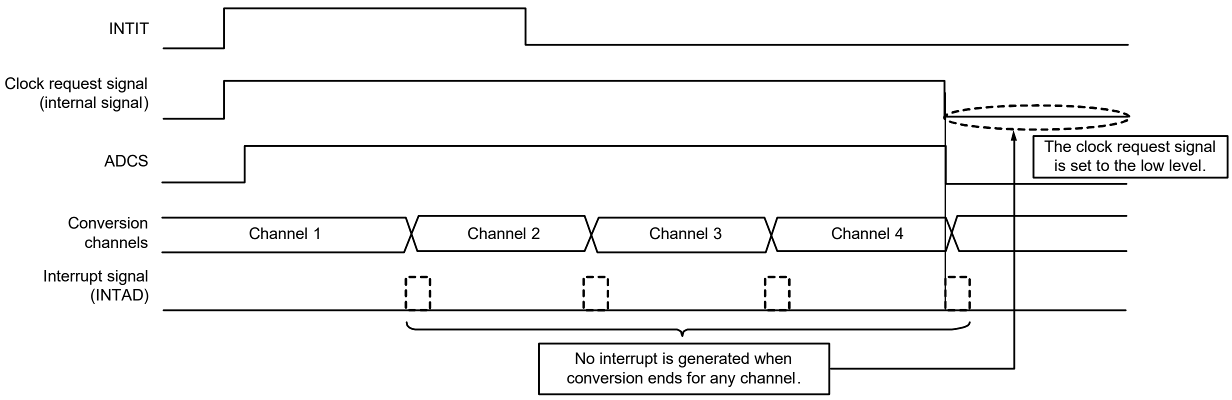
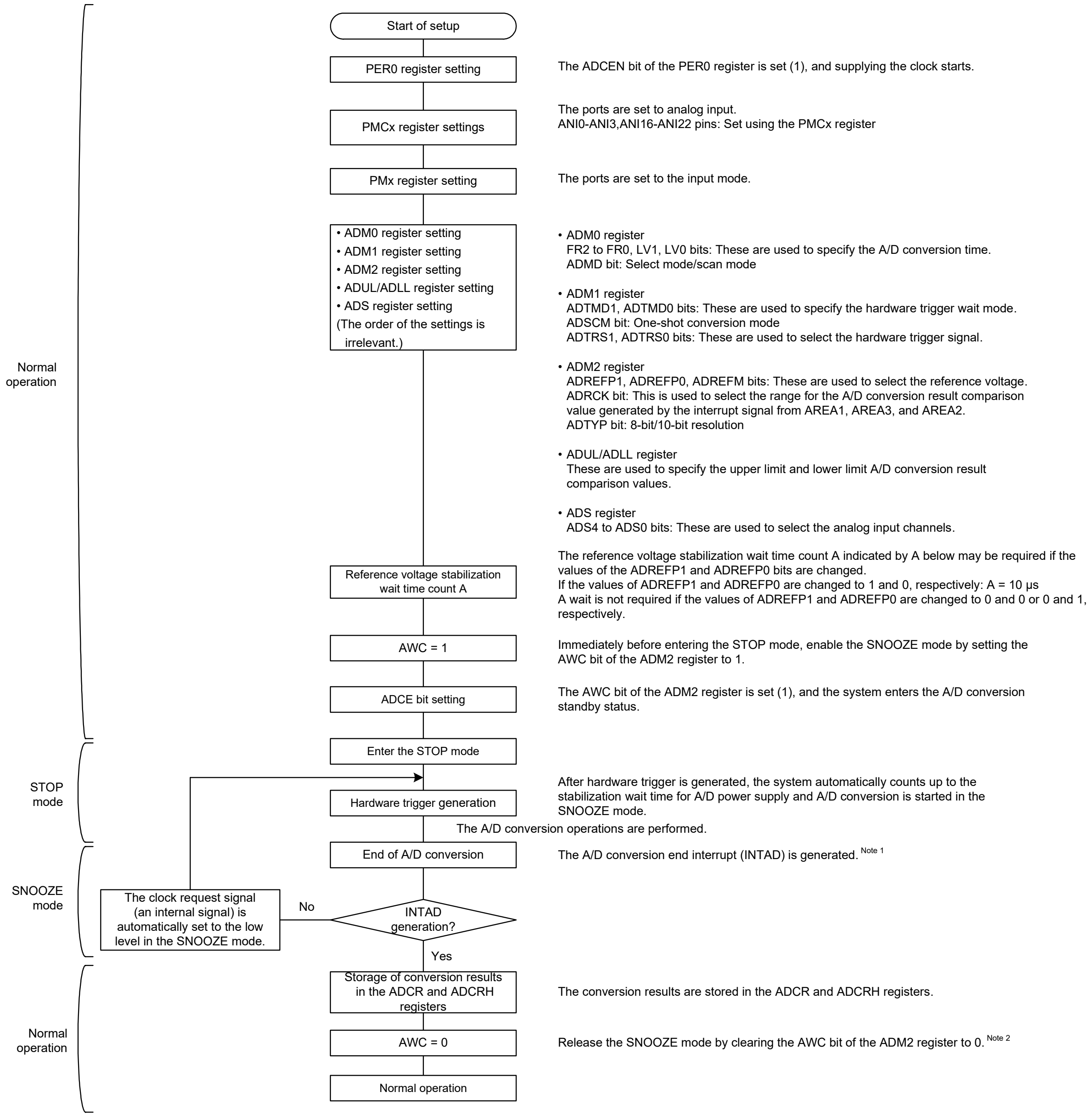

INTIT
Clock request signal (internal signal)
ADCS
Conversion channels
Interrupt signal (INTAD)
Channel 1
Channel 2
Channel 3
Channel 4
The clock request signal is set to the low level.
No interrupt is generated when conversion ends for any channel.

Figure 16 - 38 Flowchart for Setting up SNOOZE Mode



The ADCEN bit of the PER0 register is set (1), and supplying the clock starts.

The ports are set to analog input.
ANI0-ANI3,ANI16-ANI22 pins: Set using the PMCx register

The ports are set to the input mode.

- ADM0 register
  FR2 to FR0, LV1, LV0 bits: These are used to specify the A/D conversion time.
  ADMD bit: Select mode/scan mode
- ADM1 register
  ADTMD1, ADTMD0 bits: These are used to specify the hardware trigger wait mode.
  ADSCM bit: One-shot conversion mode
  ADTRS1, ADTRS0 bits: These are used to select the hardware trigger signal.
- ADM2 register
  ADREFP1, ADREFP0, ADREFM bits: These are used to select the reference voltage.
  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/10-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.

The reference voltage stabilization wait time count A indicated by A below may be required if the values of the ADREFP1 and ADREFP0 bits are changed.
If the values of ADREFP1 and ADREFP0 are changed to 1 and 0, respectively: A = 10 μs
A wait is not required if the values of ADREFP1 and ADREFP0 are changed to 0 and 0 or 0 and 1, respectively.

Immediately before entering the STOP mode, enable the SNOOZE mode by setting the AWC bit of the ADM2 register to 1.

The AWC bit of the ADM2 register is set (1), and the system enters the A/D conversion standby status.

After hardware trigger is generated, the system automatically counts up to the stabilization wait time for A/D power supply and A/D conversion is started in the SNOOZE mode.

The A/D conversion operations are performed.

The A/D conversion end interrupt (INTAD) is generated. Note 1

The conversion results are stored in the ADCR and ADCRH registers.

Release the SNOOZE mode by clearing the AWC bit of the ADM2 register to 0. Note 2

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.

16.9 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.

$1\text{ LSB} = 1/2^{10} = 1/1024$
$= 0.098\%\text{FSR}$

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 16 - 39 Overall Error

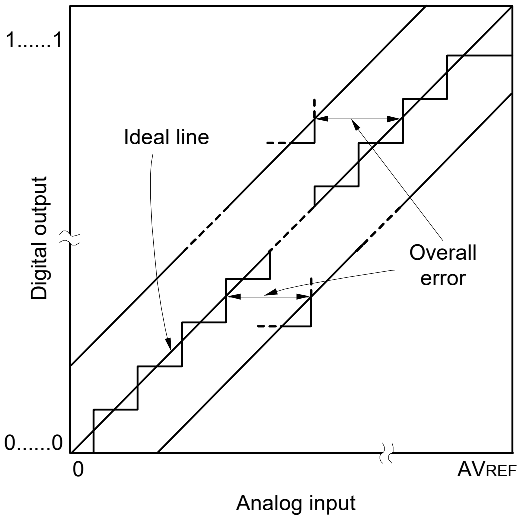
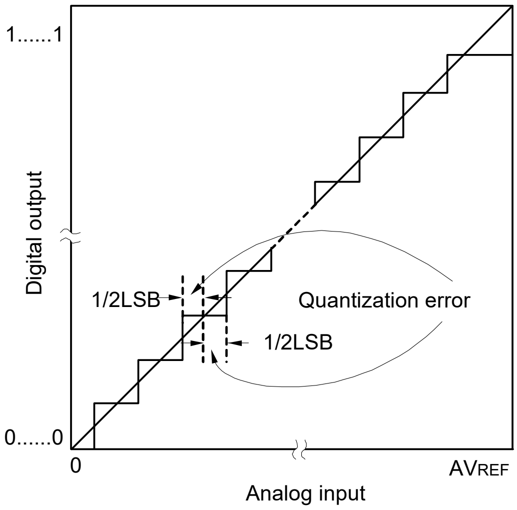


Figure 16 - 40 Quantization Error



(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 16 - 41 Zero-Scale Error

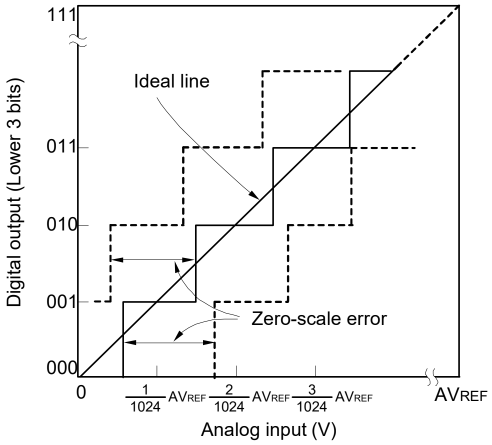


Figure 16 - 42 Full-Scale Error

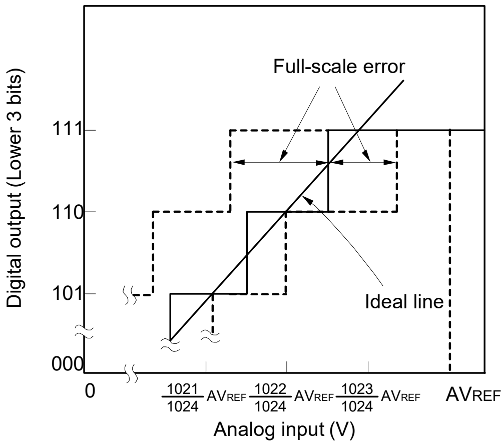


Figure 16 - 43 Integral Linearity Error

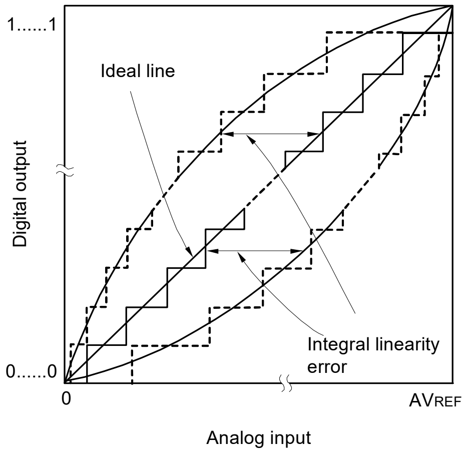
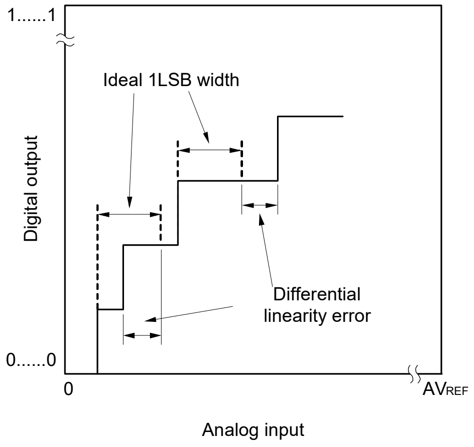


Figure 16 - 44 Differential Linearity Error



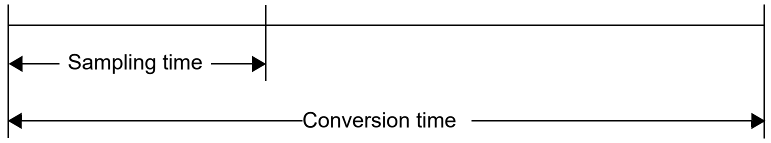
(8) Conversion time

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.



16.10 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 ANI3 and ANI16 to ANI22

Observe the rated range of the ANI0 to ANI3 and ANI16 to ANI22 pins input voltage. If a voltage exceeding V_{DD} and AV_{REFP} or below V_{SS} and AV_{REFM} (even in the range of absolute maximum ratings) is input to an analog input channel, the converted value of that channel becomes undefined. In addition, the converted values of the other channels may also be affected.

When internal reference voltage (1.45 V) is selected reference voltage for the + side of the A/D converter, do not input voltage exceeding internal reference voltage (1.45 V) to a pin selected by the ADS register. However, it is no problem that a pin not selected by the ADS register is input voltage exceeding the internal reference voltage (1.45 V).

(3) Conflicting operations

<1> Conflict between the A/D conversion result register (ADCR, ADCRH) write and the ADCR or ADCRH register read by instruction upon the end of conversion

The ADCR or ADCRH register read has priority. After the read operation, the new conversion result is written to the ADCR or ADCRH registers.

<2> Conflict between the ADCR or ADCRH register write and the A/D converter mode register 0 (ADM0) write, the analog input channel specification register (ADS) write upon the end of conversion

The ADM0 and ADS registers write has priority. The ADCR or ADCRH register write is not performed, nor is the conversion end interrupt signal (INTAD) generated.

(4) Noise countermeasures

To maintain the 10-bit resolution, attention must be paid to noise input to the AV_{REFP}, V_{DD}, ANI0 to ANI3 and ANI16 to ANI22 pins.

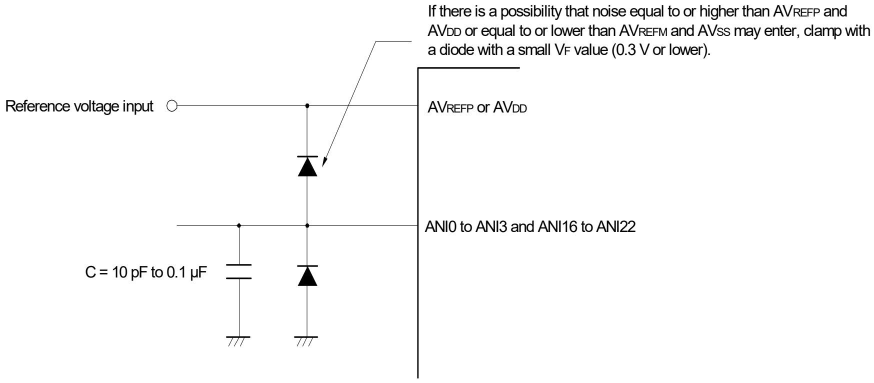
<1> Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.

<2> The higher the output impedance of the analog input source, the greater the influence. To reduce the noise, connecting external capacitor as shown in Figure 16 - 45 is recommended.

<3> Do not switch these pins with other pins during conversion.

<4> The accuracy is improved if the HALT mode is set immediately after the start of conversion.

Figure 16 - 45 Analog Input Pin Connection


If there is a possibility that noise equal to or higher than AVREFP and AVDD or equal to or lower than AVREFM and AVSS may enter, clamp with a diode with a small VF value (0.3 V or lower).
Reference voltage input
AVREFP or AVDD
ANI0 to ANI3 and ANI16 to ANI22
C = 10 pF to 0.1 μF

(5) Analog input (ANIn) pins

<1> The analog input pins (ANI0 to ANI3 and ANI16 to ANI22) are also used as input port pins (P20 to P23, P01, P00, P33 to P30, P56).

When A/D conversion is performed with any of the ANI0 to ANI3, ANI16 to ANI22 pins selected, do not change to output value P20 to P23, P01, P00, P33 to P30 and P56 while conversion is in progress; otherwise the conversion resolution may be degraded.

<2> If a pin adjacent to a pin that is being A/D converted is used as a digital I/O port pin, the A/D conversion result might differ from the expected value due to a coupling noise. Be sure to prevent such a pulse from being input or output.

(6) Input impedance of analog input (ANIn) pins

This A/D converter charges a sampling capacitor for sampling during sampling time.

Therefore, only a leakage current flows when sampling is not in progress, and a current that charges the capacitor flows during sampling. Consequently, the input impedance fluctuates depending on whether sampling is in progress, and on the other states.

To make sure that sampling is effective, however, we recommend using the converter with analog input sources that have output impedances no greater than 1 kΩ. If a source has a higher output impedance, lengthen the sampling time or connect a larger capacitor (with a value of about 0.1 μF) to the pin from among ANI0 to ANI3 and ANI16 to ANI22 to which the source is connected (see **Figure 16 - 45**). The sampling capacitor may be being charged while the setting of the ADCS bit is 0 and immediately after sampling is restarted and so is not defined at these times. Accordingly, the state of conversion is undefined after charging starts in the next round of conversion after the value of the ADCS bit has been 1 or when conversion is repeated. Thus, to secure full charging regardless of the size of fluctuations in the analog signal, ensure that the output impedances of the sources of analog inputs are low or secure sufficient time for the completion of conversion.

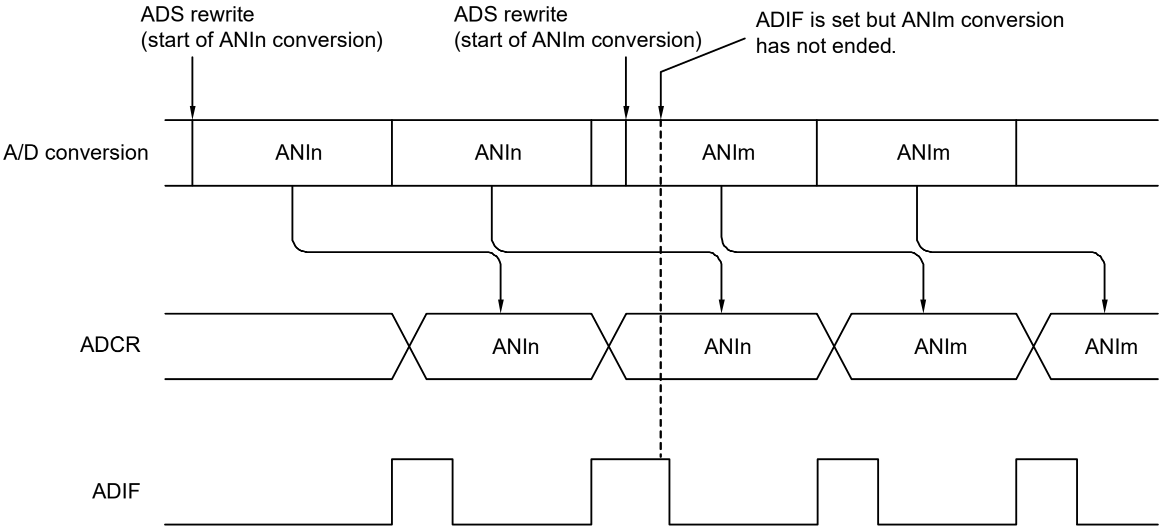
(7) Interrupt request flag (ADIF)

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 16 - 46 Timing of A/D Conversion End Interrupt Request Generation



(8) Conversion results just after A/D conversion start

While in the software trigger mode or hardware trigger no-wait mode, the first A/D conversion value immediately after A/D conversion starts may not fall within the rating range if the ADCS bit is set to 1 within 1 μs after the ADCE bit was set to 1. Take measures such as polling the A/D conversion end interrupt request (INTAD) and removing the first conversion result.

(9) A/D conversion result register (ADCR, ADCRH) read operation

When a write operation is performed to A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), and port mode control register (PMCxx), the contents of the ADCR and ADCRH registers may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, or PMC register. Using a timing other than the above may cause an incorrect conversion result to be read.

(10) Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 16 - 47 Internal Equivalent Circuit of ANIn Pin

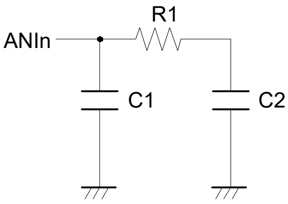


Table 16 - 4 Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

AVREFP, VDD	ANIn Pins	R1 [kΩ]	C1 [pF]	C2 [pF]
3.6 V ≤ VDD ≤ 5.5 V	ANI0 to ANI3	14	8	2.5
	ANI16 to ANI22	18	8	7.0
2.7 V ≤ VDD < 3.6 V	ANI0 to ANI3	39	8	2.5
	ANI16 to ANI22	53	8	7.0
1.8 V ≤ VDD < 2.7 V	ANI0 to ANI3	231	8	2.5
	ANI16 to ANI22	321	8	7.0
1.6 V ≤ VDD < 2.7 V	ANI0 to ANI3	632	8	2.5
	ANI16 to ANI22	902	8	7.0

Remark The resistance and capacitance values shown in Table 16 - 4 are not guaranteed values.

(11) Starting the A/D converter

Start the A/D converter after the AVREFP and VDD voltages stabilize.

CHAPTER 17 D/A CONVERTER

The number of D/A converter channels differs depending on the product.

Table 17 - 1 Output Pin of D/A Converter

D/A output pins	10 pin	16 pin	20 pin	24 pin	25 pin
ANO1	—	√	√	√	√

17.1 Functions of D/A Converter

The D/A converter is an 8-bit resolution converter that converts digital inputs into analog signals. It is used to control analog outputs for CMP0 and one independent channel (ANO1).

The D/A converter has the following features.

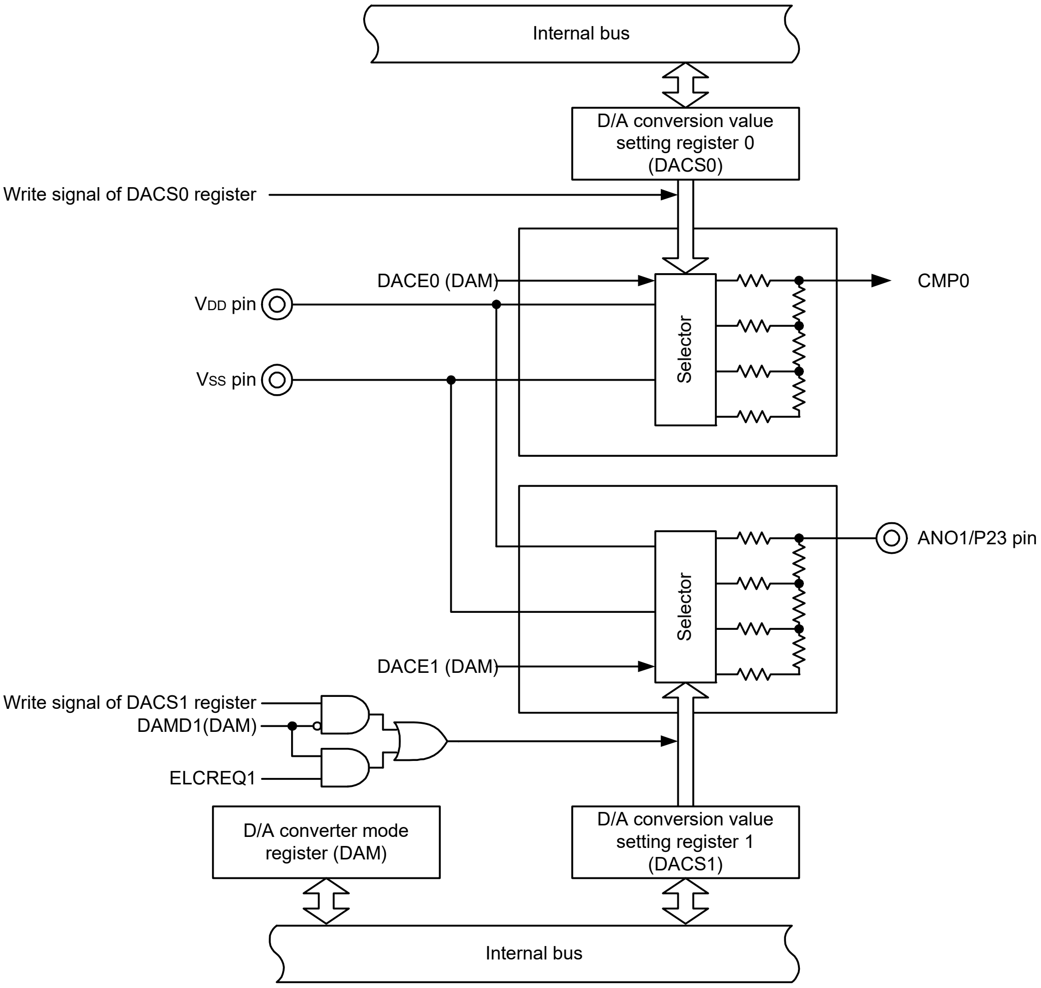
- 8-bit resolution × 2 channels
- R-2R ladder method
- Output analog voltage
    - 8-bit resolution: V_{DD} × m8/256 (m8: Value set to DACSi register)
- Operation mode
    - Normal mode
    - Real-time output mode

Remark i = 0, 1

17.2 Configuration of D/A Converter

Figure 17 - 1 shows the Block Diagram of D/A Converter.

Figure 17 - 1 Block Diagram of D/A Converter


Internal bus
D/A conversion value setting register 0 (DACS0)
Write signal of DACS0 register
DACE0 (DAM)
VDD pin
Vss pin
Selector
CMP0
Selector
ANO1/P23 pin
DACE1 (DAM)
Write signal of DACS1 register
DAMD1(DAM)
ELCREQ1
D/A converter mode register (DAM)
D/A conversion value setting register 1 (DACS1)
Internal bus

Remark ELCREQ1 is trigger signal (event signals from the ELC) that is used in the real-time output mode.

17.3 Registers Controlling D/A Converter

The D/A converter is controlled by the following registers.

- Peripheral enable register 1 (PER1)
- Peripheral reset control register 1 (PRR1)
- D/A converter mode register (DAM)
- D/A conversion value setting registers 0, 1 (DACS0, DACS1)
- Event output destination select register n (ELSELRn), n = 00 to 17
- Port mode control register 2 (PMC2)
- Port mode register 2 (PM2)

17.3.1 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 D/A converter is used, be sure to set bit 7 (DACEN) 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 17 - 2 Format of Peripheral enable register 1 (PER1)

Address: F007AH After reset: 00H R/W

Symbol	<7>	6	<5>	4	<3>	<2>	1	0
PER1	DACEN	0	CMPEN	0	DTCEN	PGA0EN	0	0

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

Caution 1. When setting the D/A converter, be sure to set DACEN to 1 first.
If DACEN = 0, writing to a control register of the D/A converter is ignored, and all read values are default values (except for port mode register 2 (PM2), and port register 2 (P2)).

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

17.3.2 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 D/A converter, be sure to set bit 7 (DACRES) to 1.

The PRR1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the PRR1 register to 00H.

Figure 17 - 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	DACRES	0	CMPRES	0	0	PGA0RES	0	0

DACRES	Reset control of DAC
0	D/A converter is released from the reset state.
1	D/A converter is in the reset state.

17.3.3 D/A converter mode register (DAM)

This register controls the operation of the D/A converter.
The DAM register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 17 - 4 Format of D/A converter mode register (DAM)

Address: F00FBH After reset: 00H R/W

Symbol	7	6	<5>	<4>	3	2	1	0
DAM	0	0	DACE1	DACE0	0	0	DAMD1	0

DACEi	D/A conversion operation control
0	Stops D/A conversion operation
1	Enables D/A conversion operation

DAMD1	D/A converter operation mode selection
0	Normal mode
1	Real-time output mode

Remark i = 0, 1

17.3.4 D/A conversion value setting register i (DACSi) (i = 0, 1)

This register is used to set the analog voltage value to be output to the CMP0 and ANO1 pins when the D/A converter is used.
The DACSi register can be read by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 17 - 5 Format of D/A conversion value setting register i (DACSi) (i = 0, 1)

Address: FFF3CH (DACS0), FFF3DH (DACS1) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
DACSi	DACSi7	DACSi6	DACSi5	DACSi4	DACSi3	DACSi2	DACSi1	DACSi0

Remark The relation between the resolution and analog output voltage (VANOi) of the D/A converter are as follows.
VANOi = V_{DD} × (DACSi)/256

When the D/A converter is not used, set the DACEi bit to 0 (output disable) and set the DACSi register to 00H to prevent current from flowing into the R-2R resistor ladder to reduce unnecessary current consumption.

17.3.5 Event output destination select register n (ELSELRn), n = 00 to 17

When the real-time output mode of the D/A converter is used, D/A conversion is performed using an event signal from the event link controller as an activation trigger.
For details, see **21.3.1 Event output destination select register n (ELSELRn) (n = 00 to 17)**.

17.3.6 Registers that control port functions of analog output pins

Set the registers that control the port functions for the port pins with which the analog output pin functions of the D/A converter are multiplexed (port mode register (PMxx) and port mode control register (PMCxx)).
For details, see **4.3.1 Port mode registers (PMxx)** and **4.3.6 Port mode control registers (PMCxx)**.
Using the ANO1 pin for analog output from the D/A converter requires setting bit 3 in port mode register 2 (PM2) and bit 3 in port mode control register 2 (PMC2) to 1.

17.4 Operations of D/A Converter

17.4.1 Operation in Normal Mode

D/A conversion is performed using write operation to the DACSi register as the trigger.
The setting method is described below.

Procedure of setting the D/A converter channel 0 :

<1> Set the DACEN bit of the PER1 register (peripheral enable register 1) to 1 to start the supply of the input clock to the D/A converter.
<2> Set the analog voltage value to be output to the CMP0 to the DACS0 register (D/A conversion value setting register 0).

Steps <1> and <2> above constitute the initial settings.

<3> Set the DACE0 bit of the DAM register to 1 (D/A conversion enable).
D/A conversion starts, and then, after the settling time elapses, the analog voltage set in step <4> is output to the CMP0.
<4> To perform subsequent D/A conversions, write to the DACSi register.

Procedure of setting the D/A converter channel 1 :

<1> Set the DACEN bit of the PER1 register (peripheral enable register 1) to 1 to start the supply of the input clock to the D/A converter.
<2> Use the port mode register (PMxx) and port mode control register (PMCxx) to set the ports to analog pins.
<3> Set the DAMD1 bit of the DAM register (D/A converter mode register) to 0 (normal mode).
<4> Set the analog voltage value to be output to the ANO1 pin to the DACS1 register (D/A conversion value setting register 1).

Steps <1> and <4> above constitute the initial settings.

<5> Set the DACE1 bit of the DAM register to 1 (D/A conversion enable).
D/A conversion starts, and then, after the settling time elapses, the analog voltage set in step <4> is output to the ANO1 pin.
<6>To perform subsequent D/A conversions, write to the DACS1 register.

The previous D/A conversion result is held until the next D/A conversion is performed.
When the DACEi bit of the DAM register is set to 0 (D/A conversion operation stop), D/A conversion stops.

Caution 1. Even if 1, 0, and then 1 is set to the DACEi bit, the analog voltage set by the DACSi register is output to the ANO1 pin and CMP0 when a settling time has elapsed after 1 is set for the last time.

Caution 2. If the DACSi register is rewritten during the settling time, D/A conversion is aborted and reconversion by using the rewritten values starts.

Remark i = 0, 1

17.4.2 Operation in Real-Time Output Mode (channel 1 (i =1))

D/A conversion is performed on each channel using the event signals from the ELC as triggers. The setting method is described below.

<1> Set the DACEN bit of the PER1 register (peripheral enable register 1) to 1 to start the supply of the input clock to the D/A converter.
<2> Use the port mode register (PMxx) and port mode control register (PMCxx) to set the ports to analog pins.
<3> Set the DAMDi bit of the DAM register (D/A converter mode register) to 0 (normal mode).
<4> Set the analog voltage value to be output to the ANOi pin to the DACSi register (D/A conversion value setting register i).
<5> Set the DACEi bit of the DAM register to 1 (D/A conversion enable).
D/A conversion starts, and then, after the settling time elapses, the analog voltage set in step <3> is output to the ANOi pin.
<6> Use the event output destination select register (ELSELRn; n = 00 to 17) to set the trigger signal used for real-time output mode.
<7> Set the DAMDi bit of the DAM register to 1 (real-time output mode).
<8> Start the operation of the event source.

Steps <1> to <8> above constitute the initial settings.

<9> Upon generation of the trigger signals used for real-time output mode, D/A conversion starts and the analog voltage set in step <4> will be output to the ANOi pin after a settling time has elapsed.
Set the analog voltage value to be output to the ANOi pin, to the DACSi register before performing the next D/A conversion (trigger signal used for real-time output mode is generated).

Set the analog voltage value to be output to the ANOi pin, to the DACSi register before performing the next D/A conversion (trigger signal used for real-time output mode is generated).
When the DACEi bit of the DAM register is set to 0 (D/A conversion operation stop), D/A conversion stops.

Caution 1. Even if 1, 0, and then 1 is set to the DACEi bit, the analog voltage set by the DACSi register is output to the ANOi pin when a settling time has elapsed after 1 is set for the last time.

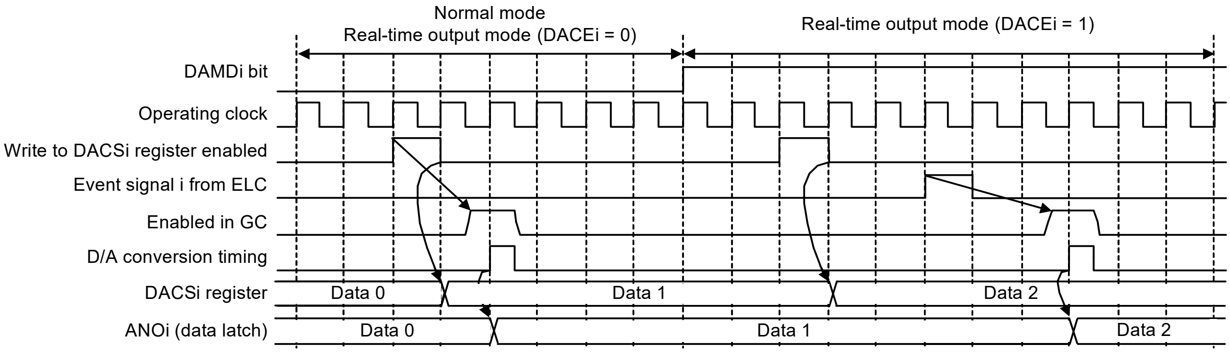
Caution 2. Set the interval between each generation of the trigger signal used for real-time output mode of the same channel to longer than the settling time. If a trigger signal used for real-time output mode is generated during the settling time, D/A conversion is aborted and reconversion starts.

Caution 3. Set the interval between each generation of the trigger signal used for real-time output mode of the same channel to longer than the three clocks of fCLK. When a trigger is generated consecutively at intervals of three or fewer fCLK clock cycles, D/A conversion is performed using only the first trigger.

17.4.3 Timing for Outputting D/A Conversion Value

Figure 17 - 6 shows the Timing for Outputting D/A Conversion Value.

Figure 17 - 6 Timing for Outputting D/A Conversion Value



Remark i = 1

- Normal mode and real-time output mode (when conversion operation is disabled)
  The value is written to the data latch after one cycle of the operating clock when the DACSi register is written.
- Real-time output mode (when conversion operation is enabled)
  The value is written to the data latch (output from the ANOi pin) after three cycles of the operating clock when the event signal from the ELC is accepted.

17.5 Cautions for D/A Converter

Observe the following cautions when using the D/A converter.
Note that the caution applicable to channel 0 is only (2).

(1) The digital port I/O function, which is the alternate function of the ANO1 pins, does not operate if the ports are set to analog pins by using the port mode register (PMxx) and port mode control register (PMCxx).
When the P2 register is read while the ports are set to analog pins by using the port mode register (PMxx) and the port mode control register (PMCxx), 0 is read in the input mode and the set value of the P2 register is read in the output mode. If the digital output mode is set, no output data is output to pins.

(2) The operation of the D/A converter continues in the HALT and STOP modes. To lower the power consumption, therefore, clear the DACEi bit to 0, and execute the HALT or STOP instruction after stopping the operation of the D/A converter.

Remark i = 0, 1

(3) To stop the real-time output mode (including when changing to normal mode), one of the following procedures must be used:

- Wait for at least three clocks after stopping the trigger output source and then set bits DACEi and DAMDi to 0.
- After setting bits DACEi and DAMDi to 0, set the DACEN bit in the PER1 register to 0 (stopping input clock supply to the D/A converter).
  Note that the D/A converter is not reset even if the DACEN bit is set to 0.
  To reset the D/A converter and SRFs in the D/A converter, use bit 7 (DACRES) in PRR1.

(4) When D/A conversion operation is enabled, do not perform A/D conversions from the analog input pins multiplexed with the ANO1 pins.

(5) In real-time output mode, set the value of the DACSi register before a trigger signal used for real-time output mode is generated. Do not change the set value of the DACSi register while the trigger signal is output.

(6) Since the output impedance of the D/A converter is high, no current can be taken out from the ANO1 pin. If the input impedance of the load is low, insert a follower amplifier between the load and the ANO1 pins before use. In addition, the wiring length between the follower amplifier and the load must be as short as possible due to the high output impedance. If the wiring length is long, take measures such as placing a ground pattern around the wiring area.

(7) When entering STOP mode while real-time output mode is enabled, disable linking of ELC events before entering STOP mode.

Remark i = 1

CHAPTER 18 PROGRAMMABLE GAIN AMPLIFIER (PGA)

A circuit of programmable gain amplifier is incorporated in RL78/G1F.

Item	RL78G11
Analog input channels	PGAI
GND of feedback resistance of the programmable gain amplifier	PGAGND

18.1 Functions of Programmable Gain Amplifier

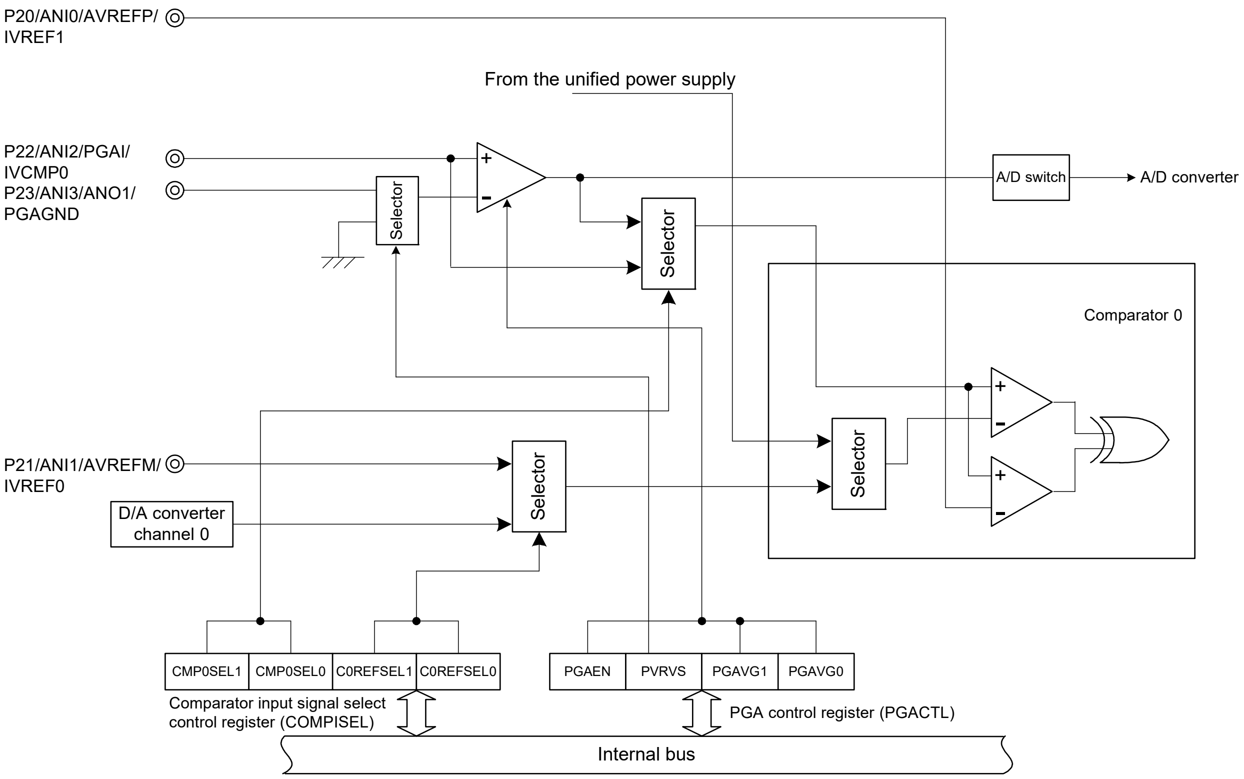
The programmable gain amplifier is provided with the following functions.

- One among four amplification factors can be selected.
- The output signal of a programmable gain amplifier can be set as the analog input of A/D converter and the input signal on the positive side of comparator 0 (CMP0).

18.2 Configuration of Programmable Gain Amplifier

The programmable gain amplifier includes the following hardware.

Figure 18 - 1 Block Diagram of Programmable Gain Amplifier



18.3 Registers Controlling Programmable Gain Amplifier

Table 18 - 1 lists the registers controlling the programmable gain amplifier.

Table 18 - 1 Registers Controlling Programmable Gain Amplifier

Register Name	Symbol
Peripheral enable register 1	PER1
Peripheral reset control register 1	PRR1
PGA control register	PGACTL
Comparator input signal select control register	COMPISEL
Port mode control register 2	PMC2
Port mode register 2	PM2

18.3.1 Peripheral enable register 1 (PER1)

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 PGA is used, be sure to set bit 2 (PGA0EN) 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 18 - 2 Format of Peripheral Enable Register 1 (PER1)

Address: F000AH After reset: 00H R/W

Symbol	<7>	6	<5>	4	<3>	<2>	1	0
PER1	DACEN	0	CMPEN	0	DTCEN	PGA0EN	0	0

PGA0EN Note	Programmable gain amplifier input clock control
0	• SFR used by the programmable gain amplifier cannot be written. • Programmable gain amplifier is not initialized. Note
1	• SFR used by the programmable gain amplifier can be read/written.

Note To initialize the SFR used by PGA0EN, use bit 2 (PGA0RES) of PRR1.

Caution 1. When setting programmable gain amplifier, be sure to set the PGA0EN bit to 1 first.
If PGA0EN = 0, writing to a control register of programmable gain amplifier is ignored, and all read values are default values (except for port mode register 2 (PM2), port mode control register 2 (PMC2), and port register 2 (P2)).

Caution 2. Be sure to clear bits 0, 1, 4, and 6 to 0.

18.3.2 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 PGA, be sure to set bit 2 (PGA0RES) to 1.

The PRR1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the PRR1 register to 00H.

Figure 18 - 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	DACRES	0	CMPRES	0	0	PGA0RES	0	0

PGA0RES	Reset control of PGA
0	PGA is released from the reset state.
1	PGA is in the reset state.

18.3.3 PGA control register (PGACTL)

The PGACLT register sets operation enable/disable and amplification factor of the programmable gain amplifier.

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

Reset signal generation clears this register to 00H.

Figure 18 - 4 Format of PGA Control Register (PGACTL)

Address: F0349H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
PGACTL	PGAEN	0	0	0	PVRVS	0	PGAVG1	PGAVG0

PGAEN	Programmable gain amplifier operation control
0	Stops operation of programmable gain amplifier.
1	Enables operation of programmable gain amplifier.

PVRVS	GND selection of feedback resistance of the programmable gain amplifier
0	Selects V_{SS}.
1	Selects PGAGND.

PGAVG1	PGAVG0	Programmable gain amplifier amplification factor selection
0	0	×4
0	1	×8
1	0	×16
1	1	×32

Caution 1. Rewrite the bits of the PGACTL register other than PGAEN when PGA operation is stopped (PGAEN = 0).

Caution 2. For the programmable gain amplifier, an operation stabilization wait time (10 μs) is required after setting PGAEN = 1.

Caution 3. Be sure to clear bits 6 to 4 and 2 to 0.

18.3.4 Comparator input signal select control register (COMPISEL)

The COMPISEL register is used to select the input signal on the + or - pin of the comparator 0.

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

Reset signal generation clears this register to 00H.

Figure 18 - 5 Format of Comparator Input Signal Select Control Register (COMPISEL)

Address: F0348H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
COMPISEL	0	0	0	0	COMP0SEL1	COMP0SEL0	C0REFSEL1	C0REFSEL0

COMP0SEL1	COMP0SEL0	Selection of the input signal on + pin of the comparator 0
0	0	Nothing is selected.
0	1	IVCMP0 pin is selected.
1	0	The output signal from the programmable gain amplifier is selected.
1	1	Setting prohibited

C0REFSEL1	C0REFSEL0	Selection of the input signal on - pin of the comparator 0
0	0	Nothing is selected.
0	1	IVREF0 pin is selected.
1	0	The output signal from channel 0 of the on-chip D/A converter is selected.
1	1	Setting prohibited

Caution Be sure to clear bits 7 to 4 to 0.

18.3.5 Port mode control register 2 (PMC2)

This register is used to set the digital I/O/analog input in 1-bit units.

To use the programmable gain amplifier, set bit 2 (PMC22) to 1. To select PGAGND as GND of the feedback resistance of the programmable gain amplifier, set bit 3 (PMC23) to 1.

Port mode control register 2 (PMC2) can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 18 - 6 Format of Port Mode Control Register 2 (PMC2)

Address: F0062H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PMC2	1	1	1	1	PMC23	PMC22	PMC21	PMC20

PMC2n	P2n pin digital I/O/analog input selection
0	Digital I/O (alternate function other than analog input)
1	Analog input

Remark n: Channel number (n = 0 to 3)

18.3.6 Port mode register 2 (PM2)

This register is used to set the port I/O in 1-bit units.

To use the programmable gain amplifier, set bit 2 (PM22) to 1. To select PGAGND as GND of the feedback resistance of the programmable gain amplifier, set bit 3 (PM23) to 1.

Port mode register 2 (PM2) can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 18 - 7 Format of Port Mode Register 2 (PM2)

Address: FFF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	1	1	1	1	PM23	PM22	PM21	PM20

PM2n	P2n pin I/O mode selection
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Remark n: Channel number (n = 0 to 3)

18.4 Operation of Programmable Gain Amplifier

The analog voltage input from the PGAI pin is amplified within the microcontroller. The gain can be selected from four types (×4, ×8, ×16, and ×32).

The amplified voltage can be used as an analog input of the A/D converter and the input signal on the positive side of the comparator 0 (CMP0).

The procedure for starting operation of the programmable gain amplifier is described below.

(1) Set input clock supply for the programmable gain amplifier in the PGA0EN bit of the PER1 register.

(2) Use the PMC2 register to set the pins (PMC22, PMC23) to be used in the programmable gain amplifier as analog inputs.

(3) Use the PM2 register to set the pins (PM22, PM23) to be used in the programmable gain amplifier to input mode.

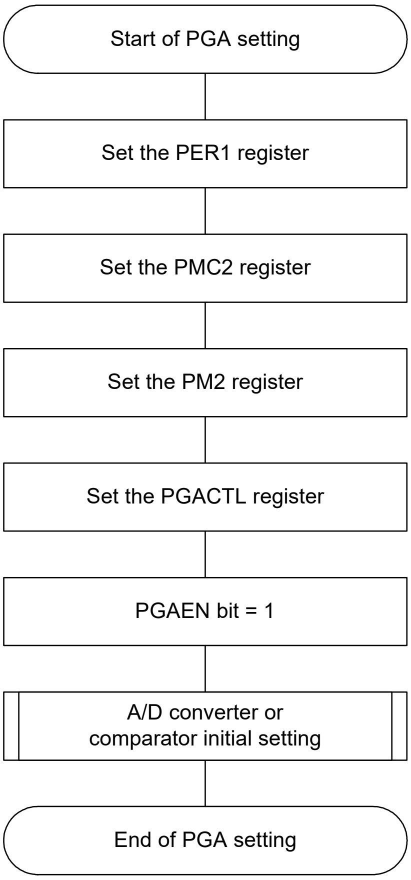
(4) Use the PGAVG0 and PGAVG1 bits to select the gain (×4, ×8, ×16, and ×32).

(5) To use the programmable gain amplifier output as the input signal on the positive side of Comparator 0, set the output signal from the programmable gain amplifier by the COMP0SEL1 and COMP0SEL0 bits of the COMPISEL register.

(6) Set (1) the PGAEN bit and enable operation of the programmable gain amplifier.

18.4.1 Setting procedure of programmable gain amplifier

Figure 18 - 8 Operation Setting Flow Chart of Programmable Gain Amplifier (PGA)



- Set the PGA0EN bit (1)

- Set the PMC22 bit (1)
- Set the PMC23 bit (1) to select PGAGND as the feedback resistance of PGA.

- Set the PM22 bit (1)
- Set the PM23 bit (1) to select PGAGND as the feedback resistance of PGA.

- The PGAEN bit is cleared (0).
- Select GND of the feedback resistance of PGA.
- Select the amplification factor of PGA.

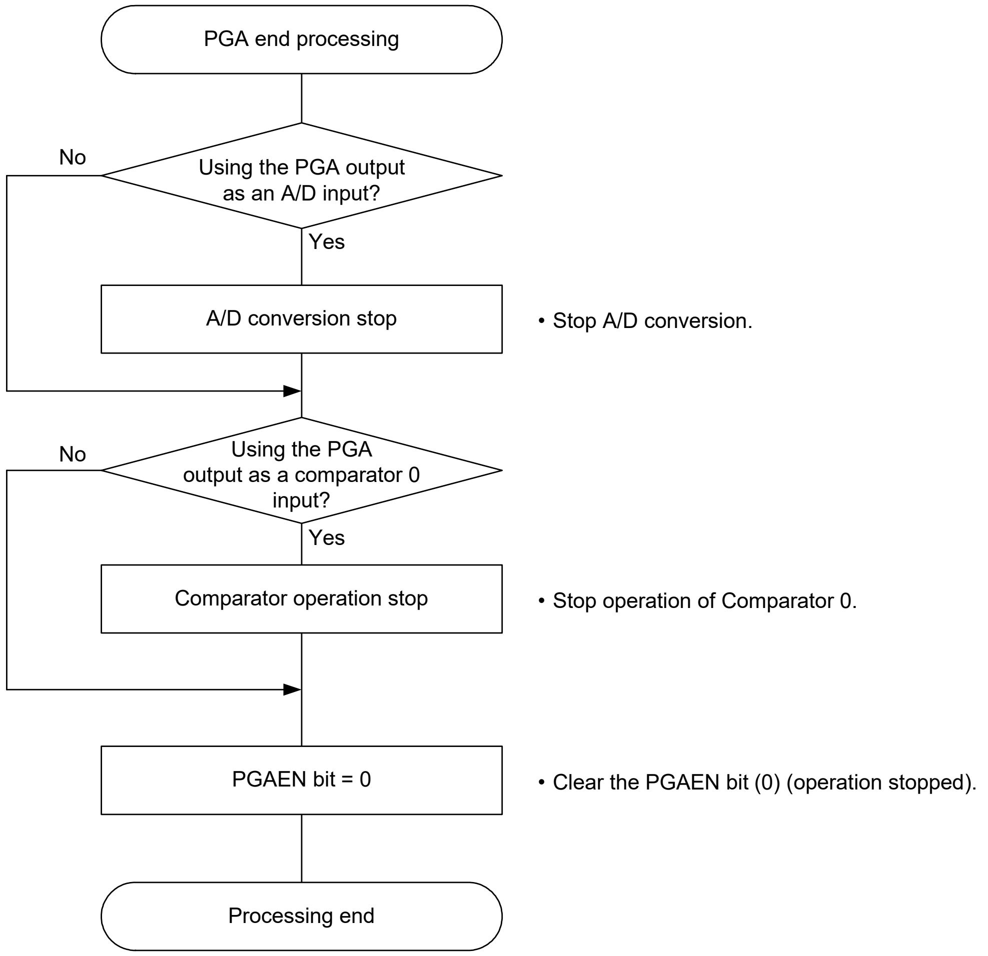
- Set the PGAEN bit (1) (operation enabled)

- See CHAPTER 16 A/D CONVERTER.
- See CHAPTER 19 COMPARATOR.

Caution After setting the PGAEN bit (1), start A/D conversion after 10 μs elapse as PGA operation stabilization wait time.

18.4.2 Setting procedure of programmable gain amplifier

Figure 18 - 9 Operation Stopping Flow Chart of Programmable Gain Amplifier (PGA)


PGA end processing
No
Using the PGA output as an A/D input?
Yes
A/D conversion stop
• Stop A/D conversion.
No
Using the PGA output as a comparator 0 input?
Yes
Comparator operation stop
• Stop operation of Comparator 0.
PGAEN bit = 0
• Clear the PGAEN bit (0) (operation stopped).
Processing end

Caution 1. When restarting PGA and A/D converter or comparator, start the function after 10 μs elapse as PGA operation stabilization wait time after setting the PGAEN bit (1).

Caution 2. The A/D conversion pins and comparators to which PGA output is not connected can be used even when PGA operation is stopped.

CHAPTER 19 COMPARATOR

19.1 Functions of Comparator

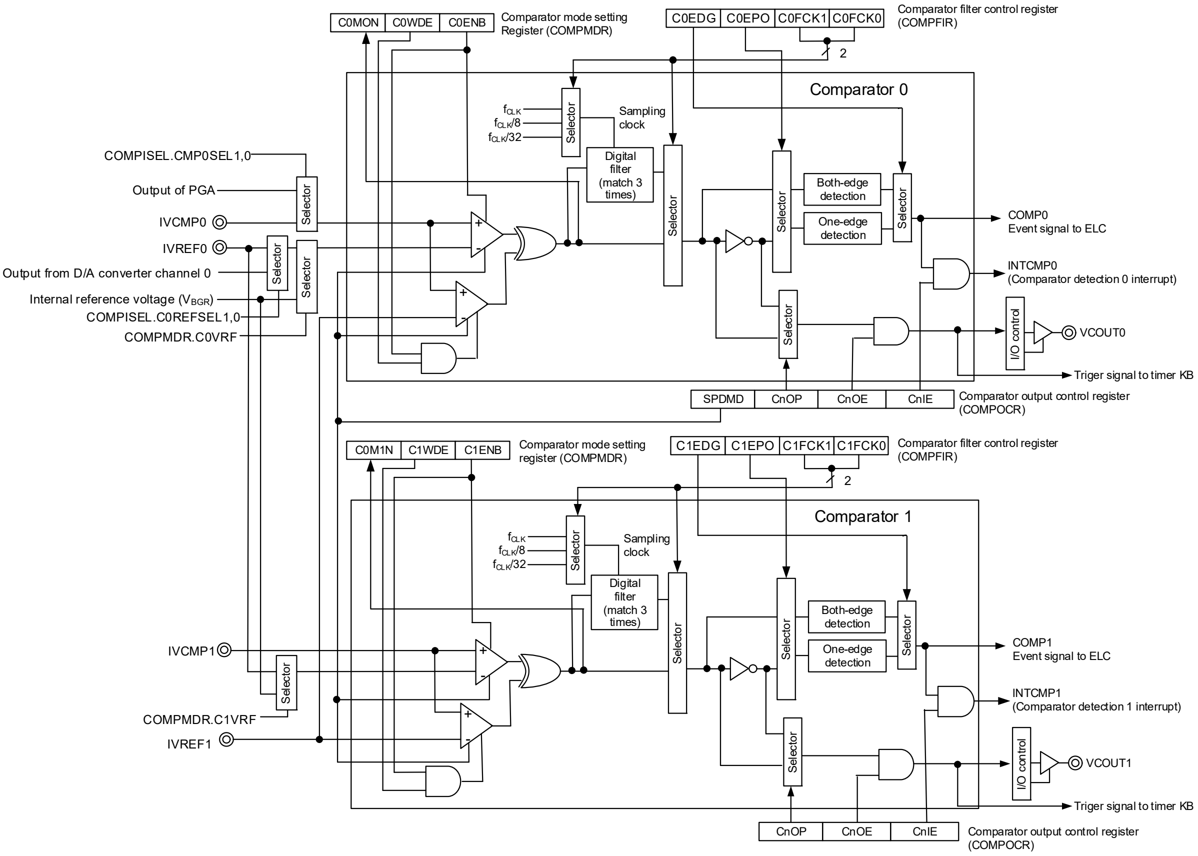
The comparator has the following functions.

- Comparator high-speed mode, comparator low-speed mode, comparator high-speed window mode, or comparator low-speed window mode can be selected.
- The reference voltage can be selected from among internal reference voltage (1.45 V), externally-input reference voltage, and voltage output from the D/A converter channel 0 (only in comparator 0).
- 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.

19.2 Configuration of Comparator

Figure 19 - 1 shows the Comparator Block Diagram.

Figure 19 - 1 Comparator Block Diagram


C0MON
C0WDE
C0ENB
Comparator mode setting Register (COMPMDR)
C0EDG
C0EPO
C0FCK1
C0FCK0
Comparator filter control register (COMPFIR)
2
Comparator 0
fCLK
fCLK/8
fCLK/32
Selector
Sampling clock
Digital filter (match 3 times)
COMPISEL.CMP0SEL1,0
Output of PGA
IVCMP0
IVREF0
Output from D/A converter channel 0
Internal reference voltage (VBGR)
COMPISEL.C0REFSEL1,0
COMPMDR.C0VRF
+
-
Both-edge detection
One-edge detection
COMP0
Event signal to ELC
INTCMP0
(Comparator detection 0 interrupt)
I/O control
VCOUT0
Triger signal to timer KB
SPDMD
CnOP
CnOE
CnIE
Comparator output control register (COMPOCR)
C0M1N
C1WDE
C1ENB
Comparator mode setting register (COMPMDR)
C1EDG
C1EPO
C1FCK1
C1FCK0
Comparator filter control register (COMPFIR)
2
Comparator 1
IVCMP1
COMPMDR.C1VRF
IVREF1
COMP1
Event signal to ELC
INTCMP1
(Comparator detection 1 interrupt)
VCOUT1
Triger signal to timer KB
CnOP
CnOE
CnIE
Comparator output control register (COMPOCR)

Remark n = 0, 1

19.3 Registers Controlling Comparator

Table 19 - 1 lists the Registers Controlling Comparator.

Table 19 - 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
Comparator input select control register	COMPISEL
Port mode control register 0, 2, 3	PMC0, PMC2, PMC3
Port mode registers 0, 2, 3, 4, 5	PM0, PM2, PM3, PM4, PM5
Port registers 0, 3, 4, 5	P0, P3, P4, P5

19.3.1 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 19 - 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	DACEN	0	CMPEN	0	DTCEN	PGA0EN	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 0, 2, 3 (PMC0, PMC2, PMC3), port mode registers 0, 2, 3, 4, 5 (PM0, PM2, PM3, PM4, PM5), and port registers 0, 3, 4, 5 (P0, P3, P4, P5)).

Caution 2. Be sure to clear bits 0, 1, 4, 6 to 0.

19.3.2 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 19 - 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	DACRES	0	CMPRES	0	0	PGA0RES	0	0

CMPRES	Reset control of comparator
0	Comparator is released from the reset state.
1	Comparator is in the reset state.

19.3.3 Comparator mode setting register (COMPMDR)

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

Figure 19 - 4 Format of Comparator mode setting register (COMPMDR) (1/2)

Address: F0340H After reset: 00H R/W

Symbol	7	6	5	<4>	3	2	1	<0>
COMPMDR	C1MON	C1VRF	C1WDE	C1ENB	C0MON	C0VRF	C0WDE	C0ENB

C1MON	Comparator 1 monitor flag [Notes 1, 2]
0	In standard mode: VCMP1 < Comparator 1 reference voltage [Note 5] In window mode: IVCMP1 < IVREF0 or IVCMP1< IVREF1
1	In standard mode: IVCMP1 > Comparator 1 reference voltage [Note 5] In window mode: IVREF0 < IVCMP1 < IVREF1

C1VRF	Comparator 1 reference voltage selection [Note 6]
0	The input voltage to the IVREF0 pin is selected as the reference voltage for comparator 1.
1	The internal reference voltage (1.45 V) is selected as the reference voltage for comparator 1.

C1WDE	Comparator 1 window mode selection [Note 4]
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 < Comparator 1 reference voltage [Note 3] In window mode: IVCMP0 < Reference voltage specified in COMPISEL.C0REFSEL or IVCMP0 > IVREF1
1	In standard mode: VCMP0 > Comparator 1 reference voltage [Note 3] In window mode: Reference voltage specified in COMPISEL.C0REFSEL < IVCMP0 < IVREF1

C0VRF	Comparator 0 reference voltage selection [Note 6]
0	The voltage specified in COMPISEL.C0REFSEL is selected as the reference voltage for comparator 0.
1	The internal reference voltage (1.45 V) is selected as the reference voltage for comparator 0.

C0WDE	Comparator 0 window mode selection [Note 4]
0	Comparator 0 standard mode
1	Comparator 0 window mode

Figure 19 - 4 Format of Comparator mode setting register (COMPMDR) (2/2)

Address: F0340H After reset: 00H R/W

Symbol	7	6	5	<4>	3	2	1	<0>
COMPMDR	C1MON	C1VRF	C1WDE	C1ENB	C0MON	C0VRF	C0WDE	C0ENB

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. In the standard mode, the internal reference voltage (1.45 V), voltage input to the IVREF0 pin, or voltage output from channel 0 of the D/A converter can be selected as the reference voltage for comparator 0.

Note 4. When using window mode, make sure that IVREF1 > IVREF0 for use as the reference voltage.

Note 5. In the standard mode, the internal reference voltage (1.45 V) or voltage input to the IVREF0 pin can be selected as the reference voltage for comparator 1.

Note 6. When the internal reference voltage (1.45 V) is selected, the temperature sensor output cannot be A/D-converted by the A/D converter.

Remark This comparator can also be used in window mode when the low-speed mode is selected (the SPDMD bit in the COMPOCR register is 0).

19.3.4 Comparator filter control register (COMPFIR)

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

Figure 19 - 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 ELSELR16 register for the ELC to 0 (not linked to comparator 1 output). In addition, clear bit 1 (CMPIF1) in interrupt request flag register 2L (IF2L) 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 ELSELR 15 register for the ELC to 0 (not linked to comparator 0 output). In addition, clear bit 0 (CMPIF0) in interrupt request flag register 2L (IF2L) 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.

19.3.5 Comparator output control register (COMPOCR)

Figure 19 - 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 Note 4

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 Note 4

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 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 1 (CMPIF1) in interrupt request flag register 2L (IF2L) may set to 1 (interrupt requested), clear bit 1 (CMPIF1) in interrupt request flag register 2L (IF2L) to 0 before using an interrupt.

Note 3. If C0IE is changed from 0 (interrupt request disabled) to 1 (interrupt request enabled), since bit 0 (CMPIF0) in interrupt request flag register 2L (IF2L) may set to 1 (interrupt requested), clear bit 0 (CMPIF0) in interrupt request flag register 2L (IF2L) to 0 before using an interrupt.

Note 4. When comparator i output is used as a trigger for functions that operate simultaneously with the timer KB, such as the forced output stop function and the timer restart function, set the CiOE bit to 1.

Remark i = 0, 1

19.3.6 Registers that control port functions of input/output pins for the comparators

Using the IVCMP0, IVCMP1, IVREF0, and IVREF1 pins for analog input to a comparator requires setting the corresponding bits in the port mode register (PMxx) and port mode control register (PMCxx) to 1.

Using the VCOUT0 and VCOUT1 functions requires setting of the registers that control the port functions for the port pins with which the VCOUT0 and VCOUT1 functions for the target channel are multiplexed (port mode control register (PMCxx), port mode register (PMxx), and port register (Pxx)). 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)**.

19.4 Operation

Comparator 0 and comparator 1 operate independently. Table 19 - 2 lists the Procedure for Setting Comparator Associated Registers.

Table 19 - 2 Procedure for Setting Comparator Associated Registers

Step		Register	Bit	Setting Value
CM P0	CM P1			
1	1	PRR1	CMPRES	0 (reset release of comparator i)
2	-	PRR1	PGA0RES	0 (reset release of PGA)
3	-	PRR1	DACRES	0 (reset release of DAC)
4	2	PER1	CMPEN	1 (input clock supply)
5	-	PER1	PGA0EN	1 (input clock supply)
6	-	PER1	DACEN	1 (input clock supply)
7	3	PMCxx	PMC20, PMC21, PMC22, PMC33	Select the function of pins IVCMPi and IVREFi. • Set the PMC20, PMC21, PMC22, and PMC33 bits to 1 (analog input). • Set the PM20, PM21, PM22, and PM33 bits to 1 (input mode).
		PMxx	PM20, PM21, PM22, PM33	
8	-	PGACTL	PGAVG0, PGAVG1	(Programmable gain amplifier amplification factor selection)
9	-	PGACTL	PVRVS	(GND selection of feedback resistance of the programmable gain amplifier)
10	-	PGACTL	PGAEN	1(Enables operation of programmable gain amplifier)
11	-	COMPISEL	CMP0SEL0, CMP0SEL1	(Selection of the signal input to the + pin of comparator 0)
12	-	COMPISEL	C0REFSEL0, C0REFSEL1	(Selection of the signal input to the - pin of comparator 0)
13	-	DACS0		(D/A conversion value setting)
14	-	DAM	DACE0	(Enables D/A conversion operation)
15	4	COMPMDR	C0VRF, C1VRF	Selection of the reference voltage for comparators 0 and 1
16	5	COMPOCR	SPDMD	Select the comparator response speed (0: Low-speed mode/1: High-speed mode). Note 1
17	6	COMPMDR	CiWDE	0 (standard mode) 1 (window mode) Note 2
18	7	COMPMDR	CiENB	1 (operation enabled)
19	8	Wait for comparator stabilization time t_{CMP}.		
20	9	COMPFIR	CiFCK1 to CiFCK0	Select whether the digital filter is used or not and the sampling clock.
			CiEPO, CiEDG	Select the edge detection condition for an interrupt request (rising edge/falling edge/both edges).
21	10	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.
22	11	Select the port logic output pin: VCOUTi Refer to **19.4.4 Comparator i Output (i = 0 or 1)**.		
23	12	PR02L, PR12L	CMPPR0i, CMPPR1i	When using an interrupt: Select the interrupt priority level.
24	13	MK2L	CMPMKi	When using an interrupt: Select the interrupt masking.
25	14	IF2L	CMPIFi	When using an interrupt: 0 (no interrupt requested: initialization) Note 3

Note 1. Comparator 0 and comparator 1 cannot be set independently.

Note 2. Can be set in high-speed mode (SPDMD = 1).

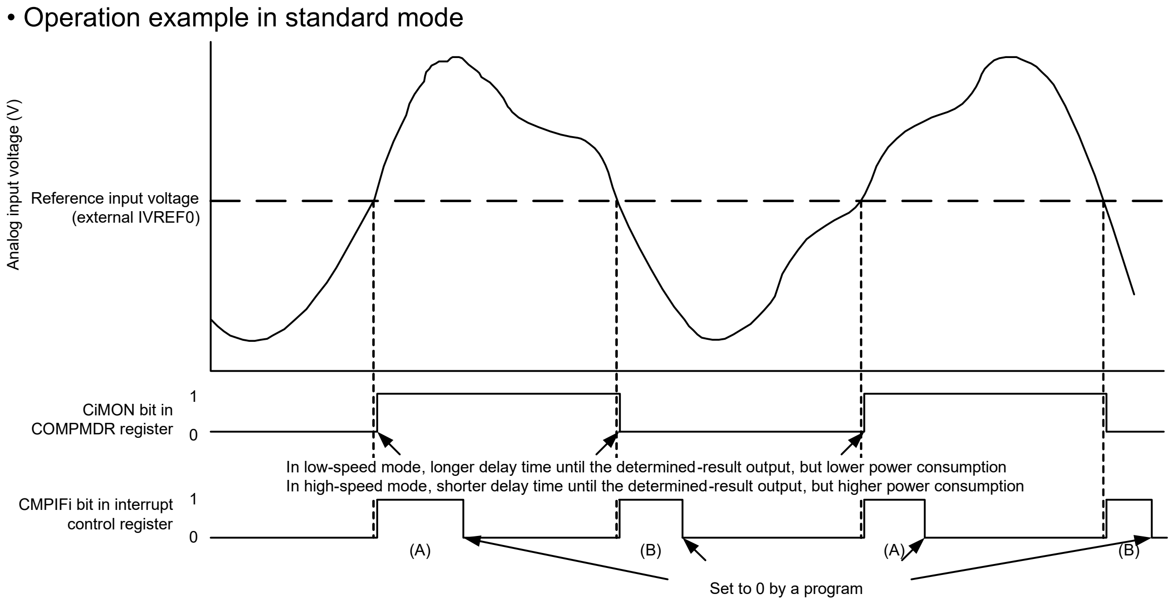
Note 3. After the setting of the comparator, an unnecessary interrupt may occur until operation becomes stable, so initialize the interrupt flag.

Caution **To initialize all circuits of converter i, set the CMPRES bit in the PRR1 register to 1.**

Remark i = 0, 1

Figures 19 - 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 19 - 7 Comparator i (i = 0 or 1) Operation Example in Standard Mode

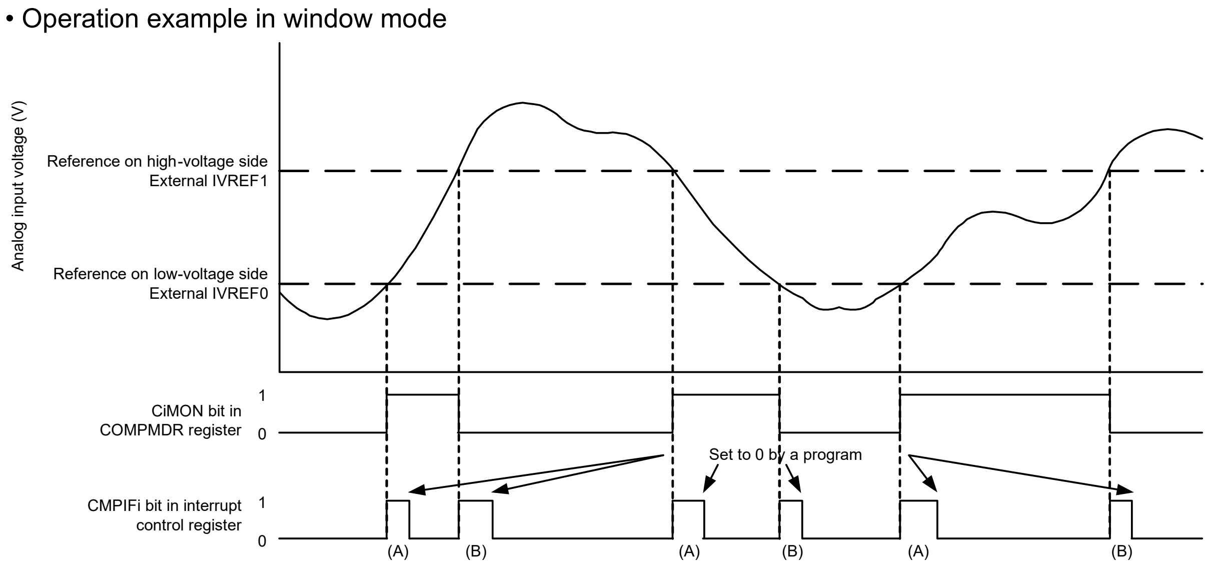

• Operation example in standard mode
Analog input voltage (V)
Reference input voltage
(external IVREF0)
CiMON bit in
COMPMDR register
1
0
In low-speed mode, longer delay time until the determined-result output, but lower power consumption
In high-speed mode, shorter delay time until the determined-result output, but higher power consumption
CMPIFi bit in interrupt
control register
1
0
(A)
(B)
(A)
(B)
Set to 0 by a program

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 19 - 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 19 - 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.**

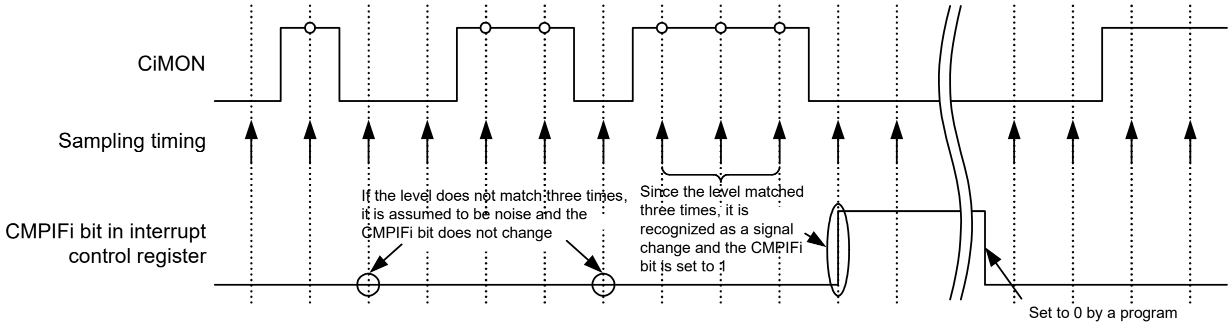
When using the comparator i interrupt, set CiIE in the COMPOCR register to 1 (interrupt request enabled). If the comparison result changes at this time, a comparator i interrupt request is generated. For details on interrupt requests, refer to **19.4.2 Comparator i (i = 0 or 1) Interrupts**.

19.4.1 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 19 - 9 shows the Comparator i (i = 0 or 1) Digital Filter and Interrupt Operation Example.

Figure 19 - 9 Comparator i (i = 0 or 1) Digital Filter and Interrupt Operation Example



If the level does not match three times, it is assumed to be noise and the CMPIFi bit does not change

Since the level matched three times, it is recognized as a signal change and the CMPIFi bit is set to 1

Set to 0 by a program

Caution **The above operation example applies when bits CiFCK1 to CiFCK0 in the COMPFIR register is 01B, 10B, or 11B (digital filter enabled).**

19.4.2 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.

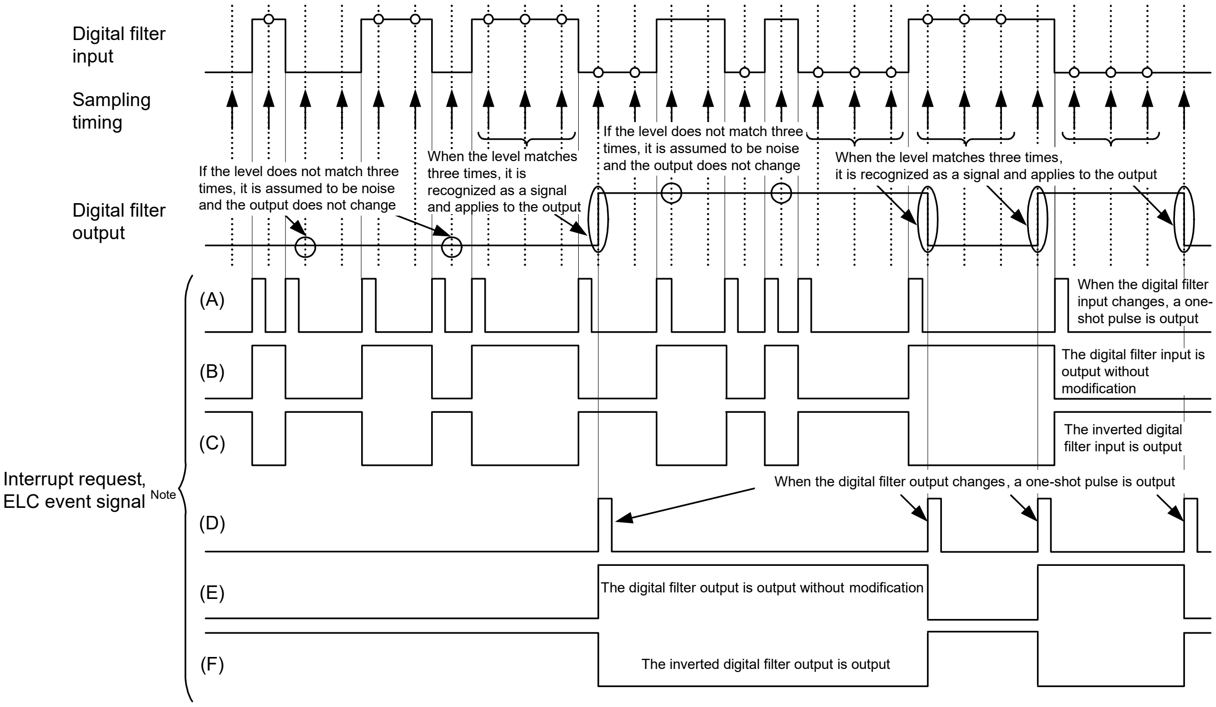
When using the comparator i interrupt, set the CiIE bit in the COMPOCR register to 1 (interrupt request output enabled). The condition for interrupt request generation can be set by the COMPFIR register. The comparator outputs can also be passed through the digital filter. Three different sampling clocks can be selected for the digital filter.

For details on the register setting and interrupt request generation, refer to **19.3.4 Comparator filter control register (COMPFIR)** and **19.3.5 Comparator output control register (COMPOCR)**.

19.4.3 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 ELSELR15 and ELSELR16 for the ELC to select the event output destination and to stop linking events.

Figure 19 - 10 Digital Filter and Interrupt Request/Event Signal Output to the ELC Operation


Digital filter input
Sampling timing
Digital filter output
If the level does not match three times, it is assumed to be noise and the output does not change
When the level matches three times, it is recognized as a signal and applies to the output
If the level does not match three times, it is assumed to be noise and the output does not change
When the level matches three times, it is recognized as a signal and applies to the output
Interrupt request, ELC event signal Note
(A)
When the digital filter input changes, a one-shot pulse is output
(B)
The digital filter input is output without modification
(C)
The inverted digital filter input is output
When the digital filter output changes, a one-shot pulse is output
(D)
(E)
The digital filter output is output without modification
(F)
The inverted digital filter output is output

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).

19.4.4 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 **19.3.5 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 19 - 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 19 - 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 “0” (start outputting from the pin).

19.4.5 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 IF2L 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 (CiEDG = 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 (CiEDG = 0), an interrupt request at the falling edge for the comparator, and IVCMP < IVREF

19.5 Caution for Using Timer KB Simultaneous Operation Function

In addition to their use as an external interrupt input, the INTP10, 11 pin output and the comparator output signal can be used as a trigger for functions that operate simultaneously with timer KB, such as the forced output stop function and timer restart function. The settings in peripheral function switch register 0 (PFSEL0), the edge selection resisters and comparator output control resister (COMPOCR) must be specified according to the function used.

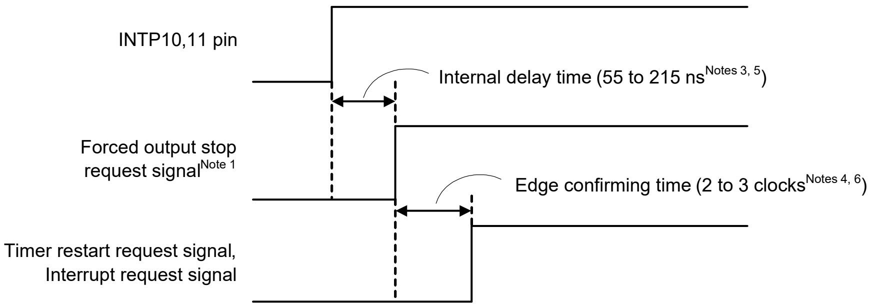
The width of the active signal required until each function starts operating differs.

When using INTP10, 11 or the comparator output signal, see **Table 19 - 3** and **Table 19 - 4** to specify the necessary register settings, and configure external circuits so that the required active signal width is assured.

Table 19 - 3 Relationship of INTP10, 11 Function, Register Settings, and Active Signal Width

Function	Peripheral Enable Register Setting	Edge Setting Registers	Necessary Active Signal Width to Operate Each Function		
			Interrupt	Forced Output Stop	Timer Restart
External interrupt (STOP release is enabled)	TMRSTENm = 0	EGPn, EGNn	Up to 1 µs	–	–
Forced output stop Note 1	TMRSTENm = 1	INTPEG[3:0]	55 to 215 ns Note 3 + 2 to 3 clocks Note 4	55 to 215 ns Notes 3, 5	–
Timer restart	TMRSTENm = 1	INTPEG[3:0]	55 to 215 ns Note 3 + 2 to 3 clocks Note 4	–	55 to 215 ns Note 3 + 2 to 3 clocksNotes 4, 6

Figure 19 - 11 Generation Timing of Forced Output Stop Signal and Timer Restart Request Signal by INTP10, 11


INTP10,11 pin
Internal delay time (55 to 215 nsNotes 3, 5)
Forced output stop request signalNote 1
Edge confirming time (2 to 3 clocksNotes 4, 6)
Timer restart request signal, Interrupt request signal

Note 1. Only INTP10, 11 can be used as a trigger for forced output stop function 2.

Note 2. The active level of the forced output stop function 2 is high. Edge selection is only applied to detection of an interrupt signal.

Note 3. 5 to 15 ns when noise filtering on INTP10, 11 is disabled (PNFEN = 1)

Note 4. For fCLK or fHOCO

Note 5. An additional output delay time (10 to 40 ns) is required from when forced output stop function 2 starts operating to when the level of the timer KB output changes.

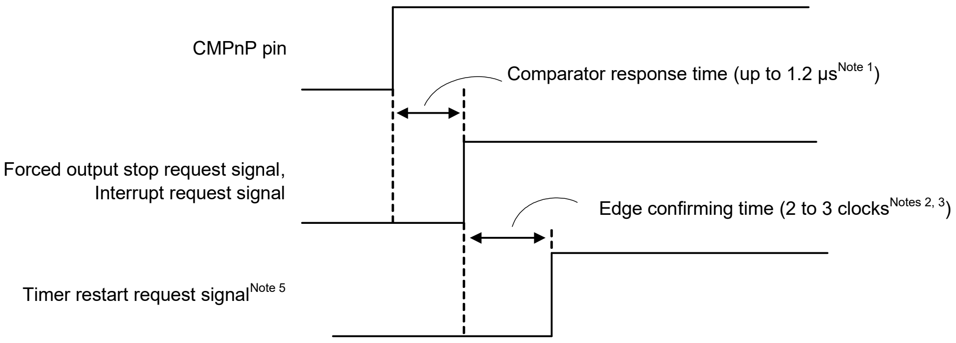
Note 6. Until the timer restart function starts operating, an additional clock cycle is required after the timer restart request signal is received, and an additional output delay time (10 to 40 ns) is required until the level of the timer KB output changes.

Remark m = 0, 1 n = 10, 11 p = 7, 6

Table 19 - 4 Relationship of Comparator 0 and 1 Functions, Register Settings, and Active Signal Width

Function	Peripheral Enable Register Setting	Edge Setting Registers	Comparator Output Control Resister Setting	Necessary Active Signal Width to Operate Each Function		
				Interrupt	Forced Output Stop	Timer Restart
External interrupt (STOP release is enabled) Note 1	–	CnEDG,CnEPO (n = 0,1)	–	Up to 1.2 μs Note 1	–	–
Forced output stop	–	Note 4	CnOE = 1 Note 6	Up to 1.2 μs Note 1	Up to 1.2 μs Notes 1, 5	–
Timer restart Note 6	–	CnEDG,CnEPO (n = 0,1)	CnOE = 1 Note 6	Up to 1.2 μs Note 1	–	Up to 1.2 μs Note 1+ 2 to 3 clocks Notes 3, 4

Figure 19 - 12 Generation Timing of Forced Output Stop Request Signal and Timer Restart Request Signal by Comparator 0 and 1


CMPnP pin
Comparator response time (up to 1.2 μsNote 1)
Forced output stop request signal,
Interrupt request signal
Edge confirming time (2 to 3 clocksNotes 2, 3)
Timer restart request signalNote 5

Note 1. When noise filtering is set to “0, 0” by using the CnDFS1 and CnDFS0 bits in the comparator n control register (CnCTL). If a setting other than “0, 0” is specified, the specified noise elimination width is added.

Note 2. For f_{CLK} or f_{HOCO}

Note 3. Until the timer restart function starts operating, an additional clock cycle is required after the timer restart request signal is received, and an additional output delay time (10 to 40 ns) is required until the level of the timer KB output changes.

Note 4. The active level of the forced output stop function is high.

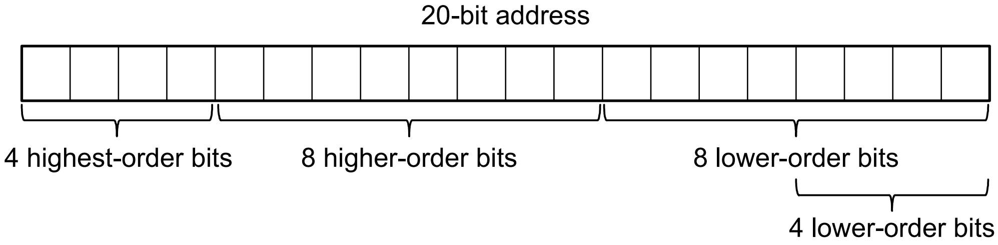
Note 5. An additional output delay time (10 to 40 ns) is required from when forced output stop function starts operating to when the level of the timer KB output changes.

Note 6. The trigger signal of the comparator to the timer KB0 is connected to internally. When the comparator n is used as the trigger signal of timer KB0, other output function that shared with the VCOUTn output pin cannot be used, but the input function can be used. When the input function that shared pin with VCOUTn is used, it requires setting the corresponding bit in the port mode register (PMxx) to 1.

Remark n = 0, 1

CHAPTER 20 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).

20.1 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 20 - 1 lists the DTC Specifications.

Table 20 - 1 DTC Specifications

Item		Specification
Activation sources		23 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 20 - 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).

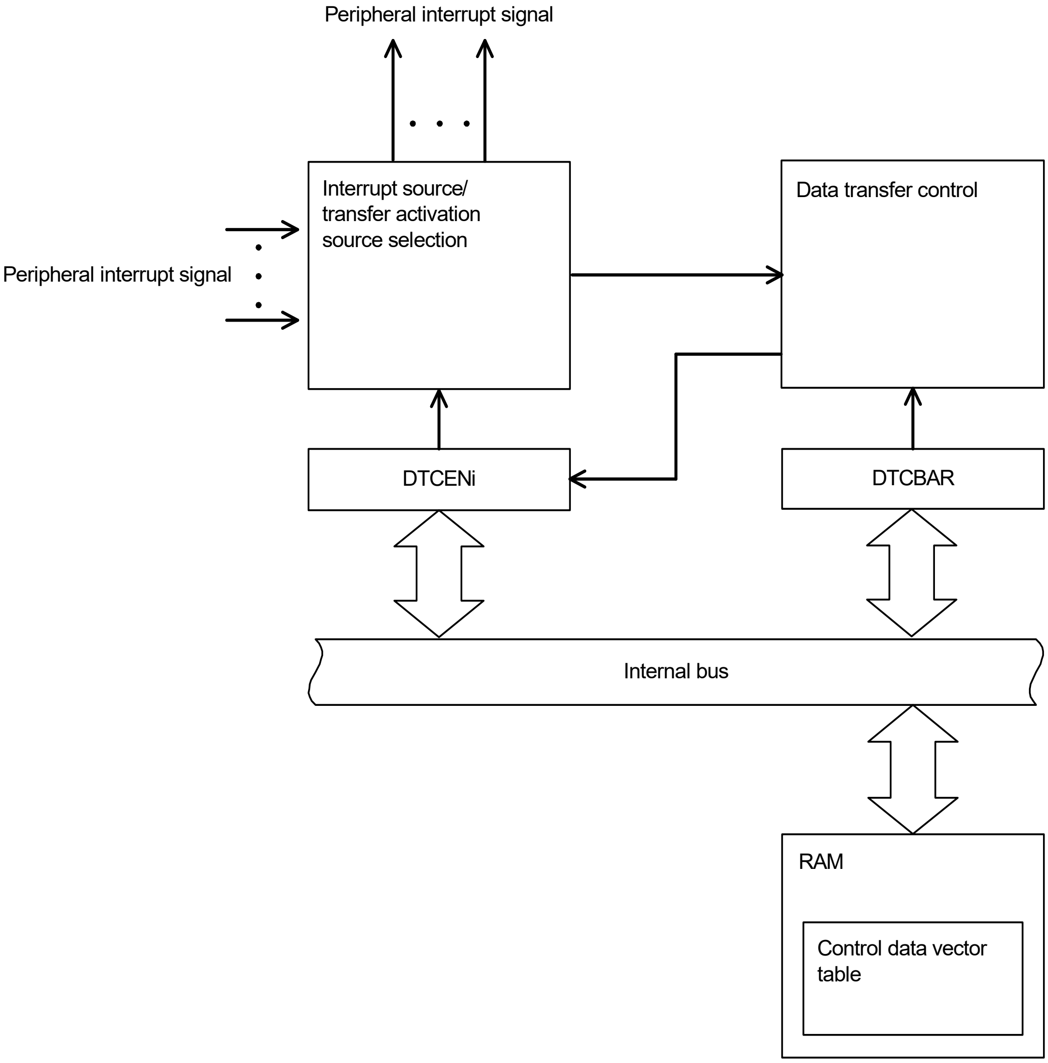
Note In the HALT mode or SNOOZE mode, these areas cannot be set as the sources for DTC transfer since the flash memory is stopped.

Remark i = 0 to 2, j = 0 to 23

20.2 Configuration of DTC

Figure 20 - 1 shows the DTC Block Diagram.

Figure 20 - 1 DTC Block Diagram


Peripheral interrupt signal
Peripheral interrupt signal
Interrupt source/ transfer activation source selection
Data transfer control
DTCENi
DTCBAR
Internal bus
RAM
Control data vector table

20.3 Registers Controlling DTC

Table 20 - 2 lists the Registers Controlling DTC.

Table 20 - 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 20 - 3 lists DTC Control Data.

DTC control data is allocated in the DTC control data area in RAM.

The DTCBAR register is used to set the 256-byte area, including the DTC control data area and the DTC vector table area where the start address for control data is stored.

Table 20 - 3 DTC Control Data

Register Name	Symbol
DTC Control Register j	DTCCRj
DTC Block Size Register j	DTBLSj
DTC Transfer Count Register j	DTCCTj
DTC Transfer Count Reload Register j	DTRLDj
DTC Source Address Register j	DTSARj
DTC Destination Address Register j	DTDARj

Remark j = 0 to 23

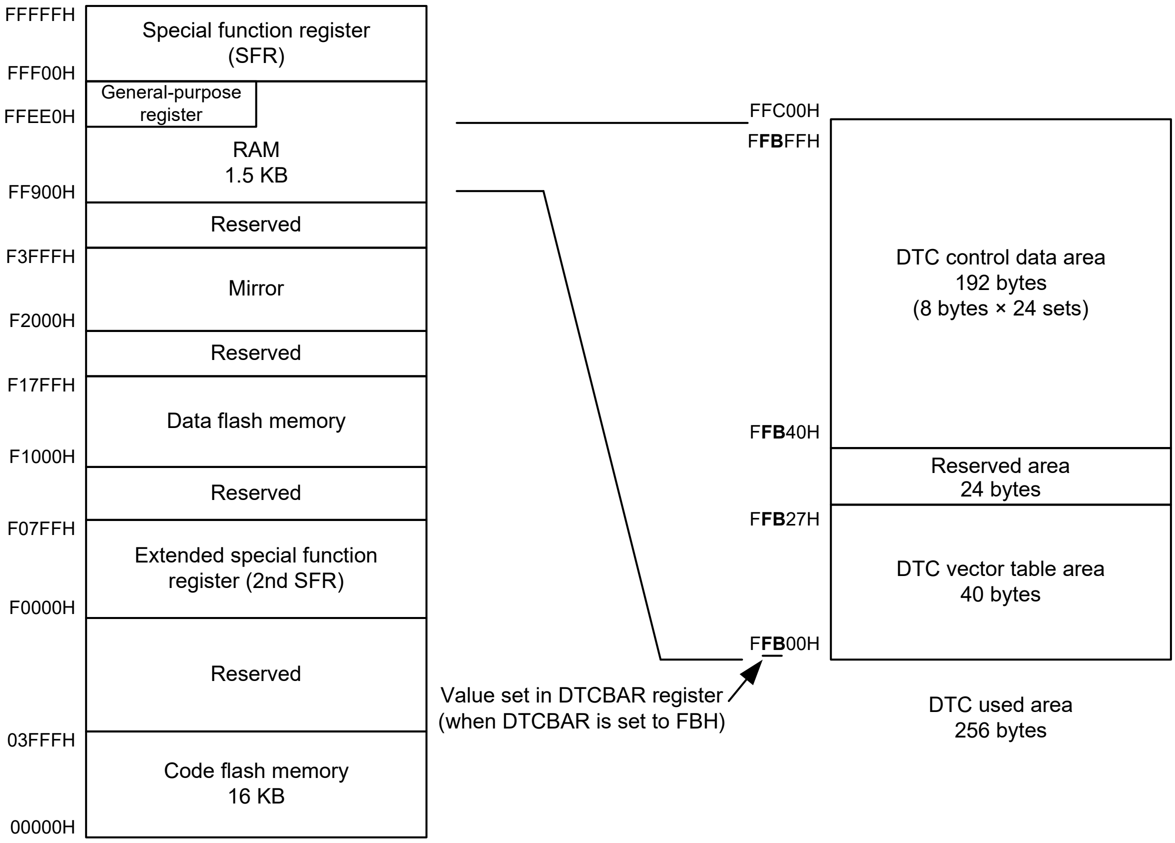
20.3.1 Allocation of DTC Control Data Area and DTC Vector Table Area

The DTCBAR register is used to set the 256-byte area where DTC control data and the vector table within the RAM area.

Figure 20 - 2 shows a Memory Map Example when DTCBAR Register is Set to FBH.

In the 192-byte DTC control data area, the space not used by the DTC can be used as RAM.

Figure 20 - 2 Memory Map Example when DTCBAR Register is Set to FBH



The areas where the DTC control data and vector table can be allocated differ depending on the 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

20.3.2 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).

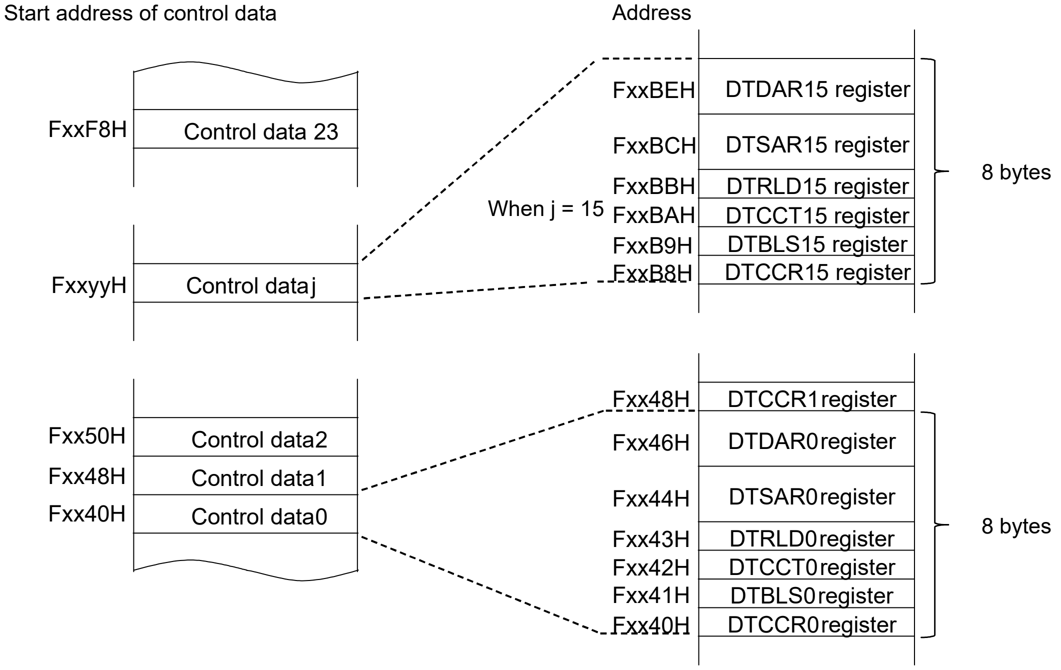
The higher 8 bits for start addresses 0 to 23 are set by the DTCBAR register, and the lower 8 bits are separately set according to the vector table assigned to each activation source.

Figure 20 - 3 shows the control data allocation.

Caution 1. Change the data in registers DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, and DTDARj when the corresponding bit among bits DTCENi0 to DTCENi7 (i = 0 to 2) in the DTCENi register is set to 0 (activation disabled).

Caution 2. Do not access DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, or DTDARj using a DTC transfer.

Figure 20 - 3 Control Data Allocation



Remark xx: Value set in DTCBAR register

Table 20 - 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

Remark xx: Value set in DTCBAR register

20.3.3 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 17H 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 20 - 5 lists the DTC Activation Sources and Vector Addresses.

Caution Change the start address of the DTC control data area to be set in the vector table when the corresponding bit among bits DTCENi0 to DTCENi7 (i = 0 to 2) in the DTCENi register is set to 0 (activation disabled).

Figure 20 - 4 Start Address of Control Data and Vector Table

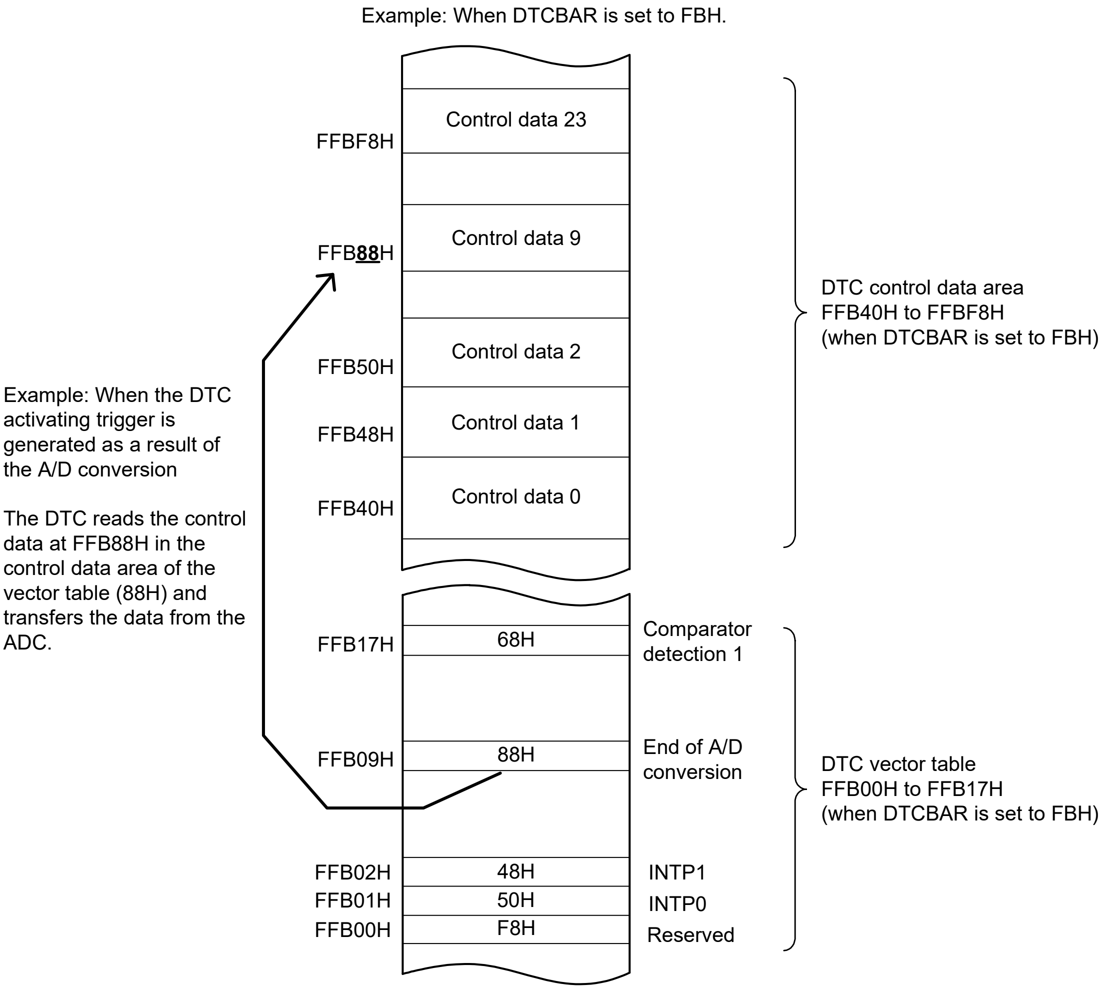

Example: When DTCBAR is set to FBH.
FFBF8H
Control data 23
FFB88H
Control data 9
FFB50H
Control data 2
FFB48H
Control data 1
FFB40H
Control data 0
DTC control data area
FFB40H to FFBF8H
(when DTCBAR is set to FBH)
Example: When the DTC activating trigger is generated as a result of the A/D conversion
The DTC reads the control data at FFB88H in the control data area of the vector table (88H) and transfers the data from the ADC.
FFB17H
68H
Comparator detection 1
FFB09H
88H
End of A/D conversion
DTC vector table
FFB00H to FFB17H
(when DTCBAR is set to FBH)
FFB02H
48H
INTP1
FFB01H
50H
INTP0
FFB00H
F8H
Reserved

Table 20 - 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	
UART1 reception transfer end/CSI11 transfer end or buffer empty/IIC11 transfer end	12	Address set in DTCBAR register +0CH	
UART1 transmission transfer end/CSI10 transfer end or buffer empty/IIC10 transfer end	13	Address set in DTCBAR register +0DH	
End of channel 0 of timer array unit 0 count or capture	14	Address set in DTCBAR register +0EH	
End of channel 1 of timer array unit 0 count or capture	15	Address set in DTCBAR register +0FH	
End of channel 2 of timer array unit 0 count or capture	16	Address set in DTCBAR register +10H	
End of channel 3 of timer array unit 0 count or capture	17	Address set in DTCBAR register +11H	
12-bit interval timer	18	Address set in DTCBAR register +12H	
8-bit interval timer 00	19	Address set in DTCBAR register +13H	
8-bit interval timer 01	20	Address set in DTCBAR register +14H	
End of TMKB count	21	Address set in DTCBAR register +15H	
Comparator detection 0	22	Address set in DTCBAR register +16H	▼
Comparator detection 1	23	Address set in DTCBAR register +17H	Lowest

20.3.4 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 20 - 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	DACEN	0	CMPEN	0	DTCEN	PGA0EN	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, 1 , 4, and 6 to 0.

20.3.5 DTC control register j (DTCCRj) (j = 0 to 23)

The DTCCRj register is used to control the DTC operating mode.

Figure 20 - 6 Format of DTC control register j (DTCCRj)

Address: Refer to **20.3.2 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.

20.3.6 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 20 - 7 Format of DTC block size register j (DTBLSj)

Address: Refer to **20.3.2 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.

20.3.7 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 20 - 8 Format of DTC transfer count register j (DTCCTj)

Address: Refer to **20.3.2 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.

20.3.8 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 20 - 9 Format of DTC transfer count reload register j (DTRLDj)

Address: Refer to **20.3.2 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.

20.3.9 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 20 - 10 Format of DTC source address register j (DTSARj)

Address: Refer to **20.3.2 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.

20.3.10 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 20 - 11 Format of DTC destination address register j (DTDARj)

Address: Refer to **20.3.2 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.

20.3.11 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 20 - 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 20 - 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 20 - 6 Correspondences between Interrupt Sources and Bits DTCENi0 to DTCENi7

Register	DTCENi7 Bit	DTCENi6 Bit	DTCENi5 Bit	DTCENi4 Bit	DTCENi3 Bit	DTCENi2 Bit	DTCENi1 Bit	DTCENi0 Bit
DTCEN0	Reserved	INTP0	INTP1	INTP2	INTP3	INTP4	INTP5	INTP6
DTCEN1	Key input	A/D conversion end	UART0 reception transfer end/ CSI01 transfer end or buffer empty/ IIC01 transfer end	UART0 transmission transfer end/ CSI00 transfer end or buffer empty/ IIC00 transfer end	UART1 reception transfer end/ CSI11 transfer end or buffer empty/ IIC11 transfer end	UART1 transmission transfer end/ CSI10 transfer end or buffer empty/ IIC10 transfer end	End of channel 0 of timer array unit 0 count or capture	End of channel 1 of timer array unit 0 count or capture
DTCEN2	End of channel 2 of timer array unit 0 count or capture	End of channel 3 of timer array unit 0 count or capture	12-bit interval timer	8-bit interval timer 00	8-bit interval timer 01	End of timer KB counter	Comparator detection 0	Comparator detection 1

Caution For the bits to which no function is assigned, be sure to set their values to 0.

Remark i = 0 to 2

20.3.12 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 20.3.1 Allocation of DTC Control Data Area and DTC Vector Table Area.

Figure 20 - 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

20.4 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.

20.4.1 Activation Sources

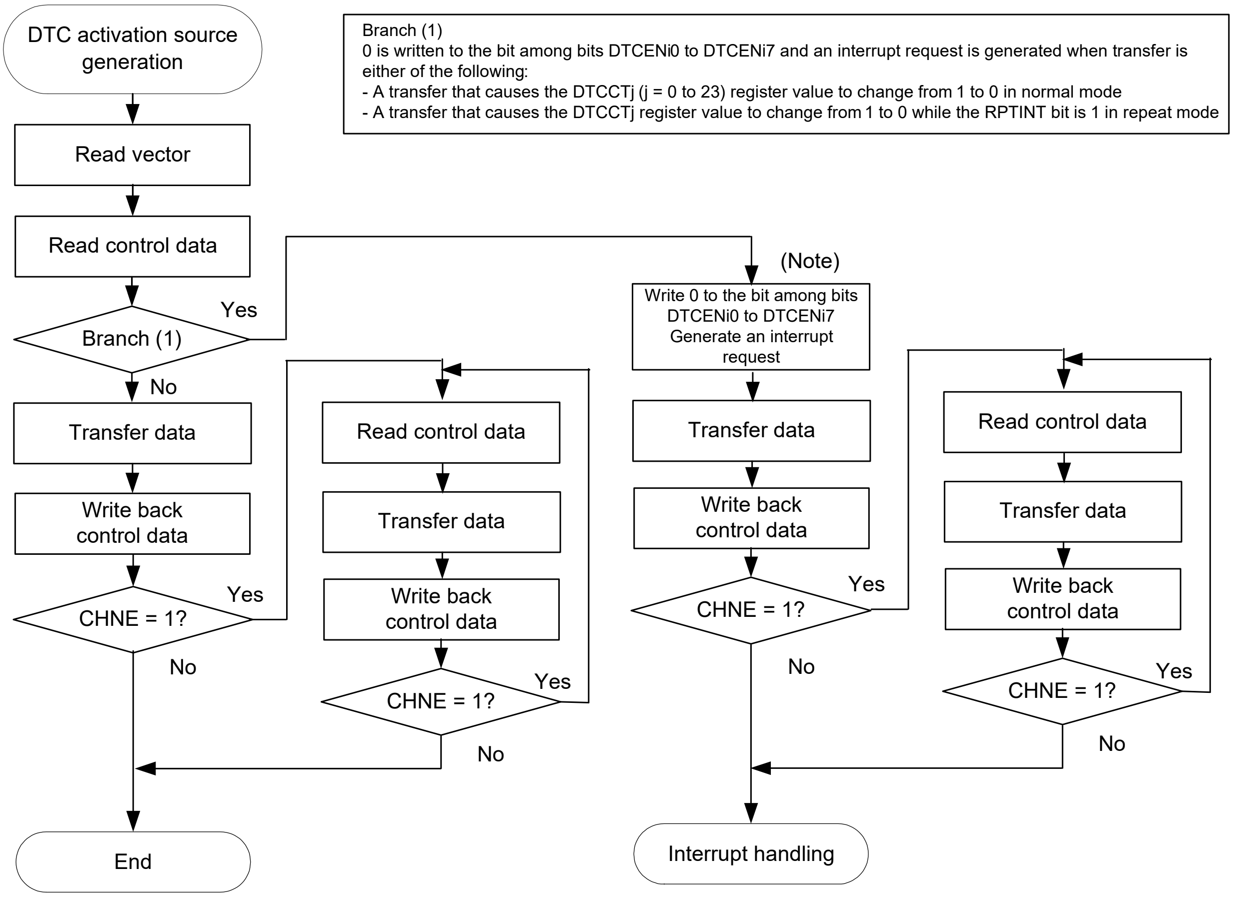
The DTC is activated by an interrupt signal from the peripheral functions. The interrupt signals to activate the DTC are selected with the DTCENi (i = 0 to 2) register.

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 20 - 14 shows the DTC Internal Operation Flowchart.

Figure 20 - 14 DTC Internal Operation Flowchart



Branch (1)
0 is written to the bit among bits DTCENi0 to DTCENi7 and an interrupt request is generated when transfer is either of the following:
- A transfer that causes the DTCCTj (j = 0 to 23) register value to change from 1 to 0 in normal mode
- A transfer that causes the DTCCTj register value to change from 1 to 0 while the RPTINT bit is 1 in repeat mode

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.

Remark DTCENi0 to DTCENi7: Bits in DTCENi (i = 0 to 2) register
RPTINT, CHNE: Bits in DTCCRj (j = 0 to 23) register

20.4.2 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 20 - 7 shows Register Functions in Normal Mode. Figure 20 - 15 shows Data Transfers in Normal Mode.

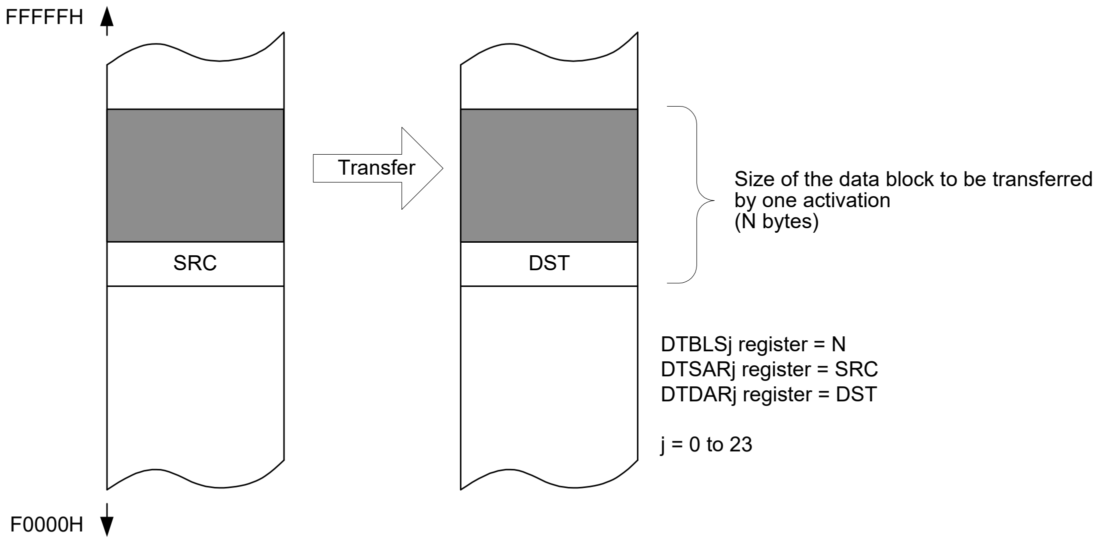
Table 20 - 7 Register Functions in Normal Mode

Register Name	Symbol	Function
DTC block size register j	DTBLSj	Size of the data block to be transferred by one activation
DTC transfer count register j	DTCCTj	Number of data transfers
DTC transfer count reload register j	DTRLDj	Not used [Note]
DTC source address register j	DTSARj	Data transfer source address
DTC destination address register j	DTDARj	Data transfer destination address

Note Initialize 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 20 - 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

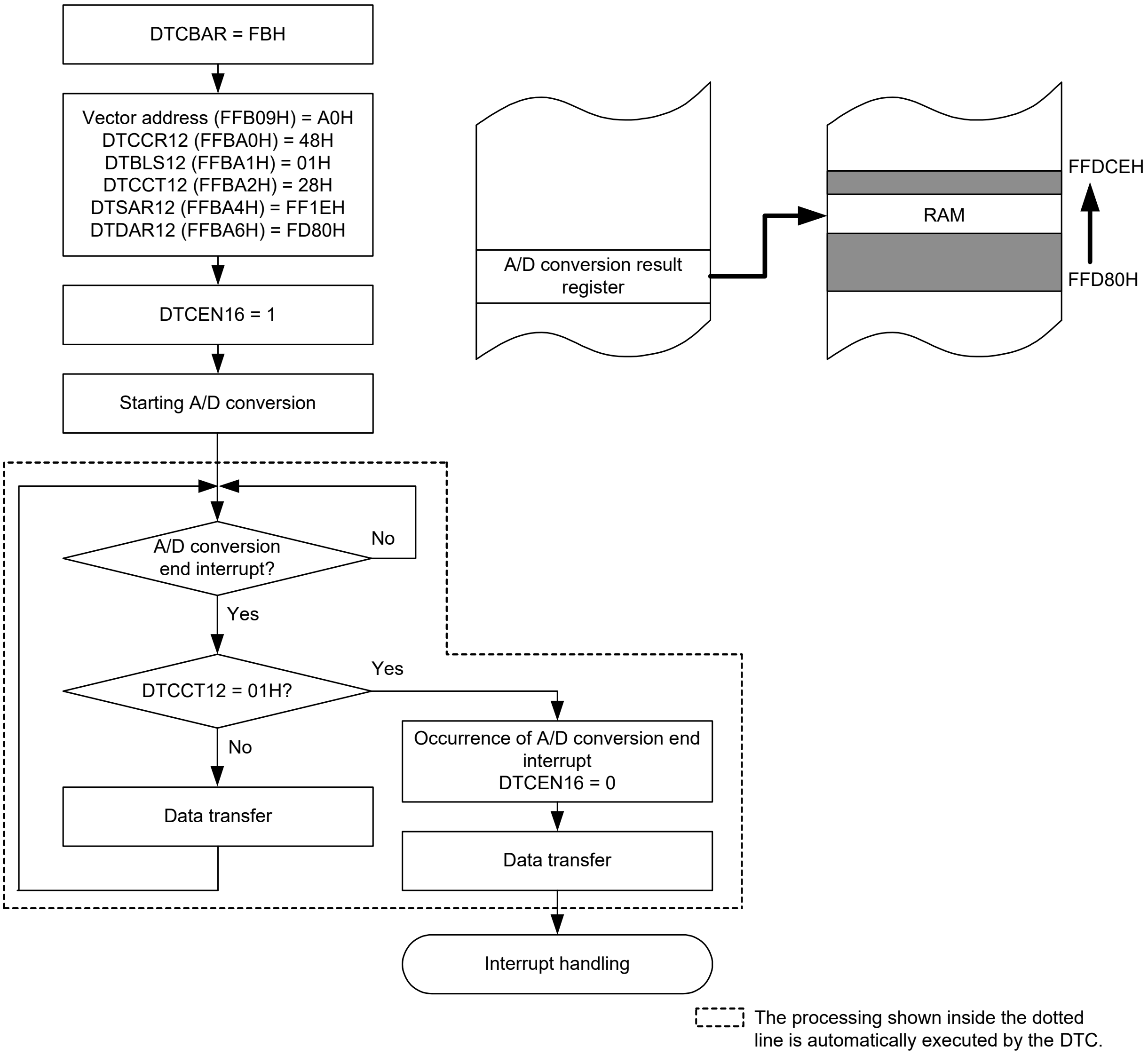
X: 0 or 1

(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 20 - 16 Example 1 of using normal mode: Consecutively capturing A/D conversion results



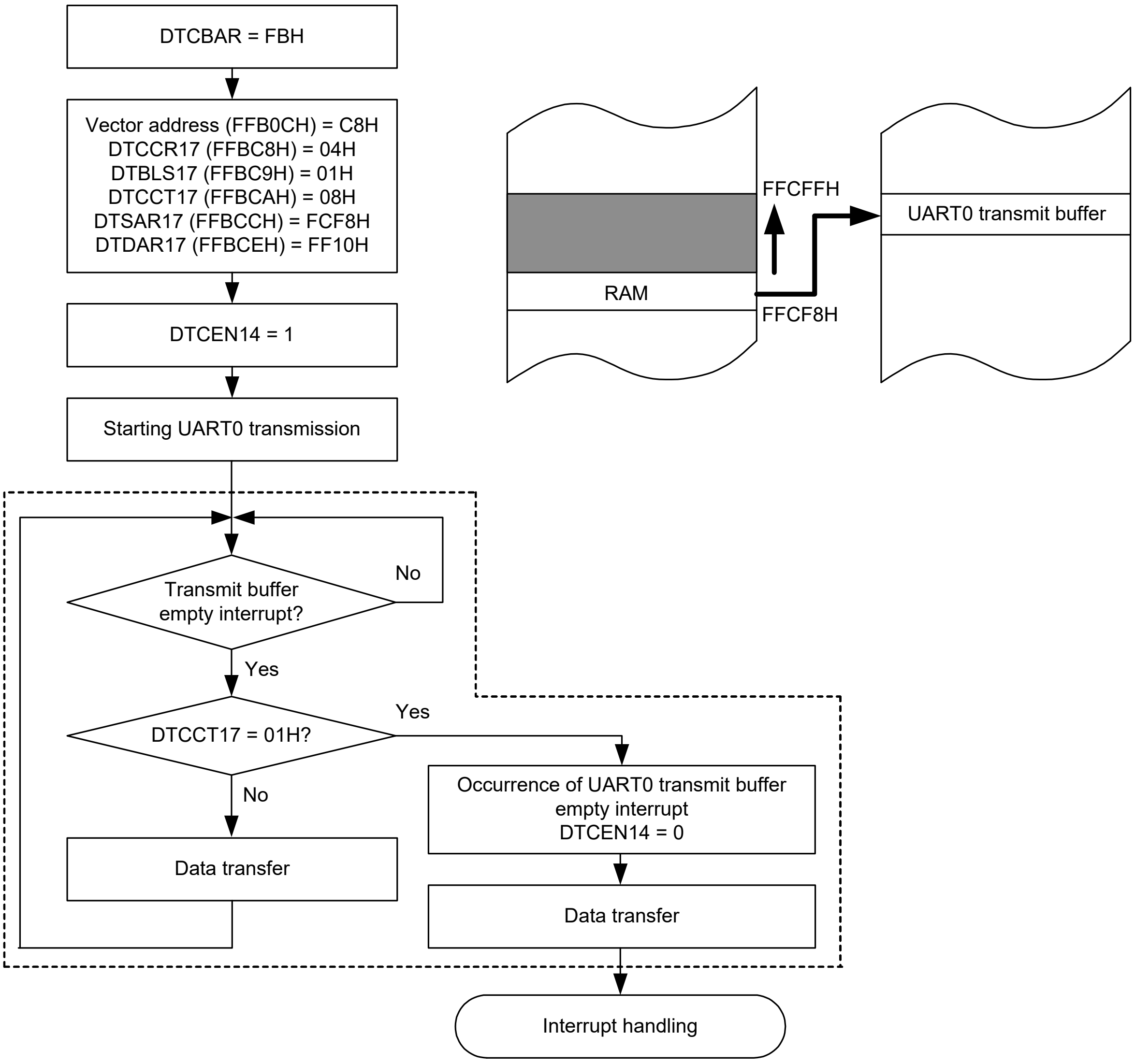
The value of the DTRLD12 register is not used because of normal mode, but initialize the register to 00H when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.

(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 20 - 17 Example 2 of using normal mode: UART0 consecutive transmission



The processing shown inside the dotted line is automatically executed by the DTC.

The value of the DTRLD17 register is not used because of normal mode, but initialize the register to 00H when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.

Start the first UART0 transmission by software. The second and subsequent transmissions are automatically sent when the DTC is activated by a transmit buffer empty interrupt.

20.4.3 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 (j = 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 in the DTCENi (i = 0 to 2) to 0 (activation disabled). When the RPTINT bit in the DTCCRj register is 0 (interrupt generation disabled), no interrupt request is generated even if the data transfer causing the DTCCTj register value to change to 0 is performed. Also, bits DTCENi0 to DTCENi7 are not set to 0.

Table 20 - 8 lists Register Functions in Repeat Mode. Figure 20 - 18 shows Data Transfers in Repeat Mode.

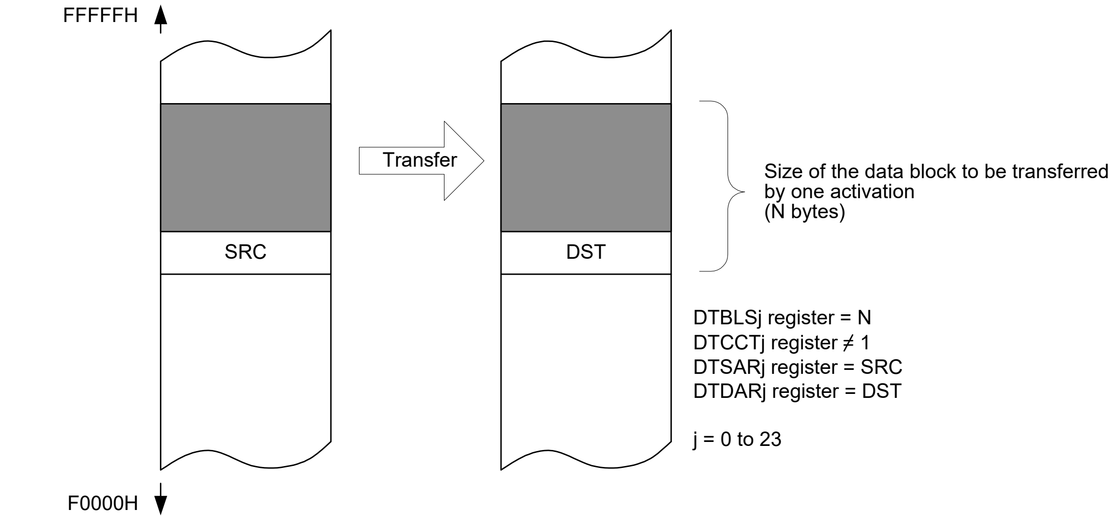
Table 20 - 8 Register Functions in Repeat Mode

Register Name	Symbol	Function
DTC block size register j	DTBLSj	Size of the data block to be transferred by one activation
DTC transfer count register j	DTCCTj	Number of data transfers
DTC transfer count reload register j	DTRLDj	This register value is reloaded to the DTCCTj register (the number of transfers is initialized).
DTC source address register j	DTSARj	Data transfer source address
DTC destination address register j	DTDARj	Data transfer destination address

Remark j = 0 to 23

Figure 20 - 18 Data Transfers in Repeat Mode

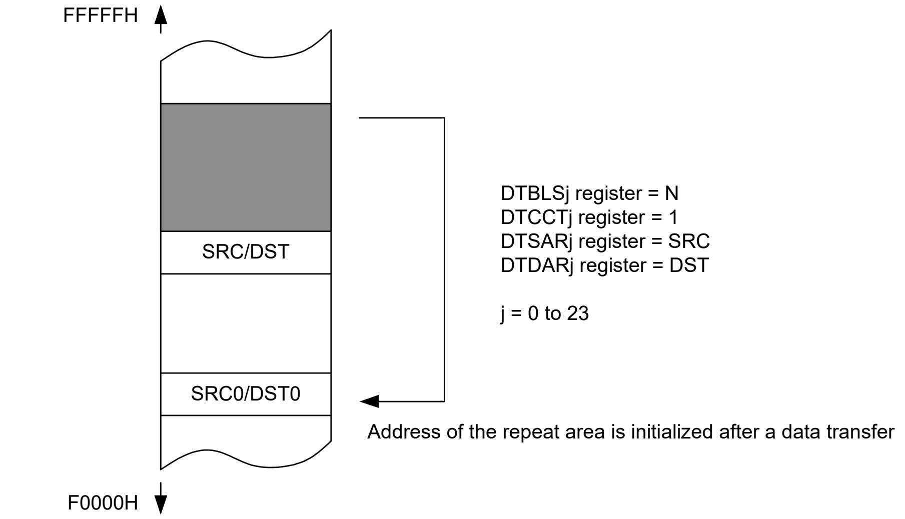
DTCCTj register ≠ 1



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.

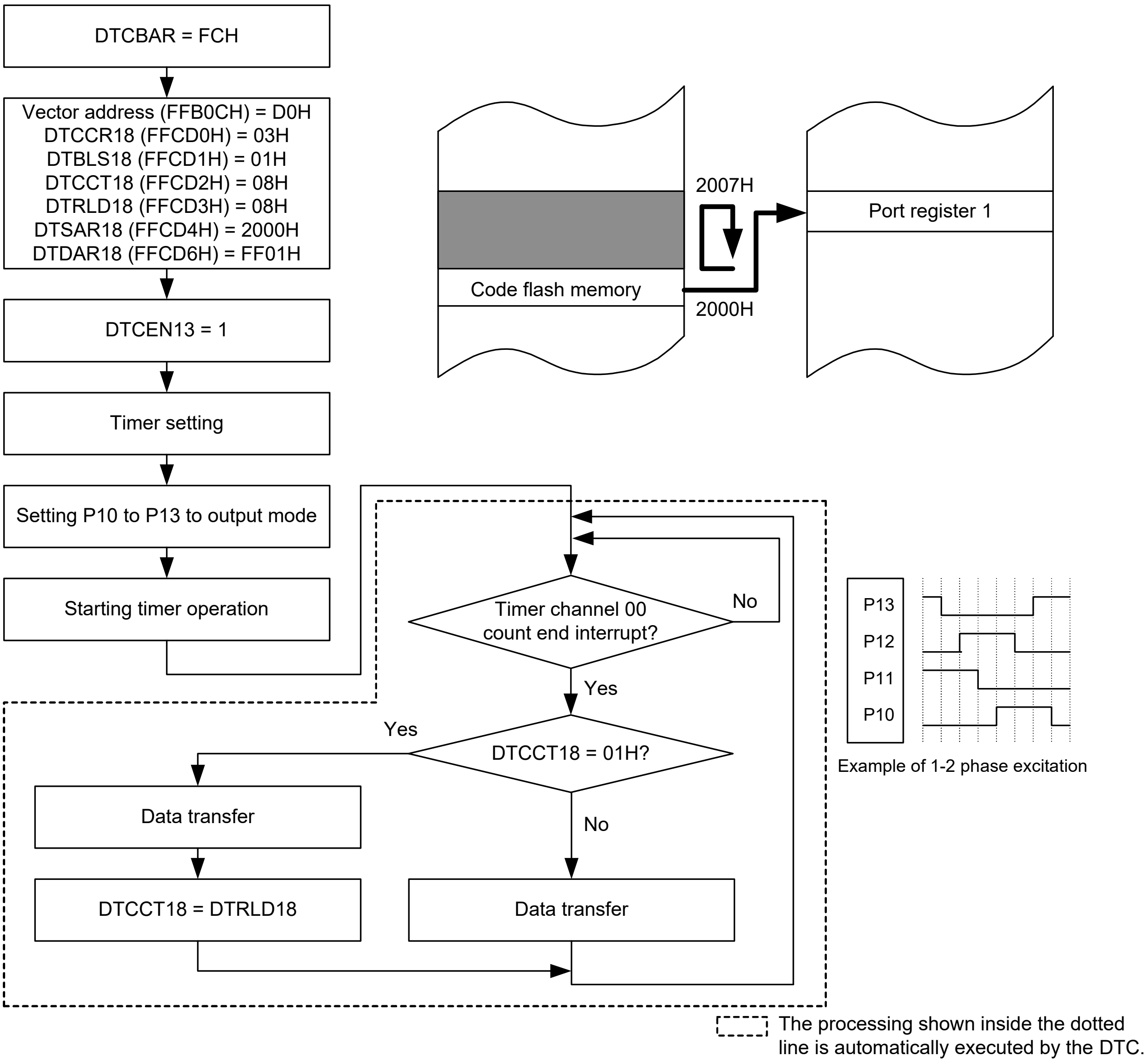
Caution 2. When repeat mode is used, the data size of the repeat area must be set to 255 bytes or less.

(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 20 - 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.

20.4.4 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.

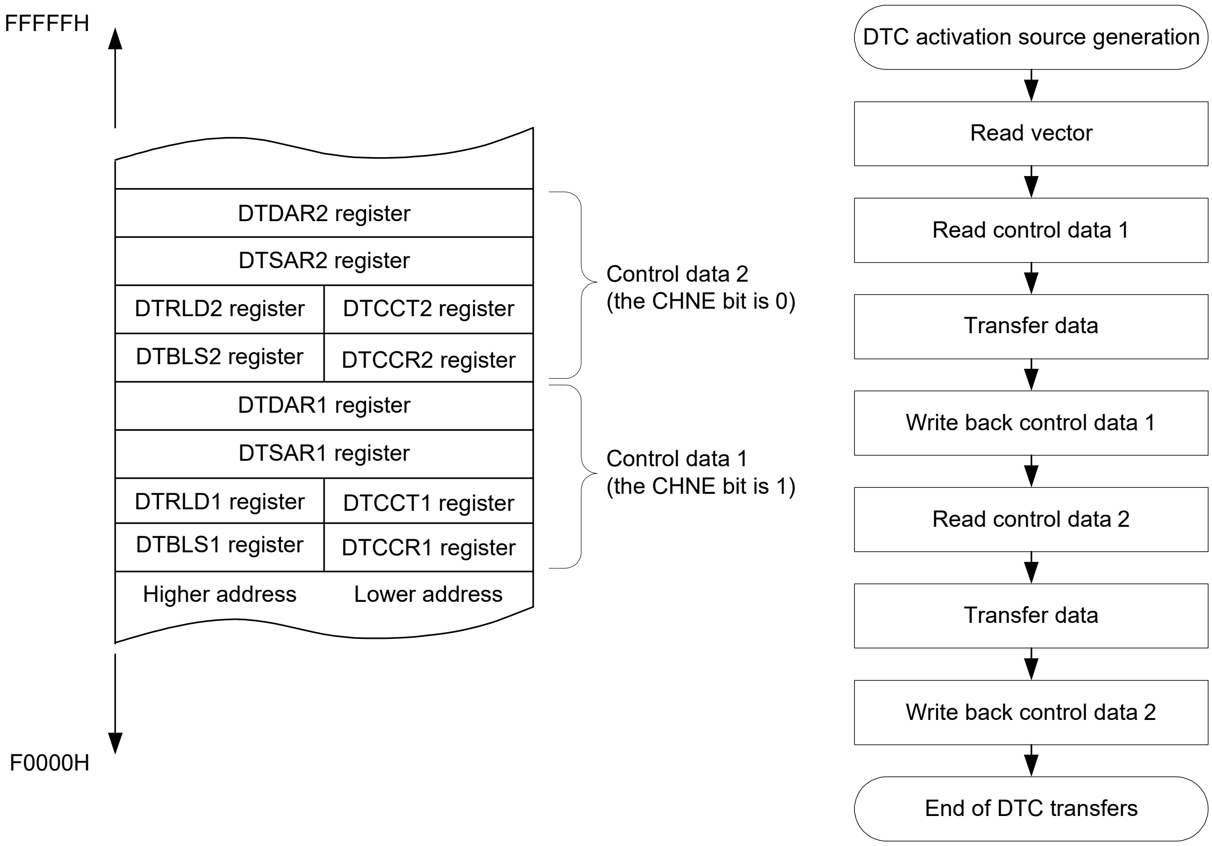
When the DTC is activated, one control data is selected according to the data read from the vector address corresponding to the activation source, and the selected control data is read from the DTC control data area.

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 20 - 20 shows Data Transfers during Chain Transfers.

Figure 20 - 20 Data Transfers during Chain Transfers



Caution 1. Set the CHNE bit in the DTCCR23 register to 0 (chain transfers disabled).

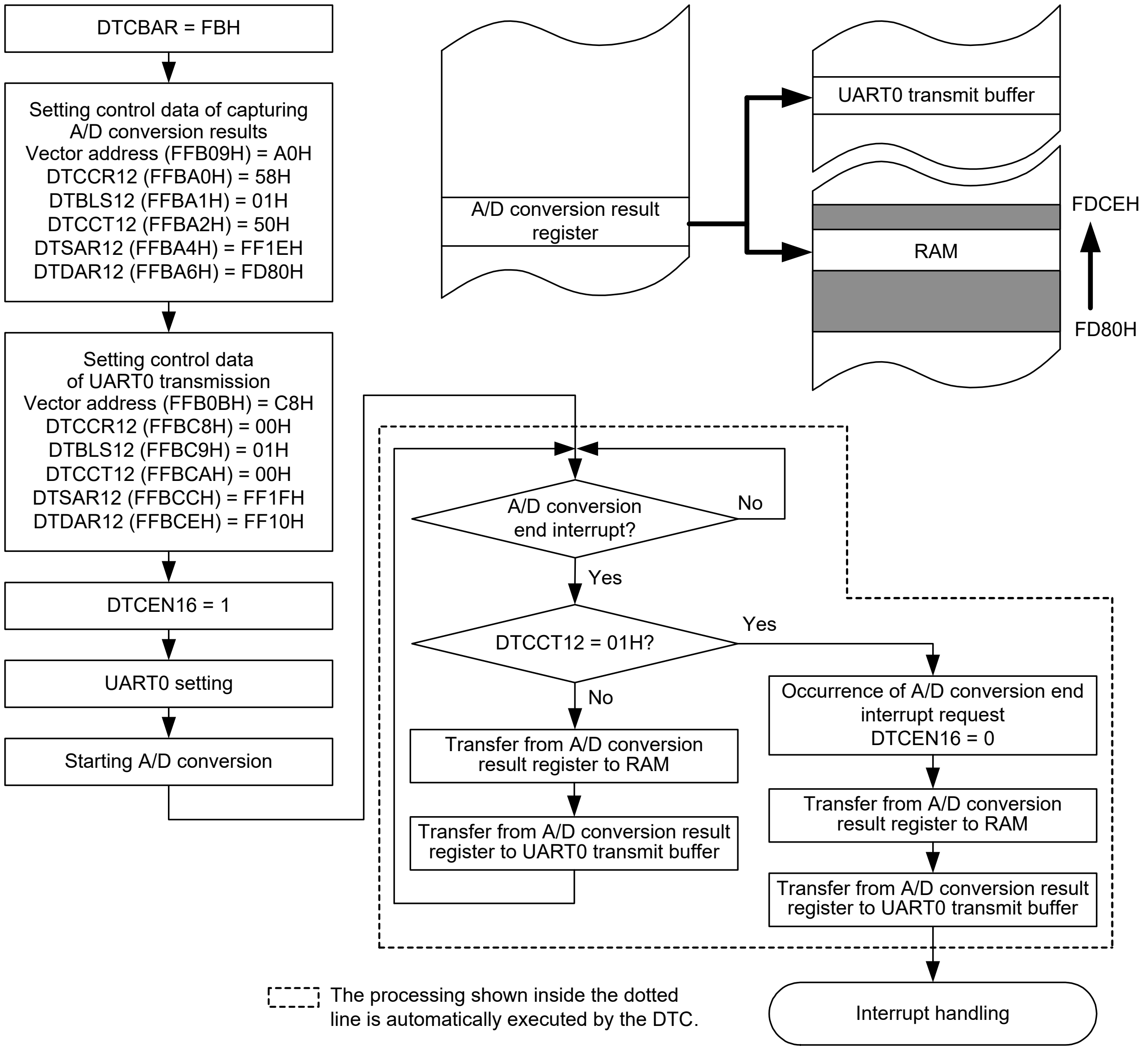
Caution 2. During chain transfers, bits DTCENi0 to DTCENi7 (i = 0 to 2) in the DTCENi register are not set to 0 (activation disabled) for the second and subsequent transfers. Also, no interrupt request is generated.

(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 20 - 21 Example of using chain transfers: Consecutively capturing A/D conversion results and UART0 transmission


DTCBAR = FBH
Setting control data of capturing A/D conversion results
Vector address (FFB09H) = A0H
DTCCR12 (FFBA0H) = 58H
DTBLS12 (FFBA1H) = 01H
DTCCT12 (FFBA2H) = 50H
DTSAR12 (FFBA4H) = FF1EH
DTDAR12 (FFBA6H) = FD80H
Setting control data of UART0 transmission
Vector address (FFB0BH) = C8H
DTCCR12 (FFBC8H) = 00H
DTBLS12 (FFBC9H) = 01H
DTCCT12 (FFBCAH) = 00H
DTSAR12 (FFBCCH) = FF1FH
DTDAR12 (FFBCEH) = FF10H
DTCEN16 = 1
UART0 setting
Starting A/D conversion
A/D conversion result register
UART0 transmit buffer
RAM
FDCEH
FD80H
A/D conversion end interrupt?
No
Yes
DTCCT12 = 01H?
Yes
No
Transfer from A/D conversion result register to RAM
Transfer from A/D conversion result register to UART0 transmit buffer
Occurrence of A/D conversion end interrupt request
DTCEN16 = 0
Transfer from A/D conversion result register to RAM
Transfer from A/D conversion result register to UART0 transmit buffer
Interrupt handling
The processing shown inside the dotted line is automatically executed by the DTC.

20.5 Notes on DTC

20.5.1 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 (j = 0 to 23) 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.

20.5.2 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.
- FFE20H-FFEDFH of the internal RAM area cannot be used as the DTC control data area or DTC vector table area when using the self-programming and data-flash functions.
- 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.

20.5.3 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.

20.5.4 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
```

20.5.5 Number of DTC Execution Clock Cycles

Table 20 - 9 lists the Operations Following DTC Activation and Required Number of Cycles for each operation.

Table 20 - 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 20 - 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 20 - 11 Number of Clock Cycles Required for One Data Read/Write Operation**.

Table 20 - 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 20 - 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.

20.5.6 DTC Response Time

Table 20 - 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 20 - 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 **20.5.3 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/fCLK (fCLK: CPU/peripheral hardware clock)

20.5.7 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 **20.3.3 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.
  - 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

Remark n = 0 to 1

20.5.8 Operation in Standby Mode Status

Status	DTC Operation
HALT mode	Operable
STOP mode	DTC activation sources can be accepted Note 1
SNOOZE mode	Operable Notes 2, 3, 4, 5

Note 1. In the STOP mode, detecting a DTC activation source enables transition to SNOOZE mode and DTC transfer. After completion of transfer, the system returns to the STOP mode. However, since the code flash memory and the data flash memory are stopped during the SNOOZE mode, the flash memory cannot be set as the transfer source.

Note 2. 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 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).

Remark p = 00; q = 0; m = 0

CHAPTER 21 EVENT LINK CONTROLLER (ELC)

21.1 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.

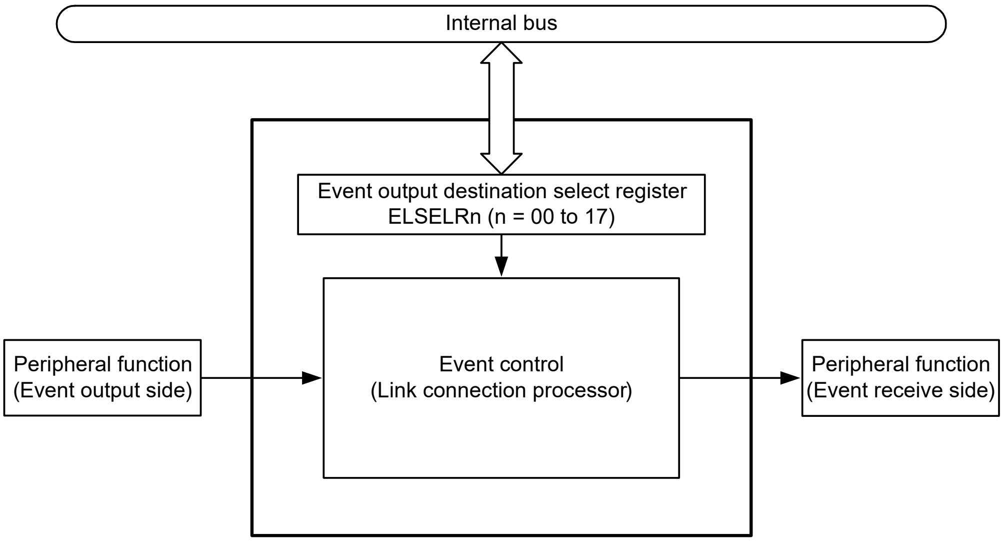
The ELC has the following functions.

- Capable of directly linking event signals from 18 types of peripheral functions to specified peripheral functions
- Event signals can be used as activation sources for operating any one of four types of peripheral functions

21.2 Configuration of ELC

Figure 21 - 1 shows the ELC Block Diagram.

Figure 21 - 1 ELC Block Diagram



21.3 Registers Controlling ELC

Table 21 - 1 lists the Registers Controlling ELC.

Table 21 - 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	ELSELR03
Event output destination select register 04	ELSELR04
Event output destination select register 05	ELSELR05
Event output destination select register 06 [Note]	ELSELR06
Event output destination select register 07	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

Note Do not set any value other than the initial value (event link disabled) in 20-pin products.

21.3.1 Event output destination select register n (ELSELRn) (n = 00 to 17)

An ELSELRn register links each event signal to an operation of an event-receiving peripheral function (link destination peripheral function) after reception.

Do not set multiple event inputs to the same event output destination (event receive side). The operation of the event-receiving peripheral function will become undefined, and event signals may not be received correctly. In addition, do not set the event link generation source and the event link output destination to the same function.

Set an ELSELRn register during a period when no event output peripheral functions are generating event signals and the function of the event output destination (event receive side) is stopped.

Table 21 - 2 lists the Correspondence Between ELSELRn (n = 00 to 17) Registers and Peripheral Functions, and Table 21 - 3 lists the Correspondence Between Values Set to ELSELRn (n = 00 to 17) Registers and Operation of Link Destination Peripheral Functions at Reception.

Figure 21 - 2 Format of Event output destination select register n (ELSELRn)

Address: F0240H (ELSELR00) to F0251H (ELSELR17) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ELSELRn	0	0	0	0	0	ELSELn2	ELSELn1	ELSELn0

ELSELn2	ELSELn1	ELSELn0	Event Link Selection
0	0	0	Event link disabled
0	0	1	Select operation of peripheral function 1 to link Note
0	1	0	Select operation of peripheral function 2 to link Note
0	1	1	Select operation of peripheral function 3 to link Note
1	0	0	Select operation of peripheral function 4 to link Note
Other than the above			Setting prohibited

Note See **Table 21 - 3 Correspondence Between Values Set to ELSELRn (n = 00 to 17) Registers and Operation of Link Destination Peripheral Functions at Reception**.

Table 21 - 2 Correspondence Between ELSELRn (n = 00 to 17) 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	12-bit interval timer interval signal detection	INTIT
ELSELR09	8-bit interval timer channel 00 compare match or 16-bit interval timer channel 0 compare match (cascaded)	INTIT00
ELSELR10	8-bit interval timer channel 01 compare match	INTIT01
ELSELR11	TAU channel 00 count end/capture end	INTTM00
ELSELR12	TAU channel 01 count end/capture end	INTTM01
ELSELR13	TAU channel 02 count end/capture end	INTTM02
ELSELR14	TAU channel 03 count end/capture end	INTTM03
ELSELR15	Comparator detection 0	INTCMP0
ELSELR16	Comparator detection 1	INTCMP1
ELSELR17	TMKB trigger output	INTTMKB0

Table 21 - 3 Correspondence Between Values Set to ELSELRn (n = 00 to 17) Registers and Operation of Link Destination Peripheral Functions at Reception

Bits ELSELn2 to ELSELn0 in ELSELRn Register	Link Destination Number	Link Destination Peripheral Function	Operation When Receiving Event
001B	1	A/D converter	A/D conversion starts
010B	2	Timer input of timer array unit 0 channel 0 Note 1	Delay counter, input pulse interval measurement, external event counter
011B	3	Timer input of timer array unit 0 channel 1 Note 2	Delay counter, input pulse interval measurement, external event counter
100B	4	D/A converter channel 1	D/A conversion 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).

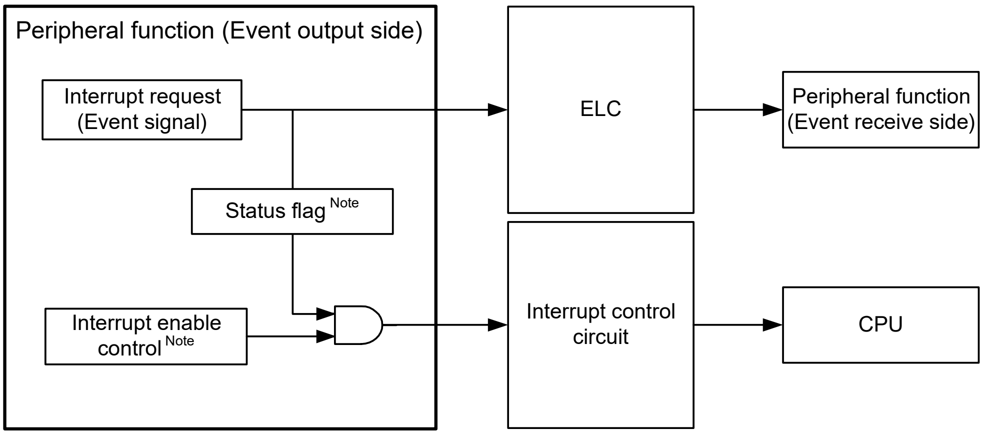
21.4 ELC Operation

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 21 - 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 21 - 3 Correspondence Between Values Set to ELSELRn (n = 00 to 17) Registers and Operation of Link Destination Peripheral Functions at Reception**).

Figure 21 - 3 Relationship Between Interrupt Handling and ELC


Peripheral function (Event output side)
Interrupt request (Event signal)
Status flag Note
Interrupt enable control Note
ELC
Interrupt control circuit
Peripheral function (Event receive side)
CPU

Note Not available depending on the peripheral function.

Table 21 - 4 lists the Response of Peripheral Functions That Receive Events.

Table 21 - 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	D/A converter channel 1	D/A conversion starts	An event from the ELC is directly used as a hardware trigger of D/A conversion.

CHAPTER 22 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.

		10-pin	16-pin	20-pin	24, 25-pin
Maskable interrupts	External	3	9	10	13
	Internal	20	25	26	26

22.1 Interrupt Function Types

The following two types of interrupt functions are used.

(1) Maskable interrupts

These interrupts undergo mask control. Maskable interrupts can be divided into four priority groups by setting the priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H).
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 22 - 1**.
A standby release signal is generated and STOP, HALT, and SNOOZE modes are released.
External interrupt requests and internal interrupt requests are provided as maskable interrupts.

(2) Software interrupt

This is a vectored interrupt generated by executing the BRK instruction. It is acknowledged even when interrupts are disabled. The software interrupt does not undergo interrupt priority control.

22.2 Interrupt Sources and Configuration

Interrupt sources include maskable interrupts and software interrupts. In addition, they also have up to seven reset sources (see **Table 22 - 1**). The vector codes that store the program start address when branching due to the generation of a reset or various interrupt requests are two bytes each, so interrupts jump to a 64 K address of 00000H to 0FFFFH.

Table 22 - 1 Interrupt Source List (1/2)

Interrupt Type	Default Priority Note 1	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type Note 2	24, 25-pin	20-pin	16-pin	10-pin
		Name	Trigger							
Maskable	0	INTWDTI	Watchdog timer interval Note 3 (75% of overflow time + 1/2 f_{IL})	Internal	00004H	(A)	√	√	√	√
	1	INTLVI	Voltage detection Note 4		00006H		√	√	√	√
	2	INTP0	Pin input edge detection	External	00008H	(B)	√	√	√	—
	3	INTP1			0000AH		√	√	√	√
	4	INTP2			0000CH		√	√	√	—
	5	INTP3			0000EH		√	√	√	—
	6	INTP4			00010H		√	√	√	—
	7	INTP5			00012H		√	√	—	—
	8	INTP6			00014H		√	—	—	—
	9	INTST0	UART0 transmission transfer end or buffer empty interrupt	Internal	00016H	(A)	√	√	√	—
		INTCSI00	CSI00 transfer end or buffer empty interrupt				√	√	—	—
		INTIIC00	IIC00 transfer end				√	√	—	—
	10	INTSR0	UART0 reception transfer end		00018H		√	√	√	—
		INTCSI01	CSI01 transfer end or buffer empty interrupt				√	—	—	—
		INTIIC01	IIC01 transfer end				√	—	—	—
	11	INTSRE0	UART0 reception communication error occurrence		0001EH		√	√	√	—
	12	INTTM00	End of TAU channel 00 count or capture (at 16-bit operation or lower 8-bit operation)		00020H		√	√	√	√
	13	INTST1	End of UART1 transmission		00022H		√	√	√	√
		INTCSI10	End of CSI10 communication				√	√	√	√
		INTIIC10	End of IIC10 communication				√	√	—	—
	14	INTSR1	End of UART1 reception		00024H		√	√	√	√
		INTCSI11	End of CSI11 communication				√	√	√	—
		INTIIC11	End of IIC11 communication				√	√	√	—
	15	INTSRE1	UART1 reception communication error occurrence		00026H		√	√	√	√
	16	INTIICA0	End of IICA0 communication		00028H		√	√	√	—
	17	INTTM01H	End of TAU channel 01 count or capture (at higher 8-bit operation)		0002AH		√	√	√	√
	18	INTTM03H	End of TAU channel 03 count or capture (at higher 8-bit operation)		0002CH		√	√	√	√
	19	INTTM01	End of TAU channel 01 count or capture (at 16-bit operation or lower 8-bit operation)		0002EH		√	√	√	√
	20	INTTM02	End of TAU channel 02 count or capture (at 16-bit operation or lower 8-bit operation)		00030H		√	√	√	√

Note 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 38 indicates the lowest priority.

Note 2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figures 22 - 1.

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 22 - 1 Interrupt Source List (2/2)

Interrupt Type	Default Priority [Note 1]	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type [Note 2]	24, 25-pin	20-pin	16-pin	10-pin
		Name	Trigger							
Maskable	21	INTTM03	End of TAU channel 03 count or capture (at 16-bit operation or lower 8-bit operation)	Internal	00032H	(A)	√	√	√	√
	22	INTAD	End of A/D conversion		00034H		√	√	√	√
	23	INTIT	12-bit interval timer interval signal detection		00036H		√	√	√	√
	24	INTKR	Key return signal detection	External	00038H	(C)	√	√	√	—
	25	INTP7	External interrupt edge detection 7	External	0003AH	(B)	√	—	—	—
	26	INTP8	External interrupt edge detection 8		0003CH		√	—	—	—
	27	INTP9	External interrupt edge detection 9		0003EH		√	√	√	√
	28	INTP10	External interrupt edge detection 10		00040H		√	√	√	√
	29	INTP11	External interrupt edge detection 11		00042H		√	√	√	—
	30	INTCMP0	Comparator detection 0	Internal	00044H	(A)	√	√	√	√
	31	INTCMP1	Comparator detection 1		00046H		√	√	√	—
	32	INTDOC	DOC operation result detection		00048H		√	√	√	√
	33	INTIT00	8-bit interval timer channel 00 compare match or 16-bit interval timer channel 0 compare match (cascaded)		0004AH		√	√	√	√
	34	INTIT01	8-bit interval timer channel 01		0004CH		√	√	√	√
	35	INTTMKB0	End of TMKB counter		0004EH		√	√	√	√
	36	INTIICA1	End of IICA1 communication		00050H		√	√	—	—
	37	INTFL	Reserved		00052H		√	√	√	√
	38	INTFO	Interrupt flag output signal		00054H		√	√	√	√
Software	—	BRK	Execution of BRK instruction	—	0007EH	(D)	√	√	√	√
Reset	—	RESET	$\overline{\text{RESET}}$ pin input	—	00000H	—	√	√	√	√
		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 38 indicates the lowest priority.

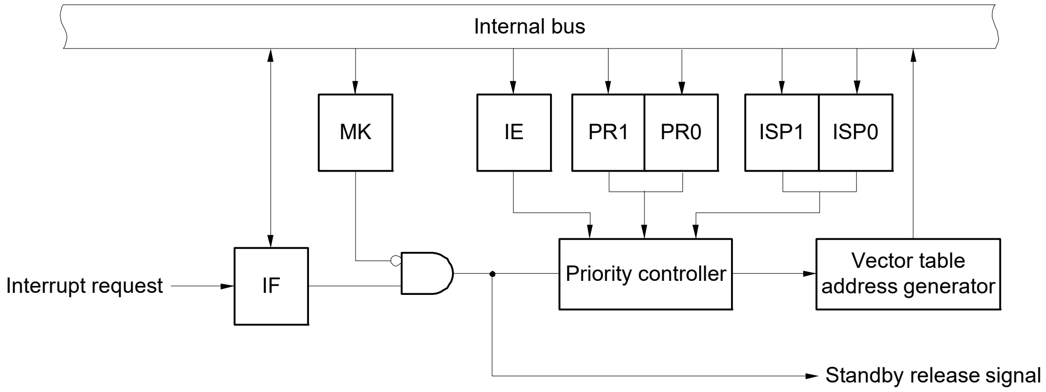
Note 2. Basic configuration types (A) to (D) correspond to (A) to (D) in Figures 22 - 1.

Note 3. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is set to 1.

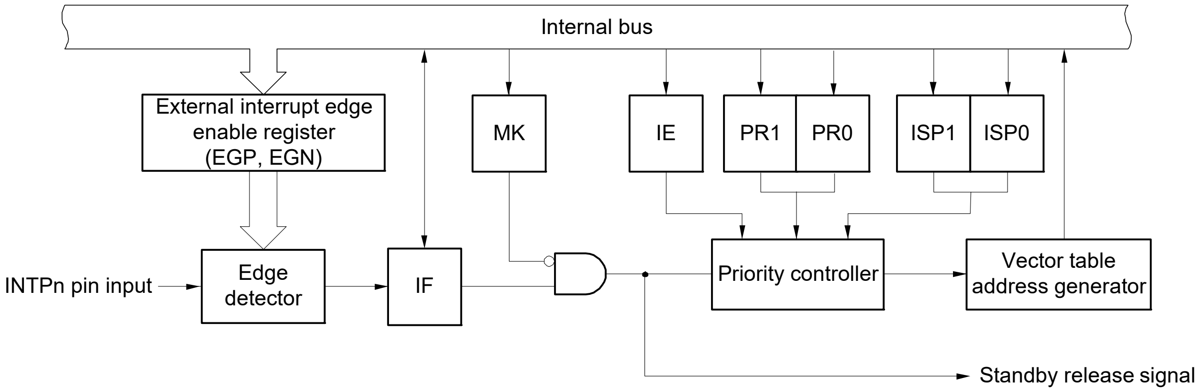
Note 4. When the instruction code in FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Figure 22 - 1 Basic Configuration of Interrupt Function (1/2)

(A) Internal maskable interrupt



(B) External maskable interrupt (INTPn)

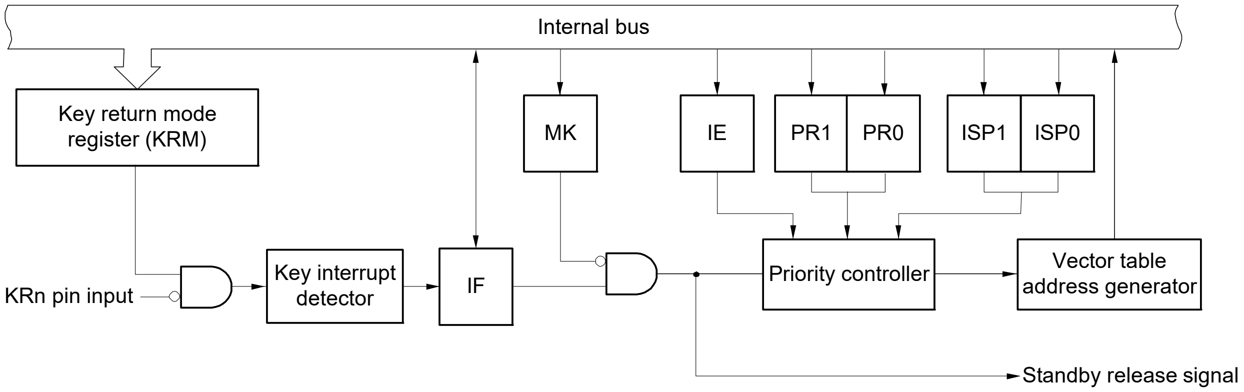


IF:	Interrupt request flag
IE:	Interrupt enable flag
ISP0:	In-service priority flag 0
ISP1:	In-service priority flag 1
MK:	Interrupt mask flag
PR0:	Priority specification flag 0
PR1:	Priority specification flag 1

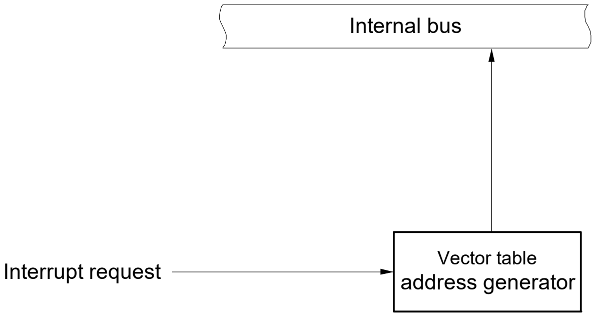
Remark	10-pin:	n = 1, 9, 10
	16-pin:	n = 0 to 4, 9 to 11
	20-pin:	n = 0 to 5, 9 to 11
	24, 25-pin:	n = 0 to 11

Figure 22 - 1 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 n = 0 to 7

22.3 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, IF2H)
- Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)
- Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H)
- External interrupt rising edge enable register (EGP0)
- External interrupt falling edge enable register (EGN0)
- Interrupt flag output control register (INTFE)
- Program status word (PSW)

Table 22 - 2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 22 - 2 Flags Corresponding to Interrupt Request Sources (1/2)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		24, 25-pin	20-pin	16-pin	10-pin
		Register		Register		Register				
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L	√	√	√	√
INTLVI	LVIIF		LVIMK		LVIPR0, LVIPR1		√	√	√	√
INTP0	PIF0		PMK0		PPR00, PPR10		√	√	√	—
INTP1	PIF1		PMK1		PPR01, PPR11		√	√	√	√
INTP2	PIF2		PMK2		PPR02, PPR12		√	√	√	—
INTP3	PIF3		PMK3		PPR03, PPR13		√	√	√	—
INTP4	PIF4		PMK4		PPR04, PPR14		√	√	√	—
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		√	√	√	—
INTCSI01Note 2	CSIIF01		CSIMK01		CSIPR001, CSIPR101		√	—	—	—
INTIIC01 Note 2	IICIF01		IICMK01		IICPR001, IICPR101		√	—	—	—
INTSRE0	SREIF0		SREMK0		SREPR00, SREPR10		√	√	√	—
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100		√	√	√	√
INTST1	STIF1		STMK1		STPR01, STPR11		√	√	√	√
INTCSI10	CSIIF10		CSIMK10		CSIPR010, CSIPR110		√	√	√	√
INTIIC10	IICIF10		IICMK10		IICPR010, IICPR110		√	√	—	—
INTSR1	SRIF1	IF1L	SRMK1	MK1L	SRPR01, SRPR11	PR01L, PR11L	√	√	√	√
INTCSI11	CSIIF11		CSIMK11		CSIPR011, CSIPR111		√	√	√	—
INTIIC11	IICIF11		IICMK11		IICPR011, IICPR111		√	√	√	—
INTSRE1	SREIF1		SREMK1		SREPR01, SREPR11		√	√	√	√
INTIICA0	IICAIF0		IICAMK0		IICAPR00, IICAPR10		√	√	√	—
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	√	√	√	√
INTIT	TMKAIF		TMKAMK		TMKAPR0, TMKAPR1		√	√	√	√
INTKR	KRIF		KRMK		KRPR0, KRPR1		√	√	√	—
INTP7	PIF7		PMK7		PPR07, PPR17		√	—	—	—
INTP8	PIF8		PMK8		PPR08, PPR18		√	—	—	—
INTP9	PIF9		PMK9		PPR09, PPR19		√	√	√	√
INTP10	PIF10		PMK10		PPR010, PPR110		√	√	√	√

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 22 - 2 Flags Corresponding to Interrupt Request Sources (2/2)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		24, 25-pin	20-pin	16-pin	10-pin
		Register		Register		Register				
INTP11	PIF11	IF1H	PMK11	MK1H	PPR011, PPR111	PR01H, PR11H	√	√	√	—
INTCMP0	CMPIF0	IF2L	CMPMK0	MK2L	CMPPR00, CMPPR01	PR02L, PR12L	√	√	√	√
INTCMP1	CMPIF1		CMPMK1		CMPPR01, CMPPR11		√	√	√	—
INTDOC	DOCIF		DOCMK		DOCPR0, DOCPR1		√	√	√	√
INTIT00	ITIF00		ITMK00		ITPR000, ITPR100		√	√	√	√
INTIT01	ITIF01		ITMK01		ITPR001, ITPR101		√	√	√	√
INTTMKB0	TMKBIF0		TMKBMK0		TMKBPR00, TMKBPR10		√	√	√	√
INTIICA1	IICAIF1		IICAMK1		IICAPR01, IICAPR11		√	√	—	—
INTFL	FLIF		FLMK		FLPR0, FLPR1		√	√	√	√
INTFO	FOIF	IF2H	FOMK	MK2H	FOPR0, FOPR1	PR02H, PR12H	√	√	√	√

22.3.1 Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)

The interrupt request flags are set to 1 when the corresponding interrupt request is generated or an instruction is executed. They are cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon reset signal generation.
When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is entered.
The IF0L, IF0H, IF1L, IF1H, IF2L, and IF2H 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 22 - 2 Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)

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	STIF1 CSIIF10 IICIF10	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	TMIF03	TMIF02	TMIF01	TMIF03H	TMIF01H	IICAIF0	SREIF1	SRIF1 CSIIF11 IICIF11

Address: FFFE3H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1H	PIF11	PIF10	PIF9	PIF8	PIF7	KRIF	TMKAIF	ADIF

Address: FFFD0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2L	FLIF	IICAIF1	TMKBIF0	ITIF01	ITIF00	DOCIF	CMPIF1	CMPIF0

Address: FFFD1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
IF2H	0	0	0	0	0	0	0	FOIF

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 22 - 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.

22.3.2 Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt.

The MK0L, MK0H, MK1L, MK1H, MK2L, and MK2H 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 22 - 3 Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)

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	STMK1 CSIMK10 IICMK10	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	TMMK03	TMMK02	TMMK01	TMMK03H	TMMK01H	IICAMK0	SREMK1	SRMK1 CSIMK11 IICMK11

Address: FFFE7H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1H	PMK11	PMK10	PMK9	PMK8	PMK7	KRMK	TMKAMK	ADMK

Address: FFFD4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2L	FLMK	IICAMK1	TMKBMK0	ITMK01	ITMK00	DOCMK	CMPMK1	CMPMK0

Address: FFFD5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2H	1	1	1	1	1	1	1	FOMK

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 22 - 2. Be sure to set bits that are not available to the initial value.**

22.3.3 Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H)

The priority specification flag registers are used to set the corresponding maskable interrupt priority level.
A priority level is set by using the PR0xy and PR1xy registers in combination (xy = 0L, 0H, 1L, 1H, 2L, or 2H).
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 22 - 4 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (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	STPR01 CSIPR010 IICPR010	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	STPR11 CSIPR110 IICPR110	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	TMPR003	TMPR002	TMPR001	TMPR003H	TMPR001H	IICAPR00	SREPR01	SRPR01 CSIPR011 IICPR011

Figure 22 - 4 Format of Priority Specification Flag Registers

(PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (2/2)

Address: FFFEEH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11L	TMPR103	TMPR102	TMPR101	TMPR103H	TMPR101H	IICAPR10	SREPR11	SRPR11 CSIPR111 IICPR111

Address: FFFEBH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01H	PPR011	PPR010	PPR09	PPR08	PPR07	KRPR0	TMKAPR0	ADPR0

Address: FFFEFH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11H	PPR111	PPR110	PPR19	PPR18	PPR17	KRPR1	TMKAPR1	ADPR1

Address: FFFD8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02L	FLPR0	IICAPR01	TMKBPR00	ITPR001	ITPR000	DOCPR0	CMPPR01	CMPPR00

Address: FFFDCH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12L	FLPR1	IICAPR11	TMKBPR10	ITPR101	ITPR100	DOCPR	CMPPR11	CMPPR10

Address: FFFD9H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	<0>
PR02H	1	1	1	1	1	1	1	FOPR0

Address: FFFDDH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	<0>
PR12H	1	1	1	1	1	1	1	FOPR1

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 22 - 2. Be sure to set bits that are not available to the initial value.

22.3.4 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 22 - 5 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	EGP7	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FFF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

Address: FFF3AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP1	0	0	0	0	EGP11	EGP10	EGP9	EGP8

Address: FFF3BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN1	0	0	0	0	EGN11	EGN10	EGN9	EGN8

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 11)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 22 - 3 shows the Ports Corresponding to EGPn and EGNn Bits.

Table 22 - 3 Ports Corresponding to EGPn and EGNn Bits

Detection Enable Bit		Interrupt Request Signal	24, 25-pin	20-pin	16-pin	10-pin
EGP0	EGN0	INTP0	√	√	√	—
EGP1	EGN1	INTP1	√	√	√	√
EGP2	EGN2	INTP2	√	√	√	—
EGP3	EGN3	INTP3	√	√	√	—
EGP4	EGN4	INTP4	√	√	√	—
EGP5	EGN5	INTP5	√	√	—	—
EGP6	EGN6	INTP6	√	—	—	—
EGP7	EGN7	INTP7	√	—	—	—
EGP8	EGN8	INTP8	√	—	—	—
EGP9	EGN9	INTP9	√	√	√	√
EGP10	EGN10	INTP10	√	√	√	√
EGP11	EGN11	INTP11	√	√	√	—

Caution When the input port pins used for the external interrupt functions are switched to the output mode, the INTPn interrupt might be generated upon detection of a valid edge.

When switching the input port pins to the output mode, set the 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 11

22.3.5 Interrupt flag control register

These registers control the outputs interrupt request flag statuses generated in A/D converter, 12-bit interval timer, 8-bit interval timer, end of timer KB count, data operation circuit (DOC), and comparator to INTFO pin.

(1) Interrupt flag enable register (INTFE)

This register is used to set enable/disable of changing interrupt request flags.
The INTFE register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 22 - 6 Format of Interrupt flag enable register (INTFE)

Address: F0448H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
INTFE	INTCMP1FE	INTCMP0FE	INTDOCFE	INTTMKB0FE	INTIT01FE	INTIT00FE	INTITFE	INTADFE

INTCMP1FE	Selection of whether to enable or disable the changing interrupt request flag of comparator 1
0	Disable the changing interrupt request flag
1	Enable the changing interrupt request flag

INTCMP0FE	Selection of whether to enable or disable the changing interrupt request flag of comparator 0
0	Disable the changing interrupt request flag
1	Enable the changing interrupt request flag

INTDOCFE	Selection of whether to enable or disable the changing interrupt request flag of data operation circuit (DOC)
0	Disable the changing interrupt request flag
1	Enable the changing interrupt request flag

INTTMKB0FE	Selection of whether to enable or disable the changing interrupt request flag of timer KB
0	Disable the changing interrupt request flag
1	Enable the changing interrupt request flag

INTIT01FE	Selection of whether to enable or disable the changing interrupt request flag of 8-bit interval timer 01
0	Disable the changing interrupt request flag
1	Enable the changing interrupt request flag

INTIT00FE	Selection of whether to enable or disable the changing interrupt request flag of 8-bit interval timer 00
0	Disable the changing interrupt request flag
1	Enable the changing interrupt request flag

INTITFE	Selection of whether to enable or disable the changing interrupt request flag of 12-bit interval timer
0	Disable the changing interrupt request flag
1	Enable the changing interrupt request flag

INTADFE	Selection of whether to enable or disable the changing interrupt request flag of A/D converter
0	Disable the changing interrupt request flag
1	Enable the changing interrupt request flag

(2) Interrupt flag output control register 0, 1 (INTFOCTL0, INTFOCTL1)

These registers control output statuses of interrupt request flags.

The INTFOCTL0 and INTFOCTL1 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 22 - 7 Format of Interrupt flag output control register 0, 1 (INTFOCTL0, INTFOCTL1)

Address: F0449H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
INTFOCTL0	0	0	0	0	0	INTFOEN	INTFINV	INTFM[Note 2]

Address: F044AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
INTFOCTL1	0	0	0	0	0	0	INTFRD[Note 1]	IINTFCLR[Note 3]

INTFOEN	Selection of whether to enable or disable the output of statuses of interrupt request flags
0	Disable the output statuses
1	Enable the output statuses

INTFINV	Select the logic of output status of interrupt request flags
0	Outputs with positive logic (non-inverted)
1	Outputs with negative logic (inverted)

INTFM[Note 2]	Select the output mode of status of interrupt request flags
0	Hardware clear mode The output is same as interrupt request signal and is not able to be cleared by software
1	Software clear mode The output is only able to be cleared by software The output is not able to be cleared by hardware

INTFRD[Note 1]	Select the output level of status of interrupt request flags with positive logic
0	High level
1	Low level

INTFCLR[Note 3]	Control to clear the output of status of interrupt request flags
0	Do not clear the output
1	Clear the output when INTFM = 1

Note 1. The read value is fixed to 0 when INTFM = 0.
The read value is the output level of status of interrupt request flags with positive logic0 when INTFM = 1.

Note 2. INTFRD bit is read only.

Note 3. INTFCLR bit is write only, and the read value is fixed to 0.
INTFM = 0: writing 0 or 1 is invalid and it has no effect on output of interrupt request flags.
INTFM = 1: writing 0 does not clear output of interrupt request flags.
writing 1 clears output of interrupt request flags.

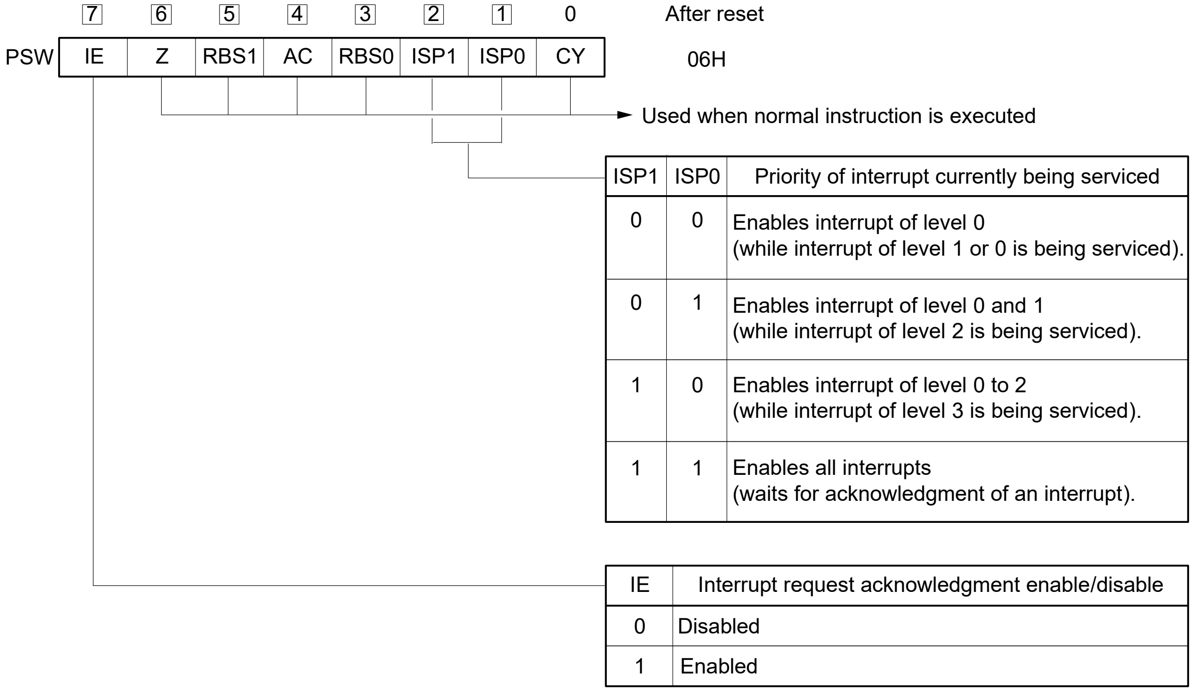
22.3.6 Program status word (PSW)

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 22 - 8 Configuration of Program Status Word



ISP1	ISP0	Priority of interrupt currently being serviced
0	0	Enables interrupt of level 0 (while interrupt of level 1 or 0 is being serviced).
0	1	Enables interrupt of level 0 and 1 (while interrupt of level 2 is being serviced).
1	0	Enables interrupt of level 0 to 2 (while interrupt of level 3 is being serviced).
1	1	Enables all interrupts (waits for acknowledgment of an interrupt).

IE	Interrupt request acknowledgment enable/disable
0	Disabled
1	Enabled

22.4 Interrupt Servicing Operations

22.4.1 Maskable interrupt request acknowledgment

A maskable interrupt request becomes acknowledgeable when the interrupt request flag is set to 1 and the mask (MK) flag corresponding to that interrupt request is cleared to 0. A vectored interrupt request is acknowledged if interrupts are in the interrupt enabled state (when the IE flag is set to 1). However, a low-priority interrupt request is not acknowledged during servicing of a higher priority interrupt request.

The times from generation of a maskable interrupt request until vectored interrupt servicing is performed are listed in Table 22 - 4 below.

For the interrupt request acknowledgment timing, see **Figures 22 - 10** and **22 - 11**.

Table 22 - 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/fCLK (fCLK: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specification flag is acknowledged first. If two or more interrupts requests have the same priority level, the request with the highest default priority is acknowledged first.

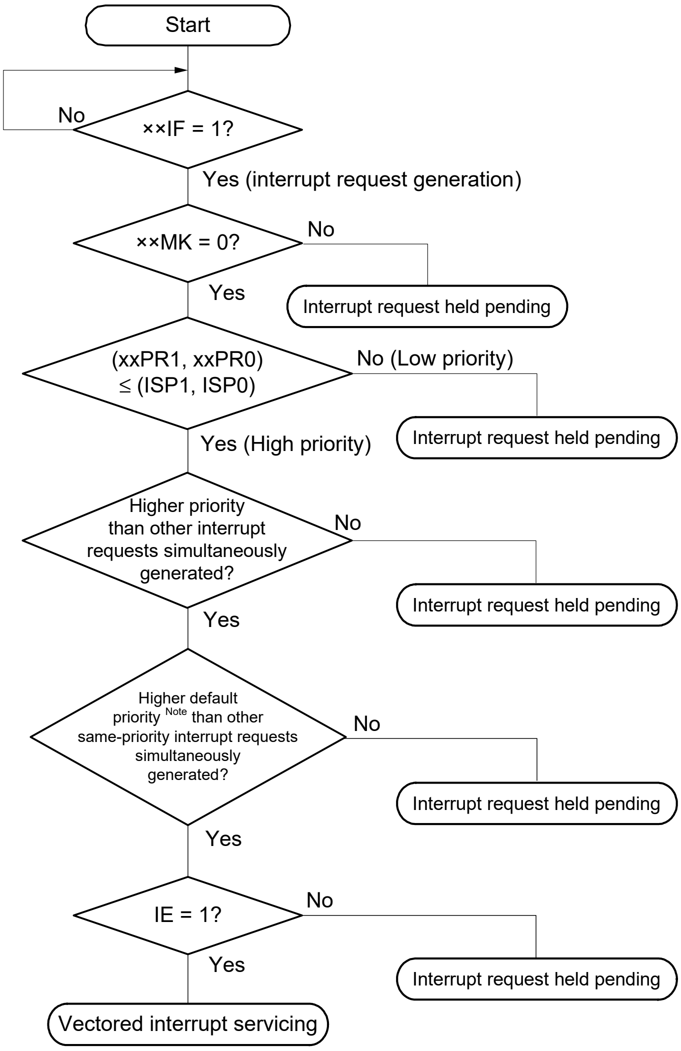
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 22 - 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.

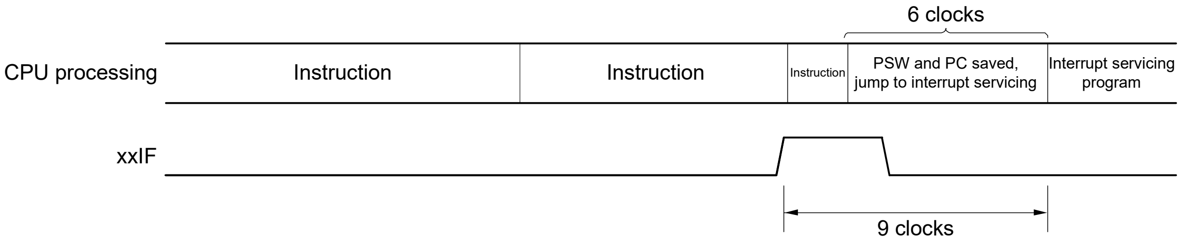
Figure 22 - 9 Interrupt Request Acknowledgment Processing Algorithm



Note For the default priority, refer to **Table 22 - 1 Interrupt Source List**.

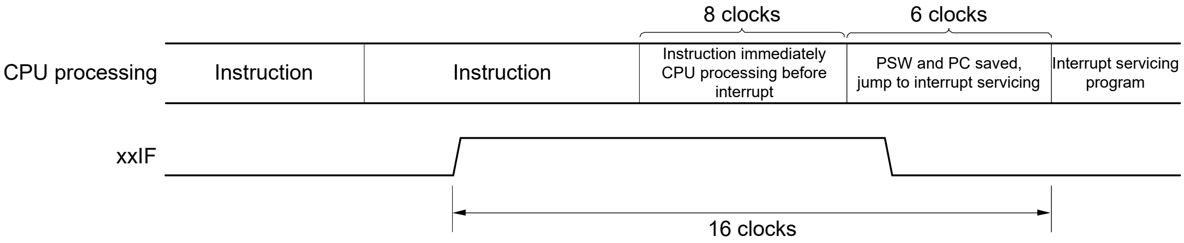
Remark xxIF: Interrupt request flag
xxMK: Interrupt mask flag
xxPR0: Priority specification flag 0
xxPR1: Priority specification flag 1
IE: Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)
ISP0, ISP1: Flag that indicates the priority level of the interrupt currently being serviced (see **Figure 22 - 8**)

Figure 22 - 10 Interrupt Request Acknowledgment Timing (Minimum Time)


6 clocks
CPU processing
Instruction
Instruction
Instruction
PSW and PC saved, jump to interrupt servicing
Interrupt servicing program
xxIF
9 clocks

Remark 1 clock: 1/fCLK (fCLK: CPU clock)

Figure 22 - 11 Interrupt Request Acknowledgment Timing (Maximum Time)


8 clocks
6 clocks
CPU processing
Instruction
Instruction
Instruction immediately CPU processing before interrupt
PSW and PC saved, jump to interrupt servicing
Interrupt servicing program
xxIF
16 clocks

Remark 1 clock: 1/fCLK (fCLK: CPU clock)

22.4.2 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.

22.4.3 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 22 - 5 shows Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing and Figures 22 - 12 show multiple interrupt servicing examples.

Table 22 - 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

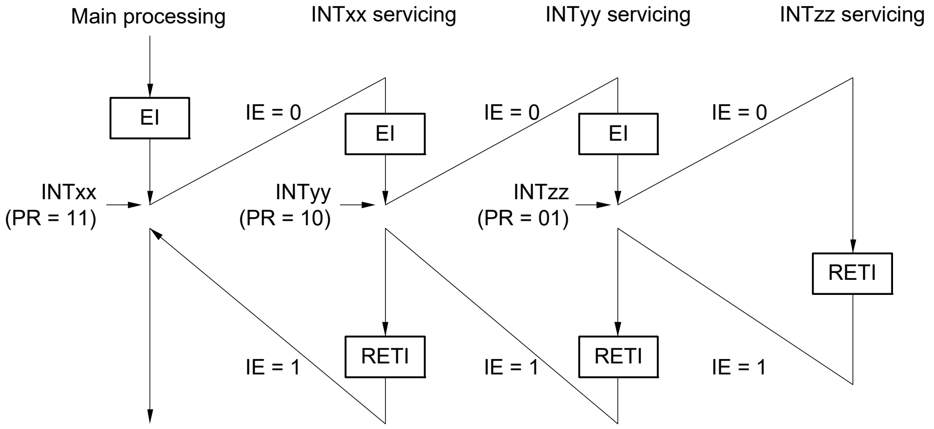
Remark 2. ×: Multiple interrupt servicing disabled

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)

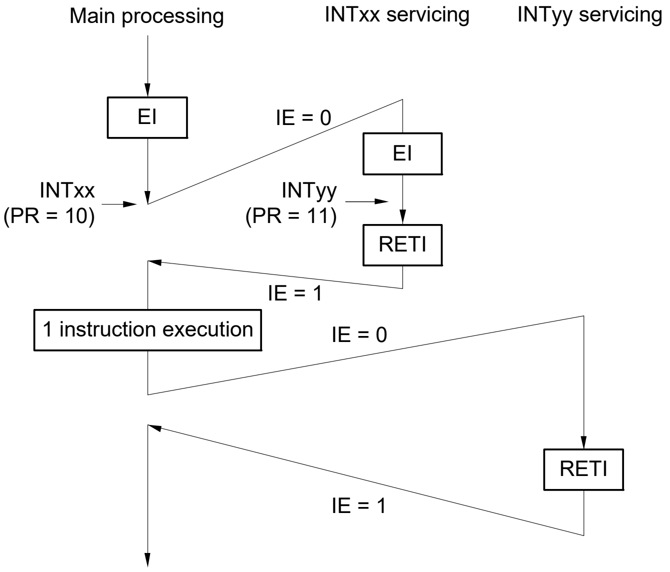
Figure 22 - 12 Examples of Multiple Interrupt Servicing (1/2)

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

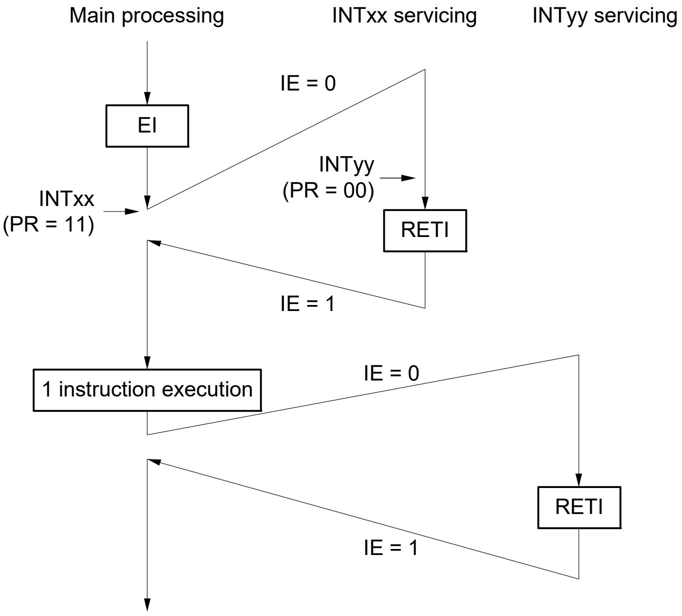


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

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 22 - 12 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.

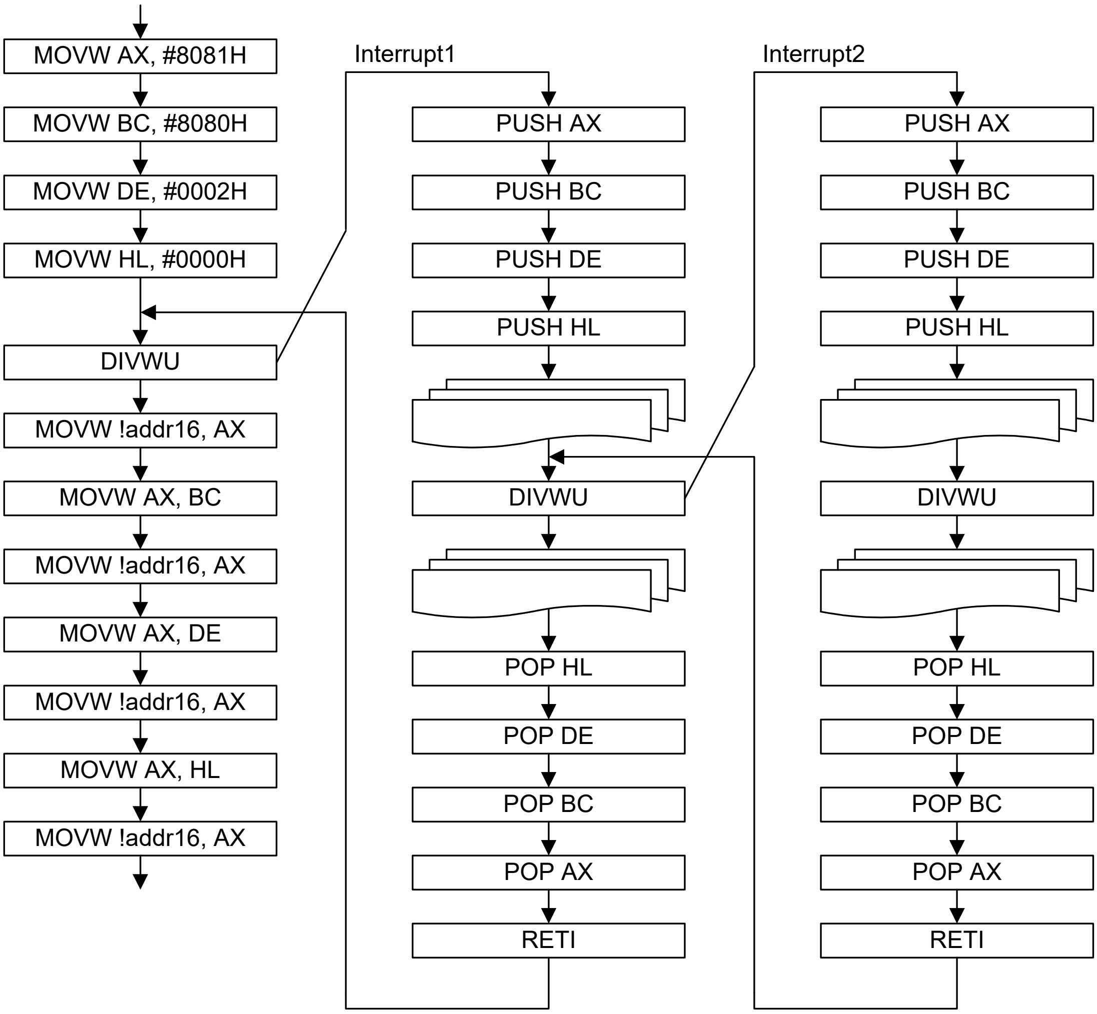
22.4.4 Interrupt servicing during division instruction

The RL78/G11 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.


MOVW AX, #8081H
MOVW BC, #8080H
MOVW DE, #0002H
MOVW HL, #0000H
DIVWU
MOVW !addr16, AX
MOVW AX, BC
MOVW !addr16, AX
MOVW AX, DE
MOVW !addr16, AX
MOVW AX, HL
MOVW !addr16, AX
Interrupt1
PUSH AX
PUSH BC
PUSH DE
PUSH HL
DIVWU
POP HL
POP DE
POP BC
POP AX
RETI
Interrupt2
PUSH AX
PUSH BC
PUSH DE
PUSH HL
DIVWU
POP HL
POP DE
POP BC
POP AX
RETI

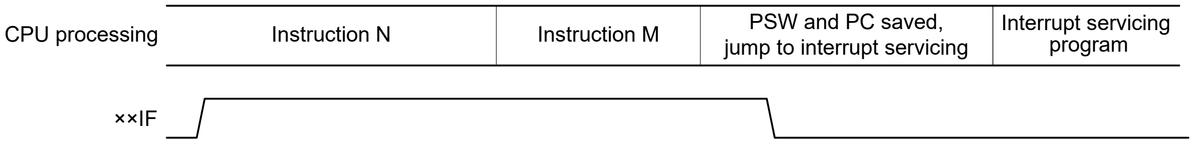
22.4.5 Interrupt request hold

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 22 - 13 shows the timing at which interrupt requests are held pending.

Figure 22 - 13 Interrupt Request Hold



Remark 1. Instruction N: Interrupt request hold instruction
Remark 2. Instruction M: Instruction other than interrupt request hold instruction

CHAPTER 23 KEY INTERRUPT FUNCTION

The number of key interrupt input channels differs, depending on the product.

	16-pin	20-pin	24, 25-pin
Key interrupt input channels	3 ch	5 ch	8 ch

23.1 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 KR7).

Table 23 - 1 Assignment of Key Interrupt Detection Pins

Key interrupt pins	Key return mode register 0 (KRM0)
KR0	KRM00
KR1	KRM01
KR2	KRM02
KR3	KRM03
KR4	KRM04
KR5	KRM05
KR6	KRM06
KR7	KRM07

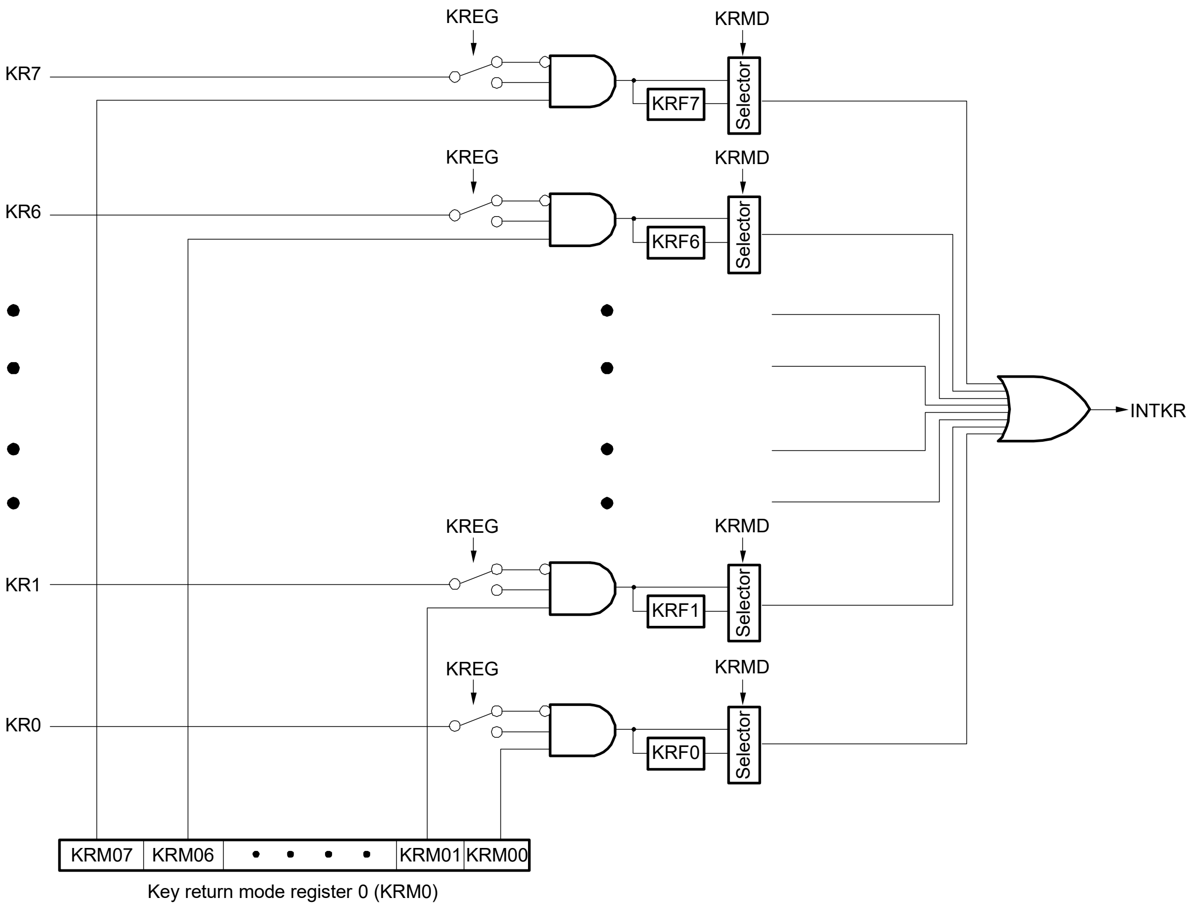
23.2 Configuration of Key Interrupt

The key interrupt includes the following hardware.

Table 23 - 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 23 - 1 Block Diagram of Key Interrupt



23.3 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)

23.3.1 Key return control register (KRCTL)

This register controls the usage of the key return flags (KRF0 to KRF7) and sets the detection edge.

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

Reset signal generation clears this register to 00H.

Figure 23 - 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 KRF7)
0	Does not use key return flags
1	Uses key return flags

KREG	Selection of detection edge (KR0 to KR7)
0	Falling edge
1	Rising edge

23.3.2 Key return mode register 0 (KRM0)

The KRM0 register controls the KR0 to KR7 signals.

The KRM0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 23 - 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	KRM07	KRM06	KRM05	KRM04	KRM03	KRM02	KRM01	KRM00

KRM0n	Key interrupt mode control
0	Does not detect key interrupt signal
1	Detects key interrupt signal

Caution 1. 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 35.4, 36.4 AC Characteristics).

Caution 3. The pins not used in the key interrupt mode can be used as normal ports.

Remark n = 0 to 7

23.3.3 Key return flag register (KRF)

This register controls the key interrupt flags (KRF0 to KRF7).

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

Reset signal generation clears this register to 00H.

Figure 23 - 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	KRF07	KRF06	KRF05	KRF04	KRF3	KRF2	KRF1	KRF0

KRFn	Key interrupt flag (n = 0 to 7)
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.

23.3.4 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 KR7), 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 23 - 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 24 STANDBY FUNCTION

24.1 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 (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 13.3 Registers Controlling Serial Array Unit and 16.3 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 30 OPTION BYTE.

24.2 Registers controlling standby function

The registers which control the standby function are described below.

- Operation speed 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 16 A/D CONVERTER** and **CHAPTER 13 SERIAL ARRAY UNIT**.

24.3 Standby Function Operation

24.3.1 HALT mode

(1) HALT mode

The HALT mode is set by executing the HALT instruction. HALT mode can be set regardless of whether the CPU clock before the setting was the high-speed system clock, high-speed on-chip oscillator clock, 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 24 - 1 Operating Statuses in HALT Mode (1/2)

HALT Mode Setting Item			When HALT Instruction is Executed While CPU is Operating on Main System Clock			
			When CPU is Operating on High-speed On-chip Oscillator Clock (f_{IH})	When CPU is Operating on 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)
	Low-speed on-chip oscillator clock	f_{IL}	Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of operation speed mode control register (OSMC) WUTMMCK0 = 1 or SELLOSC = 1: Oscillates 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			
Timer KB						
12-bit Interval timer						
8-bit Interval timer						
Watchdog timer			See **CHAPTER 12 WATCHDOG TIMER**.			
Clock output/buzzer output			Operable			
10-bit A/D converter						
D/A converter						
Comparator						
Programmable gain amplifier (PGA)						
Serial array unit (SAU)						
Serial interface IICA						
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

Table 24 - 1 Operating Statuses in HALT Mode (2/2)

HALT Mode Setting Item			When HALT Instruction is Executed While CPU is Operating on Subsystem Clock
			When 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}	
	Low-speed on-chip oscillator clock	f_{IL}	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			Operable
Timer KB			
12-bit Interval timer			
8-bit Interval timer			
Watchdog timer			See **CHAPTER 12 WATCHDOG TIMER**.
Clock output/buzzer output			Operable
10-bit A/D converter			Operation disabled
D/A converter			Operable
Comparator			
Programmable gain amplifier (PGA)			
Serial array unit (SAU)			
Serial interface IICA			
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 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

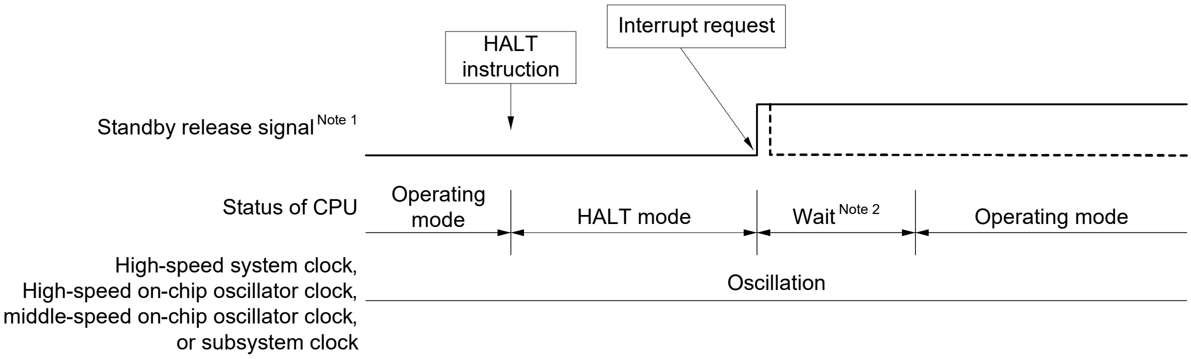
(2) HALT mode release

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 24 - 1 HALT Mode Release by Interrupt Request Generation


Interrupt request
HALT instruction
Standby release signal Note 1
Status of CPU
Operating mode
HALT mode
Wait Note 2
Operating mode
High-speed system clock, High-speed on-chip oscillator clock, middle-speed on-chip oscillator clock, or subsystem clock
Oscillation

Note 1. For details of the standby release signal, see **Figure 22 - 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	10 to 11 clocks

When vectored interrupt servicing is not carried out

Main system clock:	9 to 10 clocks
Subsystem clock:	4 to 5 clocks

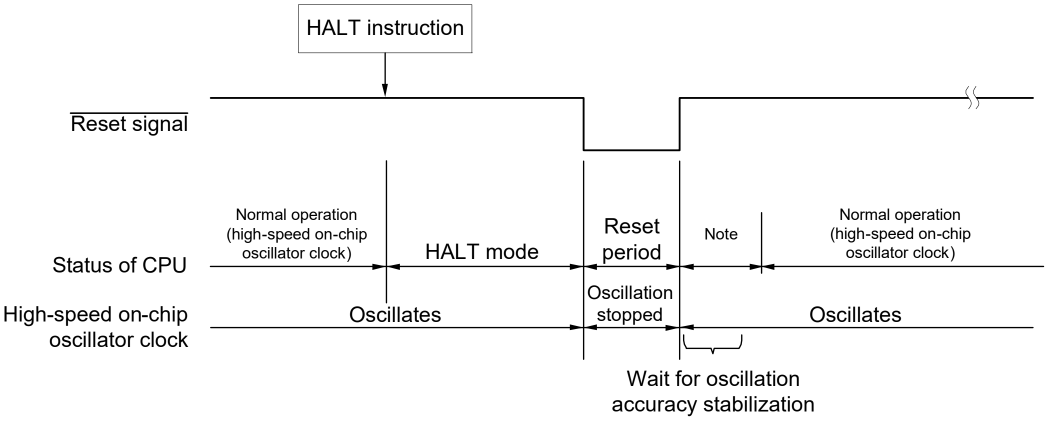
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) Release by reset signal generation

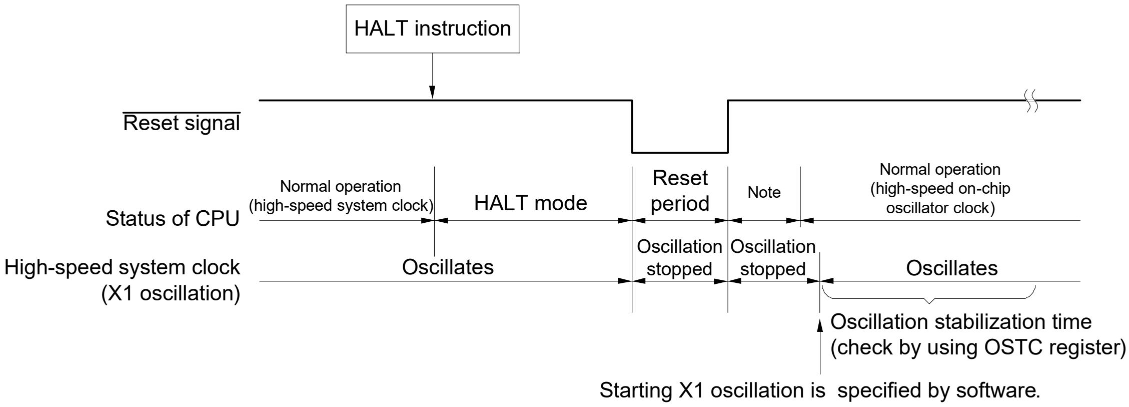
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 24 - 2 HALT Mode Release by Reset (1/2)

(1) When high-speed on-chip oscillator clock is used as CPU clock


HALT instruction
Reset signal
Status of CPU
Normal operation (high-speed on-chip oscillator clock)
HALT mode
Reset period
Note
Normal operation (high-speed on-chip oscillator clock)
High-speed on-chip oscillator clock
Oscillates
Oscillation stopped
Oscillates
Wait for oscillation accuracy stabilization

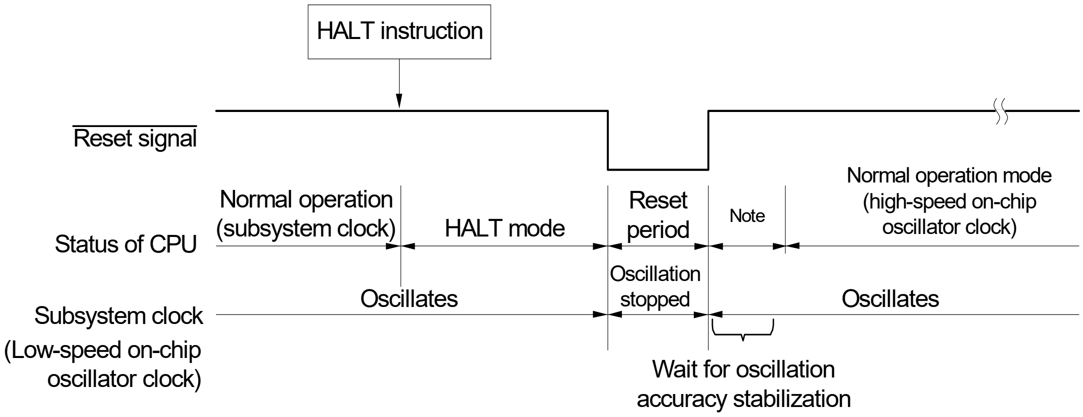
(2) When high-speed system clock is used as CPU clock


HALT instruction
Reset signal
Status of CPU
Normal operation (high-speed system clock)
HALT mode
Reset period
Note
Normal operation (high-speed on-chip oscillator clock)
High-speed system clock (X1 oscillation)
Oscillates
Oscillation stopped
Oscillation stopped
Oscillates
Oscillation stabilization time (check by using OSTC register)
Starting X1 oscillation is specified by software.

Note For the reset processing time, see **CHAPTER 25 RESET FUNCTION**.
For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 26 POWER-ON-RESET CIRCUIT**.

Figure 24 - 2 HALT Mode Release by Reset (2/2)

(3) When subsystem clock is used as CPU clock


HALT instruction
Reset signal
Status of CPU
Normal operation (subsystem clock)
HALT mode
Reset period
Note
Normal operation mode (high-speed on-chip oscillator clock)
Subsystem clock (Low-speed on-chip oscillator clock)
Oscillates
Oscillation stopped
Oscillates
Wait for oscillation accuracy stabilization

Note For the reset processing time, see **CHAPTER 25 RESET FUNCTION**.
For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 26 POWER-ON-RESET CIRCUIT**.

24.3.2 STOP mode

(1) STOP mode setting and operating statuses

The STOP mode is set by executing the STOP instruction, and it can be set only when the CPU clock before the setting was the high-speed on-chip oscillator clock, middle-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 24 - 2 Operating Statuses in STOP Mode

STOP Mode Setting Item			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}				
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of operation speed mode control register (OSMC) WUTMMCK0 = 1 or SELLOSC = 1: Oscillates 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			
Timer KB						
12-bit Interval timer			Operation is disabled when a clock f_{CLK} is selected as the clock source for counting-Operable when the f_{IL} is selected.			
8-bit Interval timer			Operable			
Watchdog timer			See **CHAPTER 12 WATCHDOG TIMER**.			
Clock output/buzzer output			Operates when the subsystem clock is selected as the clock source for counting (operation is disabled when a clock other than the subsystem clock is selected).			
10-bit A/D converter			Wakeup operation is enabled (switching to SNOOZE mode)			
D/A converter			Operable (status before STOP mode was set is retained)			
Comparator			Operable (only when the digital filter is not used)			
Programmable gain amplifier (PGA)			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			
Serial interface IICA			Wakeup operation is enabled by address match			
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

(2) STOP mode release

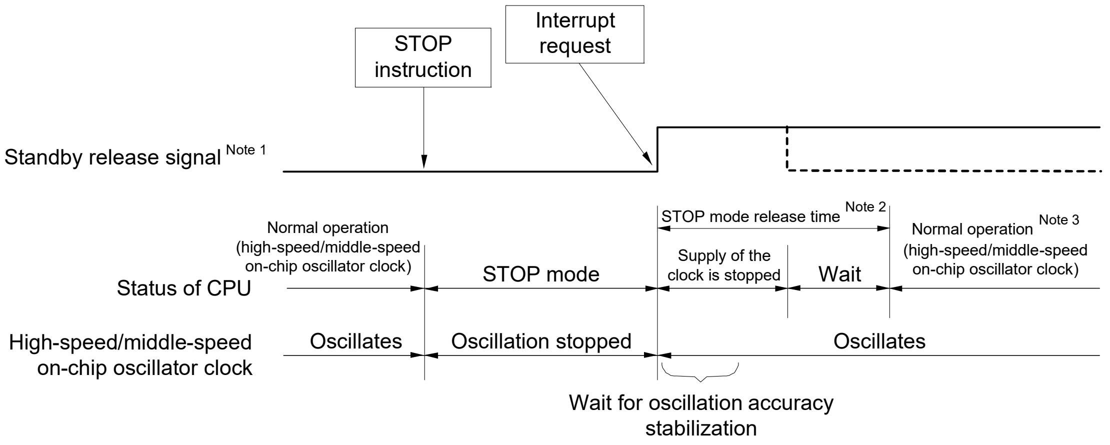
The STOP mode can be released by the following two sources.

(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 24 - 3 STOP Mode Release by Interrupt Request Generation (1/2)

(1) When high-speed/middle-speed on-chip oscillator clock is used as CPU clock


STOP instruction
Interrupt request
Standby release signal Note 1
Normal operation (high-speed/middle-speed on-chip oscillator clock)
STOP mode
STOP mode release time Note 2
Supply of the clock is stopped
Wait
Normal operation Note 3 (high-speed/middle-speed on-chip oscillator clock)
Status of CPU
High-speed/middle-speed on-chip oscillator clock
Oscillates
Oscillation stopped
Oscillates
Wait for oscillation accuracy stabilization

Note 1. For details of the standby release signal, see **Figure 22 - 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

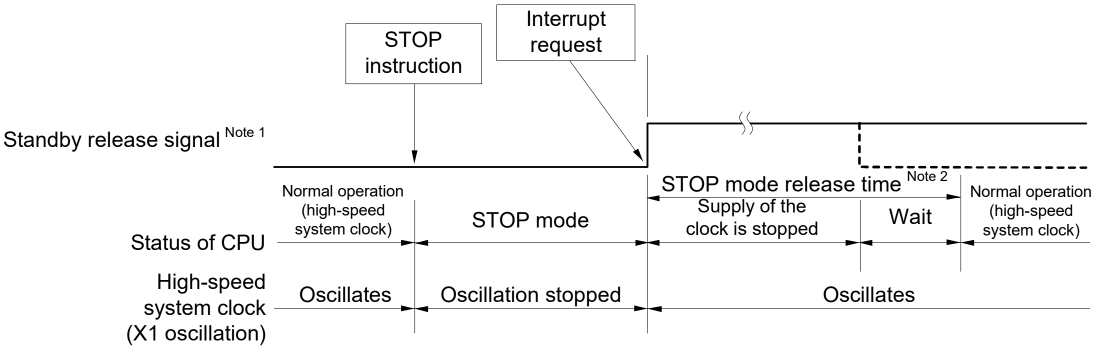
Caution To shorten oscillation stabilization time after the STOP mode is released when the CPU operates with the high-speed system clock (X1 oscillation), temporarily switch the CPU clock to the high-speed on-chip oscillator clock before the execution of the STOP instruction.

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 24 - 3 STOP Mode Release by Interrupt Request Generation (2/2)

(2) When high-speed system clock (X1 oscillation) is used as CPU clock


STOP instruction
Interrupt request
Standby release signal Note 1
STOP mode release time Note 2
Normal operation (high-speed system clock)
STOP mode
Supply of the clock is stopped
Wait
Normal operation (high-speed system clock)
Status of CPU
High-speed system clock (X1 oscillation)
Oscillates
Oscillation stopped
Oscillates

Note 1. For details of the standby release signal, see **Figure 22 - 1 Basic Configuration of Interrupt Function (1/2)**.

Note 2. STOP mode release time

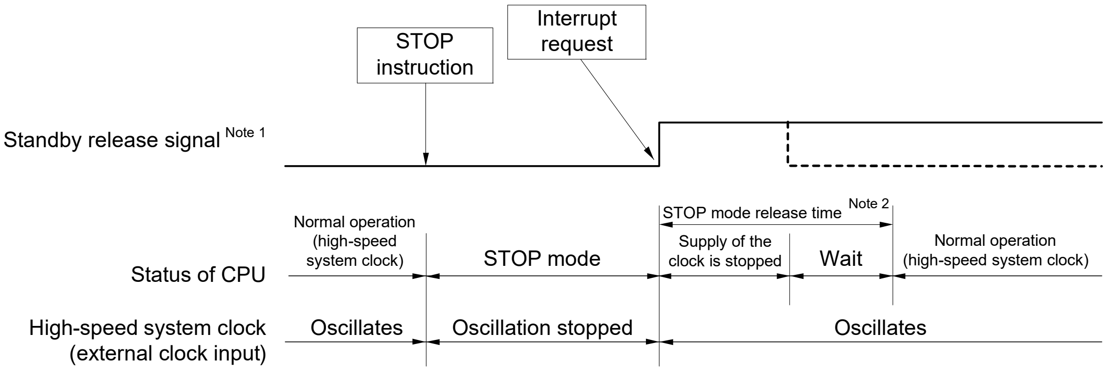
Supply of the clock is stopped:

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


STOP instruction
Interrupt request
Standby release signal Note 1
STOP mode release time Note 2
Normal operation (high-speed system clock)
STOP mode
Supply of the clock is stopped
Wait
Normal operation (high-speed system clock)
Status of CPU
High-speed system clock (external clock input)
Oscillates
Oscillation stopped
Oscillates

Note 1. For details of the standby release signal, see **Figure 22 - 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.

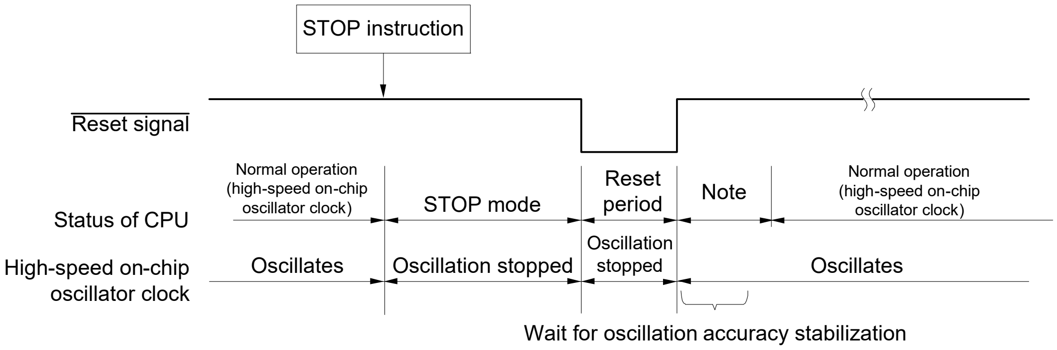
Remark 2. The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(b) Release by reset signal generation

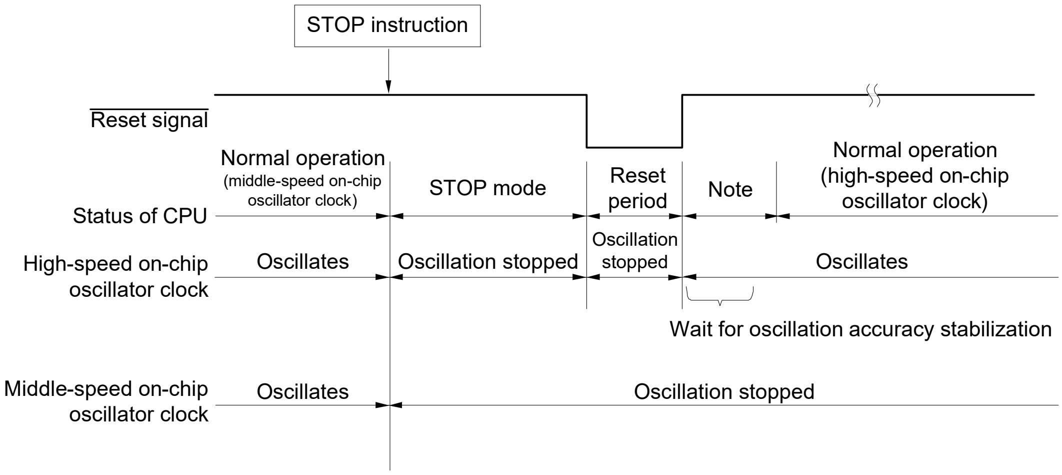
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 24 - 4 STOP Mode Release by Reset

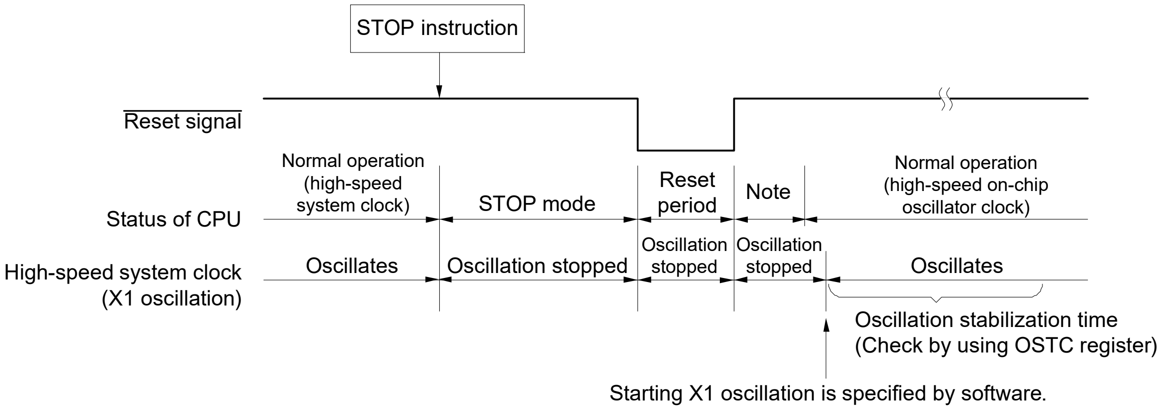
(1) When high-speed on-chip oscillator clock is used as CPU clock


STOP instruction
Reset signal
Status of CPU
Normal operation (high-speed on-chip oscillator clock)
STOP mode
Reset period
Note
Normal operation (high-speed on-chip oscillator clock)
High-speed on-chip oscillator clock
Oscillates
Oscillation stopped
Oscillation stopped
Oscillates
Wait for oscillation accuracy stabilization

(2) When middle-speed on-chip oscillator clock is used as CPU clock


STOP instruction
Reset signal
Status of CPU
Normal operation (middle-speed on-chip oscillator clock)
STOP mode
Reset period
Note
Normal operation (high-speed on-chip oscillator clock)
High-speed on-chip oscillator clock
Oscillates
Oscillation stopped
Oscillation stopped
Oscillates
Wait for oscillation accuracy stabilization
Middle-speed on-chip oscillator clock
Oscillates
Oscillation stopped

(3) When high-speed system clock is used as CPU clock


STOP instruction
Reset signal
Status of CPU
Normal operation (high-speed system clock)
STOP mode
Reset period
Note
Normal operation (high-speed on-chip oscillator clock)
High-speed system clock (X1 oscillation)
Oscillates
Oscillation stopped
Oscillation stopped
Oscillation stopped
Oscillates
Oscillation stabilization time (Check by using OSTC register)
Starting X1 oscillation is specified by software.

Note For the reset processing time, see **CHAPTER 25 RESET FUNCTION**.

For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 26 POWER-ON-RESET CIRCUIT**.

24.3.3 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 **13.3 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 **16.3 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 **20.3 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)
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)

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 24 - 3 Operating Statuses in SNOOZE Mode

STOP Mode Setting Item			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}		

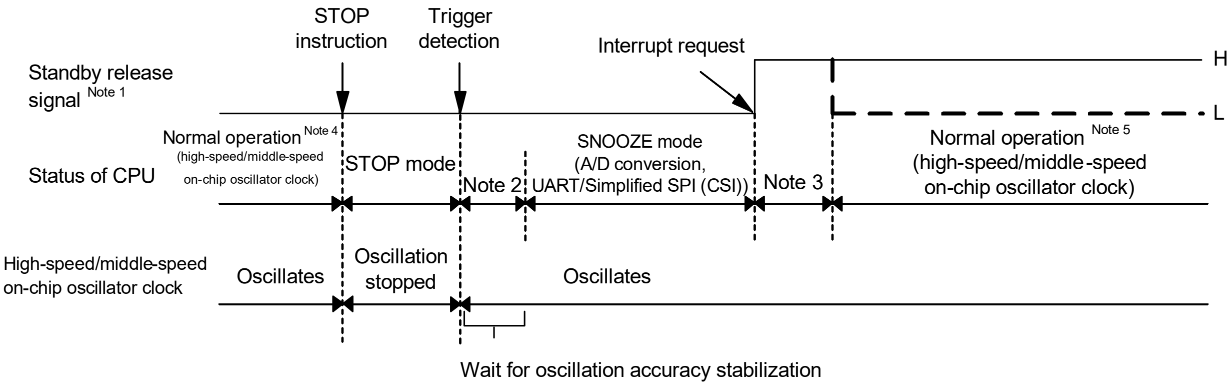
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of operation speed mode control register (OSMC)
			WUTMMCK0 = 1 or SELLOSC = 1: Oscillates
			WUTMMCK0 = 0, SELLOSC = 0, and WDTON = 0: Stop
			WUTMMCK0 = 0, SELLOSC = 0, WDTON = 1, and WDSTBYON = 1: Oscillates
			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	
Timer KB				
12-bit Interval timer			Operation is disabled when a clock f_{CLK} is selected as the clock source for counting-Operable when the f_{IL} is selected.	
8-bit Interval timer			Operable	
Watchdog timer			See **CHAPTER 12 WATCHDOG TIMER**.	
Clock output/buzzer output			Operates when the f_{IL} clock is selected for counting. (operation is disabled when another clock is selected).	
10-bit A/D converter			Operable	
D/A converter			Operation(status before STOP mode was set is retained)	
Comparator			Operable (when digital filter is not used)	
Programmable gain amplifier (PGA)			Operable	
Serial array unit (SAU)			Operable only CSIp and UARTq only. Operation disabled other than CSIp and UARTq.	
Serial interface IICA			Operation disabled	
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

(2) Timing diagram when the interrupt request signal is generated in the SNOOZE mode

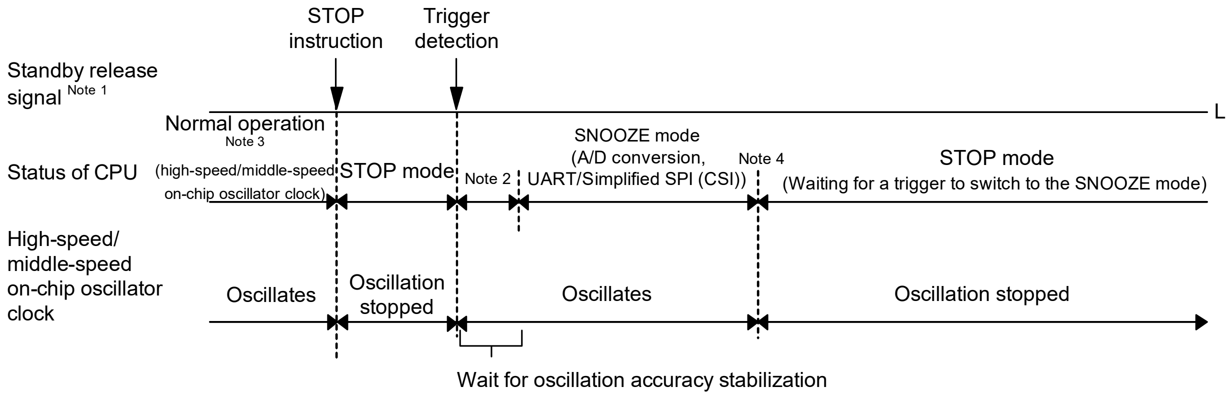
Figure 24 - 5 When the Interrupt Request Signal is Generated in the SNOOZE Mode


STOP instruction
Trigger detection
Interrupt request
Standby release signal Note 1
H
L
Normal operation Note 4 (high-speed/middle-speed on-chip oscillator clock)
STOP mode
Note 2
SNOOZE mode (A/D conversion, UART/Simplified SPI (CSI))
Note 3
Normal operation Note 5 (high-speed/middle-speed on-chip oscillator clock)
Status of CPU
High-speed/middle-speed on-chip oscillator clock
Oscillates
Oscillation stopped
Oscillates
Wait for oscillation accuracy stabilization

Note 1. For details of the standby release signal, see **Figure 22 - 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 SWCm = 1) immediately before switching to the STOP mode.
Note 5. Be sure to release the SNOOZE mode (AWC = 0 or SWCm = 0) immediately after return to the normal operation.

(3) Timing diagram when the interrupt request signal is not generated in the SNOOZE mode

Figure 24 - 6 When the Interrupt Request Signal is not Generated in the SNOOZE Mode


STOP instruction
Trigger detection
Standby release signal Note 1
L
Normal operation Note 3 (high-speed/middle-speed on-chip oscillator clock)
STOP mode
Note 2
SNOOZE mode (A/D conversion, UART/Simplified SPI (CSI))
Note 4
STOP mode (Waiting for a trigger to switch to the SNOOZE mode)
Status of CPU
High-speed/ middle-speed on-chip oscillator clock
Oscillates
Oscillation stopped
Oscillates
Oscillation stopped
Wait for oscillation accuracy stabilization

Note 1. For details of the standby release signal, see **Figure 22 - 1**.
Note 2. Transition time from STOP mode to SNOOZE mode
Note 3. Enable the SNOOZE mode (AWC = 1 or SWCm = 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 16 A/D CONVERTER** and **CHAPTER 13 SERIAL ARRAY UNIT**.

CHAPTER 25 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 00000H and 00001H 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 25 - 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.

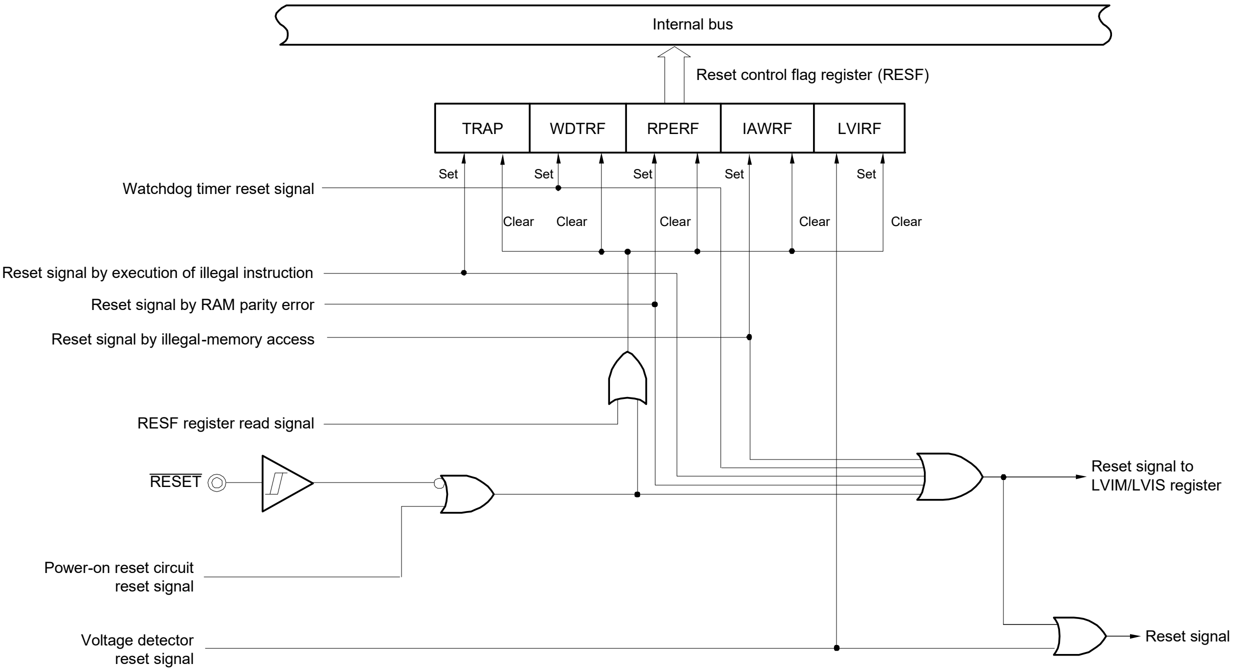
Caution 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.
To perform an external reset upon power application, input a low level to the $\overline{\text{RESET}}$ pin, turn power on, continue to input a low level to the pin for 10 μs or more within the operating voltage range shown in 35.4, 36.4 AC Characteristics and then input a high level to the pin.

Caution 2. During generation of a reset signal, the X1 clock, high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock, and low-speed on-chip oscillator clock stop oscillating. External main system clock input 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 25 - 1 Block Diagram of Reset Function


Internal bus
Reset control flag register (RESF)
TRAP
WDTRF
RPERF
IAWRF
LVIRF
Set
Set
Set
Set
Set
Clear
Clear
Clear
Clear
Clear
Watchdog timer reset signal
Reset signal by execution of illegal instruction
Reset signal by RAM parity error
Reset signal by illegal-memory access
RESF register read signal
RESET
Power-on reset circuit reset signal
Voltage detector reset signal
Reset signal to LVIM/LVIS register
Reset signal

Caution An LVD circuit internal reset does not reset the LVD circuit.

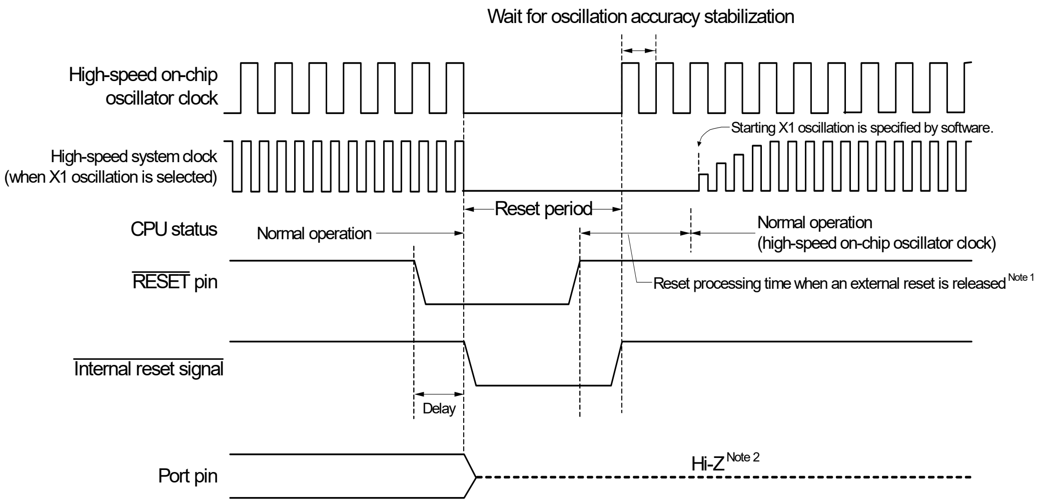
Remark 1. LVIM: Voltage detection register

Remark 2. LVIS: Voltage detection level register

25.1 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.

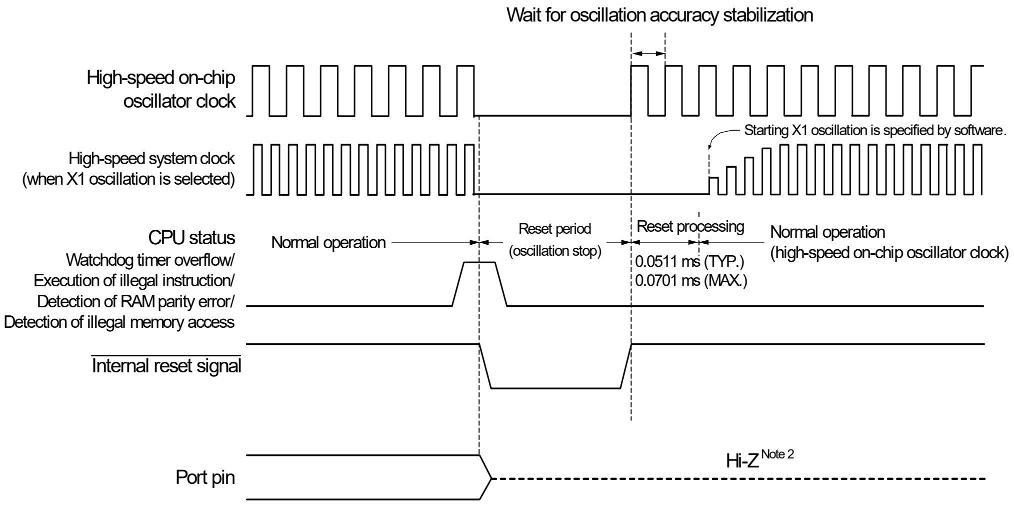
Figure 25 - 2 Timing of Reset by $\overline{\text{RESET}}$ Input


Wait for oscillation accuracy stabilization
High-speed on-chip oscillator clock
Starting X1 oscillation is specified by software.
High-speed system clock (when X1 oscillation is selected)
Reset period
CPU status
Normal operation
Normal operation (high-speed on-chip oscillator clock)
RESET pin
Reset processing time when an external reset is released Note 1
Internal reset signal
Delay
Port pin
Hi-Z Note 2

(**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 25 - 3 Timing of Reset Due to Watchdog Timer Overflow, Execution of Illegal Instruction, Detection of RAM Parity Error, or Detection of Illegal Memory Access


Wait for oscillation accuracy stabilization
High-speed on-chip oscillator clock
Starting X1 oscillation is specified by software.
High-speed system clock (when X1 oscillation is selected)
CPU status
Normal operation
Reset period (oscillation stop)
Reset processing
0.0511 ms (TYP.)
0.0701 ms (MAX.)
Normal operation (high-speed on-chip oscillator clock)
Watchdog timer overflow/ Execution of illegal instruction/ Detection of RAM parity error/ Detection of illegal memory access
Internal reset signal
Port pin
Hi-Z Note 2

(**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).

Caution The watchdog timer is also reset without exception when an internal reset occurs.

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 26 POWER-ON-RESET CIRCUIT** or **CHAPTER 27 VOLTAGE DETECTOR**.

Remark V_{POR}: POR power supply rise detection voltage
V_{LVD}: LVD detection voltage

Table 25 - 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)
	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
Timer KB			
12-bit Interval timer			
8-bit Interval timer			
Watchdog timer			
Clock output/buzzer output			
10-bit A/D converter			
D/A converter			
Comparator			
Programmable gain amplifier (PGA)			
Serial array unit (SAU)			
Serial interface IICA			
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, P125 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).
- Low level during the external reset period. High level during other types of reset (connected to the on-chip pull-up resistor).

Remark f_{IH}: High-speed on-chip oscillator clock f_X: X1 oscillation clock
f_{IM}: Middle-speed on-chip oscillator clock f_{EX}: External main system clock
f_{IL}: Low-speed on-chip oscillator clock

Table 25 - 2 Hardware Statuses After Reset Acknowledgment

Hardware		After Reset Acknowledgment [Note]
Program counter (PC)		The contents of the reset vector table (00000H, 00001H) 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**.

25.2 Register for Confirming Reset Source

25.2.1 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 25 - 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 25 - 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 28.3.3 RAM parity error detection function.

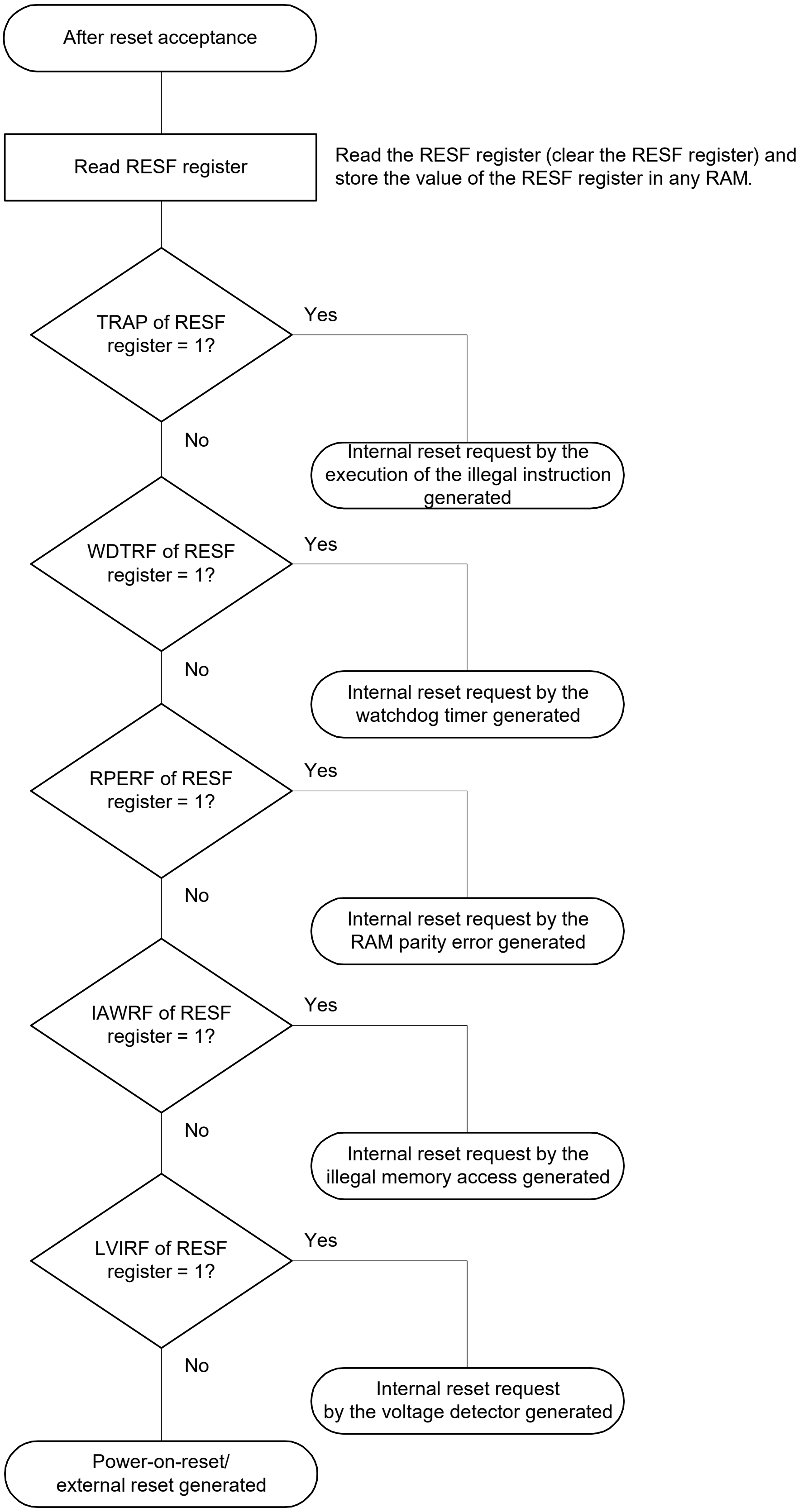
The status of the RESF register when a reset request is generated is shown in Table 25 - 3.

Table 25 - 3 RESF Register Status When Reset Request Is Generated

Reset Source Flag	$\overline{\text{RESET}}$ Input	Reset by POR	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by RAM parity error	Reset by illegal-memory access	Reset by 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 25 - 5 shows the procedure for checking a reset source.

Figure 25 - 5 Procedure for Checking Reset Source


After reset acceptance
Read RESF register
TRAP of RESF register = 1?
Yes
No
Internal reset request by the execution of the illegal instruction generated
WDTRF of RESF register = 1?
Yes
No
Internal reset request by the watchdog timer generated
RPERF of RESF register = 1?
Yes
No
Internal reset request by the RAM parity error generated
IAWRF of RESF register = 1?
Yes
No
Internal reset request by the illegal memory access generated
LVIRF of RESF register = 1?
Yes
No
Internal reset request by the voltage detector generated
Power-on-reset/ external reset generated

Read the RESF register (clear the RESF register) and store the value of the RESF register in any RAM.

25.2.2 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.

The PRR0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Figure 25 - 6 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	IICA1RES	ADCRES	IICA0RES	0	SAU0RES	0	TAU0RES

PRR0n	Peripheral reset control on each peripheral hardware
0	Peripheral module is released from the reset state.
1	Peripheral module is in the reset state.

Remark n = 0, 2, 4 to 6

The controlled hardware by each bit are as follows.

Table 25 - 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)
4	IICA0RES	Serial interface IICA (Unit 0)
5	ADCRES	10-bit A/D Converter
6	IICA1RES	Serial interface IICA (Unit 1)

25.2.3 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.

The PRR1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Figure 25 - 7 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	DACRES	0	CMPRES	0	0	PGA0RES	0	0

PRR1n	Peripheral reset control on each peripheral hardware
0	Peripheral module is released from the reset state.
1	Peripheral module is in the reset state.

Remark n = 2, 5, 7

The controlled hardware by each bit are as follows.

Table 25 - 5 Macro Controlled by Each Bit in PRR1

Bit	Bit Name	Controlled Hardware
2	PGA0RES	Programmable gain amplifier (PGA)
5	CMPRES	Comparators 0 and 1
7	DACRES	D/A Converter

25.2.4 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.

The PRR2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Figure 25 - 8 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	TKB0RES

PRR2n	Peripheral reset control on each peripheral hardware macro
0	Peripheral module is released from the reset state.
1	Peripheral module is in the reset state.

Remark n = 0, 5, 7

The controlled hardware by each bit are as follows.

Table 25 - 6 Macro Controlled by Each Bit in PRR2

Bit	Bit Name	Controlled Hardware
0	TKB0RES	Timer KB
5	DOCRES	Data operation circuit (DOC)
7	TMKARES	12-bit interval timer

CHAPTER 26 POWER-ON-RESET CIRCUIT

26.1 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 **35.4 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 **35.4 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) are cleared (00H).

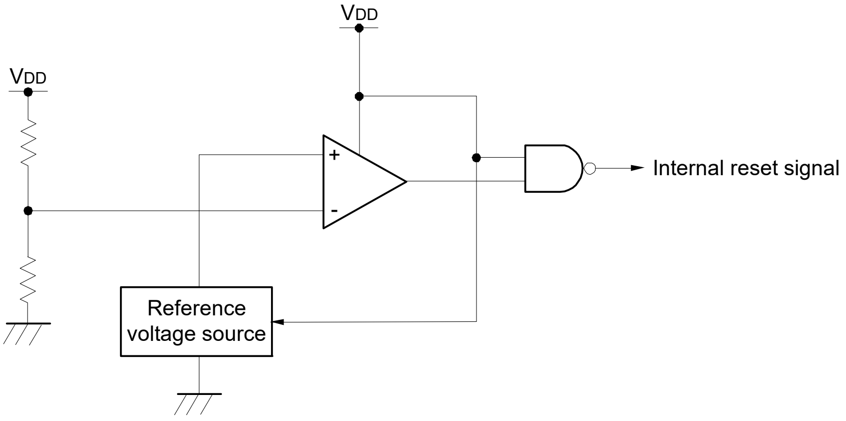
Remark 1. The RL78 microcontroller incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset source is located in the reset control flag register (RESF) for when an internal reset signal is generated by the watchdog timer (WDT), voltage-detector (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 25 RESET FUNCTION**.

Remark 2. V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage
For details, see **35.6.6 POR circuit characteristics**.

26.2 Configuration of Power-on-reset Circuit

The block diagram of the power-on-reset circuit is shown in Figure 26 - 1.

Figure 26 - 1 Block Diagram of Power-on-reset Circuit

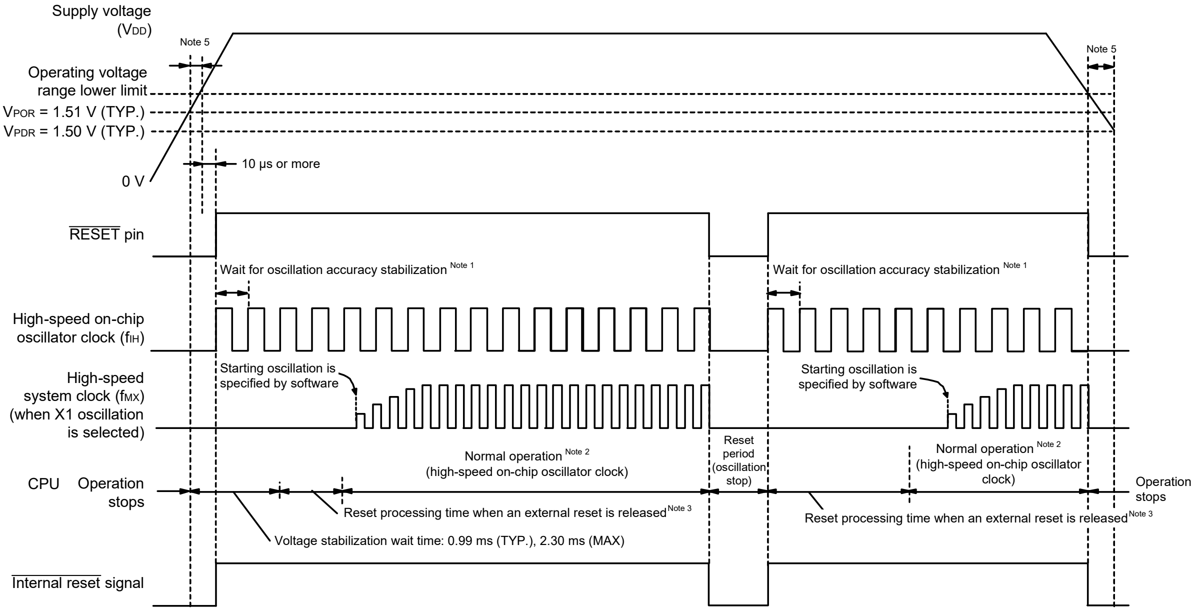


26.3 Operation of Power-on-reset Circuit

The timing of generation of the internal reset signal by the power-on-reset circuit and voltage detector is shown next.

Figure 26 - 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


Supply voltage (VDD)
Note 5
Operating voltage range lower limit
VPOR = 1.51 V (TYP.)
VPDR = 1.50 V (TYP.)
0 V
10 μs or more
Note 5
RESET pin
Wait for oscillation accuracy stabilization Note 1
Wait for oscillation accuracy stabilization Note 1
High-speed on-chip oscillator clock (fIH)
High-speed system clock (fMX) (when X1 oscillation is selected)
Starting oscillation is specified by software
Starting oscillation is specified by software
CPU
Operation stops
Normal operation Note 2 (high-speed on-chip oscillator clock)
Reset period (oscillation stop)
Normal operation Note 2 (high-speed on-chip oscillator clock)
Operation stops
Reset processing time when an external reset is released Note 3
Voltage stabilization wait time: 0.99 ms (TYP.), 2.30 ms (MAX)
Reset processing time when an external reset is released Note 3
Internal reset signal

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.

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 **35.4 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.

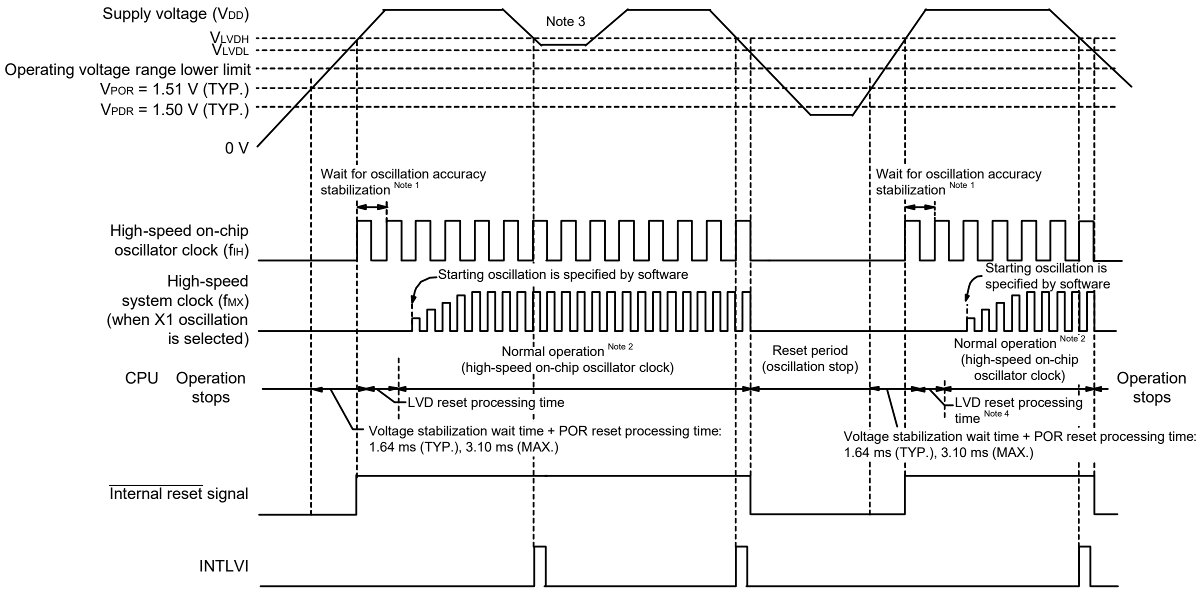
Remark VPOR: POR power supply rise detection voltage

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 27 VOLTAGE DETECTOR.**

Figure 26 - 2 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)


Supply voltage (VDD)
Note 3
VLVDH
VLVDL
Operating voltage range lower limit
VPOR = 1.51 V (TYP.)
VPDR = 1.50 V (TYP.)
0 V
Wait for oscillation accuracy stabilization Note 1
Wait for oscillation accuracy stabilization Note 1
High-speed on-chip oscillator clock (fIH)
Starting oscillation is specified by software
Starting oscillation is specified by software
High-speed system clock (fMX) (when X1 oscillation is selected)
Normal operation Note 2 (high-speed on-chip oscillator clock)
Reset period (oscillation stop)
Normal operation Note 2 (high-speed on-chip oscillator clock)
CPU Operation stops
Operation stops
LVD reset processing time
LVD reset processing time Note 4
Voltage stabilization wait time + POR reset processing time: 1.64 ms (TYP.), 3.10 ms (MAX.)
Voltage stabilization wait time + POR reset processing time: 1.64 ms (TYP.), 3.10 ms (MAX.)
Internal reset signal
INTLVI

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.

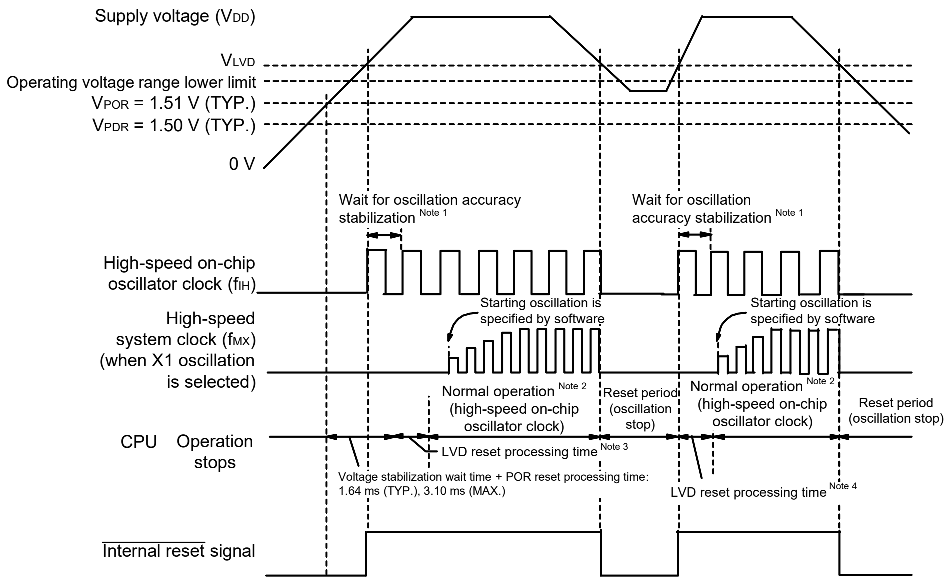
Note 3. After the interrupt request signal (INTLVI) is generated, the LVIL and LVIMD bits of the voltage detection level register (LVIS) are automatically set to 1. After INTLVI is generated, appropriate settings should be made according to **Figure 27 - 7 Setting Procedure for Operating Voltage Check 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 26 - 2 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)


Supply voltage (VDD)
VLVD
Operating voltage range lower limit
VPOR = 1.51 V (TYP.)
VPDR = 1.50 V (TYP.)
0 V
Wait for oscillation accuracy stabilization Note 1
Wait for oscillation accuracy stabilization Note 1
High-speed on-chip oscillator clock (fIH)
Starting oscillation is specified by software
Starting oscillation is specified by software
High-speed system clock (fMX) (when X1 oscillation is selected)
Normal operation Note 2 (high-speed on-chip oscillator clock)
Reset period (oscillation stop)
Normal operation Note 2 (high-speed on-chip oscillator clock)
Reset period (oscillation stop)
CPU Operation stops
LVD reset processing time Note 3
Voltage stabilization wait time + POR reset processing time: 1.64 ms (TYP.), 3.10 ms (MAX.)
LVD reset processing time Note 4
Internal reset signal

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.

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 26 - 2 (3).

CHAPTER 27 VOLTAGE DETECTOR

27.1 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 **27.3.2 Voltage detection level register (LVIS)** and **CHAPTER 30 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 **35.4, 36.4 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.

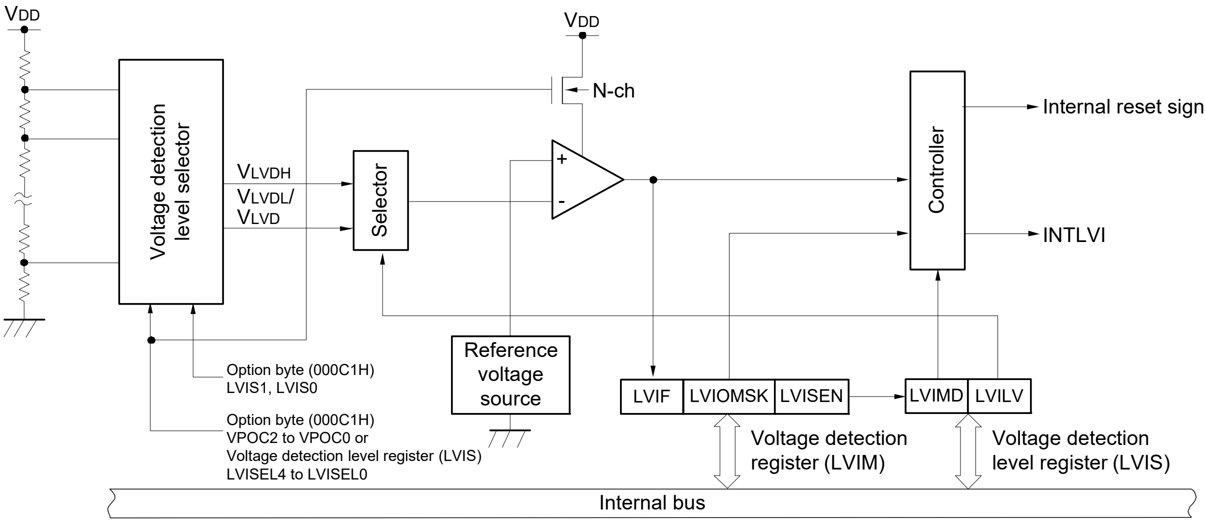
Interrupt & reset mode (LVIMDS1, LVIMDS0 = 1, 0)	Reset mode (LVIMDS1, LVIMDS0 = 1, 1)	Interrupt mode (LVIMDS1, LVIMDS0 = 0, 1)
Generates an interrupt request signal by detecting $V_{DD} < V_{LVDH}$ when the operating voltage falls, and an internal reset by detecting $V_{DD} < V_{LVDL}$. Releases an internal 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 $V_{DD} < V_{LVD}$.	Releases an internal reset by detecting $V_{DD} \geq V_{LVD}$ at power on after the first release of the POR. Generates an interrupt request signal by detecting $V_{DD} < V_{LVD}$ or $V_{DD} \geq V_{LVD}$ at power on after the second release of the POR.

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 25 RESET FUNCTION**.

27.2 Configuration of Voltage Detector

The block diagram of the voltage detector is shown in Figure 27 - 1.

Figure 27 - 1 Block Diagram of Voltage Detector



27.3 Registers Controlling Voltage Detector

The voltage detector is controlled by the following registers.

- Voltage detection register (LVIM)
- Voltage detection level register (LVIS)

27.3.1 Voltage detection register (LVIM)

This register is used to specify whether to enable or disable rewriting the voltage detection level register (LVIS), 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 27 - 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

27.3.2 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.

(Note1 is listed on the next page.)

Figure 27 - 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 27 - 7.

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. Table 27 - 1 shows the format of the user option byte (000C1H/010C1H). For details about the option byte, see CHAPTER 30 OPTION BYTE.

Table 27 - 1 Format of User Option Byte (000C1H/010C1H) (1/2)

Address: 000C1H/010C1H[Note]

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	PORTSELB	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		
3.75 V	3.67 V					0	0		
2.92 V	2.86 V	2.75 V		1	1	1	0		
3.02 V	2.96 V					0	1		
4.06 V	3.98 V					0	0		
—			Settings other than the above are prohibited						

- LVD setting (reset mode)

Detection voltage		Option byte Setting Value						
VLVD		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		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	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 27 VOLTAGE DETECTOR**.

Remark 2. The detection voltage is a TYP. value. For details, see **35.6.7, 36.6.7 LVD circuit characteristics**.

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

Table 27 - 1 Format of User Option Byte (000C1H/010C1H) (2/2)

Address: 000C1H/010C1H$^{\text{Note}}$

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	PORTSELB	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		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	1	0	0		
—		Settings other than the above are prohibited						

- 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						

- Setting of P125/$\overline{\text{RESET}}$/INTP9 pin

PORTSELB	P125/$\overline{\text{RESET}}$ pin control
0	Port functions (P125/INTP9)
1	$\overline{\text{RESET}}$ input (the on-chip pull-up resistor is valid)

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Caution After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 35.4, 36.4 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 27 VOLTAGE DETECTOR**.

Remark 3. The detection voltage is a TYP. value. For details, see **35.6.7, 36.6.7 LVD circuit characteristics**.

27.4 Operation of Voltage Detector

27.4.1 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 **27.3.2 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}).

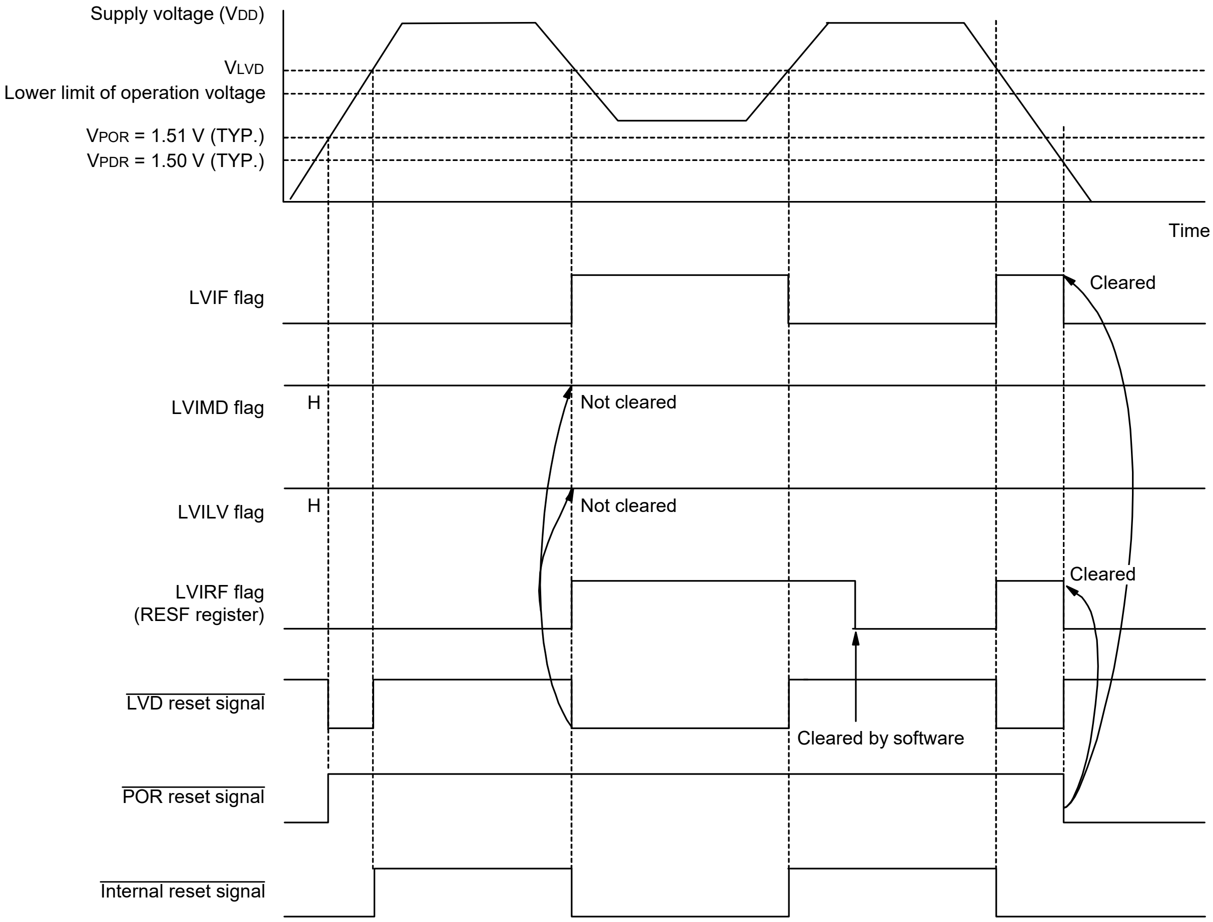
At the fall of the operating voltage, an internal reset by LVD is generated when the supply voltage (V_{DD}) falls below the voltage detection level (V_{LVD}).

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 27 - 4 shows the timing of the internal reset signal generated in the LVD reset mode.

Figure 27 - 4 Timing of Voltage Detector Internal Reset Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 1)



Remark VPOR: POR power supply rise detection voltage
VPDR: POR power supply fall detection voltage

27.4.2 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 **27.3.2 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

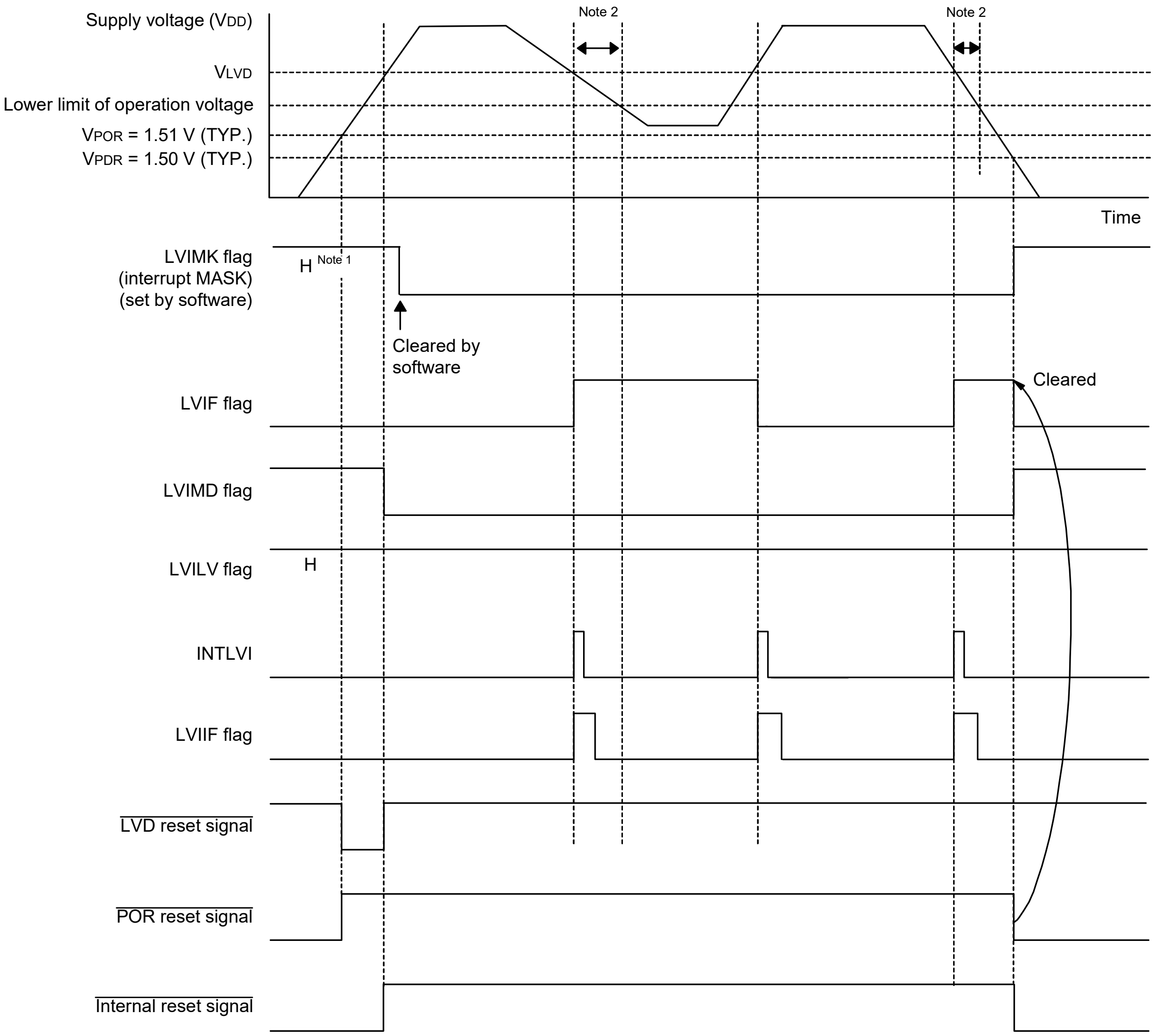
In the interrupt mode (option byte LVIMDS1, LVIMDS0 = 0, 1), the state of an internal reset by LVD is retained until the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}) after power is supplied (after the first release of the POR). The internal reset is released when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}).

An interrupt request signal by LVD (INTLVI) is generated, when the supply voltage (V_{DD}) falls below the voltage detection level (V_{LVD}) or when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}) after the second release of the POR. 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 **35.4, 36.4 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 by the LVD is generated.

Figure 27 - 5 shows the timing of the interrupt request signal generated in the LVD interrupt mode.

Figure 27 - 5 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 **35.4, 36.4 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

27.4.3 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.

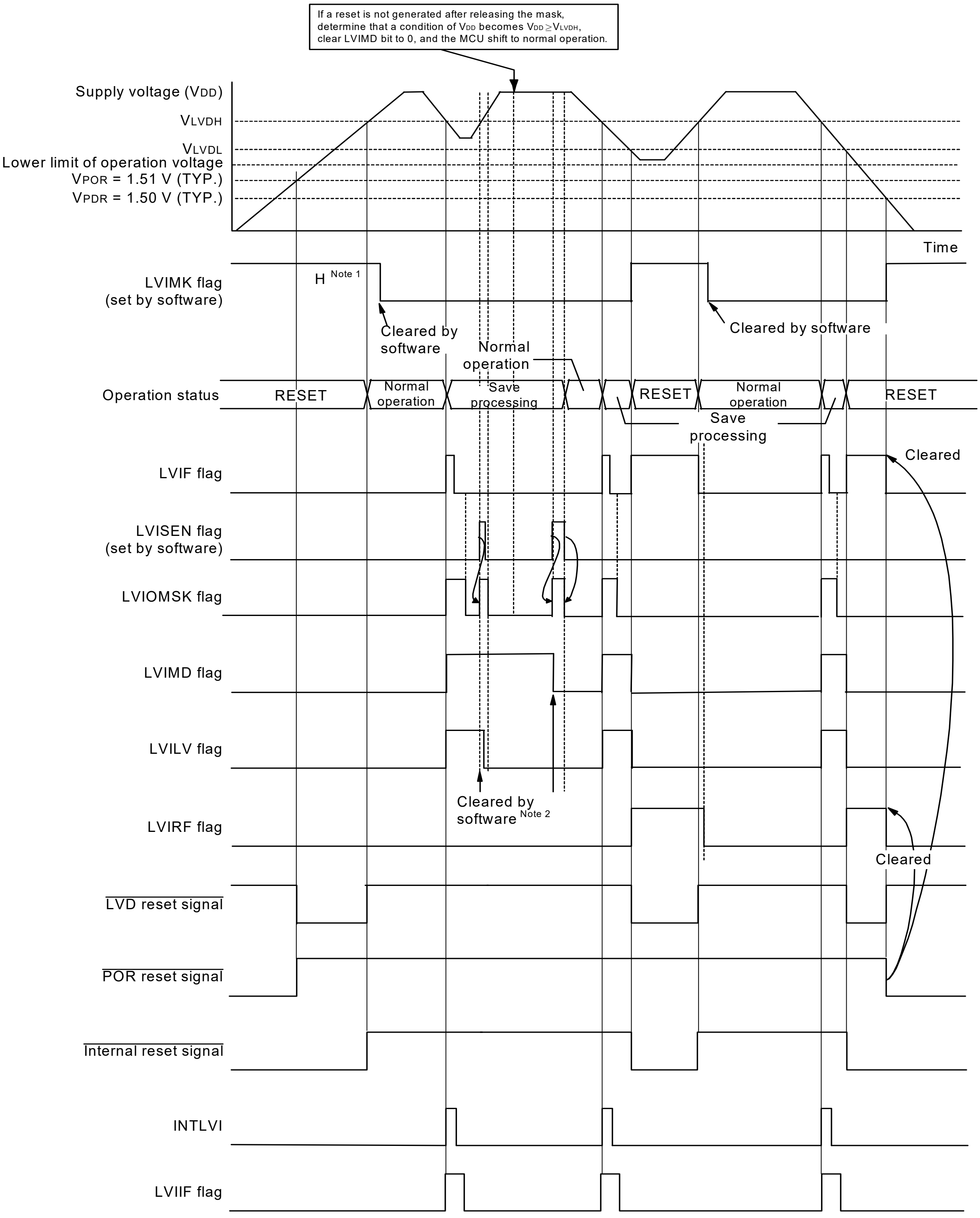
The operation is started in the following initial setting state when the interrupt & reset mode is set.

- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- See **27.3.2 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 27 - 7 Setting Procedure for Operating Voltage Check and Reset**.

Figures 27 - 6 (1/2) and 27 - 6 (2/2) show the timing of the internal reset signal and interrupt signal generated in the LVD interrupt & reset mode.

Figure 27 - 6 Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (1/2)



If a reset is not generated after releasing the mask, determine that a condition of V_{DD} becomes $V_{DD} \geq V_{LVDH}$, clear LVIMD bit to 0, and the MCU shift to normal operation.

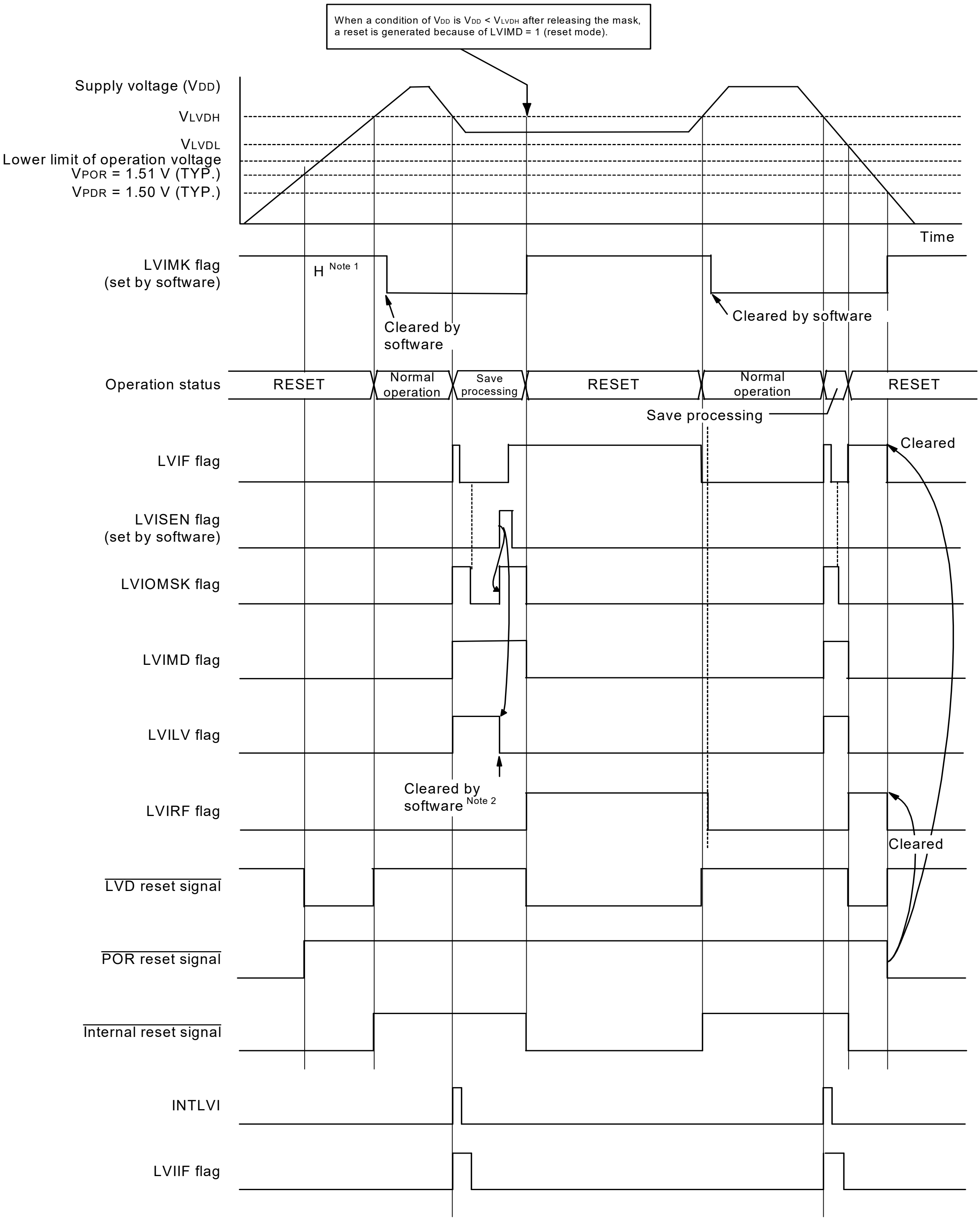
(**Notes** and **Remark** are listed on the next page.)

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 27 - 7 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 27 - 6 Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (2/2)



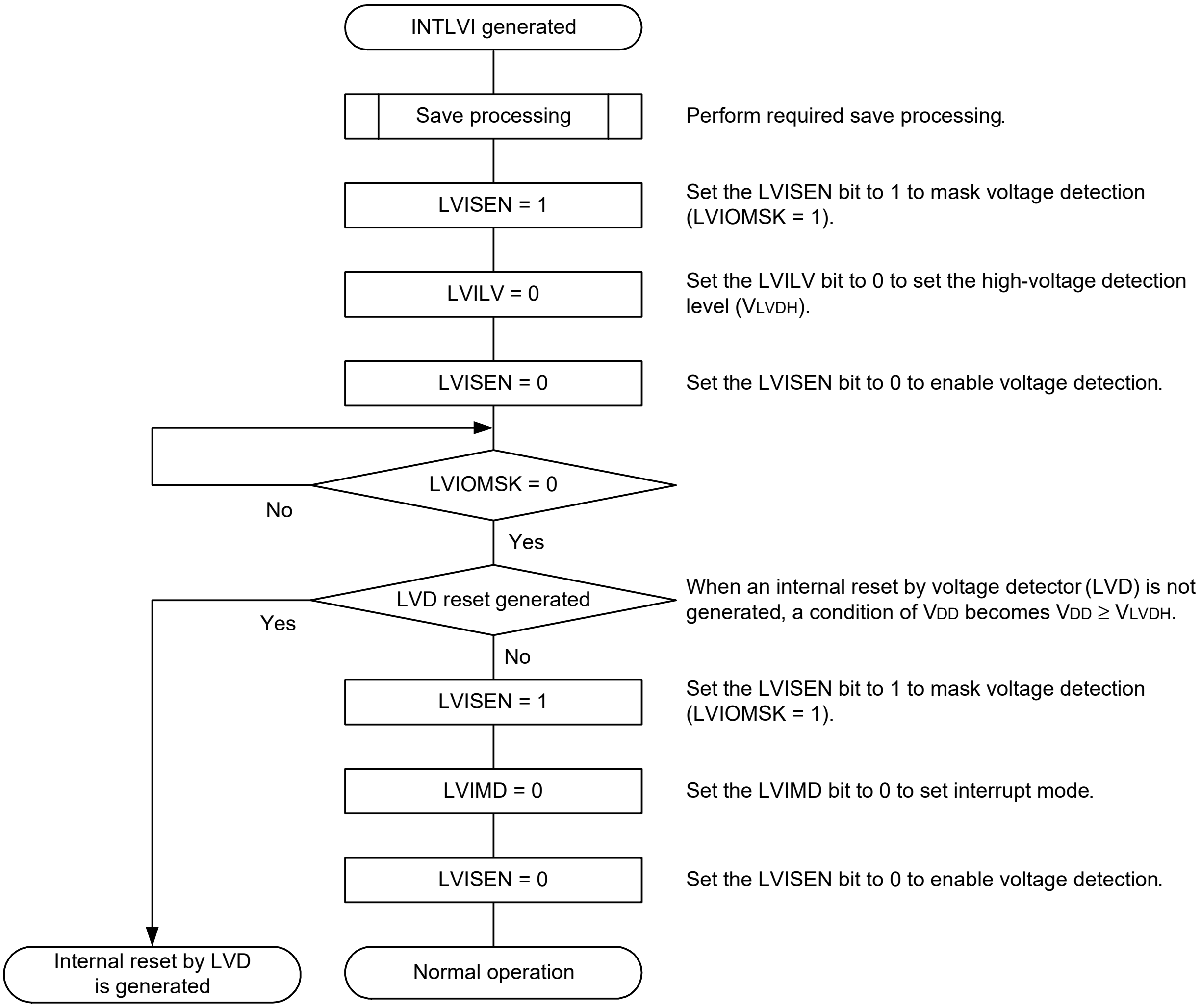
(**Notes** and **Remark** are listed on the next page.)

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 27 - 7 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 27 - 7 Setting Procedure for Operating Voltage Check and Reset


INTLVI generated
Save processing
Perform required save processing.
LVISEN = 1
Set the LVISEN bit to 1 to mask voltage detection (LVIOMSK = 1).
LVILV = 0
Set the LVILV bit to 0 to set the high-voltage detection level (VLVDH).
LVISEN = 0
Set the LVISEN bit to 0 to enable voltage detection.
LVIOMSK = 0
No
Yes
LVD reset generated
When an internal reset by voltage detector (LVD) is not generated, a condition of VDD becomes VDD ≥ VLVDH.
Yes
No
LVISEN = 1
Set the LVISEN bit to 1 to mask voltage detection (LVIOMSK = 1).
LVIMD = 0
Set the LVIMD bit to 0 to set interrupt mode.
LVISEN = 0
Set the LVISEN bit to 0 to enable voltage detection.
Internal reset by LVD is generated
Normal operation

27.5 Changing of LVD Detection Voltage Setting

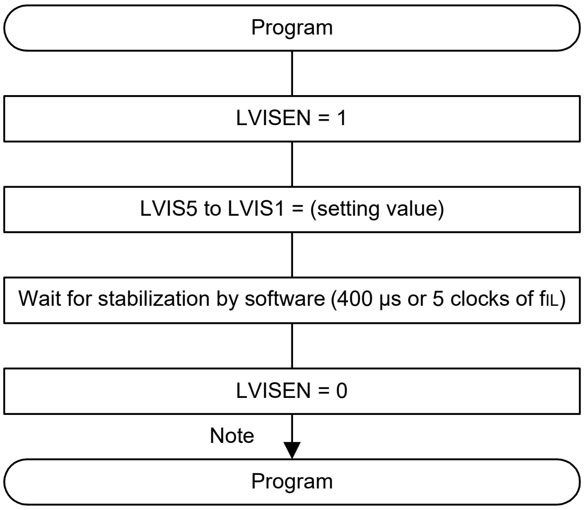
To change the LVD detection voltage by software, use the following procedure.

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 27 - 8 Changing of LVD Detection Voltage Setting

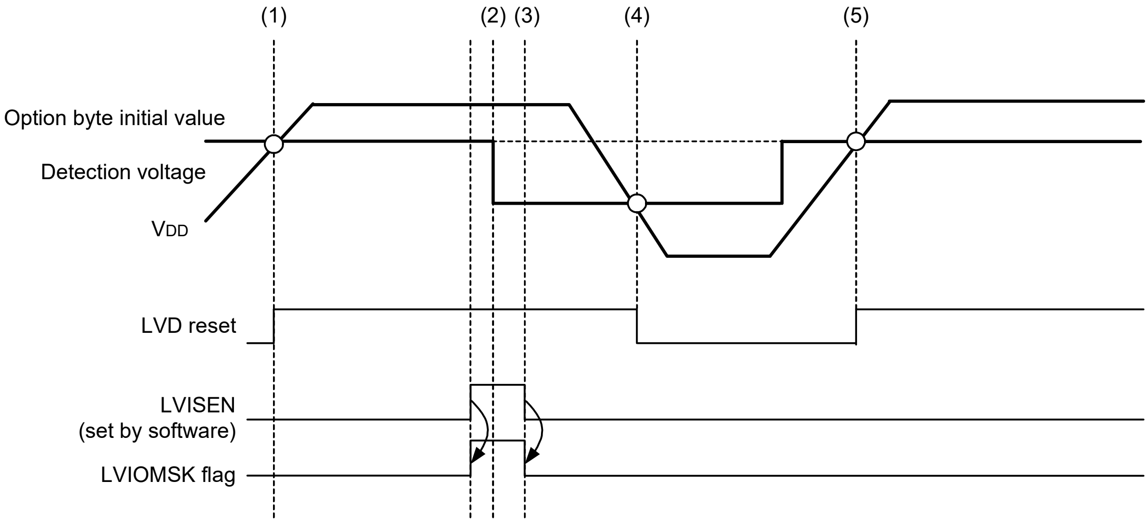

Program
LVISEN = 1
LVIS5 to LVIS1 = (setting value)
Wait for stabilization by software (400 μs or 5 clocks of fIL)
LVISEN = 0
Note
Program

Note After LVISEN is set to 0, LVD is detected if V_{LVD} > V_{DD}, and a reset/interrupt is generated.

27.5.1 Changing of LVD detection voltage setting in LVD reset mode

Figure 27 - 9 shows an Example of Timing for Changing LVD Detection Voltage Setting in LVD Reset Mode.

Figure 27 - 9 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 fIL 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

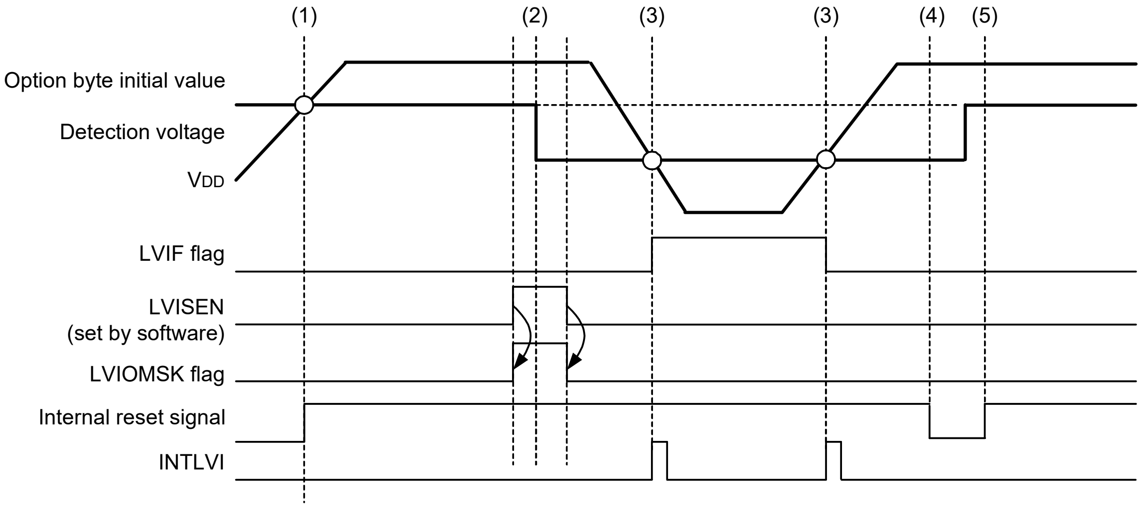
When changing the LVD detection voltage setting, note the following.

Caution The value of the reset release voltage in LVD reset mode is set to the set value in the option byte.

27.5.2 Changing of LVD detection voltage setting in LVD interrupt mode

Figure 27 - 10 shows an Example of Timing for Changing LVD Detection Voltage Setting in LVD Interrupt Mode.

Figure 27 - 10 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.

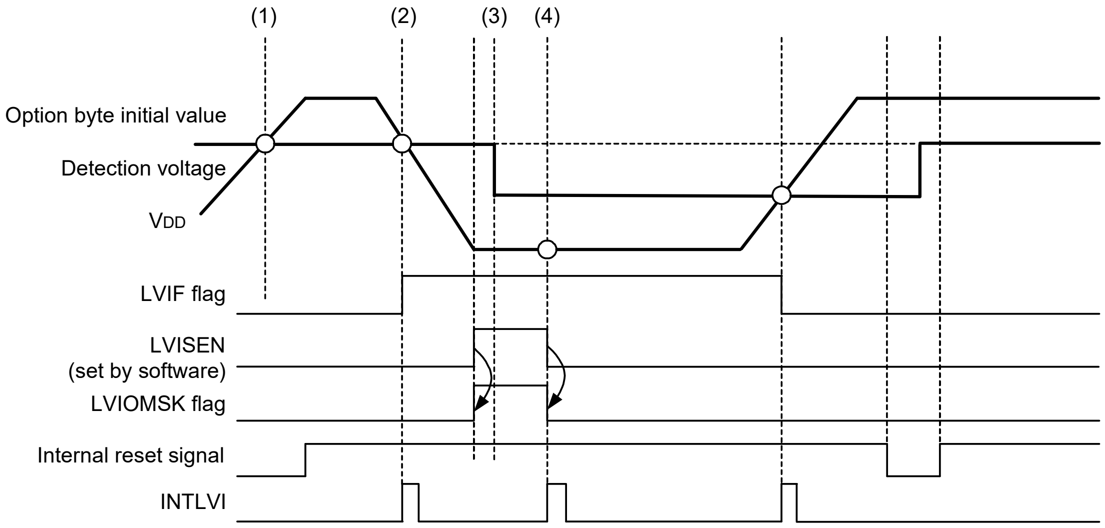
When changing the LVD detection voltage setting, note the following.

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 27 - 11.

Figure 27 - 11 Example of Timing for Changing LVD Detection Voltage Using LVDIS When VDD < VLVD



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 VDD < VLVD at the same time the masking is released, an interrupt is generated.

27.6 Cautions for Voltage Detector

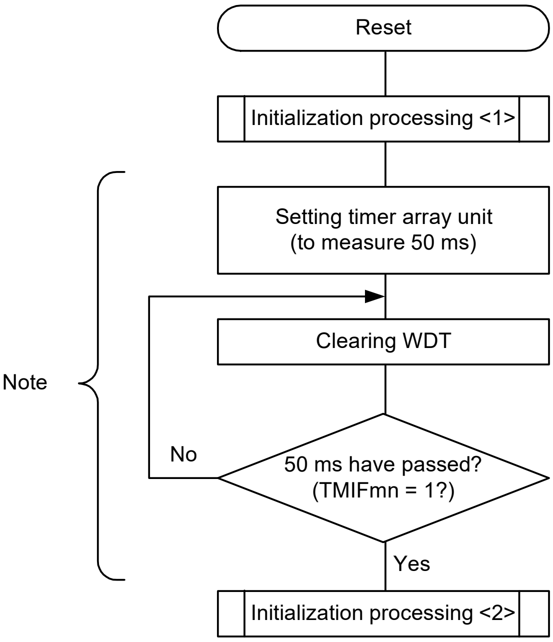
(1) Voltage fluctuation when power is supplied

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 27 - 12 Example of Software Processing If Supply Voltage Fluctuation is 50 ms or Less in Vicinity of LVD Detection Voltage



Refer to **Figure 25 - 5 Procedure for Checking Reset Source**.

; e.g. f_{CLK} = High-speed on-chip oscillator clock (4.04 MHz (MAX.))
Source: f_{MCK} = (4.04 MHz (MAX.))/2^8,
where comparison value = 789: ≈ 50 ms
Timer starts (TSmn = 1).

; Initial setting for port.
Setting of division ratio of system clock,
such as setting of timer or A/D converter.

Note If reset is generated again during this period, initialization processing <2> is not started.

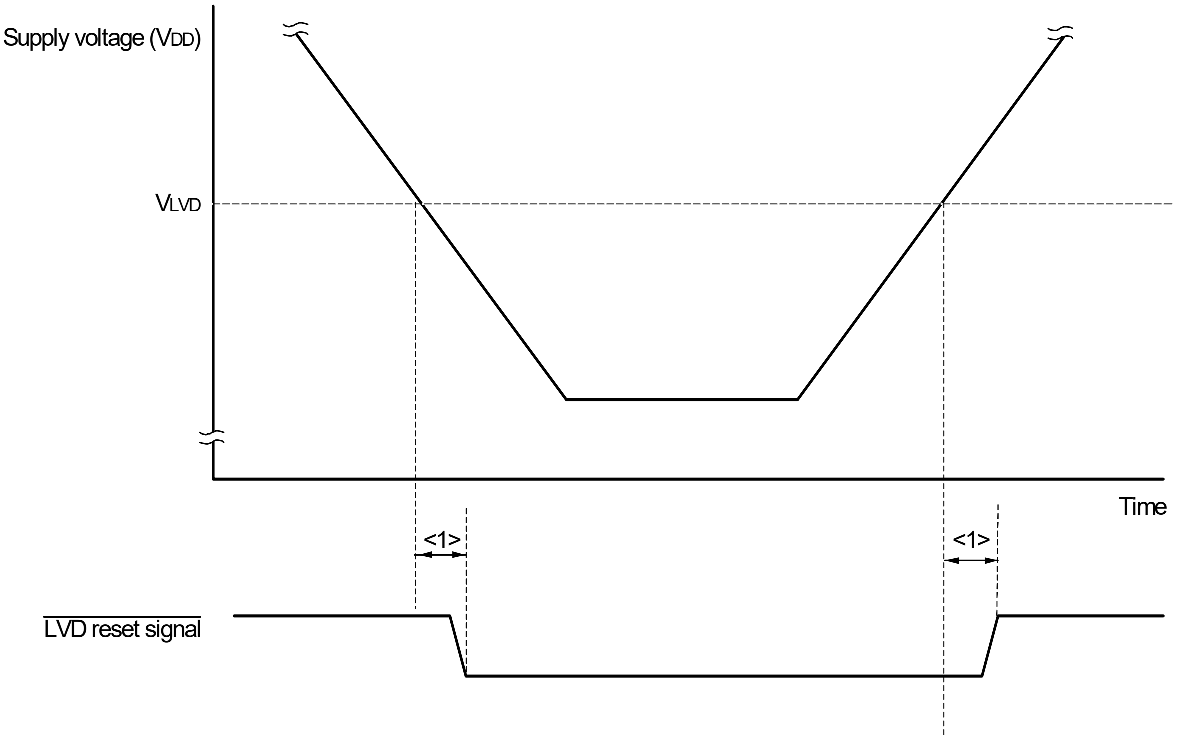
Remark m = 0
n = 0 to 3

(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 27 - 13**).

Figure 27 - 13 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 **35.4, 36.4 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 **35.4, 36.4 AC Characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

CHAPTER 28 SAFETY FUNCTIONS

28.1 Overview of Safety Functions

<R> The following safety functions are provided in the RL78/G11 to comply with the IEC60730 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/G11 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 safety standards. <R>

28.2 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 28.3 Operation of Safety Functions.

28.3 Operation of Safety Functions

28.3.1 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/G11 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 (170 µs@24 MHz with 16-Kbytes 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.

28.3.1.1 Flash memory CRC control register (CRC0CTL)

This register is used to control the operation of the high-speed CRC ALU.

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 28 - 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	0	0	0	0	0	0

CRC0EN	Control of high-speed CRC ALU operation
0	Stop the operation.
1	Start the operation according to HALT instruction execution.

The operation range of the high-speed CRC is from 00000H to 03FFBH (16 K - 4 bytes).

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.

28.3.1.2 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 28 - 2 Format of Flash memory CRC operation result register (PGCRCL)

Address: F02F2H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
PGCRCL	PGCRC15	PGCRC14	PGCRC13	PGCRC12	PGCRC11	PGCRC10	PGCRC9	PGCRC8

7	6	5	4	3	2	1	0
PGCRC7	PGCRC6	PGCRC5	PGCRC4	PGCRC3	PGCRC2	PGCRC1	PGCRC0

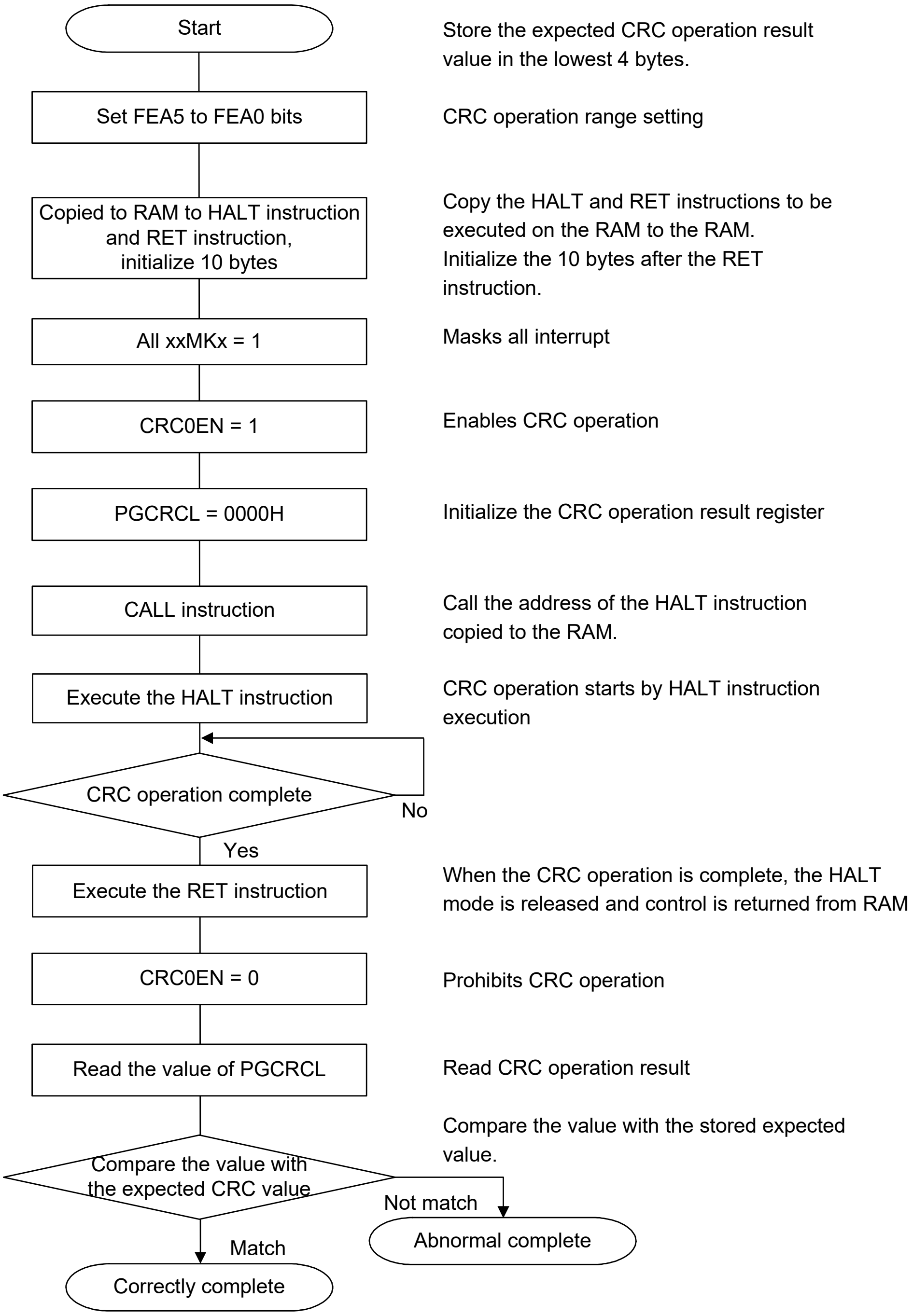
PGCRC15 to PGCRC0	High-speed CRC operation results
0000H to FFFFH	Store the high-speed CRC operation results.

Caution The PGCRCL register can only be written if CRC0EN (bit 7 of the CRC0CTL register) = 1.

Figure 28 - 3 shows the Flowchart of Flash Memory CRC Operation Function (High-speed CRC).

<Operation flow>

Figure 28 - 3 Flowchart of Flash Memory CRC Operation Function (High-speed CRC)



Store the expected CRC operation result value in the lowest 4 bytes.

CRC operation range setting

Copy the HALT and RET instructions to be executed on the RAM to the RAM.
Initialize the 10 bytes after the RET instruction.

Masks all interrupt

Enables CRC operation

Initialize the CRC operation result register

Call the address of the HALT instruction copied to the RAM.

CRC operation starts by HALT instruction execution

When the CRC operation is complete, the HALT mode is released and control is returned from RAM

Prohibits CRC operation

Read CRC operation result

Compare the value with the stored expected value.

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.

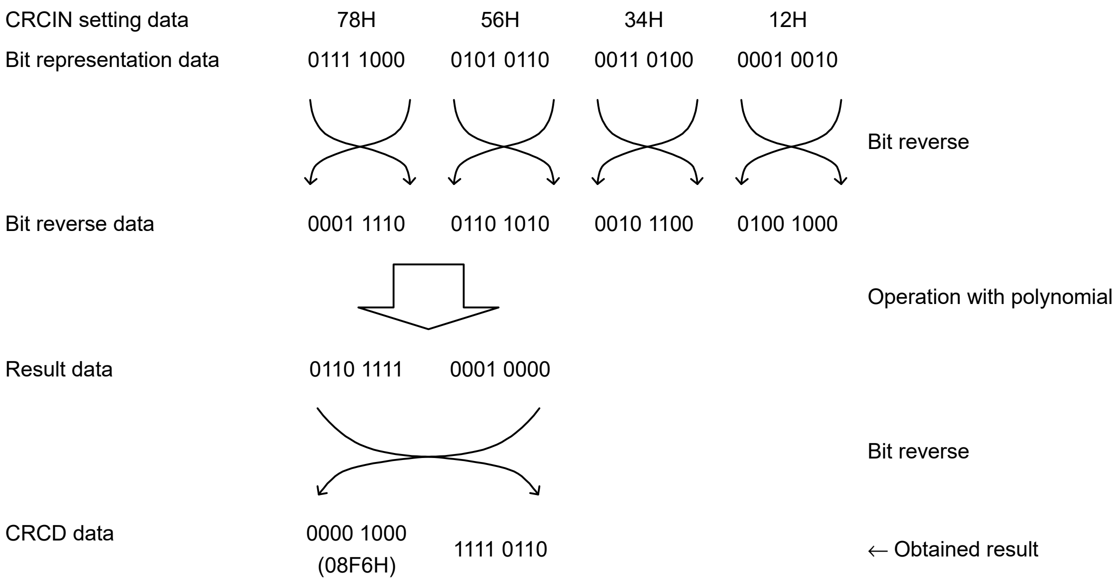
The expected CRC operation value can be calculated by using the integrated development environment CubeSuite+ development environment. Refer to the CubeSuite+ integrated development environment user's manual for details.

28.3.2 CRC operation function (general-purpose CRC)

<R> In the RL78/G11, 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.

28.3.2.1 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 28 - 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.

28.3.2.2 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 (fCLK) 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 28 - 5 Format of CRC data register (CRCD)

Address: F02FAH After reset: 0000H R/W

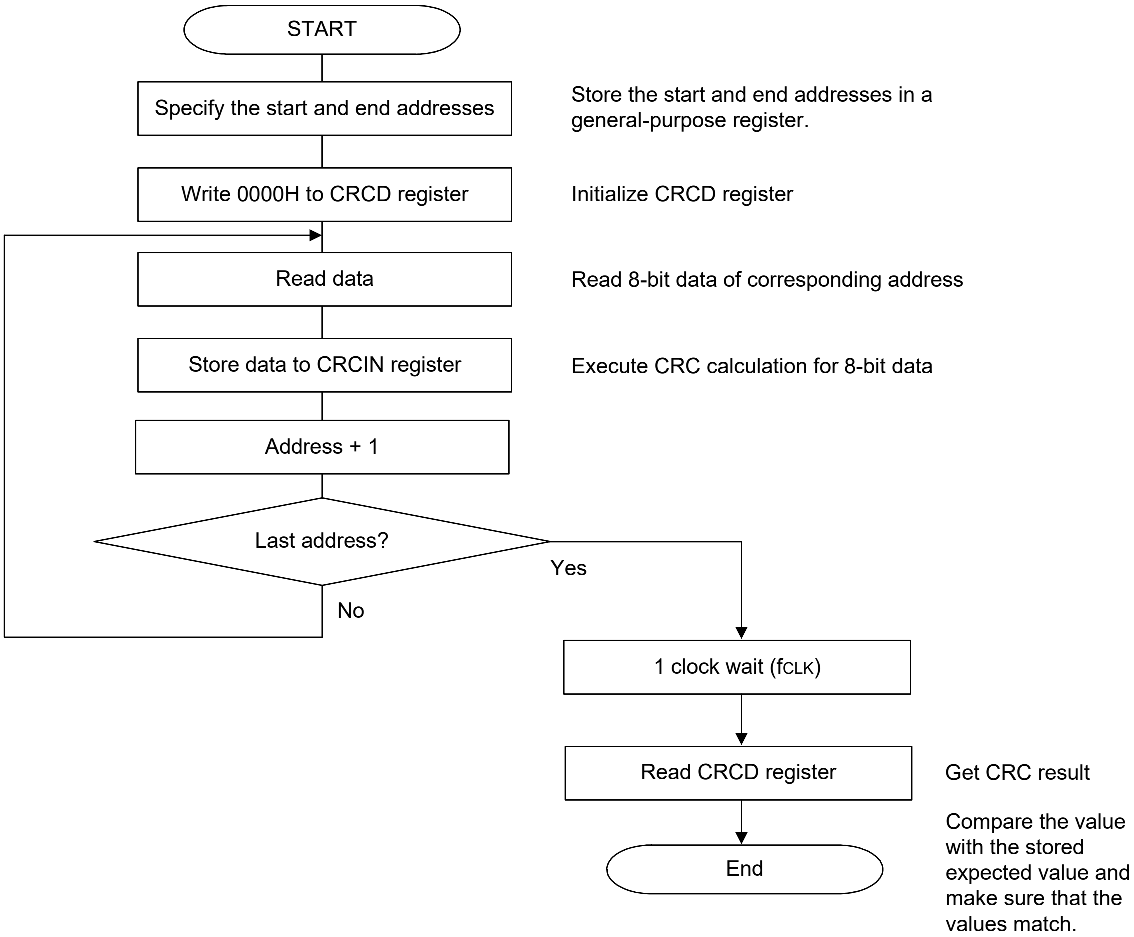
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCD																

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 28 - 6 CRC Operation Function (General-Purpose CRC)


START
Specify the start and end addresses
Store the start and end addresses in a general-purpose register.
Write 0000H to CRCD register
Initialize CRCD register
Read data
Read 8-bit data of corresponding address
Store data to CRCIN register
Execute CRC calculation for 8-bit data
Address + 1
Last address?
Yes
No
1 clock wait (fCLK)
Read CRCD register
Get CRC result
Compare the value with the stored expected value and make sure that the values match.
End

28.3.3 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/G11'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.

28.3.3.1 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 28 - 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

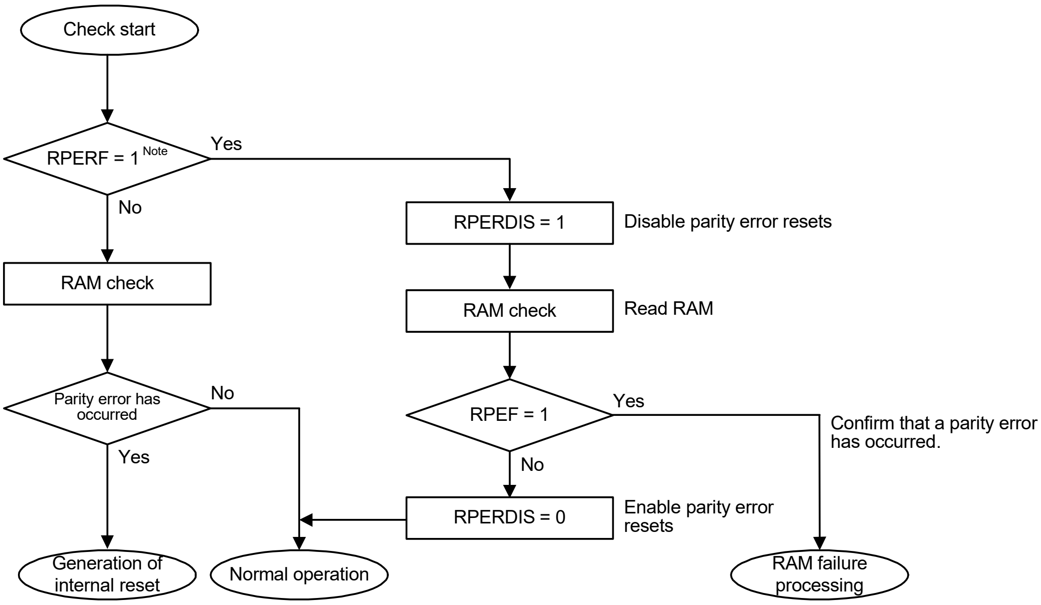
RPERDIS	Parity error reset mask flag
0	Enable parity error resets.
1	Disable parity error resets.

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 28 - 8 RAM Parity Error Check Flow


Check start
RPERF = 1 Note
Yes
No
RPERDIS = 1
Disable parity error resets
RAM check
RAM check
Read RAM
Parity error has occurred
No
Yes
RPEF = 1
Yes
No
Confirm that a parity error has occurred.
RPERDIS = 0
Enable parity error resets
Generation of internal reset
Normal operation
RAM failure processing

Note See **CHAPTER 25 RESET FUNCTION** for details on how to confirm internal resets due to RAM parity errors.

28.3.4 RAM guard function

<R>

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.

28.3.4.1 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 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 28 - 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
0	0	The guard function is disabled. Writing to the RAM is allowed.
1	0	256-byte space from the start address of the RAM
1	1	512-byte space from the start address of the RAM

28.3.5 SFR guard function

<R>

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.

28.3.5.1 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 28 - 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, 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, INTPEG

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

Note Pxx (Port register) is not guarded.

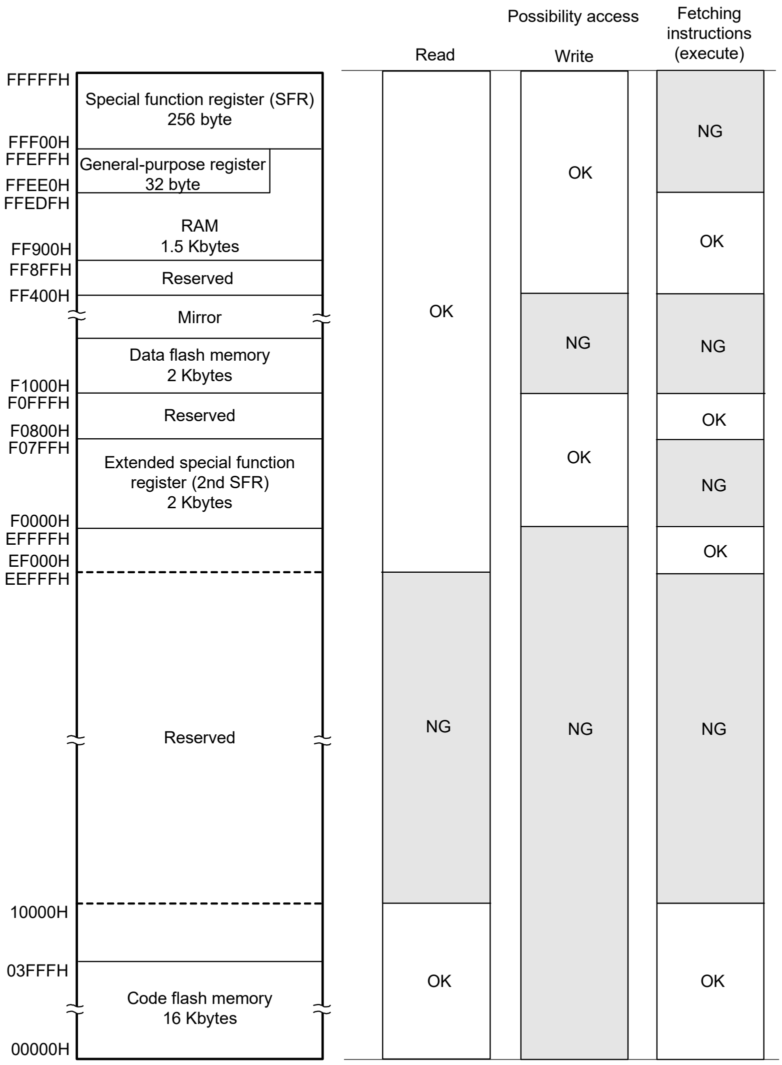
28.3.6 Invalid memory access detection function

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 28 - 11.

Figure 28 - 11 Invalid access detection area



28.3.6.1 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 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 28 - 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	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

IAWEN[Note]	Invalid memory access
0	Detection of the invalid memory access is disabled.
1	Detection of the invalid memory access is enabled.

Note Writing 1 to the IAWEN bit is only valid. Writing 0 to the IAWEN bit is invalid once it is set to 1.

Remark Detection of the invalid memory access is enabled even if the IAWEN bit is set to 0 while the option byte WDTON is set to 1 (operation of the watchdog timer counter is enabled).

28.3.7 Frequency detection function

The IEC60730 standard mandates checking that the oscillation frequency is correct.

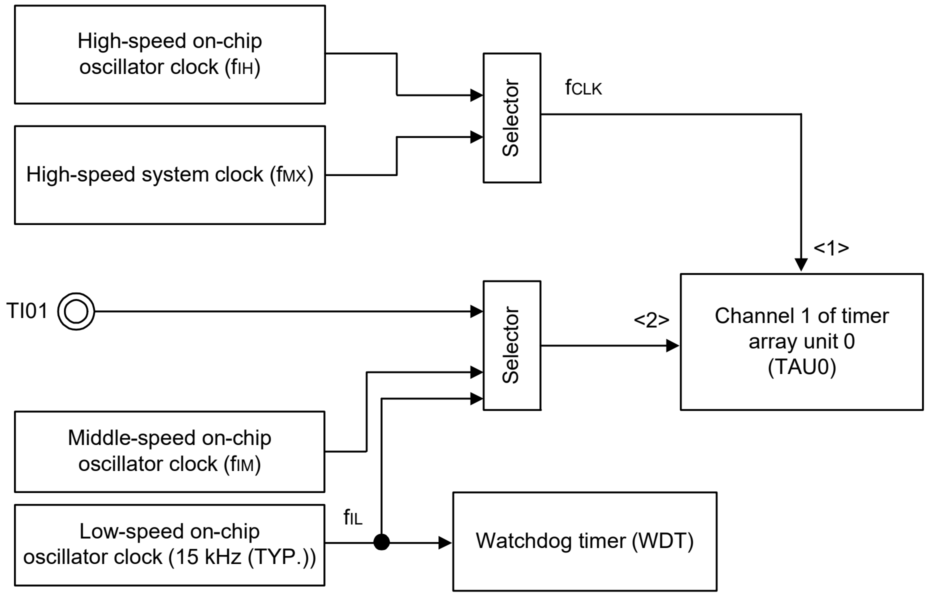
By using the CPU/peripheral hardware clock frequency (f_{CLK}) and measuring the pulse width of the input signal to channel 1 of the timer array unit 0 (TAU0), whether the proportional relationship between the two clock frequencies is correct can be determined. Note that, however, if one or both clock 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})

Figure 28 - 13 Configuration of Frequency Detection Function


High-speed on-chip oscillator clock (fIH)
High-speed system clock (fMX)
Selector
fCLK
<1>
TI01
Selector
<2>
Channel 1 of timer array unit 0 (TAU0)
Middle-speed on-chip oscillator clock (fIM)
Low-speed on-chip oscillator clock (15 kHz (TYP.))
fIL
Watchdog timer (WDT)

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**.

28.3.7.1 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 28 - 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	TIS06	TIS05	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})
Other than above			Setting prohibited

TIS06	TIS05	Selection of timer input used with channel 3
0	0	Input signal of timer input pin (TI03)
0	1	VCOUT1 output signal of CMP 1 [Note 1]
1	0	VCOUT0 output signal of CMP0 [Note 2]
1	1	Setting prohibited

Note 1. When VCOUT1 is selected as the input signal of TI03, the VCOUT1 signal cannot be output to the external pin.

Note 2. When VCOUT0 is selected as the input signal of TI03, the VCOUT0 signal cannot be output to the external pin.

28.3.8 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.

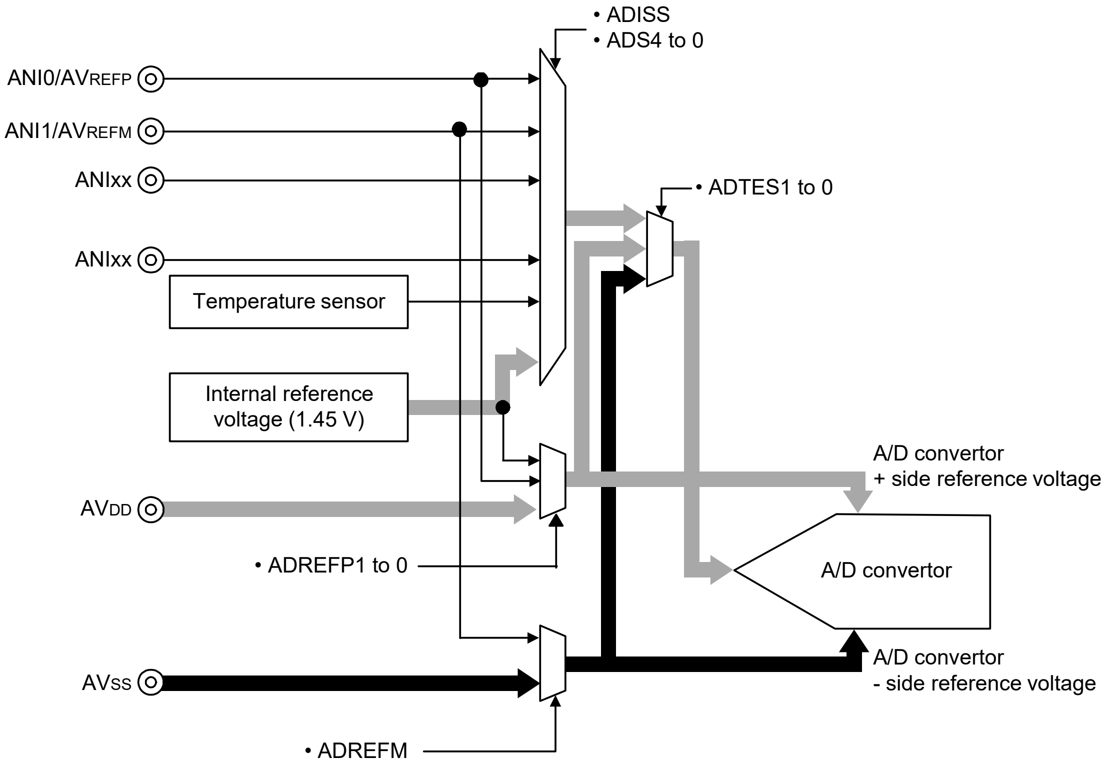
(1) Select the ANIx pin as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 0, 0).
(2) Perform A/D conversion for the ANIx pin (conversion result 1-1).
(3) Select the negative reference voltage of the A/D converter as the target for A/D conversion by setting the ADTES register (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 28 - 15 Configuration of A/D Test Function



28.3.8.1 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 28 - 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

28.3.8.2 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 28 - 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	P20/ANI0/AVREFP pin
0	0	0	0	0	1	ANI1	P21/ANI1/AVREFM pin
0	0	0	0	1	0	ANI2	P22/ANI2 pin
0	0	0	0	1	1	ANI3	P23/ANI3 pin
0	1	0	0	0	0	ANI16	P01/ANI16 pin
0	1	0	0	0	1	ANI17	P00/ANI17 pin
0	1	0	0	1	0	ANI18	P33/ANI18 pin
0	1	0	0	1	1	ANI19 [Note]	P32/ANI19 [Note]
0	1	0	1	0	0	ANI20	P31/ANI20
0	1	0	1	0	1	ANI21	P30/ANI21
0	1	0	1	1	0	ANI22	P56/ANI22
0	1	0	1	1	1	—	PGAOUT
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	

Note 25-pin and 24-pin products only

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, 2, 3 and 5 (PM0, PM2, PM3 and PM5).

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, 2, 3 and 5 (PMC0, PMC2, PMC3 and PMC5).

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 16.7.4 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 35.3.2, 36.3.2 Supply current characteristics is added.

28.3.9 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.

28.3.9.1 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 28 - 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.

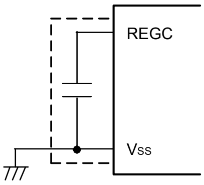
Caution While the PMS0 bit in the PMS register is set to 1, do not change the value of the port register (Pxx) using a bit manipulation instruction. To change the value of the port register (Pxx), use an 8-bit data manipulation instruction.

Remark m = 0, 2 to 5, 12, 13
n = 0 to 7
For input port pins (P121, P122, P125, and P137), setting PMS0 to 1 is invalid.

CHAPTER 29 REGULATOR

29.1 Regulator Overview

The RL78/G11 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 Vss via a capacitor (0.47 to 1 μF). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.



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

The regulator output voltage, see **Table 29 - 1**.

Table 29 - 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).

29.2 Register Controlling Regulator

The following register is used to control the regulator.

- Regulator mode control register (PMMC)

29.2.1 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 29 - 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 30 OPTION BYTE

30.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory of the RL78/G11 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.

Caution The option bytes should always be set regardless of whether each function is used.

30.1.1 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})

○ Control of P125/$\overline{\text{RESET}}$
- Select P125/INTP9 or $\overline{\text{RESET}}$

Caution 1. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 35.4, 36.4 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

- 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, 48 MHz.

Caution Set the same value as 000C2H to 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.

30.1.2 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.

30.2 Format of User Option Byte

The format of user option byte is shown below.

Figure 30 - 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 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 30 - 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	PORTSELB	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		
3.75 V	3.67 V					0	0		
2.92 V	2.86 V	2.75 V		1	1	1	0		
3.02 V	2.96 V					0	1		
4.06 V	3.98 V					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 27 VOLTAGE DETECTOR**.

Remark 2. The detection voltage is a typical value. For details, see **35.6.7, 36.6.7 LVD circuit characteristics**.

Figure 30 - 2 Format of User Option Byte (000C1H/010C1H) (2/4)

Address: 000C1H/010C1H [Note]

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	PORTSELB	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		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	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 27 VOLTAGE DETECTOR**.

Remark 2. The detection voltage is a typical value. For details, see **35.6.7, 36.6.7 LVD circuit characteristics**.

Figure 30 - 2 Format of User Option Byte (000C1H/010C1H) (3/4)

Address: 000C1H/010C1H [Note]

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	PORTSELB	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		
3.75 V	3.67 V		1	0	0	0		
4.06 V	3.98 V		1	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 27 VOLTAGE DETECTOR**.

Remark 2. The detection voltage is a typical value. For details, see **35.6.7, 36.6.7 LVD circuit characteristics**.

Figure 30 - 2 Format of User Option Byte (000C1H/010C1H) (4/4)

Address: 000C1H/010C1H Note

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	PORTSELB	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						

- Setting of P125/$\overline{\text{RESET}}$/INTP9 pin

PORTSELB	P125/$\overline{\text{RESET}}$ pin control
0	Port functions (P125/INTP9)
1	$\overline{\text{RESET}}$ input (the on-chip pull-up resistor is valid)

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Caution 1. After the power is turned on, P125 functions as $\overline{\text{RESET}}$ input. Even if an internal reset signal is released by power-on-reset (POR), the reset status continues as long as the low level is output to this pin. To use P125/INTP9, select the port function (PORTSELB = 0) by the option byte (000C1H/010C1H) and release all reset sources.

Caution 2. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 35.4, 36.4 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 27 VOLTAGE DETECTOR**.

Remark 3. The detection voltage is a typical value. For details, see **35.6.7, 36.6.7 LVD circuit characteristics**.

Figure 30 - 3 Format of Option Byte (000C2H/010C2H)

Address: 000C2H/010C2H Note

7	6	5	4	3	2	1	0
CMODE1	CMODE0	1	FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0

CMODE1	CMODE0	Setting of flash operation mode		
			Operating Frequency Range	Operating Voltage Range
0	0	LV (low-voltage main) mode	1 to 4 MHz	1.6 to 5.5 V
1	0	LS (low-speed main) mode	1 to 8 MHz	1.8 to 5.5 V
1	1	HS (high-speed main) mode	1 to 16 MHz	2.4 to 5.5 V
			1 to 24 MHz	2.7 to 5.5 V
Other than above		Setting prohibited		

FRQSEL4	FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator clock	
					f_{HOCO}	f_{IH}
1	0	0	0	0	48 MHz	24 MHz
0	0	0	0	0	24 MHz	24 MHz
0	1	0	0	1	16 MHz	16 MHz
0	0	0	0	1	12 MHz	12 MHz
0	1	0	1	0	8 MHz	8 MHz
0	0	0	1	0	6 MHz	6 MHz
0	1	0	1	1	4 MHz	4 MHz
0	0	0	1	1	3 MHz	3 MHz
0	1	1	0	0	2 MHz	2 MHz
0	1	1	0	1	1 MHz	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 bit 5 to 1.

Caution 2. The operating frequency range and operating voltage range depend on each operating mode of the flash memory. See 35.4, 36.4 AC Characteristics for details.

30.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

Figure 30 - 4 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.

30.4 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
                                        ; Use RESET input
                                        ; 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
                                        ; Select fCLK as the operating clock for timer KB
                DB          85H         ; Enables on-chip debug operation, does not erase flash memory
                                        ; data when security ID authorization fails
```

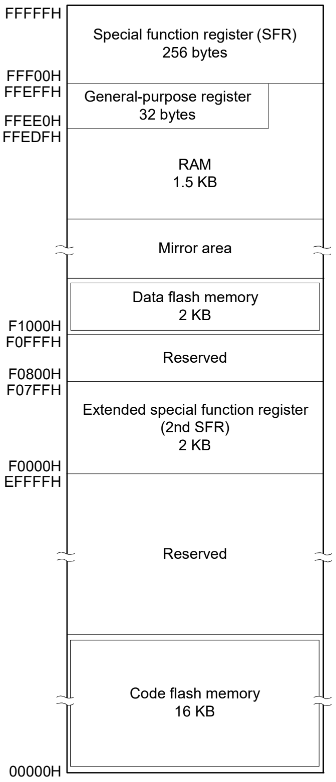
When the boot swap function is used during self-programming, 000C0H to 000C3H is switched to 010C0H to 010C3H. Describe to 010C0H to 010C3H, therefore, the same values as 000C0H to 000C3H as follows.

```
  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
                                            ; Use RESET input
                                            ; 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
                                            ; Select fCLK as the operating clock for timer KB
            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 31 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 **31.1**)
  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 **31.2**)
  Data can be written to the flash memory on-board through UART communication with an external device (microcontroller or ASIC).
- Self-Programming (see **31.6**)
  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 **31.8 Data Flash**.

31.1 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-FP6, FL-PR6
- 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.

Remark FL-PR6 and FA series are products of Naito Densei Machida Mfg. Co., Ltd.

Table 31 - 1 Wiring Between RL78/G11 and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer				Pin Name	Pin No.				
Signal Name		I/O	Pin Function		10-pin	16-pin	20-pin	24-pin	25-pin
PG-FP6, FL-PR6	E1, E2, E2 Lite, E20 on-chip debugging emulator				LSSOP	SSOP	LSSOP	HWQFN (4 × 4)	WFLGA (3 × 3)
—	TOOL0	I/O	Transmit/ receive signal	TOOL0/ P40	7	1	3	23	A5
SI/RxD	—	I/O	Transmit/ receive signal						
—	$\overline{\text{RESET}}$	Output	Reset signal	$\overline{\text{RESET}}$/ P125	1	2	4	24	B5
/RESET	—	Output							
V_{DD}		I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	5	8	10	6	B3
GND		—	Ground	V_{SS}	4	7	9	5	B2
				REGC [Note]	3	6	8	4	A2
FLMD1	EMV_{DD}	—	Driving power for TOOL0 pin	V_{DD}	5	8	10	6	B3

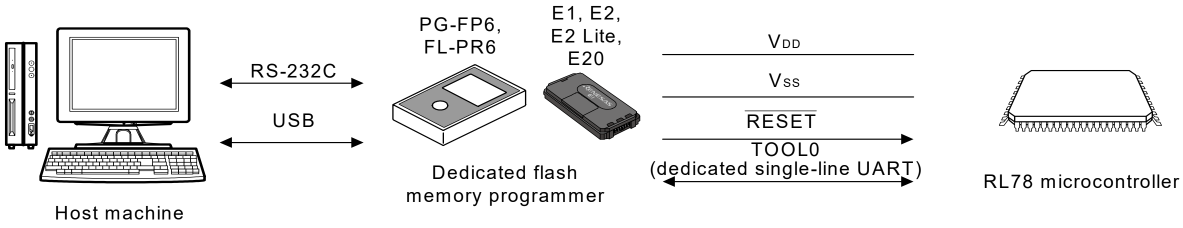
Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

Remark Pins that are not indicated in the above table can be left open when using the flash memory programmer for flash programming.

31.1.1 Programming Environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 31 - 1 Environment for Writing Program to Flash Memory



A host machine that controls the dedicated flash memory programmer is necessary.

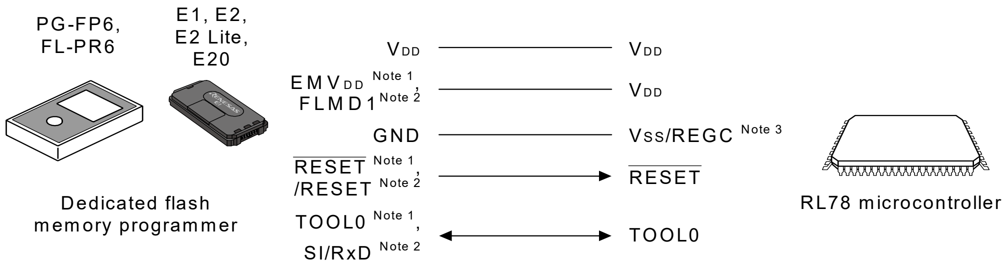
To interface between the dedicated flash memory programmer and the RL78 microcontroller, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART.

31.1.2 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 31 - 2 Communication with Dedicated Flash Memory Programmer



Note 1. When using E1, E2, E2 Lite, E20 on-chip debugging emulator.

Note 2. When using PG-FP6 or FL-PR6.

Note 3. 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-FP6, FL-PR6, or E1, E2, E2 Lite, E20 on-chip debugging emulator for details.

Table 31 - 2 Pin Connection

Dedicated Flash Memory Programmer				RL78 microcontroller
Signal Name		I/O	Pin Function	Pin Name [Note 1]
PG-FP6, FL-PR6	E1, E2, E2 Lite, E20 on-chip debugging emulator			
V_{DD}		I/O	V_{DD} voltage generation/power monitoring	V_{DD}
GND		—	Ground	V_{SS}, REGC [Note 2]
FLMD1	EMV_{DD}	—	Driving power for TOOL0 pin	V_{DD}, EV_{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 31 - 1**.

Note 2. Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

31.2 Serial Programming Using External Device (that Incorporates UART)

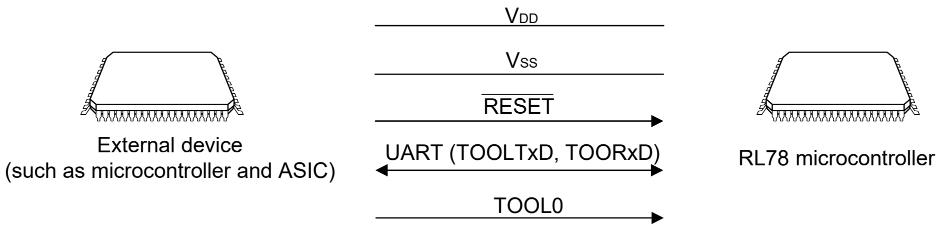
On-board data writing to the internal flash memory is possible by using the RL78 microcontroller and an external device (a microcontroller or ASIC) connected to a UART.

On the development of flash memory programmer by user, refer to the **RL78 Microcontrollers (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

31.2.1 Programming Environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 31 - 3 Environment for Writing Program to Flash Memory



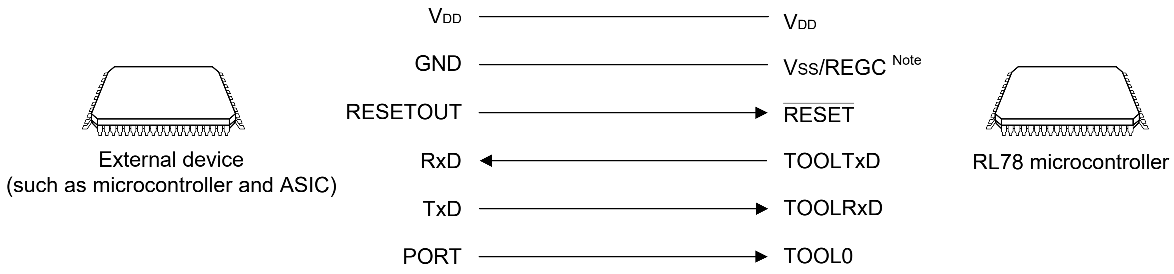
Processing to write data to or delete data from the RL78 microcontroller by using an external device is performed on-board. Off-board writing is not possible.

31.2.2 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 31 - 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 31 - 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).

31.3 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 **31.4.2 Flash memory programming mode**.

31.3.1 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 **35.10, 36.10 Timing of Entry to Flash Memory Programming Modes**).

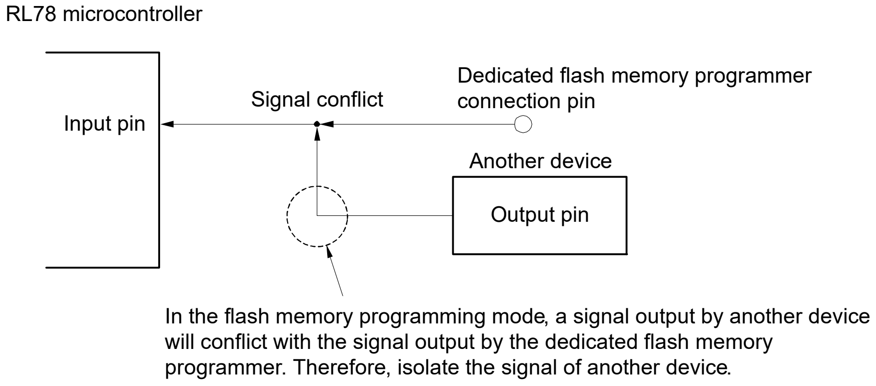
Remark 2. The SAU and IICA pins are not used for communication between the RL78 microcontroller and dedicated flash memory programmer, because single-line UART (TOOL0 pin) is used.

31.3.2 $\overline{\text{RESET}}$ 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 31 - 5 Signal Conflict ($\overline{\text{RESET}}$ Pin)


RL78 microcontroller
Input pin
Signal conflict
Dedicated flash memory programmer connection pin
Another device
Output pin
In the flash memory programming mode, a signal output by another device will conflict with the signal output by the dedicated flash memory programmer. Therefore, isolate the signal of another device.

31.3.3 Port pins

When the flash memory programming mode is set, all the pins not used for flash memory programming enter the same status as that immediately after reset. If external devices connected to the ports do not recognize the port status immediately after reset, the port pin must be connected to V_{DD}, or V_{SS} via a resistor

31.3.4 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.

31.3.5 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.

31.3.6 Power supply

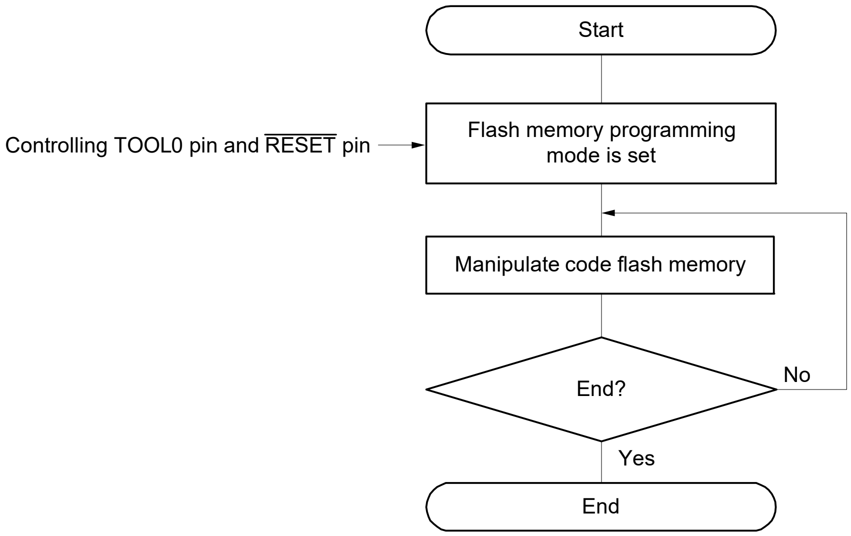
To use the supply voltage output of the flash memory programmer, connect the V_{DD} pin to V_{DD} of the flash memory programmer, and the V_{SS} pin to GND of the flash memory programmer.
To use the on-board supply voltage, connect in compliance with the normal operation mode.
However, when writing to the flash memory by using the flash memory programmer and using the on-board supply voltage, be sure to connect the V_{DD} and V_{SS} pins to V_{DD} and GND of the flash memory programmer to use the power monitor function with the flash memory programmer.

31.4 Programming Method

31.4.1 Serial programming procedure

The following figure illustrates a flow for rewriting the code flash memory through serial programming.

Figure 31 - 6 Code Flash Memory Manipulation Procedure


Start
Controlling TOOL0 pin and $\overline{\text{RESET}}$ pin
Flash memory programming mode is set
Manipulate code flash memory
End?
No
Yes
End

31.4.2 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.

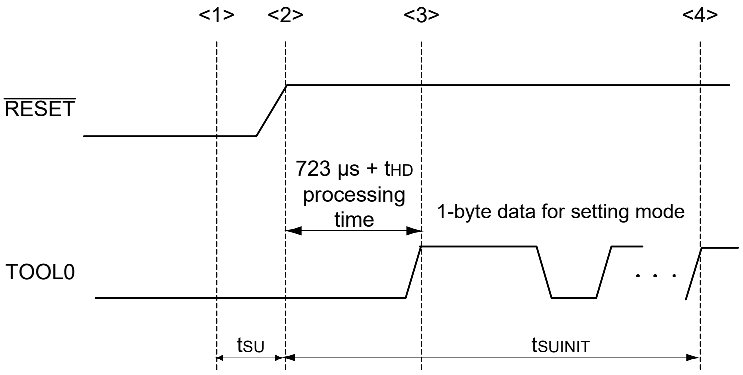
<When serial programming by using the dedicated flash memory programmer>
Connect the RL78 microcontroller to a dedicated flash memory programmer. Communication from the dedicated flash memory programmer is performed to automatically switch to the flash memory programming mode.

<When serial programming by using an external device>
Set the TOOL0 pin to the low level, and then cancel the reset (see **Table 31 - 4**). After that, enter flash memory programming mode according to the procedures <1> to <4> shown in **Figure 31 - 7**. For details, refer to the **RL78 microcontrollers (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

Table 31 - 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 31 - 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 **35.10, 36.10 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 31 - 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})	
$2.7\ V \leq V_{DD} \leq 5.5\ V$	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 **31.4.4 Communication commands**.

31.4.3 Selecting communication mode

Communication mode of the RL78 microcontroller as follows.

Table 31 - 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.

31.4.4 Communication commands

The RL78 microcontroller executes serial programming through the commands listed in **Table 31 - 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 31 - 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 31 - 8 and 31 - 9 show signature data list and example of signature data list.

Table 31 - 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 03FFFH (16 KB) → FFH, 3FH, 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 31 - 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	R5F1057A	10 bytes	52 = “R” 35 = “5” 46 = “F” 31 = “1” 30 = “0” 35 = “5” 37 = “7” 41 = “A” 20 = “ ” 20 = “ ”
Code flash memory area last address	Code flash memory area 00000H to 03FFFH (16 KB)	3 bytes	FFH 3FH 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

31.5 Processing Time for Each Command When PG-FP6 Is in Use (Reference Value)

The following shows the processing time for each command (reference value) when PG-FP6 is used as a dedicated flash memory programmer.

Table 31 - 10 Processing Time for Each Command When PG-FP6 Is in Use (Reference Value)

PG-FP6 Command	Port: TOOL0 (UART)
	Speed: 1M bps
	16 Kbytes
Erasing	1 s
Writing	1.4 s
Verification	1.5 s
Writing after erasing	1.5 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)

31.6 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)**.

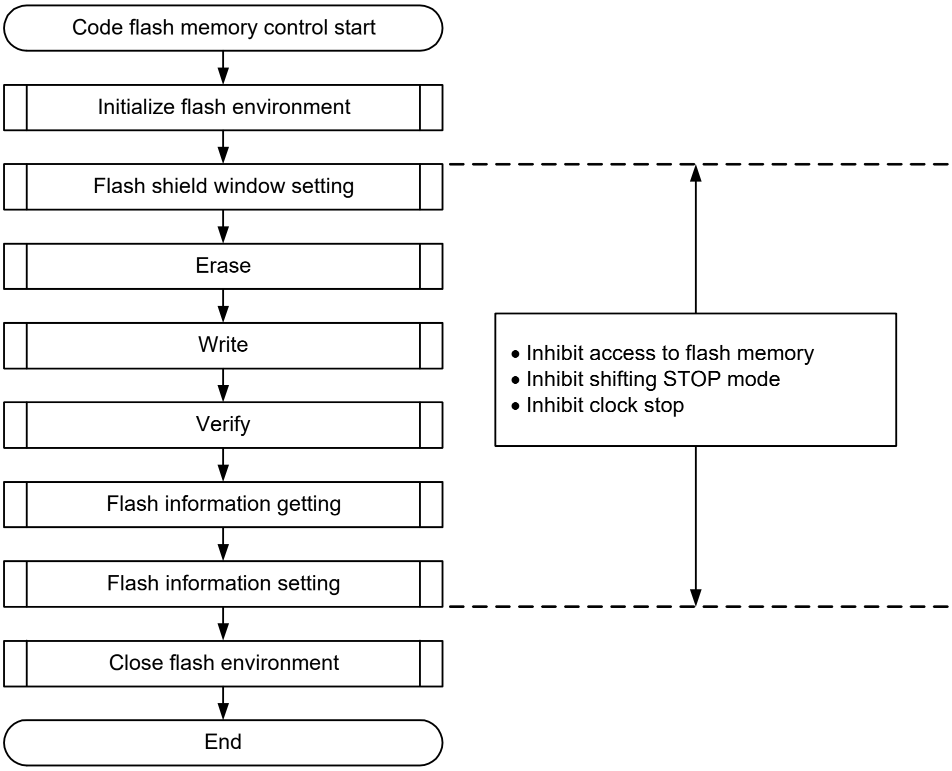
The self-programming function has two flash memory programming modes; wide voltage mode and full speed mode. Specify the mode that corresponds to the flash operation mode specified in bits CMODE1 and CMODE0 in option byte 000C2H.
Specify the full speed mode when the HS (high-speed main) mode is specified. 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.

31.6.1 Self-programming procedure

The following figure illustrates a flow for rewriting the code flash memory by using a flash self-programming library.

Figure 31 - 8 Flow of Self-Programming (Rewriting Flash Memory)


Code flash memory control start
Initialize flash environment
Flash shield window setting
Erase
Write
Verify
Flash information getting
Flash information setting
Close flash environment
End
• Inhibit access to flash memory
• Inhibit shifting STOP mode
• Inhibit clock stop

31.6.2 Boot swap function

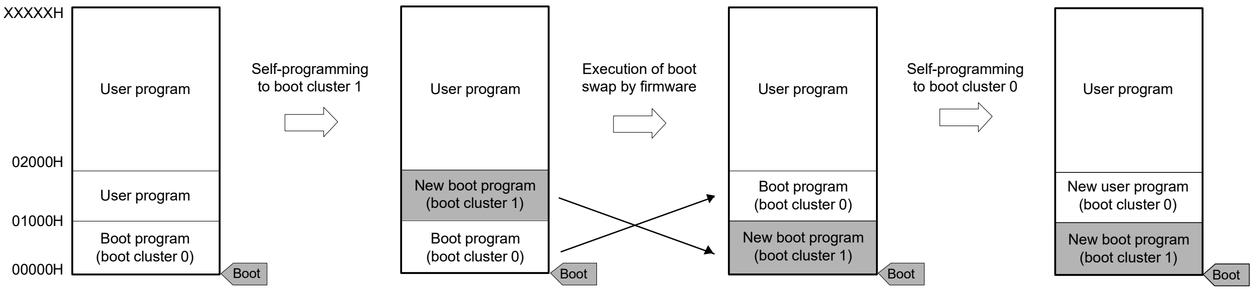
If rewriting the boot area failed by temporary power failure or other reasons, restarting a program by resetting or overwriting is disabled due to data destruction in the boot area.

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 31 - 9 Boot Swap Function

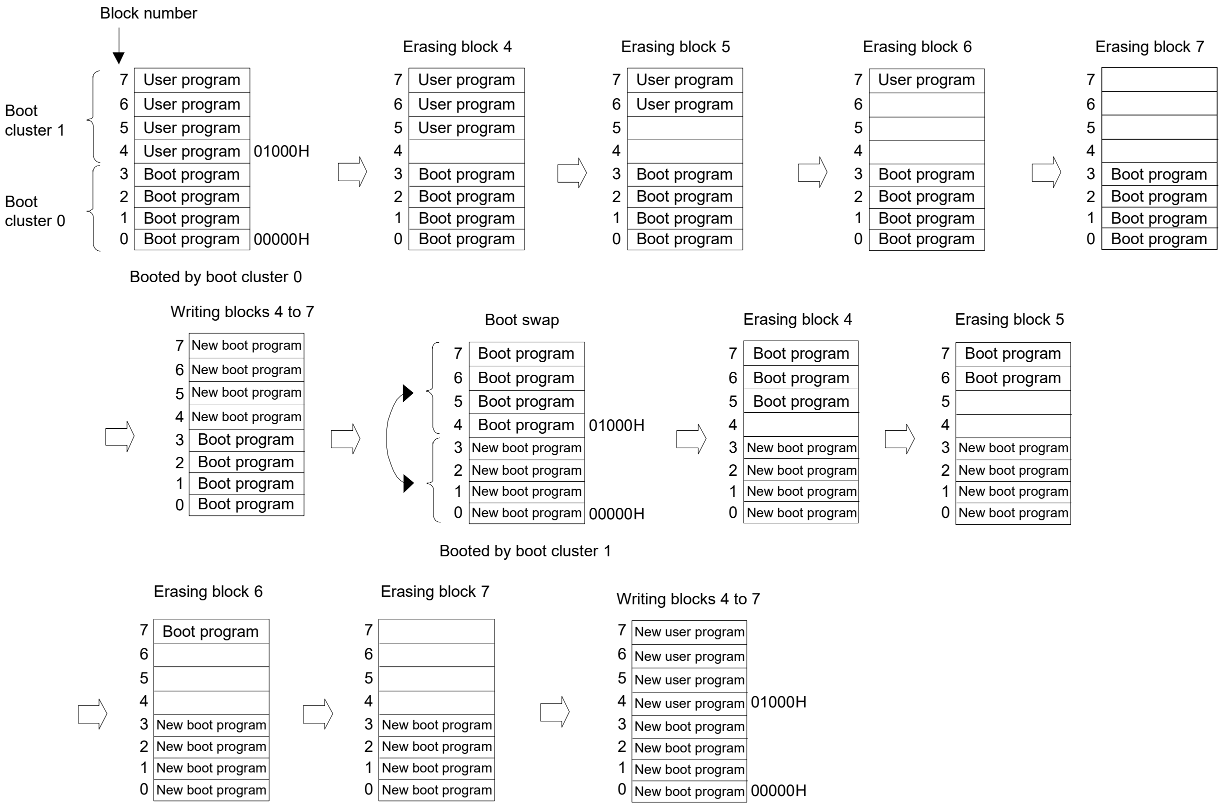


In an example of above figure, it is as follows.

Boot cluster 0: Boot area before boot swap

Boot cluster 1: Boot area after boot swap

Figure 31 - 10 Example of Executing Boot Swapping



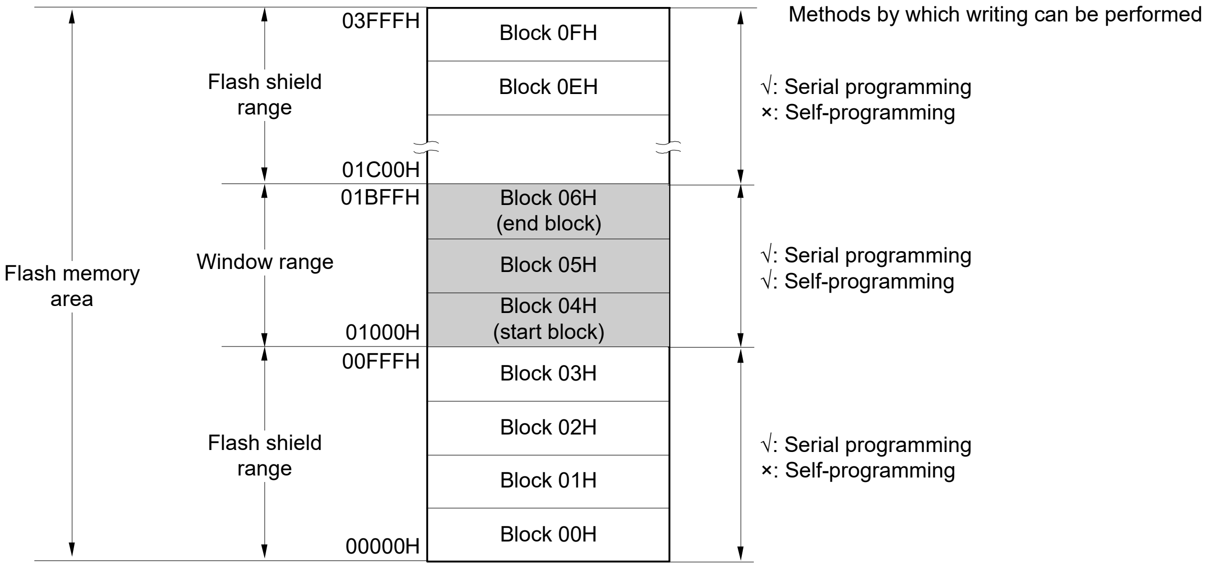
31.6.3 Flash shield window function

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 31 - 11 Flash Shield Window Setting Example (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 31 - 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 **31.7 Security Settings** to prohibit writing/erasing during serial programming.

31.7 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 only be set by serial programming (Security cannot be set by self-programming).
Each security setting can be used in combination.
Table 31 - 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 **31.6.3** for detail).

Table 31 - 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 **31.6.3** for detail).

Table 31 - 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.**

31.8 Data Flash

31.8.1 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 **31.6 Self-Programming**.

31.8.2 Register controlling data flash memory

31.8.2.1 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 31 - 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.

31.8.3 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

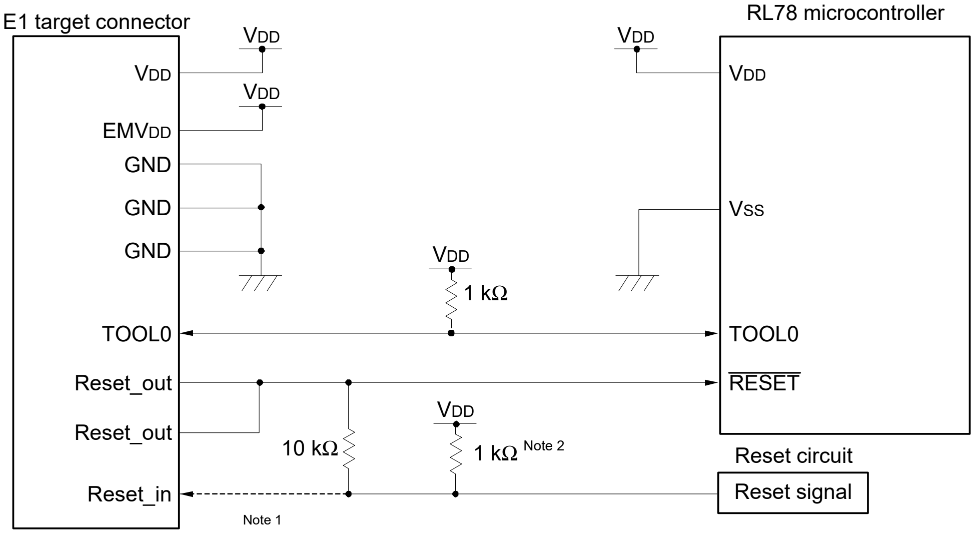
CHAPTER 32 ON-CHIP DEBUG FUNCTION

32.1 Connecting E1 On-chip Debugging Emulator

The RL78 microcontroller uses the V_{DD}, $\overline{\text{RESET}}$, TOOL0, and V_{SS} pins to communicate with the host machine via an E1 on-chip debugging emulator. Serial communication is performed by using a single-line UART that uses the TOOL0 pin.

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 32 - 1 Connection Example of E1 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)

32.2 On-Chip Debug Security ID

The RL78 microcontroller has an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 30 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 32 - 1 On-Chip Debug Security ID

Address	On-Chip Debug Security ID
000C4H to 000CDH	Any ID code of 10 bytes
010C4H to 010CDH	

32.3 Securing of User Resources

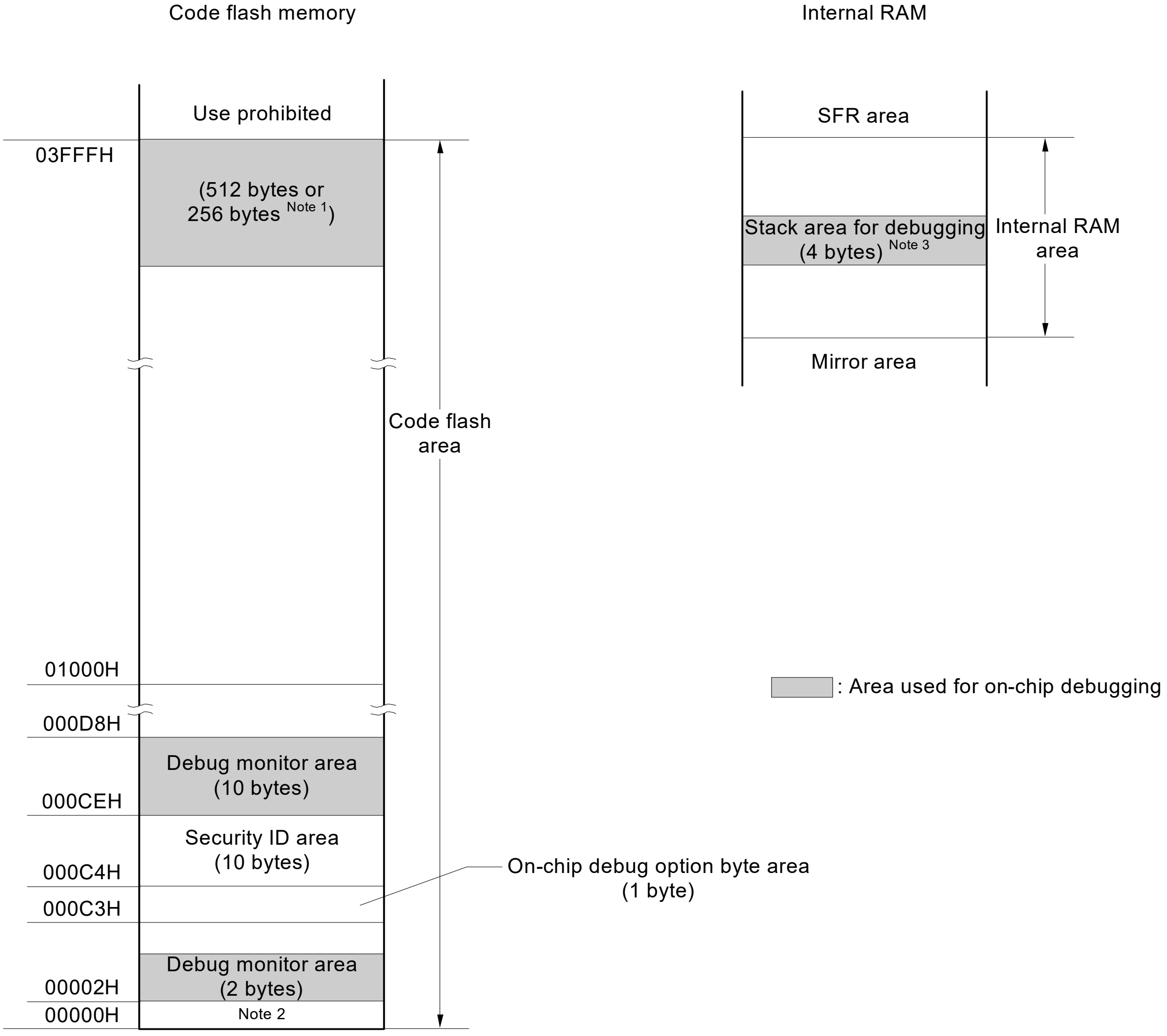
To perform communication between the RL78 microcontroller and E1 on-chip debugging emulator, as well as each debug function, the securing of memory space must be done beforehand.

If Renesas Electronics assembler or compiler is used, the items can be set by using link options.

(1) Securement of memory space

The shaded portions in Figure 32 - 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 32 - 2 Memory Spaces Where Debug Monitor Programs Are Allocated


Code flash memory
Use prohibited
03FFFH
(512 bytes or 256 bytes Note 1)
Code flash area
01000H
000D8H
Debug monitor area (10 bytes)
000CEH
Security ID area (10 bytes)
000C4H
000C3H
On-chip debug option byte area (1 byte)
Debug monitor area (2 bytes)
00002H
00000H
Note 2
Internal RAM
SFR area
Stack area for debugging (4 bytes) Note 3
Internal RAM area
Mirror area
▨ : Area used for on-chip debugging

Note 1. When real-time RAM monitor (RRM) function and dynamic memory modification (DMM) function are not used, it is 256 bytes.

Note 2. In debugging, reset vector is rewritten to address allocated to a monitor program.

Note 3. 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.

32.4 Notes on Usage

The RL78/G11 does not support the trace function of emulator.

CHAPTER 33 BCD CORRECTION CIRCUIT

33.1 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).

33.2 Registers Used by BCD Correction Circuit

The BCD correction circuit uses the following registers.

- BCD correction result register (BCDADJ)

33.2.1 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 33 - 1 Format of BCD correction result register (BCDADJ)

Address: F00FEH After reset: Undefined R

Symbol	7	6	5	4	3	2	1	0
BCDADJ								

33.3 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 34 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)**.

34.1 Conventions Used in Operation List

34.1.1 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 34 - 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 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 **Table 3 - 6 Extended Special Function Register (2nd SFR) List** for the symbols of the extended special function registers.

34.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

Table 34 - 2 Symbols in “Operation” Column

Symbol	Function
A	A register; 8-bit accumulator
X	X register
B	B register
C	C register
D	D register
E	E register
H	H register
L	L register
ES	ES register
CS	CS register
AX	AX register pair; 16-bit accumulator
BC	BC register pair
DE	DE register pair
HL	HL register pair
PC	Program counter
SP	Stack pointer
PSW	Program status word
CY	Carry flag
AC	Auxiliary carry flag
Z	Zero flag
RBS	Register bank select flag
IE	Interrupt request enable flag
()	Memory contents indicated by address or register contents in parentheses
X_H, X_L X_S, X_H, X_L	16-bit registers: X_H = higher 8 bits, X_L = lower 8 bits 20-bit registers: X_S = (bits 19 to 16), X_H = (bits 15 to 8), X_L = (bits 7 to 0)
∧	Logical product (AND)
∨	Logical sum (OR)
⊻	Exclusive logical sum (exclusive OR)
—	Inverted data
addr5	16-bit immediate data (even addresses only in 0080H to 00BFH)
addr16	16-bit immediate data
addr20	20-bit immediate data
jdisp8	Signed 8-bit data (displacement value)
jdisp16	Signed 16-bit data (displacement value)

34.1.3 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 34 - 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

34.1.4 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 34 - 4 Use Example of PREFIX Operation Code

<table>
<tr><th rowspan="2">Instruction</th><th colspan="5">Opcode</th></tr>
<tr><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th></tr>
<tr><td>MOV !addr16, #byte</td><td>CFH</td><td colspan="2">!addr16</td><td>#byte</td><td>—</td></tr>
<tr><td>MOV ES:!addr16, #byte</td><td>11H</td><td>CFH</td><td colspan="2">!addr16</td><td>#byte</td></tr>
<tr><td>MOV A, [HL]</td><td>8BH</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>MOV A, ES: [HL]</td><td>11H</td><td>8BH</td><td>—</td><td>—</td><td>—</td></tr>
</table>

Caution Set the ES register value with MOV ES, A, etc., before executing the PREFIX instruction.

34.2 Operation List

Table 34 - 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 (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 34 - 5 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 (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.

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 34 - 5 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 34 - 5 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 (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

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 34 - 5 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 34 - 5 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 34 - 5 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, r Note 3	2	1	—	A, CY ← A + r + CY	×	×	×
		r, A	2	1	—	r, CY ← r + A + CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16) + CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16) + CY	×	×	×
		A, saddr	2	1	—	A, CY ← A + (saddr) + CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A + (HL) + CY	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A + (ES, HL) + CY	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A + (HL + byte) + CY	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A + ((ES, HL) + byte) + CY	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A + (HL + B) + CY	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A + ((ES, HL) + B) + CY	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A + (HL + C) + CY	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A + ((ES, HL) + C) + CY	×	×	×
	SUB	A, #byte	2	1	—	A, CY ← A - byte	×	×	×
		saddr, #byte	3	2	—	(saddr), CY ← (saddr) - byte	×	×	×
		A, r Note 3	2	1	—	A, CY ← A - r	×	×	×
		r, A	2	1	—	r, CY ← r - A	×	×	×
		A, !addr16	3	1	4	A, CY ← A - (addr16)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A - (ES, addr16)	×	×	×
		A, saddr	2	1	—	A, CY ← A - (saddr)	×	×	×
		A, [HL]	1	1	4	A, CY ← A - (HL)	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A - (ES, HL)	×	×	×
		A, [HL+byte]	2	1	4	A, CY ← A - (HL + byte)	×	×	×
		A, ES:[HL+byte]	3	2	5	A,CY ← A - ((ES, HL) + byte)	×	×	×
		A, [HL+B]	2	1	4	A, CY ← A - (HL + B)	×	×	×
		A, ES:[HL+B]	3	2	5	A,CY ← A - ((ES, HL) + B)	×	×	×
		A, [HL+C]	2	1	4	A, CY ← A - (HL + C)	×	×	×
		A, ES:[HL+C]	3	2	5	A,CY ← A - ((ES, HL) + C)	×	×	×

Note 1. 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 34 - 5 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 34 - 5 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 34 - 5 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 (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 34 - 5 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 34 - 5 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 34 - 5 Operation List (13/18)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Increment/ decrement	INC	r	1	1	—	r ← r + 1	×	×	
		!addr16	3	2	—	(addr16) ← (addr16) + 1	×	×	
		ES:!addr16	4	3	—	(ES, addr16) ← (ES, addr16) + 1	×	×	
		saddr	2	2	—	(saddr) ← (saddr) + 1	×	×	
		[HL+byte]	3	2	—	(HL + byte) ← (HL + byte) + 1	×	×	
		ES: [HL+byte]	4	3	—	((ES:HL) + byte) ← ((ES:HL) + byte) + 1	×	×	
	DEC	r	1	1	—	r ← r - 1	×	×	
		!addr16	3	2	—	(addr16) ← (addr16) - 1	×	×	
		ES:!addr16	4	3	—	(ES, addr16) ← (ES, addr16) - 1	×	×	
		saddr	2	2	—	(saddr) ← (saddr) - 1	×	×	
		[HL+byte]	3	2	—	(HL + byte) ← (HL + byte) - 1	×	×	
		ES: [HL+byte]	4	3	—	((ES:HL) + byte) ← ((ES:HL) + byte) - 1	×	×	
	INCW	rp	1	1	—	rp ← rp + 1			
		!addr16	3	2	—	(addr16) ← (addr16) + 1			
		ES:!addr16	4	3	—	(ES, addr16) ← (ES, addr16) + 1			
		saddrp	2	2	—	(saddrp) ← (saddrp) + 1			
		[HL+byte]	3	2	—	(HL + byte) ← (HL + byte) + 1			
		ES: [HL+byte]	4	3	—	((ES:HL) + byte) ← ((ES:HL) + byte) + 1			
	DECW	rp	1	1	—	rp ← rp - 1			
		!addr16	3	2	—	(addr16) ← (addr16) - 1			
		ES:!addr16	4	3	—	(ES, addr16) ← (ES, addr16) - 1			
		saddrp	2	2	—	(saddrp) ← (saddrp) - 1			
		[HL+byte]	3	2	—	(HL + byte) ← (HL + byte) - 1			
		ES: [HL+byte]	4	3	—	((ES:HL) + byte) ← ((ES:HL) + byte) - 1			
Shift	SHR	A, cnt	2	1	—	(CY ← A_0, A_{m-1} ← A_m, A_7 ← 0) × cnt			×
	SHRW	AX, cnt	2	1	—	(CY ← AX_0, AX_{m-1} ← AX_m, AX_{15} ← 0) × cnt			×
	SHL	A, cnt	2	1	—	(CY ← A_7, A_m ← A_{m-1}, A_0 ← 0) × cnt			×
		B, cnt	2	1	—	(CY ← B_7, B_m ← B_{m-1}, B_0 ← 0) × cnt			×
		C, cnt	2	1	—	(CY ← C_7, C_m ← C_{m-1}, C_0 ← 0) × cnt			×
	SHLW	AX, cnt	2	1	—	(CY ← AX_{15}, AX_m ← AX_{m-1}, AX_0 ← 0) × cnt			×
		BC, cnt	2	1	—	(CY ← BC_{15}, BC_m ← BC_{m-1}, BC_0 ← 0) × cnt			×
	SAR	A, cnt	2	1	—	(CY ← A_0, A_{m-1} ← A_m, A_7 ← A_7) × cnt			×
	SARW	AX, cnt	2	1	—	(CY ← AX_0, AX_{m-1} ← AX_m, AX_{15} ← AX_{15}) × cnt			×

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 34 - 5 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	—	$(CY, A_7 \leftarrow A_0, A_{m-1} \leftarrow A_m) \times 1$			×
	ROL	A, 1	2	1	—	$(CY, A_0 \leftarrow A_7, A_{m+1} \leftarrow A_m) \times 1$			×
	RORC	A, 1	2	1	—	$(CY \leftarrow A_0, A_7 \leftarrow CY, A_{m-1} \leftarrow A_m) \times 1$			×
	ROLC	A, 1	2	1	—	$(CY \leftarrow A_7, A_0 \leftarrow CY, A_{m+1} \leftarrow A_m) \times 1$			×
	ROLWC	AX,1	2	1	—	$(CY \leftarrow AX_{15}, AX_0 \leftarrow CY, AX_{m+1} \leftarrow AX_m) \times 1$			×
		BC,1	2	1	—	$(CY \leftarrow BC_{15}, BC_0 \leftarrow CY, BC_{m+1} \leftarrow BC_m) \times 1$			×
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 34 - 5 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 ← $\overline{\text{CY}}$			×

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 34 - 5 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	—	$(SP - 2) \leftarrow (PC + 2)_S$, $(SP - 3) \leftarrow (PC + 2)_H$, $(SP - 4) \leftarrow (PC + 2)_L$, $PC \leftarrow CS, rp$, $SP \leftarrow SP - 4$			
		$!addr20	3	3	—	$(SP - 2) \leftarrow (PC + 3)_S$, $(SP - 3) \leftarrow (PC + 3)_H$, $(SP - 4) \leftarrow (PC + 3)_L$, $PC \leftarrow PC + 3 + jdisp16$, $SP \leftarrow SP - 4$			
		!addr16	3	3	—	$(SP - 2) \leftarrow (PC + 3)_S$, $(SP - 3) \leftarrow (PC + 3)_H$, $(SP - 4) \leftarrow (PC + 3)_L$, $PC \leftarrow 0000, addr16$, $SP \leftarrow SP - 4$			
		!!addr20	4	3	—	$(SP - 2) \leftarrow (PC + 4)_S$, $(SP - 3) \leftarrow (PC + 4)_H$, $(SP - 4) \leftarrow (PC + 4)_L$, $PC \leftarrow addr20$, $SP \leftarrow SP - 4$			
	CALLT	[addr5]	2	5	—	$(SP - 2) \leftarrow (PC + 2)_S$, $(SP - 3) \leftarrow (PC + 2)_H$, $(SP - 4) \leftarrow (PC + 2)_L$, $PC_S \leftarrow 0000$, $PC_H \leftarrow (0000, addr5 + 1)$, $PC_L \leftarrow (0000, addr5)$, $SP \leftarrow SP - 4$			
	BRK	—	2	5	—	$(SP - 1) \leftarrow PSW$, $(SP - 2) \leftarrow (PC + 2)_S$, $(SP - 3) \leftarrow (PC + 2)_H$, $(SP - 4) \leftarrow (PC + 2)_L$, $PC_S \leftarrow 0000$, $PC_H \leftarrow (0007FH)$, $PC_L \leftarrow (0007EH)$, $SP \leftarrow SP - 4$, $IE \leftarrow 0$			
	RET	—	1	6	—	$PC_L \leftarrow (SP)$, $PC_H \leftarrow (SP + 1)$, $PC_S \leftarrow (SP + 2)$, $SP \leftarrow SP + 4$			
	RETI	—	2	6	—	$PC_L \leftarrow (SP)$, $PC_H \leftarrow (SP + 1)$, $PC_S \leftarrow (SP + 2)$, $PSW \leftarrow (SP + 3)$, $SP \leftarrow SP + 4$	R	R	R
	RETB	—	2	6	—	$PC_L \leftarrow (SP)$, $PC_H \leftarrow (SP + 1)$, $PC_S \leftarrow (SP + 2)$, $PSW \leftarrow (SP + 3)$, $SP \leftarrow SP + 4$	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 34 - 5 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 34 - 5 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 (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”.

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 35 ELECTRICAL SPECIFICATIONS (T_A = -40 to +85°C)

This chapter describes the following electrical specifications.

Target products A: Consumer applications (T_A = −40 to +85°C)
R5F105xxAxx
G: When the products “G: Industrial applications (T_A = −40 to +105°C)" is used in the range of T_A = -40 to +85°C
R5F105xxGxx

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. The EV_{DD} pin is not present on products with 24 or less pins. Accordingly, replace EV_{DD} with V_{DD} and the voltage condition 1.6 ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V with 1.6 ≤ V_{DD} ≤ 5.5 V.

35.1 Absolute Maximum Ratings

(1/2)

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}		-0.5 to +6.5	V
	EV_{DD}		-0.5 to +6.5	V
	AV_{REFP}		0.3 to V_{DD} + 0.3 Note 2	V
	AV_{REFM}		-0.3 to V_{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, P01, P30 to P33, P40, and P51 to P56	-0.3 to EV_{DD} + 0.3 and -0.3 to V_{DD} + 0.3 Note 2	V
	V_{I2}	P20 to P23, P121, P122, P125, P137, EXCLK, $\overline{\text{RESET}}$	-0.3 to V_{DD} + 0.3 Note 2	V
Output voltage	V_{O1}	P00, P01, P30 to P33, P40, and P51 to P56	-0.3 to EV_{DD} + 0.3 and -0.3 to V_{DD} + 0.3 Note 2	V
	V_{O2}	P20 to P23	-0.3 to V_{DD} + 0.3 Note 2	V
Analog input voltage	V_{AI1}	ANI16 to ANI22	-0.3 to EV_{DD} + 0.3 and -0.3 to $AV_{REF}(+)$ + 0.3 Notes 2, 3	V
	V_{AI2}	ANI0 to ANI3	-0.3 to V_{DD} + 0.3 and -0.3 to $AV_{REF}(+)$ + 0.3 Notes 2, 3	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 6.5 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

(2/2)

Parameter	Symbols	Conditions		Ratings	Unit
Output current, high	I_{OH1}	Per pin		-40	mA
		Total of all pins -170 mA	P00, P01, P40	-70	mA
			P30 to P33, P51 to P56	-100	mA
	I_{OH2}	Per pin	P20 to P23	-0.5	mA
		Total of all pins		-2	mA
Output current, low	I_{OL1}	Per pin		40	mA
		Total of all pins 170 mA	P00, P01, P40	70	mA
			P30 to P33, P51 to P56	100	mA
	I_{OL2}	Per pin	P20 to P23	1	mA
		Total of all pins		4	mA
Operating ambient temperature	T_A	In normal operation mode		-40 to +85	°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.

35.2 Oscillator Characteristics

35.2.1 X1 characteristics

(T_A = -40 to +85°C, 1.6 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Resonator	Resonator	Conditions	MIN.	TYP.	MAX.	Unit
X1 clock oscillation frequency (fx) [Note]	Ceramic resonator/ crystal resonator	2.7 V ≤ V_{DD} ≤ 5.5 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	

Note Indicates only permissible oscillator frequency ranges. Refer to **35.4 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, refer to **6.4 System Clock Oscillator**.

35.2.2 On-chip oscillator characteristics

(T_A = -40 to +85°C, 1.6 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Oscillators	Parameters	Conditions		MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator clock frequency [Notes 1, 2]	f_{IH}	2.7 V ≤ V_{DD} ≤ 5.5 V		1		24	MHz
		2.4 V ≤ V_{DD} ≤ 5.5 V		1		16	
		1.8 V ≤ V_{DD} ≤ 5.5 V		1		8	
		1.6 V ≤ V_{DD} ≤ 5.5 V		1		4	
High-speed on-chip oscillator clock frequency accuracy		T_A = -20 to +85°C	1.8 V ≤ V_{DD} ≤ 5.5 V	-1		1	%
			1.6 V ≤ V_{DD} < 1.8 V	-5		5	
		T_A = -40 to -20°C	1.8 V ≤ V_{DD} ≤ 5.5 V	-1.5		1.5	%
			1.6 V ≤ V_{DD} < 1.8 V	-5.5		5.5	
Middle-speed on-chip oscillator oscillation frequency [Note 2]	f_{IM}			1		4	MHz
Middle-speed on-chip oscillator oscillation frequency accuracy				-12		+12	%
Temperature drift of Middle-speed on-chip oscillator oscillation frequency accuracy	D_{IMT}				0.008		%/°C
Voltage drift of Middle-speed on-chip oscillator oscillation frequency accuracy	D_{IMV}	T_A = 25°C	2.1 V ≤ V_{DD} ≤ 5.5 V		0.02		%/V
			2.0 V ≤ V_{DD} < 2.1 V		-12		
			1.6 V ≤ V_{DD} < 2.0 V		10		
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 **35.4 AC Characteristics** for instruction execution time.

35.3 DC Characteristics

35.3.1 Pin characteristics

(T_A = -40 to +85°C, 1.6 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(1/5)**

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high Note 1	I_{OH1}	Per pin for P00, P01, P30 to P33, P40, and P51 to P56				-10.0 Note 2	mA
		Total of P00, P01, and P40 (When duty ≤ 70% Note 3)	4.0 V ≤ EV_{DD} ≤ 5.5 V			-42.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			-10.0	mA
			1.8 V ≤ EV_{DD} < 2.7 V			-5.0	mA
			1.6 V ≤ EV_{DD} < 1.8 V			-2.5	mA
		Total of P30 to P33, and P51 to P56 (When duty ≤ 70% Note 3)	4.0 V ≤ EV_{DD} ≤ 5.5 V			-80.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			-19.0	mA
			1.8 V ≤ EV_{DD} < 2.7 V			-10.0	mA
			1.6 V ≤ EV_{DD} < 1.8 V			-5.0	mA
		Total of all pins (When duty ≤ 70% Note 3)				-122.0	mA
	I_{OH2}	Per pin for P20 to P23				-0.1 Note 2	mA
		Total of all pins (When duty ≤ 70% Note 3)	1.6 V ≤ V_{DD} ≤ 5.5 V			-0.4	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 **P00, P01, P20, P30 to P33, P40 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 ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(2/5)**

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low Note 1	I_{OL1}	Per pin for P00, P01, P30 to P33, P40, and P51 to P56				20.0 Note 2	mA
		Total of P00, P01, and P40 (When duty ≤ 70% Note 3)	4.0 V ≤ EV_{DD} ≤ 5.5 V			70.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			15.0	mA
			1.8 V ≤ EV_{DD} < 2.7 V			9.0	mA
			1.6 V ≤ EV_{DD} < 1.8 V			4.5	mA
		Total of P30 to P33, and P51 to P56 (When duty ≤ 70% Note 3)	4.0 V ≤ EV_{DD} ≤ 5.5 V			80.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			35.0	mA
			1.8 V ≤ EV_{DD} < 2.7 V			20.0	mA
			1.6 V ≤ EV_{DD} < 1.8 V			10.0	mA
		Total of all pins (When duty ≤ 70% Note 3)				150.0	mA
	I_{OL2}	Per pin for P20 to P23				0.4 Note 2	mA
		Total of all pins (When duty ≤ 70% Note 3)	1.6 V ≤ V_{DD} ≤ 5.5 V			1.6	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 ≤ EV$_{DD}$ ≤ V$_{DD}$ ≤ 5.5 V, V$_{SS}$ = 0 V) **(3/5)**

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P00, P01, P30 to P33, P40, and P51 to P56	Normal mode	0.8 EV$_{DD}$		EV$_{DD}$	V
	V_{IH2}	P00, P30 to P32, P40, P51 to P56	TTL mode 4.0 V ≤ EV$_{DD}$ ≤ 5.5 V	2.2		EV$_{DD}$	V
			TTL mode 3.3 V ≤ EV$_{DD}$ < 4.0 V	2.0		EV$_{DD}$	V
			TTL mode 1.6 V ≤ EV$_{DD}$ < 3.3 V	1.5		EV$_{DD}$	V
	V_{IH3}	P20 to P23 (digital input)		0.7 V$_{DD}$		V$_{DD}$	V
	V_{IH4}	P121, P122, P125, P137, EXCLK, $\overline{\text{RESET}}$		0.8 V$_{DD}$		V$_{DD}$	V
Input voltage, low	V_{IL1}	P00, P01, P30 to P33, P40, and P51 to P56	Normal mode	0		0.2 EV$_{DD}$	V
	V_{IL2}	P00, P30 to P32, P40, P51 to P56	TTL mode 4.0 V ≤ EV$_{DD}$ ≤ 5.5 V	0		0.8	V
			TTL mode 3.3 V ≤ EV$_{DD}$ < 4.0 V	0		0.5	V
			TTL mode 1.6 V ≤ EV$_{DD}$ < 3.3 V	0		0.32	V
	V_{IH3}	P20 to P23 (digital input)		0		0.3 V$_{DD}$	V
	V_{IH4}	P121, P122, P125, P137, EXCLK, $\overline{\text{RESET}}$		0		0.2 V$_{DD}$	V

Caution The maximum value of V$_{IH}$ of pins P00, P01, P20, P30 to P33, P40 and P51 to P56 is V$_{DD}$ or EV$_{DD}$, even in the N-ch open-drain mode.
(P20: V$_{DD}$
P00, P01, P30 to P33, P40, P51 to P56: EV$_{DD}$)

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 ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(4/5)**

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, high	V_{OH1}	P00, P01, P30 to P33, P40, and P51 to P56	4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OH} = -10.0 mA	EV_{DD} - 1.5			V
			4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OH} = -3.0 mA	EV_{DD} - 0.7			V
			2.7 V ≤ EV_{DD} ≤ 5.5 V, I_{OH} = -2.0 mA	EV_{DD} - 0.6			V
			1.8 V ≤ EV_{DD} ≤ 5.5 V I_{OH} = -1.5 mA	EV_{DD} - 0.5			V
			1.6 V ≤ EV_{DD} ≤ 5.5 V, I_{OH} = -1.0 mA	EV_{DD} - 0.5			V
	V_{OH2}	P20 to P23	1.6 V ≤ V_{DD} ≤ 5.5 V, I_{OH} = -100 μA	V_{DD} - 0.5			V
Output voltage, low	V_{OL1}	P00, P01, P30 to P33, P40, and P51 to P56	4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OL} = 20.0 mA			1.3	V
			4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OL} = 8.5 mA			0.7	V
			2.7 V ≤ EV_{DD} ≤ 5.5 V, I_{OL} = 3.0 mA			0.6	V
			2.7 V ≤ EV_{DD} ≤ 5.5 V, I_{OL} = 1.5 mA			0.4	V
			1.8 V ≤ EV_{DD} ≤ 5.5 V, I_{OL} = 0.6 mA			0.4	V
			1.6 V ≤ EV_{DD} ≤ 5.5 V, I_{OL} = 0.3 mA			0.4	V
	V_{OL2}	P20 to P23	1.6 V ≤ V_{DD} ≤ 5.5 V, I_{OL} = 400 μA			0.4	V

Caution **P00, P01, P20, P30 to P33, P40 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 ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(5/5)**

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P00, P01, P30 to P33, P40, and P51 to P56	$V_I = EV_{DD}$				1	μA
	I_{LIH2}	P20 to P23, P125, P137, $\overline{\text{RESET}}$	$V_I = V_{DD}$				1	μA
	I_{LIH3}	P121, P122, X1, X2, EXCLK	$V_I = V_{DD}$	In input port or external clock input			1	μA
				In resonator connection			10	μA
Input leakage current, low	I_{LIL1}	P00, P01, P30 to P33, P40, and P51 to P56	$V_I = V_{SS}$				-1	μA
	I_{LIL2}	P20 to P23, P125, P137, $\overline{\text{RESET}}$	$V_I = V_{SS}$				-1	μA
	I_{LIL3}	P121, P122, X1, X2, EXCLK	$V_I = V_{SS}$	In input port or external clock input			-1	μA
				In resonator connection			-10	μA
On-chip pull-up resistance	R_U	P00, P01, P30 to P33, P40, P51 to P56, P125	$V_I = V_{SS}$, 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.

35.3.2 Supply current characteristics

(T_A = -40 to +85°C, 1.6 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(1/4)**

Parameter	Symbol	Conditions						MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD1}	Operating mode	Basic operation	HS (high-speed main) mode	f_{HOCO} = 48 MHz Note 3 f_{IH} = 24 MHz Note 3	V_{DD} = 5.0 V			1.7		mA
						V_{DD} = 3.0 V			1.7		
					f_{HOCO} = 24 MHz Note 3 f_{IH} = 24 MHz Note 3	V_{DD} = 5.0 V			1.4		
						V_{DD} = 3.0 V			1.4		
			Normal operation	HS (high-speed main) mode	f_{HOCO} = 48 MHz Note 3 f_{IH} = 24 MHz Note 3	V_{DD} = 5.0 V			3.5	6.9	mA
						V_{DD} = 3.0 V			3.5	6.9	
					f_{HOCO} = 24 MHz Note 3 f_{IH} = 24 MHz Note 3	V_{DD} = 5.0 V			3.2	6.3	
						V_{DD} = 3.0 V			3.2	6.3	
					f_{HOCO} = 16 MHz Note 3 f_{IH} = 16 MHz Note 3	V_{DD} = 5.0 V			2.4	4.6	
						V_{DD} = 3.0 V			2.4	4.6	
			Normal operation	LS (low-speed main) mode (MCSEL = 0)	f_{IH} = 8 MHz Note 3	V_{DD} = 3.0 V			1.1	2.0	mA
						V_{DD} = 2.0 V			1.1	2.0	
			Normal operation	LS (low-speed main) mode (MCSEL = 1)	f_{IH} = 4 MHz Note 3	V_{DD} = 3.0 V			0.72	1.3	mA
						V_{DD} = 2.0 V			0.72	1.3	
					f_{IM} = 4 MHz Note 6	V_{DD} = 3.0 V			0.58	1.1	
						V_{DD} = 2.0 V			0.58	1.1	
			Normal operation	LV (low-voltage main) mode	f_{IH} = 4 MHz Note 3	V_{DD} = 3.0 V			1.2	1.8	mA
						V_{DD} = 2.0 V			1.2	1.8	
			Normal operation	LP (low-power main) mode (MCSEL = 1)	f_{IH} = 1 MHz Note 3	V_{DD} = 3.0 V			290	480	μA
						V_{DD} = 2.0 V			290	480	
					f_{IM} = 1 MHz Note 6	V_{DD} = 3.0 V			124	230	
						V_{DD} = 2.0 V			124	230	
			Normal operation	HS (high-speed main) mode	f_{MX} = 20 MHz Note 2	V_{DD} = 5.0 V	Square wave input		2.7	5.3	mA
							Resonator connection		2.8	5.5	
						V_{DD} = 3.0 V	Square wave input		2.7	5.3	
							Resonator connection		2.8	5.5	
					f_{MX} = 10 MHz Note 2	V_{DD} = 5.0 V	Square wave input		1.8	3.1	
							Resonator connection		1.9	3.2	
						V_{DD} = 3.0 V	Square wave input		1.8	3.1	
							Resonator connection		1.9	3.2	
			Normal operation	LS (low-speed main) mode (MCSEL = 0)	f_{MX} = 8 MHz Note 2	V_{DD} = 3.0 V	Square wave input		0.9	1.9	mA
							Resonator connection		1.0	2.0	
			Normal operation		f_{MX} = 8 MHz Note 2	V_{DD} = 2.0 V	Square wave input		0.9	1.9	
							Resonator connection		1.0	2.0	
			Normal operation	LS (low-speed main) mode (MCSEL = 1)	f_{MX} = 4 MHz Note 2	V_{DD} = 3.0 V	Square wave input		0.6	1.1	mA
							Resonator connection		0.6	1.2	
			Normal operation		f_{MX} = 4 MHz Note 2	V_{DD} = 2.0 V	Square wave input		0.6	1.1	
							Resonator connection		0.6	1.2	
			Normal operation	LP (low-power main) mode (MCSEL = 1)	f_{MX} = 1 MHz Note 2	V_{DD} = 3.0 V	Square wave input		100	190	μA
							Resonator connection		145	250	
			Normal operation		f_{MX} = 1 MHz Note 2	V_{DD} = 2.0 V	Square wave input		100	190	
							Resonator connection		145	250	

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

(T_A = -40 to +85°C, 1.6 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(2/4)**

Parameter	Symbol	Conditions						MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD1}	Operating mode	Normal operation	Subsystem clock operation	f_{IL} = 15 kHz, T_A = -40°C Note 5	Normal operation			1.8	5.9	μA
					f_{IL} = 15 kHz, T_A = +25°C Note 5	Normal operation			1.9	5.9	
					f_{IL} = 15 kHz, T_A = +85°C Note 5	Normal operation			2.3	8.7	

Note 1. Total current flowing into V_{DD} and EV_{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, D/A converter, comparator, programmable gain amplifier, 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.

Note 2. When the high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock and low-speed on-chip oscillator clock are stopped.

Note 3. When the high-speed system clock, middle-speed on-chip oscillator clock and low-speed on-chip oscillator clock are stopped.

Note 4. When the high-speed system clock is stopped.

Note 5. When the high-speed system clock, high-speed on-chip oscillator clock and middle-speed on-chip oscillator clock are stopped.

Note 6. When the high-speed system clock, high-speed on-chip oscillator clock and low-speed on-chip oscillator 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_{SUB}: Subsystem clock frequency (Low-speed on-chip oscillator clock frequency)

Remark 6. Except subsystem clock operation, temperature condition of the TYP. value is T_A = 25°C

(T_A = -40 to +85°C, 1.6 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{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_{HOCO} = 48 MHz Note 4	V_{DD} = 5.0 V			0.59	2.43	mA
				f_{IH} = 24 MHz Note 4	V_{DD} = 3.0 V			0.59	2.43	
				f_{HOCO} = 24 MHz Note 4	V_{DD} = 5.0 V			0.41	1.83	
				f_{IH} = 24 MHz Note 4	V_{DD} = 3.0 V			0.41	1.83	
				f_{HOCO} = 16 MHz Note 4	V_{DD} = 5.0 V			0.39	1.38	
				f_{IH} = 16 MHz Note 4,	V_{DD} = 3.0 V			0.39	1.38	
			LS (low-speed main) mode (MCSEL = 0)	f_{IH} = 8 MHz Note 4	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	V_{DD} = 3.0 V			204	400	µA
					V_{DD} = 2.0 V			204	400	
				f_{IM} = 4 MHz Note 6	V_{DD} = 3.0 V			43	250	
					V_{DD} = 2.0 V			43	250	
			LV (low-voltage main) mode	f_{IH} = 4 MHz Note 4	V_{DD} = 3.0 V			450	700	µA
					V_{DD} = 2.0 V			450	700	
			LP (low-power main) mode (MCSEL = 1)	f_{IH} = 1 MHz Note 4	V_{DD} = 3.0 V			192	400	µA
					V_{DD} = 2.0 V			192	400	
				f_{IM} = 1 MHz Note 6	V_{DD} = 3.0 V			28	100	
					V_{DD} = 2.0 V			28	100	
			HS (high-speed main) mode	f_{MX} = 20 MHz Note 3	V_{DD} = 5.0 V	Square wave input		0.20	1.55	mA
						Resonator connection		0.40	1.74	
					V_{DD} = 3.0 V	Square wave input		0.20	1.55	
						Resonator connection		0.40	1.74	
				f_{MX} = 10 MHz Note 3	V_{DD} = 5.0 V	Square wave input		0.15	0.86	
						Resonator connection		0.30	0.93	
					V_{DD} = 3.0 V	Square wave input		0.15	0.86	
						Resonator connection		0.30	0.93	
			LS (low-speed main) mode (MCSEL = 0)	f_{MX} = 8 MHz Note 3	V_{DD} = 3.0 V	Square wave input		68	550	µA
						Resonator connection		125	590	
				f_{MX} = 8 MHz Note 3	V_{DD} = 2.0 V	Square wave input		68	550	
						Resonator connection		125	590	
			LS (low-speed main) mode (MCSEL = 1)	f_{MX} = 4 MHz Note 3	V_{DD} = 3.0 V	Square wave input		23	128	µA
						Resonator connection		65	200	
				f_{MX} = 1 MHz Note 3	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	V_{DD} = 3.0 V	Square wave input		10	64	µA
						Resonator connection		59	150	
				f_{MX} = 1 MHz Note 3	V_{DD} = 2.0 V	Square wave input		10	64	
						Resonator connection		59	150	
			Subsystem clock operation	f_{IL} = 15 kHz, T_A = -40°C Note 5				0.48	1.22	µA
				f_{IL} = 15 kHz, T_A = +25°C Note 5				0.55	1.22	
				f_{IL} = 15 kHz, T_A = +85°C Note 5				0.80	3.30	

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

Note 1. Total current flowing into V_{DD} and EV_{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, D/A converter, comparator, programmable gain amplifier, 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.

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, and low-speed on-chip oscillator clock are stopped.

Note 4. When the high-speed system clock, middle-speed on-chip oscillator clock and low-speed on-chip oscillator clock are stopped.

Note 5. When the high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock and high-speed system clock are stopped.

Note 6. When the high-speed system clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator 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_{SUB}: Subsystem clock frequency (Low-speed on-chip oscillator clock frequency)

Remark 6. Except subsystem clock operation, temperature condition of the TYP. value is T_A = 25°C

(T_A = -40 to +85°C, 1.6 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(4/4)**

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD3}	STOP mode Note 2	T_A = -40°C		0.19	0.51	µA
			T_A = +25°C		0.25	0.51	
			T_A = +50°C		0.28	1.10	
			T_A = +70°C		0.38	1.90	
			T_A = +85°C		0.60	3.30	

Note 1. Total current flowing into V_{DD} and EV_{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.

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 ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Low-speed on-chip oscillator operating current	I_{FIL} Note 1				0.22		μA
12-bit interval timer operating current	I_{TMKA} Notes 1, 3, 4	f_{IL} = 15 kHz f_{MAIN} stopped (per unit)			0.02		μA
8-bit interval timer operating current Notes 1, 9	I_{TMT}	f_{IL} = 15 kHz f_{MAIN} stopped (per unit)	8-bit counter mode × 2-channel operation		0.04		μA
			16-bit counter mode operation		0.03		μA
Watchdog timer operating current	I_{WDT} Notes 1, 3, 5	f_{IL} = 15 kHz f_{MAIN} stopped (per unit)			0.22		μA
A/D converter operating current	I_{ADC} Notes 1, 6	During maximum-speed conversion	Normal mode, AV_{VREFP} = V_{DD} = 5.0 V		1.3	1.7	mA
			Low voltage mode, AV_{VREFP} = V_{DD} = 3.0 V		0.5	0.7	mA
Internal reference voltage (1.45 V) current Notes 1, 10	I_{ADREF}				85.0		μA
Temperature sensor operating current	I_{TMPS} Note 1				85.0		μA
D/A converter operating current	I_{DAC} Notes 1, 14	Per channel				1.5	mA
PGA operating current	I_{PGA} Notes 1, 2				480	700	μA
Comparator operating current	I_{CMP} Note 8	V_{DD} = 5.0 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		1.9		
		V_{DD} = 5.0 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		
LVD operating current	I_{LVD} Notes 1, 7				0.10		μA
Self-programming operating current	IFSP Notes 1, 12				2.0	12.20	mA
BGO current	IBGO Notes 1, 11				2.0	12.20	mA
SNOOZE operating current	ISNOZ Note 1	ADC operation f_{IH} = 24 MHz, AVREFP = V_{DD} = 3.0 V	Mode transition Note 13		0.50	0.60	mA
			The A/D conversion operations are performed		1.20	1.44	mA
		Simplified SPI (CSI)/UART operation f_{IH} = 24 MHz			0.70	0.84	mA
	ISNOZM Note 1	ADC operation f_{IM} = 4 MHz, AVREFP = V_{DD} = 3.0 V	Mode transition Note 13		0.05	0.08	mA
			The A/D conversion operations are performed		0.67	0.78	mA
		Simplified SPI (CSI) operation, f_{IM} = 4 MHz			0.06	0.08	mA

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

Note 1. Current flowing to V_{DD}.

Note 2. Operable range is 2.7 to 5.5 V.

Note 3. When the high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock, and high-speed system clock are stopped.

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 consumed by generating the internal reference voltage (1.45 V).

Note 11. Current flowing during programming of the data flash.

Note 12. Current flowing during self-programming.

Note 13. For transition time to the SNOOZE mode, see **24.3.3 SNOOZE mode**.

Note 14. Current flowing only to the D/A converter. The supply current of the RL78 microcontrollers is the sum of I_{DD1} or I_{DD2} and I_{DAC} when the D/A converter operates in an operation mode or the HALT mode.

Remark 1. f_{IL}: Low-speed on-chip oscillator clock frequency

Remark 2. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 3. Temperature condition of the TYP. value is T_A = 25°C

35.4 AC Characteristics

(T_A = -40 to +85°C, 1.6 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{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} ≤ 5.5 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} ≤ 5.5 V PMMC. MCSEL = 0	0.125		1	μs
				1.8 V ≤ V_{DD} ≤ 5.5 V PMMC. MCSEL = 1	0.25		1	
			LP (low-power main) mode	1.8 V ≤ V_{DD} ≤ 5.5 V	1			μs
			LV (low-voltage main) mode	1.6 V ≤ V_{DD} ≤ 5.5 V	0.25		1	μs
		Subsystem clock (f_{SUB}) operation	f_{IL}	1.8 V ≤ V_{DD} ≤ 5.5 V		66.7		μs
		In the self-programming mode	HS (high-speed main) mode	2.7 V ≤ V_{DD} ≤ 5.5 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} ≤ 5.5 V	0.125		1	μs
			LV (low-voltage main) mode	1.8 V ≤ V_{DD} ≤ 5.5 V	0.25		1	μs
External system clock frequency	f_{EX}	2.7 V ≤ V_{DD} ≤ 5.5 V			1		20	MHz
		2.4 V ≤ V_{DD} < 2.7 V			1		16	MHz
		1.8 V ≤ V_{DD} < 2.4 V			1		8	MHz
		1.6 V ≤ V_{DD} < 1.8 V			1		4	MHz
External system clock input high-/low-level width	t_{EXH}, t_{EXL}	2.7 V ≤ V_{DD} ≤ 5.5 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
TI00 to TI03 input high-/low-level width	t_{TIH}, t_{TIL}[Note]				1/f_{MCK} + 10			ns

Note Following conditions must be satisfied on low level interface of EV_{DD} < V_{DD}.
1.8 V ≤ EV_{DD} ≤ 2.7 V: MIN. 125 ns
1.6 V ≤ EV_{DD} < 1.8 V: MIN. 250 ns

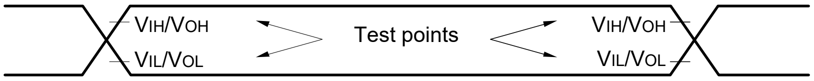
Remark f_{MCK}: Timer array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of timer mode register mn (TMRmn). m: Unit number (m = 0), n: Channel number (n = 0 to 3))

(T_A = -40 to +85°C, 1.6 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(2/2)**

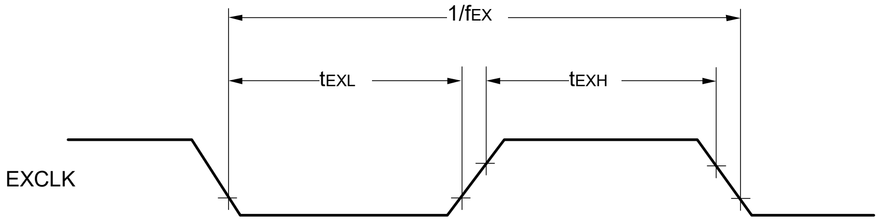
Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
TO00 to TO03, TKBO0, and TKBO1 output frequency Note	f_{TO}	TO00 to TO03, TKBO0, and TKBO1 (in the case of output from port pins other than P20)	HS (high-speed main) mode	4.0 V ≤ EV_{DD} ≤ 5.5 V			12	MHz
				2.7 V ≤ EV_{DD} < 4.0 V			8	
				1.8 V ≤ EV_{DD} < 2.7 V			4	
				1.6 V ≤ EV_{DD} < 1.8 V			2	
			LS (low-speed main) mode	1.8 V ≤ EV_{DD} ≤ 5.5 V			4	
				1.6 V ≤ EV_{DD} < 1.8 V			2	
			LP (low-power main) mode	1.8 V ≤ EV_{DD} ≤ 5.5 V			0.5	
			LV (low-voltage main) mode	1.6 V ≤ EV_{DD} ≤ 5.5 V			2	
		TKBO1 (in the case of output from P20)	HS (high-speed main) mode	4.0 V ≤ V_{DD} ≤ 5.5 V			1.5	MHz
				2.7 V ≤ V_{DD} < 4.0 V			1.2	
				2.4 V ≤ V_{DD} < 2.7 V			1	
			LS (low-speed main) mode	4.0 V ≤ V_{DD} ≤ 5.5 V			1.5	
				2.7 V ≤ V_{DD} < 4.0 V			1.2	
				2.4 V ≤ V_{DD} < 2.7 V			1	
				1.8 V ≤ V_{DD} < 2.4 V			0.75	
			LP (low-power main) mode	1.8 V ≤ V_{DD} ≤ 5.5 V			0.5	
			LV (low-voltage main) mode	4.0 V ≤ V_{DD} ≤ 5.5 V			1.5	
				2.7 V ≤ V_{DD} < 4.0 V			1.2	
				2.4 V ≤ V_{DD} < 2.7 V			1	
				1.8 V ≤ V_{DD} < 2.4 V			0.75	
				1.6 V ≤ V_{DD} < 1.8 V			0.5	
PCLBUZ0, PCLBUZ1 output frequency	f_{PCL}	HS (high-speed main) mode		4.0 V ≤ EV_{DD} ≤ 5.5 V			16	MHz
				2.7 V ≤ EV_{DD} < 4.0 V			8	
				1.8 V ≤ EV_{DD} < 2.7 V			4	
				1.6 V ≤ EV_{DD} < 1.8 V			2	
		LS (low-speed main) mode		1.8 V ≤ EV_{DD} ≤ 5.5 V			4	
				1.6 V ≤ EV_{DD} < 1.8 V			2	
		LP (low-power main) mode		1.6 V ≤ EV_{DD} ≤ 5.5 V			1	
		LV (low-voltage main) mode		1.8 V ≤ EV_{DD} ≤ 5.5 V			4	
				1.6 V ≤ EV_{DD} < 1.8 V			2	
Interrupt input high-/low-level width	t_{INTH}, t_{INTL}	INTP0 to INTP2, INTP9		1.6 V ≤ V_{DD} ≤ 5.5 V	1			μs
		INTP3 to INTP8, INTP10, INTP11		1.6 V ≤ EV_{DD} ≤ 5.5 V	1			
Key interrupt input low-level width	t_{KR}	KR0 to KR7		1.8 V ≤ EV_{DD} ≤ 5.5 V	250			ns
				1.6 V ≤ EV_{DD} < 1.8 V	1			μs
$\overline{\text{RESET}}$ low-level width	t_{RSL}				10			μs

Note When duty is 50%.

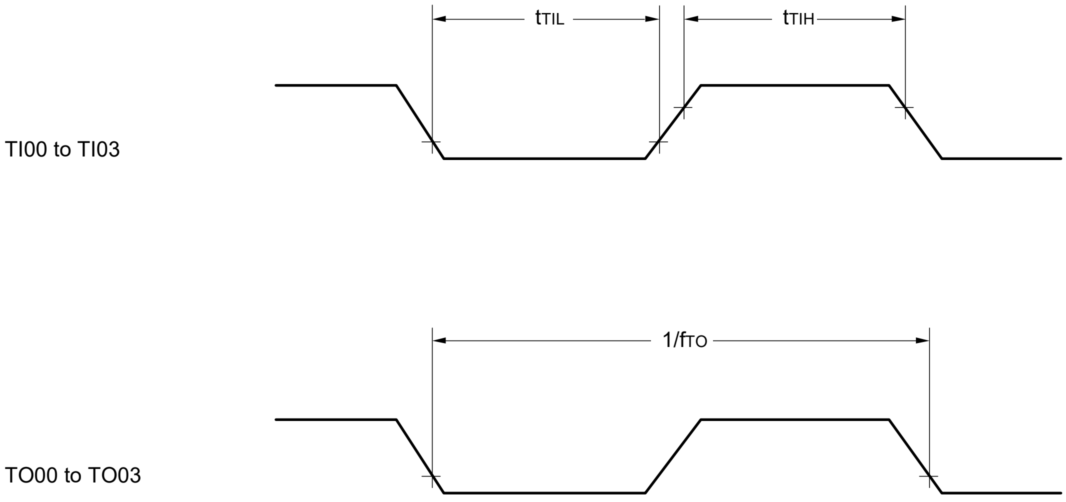
AC Timing Test Points



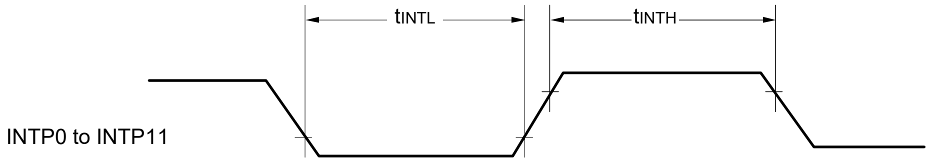
External System Clock Timing



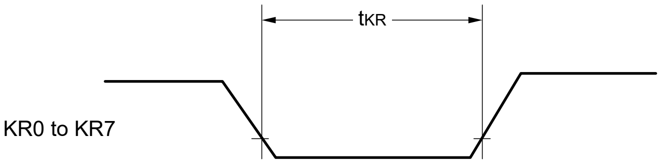
TI/TO Timing



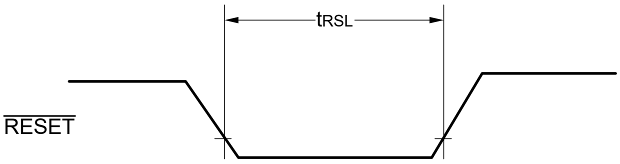
Interrupt Request Input Timing



Key Interrupt Input Timing

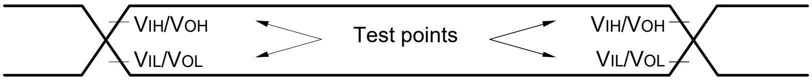


$\overline{\text{RESET}}$ Input Timing



35.5 Peripheral Functions Characteristics

AC Timing Test Points



35.5.1 Serial array unit

(1) During communication at same potential (UART mode)

When P01, P30, P31 and P54 are used as TxDq pins
(T_A = -40 to +85°C, 1.6 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{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		2.7 V ≤ EV_{DD} ≤ 5.5V			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 3		4.0		1.3		0.1		0.6	Mbps
		1.8 V ≤ EV_{DD} ≤ 5.5 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 3		4.0		1.3		0.1		0.6	Mbps
		1.7 V ≤ EV_{DD} ≤ 5.5 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 3		4.0		1.3		0.1		0.6	Mbps
		1.6 V ≤ EV_{DD} ≤ 5.5 V		—			f_{MCK}/6		f_{MCK}/6		f_{MCK}/6	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 3	—			1.3		0.1		0.6	Mbps

Note 1. Transfer rate in the SNOOZE mode is 4800 bps only.

Note 2. Following conditions must be satisfied on low level interface of EV_{DD} < V_{DD}.
2.4 V ≤ EV_{DD} < 2.7 V: MAX.2.6 Mbps
1.8 V ≤ EV_{DD} < 2.4 V: MAX.1.3 Mbps
1.6 V ≤ EV_{DD} < 1.8 V: MAX.0.6 Mbps

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 ≤ EV_{DD} ≤ 5.5 V)
	16 MHz (2.4 V ≤ EV_{DD} ≤ 5.5 V)
LS (low-speed main) mode:	8 MHz (1.8 V ≤ EV_{DD} ≤ 5.5 V)
LP (low-power main) mode:	1 MHz (1.8 V ≤ EV_{DD} ≤ 5.5 V)
LV (low-voltage main) mode:	4 MHz (1.6 V ≤ EV_{DD} ≤ 5.5 V)

Caution **Select the normal input buffer for the RxDq pin and the normal output mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg).**

When P20 is used as TxD1 pin

(T_A = -40 to +85°C, 1.6 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{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		4.0 V ≤ V_{DD} ≤ 5.5 V			$f_{MCK}/6$ Notes 1, 2, 3		$f_{MCK}/6$ Notes 1, 2		$f_{MCK}/6$ Notes 1, 2		$f_{MCK}/6$ Notes 1, 2	bps
			Theoretical value of the maximum transfer rate $f_{MCK} = f_{CLK}$ Notes 1, 3		1.5		1.3		0.1		0.6	Mbps
		2.7 V ≤ V_{DD} ≤ 5.5 V			$f_{MCK}/6$ Notes 1, 2, 3		$f_{MCK}/6$ Notes 1, 2		$f_{MCK}/6$ Notes 1, 2		$f_{MCK}/6$ Notes 1, 2	bps
			Theoretical value of the maximum transfer rate $f_{MCK} = f_{CLK}$ Notes 1, 3		1.2		1.2		0.1		0.6	Mbps
		2.4 V ≤ V_{DD} ≤ 5.5 V			$f_{MCK}/6$ Notes 1, 2, 3		$f_{MCK}/6$ Notes 1, 2		$f_{MCK}/6$ Notes 1, 2		$f_{MCK}/6$ Notes 1, 2	bps
			Theoretical value of the maximum transfer rate $f_{MCK} = f_{CLK}$ Notes 1, 3		1.0		1.0		0.1		0.6	Mbps
		1.8 V ≤ V_{DD} ≤ 5.5 V			Using prohibited		$f_{MCK}/6$ Notes 1, 2		$f_{MCK}/6$ Notes 1, 2		$f_{MCK}/6$ Notes 1, 2	bps
			Theoretical value of the maximum transfer rate $f_{MCK} = f_{CLK}$ Notes 1, 3				0.6		0.1		0.6	Mbps
		1.7 V ≤ V_{DD} ≤ 5.5 V					Using prohibited		Using prohibited		$f_{MCK}/6$ Notes 1, 2	bps
			Theoretical value of the maximum transfer rate $f_{MCK} = f_{CLK}$ Notes 1, 3								0.5	Mbps
		1.6 V ≤ V_{DD} ≤ 5.5 V									$f_{MCK}/6$ Notes 1, 2	bps
			Theoretical value of the maximum transfer rate $f_{MCK} = f_{CLK}$ Notes 1, 3								0.5	Mbps

Note 1. f_{MCK} is a frequency selected by setting the CKS bit in the SPS and SMR registers.

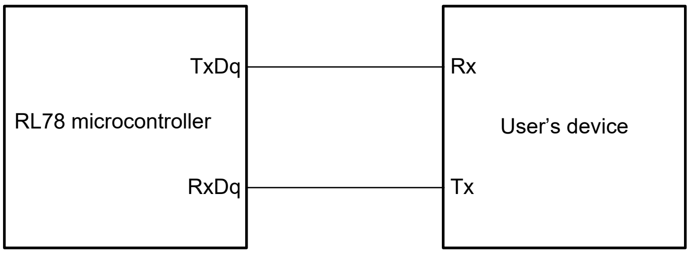
Note 2. The transfer rate of 4800 bps is only supported in the SNOOZE mode.
Note that the SNOOZE mode is not supported when fHOCO is 48 MHz.

Note 3. f_{CLK} in each operating mode is as follows.:

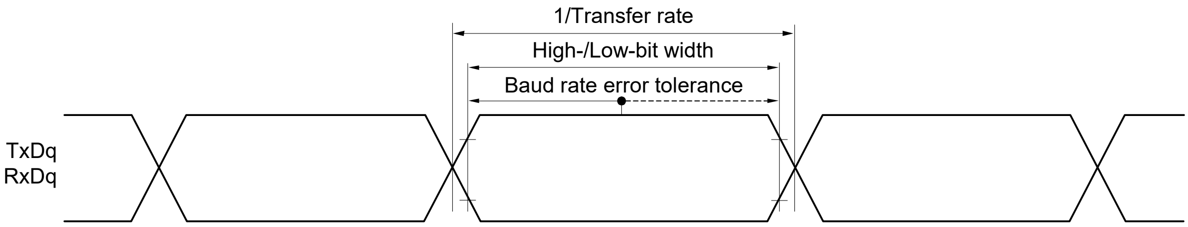
HS (high-speed main) mode:	24 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)
	16 MHz (2.4 V ≤ V_{DD} ≤ 5.5 V)
LS (low-speed main) mode:	8 MHz (1.8 V ≤ V_{DD} ≤ 5.5 V)
LP (low-power main) mode:	1 MHz (1.8 V ≤ V_{DD} ≤ 5.5 V)
LV (low-voltage main) mode:	4 MHz (1.6 V ≤ V_{DD} ≤ 5.5 V)

Caution **Select the normal input buffer for the RxDq pin and the normal output mode for the TxDq pin by using 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 and 1), g: PIM and POM number (g = 0, 2, 3 and 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 to 03))

(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 ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{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} ≥ 2/f_{CLK}	83.3		250		2000		500		ns
SCKp high-/low-level width	t_{KL1}	4.0 V ≤ EV_{DD} ≤ 5.5 V	t_{KCY1}/2 - 7		t_{KCY1}/2 - 50		t_{KCY1}/2 - 50		t_{KCY1}/2 - 50		ns
		2.7 V ≤ EV_{DD} ≤ 5.5 V	t_{KCY1}/2 - 10								ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	4.0 V ≤ EV_{DD} ≤ 5.5 V	23		110		110		110		ns
		2.7 V ≤ EV_{DD} ≤ 5.5 V	33								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)

When P01, P32, P53, P54 and P56 are used as SOmn pins
(T_A = -40 to +85°C, 1.6 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{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} ≥ 4/f_{CLK}	2.7 V ≤ EV_{DD} ≤ 5.5 V	167		500		4000		1000		ns
			2.4 V ≤ EV_{DD} ≤ 5.5 V	250								
			1.8 V ≤ EV_{DD} ≤ 5.5 V	500								
			1.7 V ≤ EV_{DD} ≤ 5.5 V	1000		1000						
			1.6 V ≤ EV_{DD} ≤ 5.5 V	Using prohibited								
SCKp high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ EV_{DD} ≤ 5.5 V		t_{KCY1}/2- 12		t_{KCY1}/2 - 50		t_{KCY1}/2 - 50		t_{KCY1}/2 - 50		ns
		2.7 V ≤ EV_{DD} ≤ 5.5 V		t_{KCY1}/2- 18								
		2.4 V ≤ EV_{DD} ≤ 5.5 V		t_{KCY1}/2- 38								
		1.8 V ≤ EV_{DD} ≤ 5.5 V		t_{KCY1}/2- 50								
		1.7 V ≤ EV_{DD} ≤ 5.5 V		t_{KCY1}/2- 100		t_{KCY1}/2 - 100		t_{KCY1}/2 - 100		t_{KCY1}/2 - 100		
		1.6 V ≤ EV_{DD} ≤ 5.5 V		Using prohibited								
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	4.0 V ≤ EV_{DD} ≤ 5.5 V		44		110		110		110		ns
		2.7 V ≤ EV_{DD} ≤ 5.5 V										
		2.4 V ≤ EV_{DD} ≤ 5.5 V		75								
		1.8 V ≤ EV_{DD} ≤ 5.5 V		110								
		1.7 V ≤ EV_{DD} ≤ 5.5 V		220		220		220		220		
		1.6 V ≤ EV_{DD} ≤ 5.5 V		Using prohibited								
SIp hold time (from SCKp↑) Note 2	t_{KSI1}	1.7 V ≤ EV_{DD} ≤ 5.5 V		19		19		19		19		ns
		1.6 V ≤ EV_{DD} ≤ 5.5 V		Using prohibited								
Delay time from SCKp↓ to SOp output Note 3	t_{KSO1}	C = 30 pF Note 4	1.7 V ≤ EV_{DD} ≤ 5.5 V		33.4		33.4		33.4		33.4	ns
			1.6 V ≤ EV_{DD} ≤ 5.5 V		Using prohibited							

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, 01, 10 and 11), m: Unit number (m = 0), n: Channel number (n = 0 to 3), g: PIM and POM numbers (g = 0, 2, 3 to 5 and 12)

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 to 03))

When P20 is used as SO10 pin

(T_A = -40 to +85°C, 1.6 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{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}	$t_{KCY1} \geq 4/f_{CLK}$	4.0 V ≤ V_{DD} ≤ 5.5 V	600		600		4000		1000		ns
			2.7 V ≤ V_{DD} ≤ 5.5 V	850		850						
			2.4 V ≤ V_{DD} ≤ 5.5 V	1000		1000						
			1.8 V ≤ V_{DD} ≤ 5.5 V	—		1500				1500		
			1.7 V ≤ V_{DD} ≤ 5.5 V	—		—		—		2000		
			1.6 V ≤ V_{DD} ≤ 5.5 V	—		—		—				
SCKp high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V		$t_{KCY1}/2$ - 12		$t_{KCY1}/2$ - 50		$t_{KCY1}/2$ - 50		$t_{KCY1}/2$ - 50		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V		$t_{KCY1}/2$ - 18								
		2.4 V ≤ V_{DD} ≤ 5.5 V		$t_{KCY1}/2$ - 38								
		1.8 V ≤ V_{DD} ≤ 5.5 V		—								
		1.7 V ≤ V_{DD} ≤ 5.5 V		—		—		—		$t_{KCY1}/2$ - 100		
		1.6 V ≤ V_{DD} ≤ 5.5 V		—		—		—				
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V		44		110		110		110		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V										
		2.4 V ≤ V_{DD} ≤ 5.5 V		75								
		1.8 V ≤ V_{DD} ≤ 5.5 V		—								
		1.7 V ≤ V_{DD} ≤ 5.5 V		—		—		—		220		
		1.6 V ≤ V_{DD} ≤ 5.5 V		—		—		—				
SIp hold time (from SCKp↑) Note 2	t_{KSI1}	2.4 V ≤ V_{DD} ≤ 5.5 V		19		19		19		19		ns
		1.8 V ≤ V_{DD} ≤ 5.5 V		—								
		1.6 V ≤ V_{DD} ≤ 5.5 V		—		—		—				
Delay time from SCKp↓ to SOp output Note 3	t_{KSO1}	C = 30 pF Note 4	2.4 V ≤ V_{DD} ≤ 5.5 V		150		250		250		300	ns
			1.8 V ≤ V_{DD} ≤ 5.5 V		—							
			1.6 V ≤ V_{DD} ≤ 5.5 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, 01, 10 and 11), m: Unit number (m = 0), n: Channel number (n = 0 to 3), g: PIM and POM numbers (g = 0, 4 and 12)

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 to 03))

(4) During communication at same potential (Simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

When P01, P32, P53, P54 and P56 are used as SOmn pins

(T_A = -40 to +85°C, 1.6 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{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 [Note 3]	t_{KCY2}	$4.0\ V \le EV_{DD} \le 5.5\ V$	f_{MCK} > 20 MHz	$8/f_{MCK}$		—		—		—		ns
			$f_{MCK} \le$ 20 MHz	$6/f_{MCK}$		$6/f_{MCK}$		$6/f_{MCK}$		$6/f_{MCK}$		
		$2.7\ V \le EV_{DD} \le 5.5\ V$	f_{MCK} > 16 MHz	$8/f_{MCK}$		—		—		—	—	
			$f_{MCK} \le$ 16 MHz	$6/f_{MCK}$		$6/f_{MCK}$		$6/f_{MCK}$		$6/f_{MCK}$		
		$2.4\ V \le EV_{DD} \le 5.5\ V$		$6/f_{MCK}$ and 500								
		$1.8\ V \le EV_{DD} \le 5.5\ V$		$6/f_{MCK}$ and 750								
		$1.7\ V \le EV_{DD} \le 5.5V$		$6/f_{MCK}$ and 1500		$6/f_{MCK}$ and 1500						
		$1.6\ V \le EV_{DD} \le 5.5\ V$		—								
SCKp high-/ low-level width	t_{KH2}, t_{KL2}	$4.0\ V \le EV_{DD} \le 5.5\ V$		$t_{KCY2}/2$ - 7		$t_{KCY2}/2$ - 7		$t_{KCY2}/2$ - 7		$t_{KCY2}/2$ - 7		ns
		$2.7\ V \le EV_{DD} \le 5.5\ V$		$t_{KCY2}/2$ - 8		$t_{KCY2}/2$ - 8		$t_{KCY2}/2$ - 8		$t_{KCY2}/2$ - 8		
		$1.8\ V \le EV_{DD} \le 5.5\ V$		$t_{KCY2}/2$ - 18		$t_{KCY2}/2$ - 18		$t_{KCY2}/2$ - 18		$t_{KCY2}/2$ - 18		
		$1.7\ V \le EV_{DD} \le 5.5\ V$		$t_{KCY2}/2$ - 66		$t_{KCY2}/2$ - 66		$t_{KCY2}/2$ - 66		$t_{KCY2}/2$ - 66		
		$1.6\ V \le EV_{DD} \le 5.5\ V$		—								
SIp setup time (to SCKp↑) [Note 1]	t_{SIK2}	$2.7\ V \le EV_{DD} \le 5.5\ V$		$1/f_{MCK}$ + 20		$1/f_{MCK}$ + 30		$1/f_{MCK}$ + 30		$1/f_{MCK}$ + 30		ns
		$1.8\ V \le EV_{DD} \le 5.5\ V$		$1/f_{MCK}$ + 30		$1/f_{MCK}$ + 30		$1/f_{MCK}$ + 30		$1/f_{MCK}$ + 30		
		$1.7\ V \le EV_{DD} \le 5.5\ V$		$1/f_{MCK}$ + 40		$1/f_{MCK}$ + 40		$1/f_{MCK}$ + 40		$1/f_{MCK}$ + 40		
		$1.6\ V \le EV_{DD} \le 5.5\ V$		—								
SIp hold time (from SCKp↑) [Note 2]	t_{KSI2}	$1.8\ V \le EV_{DD} \le 5.5\ V$		$1/f_{MCK}$ + 31		$1/f_{MCK}$ + 31		$1/f_{MCK}$ + 31		$1/f_{MCK}$ + 31		ns
		$1.7\ V \le EV_{DD} \le 5.5\ V$		$1/f_{MCK}$ + 250		$1/f_{MCK}$ + 250		$1/f_{MCK}$ + 250		$1/f_{MCK}$ + 250		
		$1.6\ V \le EV_{DD} \le 5.5\ V$		—								

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. 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, 01, 10 and 11), m: Unit number (m = 0), n: Channel number (n = 0 to 3), g: PIM and POM numbers (g = 0, 2, 3 to 5 and 12)

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 to 03))

(T_A = -40 to +85°C, 1.6 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{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.	
Delay time from SCKp↓ to SOp output Note 1	t_{KSO2}	C = 30 pF Note 2	2.7 V ≤ EV_{DD} ≤ 5.5 V		$2/f_{MCK}$ + 44		$2/f_{MCK}$ + 110		$2/f_{MCK}$ + 110		$2/f_{MCK}$ + 110	ns
			2.4 V ≤ EV_{DD} ≤ 5.5 V		$2/f_{MCK}$ + 75							
			1.8 V ≤ EV_{DD} ≤ 5.5 V		$2/f_{MCK}$ + 110							
			1.7 V ≤ EV_{DD} ≤ 5.5 V		$2/f_{MCK}$ + 220		$2/f_{MCK}$ + 220		$2/f_{MCK}$ + 220		$2/f_{MCK}$ + 220	
			1.6 V ≤ EV_{DD} ≤ 5.5 V		—							
$\overline{SSI00}$ setup time	t_{SSIK}	DAPmn = 0	2.7 V ≤ V_{DD} ≤ 5.5 V	120		120		120		120		ns
			1.8 V ≤ V_{DD} < 2.7 V	200		200		200		200		
			1.7 V ≤ V_{DD} < 1.8 V	400		400		400		400		
			1.6 V ≤ V_{DD} < 1.7 V	—								
		DAPmn = 1	2.7 V ≤ V_{DD} ≤ 5.5 V	$1/f_{MCK}$ + 120		$1/f_{MCK}$ + 120		$1/f_{MCK}$ + 120		$1/f_{MCK}$ + 120		ns
			1.8 V ≤ V_{DD} < 2.7 V	$1/f_{MCK}$ + 200		$1/f_{MCK}$ + 200		$1/f_{MCK}$ + 200		$1/f_{MCK}$ + 200		
			1.7 V ≤ V_{DD} < 1.8 V	$1/f_{MCK}$ + 400		$1/f_{MCK}$ + 400		$1/f_{MCK}$ + 400		$1/f_{MCK}$ + 400		
			1.6 V ≤ V_{DD} < 1.7 V	—								
$\overline{SSI00}$ hold time	t_{KSSI}	DAPmn = 0	2.7 V ≤ V_{DD} ≤ 5.5 V	$1/f_{MCK}$ + 120		$1/f_{MCK}$ + 120		$1/f_{MCK}$ + 120		$1/f_{MCK}$ + 120		ns
			1.8 V ≤ V_{DD} < 2.7 V	$1/f_{MCK}$ + 200		$1/f_{MCK}$ + 200		$1/f_{MCK}$ + 200		$1/f_{MCK}$ + 200		
			1.7 V ≤ V_{DD} < 1.8 V	$1/f_{MCK}$ + 400		$1/f_{MCK}$ + 400		$1/f_{MCK}$ + 400		$1/f_{MCK}$ + 400		
			1.6 V ≤ V_{DD} < 1.7 V	—								
		DAPmn = 1	2.7 V ≤ V_{DD} ≤ 5.5 V	120		120		120		120		ns
			1.8 V ≤ V_{DD} < 2.7 V	200		200		200		200		
			1.7 V ≤ V_{DD} < 1.8 V	400		400		400		400		
			1.6 V ≤ V_{DD} < 1.7 V	—								

Note 1. 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 2. C is the load capacitance of the SOp output lines.

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 1. p: CSI number (p = 00, 01, 10 and 11), m: Unit number (m = 0), n: Channel number (n = 0 to 3), g: PIM and POM numbers (g = 0, 2, 3 to 5 and 12)

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 to 03))

When P20 is used as SO10 pin

(T_A = -40 to +85°C, 1.6 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{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 Note 5	t_{KCY2}	4.0 V ≤ V_{DD} ≤ 5.5 V	f_{MCK} > 20 MHz	14/f_{MCK}		—		—		—		ns
			f_{MCK} ≤ 20 MHz	12/f_{MCK}		12/f_{MCK}		12/f_{MCK}		12/f_{MCK}		
		2.7 V ≤ V_{DD} ≤ 5.5 V	f_{MCK} > 16 MHz	14/f_{MCK} and 850		—		—		—		
			f_{MCK} ≤ 16 MHz	12/f_{MCK} and 850		12/f_{MCK}		12/f_{MCK}		12/f_{MCK}		
		2.4 V ≤ V_{DD} ≤ 5.5 V		12/f_{MCK} and 1000		12/f_{MCK}		12/f_{MCK}		12/f_{MCK}		
		1.8 V ≤ V_{DD} ≤ 5.5 V		—		12/f_{MCK}		12/f_{MCK}		12/f_{MCK}		
		1.7 V ≤ V_{DD} ≤ 5.5V		—		—		—		12/f_{MCK}		
		1.6 V ≤ V_{DD} ≤ 5.5 V		—		—		—				
SCKp high-/ low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ V_{DD} ≤ 5.5 V		t_{KCY2}/2 - 7		t_{KCY2}/2 - 7		t_{KCY2}/2 - 7		t_{KCY2}/2 - 7		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V		t_{KCY2}/2 - 8		t_{KCY2}/2 - 8		t_{KCY2}/2 - 8		t_{KCY2}/2 - 8		
		1.8 V ≤ V_{DD} ≤ 5.5 V		—		t_{KCY2}/2 - 18		t_{KCY2}/2 - 18		t_{KCY2}/2 - 18		
		1.7 V ≤ V_{DD} ≤ 5.5 V		—		—		—		t_{KCY2}/2 - 66		
		1.6 V ≤ V_{DD} ≤ 5.5 V		—		—		—				
SIp setup time (to SCKp↑) Note 1	t_{SIK2}	2.7 V ≤ V_{DD} ≤ 5.5 V		1/f_{MCK} + 20		1/f_{MCK} + 30		1/f_{MCK} + 30		1/f_{MCK} + 30		ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		1/f_{MCK} + 30								
		1.8 V ≤ V_{DD} ≤ 5.5 V		—								
		1.7 V ≤ V_{DD} ≤ 5.5 V		—		—		—		1/f_{MCK} + 40		
		1.6 V ≤ V_{DD} ≤ 5.5 V		—		—		—				
SIp hold time (from SCKp↑) Note 2	t_{KSI2}	2.4 V ≤ V_{DD} ≤ 5.5 V		1/f_{MCK} + 31		1/f_{MCK} + 31		1/f_{MCK} + 31		1/f_{MCK} + 31		ns
		1.8 V ≤ V_{DD} ≤ 5.5 V		—		1/f_{MCK} + 31		1/f_{MCK} + 31		1/f_{MCK} + 31		
		1.7 V ≤ V_{DD} ≤ 5.5 V		—		—		—		1/f_{MCK} + 250		
		1.6 V ≤ V_{DD} ≤ 5.5 V		—		—		—				
Delay time from SCKp↓ to SOp output Note 3	t_{KSO2}	C = 30 pF Note 4	2.7 V ≤ V_{DD} ≤ 5.5 V		2/f_{MCK} + 160		2/f_{MCK} + 260		2/f_{MCK} + 260		2/f_{MCK} + 260	ns
			2.4 V ≤ V_{DD} ≤ 5.5 V		2/f_{MCK} + 190							
			1.8 V ≤ V_{DD} ≤ 5.5 V		—							
			1.7 V ≤ V_{DD} ≤ 5.5 V		—		—		—		2/f_{MCK} + 320	
			1.6 V ≤ V_{DD} ≤ 5.5 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.

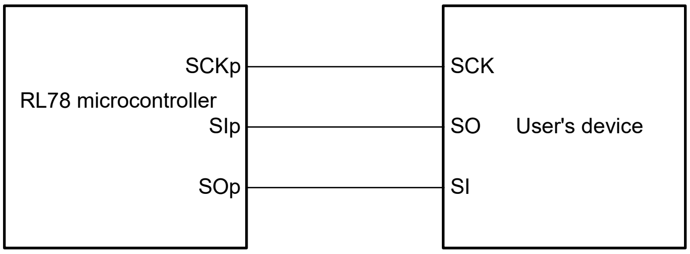
Note 5. The maximum transfer rate when using the SNOOZE mode is 1 Mbps.

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).**

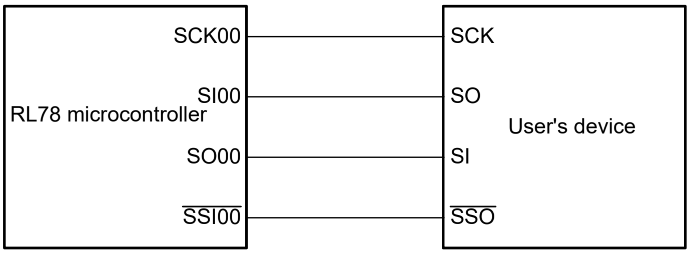
Remark 1. p: CSI number (p = 00, 01, 10 and 11), m: Unit number (m = 0), n: Channel number (n = 0 to 3), g: PIM and POM numbers (g = 0, 4 and 12)

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 to 03))

Simplified SPI (CSI) mode connection diagram (during communication at same potential)

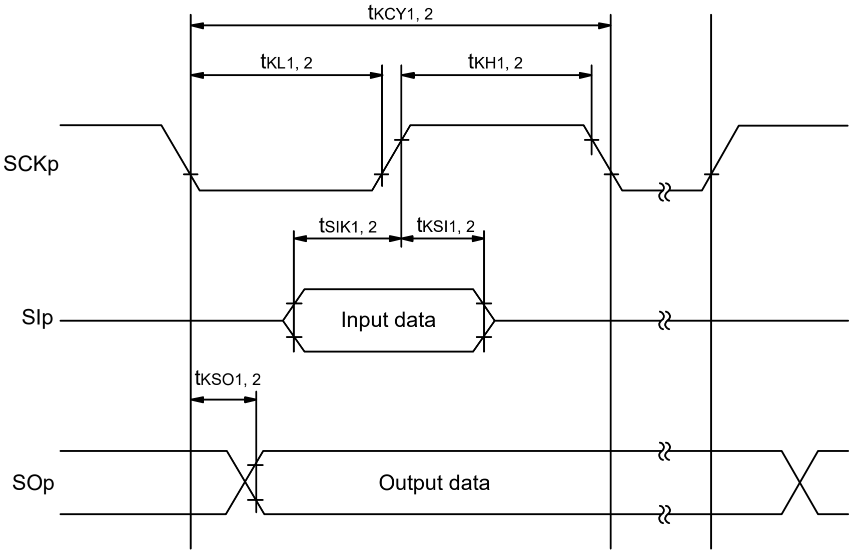


Simplified SPI (CSI) mode connection diagram (during communication at same potential)
(Slave Transmission of slave select input function (CSI00))

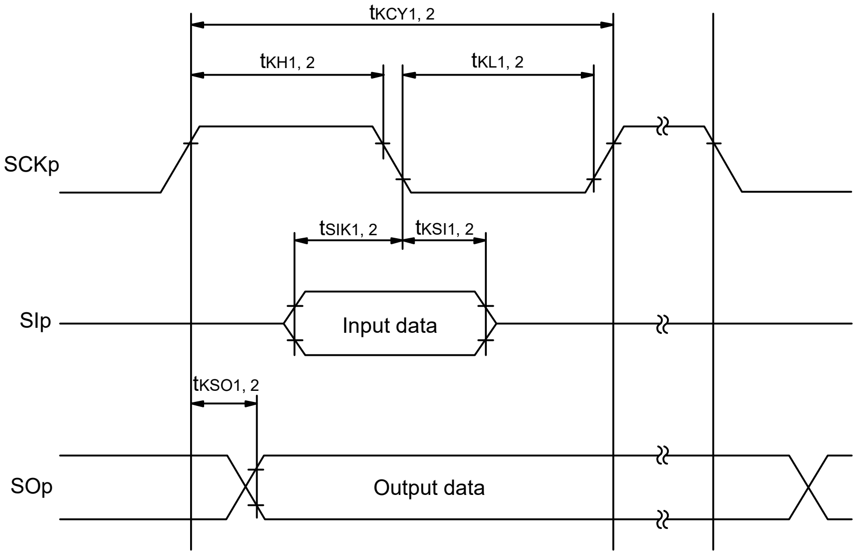


Remark p: CSI number (p = 00, 01, 10 and 11)

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, 10 and 11)
Remark 2. m: Unit number, n: Channel number (mn = 00 to 03)

(5) During communication at same potential (simplified I^2C mode)

(T_A = -40 to +85°C, 1.6 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

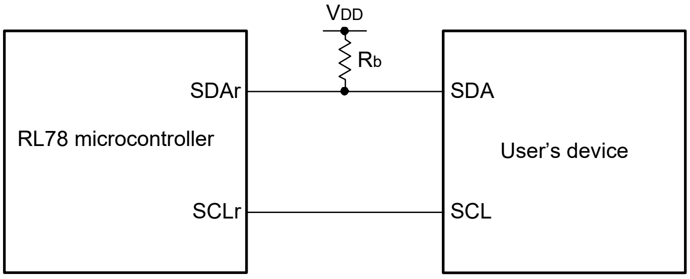
Parameter	Symbol	Conditions	HS (high-speed main) Mode		LS (low-speed main) Mode		LP (Low-power main) mode		LV (low-voltage main) Mode		Unit
			MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	MIN.	MAX.	
SCLr clock frequency	f_{SCL}	2.7 V ≤ EV_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ		1000 Note 1		400 Note 1		250 Note 1		400 Note 1	kHz
		1.8 V ≤ EV_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ		400 Note 1							
		1.8 V ≤ EV_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ		300 Note 1		300 Note 1		250 Note 1		300 Note 1	
		1.7 V ≤ EV_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ		250 Note 1		250 Note 1		250 Note 1		250 Note 1	
		1.6 V ≤ EV_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ		—							
Hold time when SCLr = “L”	t_{LOW}	2.7 V ≤ EV_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	475		1150		1150		1150		ns
		1.8 V ≤ EV_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1150								
		1.8 V ≤ EV_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ	1550		1550		1550		1550		
		1.7 V ≤ EV_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	1850		1850		1850		1850		
		1.6 V ≤ EV_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	—								
Hold time when SCLr = “H”	t_{HIGH}	2.7 V ≤ EV_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	475		1150		1150		1150		ns
		1.8 V ≤ EV_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	1150								
		1.8 V ≤ EV_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ	1550		1550		1550		1550		
		1.7 V ≤ EV_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	1850		1850		1850		1850		
		1.6 V ≤ EV_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	—								
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ EV_{DD} ≤ 5.5 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 ≤ EV_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	$1/f_{MCK}$ + 145 Note 2								
		1.8 V ≤ EV_{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		$1/f_{MCK}$ + 230 Note 2		
		1.7 V ≤ EV_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	$1/f_{MCK}$ + 290 Note 2		$1/f_{MCK}$ + 290 Note 2		$1/f_{MCK}$ + 290 Note 2		$1/f_{MCK}$ + 290 Note 2		
		1.6 V ≤ EV_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	—								
Data hold time (transmission)	$t_{HD:DAT}$	2.7 V ≤ EV_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	0	305	0	305	0	305	0	305	ns
		1.8 V ≤ EV_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ		355		355		355		355	
		1.8 V ≤ EV_{DD} < 2.7 V, C_b = 100 pF, R_b = 5 kΩ		405		405		405		405	
		1.7 V ≤ EV_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ									
		1.6 V ≤ EV_{DD} < 1.8 V, C_b = 100 pF, R_b = 5 kΩ	—	—							

Note 1. The value must be equal to or less than f_{MCK}/4.

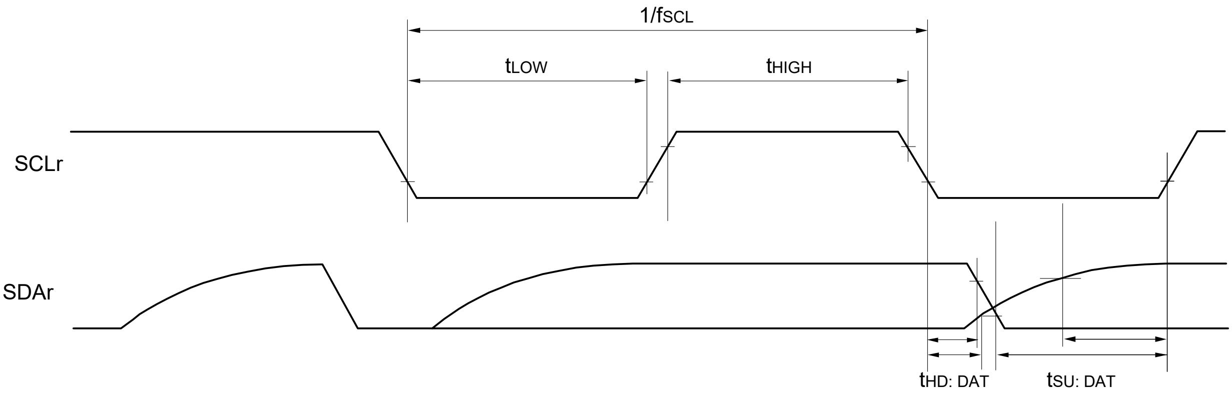
Note 2. Set the f_{MCK} value to keep the hold time of SCLr = "L" and SCLr = "H".

Caution Select the normal input buffer and the N-ch open drain output (EV$_{DD}$ 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 I^{2}C mode connection diagram (during communication at same potential)



Simplified I^{2}C 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
r: IIC number (r = 00, 01, 10 and 11), g: PIM number (g = 0, 3 and 5), h: POM number (h = 0, 3 and 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 (m = 0), n: Channel number (n = 0 to 3), mn = 00 to 03)

(6) Communication at different potential (1.8 V, 2.5 V, 3.0 V) (UART mode) (dedicated baud rate generator output)

(T_A = -40 to +85°C, 1.8 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{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.	
Transfer rate		reception	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		f_{MCK}/6 Note 1		f_{MCK}/6 Note 1		f_{MCK}/6 Note 1		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
			2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		f_{MCK}/6 Note 1		f_{MCK}/6 Note 1		f_{MCK}/6 Note 1		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 ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V		f_{MCK}/6 Notes 1, 2, 4		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 EV_{DD} ≥ V_b.

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} ≤ 5.5 V)
	16 MHz (2.4 V ≤ V_{DD} ≤ 5.5 V)
LS (low-speed main) mode:	8 MHz (1.8 V ≤ V_{DD} ≤ 5.5 V)
LP (low-power main) mode:	1 MHz (1.8 V ≤ V_{DD} ≤ 5.5 V)
LV (low-voltage main) mode:	4 MHz (1.6 V ≤ V_{DD} ≤ 5.5 V)

Note 4. The following conditions are required for low voltage interface when EV_{DD} < V_{DD}
2.4 V ≤ EV_{DD} < 2.7 V: MAX. 2.6 Mbps
1.8 V ≤ EV_{DD} < 2.4 V: MAX. 1.3 Mbps

Caution **Select the TTL input buffer for the RxDq pin and the N-ch open drain output (EV_{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 and 1), g: PIM and POM number (g = 0, 2, 3, 5 and 12)

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 to 03))

(T_A = -40 to +85°C, 1.8 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{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		Transmission	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		Note 1		Note 1		Note 1		Note 1	bps
			Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 1.4 kΩ, V_b = 2.7 V		2.8 Note 2		2.8 Note 2		2.8 Note 2		2.8 Note 2	Mbps
			2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		Note 3		Note 3		Note 3		Note 3	bps
			Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 2.7 kΩ, V_b = 2.3 V		1.2 Note 4		1.2 Note 4		1.2 Note 4		1.2 Note 4	Mbps
			1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V		Notes 5, 6		Notes 5, 6		Notes 5, 6		Notes 5, 6	bps
			Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 5.5 kΩ, V_b = 1.6 V		0.43 Note 7		0.43 Note 7		0.43 Note 7		0.43 Note 7	Mbps

Note 1. The smaller maximum transfer rate derived by using $f_{MCK}/6$ or the following expression is the valid maximum transfer rate.
Expression for calculating the transfer rate when 4.0 V ≤ EV_{DD} ≤ 5.5 V and 2.7 V ≤ V_b ≤ 4.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln\left(1 - \frac{2.2}{V_b}\right)\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln\left(1 - \frac{2.2}{V_b}\right)\}}{\left(\frac{1}{\text{Transfer rate}}\right) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* 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. The smaller maximum transfer rate derived by using $f_{MCK}/6$ or the following expression is the valid maximum transfer rate.
Expression for calculating the transfer rate when 2.7 V ≤ EV_{DD} ≤ 4.0 V and 2.3 V ≤ V_b ≤ 2.7 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln\left(1 - \frac{2.0}{V_b}\right)\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln\left(1 - \frac{2.0}{V_b}\right)\}}{\left(\frac{1}{\text{Transfer rate}}\right) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides

Note 4. This value as an example is calculated when the conditions described in the “Conditions” column are met.
Refer to Note 3 above to calculate the maximum transfer rate under conditions of the customer.

Note 5. Use it with EV_{DD} ≥ V_b.

Note 6. The smaller maximum transfer rate derived by using $f_{MCK}/6$ or the following expression is the valid maximum transfer rate.
Expression for calculating the transfer rate when $1.8\ V \leq EV_{DD} < 3.3\ V$ and $1.6\ V \leq V_b \leq 2.0\ V$

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln\left(1 - \frac{1.5}{V_b}\right)\} \times 3}\ [\text{bps}]$$

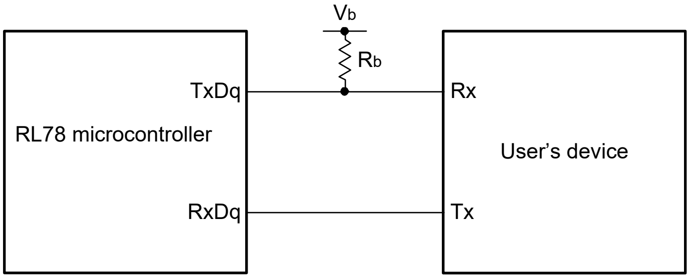
$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln\left(1 - \frac{1.5}{V_b}\right)\}}{\left(\frac{1}{\text{Transfer rate}}\right) \times \text{Number of transferred bits}} \times 100\ [\%]$$

* This value is the theoretical value of the relative difference between the transmission and reception sides

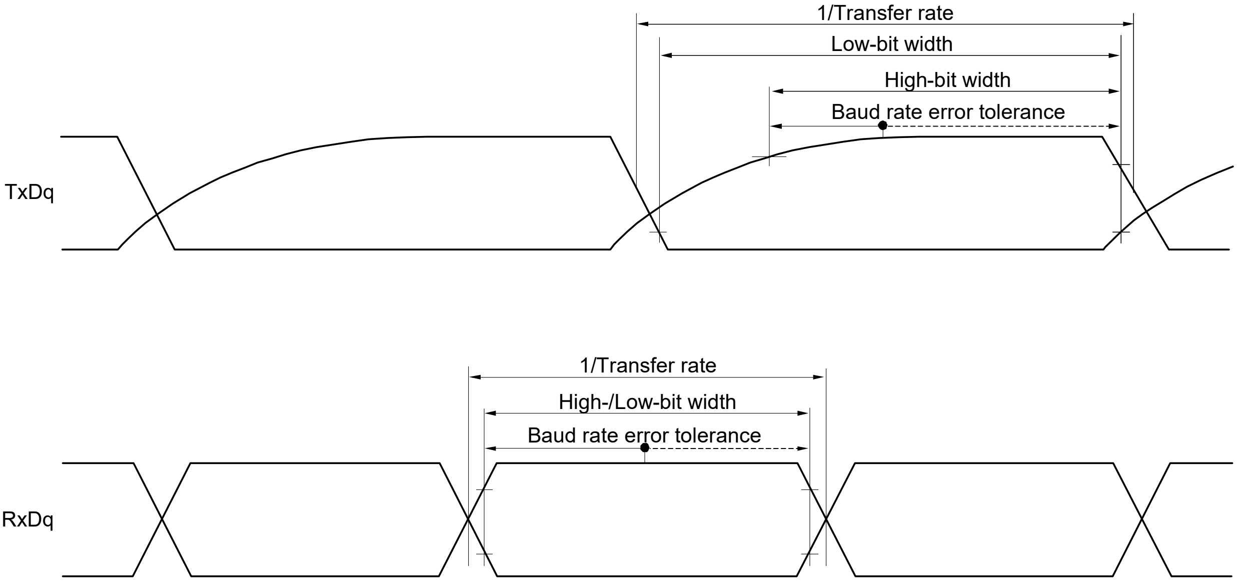
Note 7. This value as an example is calculated when the conditions described in the “Conditions” column are met.
Refer to Note 6 above to calculate the maximum transfer rate under conditions of the customer.

Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (EV_{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 and 1), g: PIM and POM number (g = 0, 2, 3, 5 and 12)

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 to 03))

(7) Communication at different potential (1.8 V, 2.5 V, 3.0 V) (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output, corresponding CSI00 only)

(T_A = -40 to +85°C, 2.7 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{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} ≥ $2/f_{CLK}$	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ	200		1150		1150		1150		ns
		t_{KCY1} ≥ $2/f_{CLK}$	2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ	300								ns
SCKp high-level width	t_{KH1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		$t_{KCY1}/2$ - 50		$t_{KCY1}/2$ - 50		$t_{KCY1}/2$ - 50		$t_{KCY1}/2$ - 50		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		$t_{KCY1}/2$ - 120		$t_{KCY1}/2$ - 120		$t_{KCY1}/2$ - 120		$t_{KCY1}/2$ - 120		ns
SCKp low-level width	t_{KL1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		$t_{KCY1}/2$ - 7		$t_{KCY1}/2$ - 50		$t_{KCY1}/2$ - 50		$t_{KCY1}/2$ - 50		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		$t_{KCY1}/2$ - 10								
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		58		479		479		479		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		121								
SIp hold time (from SCKp↑) Note 1	t_{KSI1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		10		10		10		10		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ										
Delay time from SCKp↓ to SOp output Note 1	t_{KSO1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ			60		60		60		60	ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ			130		130		130		130	
SIp setup time (to SCKp↓) Note 2	t_{SIK1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		23		110		110		110		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ		33								
SIp hold time (from SCKp↓) Note 2	t_{KSI1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		10		10		10		10		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ										

(T_A = -40 to +85°C, 2.7 V $\leq$ EV_{DD} $\leq$ V_{DD} $\leq$ 5.5 V, V_{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.	
Delay time from SCKp↑ to SOp output Note 2	t_{KSO1}	4.0 V $\leq$ EV_{DD} $\leq$ 5.5 V, 2.7 V $\leq$ V_b $\leq$ 4.0 V, C_b = 20 pF, R_b = 1.4 kΩ		10		10		10		10	ns
		2.7 V $\leq$ EV_{DD} < 4.0 V, 2.3 V $\leq$ V_b $\leq$ 2.7 V, C_b = 20 pF, R_b = 2.7 kΩ									

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 (EV_{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, 3.0 V) (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = -40 to +85°C, 1.8 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{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} ≥ 4/f_{CLK}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	300		1150		1150		1150		ns
			2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	500								ns
			1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note, C_b = 30 pF, R_b = 5.5 kΩ	1150								ns
SCKp high-level width	t_{KH1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KCY1}/2 - 75		t_{KCY1}/2 - 75		t_{KCY1}/2 - 75		t_{KCY1}/2 - 75		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 - 170		t_{KCY1}/2 - 170		t_{KCY1}/2 - 170		t_{KCY1}/2 - 170		ns
		1.8 V ≤ EV_{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}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KCY1}/2 - 12		t_{KCY1}/2 - 50		t_{KCY1}/2 - 50		t_{KCY1}/2 - 50		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 - 18								
		1.8 V ≤ EV_{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								ns

Note Use it with EV_{DD} ≥ V_b.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (EV_{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, 3.0 V) (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = -40 to +85°C, 1.8 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{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}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	81		479		479		479		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	177								
		1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ	479								
SIp hold time (from SCKp↑) Note 1	t_{KSI1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	19		19		19		19		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ									
		1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ									
Delay time from SCKp↓ to SOp output Note 1	t_{KSO1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		100		100		100		100	ns
		2.7 V ≤ EV_{DD} < 4.0 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 ≤ EV_{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}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	44		110		110		110		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ									
		1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ	110								
SIp hold time (from SCKp↓) Note 2	t_{KSI1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	19		19		19		19		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ									
		1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ									
Delay time from SCKp↑ to SOp output Note 2	t_{KSO1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		25		25		25		25	ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ									
		1.8 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ									

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

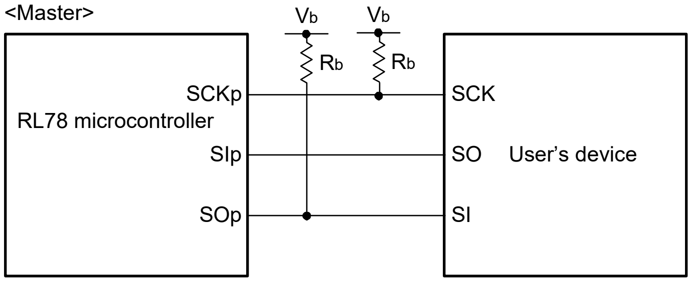
Note 2. When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. Use it with EV_{DD} ≥ V_b.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (EV_{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)

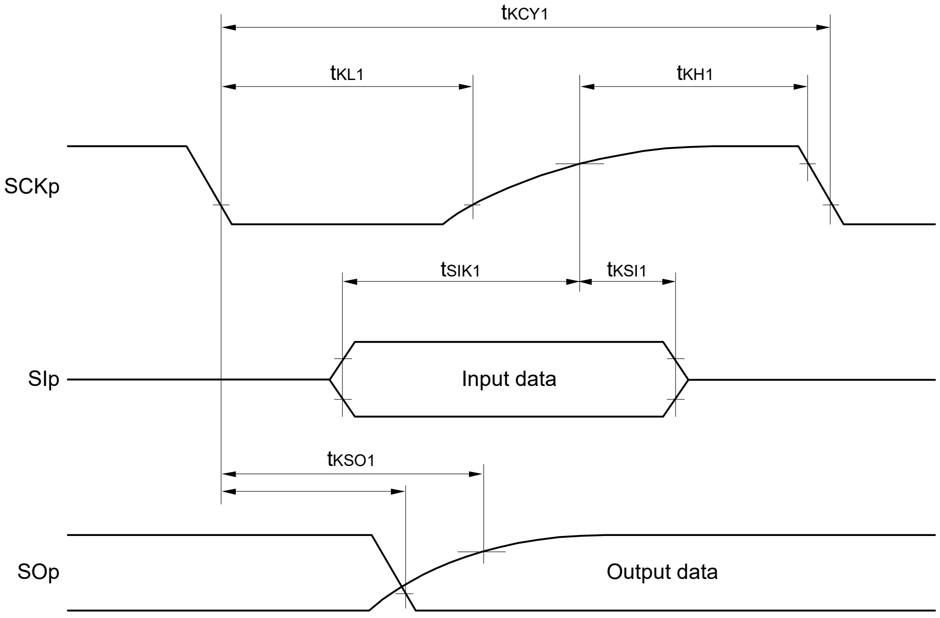


Remark 1. R_b[Ω]: Communication line (SCKp, SOp) pull-up resistance, C_b[F]: Communication line (SCKp, SOp) load capacitance, V_b[V]: Communication line voltage

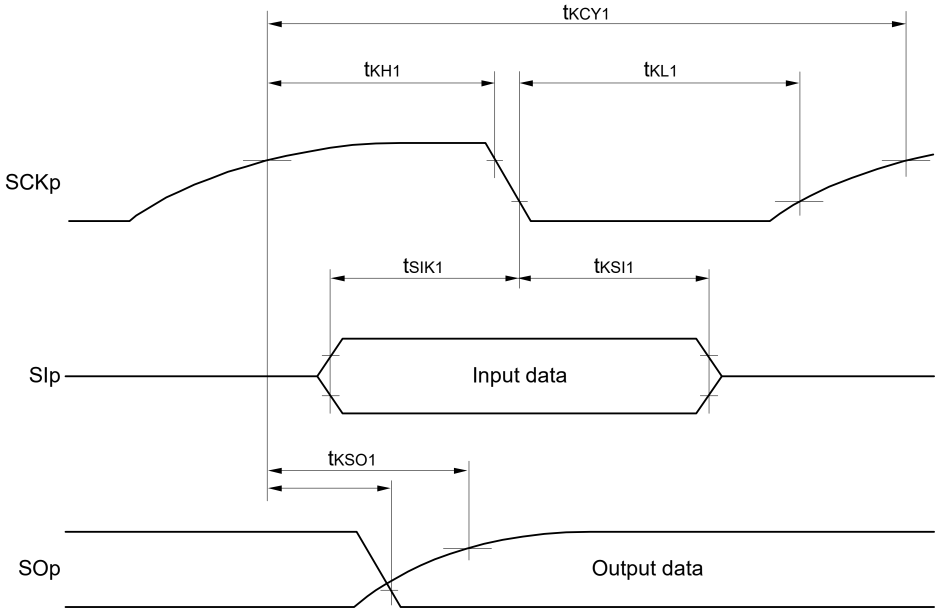
Remark 2. p: CSI number (p = 00, 01, 10 and 11), m: Unit number (m = 0), n: Channel number (n = 0 to 3), g: PIM and POM numbers (g = 0, 2, 3 to 5 and 12)

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 to 03))

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, 10 and 11), m: Unit number (m = 0), n: Channel number (n = 0 to 3), g: PIM and POM numbers (g = 0, 2, 3 to 5 and 12)

(9) Communication at different potential (1.8 V, 2.5 V, 3.0 V) (Simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

(T_A = -40 to 85°C, 1.8 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{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 [Note 1]	t_{KCY2}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V	20 MHz < f_{MCK} ≤ 24 MHz	12/f_{MCK}		—		—		—		ns
			8 MHz < f_{MCK} ≤ 20 MHz	10/f_{MCK}		—		—		—		ns
			4 MHz < f_{MCK} ≤ 8 MHz	8/f_{MCK}		16/f_{MCK}		—		—		ns
			f_{MCK} ≤ 4 MHz	6/f_{MCK}		10/f_{MCK}		10/f_{MCK}		10/f_{MCK}		ns
		2.7 V ≤ EV_{DD} < 4.0 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 ≤ EV_{DD} < 2.7 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}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		t_{KCY2}/2 - 12		t_{KCY2}/2 - 50		t_{KCY2}/2 - 50		t_{KCY2}/2 - 50		ns
		2.7 V ≤ EV_{DD} < 4.0 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 ≤ EV_{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}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		1/f_{MCK} + 20		1/f_{MCK} + 30		1/f_{MCK} + 30		1/f_{MCK} + 30		ns
		2.7 V ≤ EV_{DD} < 4.0 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 ≤ EV_{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 3	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 4]	t_{KSO2}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ			2/f_{MCK} + 120		2/f_{MCK} + 573		2/f_{MCK} + 573		2/f_{MCK} + 573	ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			2/f_{MCK} + 214		2/f_{MCK} + 573		2/f_{MCK} + 573		2/f_{MCK} + 573	ns
		1.8 V ≤ EV_{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, Caution** and **Remarks** are listed on the next page.)

Note 1. Transfer rate in the SNOOZE mode: MAX. 1 Mbps

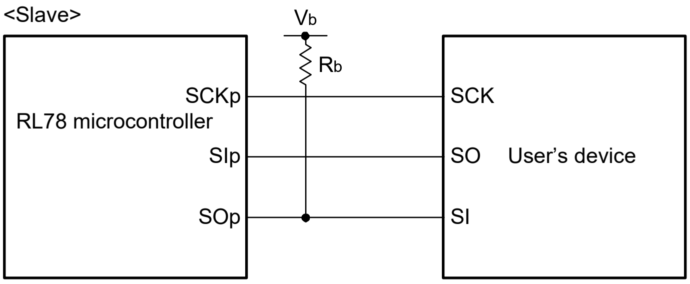
Note 2. Use it with $EV_{DD} \geq V_b$.

Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" and the SIp hold time becomes "from 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 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 (EV_{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.

Simplified SPI (CSI) mode connection diagram (during communication at different potential)

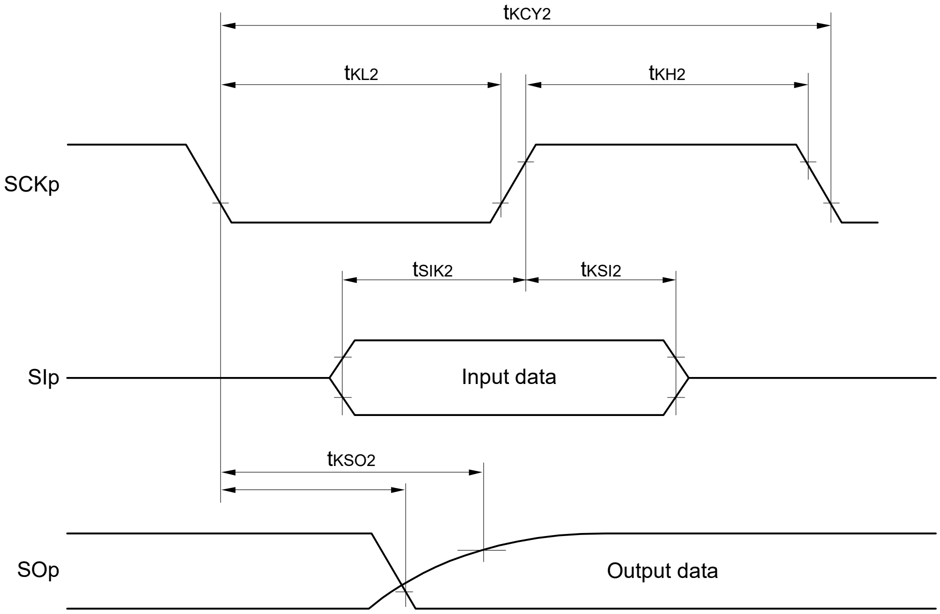


Remark 1. $R_b[\Omega]$: Communication line (SOp) pull-up resistance, $C_b[F]$: Communication line (SOp) load capacitance, $V_b[V]$: Communication line voltage

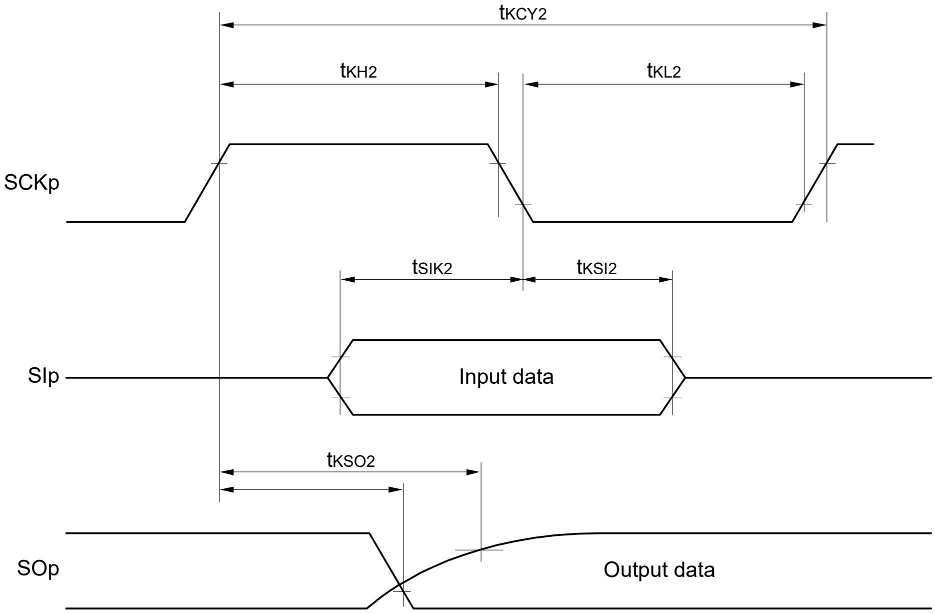
Remark 2. p: CSI number (p = 00 to 03), m: Unit number (m = 0), n: Channel number (n = 0 to 3), g: PIM and POM numbers (g = 0, 2, 3 to 5 and 12)

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 to 03))

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, 10 and 11), m: Unit number (m = 0), n: Channel number (n = 0 to 3), g: PIM and POM numbers (g = 0, 2, 3 to 5 and 12)

(10) Communication at different potential (1.8 V, 2.5 V, 3.0 V) (simplified I²C mode)

(T_A = -40 to 85°C, 1.8 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{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}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 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 ≤ EV_{DD} < 4.0 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
		4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ		400 Note 1		300 Note 1		250 Note 1		300 Note 1	kHz
		2.7 V ≤ EV_{DD} < 4.0 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 ≤ EV_{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}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	475		1550		1550		1550		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	475		1550		1550		1550		ns
		4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	1150		1550		1550		1550		ns
		2.7 V ≤ EV_{DD} < 4.0 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 ≤ EV_{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}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	245		610		610		610		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	200		610		610		610		ns
		4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	675		610		610		610		ns
		2.7 V ≤ EV_{DD} < 4.0 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 ≤ EV_{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)	$t_{SU:DAT}$	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	$1/f_{MCK}$ + 135 Note 3		$1/f_{MCK}$ + 190 Note 2		$1/f_{MCK}$ + 190 Note 3		$1/f_{MCK}$ + 190 Note 3		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	$1/f_{MCK}$ + 135 Note 3		$1/f_{MCK}$ + 190 Note 2		$1/f_{MCK}$ + 190 Note 3		$1/f_{MCK}$ + 190 Note 3		ns
		4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	$1/f_{MCK}$ + 190 Note 3		$1/f_{MCK}$ + 190 Note 3		$1/f_{MCK}$ + 190 Note 3		$1/f_{MCK}$ + 190 Note 3		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	$1/f_{MCK}$ + 190 Note 3		$1/f_{MCK}$ + 190 Note 3		$1/f_{MCK}$ + 190 Note 3		$1/f_{MCK}$ + 190 Note 3		ns
		1.8 V ≤ EV_{DD} < 4.0 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)	$t_{HD:DAT}$	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	0	305	0	305	0	305	0	305	ns
		2.7 V ≤ EV_{DD} < 4.0V, 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
		4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	0	355	0	355	0	355	0	355	ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	0	355	0	355	0	355	0	355	ns
		1.8 V ≤ EV_{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

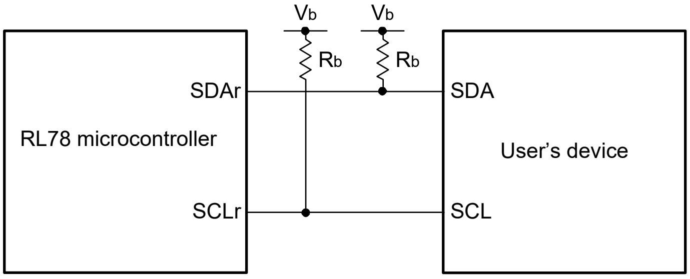
Note 1. The value must be equal to or less than $f_{MCK}/4$.

Note 2. Use it with $EV_{DD} \geq V_b$.

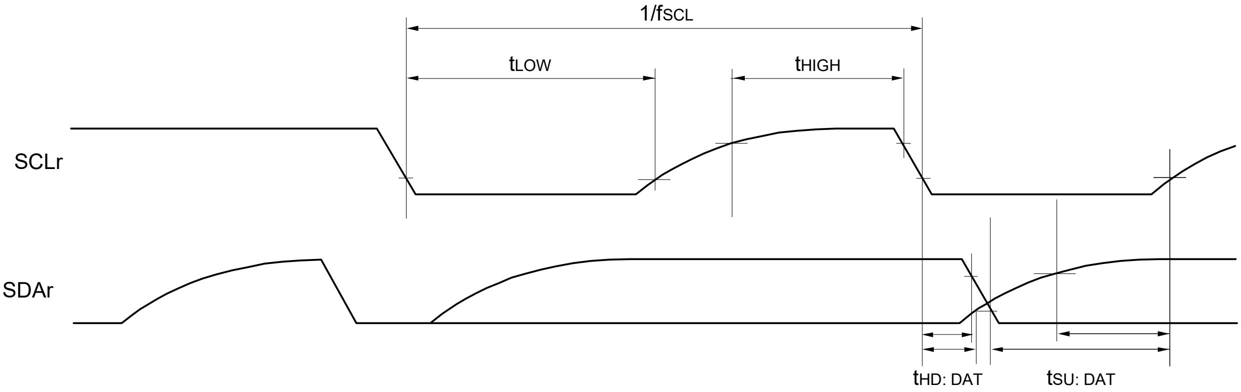
Note 3. Set the f_{MCK} value to keep the hold time of SCLr = "L" and SCLr = "H".

Caution **Select the TTL input buffer and the N-ch open drain output (EV_{DD} tolerance) mode for the SDAr pin and the N-ch open drain output (EV_{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.**

Simplified I²C mode connection diagram (during communication at different potential)



Simplified I²C mode serial transfer timing (during communication at different potential)



Remark 1. $R_b[\Omega]$: Communication line (SDAr, SCLr) pull-up resistance, $C_b[F]$: Communication line (SDAr, SCLr) load capacitance, $V_b[V]$: Communication line voltage

Remark 2. r: IIC number (r = 00, 01, 10 and 11), g: PIM, POM number (g = 0, 3 and 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 to 3), mn = 00 to 03)

35.5.2 Serial interface IICA

(1) I²C standard mode

(T_A = -40 to +85°C, 1.6 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{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.	
SCLA0 clock frequency	f_{SCL}	Standard mode: f_{CLK} ≥ 1 MHz	2.7 V ≤ EV_{DD} ≤ 5.5 V	0	100	0	100	0	100	0	100	kHz
			1.8 V ≤ EV_{DD} ≤ 5.5 V	0	100	0	100	0	100	0	100	kHz
			1.7 V ≤ EV_{DD} ≤ 5.5 V	0	100	0	100	0	100	0	100	kHz
			1.6 V ≤ EV_{DD} ≤ 5.5 V	—		0	100	0	100	0	100	kHz
Setup time of restart condition	$t_{SU: STA}$	2.7 V ≤ EV_{DD} ≤ 5.5 V		4.7		4.7		4.7		4.7		μs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		4.7		4.7		4.7		4.7		μs
		1.7 V ≤ EV_{DD} ≤ 5.5 V		4.7		4.7		4.7		4.7		μs
		1.6 V ≤ EV_{DD} ≤ 5.5 V		—		4.7		4.7		4.7		μs
Hold time Note 1	$t_{HD: STA}$	2.7 V ≤ EV_{DD} ≤ 5.5 V		4.0		4.0		4.0		4.0		μs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		4.0		4.0		4.0		4.0		μs
		1.7 V ≤ EV_{DD} ≤ 5.5 V		4.0		4.0		4.0		4.0		μs
		1.6 V ≤ EV_{DD} ≤ 5.5 V		—		4.0		4.0		4.0		μs
Hold time when SCLA0 = "L"	t_{LOW}	2.7 V ≤ EV_{DD} ≤ 5.5 V		4.7		4.7		4.7		4.7		μs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		4.7		4.7		4.7		4.7		μs
		1.7 V ≤ EV_{DD} ≤ 5.5 V		4.7		4.7		4.7		4.7		μs
		1.6 V ≤ EV_{DD} ≤ 5.5 V		—		4.7		4.7		4.7		μs
Hold time when SCLA0 = "H"	t_{HIGH}	2.7 V ≤ EV_{DD} ≤ 5.5 V		4.0		4.0		4.0		4.0		μs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		4.0		4.0		4.0		4.0		μs
		1.7 V ≤ EV_{DD} ≤ 5.5 V		4.0		4.0		4.0		4.0		μs
		1.6 V ≤ EV_{DD} ≤ 5.5 V		—		4.0		4.0		4.0		μs
Data setup time (reception)	$t_{SU: DAT}$	2.7 V ≤ EV_{DD} ≤ 5.5 V		250		250		250		250		ns
		1.8 V ≤ EV_{DD} ≤ 5.5 V		250		250		250		250		ns
		1.7 V ≤ EV_{DD} ≤ 5.5 V		250		250		250		250		ns
		1.6 V ≤ EV_{DD} ≤ 5.5 V		—		250		250		250		ns
Data hold time (transmission) Note 2	$t_{HD: DAT}$	2.7 V ≤ EV_{DD} ≤ 5.5 V		0	3.45	0	3.45	0	3.45	0	3.45	μs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		0	3.45	0	3.45	0	3.45	0	3.45	μs
		1.7 V ≤ EV_{DD} ≤ 5.5 V		0	3.45	0	3.45	0	3.45	0	3.45	μs
		1.6 V ≤ EV_{DD} ≤ 5.5 V		—		0	3.45	0	3.45	0	3.45	μs
Setup time of stop condition	$t_{SU: STO}$	2.7 V ≤ EV_{DD} ≤ 5.5 V		4.0		4.0		4.0		4.0		μs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		4.0		4.0		4.0		4.0		μs
		1.7 V ≤ EV_{DD} ≤ 5.5 V		4.0		4.0		4.0		4.0		μs
		1.6 V ≤ EV_{DD} ≤ 5.5 V		—		4.0		4.0		4.0		μs
Bus-free time	t_{BUF}	2.7 V ≤ EV_{DD} ≤ 5.5 V		4.7		4.7		4.7		4.7		μs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		4.7		4.7		4.7		4.7		μs
		1.7 V ≤ EV_{DD} ≤ 5.5 V		4.7		4.7		4.7		4.7		μs
		1.6 V ≤ EV_{DD} ≤ 5.5 V		—		4.7		4.7		4.7		μs

Note 1. The first clock pulse is generated after this period when the start/restart condition is detected.

Note 2. The maximum value (MAX.) of $t_{HD: DAT}$ is during normal transfer and a clock stretch state is inserted in the ACK (acknowledge) timing.

Caution **The values in the above table are applied even when bit 2 (PIOR02) in the peripheral I/O redirection register 0 (PIOR0) is 1. At this time, the pin characteristics (I_{OH1}, I_{OL1}, V_{OH1}, V_{OL1}) must satisfy the values in the redirect destination.**

Remark The maximum value of C_b (communication line capacitance) and the value of R_b (communication line pull-up resistor) at that time in each mode are as follows.

Standard mode: C_b = 400 pF, R_b = 2.7 kΩ

(2) I^2C fast mode

(T_A = -40 to +85°C, 1.6 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{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.	
SCLA0 clock frequency	f_{SCL}	Fast mode: f_{CLK} ≥ 3.5 MHz	2.7 V ≤ EV_{DD} ≤ 5.5 V	0	400	0	400	0	400	0	400	kHz
			1.8 V ≤ EV_{DD} ≤ 5.5 V	0	400	0	400	0	400	0	400	kHz
Setup time of restart condition	$t_{SU: STA}$	2.7 V ≤ EV_{DD} ≤ 5.5 V		0.6		0.6		0.6		0.6		μs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		0.6		0.6		0.6		0.6		μs
Hold time Note 1	$t_{HD: STA}$	2.7 V ≤ EV_{DD} ≤ 5.5 V		0.6		0.6		0.6		0.6		μs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		0.6		0.6		0.6		0.6		μs
Hold time when SCLA0 = “L”	t_{LOW}	2.7 V ≤ EV_{DD} ≤ 5.5 V		1.3		1.3		1.3		1.3		μs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		1.3		1.3		1.3		1.3		μs
Hold time when SCLA0 = “H”	t_{HIGH}	2.7 V ≤ EV_{DD} ≤ 5.5 V		0.6		0.6		0.6		0.6		μs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		0.6		0.6		0.6		0.6		μs
Data setup time (reception)	$t_{SU: DAT}$	2.7 V ≤ EV_{DD} ≤ 5.5 V		100		100		100		100		ns
		1.8 V ≤ EV_{DD} ≤ 5.5 V		100		100		100		100		ns
Data hold time (transmission) Note 2	$t_{HD: DAT}$	2.7 V ≤ EV_{DD} ≤ 5.5 V		0	0.9	0	0.9	0	0.9	0	0.9	μs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		0	0.9	0	0.9	0	0.9	0	0.9	μs
Setup time of stop condition	$t_{SU: STO}$	2.7 V ≤ EV_{DD} ≤ 5.5 V		0.6		0.6		0.6		0.6		μs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		0.6		0.6		0.6		0.6		μs
Bus-free time	t_{BUF}	2.7 V ≤ EV_{DD} ≤ 5.5 V		1.3		1.3		1.3		1.3		μs
		1.8 V ≤ EV_{DD} ≤ 5.5 V		1.3		1.3		1.3		1.3		μs

Note 1. The first clock pulse is generated after this period when the start/restart condition is detected.

Note 2. The maximum value (MAX.) of $t_{HD: DAT}$ is during normal transfer and a clock stretch state is inserted in the ACK (acknowledge) timing.

Caution The values in the above table are applied even when bit 2 (PIOR02) in the peripheral I/O redirection register 0 (PIOR0) is 1. At this time, the pin characteristics (I_{OH1}, I_{OL1}, V_{OH1}, V_{OL1}) must satisfy the values in the redirect destination.

Remark The maximum value of C_b (communication line capacitance) and the value of R_b (communication line pull-up resistor) at that time in each mode are as follows.

Fast mode: C_b = 320 pF, R_b = 1.1 kΩ

(3) I²C fast mode plus

(T_A = -40 to +85°C, 1.6 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{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.	
SCLA0 clock frequency	f_{SCL}	Fast mode plus: f_{CLK} ≥ 10 MHz	2.7 V ≤ EV_{DD} ≤ 5.5 V	0	1000	—		—		—		kHz
Setup time of restart condition	$t_{SU: STA}$	2.7 V ≤ EV_{DD} ≤ 5.5 V		0.26		—		—		—		μs
Hold time Note 1	$t_{HD: STA}$	2.7 V ≤ EV_{DD} ≤ 5.5 V		0.26		—		—		—		μs
Hold time when SCLA0 = "L"	t_{LOW}	2.7 V ≤ EV_{DD} ≤ 5.5 V		0.5		—		—		—		μs
Hold time when SCLA0 = "H"	t_{HIGH}	2.7 V ≤ EV_{DD} ≤ 5.5 V		0.26		—		—		—		μs
Data setup time (reception)	$t_{SU: DAT}$	2.7 V ≤ EV_{DD} ≤ 5.5 V		50		—		—		—		ns
Data hold time (transmission) Note 2	$t_{HD: DAT}$	2.7 V ≤ EV_{DD} ≤ 5.5 V		0	0.45	—		—		—		μs
Setup time of stop condition	$t_{SU: STO}$	2.7 V ≤ EV_{DD} ≤ 5.5 V		0.26		—		—		—		μs
Bus-free time	t_{BUF}	2.7 V ≤ EV_{DD} ≤ 5.5 V		0.5		—		—		—		μs

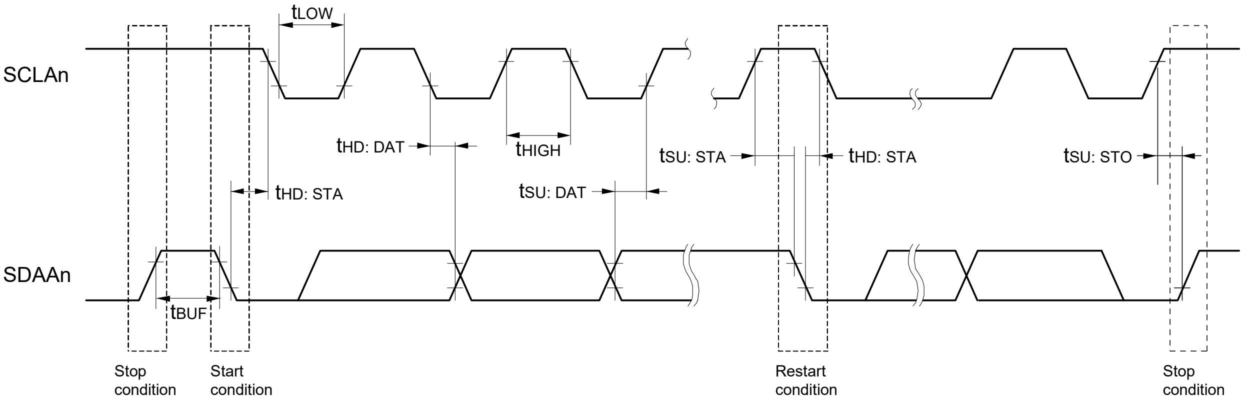
Note 1. The first clock pulse is generated after this period when the start/restart condition is detected.

Note 2. The maximum value (MAX.) of $t_{HD: DAT}$ is during normal transfer and a clock stretch state is inserted in the ACK (acknowledge) timing.

Caution The values in the above table are applied even when bit 2 (PIOR02) in the peripheral I/O redirection register 0 (PIOR0) is 1. At this time, the pin characteristics (I_{OH1}, I_{OL1}, V_{OH1}, V_{OL1}) must satisfy the values in the redirect destination.

Remark The maximum value of C_b (communication line capacitance) and the value of R_b (communication line pull-up resistor) at that time in each mode are as follows.
Fast mode plus: C_b = 120 pF, R_b = 1.1 kΩ

IICA serial transfer timing



Remark n = 0, 1

35.6 Analog Characteristics

35.6.1 A/D converter characteristics

Classification of A/D converter characteristics

Reference Voltage Input channel	Reference voltage (+) = AV_{REFP} Reference voltage (-) = AV_{REFM}	Reference voltage (+) = V_{DD} Reference voltage (-) = V_{SS}	Reference voltage (+) = V_{BGR} Reference voltage (-)= AV_{REFM}
ANI0 to ANI3	Refer to **35.6.1 (1)**.	Refer to **35.6.1 (3)**.	Refer to **35.6.1 (4)**.
ANI16 to ANI22	Refer to **35.6.1 (2)**.		
Internal reference voltage Temperature sensor output voltage	Refer to **35.6.1 (1)**.		—

(1) When reference voltage (+) = AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (-) = AV_{REFM}/ANI1 (ADREFM = 1), target pin: ANI2 and ANI3, internal reference voltage, and temperature sensor output voltage

(T_A = -40 to +85°C, 1.6 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = AV_{REFP}, Reference voltage (-) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error Note 1	AINL	10-bit resolution AV_{REFP} = V_{DD} Note 3	1.8 V ≤ AV_{REFP} ≤ 5.5 V		1.2	±3.5	LSB
			1.6 V ≤ AV_{REFP} ≤ 5.5 V Note 4		1.2	±7.0	LSB
Conversion time	t_{CONV}	10-bit resolution Target pin: ANI2 and ANI3	3.6 V ≤ V_{DD} ≤ 5.5 V	2.125		39	µs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	µs
			1.8 V ≤ V_{DD} ≤ 5.5 V	17		39	µs
			1.6 V ≤ V_{DD} ≤ 5.5 V	57		95	µs
		10-bit resolution Target pin: Internal reference voltage, and temperature sensor output voltage	3.6 V ≤ V_{DD} ≤ 5.5 V	2.375		39	µs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.5625		39	µs
			1.8 V ≤ V_{DD} ≤ 5.5 V	17		39	µs
Zero-scale error Notes 1, 2	E_{ZS}	10-bit resolution AV_{REFP} = V_{DD} Note 3	1.8 V ≤ AV_{REFP} ≤ 5.5 V			±0.25	%FSR
			1.6 V ≤ AV_{REFP} ≤ 5.5 V Note 4			±0.50	%FSR
Full-scale error Notes 1, 2	E_{FS}	10-bit resolution AV_{REFP} = V_{DD} Note 3	1.8 V ≤ AV_{REFP} ≤ 5.5 V			±0.25	%FSR
			1.6 V ≤ AV_{REFP} ≤ 5.5 V Note 4			±0.50	%FSR
Integral linearity error Note 1	ILE	10-bit resolution AV_{REFP} = V_{DD} Note 3	1.8 V ≤ AV_{REFP} ≤ 5.5 V			±2.5	LSB
			1.6 V ≤ AV_{REFP} ≤ 5.5 V Note 4			±5.0	LSB
Differential linearity error Note 1	DLE	10-bit resolution AV_{REFP} = V_{DD} Note 3	1.8 V ≤ AV_{REFP} ≤ 5.5 V			±1.5	LSB
			1.6 V ≤ AV_{REFP} ≤ 5.5 V Note 4			±2.0	LSB
Analog input voltage	V_{AIN}	ANI2 and ANI3		0		AV_{REFP}	V
		Internal reference voltage (1.8 V ≤ V_{DD} ≤ 5.5 V)		V_{BGR} Note 5			V
		Temperature sensor output voltage (1.8 V ≤ V_{DD} ≤ 5.5 V)		V_{TMPS25} Note 5			V

Note 1. Excludes quantization error (±1/2 LSB).

Note 2. This value is indicated as a ratio (%FSR) to the full-scale value.

Note 3. When AV_{REFP} < V_{DD}, the MAX. values are as follows.

Overall error:	Add ±1.0 LSB to the MAX. value when AV_{REFP} = V_{DD}.
Zero-scale error/Full-scale error:	Add ±0.05%FSR to the MAX. value when AV_{REFP} = V_{DD}.
Integral linearity error/ Differential linearity error:	Add ±0.5 LSB to the MAX. value when AV_{REFP} = V_{DD}.

Note 4. Values when the conversion time is set to 57 µs (min.) and 95 µs (max.).

Note 5. Refer to **35.6.2 Temperature sensor characteristics/internal reference voltage characteristic**.

(2) When reference voltage (+) = AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (-) = AV_{REFM}/ANI1 (ADREFM = 1), target pin: ANI16 to ANI22

(T_A = -40 to +85°C, 1.6 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, 1.6 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = AV_{REFP}, Reference voltage (-) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error [Note 1]	AINL	10-bit resolution EV_{DD} ≤ AV_{REFP} = V_{DD} [Notes 3, 4]	1.8 V ≤ AV_{REFP} ≤ 5.5 V		1.2	±5.0	LSB
			1.6 V ≤ AV_{REFP} ≤ 5.5 V [Note 5]		1.2	±8.5	LSB
Conversion time	t_{CONV}	10-bit resolution Target ANI pin: ANI16 to ANI22	3.6 V ≤ V_{DD} ≤ 5.5 V	2.125		39	µs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	µs
			1.8 V ≤ V_{DD} ≤ 5.5 V	17		39	µs
			1.6 V ≤ V_{DD} ≤ 5.5 V	57		95	µs
Zero-scale error [Notes 1, 2]	E_{ZS}	10-bit resolution EV_{DD} ≤ AV_{REFP} = V_{DD} [Notes 3, 4]	1.8 V ≤ AV_{REFP} ≤ 5.5 V			±0.35	%FSR
			1.6 V ≤ AV_{REFP} ≤ 5.5 V [Note 5]			±0.60	%FSR
Full-scale error [Notes 1, 2]	E_{FS}	10-bit resolution EV_{DD} ≤ AV_{REFP} = V_{DD} [Notes 3, 4]	1.8 V ≤ AV_{REFP} ≤ 5.5 V			±0.35	%FSR
			1.6 V ≤ AV_{REFP} ≤ 5.5 V [Note 5]			±0.60	%FSR
Integral linearity error [Note 1]	ILE	10-bit resolution EV_{DD} ≤ AV_{REFP} = V_{DD} [Notes 3, 4]	1.8 V ≤ AV_{REFP} ≤ 5.5 V			±3.5	LSB
			1.6 V ≤ AV_{REFP} ≤ 5.5 V [Note 5]			±6.0	LSB
Differential linearity error [Note 1]	DLE	10-bit resolution EV_{DD} ≤ AV_{REFP} = V_{DD} [Notes 3, 4]	1.8 V ≤ AV_{REFP} ≤ 5.5 V			±2.0	LSB
			1.6 V ≤ AV_{REFP} ≤ 5.5 V [Note 5]			±2.5	LSB
Analog input voltage	V_{AIN}	ANI16 to ANI22		0		AV_{REFP} and EV_{DD}	V

Note 1. Excludes quantization error (±1/2 LSB).

Note 2. This value is indicated as a ratio (%FSR) to the full-scale value.

Note 3. When EV_{DD} ≤ AV_{REFP} < V_{DD}, the MAX. values are as follows.

Overall error:	Add ±1.0 LSB to the MAX. value when AV_{REFP} = V_{DD}.
Zero-scale error/Full-scale error:	Add ±0.05%FSR to the MAX. value when AV_{REFP} = V_{DD}.
Integral linearity error/ Differential linearity error:	Add ±0.5 LSB to the MAX. value when AV_{REFP} = V_{DD}.

Note 4. When AV_{REFP} < EV_{DD} ≤ V_{DD}, the MAX. values are as follows.

Overall error:	Add ±4.0 LSB to the MAX. value when AV_{REFP} = V_{DD}.
Zero-scale error/Full-scale error:	Add ±0.20%FSR to the MAX. value when AV_{REFP} = V_{DD}.
Integral linearity error/ Differential linearity error:	Add ±2.0 LSB to the MAX. value when AV_{REFP} = V_{DD}.

Note 5. When the conversion time is set to 57 µs (min.) and 95 µs (max.).

(3) When reference voltage (+) = V_{DD} (ADREFP1 = 0, ADREFP0 = 0), reference voltage (-) = V_{SS} (ADREFM = 0), target pin: ANI0 to ANI3, ANI16 to ANI22, internal reference voltage, and temperature sensor output voltage

(T_A = -40 to +85°C, 1.6 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = V_{DD}, Reference voltage (-) = V_{SS})

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error Note 1	AINL	10-bit resolution	1.8 V ≤ V_{DD} ≤ 5.5 V		1.2	±7.0	LSB
			1.6 V ≤ V_{DD} ≤ 5.5 V Note 3		1.2	±10.5	LSB
Conversion time	t_{CONV}	10-bit resolution Target pin: ANI0 to ANI3, ANI16 to ANI22	3.6 V ≤ V_{DD} ≤ 5.5 V	2.125		39	µs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	µs
			1.8 V ≤ V_{DD} ≤ 5.5 V	17		39	µs
			1.6 V ≤ V_{DD} ≤ 5.5 V	57		95	µs
		10-bit resolution Target pin: internal reference voltage, and temperature sensor output voltage	3.6 V ≤ V_{DD} ≤ 5.5 V	2.375		39	µs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.5625		39	µs
			1.8 V ≤ V_{DD} ≤ 5.5 V	17		39	µs
Zero-scale error Notes 1, 2	E_{ZS}	10-bit resolution	1.8 V ≤ V_{DD} ≤ 5.5 V			±0.60	%FSR
			1.6 V ≤ V_{DD} ≤ 5.5 V Note 3			±0.85	%FSR
Full-scale error Notes 1, 2	E_{FS}	10-bit resolution	1.8 V ≤ V_{DD} ≤ 5.5 V			±0.60	%FSR
			1.6 V ≤ V_{DD} ≤ 5.5 V Note 3			±0.85	%FSR
Integral linearity error Note 1	ILE	10-bit resolution	1.8 V ≤ V_{DD} ≤ 5.5 V			±4.0	LSB
			1.6 V ≤ V_{DD} ≤ 5.5 V Note 3			±6.5	LSB
Differential linearity error Note 1	DLE	10-bit resolution	1.8 V ≤ V_{DD} ≤ 5.5 V			±2.0	LSB
			1.6 V ≤ V_{DD} ≤ 5.5 V Note 3			±2.5	LSB
Analog input voltage	V_{AIN}	ANI0 to ANI3		0		V_{DD}	V
		ANI16 to ANI22		0		EV_{DD}	V
		Internal reference voltage (1.8 V ≤ V_{DD} ≤ 5.5 V)		V_{BGR} Note 4			V
		Temperature sensor output voltage (1.8 V ≤ V_{DD} ≤ 5.5 V)		V_{TMPS25} Note 4			V

Note 1. Excludes quantization error (±1/2 LSB).

Note 2. This value is indicated as a ratio (% FSR) to the full-scale value.

Note 3. When the conversion time is set to 57 µs (min.) and 95 µs (max.).

Note 4. Refer to **35.6.2 Temperature sensor characteristics/internal reference voltage characteristic**.

(4) When reference voltage (+) = Internal reference voltage (ADREFP1 = 1, ADREFP0 = 0), reference voltage (-) = AVREFM/ANI1 (ADREFM = 1), target pin: ANI0, ANI2 and ANI3, ANI16 to ANI22

(TA = -40 to +85°C, 1.8 V ≤ VDD ≤ 5.5 V, 1.6 V ≤ EVDD ≤ VDD, VSS = 0 V, Reference voltage (+) = VBGR Note 3, Reference voltage (-) = AVREFM = 0 V Note 4)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	RES		8			bit
Conversion time	tCONV		17		39	µs
Zero-scale error Notes 1, 2	EZS				±0.60	% FSR
Integral linearity error Note 1	ILE				±2.0	LSB
Differential linearity error Note 1	DLE				±1.0	LSB
Analog input voltage	VAIN		0		VBGR Note 3	V

Note 1. Excludes quantization error (±1/2 LSB).

Note 2. This value is indicated as a ratio (% FSR) to the full-scale value.

Note 3. Refer to **35.6.2 Temperature sensor characteristics/internal reference voltage characteristic**.

Note 4. When reference voltage (-) = VSS, the MAX. values are as follows.

Zero-scale error:	Add ±0.35%FSR to the MAX. value when reference voltage (-) = AVREFM.
Integral linearity error:	Add ±0.5 LSB to the MAX. value when reference voltage (-) = AVREFM.
Differential linearity error:	Add ±0.2 LSB to the MAX. value when reference voltage (-) = AVREFM.

35.6.2 Temperature sensor characteristics/internal reference voltage characteristic

(T_A = -40 to +85°C, 1.8 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Temperature sensor output voltage	V_{TMPS25}	Setting ADS register = 80H, T_A = +25°C		1.05		V
Internal reference voltage	V_{BGR}	Setting ADS register = 81H	1.38	1.45	1.5	V
Temperature coefficient	F_{VTMPS}	Temperature sensor that depends on the temperature		-3.6		mV/°C
Operation stabilization wait time	t_{AMP}	2.4 V ≤ V_{DD} ≤ 5.5 V	5			µs
		1.8 V ≤ V_{DD} < 2.4 V	10			µs

35.6.3 D/A converter (channel 1)

(T_A = -40 to +85°C, 1.6 V ≤ EV_{SS} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES					8	bit
Overall error	AINL	Rload = 4 MΩ	1.8 V ≤ V_{DD} ≤ 5.5 V			±2.5	LSB
		Rload = 8 MΩ	1.8 V ≤ V_{DD} ≤ 5.5 V			±2.5	LSB
Settling time	t_{SET}	Cload = 20 pF	2.7 V ≤ V_{DD} ≤ 5.5 V			3	µs
			1.6 V ≤ V_{DD} < 2.7 V			6	µs

35.6.4 Comparator

(Comparator 0: T_A = -40 to +85°C, 2.7 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)
(Comparator 1: T_A = -40 to +85°C, 1.6 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage range	V_{IREF0}	IVREF0 pin		0		V_{DD} - 1.4 Note 1	V
	V_{IREF1}	IVREF1 pin		1.4 Note 1		V_{DD}	V
	V_{ICMP}	IVCMP0 pin		-0.3		V_{DD} + 0.3	V
		IVCMP1 pin		-0.3		EV_{DD} + 0.3	V
Output delay	td	V_{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		µs
			Comparator low-speed mode, window mode		4		µs
Operation stabilization wait time	t_{CMP}			100			µs
Reference voltage declination in channel 0 of internal DAC Note 2	⊿VIDAC					± 2.5	LSB

Note 1. In window mode, make sure that V_{REF1} - V_{REF0} ≥ 0.2 V.

Note 2. Only in CMP0

35.6.5 PGA

(T_A = -40 to +85°C, 2.7 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input offset voltage	V_{IOPGA}					±10	mV
Input voltage range	V_{IPGA}			0		0.9 × V_{DD}/Gain	V
Output voltage range	V_{IOHPGA}			0.93 × V_{DD}			V
	V_{IOLPGA}					0.07 × V_{DD}	V
Gain error		x4, x8				±1	%
		x16				±1.5	%
		x32				±2	%
Slew rate	SR_{RPGA}	Rising When V_{IN} = 0.1V_{DD}/gain to 0.9V_{DD}/gain. 10 to 90% of output voltage amplitude	4.0 V ≤ V_{DD} ≤ 5.5 V (Other than x32)	3.5			V/μs
			4.0 V ≤ V_{DD} ≤ 5.5 V (x32)	3.0			
			2.7 V ≤ V_{DD} ≤ 4.0V	0.5			
	SR_{FPGA}	Falling When V_{IN}= 0.1V_{DD}/gain to 0.9V_{DD}/gain. 90 to 10% of output voltage amplitude	4.0 V ≤ V_{DD} ≤ 5.5 V (Other than x32)	3.5			
			4.0 V ≤ V_{DD} ≤ 5.5 V (x32)	3.0			
			2.7 V ≤ V_{DD} ≤ 4.0V	0.5			
Reference voltage stabilization wait time[Note]	t_{PGA}	x4, x8				5	μs
		x16, x32				10	μs

Note Time required until a state is entered where the DC and AC specifications of the PGA are satisfied after the PGA operation has been enabled (PGAEN = 1).

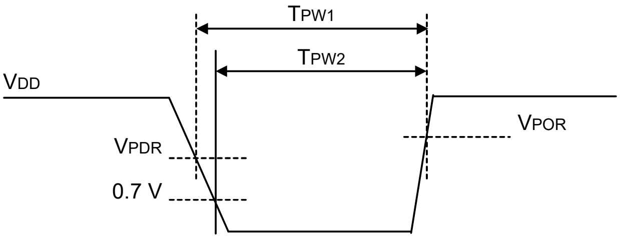
35.6.6 POR circuit characteristics

(T_A = -40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POR}	The power supply voltage is rising.	1.47	1.51	1.55	V
	V_{PDR}	The power supply voltage is falling. Note 1	1.46	1.50	1.54	V
Minimum pulse width Note 2	T_{PW1}	Other than STOP/SUB HALT/SUB RUN	300			μs
	T_{PW2}	STOP/SUB HALT/SUB RUN	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 35.4 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).



35.6.7 LVD circuit characteristics

(1) LVD Detection Voltage of Reset Mode and Interrupt Mode

(T_A = -40 to +85°C, $V_{PDR} \leq EV_{DD} \leq V_{DD} \leq$ 5.5 V, V_{SS} = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVD0}	The power supply voltage is rising.	3.98	4.06	4.14	V
			The power supply voltage is falling.	3.90	3.98	4.06	V
		V_{LVD1}	The power supply voltage is rising.	3.68	3.75	3.82	V
			The power supply voltage is falling.	3.60	3.67	3.74	V
		V_{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

(2) LVD Detection Voltage of Interrupt & Reset Mode

(T_A = -40 to +85°C, V_{PDR} ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{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_{LVDC3}		LVIS0, LVIS1 = 0, 0	Rising release reset voltage	3.68	3.75	3.82	V
				Falling interrupt voltage	3.60	3.67	3.74	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
	V_{LVDD3}		LVIS0, LVIS1 = 0, 0	Rising release reset voltage	3.98	4.06	4.14	V
				Falling interrupt voltage	3.90	3.98	4.06	V

35.6.8 Power supply voltage rising slope characteristics

(T_A = -40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Power supply voltage rising slope	S_{VDD}				54	V/ms

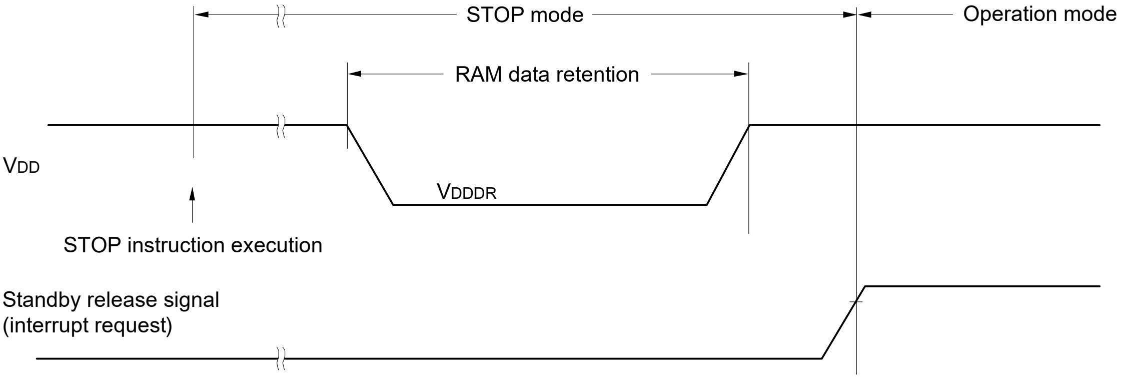
Caution **Make sure to keep the internal reset state by the LVD circuit or an external reset until V_{DD} reaches the operating voltage range shown in 35.4 AC Characteristics.**

35.7 RAM Data Retention Characteristics

(T_A = -40 to +85°C, 1.8 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		1.46 Note		5.5	V

Note The value depends on the POR detection voltage. When the voltage drops, the RAM data is retained before a POR reset is effected, but RAM data is not retained when a POR reset is effected.



35.8 Flash Memory Programming Characteristics

(T_A = -40 to +85°C, 1.8 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
System clock frequency	f_{CLK}	1.8 V ≤ V_{DD} ≤ 5.5 V		1		24	MHz
Number of code flash rewrites Notes 1, 2, 3	C_{erwr}	Retained for 20 years	T_A = 85°C	1,000			Times
Number of data flash rewrites Notes 1, 2, 3		Retained for 1 year	T_A = 25°C		1,000,000		
		Retained for 5 years	T_A = 85°C	100,000			
		Retained for 20 years	T_A = 85°C	10,000			

Note 1. 1 erase + 1 write after the erase is regarded as 1 rewrite. The retaining years are until next rewrite after the rewrite.

Note 2. 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.

35.9 Dedicated Flash Memory Programmer Communication (UART)

(T_A = -40 to +85°C, 1.8 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		During serial programming	115,200		1,000,000	bps

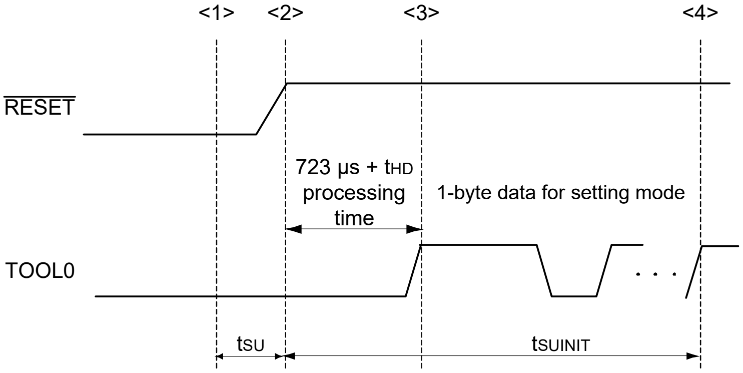
35.10 Timing of Entry to Flash Memory Programming Modes

(T_A = -40 to +85°C, 1.8 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{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 36 ELECTRICAL SPECIFICATIONS (T_A = -40 to +105°C)

This chapter describes the following electrical specifications.
Target products G: Industrial applications (T_A = −40 to +105°C)
R5F105xxGxx

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).

Caution 5. The EV_{DD} pin is not present on products with 24 or less pins. Accordingly, replace EV_{DD} with V_{DD} and the voltage condition 1.6 ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V with 1.6 ≤ V_{DD} ≤ 5.5 V.

Remark When the products “G: Industrial applications" is used in the range of T_A = -40 to +85°C, see **CHAPTER 35 ELECTRICAL SPECIFICATIONS (T_A = -40 to +85°C)**.

Fields of application	A: Consumer applications	G: Industrial applications
Operating ambient temperature	T_A = −40 to +85°C	T_A = −40 to +105°C
Operating mode Operating Voltage Range	HS (High-speed main) mode: 2.7 V ≤ V_{DD} ≤ 5.5 V @ 1 MHz to 24 MHz 2.4 V ≤ V_{DD} ≤ 5.5 V @ 1 MHz to 16 MHz LS (Low-speed main) mode: 1.8 V ≤ V_{DD} ≤ 5.5 V @ 1 MHz to 8 MHz LV (Low-voltage main) mode: 1.8 V ≤ V_{DD} ≤ 5.5 V @ 1 MHz to 4 MHz	Only in HS (High-speed main) mode: 2.7 V ≤ V_{DD} ≤ 5.5 V @ 1 MHz to 24 MHz 2.4 V ≤ V_{DD} ≤ 5.5 V @ 1 MHz to 16 MHz
High-speed on-chip oscillator clock to an accuracy	1.8 V ≤ V_{DD} ≤ 5.5 V: ±1.0% @ T_A = -20 to +85°C ±1.5% @ T_A = -40 to -20°C 1.6 V ≤ V_{DD} < 1.8 V: ±5.0% @ T_A = -20 to +85°C ±5.5% @ T_A = -40 to -20°C	2.4 V ≤ V_{DD} ≤ 5.5 V: ±2.0% @ T_A = +85 to +105°C ±1.0% @ T_A = -20 to +85°C ±1.5% @ T_A = -40 to -20°C
Serial array unit	UART Simplified SPI (CSI): f_{CLK}/2 (12 Mbps are supported), f_{CLK}/4 Simplified I^2C	UART Simplified SPI (CSI): f_{CLK}/4 Simplified I^2C
IICA	Standard mode Fast mode Fast mode plus	Standard mode Fast mode
Voltage Detector	• Rising: 1.67 V to 4.06 V (14 levels) • Falling: 1.63 V to 3.98 V (14 levels)	• Rising: 2.61 V to 4.06 V (8 levels) • Falling: 2.55 V to 3.98 V (8 levels)

Remark The electrical characteristics for "G: Industrial applications" differ from those for "A: Consumer applications" when the product is in use in an ambient temperature over 85°C. For details, see **36.1** to **36.10** in the following pages.

36.1 Absolute Maximum Ratings

(1/2)

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}		-0.5 to +6.5	V
	EV_{DD}		-0.5 to +6.5	V
	AV_{REFP}		0.3 to V_{DD} + 0.3 [Note 2]	V
	AV_{REFM}		-0.3 to V_{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, P01, P30 to P33, P40, and P51 to P56	-0.3 to EV_{DD} + 0.3 and -0.3 to V_{DD} + 0.3 [Note 2]	V
	V_{I2}	P20 to P23, P121, P122, P125, P137, EXCLK, $\overline{\text{RESET}}$	-0.3 to V_{DD} + 0.3 [Note 2]	V
Output voltage	V_{O1}	P00, P01, P30 to P33, P40, and P51 to P56	-0.3 to EV_{DD} + 0.3 and -0.3 to V_{DD} + 0.3 [Note 2]	V
	V_{O2}	P20 to P23	-0.3 to V_{DD} + 0.3 [Note 2]	V
Analog input voltage	V_{AI1}	ANI16 to ANI22	-0.3 to EV_{DD} + 0.3 and -0.3 to $AV_{REF}(+)$ + 0.3 [Notes 2, 3]	V
	V_{AI2}	ANI0 to ANI3	-0.3 to V_{DD} + 0.3 and -0.3 to $AV_{REF}(+)$ + 0.3 [Notes 2, 3]	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 6.5 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

(2/2)

Parameter	Symbols	Conditions		Ratings	Unit
Output current, high	I_{OH1}	Per pin	P00, P01, P30 to P33, P40, P51 to P56	-40	mA
		Total of all pins -170 mA	P00, P01, P40	-70	mA
			P30 to P33, P51 to P56	-100	mA
	I_{OH2}	Per pin	P20 to P23	-0.5	mA
		Total of all pins		-2	mA
Output current, low	I_{OL1}	Per pin	P00, P01, P30 to P33, P40, P51 to P56	40	mA
		Total of all pins 170 mA	P00, P01, P40	70	mA
			P30 to P33, P51 to P56	100	mA
	I_{OL2}	Per pin	P20 to P23	1	mA
		Total of all pins		4	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.

36.2 Oscillator Characteristics

36.2.1 X1 characteristics

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{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} ≤ 5.5 V	1.0		20.0	MHz
		2.4 V ≤ V_{DD} < 2.7 V	1.0		16.0	

Note Indicates only permissible oscillator frequency ranges. Refer to **36.4 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, refer to **6.4 System Clock Oscillator**.

36.2.2 On-chip oscillator characteristics

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Oscillators	Parameters	Conditions	MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator clock frequency Notes 1, 2	f_{IH}	2.7 V ≤ V_{DD} ≤ 5.5 V	1		24	MHz
		2.4 V ≤ V_{DD} ≤ 5.5 V	1		16	
High-speed on-chip oscillator clock frequency accuracy		T_A = +85°C to +105°C	-2		2	%
		T_A = -20°C to +85°C	-1		1	%
		T_A = -40°C to -20°C	-1.5		1.5	%
Middle-speed on-chip oscillator oscillation frequency Note 2	f_{IM}		1		4	MHz
Middle-speed on-chip oscillator oscillation frequency accuracy			-12		+12	%
Temperature drift of Middle-speed on-chip oscillator oscillation frequency accuracy	D_{IMT}			0.008		%/°C
Voltage drift of Middle-speed on-chip oscillator oscillation frequency accuracy	D_{IMV}	T_A = 25°C		0.02		%/V
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 **36.4 AC Characteristics** for instruction execution time.

36.3 DC Characteristics

36.3.1 Pin characteristics

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(1/5)**

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high Note 1	I_{OH1}	Per pin for P00, P01, P30 to P33, P40, and P51 to P56				-3.0 Note 2	mA
		Total of P00, P01, and P40 (When duty ≤ 70% Note 3)	4.0 V ≤ EV_{DD} ≤ 5.5 V			-12.5	mA
			2.7 V ≤ EV_{DD} < 4.0 V			-10.0	mA
			2.4 V ≤ EV_{DD} < 2.7 V			-5.0	mA
		Total of P30 to P33, and P51 to P56 (When duty ≤ 70% Note 3)	4.0 V ≤ EV_{DD} ≤ 5.5 V			-30.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			-19.0	mA
			2.4 V ≤ EV_{DD} < 2.7 V			-10.0	mA
		Total of all pins (When duty ≤ 70% Note 3)				-42.5	mA
	I_{OH2}	Per pin for P20 to P23				-0.1 Note 2	mA
		Total of all pins (When duty ≤ 70% Note 3)	2.4 V ≤ V_{DD} ≤ 5.5 V			-0.4	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 P00, P01, P20, P30 to P33, P40 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 +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) (2/5)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low Note 1	I_{OL1}	Per pin for P00, P01, P30 to P33, P40, and P51 to P56				8.5 Note 2	mA
		Total of P00, P01, and P40 (When duty ≤ 70% Note 3)	4.0 V ≤ EV_{DD} ≤ 5.5 V			36.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			15.0	mA
			2.4 V ≤ EV_{DD} < 2.7 V			9.0	mA
		Total of P30 to P33, and P51 to P56 (When duty ≤ 70% Note 3)	4.0 V ≤ EV_{DD} ≤ 5.5 V			40.0	mA
			2.7 V ≤ EV_{DD} < 4.0 V			35.0	mA
			2.4 V ≤ EV_{DD} < 2.7 V			20.0	mA
		Total of all pins (When duty ≤ 70% Note 3)				76.0	mA
	I_{OL2}	Per pin for P20 to P23				0.4 Note 2	mA
		Total of all pins (When duty ≤ 70% Note 3)	2.4 V ≤ V_{DD} ≤ 5.5 V			1.6	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 +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(3/5)**

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P00, P01, P30 to P33, P40, and P51 to P56	Normal mode	0.8 EV_{DD}		EV_{DD}	V
	V_{IH2}	P00, P30 to P32, P40, P51 to P56	TTL mode 4.0 V ≤ EV_{DD} ≤ 5.5 V	2.2		EV_{DD}	V
			TTL mode 3.3 V ≤ EV_{DD} < 4.0 V	2.0		EV_{DD}	V
			TTL mode 2.4 V ≤ EV_{DD} < 3.3 V	1.5		EV_{DD}	V
	V_{IH3}	P20 to P23 (digital input)		0.7 V_{DD}		V_{DD}	V
	V_{IH4}	P121, P122, P125, P137, EXCLK, $\overline{\text{RESET}}$		0.8 V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	P00, P01, P30 to P33, P40, and P51 to P56	Normal mode	0		0.2 EV_{DD}	V
	V_{IL2}	P00, P30 to P32, P40, P51 to P56	TTL mode 4.0 V ≤ EV_{DD} ≤ 5.5 V	0		0.8	V
			TTL mode 3.3 V ≤ EV_{DD} < 4.0 V	0		0.5	V
			TTL mode 2.4 V ≤ EV_{DD} < 3.3 V	0		0.32	V
	V_{IH3}	P20 to P23 (digital input)		0		0.3 V_{DD}	V
	V_{IH4}	P121, P122, P125, P137, EXCLK, $\overline{\text{RESET}}$		0		0.2 V_{DD}	V

Caution **The maximum value of V_{IH} of pins P00, P01, P20, P30 to P33, P40 and P51 to P56 is V_{DD} or EV_{DD}, even in the N-ch open-drain mode.**
(P20: V_{DD}
P00, P01, P30 to P33, P40, P51 to P56: EV_{DD})

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(4/5)**

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, high	V_{OH1}	P00, P01, P30 to P33, P40, and P51 to P56	4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OH} = -3.0 mA	EV_{DD} - 0.7			V
			2.7 V ≤ EV_{DD} ≤ 5.5 V, I_{OH} = -2.0 mA	EV_{DD} - 0.6			V
			2.4 V ≤ EV_{DD} ≤ 5.5 V I_{OH} = -1.5 mA	EV_{DD} - 0.5			V
	V_{OH2}	P20 to P23	2.4 V ≤ V_{DD} ≤ 5.5 V, I_{OH} = -100 μA	V_{DD} - 0.5			V
Output voltage, low	V_{OL1}	P00, P01, P30 to P33, P40, and P51 to P56	4.0 V ≤ EV_{DD} ≤ 5.5 V, I_{OL} = 8.5 mA			0.7	V
			2.7 V ≤ EV_{DD} ≤ 5.5 V, I_{OL} = 3.0 mA			0.6	V
			2.7 V ≤ EV_{DD} ≤ 5.5 V, I_{OL} = 1.5 mA			0.4	V
			2.4 V ≤ EV_{DD} ≤ 5.5 V, I_{OL} = 0.6 mA			0.4	V
	V_{OL2}	P20 to P23	2.4 V ≤ V_{DD} ≤ 5.5 V, I_{OL} = 400 μA			0.4	V

Caution P00, P01, P20, P30 to P33, P40 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 +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(5/5)**

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P00, P01, P30 to P33, P40, and P51 to P56	$V_I = EV_{DD}$				1	µA
	I_{LIH2}	P20 to P23, P125, P137, $\overline{\text{RESET}}$	$V_I = V_{DD}$				1	µA
	I_{LIH3}	P121, P122, X1, X2, EXCLK	$V_I = V_{DD}$	In input port or external clock input			1	µA
				In resonator connection			10	µA
Input leakage current, low	I_{LIL1}	P00, P01, P30 to P33, P40, and P51 to P56	$V_I = V_{SS}$				-1	µA
	I_{LIL2}	P20 to P23, P125, P137, $\overline{\text{RESET}}$	$V_I = V_{SS}$				-1	µA
	I_{LIL3}	P121, P122, X1, X2, EXCLK	$V_I = V_{SS}$	In input port or external clock input			-1	µA
				In resonator connection			-10	µA
On-chip pull-up resistance	R_U	P00, P01, P30 to P33, P40, P51 to P56, P125	$V_I = V_{SS}$, 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.

36.3.2 Supply current characteristics

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(1/3)**

Parameter	Symbol	Conditions						MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD1}	Operating mode	Basic operation	HS (high-speed main) mode	f_{HOCO} = 48 MHz Note 3 f_{IH} = 24 MHz Note 3	V_{DD} = 5.0 V			1.7		mA
						V_{DD} = 3.0 V			1.7		
					f_{HOCO} = 24 MHz Note 3 f_{IH} = 24 MHz Note 3	V_{DD} = 5.0 V			1.4		
						V_{DD} = 3.0 V			1.4		
			Normal operation	HS (high-speed main) mode	f_{HOCO} = 48 MHz Note 3 f_{IH} = 24 MHz Note 3	V_{DD} = 5.0 V			3.5	7.3	mA
						V_{DD} = 3.0 V			3.5	7.3	
					f_{HOCO} = 24 MHz Note 3 f_{IH} = 24 MHz Note 3	V_{DD} = 5.0 V			3.2	6.7	
						V_{DD} = 3.0 V			3.2	6.7	
					f_{HOCO} = 16 MHz Note 3 f_{IH} = 16 MHz Note 3	V_{DD} = 5.0 V			2.4	4.9	
						V_{DD} = 3.0 V			2.4	4.9	
			Normal operation	HS (high-speed main) mode	f_{MX} = 20 MHz Note 2	V_{DD} = 5.0 V	Square wave input		2.7	5.7	mA
							Resonator connection		2.8	5.8	
						V_{DD} = 3.0 V	Square wave input		2.7	5.7	
							Resonator connection		2.8	5.8	
					f_{MX} = 10 MHz Note 2	V_{DD} = 5.0 V	Square wave input		1.8	3.4	
							Resonator connection		1.9	3.5	
						V_{DD} = 3.0 V	Square wave input		1.8	3.4	
							Resonator connection		1.9	3.5	
			Normal operation	Subsystem clock operation	f_{IL} = 15 kHz, T_A = -40°C Note 4				1.8	5.9	μA
					f_{IL} = 15 kHz, T_A = +25°C Note 4				1.9	5.9	
					f_{IL} = 15 kHz, T_A = +85°C Note 4				2.3	8.7	
					f_{IL} = 15 kHz, T_A = +105°C Note 4				3.0	20.9	

Note 1. Total current flowing into V_{DD} and EV_{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) 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, D/A converter, comparator, programmable gain amplifier, 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.

Note 2. When the high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock and low-speed on-chip oscillator clock are stopped.

Note 3. When the high-speed system clock, middle-speed on-chip oscillator clock and low-speed on-chip oscillator clock are stopped.

Note 4. When the high-speed system clock, high-speed on-chip oscillator clock and middle-speed on-chip oscillator 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_{SUB}: Subsystem clock frequency (Low-speed on-chip oscillator clock frequency)

Remark 6. Except subsystem clock operation, temperature condition of the TYP. value is T_A = 25°C

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(2/3)**

Parameter	Symbol	Conditions					MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD2} Note 2	HALT mode	HS (high-speed main) mode	f_{HOCO} = 48 MHz Note 3	V_{DD} = 5.0 V			0.59	3.45	mA
				f_{IH} = 24 MHz Note 4	V_{DD} = 3.0 V			0.59	3.45	
				f_{HOCO} = 24 MHz Note 3	V_{DD} = 5.0 V			0.41	2.85	
				f_{IH} = 16 MHz Note 4	V_{DD} = 3.0 V			0.41	2.85	
				f_{HOCO} = 16 MHz Note 3	V_{DD} = 5.0 V			0.39	2.08	
				f_{IH} = 16 MHz Note 4	V_{DD} = 3.0 V			0.39	2.08	
			HS (high-speed main) mode	f_{MX} = 20 MHz Note 3	V_{DD} = 5.0 V	Square wave input		0.20	2.45	mA
						Resonator connection		0.40	2.57	
					V_{DD} = 3.0 V	Square wave input		0.20	2.45	
						Resonator connection		0.40	2.57	
				f_{MX} = 10 MHz Note 3	V_{DD} = 5.0 V	Square wave input		0.15	1.28	
						Resonator connection		0.30	1.36	
					V_{DD} = 3.0 V	Square wave input		0.15	1.28	
						Resonator connection		0.30	1.36	
			Subsystem clock operation	f_{IL} = 15 kHz, T_A = -40°C Note 5				0.48	1.22	µA
				f_{IL} = 15 kHz, T_A = +25°C Note 5				0.55	1.22	
				f_{IL} = 15 kHz, T_A = +85°C Note 5				0.80	3.30	
				f_{IL} = 15 kHz, T_A = +105°C Note 5				2.00	17.3	

Note 1. Total current flowing into V_{DD} and EV_{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) 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, D/A converter, comparator, programmable gain amplifier, 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.

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, and low-speed on-chip oscillator clock are stopped.

Note 4. When the high-speed system clock, middle-speed on-chip oscillator clock and low-speed on-chip oscillator clock are stopped.

Note 5. When the high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock and high-speed system 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_{SUB}: Subsystem clock frequency (Low-speed on-chip oscillator clock frequency)

Remark 6. Except subsystem clock operation, temperature condition of the TYP. value is T_A = 25°C

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(3/3)**

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD3}	STOP mode Note 2	T_A = -40°C		0.19	0.51	μA
			T_A = +25°C		0.25	0.51	
			T_A = +50°C		0.28	1.10	
			T_A = +70°C		0.38	1.90	
			T_A = +85°C		0.60	3.30	
			T_A = +105°C		1.5	17.0	

Note 1. Total current flowing into V_{DD} and EV_{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.

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 +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Low-speed on-chip oscillator operating current	I_{FIL} Note 1				0.22		μA
12-bit interval timer operating current	I_{TMKA} Notes 1, 3, 4	f_{IL} = 15 kHz f_{MAIN} stopped (per unit)			0.02		μA
8-bit interval timer operating current Notes 1, 9	I_{TMT}	f_{IL} = 15 kHz f_{MAIN} stopped (per unit)	8-bit counter mode × 2-channel operation		0.04		μA
			16-bit counter mode operation		0.03		μA
Watchdog timer operating current	I_{WDT} Notes 1, 3, 5	f_{IL} = 15 kHz f_{MAIN} stopped (per unit)			0.22		μA
A/D converter operating current	I_{ADC} Notes 1, 6	During maximum-speed conversion	Normal mode, AV_{VREFP} = V_{DD} = 5.0 V		1.3	1.7	mA
			Low voltage mode, AV_{VREFP} = V_{DD} = 3.0 V		0.5	0.7	mA
Internal reference voltage (1.45 V) current Notes 1, 10	I_{ADREF}				85.0		μA
Temperature sensor operating current	I_{TMPS} Note 1				85.0		μA
D/A converter operating current	I_{DAC} Note 1	Per channel				1.5	mA
PGA operating current	I_{PGA} Notes 1, 2				480	700	μA
Comparator operating current	I_{CMP} Note 8	V_{DD} = 5.0 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		1.9		
		V_{DD} = 5.0 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		
LVD operating current	I_{LVD} Notes 1, 7				0.10		μA
Self-programming operating current	IFSP Notes 1, 12				2.0	12.20	mA
BGO current	IBGO Notes 1, 11				2.0	12.20	mA
SNOOZE operating current	ISNOZ Note 1	ADC operation f_{IH} = 24 MHz, AVREFP = V_{DD} = 3.0 V	Mode transition Note 13		0.50	1.10	mA
			The A/D conversion operations are performed		1.20	2.04	mA
		Simplified SPI (CSI)/UART operation f_{IH} = 24 MHz			0.70	1.54	mA
	ISNOZM Note 1	ADC operation f_{IM} = 4 MHz, AVREFP = V_{DD} = 3.0 V	Mode transition Note 13		0.05	0.13	mA
			The A/D conversion operations are performed		0.67	0.84	mA
		Simplified SPI (CSI) operation, f_{IM} = 4 MHz			0.06	0.15	mA

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

Note 1. Current flowing to V_{DD}.

Note 2. Operable range is 2.7 to 5.5 V.

Note 3. When the high-speed on-chip oscillator clock, middle-speed on-chip oscillator clock, and high-speed system clock are stopped.

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 consumed by generating the internal reference voltage (1.45 V).

Note 11. Current flowing during programming of the data flash.

Note 12. Current flowing during self-programming.

Note 13. For transition time to the SNOOZE mode, see **24.3.3 SNOOZE mode**.

Remark 1. f_{IL}: Low-speed on-chip oscillator clock frequency

Remark 2. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 3. Temperature condition of the TYP. value is T_A = 25°C

36.4 AC Characteristics

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

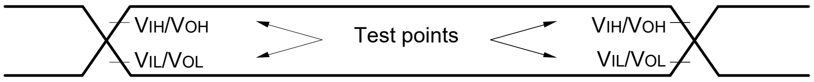
Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation	HS (high-speed main) mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.04167		1	μs
				2.4 V ≤ V_{DD} < 2.7 V	0.0625		1	μs
		Subsystem clock (f_{SUB}) operation	f_{IL}	2.4 V ≤ V_{DD} ≤ 5.5 V		66.7		μs
		In the self-programming mode	HS (high-speed main) mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.04167		1	μs
				2.4 V ≤ V_{DD} < 2.7 V	0.0625		1	μs
External system clock frequency	f_{EX}	2.7 V ≤ V_{DD} ≤ 5.5 V			1		20	MHz
		2.4 V ≤ V_{DD} < 2.7 V			1		16	MHz
External system clock input high-/low-level width	t_{EXH}, t_{EXL}	2.7 V ≤ V_{DD} ≤ 5.5 V			24			ns
		2.4 V ≤ V_{DD} < 2.7 V			30			ns
TI00 to TI03 input high-/low-level width	t_{TIH}, t_{TIL} Note 1				1/f_{MCK} + 10			ns
TO00 to TO03, TKBO0, and TKBO1 output frequency Note 2	f_{TO}	TO00 to TO03, TKBO0, and TKBO1 (in the case of output from port pins other than P20)	HS (high-speed main) mode	4.0 V ≤ EV_{DD} ≤ 5.5 V			12	MHz
				2.7 V ≤ EV_{DD} < 4.0 V			8	
				2.4 V ≤ EV_{DD} < 2.7 V			4	
		TKBO1 (in the case of output from P20)	HS (high-speed main) mode	4.0 V ≤ V_{DD} ≤ 5.5 V			1.5	MHz
				2.7 V ≤ V_{DD} < 4.0 V			1.2	
				2.4 V ≤ V_{DD} < 2.7 V			1	
PCLBUZ0, PCLBUZ1 output frequency	f_{PCL}	HS (high-speed main) mode		4.0 V ≤ EV_{DD} ≤ 5.5 V			16	MHz
				2.7 V ≤ EV_{DD} < 4.0 V			8	
				2.4 V ≤ EV_{DD} < 2.7 V			4	
Interrupt input high-/low-level width	t_{INTH}, t_{INTL}	INTP0 to INTP2, INTP9		2.4 V ≤ V_{DD} ≤ 5.5 V	1			μs
		INTP3 to INTP8, INTP10, INTP11		2.4 V ≤ EV_{DD} ≤ 5.5 V	1			
Key interrupt input low-level width	t_{KR}	KR0 to KR7		2.4 V ≤ EV_{DD} ≤ 5.5 V	250			ns
$\overline{\text{RESET}}$ low-level width	t_{RSL}				10			μs

Note 1. Following conditions must be satisfied on low level interface of EV_{DD} < V_{DD}.
2.4 V ≤ EV_{DD} ≤ 2.7 V: MIN.125 ns

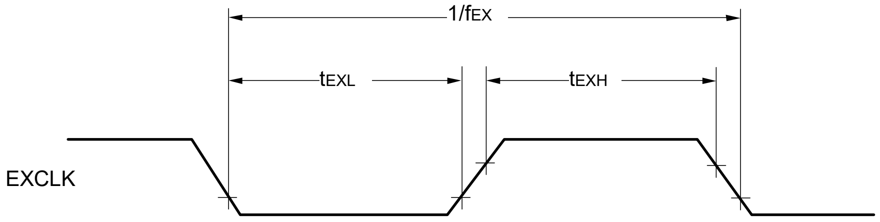
Note 2. When duty is 50%.

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))

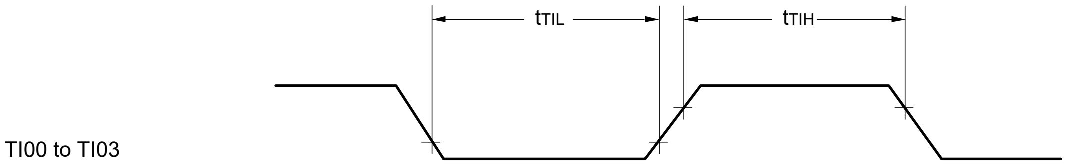
AC Timing Test Points

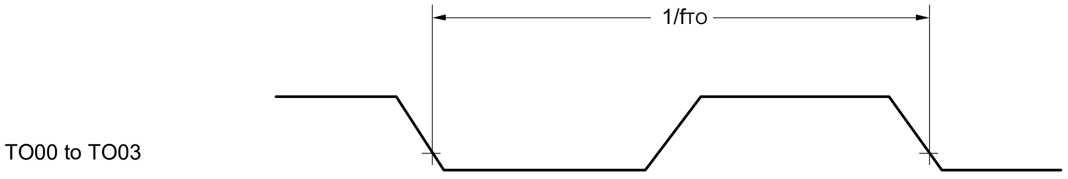


External System Clock Timing

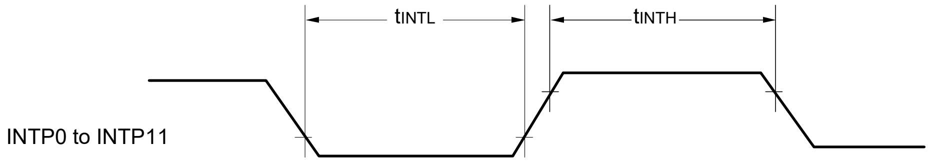


TI/TO Timing

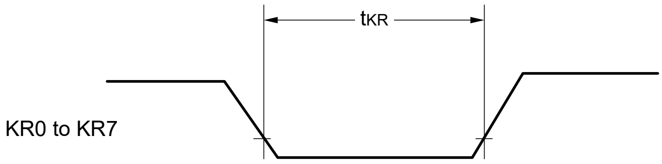




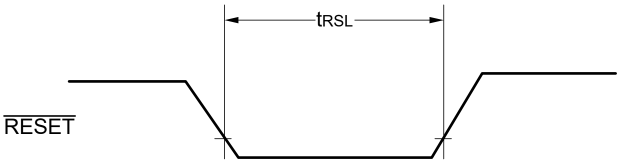
Interrupt Request Input Timing



Key Interrupt Input Timing

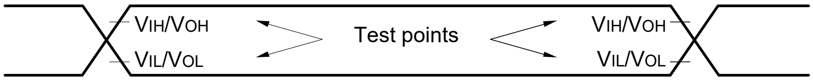


$\overline{\text{RESET}}$ Input Timing



36.5 Peripheral Functions Characteristics

AC Timing Test Points



36.5.1 Serial array unit

(1) during communication at same potential (UART mode)
When P01, P30, P31 and P54 are used as TxDq pin

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
Transfer rate		Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} = 24 MHz		$f_{MCK}/12$ Notes 1, 2	bps
				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:
2.4 V ≤ EV_{DD} ≤ 2.7 V: MAX. 1.3 Mbps

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).**

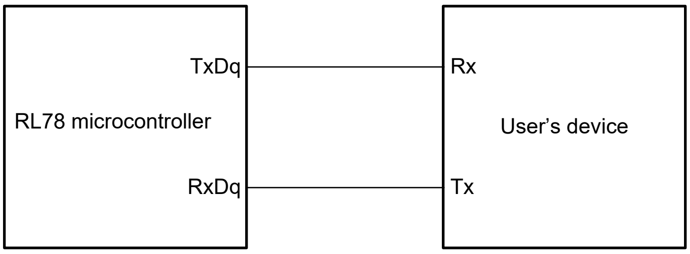
When P20 is used as TxD1 pin

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

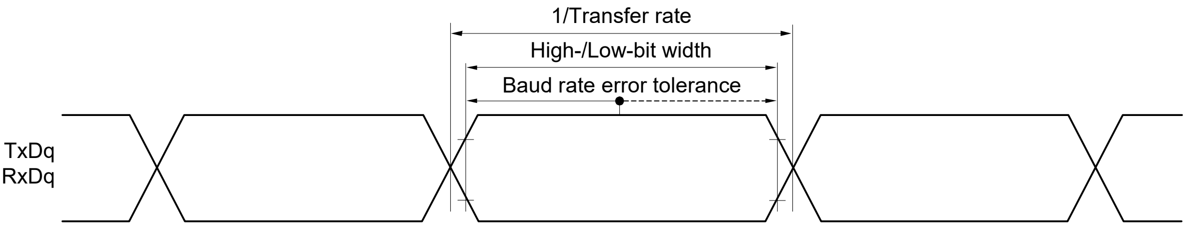
Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
Transfer rate		4.0 V ≤ V_{DD} ≤ 5.5 V		$f_{MCK}/16$ Note	bps
		Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} = 24 MHz		1.5	Mbps
		2.7 V ≤ V_{DD} ≤ 5.5 V		$f_{MCK}/20$ Note	bps
		Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} = 24 MHz		1.2	Mbps
		2.4 V ≤ V_{DD} ≤ 5.5 V		$f_{MCK}/16$ Note	bps
		Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} = 16 MHz		1.0	Mbps

Note Transfer rate in the SNOOZE mode is 4800 bps only. When f_{HOCO} = 48 MHz, SNOOZE mode is not supported.

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 and 1), g: PIM and POM number (g = 0, 2, 3 and 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 to 03)

(2) During communication at same potential (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

When P01, P32, P53, P54 and P56 are used as SOmn pins

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ 4/f_{CLK}	2.7 V ≤ EV_{DD} ≤ 5.5 V	250		ns
			2.4 V ≤ EV_{DD} ≤ 5.5 V	500		ns
SCKp high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ EV_{DD} ≤ 5.5 V		t_{KCY1}/2 - 24		ns
		2.7 V ≤ EV_{DD} ≤ 5.5 V		t_{KCY1}/2 - 36		ns
		2.4 V ≤ EV_{DD} ≤ 5.5 V		t_{KCY1}/2 - 76		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	4.0 V ≤ EV_{DD} ≤ 5.5 V		66		ns
		2.7 V ≤ EV_{DD} ≤ 5.5 V				ns
		2.4 V ≤ EV_{DD} ≤ 5.5 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, 01, 10 and 11), m: Unit number (m = 0), n: Channel number (n = 0 to 3), g: PIM and POM numbers (g = 0, 2, 3 to 5 and 12)

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 to 03))

When P20 is used as SO10 pin

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ 4/f_{CLK}	2.7 V ≤ V_{DD} ≤ 5.5 V	1000		ns
			2.4 V ≤ V_{DD} ≤ 5.5 V	1200		ns
SCKp high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2 - 24		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2 - 36		ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2 - 76		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	2.7 V ≤ V_{DD} ≤ 5.5 V		66		ns
		2.4 V ≤ V_{DD} ≤ 5.5 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			180	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, 01, 10 and 11), m: Unit number (m = 0), n: Channel number (n = 0 to 3), g: PIM and POM numbers (g = 0, 2, 3 to 5 and 12)

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 to 03))

(3) During communication at same potential (Simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

When P01, P32, P53, P54 and P56 are used as SOmn pins

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
SCKp cycle time Note 4	t_{KCY2}	4.0 V ≤ EV_{DD} ≤ 5.5 V	f_{MCK} > 20 MHz	16/f_{MCK}		ns
			f_{MCK} ≤ 20 MHz	12/f_{MCK}		ns
		2.7 V ≤ EV_{DD} < 4.0 V	f_{MCK} > 16 MHz	16/f_{MCK}		ns
			f_{MCK} ≤ 16 MHz	12/f_{MCK}		ns
		2.4 V ≤ EV_{DD} < 2.7 V		12/f_{MCK} and 1000		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ EV_{DD} ≤ 5.5 V		t_{KCY2}/2 - 14		ns
	t_{KH2}, t_{KL2}	2.7 V ≤ EV_{DD} < 4.0 V		t_{KCY2}/2 - 16		ns
		2.4 V ≤ EV_{DD} < 2.7 V		t_{KCY2}/2 - 36		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK2}	2.7 V ≤ EV_{DD} ≤ 5.5 V		1/f_{MCK} + 40		ns
		2.4 V ≤ EV_{DD} < 2.7 V		1/f_{MCK} + 60		ns
SIp hold time (from SCKp↑) Note 1	t_{KSI2}			1/f_{MCK} + 62		ns
Delay time from SCKp↓ to SOp output Note 2	t_{KSO2}	C = 30 pF Note 3	2.7 V ≤ EV_{DD} ≤ 5.5 V		2/f_{MCK} + 66	ns
			2.4 V ≤ EV_{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↓” and the SIp hold time becomes “from 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 delay time to SOp output becomes “from SCKp↑” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. C is the load capacitance of the SOp output lines.

Note 4. 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, 01, 10 and 11), m: Unit number (m = 0), n: Channel number (n = 0 to 3), g: PIM and POM numbers (g = 0, 2, 3 to 5 and 12)

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 to 03))

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(2/2)**

Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
$\overline{\text{SSI00}}$ 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
$\overline{\text{SSI00}}$ 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, 01, 10 and 11), m: Unit number (m = 0), n: Channel number (n = 0 to 3), g: PIM and POM numbers (g = 0, 2, 3 to 5, 12)

When P20 is used as SO10 pin

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} = V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
SCKp cycle time [Note 4]	t_{KCY2}	4.0 V ≤ V_{DD} ≤ 5.5 V	f_{MCK} > 20 MHz	20/f_{MCK}		ns
			f_{MCK} ≤ 20 MHz	18/f_{MCK}		ns
		2.7 V ≤ V_{DD} < 4.0 V	f_{MCK} > 16 MHz	20/fMCK and 1000		ns
			f_{MCK} ≤ 16 MHz	18/f_{MCK}		ns
		2.4 V ≤ V_{DD} < 2.7 V		18/f_{MCK} and 1200		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ V_{DD} ≤ 5.5 V		t_{KCY2}/2 - 14		ns
	t_{KH2}, t_{KL2}	2.7 V ≤ V_{DD} < 4.0 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} ≤ 5.5 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 1]	t_{KSI2}			1/f_{MCK} + 62		ns
Delay time from SCKp↓ to SOp output [Note 2]	t_{KSO2}	C = 30 pF [Note 3]	2.7 V ≤ V_{DD} ≤ 5.5 V		2/f_{MCK} + 190	ns
			2.4 V ≤ V_{DD} < 2.7 V		2/f_{MCK} + 250	ns

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" and the SIp hold time becomes "from 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 delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. C is the load capacitance of the SOp output lines.

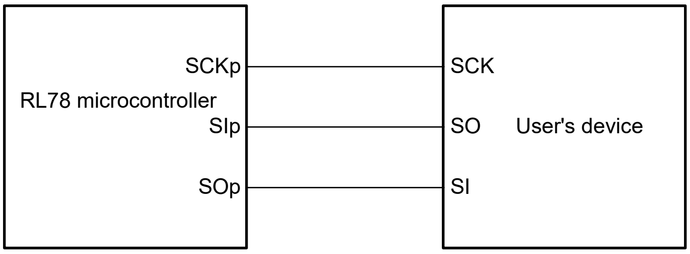
Note 4. The maximum transfer rate when using the SNOOZE mode is 1 Mbps.

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).**

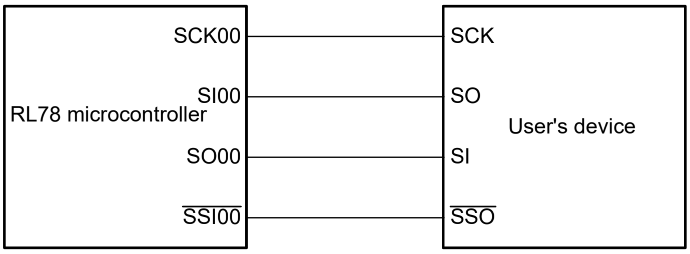
Remark 1. p: CSI number (p = 00, 01, 10 and 11), m: Unit number (m = 0), n: Channel number (n = 0 to 3), g: PIM and POM numbers (g = 0, 2, 3 to 5 and 12)

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 to 03))

Simplified SPI (CSI) mode connection diagram (during communication at same potential)

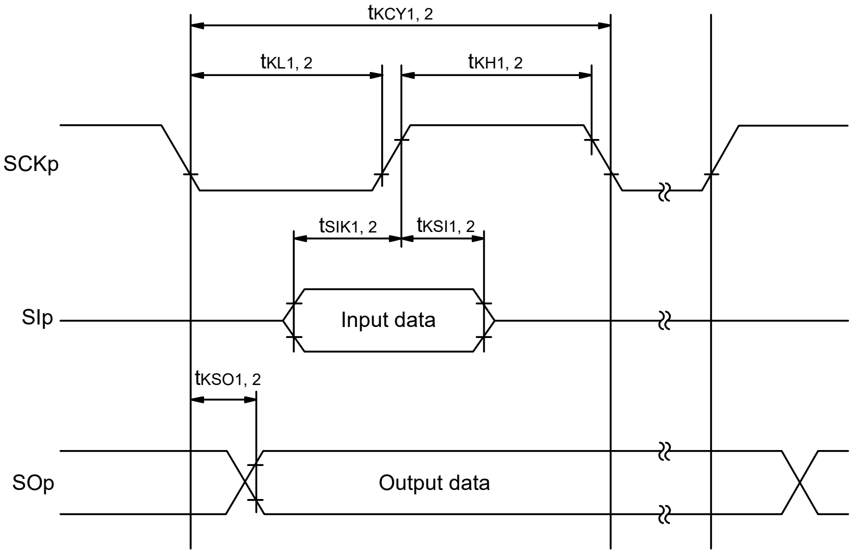


Simplified SPI (CSI) mode connection diagram (during communication at same potential)
(Slave Transmission of slave select input function (CSI00))

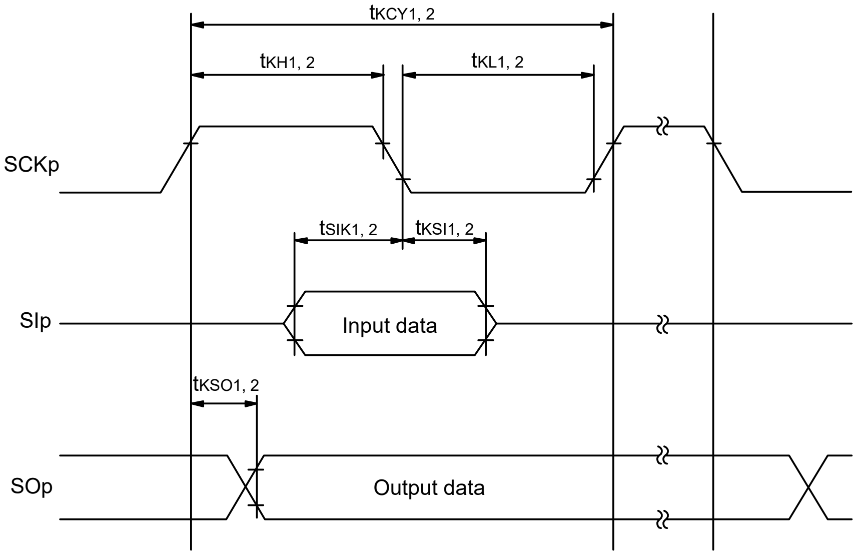


Remark p: CSI number (p = 00, 01, 10 and 11)

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, 10 and 11)
Remark 2. m: Unit number, n: Channel number (mn = 00 to 03)

(4) During communication at same potential (simplified I²C mode)

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

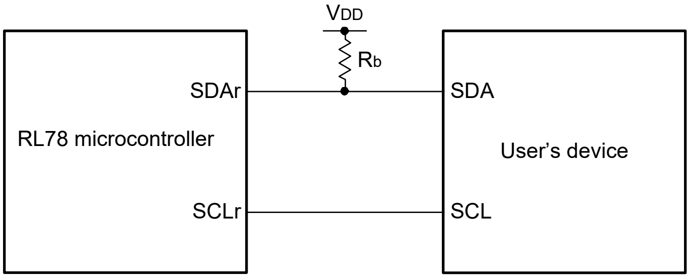
Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
SCLr clock frequency	f_{SCL}	2.7 V ≤ EV_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ		400 Note 1	kHz
		2.4 V ≤ EV_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ		100 Note 1	kHz
Hold time when SCLr = "L"	t_{LOW}	2.7 V ≤ EV_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	1200		ns
		2.4 V ≤ EV_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	4600		ns
Hold time when SCLr = "H"	t_{HIGH}	2.7 V ≤ EV_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	1200		ns
		2.4 V ≤ EV_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	4600		ns
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ EV_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	$1/f_{MCK}$ + 220 Note 2		ns
		2.4 V ≤ EV_{DD} ≤ 5.5 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 ≤ EV_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 kΩ	0	770	ns
		2.4 V ≤ EV_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	0	1420	ns

Note 1. The value must be equal to or less than f_{MCK}/4.

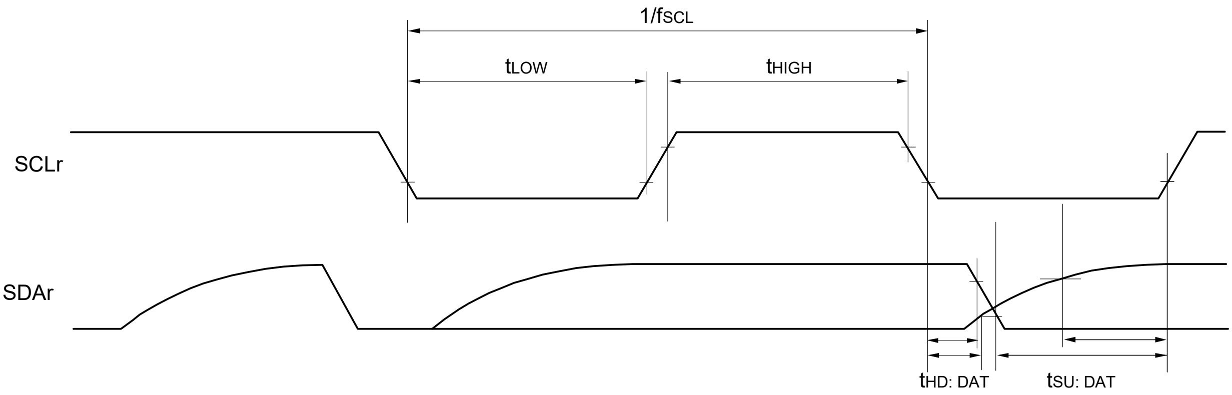
Note 2. Set the f_{MCK} value to keep the hold time of SCLr = "L" and SCLr = "H".

Caution **Select the normal input buffer and the N-ch open drain output (EV_{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. $R_b[\Omega]$: Communication line (SDAr) pull-up resistance, C_b[F]: Communication line (SDAr, SCLr) load capacitance

Remark 2. r: IIC number (r = 00, 01, 10 and 11), g: PIM number (g = 0, 3 and 5), h: POM number (h = 0, 3 and 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 to 3), mn = 00 to 03)

(5) Communication at different potential (1.8 V, 2.5 V, 3.0 V) (UART mode) (dedicated baud rate generator output)

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions			HS (high-speed main) Mode		Unit
					MIN.	MAX.	
Transfer rate		Reception	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.3 V ≤ V_b ≤ 4.0 V			f_{MCK}/12 Note 1	bps
				Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 3		2.0	Mbps
			2.7 V ≤ EV_{DD} < 4.0 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 ≤ EV_{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		1.3	Mbps

Note 1. Transfer rate in the SNOOZE mode is 4,800 bps only.

Note 2. Use it with EV_{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} ≤ 5.5 V)
16 MHz (2.4 V ≤ V_{DD} ≤ 5.5 V)

Caution **Select the TTL input buffer for the RxDq pin and the N-ch open drain output (EV_{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 and 1), g: PIM and POM numbers (g = 0, 2, 3, 5, 12)

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 to 03))

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(2/2)**

Parameter	Symbol	Conditions			HS (high-speed main) Mode		Unit
					MIN.	MAX.	
Transfer rate		Transmission	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V			Note 1	bps
				Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 1.4 kΩ, V_b = 2.7 V		2.6 Note 2	Mbps
			2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V			Note 3	bps
				Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 2.7 kΩ, V_b = 2.3 V		1.2 Note 4	Mbps
			2.4 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V			Notes 5, 6	bps
				Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 5.5 kΩ, V_b = 1.6 V		0.43 Note 7	Mbps

Note 1. The smaller maximum transfer rate derived by using $f_{MCK}/6$ or the following expression is the valid maximum transfer rate.
Expression for calculating the transfer rate when 4.0 V ≤ EV_{DD} ≤ 5.5 V and 2.7 V ≤ V_b ≤ 4.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides

Note 2. This 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. The smaller maximum transfer rate derived by using $f_{MCK}/6$ or the following expression is the valid maximum transfer rate.
Expression for calculating the transfer rate when 2.7 V ≤ EV_{DD} ≤ 4.0 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 4. This value as an example is calculated when the conditions described in the “Conditions” column are met.
Refer to Note 3 above to calculate the maximum transfer rate under conditions of the customer.

Note 5. Use it with EV_{DD} ≥ V_b.

Note 6. The smaller maximum transfer rate derived by using $f_{MCK}/6$ or the following expression is the valid maximum transfer rate.
Expression for calculating the transfer rate when 2.4 V ≤ EV_{DD} < 3.3 V and 1.6 V ≤ V_b ≤ 2.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln (1 - \frac{1.5}{V_b})\} \times 3} \text{ [bps]}$$

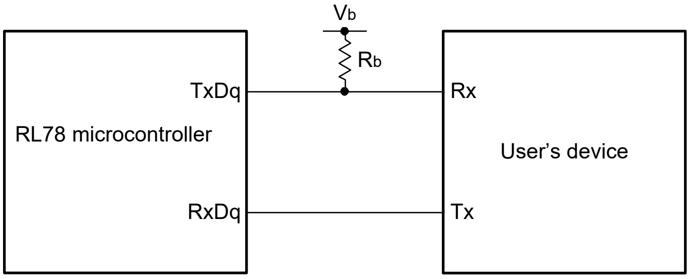
$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln (1 - \frac{1.5}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides

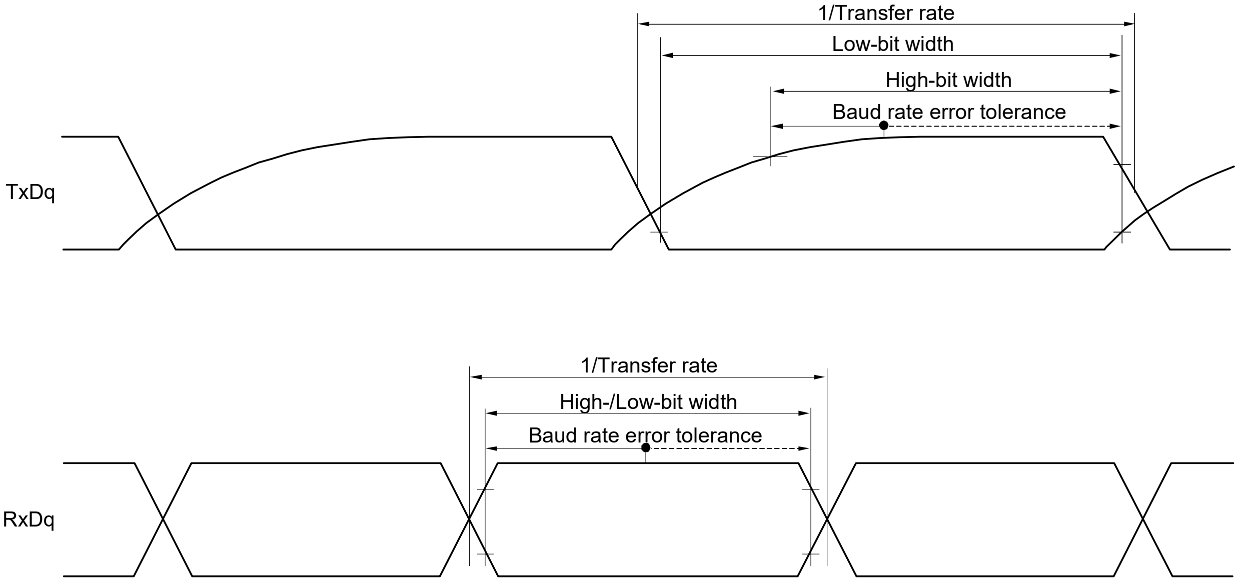
Note 7. This value as an example is calculated when the conditions described in the “Conditions” column are met.
Refer to Note 6 above to calculate the maximum transfer rate under conditions of the customer.

Caution **Select the TTL input buffer for the RxDq pin and the N-ch open drain output (EV_{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 and 1), g: PIM and POM number (g = 0, 2, 3, 5 and 12)

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 to 03))

(6) Communication at different potential (1.8 V, 2.5 V, 3.0 V) (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ $4/f_{CLK}$	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	600		ns
			2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	1000		ns
			2.4 V ≤ EV_{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}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		$t_{KCY1}/2$ - 150		ns
		2.7 V ≤ EV_{DD} < 4.0 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 ≤ EV_{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}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		$t_{KCY1}/2$ - 24		ns
		2.7 V ≤ EV_{DD} < 4.0 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 ≤ EV_{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 (EV_{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.)

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{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}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	162		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	354		ns
		2.4 V ≤ EV_{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}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	38		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			ns
		2.4 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ			ns
Delay time from SCKp↓ to SOp output Note 1	t_{KSO1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		200	ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		390	ns
		2.4 V ≤ EV_{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}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	88		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			ns
		2.4 V ≤ EV_{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}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	38		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			
		2.4 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ			ns
Delay time from SCKp↑ to SOp output Note 2	t_{KSO1}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		50	ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			
		2.4 V ≤ EV_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V Note 3, C_b = 30 pF, R_b = 5.5 kΩ			ns

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

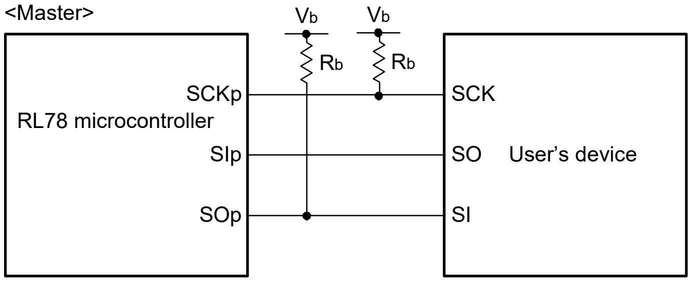
Note 2. When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. Use it with EV_{DD} ≥ V_b.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (EV_{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)

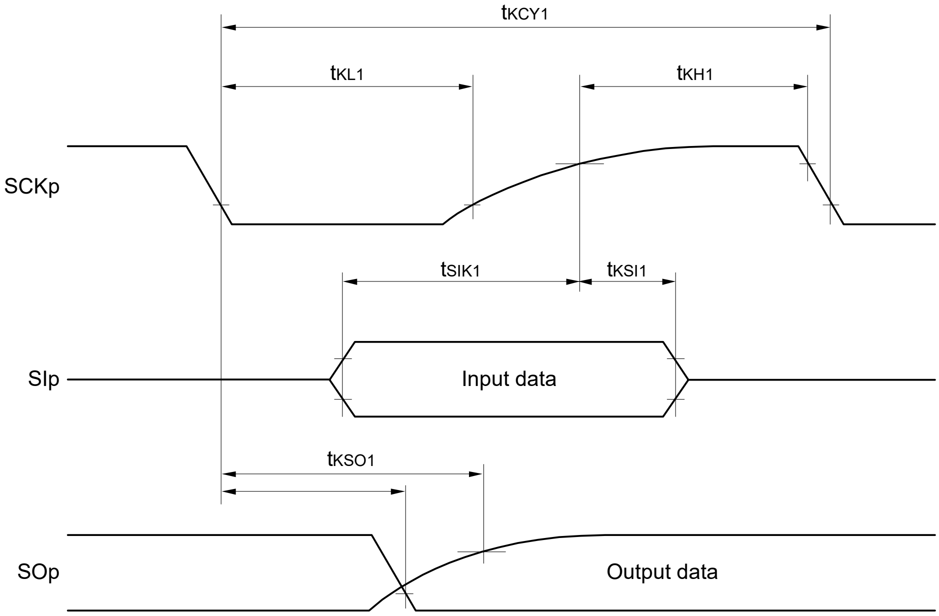


Remark 1. R_b[Ω]: Communication line (SCKp, SOp) pull-up resistance, C_b[F]: Communication line (SCKp, SOp) load capacitance, V_b[V]: Communication line voltage

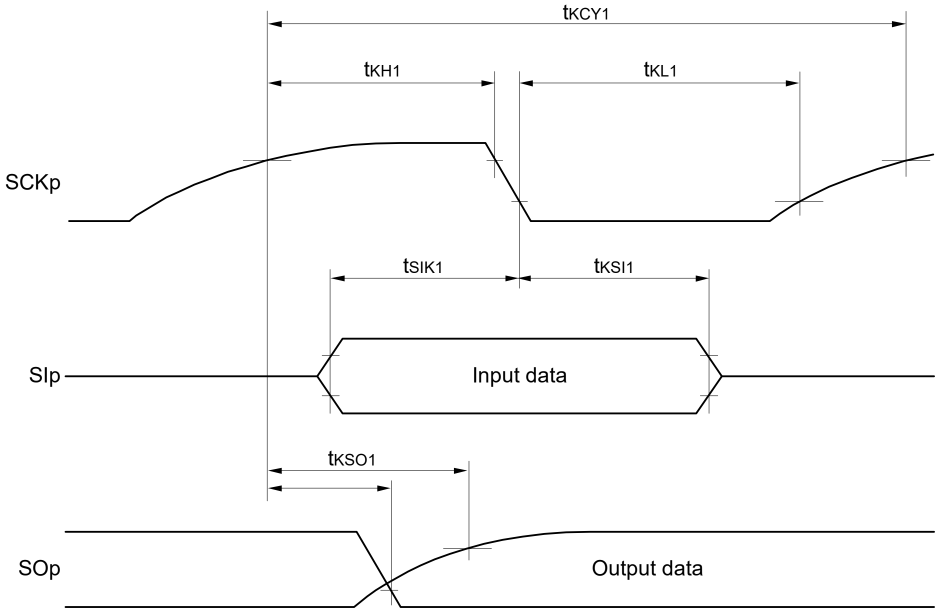
Remark 2. p: CSI number (p = 00, 01, 10 and 11), m: Unit number (m = 0), n: Channel number (n = 0 to 3), g: PIM and POM numbers (g = 0, 2, 3 to 5 and 12)

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 to 03))

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, 10 and 11), m: Unit number (m = 0), n: Channel number (n = 0 to 3), g: PIM and POM numbers (g = 0, 2, 3 to 5 and 12)

(7) Communication at different potential (1.8 V, 2.5 V, 3.0 V) (Simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) Mode		Unit
				MIN.	MAX.	
SCKp cycle time Note 1	t_{KCY2}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V	20 MHz < f_{MCK} ≤ 24 MHz	24/f_{MCK}		ns
			8 MHz < f_{MCK} ≤ 20 MHz	20/f_{MCK}		ns
			4 MHz < f_{MCK} ≤ 8 MHz	16/f_{MCK}		ns
			f_{MCK} ≤ 4 MHz	12/f_{MCK}		ns
		2.7 V ≤ EV_{DD} < 4.0 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 ≤ EV_{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}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		t_{KCY2}/2 - 24		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		t_{KCY2}/2 - 36		ns
		2.4 V ≤ EV_{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 ≤ EV_{DD} ≤ 5.5 V, 2.3 V ≤ V_b ≤ 4.0 V		1/f_{MCK} + 40		ns
		2.4 V ≤ EV_{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}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V C_b = 30 pF, R_b = 1.4 kΩ			2/f_{MCK} + 240	ns
		2.7 V ≤ EV_{DD} < 4.0 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 ≤ EV_{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**, **Caution** and **Remarks** are listed on the next page.)

Note 1. Transfer rate in the SNOOZE mode: MAX. 1 Mbps

Note 2. Use it with $EV_{DD} \geq V_b$.

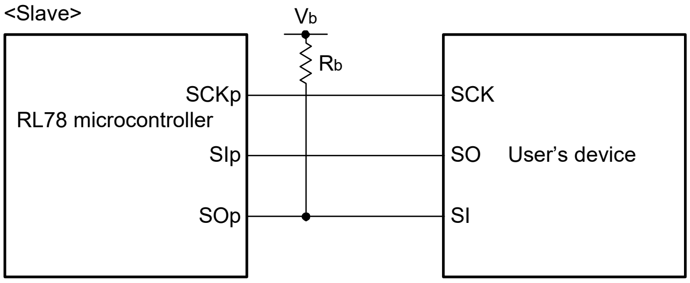
Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes “to SCKp↓” when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

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 (EV_{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.**

Simplified SPI (CSI) mode connection diagram (during communication at different potential)

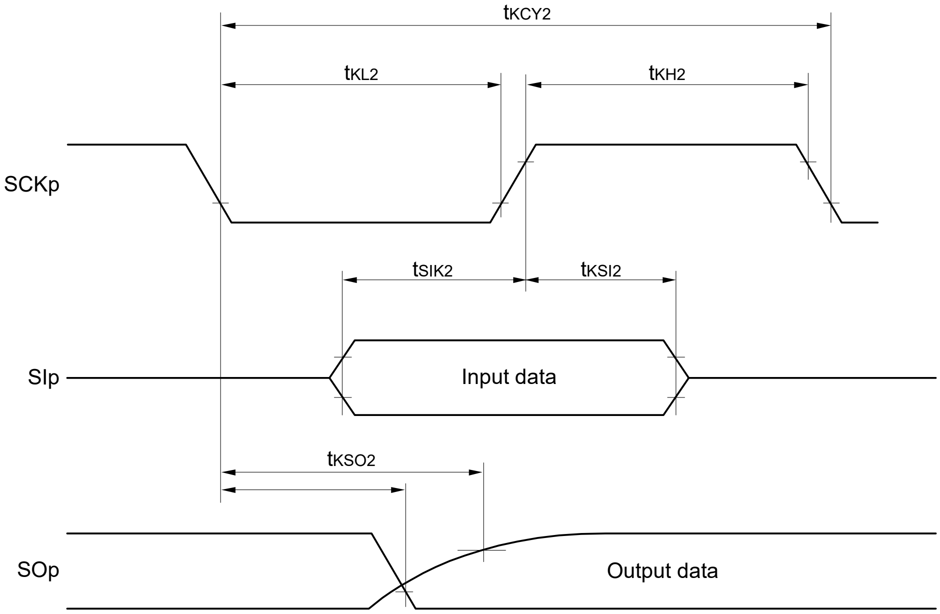


Remark 1. R_b[Ω]: Communication line (SOp) pull-up resistance, C_b[F]: Communication line (SOp) load capacitance, V_b[V]: Communication line voltage

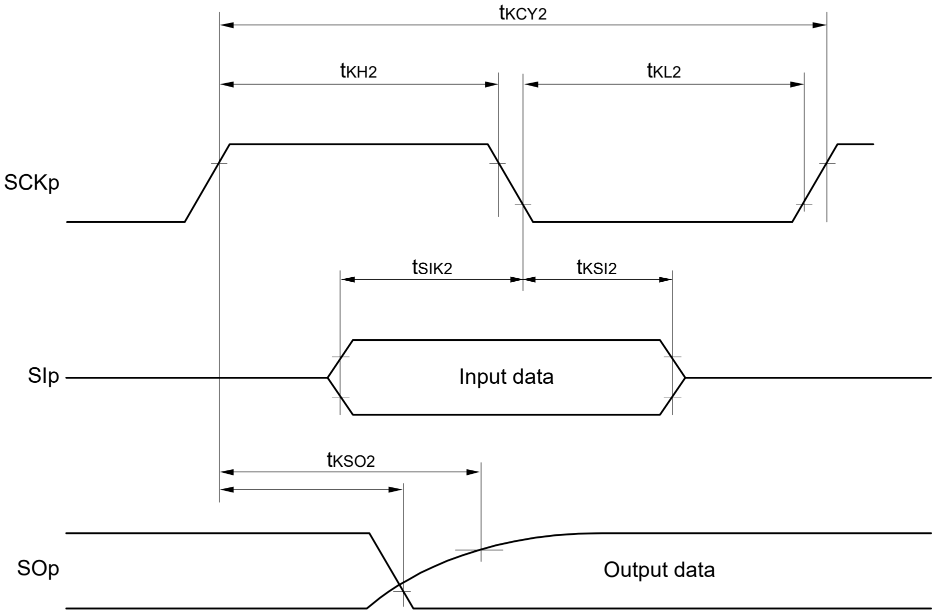
Remark 2. p: CSI number (p = 00, 01, 10 and 11), m: Unit number (m = 0), n: Channel number (n = 0 to 3), g: PIM and POM numbers (g = 0, 2, 3 to 5 and 12)

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 to 03))

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, 10 and 11), m: Unit number (m = 0), n: Channel number (n = 0 to 3), g: PIM and POM numbers (g = 0, 2, 3 to 5 and 12)

(8) Communication at different potential (1.8 V, 2.5 V, 3.0 V) (simplified I2C mode)

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
SCLr clock frequency	f_{SCL}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ		400 Note 1	kHz
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ		400 Note 1	kHz
		4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ		100 Note 1	kHz
		2.7 V ≤ EV_{DD} < 4.0 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 ≤ EV_{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}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	1200		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	1200		ns
		4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	4600		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	4600		ns
		2.4 V ≤ EV_{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}	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	620		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	500		ns
		4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	2700		ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	2400		ns
		2.4 V ≤ EV_{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)	$t_{SU:DAT}$	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	$1/f_{MCK}$ + 340 Note 3		ns
		2.7 V ≤ EV_{DD} < 4.0 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
		4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	$1/f_{MCK}$ + 760 Note 3		ns
		2.7 V ≤ EV_{DD} < 4.0 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 ≤ EV_{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)	$t_{HD:DAT}$	4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	0	770	ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	0	770	ns
		4.0 V ≤ EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	0	1420	ns
		2.7 V ≤ EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	0	1420	ns
		2.4 V ≤ EV_{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

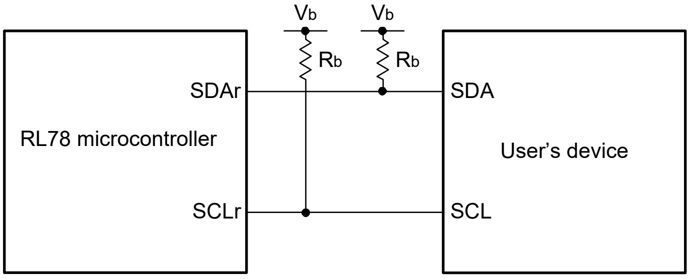
Note 1. The value must be equal to or less than $f_{MCK}/4$.

Note 2. Use it with $EV_{DD} \geq V_b$.

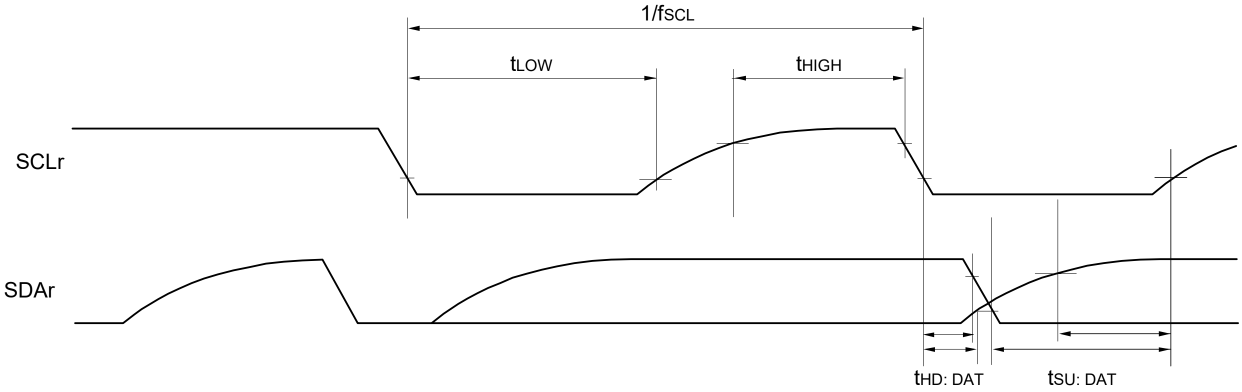
Note 3. Set the f_{MCK} value to keep the hold time of SCLr = "L" and SCLr = "H".

Caution **Select the TTL input buffer and the N-ch open drain output (EV_{DD} tolerance) mode for the SDAr pin and the N-ch open drain output (EV_{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.**

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, 10 and 11), g: PIM, POM number (g = 0, 3 and 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 to 3), mn = 00 to 03)

36.5.2 Serial interface IICA

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) mode				Unit
			Standard mode		Fast mode		
			MIN.	MAX.	MIN.	MAX.	
SCLA0 clock frequency	f_{SCL}	Fast mode: f_{CLK} ≥ 3.5 MHz	—	—	0	400	kHz
		Standard mode: f_{CLK} ≥ 1 MHz	0	100	—	—	kHz
Setup time of restart condition	$t_{SU: STA}$		4.7		0.6		µs
Hold time [Note 1]	$t_{HD: STA}$		4.0		0.6		µs
Hold time when SCLA0 = “L”	t_{LOW}		4.7		1.3		µs
Hold time when SCLA0 = “H”	t_{HIGH}		4.0		0.6		µs
Data setup time (reception)	$t_{SU: DAT}$		250		100		ns
Data hold time (transmission) [Note 2]	$t_{HD: DAT}$		0	3.45	0	0.9	µs
Setup time of stop condition	$t_{SU: STO}$		4.0		0.6		µs
Bus-free time	t_{BUF}		4.7		1.3		µs

Note 1. The first clock pulse is generated after this period when the start/restart condition is detected.

Note 2. The maximum value (MAX.) of $t_{HD: DAT}$ is during normal transfer and a clock stretch state is inserted in the ACK (acknowledge) timing.

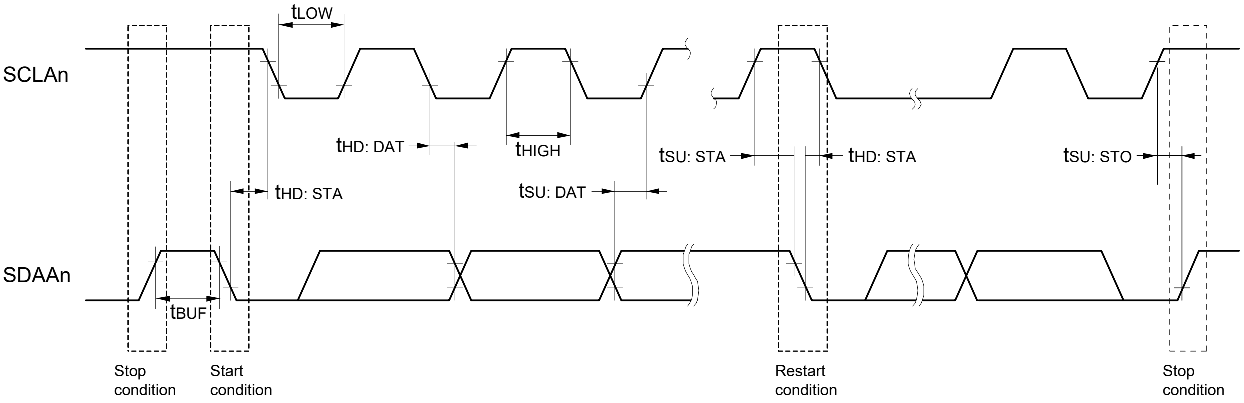
Caution **The values in the above table are applied even when bit 2 (PIOR02) in the peripheral I/O redirection register 0 (PIOR0) is 1. At this time, the pin characteristics (I_{OH1}, I_{OL1}, V_{OH1}, V_{OL1}) must satisfy the values in the redirect destination.**

Remark The maximum value of C_b (communication line capacitance) and the value of R_b (communication line pull-up resistor) at that time in each mode are as follows.

Standard mode: C_b = 400 pF, R_b = 2.7 kΩ

Fast mode: C_b = 320 pF, R_b = 1.1 kΩ

IICA serial transfer timing



Remark n = 0, 1

36.6 Analog Characteristics

36.6.1 A/D converter characteristics

Classification of A/D converter characteristics

Reference Voltage Input channel	Reference voltage (+) = AV_{REFP} Reference voltage (-) = AV_{REFM}	Reference voltage (+) = V_{DD} Reference voltage (-) = V_{SS}	Reference voltage (+) = V_{BGR} Reference voltage (-) = AV_{REFM}
ANI0 to ANI3	Refer to **36.6.1 (1)**.	Refer to **36.6.1 (3)**.	Refer to **36.6.1 (4)**.
ANI16 to ANI22	Refer to **36.6.1 (2)**.		
Internal reference voltage Temperature sensor output voltage	Refer to **36.6.1 (1)**.		—

(1) When reference voltage (+) = AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (-) = AV_{REFM}/ANI1 (ADREFM = 1), target pin: ANI2 and ANI3, internal reference voltage, and temperature sensor output voltage

(T_A = -40 to +105°C, 2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = AV_{REFP}, Reference voltage (-) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error Note 1	AINL	10-bit resolution AV_{REFP} = V_{DD} Note 3	2.4 V ≤ AV_{REFP} ≤ 5.5 V		1.2	±3.5	LSB
Conversion time	t_{CONV}	10-bit resolution Target pin: ANI2 and ANI3	3.6 V ≤ V_{DD} ≤ 5.5 V	2.125		39	µs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	µs
			2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	µs
		10-bit resolution Target pin: Internal reference voltage, and temperature sensor output voltage	3.6 V ≤ V_{DD} ≤ 5.5 V	2.375		39	µs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.5625		39	µs
			2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	µs
Zero-scale error Notes 1, 2	E_{ZS}	10-bit resolution AV_{REFP} = V_{DD} Note 3	2.4 V ≤ AV_{REFP} ≤ 5.5 V			±0.25	%FSR
Full-scale error Notes 1, 2	E_{FS}	10-bit resolution AV_{REFP} = V_{DD} Note 3	2.4 V ≤ AV_{REFP} ≤ 5.5 V			±0.25	%FSR
Integral linearity error Note 1	ILE	10-bit resolution AV_{REFP} = V_{DD} Note 3	2.4 V ≤ AV_{REFP} ≤ 5.5 V			±2.5	LSB
Differential linearity error Note 1	DLE	10-bit resolution AV_{REFP} = V_{DD} Note 3	2.4 V ≤ AV_{REFP} ≤ 5.5 V			±1.5	LSB
Analog input voltage	V_{AIN}	ANI2 and ANI3		0		AV_{REFP}	V
		Internal reference voltage (2.4 V ≤ V_{DD} ≤ 5.5 V)		V_{BGR} Note 4			V
		Temperature sensor output voltage (2.4 V ≤ V_{DD} ≤ 5.5 V)		V_{TMPS25} Note 4			V

Note 1. Excludes quantization error (±1/2 LSB).

Note 2. This value is indicated as a ratio (%FSR) to the full-scale value.

Note 3. When AV_{REFP} < V_{DD}, the MAX. values are as follows.

Overall error:	Add ±1.0 LSB to the MAX. value when AV_{REFP} = V_{DD}.
Zero-scale error/Full-scale error:	Add ±0.05%FSR to the MAX. value when AV_{REFP} = V_{DD}.
Integral linearity error/ Differential linearity error:	Add ±0.5 LSB to the MAX. value when AV_{REFP} = V_{DD}.

Note 4. Refer to **36.6.2 Temperature sensor characteristics/internal reference voltage characteristic**.

(2) When reference voltage (+) = AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (-) = AV_{REFM}/ANI1 (ADREFM = 1), target pin: ANI16 to ANI22

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, 2.4 V ≤ AV_{REFP} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V, Reference voltage (+) = AV_{REFP}, Reference voltage (-) = AV_{REFM} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error [Note 1]	AINL	10-bit resolution EV_{DD} ≤ AV_{REFP} = V_{DD} [Notes 3, 4]	2.4 V ≤ AV_{REFP} ≤ 5.5 V		1.2	±5.0	LSB
Conversion time	t_{CONV}	10-bit resolution Target ANI pin: ANI16 to ANI22	3.6 V ≤ V_{DD} ≤ 5.5 V	2.125		39	µs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	µs
			2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	µs
Zero-scale error [Notes 1, 2]	E_{ZS}	10-bit resolution EV_{DD} ≤ AV_{REFP} = V_{DD} [Notes 3, 4]	2.4 V ≤ AV_{REFP} ≤ 5.5 V			±0.35	%FSR
Full-scale error [Notes 1, 2]	E_{FS}	10-bit resolution EV_{DD} ≤ AV_{REFP} = V_{DD} [Notes 3, 4]	2.4 V ≤ AV_{REFP} ≤ 5.5 V			±0.35	%FSR
Integral linearity error [Note 1]	ILE	10-bit resolution EV_{DD} ≤ AV_{REFP} = V_{DD} [Notes 3, 4]	2.4 V ≤ AV_{REFP} ≤ 5.5 V			±3.5	LSB
Differential linearity error [Note 1]	DLE	10-bit resolution EV_{DD} ≤ AV_{REFP} = V_{DD} [Notes 3, 4]	2.4 V ≤ AV_{REFP} ≤ 5.5 V			±2.0	LSB
Analog input voltage	V_{AIN}	ANI16 to ANI22		0		AV_{REFP} and EV_{DD}	V

Note 1. Excludes quantization error (±1/2 LSB).

Note 2. This value is indicated as a ratio (%FSR) to the full-scale value.

Note 3. When EV_{DD} ≤ AV_{REFP} < V_{DD}, the MAX. values are as follows.

Overall error:	Add ±1.0 LSB to the MAX. value when AV_{REFP} = V_{DD}.
Zero-scale error/Full-scale error:	Add ±0.05%FSR to the MAX. value when AV_{REFP} = V_{DD}.
Integral linearity error/ Differential linearity error:	Add ±0.5 LSB to the MAX. value when AV_{REFP} = V_{DD}.

Note 4. When AV_{REFP} < EV_{DD} ≤ V_{DD}, the MAX. values are as follows.

Overall error:	Add ±4.0 LSB to the MAX. value when AV_{REFP} = V_{DD}.
Zero-scale error/Full-scale error:	Add ±0.20%FSR to the MAX. value when AV_{REFP} = V_{DD}.
Integral linearity error/ Differential linearity error:	Add ±2.0 LSB to the MAX. value when AV_{REFP} = V_{DD}.

(3) When reference voltage (+) = V_{DD} (ADREFP1 = 0, ADREFP0 = 0), reference voltage (-) = V_{SS} (ADREFM = 0), target pin: ANI0 to ANI3, ANI16 to ANI22, internal reference voltage, and temperature sensor output voltage

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V,
Reference voltage (+) = V_{DD}, Reference voltage (-) = V_{SS})

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error Note 1	AINL	10-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V		1.2	±7.0	LSB
Conversion time	t_{CONV}	10-bit resolution Target pin: ANI0 to ANI3, ANI16 to ANI22	3.6 V ≤ V_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.1875		39	μs
			2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
		10-bit resolution Target pin: internal reference voltage, and temperature sensor output voltage	3.6 V ≤ V_{DD} ≤ 5.5 V	2.375		39	μs
			2.7 V ≤ V_{DD} ≤ 5.5 V	3.5625		39	μs
			2.4 V ≤ V_{DD} ≤ 5.5 V	17		39	μs
Zero-scale error Notes 1, 2	E_{ZS}	10-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V			±0.60	%FSR
Full-scale error Notes 1, 2	E_{FS}	10-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V			±0.60	%FSR
Integral linearity error Note 1	ILE	10-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V			±4.0	LSB
Differential linearity error Note 1	DLE	10-bit resolution	2.4 V ≤ V_{DD} ≤ 5.5 V			±2.0	LSB
Analog input voltage	V_{AIN}	ANI0 to ANI3		0		V_{DD}	V
		ANI16 to ANI22		0		EV_{DD}	V
		Internal reference voltage (2.4 V ≤ V_{DD} ≤ 5.5 V)		V_{BGR} Note 3			V
		Temperature sensor output voltage (2.4 V ≤ V_{DD} ≤ 5.5 V)		V_{TMPS25} Note 3			V

Note 1. Excludes quantization error (±1/2 LSB).

Note 2. This value is indicated as a ratio (% FSR) to the full-scale value.

Note 3. Refer to **36.6.2 Temperature sensor characteristics/internal reference voltage characteristic**.

(4) When reference voltage (+) = Internal reference voltage (ADREFP1 = 1, ADREFP0 = 0), reference voltage (-) = AV_{REFM}/ANI1 (ADREFM = 1), target pin: ANI0 to ANI3, ANI16 to ANI22

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, 2.4 V ≤ EV_{DD} ≤ V_{DD} = 0 V,
Reference voltage (+) = V_{BGR} Note 3, Reference voltage (-) = AV_{REFM} = 0 V Note 4)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	RES		8			bit
Conversion time	t_{CONV}		17		39	μs
Zero-scale error Notes 1, 2	E_{ZS}				±0.60	% FSR
Integral linearity error Note 1	ILE				±2.0	LSB
Differential linearity error Note 1	DLE	8-bit resolution			±1.0	LSB
Analog input voltage	V_{AIN}		0		V_{BGR} Note 3	V

Note 1. Excludes quantization error (±1/2 LSB).

Note 2. This value is indicated as a ratio (% FSR) to the full-scale value.

Note 3. Refer to **36.6.2 Temperature sensor characteristics/internal reference voltage characteristic**.

Note 4. When reference voltage (-) = V_{SS}, the MAX. values are as follows.

Zero-scale error:	Add ±0.35%FSR to the MAX. value when reference voltage (-) = AV_{REFM}.
Integral linearity error:	Add ±0.5 LSB to the MAX. value when reference voltage (-) = AV_{REFM}.
Differential linearity error:	Add ±0.2 LSB to the MAX. value when reference voltage (-) = AV_{REFM}.

36.6.2 Temperature sensor characteristics/internal reference voltage characteristic

(T_A = -40 to +105°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Temperature sensor output voltage	V_{TMPS25}	Setting ADS register = 80H, T_A = +25°C		1.05		V
Internal reference voltage	V_{BGR}	Setting ADS register = 81H	1.38	1.45	1.5	V
Temperature coefficient	F_{VTMPS}	Temperature sensor that depends on the temperature		-3.6		mV/°C
Operation stabilization wait time	t_{AMP}	2.4 V ≤ V_{DD} ≤ 3.6 V	5			μs

36.6.3 D/A converter (channel 1)

(T_A = -40 to +105°C, 2.4 V ≤ EV_{SS} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES					8	bit
Overall error	AINL	Rload = 4 MΩ	2.4 V ≤ V_{DD} ≤ 5.5 V			±2.5	LSB
		Rload = 8 MΩ	2.4 V ≤ V_{DD} ≤ 5.5 V			±2.5	LSB
Settling time	t_{SET}	Cload = 20 pF	2.7 V ≤ V_{DD} ≤ 5.5 V			3	μs
			2.4 V ≤ V_{DD} < 2.7 V			6	μs

36.6.4 Comparator

(Comparator 0: T_A = -40 to +105°C, 2.7 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)
(Comparator 1: T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage range	V_{IREF0}	IVREF0 pin		0		V_{DD} - 1.4 Note 1	V
	V_{IREF1}	IVREF1 pin		1.4 Note 1		V_{DD}	V
	V_{ICMP}	IVCMP0 pin		-0.3		V_{DD} + 0.3	V
		IVCMP1 pin		-0.3		EV_{DD} + 0.3	V
Output delay	td	V_{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		µs
			Comparator low-speed mode, window mode		4		µs
Operation stabilization wait time	t_{CMP}			100			µs
Reference voltage declination in channel 0 of internal DAC Note 2	⊿VIDAC					±2.5	LSB

Note 1. In window mode, make sure that V_{REF1} - V_{REF0} ≥ 0.2 V.

Note 2. Only in CMP0

36.6.5 PGA

(T_A = -40 to +105°C, 2.7 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input offset voltage	V_{IOPGA}					±10	mV
Input voltage range	V_{IPGA}			0		0.9 × V_{DD}/Gain	V
Output voltage range	V_{IOHPGA}			0.93 × V_{DD}			V
	V_{IOLPGA}					0.07 × V_{DD}	V
Gain error		x4, x8				±1	%
		x16				±1.5	%
		x32				±2	%
Slew rate	SR_{RPGA}	Rising When V_{IN} = 0.1V_{DD}/gain to 0.9V_{DD}/gain. 10 to 90% of output voltage amplitude	4.0 V ≤ V_{DD} ≤ 5.5 V (Other than x32)	3.5			V/μs
			4.0 V ≤ V_{DD} ≤ 5.5 V (x32)	3.0			
			2.7 V ≤ V_{DD} ≤ 4.0V	0.5			
	SR_{FPGA}	Falling When V_{IN}= 0.1V_{DD}/gain to 0.9V_{DD}/gain. 90 to 10% of output voltage amplitude	4.0 V ≤ V_{DD} ≤ 5.5 V (Other than x32)	3.5			
			4.0 V ≤ V_{DD} ≤ 5.5 V (x32)	3.0			
			2.7 V ≤ V_{DD} ≤ 4.0V	0.5			
Reference voltage stabilization wait time[Note]	t_{PGA}	x4, x8				5	μs
		x16, x32				10	μs

Note Time required until a state is entered where the DC and AC specifications of the PGA are satisfied after the PGA operation has been enabled (PGAEN = 1).

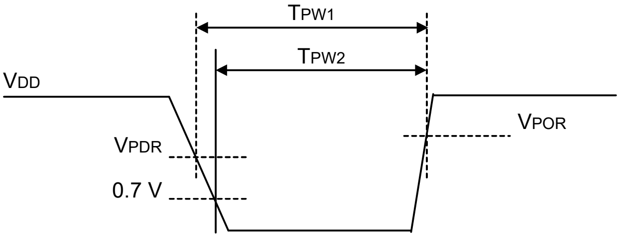
36.6.6 POR circuit characteristics

(T_A = -40 to +105°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POR}	The power supply voltage is rising.	1.45	1.51	1.57	V
	V_{PDR}	The power supply voltage is falling.[Note 1]	1.44	1.50	1.56	V
Minimum pulse width [Note 2]	T_{PW1}	Other than STOP/SUB HALT/SUB RUN	300			µs
	T_{PW2}	STOP/SUB HALT/SUB RUN	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 36.4 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).



36.6.7 LVD circuit characteristics

(1) LVD Detection Voltage of Reset Mode and Interrupt Mode

(T_A = -40 to +105°C, $V_{PDR} \leq EV_{DD} \leq V_{DD} \leq 5.5$ V, V_{SS} = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVD0}	The power supply voltage is rising.	3.90	4.06	4.22	V
			The power supply voltage is falling.	3.83	3.98	4.13	V
		V_{LVD1}	The power supply voltage is rising.	3.60	3.75	3.90	V
			The power supply voltage is falling.	3.53	3.67	3.81	V
		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 +105°C, $V_{PDR} \leq EV_{DD} \leq V_{DD} \leq 5.5$ V, V_{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
	V_{LVDD3}		LVIS0, LVIS1 = 0, 0	Rising release reset voltage	3.90	4.06	4.22	V
				Falling interrupt voltage	3.83	3.98	4.13	V

36.6.8 Power supply voltage rising slope characteristics

(T_A = -40 to +105°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Power supply voltage rising slope	S_{VDD}				54	V/ms

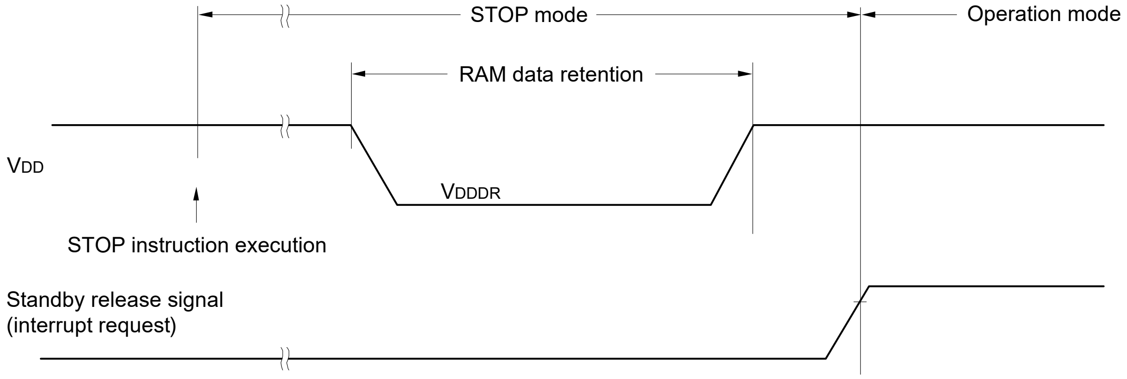
Caution **Make sure to keep the internal reset state by the LVD circuit or an external reset until V_{DD} reaches the operating voltage range shown in 36.4 AC Characteristics.**

36.7 RAM Data Retention Characteristics

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		1.44 Note		5.5	V

Note The value depends on the POR detection voltage. When the voltage drops, the RAM data is retained before a POR reset is effected, but RAM data is not retained when a POR reset is effected.



36.8 Flash Memory Programming Characteristics

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
System clock frequency	f_{CLK}	2.4 V ≤ V_{DD} ≤ 5.5 V		1		24	MHz
Number of code flash rewrites Notes 1, 2, 3	C_{erwr}	Retained for 20 years	T_A = 85°C	1,000			Times
Number of data flash rewrites Notes 1, 2, 3		Retained for 1 year	T_A = 25°C		1,000,000		
		Retained for 5 years	T_A = 85°C	100,000			
		Retained for 20 years	T_A = 85°C	10,000			

Note 1. 1 erase + 1 write after the erase is regarded as 1 rewrite. The retaining years are until next rewrite after the rewrite.

Note 2. 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.

36.9 Dedicated Flash Memory Programmer Communication (UART)

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		During serial programming	115,200		1,000,000	bps

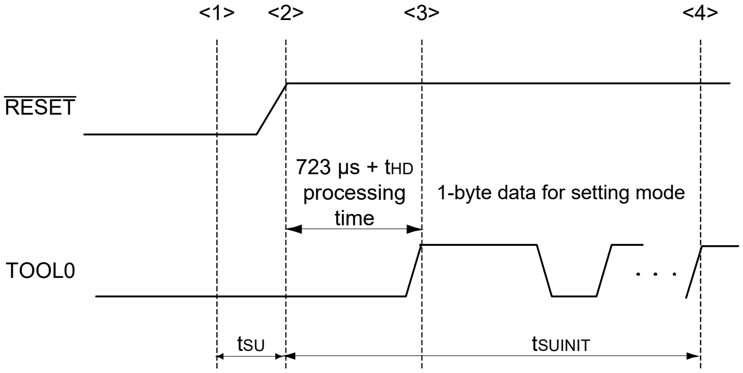
36.10 Timing of Entry to Flash Memory Programming Modes

(T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{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.

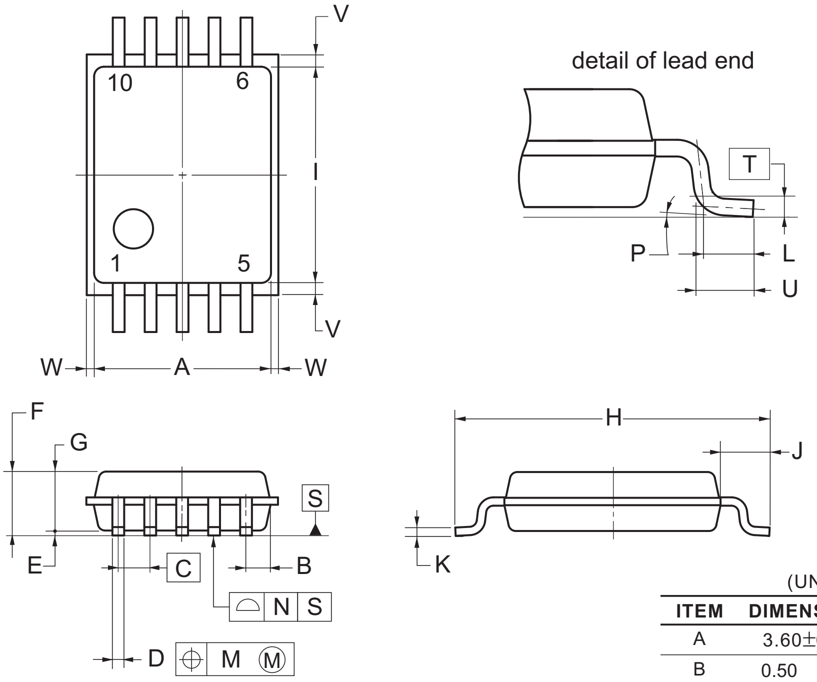
t_{SU}: How long from when the TOOL0 pin is placed at the low level until a pin reset ends

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 37 PACKAGE DRAWINGS

37.1 10-pin package

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LSSOP10-4.4x3.6-0.65	PLSP0010JA-A	P10MA-65-CAC-2	0.05



(UNIT:mm)

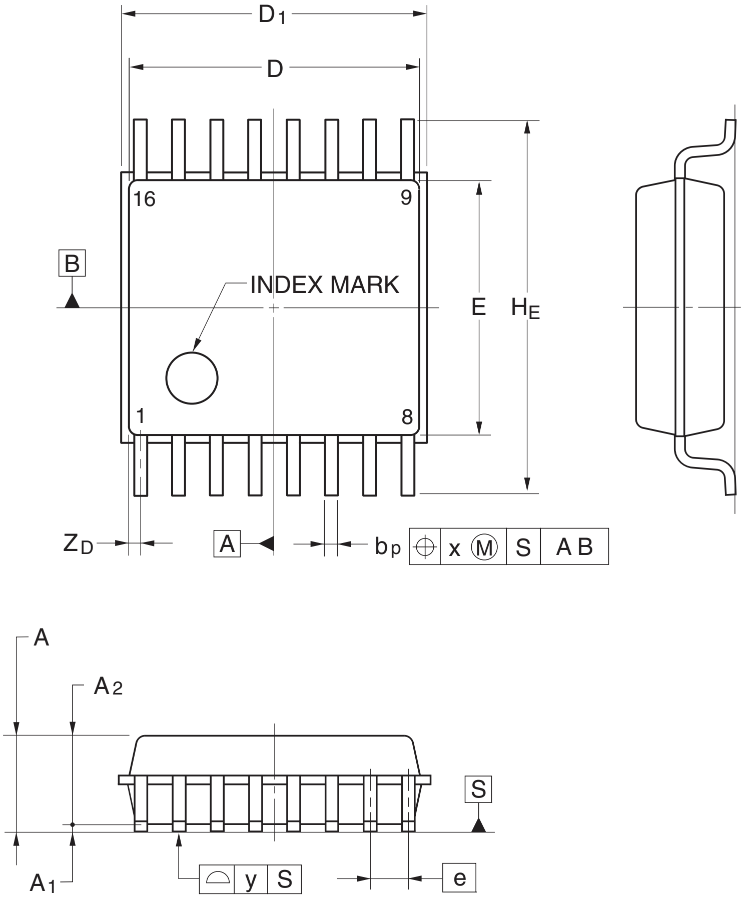
ITEM	DIMENSIONS
A	3.60±0.10
B	0.50
C	0.65 (T.P.)
D	0.24±0.08
E	0.10±0.05
F	1.45 MAX.
G	1.20±0.10
H	6.40±0.20
I	4.40±0.10
J	1.00±0.20
K	$0.17^{+0.08}_{-0.07}$
L	0.50
M	0.13
N	0.10
P	$3°^{+5°}_{-3°}$
T	0.25 (T.P.)
U	0.60±0.15
V	0.25 MAX.
W	0.15 MAX.

NOTE

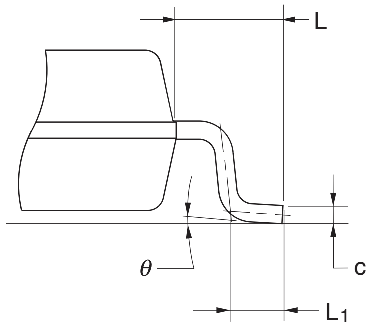
Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

37.2 16-pin package

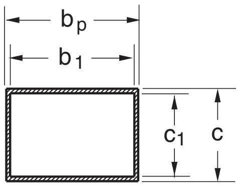
JEITA Package code	RENESAS code	Previous code	MASS(TYP.)[g]
P-SSOP16-4.4x5-0.65	PRSP0016JC-B	P16MA-65-FAB	0.08



detail of lead end

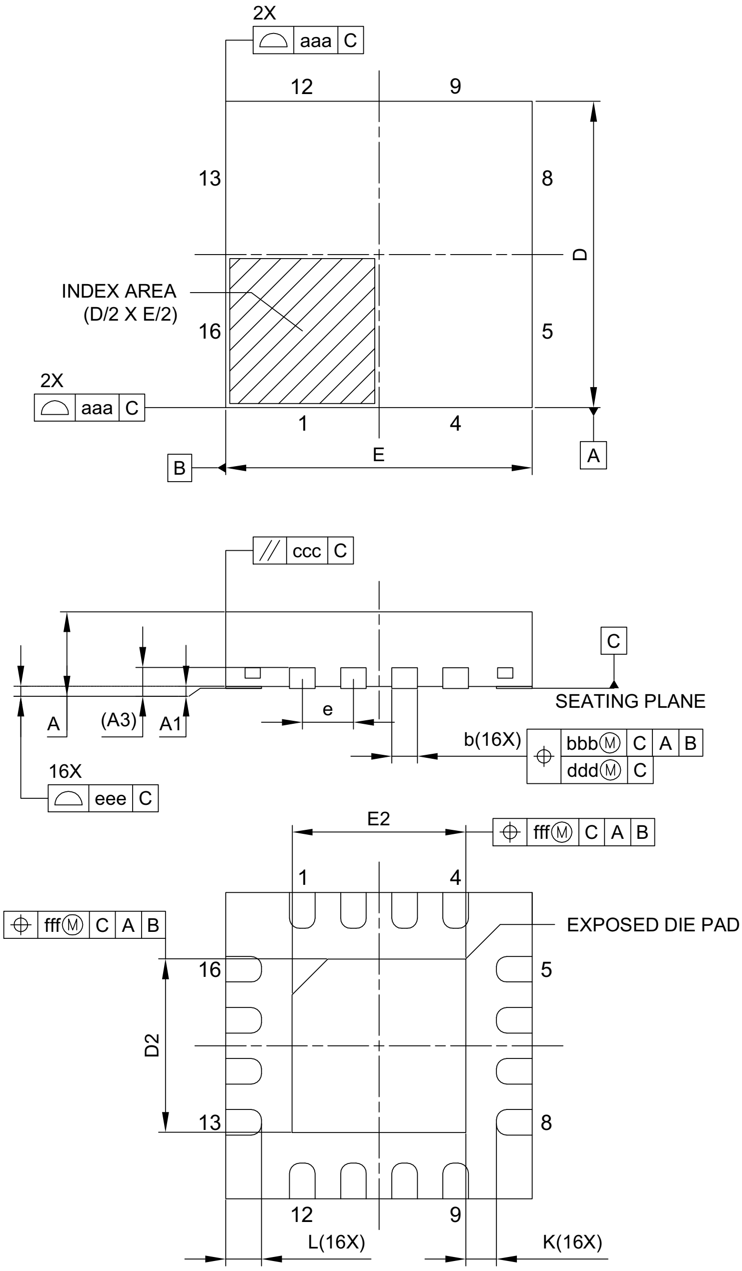


Terminal cross section



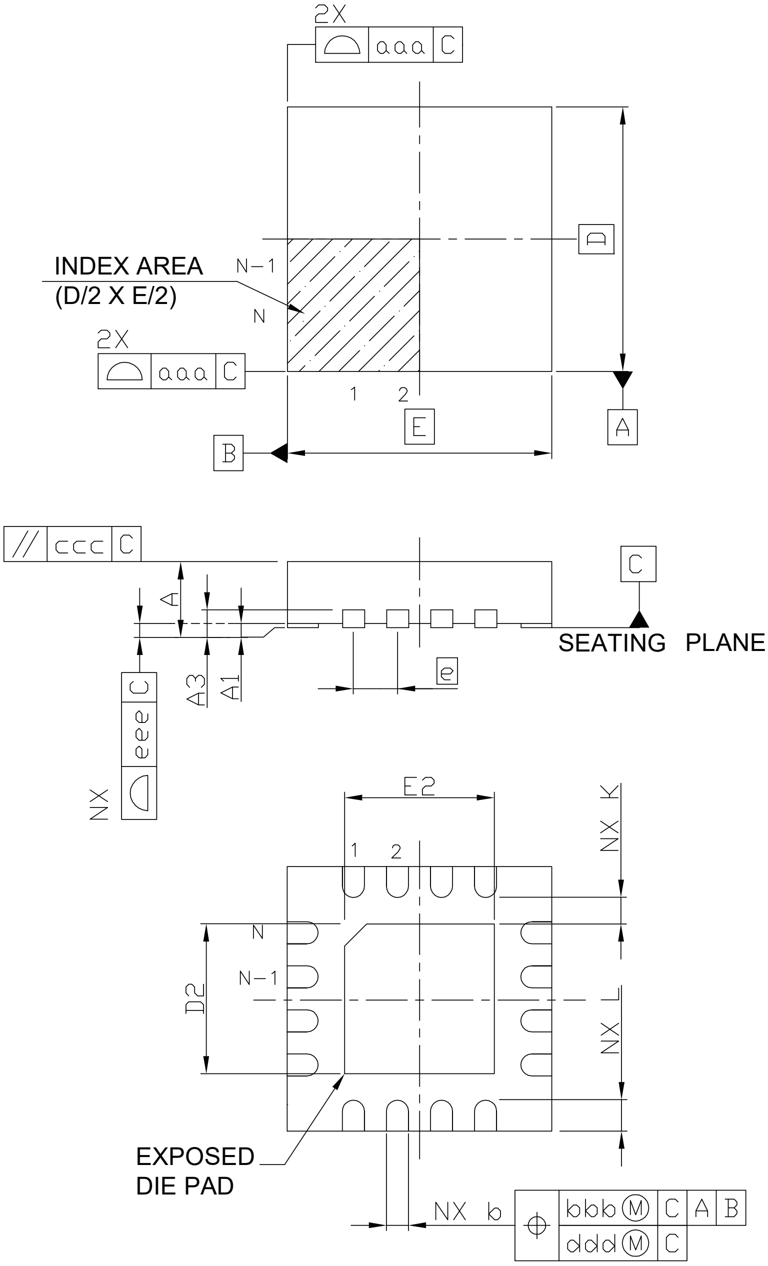
Referance Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	4.85	5.00	5.15
D_1	5.05	5.20	5.35
E	4.20	4.40	4.60
A_2	—	1.50	—
A_1	0.075	0.125	0.175
A	—	—	1.725
b_p	0.17	0.24	0.32
b_1	—	0.22	—
c	0.14	0.17	0.20
c_1	—	0.15	—
θ	0°	—	8°
H_E	6.20	6.40	6.60
e	—	0.65	—
x	—	—	0.13
y	—	—	0.10
Z_D	—	0.225	—
L	0.35	0.50	0.65
L_1	—	1.00	—

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN016-3x3-0.50	PWQN0016KD-A	0.02



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A1	0.00	0.02	0.05
A3	0.203 REF.		
b	0.20	0.25	0.30
D	3.00 BSC		
E	3.00 BSC		
e	0.50 BSC		
L	0.30	0.35	0.40
K	0.20	—	—
D2	1.65	1.70	1.75
E2	1.65	1.70	1.75
aaa	0.15		
bbb	0.10		
ccc	0.10		
ddd	0.05		
eee	0.08		
fff	0.10		

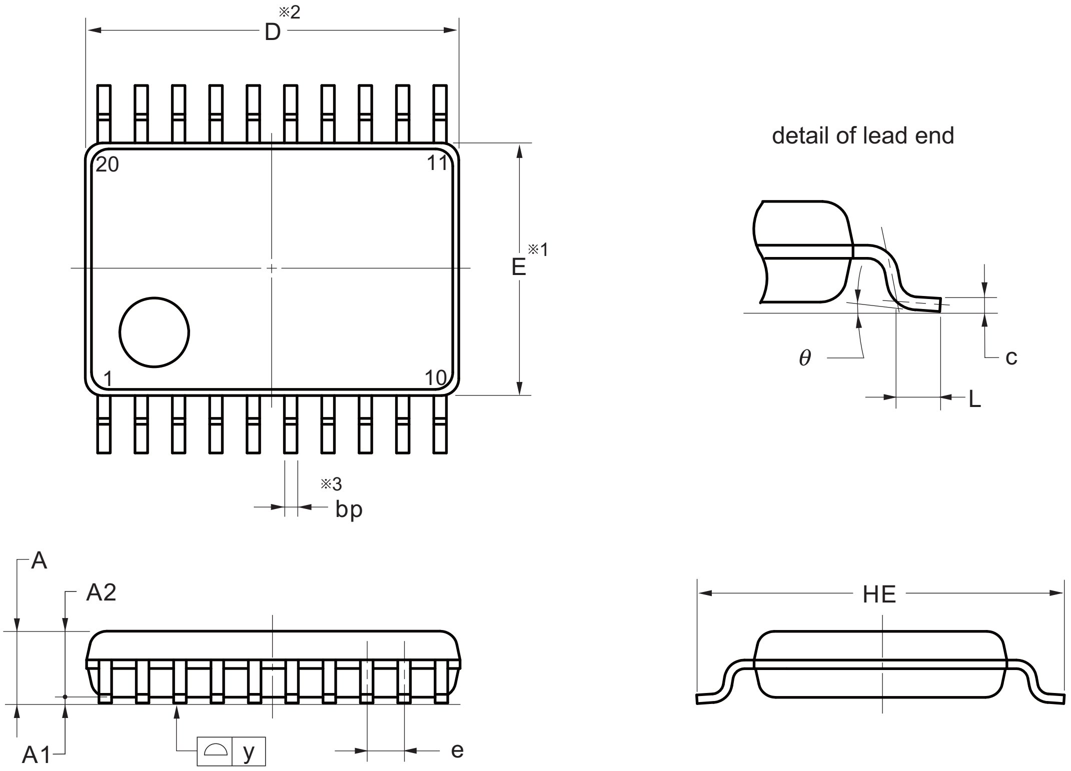
JEITA Package Code	RENESAS Code	MASS(Typ.)[g]
P-HWQFN16-3×3-0.50	PWQN0016KE-A	0.02



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
D	—	3.00	—
E	—	3.00	—
e	—	0.50	—
N	16		
L	0.25	0.35	0.45
K	0.20	—	—
D_2	1.60	1.70	1.80
E_2	1.60	1.70	1.80
aaa	—	—	0.15
bbb	—	—	0.10
ccc	—	—	0.10
ddd	—	—	0.05
eee	—	—	0.08

37.3 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)

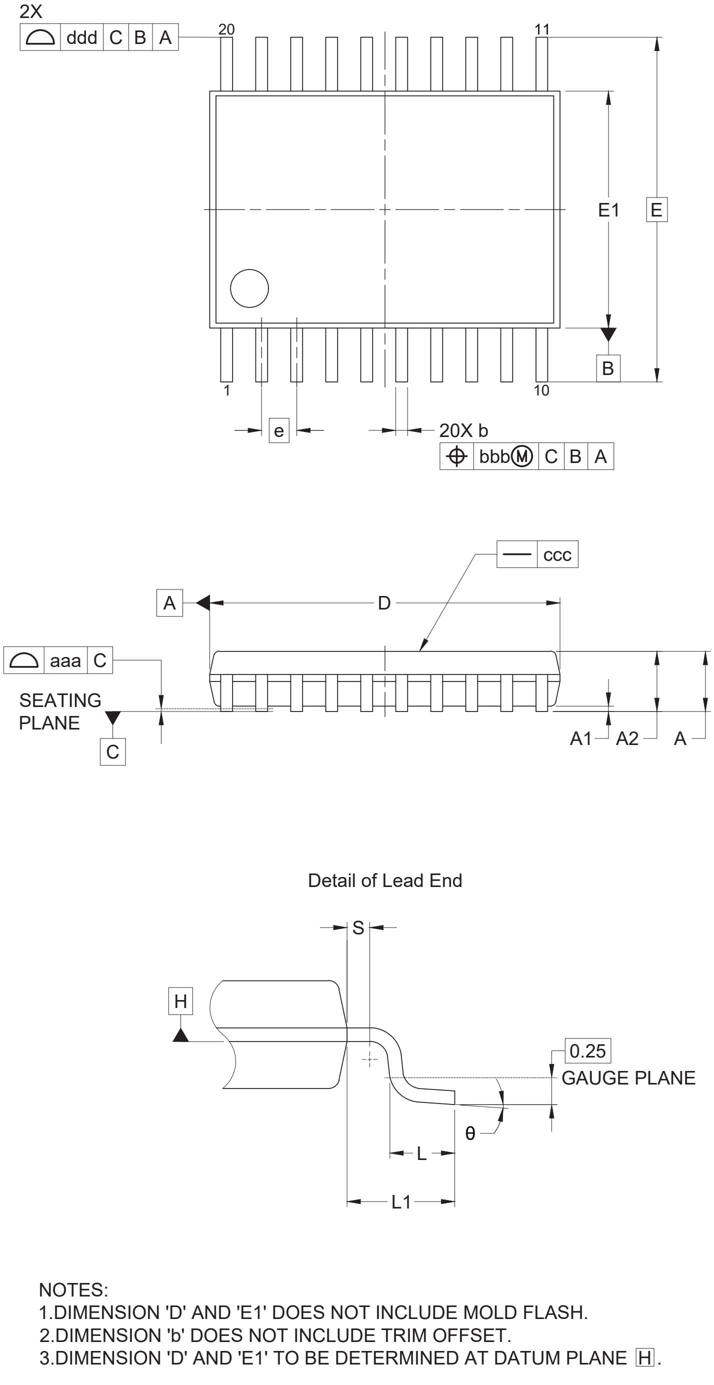
ITEM	DIMENSIONS
D	6.50±0.10
E	4.40±0.10
HE	6.40±0.20
A	1.45 MAX.
A1	0.10±0.10
A2	1.15
e	0.65±0.12
bp	$0.22^{+0.10}_{-0.05}$
c	$0.15^{+0.05}_{-0.02}$
L	0.50±0.20
y	0.10
θ	0° to 10°

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

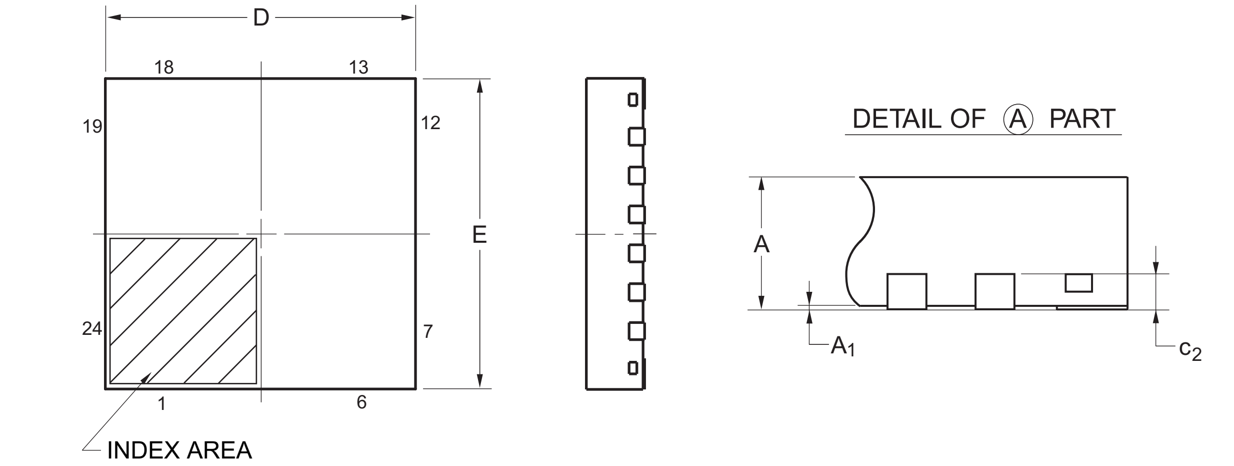


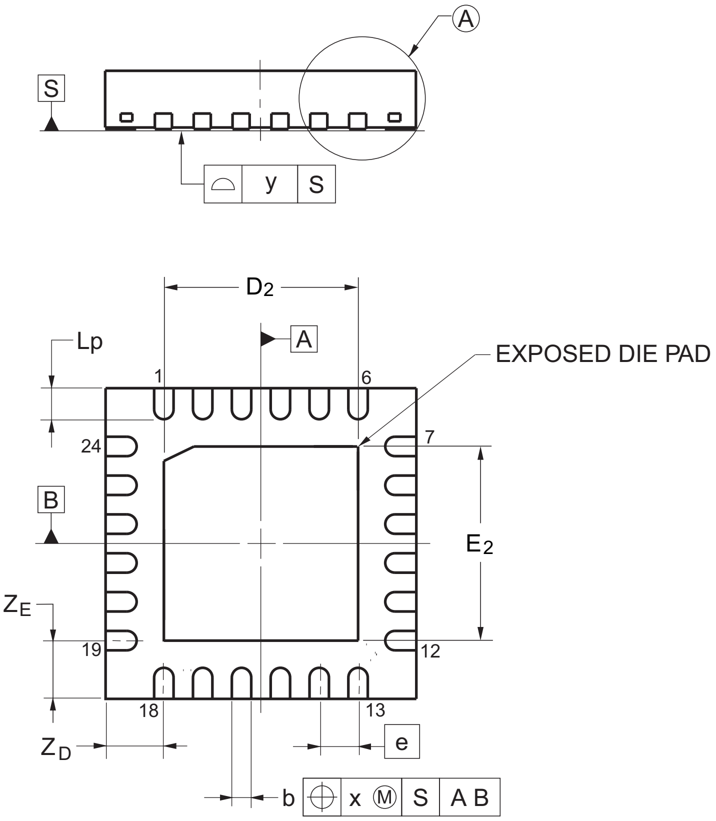
NOTES:
1.DIMENSION 'D' AND 'E1' DOES NOT INCLUDE MOLD FLASH.
2.DIMENSION 'b' DOES NOT INCLUDE TRIM OFFSET.
3.DIMENSION 'D' AND 'E1' TO BE DETERMINED AT DATUM PLANE H.

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		

37.4 24-pin package

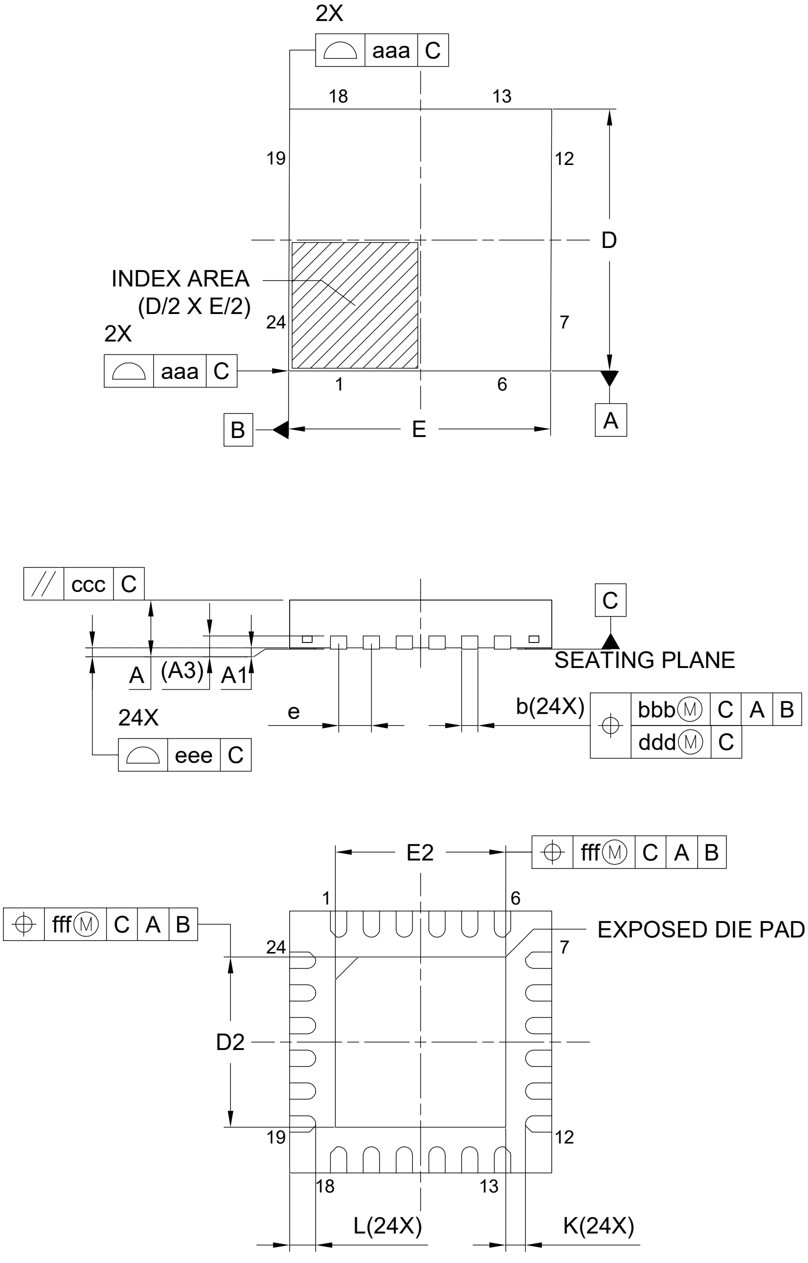
JEITA Package code	RENESAS code	Previous code	MASS(TYP.)[g]
P-HWQFN24-4x4-0.50	PWQN0024KE-A	P24K8-50-CAB-3	0.04





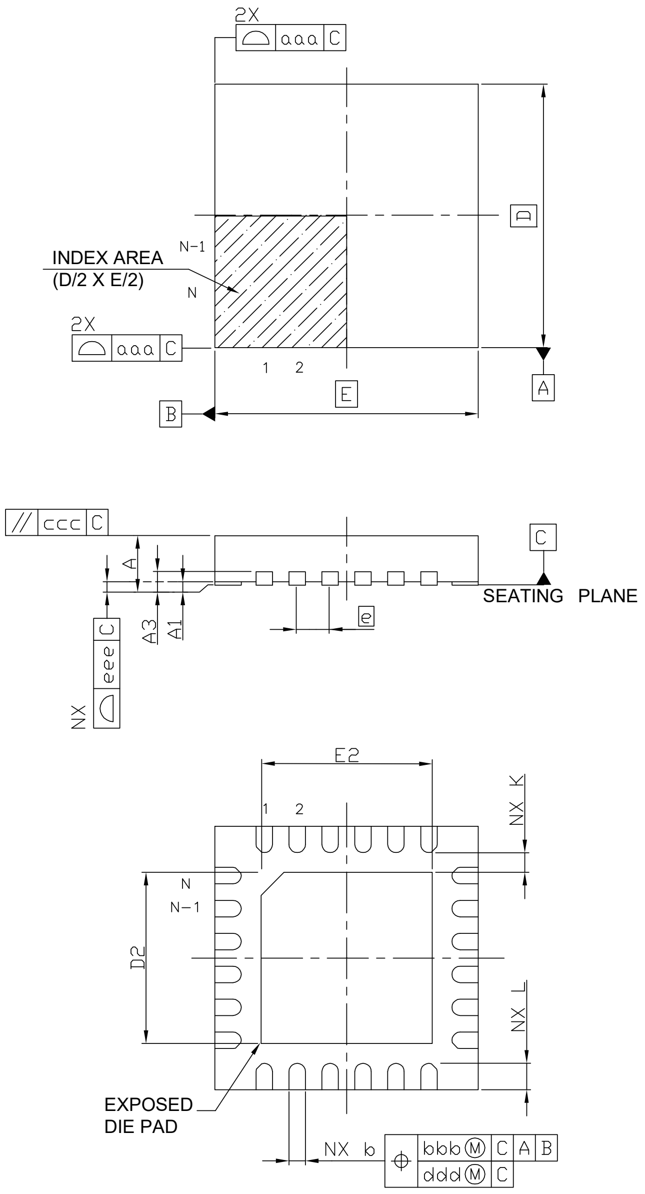
Referance Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	3.95	4.00	4.05
E	3.95	4.00	4.05
A	—	—	0.80
A_1	0.00	—	—
b	0.18	0.25	0.30
e	—	0.50	—
Lp	0.30	0.40	0.50
x	—	—	0.05
y	—	—	0.05
Z_D	—	0.75	—
Z_E	—	0.75	—
c_2	0.15	0.20	0.25
D_2	—	2.50	—
E_2	—	2.50	—

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		

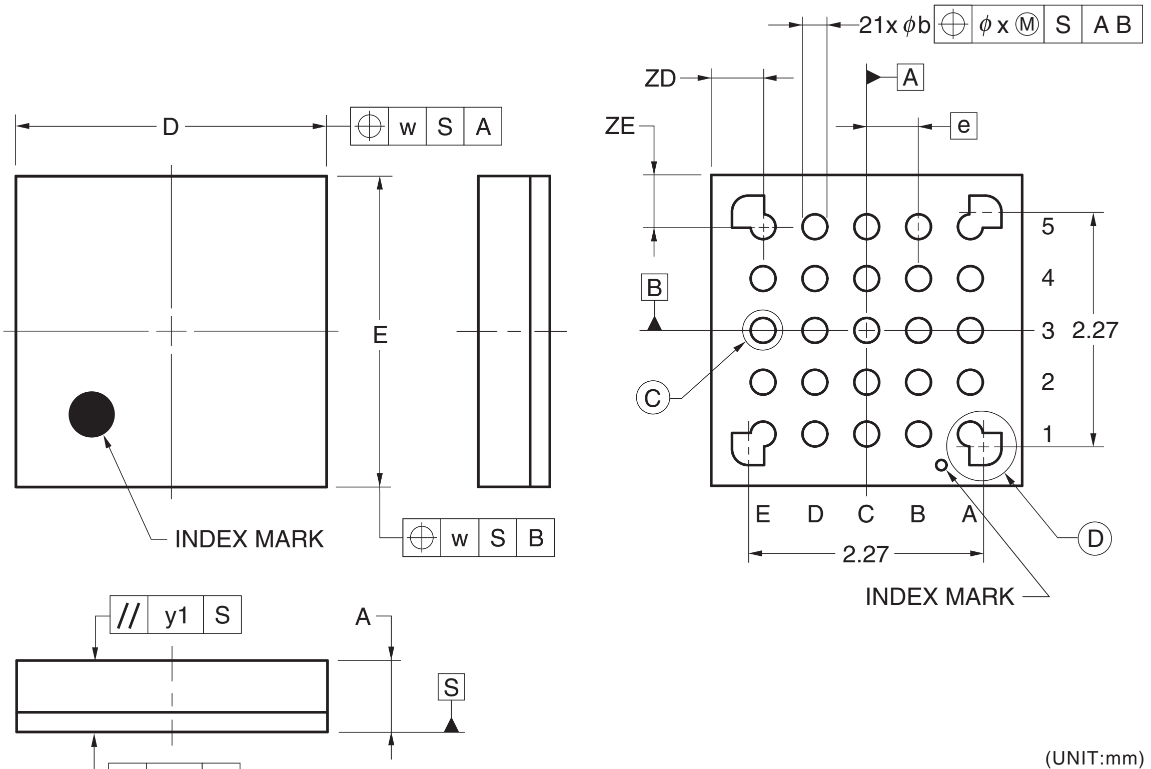
JEITA Package Code	RENESAS Code	MASS(Typ.)[g]
P-HWQFN24-4×4-0.50	PWQN0024KH-A	0.04



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

37.5 25-pin package

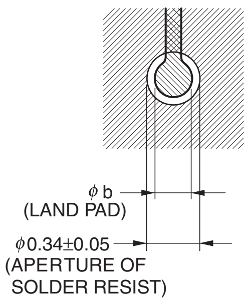
JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-WFLGA25-3x3-0.50	PWLG0025KA-A	P25FC-50-2N2-2	0.01



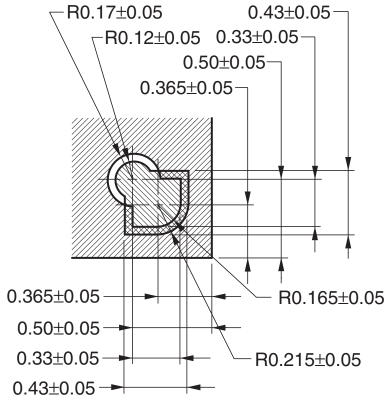
(UNIT:mm)

ITEM	DIMENSIONS
D	3.00±0.10
E	3.00±0.10
w	0.20
e	0.50
A	0.69±0.07
b	0.24±0.05
x	0.05
y	0.08
y1	0.20
ZD	0.50
ZE	0.50

DETAIL OF Ⓒ PART



DETAIL OF Ⓓ PART



APPENDIX A REVISION HISTORY

A.1 Major Revisions in This Edition

Page	Description	Classification
CHAPTER 1 OUTLINE		
p.4	Modification of Figure 1 - 1 Part Number, Memory Size, and Package of RL78/G11	(c)
p.5	Modification of Table 1 - 1 List of Ordering Part Numbers	(c)
CHAPTER 2 PIN FUNCTIONS		
p.28	Modification of Table 2 - 3 Connection of Unused Pins	(c)
CHAPTER 12 WATCHDOG TIMER		
p.409	Addition of Note in Table 12 - 3 Setting of Overflow Time of Watchdog Timer	(c)
CHAPTER 28 SAFETY FUNCTIONS		
p.911	Modification of 28.1 Overview of Safety Functions	(c)
p.911	Modification of Remark in 28.1 Overview of Safety Functions	(c)
p.916	Modification of 28.3.2 CRC operation function (general-purpose CRC)	(c)
p.920	Modification of 28.3.4 RAM guard function	(c)
p.922	Modification of 28.3.5 SFR guard function	(c)

Remark “Classification” in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note,
(d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

A.2 Revision History of Preceding Editions

Here is the revision history of the preceding editions. Chapter indicates the chapter of each edition.

(1/14)

Edition	Description	Chapter
Rev.0.50	First Edition issued.	Throughout
Rev.1.00	Modification of Pin Configuration in 1.3.3 25-pin products	CHAPTER 1 OUTLINE
	Addition of 1.5.1 20-pin products	
	Addition of product name and Modification of Block Diagram in 1.5.2 24-pin, 25-pin products	
	Addition of I^{2}C bus in 1.6 Outline of Functions	
	Modification of Alternate Function in 2.1.1 20-pin Products	CHAPTER 2 PIN FUNCTIONS
	Modification of Remark in 2.2.2 Pins for each product	
	Modification of R/W of P12 and P13 registers in Table 3 - 5 Special Function Register (SFR) List (1/3)	CHAPTER 3 ARCHITECTURE
	Addition of CKSEL register in Table 3 - 5 Special Function Register (SFR) List (2/3)	
	Addition of OSMC register and Deletion of PORSR register in Table 3 - 6 Extended Special Function Register (2nd SFR) List (2/7)	
	Addition of Manipulable Bit Range of TPS2, TPS3 DTCBAR in Table 3 - 6 Extended Special Function Register (2nd SFR) List (5/7)	
	Addition of TRT00, TRT01, DOCR, DODIR and DODSR in Table 3 - 6 Extended Special Function Register (2nd SFR) List (7/7)	
	Modification of PIOR25 and PIOR22 bit in Figure 4 - 7 Format of Peripheral I/O redirection register (PIORx)	CHAPTER 4 PORT FUNCTIONS
	Modification of description in 4.5.2 Register settings for alternate function whose output function is not used	
	Modification of Operating Current in Table 5 - 2 Features of Each Flash Operation Mode	CHAPTER 5 OPERATION STATE CONTROL
	Modification of bit0 in Figure 5 - 2 Format of Flash operating mode select register (FLMODE)	
	Modification of description in 6.3.8 Operation speed mode control register (OSMC)	CHAPTER 6 CLOCK GENERATOR
	Modification of After reset value in Figure 6 - 11 Format of Operation speed mode control register (OSMC)	
	Modification of setting in Figure 6 - 12 Format of High-speed on-chip oscillator frequency select register (HOCODIV)	
	Modification of setting in 6.6.1 Example of setting high-speed on-chip oscillator	
	Modification of CSC register in 6.6.2 Example of setting X1 oscillation clock	
	Modification of (6) in Table 6 - 3 CPU Clock Transition and SFR Register Setting Examples (4/4)	
	Addition of (2) in 7.1.2 Simultaneous channel operation function	CHAPTER 7 TIMER ARRAY UNIT
	Addition of description of bit10 to 8 in Figure 7 - 12 Format of Timer mode register mn (TMRmn) (2/4)	
	Modification of Symbol in Figure 7 - 22 Format of Input switch control register (ISC)	
	Modification of description in 7.8.3 Operation as frequency divider	
	Modification of Figure 7 - 50 Block Diagram of Operation as Frequency Divider	

(2/14)

Edition	Description	Chapter
Rev.1.00	Modification of Figure 7 - 51 Example of Basic Timing of Operation as Frequency Divider (MDmn0 = 1)	CHAPTER 7 TIMER ARRAY UNIT
	Modification of Figure 7 - 52 Example of Set Contents of Registers During Operation as Frequency Divider	
	Modification of Figure 7 - 53 Operation Procedure When Frequency Divider Function Is Used	
	Modification of description in 7.9.2 Two-channel input with one-shot pulse output function	
	Modification of Block Diagram and Addition of Note in Figure 7 - 71 Block Diagram of Operation for Two-channel Input with One-shot Pulse Output Function	
	Modification of bit name and setting value in Figure 7 - 73 Example of Set Contents of Registers for Two-channel Input with One-shot Pulse Output Function (Master Channel)	
	Modification of description in 8.3.3 Operation speed mode control register (OSMC)	CHAPTER 8 12-BIT INTERVAL TIMER
	Modification of after reset value in Figure 8 - 4 Format of operation speed mode control register (OSMC)	
	Modification of description ob bit 11 and 10 in Figure 8 - 6 Format of 12-bit interval timer control register (ITMC)	
	Modification of Symbol in Figure 10 - 19 Format of Peripheral Function Switch Register 0 (PFSEL0)	CHAPTER 10 16-BIT TIMER KB0
	Modification of R/W and Addition of Remark in Figure 10 - 69 Format of Forced Output Stop Function 1 Start Trigger Register n (TKBPAHFSn)	
	Modification of R/W in Figure 10 - 70 Format of Forced Output Stop Function Cancel 1 Trigger Register n (TKBPAHFTn)	
	Modification of Note in Figure 13 - 6 Format of Serial mode register mn (SMRmn) (1/2)	CHAPTER 13 SERIAL ARRAY UNIT
	Modification of Note in Figure 13 - 6 Format of Serial mode register mn (SMRmn) (2/2)	
	Modification of description of EOC bit and Note 1 in Figure 13 - 7 Format of Serial communication operation setting register mn (SCRmn) (1/2)	
	Modification of description of SLCmn1,0 bit and Note 1 in Figure 13 - 7 Format of Serial communication operation setting register mn (SCRmn) (2/2)	
	Modification of Note in Figure 13 - 9 Format of Serial flag clear trigger register mn (SIRmn)	
	Modification of Note in Figure 13 - 10 Format of Serial status register mn (SSRmn) (1/2)	
	Modification of bit3,2 in Figure 13 - 11 Format of Serial channel start register m (SSm)	
	Modification of bit3,2 in Figure 13 - 12 Format of Serial channel stop register m (STm)	
	Modification of bit3,2 in Figure 13 - 13 Format of Serial channel enable status register m (SEm)	
	Modification of bit3,2 in Figure 13 - 14 Format of Serial output enable register m (SOEm)	
	Modification of bit3,2 in Figure 13 - 15 Format of Serial output register m (SOm)	

(3/14)

Edition	Description	Chapter
Rev.1.00	Modification of bit3,2,1 in Figure 13 - 16 Format of Serial output level register m (SOLm)	CHAPTER 13 SERIAL ARRAY UNIT
	Modification of Symbol in Figure 13 - 20 Format of Input switch control register (ISC)	
	Modification of Note 2 in Figure 13 - 108 Example of Contents of Registers for UART Reception of UART (UART0, UART1) (1/2)	
	Addition of PRR0 register in 14.3 Registers Controlling Serial Interface IICA	CHAPTER 14 SERIAL INTERFACE IICA
	Addition of description in 14.3.2 Peripheral reset control register 0 (PRR0)	
	Modification of value of IICSn in 14.5.17 Timing of I2C interrupt request (INTIICAn) occurrence	
	Modification of the reference voltage stabilization wait time A in Figure 16 - 8 Format of A/D converter mode register 2 (ADM2) (1/2)	CHAPTER 16 A/D CONVERTER
	Modification of the reference voltage stabilization wait time A in Figure 16 - 30 Setting up Software Trigger Mode	
	Modification of the reference voltage stabilization wait time A in Figure 16 - 31 Setting up Hardware Trigger No-Wait Mode	
	Modification of the reference voltage stabilization wait time A in Figure 16 - 32 Setting up Hardware Trigger Wait Mode	
	Modification of the reference voltage stabilization wait time A in Figure 16 - 34 Setting up Test Mode	
	Modification of the reference voltage stabilization wait time A in Figure 16 - 38 Flowchart for Setting up SNOOZE Mode	
	Deletion of Caution in 16.10 Cautions for A/D Converter	
	Modification of description in 20.3.3 Vector Table	CHAPTER 20 DATA TRANSFER CONTROLLER (DTC)
	Modification of Figure 20 - 4 Start Address of Control Data and Vector Table	
	Modification of bit of IF0H and IF1H register in Figure 22 - 2 Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)	CHAPTER 22 INTERRUPT FUNCTIONS
	Modification of description in CHAPTER 23 KEY INTERRUPT FUNCTION	CHAPTER 23 KEY INTERRUPT FUNCTION
	Modification of Note 2 in Figure 24 - 3 STOP Mode Release by Interrupt Request Generation (1/2)	CHAPTER 24 STANDBY FUNCTION
	Modification of Note in 24.3.3 SNOOZE mode	
	Addition of Note 4 in Figure 24 - 6 When the Interrupt Request Signal is not Generated in the SNOOZE Mode	
	Addition of description in 28.3.4.1 Invalid memory access detection control register (IAWCTL)	CHAPTER 28 SAFETY FUNCTIONS
	Modification of bit7,5,4 in Figure 28 - 10 Format of Invalid memory access detection control register (IAWCTL)	
	Addition of description in 28.3.6.1 Invalid memory access detection control register (IAWCTL)	

(4/14)

Edition	Description	Chapter
Rev.1.00	Addition of note 3 in Figure 30 - 1 Format of User Option Byte (000C0H/010C0H)	CHAPTER 30 OPTION BYTE
	Deletion of Figure 30 - 3 Format of Option Byte (000C2H/010C2H) (2/2)	
	Modification of Conditions of I_{OH1}, I_{OL1} in 35.1 Absolute Maximum Ratings	CHAPTER 35 ELECTRICAL SPECIFICATIONS (T_A = -40 to +85°C)
	Modification of High-speed on-chip oscillator clock frequency accuracy and addition of D_{IMT}, D_{IMV} in 35.2.2 On-chip oscillator characteristics	
	Modification of Caution in 35.3.1 Pin characteristics	
	Modification of Input voltage, high and Input voltage, low in 35.3.1 Pin characteristics	CHAPTER 35 ELECTRICAL SPECIFICATIONS (T_A = -40 to +85°C)
	Modification of Caution in 35.3.1 Pin characteristics	
	Modification of specifications in 35.3.2 Supply current characteristics	
	Modification of specification in 35.4 AC Characteristics	
	Modification of specifications in 35.5.1 Serial array unit (1)	
	Modification of specifications in 35.5.1 Serial array unit (3)	
	Modification of specification in 35.5.1 Serial array unit (4)	
	Addition of LP (Low-power main) mode in 35.5.2 Serial interface IICA (1)	
	Addition of LP (Low-power main) mode in 35.5.2 Serial interface IICA (2)	
	Addition of LP (Low-power main) mode in 35.5.2 Serial interface IICA (3)	
	Modification of Conditions in 35.6.2 Temperature sensor haracteristics/internal reference voltage characteristic	
	Addition of description in CHAPTER 36 ELECTRICAL SPECIFICATIONS (T_A = -40 to +105°C)	CHAPTER 36 ELECTRICAL SPECIFICATIONS (T_A = -40 to +105°C)
	Modification of High-speed on-chip oscillator clock frequency accuracy and addition of D_{IMT}, D_{IMV} in 36.2.2 On-chip oscillator characteristics	
	Modification of Caution in 36.3.1 Pin characteristics	
	Modification of Input voltage, high and Input voltage, low in 36.3.1 Pin characteristics	
	Modification of Caution in 36.3.1 Pin characteristics	
	Modification of specifications in 36.3.2 Supply current characteristics	
	Modification of specifications and specification table in 36.5.1 Serial array unit (1)	
	Modification of specifications in 36.5.1 Serial array unit (3)	
	Modification of Conditions in 36.6.1 A/D converter characteristics (4)	
	Modification of Conditions in 36.6.2 Temperature sensor haracteristics/internal reference voltage characteristic	
Rev.1.01	Modification of errors in Function in 2.2.2 Pins for each product (pins other than port pins)	CHAPTER 2 PIN FUNCTIONS
	Modification of errors in Figure 10 - 39 Timing Sample for Standalone Mode (Period Controlled by TKBCRn0) (at Default Value of Output Is Low Level (TKBTODnp = 0) and Active Level Is High Level (TKBTOLnp = 0))	CHAPTER 10 16-BIT TIMER KB0
	Modification of errors in Figure 10 - 41 Batch Overwrite Function: Figure of the Timing of Buffer Updating During Counting Operation	
	Modification of errors in Figure 10 - 56 Trigger Output Function for Standalone Mode (Period Controlled by External Trigger Input)	

(5/14)

Edition	Description	Chapter
Rev.1.10	Modification of 1.2 Ordering Information	CHAPTER 1 OUTLINE
	Change of description in 6.4.4 Low-speed on-chip oscillator	CHAPTER 6 CLOCK GENERATOR
	Addition of note in Table 12 - 4 Setting Window Open Period of Watchdog Timer	CHAPTER 12 WATCHDOG TIMER
	Change of note in Figure 16 - 8 Format of A/D converter mode register 2 (ADM2) (1/2)	CHAPTER 16 A/D CONVERTER
	Deletion of note in Figure 16 - 12 Format of Analog input channel specification register (ADS) (1/2)	
	Deletion of note and addition of caution in Figure 16 - 15 Format of A/D test register (ADTES)	
	Addition of note 3 in Figure 30 - 1 Format of User Option Byte (000C0H/010C0H)	CHAPTER 30 OPTION BYTE
Rev.2.00	Addition of specifications of 10-pin and 16-pin products	Throughout
	PG-FP5 has been modified to PG-FP6, FL-PR5 has been modified to FL-PR6, and description of E2, E2 Lite, and E20 has been added.	
	Modification of description in 1.1 Features	CHAPTER 1 OUTLINE
	Modification of figure in 1.3.4 24-pin products	
	Modification of figure in 1.5.3 20-pin products	
	Modification of figure in 1.5.4 24-pin, 25-pin products	
	Modification of table in 1.6 Outline of Functions	
	Modification of table in 2.1.3 20-pin Products	CHAPTER 2 PIN FUNCTIONS
	Modification of table in 2.1.4 24-pin Products	
	Modification of table in 2.1.5 25-pin Products	
	Modification of table in 2.2.2 Pins for each product (pins other than port pins)	
	Modification of Table 2 - 3 Connection of Unused Pins	
	Modification of figure and addition of caution in Figure 2 - 5 Pin Block Diagram of Pin Type 4-9-3	
	Modification of figure in Figure 2 - 7 Pin Block Diagram of Pin Type 4-18-2	
	Modification of note 1 and addition of caution in Figure 3 - 1 Memory Map	CHAPTER 3 CPU ARCHITECTURE
	Addition of caution 3 in Table 3 - 4 Internal RAM Capacity	
	Modification of Table 3 - 6 Extended Special Function Register (2nd SFR) List	
	Modification of description in 3.3.4 Register indirect addressing	
	Modification of Table 4 - 1 Port Configuration	CHAPTER 4 PORT FUNCTIONS
	Modification of description in 4.2.2 Port 2	
	Modification of description in 4.2.4 Port 4	
	Modification of description in 4.2.5 Port 5	
	Modification of table and addition of notes in Table 4 - 2 PMxx, Pxx, PUxx, PIMxx, POMxx, PMCxx registers and the bits mounted on each product	
	Modification of note in 4.3.2 Port registers (Pxx)	
	Modification of Figure 4 - 3 Format of Pull-up resistor option register	
	Modification of notes 1 and 2 in Figure 4 - 4 Format of Port input mode register	
	Modification of description in 4.3.5 Port output mode registers (POMxx)	
	Modification of figure and addition of notes 1 to 4 in Figure 4 - 5 Format of Port output mode register	
	Modification of description in 4.4.4 Handling different potential (1.8 V, 2.5 V, 3.0 V) by using I/O buffers	

(6/14)

Edition	Description	Chapter
Rev.2.00	Modification of Table 4 - 4 Setting Examples of Registers When Using P00 and P01 Pin Function	CHAPTER 4 PORT FUNCTIONS
	Modification of Table 4 - 5 Setting Examples of Registers When Using P20 to P23 Pin Function	
	Modification of Table 4 - 6 Setting Examples of Registers When Using P30 to P33 Pin Function	
	Modification of Table 4 - 7 Setting Examples of Registers When Using P40 Pin Function	
	Modification of Table 4 - 8 Setting Examples of Registers When Using P51 to P56 Pin Function	
	Modification of Table 4 - 9 Setting Examples of Registers When Using P121, P122 and P125 Pin Function	
	Modification of Table 4 - 10 Setting Examples of Registers When Using P137 Pin Function	
	Modification of description and addition of cautions 3 and 10 in Figure 5 - 2 Format of Flash operating mode select register (FLMODE)	CHAPTER 5 OPERATION STATE CONTROL
	Modification of Figure 5 - 5 Format of User option byte (000C2H)	
	Modification of Figure 5 - 6 State Transitions between Flash Operation Modes	
	Modification of figure and addition of remark in Figure 6 - 1 Block Diagram of Clock Generator	CHAPTER 6 CLOCK GENERATOR
	Modification of caution 2 and addition of caution 3 in Figure 6 - 3 Format of System clock control register (CKC)	
	Modification of caution 2 in Figure 6 - 8 Format of Peripheral enable register 0 (PER0)	
	Modification of notes 1 and 2 in Figure 6 - 9 Format of Peripheral enable register 1 (PER1)	
	Modification of notes 1 to 3 in Figure 6 - 10 Format of Peripheral enable register 2 (PER2)	
	Deletion of description in 6.3.8 Operation speed mode control register (OSMC)	
	Modification of note 2 in Figure 6 - 11 Format of Operation speed mode control register (OSMC)	
	Modification of Figure 6 - 12 Format of High-speed on-chip oscillator frequency select register (HOCODIV)	
	Modification of description in 6.4.4 Low-speed on-chip oscillator	
	Modification of figure in 6.6.1 Example of setting high-speed on-chip oscillator	
	Modification of figure and addition of caution in 6.6.2 Example of setting X1 oscillation clock	
	Modification of Table 6 - 4 Changing CPU Clock	
	Modification of Table 6 - 5 Maximum Time Required for Main System Clock Switchover	
	Modification of 7.1.2 (2) Two-channel input with one-shot pulse output function	CHAPTER 7 TIMER ARRAY UNIT
	Modification of Figure 7 - 1 Entire Configuration of Timer Array Unit 0	
	Modification of Figure 7 - 2 Internal Block Diagram of Channel 0 of Timer Array Unit 0	
	Modification of Figure 7 - 3 Internal Block Diagram of Channel 1 of Timer Array Unit 0	
	Modification of Figure 7 - 5 Internal Block Diagram of Channel 3 of Timer Array Unit 0	
	Modification of caution 1 in Figure 7 - 9 Format of Peripheral enable register 0 (PER0)	
	Deletion of caution in Figure 7 - 38 TO0n Pin Statuses by Collective Manipulation of TO0n Bit	
	Modification of Figure 7 - 71 Block Diagram of Operation for Two-channel Input with One-shot Pulse Output Function	

(7/14)

Edition	Description	Chapter
Rev.2.00	Modification of Figure 7 - 73 Example of Set Contents of Registers for Two-channel Input with One-shot Pulse Output Function (Master Channel)	CHAPTER 7 TIMER ARRAY UNIT
	Modification of Table 8 - 1 Configuration of 12-bit Interval Timer	CHAPTER 8 12-BIT INTERVAL TIMER
	Modification of description in 8.3 Registers Controlling 12-bit Interval Timer	
	Modification of title of 8.3.4 Timer clock select register 3 (TPS3)	
	Modification of title in Figure 8 - 5 Format of timer clock select register 3 (TPS3)	
	Modification of description in 9.3.3 8-bit interval timer compare register ni (TRTCMPni)	CHAPTER 9 8-BIT INTERVAL TIMER
	Modification of Figure 9 - 5 Format of 8-bit interval timer compare register n (TRTCMPn)	
	Modification of description in 9.4.2 Timer Operation	
	Modification of description in 9.4.3.2 When Count Source ($f_{IL}/2^m$) is Selected	
	Addition of 9.4.5 Procedure for Setting the 8-bit Interval Timer	
	Modification of description in 10.1 Functions of 16-bit Timer KB0	CHAPTER 10 16-BIT TIMER KB0
	Modification of Table 10 - 1 Configuration of 16-bit Timers KB0	
	Modification of Figure 10 - 1 Block Diagram of 16-bit Timer KB0	
	Modification of Figure 10 - 2 Format of 16-bit Timer KB Compare Register nm (TKBCRnm)	
	Modification of Figure 10 - 3 Format of 16-bit Timer KB Trigger Compare Register n (TKBTGCRn)	
	Modification of description in 10.3 Registers Controlling 16-bit Timers KB0	
	Modification of caution in Figure 10 - 4 Format of Peripheral Enable Register 2 (PER2)	
	Addition of 10.3.2 Peripheral reset control register 2 (PRR2)	
	Modification of table and remark in Figure 10 - 6 Format of Timer Clock Select Register 2 (TPS2)	
	Modification of Figure 10 - 7 Format of 16-bit Timer KB Operation Control Register n0 (TKBCTLn0)	
	Modification of description in 10.3.19 Registers that control port functions of 16-bit timer KB input/output pins	
	Modification of Figure 10 - 22 Timer KB Operation Setting Example (Operation Start Flow)	
	Modification of 10.4.5 (5) Sample of register setting details at standalone mode (period controlled by TKBCRn0)	
	Modification of 10.4.6 (4) Sample of register setting details at standalone mode (period controlled by external trigger input)	
	Modification of 10.4.7 (2) List of register setting at interleave PFC output mode	
	Modification of description in 10.6.1 (2) Start/cancel of Forced Output Stop Function 1 & 2	
	Modification of remark in Figure 10 - 69 Format of Forced Output Stop Function 1 Start Trigger Register n (TKBPAHFSn)	
	Modification of Table 11 - 1 Configuration of Clock Output/Buzzer Output Controller	CHAPTER 11 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER
	Modification of 11.3 Registers Controlling Clock Output/Buzzer Output Controller	
	Modification of description in 11.3.2 Registers that control port functions of clock output/buzzer output pins	
	Modification of description in 11.5 Cautions of clock output/buzzer output controller	

(8/14)

Edition	Description	Chapter
Rev.2.00	Modification of remark in Table 13 - 1 Configuration of Serial Array Unit	CHAPTER 13 SERIAL ARRAY UNIT
	Modification of Figure 13 - 1 Block Diagram of Serial Array Unit 0	
	Modification of note 2 in 13.2.2 Lower 8/9 bits of the serial data register mn (SDRmn)	
	Modification of Figure 13 - 2 Format of Serial data register mn (SDRmn) (mn = 00, 01)	
	Addition of Figure 13 - 3 Format of Serial data register mn (SDRmn) (mn = 02, 03)	
	Modification of caution in Figure 13 - 7 Format of Serial mode register mn (SMRmn)	
	Modification of caution in Figure 13 - 8 Format of Serial communication operation setting register mn (SCRmn)	
	Modification of figure and caution 3 in Figure 13 - 9 Format of Serial data register mn (SDRmn)	
	Modification of caution in Figure 13 - 10 Format of Serial flag clear trigger register mn (SIRmn)	
	Modification of note in Figure 13 - 11 Format of Serial status register mn (SSRmn)	
	Modification of caution 1 in Figure 13 - 12 Format of Serial channel start register m (SSm)	
	Modification of caution in Figure 13 - 13 Format of Serial channel stop register m (STm)	
	Addition of caution in Figure 13 - 14 Format of Serial channel enable status register m (SEm)	
	Modification of caution in Figure 13 - 15 Format of Serial output enable register m (SOEm)	
	Modification of description in 13.3.13 Serial output register m (SOm)	
	Modification of figure and caution in Figure 13 - 16 Format of Serial output register m (SOm)	
	Modification of figure and caution in Figure 13 - 17 Format of Serial output level register m (SOLm)	
	Modification of Figure 13 - 18 Examples of Reverse Transmit Data	
	Addition of description in 13.3.16 Input switch control register (ISC)	
	Modification of caution in Figure 13 - 22 Format of Noise filter enable register 0 (NFEN0)	
	Modification of description in 13.3.18 Registers that control port functions of serial input/output pins	
	Modification of Figure 13 - 24 Each Register Setting When Stopping the Operation by Channels	
	Modification of description in 13.5 Operation of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11) Communication	
	Modification of remark in 13.5.1 Master transmission	
	Modification of Figure 13 - 25 Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11)	
	Modification of Figure 13 - 33 Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11)	
	Modification of Figure 13 - 41 Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11)	
	Modification of note 1 in 13.5.4 Slave transmission	
	Modification of Figure 13 - 49 Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11)	
	Modification of note 1 in 13.5.5 Slave reception	
	Modification of Figure 13 - 57 Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11)	

(9/14)

Edition	Description	Chapter
Rev.2.00	Modification of note 1 in 13.5.6 Slave transmission/reception	CHAPTER 13 SERIAL ARRAY UNIT
	Modification of Figure 13 - 63 Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI01, CSI10, CSI11)	
	Modification of description in 13.5.7 SNOOZE mode function	
	Modification of Figure 13 - 72 Flowchart of SNOOZE Mode Operation (once startup)	
	Modification of Figure 13 - 74 Flowchart of SNOOZE Mode Operation (continuous startup)	
	Modification of caution in Figure 13 - 76 Example of Slave Select Input Function Configuration	
	Modification of Figure 13 - 78 Example of Contents of Registers for Slave Transmission of Slave Select Input Function (CSI00)	
	Modification of Figure 13 - 86 Example of Contents of Registers for Slave Reception of Slave Select Input Function (CSI00)	
	Modification of Figure 13 - 92 Example of Contents of Registers for Slave Transmission/Reception of Slave Select Input Function (CSI00)	
	Modification of description in 13.7 Operation of UART (UART0, UART1) Communication	
	Modification of Figure 13 - 101 Example of Contents of Registers for UART Transmission of UART (UART0, UART1)	
	Modification of Figure 13 - 106 Flowchart of UART Transmission (in Single-Transmission Mode)	
	Modification of table in 13.7.2 UART reception	
	Modification of Figure 13 - 109 Example of Contents of Registers for UART Reception of UART (UART0, UART1)	
	Modification of description in 13.7.3 SNOOZE mode function	
	Modification of Figure 13 - 117 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)	
	Modification of Figure 13 - 119 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)	
	Modification of Figure 13 - 123 Transmission Operation of LIN	
	Modification of Figure 13 - 124 Flowchart for LIN Transmission	
	Modification of Figure 13 - 125 Reception Operation of LIN	
	Modification of Figure 13 - 126 Flowchart for LIN Reception	
	Modification of Figure 13 - 127 Port Configuration for Manipulating Reception of LIN	
	Modification of Figure 13 - 128 Example of Contents of Registers for Address Field Transmission of Simplified I^{2}C (IIC00, IIC01, IIC10, IIC11)	
	Modification of Figure 13 - 132 Example of Contents of Registers for Data Transmission of Simplified I^{2}C (IIC00, IIC01, IIC10, IIC11)	
	Modification of Figure 13 - 135 Example of Contents of Registers for Data Reception of Simplified I^{2}C (IIC00, IIC01, IIC00, IIC01)	
	Modification of Table 14 - 1 Configuration of Serial Interface IICA	CHAPTER 14 SERIAL INTERFACE IICA
	Modification of 14.3 Registers Controlling Serial Interface IICA	
	Modification of caution 1 in Figure 14 - 5 Format of Peripheral enable register 0 (PER0)	
	Modification of description in 14.3.9 Registers that control port functions of IICA input/output pins	

(10/14)

Edition	Description	Chapter
Rev.2.00	Modification of caution 1 in Figure 15 - 2 Format of Peripheral enable register 2 (PER2)	CHAPTER 15 DATA OPERATION CIRCUIT (DOC)
	Addition of description in 15.2.3 DOC control register (DOCR)	
	Modification of table in Figure 16 - 12 Format of Analog input channel specification register (ADS)	CHAPTER 16 A/D CONVERTER
	Deletion of caution in Figure 16 - 33 Setup when temperature sensor output voltage/internal reference voltage is selected	
	Addition of caution 2 in Figure 17 - 2 Format of Peripheral enable register 1 (PER1)	CHAPTER 17 D/A CONVERTER
	Modification of caution 1 in 17.4.1 Operation in Normal Mode	
	Modification of remark in Figure 17 - 6 Timing for Outputting D/A Conversion Value	
	Addition of remark in 17.5 Cautions for D/A Converter	
	Modification of Figure 18 - 1 Block Diagram of Programmable Gain Amplifier	CHAPTER 18 PROGRAMMABLE GAIN AMPLIFIER (PGA)
	Modification of caution 1 in Figure 18 - 2 Format of Peripheral Enable Register 1 (PER1)	
	Modification of (1) and (5) in 18.4 Operation of Programmable Gain Amplifier	
	Modification of Figure 18 - 8 Operation Setting Flow Chart of Programmable Gain Amplifier (PGA)	
	Modification of figure and deletion of caution in Figure 19 - 1 Comparator Block Diagram	CHAPTER 19 COMPARATOR
	Addition of description in 19.3.3 Comparator mode setting register (COMPMDR)	
	Addition of description in 19.3.4 Comparator filter control register (COMPFIR)	
	Modification of Table 19 - 2 Procedure for Setting Comparator Associated Registers	
	Addition of description in 20.5.7 DTC Activation Sources	CHAPTER 20 DATA TRANSFER CONTROLLER (DTC)
	Modification of description in 21.3.1 Event output destination select register n (ELSELRn)	CHAPTER 21 EVENT LINK CONTROLLER (ELC)
	Modification of note 1 in Table 22 - 1 Interrupt Source List	CHAPTER 22 INTERRUPT FUNCTIONS
	Modification of Table 24 - 1 Operating Statuses in HALT Mode	CHAPTER 24 STANDBY FUNCTION
	Modification of Table 24 - 2 Operating Statuses in STOP Mode	
	Modification of description in 24.3.3 SNOOZE mode	
	Modification of Table 24 - 3 Operating Statuses in SNOOZE Mode	
	Addition of description in 25.2.2 Peripheral reset control register 0 (PRR0)	CHAPTER 25 RESET FUNCTION
	Addition of description in 25.2.3 Peripheral reset control register 1 (PRR1)	
	Addition of description in 25.2.4 Peripheral reset control register 2 (PRR2)	
	Modification of Table 27 - 1 Format of User Option Byte (000C1H/010C1H)	CHAPTER 27 VOLTAGE DETECTOR
	Modification of Figure 28 - 11 Invalid access detection area	CHAPTER 28 SAFETY FUNCTIONS
	Modification of Figure 31 - 7 Setting of Flash Memory Programming Mode	CHAPTER 31 FLASH MEMORY
	Modification of 35.2.2 On-chip oscillator characteristics	CHAPTER 35 ELECTRICAL SPECIFICATIONS (T_A = -40 to +85ºC)
	Modification of 35.3.1 Pin characteristics	
	Modification of 35.3.2 Supply current characteristics	
	Modification of 35.4 AC Characteristics	
	Modification of figure in 35.10 Timing of Entry to Flash Memory Programming Modes	

(11/14)

Edition	Description	Chapter
Rev.2.00	Modification of 36.2.1 X1 characteristics	CHAPTER 36 ELECTRICAL SPECIFICATIONS (T_A = -40 to +105ºC)
	Modification of 36.2.2 On-chip oscillator characteristics	
	Modification of 36.3.1 Pin characteristics	
	Modification of 36.4 AC Characteristics	
	Modification of note in 36.5.1 Serial array unit (1)	
	Modification of figure in 36.10 Timing of Entry to Flash Memory Programming Modes	
Rev.2.10	Modification of specification in 35.6.1 A/D converter characteristics, (3) When reference voltage (+) = V_{DD} (ADREFP1 = 0, ADREFP0 = 0), reference voltage (-) = V_{SS} (ADREFM = 0), target pin: ANI0 to ANI3, ANI16 to ANI22, internal reference voltage, and temperature sensor output voltage	CHAPTER 35 ELECTRICAL SPECIFICATIONS (T_A = -40 to +85ºC)
	Modification of specification in 35.6.1 A/D converter characteristics, (4) When reference voltage (+) = Internal reference voltage (ADREFP1 = 1, ADREFP0 = 0), reference voltage (-) = AV_{REFM}/ANI1 (ADREFM = 1), target pin: ANI0, ANI2 and ANI3, ANI16 to ANI22	
	Modification of specification in 36.6.1 A/D converter characteristics, (1) When reference voltage (+) = AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (-) = AV_{REFM}/ANI1 (ADREFM = 1), target pin: ANI2 and ANI3, internal reference voltage, and temperature sensor output voltage	CHAPTER 36 ELECTRICAL SPECIFICATIONS (T_A = -40 to +105ºC)
Rev.2.20	Addition of note in Figure 1 - 1 Part Number, Memory Size, and Package of RL78/G11	CHAPTER 1 OUTLINE
	Modification of figure in 1.3.1 10-pin products	
	Modification of figure in 1.3.2 16-pin products	
	Modification of figure in 1.3.3 20-pin products	
	Modification of figure in 1.3.4 24-pin products	
	Modification of table in 1.6 Outline of Functions	
	Modification of Figure 2 - 11 Pin Block Diagram of Pin Type 8-3-4	CHAPTER 2 PIN FUNCTIONS
	Modification of Table 4 - 5 Setting Examples of Registers When Using P20 to P23 Pin Function	CHAPTER 4 PORT FUNCTIONS
	Modification of description in Figure 7 - 16 Format of Timer channel stop register m (TTm)	CHAPTER 7 TIMER ARRAY UNIT
	Modification of figure and note in Figure 11 - 2 Format of Clock output select registers n (CKSn)	CHAPTER 11 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER
	Modification of description in Figure 16 - 8 Format of A/D converter mode register 2 (ADM2) (2/2)	CHAPTER 16 A/D CONVERTER
	Addition of note in Figure 16 - 12 Format of Analog input channel specification register (ADS)	
	Modification of caution in 16.8 SNOOZE Mode Function	
	Modification of Figure 19 - 1 Comparator Block Diagram	CHAPTER 19 COMPARATOR
	Modification of figure and notes 3 and 5, and addition of note 6 in Figure 19 - 4 Format of Comparator mode setting register (COMPMDR)	
	Modification of Table 19 - 4 Relationship of Comparator 0 and 1 Functions, Register Settings, and Active Signal Width	
	Modification of Figure 19 - 12 Generation Timing of Forced Output Stop Request Signal and Timer Restart Request Signal by Comparator 0 and 1	
	Modification of specification in 35.1 Absolute Maximum Ratings	CHAPTER 35 ELECTRICAL SPECIFICATIONS (T_A = -40 to +85ºC)

(12/14)

Edition	Description	Chapter
Rev.2.20	Modification of specification in 35.3.1 Pin characteristics	CHAPTER 35 ELECTRICAL SPECIFICATIONS (T_A = -40 to +85°C)
	Modification of note 1 in 35.3.2 Supply current characteristics	
	Modification of specification and addition of note 14 in 35.3.2 Supply current characteristics, Peripheral Functions (Common to all products)	
	Modification of specification in 35.4 AC Characteristics	
	Modification of note 2 in 35.5.1 Serial array unit, (1) During communication at same potential (UART mode)	
	Modification of specification in 35.5.1 Serial array unit, (3) During communication at same potential (CSI mode) (master mode, SCKp... internal clock output), When P20 is used as SO10 pin	
	Modification of specification in 35.5.1 Serial array unit, (4) During communication at same potential (CSI mode) (slave mode, SCKp... external clock input), When P01, P32, P53, P54 and P56 are used as SOmn pins	
	Modification of specification in 35.5.1 Serial array unit, (4) During communication at same potential (CSI mode) (slave mode, SCKp... external clock input), When P20 is used as SO10 pin	
	Modification of specification in 35.5.1 Serial array unit, (5) During communication at same potential (simplified I^{2}C mode)	
	Modification of specification in 35.5.1 Serial array unit, (7) Communication at different potential (1.8 V, 2.5 V, 3.0 V) (CSI mode) (master mode, SCKp... internal clock output, corresponding CSI00 only)	
	Modification of note 3 in 35.5.1 Serial array unit, (9) Communication at different potential (1.8 V, 2.5 V, 3.0 V) (CSI mode) (slave mode, SCKp... external clock input)	
	Modification of note 3 in 35.6.1 A/D converter characteristics, (2) When reference voltage (+) = AV_{REFP}/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (-) = AV_{REFM}/ANI1 (ADREFM = 1), target pin: ANI16 to ANI22	
	Modification of title in 35.6.3 D/A converter (channel 1)	
	Modification of specification in 35.6.4 Comparator	
	Modification of specification in 36.1 Absolute Maximum Ratings	CHAPTER 36 ELECTRICAL SPECIFICATIONS (T_A = -40 to +105°C)
	Modification of specification in 36.2.1 X1 characteristics	
	Modification of specification in 36.3.1 Pin characteristics	
	Modification of specification in 36.3.2 Supply current characteristics, Peripheral Functions (Common to all products)	
	Modification of specification in 36.5.1 Serial array unit, (1) during communication at same potential (UART mode), When P20 is used as TxD1 pin	
	Modification of specification in 36.5.1 Serial array unit, (2) During communication at same potential (CSI mode) (master mode, SCKp… internal clock output), When P01, P32, P53, P54 and P56 are used as Somn pins	
	Modification of specification in 36.5.1 Serial array unit, (2) During communication at same potential (CSI mode) (master mode, SCKp... internal clock output), When P20 is used as SO10 pin	
	Modification of note 1 in 36.5.1 Serial array unit, (3) During communication at same potential (CSI mode) (slave mode, SCKp... external clock input), When P01, P32, P53, P54 and P56 are used as SOmn pins	
	Modification of specification and note 1 in 36.5.1 Serial array unit, (3) During communication at same potential (CSI mode) (slave mode, SCKp... external clock input), When P20 is used as SO10 pin	

(13/14)

Edition	Description	Chapter
Rev.2.20	Modification of note 3 in 36.6.1 A/D converter characteristics, (2) When reference voltage (+) = AVREFP/ANI0 (ADREFP1 = 0, ADREFP0 = 1), reference voltage (-) = AVREFM/ANI1 (ADREFM = 1), target pin: ANI16 to ANI22	CHAPTER 36 ELECTRICAL SPECIFICATIONS (TA = -40 to +105ºC)
	Modification of specification in 36.6.1 A/D converter characteristics, (4) When reference voltage (+) = Internal reference voltage (ADREFP1 = 1, ADREFP0 = 0), reference voltage (-) = AVREFM/ANI1 (ADREFM = 1), target pin: ANI0 to ANI3, ANI16 to ANI22	
	Modification of title in 36.6.3 D/A converter (channel 1)	
	Modification of specification in 36.6.4 Comparator	
	Modification of specification in 36.6.6 POR circuit characteristics	
	Modification of specification in 36.6.7 LVD circuit characteristics, (1) LVD Detection Voltage of Reset Mode and Interrupt Mode	
Rev.2.30	Modification of Figure 1 - 1 Part Number, Memory Size, and Package of RL78/G11	CHAPTER 1 OUTLINE
	Modification of table in 1.2 Ordering Information	
	Modification of description in 1.3.3 20-pin products	
	Modification of note in Figure 8 - 2 Format of Peripheral enable register 2 (PER2)	CHAPTER 8 12-BIT INTERVAL TIMER
	Modification of description in 12.4.4 Setting watchdog timer interval interrupt	CHAPTER 12 WATCHDOG TIMER
	Modification of description in Table 12 - 5 Setting of Watchdog Timer Interval Interrupt	
	Modification of caution 1 in Figure 13 - 11 Format of Serial status register mn (SSRmn) (2/2)	CHAPTER 13 SERIAL ARRAY UNIT
	Modification of description in Figure 13 - 19 Format of Serial standby control register m (SSCm)	
	Modification of Figure 13 - 32 Flowchart of Master Transmission (in Continuous Transmission Mode)	
	Modification of Figure 13 - 108 Flowchart of UART Transmission (in Continuous Transmission Mode)	
	Modification of note 1 in Figure 16 - 5 Timing Chart When A/D Voltage Comparator Is Used	CHAPTER 16 A/D CONVERTER
	Modification of description in Figure 30 - 1 Format of User Option Byte (000C0H/010C0H)	CHAPTER 30 OPTION BYTE
	Addition of caution 4 in 31.8.3 Procedure for accessing data flash memory	CHAPTER 31 FLASH MEMORY
	Modification of specification in 35.3.2 Supply current characteristics	CHAPTER 35 ELECTRICAL SPECIFICATIONS (TA = -40 to +85°C)
	Addition of package drawing in 37.3 20-pin package	CHAPTER 37 PACKAGE DRAWINGS
	Addition of package drawing in 37.4 24-pin package	
Rev.2.40	Modification of Figure 1 - 1 Part Number, Memory Size, and Package of RL78/G11	CHAPTER 1 OUTLINE
	Modification of table in 1.2 Ordering Information	
	Addition of description and figure in 1.3.2 16-pin products	
	Addition of package drawing in 37.2 16-pin package	CHAPTER 37 PACKAGE DRAWINGS
Rev.2.50	"3-Wire Serial I/O" and "3-wire serial" were modified to " Simplified SPI"	All
	The module name for CSI was changed to Simplified SPI	
	"wait" for IIC was modified to "clock stretch"	
	Addition of Note 1 in 1 Features	CHAPTER 1 OUTLINE
	Addition of Table Title and modification of Table 1-1 List of Ordering Part Numbers	

(14/14)

Edition	Description	Chapter
Rev.2.50	Addition of Note in 4.4.4 Handling different potential (1.8 V, 2.5 V) by using I/O buffers	CHAPTER 4 PORT FUNCTIONS
	Addition of Note and modification of Remark in Figure 10 - 7 Format of 16-bit Timer KB Operation Control Register n0 (TKBCTLn0)	CHAPTER 10 16-BIT TIMER KB0
	Addition of Note 1 and modification of Remark in Figure 10 - 20 Format of Peripheral Function Switch Register 0 (PFSEL0)	
	Addition of Note 3 and modification of Remark in Figure 10 - 66 Format of Forced Output Stop Function Control Register 0p (TKBPACTL0p)	
	Addition of Note in CHAPTER 13 SERIAL ARRAY UNIT	CHAPTER 13 SERIAL ARRAY UNIT
	Modification of Figure 19 - 1 Comparator Block Diagram	CHAPTER 19 COMPARATOR
	Addition of Note 4 and Remark in Figure 19 - 6 Format of Comparator output control register (COMPOCR)	
	Modification of description in 19.5 Caution for Using Timer KB Simultaneous Operation Function	
	Modification of Table 19 - 4 Relationship of Comparator 0 and 1 Functions, Register Settings, and Active Signal Width	
	Modification of Note 6 and Remark in Figure 19 - 12 Generation Timing of Forced Output Stop Request Signal and Timer Restart Request Signal by Comparator 0 and 1	
	Modification of description in 20.5.1 Setting DTC Control Data and Vector Table	CHAPTER 20 DATA TRANSFER CONTROLLER (DTC)
	Modification of Note and Addition of Remark in Table of 20.5.8 Operation in Standby Mode Status	
	Modification of Note 1 in Table (T_A = -40 to +85°C, 1.6 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) (2/4) of 35.3.2 Supply current characteristics	CHAPTER 35 ELECTRICAL SPECIFICATIONS (T_A = -40 to +85°C)
	Modification of Note 1 in Table (T_A = -40 to +85°C, 1.6 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) (3/4) of 35.3.2 Supply current characteristics	
	Modification of Notes in Table (T_A = -40 to +85°C, 1.6 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) (4/4) of 35.3.2 Supply current characteristics	
	Modification of Note 1 in Table (T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) (1/3) of 36.3.2 Supply current characteristics	CHAPTER 36 ELECTRICAL SPECIFICATIONS (T_A = -40 to +105°C)
	Modification of Note 1 in Table (T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) (2/3) of 36.3.2 Supply current characteristics	
	Modification of Notes in Table (T_A = -40 to +105°C, 2.4 V ≤ EV_{DD} ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V) (3/3) of 36.3.2 Supply current characteristics	
	Addition of package drawing in 37.2 16-pin package	CHAPTER 37 PACKAGE DRAWINGS
	Addition of package drawing in 37.4 24-pin package	

RL78/G11 User’s Manual: Hardware

Publication Date: Rev.0.50 Mar 31, 2016
Rev.2.60 Mar 29, 2024

Published by: Renesas Electronics Corporation

RL78/G11

Renesas Electronics Corporation