User's Manual

16

RL78/G16

User's Manual: Hardware

16-Bit Single-Chip Microcontrollers

Renesas Electronics
www.renesas.com

Rev.1.30 May, 2025

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

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

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

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

www.renesas.com

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/G16 and design and develop application systems and programs for these devices.

The target products are as follows.

- 10-pin: R5F1211x (x = A, C)
- 16-pin: R5F1214x (x = A, C)
- 20-pin: R5F1216x (x = A, C)
- 24-pin: R5F1217x (x = A, C)
- 32-pin R5F121Bx (x = A, C)

Purpose

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

Organization

The RL78/G16 manual is separated into two parts: this manual and the software edition (common to the RL78 family).

RL78/G16 User's Manual Hardware	RL78 Family User's Manual Software
● Pin functions ● Internal block functions ● Interrupts ● Other on-chip peripheral functions ● Electrical specifications	● CPU functions ● Instruction set ● Explanation of each instruction

How to Read This Manual

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

- To gain a general understanding of functions:
  → Read this manual in the order of the CONTENTS. 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/G16 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/G16 User's Manual Hardware	R01UH0980E
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
E2 Emulator User's Manual	R20UT3538E
E2 Lite Emulator User's Manual	R20UT3240E
Renesas Flash Programmer Flash Memory Programming Software User's Manual	R20UT4066E

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.

Table of Contents

CHAPTER 1 OUTLINE

1.1 Features

Low power consumption technology

- V_{DD} = single power supply voltage of 2.4 to 5.5 V
- HALT mode
- STOP mode

RL78 CPU core

- CISC architecture with 3-stage pipeline
- Minimum instruction execution time: Can be changed from high speed (0.0625 µs: @ 16 MHz operation with high-speed on-chip oscillator) to ultra-low speed (30.5 µs: @ 32.768 kHz operation with subsystem clock)
- Address space: 1 MB
- General-purpose registers: (8-bit register × 8) × 4 banks
- On-chip RAM: 2 KB

Code flash memory

- Code flash memory: 16 to 32 KB
- Block size: 1 KB
- Only write after erase is possible
- On-chip debug function
- Self-programming (with no boot swap function/flash shield window function)

Data flash memory

- Data flash memory: 1 KB
- Block size: 512 B
- Unit of rewrites: 32 bits
- Background operation (BGO) is not supported (instructions cannot be executed from the code flash memory while rewriting the data flash memory)
- Number of rewrites: 1,000,000 times (TYP.)
- Voltage of rewrites: V_{DD} = 2.4 to 5.5 V

High-speed on-chip oscillator

- Select from 16 MHz, 8 MHz, 4 MHz, 2 MHz, and 1 MHz
- Frequency accuracy ±1.0% (V_{DD} = 2.4 to 5.5 V, T_A = −20 to +85°C)
  (G: Industrial applications, M: Industrial applications)
- Frequency accuracy ±1.5% (V_{DD} = 2.4 to 5.5 V, T_A = −40 to −20°C)
  (G: Industrial applications, M: Industrial applications)
- Frequency accuracy ±1.5% (V_{DD} = 2.4 to 5.5 V, T_A = +85 to +105°C)
  (G: Industrial applications)
- Frequency accuracy ±1.5% (V_{DD} = 2.4 to 5.5 V, T_A = +85 to +125°C)
  (M: Industrial applications)
- Frequency accuracy ±2.0% (V_{DD} = 2.4 to 5.5 V, T_A = −40 to +85°C)
  (A: Consumer applications)

Operating ambient temperature

- T_A = −40 to +85°C (A: Consumer applications)
- T_A = −40 to +105°C (G: Industrial applications)
- T_A = −40 to +125°C (M: Industrial applications)

Power management and reset function

- On-chip selectable power-on-reset (SPOR) circuit (Select reset from 3 levels, stop setting is available)

Serial interface

- Simplified SPI (CSI[Note 1]): 1 to 3 channels
- UART: 1 to 3 channel
- Simplified I^2C: 1 to 3 channels
- I^2C: 1 channel

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.

Timer

- 16-bit timer: 8 channels
- 12-bit interval timer: 1 channel
- Watchdog timer: 1 channel (operable with the dedicated low-speed on-chip oscillator)
- Real-time clock 2: 1 channel (99-year calendar, alarm function, and clock correction function)

A/D converter

- 8/10-bit resolution A/D converter (V_{DD} = 2.4 to 5.5 V)
- Analog input: 4 to 11 channels

- Internal reference voltage (0.815 V (TYP.)), temperature sensor, and touch TSCAP voltage selection

Comparator

- 1 to 2 channels
- Operation mode: High-speed mode, low-speed mode
- External reference voltage or internal reference voltage can be selected as the reference voltage.

Capacitive touch sensing unit (CTSUb)

- 15 channels
- Self-capacitance method: A single pin configures a single key, supporting up to 15 keys
<R> - Mutual capacitance method: A key can be created with a matrix configuration by selecting transmit/receive pins from 15 pins. Up to 16 keys are supported.

I/O port

- I/O port: 8 to 30 (N-ch open drain I/O [withstand voltage of 6 V]: 0 to 2)
- Can be set to N-ch open drain and on-chip pull-up resistor
- External interrupt function: 10 channels
- On-chip clock output/buzzer output controller

Others

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

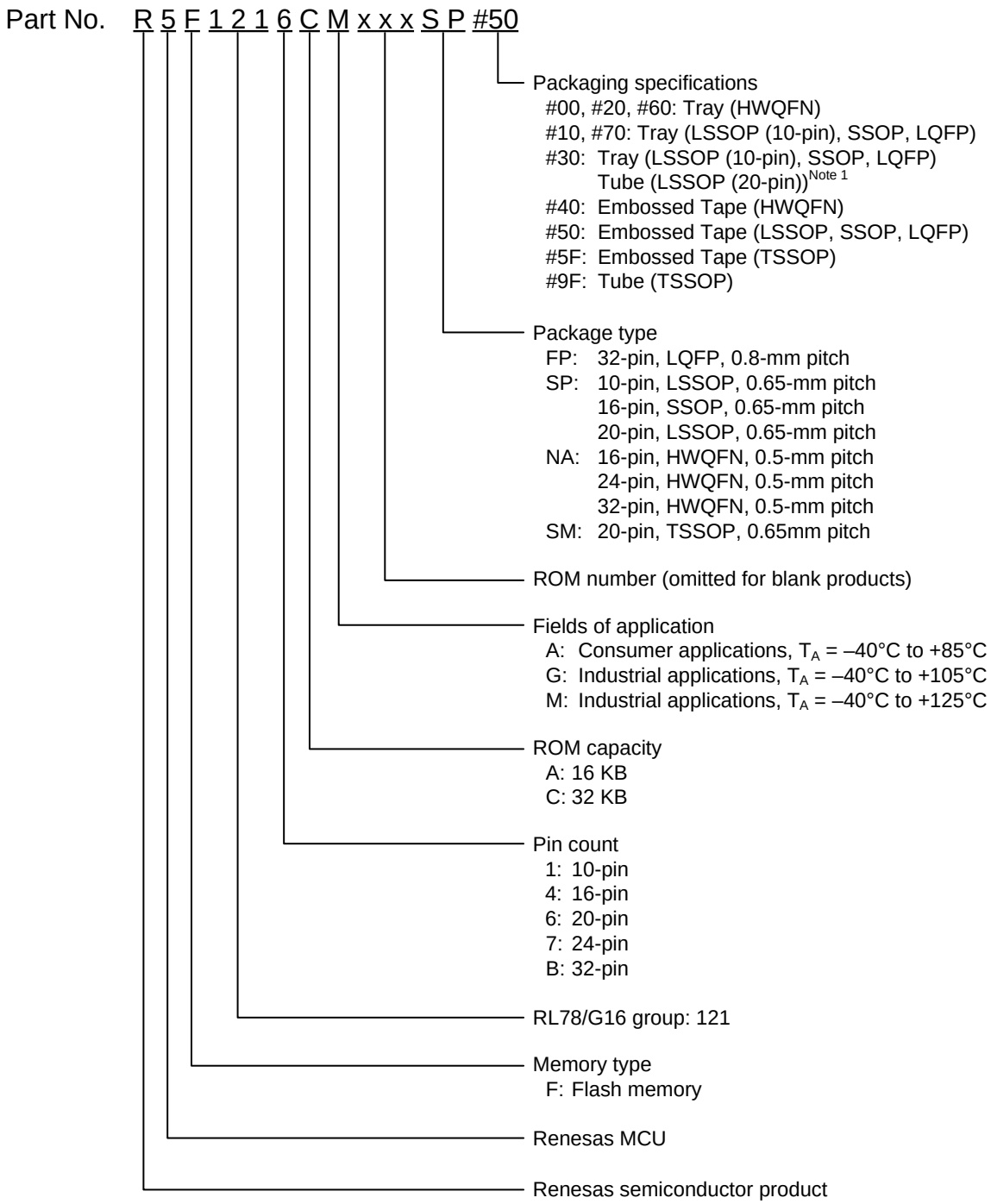
Remark The functions mounted depend on the product. See **1.6 Outline of Functions**.

ROM, RAM capacities

Flash ROM	Data flash	RAM	RL78/G16				
			10 pins	16 pins	20 pins	24 pins	32 pins
32 KB	1 KB	2 KB	R5F1211C	R5F1214C	R5F1216C	R5F1217C	R5F121BC
16 KB	1 KB	2 KB	R5F1211A	R5F1214A	R5F1216A	R5F1217A	R5F121BA

1.2 List of Part Numbers

<R> Figure 1-1. Part Number, Memory Size, and Package of RL78/G16



<R> Note 1. The packaging specification of the 20-pin LSSOP products is Tube.

Table 1-1. List of Ordering Part Numbers

Pin count	Package	Fields of Application Note 1	Ordering Part Number		RENESAS Code
			Product Name	Packaging Specifications	
10 pins	10-pin plastic LSSOP (4.4 × 3.6 mm, 0.65-mm pitch)	A	R5F1211CASP, R5F1211AASP	#10, #30, #50, #70	PLSP0010JA-A
		G	R5F1211CGSP, R5F1211AGSP		
		M	R5F1211CMSP, R5F1211AMSP		
16 pins	16-pin plastic SSOP (4.4 × 5.0 mm, 0.65-mm pitch)	A	R5F1214CASP, R5F1214AASP	#10, #30, #50, #70	PRSP0016JC-B
		G	R5F1214CGSP, R5F1214AGSP		
		M	R5F1214CMSP, R5F1214AMSP		
16 pins	16-pin plastic HWQFN (3 × 3 mm, 0.5-mm pitch)	A	R5F1214CANA, R5F1214AANA	#00, #20, #40, #60	PWQN0016KD-A
		G	R5F1214CGNA, R5F1214AGNA		
		M	R5F1214CMNA, R5F1214AMNA		
20 pins	20-pin plastic LSSOP (4.4 × 6.5 mm, 0.65-mm pitch)	A	R5F1216CASP, R5F1216AASP	#30, #50	PLSP0020JB-A
		G	R5F1216CGSP, R5F1216AGSP		
		M	R5F1216CMSP, R5F1216AMSP		
<R> 20 pins	20-pin plastic TSSOP (4.4 × 6.5 mm, 0.65-mm pitch)	A	R5F1216CASM	#5F, #9F	PTSP0020JI-A
		M	R5F1216CMSM		
24 pins	24-pin plastic HWQFN (4.0 × 4.0 mm, 0.5-mm pitch)	A	R5F1217CANA, R5F1217AANA	#00, #20, #40, #60	PWQN0024KF-A
		G	R5F1217CGNA, R5F1217AGNA		
		M	R5F1217CMNA, R5F1217AMNA		
32 pins	32-pin plastic HWQFN (5.0 × 5.0 mm, 0.5-mm pitch)	A	R5F121BCANA, R5F121BAANA	#00, #20, #40, #60	PWQN0032KE-A
		G	R5F121BCGNA, R5F121BAGNA		
		M	R5F121BCMNA, R5F121BAMNA		
32 pins	32-pin plastic LQFP (7.0 × 7.0 mm, 0.8-mm pitch)	A	R5F121BCAFP, R5F121BAAFP	#10, #30, #50, #70	PLQP0032GB-A
		G	R5F121BCGFP, R5F121BAGFP		
		M	R5F121BCMFP, R5F121BAMFP		

Note 1. For the fields of application, refer to **Figure 1-1 Part Number, Memory Size, and Package of RL78/G16**.

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

1.3 Pin Configuration (Top View)

1.3.1 10-pin products

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

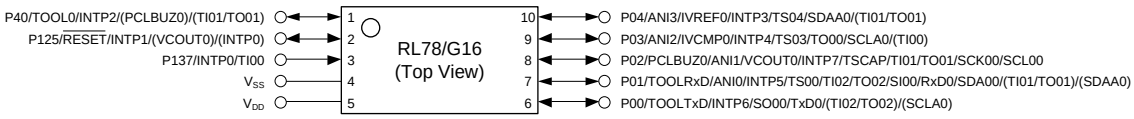


Table 1-2. Multiplexed Functions of 10-pin Products

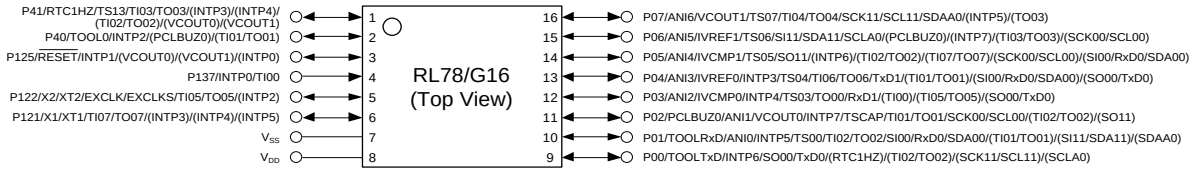
Pin No.	I/O		Analog		HMI		Timer	Communications Interface	
10LSSOP	Digital port	Power supply, system, clock, debug	A/D converter	Comparator	Interrupt function	Capacitive touch sensing unit (CTSU)	Timer array unit	Serial array unit	Serial interface IICA
1	P40	TOOL0 (PCLBUZ0)	—	—	INTP2	—	(TI01/TO01)	—	—
2	P125	$\overline{\text{RESET}}$	—	(VCOUT0)	INTP1 (INTP0)	—	—	—	—
3	P137	—	—	—	INTP0	—	TI00	—	—
4	—	V_{SS}	—	—	—	—	—	—	—
5	—	V_{DD}	—	—	—	—	—	—	—
6	P00	TOOLTxD	—	—	INTP6	—	(TI02/TO02)	SO00/TxD0	(SCLA0)
7	P01	TOOLRxD	ANI0	—	INTP5	TS00	TI02/TO02 (TI01/TO01)	SI00/RxD0/ SDA00	(SDAA0)
8	P02	PCLBUZ0	ANI1	VCOUT0	INTP7	TSCAP	TI01/TO01	SCK00/SCL00	—
9	P03	—	ANI2	IVCMP0	INTP4	TS03	TO00 (TI00)	—	SCLA0
10	P04	—	ANI3	IVREF0	INTP3	TS04	(TI01/TO01)	—	SDAA0

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

Remark 2. Functions in parentheses can be assigned via settings in the peripheral I/O redirection register (PIOR). Refer to **Figure 4-6 Format of Peripheral I/O Redirection Registers 0 to 6 (PIOR0 to PIOR6)**.

1.3.2 16-pin products

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



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

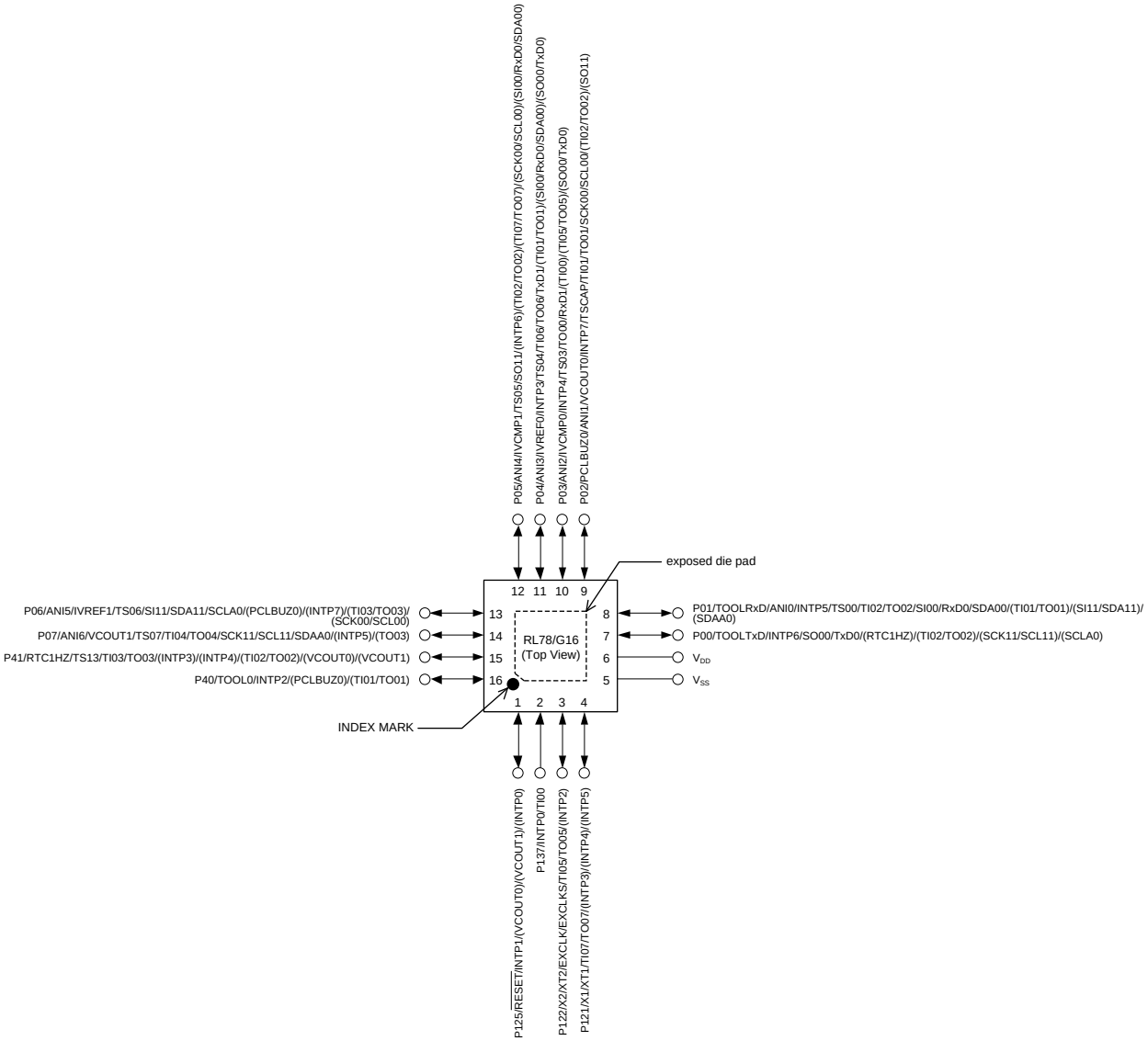


Table 1-3. Multiplexed Functions of 16-pin Products

Pin No.		I/O		Analog		HMI		Timer	Communications Interface	
16SSOP	16HWQFN	Digital port	Power supply, system, clock, debug	A/D converter	Comparator	Interrupt function	Capacitive touch sensing unit (CTSU)	Timer array unit	Serial array unit	Serial interface IICA
1	15	P41	RTC1HZ	—	(VCOUT0) (VCOUT1)	(INTP3) (INTP4)	TS13	TI03/TO03 (TI02/TO02)	—	—
2	16	P40	TOOL0 (PCLBUZ0)	—	—	INTP2	—	(TI01/TO01)	—	—
3	1	P125	$\overline{\text{RESET}}$	—	(VCOUT0) (VCOUT1)	INTP1 (INTP0)	—	—	—	—
4	2	P137	INTP0	—	—	—	—	TI00	—	—
5	3	P122	X2/XT2/ EXCLK/ EXCLKS	—	—	(INTP2)	—	TI05/TO05	—	—
6	4	P121	X1/XT1	—	—	(INTP3) (INTP4) (INTP5)	—	TI07/TO07	—	—
7	5	—	V_{SS}	—	—	—	—	—	—	—
8	6	—	V_{DD}	—	—	—	—	—	—	—
9	7	P00	TOOLTxD (RTC1HZ)	—	—	INTP6	—	(TI02/TO02)	SO00/TxD0 (SCK11/SCL11)	(SCLA0)
10	8	P01	TOOLRxD	ANI0	—	INTP5	TS00	TI02/TO02 (TI01/TO01)	SI00/RxD0/ SDA00 (SI11/SDA11)	(SDAA0)
11	9	P02	PCLBUZ0	ANI1	VCOUT0	INTP7	TSCAP	TI01/TO01 (TI02/TO02)	SCK00/SCL00 (SO11)	—
12	10	P03	—	ANI2	IVCMP0	INTP4	TS03	TO00 (TI00) (TI05/TO05)	RxD1 (SO00/TxD0)	—
13	11	P04	—	ANI3	IVREF0	INTP3	TS04	TI06/TO06 (TI01/TO01)	TxD1 (SI00/RxD0/ SDA00) (SO00/TxD0)	—
14	12	P05	—	ANI4	IVCMP1	(INTP6)	TS05	(TI02/TO02) (TI07/TO07)	SO11 (SCK00/SCL00) (SI00/RxD0/ SDA00)	—
15	13	P06	(PCLBUZ0)	ANI5	IVREF1	(INTP7)	TS06	(TI03/TO03)	SI11/SDA11 (SCK00/SCL00)	SCLA0
16	14	P07	—	ANI6	VCOUT1	(INTP5)	TS07	TI04/TO04 (TO03)	SCK11/SCL11	SDAA0

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

Remark 2. Functions in parentheses can be assigned via settings in the peripheral I/O redirection register (PIOR). Refer to **Figure 4-6 Format of Peripheral I/O Redirection Registers 0 to 6 (PIOR0 to PIOR6)**.

Remark 3. For the product in a QFN package, solder the exposed die pad onto a plated area of the PCB that has no electrical connections.

1.3.3 20-pin products

<R> ● 20-pin plastic LSSOP/TSSOP (4.4 × 6.5 mm, 0.65-mm pitch)

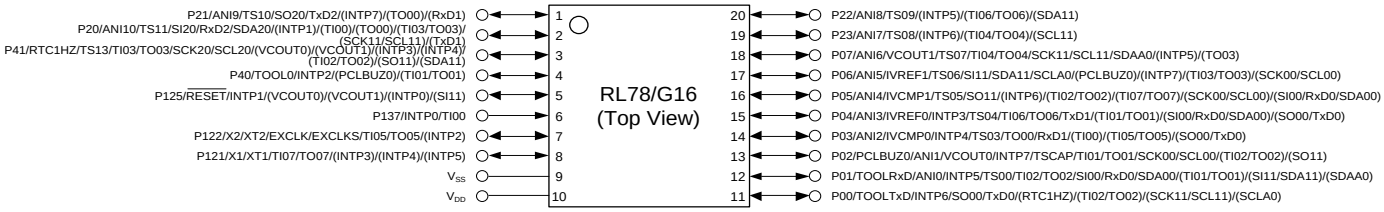


Table 1-4. Multiplexed Functions of 20-pin Products (1/2)

Pin No.	I/O		Analog		HMI		Timer	Communications Interface	
20LSSOP/ 20TSSOP	Digital port	Power supply, system, clock, debug	A/D converter	Comparator	Interrupt function	Capacitive touch sensing unit (CTSU)	Timer array unit	Serial array unit	Serial interface IICA
1	P21	—	ANI9	—	(INTP7)	TS10	(TO00)	SO20/TxD2 (RxD1)	—
2	P20	—	ANI10	—	(INTP1)	TS11	(TI00) (TO00) (TI03/TO03)	SI20/RxD2/SDA20 (SCK11/SCL11) (TxD1)	—
3	P41	RTC1HZ	—	(VCOUT0) (VCOUT1)	(INTP3) (INTP4)	TS13	TI03/TO03 (TI02/TO02)	SCK20/SCL20 (SO11)/(SDA11)	—
4	P40	TOOL0 (PCLBUZ0)	—	—	INTP2	—	(TI01/TO01)	—	—
5	P125	$\overline{\text{RESET}}$	—	(VCOUT0) (VCOUT1)	INTP1 (INTP0)	—	—	(SI11)	—
6	P137	—	—	—	INTP0	—	TI00	—	—
7	P122	X2/XT2 EXCLK/EXCLKS	—	—	(INTP2)	—	TI05/TO05	—	—
8	P121	X1/XT1	—	—	(INTP3) (INTP4) (INTP5)	—	TI07/TO07	—	—
9	—	V_{SS}	—	—	—	—	—	—	—
10	—	V_{DD}	—	—	—	—	—	—	—
11	P00	TOOLTxD (RTC1HZ)	—	—	INTP6	—	(TI02/TO02)	SO00/TxD0 (SCK11/SCL11)	(SCLA0)
12	P01	TOOLRxD	ANI0	—	INTP5	TS00	TI02/TO02 (TI01/TO01)	SI00/RxD0/SDA00 (SI11/SDA11)	(SDAA0)
13	P02	PCLBUZ0	ANI1	VCOUT0	INTP7	TSCAP	TI01/TO01 (TI02/TO02)	SCK00/SCL00 (SO11)	—
14	P03	—	ANI2	IVCMP0	INTP4	TS03	TO00 (TI00) (TI05/TO05)	RxD1 (SO00/TxD0)	—
15	P04	—	ANI3	IVREF0	INTP3	TS04	TI06/TO06 (TI01/TO01)	TxD1 (SI00/RxD0/SDA00) (SO00/TxD0)	—
16	P05	—	ANI4	IVCMP1	(INTP6)	TS05	(TI02/TO02) (TI07/TO07)	SO11 (SCK00/SCL00) (SI00/RxD0/SDA00)	—

Table 1-4. Multiplexed Functions of 20-pin Products (2/2)

Pin No.	I/O		Analog		HMI		Timer	Communications Interface	
20LSSOP/ 20TSSOP	Digital port	Power supply, system, clock, debug	A/D converter	Comparator	Interrupt function	Capacitive touch sensing unit (CTSU)	Timer array unit	Serial array unit	Serial interface IICA
17	P06	(PCLBUZ0)	ANI5	IVREF1	(INTP7)	TS06	(TI03/TO03)	SI11/SDA11 (SCK00/SCL00)	SCLA0
18	P07	—	ANI6	VCOUT1	(INTP5)	TS07	TI04/TO04 (TO03)	SCK11/SCL11	SDAA0
19	P23	—	ANI7	—	(INTP6)	TS08	(TI04/TO04)	(SCL11)	—
20	P22	—	ANI8	—	(INTP5)	TS09	(TI06/TO06)	(SDA11)	—

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

Remark 2. Functions in parentheses can be assigned via settings in the peripheral I/O redirection register (PIOR). Refer to **Figure 4-6 Format of Peripheral I/O Redirection Registers 0 to 6 (PIOR0 to PIOR6)**.

1.3.4 24-pin products

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

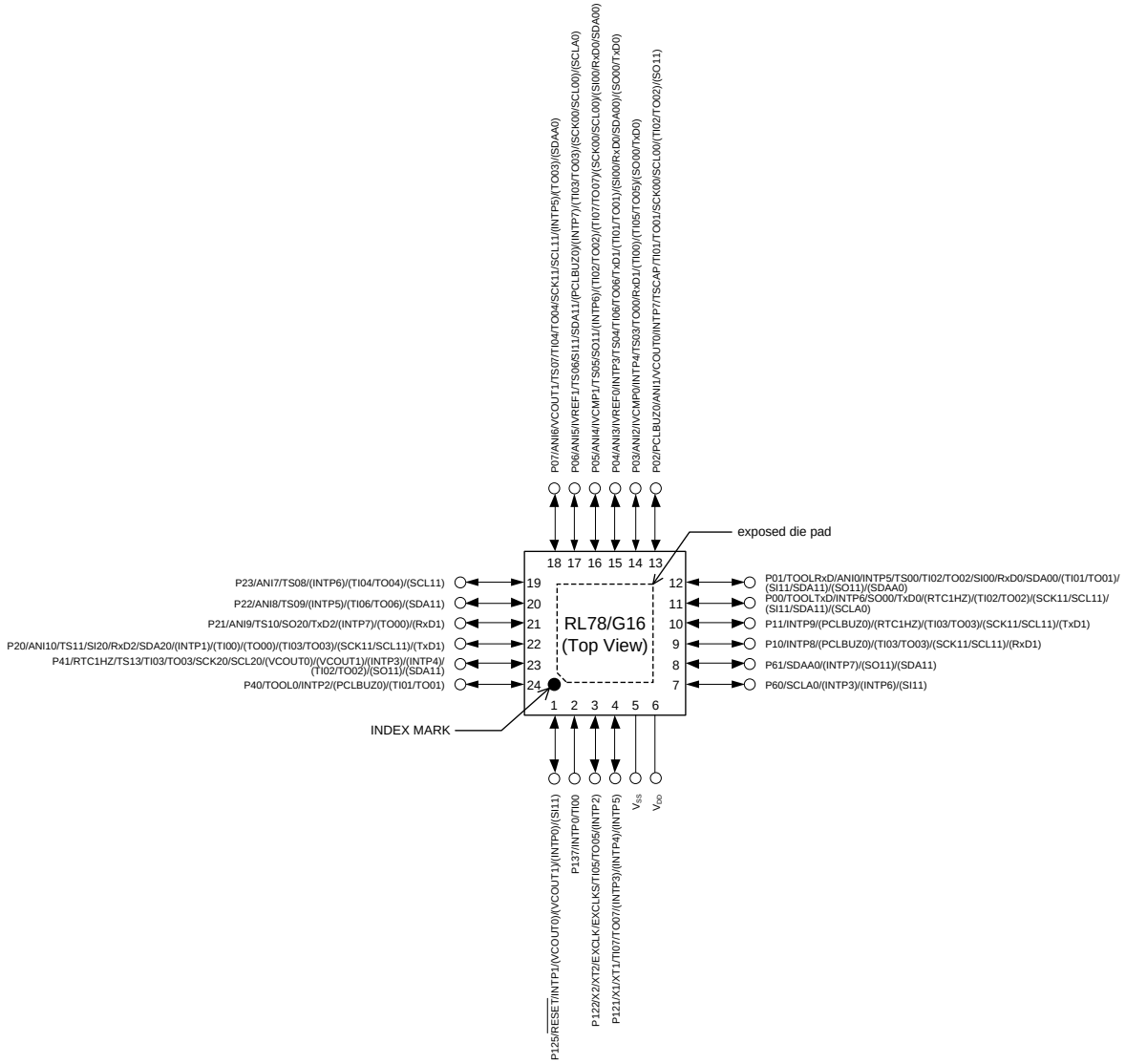


Table 1-5. Multiplexed Functions of 24-pin Products (1/2)

Pin No.	I/O		Analog		HMI		Timer	Communications Interface	
24HWQFN	Digital port	Power supply, system, clock, debug	A/D converter	Comparator	Interrupt function	Capacitive touch sensing unit (CTSU)	Timer array unit	Serial array unit	Serial interface IICA
1	P125	$\overline{\text{RESET}}$	—	(VCOUT0) (VCOUT1)	INTP1 (INTP0)	—	—	(SI11)	—
2	P137	—	—	—	INTP0	—	TI00	—	—
3	P122	X2/XT2 EXCLK/EXCLKS	—	—	(INTP2)	—	TI05/TO05	—	—
4	P121	X1/XT1	—	—	(INTP3) (INTP4) (INTP5)	—	TI07/TO07	—	—
5	—	V_{SS}	—	—	—	—	—	—	—
6	—	V_{DD}	—	—	—	—	—	—	—
7	P60	—	—	—	(INTP3) (INTP6)	—	—	(SI11)	SCLA0
8	P61	—	—	—	(INTP7)	—	—	(SO11) (SDA11)	SDAA0
9	P10	(PCLBUZ0)	—	—	INTP8	—	(TI03/TO03)	(SCK11/SCL11) (RxD1)	—
10	P11	(PCLBUZ0) (RTC1HZ)	—	—	INTP9	—	(TI03/TO03)	(SCK11/SCL11) (TxD1)	—
11	P00	TOOLTxD (RTC1HZ)	—	—	INTP6	—	(TI02/TO02)	SO00/TxD0 (SCK11/SCL11) (SI11/SDA11)	(SCLA0)
12	P01	TOOLRxD	ANI0	—	INTP5	TS00	TI02/TO02 (TI01/TO01)	SI00/RxD0/SDA00 (SI11/SDA11) (SO11)	(SDAA0)
13	P02	PCLBUZ0	ANI1	VCOUT0	INTP7	TSCAP	TI01/TO01 (TI02/TO02)	SCK00/SCL00 (SO11)	—
14	P03	—	ANI2	IVCMP0	INTP4	TS03	TO00 (TI00) (TI05/TO05)	RxD1 (SO00/TxD0)	—
15	P04	—	ANI3	IVREF0	INTP3	TS04	TI06/TO06 (TI01/TO01)	TxD1 (SI00/RxD0/SDA00) (SO00/TxD0)	—
16	P05	—	ANI4	IVCMP1	(INTP6)	TS05	(TI02/TO02) (TI07/TO07)	SO11 (SCK00/SCL00) (SI00/RxD0/SDA00)	—
17	P06	(PCLBUZ0)	ANI5	IVREF1	(INTP7)	TS06	(TI03/TO03)	SI11/SDA11 (SCK00/SCL00)	(SCLA0)
18	P07	—	ANI6	VCOUT1	(INTP5)	TS07	TI04/TO04 (TO03)	SCK11/SCL11	(SDAA0)
19	P23	—	ANI7	—	(INTP6)	TS08	(TI04/TO04)	(SCL11)	—
20	P22	—	ANI8	—	(INTP5)	TS09	(TI06/TO06)	(SDA11)	—
21	P21	—	ANI9	—	(INTP7)	TS10	(TO00)	SO20/TxD2 (RxD1)	—
22	P20	—	ANI10	—	(INTP1)	TS11	(TI00) (TO00) (TI03/TO03)	SI20/RxD2/SDA20 (SCK11/SCL11) (TxD1)	—
23	P41	RTC1HZ	—	(VCOUT0) (VCOUT1)	(INTP3) (INTP4)	TS13	TI03/TO03 (TI02/TO02)	SCK20/SCL20 (SO11) (SDA11)	—

Table 1-5. Multiplexed Functions of 24-pin Products (2/2)

Pin No.	I/O		Analog		HMI		Timer	Communications Interface	
24HWQFN	Digital port	Power supply, system, clock, debug	A/D converter	Comparator	Interrupt function	Capacitive touch sensing unit (CTSU)	Timer array unit	Serial array unit	Serial interface IICA
24	P40	TOOL0 (PCLBUZ0)	—	—	INTP2	—	(TI01/TO01)	—	—

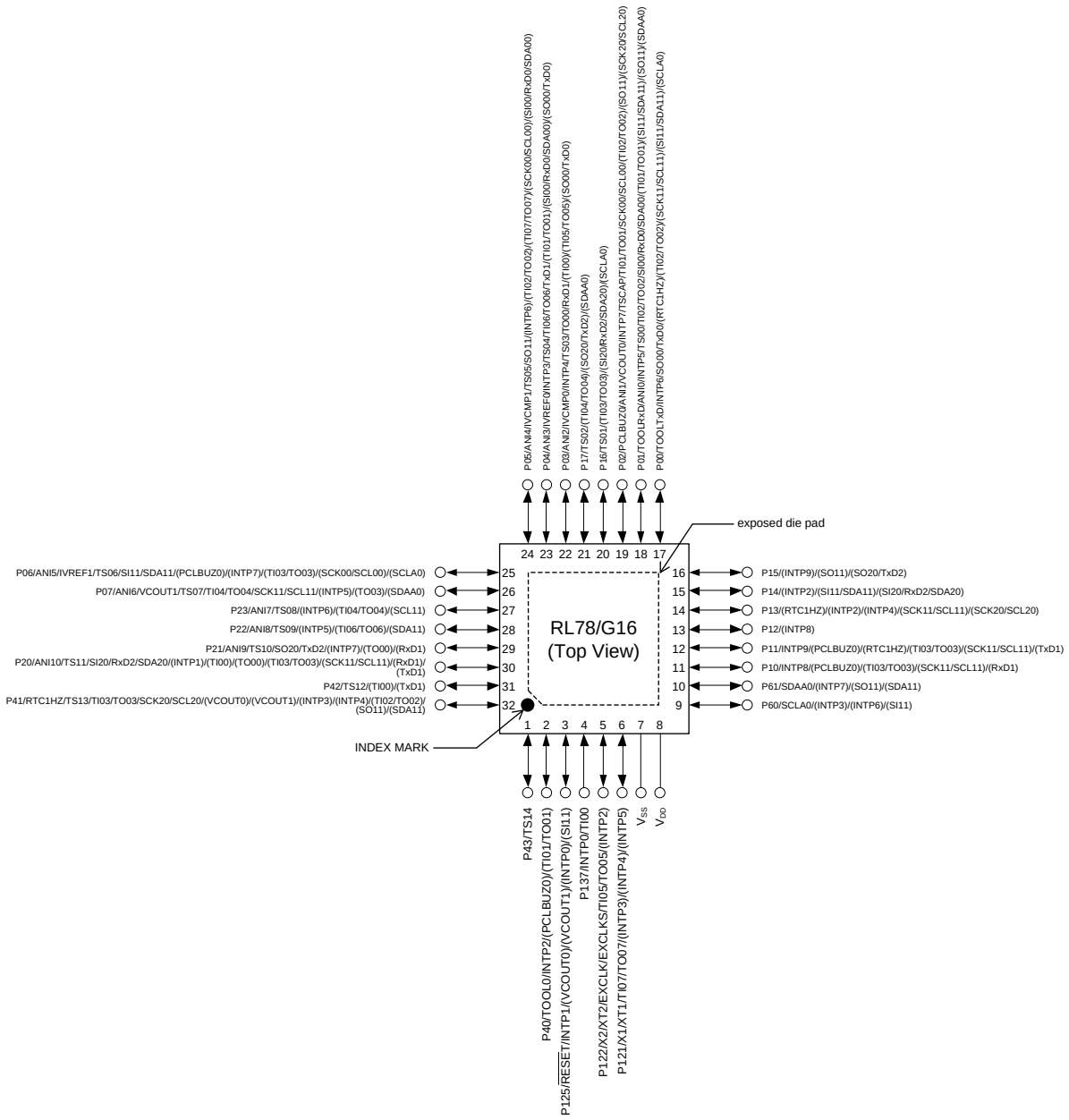
Remark 1. For pin identification, see **1.4 Pin Identification**.

Remark 2. Functions in parentheses can be assigned via settings in the peripheral I/O redirection register (PIOR). Refer to **Figure 4-6 Format of Peripheral I/O Redirection Registers 0 to 6 (PIOR0 to PIOR6)**.

Remark 3. For the product in a QFN package, solder the exposed die pad onto a plated area of the PCB that has no electrical connections.

1.3.5 32-pin products

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



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

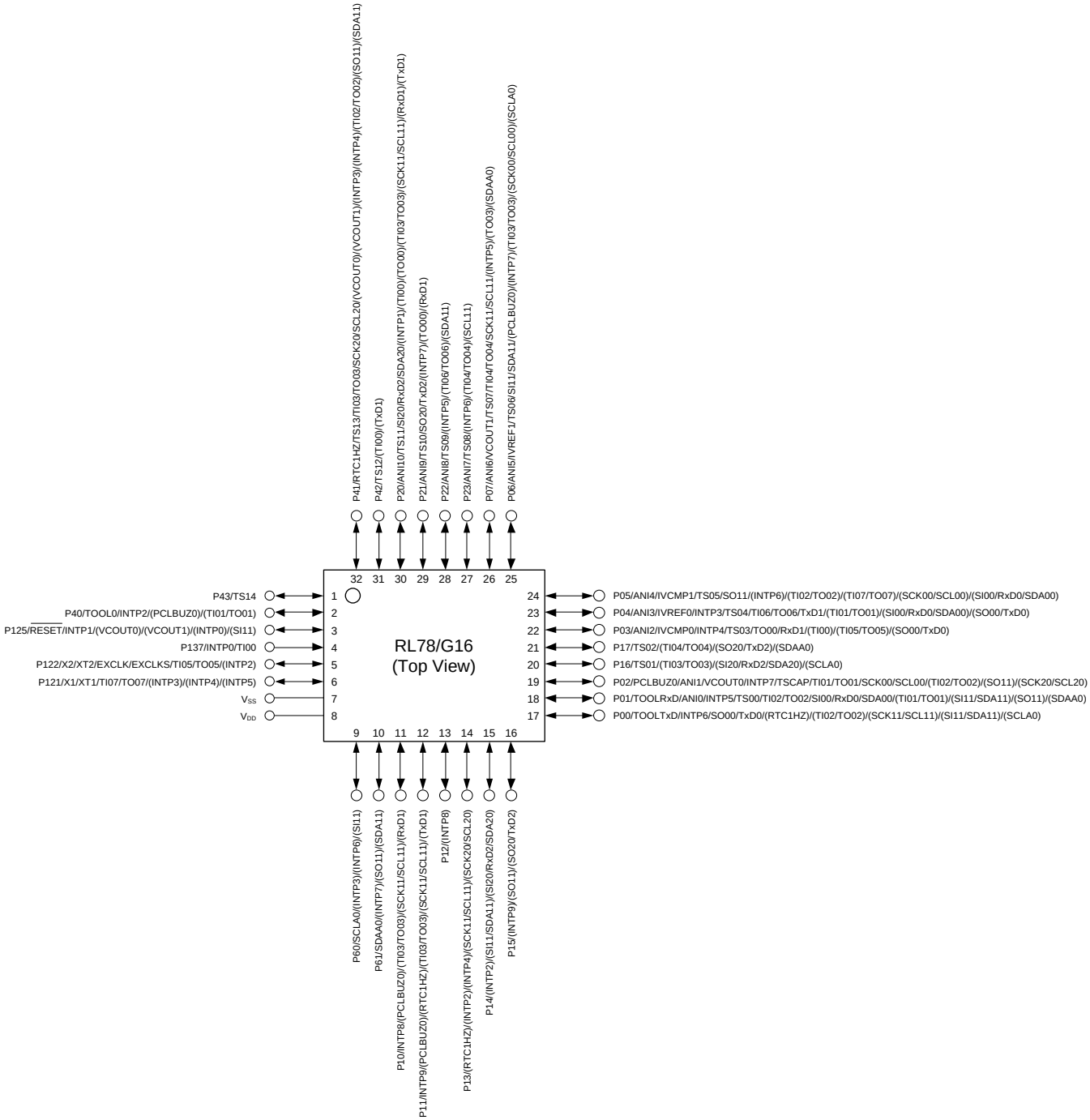


Table 1-6. Multiplexed Functions of 32-pin Products (1/2)

Pin No.	I/O		Analog		HMI		Timer	Communications Interface	
32HWQFN 32LQFP	Digital port	Power supply, system, clock, debug	A/D converter	Comparator	Interrupt function	Capacitive touch sensing unit (CTSU)	Timer array unit	Serial array unit	Serial interface IICA
1	P43	—	—	—	—	TS14	—	—	—
2	P40	TOOL0 (PCLBUZ0)	—	—	INTP2	—	(TI01/TO01)	—	—
3	P125	$\overline{\text{RESET}}$	—	(VCOUT0) (VCOUT1)	INTP1 (INTP0)	—	—	(SI11)	—
4	P137	—	—	—	INTP0	—	TI00	—	—
5	P122	X2/XT2 EXCLK/EXCLKS	—	—	(INTP2)	—	TI05/TO05	—	—
6	P121	X1/XT1	—	—	(INTP3) (INTP4) (INTP5)	—	TI07/TO07	—	—
7	—	VSS	—	—	—	—	—	—	—
8	—	VDD	—	—	—	—	—	—	—
9	P60	—	—	—	(INTP3) (INTP6)	—	—	(SI11)	SCLA0
10	P61	—	—	—	(INTP7)	—	—	(SO11) (SDA11)	SDAA0
11	P10	(PCLBUZ0)	—	—	INTP8	—	(TI03/TO03)	(SCK11/SCL11) (RxD1)	—
12	P11	(PCLBUZ0) (RTC1HZ)	—	—	INTP9	—	(TI03/TO03)	(SCK11/SCL11) (TxD1)	—
13	P12	—	—	—	(INTP8)	—	—	—	—
14	P13	(RTC1HZ)	—	—	(INTP2) (INTP4)	—	—	(SCK11/SCL11) (SCK20/SCL20)	—
15	P14	—	—	—	(INTP2)	—	—	(SI11/SDA11) (SI20/RxD2/SDA20)	—
16	P15	—	—	—	(INTP9)	—	—	(SO11) (SO20/TxD2)	—
17	P00	TOOLTxD (RTC1HZ)	—	—	INTP6	—	(TI02/TO02)	SO00/TxD0 (SCK11/SCL11) (SI11/SDA11)	(SCLA0)
18	P01	TOOLRxD	ANI0	—	INTP5	TS00	TI02/TO02 (TI01/TO01)	SI00/RxD0/SDA00 (SI11/SDA11) (SO11)	(SDAA0)
19	P02	PCLBUZ0	ANI1	VCOUT0	INTP7	TSCAP	TI01/TO01 (TI02/TO02)	SCK00/SCL00 (SO11) (SCK20/SCL20)	—
20	P16	—	—	—	—	TS01	(TI03/TO03)	(SI20/RxD2/SDA20)	(SCLA0)
21	P17	—	—	—	—	TS02	(TI04/TO04)	(SO20/TxD2)	(SDAA0)
22	P03	—	ANI2	IVCMP0	INTP4	TS03	TO00 (TI00) (TI05/TO05)	RxD1 (SO00/TxD0)	—
23	P04	—	ANI3	IVREF0	INTP3	TS04	TI06/TO06 (TI01/TO01)	TxD1 (SI00/RxD0/SDA00) (SO00/TxD0)	—
24	P05	—	ANI4	IVCMP1	(INTP6)	TS05	(TI02/TO02) (TI07/TO07)	SO11 (SCK00/SCL00) (SI00/RxD0/SDA00)	—

Table 1-6. Multiplexed Functions of 32-pin Products (2/2)

Pin No.	I/O		Analog		HMI		Timer	Communications Interface	
32HWQFN 32LQFP	Digital port	Power supply, system, clock, debug	A/D converter	Comparator	Interrupt function	Capacitive touch sensing unit (CTSU)	Timer array unit	Serial array unit	Serial interface IICA
25	P06	(PCLBUZ0)	ANI5	IVREF1	(INTP7)	TS06	(TI03/TO03)	SI11/SDA11 (SCK00/SCL00)	(SCLA0)
26	P07	—	ANI6	VCOUT1	(INTP5)	TS07	TI04/TO04 (TO03)	SCK11/SCL11	(SDAA0)
27	P23	—	ANI7	—	(INTP6)	TS08	(TI04/TO04)	(SCL11)	—
28	P22	—	ANI8	—	(INTP5)	TS09	(TI06/TO06)	(SDA11)	—
29	P21	—	ANI9	—	(INTP7)	TS10	(TO00)	SO20/TxD2 (RxD1)	—
30	P20	—	ANI10	—	(INTP1)	TS11	(TI00) (TO00) (TI03/TO03)	SI20/RxD2/SDA20 (SCK11/SCL11) (RxD1) (TxD1)	—
31	P42	—	—	—	—	TS12	(TI00)	(TxD1)	—
32	P41	RTC1HZ	—	(VCOUT0) (VCOUT1)	(INTP3) (INTP4)	TS13	TI03/TO03 (TI02/TO02)	SCK20/SCL20 (SO11) (SDA11)	—

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

Remark 2. Functions in parentheses can be assigned via settings in the peripheral I/O redirection register (PIOR). Refer to **Figure 4-6 Format of Peripheral I/O Redirection Registers 0 to 6 (PIOR0 to PIOR6)**.

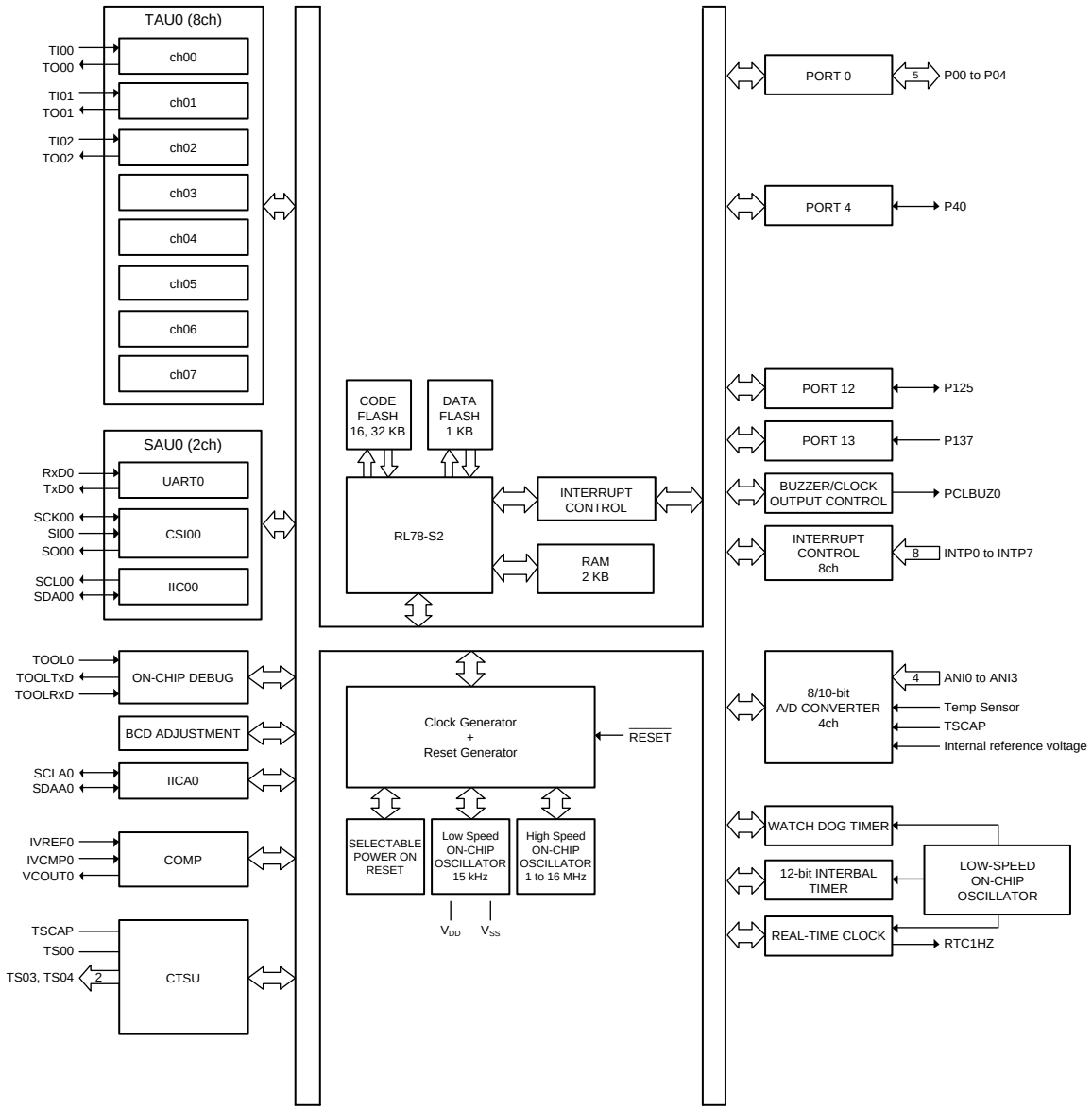
Remark 3. For the product in a QFN package, solder the exposed die pad onto a plated area of the PCB that has no electrical connections.

1.4 Pin Identification

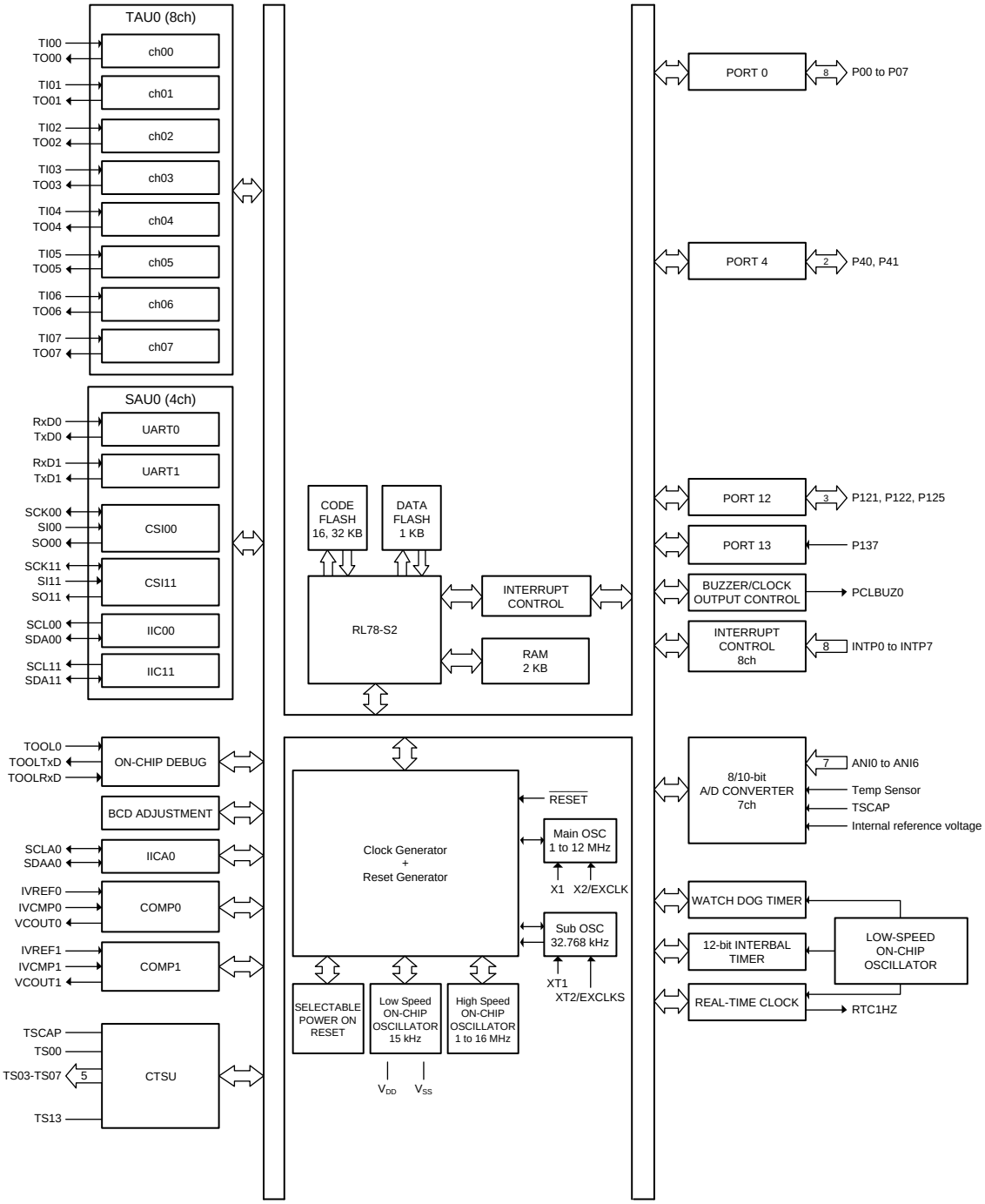
ANI0 to ANI10	: Analog Input
INTP0 to INTP9	: Interrupt Request From Peripherals
P00 to P07	: Port 0
P10 to P17	: Port 1
P20 to P23	: Port 2
P40 to P43	: Port 4
P60, P61	: Port 6
P121, P122, P125	: Port 12
P137	: Port 13
PCLBUZ0	: Programmable Clock Output/Buzzer Output
EXCLK	: External Clock Input (Main System Clock)
EXCLKS	: External Clock Input (Subsystem Clock)
X1, X2	: Crystal Oscillator (Main System Clock)
XT1, XT2	: Crystal Oscillator (Subsystem Clock)
IVCMP0, IVCMP1	: Comparator Input
VCOUT0, VCOUT1	: Comparator Output
IVREF0, IVREF1	: Comparator Reference Input
$\overline{\text{RESET}}$	: Reset
RxD0, RxD1, RxD2	: Receive Data
RTC1HZ	: Real-time Clock Correction Clock (1 Hz) Output
SCK00, SCK11, SCK20	: Serial Clock Input/Output
SCL00, SCL11, SCL20, SCLA0	: Serial Clock Output
SDA00, SDA11, SDA20, SDAA0	: Serial Data Input/Output
SI00, SI11, SI20	: Serial Data Input
SO00, SO11, SO20	: Serial Data Output
TI00 to TI07	: Timer Input
TO00 to TO07	: Timer Output
TOOL0	: Data Input/Output for Tool
TOOLRxD, TOOLTxD	: Data Input/Output for External Device
TxD0, TxD1, TxD2	: Transmit Data
TS00 to TS14	: Electrostatic Capacitance Measurement Pin (Touch Pin)
TSCAP	: LPF (low-pass filter) Connection for CTSU
V_{DD}	: Power Supply
V_{SS}	: Ground

1.5 Block Diagram

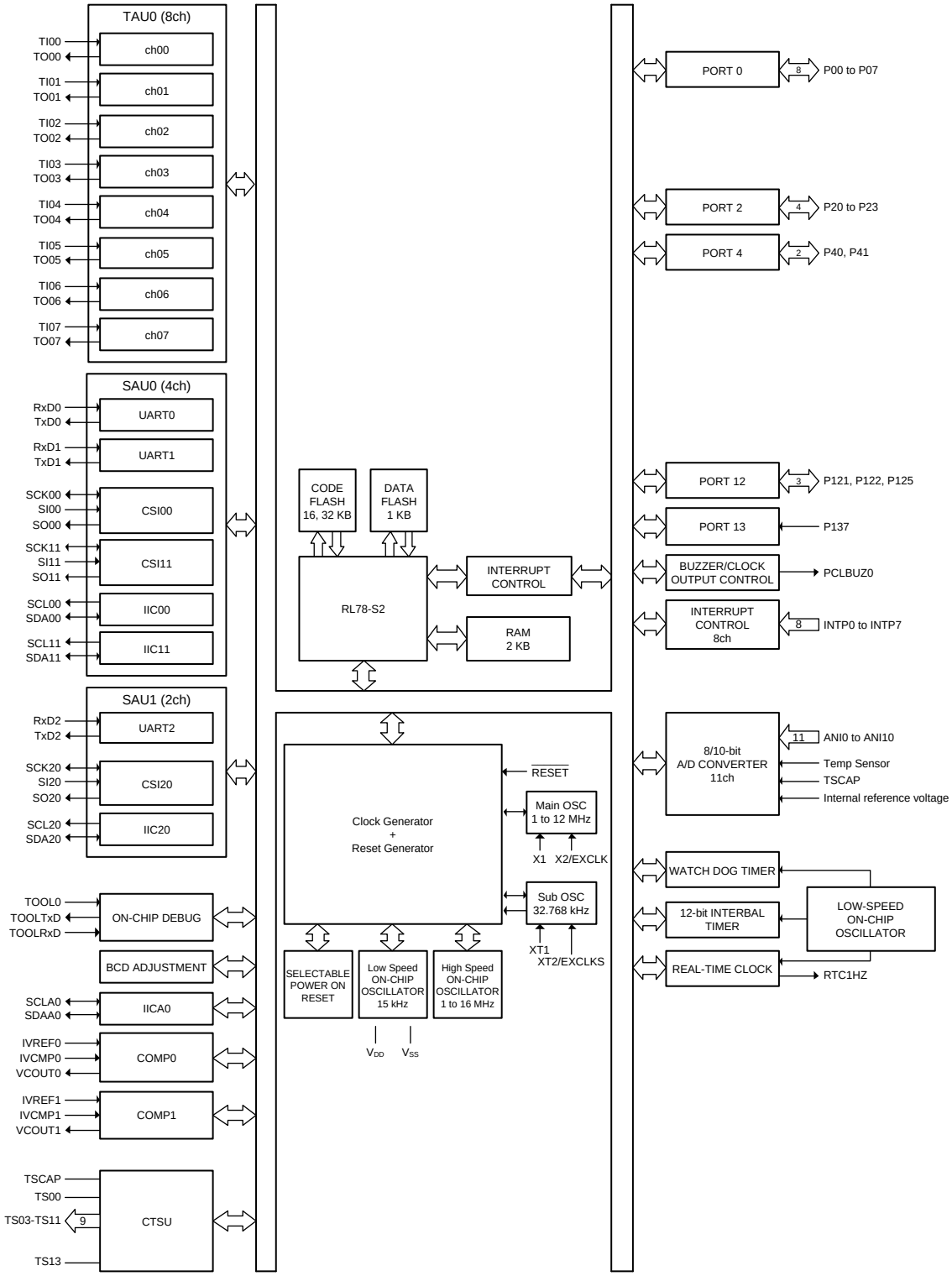
1.5.1 10-pin products



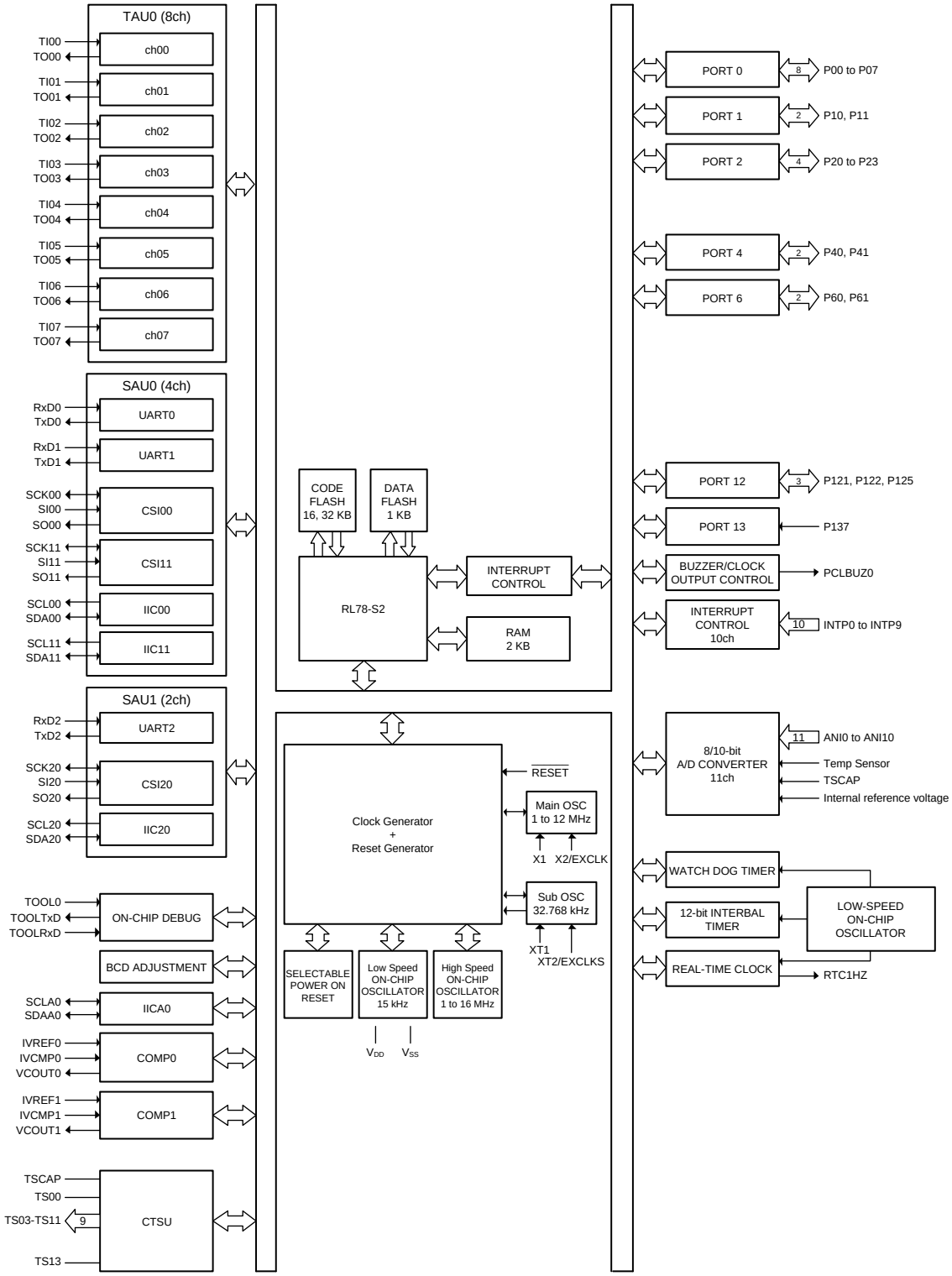
1.5.2 16-pin products



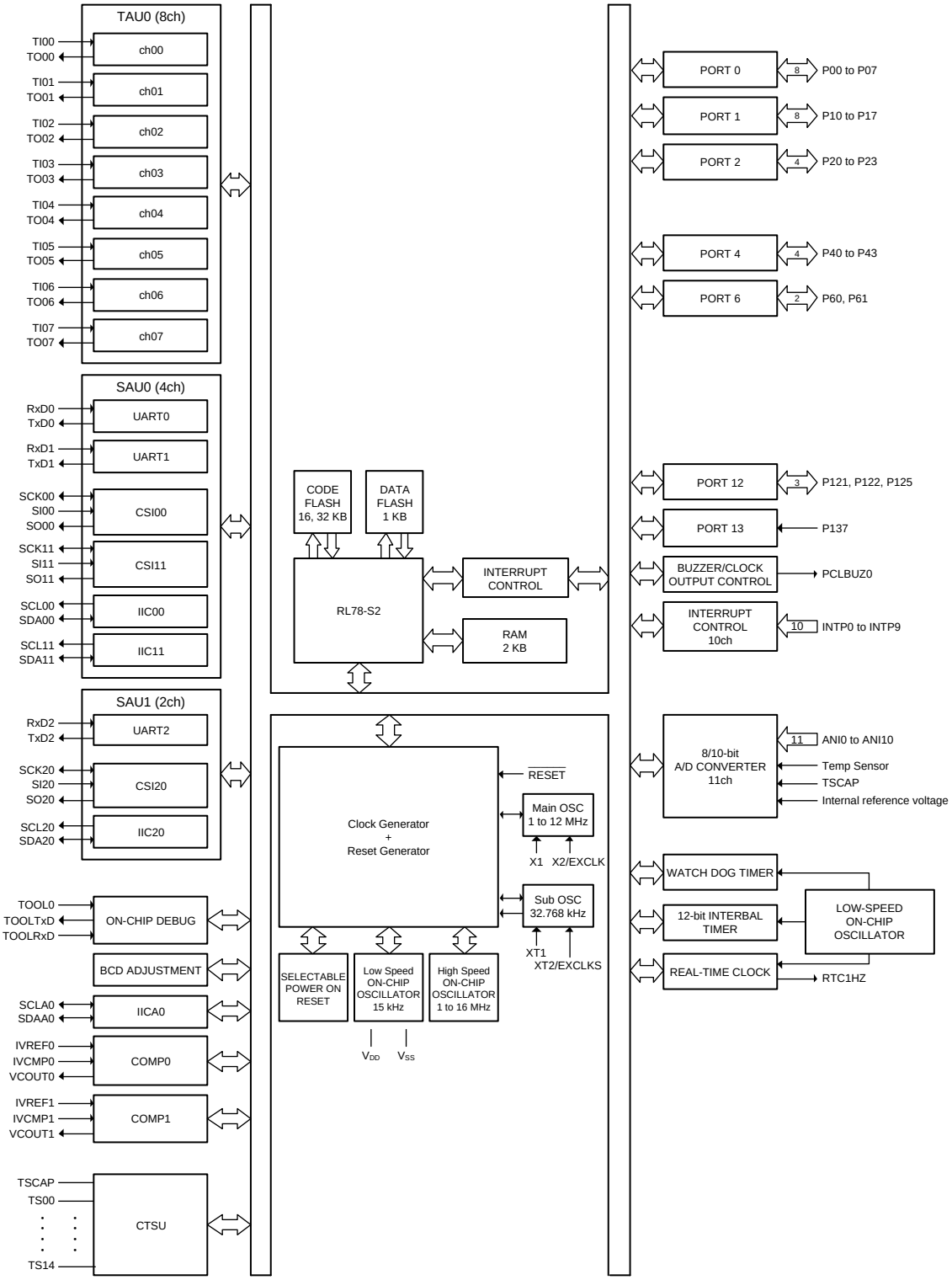
1.5.3 20-pin products



1.5.4 24-pin products



1.5.5 32-pin products



1.6 Outline of Functions

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

Item		10-pin		16-pin		20-pin		24-pin		32-pin	
		R5F1211A	R5F1211C	R5F1214A	R5F1214C	R5F1216A	R5F1216C	R5F1217A	R5F1217C	R5F121BA	R5F121BC
Code flash memory		16 KB	32 KB	16 KB	32 KB	16 KB	32 KB	16 KB	32 KB	16 KB	32 KB
Data flash memory		1 KB									
RAM		2 KB									
Main system clock	High-speed system clock	—		X1, X2 (crystal/ceramic) oscillation: 1 to 12 MHz: V_{DD} = 2.4 to 5.5 V External main system clock input (EXCLK): 1 to 16 MHz: V_{DD} = 2.4 to 5.5 V							
	High-speed on-chip oscillator	1 to 16 MHz (V_{DD} = 2.4 to 5.5 V)									
Subsystem clock		—		XT1 (crystal) oscillation, external subsystem clock input (EXCLKS) 32.768 kHz (TYP.): V_{DD} = 2.4 to 5.5 V							
Low-speed on-chip oscillator clock		15 kHz (TYP.)									
General-purpose registers		(8-bit register × 8) × 4 banks									
Minimum instruction execution time		0.0625 μs (16 MHz operation)									
Instruction set		• Data transfer (8/16 bits) • Adder and subtractor/logical operation (8/16 bits) • Multiplication (8 bits × 8 bits) • Rotate, barrel shift, and bit manipulation (Set, reset, test, and Boolean operation), etc.									
I/O port	Total	8		14		18		22		30	
	CMOS I/O	7		13		17		19		27	
	CMOS input	1									
	N-ch open drain I/O (withstand voltage of 6 V)	—						2			
Timer	16-bit timer	8 channels									
	Watchdog timer	1 channel									
	12-bit interval timer	1 channel									
	Real-time clock 2	—		1 channel							
	Timer output	3 channels (PWM outputs: 2)Note 1		8 channels (PWM outputs: 7)Note 1		8 channels (PWM outputs: 7)Note 1		8 channels (PWM outputs: 7)Note 1		8 channels (PWM outputs: 7)Note 1	
	RTC output	—		1 Hz (subsystem clock: f_{SUB} = 32.768 kHz)							
Clock output/buzzer output		1									
		Up to 10 MHz (peripheral hardware clock: f_{MAIN} = 10 MHz operation)									
Comparator		1 channel		2 channels		2 channels		2 channels		2 channels	
8/10-bit resolution A/D converter		4 channels		7 channels		11 channels		11 channels		11 channels	
Serial interface		Simplified SPI (CSI): 1 channel/ simplified I^2C: 1 channel/ UART: 1 channel		Simplified SPI (CSINote 2): 2 channels/ simplified I^2C: 2 channels/ UART: 2 channels		Simplified SPI (CSI): 3 channels/simplified I^2C: 3 channels/UART: 3 channels					
	I^2C bus	1 channel									
CTSU		3		7		11		11		15	
Number of Vectored interrupt sources	Internal	23		26		30		30		30	
	External	8		8		8		10		10	
Reset		• Reset by $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by selectable power-on-reset • Internal reset by illegal instruction executionNote 3 • Internal reset by data retention lower limit voltage • Internal reset by illegal-memory access • Internal reset by RAM parity error									
Selectable power-on-reset circuit		• Detection voltage Rising edge (V_{SPOR}): 2.25 V/2.68 V/3.02 V/4.45 V (MAX.) Falling edge (V_{SPDR}): 2.20 V/2.62 V/2.96 V/4.37 V (MAX.)									
On-chip debug function		Provided									
Power supply voltage		V_{DD} = 2.4 to 5.5 V									
Operating ambient temperature		T_A = −40 to +85°C (A: Consumer applications), T_A = −40 to +105°C (G: Industrial applications), T_A = −40 to +125°C (M: Industrial applications)									

Note 1. The number of outputs varies, depending on the setting of channels in use and the number of the master (see **6.9.3 Operation as multiple PWM output function**).

Note 2. 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 3. The illegal instruction is generated when instruction code FFH is executed. Reset by the illegal instruction execution not issued by emulation with the on-chip debug emulator.

CHAPTER 2 PIN FUNCTIONS

2.1 Port Function

The input or output, buffer, and pull-up resistor settings on each port are also valid for the alternate functions.

2.1.1 10-pin products

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-1-2	I/O	Input port	TOOLTxD/INTP6/SO00/ TxD0/(TI02/TO02)/(SCLA0)	Port 0. 5-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Output of P00, P01, P03, and P04 can be set to N-ch open-drain output (V_{DD} tolerance). P01 to P04 can be set to analog input[Note 1]. P01, P03, and P04 can be set to touch pin output[Note 1].
P01	7-33-5		Analog input	TOOLRxD/ANI0/INTP5/TS00/ TI02/TO02/SI00/RxD0/ SDA00/(TI01/TO01)/(SDAA0)	
P02	7-31-5			PCLBUZ0/ANI1/VCOUT0/ INTP7/TSCAP/TI01/TO01/ SCK00/SCL00	
P03	7-33-7			ANI2/IVCMP0/INTP4/TS03/ TO00/SCLA0/(TI00)	
P04				ANI3/IVREF0/INTP3/TS04/ SDAA0/(TI01/TO01)	
P40	7-1-1	I/O	Input port	TOOL0/INTP2/(PCLBUZ0)/ (TI01/TO01)	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 port.
P125	3-2-1	I/O	Input port	$\overline{\text{RESET}}$/INTP1/(VCOUT0)/ (INTP0)	Port 12. 1-bit I/O port. Use of an on-chip pull-up resistor can be specified by a software setting. 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.
P137	2-1-2	Input	Input port	INTP0/TI00	Port 13. 1-bit input only port.

Note 1. Setting digital, analog, or touch to each pin can be done in the port mode registers 0, 4, 12 (PM0, PM4, PM12), port mode control register 0 (PMC0), and touch pin function select register 0 (TSSEL0) (can be set in 1-bit units).

Remark Functions in parentheses can be assigned via settings in the peripheral I/O redirection registers 0 to 6 (PIOR0 to PIOR6). Refer to **Figure 4-6 Format of Peripheral I/O Redirection Registers 0 to 6 (PIOR0 to PIOR6)**.

2.1.2 16-pin products

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-1-2	I/O	Input port	TOOLTxD/INTP6/SO00/ TxD0/(RTC1HZ)/(TI02/TO02)/ (SCK11/SCL11)/(SCLA0)	Port 0. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Output of P00, P01, and P03 to P07 can be set to N-ch open-drain output (V_{DD} tolerance). P01 to P07 can be set to analog input[Note 1]. P01 and P03 to P07 can be set to touch pin output[Note 1].
P01	7-33-5		Analog input	TOOLRxD/ANI0/INTP5/TS00/ TI02/TO02/SI00/RxD0/ SDA00/(TI01/TO01)/ (SI11/SDA11)/(SDAA0)	
P02	7-31-5			PCLBUZ0/ANI1/VCOUT0/ INTP7/TSCAP/TI01/TO01/ SCK00/SCL00/(TI02/TO02)/ (SO11)	
P03	7-33-7			ANI2/IVCMP0/INTP4/TS03/ TO00/RxD1/(TI00)/ (TI05/TO05)/(SO00/TxD0)	
P04				ANI3/IVREF0/INTP3/TS04/ TI06/TO06/TxD1/ (TI01/TO01)/ (SI00/RxD0/SDA00)/ (SO00/TxD0)	
P05				ANI4/IVCMP1/TS05/SO11/ (INTP6)/(TI02/TO02)/ (TI07/TO07)/(SCK00/SCL00)/ (SI00/RxD0/SDA00)	
P06				ANI5/IVREF1/TS06/SI11/ SDA11/SCLA0/(PCLBUZ0)/ (INTP7)/(TI03/TO03)/ (SCK00/SCL00)	
P07	7-33-5			ANI6/VCOUT1/TS07/TI04/ TO04/SCK11/SCL11/SDAA0/ (INTP5)/(TO03)	
P40	7-1-1	I/O	Input port	TOOL0/INTP2/(PCLBUZ0)/ (TI01/TO01)	Port 4. 2-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Output of P41 can be set to N-ch open-drain output (V_{DD} tolerance). P41 can be set to touch pin output[Note 1].
P41	7-33-8			RTC1HZ/TS13/TI03/TO03/ (INTP3)/(INTP4)/(TI02/TO02)/ (VCOUT0)/(VCOUT1)	

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P121	7-2-3	I/O	Input port	X1/XT1/TI07/TO07/(INTP3)/(INTP4)/(INTP5)	Port 12. 3-bit I/O port. Use of an on-chip pull-up resistor can be specified by a software setting at P121, P122, and P125. 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.
P122				X2/XT2/EXCLK/EXCLKS/TI05/TO05/(INTP2)	
P125	3-2-1			$\overline{\text{RESET}}$/INTP1/(VCOUT0)/(VCOUT1)/(INTP0)	
P137	2-1-2	I/O	Input port	INTP0/TI00	Port 13. 1-bit input only port.

Note 1. Setting digital, analog, or touch to each pin can be done in the port mode registers 0, 4, 12 (PM0, PM4, PM12), port mode control register 0 (PMC0), and touch pin function select registers 0, 1 (TSSEL0, TSSEL1) (can be set in 1-bit units).

Remark Functions in parentheses can be assigned via settings in the peripheral I/O redirection registers 0 to 6 (PIOR0 to PIOR6). Refer to **Figure 4-6 Format of Peripheral I/O Redirection Registers 0 to 6 (PIOR0 to PIOR6)**.

2.1.3 20-pin products

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-1-2	I/O	Input port	TOOLTxD/INTP6/SO00/ TxD0/(RTC1HZ)/(TI02/TO02)/ (SCK11/SCL11)/(SCLA0)	Port 0. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Output of P00, P01, and P03 to P07 can be set to N-ch open-drain output (V_{DD} tolerance). P01 to P07 can be set to analog input[Note 1]. P01 and P03 to P07 can be set to touch pin output[Note 1].
P01	7-33-5		Analog input	TOOLRxD/ANI0/INTP5/TS00/ TI02/TO02/SI00/RxD0/ SDA00/(TI01/TO01)/ (SI11/SDA11)/(SDAA0)	
P02	7-31-5			PCLBUZ0/ANI1/VCOUT0/ INTP7/TSCAP/TI01/TO01/ SCK00/SCL00/(TI02/TO02)/ (SO11)	
P03	7-33-7			ANI2/IVCMP0/INTP4/TS03/ TO00/RxD1/(TI00)/ (TI05/TO05)/(SO00/TxD0)	
P04				ANI3/IVREF0/INTP3/TS04/ TI06/TO06/TxD1/ (TI01/TO01)/ (SI00/RxD0/SDA00)/ (SO00/TxD0)	
P05				ANI4/IVCMP1/TS05/SO11/ (INTP6)/(TI02/TO02)/ (TI07/TO07)/(SCK00/SCL00)/ (SI00/RxD0/SDA00)	
P06				ANI5/IVREF1/TS06/SI11/ SDA11/SCLA0/(PCLBUZ0)/ (INTP7)/(TI03/TO03)/ (SCK00/SCL00)	
P07	7-33-5			ANI6/VCOUT1/TS07/TI04/ TO04/SCK11/SCL11/SDAA0/ (INTP5)/(TO03)	
P20	7-33-5	I/O	Analog input	ANI10/TS11/SI20/RxD2/ SDA20/(INTP1)/(TI00)/ (TO00)/(TI03/TO03)/ (SCK11/SCL11)/(TxD1)	Port 2. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Output of P20 and P22 can be set to N-ch open-drain output (V_{DD} tolerance). P20 to P23 can be set to analog input[Note 1]. P20 to P23 can be set to touch pin output[Note 1].
P21	7-33-6			ANI9/TS10/SO20/TxD2/ (INTP7)/(TO00)/(RxD1)	
P22	7-33-5			ANI8/TS09/(INTP5)/ (TI06/TO06)/(SDA11)	
P23	7-33-6			ANI7/TS08/(INTP6)/ (TI04/TO04)/(SCL11)	

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P40	7-1-1	I/O	Input port	TOOL0/INTP2/(PCLBUZ0)/ (TI01/TO01)	Port 4. 2-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Output of P41 can be set to N-ch open-drain output (V_{DD} tolerance). P41 can be set to touch pin output[Note 1].
P41	7-33-8			RTC1HZ/TS13/TI03/TO03/ SCK20/SCL20/(VCOUT0)/ (VCOUT1)/(INTP3)/(INTP4)/ (TI02/TO02)/(SO11)/(SDA11)	
P121	7-2-3	I/O	Input port	X1/XT1/TI07/TO07/(INTP3)/ (INTP4)/(INTP5)	Port 12. 3-bit I/O port. Use of an on-chip pull-up resistor can be specified by a software setting at P121, P122, and P125. 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.
P122				X2/XT2/EXCLK/EXCLKS/ TI05/TO05/(INTP2)	
P125	3-2-1			$\overline{\text{RESET}}$/INTP1/(VCOUT0)/ (VCOUT1)/(INTP0)/(SI11)	
P137	2-1-2	Input	Input port	INTP0/TI00	Port 13. 1-bit input only port.

Note 1. Setting digital, analog, or touch to each pin can be done in the port mode registers 0, 2, 4, 12 (PM0, PM2, PM4, PM12), port mode control registers 0, 2 (PMC0, PMC2), and touch pin function select registers 0, 1 (TSSEL0, TSSEL1) (can be set in 1-bit units).

Remark Functions in parentheses can be assigned via settings in the peripheral I/O redirection registers 0 to 6 (PIOR0 to PIOR6). Refer to **Figure 4-6 Format of Peripheral I/O Redirection Registers 0 to 6 (PIOR0 to PIOR6)**.

2.1.4 24-pin products

(1/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-1-2	I/O	Input port	TOOLTxD/INTP6/SO00/ TxD0/(RTC1HZ)/(TI02/TO02)/ (SCK11/SCL11)/ (SI11/SDA11)/(SCLA0)	Port 0. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Output of P00, P01, and P03 to P07 can be set to N-ch open-drain output (V_{DD} tolerance). P01 to P07 can be set to analog input[Note 1]. P01 and P03 to P07 can be set to touch pin output[Note 1].
P01	7-33-5			TOOLRxD/ANI0/INTP5/TS00/ TI02/TO02/SI00/RxD0/ SDA00/(TI01/TO01)/ (SI11/SDA11)/(SO11)/ (SDAA0)	
P02	7-31-5			PCLBUZ0/ANI1/VCOUT0/ INTP7/TSCAP/TI01/TO01/ SCK00/SCL00/(TI02/TO02)/ (SO11)	
P03	7-33-7			ANI2/IVCMP0/INTP4/TS03/ TO00/RxD1/(TI00)/ (TI05/TO05)/(SO00/TxD0)	
P04				ANI3/IVREF0/INTP3/TS04/ TI06/TO06/TxD1/ (TI01/TO01)/ (SI00/RxD0/SDA00)/ (SO00/TxD0)	
P05				ANI4/IVCMP1/TS05/SO11/ (INTP6)/(TI02/TO02)/ (TI07/TO07)/(SCK00/SCL00)/ (SI00/RxD0/SDA00)	
P06				ANI5/IVREF1/TS06/SI11/ SDA11/(PCLBUZ0)/(INTP7)/ (TI03/TO03)/(SCK00/SCL00)/ (SCLA0)	
P07	7-33-5			ANI6/VCOUT1/TS07/TI04/ TO04/SCK11/SCL11/(INTP5)/ (TO03)/(SDAA0)	
P10	7-1-1	I/O	Input port	INTP8/(PCLBUZ0)/ (TI03/TO03)/(SCK11/SCL11)/ (RxD1)	Port 1. 2-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P11				INTP9/(PCLBUZ0)/ (RTC1HZ)/(TI03/TO03)/ (SCK11/SCL11)/(TxD1)	

(2/2)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P20	7-33-5	I/O	Analog input	ANI10/TS11/SI20/RxD2/ SDA20/(INTP1)/(TI00)/ (TO00)/(TI03/TO03)/ (SCK11/SCL11)/(TxD1)	Port 2. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Output of P20 and P22 can be set to N-ch open-drain output (V_{DD} tolerance). P20 to P23 can be set to analog input[Note 1]. P20 to P23 can be set to touch pin output[Note 1].
P21	7-33-6			ANI9/TS10/SO20/TxD2/ (INTP7)/(TO00)/(RxD1)	
P22	7-33-5			ANI8/TS09/(INTP5)/ (TI06/TO06)/(SDA11)	
P23	7-33-6			ANI7/TS08/(INTP6)/ (TI04/TO04)/(SCL11)	
P40	7-1-1	I/O	Input port	TOOL0/INTP2/(PCLBUZ0)/ (TI01/TO01)	Port 4. 2-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Output of P41 can be set to N-ch open-drain output (V_{DD} tolerance). P41 can be set to touch pin output[Note 1].
P41	7-33-8			RTC1HZ/TS13/TI03/TO03/ SCK20/SCL20/(VCOUT0)/ (VCOUT1)/(INTP3)/(INTP4)/ (TI02/TO02)/(SO11)/(SDA11)	
P60	12-1-1	I/O	Input port	SCLA0/(INTP3)/(INTP6)/ (SI11)	Port 6. 2-bit I/O port. Input/output can be specified in 1-bit units. Output of P60 and P61 can be set to N-ch open-drain output (6-V tolerance).
P61				SDAA0/(INTP7)/(SO11)/ (SDA11)	
P121	7-2-3	I/O	Input port	X1/XT1/TI07/TO07/(INTP3)/ (INTP4)/(INTP5)	Port 12. 3-bit I/O port. Use of an on-chip pull-up resistor can be specified by a software setting at P121, P122, and P125. 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.
P122				X2/XT2/EXCLK/EXCLKS/ TI05/TO05/(INTP2)	
P125	3-2-1			$\overline{\text{RESET}}$/INTP1/(VCOUT0)/ (VCOUT1)/(INTP0)/(SI11)	
P137	2-1-2	Input	Input port	INTP0/TI00	Port 13. 1-bit input only port.

Note 1. Setting digital, analog, or touch to each pin can be done in the port mode registers 0, 1, 2, 4, 6, 12 (PM0, PM1, PM2, PM4, PM6, PM12), port mode control registers 0, 2 (PMC0, PMC2), and touch pin function select registers 0, 1 (TSSEL0, TSSEL1) (can be set in 1-bit units).

Remark Functions in parentheses can be assigned via settings in the peripheral I/O redirection registers 0 to 6 (PIOR0 to PIOR6). Refer to **Figure 4-6 Format of Peripheral I/O Redirection Registers 0 to 6 (PIOR0 to PIOR6)**.

2.1.5 32-pin products

(1/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P00	7-1-2	I/O	Input port	TOOLTxD/INTP6/SO00/ TxD0/(RTC1HZ)/(TI02/TO02)/ (SCK11/SCL11)/ (SI11/SDA11)/(SCLA0)	Port 0. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Output of P00, P01, and P03 to P07 can be set to N-ch open-drain output (V_{DD} tolerance). P01 to P07 can be set to analog input[Note 1]. P01 and P03 to P07 can be set to touch pin output[Note 1].
P01	7-33-5			TOOLRxD/ANI0/INTP5/TS00/ TI02/TO02/SI00/RxD0/ SDA00/(TI01/TO01)/ (SI11/SDA11)/(SO11)/ (SDAA0)	
P02	7-31-5			PCLBUZ0/ANI1/VCOUT0/ INTP7/TSCAP/TI01/TO01/ SCK00/SCL00/(TI02/TO02)/ (SO11)/(SCK20/SCL20)	
P03	733-7			ANI2/IVCMP0/INTP4/TS03/ TO00/RxD1/(TI00)/ (TI05/TO05)/(SO00/TxD0)	
P04				ANI3/IVREF0/INTP3/TS04/ TI06/TO06/TxD1/ (TI01/TO01)/ (SI00/RxD0/SDA00)/ (SO00/TxD0)	
P05				ANI4/IVCMP1/TS05/SO11/ (INTP6)/(TI02/TO02)/ (TI07/TO07)/(SCK00/SCL00)/ (SI00/RxD0/SDA00)	
P06				ANI5/IVREF1/TS06/SI11/ SDA11/(PCLBUZ0)/(INTP7)/ (TI03/TO03)/(SCK00/SCL00)/ (SCLA0)	
P07	7-33-5			ANI6/VCOUT1/TS07/TI04/ TO04/SCK11/SCL11/(INTP5)/ (TO03)/(SDAA0)	

(2/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P10	7-1-1	I/O	Input port	INTP8/(PCLBUZ0)/ (TI03/TO03)/(SCK11/SCL11)/ (RxD1)	Port 1. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Output of P14, P16, and P17 can be set to N-ch open-drain output (V_{DD} tolerance). P16 and P17 can be set to touch pin output[Note 1].
P11				INTP9/(PCLBUZ0)/ (RTC1HZ)/(TI03/TO03)/ (SCK11/SCL11)/(TxD1)	
P12				(INTP8)	
P13				(RTC1HZ)/(INTP2)/(INTP4)/ (SCK11/SCL11)/ (SCK20/SCL20)	
P14	7-1-2			(INTP2)/(SI11/SDA11)/ (SI20/RxD2/SDA20)	
P15	7-1-1			(INTP9)/(SO11)/(SO20/TxD2)	
P16	7-33-8			TS01/(TI03/TO03)/ (SI20/RxD2/SDA20)/(SCLA0)	
P17				TS02/(TI04/TO04)/ (SO20/TxD2)/(SDAA0)	
P20	7-33-5	I/O	Analog input	ANI10/TS11/SI20/RxD2/ SDA20/(INTP1)/(TI00)/ (TO00)/(TI03/TO03)/ (SCK11/SCL11)/(RxD1)/ (TxD1)	Port 2. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Output of P20 and P22 can be set to N-ch open-drain output (V_{DD} tolerance). P20 to P23 can be set to analog input[Note 1]. P20 to P23 can be set to touch pin output[Note 1].
P21	7-33-6			ANI9/TS10/SO20/TxD2/ (INTP7)/(TO00)/(RxD1)	
P22	7-33-5			ANI8/TS09/(INTP5)/ (TI06/TO06)/(SDA11)	
P23	7-33-6			ANI7/TS08/(INTP6)/ (TI04/TO04)/(SCL11)	
P40	7-1-1	I/O	Input port	TOOL0/INTP2/(PCLBUZ0)/ (TI01/TO01)	Port 4. 4-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Output of P41 can be set to N-ch open-drain output (V_{DD} tolerance). P41 to P43 can be set to touch pin output[Note 1].
P41	7-33-8			RTC1HZ/TS13/TI03/TO03/ SCK20/SCL20/(VCOUT0)/ (VCOUT1)/(INTP3)/(INTP4)/ (TI02/TO02)/(SO11)/(SDA11)	
P42	7-33-9			TS12/(TI00)/(TxD1)	
P43				TS14	
P60	12-1-1	I/O	Input port	SCLA0/(INTP3)/(INTP6)/ (SI11)	Port 6. 2-bit I/O port. Input/output can be specified in 1-bit units. Output of P60 and P61 can be set to N-ch open-drain output (6-V tolerance).
P61				SDAA0/(INTP7)/(SO11)/ (SDA11)	

(3/3)

Function Name	Pin Type	I/O	After Reset Release	Alternate Function	Function
P121	7-2-3	I/O	Input port	X1/XT1/TI07/TO07/(INTP3)/ (INTP4)/(INTP5)	Port 12. 3-bit I/O port. Use of an on-chip pull-up resistor can be specified by a software setting at P121, P122, and P125. 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.
P122				X2/XT2/EXCLK/EXCLKS/ TI05/TO05/(INTP2)	
P125	3-2-1			$\overline{\text{RESET}}$/INTP1/(VCOUT0)/ (VCOUT1)/(INTP0)/(SI11)	
P137	2-1-2	Input	Input port	INTP0/TI00	Port 13. 1-bit input only port.

Note 1. Setting digital, analog, or touch to each pin can be done in the port mode registers 0, 1, 2, 4, 6, 12 (PM0, PM1, PM2, PM4, PM6, PM12), port mode control registers 0, 2 (PMC0, PMC2), and touch pin function select registers 0, 1 (TSSEL0, TSSEL1) (can be set in 1-bit units).

Remark Functions in parentheses can be assigned via settings in the peripheral I/O redirection registers 0 to 6 (PIOR0 to PIOR6). Refer to **Figure 4-6 Format of Peripheral I/O Redirection Registers 0 to 6 (PIOR0 to PIOR6)**.

2.2 Functions other than port pins

2.2.1 Functions for each product

(1/2)

Function Name	32-pin products	24-pin products	20-pin products	16-pin products	10-pin products
ANI0	✓	✓	✓	✓	✓
ANI1	✓	✓	✓	✓	✓
ANI2	✓	✓	✓	✓	✓
ANI3	✓	✓	✓	✓	✓
ANI4	✓	✓	✓	✓	—
ANI5	✓	✓	✓	✓	—
ANI6	✓	✓	✓	✓	—
ANI7	✓	✓	✓	—	—
ANI8	✓	✓	✓	—	—
ANI9	✓	✓	✓	—	—
ANI10	✓	✓	✓	—	—
TS00	✓	✓	✓	✓	✓
TS01	✓	—	—	—	—
TS02	✓	—	—	—	—
TS03	✓	✓	✓	✓	✓
TS04	✓	✓	✓	✓	✓
TS05	✓	✓	✓	✓	—
TS06	✓	✓	✓	✓	—
TS07	✓	✓	✓	✓	—
TS08	✓	✓	✓	—	—
TS09	✓	✓	✓	—	—
TS10	✓	✓	✓	—	—
TS11	✓	✓	✓	—	—
TS12	✓	—	—	—	—
TS13	✓	✓	✓	✓	—
TS14	✓	—	—	—	—
TSCAP	✓	✓	✓	✓	✓
VCOUT0	✓	✓	✓	✓	✓
VCOUT1	✓	✓	✓	✓	—
IVCMP0	✓	✓	✓	✓	✓
IVCMP1	✓	✓	✓	✓	—
IVREF0	✓	✓	✓	✓	✓
IVREF1	✓	✓	✓	✓	—
INTP0	✓	✓	✓	✓	✓

Function Name	32-pin products	24-pin products	20-pin products	16-pin products	10-pin products
RxD0	✓	✓	✓	✓	✓
TxD0	✓	✓	✓	✓	✓
RxD1	✓	✓	✓	✓	—
TxD1	✓	✓	✓	✓	—
RxD2	✓	✓	✓	—	—
TxD2	✓	✓	✓	—	—
TOOLRxD	✓	✓	✓	✓	✓
TOOLTxD	✓	✓	✓	✓	✓
TOOL0	✓	✓	✓	✓	✓
SCL00	✓	✓	✓	✓	✓
SDA00	✓	✓	✓	✓	✓
SCL11	✓	✓	✓	✓	—
SDA11	✓	✓	✓	✓	—
SCL20	✓	✓	✓	—	—
SDA20	✓	✓	✓	—	—
SCK00	✓	✓	✓	✓	✓
SI00	✓	✓	✓	✓	✓
SO00	✓	✓	✓	✓	✓
SCK11	✓	✓	✓	✓	—
SI11	✓	✓	✓	✓	—
SO11	✓	✓	✓	✓	—
SCK20	✓	✓	✓	—	—
SI20	✓	✓	✓	—	—
SO20	✓	✓	✓	—	—
SCLA0	✓	✓	✓	✓	✓
SDAA0	✓	✓	✓	✓	✓
TI00	✓	✓	✓	✓	✓
TO00	✓	✓	✓	✓	✓
TI01	✓	✓	✓	✓	✓
TO01	✓	✓	✓	✓	✓
TI02	✓	✓	✓	✓	✓
TO02	✓	✓	✓	✓	✓
TI03	✓	✓	✓	✓	—
TO03	✓	✓	✓	✓	—

(2/2)

Function Name	32-pin products	24-pin products	20-pin products	16-pin products	10-pin products
INTP1	✓	✓	✓	✓	✓
INTP2	✓	✓	✓	✓	✓
INTP3	✓	✓	✓	✓	✓
INTP4	✓	✓	✓	✓	✓
INTP5	✓	✓	✓	✓	✓
INTP6	✓	✓	✓	✓	✓
INTP7	✓	✓	✓	✓	✓
INTP8	✓	✓	—	—	—
INTP9	✓	✓	—	—	—
PCLBUZ0	✓	✓	✓	✓	✓
$\overline{\text{RESET}}$	✓	✓	✓	✓	✓
X1	✓	✓	✓	✓	—
XT1	✓	✓	✓	✓	—
X2	✓	✓	✓	✓	—
XT2	✓	✓	✓	✓	—
EXCLK	✓	✓	✓	✓	—
EXCLKS	✓	✓	✓	✓	—

Function Name	32-pin products	24-pin products	20-pin products	16-pin products	10-pin products
TI04	✓	✓	✓	✓	—
TO04	✓	✓	✓	✓	—
TI05	✓	✓	✓	✓	—
TO05	✓	✓	✓	✓	—
TI06	✓	✓	✓	✓	—
TO06	✓	✓	✓	✓	—
TI07	✓	✓	✓	✓	—
TO07	✓	✓	✓	✓	—
RTC1HZ	✓	✓	✓	✓	—
V_{DD}	✓	✓	✓	✓	✓
V_{SS}	✓	✓	✓	✓	✓

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

Function Name	I/O	Function
<R> ANI0 to ANI10	Input	A/D converter analog input (see **Figure 11-22 Analog Input Pin Connection**).
TS00 to TS14	I/O	Capacitance measurement pin (touch pin)
<R> TSCAP	—	Pin for connecting a power supply capacitor for use in measuring electrostatic capacitance. Connect this pin to V_{SS} via a capacitor (10 nF). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.
VCOUT0, VCOUT1	Output	Comparator output
IVCMP0, IVCMP1	Input	Analog input for the comparator
IVREF0, IVREF1	Input	Reference voltage input for the comparator
INTP0 to INTP9	Input	External interrupt request input pin for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.
PCLBUZ0	Output	Clock output/buzzer output
<R> $\overline{\text{RESET}}$	Input	This is the active-low system reset input pin. PORTSELB = 1: When the external reset pin is not used, connect this pin directly to V_{DD}.
RxD0	Input	Serial data input pin of serial interface UART0
RxD1	Input	Serial data input pin of serial interface UART1
RxD2	Input	Serial data input pin of serial interface UART2
TxD0	Output	Serial data output pin of serial interface UART0
TxD1	Output	Serial data output pin of serial interface UART1
TxD2	Output	Serial data output pin of serial interface UART2
SCK00, SCK11, SCK20	I/O	Serial clock I/O pins of serial interfaces CSI00, CSI11, and CSI20
SI00, SI11, SI20	Input	Serial data input pins of serial interfaces CSI00, CSI11, and CSI20
SO00, SO11, SO20	Output	Serial data output pins of serial interfaces CSI00, CSI11, and CSI20
SCL00, SCL11, SCL20	Output	Serial clock output pins of serial interface simple I^2C (IIC00, IIC11, and IIC20)
SDA00, SDA11, SDA20	I/O	Serial data I/O pins of serial interface simple I^2C (IIC00, IIC11, and IIC20)
SCLA0	I/O	Clock I/O pin of serial interface IICA0
SDAA0	I/O	Serial data I/O pin of serial interface IICA0
TI00 to TI07	Input	The pins for inputting an external count clock/capture trigger to 16-bit timers 00 to 07
TO00 to TO07	Output	Timer output pins of 16-bit timers 00 to 07
X1, X2	—	Resonator connection for main system clock
XT1, XT2	—	Resonator connection for subsystem clock
EXCLK	Input	External clock input for main system clock
EXCLKS	Input	External clock input for subsystem clock
RTC1HZ	Output	Real-time clock correction clock (1 Hz) output
V_{DD}	—	Positive power supply
V_{SS}	—	Ground potential
TOOL0	I/O	Data I/O for flash memory programmer/debugger
TOOLRxD	Input	UART reception pin for the external device connection used during flash memory programmer
TOOLTxD	Output	UART transmission pin for the external device connection used during flash memory programmer

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

Table 2-1. Relationships Between P40/TOOL0 and Operation Mode After Reset Release

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

For details, see 22.4.2 Flash memory programming mode.

Remark Use bypass capacitors (about 0.1 µF) as noise and latch up countermeasures with relatively thick wires at the shortest distance to V_{DD} to V_{SS} line.

2.3 Connection of Unused Pins

Table 2-2 shows the connections of unused pins.

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

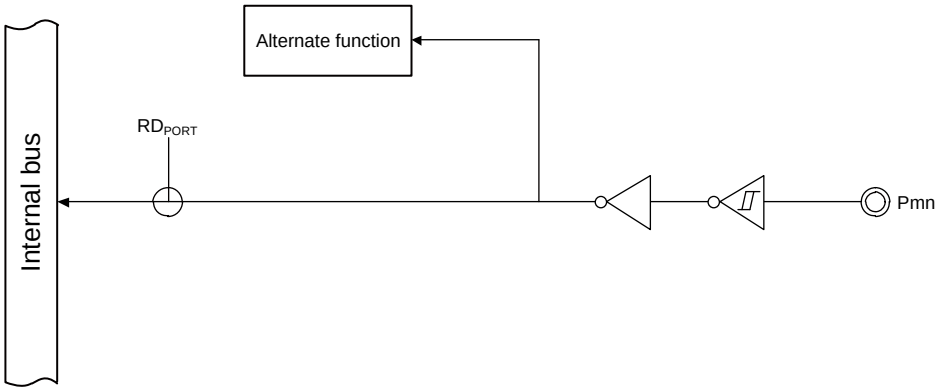
Table 2-2. Connections of Unused Pins

Pin Name	I/O	Recommended Connection of Unused Pins
P00 to P07	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P10 to P17		Input: Independently connect to V_{DD} 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.
P40/TOOL0		Input: Independently connect to V_{DD} via a resistor. Output: Leave open.
P41 to P43		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P60, P61		Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Set the port's output latch to 0 and leave the pins open, or set the port's output latch to 1 and independently connect the pins to V_{DD} or V_{SS} via a resistor.
P121, P122	I/O	Input: Independently connect to V_{DD} or V_{SS} via a resistor. Output: Leave open.
P125/$\overline{\text{RESET}}$	I/O	PORTSELB = 0: Input: Independently connect to V_{DD} via a resistor. Output: Leave open. PORTSELB = 1: Leave open, or connect to V_{DD}.
P137	Input	Independently connect to V_{DD} or V_{SS} via a resistor.

2.4 Block Diagrams of Pins

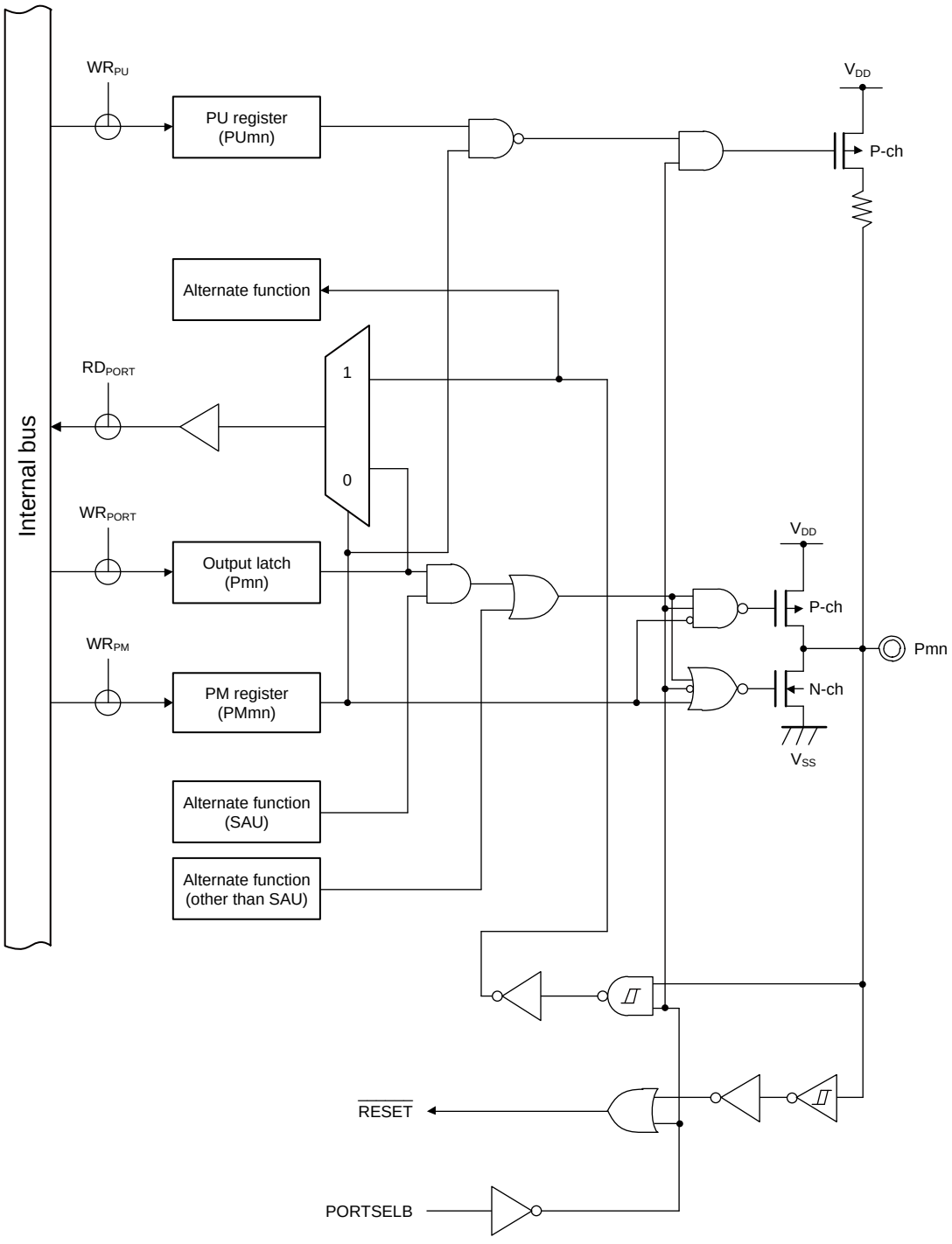
Figure 2-1 to **Figure 2-12** show the block diagrams of the pins described in **2.1.1 10-pin products** to **2.1.5 32-pin products**.

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



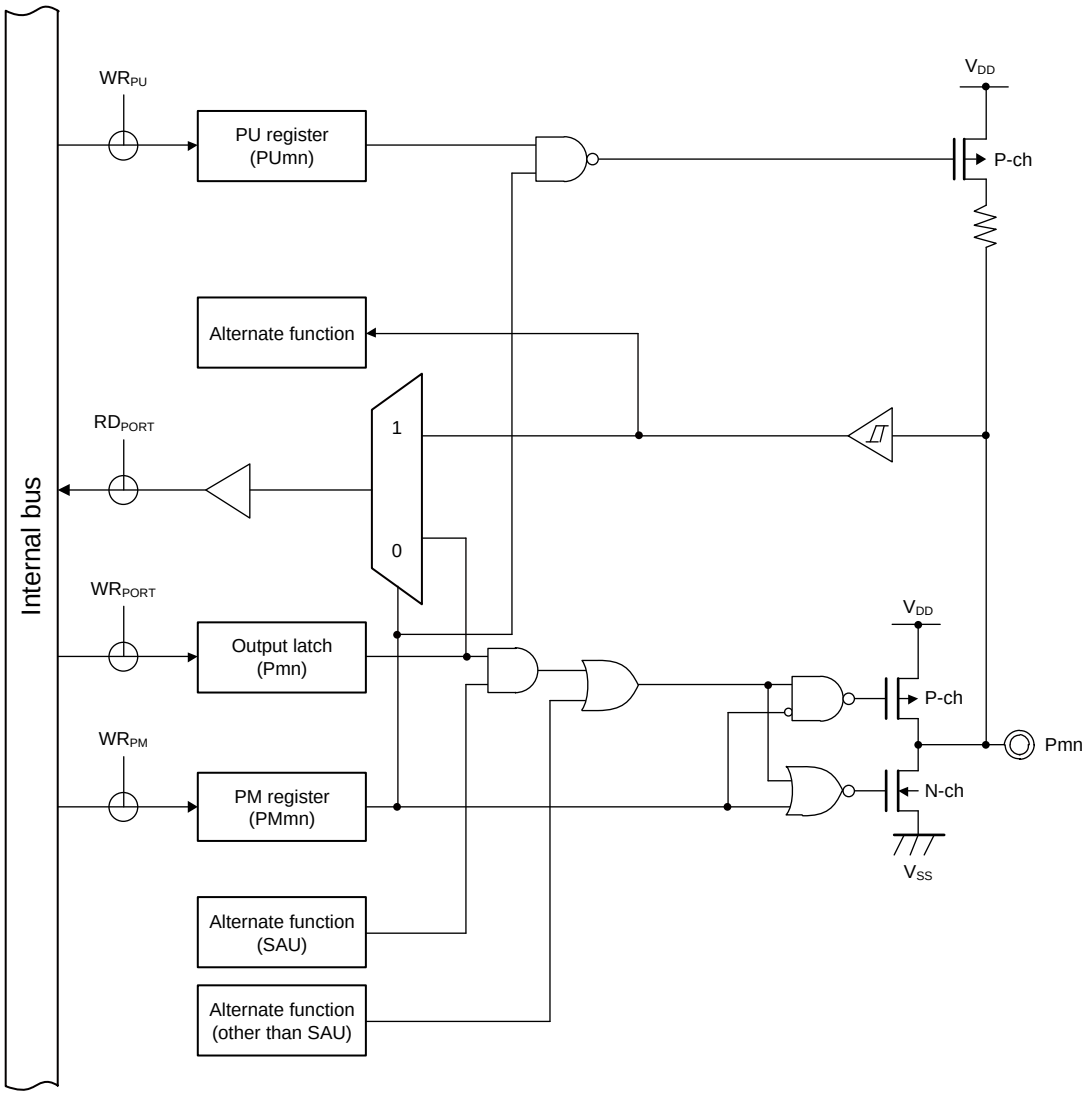
Remark For alternate functions, see **2.1 Port Function**.

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



Remark For alternate functions, see **2.1 Port Function**.

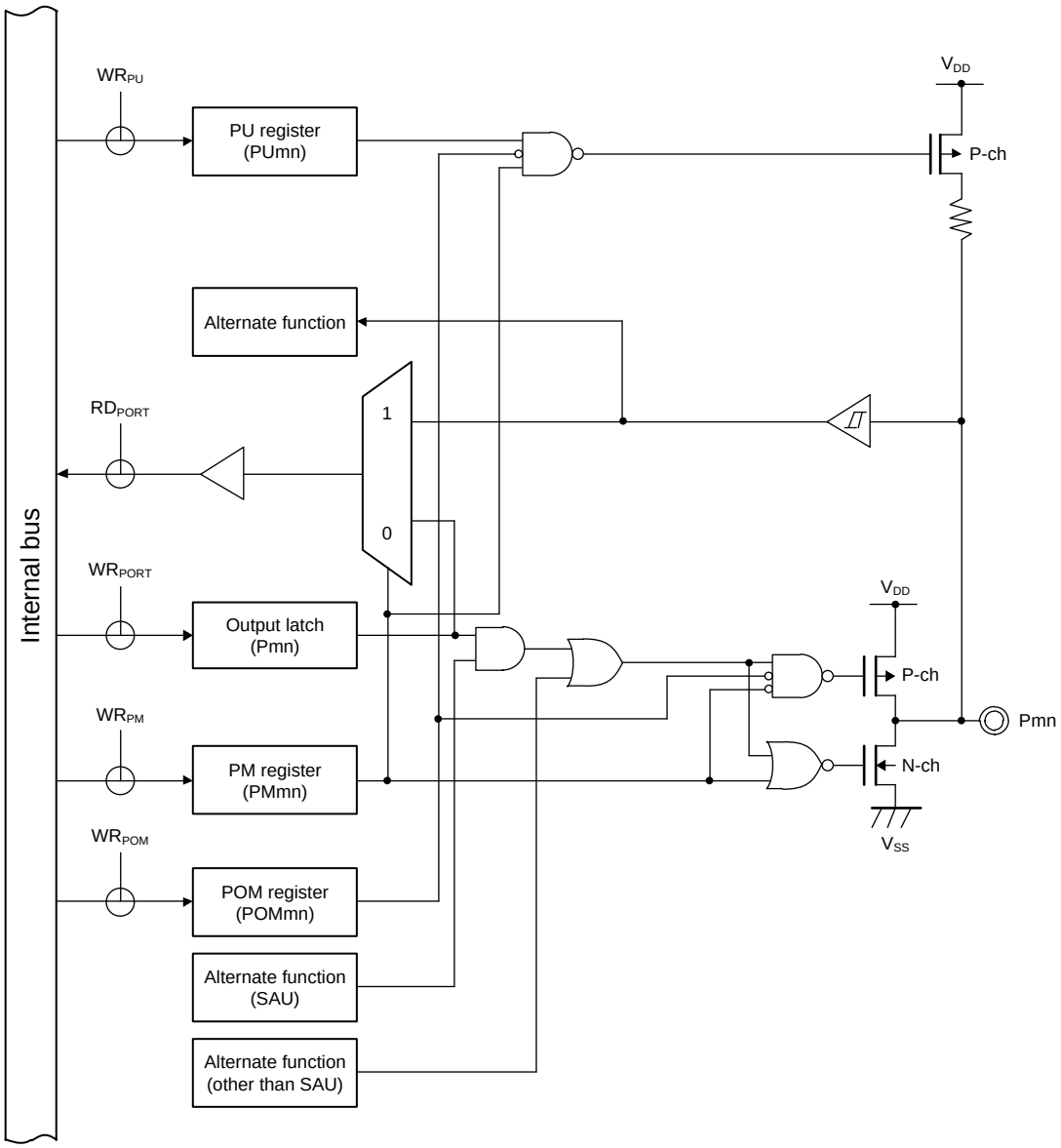
Figure 2-3. Pin Block Diagram for Pin Type 7-1-1



Remark 1. For alternate functions, see **2.1 Port Function**.

Remark 2. SAU: Serial array unit

Figure 2-4. Pin Block Diagram for Pin Type 7-1-2



Caution The input buffer is enabled even if the type 7-1-2 pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMxx). This may lead to a through current flowing through the type 7-1-2 pin when the voltage level on this pin is intermediate.

Remark 1. For alternate functions, see **2.1 Port Function**.

Remark 2. SAU: Serial array unit

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

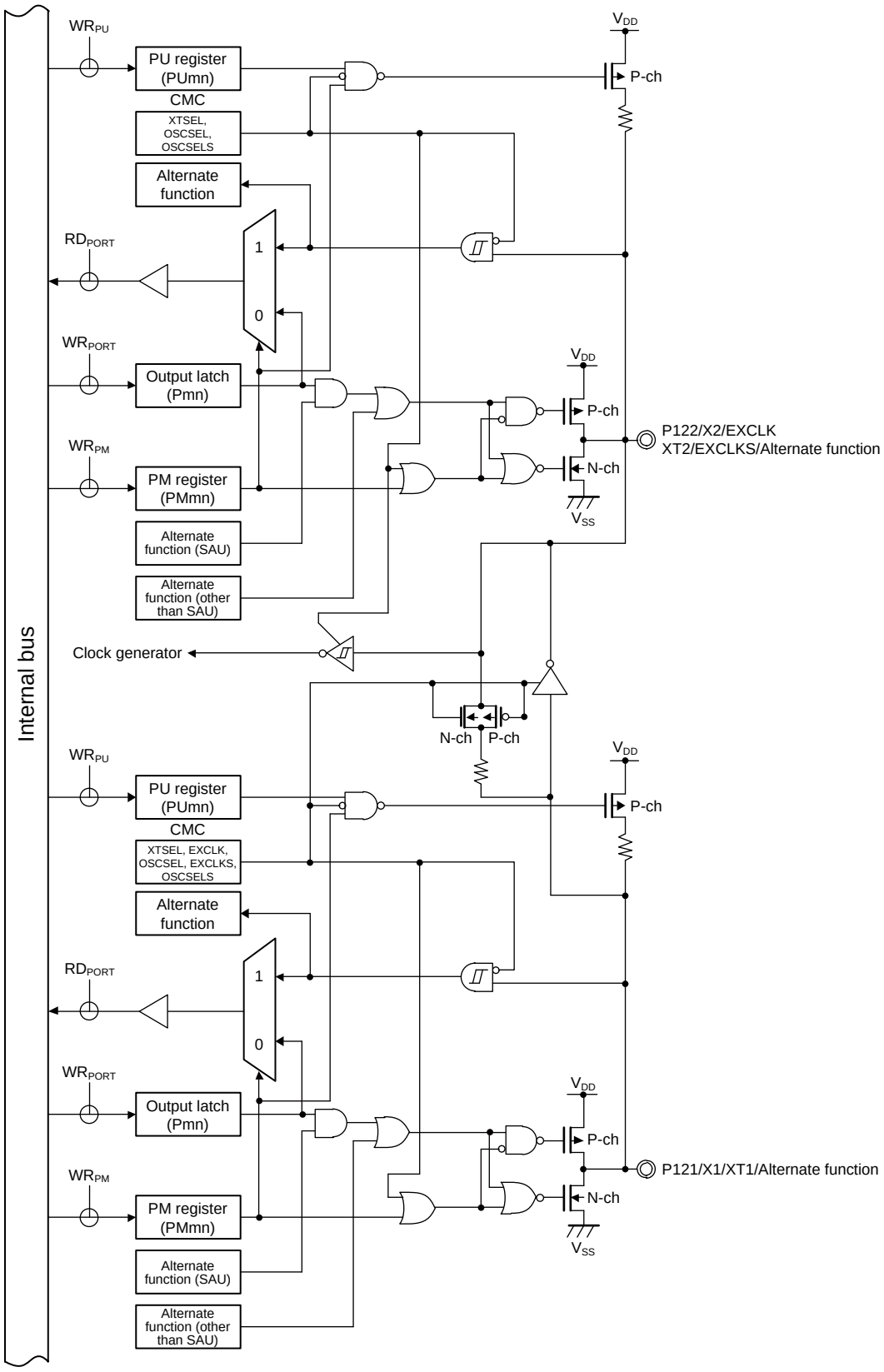
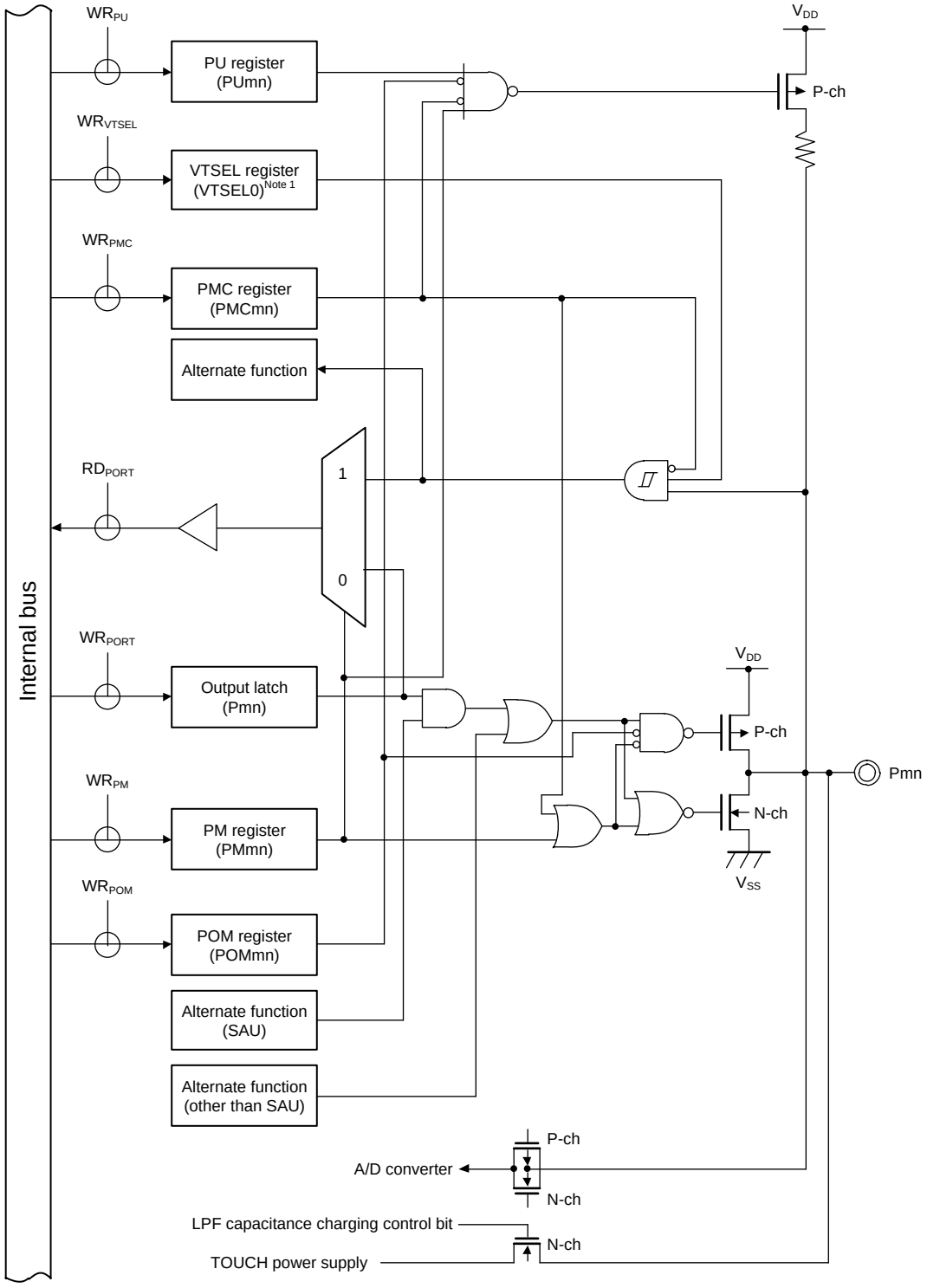


Figure 2-6. Pin Block Diagram for Pin Type 7-31-5

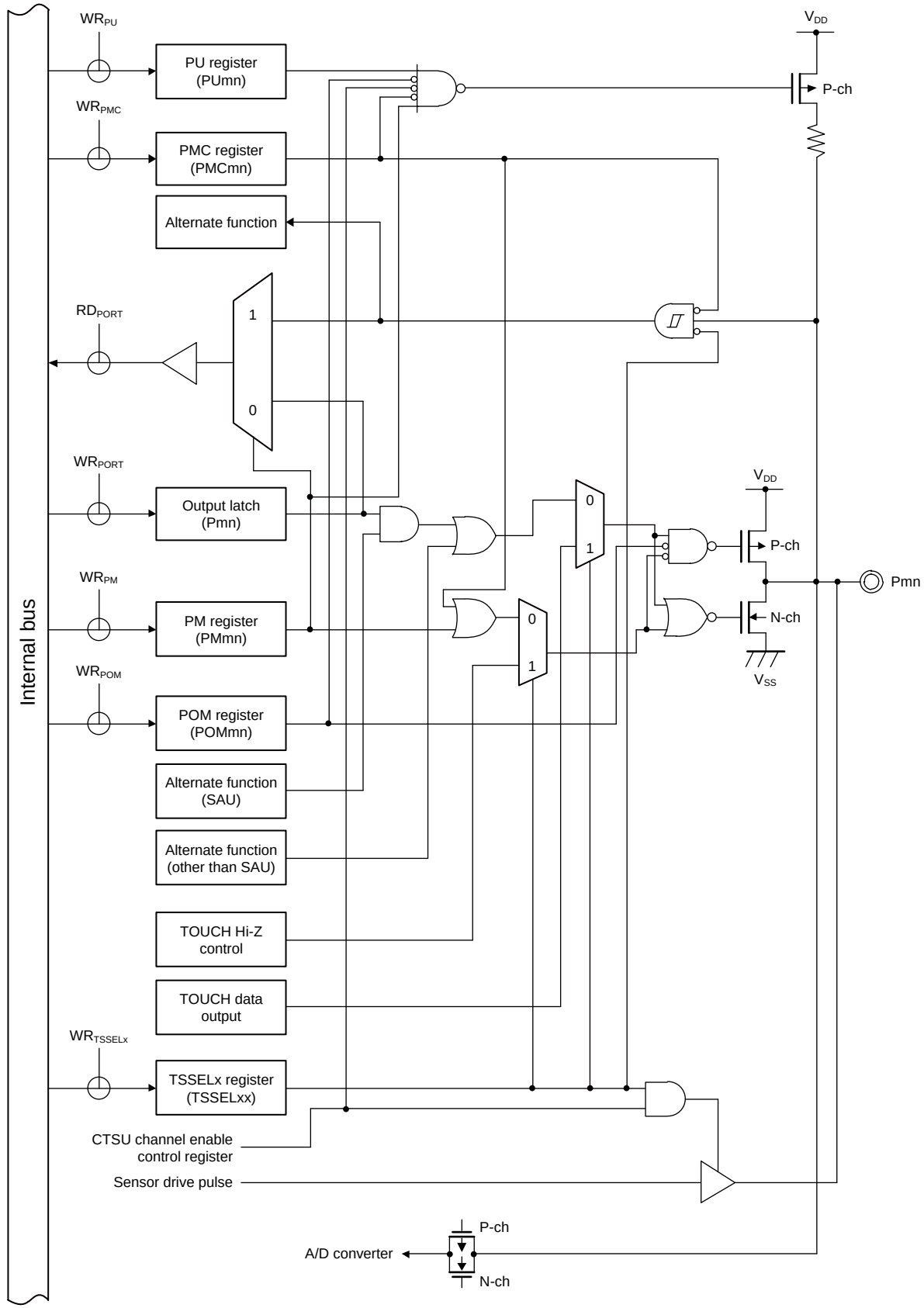


Note 1. The function of the VTSEL register in the figure of I/O circuit 7-31-5 is only enabled when any of the bits in the TSSELn (n = 0 to 2) registers is set to 1.

Remark 1. For alternate functions, see **2.1 Port Function**.

Remark 2. SAU: Serial array unit

Figure 2-7. Pin Block Diagram for Pin Type 7-33-5

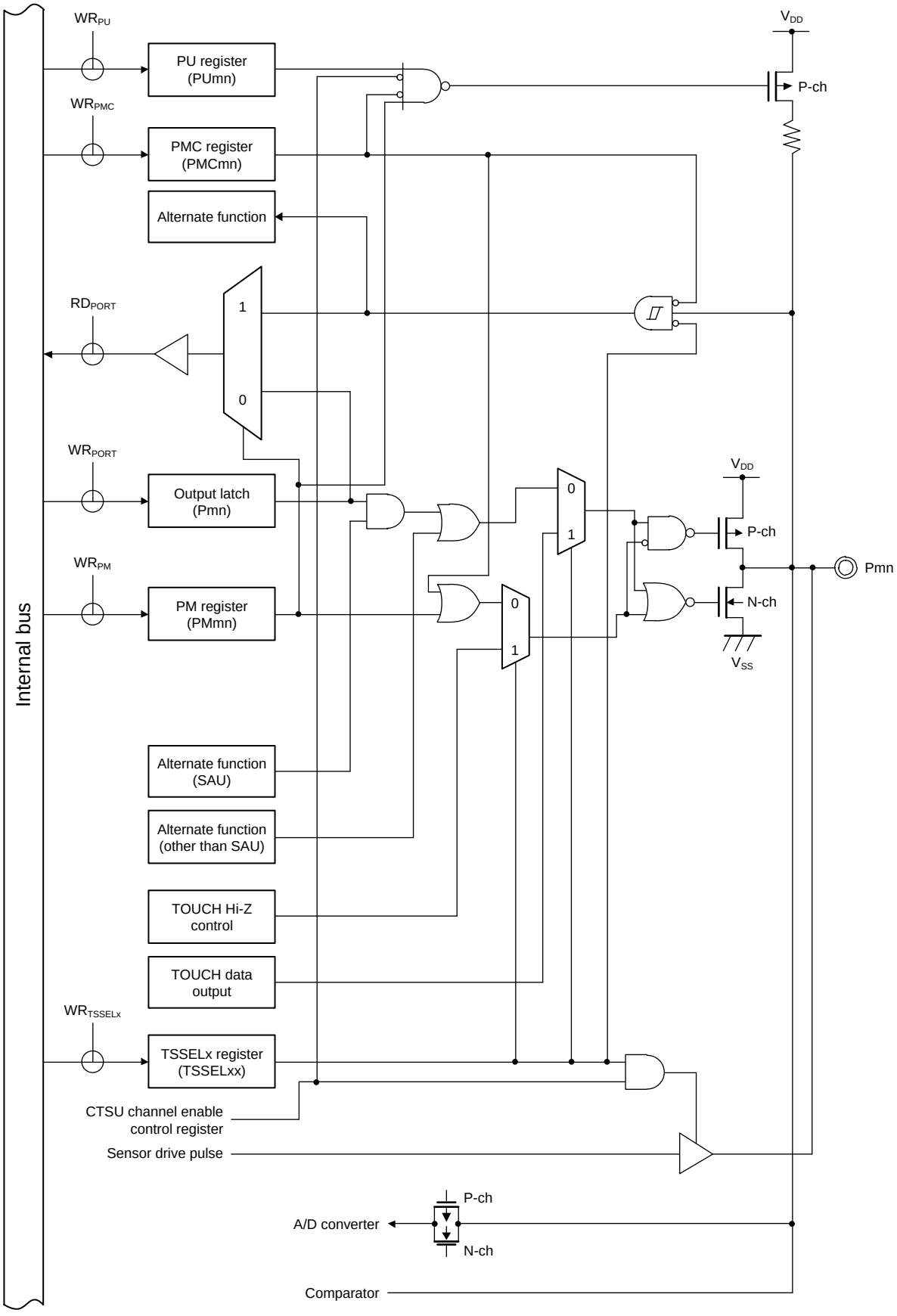


Caution The input buffer is enabled even if the type 7-33-5 pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMxx). This may lead to a through current flowing through the type 7-33-5 pin when the voltage level on this pin is intermediate.

Remark 1. For alternate functions, see **2.1 Port Function**.

Remark 2. SAU: Serial array unit

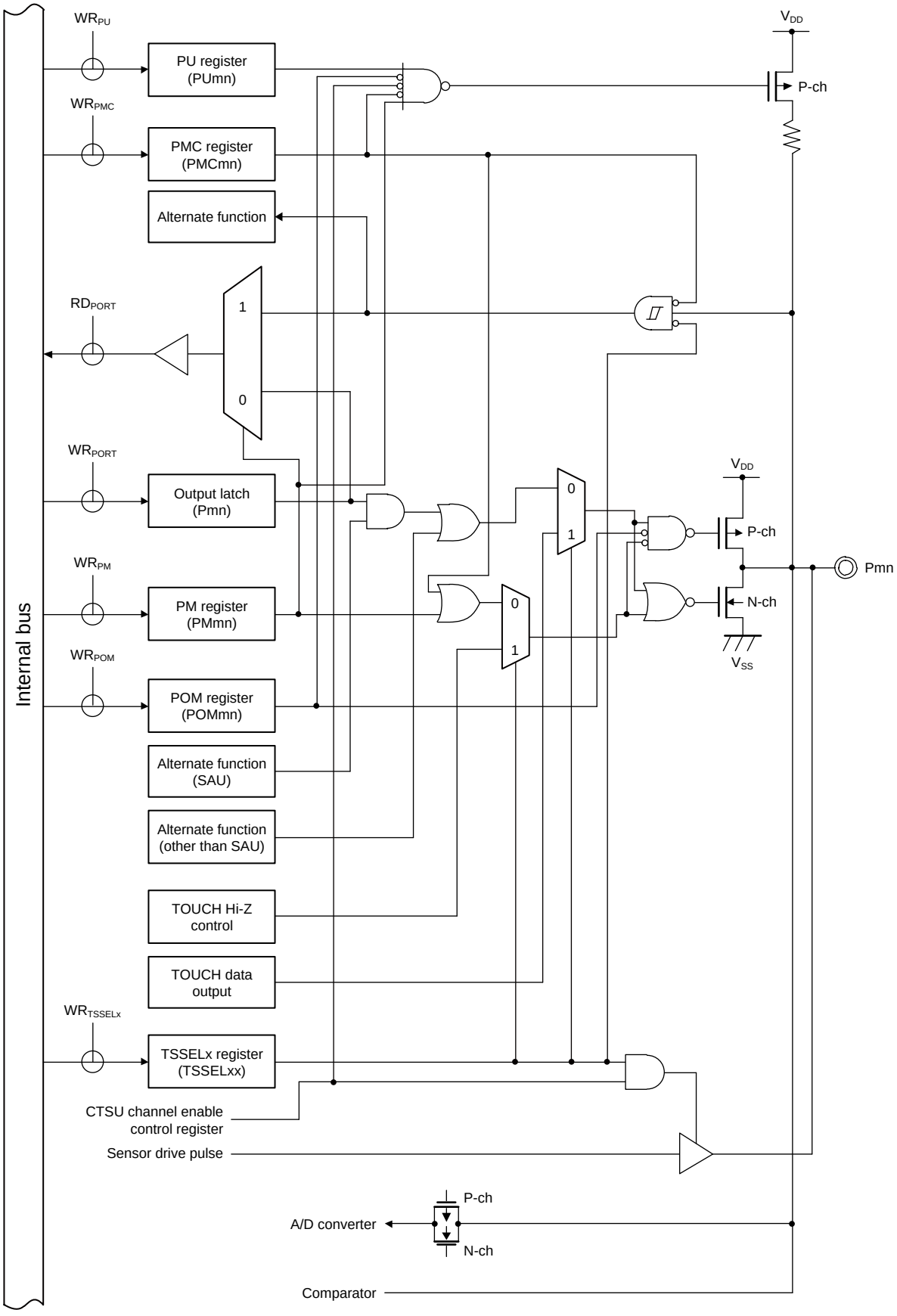
Figure 2-8. Pin Block Diagram for Pin Type 7-33-6



Remark 1. For alternate functions, see **2.1 Port Function**.

Remark 2. SAU: Serial array unit

Figure 2-9. Pin Block Diagram for Pin Type 7-33-7

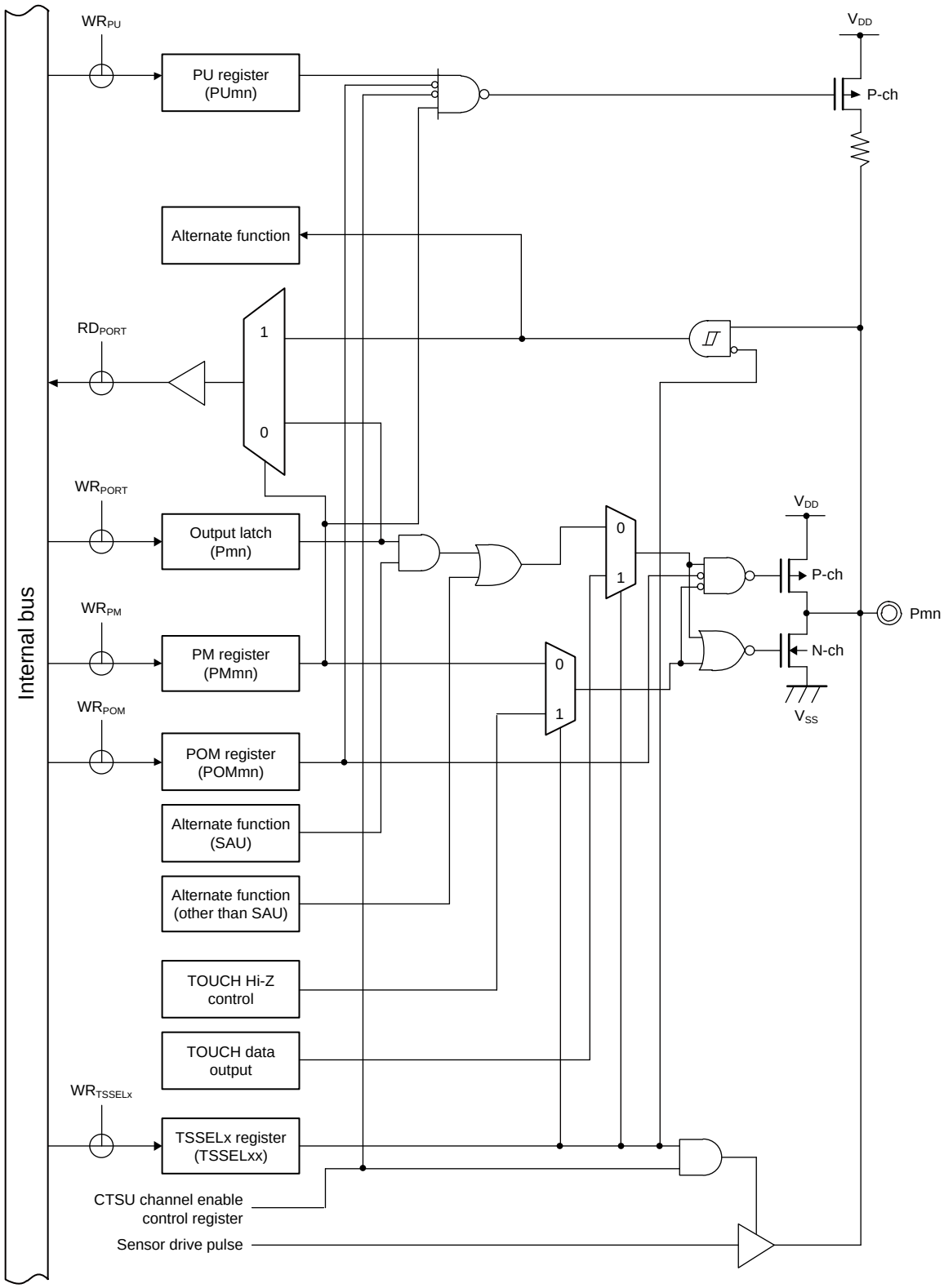


Caution The input buffer is enabled even if the type 7-33-7 pin is operating as an output when the N-ch open drain output mode is selected by the corresponding bit in the port output mode register (POMxx). This may lead to a through current flowing through the type 7-33-7 pin when the voltage level on this pin is intermediate.

Remark 1. For alternate functions, see **2.1 Port Function**.

Remark 2. SAU: Serial array unit

Figure 2-10. Pin Block Diagram for Pin Type 7-33-8



Remark 1. For alternate functions, see **2.1 Port Function**.

Remark 2. SAU: Serial array unit

Figure 2-11. Pin Block Diagram for Pin Type 7-33-9

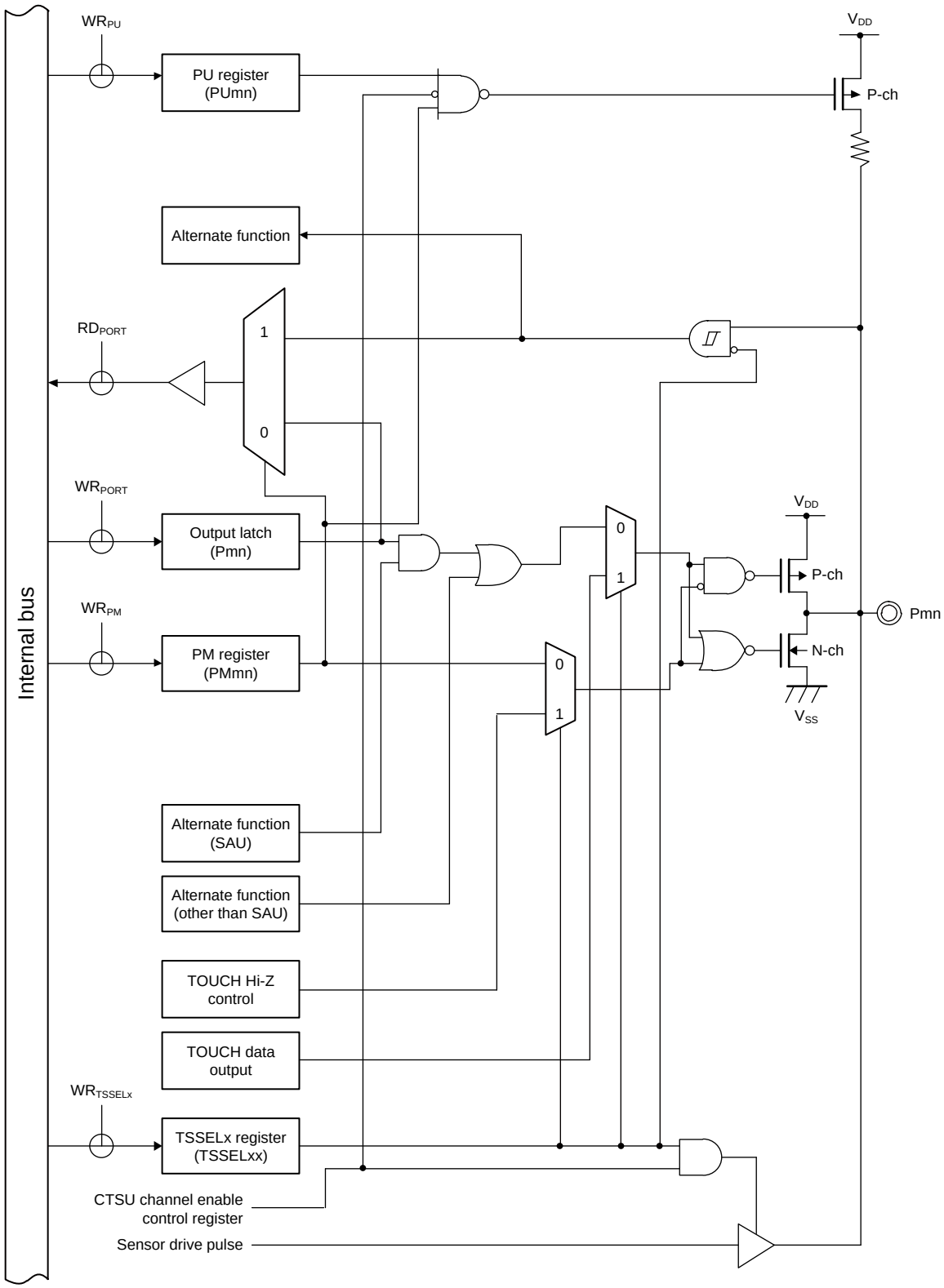
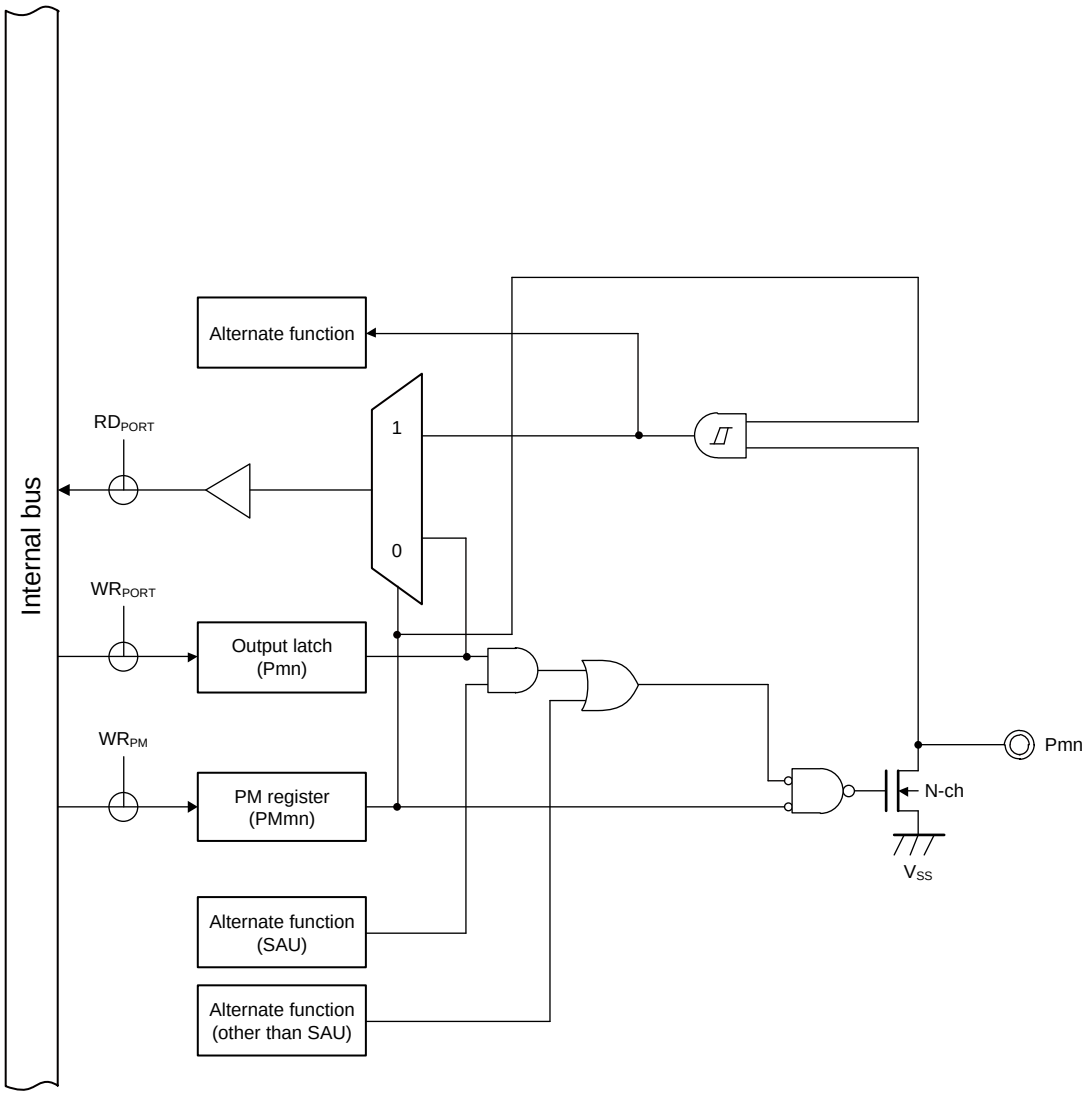


Figure 2-12. Pin Block Diagram for Pin Type 12-1-1



Remark 1. For alternate functions, see **2.1 Port Function**.

Remark 2. SAU: Serial array unit

CHAPTER 3 CPU ARCHITECTURE

3.1 Overview

The CPU core of the RL78 microcontroller is based on the Harvard architecture where instruction fetch buses, address buses, and data buses are kept separate. In addition, through the adoption of three-stage pipeline control of fetch, decode, and memory access, the operation efficiency is improved drastically 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.

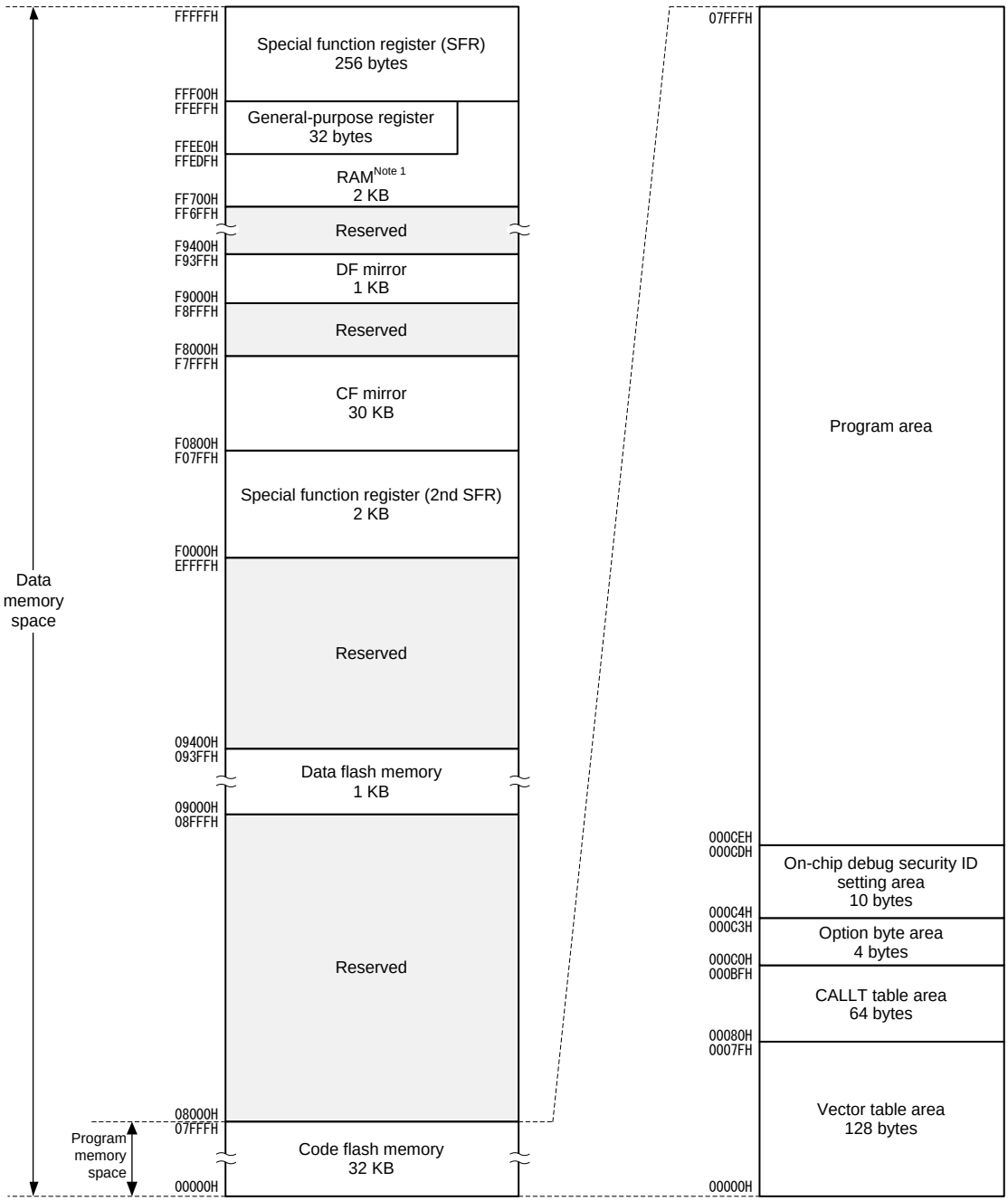
The RL78/G16 has the RL78-S2 CPU core. Its main features are as follows.

- 3-stage pipeline CISC architecture
- Address space: 1 Mbyte
- Minimum instruction execution time: One clock cycle for one instruction
- General-purpose register: 8-bit registers × 8 × 4 banks
- Types of instruction: 75
- Data allocation: Little endian

3.2 Memory Space

Products in the RL78/G16 can access a 1-MB address space. **Figure 3-1** and **Figure 3-2** show the memory maps.

Figure 3-1. Memory Map (R5F121xC (x = 1, 4, 6, 7, B))

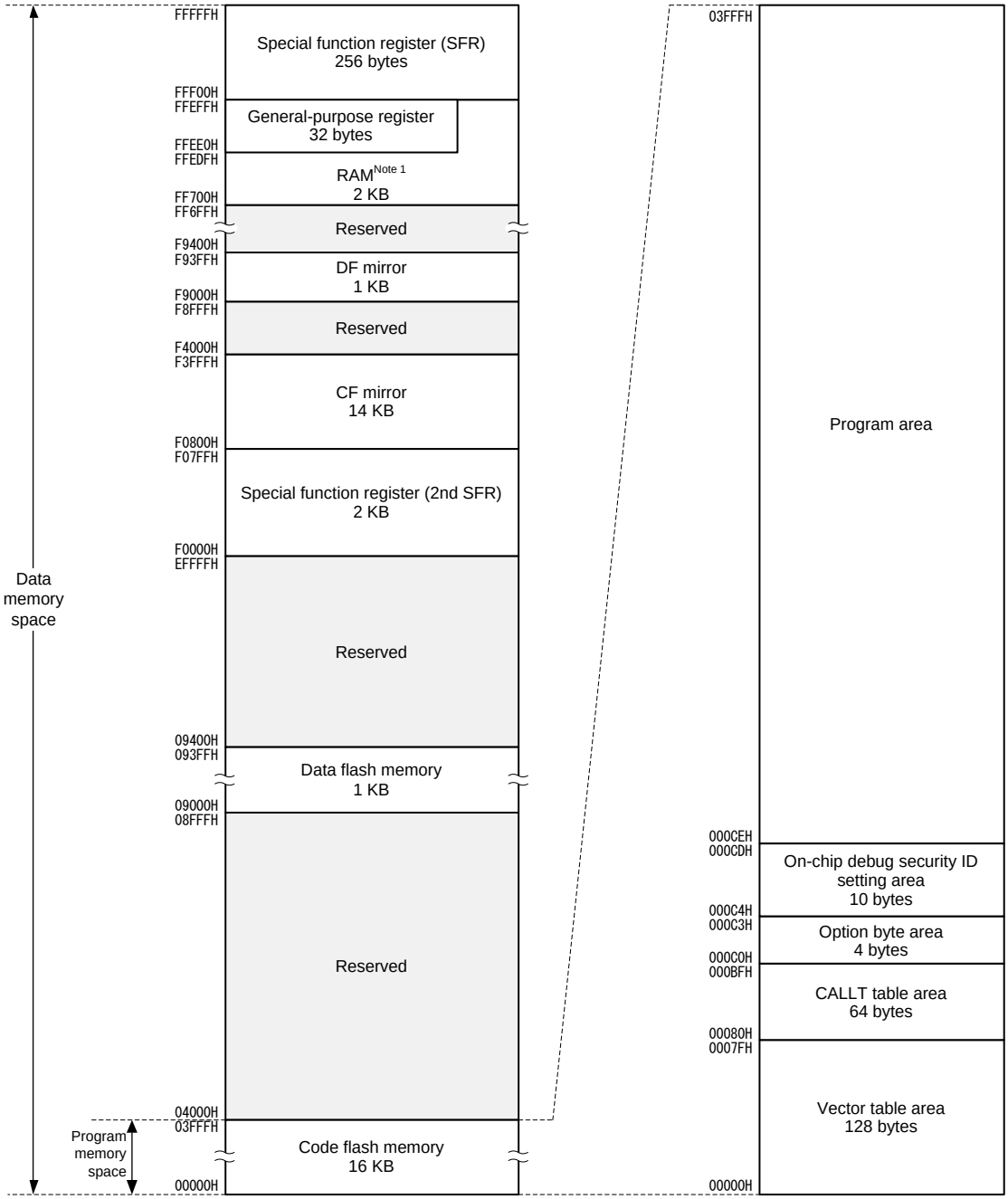


Note 1. Instructions can be executed from the RAM area excluding the general-purpose register area.

Caution If a generation of a reset by a RAM parity error is enabled (RPERDIS = 0), make sure to initialize "the RAM area to be used" when accessing data or "the RAM area to be used + 10-byte area" when executing instructions from the RAM area. A generation of a reset by a RAM parity error will be

enabled (RPERDIS = 0) in response to a generation of a reset. For details, refer to 20.3.2 RAM parity error detection.

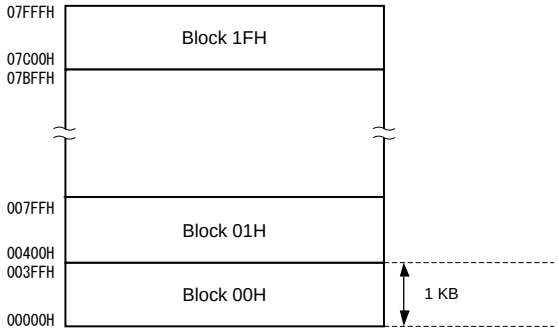
Figure 3-2. Memory Map (R5F121xA (x = 1, 4, 6, 7, B))



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

Caution If a generation of a reset by a RAM parity error is enabled (RPERDIS = 0), make sure to initialize "the RAM area to be used" when accessing data or "the RAM area to be used + 10-byte area" when executing instructions from the RAM area. A generation of a reset by a RAM parity error will be enabled (RPERDIS = 0) in response to a generation of a reset. For details, refer to 20.3.2 RAM parity error detection.

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



R5F121xC (x = 1, 4, 6, 7, B)

Table 3-1. Correspondence Between Address Values and Block Numbers in Flash Memory (1/2)

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

Table 3-1. Correspondence Between Address Values and Block Numbers in Flash Memory (2/2)

Address Value	Block Number
06800H to 06BFFH	1AH
06C00H to 06FFFH	1BH
07000H to 073FFH	1CH
07400H to 077FFH	1DH
07800H to 07BFFH	1EH
07C00H to 07FFFH	1FH

Remark R5F121xC: Block number 00H to 1FH
R5F121xA: Block number 00H to 0FH

3.2.1 Internal program memory space

The internal program memory space stores the program and table data. The RL78/G16 products incorporate internal ROM (flash memory), as shown below.

Table 3-2. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
R5F121xC (x = 1, 4, 6, 7, B)	Flash memory	32768 × 8 bits (00000H to 07FFFH)
R5F121xA (x = 1, 4, 6, 7, B)		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.

Table 3-3. Vector Table (1/2)

Vector Table Address	Interrupt Source	32-pin	24-pin	20-pin	16-pin	10-pin
00000H	RESET, SPOR, WDT, TRAP, IAW, RPE, DDDR	✓	✓	✓	✓	✓
00004H	INTWDTI	✓	✓	✓	✓	✓
00006H	INTP0	✓	✓	✓	✓	✓
00008H	INTP1	✓	✓	✓	✓	✓
0000AH	INTP2	✓	✓	✓	✓	✓
0000CH	INTP3	✓	✓	✓	✓	✓
0000EH	INTP4	✓	✓	✓	✓	✓
00010H	INTP5	✓	✓	✓	✓	✓
00012H	INTST0, INTCSI00, INTIIC00	✓	✓	✓	✓	✓
00014H	INTSR0	✓	✓	✓	✓	✓
00016H	INTSRE0	✓	✓	✓	✓	✓
00018H	INTTM01H	✓	✓	✓	✓	✓
0001AH	INTTM00	✓	✓	✓	✓	✓
0001CH	INTTM01	✓	✓	✓	✓	✓
0001EH	INTST1	✓	✓	✓	✓	—
00020H	INTSR1, INTCSI11, INTIIC11	✓	✓	✓	✓	—
00022H	INTSRE1	✓	✓	✓	✓	—
00024H	INTST2, INTCSI20, INTIIC20	✓	✓	✓	—	—
00026H	INTSR2	✓	✓	✓	—	—
00028H	INTSRE2	✓	✓	✓	—	—
0002AH	INTTM03H	✓	✓	✓	✓	✓
0002CH	INTIICA0	✓	✓	✓	✓	✓
0002EH	INTTM02	✓	✓	✓	✓	✓
00030H	INTTM03	✓	✓	✓	✓	✓
00032H	INTAD	✓	✓	✓	✓	✓
00034H	INTRTC	✓	✓	✓	✓	—
00036H	INTRTIT	✓	✓	✓	✓	—
00038H	INTIT	✓	✓	✓	✓	✓
0003AH	INTTM04	✓	✓	✓	✓	✓
0003CH	INTTM05	✓	✓	✓	✓	✓

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

Vector Table Address	Interrupt Source	32-pin	24-pin	20-pin	16-pin	10-pin
0003EH	INTTM06	✓	✓	✓	✓	✓
00040H	INTTM07	✓	✓	✓	✓	✓
00042H	INTP6	✓	✓	✓	✓	✓
00044H	INTP7	✓	✓	✓	✓	✓
00046H	INTP8	✓	✓	—	—	—
00048H	INTP9	✓	✓	—	—	—
0004AH	INTCMP0	✓	✓	✓	✓	✓
0004CH	INTCMP1	✓	✓	✓	✓	—
0004EH	INTCTSUWR	✓	✓	✓	✓	✓
00050H	INTCTSURD	✓	✓	✓	✓	✓
00052H	INTCTSUFN	✓	✓	✓	✓	✓
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 of 2 bytes).

(3) Option byte area

A 4-byte area of 000C0H to 000C3H can be used as an option byte area. For details, see **CHAPTER 21 OPTION BYTE**.

(4) On-chip debug security ID setting area

A 10-byte area of 000C4H to 000CDH can be used as an on-chip debug security ID setting area. For details, see **CHAPTER 23 ON-CHIP DEBUG FUNCTION**.

3.2.2 Mirror area

The RL78/G16 mirrors the code flash area of 00800H to 07FFFH, to F0800H to F7FFFH. It also mirrors the data flash area of 09000H to 093FFH, to F9000H to F93FFH.

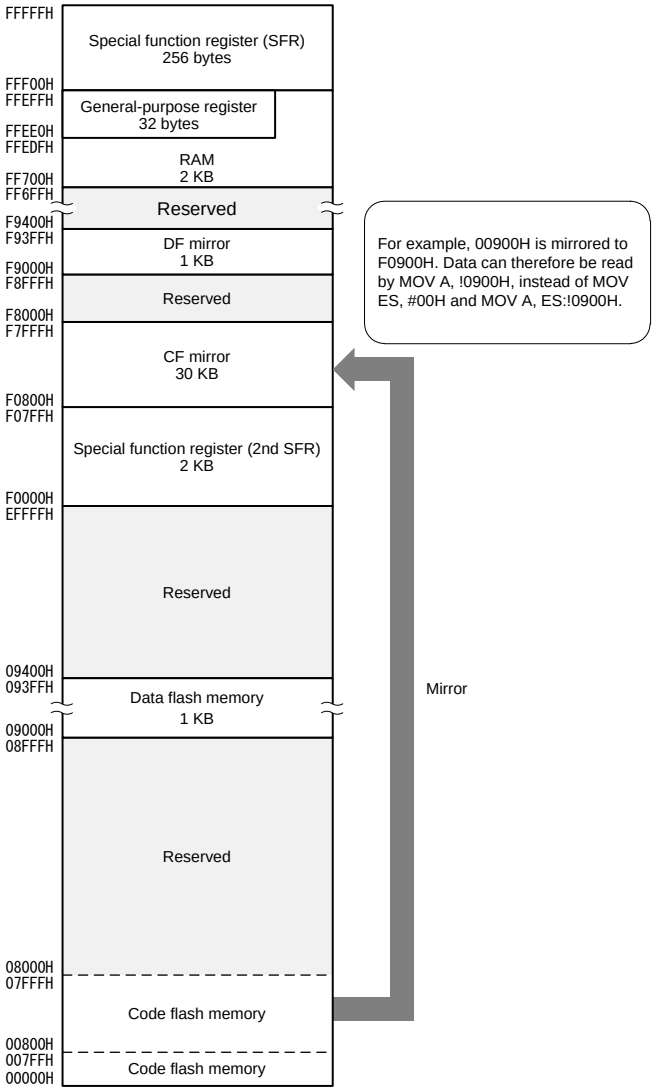
By reading data from F0800H to F7FFFH and F9000H to F93FFH, an instruction that does not have the ES register as an operand can be used, and thus the contents of the code flash can be read with the shorter code. However, the code flash area is not mirrored to the SFR, extended SFR, RAM, and use prohibited areas.

See **3.2 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 shows examples.

Example) R5F121xC (x = 1, 4, 6, 7, B) (Flash memory: 32 KB, RAM: 2 KB)



3.2.3 Internal data memory space

The RL78/G16 products incorporate the following RAMs.

Table 3-4. Internal RAM Capacity

Part Number	Internal RAM
R5F121xC (x = 1, 4, 6, 7, B)	2048 × 8 bits (FF700H to FFEFFH)
R5F121xA (x = 1, 4, 6, 7, B)	

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

3.2.4 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 **3.3.4 Special function registers (SFRs)**).

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

3.2.5 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 **3.3.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

SFRs other than those in the SFR area (FFF00H to FFFFFH) are allocated to this area. An instruction that accesses the extended SFR area, however, is 1 byte longer than an instruction that accesses the SFR area.

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

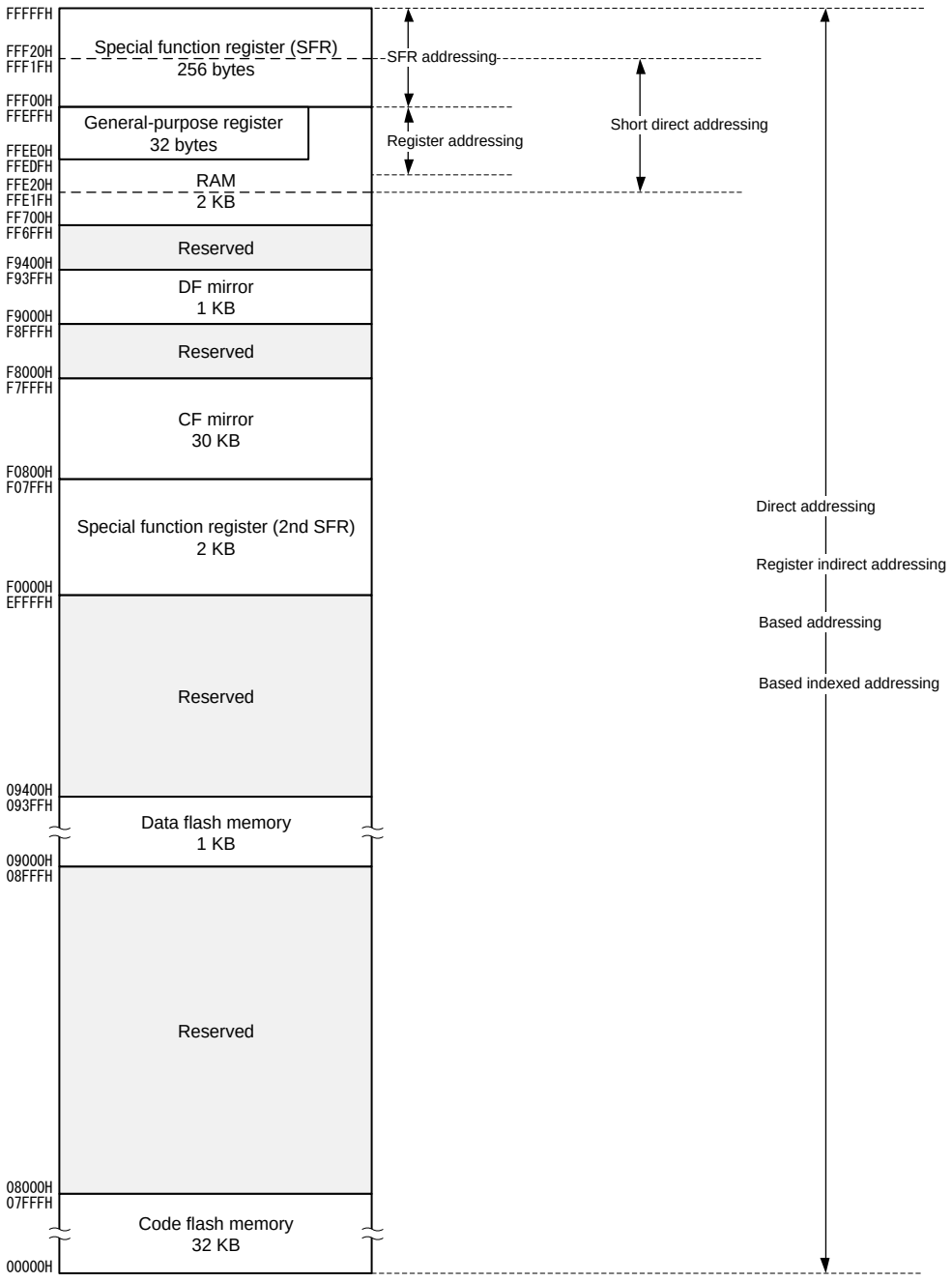
3.2.6 Data memory addressing

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/G16, based on operability and other considerations. 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 **3.5 Addressing for Processing Data Addresses**.

Figure 3-3. Correspondence Between Data Memory and Addressing



3.3 Processor Registers

The RL78/G16 products incorporate the following processor registers.

3.3.1 Control registers

The control registers control the program sequence, status, 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 16 lower-order bits of the program counter. The four higher-order bits of the program counter are cleared to 0000.

Figure 3-4. Format of Program Counter

Symbol	19															0
PC																

(2) Program status word (PSW)

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

Program status word contents are stored in the stack area upon vectored interrupt request is acknowledged or PUSH PSW instruction execution and are restored upon execution of the RETB, RETI and POP PSW instructions.

Reset signal generation sets the PSW register to 06H.

Figure 3-5. Format of Program Status Word

Symbol	7	6	5	4	3	2	1	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 maskable 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 or comparison result is zero or equal, 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)

These flags manage 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) (see **14.3.3**) cannot be acknowledged. Actual vectored interrupt 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

Symbol	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 a stack area.

3.3.2 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).

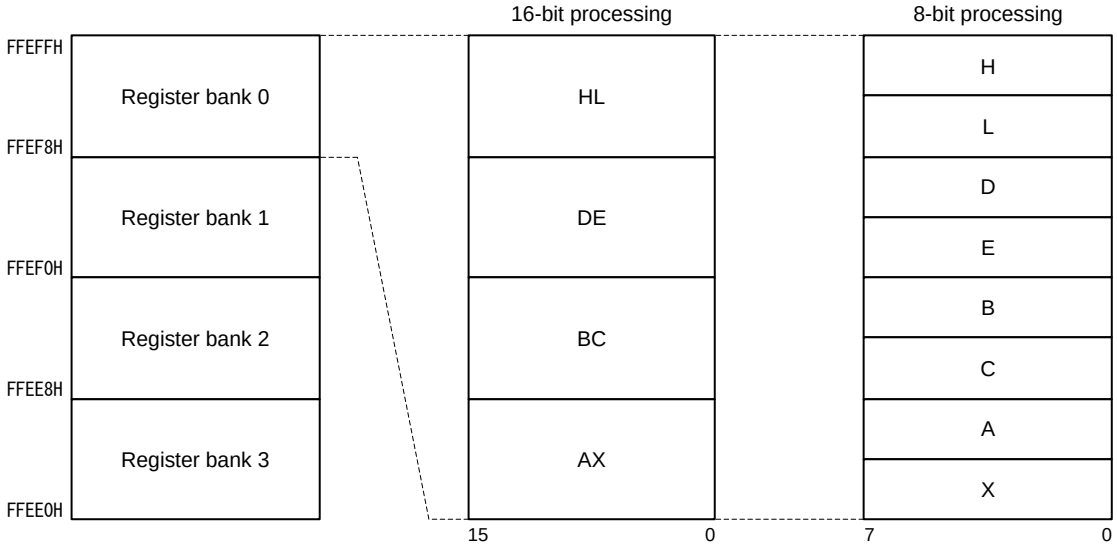
Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

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



3.3.3 ES and CS registers

The ES register and CS register are used to specify the higher address for data access and when a branch instruction is executed (register indirect addressing), respectively.

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

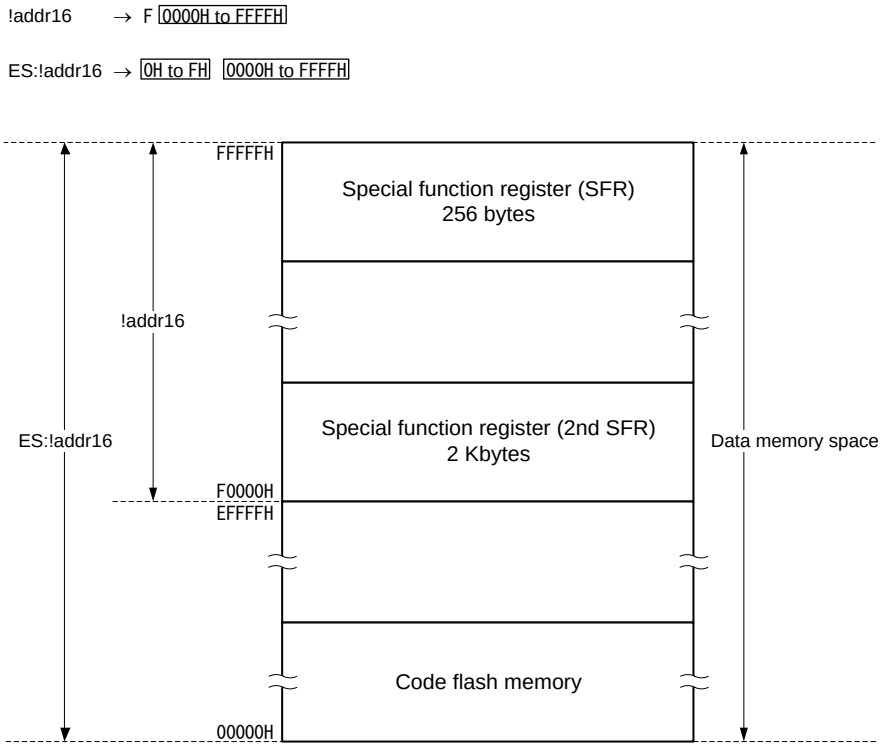
Figure 3-8. Configuration of ES and CS Registers

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

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

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

Figure 3-9. Extension of Data Area Which Can Be Accessed



3.3.4 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 a list of the SFRs. The meanings of items in the table are as follows.

[Symbol]

Symbol indicating 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]

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]

Indicates each register status upon reset signal generation.

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

Remark For extended SFRs (2nd SFRs), see **3.3.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**.

Table 3-5. SFR List (1/3)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF00H	Port register 0	P0		R/W	✓	✓	—	00H
FFF01H	Port register 1	P1		R/W	✓	✓	—	00H
FFF02H	Port register 2	P2		R/W	✓	✓	—	00H
FFF04H	Port register 4	P4		R/W	✓	✓	—	00H
FFF06H	Port register 6	P6		R/W	✓	✓	—	00H
FFF0CH	Port register 12	P12		R/W	✓	✓	—	Undefined
FFF0DH	Port register 13	P13		R/W	✓	✓	—	Undefined
FFF10H	Serial data register 00	TXD0/SIO00	SDR00	R/W	—	✓	✓	0000H
FFF11H		—			—	—		
FFF12H	Serial data register 01	RXD0	SDR01	R/W	—	✓	✓	0000H
FFF13H		—			—	—		
FFF18H	Timer data register 00	TDR00		R/W	—	—	✓	0000H
FFF19H								
FFF1AH	Timer data register 01	TDR01L	TDR01	R/W	—	✓	✓	00H
FFF1BH		TDR01H			—	✓		00H
FFF1EH	10-bit A/D conversion result register	ADCR		R	—	—	✓	0000H
FFF1FH	8-bit A/D conversion result register	ADCRH		R	—	✓	—	00H
FFF20H	Port mode register 0	PM0		R/W	✓	✓	—	FFH
FFF21H	Port mode register 1	PM1		R/W	✓	✓	—	FFH
FFF22H	Port mode register 2	PM2		R/W	✓	✓	—	FFH
FFF24H	Port mode register 4	PM4		R/W	✓	✓	—	FFH
FFF26H	Port mode register 6	PM6		R/W	✓	✓	—	FFH
FFF2CH	Port mode register 12	PM12		R/W	✓	✓	—	FFH
FFF30H	A/D converter mode register 0	ADM0		R/W	✓	✓	—	00H
FFF31H	Analog input channel specification register	ADS		R/W	✓	✓	—	00H
FFF38H	External interrupt rising edge enable register 0	EGP0		R/W	✓	✓	—	00H
FFF39H	External interrupt falling edge enable register 0	EGN0		R/W	✓	✓	—	00H
FFF3AH	External interrupt rising edge enable register 1	EGP1		R/W	✓	✓	—	00H
FFF3BH	External interrupt falling edge enable register 1	EGN1		R/W	✓	✓	—	00H
FFF44H	Serial data register 02	TXD1	SDR02	R/W	—	✓	✓	0000H
FFF45H		—			—	—		
FFF46H	Serial data register 03	RXD1/SIO11	SDR03	R/W	—	✓	✓	0000H
FFF47H		—			—	—		
FFF48H	Serial data register 10	TXD2/SIO20	SDR10	R/W	—	✓	✓	0000H
FFF49H		—			—	—		
FFF4AH	Serial data register 11	RXD2	SDR11	R/W	—	✓	✓	0000H
FFF4BH		—			—	—		
FFF50H	IICA shift register 0	IICA0		R/W	—	✓	—	00H
FFF51H	IICA status register 0	IICS0		R	✓	✓	—	00H
FFF52H	IICA flag register 0	IICF0		R/W	✓	✓	—	00H

Table 3-5. SFR List (2/3)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF60H	Comparator mode setting register	COMPMDR		R/W	✓	✓	—	00H
FFF61H	Comparator filter control register	COMPFIR		R/W	✓	✓	—	00H
FFF62H	Comparator output control register	COMPOCR		R/W	✓	✓	—	00H
FFF64H	Timer data register 02	TDR02		R/W	—	—	✓	0000H
FFF65H								
FFF66H	Timer data register 03	TDR03L	TDR03	R/W	—	✓	✓	00H
FFF67H		TDR03H			—	✓		00H
FFF68H	Timer data register 04	TDR04		R/W	—	—	✓	0000H
FFF69H								
FFF6AH	Timer data register 05	TDR05		R/W	—	—	✓	0000H
FFF6BH								
FFF6CH	Timer data register 06	TDR06		R/W	—	—	✓	0000H
FFF6DH								
FFF6EH	Timer data register 07	TDR07		R/W	—	—	✓	0000H
FFF6FH								
FFF90H	Interval timer control register	ITMC		R/W	—	—	✓	0FFFH
FFF91H								
FFF92H	Second count register	SEC		R/W	—	✓	—	Undefined
FFF93H	Minute count register	MIN		R/W	—	✓	—	Undefined
FFF94H	Hour count register	HOUR		R/W	—	✓	—	Undefined
FFF95H	Week count register	WEEK		R/W	—	✓	—	Undefined
FFF96H	Day count register	DAY		R/W	—	✓	—	Undefined
FFF97H	Month count register	MONTH		R/W	—	✓	—	Undefined
FFF98H	Year count register	YEAR		R/W	—	✓	—	Undefined
FFF9AH	Alarm minute register	ALARMWM		R/W	—	✓	—	Undefined
FFF9BH	Alarm hour register	ALARMWH		R/W	—	✓	—	Undefined
FFF9CH	Alarm week register	ALARMWW		R/W	—	✓	—	Undefined
FFF9DH	Real-time clock control register 0	RTCC0		R/W	✓	✓	—	00H Note 1
FFF9EH	Real-time clock control register 1	RTCC1		R/W	✓	✓	—	00H Note 1
FFFA0H	Clock operation mode control register	CMC		R/W	—	✓	—	00H
FFFA1H	Clock operation status control register	CSC		R/W	✓	✓	—	C0H
FFFA2H	Oscillation stabilization time counter status register	OSTC		R	✓	✓	—	00H
FFFA3H	Oscillation stabilization time select register	OSTS		R/W	—	✓	—	07H
FFFA4H	System clock control register	CKC		R/W	✓	✓	—	00H
FFFA5H	Clock output select register 0	CKS0		R/W	✓	✓	—	00H
FFFA8H	Reset control flag register	RESF		R	—	✓	—	Undefined Note 2
FFFABH	Watchdog timer enable register	WDTE		R/W	—	✓	—	1AH/9AH Note 3
FFFACH	CRC input register	CRCIN		R/W	—	✓	—	00H
FFFD0H	Interrupt request flag register 2L	IF2L	IF2	R/W	✓	✓	✓	00H
FFFD4H	Interrupt mask flag register 2L	MK2L	MK2	R/W	✓	✓	✓	FFH
FFFD8H	Priority specification flag register 02L	PR02L	PR02	R/W	✓	✓	✓	FFH
FFFDCH	Priority specification flag register 12L	PR12L	PR12	R/W	✓	✓	✓	FFH

Table 3-5. SFR List (3/3)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFFE0H	Interrupt request flag register 0	IF0L	IF0	R/W	✓	✓	✓	00H
FFFE1H		IF0H		R/W	✓	✓		00H
FFFE2H	Interrupt request flag register 1	IF1L	IF1	R/W	✓	✓	✓	00H
FFFE3H		IF1H		R/W	✓	✓		00H
FFFE4H	Interrupt mask flag register 0	MK0L	MK0	R/W	✓	✓	✓	FFH
FFFE5H		MK0H		R/W	✓	✓		FFH
FFFE6H	Interrupt mask flag register 1	MK1L	MK1	R/W	✓	✓	✓	FFH
FFFE7H		MK1H		R/W	✓	✓		FFH
FFFE8H	Priority specification flag register 00	PR00L	PR00	R/W	✓	✓	✓	FFH
FFFE9H		PR00H		R/W	✓	✓		FFH
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

Note 1. It is initialized only in case of a reset by data retention lower limit voltage.

Note 2. The reset values of the registers vary depending on the reset source as shown below.

Register \ Reset Source		$\overline{\text{RESET}}$ input	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by RAM Parity Error	Reset by Illegal Access to Memory	Reset by SPOR	Reset by data retention lower limit voltage
RESF	TRAP	Cleared (0)	Set (1)	Held			Held	Cleared (0)
	WDTRF		Held	Set (1)	Held			
	RPERF		Held		Set (1)	Held		
	IAWRF		Held			Set (1)		
	SPORF		Held				Set (1)	

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

Remark For extended SFRs (2nd SFRs), see **Table 3-6. Extended SFR (2nd SFR) List (1/6)**.

3.3.5 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 extended 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 a list of the extended SFRs. The meanings of items in the table are as follows.

[Symbol]

Symbol indicating 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]

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]

Indicates each register status upon reset signal generation.

Caution **Do not access addresses to which extended SFRs (2nd SFRs) are not assigned.**

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

Table 3-6. Extended SFR (2nd SFR) List (1/6)

Address	Special Function Register (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
F0013H	A/D test register	ADTES	R/W	—	✓	—	00H
F0030H	Pull-up resistor option register 0	PU0	R/W	✓	✓	—	00H
F0031H	Pull-up resistor option register 1	PU1	R/W	✓	✓	—	00H
F0032H	Pull-up resistor option register 2	PU2	R/W	✓	✓	—	00H
F0034H	Pull-up resistor option register 4	PU4	R/W	✓	✓	—	01H
F003CH	Pull-up resistor option register 12	PU12	R/W	✓	✓	—	00H
F004EH	Port input mode register 14	PIM14	R/W	✓	✓	—	00H
F0050H	Port output mode register 0	POM0	R/W	✓	✓	—	00H
F0051H	Port output mode register 1	POM1	R/W	✓	✓	—	00H
F0052H	Port output mode register 2	POM2	R/W	✓	✓	—	00H
F0054H	Port output mode register 4	POM4	R/W	✓	✓	—	00H
F0060H	Port mode control register 0	PMC0	R/W	✓	✓	—	FFH
F0062H	Port mode control register 2	PMC2	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
F0076H	Peripheral I/O redirection register 6	PIOR6	R/W	—	✓	—	00H
F0077H	Peripheral I/O redirection register 0	PIOR0	R/W	—	✓	—	00H
F0078H	Illegal memory access detection control register	IAWCTL	R/W	—	✓	—	00H
F0079H	Peripheral I/O redirection register 1	PIOR1	R/W	—	✓	—	00H
F007AH	Peripheral enable register 1	PER1	R/W	✓	✓	—	00H
F007BH	Peripheral I/O redirection register 4	PIOR4	R/W	—	✓	—	00H
F007CH	Peripheral I/O redirection register 3	PIOR3	R/W	—	✓	—	00H
F007DH	Peripheral I/O redirection register 5	PIOR5	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
F00BEH	Flash sequencer frequency setting register	FSSET	R/W	—	✓	—	00H
F00C0H	Flash programming mode control register	FLPMC	R/W	—	✓	—	08H
F00C1H	Flash sequencer control register	FSSQ	R/W	—	✓	—	00H
F00C2H	Flash address pointer L	FLAPL	R/W	—	✓	—	00H
F00C3H	Flash address pointer H	FLAPH	R/W	—	✓	—	00H
F00C4H	Flash end address pointer L	FLSEDL	R/W	—	✓	—	00H
F00C5H	Flash end address pointer H	FLSEDH	R/W	—	✓	—	00H
F00C6H	Flash sequencer status register L	FSASTL	R	—	✓	—	00H
F00C7H	Flash sequence status register H	FSASTH	R	—	✓	—	00H

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F00C8H	Flash write buffer register LL	FLWLL		R/W	—	✓	—	00H
F00C9H	Flash write buffer register LH	FLWLH		R/W	—	✓	—	00H
F00CAH	Flash write buffer register HL	FLWHL		R/W	—	✓	—	00H
F00CBH	Flash write buffer register HH	FLWHH		R/W	—	✓	—	00H
F00F0H	Peripheral enable register 0	PER0		R/W	✓	✓	—	00H
F00F3H	Operation speed mode control register	OSMC		R/W	—	✓	—	00H
F00F5H	RAM parity error control register	RPECTL		R/W	✓	✓	—	00H
F00F9H	Internal reset status register by data retention power voltage	PORSR		R/W	—	✓	—	00H[Note 3]
F00FEH	BCD adjust 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								
F0112H	Serial mode register 01	SMR01		R/W	—	—	✓	0020H
F0113H								
F0114H	Serial mode register 02	SMR02		R/W	—	—	✓	0020H
F0115H								
F0016H	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		—			—	—		

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0122H	Serial channel start register 0	SS0L	SS0	R/W	✓	✓	✓	0000H
F0123H		—			—	—		
F0124H	Serial channel stop register 0	ST0L	ST0	R/W	✓	✓	✓	0000H
F0125H		—			—	—		
F0126H	Serial clock select register 0	SPS0L	SPS0	R/W	—	✓	✓	0000H
F0127H		—			—	—		
F0128H	Serial output register 0	SO0		R/W	—	—	✓	0303H
F0129H								
F012AH	Serial output enable register 0	SOE0L	SOE0	R/W	✓	✓	✓	0000H
F012BH		—			—	—		
F0134H	Serial output level register 0	SOL0L	SOL0	R/W	—	✓	✓	0000H
F0135H		—			—	—		
F0140H	Serial status register 10	SSR10L	SSR10	R	—	✓	✓	0000H
F0141H		—			—	—		
F0142H	Serial status register 11	SSR11L	SSR11	R	—	✓	✓	0000H
F0143H		—			—	—		
F0148H	Serial flag clear trigger register 10	SIR10L	SIR10	R/W	—	✓	✓	0000H
F0149H		—			—	—		
F014AH	Serial flag clear trigger register 11	SIR11L	SIR11	R/W	—	✓	✓	0000H
F014BH		—			—	—		
F0150H	Serial mode register 10	SMR10		R/W	—	—	✓	0020H
F0151H								
F0152H	Serial mode register 11	SMR11		R/W	—	—	✓	0020H
F0153H								
F0158H	Serial communication operation setting register 10	SCR10		R/W	—	—	✓	0087H
F0159H								
F015AH	Serial communication operation setting register 11	SCR11		R/W	—	—	✓	0087H
F015BH								
F0160H	Serial channel enable status register 1	SE1L	SE1	R	✓	✓	✓	0000H
F0161H		—			—	—		
F0162H	Serial channel start register 1	SS1L	SS1	R/W	✓	✓	✓	0000H
F0163H		—			—	—		
F0164H	Serial channel stop register 1	ST1L	ST1	R/W	✓	✓	✓	0000H
F0165H		—			—	—		
F0166H	Serial clock select register 1	SPS1L	SPS1	R/W	—	✓	✓	0000H
F0167H		—			—	—		
F0168H	Serial output register 1	SO1		R/W	—	—	✓	0303H
F0169H								
F016AH	Serial output enable register 1	SOE1L	SOE1	R/W	✓	✓	✓	0000H
F016BH		—			—	—		
F0174H	Serial output level register 1	SOL1L	SOL1	R/W	—	✓	✓	0000H
F0175H		—			—	—		
F0180H	Timer counter register 00	TCR00		R	—	—	✓	FFFFH
F0181H								

Table 3-6. Extended SFR (2nd SFR) List (4/6)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0182H	Timer counter register 01	TCR01		R	—	—	✓	FFFFH
F0183H								
F0184H	Timer counter register 02	TCR02		R	—	—	✓	FFFFH
F0185H								
F0186H	Timer counter register 03	TCR03		R	—	—	✓	FFFFH
F0187H								
F0188H	Timer counter register 04	TCR04		R	—	—	✓	FFFFH
F0189H								
F018AH	Timer counter register 05	TCR05		R	—	—	✓	FFFFH
F018BH								
F018CH	Timer counter register 06	TCR06		R	—	—	✓	FFFFH
F018DH								
F018EH	Timer counter register 07	TCR07		R	—	—	✓	FFFFH
F018FH								
F0190H	Timer mode register 00	TMR00		R/W	—	—	✓	0000H
F0191H								
F0192H	Timer mode register 01	TMR01		R/W	—	—	✓	0000H
F0193H								
F0194H	Timer mode register 02	TMR02		R/W	—	—	✓	0000H
F0195H								
F0196H	Timer mode register 03	TMR03		R/W	—	—	✓	0000H
F0197H								
F0198H	Timer mode register 04	TMR04		R/W	—	—	✓	0000H
F0199H								
F019AH	Timer mode register 05	TMR05		R/W	—	—	✓	0000H
F019BH								
F019CH	Timer mode register 06	TMR06		R/W	—	—	✓	0000H
F019DH								
F019EH	Timer mode register 07	TMR07		R/W	—	—	✓	0000H
F019FH								
F01A0H	Timer status register 00	TSR00L	TSR00	R	—	✓	✓	0000H
F01A1H		—			—	—		
F01A2H	Timer status register 01	TSR01L	TSR01	R	—	✓	✓	0000H
F01A3H		—			—	—		
F01A4H	Timer status register 02	TSR02L	TSR02	R	—	✓	✓	0000H
F01A5H		—			—	—		
F01A6H	Timer status register 03	TSR03L	TSR03	R	—	✓	✓	0000H
F01A7H		—			—	—		
F01A8H	Timer status register 04	TSR04L	TSR04	R	—	✓	✓	0000H
F01A9H		—			—	—		
F01AAH	Timer status register 05	TSR05L	TSR05	R	—	✓	✓	0000H
F01ABH		—			—	—		
F01ACH	Timer status register 06	TSR06L	TSR06	R	—	✓	✓	0000H
F01ADH		—			—	—		

Table 3-6. Extended SFR (2nd SFR) List (5/6)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F01AEH	Timer status register 07	TSR07L	TSR07	R	—	✓	✓	0000H
F01AFH		—			—	—		
F01B0H	Timer channel enable status register 0	TE0L	TE0	R	✓	✓	✓	0000H
F01B1H		—			—	—		
F01B2H	Timer channel start register 0	TS0L	TS0	R/W	✓	✓	✓	0000H
F01B3H		—			—	—		
F01B4H	Timer channel stop register 0	TT0L	TT0	R/W	✓	✓	✓	0000H
F01B5H		—			—	—		
F01B6H	Timer clock select register 0	TPS0		R/W	—	—	✓	0000H
F01B7H								
F01B8H	Timer output register 0	TO0L	TO0	R/W	—	✓	✓	0000H
F01B9H		—			—	—		
F01BAH	Timer output enable register 0	TOE0L	TOE0	R/W	✓	✓	✓	0000H
F01BBH		—			—	—		
F01BCH	Timer output level register 0	TOL0L	TOL0	R/W	—	✓	✓	0000H
F01BDH		—			—	—		
F01BEH	Timer output mode register 0	TOM0L	TOM0	R/W	—	✓	✓	0000H
F01BFH		—			—	—		
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
F02FAH	CRC data register	CRCD		R/W	—	—	✓	0000H
F02FBH								
F030AH	Touch pin function select register 0	TSSEL0		R/W	✓	✓	—	00H
F030BH	Touch pin function select register 1	TSSEL1		R/W	✓	✓	—	00H
F030DH	TSCAP pin setting register	VTSEL		R/W	✓	✓	—	00H
F0310H	Clock error correction register	SUBCUD		R/W	—	—	✓	0020H[Note 3]
F0380H	CTSU Control Register 0	CTSUCR0		R/W	✓	✓	—	00H
F0381H	CTSU Control Register 1	CTSUCR1		R/W	✓	✓	—	00H
F0382H	CTSU synchronous noise reduction setting register	CTSUSDPRS		R/W	✓	✓	—	00H
F0383H	CTSU sensor stabilization waiting time register	CTSUSST		R/W	—	✓	—	00H
F0384H	CTSU measurement channel register 0	CTSUMCH0		R/W	—	✓	—	1FH
F0385H	CTSU measurement channel register 1	CTSUMCH1		R	—	✓	—	1FH
F0386H	CTSU channel enable control register 0	CTSUCHAC0		R/W	✓	✓	—	00H
F0387H	CTSU channel enable control register 1	CTSUCHAC1		R/W	✓	✓	—	00H
F038BH	CTSU channel transmission/reception control register 0	CTSUCHTRC0		R/W	✓	✓	—	00H
F038CH	CTSU channel transmission/reception control register 1	CTSUCHTRC1		R/W	✓	✓	—	00H
F0390H	CTSU high-pass noise reduction control register	CTSUDCLKC		R/W	✓	✓	—	00H
F0391H	CTSU status register	CTSUST		R/W	✓	✓	—	00H

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0392H	CTSU high-pass noise reduction spectrum diffusion control register	CTSUSSC	R/W	—	—	✓	0000H
F0393H							
F0394H	CTSU sensor offset register 0	CTSUSO0	R/W	—	—	✓	0000H
F0395H							
F0396H	CTSU sensor offset register 1	CTSUSO1	R/W	—	—	✓	0000H
F0397H							
F0398H	CTSU sensor counter	CTSUSC	R	—	—	✓	0000H
F0399H							
F039AH	CTSU reference counter	CTSURC	R	—	—	✓	0000H
F039BH							
F039CH	CTSU error status register	CTSUERRS	R/W Note 4	—	—	✓	0000H
F039DH							
F0730H	CTSU trimming register	RTRIM	R/W	—	✓	—	Undefined Note 1
F0731H	CTSU trimming result register	CTSUTRESULT	R/W	—	✓	—	Undefined Note 1

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

Note 2. The value is determined by the FRQSEL2 to FRQSEL0 settings in option byte 000C2H.

Note 3. It is initialized only in case of an internal reset by data retention power voltage.

Note 4. Bit 15 (CTSUICOMP) is read-only.

Remark For SFRs in the SFR area, see **Table 3-5. SFR List (1/3)**.

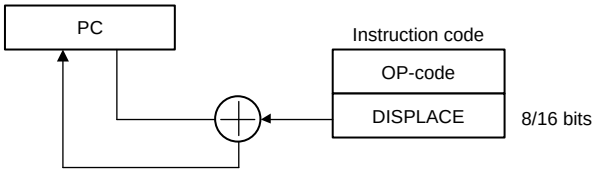
3.4 Instruction Address Addressing

3.4.1 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



3.4.2 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

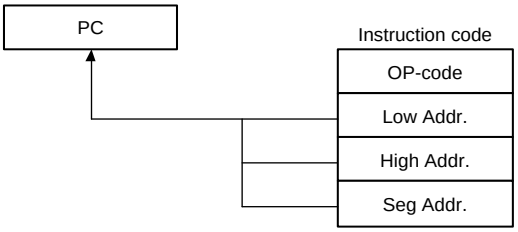
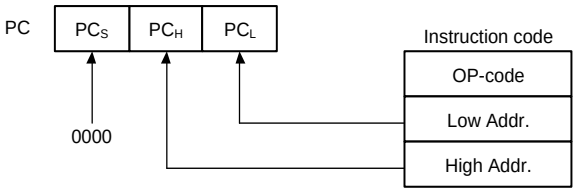


Figure 3-12. Example of CALL !addr16/BR !addr16



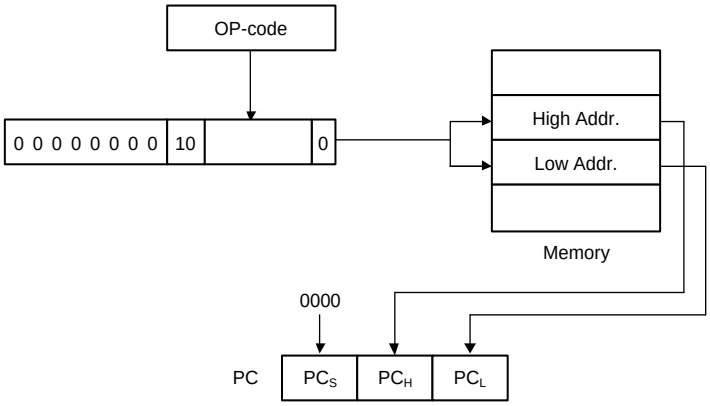
3.4.3 Table indirect addressing

[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

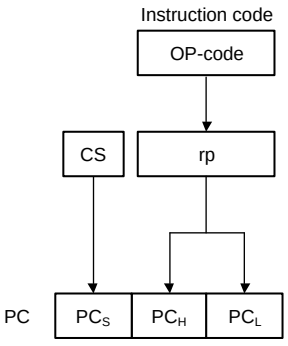


3.4.4 Register indirect addressing

[Function]

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

Figure 3-14. Outline of Register Indirect Addressing



3.5 Addressing for Processing Data Addresses

3.5.1 Implied addressing

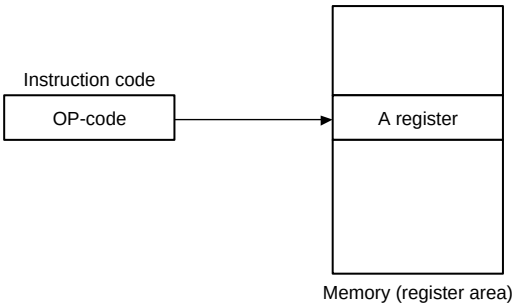
[Function]

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



3.5.2 Register addressing

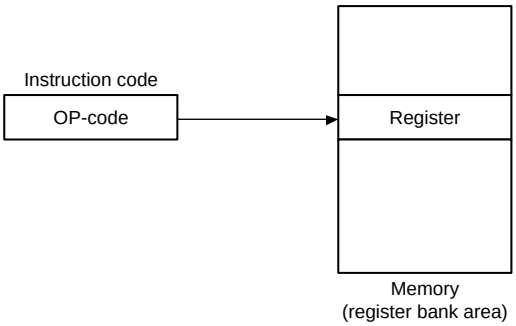
[Function]

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



3.5.3 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

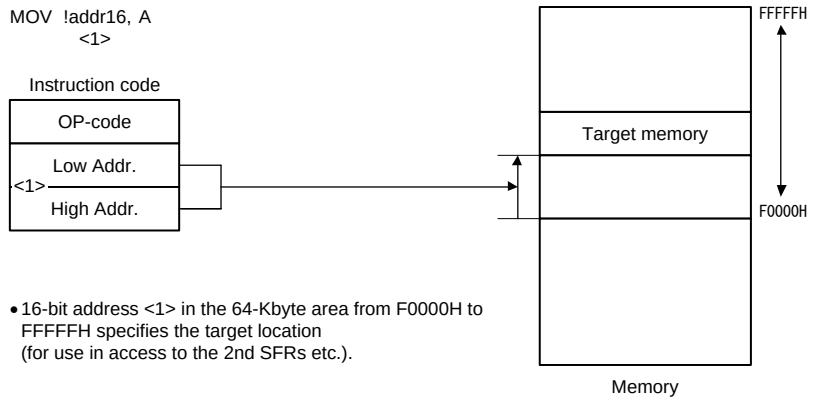
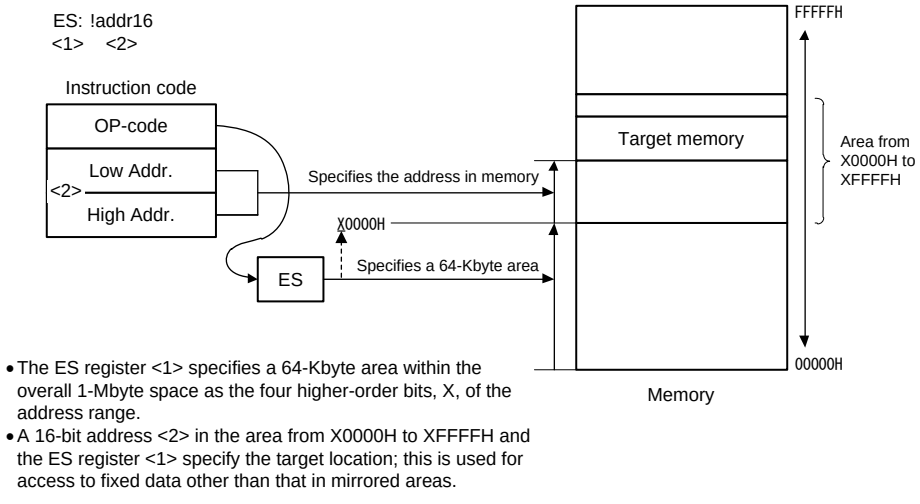


Figure 3-18. Example of ES:!addr16



3.5.4 Short direct addressing

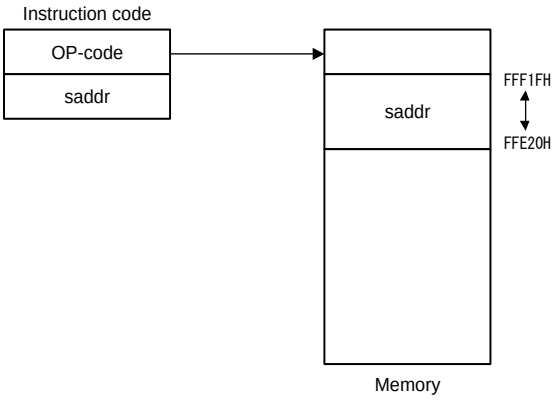
[Function]

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.

3.5.5 SFR addressing

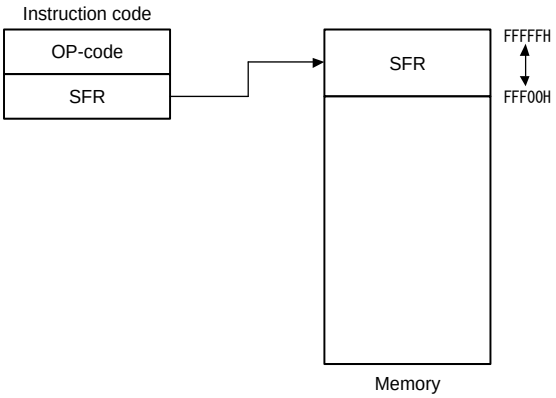
[Function]

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



3.5.6 Register indirect addressing

[Function]

Register indirect addressing 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]

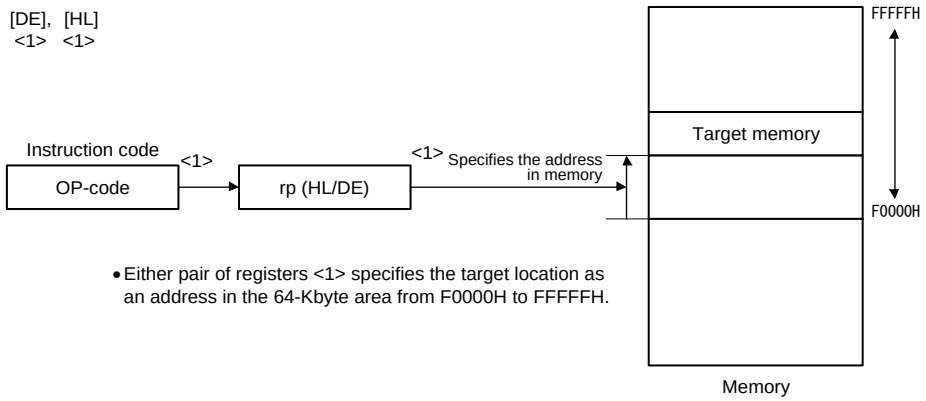
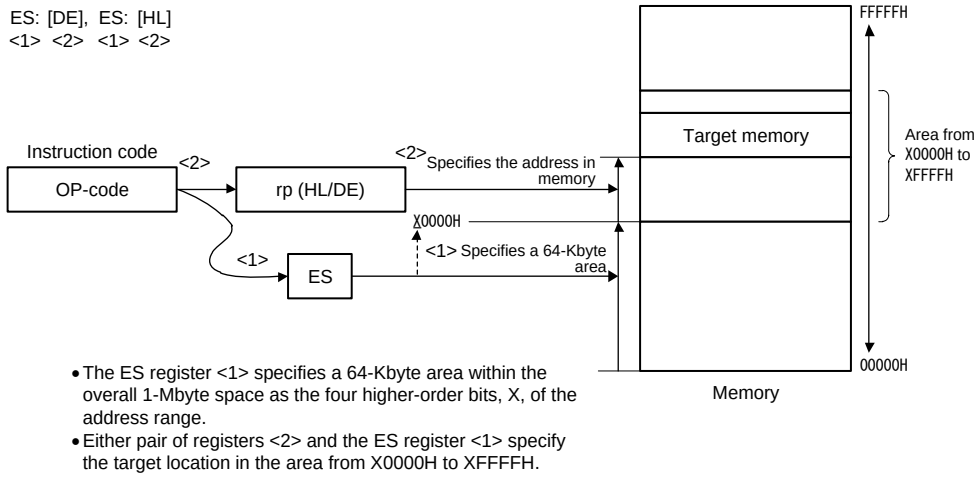


Figure 3-22. Example of ES:[DE], ES:[HL]



3.5.7 Based addressing

[Function]

Based addressing uses the contents of a register pair specified with the instruction word or 16-bit immediate data as the base address and specifies the target addresses using the result of adding the 8-bit immediate data or 16-bit immediate data as the offset data to the base 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]

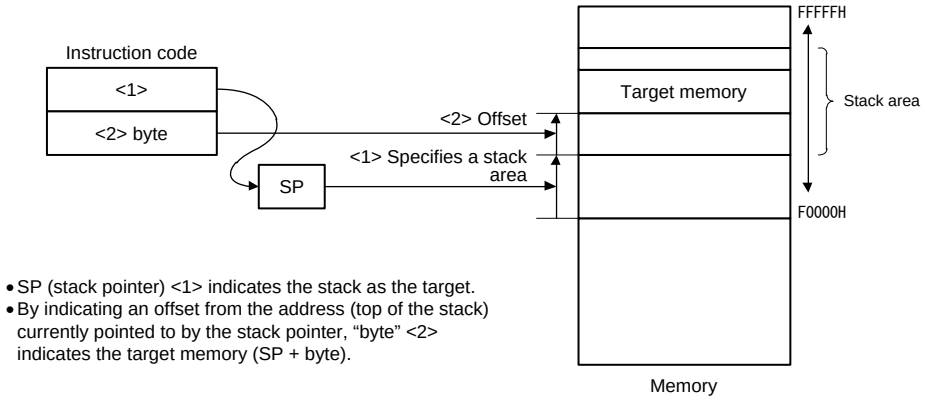


Figure 3-24. Example of [HL + byte], [DE + byte]

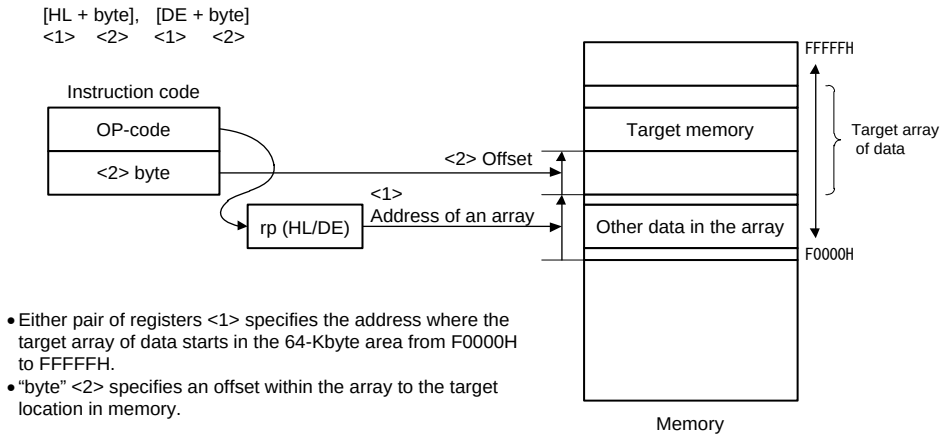


Figure 3-25. Example of word[B], word[C]

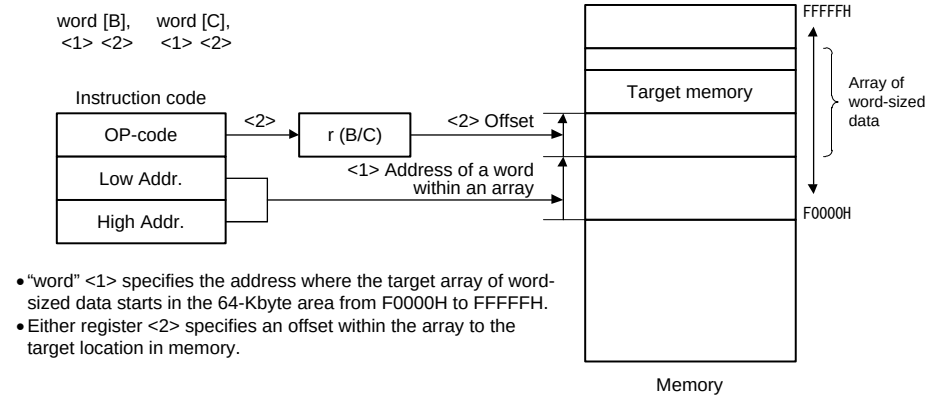


Figure 3-26. Example of word[BC]

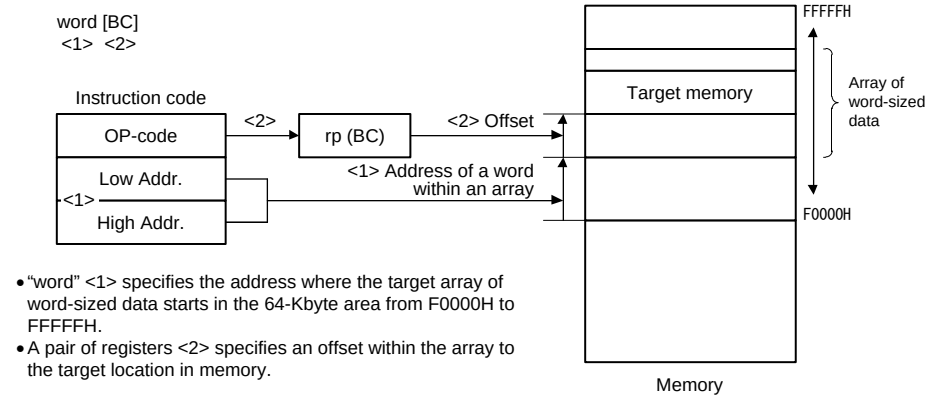
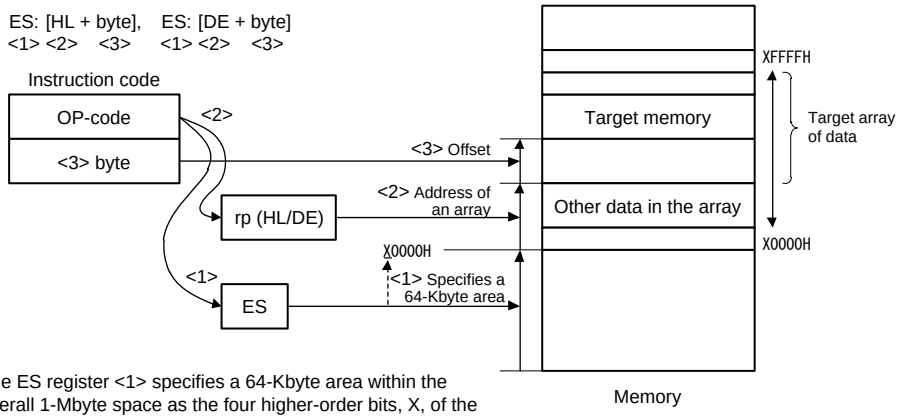
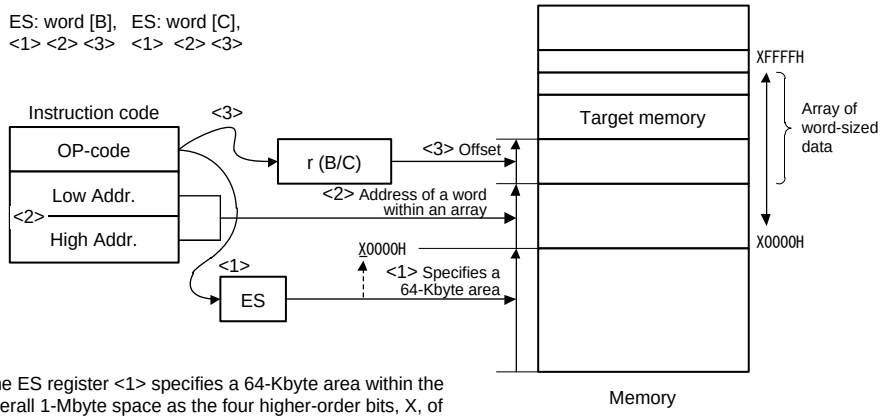


Figure 3-27. Example of ES:[HL + byte], ES:[DE + byte]



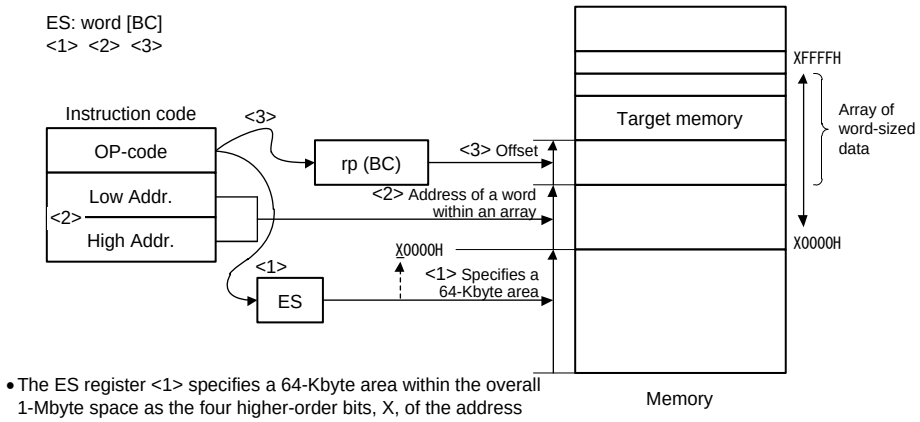
- The ES register <1> specifies a 64-Kbyte area within the overall 1-Mbyte space as the four higher-order bits, X, of the address range.
- Either pair of registers <2> specifies the address where the target array of data starts in the 64-Kbyte area specified in the ES register <1>.
- "byte" <3> specifies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

Figure 3-28. Example of ES:word[B], ES:word[C]



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

Figure 3-29. Example of ES:word[BC]



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

3.5.8 Based indexed addressing

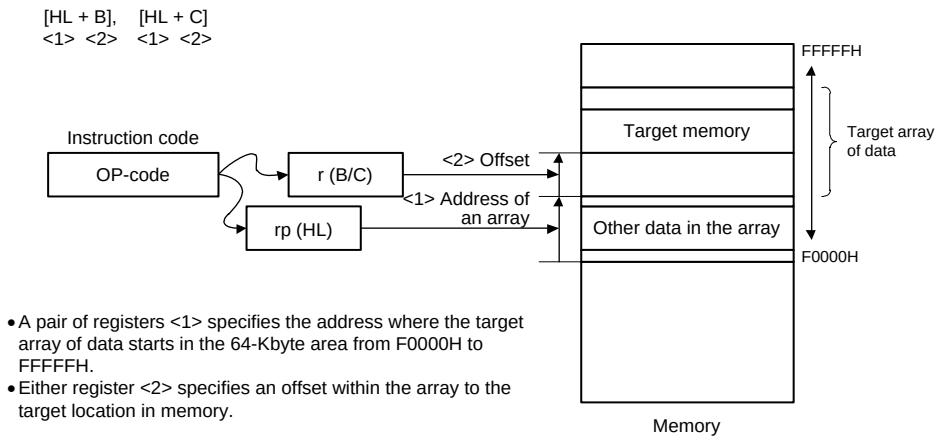
[Function]

Based indexed addressing uses the contents of a register pair specified with the instruction word as the base address and specifies the target addresses using the result of adding the contents of the B register or C register similarly specified with the instruction word as the offset address to the base address.

[Operand format]

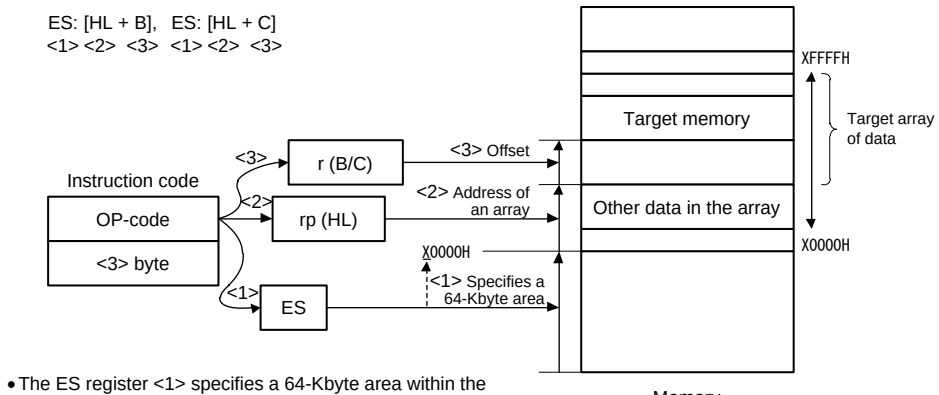
Identifier	Description
—	[HL + B], [HL + C] (only the space from F0000H to FFFFFH is specifiable)
—	ES:[HL + B], ES:[HL + C] (higher 4-bit addresses are specified by the ES register)

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.

3.5.9 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.

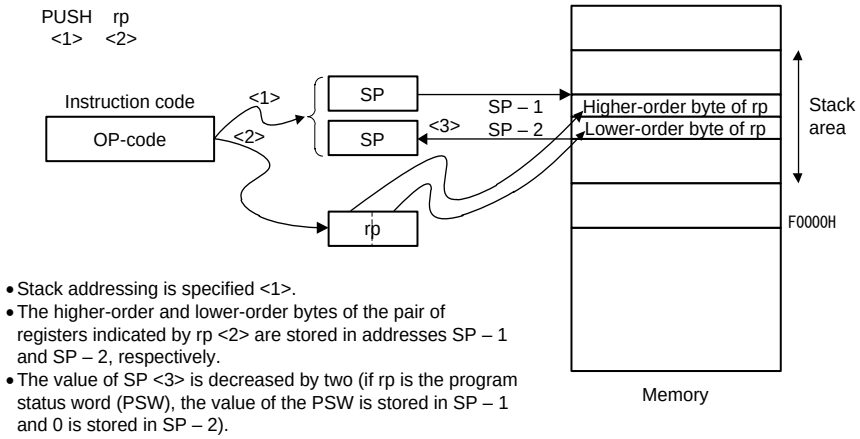
Only the internal RAM area can be set as the stack area.

[Description format]

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

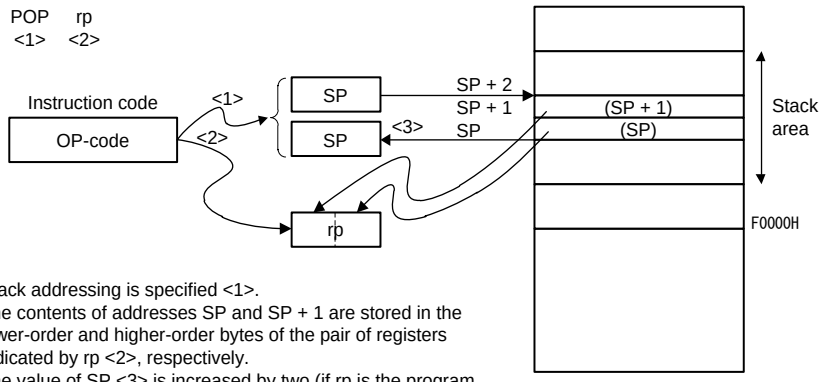
Each stack operation saves or restores data as shown in **Figure 3-32** to **Figure 3-37**.

Figure 3-32. Example of PUSH rp



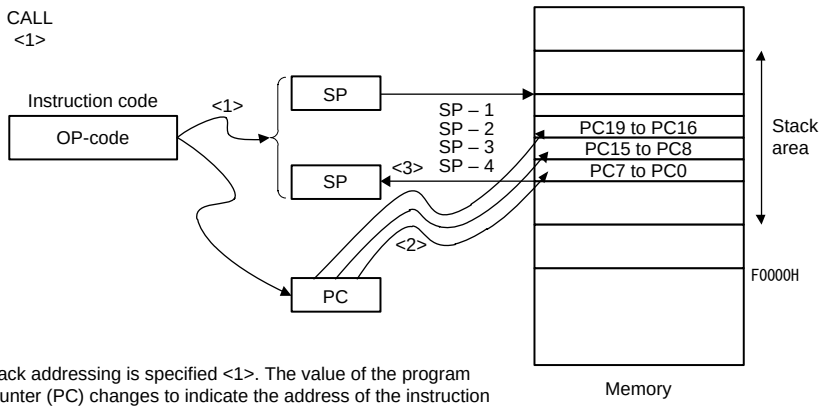
- Stack addressing is specified <1>.
- The higher-order and lower-order bytes of the pair of registers indicated by rp <2> are stored in addresses SP – 1 and SP – 2, respectively.
- The value of SP <3> is decreased by two (if rp is the program status word (PSW), the value of the PSW is stored in SP –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

Figure 3-33. Example of POP



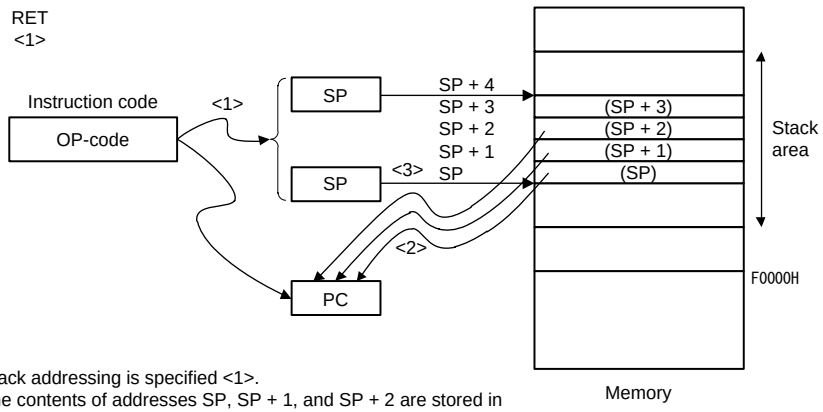
- Stack addressing is specified <1>.
- The contents of addresses SP and SP + 1 are stored in the lower-order and higher-order bytes of the pair of registers indicated by rp <2>, respectively.
- The value of SP <3> is increased by two (if rp is the program status word (PSW), the content of address SP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

Figure 3-34. Example of CALL, CALLT



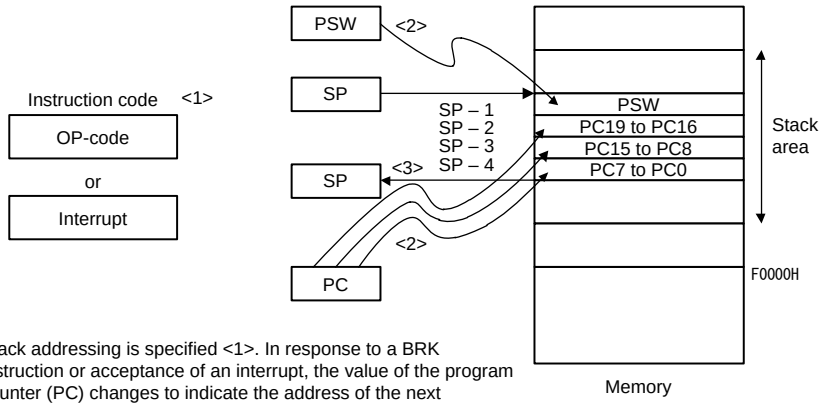
- Stack addressing is specified <1>. The value of the program counter (PC) changes to indicate the address of the instruction following the CALL instruction.
- The values of PC bits 19 to 16, 15 to 8, and 7 to 0 are stored in addresses SP – 2, SP - 3, and SP – 4, respectively <2>.
- The value of the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

Figure 3-35. Example of RET



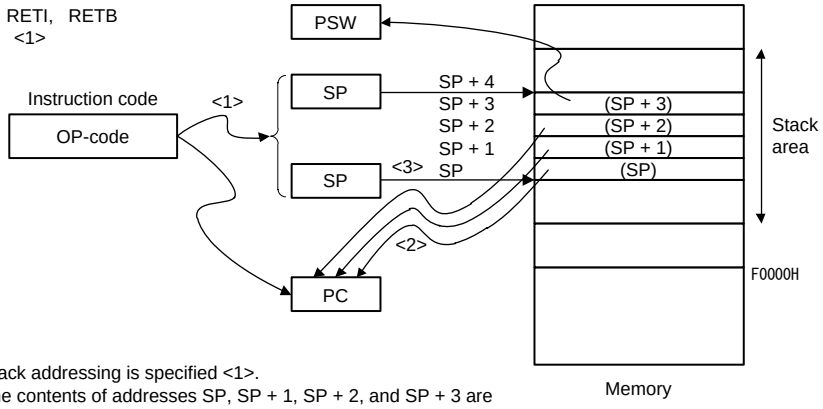
- Stack addressing is specified <1>.
- The contents of addresses SP, SP + 1, and SP + 2 are stored in PC bits 7 to 0, 15 to 8, and 19 to 16, respectively <2>.
- The value of SP <3> is increased by four.

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 4.

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The RL78 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 0, 1, 2, 4, 6, 12 (PM0, PM1, PM2, PM4, PM6, PM12) Port registers 0, 1, 2, 4, 6, 12, 13 (P0, P1, P2, P4, P6, P12, P13) Pull-up resistor option registers 0, 1, 2, 4, 12 (PU0, PU1, PU2, PU4, PU12) Port output mode registers 0, 1, 2, 4 (POM0, POM1, POM2, POM4) Port mode control registers 0, 2 (PMC0 ,PMC2) Peripheral I/O redirection registers 0 to 6 (PIOR0 to PIOR6) Touch pin function select registers (TSSEL0, TSSEL1) TSCAP pin setting register (VTSEL)
Port	• 10-pin products Total: 8 (CMOS I/O: 7 (N-ch open drain output (V_{DD} tolerance): 4), CMOS input: 1) • 16-pin products Total: 14 (CMOS I/O: 13 (N-ch open drain output (V_{DD} tolerance): 8), CMOS input: 1) • 20-pin products Total: 18 (CMOS I/O: 17 (N-ch open drain output (V_{DD} tolerance): 10), CMOS input: 1) • 24-pin products Total: 22 (CMOS I/O: 19 (N-ch open drain output (VDD tolerance): 10), CMOS input: 1, N-ch open drain output (6-V tolerance): 2) • 32-pin products Total: 30 (CMOS I/O: 27 (N-ch open drain output (VDD tolerance): 13), CMOS input: 1, N-ch open drain output (6-V tolerance): 2)
On-chip pull-up resistor	• 10-pin products Total: 7 • 16-pin products Total: 13 • 20-pin products Total: 17 • 24-pin products Total: 19 • 32-pin products Total: 27

4.2.1 Port 0

Port 0 is an I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00 to P07[Note 1] 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).

Output from the P00, P01, and P03 to P07 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 0 (POM0).

This port can also be used for real-time clock correction clock output, serial interface data I/O and clock I/O, UART transmission and reception for the external device connection used during flash memory programming, analog input, comparator output, clock/buzzer output, timer I/O, capacitance measurement pin, LPF connection pin, and external interrupt request input.

Reset signal generation sets P00 to input mode, and sets P01 to P07 to analog input mode.

Note 1. P00 to P04 for 10-pin products; P00 to P07 for 16-pin, 20-pin, 24-pin, and 32-pin products.

4.2.2 Port 1

Port 1 is an I/O port with an output latch. Port 1 can be set to the input mode or output mode in 1-bit units using port mode register 1 (PM1). When the P10 to P17[Note 1] pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 1 (PU1).

Output from the P14, P16, and P17 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 1 (POM1).

This port can also be used for real-time clock correction clock output, programming UART transmission/reception, serial interface data I/O and clock I/O, analog input, clock/buzzer output, timer I/O, capacitance measurement pin, and external interrupt request input.

Note 1. P10 and P11 for 24-pin products; P10 to P17 for 32-pin products.

4.2.3 Port 2

Port 2 is an I/O port with an output latch. Port 2 can be set to the input mode or output mode in 1-bit units using port mode register 2 (PM2). When the P20 to P23[Note 1] 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 2 (PU2).

Output from the P22 and P22 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 2 (POM2).

This port can also be used for serial interface data I/O and clock I/O, analog input, timer I/O, capacitance measurement pin, and external interrupt request input.

Reset signal generation sets P20 to P23 to analog input mode.

Note 1. For 20-pin, P24-pin, and P32-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 to P43[Note 1] pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 4 (PU4).

Output from the P41 pin can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 4 (POM4).

This port can also be used for real-time clock correction clock output, data I/O for a flash memory programmer/debugger, serial interface data I/O and clock I/O, comparator output, clock/buzzer output, timer I/O, capacitance measurement pin, and external interrupt request input. Reset signal generation sets port 4 to input mode.

Note 1. P40 for 10-pin products; P40 and P41 for 16-pin, 20-pin, and 24-pin products; P40 to P43 for 32-pin products.

4.2.5 Port 6

Port 6 is an I/O port with an output latch. Port 6 can be set to the input mode or output mode in 1-bit units using port mode register 6 (PM6).

Output from the P60 and P61 pins is N-ch open-drain output (6-V tolerance).

This port can also be used for serial interface data I/O, timer I/O, and external interrupt request input.

Note 1. P60 and P61 for 24-pin and 32-pin products.

4.2.6 Port 12

Port 12 is an I/O port with an output latch. Port 12 can be set to the input mode or output mode in 1-bit units using port mode register 12 (PM12). When the P121[Note 1], P122[Note 1], and P125 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 12 (PU12) (the on-chip pull-up resistor is always valid for P125 when $\overline{\text{RESET}}$ input is selected (PORTSELB = 1)).

This port can also be used for external interrupt request input, connecting resonator for main system clock, external clock input for main system clock, connecting resonator for subsystem clock, external clock input for subsystem clock, reset input, serial interface data input, comparator output, and timer I/O. Reset signal generation sets port 12 to input mode.

Note 1. For 16-pin, 20-pin, 24-pin, and 32-pin products.

Caution Once the power is turned on, P125 functions as the $\overline{\text{RESET}}$ input. The PORTSELB bit of the option byte (000C1H) defines whether this port operates as P125 or $\overline{\text{RESET}}$. When this pin is set to P125, do not input the low level to this pin during a reset by the selectable power-on-reset (SPOR) circuit and during the period from release from the reset by the SPOR circuit to the start of normal operation. If input of the low level continues during this period, the chip will remain in the reset state in response to the external reset. Accordingly, the on-chip pull-up resistor is enabled after power is turned on.

4.2.7 Port 13

P137 is an input-only port. This port can also be used for timer input and external interrupt request input.

4.3 Registers Controlling Port Function

Port functions are controlled by the following registers.

- Port mode registers 0, 1, 2, 4, 6, 12 (PM0, PM1, PM2, PM4, PM6, PM12)
- Port registers 0, 1, 2, 4, 6, 12, 13 (P0, P1, P2, P4, P6, P12, P13)
- Pull-up resistor option registers 0, 1, 2, 4, 12 (PU0, PU1, PU2, PU4, PU12)
- Port output mode registers 0, 1, 2, 4 (POM0, POM1, POM2, POM4)
- Port mode control registers 0, 2 (PMC0, PMC2)
- Peripheral I/O redirection registers 0 to 6 (PIOR0 to PIOR6)

Caution Which registers and bits are included depends on the product. For registers and bits mounted on each product, see Table 4-2 to Table 4-6. Be sure to set bits that are not mounted to their initial values.

Table 4-2. Pm, PMn, PUp, POMq, PMCr Registers and the Bits (10-pin Products)

Port		Bit name				
		Pm register	PMn register	PUp register	POMq register	PMCr register
PORT0	0	P00	PM00	PU00	POM00	—
	1	P01	PM01	PU01	POM01	PMC01
	2	P02	PM02	PU02	—	PMC02
	3	P03	PM03	PU03	POM03	PMC03
	4	P04	PM04	PU04	POM04	PMC04
PORT4	0	P40	PM40	PU40	—	—
PORT12	5	P125	PM125	PU125	—	—
PORT13	7	P137	—	—	—	—

Table 4-3. Pm, PMn, PUp, POMq, PMCr Registers and the Bits (16-pin Products)

Port		Bit name				
		Pm register	PMn register	PUp register	POMq register	PMCr register
PORT0	0	P00	PM00	PU00	POM00	—
	1	P01	PM01	PU01	POM01	PMC01
	2	P02	PM02	PU02	—	PMC02
	3	P03	PM03	PU03	POM03	PMC03
	4	P04	PM04	PU04	POM04	PMC04
	5	P05	PM05	PU05	POM05	PMC05
	6	P06	PM06	PU06	POM06	PMC06
	7	P07	PM07	PU07	POM07	PMC07
PORT4	0	P40	PM40	PU40	—	—
	1	P41	PM41	PU41	—	—
PORT12	1	P121	PM121	PU121	—	—
	2	P122	PM122	PU122	—	—
	5	P125	PM125	PU125	—	—
PORT13	7	P137	—	—	—	—

Table 4-4. Pm, PMn, PUp, POMq, PMCr Registers and the Bits (20-pin Products)

Port		Bit name				
		Pm register	PMn register	PUp register	POMq register	PMCr register
PORT0	0	P00	PM00	PU00	POM00	—
	1	P01	PM01	PU01	POM01	PMC01
	2	P02	PM02	PU02	—	PMC02
	3	P03	PM03	PU03	POM03	PMC03
	4	P04	PM04	PU04	POM04	PMC04
	5	P05	PM05	PU05	POM05	PMC05
	6	P06	PM06	PU06	POM06	PMC06
	7	P07	PM07	PU07	POM07	PMC07
PORT2	0	P20	PM20	PU20	POM20	PMC20
	1	P21	PM21	PU21	—	PMC21
	2	P22	PM22	PU22	POM22	PMC22
	3	P23	PM23	PU23	—	PMC23
PORT4	0	P40	PM40	PU40	—	—
	1	P41	PM41	PU41	POM41	—
PORT12	1	P121	PM121	PU121	—	—
	2	P122	PM122	PU122	—	—
	5	P125	PM125	PU125	—	—
PORT13	7	P137	—	—	—	—

Table 4-5. Pm, PMn, PUp, POMq, PMCr Registers and the Bits (24-pin Products)

Port		Bit name				
		Pm register	PMn register	PUp register	POMq register	PMCr register
PORT0	0	P00	PM00	PU00	POM00	—
	1	P01	PM01	PU01	POM01	PMC01
	2	P02	PM02	PU02	—	PMC02
	3	P03	PM03	PU03	POM03	PMC03
	4	P04	PM04	PU04	POM04	PMC04
	5	P05	PM05	PU05	POM05	PMC05
	6	P06	PM06	PU06	POM06	PMC06
	7	P07	PM07	PU07	POM07	PMC07
PORT1	0	P10	PM 10	PU10	—	—
	1	P11	PM11	PU11	—	—
PORT2	0	P20	PM20	PU20	POM20	PMC20
	1	P21	PM21	PU21	—	PMC21
	2	P22	PM22	PU22	POM22	PMC22
	3	P23	PM23	PU23	—	PMC23
PORT4	0	P40	PM40	PU40	—	—
	1	P41	PM41	PU41	POM41	—
PORT6	0	P60	PM60	—	—	—
	1	P61	PM61	—	—	—
PORT12	1	P121	PM121	PU121	—	—
	2	P122	PM122	PU122	—	—
	5	P125	PM125	PU125	—	—
PORT13	7	P137	—	—	—	—

Table 4-6. Pm, PMn, PUp, POMq, PMCr Registers and the Bits (32-pin Products)

Port		Bit name				
		Pm register	PMn register	PUp register	POMq register	PMCr register
PORT0	0	P00	PM00	PU00	POM00	—
	1	P01	PM01	PU01	POM01	PMC01
	2	P02	PM02	PU02	—	PMC02
	3	P03	PM03	PU03	POM03	PMC03
	4	P04	PM04	PU04	POM04	PMC04
	5	P05	PM05	PU05	POM05	PMC05
	6	P06	PM06	PU06	POM06	PMC06
	7	P07	PM07	PU07	POM07	PMC07
PORT1	0	P10	PM10	PU10	—	—
	1	P11	PM11	PU11	—	—
	2	P12	PM12	PU12	—	—
	3	P13	PM13	PU13	—	—
	4	P14	PM14	PU14	POM14	—
	5	P15	PM15	PU15	—	—
	6	P16	PM16	PU16	POM16	—
	7	P17	PM17	PU17	POM17	—
PORT2	0	P20	PM20	PU20	POM20	PMC20
	1	P21	PM21	PU21	—	PMC21
	2	P22	PM22	PU22	POM22	PMC22
	3	P23	PM23	PU23	—	PMC23
PORT4	0	P40	PM40	PU40	—	—
	1	P41	PM41	PU41	POM41	—
	2	P42	PM42	PU42	—	—
	3	P43	PM43	PU43	—	—
PORT6	0	P60	PM60	—	—	—
	1	P61	PM61	—	—	—
PORT12	1	P121	PM121	PU121	—	—
	2	P122	PM122	PU122	—	—
	5	P125	PM125	PU125	—	—
PORT13	7	P137	—	—	—	—

Remark m = 0, 1, 2, 4, 6, 12, 13
n = 0, 1, 2, 4, 6, 12
p = 0, 1, 2, 4, 12
q = 0, 1, 2, 4
r = 0, 2

The format of each register is described below.

4.3.1 Port mode registers 0, 1, 2, 4, 6, 12 (PM0, PM1, PM2, PM4, PM6, PM12)

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 Registers 0, 1, 2, 4, 6, 12 (PM0, PM1, PM2, PM4, PM6, PM12)

10-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W
PM4	1	1	1	1	1	1	1	PM40	FFF24H	FFH	R/W
PM12	1	1	PM125	1	1	1	1	1	FFF2CH	FFH	R/W

16-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W
PM4	1	1	1	1	1	1	PM41	PM40	FFF24H	FFH	R/W
PM12	1	1	PM125	1	1	PM122	PM121	1	FFF2CH	FFH	R/W

20-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W
PM2	1	1	1	1	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W
PM4	1	1	1	1	1	1	PM41	PM40	FFF24H	FFH	R/W
PM12	1	1	PM125	1	1	PM122	PM121	1	FFF2CH	FFH	R/W

24-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W
PM1	1	1	1	1	1	1	PM11	PM10	FFF21H	FFH	R/W
PM2	1	1	1	1	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W

PM4	1	1	1	1	1	1	PM41	PM40	FFF24H	FFH	R/W
PM6	1	1	1	1	1	1	PM61	PM60	FFF26H	FFH	R/W
PM12	1	1	PM125	1	1	PM122	PM121	1	FFF2CH	FFH	R/W

32-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FFF20H	FFH	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM2	1	1	1	1	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W
PM4	1	1	1	1	PM43	PM42	PM41	PM40	FFF24H	FFH	R/W
PM6	1	1	1	1	1	1	PM61	PM60	FFF26H	FFH	R/W
PM12	1	1	PM125	1	1	PM122	PM121	1	FFF2CH	FFH	R/W

PMmn	Pmn pin I/O mode selection
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Remark m = 0, 1, 2, 4, 6, 12
n = 0 to 7

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

4.3.2 Port registers 0, 1, 2, 4, 6, 12, 13 (P0, P1, P2, P4, P6, P12, P13)

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 1].

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

Reset signal generation sets the P13 register to the undefined value, and clears the other registers to 00H.

Note 1. When a pin that is set as an analog input pin (PMCx = 1, PMx = 1) is read, the value read is always 0 regardless of the input signal level on the pin.
When the data bit for P125 is read while the setting for the P125/$\overline{\text{RESET}}$ pin is $\overline{\text{RESET}}$ input (PORTSELB = 1), the value read is always 1.

Figure 4-2. Format of Port Registers 0, 1, 2, 4, 6, 12, 13 (P0, P1, P2, P4, P6, P12, P13)

10-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	0	P04	P03	P02	P01	P00	FFF00H	00H (output latch)	R/W
P4	0	0	0	0	0	0	0	P40	FFF04H	00H (output latch)	R/W
P12	0	0	P125	0	0	0	0	0	FFF0CH	00H (output latch)	R/W
P13	P137	0	0	0	0	0	0	0	FFF0DH	Undefined	R

16-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	P07	P06	P05	P04	P03	P02	P01	P00	FFF00H	00H (output latch)	R/W
P4	0	0	0	0	0	0	P41	P40	FFF04H	00H (output latch)	R/W
P12	0	0	P125	0	0	P122	P121	0	FFF0CH	00H (output latch)	R/W
P13	P137	0	0	0	0	0	0	0	FFF0DH	Undefined	R

20-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	P07	P06	P05	P04	P03	P02	P01	P00	FFF00H	00H (output latch)	R/W
P2	0	0	0	0	P23	P22	P21	P20	FFF02H	00H (output latch)	R/W
P4	0	0	0	0	0	0	P41	P40	FFF04H	00H (output latch)	R/W
P12	0	0	P125	0	0	P122	P121	0	FFF0CH	00H (output latch)	R/W

P13	P137	0	0	0	0	0	0	0	FFF0DH	Undefined	R

24-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	P07	P06	P05	P04	P03	P02	P01	P00	FFF00H	00H (output latch)	R/W
P1	0	0	0	0	0	0	P11	P10	FFF01H	00H (output latch)	R/W
P2	0	0	0	0	P23	P22	P21	P20	FFF02H	00H (output latch)	R/W
P4	0	0	0	0	0	0	P41	P40	FFF04H	00H (output latch)	R/W
P6	0	0	0	0	0	0	P61	P60	FFF06H	00H (output latch)	R/W
P12	0	0	P125	0	0	P122	P121	0	FFF0CH	00H (output latch)	R/W
P13	P137	0	0	0	0	0	0	0	FFF0DH	Undefined	R

32-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	P07	P06	P05	P04	P03	P02	P01	P00	FFF00H	00H (output latch)	R/W
P1	P17	P16	P15	P14	P13	P12	P11	P10	FFF01H	00H (output latch)	R/W
P2	0	0	0	0	P23	P22	P21	P20	FFF02H	00H (output latch)	R/W
P4	0	0	0	0	P43	P42	P41	P40	FFF04H	00H (output latch)	R/W
P6	0	0	0	0	0	0	P61	P60	FFF06H	00H (output latch)	R/W
P12	0	0	P125	0	0	P122	P121	0	FFF0CH	00H (output latch)	R/W
P13	P137	0	0	0	0	0	0	0	FFF0DH	Undefined	R

Pmn	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

Remark m = 0, 1, 2, 4, 6, 12, 13
n = 0 to 7

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

4.3.3 Pull-up resistor option registers 0, 1, 2, 4, 12 (PU0, PU1, PU2, PU4, PU12)

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 that satisfy the following usage conditions for the pins to which the use of an on-chip pull-up resistor has been specified in these registers.

Usage conditions of the on-chip pull-up resistor:

- PMmn = 1 (Input mode)
- PMCmn = 0 (Digital I/O)
- POMmn = 0 (Normal output mode)

On-chip pull-up resistors cannot be connected to bits set to output mode and bits used as alternate-function output pins and analog setting (PMC = 1), regardless of the settings of these registers.

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

Reset signal generation sets PU4 to 01H, PU12 to 20H, and clears PU0, PU1, and PU2 to 00H.

Figure 4-3. Format of Pull-up Resistor Option Registers 0, 1, 2, 4, 12 (PU0, PU1, PU2, PU4, PU12)

10-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	PU04	PU03	PU02	PU01	PU00	F0030H	00H	R/W
PU4	0	0	0	0	0	0	0	PU40	F0034H	01H	R/W
PU12	0	0	PU125 Note 1	0	0	0	0	0	F003CH	20H	R/W

16-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	PU07	PU06	PU05	PU04	PU03	PU02	PU01	PU00	F0030H	00H	R/W
PU4	0	0	0	0	0	0	PU41	PU40	F0034H	01H	R/W
PU12	0	0	PU125 Note 1	0	0	PU122	PU121	0	F003CH	20H	R/W

20-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	PU07	PU06	PU05	PU04	PU03	PU02	PU01	PU00	F0030H	00H	R/W
PU2	0	0	0	0	PU23	PU22	PU21	PU20	F0032H	00H	R/W
PU4	0	0	0	0	0	0	PU41	PU40	F0034H	01H	R/W

PU12	0	0	PU125 Note 1	0	0	PU122	PU121	0	F003CH	20H	R/W

24-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	PU07	PU06	PU05	PU04	PU03	PU02	PU01	PU00	F0030H	00H	R/W
PU1	0	0	0	0	0	0	PU11	PU10	F0031H	00H	R/W
PU2	0	0	0	0	PU23	PU22	PU21	PU20	F0032H	00H	R/W
PU4	0	0	0	0	0	0	PU41	PU40	F0034H	01H	R/W
PU12	0	0	PU125 Note 1	0	0	PU122	PU121	0	F003CH	20H	R/W

32-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	PU07	PU06	PU05	PU04	PU03	PU02	PU01	PU00	F0030H	00H	R/W
PU1	PU17	PU16	PU15	PU14	PU13	PU12	PU11	PU10	F0031H	00H	R/W
PU2	0	0	0	0	PU23	PU22	PU21	PU20	F0032H	00H	R/W
PU4	0	0	0	0	PU43	PU42	PU41	PU40	F0034H	01H	R/W
PU12	0	0	PU125 Note 1	0	0	PU122	PU121	0	F003CH	20H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

Note 1. This bit can be only manipulated when the P125 pin is selected (PORTSELB = 0) (the on-chip pull-up resistor is always valid (PU125 = 1) when the $\overline{\text{RESET}}$ input (PORTSELB = 1) is selected).

Remark m = 0, 1, 2, 4, 12
n = 0 to 7

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

4.3.4 Port input mode registers 0, 1, 2, 4 (POM0, POM1, POM2, POM4)

These registers set CMOS output or N-ch open drain output in 1-bit units.

N-ch open drain output (V_{DD} tolerance) mode can be selected for the SDA00 and SDA01 pins during simplified I^2C communication with an external device.

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

Reset signal generation clears these registers to 00H.

Caution An on-chip pull-up resistor is not connected to a bit for which N-ch open drain output (V_{DD} tolerance) mode (POMmn = 1) is set.

Figure 4-4. Format of Port Output Mode Registers 0, 1, 2, 4 (POM0, POM1, POM2, POM4)

10-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM0	0	0	0	POM04	POM03	0	POM01	POM00	F0050H	00H	R/W

16-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM0	POM07	POM06	POM05	POM04	POM03	0	POM01	POM00	F0050H	00H	R/W

20-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM0	POM07	POM06	POM05	POM04	POM03	0	POM01	POM00	F0050H	00H	R/W
POM2	0	0	0	0	0	POM22	0	POM20	F0052H	00H	R/W
POM4	0	0	0	0	0	0	POM41	0	F0054H	00H	R/W

24-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM0	POM07	POM06	POM05	POM04	POM03	0	POM01	POM00	F0050H	00H	R/W
POM2	0	0	0	0	0	POM22	0	POM20	F0052H	00H	R/W
POM4	0	0	0	0	0	0	POM41	0	F0054H	00H	R/W

32-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM0	POM07	POM06	POM05	POM04	POM03	0	POM01	POM00	F0050H	00H	R/W
POM1	POM17	POM16	0	POM14	0	0	0	0	F0051H	00H	R/W

POM2	0	0	0	0	0	POM22	0	POM20	F0052H	00H	R/W
POM4	0	0	0	0	0	0	POM41	0	F0054H	00H	R/W

POMmn	P0n pin output mode selection
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance) mode

Remark m = 0, 1, 2, 4
n = 0 to 7

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

4.3.5 Port mode control registers 0, 2 (PMC0, PMC2)

These registers set the 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 4-5. Format of Port Mode Control Registers 0, 2 (PMC0, PMC2)

10-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMC0	1	1	1	PMC04	PMC03	PMC02	PMC01	1	F0060H	FFH	R/W

16-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMC0	PMC07	PMC06	PMC05	PMC04	PMC03	PMC02	PMC01	1	F0060H	FFH	R/W

20-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMC0	PMC07	PMC06	PMC05	PMC04	PMC03	PMC02	PMC01	1	F0060H	FFH	R/W
PMC2	1	1	1	1	PMC23	PMC22	PMC21	PMC20	F0062H	FFH	R/W

24-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMC0	PMC07	PMC06	PMC05	PMC04	PMC03	PMC02	PMC01	1	F0060H	FFH	R/W
PMC2	1	1	1	1	PMC23	PMC22	PMC21	PMC20	F0062H	FFH	R/W

32-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMC0	PMC07	PMC06	PMC05	PMC04	PMC03	PMC02	PMC01	1	F0060H	FFH	R/W
PMC2	1	1	1	1	PMC23	PMC22	PMC21	PMC20	F0062H	FFH	R/W

PMCmn	P0n pin digital I/O/analog input selection
0	Digital I/O (alternate function other than analog input)
1	Analog input

Remark m = 0, 2
n = 1 to 7

Caution 1. Select input mode by using port mode registers 0, 2 (PM0, PM2) for the ports which are set by the PMC0, PMC2 registers as analog input.

Caution 2. Be sure to set bits that are not mounted to their initial values.

4.3.6 Peripheral I/O redirection registers 0 to 6 (PIOR0 to PIOR6)

These registers are 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.

Use the PIOR0 to PIOR6 registers to assign a port to the function to redirect and enable the function.

In addition, the settings for redirection can be changed only until operation of the function is enabled.

The PIOR0 to PIOR6 registers can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 4-6. Format of Peripheral I/O Redirection Registers 0 to 6 (PIOR0 to PIOR6)

10-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIOR0	0	0	0	PIOR04	PIOR03	PIOR02	0	PIOR00	F0077H	00H	R/W
PIOR3	0	0	0	0	0	PIOR32	0	0	F007CH	00H	R/W
PIOR4	0	0	0	0	0	0	0	PIOR40	F007BH	00H	R/W
PIOR6	0	0	0	0	PIOR63	0	0	PIOR60	F0076H	00H	R/W

16-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIOR0	0	PIOR06	PIOR05	PIOR04	PIOR03	PIOR02	0	PIOR00	F0077H	00H	R/W
PIOR1	PIOR17	0	PIOR15	0	0	0	PIOR11	PIOR10	F0079H	00H	R/W
PIOR2	0	0	0	0	0	PIOR22	PIOR21	PIOR20	F0075H	00H	R/W
PIOR3	0	0	0	0	0	PIOR32	0	0	F007CH	00H	R/W
PIOR4	PIOR47	PIOR46	PIOR45	PIOR44	0	PIOR42	0	PIOR40	F007BH	00H	R/W
PIOR5	0	0	0	PIOR54	0	PIOR52	PIOR51	PIOR50	F007DH	00H	R/W
PIOR6	0	PIOR66	PIOR65	PIOR64	PIOR63	0	PIOR61	PIOR60	F0076H	00H	R/W

20-pin products

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	0	PIOR13	0	PIOR11	PIOR10	F0079H	00H	R/W

PIOR2	0	0	0	0	PIOR23	PIOR22	PIOR21	PIOR20	F0075H	00H	R/W
PIOR3	0	0	0	0	0	PIOR32	0	PIOR30	F007CH	00H	R/W
PIOR4	PIOR47	PIOR46	PIOR45	PIOR44	0	PIOR42	PIOR41	PIOR40	F007BH	00H	R/W
PIOR5	0	0	PIOR55	PIOR54	PIOR53	PIOR52	PIOR51	PIOR50	F007DH	00H	R/W
PIOR6	0	PIOR66	PIOR65	PIOR64	PIOR63	0	PIOR61	PIOR60	F0076H	00H	R/W

24-pin products

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	0	PIOR13	PIOR12	PIOR11	PIOR10	F0079H	00H	R/W
PIOR2	0	0	0	PIOR24	PIOR23	PIOR22	PIOR21	PIOR20	F0075H	00H	R/W
PIOR3	0	0	0	0	PIOR33	PIOR32	PIOR31	PIOR30	F007CH	00H	R/W
PIOR4	PIOR47	PIOR46	PIOR45	PIOR44	0	PIOR42	PIOR41	PIOR40	F007BH	00H	R/W
PIOR5	0	0	PIOR55	PIOR54	PIOR53	PIOR52	PIOR51	PIOR50	F007DH	00H	R/W
PIOR6	PIOR67	PIOR66	PIOR65	PIOR64	PIOR63	PIOR62	PIOR61	PIOR60	F0076H	00H	R/W

32-pin products

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	PIOR26	PIOR25	PIOR24	PIOR23	PIOR22	PIOR21	PIOR20	F0075H	00H	R/W
PIOR3	0	0	0	0	PIOR33	PIOR32	PIOR31	PIOR30	F007CH	00H	R/W
PIOR4	PIOR47	PIOR46	PIOR45	PIOR44	PIOR43	PIOR42	PIOR41	PIOR40	F007BH	00H	R/W
PIOR5	PIOR57	PIOR56	PIOR55	PIOR54	PIOR53	PIOR52	PIOR51	PIOR50	F007DH	00H	R/W
PIOR6	PIOR67	PIOR66	PIOR65	PIOR64	PIOR63	PIOR62	PIOR61	PIOR60	F0076H	00H	R/W

10-pin products

PIOR0 Bit	Function	Setting value			
		0	1	2	3
PIOR04	TI02/TO02	P01	P00	—	—
PIOR03, PIOR02	TI01/TO01	P02	P40	P04	P01
PIOR00	TI00	P137	P03	—	—
	TO00	P03	P03	—	—

PIOR3 Bit	Function	Setting value	
		0	1
PIOR32	SCLA0	P03	P00
	SDAA0	P04	P01

PIOR4 Bit	Function	Setting value	
		0	1
PIOR40	INTP0	P137	P125

PIOR6 Bit	Function	Setting value	
		0	1
PIOR63	VCOUT0	P02	P125
PIOR60	PCLBUZ0	P02	P40

16-pin products

PIOR0 Bit	Function	Setting value							
		0	1	2	3	4	5	6	7
PIOR06 to PIOR04	TI02/TO02	P01	P00	P02	P05	P41	—	—	—
PIOR03, PIOR02	TI01/TO01	P02	P40	P04	P01	—	—	—	—
PIOR00	TI00	P137	P03	—	—	—	—	—	—

PIOR1 Bit	Function	Setting value			
		0	1	2	3
PIOR17	TI07/TO07	P121	P05	—	—
PIOR15	TI05/TO05	P122	P03	—	—
PIOR11, PIOR10	TI03	P41	P41	P06	—
	TO03	P41	P07	P06	—

PIOR2 Bit	Function	Setting value			
		0	1	2	3
PIOR22	SCK11	P07	P00	—	—
	SCL11	P07	P00	—	—
	SI11	P06	P01	—	—
	SDA11	P06	P01	—	—
	SO11	P05	P02	—	—
PIOR21, PIOR20	SCK00/SCL00	P02	P06	P05	—
	SI00/RxD0/SDA00	P01	P05	P04	—
	SO00/TxD0	P00	P04	P03	—

PIOR3 Bit	Function	Setting value	
		0	1
PIOR32	SCLA0	P06	P00
	SDAA0	P07	P01

PIOR4 Bit	Function	Setting value			
		0	1	2	3
PIOR47, PIOR46	INTP4	P03	P41	P121	—
PIOR45, PIOR44	INTP3	P04	P121	P41	—
PIOR42	INTP2	P40	P122	—	—
PIOR40	INTP0	P137	P125	—	—

PIOR5 Bit	Function	Setting value			
		0	1	2	3
PIOR54	INTP7	P02	P06	—	—
PIOR52	INTP6	P00	P05	—	—
PIOR51, PIOR50	INTP5	P01	P07	P121	—

PIOR6 Bit	Function	Setting value							
		0	1	2	3	4	5	6	7
PIOR66	RTC1HZ	P41	P00	—	—	—	—	—	—
PIOR65 to PIOR63	VCOUT1	P07	P07	P125	P07	P41	P125	P41	—
	VCOUT0	P02	P125	P02	P41	P02	P41	P125	—
PIOR61, PIOR60	PCLBUZ0	P02	P40	P06	—	—	—	—	—

20-pin products

PIOR0 Bit	Function	Setting value							
		0	1	2	3	4	5	6	7
PIOR06 to PIOR04	TI02/TO02	P01	P00	P02	P05	P41	—	—	—
PIOR03, PIOR02	TI01/TO01	P02	P40	P04	P01	—	—	—	—
PIOR01, PIOR00	TI00	P137	P03	P20	—	—	—	—	—
	TO00	P03	P03	P21	P20	—	—	—	—

PIOR1 Bit	Function	Setting value			
		0	1	2	3
PIOR17	TI07/TO07	P121	P05	—	—
PIOR16	TI06/TO06	P04	P22	—	—
PIOR15	TI05/TO05	P122	P03	—	—
PIOR13	TI04/TO04	P07	P23	—	—
PIOR11, PIOR10	TI03	P41	P41	P06	P20
	TO03	P41	P07	P06	P20

PIOR2 Bit	Function	Setting value			
		0	1	2	3
PIOR23, PIOR22	SCK11	P07	P00	P20	P07
	SCL11	P07	P00	P20	P23
	SI11	P06	P01	P125	P06
	SDA11	P06	P01	P41	P22
	SO11	P05	P02	P41	P05
PIOR21, PIOR20	SCK00/SCL00	P02	P06	P05	—
	SI00/RxD0/SDA00	P01	P05	P04	—
	SO00/TxD0	P00	P04	P03	—

PIOR3 Bit	Function	Setting value	
		0	1
PIOR32	SCLA0	P06	P00
	SDAA0	P07	P01
PIOR30	RxD1	P03	P21
	TxD1	P04	P20

PIOR4 Bit	Function	Setting value			
		0	1	2	3
PIOR47, PIOR46	INTP4	P03	P41	P121	—
PIOR45, PIOR44	INTP3	P04	P121	P41	—
PIOR42	INTP2	P40	P122	—	—
PIOR41	INTP1	P125	P20	—	—
PIOR40	INTP0	P137	P125	—	—

PIOR5 Bit	Function	Setting value			
		0	1	2	3
PIOR55, PIOR54	INTP7	P02	P06	P21	—
PIOR53, PIOR52	INTP6	P00	P05	P23	—
PIOR51, PIOR50	INTP5	P01	P07	P121	P22

PIOR6 Bit	Function	Setting value							
		0	1	2	3	4	5	6	7
PIOR66	RTC1HZ	P41	P00	—	—	—	—	—	—
PIOR65 to PIOR63	VCOUT1	P07	P07	P125	P07	P41	P125	P41	—
	VCOUT0	P02	P125	P02	P41	P02	P41	P125	—
PIOR61, PIOR60	PCLBUZ0	P02	P40	P06	—	—	—	—	—

24-pin products

PIOR0 Bit	Function	Setting value							
		0	1	2	3	4	5	6	7
PIOR06 to PIOR04	TI02/TO02	P01	P00	P02	P05	P41	—	—	—
PIOR03, PIOR02	TI01/TO01	P02	P40	P04	P01	—	—	—	—
PIOR01, PIOR00	TI00	P137	P03	P20	—	—	—	—	—
	TO00	P03	P03	P21	P20	—	—	—	—

PIOR1 Bit	Function	Setting value							
		0	1	2	3	4	5	6	7
PIOR17	TI07/TO07	P121	P05	—	—	—	—	—	—
PIOR16	TI06/TO06	P04	P22	—	—	—	—	—	—
PIOR15	TI05/TO05	P122	P03	—	—	—	—	—	—
PIOR13	TI04/TO04	P07	P23	—	—	—	—	—	—
PIOR12 to PIOR10	TI03	P41	P41	P06	P20	p10	p11	—	—
	TO03	P41	P07	P06	P20	p10	p11	—	—

PIOR2 Bit	Function	Setting value							
		0	1	2	3	4	5	6	7
PIOR24 to PIOR22	SCK11	P07	P00	P20	P07	P10	P11	—	—
	SCL11	P07	P00	P20	P23	P10	P11	—	—
	SI11	P06	P01	P125	P06	P60	P00	—	—
	SDA11	P06	P01	P41	P22	P61	P00	—	—
	SO11	P05	P02	P41	P05	P61	P01	—	—
PIOR21, PIOR20	SCK00/SCL00	P02	P06	P05	—	—	—	—	—
	SI00/RxD0/SDA00	P01	P05	P04	—	—	—	—	—
	SO00/TxD0	P00	P04	P03	—	—	—	—	—

PIOR3 Bit	Function	Setting value			
		0	1	2	3
PIOR33, PIOR32	SCLA0	P60	P06	P00	—
	SDAA0	P61	P07	P01	—
PIOR31, PIOR30	RxD1	P03	P21	P10	—
	TxD1	P04	P20	P11	—

PIOR4 Bit	Function	Setting value			
		0	1	2	3
PIOR47, PIOR46	INTP4	P03	P41	P121	—
PIOR45, PIOR44	INTP3	P04	P121	P41	P60
PIOR42	INTP2	P40	P122	—	—
PIOR41	INTP1	P125	P20	—	—
PIOR40	INTP0	P137	P125	—	—

PIOR5 Bit	Function	Setting value			
		0	1	2	3
PIOR55, PIOR54	INTP7	P02	P06	P21	P61
PIOR53, PIOR52	INTP6	P00	P05	P23	P60
PIOR51, PIOR50	INTP5	P01	P07	P121	P22

PIOR6 Bit	Function	Setting value							
		0	1	2	3	4	5	6	7
PIOR67, PIOR66	RTC1HZ	P41	P00	P11	—	—	—	—	—
PIOR65 to PIOR63	VCOUT1	P07	P07	P125	P07	P41	P125	P41	—
	VCOUT0	P02	P125	P02	P41	P02	P41	P125	—
PIOR62 to PIOR60	PCLBUZ0	P02	P40	P06	P10	P11	—	—	—

32-pin products

PIOR0 Bit	Function	Setting value							
		0	1	2	3	4	5	6	7
PIOR06 to PIOR04	TI02/TO02	P01	P00	P02	P05	P41	—	—	—
PIOR03, PIOR02	TI01/TO01	P02	P40	P04	P01	—	—	—	—
PIOR01 ,PIOR00	TI00	P137	P03	P20	P42	—	—	—	—
	TO00	P03	P03	P21	P20	—	—	—	—

PIOR1 Bit	Function	Setting value							
		0	1	2	3	4	5	6	7
PIOR17	TI07/TO07	P121	P05	—	—	—	—	—	—
PIOR16	TI06/TO06	P04	P22	—	—	—	—	—	—
PIOR15	TI05/TO05	P122	P03	—	—	—	—	—	—
PIOR14, PIOR13	TI04/TO04	P07	P23	P17	—	—	—	—	—
PIOR12 to PIOR10	TI03	P41	P41	P06	P20	p10	p11	P16	—
	TO03	P41	P07	P06	P20	p10	p11	P16	—

PIOR2 Bit	Function	Setting value							
		0	1	2	3	4	5	6	7
PIOR26, PIOR25	SCK20/SCL20	P41	P02	P13	—	—	—	—	—
	SI20/RxD2/SDA20	P20	P16	P14	—	—	—	—	—
	SO20/TxD2	P21	P17	P15	—	—	—	—	—
PIOR24 to PIOR22	SCK11	P07	P00	P20	P07	P10	P11	P13	—
	SCL11	P07	P00	P20	P23	P10	P11	P13	—
	SI11	P06	P01	P125	P06	P60	P00	P14	—
	SDA11	P06	P01	P41	P22	P61	P00	P14	—
	SO11	P05	P02	P41	P05	P61	P01	P15	—
PIOR21, PIOR20	SCK00/SCL00	P02	P06	P05	—	—	—	—	—
	SI00/RxD0/SDA00	P01	P05	P04	—	—	—	—	—
	SO00/TxD0	P00	P04	P03	—	—	—	—	—

PIOR3 Bit	Function	Setting value			
		0	1	2	3
PIOR33, PIOR32	SCLA0	P60	P06	P00	P16
	SDAA0	P61	P07	P01	P17
PIOR31, PIOR30	RxD1	P03	P21	P10	P20
	TxD1	P04	P20	P11	P42

PIOR4 Bit	Function	Setting value			
		0	1	2	3
PIOR47, PIOR46	INTP4	P03	P41	P121	P13
PIOR45, PIOR44	INTP3	P04	P121	P41	P60
PIOR43, PIOR42	INTP2	P40	P122	P13	P14
PIOR41	INTP1	P125	P20	—	—
PIOR40	INTP0	P137	P125	—	—

PIOR5 Bit	Function	Setting value			
		0	1	2	3
PIOR57	INTP9	P11	P15	—	—
PIOR56	INTP8	P10	P12	—	—
PIOR55, PIOR54	INTP7	P02	P06	P21	P61
PIOR53, PIOR52	INTP6	P00	P05	P23	P60
PIOR51, PIOR50	INTP5	P01	P07	P121	P22

PIOR6 Bit	Function	Setting value							
		0	1	2	3	4	5	6	7
PIOR67, PIOR66	RTC1HZ	P41	P00	P11	P13	—	—	—	—
PIOR65 to PIOR63	VCOUT1	P07	P07	P125	P07	P41	P125	P41	—
	VCOUT0	P02	P125	P02	P41	P02	P41	P125	—
PIOR62 to PIOR60	PCLBUZ0	P02	P40	P06	P10	P11	—	—	—

4.3.7 Touch pin function select registers 0, 1 (TSSEL0, TSSEL1)

These registers select whether the touch pin function or another multiplexed function is used with the P01, P03 to P07, P16, P17, P20 to P23, and P41 to P43 pins.

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

Reset signal generation clears these registers to 00H.

Figure 4-7. Format of Touch Pin Function Select Registers 0, 1 (TSSEL0, TSSEL1)

Address: F030AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TSSEL0	TSSEL07	TSSEL06	TSSEL05	TSSEL04	TSSEL03	TSSEL02	TSSEL01	TSSEL00

Address: F030BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TSSEL1	0	TSSEL14	TSSEL13	TSSEL12	TSSEL11	TSSEL10	TSSEL09	TSSEL08

TSSELxx (xx = 0 to 14)	Selection of a function other than the touch pin function (multiplexed function) or the touch pin function for the Pmn pin (m = 0, 1, 2, 4; n = 0 to 7)
0	Use the Pmn pin for a function other than the touch pin function (multiplexed function).
1	Use the Pmn pin for the touch pin function.

Remark To use the Pmn pins as touch pins (TSSELxx = 1), be sure to set the PUmn bit of the PUm register, POMmn bit of the POMm register, and PIMmn bit of the PIMm register to “0”.

- 10-pin products: Only TSSEL00, 03, and 04 are valid.
- 16-pin products: Only TSSEL00, 03, 04, 05, 06, 07, and 13 are valid.
- 20- and 24-pin products: TSSEL00, 03, 04, 05, 06, 07, 08, 09, 10, 11, and 13 are valid.

4.3.8 TSCAP pin setting register (VTSEL)

When the touch pin function is in use (when the TSSELxx bit is set to 1), the setting of the VTSEL register is effective. This register disables or enables input to the P02 pin.

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

Reset signal generation clears this register to 00H.

Figure 4-8. Format of TSCAP Pin Setting Register (VTSEL)

32-pin products

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
VTSEL	0	0	0	0	0	0	0	VTSEL0	F030DH	00H	R/W

VTSEL0	Disabling or enabling of input to the P02 pin
0	When the touch pin function is in use, input to the P02 pin is disabled.
1	When the touch pin function is in use, input to the P02 pin is enabled.

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. Therefore, byte data can be written to the ports used for both input and output.

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

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

4.4.2 Reading from I/O port

1) Output mode

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

2) Input mode

The pin 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.

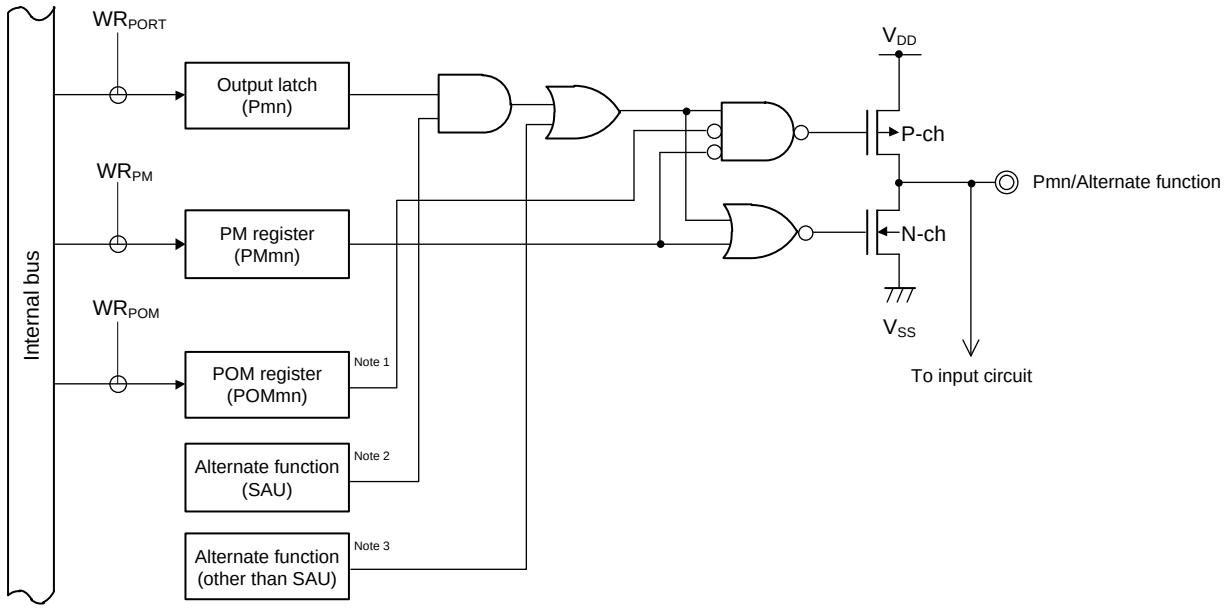
4.5 Register Settings When Using Alternate Function

4.5.1 Basic concept when using alternate function

In the beginning, for a pin also assigned to be used for analog input, use the port mode control registers 0, 2 (PMC0, PMC2) to specify whether to use the pin for analog input 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 (TAU, clock/buzzer output, IICA, etc.) is connected to the other input pin of the OR gate. When such kind of pins are used as 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 4-7**.

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).

Remark m: Port number (m = 0, 1, 2, 4, 6, 12, 13); n: Bit number (n = 0 to 7)

Table 4-7. 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: High (1)	Output: Low (0)
Output function for SAU	High (1)	—	Output: Low (0)
Output function for other than SAU	Low (0)	don't care	Output:Low (0)Note 1

Note 1. 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 the peripheral I/O redirection registers 0 to 6 (PIOR0 to PIOR6). 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 0 (SOE0) which corresponds to the unused output to 0 (output disabled) and set the SO0n bit in serial output register 0 (SO0) 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 (SE0n) in serial channel enable status register 0 (SE0) to 0 (operation stopped state), set the bit in serial output enable register 0 (SOE0) which corresponds to the unused output to 0 (output disabled), and set the SO0n and CKO0n bits in serial output register 0 (SO0) to 1 (high). These are the same settings as the initial state.

(3) TO0n = 0 (settings when the output of channel n in TAU is not used)

When the TO0n 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) SDAA0 = 0, SCLA0 = 0 (settings when IICA is not used)

When IICA is not used, set the IICE0 bit in IICA control register 00 (IICCTL00) to 0 (operation stopped). This is the same setting as the initial state.

(5) PCLBUZ0 = 0 (setting when clock/buzzer output is not used)

When the clock/buzzer output is not used, set the PCLOE0 bit in clock output select register 0 (CKS0) to 0 (output disabled). This is the same setting as the initial state.

4.5.3 Register setting examples for used port and alternate functions

Register setting examples for used port and alternate functions are shown in **Table 4-8**. The registers used to control the port functions should be set as shown in **Table 4-8**. See the following remark for legends used in **Table 4-8**.

Remark —: Not supported
×: don't care
PIORr: Peripheral I/O redirection register r (r = 0 to 6)
POMp: Port output mode register p (p = 0, 1, 2, 4)
PMCq: Port mode control register q (q = 0, 2)
PMn: Port mode register n (n = 0, 1, 2, 4, 6, 12)
Pm: Port output latch (m = 0, 1, 2, 4, 6, 12, 13)
TSSELt: Touch pin function select register t (t = 0, 1)
Functions in parentheses can be assigned via settings in the peripheral I/O redirection register 0 to 6 (PIOR0 to PIOR6).

Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function (1/14)

Pin Name	Used Function		PIORr	POMp	PMCq	PMn	Pm	TSSELt	Alternate Function Output		32-pin	24-pin	20-pin	16-pin	10-pin
	Function Name	I/O							SAU Output Function	Other than SAU					
P00	P00	Input	—	×	—	1	×	—	×	×	✓	✓	✓	✓	✓
		Output	—	0	—	0	0/1	—	TxD0/SO00 = 1 (SCK11/SCL11) = 1 Note 1 SDA11 = 1 Note 3	(TO02) = 0 (SCLA0) = 0 (RTC1HZ) = 0 Note 1					
		N-ch open drain output	—	1	—	0	0/1	—							
	SO00	Output	PIOR21 = 0 PIOR20 = 0	0/1	—	0	1	—	(SCK11/SCL11) = 1 Note 1 SDA11 = 1 Note 3	(TO02) = 0 (SCLA0) = 0 (RTC1HZ) = 0 Note 1	✓	✓	✓	✓	✓
	TxD0	Output		0/1	—	0	1	—			✓	✓	✓	✓	✓
	INTP6	Input	PIOR53 = 0 PIOR52 = 0	×	—	1	×	—	×	×	✓	✓	✓	✓	✓
	(TI02)	Input	PIOR06 = 0 PIOR05 = 0 PIOR04 = 1	×	—	1	×	—	×	×	✓	✓	✓	✓	✓
	(TO02)	Output		0	—	0	0	—	×	(SCLA0) = 0 (RTC1HZ) = 0 Note 1	✓	✓	✓	✓	✓
	(SI11)	Input	PIOR24 = 1 PIOR23 = 0 PIOR22 = 1	×	—	1	×	—	×	×	✓	✓	—	—	—
	(SDA11)	I/O		1	—	0	1	—	TxD0/SO00 = 1 (SCK11/SCL11) = 1 Note 1	(TO02) = 0 (SCLA0) = 0 (RTC1HZ) = 0 Note 1	✓	✓	—	—	—
	(SCK11)	Input	PIOR24 = 0 PIOR23 = 0 PIOR22 = 1	×	—	1	×	—	×	×	✓	✓	✓	✓	—
		Output		0/1	—	0	1	—	TxD0/SO00 = 1 SDA11 = 1 Note 3	(TO02) = 0 (SCLA0) = 0 (RTC1HZ) = 0 Note 1					
	(SCL11)	Output		0/1	—	0	1	—			✓	✓	✓	✓	—
<R>	(SCLA0)	I/O	PIOR32 = 1	1	—	0	0	—	TxD0/SO00 = 1 (SCK11/SCL11) = 1 Note 1 SDA11 = 1 Note 3	(TO02) = 0 (RTC1HZ) = 0 Note 1	—	—	✓	✓	✓
			PIOR33 = 1 PIOR32 = 0								✓	✓	—	—	—
	(RTC1HZ)	Output	PIOR67 = 0 PIOR66 = 1	0	—	0	0	—	×	(TO02) = 0 (SCLA0) = 0	✓	✓	✓	✓	—

Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function (2/14)

Pin Name	Used Function		PIORr	POMp	PMCq	PMn	Pm	TSSELt	Alternate Function Output		32-pin	24-pin	20-pin	16-pin	10-pin
	Function Name	I/O							SAU Output Function	Other than SAU					
P01	P01	Input	—	×	0	1	×	0	×	×	✓	✓	✓	✓	✓
		Output	—	0	0	0	0/1	0	SDA00 = 1 (SO11) = 1[Note 3] (SDA11) = 1[Note 1]	TO02 = 0 (TO01) = 0 (SDAA0) = 0					
		N-ch open drain output	—	1	0	0	0/1	0							
	ANI0	Analog input	—	×	1	1	×	0	×	×	✓	✓	✓	✓	✓
	TS00	I/O	×	×	×	1	0	1	×	×	✓	✓	✓	✓	✓
	SI00	Input	PIOR21 = 0 PIOR20 = 0	×	0	1	×	0	×	×	✓	✓	✓	✓	✓
	RxD0	Input		×	0	1	×	0	×	×	✓	✓	✓	✓	✓
	SDA00	I/O		1	0	0	1	0	(SO11) = 1[Note 3] (SDA11) = 1[Note 1]	TO02 = 0 (TO01) = 0 (SDAA0) = 0	✓	✓	✓	✓	✓
	INTP5	Input	PIOR51 = 0 PIOR50 = 0	×	0	1	×	0	×	×	✓	✓	✓	✓	✓
	(TI01)	Input	PIOR03 = 1 PIOR02 = 1	×	0	1	×	0	×	×	✓	✓	✓	✓	✓
	(TO01)	Output		0	0	0	0	0	×	TO02 = 0 (SDAA0) = 0	✓	✓	✓	✓	✓
	TI02	Input	PIOR06 = 0 PIOR05 = 0 PIOR04 = 0	×	0	1	×	0	×	×	✓	✓	✓	✓	✓
	TO02	Output		0	0	0	0	0	×	(TO01) = 0 (SDAA0) = 0	✓	✓	✓	✓	✓
	(SI11)	Input	PIOR24 = 0 PIOR23 = 0 PIOR22 = 1	×	0	1	×	0	×	×	✓	✓	✓	✓	—
	(SDA11)	I/O		1	0	0	1	0	SDA00 = 1 (SO11) = 1[Note 3]	TO02 = 0 (TO01) = 0 (SDAA0) = 0	✓	✓	✓	✓	—
<R>	(SDAA0)	I/O	PIOR32 = 1	1	0	0	0	0	×	TO02 = 0 (TO01) = 0	—	—	✓	✓	✓
			PIOR33 = 1 PIOR32 = 0								✓	✓	—	—	—
	(SO11)	I/O	PIOR24 = 1 PIOR23 = 0 PIOR22 = 1	0/1	0	0	1	0	SDA00 = 1 (SDA11) = 1[Note 1]	TO02 = 0 (TO01) = 0 (SDAA0) = 0	✓	✓	—	—	—

Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function (3/14)

Pin Name	Used Function		PIORr	POMp	PMCq	PMn	Pm	TSSELt	Alternate Function Output		32-pin	24-pin	20-pin	16-pin	10-pin
	Function Name	I/O							SAU Output Function	Other than SAU					
P02	P02	Input	—	—	0	1	×	—	×	×	✓	✓	✓	✓	✓
		Output	—	—	0	0	0/1	—	SCK00/SCL00 = 1 (SCK20/SCL20) = 1 Note 4 (SO11) = 1 Note 1	PCLBUZ0 = 0 VCOUT0 = 0 TO01 = 0 (TO02) = 0 Note 1					
	ANI1	Analog input	—	—	1	1	×	—	×	×	✓	✓	✓	✓	✓
	TSCAP Note 7	—	—	—	×	1	×	—	×	×	✓	✓	✓	✓	✓
	SCK00	Input	PIOR21 = 0 PIOR20 = 0	—	0	1	×	—	×	×	✓	✓	✓	✓	✓
		Output		—	0	0	1	—	(SCK20/SCL20) = 1 Note 4 (SO11) = 1 Note 1	PCLBUZ0 = 0 VCOUT0 = 0 TO01 = 0 (TO02) = 0 Note 1					
	SCL00	Output		—	0	0	1	—			✓	✓	✓	✓	✓
	(SCK20)	Input	PIOR26 = 0 PIOR25 = 1	—	0	1	×	—	×	×	✓	—	—	—	—
		Output		—	0	0	1	—	SCK00/SCL00 = 1 (SO11) = 1 Note 1	PCLBUZ0 = 0 VCOUT0 = 0 TO01 = 0 (TO02) = 0 Note 1					
	(SCL20)	Output		—	0	0	1	—			✓	—	—	—	—
	PCLBUZ0	Output	PIOR62 = 0 PIOR61 = 0 PIOR60 = 0	—	0	0	0	—	×	VCOUT0 = 0 TO01 = 0 (TO02) = 0 Note 1	✓	✓	✓	✓	✓
	VCOUT0	Output	PIOR65 = 0 PIOR64 = 0 PIOR63 = 0	—	0	0	0	—	×	PCLBUZ0 = 0 TO01 = 0 (TO02) = 0 Note 1	✓	✓	✓	✓	✓
	INTP7	Input	PIOR55 = 0 PIOR54 = 0	—	0	1	×	—	×	×	✓	✓	✓	✓	✓
	(TI02)	Input	PIOR06 = 0 PIOR05 = 1 PIOR04 = 0	—	0	1	×	—	×	×	✓	✓	✓	✓	—
	(TO02)	Output		—	0	0	0	—	×	PCLBUZ0 = 0 VCOUT0 = 0 TO01 = 0	✓	✓	✓	✓	—
	TI01	Input	PIOR03 = 0 PIOR02 = 0	—	0	1	×	—	×	×	✓	✓	✓	✓	✓
	TO01	Output		—	0	0	0	—	×	PCLBUZ0 = 0 VCOUT0 = 0 (TO02) = 0 Note 1	✓	✓	✓	✓	✓
	(SO11)	Output	PIOR24 = 0 PIOR23 = 0 PIOR22 = 1	—	0	0	1	—	SCK00/SCL00 = 1 (SCK20/SCL20) = 1 Note 4	PCLBUZ0 = 0 VCOUT0 = 0 TO01 = 0 (TO02) = 0 Note 1	✓	✓	✓	✓	—

Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function (4/14)

Pin Name	Used Function		PIORr	POMp	PMCq	PMn	Pm	TSSELt	Alternate Function Output		32-pin	24-pin	20-pin	16-pin	10-pin
	Function Name	I/O							SAU Output Function	Other than SAU					
P03	P03	Input	—	×	0	1	×	0	×	×	✓	✓	✓	✓	✓
		Output	—	×	0	0	0/1	0	(SO00/TxD0) = 1Note 1	TO00 = 0 (TO05) = 0Note 1 SCLA0 = 0Note 5					
		N-ch open drain output	—	1	0	0	0/1	0							
	ANI2	Analog input	—	×	1	1	×	0	×	×	✓	✓	✓	✓	✓
	TS03	I/O	×	×	×	1	0	1	×	×	✓	✓	✓	✓	✓
	TO00	Output	PIOR01 = 0 PIOR00 = 0	×	0	0	0	0	×	(TO05) = 0Note 1 SCLA0 = 0Note 5	✓	✓	✓	✓	✓
	INTP4	Input	PIOR47 = 0 PIOR46 = 0	×	0	1	×	0	×	×	✓	✓	✓	✓	✓
	IVCMP0	Input	—	×	1	1	×	0	×	×	✓	✓	✓	✓	✓
	(TI00)	Input	PIOR01 = 0 PIOR00 = 1	×	0	1	×	0	×	×	✓	✓	✓	✓	✓
	(TI05)	Input	PIOR15 = 1	×	0	1	×	0	×	×	✓	✓	✓	✓	—
	(TO05)	Output	PIOR15 = 1	×	0	0	0	0	×	TO00 = 0 SCLA0 = 0Note 5	✓	✓	✓	✓	—
	(SO00)	Output	PIOR21 = 1 PIOR20 = 0	0/1	0	0	1	0	×	TO00 = 0 TO05 = 0Note 1 SCLA0 = 0Note 5	✓	✓	✓	✓	—
	(TxD0)	Output		0/1	0	0	1	0	×		✓	✓	✓	✓	—
	RxD1	Input	PIOR31 = 0 PIOR30 = 0	×	0	1	×	0	×	×	✓	✓	✓	✓	—
	SCLA0	I/O	PIOR32 = 0	1	0	0	0	0	×	TO00 = 0 (TO05) = 0Note 1	—	—	—	—	✓
P04	P04	Input	—	×	0	1	×	0	×	×	✓	✓	✓	✓	✓
		Output	—	×	0	0	0/1	0	(SO00/TxD0) = 1Note 1 TxD1 = 1Note 1 (SDA00) = 1Note 1	TO06 = 0Note 1 (TO01) = 0 SDAA0 = 0Note 5					
		N-ch open drain output	—	1	0	0	0/1	0							
	ANI3	Analog input	—	×	1	1	×	0	×	×	✓	✓	✓	✓	✓
	TS04	I/O	×	×	×	1	0	1	×	×	✓	✓	✓	✓	✓
	(TI01)	Input	PIOR03 = 1 PIOR02 = 0	×	0	1	×	0	×	×	✓	✓	✓	✓	✓
	(TO01)	Output		0	0	0	0	0	×	TO06 = 0Note 1 SDAA0 = 0Note 5	✓	✓	✓	✓	✓
	TI06	Input	PIOR16 = 0	×	0	1	×	0	×	×	✓	✓	✓	✓	—
	TO06	Output		0	0	0	0	0	×	(TO01) = 0 SDAA0 = 0Note 5	✓	✓	✓	✓	—
	IVREF0	Input	—	×	1	1	×	0	×	×	✓	✓	✓	✓	✓
	INTP3	Input	PIOR45 = 0 PIOR44 = 0	×	0	1	×	0	×	×	✓	✓	✓	✓	✓
	(SI00)	Input	PIOR21 = 1 PIOR20 = 0	×	0	1	×	0	×	×	✓	✓	✓	✓	—
	(RxD0)	Input		×	0	1	×	0	×	×	✓	✓	✓	✓	—
	(SDA00)	I/O		1	0	0	1	0	(SO00/TxD0) = 1Note 1 TxD1 = 1Note 1	TO06 = 0Note 1 (TO01) = 0 SDAA0 = 0Note 5	✓	✓	✓	✓	—
	(SO00)	Output	PIOR21 = 0 PIOR20 = 1	0/1	0	0	1	0	(SO00/TxD0) = 1Note 1 TxD1 = 1Note 1 (SDA00) = 1Note 1	TO06 = 0Note 1 (TO01) = 0 SDAA0 = 0Note 5	✓	✓	✓	✓	—
	(TxD0)	Output		0/1	0	0	1	0			✓	✓	✓	✓	—
	TxD1	Output	PIOR31 = 0 PIOR30 = 0	0/1	0	0	1	0	(SO00/TxD0) = 1Note 1 (SDA00) = 1Note 1	TO06 = 0Note 1 (TO01) = 0 SDAA0 = 0Note 5	✓	✓	✓	✓	—
	SDAA0	I/O	PIOR32 = 0	1	0	0	0	0	×	TO06 = 0Note 1 (TO01) = 0	—	—	—	—	✓

Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function (5/14)

Pin Name	Used Function		PIORr	POMp	PMCq	PMn	Pm	TSSELt	Alternate Function Output		32-pin	24-pin	20-pin	16-pin	10-pin
	Function Name	I/O							SAU Output Function	Other than SAU					
P05	P05	Input	—	×	0	1	×	0	×	×	✓	✓	✓	✓	—
		Output	—	0	0	0	0/1	0	SO11 = 1 (SCK00/SCL00) = 1 (SDA00) = 1	(TO02) = 0 (TO07) = 0					
		N-ch open drain output	—	1	0	0	0/1	0							
	ANI4	Analog input	—	×	1	1	×	0	×	×	✓	✓	✓	✓	—
	TS05	I/O	×	×	×	1	0	1	×	×	✓	✓	✓	✓	—
	IVCMP1	Input	—	×	1	1	×	0	×	×	✓	✓	✓	✓	—
	(TI02)	Input	PIOR06 = 0 PIOR05 = 1 PIOR04 = 1	×	0	1	×	0	×	×	✓	✓	✓	✓	—
	(TO02)	Output		0	0	0	0	0	×	(TO07) = 0	✓	✓	✓	✓	—
	(TI07)	Input	PIOR17 = 1	×	0	1	×	0	×	×	✓	✓	✓	✓	—
	(TO07)	Output		0	0	0	0	0	×	(TO02) = 0	✓	✓	✓	✓	—
	SO11	Output	PIOR24 = 0 PIOR23 = 0 PIOR22 = 0	0/1	0	0	1	0	(SCK00/SCL00) = 1 (SDA00) = 1	(TO02) = 0 (TO07) = 0	✓	✓	✓	✓	—
	(INTP6)	Input	PIOR53 = 0 PIOR52 = 1	×	0	1	×	0	×	×	✓	✓	✓	✓	—
	(SCK00)	Input	PIOR21 = 1 PIOR20 = 0	×	0	1	×	0	×	×	✓	✓	✓	✓	—
		Output		0/1	0	0	1	0	SO11 = 1 (SDA00) = 1	(TO02) = 0 (TO07) = 0					
	(SCL00)	Output		0/1	0	0	1	0			✓	✓	✓	✓	—
	(SI00)	Input	PIOR21 = 0 PIOR20 = 1	×	0	1	×	0	×	×	✓	✓	✓	✓	—
	(RxD0)	Input		×	0	1	×	0	×	×	✓	✓	✓	✓	—
	(SDA00)	I/O		1	0	0	1	0	×	×	✓	✓	✓	✓	—
P06	P06	Input	—	×	0	1	×	0	×	×	✓	✓	✓	✓	—
		Output	—	0	0	0	0/1	0	SDA11 = 1 (SCK00/SCL00) = 1	SCLA0 = 0[Note 6] (SCLA0) = 0[Note 3] (TO03) = 0 (PCLBUZ) = 0					
		N-ch open drain output	—	1	0	0	0/1	0							
	ANI5	Analog input	—	×	1	1	×	0	×	×	✓	✓	✓	✓	—
	TS06	I/O	×	×	×	1	0	1	×	×	✓	✓	✓	✓	—
	IVREF1	Input	—	×	1	1	×	0	×	×	✓	✓	✓	✓	—
	SI11	Input	PIOR24 = 0 PIOR23 = 0 PIOR22 = 0	×	0	1	×	0	×	×	✓	✓	✓	✓	—
	SDA11	I/O	PIOR24 = 0 PIOR23 = 0 PIOR22 = 0	1	0	0	1	0	(SCK00/SCL00) = 1	SCLA0 = 0[Note 6] (SCLA0) = 0[Note 3] (TO03) = 0 (PCLBUZ) = 0	✓	✓	✓	✓	—
	SCLA0	I/O	PIOR32 = 0	1	0	0	0	0	×	(TO03) = 0 (PCLBUZ) = 0	—	—	✓	✓	—
	(SCLA0)	I/O	PIOR33 = 0 PIOR32 = 1	1	0	0	0	0	×	(TO03) = 0 (PCLBUZ) = 0	✓	✓	—	—	—
	(INTP7)	Input	PIOR55 = 0 PIOR54 = 1	×	0	1	×	0	×	×	✓	✓	✓	✓	—
	(TI03)	Input	PIOR12 = 0 PIOR11 = 1 PIOR10 = 0	×	0	1	×	0	×	×	✓	✓	✓	✓	—
	(TO03)	Output		0	0	0	0	0	×	×	✓	✓	✓	✓	—
	(PCLBUZ0)	Output	PIOR62 = 0 PIOR61 = 1 PIOR60 = 0	0	0	0	0	0	×	SCLA0 = 0[Note 6] (SCLA0) = 0[Note 3] (TO03) = 0	✓	✓	✓	✓	—
	(SCK00)	Input	PIOR21 = 0 PIOR20 = 1	×	0	1	×	0	×	×	✓	✓	✓	✓	—
		Output		0/1	0	0	1	0	SDA11 = 1	SCLA0 = 0[Note 6] (SCLA0) = 0[Note 3] (TO03) = 0 (PCLBUZ) = 0	✓	✓	✓	✓	—
	(SCL00)	Output		0/1	0	0	1	0			✓	✓	✓	✓	—

Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function (6/14)

Pin Name	Used Function		PIORr	POMp	PMCq	PMn	Pm	TSSELt	Alternate Function Output		32-pin	24-pin	20-pin	16-pin	10-pin
	Function Name	I/O							SAU Output Function	Other than SAU					
P07	P07	Input	—	×	0	1	×	0	×	×	✓	✓	✓	✓	—
		Output	—	0	0	0	0/1	0	SCK11/SCL11 = 1	VCOUT1 = 0 (TO03) = 0 TO04 = 0 SDAA0 = 0Note 6 (SDAA0) = 0Note 3					
		N-ch open drain output	—	1	0	0	0/1	0							
	ANI6	Analog input	—	×	1	1	×	0	×	×	✓	✓	✓	✓	—
	TS07	I/O	×	×	×	1	0	1	×	×	✓	✓	✓	✓	—
	TI04	Input	PIOR14 = 0 PIOR13 = 0	×	0	1	×	0	×	×	✓	✓	✓	✓	—
	TO04	Output		0	0	0	0	0	×	VCOUT1 = 0 (TO03) = 0 SDAA0 = 0Note 6 (SDAA0) = 0Note 3	✓	✓	✓	✓	—
	(TO03)	Output	PIOR12 = 0 PIOR11 = 0 PIOR10 = 1	0	0	0	0	0	×	VCOUT1 = 0 TO04 = 0 SDAA0 = 0Note 6 (SDAA0) = 0Note 3	✓	✓	✓	✓	—
	SCK11	Input	PIOR24 = 0 PIOR23 = 0 PIOR22 = 0	×	0	1	×	0	×	×	✓	✓	✓	✓	—
		Output		0/1	0	0	1	0	×	VCOUT1 = 0 (TO03) = 0 TO04 = 0 SDAA0 = 0Note 6 (SDAA0) = 0Note 3					
	SCL11	Output	PIOR24 = 0 PIOR23 = 0 PIOR22 = 0	0/1	0	0	1	0			✓	✓	✓	✓	—
	SDAA0	I/O	PIOR32 = 0	1	0	0	0	0	×	VCOUT1 = 0 (TO03) = 0 TO04 = 0	—	—	✓	✓	—
	(SDAA0)	I/O	PIOR33 = 0 PIOR32 = 1	1	0	0	0	0	×	VCOUT1 = 0 (TO03) = 0 TO04 = 0	✓	✓	—	—	—
	VCOUT1	Output	PIOR65 = 0 PIOR64 = 0 PIOR63 = 0	0	0	0	0	0	×	(TO03) = 0 TO04 = 0 SDAA0 = 0Note 6 (SDAA0) = 0Note 3	✓	✓	✓	✓	—
	(INTP5)	Input	PIOR51 = 0 PIOR50 = 1	×	0	1	×	0	×	×	✓	✓	✓	✓	—
P10	P10	Input	—	—	—	1	×	—	×	×	✓	✓	—	—	—
		Output	—	—	—	0	0/1	—	(SCK11/SCL11) = 1	(TO03) = 0 (PCLBUZ0) = 0					
	INTP8	Input	PIOR56 = 0	—	—	1	×	—	×	×	✓	✓	—	—	—
	(TI03)	Input	PIOR12 = 1 PIOR11 = 0 PIOR10 = 0	—	—	1	×	—	—	×	✓	✓	—	—	—
	(TO03)	Output		—	—	0	0	—	×	(PCLBUZ0) = 0	✓	✓	—	—	—
	(RxD1)	Input	PIOR31 = 1 PIOR30 = 0	—	—	1	×	—	×	×	✓	✓	—	—	—
	(PCLBUZ0)	Output	PIOR62 = 0 PIOR61 = 1 PIOR60 = 1	—	—	0	0	—	×	(TO03) = 0	✓	✓	—	—	—
	(SCK11)	Input	PIOR24 = 1 PIOR23 = 0 PIOR22 = 0	—	—	1	×	—	×	×	✓	✓	—	—	—
		Output		—	—	0	1	—	×	(TO03) = 0 (PCLBUZ0) = 0					
	(SCL11)	Output		—	—	0	1	—			✓	✓	—	—	—

Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function (7/14)

Pin Name	Used Function		PIORr	POMp	PMCq	PMn	Pm	TSSELt	Alternate Function Output		32-pin	24-pin	20-pin	16-pin	10-pin
	Function Name	I/O							SAU Output Function	Other than SAU					
P11	P11	Input	—	—	—	1	×	—	×	×	✓	✓	—	—	—
		Output	—	—	—	0	0/1	—	(SCK11/SCL11) = 1 (TxD1) = 1	(TO03) = 0 (PCLBUZ0) = 0 (RTC1HZ) = 0	✓	✓	—	—	—
	IVCMP1	Input	—	—	—	1	×	—	×	×	✓	✓	—	—	—
	INTP9	Input	PIOR57 = 0	—	—	1	×	—	×	×	✓	✓	—	—	—
	(TI03)	Input	PIOR12 = 1 PIOR11 = 0 PIOR10 = 1	—	—	1	×	—	×	×	✓	✓	—	—	—
	(TO03)	Output		—	—	0	0	—	×	(PCLBUZ0) = 0 (RTC1HZ) = 0	✓	✓	—	—	—
	(PCLBUZ0)	Output	PIOR62 = 1 PIOR61 = 0 PIOR60 = 0	—	—	0	0	—	×	(TO03) = 0 (RTC1HZ) = 0	✓	✓	—	—	—
	(RTC1HZ)	Output	PIOR67 = 1 PIOR66 = 0	—	—	0	0	—	×	(TO03) = 0 (PCLBUZ0) = 0	✓	✓	—	—	—
	(TxD1)	Output	PIOR31 = 1 PIOR30 = 0	—	—	0	1	—	(SCK11/SCL11) = 1	(TO03) = 0 (PCLBUZ0) = 0 (RTC1HZ) = 0	✓	✓	—	—	—
	(SCK11)	Input	PIOR24 = 1 PIOR23 = 0 PIOR22 = 1	—	—	1	×	—	×	×	✓	✓	—	—	—
		Output		—	—	0	1	—	(TxD1) = 1	(TO03) = 0 (PCLBUZ0) = 0 (RTC1HZ) = 0					
	(SCL11)	Output		—	—	0	1	—			✓	✓	—	—	—
P12	P12	Input	—	—	—	1	×	—	×	×	✓	—	—	—	—
		Output	—	—	—	0	0/1	—	×	×					
	(INTP8)	Input	PIOR56 = 1	—	—	1	×	—	×	×	✓	—	—	—	—
P13	P13	Input	—	—	—	1	×	—	×	×	✓	—	—	—	—
		Output	—	—	—	0	0/1	—	(SCK11/SCL11) = 1 (SCK20/SCL20) = 1	(RTC1HZ) = 0					
	(INTP2)	Input	PIOR43 = 1 PIOR42 = 0	—	—	1	×	—	×	×	✓	—	—	—	—
	(INTP4)	Input	PIOR47 = 1 PIOR46 = 1	—	—	1	×	—	×	×	✓	—	—	—	—
	(RTC1HZ)	Output	PIOR67 = 1 PIOR66 = 1	—	—	0	0	—	×	×	✓	—	—	—	—
	(SCK20)	Input	PIOR26 = 1 PIOR25 = 0	—	—	1	×	—	×	×	✓	—	—	—	—
		Output		—	—	0	1	—	(SCK11/SCL11) = 1	(RTC1HZ) = 0					
	(SCL20)	Output		—	—	0	1	—			✓	—	—	—	—
	(SCK11)	Input	PIOR24 = 1 PIOR23 = 1 PIOR22 = 0	—	—	1	×	—	×	×	✓	—	—	—	—
		Output		—	—	0	1	—	(SCK20/SCL20) = 1	(RTC1HZ) = 0					
	(SCL11)	Output		—	—	0	1	—			✓	—	—	—	—
P14	P14	Input	—	×	—	1	×	—	×	×	✓	—	—	—	—
		Output	—	0	—	0	0/1	—	(SDA11) = 0 (SDA20) = 0	×					
		N-ch open drain output	—	1	—	0	0/1	—							
	(INTP2)	Input	PIOR43 = 1 PIOR42 = 1	×	—	1	×	—	×	×	✓	—	—	—	—
	(SI11)	Input	PIOR24 = 1 PIOR23 = 1 PIOR22 = 0	×	—	1	×	—	×	×	✓	—	—	—	—
	(SDA11)	I/O		1	—	0	1	—	(SDA11) = 0	×	✓	—	—	—	—
	(SI20)	Input	PIOR26 = 1 PIOR25 = 0	×	—	1	×	—	×	×	✓	—	—	—	—
	(SDA20)	I/O		1	—	0	1	—	(SDA20) = 0	×	✓	—	—	—	—
	(RxD2)	Input	PIOR26 = 1 PIOR25 = 0	×	—	1	×	—	×	×	✓	—	—	—	—

Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function (8/14)

Pin Name	Used Function		PIORr	POMp	PMCq	PMn	Pm	TSSELt	Alternate Function Output		32-pin	24-pin	20-pin	16-pin	10-pin
	Function Name	I/O							SAU Output Function	Other than SAU					
P15	P15	Input	—	—	—	1	×	—	×	×	✓	—	—	—	—
		Output	—	—	—	0	0/1	—	(SO11) = 1 (SO20/TxD2) = 1	×					
	(INTP9)	Input	PIOR57 = 1	—	—	1	×	—	×	×	✓	—	—	—	—
	(TxD2)	Output	PIOR26 = 1 PIOR25 = 0	—	—	0	1	—	(SO11) = 1	×	✓	—	—	—	—
	(SO20)	Output	PIOR26 = 1 PIOR25 = 0	—	—	0	1	—	(SO11) = 1	×	✓	—	—	—	—
	(SO11)	Output	PIOR24 = 1 PIOR23 = 1 PIOR22 = 0	—	—	0	1	—	(SO20/TxD2) = 1	×	✓	—	—	—	—
P16	P16	Input	—	×	—	1	×	×	×	×	✓	—	—	—	—
		Output	—	0	—	0	0/1	0	(SDA20) = 1	(TO03) = 0 (SCLA0) = 0					
		N-ch open drain output	—	1	—	0	0/1	0							
	TS01	I/O	×	×	—	1	0	1	×	×	✓	—	—	—	—
	(TI03)	Input	PIOR12 = 1 PIOR11 = 1 PIOR10 = 0	×	—	1	×	0	×	×	✓	—	—	—	—
	(TO03)	Output		0	—	0	0	0	×	(SCLA0) = 0	✓	—	—	—	—
	(SI20)	Input	PIOR26 = 0 PIOR25 = 1	×	—	1	×	0	×	×	✓	—	—	—	—
	(SDA20)	I/O		1	—	0	1	0	×	(TO03) = 0 (SCAL0) = 0	✓	—	—	—	—
	(RxD2)	Input		×	—	1	×	0	×	×	✓	—	—	—	—
	(SCLA0)	I/O	PIOR33 = 1 PIOR32 = 1	1	—	0	0	0	×	(TO03) = 0	✓	—	—	—	—
P17	P17	Input	—	×	—	1	×	×	×	×	✓	—	—	—	—
		Output	—	0	—	0	0/1	0	(SO20/TxD2) = 1	(TO04) = 0 (SDAA0) = 0					
		N-ch open drain output	—	1	—	0	0/1	0							
	TS02	I/O	×	×	—	1	0	1	×	×	✓	—	—	—	—
	(TI04)	Input	PIOR14 = 1 PIOR13 = 0	×	—	1	×	0	×	×	✓	—	—	—	—
	(TO04)	Output		0	—	0	0	0	×	(SDAA0) = 0	✓	—	—	—	—
	(TxD2)	Output	PIOR26 = 0 PIOR25 = 1	0/1	—	0	1	0	×	(TO04) = 0 (SDAA0) = 0	✓	—	—	—	—
	(SO20)	Output		0/1	—	0	1	0			✓	—	—	—	—
	(SDAA0)	I/O	PIOR33 = 1 PIOR32 = 1	1	—	0	0	0	×	(TO04) = 0	✓	—	—	—	—

Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function (9/14)

Pin Name	Used Function		PIORr	POMp	PMCq	PMn	Pm	TSSELt	Alternate Function Output		32-pin	24-pin	20-pin	16-pin	10-pin
	Function Name	I/O							SAU Output Function	Other than SAU					
P20	P20	Input	—	×	0	1	×	0	×	×	✓	✓	✓	—	—
		Output	—	0	0	0	0/1	0	(SCK11/SCL11) = 1 (TxD1) = 1 SDA20 = 1	(TO00) = 0 (TO03) = 0	✓	✓	✓	—	—
		N-ch open drain output	—	1	0	0	0/1	0			✓	✓	✓	—	—
	ANI10	Analog input	—	×	1	1	×	0	×	×	✓	✓	✓	—	—
	(INTP1)	Input	PIOR41 = 1	×	0	1	×	0	×	×	✓	✓	✓	—	—
	(TI00)	Input	PIOR01 = 1 PIOR00 = 0	×	0	1	×	0	×	×	✓	✓	✓	—	—
	(TO00)	Output	PIOR01 = 1 PIOR00 = 1	0	0	0	0	0	×	(TO00) = 0	✓	✓	✓	—	—
	(TI03)	Input	PIOR12 = 0 PIOR11 = 1 PIOR10 = 1	×	0	1	×	0	×	×	✓	✓	✓	—	—
	(TO03)	Output		0	0	0	0	0	×	(TO03) = 0	✓	✓	✓	—	—
	(SCK11)	Input	PIOR24 = 0 PIOR23 = 1 PIOR22 = 0	×	0	1	×	0	×	×	✓	✓	✓	—	—
		Output		0/1	0	0	1	0	×	(TO00) = 0 (TO03) = 0	✓	✓	✓	—	—
	(SCL11)	Output		0/1	0	0	1	0			✓	✓	✓	—	—
	(RxD1)	Input	PIOR31 = 1 PIOR30 = 1	×	0	1	×	0	×	×	✓	—	—	—	—
	(TxD1)	Output	PIOR31 = 0 PIOR30 = 1	0/1	0	0	1	0	(SCK11/SCL11) = 1 SDA20 = 1	(TO00) = 0 (TO03) = 0	✓	✓	✓	—	—
	SI20	Input	PIOR26 = 0 PIOR25 = 0	×	0	1	×	0	×	×	✓	✓	✓	—	—
	RxD2	Input	PIOR26 = 0 PIOR25 = 0	×	0	1	×	0	×	×	✓	✓	✓	—	—
	SDA20	I/O	PIOR26 = 0 PIOR25 = 0	1	0	0	1	0	(SCK11/SCL11) = 1 (TxD1) = 1	×	✓	✓	✓	—	—
	TS11	I/O	×	×	×	1	0	1	×	×	✓	✓	✓	—	—
P21	P21	Input	—	—	0	1	×	0	×	×	✓	✓	✓	—	—
		Output	—	—	0	0	0/1	0	SO20/TxD2 = 1	(TO00) = 0	✓	✓	✓	—	—
	ANI9	Analog input	—	—	1	1	×	0	×	×	✓	✓	✓	—	—
	(INTP7)	Input	PIOR55 = 1 PIOR54 = 0	—	0	1	×	0	×	×	✓	✓	✓	—	—
	(TO00)	Output	PIOR01 = 1 PIOR00 = 0	—	0	0	0	0	×	×	✓	✓	✓	—	—
	(RxD1)	Input	PIOR31 = 0 PIOR30 = 1	—	0	1	×	0	×	×	✓	✓	✓	—	—
	SO20	Output	PIOR26 = 0 PIOR25 = 0	—	0	0	1	0	×	(TO00) = 0	✓	✓	✓	—	—
	TxD2	Output	PIOR26 = 0 PIOR25 = 0	—	0	0	1	0	×	(TO00) = 0	✓	✓	✓	—	—
	TS10	I/O	×	—	×	1	0	1	×	×	✓	✓	✓	—	—
P22	P22	Input	—	×	0	1	×	0	×	×	✓	✓	✓	—	—
		Output	—	0	0	0	0/1	0	(SDA11) = 1	(TO06) = 0	✓	✓	✓	—	—
		N-ch open drain output	—	1	0	0	0/1	0			✓	✓	✓	—	—
	ANI8	Analog input	—	×	1	1	×	0	×	×	✓	✓	✓	—	—
	(INTP5)	Input	PIOR51 = 1 PIOR50 = 1	×	0	1	×	0	×	×	✓	✓	✓	—	—
	(TI06)	Input	PIOR16 = 1	×	0	1	×	—	×	×	✓	✓	✓	—	—
	(TO06)	Output		0	0	0	0	—	×	×	✓	✓	✓	—	—
	(SDA11)	I/O	PIOR24 = 0 PIOR23 = 1 PIOR22 = 1	1	0	0	1	0	×	(TO06) = 0	✓	✓	✓	—	—
	TS09	I/O	×	×	×	1	0	1	×	×	✓	✓	✓	—	—

Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function (10/14)

Pin Name	Used Function		PIORr	POMp	PMCq	PMn	Pm	TSSELt	Alternate Function Output		32-pin	24-pin	20-pin	16-pin	10-pin
	Function Name	I/O							SAU Output Function	Other than SAU					
P23	P23	Input	—	—	0	1	×	0	×	×	✓	✓	✓	—	—
		Output	—	—	0	0	0/1	0	(SCL11) = 1	(TO04) = 1	✓	✓	✓	—	—
	ANI7	Analog input	—	—	1	1	×	0	×	×	✓	✓	✓	—	—
	(INTP6)	Input	PIOR53 = 1 PIOR52 = 0	—	0	1	×	0	×	×	✓	✓	✓	—	—
	(TI04)	Input	PIOR14 = 0 PIOR13 = 1	—	0	1	×	0	×	×	✓	✓	✓	—	—
	(TO04)	Output		—	0	0	0	0	×	×	✓	✓	✓	—	—
	(SCL11)	Output	PIOR24 = 0 PIOR23 = 1 PIOR22 = 1	—	0	0	1	0	×	(TO04) = 0	✓	✓	✓	—	—
	TS08	I/O	×	—	×	1	0	1	×	×	✓	✓	✓	—	—
P40	P40	Input	—	—	—	1	×	—	×	×	✓	✓	✓	✓	✓
		Output	—	—	—	0	0/1	—	×	(TO01) = 0 (PCLBUZ0) = 0	✓	✓	✓	✓	✓
	INTP2	Input	PIOR43 = 0 PIOR42 = 0	—	—	1	×	—	×	×	✓	✓	✓	✓	✓
	(TI01)	Input	PIOR03 = 0 PIOR02 = 1	—	—	1	×	—	×	×	✓	✓	✓	✓	✓
	(TO01)	Output		—	—	0	0	—	×	(PCLBUZ0) = 0	✓	✓	✓	✓	✓
	(PCLBUZ0)	Output	PIOR62 = 0 PIOR61 = 0 PIOR60 = 1	—	—	0	0	—	×	(TO01) = 0	✓	✓	✓	✓	✓

Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function (11/14)

Pin Name	Used Function		PIORr	POMp	PMCq	PMn	Pm	TSSELt	Alternate Function Output		32-pin	24-pin	20-pin	16-pin	10-pin
	Function Name	I/O							SAU Output Function	Other than SAU					
P41	P41	Input	—	×	—	1	×	0	×	×	✓	✓	✓	✓	—
		Output	—	0	—	0	0/1	0	SCK20/SCL20 = 1[Note 2] (SO11) = 1[Note 2] (SDA11) = 1[Note 2]	TO03 = 0 (TO02) = 0 RTC1HZ = 0 (VCOUT0) = 0 (VCOUT1) = 0	✓	✓	✓	✓	—
		N-ch open drain output	—	1	—	0	0/1	0			✓	✓	✓	✓	—
	(INTP3)	Input	PIOR45 = 1 PIOR44 = 0	×	—	1	×	0	×	×	✓	✓	✓	✓	—
	(INTP4)	Input	PIOR47 = 0 PIOR46 = 1	×	—	1	×	0	×	×	✓	✓	✓	✓	—
	(TI02)	Input	PIOR06 = 1 PIOR05 = 0 PIOR04 = 0	×	—	1	×	—	×	×	✓	✓	✓	✓	—
	(TO02)	Output		0	—	0	0	—	×	TO03 = 0 (TO02) = 0 RTC1HZ = 0 (VCOUT0) = 0 (VCOUT1) = 0	✓	✓	✓	✓	—
	TI03	Input	PIOR12 = 0 PIOR11 = 0 PIOR10 = 1	×	—	1	×	—	×	×	✓	✓	✓	✓	—
	TO03	Output	PIOR12 = 0 PIOR11 = 0 PIOR10 = 0	0	—	0	0	—	×	(TO02) = 0 RTC1HZ = 0 (VCOUT0) = 0 (VCOUT1) = 0	✓	✓	✓	✓	—
	(SO11)	Output	PIOR24 = 0 PIOR23 = 1 PIOR22 = 0	0/1	—	0	1	0	SCK20/SCL20 = 1[Note 2]	TO03 = 0 (TO02) = 0 RTC1HZ = 0 (VCOUT0) = 0 (VCOUT1) = 0	✓	✓	✓	—	—
	(SDA11)	I/O		1	—	0	1	0	SCK20/SCL20 = 1[Note 2]	TO03 = 0 (TO02) = 0 RTC1HZ = 0 (VCOUT0) = 0 (VCOUT1) = 0	✓	✓	✓	—	—
	SCK20	Input	PIOR26 = 0 PIOR25 = 0	×	—	1	×	—	×	×	✓	✓	✓	—	—
		Output		0/1	—	0	1	—	(SO11) = 1[Note 2] (SDA11) = 1[Note 2]	TO03 = 0 (TO02) = 0 RTC1HZ = 0 (VCOUT0) = 0 (VCOUT1) = 0	✓	✓	✓	—	—
	SCL20	Output		0/1	—	0	1	—			✓	✓	✓	—	—
	RTC1HZ	Output	PIOR67 = 0 PIOR66 = 0	0	—	0	0	0	×	TO03 = 0 (TO02) = 0 (VCOUT0) = 0 (VCOUT1) = 0	✓	✓	✓	✓	—
	(VCOUT0)	Output	PIOR65 = 0 PIOR64 = 1 PIOR63 = 1	0	—	0	0	0	×	TO03 = 0 (TO02) = 0 RTC1HZ = 0 (VCOUT1) = 0	✓	✓	✓	✓	—
	(VCOUT1)	Output	PIOR65 = 1 PIOR64 = 0 PIOR63 = 0	0	—	0	0	0	×	TO03 = 0 (TO02) = 0 RTC1HZ = 0 (VCOUT0) = 0	✓	✓	✓	✓	—
	TS13	I/O	×	×	—	1	0	1	×	×	✓	✓	✓	✓	—
P42	P42	Input	—	—	—	1	×	0	×	×	✓	—	—	—	—
		Output	—	—	—	0	0/1	0	(TxD1) = 1	×	✓	—	—	—	—
	(TI00)	Input	PIOR01 = 1 PIOR00 = 1	—	—	1	×	—	×	×	✓	—	—	—	—
	(TxD1)	Output	PIOR31 = 1 PIOR30 = 1	—	—	0	1	—	×	×	✓	—	—	—	—
	TS12	I/O	×	—	—	1	0	1	×	×	✓	—	—	—	—
P43	P43	Input	—	—	—	1	×	0	×	×	✓	—	—	—	—
		Output	—	—	—	0	0/1	0	×	×	✓	—	—	—	—
	TS14	I/O	—	—	—	1	0	1	×	×	✓	—	—	—	—

Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function (12/14)

Pin Name	Used Function		PIORr	POMp	PMCq	PMn	Pm	TSSELt	Alternate Function Output		32-pin	24-pin	20-pin	16-pin	10-pin
	Function Name	I/O							SAU Output Function	Other than SAU					
P60	P60	Input	—	—	—	1	×	—	×	×	✓	✓	—	—	—
		Output	—	—	—	0	0/1	—	×	SCLA0 = 0	✓	✓	—	—	—
	(INTP3)	Input	PIOR45 = 1 PIOR44 = 1	—	—	1	×	—	×	×	✓	✓	—	—	—
	(INTP6)	Input	PIOR53 = 1 PIOR52 = 1	—	—	1	×	—	×	×	✓	✓	—	—	—
	(SI11)	Input	PIOR24 = 1 PIOR23 = 0 PIOR22 = 0	—	—	1	×	—	×	×	✓	✓	—	—	—
	SCLA0	I/O	PIOR33 = 0 PIOR32 = 0	—	—	0	0	—	×	×	✓	✓	—	—	—
P61	P61	Input	—	—	—	1	×	—	×	×	✓	✓	—	—	—
		Output	—	—	—	0	0/1	—	(SO11)/(SDA11) = 1	SDAA0 = 0	✓	✓	—	—	—
	(INTP7)	Input	PIOR55 = 1 PIOR54 = 1	—	—	1	×	—	×	×	✓	✓	—	—	—
	(SO11)	Output	PIOR24 = 1 PIOR23 = 0 PIOR22 = 0	—	—	0	1	—	×	SDAA0 = 0	✓	✓	—	—	—
	(SDA11)	I/O		—	—	0	1	—	×	SDAA0 = 0	✓	✓	—	—	—
	SDAA0	I/O	PIOR33 = 0 PIOR32 = 0	—	—	0	0	—	×	×	✓	✓	—	—	—

Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function (13/14)

Pin Name	Used Function		PIORr	POMp	PMCq	PMn	Pm	TSSELt	CMC (EXCLK, OSCSEL, EXCLKS, OSCSELS, XTSEL)	32-pin	24-pin	20-pin	16-pin	10-pin
	Function Name	I/O												
P121	P121	Input	—	—	—	1	×	—	00xx0/10xx0/11xx0/xx001/xx101/xx111	✓	✓	✓	✓	—
		Output	—	—	—	0	0/1	—	00xx0/10xx0/11xx0/xx001/xx101/xx111	✓	✓	✓	✓	—
	X1	Input	—	—	—	—	×	—	01xx0	✓	✓	✓	✓	—
	XT1	Input	—	—	—	—	×	—	xx011	✓	✓	✓	✓	—
	(INTP3)	Input	PIOR45 = 0 PIOR44 = 1	—	—	1	×	—	00xx0/10xx0/11xx0/xx001/xx101/xx111	✓	✓	✓	✓	—
	(INTP4)	Input	PIOR47 = 1 PIOR46 = 0	—	—	1	×	—	00xx0/10xx0/11xx0/xx001/xx101/xx111	✓	✓	✓	✓	—
	(INTP5)	Input	PIOR51 = 1 PIOR50 = 0	—	—	1	×	—	00xx0/10xx0/11xx0/xx001/xx101/xx111	✓	✓	✓	✓	—
	TI07	Input	PIOR17 = 0	—	—	1	×	—	00xx0/10xx0/11xx0/xx001/xx101/xx111	✓	✓	✓	✓	—
	TO07	Output		—	—	0	0	—	00xx0/10xx0/11xx0/xx001/xx101/xx111	✓	✓	✓	✓	—
P122	P122	Input	—	—	—	1	×	—	00xx0/10xx0//xx001/xx101/	✓	✓	✓	✓	—
		Output	—	—	—	0	0/1	—	00xx0/10xx0//xx001/xx101/	✓	✓	✓	✓	—
	X2	Input	—	—	—	—	×	—	01xx0	✓	✓	✓	✓	—
	XT2	Input	—	—	—	—	×	—	xx011	✓	✓	✓	✓	—
	EXCLK	Input	—	—	—	—	×	—	11xx0	✓	✓	✓	✓	—
	EXCLKS	Input	—	—	—	—	×	—	xx111	✓	✓	✓	✓	—
	(INTP2)	Input	PIOR43 = 0 PIOR42 = 1	—	—	1	×	—	00xx0/10xx0//xx001/xx101/	✓	✓	✓	✓	—
	TI05	Input	PIOR15 = 0	—	—	1	×	—	00xx0/10xx0//xx001/xx101/	✓	✓	✓	✓	—
	TO05	Output		—	—	0	0	—	00xx0/10xx0//xx001/xx101/	✓	✓	✓	✓	—

Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function (14/14)

Pin Name	Used Function		PIORr	POMp	PMCq	PMn	Pm	TSSELt	Notes	32-pin	24-pin	20-pin	16-pin	10-pin
	Function Name	I/O												
P125	P125	Input	—	—	—	1	×	—	Optional bytes 000C1H PORTSELB = 0	✓	✓	✓	✓	✓
		Output	—	—	—	0	0/1	—		✓	✓	✓	✓	✓
	$\overline{\text{RESET}}$	Input	—	—	—	—	×	—	Optional bytes 000C1H PORTSELB = 1	✓	✓	✓	✓	✓
	(INTP0)	Input	PIOR40 = 1	—	—	1	×	—	Optional bytes 000C1H PORTSELB = 0	✓	✓	✓	✓	✓
	INTP1	Input	PIOR41 = 0	—	—	1	×	—		✓	✓	✓	✓	✓
	(VCOUT0)	Output	PIOR65 = 0 PIOR64 = 0 PIOR63 = 1	—	—	0	0	—		✓	✓	✓	✓	✓
	(VCOUT1)	Output	PIOR65 = 0 PIOR64 = 1 PIOR63 = 0	—	—	0	0	—		✓	✓	✓	✓	—
	(SI11)	Input	PIOR24 = 0 PIOR23 = 1 PIOR22 = 0	—	—	1	×	—		✓	✓	✓	—	—
P137	P137	Input	—	—	—	—	×	—	—	✓	✓	✓	✓	✓
	INTP0	Input	PIOR40 = 0	—	—	—	×	—	—	✓	✓	✓	✓	✓
	TI00	Input	PIOR01 = 0 PIOR00 = 0	—	—	—	×	—	—	✓	✓	✓	✓	✓

Note 1. 16-pin to 32-pin products only.

Note 2. 20-pin to 32-pin products only.

Note 3. 24-pin to 32-pin products only.

Note 4. 32-pin products only.

Note 5. 10-pin products only.

Note 6. 16-pin to 20-pin products only.

Note 7. When the touch pin function is in use (when the TSSELxx bit is set to 1), the P02/TSCAP pin automatically functions as TSCAP.

4.5.4 Operation of the port pin on which ANIxx pin and TSxx pin functions are multiplexed

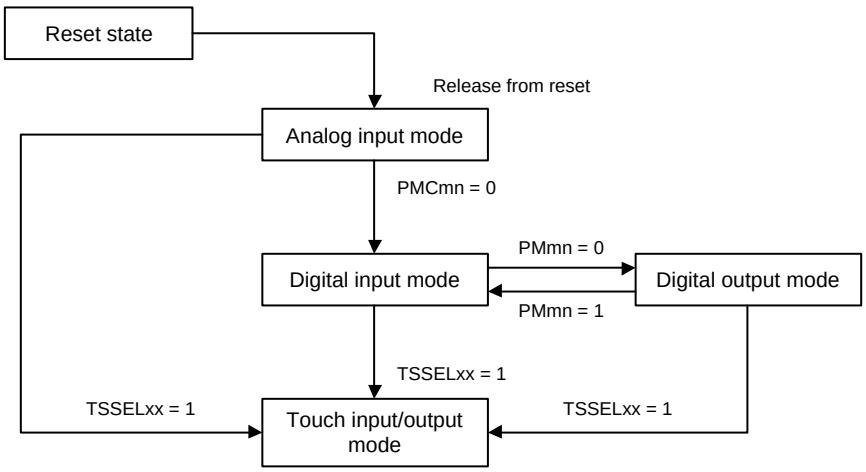
The port function multiplexed on the analog input pin (ANIxx) and touch pin (TSxx) is determined by the settings of the port mode register (PMxx), port mode control register (PMCxx), and touch pin function select registers (TSSEL0, TSSEL1).

Table 4-9. Settings for ANIxx/TSxx/Port Pin Function

TSSELxx Bit of TSSEL0 to TSSEL2 Registers	PMCxx Bit of PMC0 and PMC2 Registers	PMxx Bit of PMxx Register	Pin Function	Initial State
0	1	1	Analog input mode	✓
0	0	0	Digital output mode	—
0	0	1	Digital input mode	—
1	×	×	Touch input/output mode	—
Other than the above			Setting prohibited	

The figure below shows the state transitions of the ANIxx/TSxx/port pin function.

Figure 4-10. State Transition Diagram of ANIxx/TSxx/Port Pin Function



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 P00 is an output port, P01 to P07 are input ports (the status of all pins is high level), and the output latch value of port 0 is 00H, if the output of output port P00 is changed from low level to high level via a 1-bit manipulation instruction, the output latch value of port 0 is FFH.

[Explanation]

The targets of writing to and reading from the Pn register of a port whose PMnm bit is 1 are the output latch and pin status, respectively.

A 1-bit manipulation instruction is executed in the following order in the RL78 microcontroller.

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

<2> The targeted one bit is manipulated.

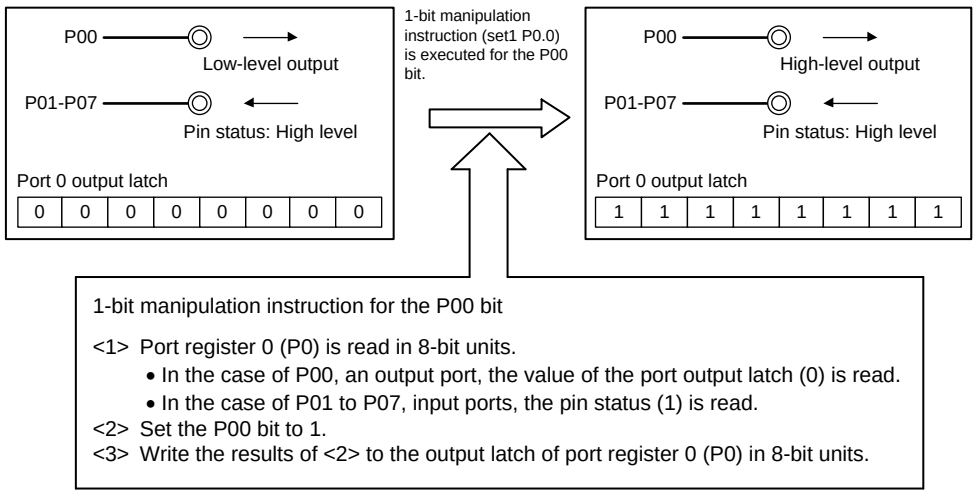
<3> The Pn register is written in 8-bit units.

In step <1>, the output latch value (0) of P00, which is an output port, is read, while the pin status of P01 to P07, which are input ports, is read. If the pin status of P01 to P07 is high level at this time, the read value is FEH.

The value is changed to FFH by the manipulation in <2>.

FFH is written to the output latch by the manipulation in <3>.

Figure 4-11. Bit Manipulation Instruction (P00)



4.6.2 Notes on specifying the pin settings

For an output pin to which multiple alternate 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 6 (PIOR0 to PIOR6). For details about the alternate output function, 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 CLOCK GENERATOR

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

	10-Pin Products	16-, 20-, 24-, and 32-Pin Products
X1 and X2 pins	—	✓
EXCLK pin	—	✓
XT1 and XT2 pins	—	✓
EXCLKS pin	—	✓

In 16-, 20-, 24-, and 32-pin products, P121 and P122 are used for both connecting resonator pin for main system clock and connecting resonator pin for subsystem clock. This can be set in the XTSEL bit of the CMC register.

5.1 Functions of Clock Generator

The clock generator generates the clock to be supplied to the CPU and peripheral hardware.

The following three kinds of system clocks and clock oscillators are selectable.

1) Main system clock

<1> X1 oscillator (16-pin, 20-pin, and 24-pin products and 32-pin products)

This circuit oscillates a clock of f_X = 1 to 12 MHz by connecting a resonator to X1 and X2 pins.
The external main system clock (f_{EX} = 1 to 16 MHz) can also be supplied from EXCLK/X2/P122 pin.
Oscillation can be stopped by executing the STOP instruction or setting of the MSTOP bit (bit 7 of the clock operation status control register (CSC)).

<2> High-speed on-chip oscillator

The frequency at which to oscillate can be selected from among f_{IH} = 16/8/4/2/1 MHz (typ.) by using the option byte (000C2H). After a reset release, the CPU always starts operating with this high-speed on-chip oscillator clock.
Oscillation can be stopped by executing the STOP instruction or setting 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 5-10 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	4	8	16
2.4 V ≤ V_{DD} ≤ 5.5 V	✓	✓	✓	✓	✓

Remark ✓: Can operate, — : Cannot operate

The external main system clock (f_{EX} = 1 to 16 MHz) can also be supplied from the EXCLK/X2/P122 pin. The 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 high-speed on-chip oscillator clock can be selected by setting of the MCM0 bit (bit 4 of the system clock control register (CKC)).

2) Subsystem clock

<1> XT1 clock oscillator (16-, 20-, and 24-pin products and 32-pin products)

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

The external subsystem clock (f_{EXS} = 32.768 kHz) can also be supplied from the EXCLKS/XT2/P122 pin.

The external subsystem clock input can be disabled by the setting of the XTSTOP bit.

3) Low-speed on-chip oscillator clock

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

The low-speed on-chip oscillator clock cannot be used as the CPU clock.

Only the following peripheral hardware runs on the low-speed on-chip oscillator clock.

- Watchdog timer
- Real-time clock 2
- 12-bit Interval timer

This clock operates when bit 4 (WDTON) of the option byte (000C0H), bit 4 (WUTMMCK0) of the operation speed mode control register (OSMC), or both are set to 1.

However, when WDTON = 1, WUTMMCK0 = 0, and bit 0 (WDSTBYON) of the option byte (000C0H) is 0, oscillation of the low-speed on-chip oscillator stops if the HALT or STOP instruction is executed.

Remark f_X: X1 clock oscillation frequency
f_{EX}: External main system clock frequency
f_{IH}: High-speed on-chip oscillator clock frequency
f_{XT}: XT1 clock oscillation frequency
f_{EXS}: External subsystem clock frequency
f_{IL}: Low-speed on-chip oscillator clock frequency

5.2 Configuration of Clock Generator

The clock generator includes the following hardware.

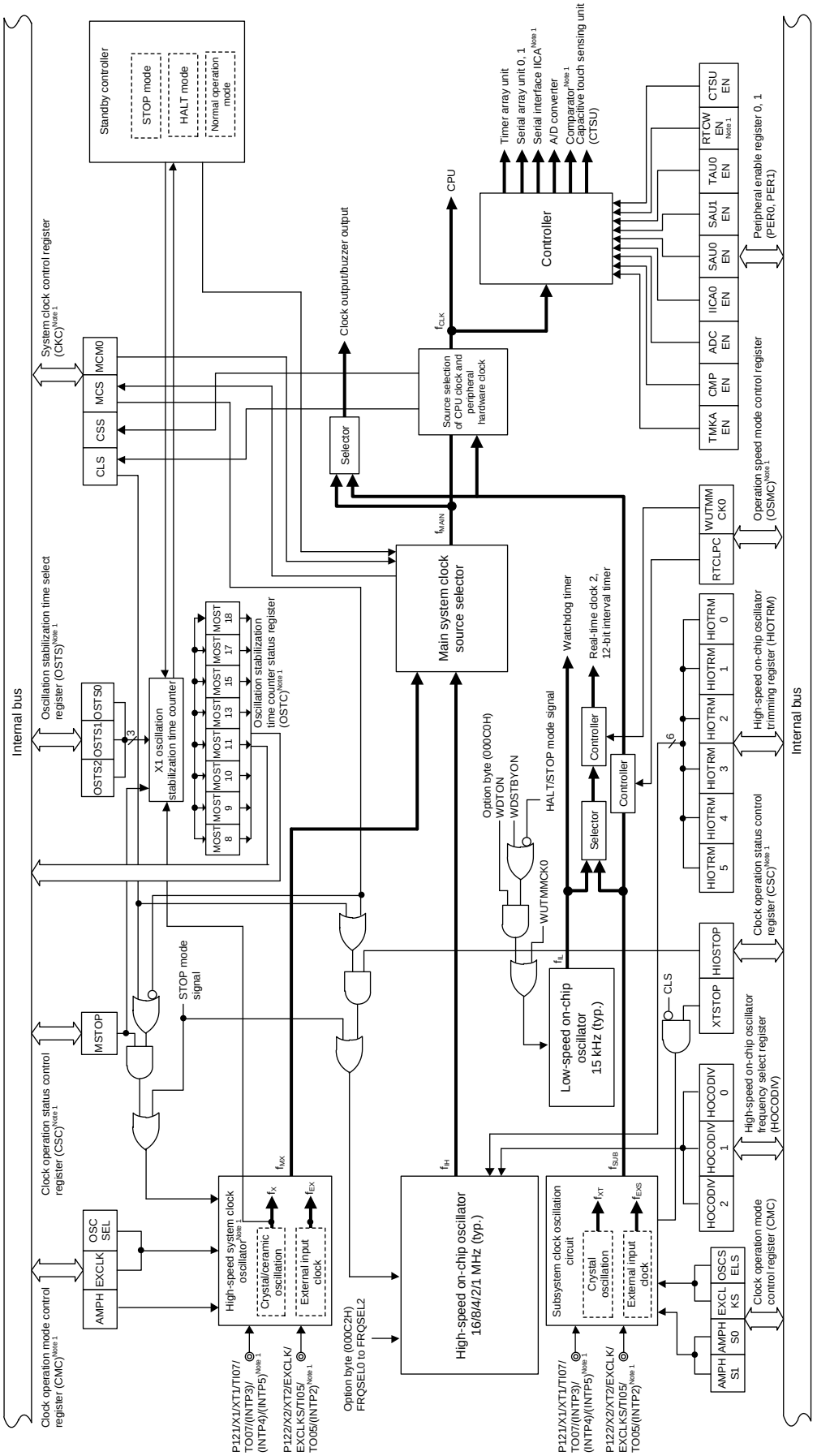
Table 5-1. Configuration of Clock Generator

Item	Configuration	10-pin products	16-pin products	20-pin products	24-pin products	32-pin products
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 register 0 (PER0)	✓	✓	✓	✓	✓
	Peripheral enable register 1 (PER1)	✓	✓	✓	✓	✓
	Operation speed mode control register (OSMC)	✓	✓	✓	✓	✓
	High-speed on-chip oscillator frequency select register (HOCODIV)	✓	✓	✓	✓	✓
	High-speed on-chip oscillator trimming register (HIOTRM)	✓	✓	✓	✓	✓
Oscillators	X1 oscillator	—	✓	✓	✓	✓
	XT1 oscillator	—	✓	✓	✓	✓
	High-speed on-chip oscillator	✓	✓	✓	✓	✓
	Low-speed on-chip oscillator	✓	✓	✓	✓	✓

Remark ✓: Provided, —: Not provided

<R>

Figure 5-1. Block Diagram of Clock Generator



Note 1. 16-, 20-, and 24-pin products and 32-pin products

Remark f_X: X1 clock oscillation frequency
f_{IH}: High-speed on-chip oscillator clock frequency
f_{EX}: External main system clock frequency
f_{MX}: High-speed system clock frequency
f_{MAIN}: Main system clock frequency
f_{XT}: XT1 clock oscillation frequency
f_{EXS}: External subsystem clock frequency
f_{SUB}: Subsystem clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency
f_{IL}: Low-speed on-chip oscillator clock frequency

5.3 Registers Controlling Clock Generator

The clock generator is controlled by the following registers depending on the products.

1) 10-pin products

- Peripheral enable register 0 (PER0)
- Peripheral enable register 1 (PER1)
- High-speed on-chip oscillator frequency select register (HOCODIV)
- Operation speed mode control register (OSMC)
- High-speed on-chip oscillator trimming register (HIOTRM)

2) 16-, 20-, and 24-pin products and 32-pin products

- 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 register 0 (PER0)
- Peripheral enable register 1 (PER1)
- Operation speed mode control register (OSMC)
- High-speed on-chip oscillator frequency select register (HOCODIV)
- High-speed on-chip oscillator trimming register (HIOTRM)

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.

5.3.1 Clock operation mode control register (CMC)

This register is used to set the operation mode of the X1/XT1/P121/(INTP3)/(INTP4)/(INTP5)/TI07/TO07 and X2/XT2/P122/EXCLK/EXCLKS/TI05/TO05/(INTP2) 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 5-2. Format of Clock Operation Mode Control Register (CMC)

Address: FFFA0H After reset: 00H R/W

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

XTSEL Note 1	EXCLK	OSCSEL	EXCLKS Note 1	OSCSELS Note 1	System clock pin operation mode	X1/XT1/P121 pin	X2/XT2/P122 pin
0	0	0	0	0	Port mode	Port	
0	0	1	0	0	X1 oscillation mode	Crystal/ceramic resonator connection	
0	1	0	0	0	Port mode	Port	
0	1	1	0	0	External clock input mode	Port	EXCLK input
1	0	0	0	0	Port mode	Port	
1	0	0	0	1	XT1 oscillation mode	Crystal resonator connection	
1	0	0	1	0	Port mode	Port	
1	0	0	1	1	External clock input mode	Port	EXCLKS input
Other than above					Setting prohibited		

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

AMPH	Control of X1 clock oscillation frequency
0	1 MHz ≤ f_X ≤ 10 MHz
1	10 MHz ≤ f_X ≤ 12 MHz

Note 1. The EXCLKS, OSCSELS, XTSEL, AMPHS1, and AMPHS0 bits are reset only by an internal reset by data retention power supply voltage; it retains the value when a reset caused by another factor occurs.

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

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

Caution 3. **Be sure to set the AMPH bit to 1 if the X1 clock oscillation frequency exceeds 10 MHz. Specify the settings for the AMPH, AMPHS1, and AMPHS0 bits while f_{IH} is selected as f_{CLK} after a reset ends (before f_{CLK} is switched to f_{MX} or f_{SUB}).**

Caution 4. **Switch the operation mode of the X1 pin and X2 pin only when MSTOP = 1.**

Caution 5. **Count the oscillation stabilization time of f_{XT} by software.**

Caution 6. **The XT1 oscillator is a circuit with low amplification in order to achieve low-power consumption. Note the following points when designing the circuit.**

- **Pins and circuit boards include parasitic capacitance. Therefore, perform oscillation evaluation using a circuit board to be actually used and confirm that there are no problems.**
- **Before using the ultra-low power consumption oscillation (AMPHS1 = 1, AMPHS0 = 0) as the mode of the XT1 oscillator, evaluate the resonators described in 5.7 Resonator and Oscillator Constants.**
- **Make the wiring between the XT1 and XT2 pins and the resonators as short as possible, and minimize the parasitic capacitance and wiring resistance. Note this particularly when the ultra-low power consumption oscillation (AMPHS1 = 1, AMPHS0 = 0) is selected.**
- **Configure the circuit of the circuit board, using material with little parasitic capacitance and wiring resistance.**
- **Place a ground pattern that has the same potential as V_{SS} as much as possible near the XT1 oscillator.**
- **Be sure that the signal lines between the XT1 and XT2 pins, and the resonators do not cross with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **The impedance between the XT1 and XT2 pins may drop and oscillation may be disturbed due to moisture absorption of the circuit board in a high-humidity environment or dew condensation on the board. When using the circuit board in such an environment, take measures to damp-proof the circuit board, such as by coating.**
- **When coating the circuit board, use material that does not cause capacitance or leakage between the XT1 and XT2 pins.**

Caution 7. **Be sure to use with V_{DD} = 2.4 V or more when setting XTSEL = 1 and setting P121/X1/XT1 pin and P122/X2/EXCLK/XT2/EXCLKS pin to the XT1 oscillation mode in the 16-pin to 32-pin products.**

Remark f_X: X1 clock frequency

5.3.2 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 5-3. 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	MCS	MCM0	0	0	0	0

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

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

MCS	Status of Main system clock (f_{MAIN})
0	High-speed on-chip oscillator clock (f_{IH})
1	High-speed system clock (f_{MX})

MCM0[Note 2]	Main system clock (f_{MAIN}) operation control
0	Selects the high-speed on-chip oscillator clock (f_{IH}) as the main system clock (f_{MAIN})
1	Selects the high-speed system clock (f_{MX}) as the main system clock (f_{MAIN})

Note 1. Bits 7 and 5 are read-only.

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

Caution 1. Be sure to set bit 0 to 3 to 0.

Caution 2. The clock set by the CSS bit is supplied to the CPU and peripheral hardware. If the CPU clock is changed, therefore, the clock supplied to peripheral hardware (except the real-time clock 2, 12-bit interval timer, clock output/buzzer output, 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 26 ELECTRICAL SPECIFICATIONS (T_A = −40 to +85°C) or CHAPTER 27 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C, T_A = −40 to +125°C).

Remark f_{IH}: High-speed on-chip oscillator clock frequency
f_{MX}: High-speed system clock frequency
f_{MAIN}: Main system clock frequency
f_{SUB}: Subsystem clock frequency

5.3.3 Clock operation status control register (CSC)

This register is used to control the operations of the high-speed system clock, high-speed on-chip oscillator clock, and subsystem clock (except the low-speed on-chip oscillator clock).

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 5-4. Format of Clock Operation Status Control Register (CSC)

Address: FFFA1H After reset: C0H R/W

Symbol	7	6	5	4	3	2	1	0
CSC	MSTOP	XTSTOP[Note 1]	0	0	0	0	0	HIOSTOP

- 16-, 20-, and 24-pin products and 32-pin products

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

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

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

Note 1. The XTSTOP bit is reset only by an internal reset by data retention power supply voltage; it retains the value when a reset caused by another factor occurs.

Caution 1. After reset release, set the clock operation mode control register (CMC) before setting the CSC register.

Caution 2. Switch the operation mode of the X1 pin and X2 pin only when MSTOP = 1.

Caution 3. When setting MSTOP bit to 0, switch the X1 pin and X2 pin to the f_X operation mode beforehand. Setting the MSTOP flag is disabled in the input port mode.

Caution 4. 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 5. 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 6. When setting MSTOP bit to 1 in the f_X operation mode, make sure that MCS in the CKC register is 0 beforehand.

Caution 7. **In the f_X operation mode, writing to the MSTOP flag is enabled but the stop control is not performed.**

Caution 8. **In the f_{XT} operation mode, writing to the XTSTOP flag is enabled but the stop control is not performed.**

Caution 9. **When starting XT1 oscillation by setting the XTSTOP bit, wait for oscillation of the subsystem clock to stabilize by setting a wait time using software.**

Caution 10. **Do not stop the clock selected for the CPU peripheral hardware clock (f_{CLK}) with the OSC register.**

Caution 11. **The setting of the flags of the register to stop clock oscillation or invalidate the external clock input and the condition before clock oscillation is to be stopped are as Table 5-2. Before stopping the clock oscillation, check the conditions before the clock oscillation is stopped.**

Table 5-2. Stopping Clock Method

Clock	Conditions Before Clock Oscillation Is Stopped	Flag Settings of CSC Register
X1 clock External main system clock	CPU/peripheral hardware clock operates with a clock other than the high-speed system clock (CLS = 0 and MCS = 0, or CLS = 1).	MSTOP = 1
XT1 clock External subsystem clock	CPU/peripheral hardware clock operates with a clock other than the subsystem clock (CLS = 0).	XTSTOP = 1
High-speed on-chip oscillator clock	CPU/peripheral hardware clock operates with the high-speed system clock (CLS = 0 and MCS = 1, or CLS = 1).	HIOSTOP = 1

5.3.4 Oscillation stabilization time counter status register (OSTC)

This register is used to indicate the count status of the X1 clock oscillation stabilization time counter.

The X1 clock oscillation stabilization time can be checked in the following case.

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

The OSTC register can be read by a 1-bit or 8-bit memory manipulation instruction.

When reset signal is generated, 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 5-5. 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	MOST10	MOST11	MOST13	MOST15	MOST17	MOST18

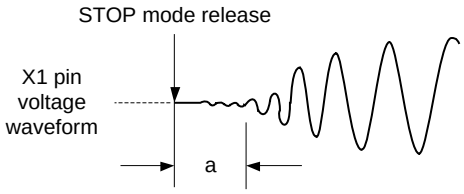
MOST8	MOST9	MOST10	MOST11	MOST13	MOST15	MOST17	MOST18	Oscillation stabilization time status		
									f_X = 10 MHz	f_X = 16 MHz
0	0	0	0	0	0	0	0	$(2^8 + 16)/f_X$ max.	27.2 μs max.	17.0 μs max.
1	0	0	0	0	0	0	0	$(2^8 + 16)/f_X$ min.	27.2 μs min.	17.0 μs min.
1	1	0	0	0	0	0	0	$(2^9 + 16)/f_X$ min.	52.8 μs min.	33.0 μs min.
1	1	1	0	0	0	0	0	$(2^{10} + 16)/f_X$ min.	104 μs min.	65.0 μs min.
1	1	1	1	0	0	0	0	$(2^{11} + 16)/f_X$ min.	206 μs min.	129 μs min.
1	1	1	1	1	0	0	0	$(2^{13} + 16)/f_X$ min.	820 μs min.	513 μs min.
1	1	1	1	1	1	0	0	$(2^{15} + 16)/f_X$ min.	3.27 ms min.	2.05 ms min.
1	1	1	1	1	1	1	0	$(2^{17} + 16)/f_X$ min.	13.1 ms min.	8.19 ms min.
1	1	1	1	1	1	1	1	$(2^{18} + 16)/f_X$ min.	26.2 ms min.	16.4 ms min.

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

Caution 2. **The oscillation stabilization time counter counts up to the oscillation stabilization time set by the oscillation stabilization time select register (OSTS).**
In the following cases, set the oscillation stabilization time of the OSTS register to the value greater than the count value which is to be checked by the OSTC register.

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

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



Remark f_X: X1 clock oscillation frequency

5.3.5 Oscillation stabilization time select register (OSTS)

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

When the X1 clock is made to oscillate, the operation is automatically waits for the time set using the OSTS register after operation of the X1 oscillation circuit is started (MSTOP = 0). When switching the CPU clock from the high-speed on-chip oscillator clock or the subsystem clock to the X1 clock, and when 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, use the oscillation stabilization time counter status register (OSTC) to confirm that the oscillation stabilization time has elapsed.

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 5-6. 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		
				f_X = 10 MHz	f_X = 16 MHz
0	0	0	$(2^8 + 16)/f_X$	27.2 μs	17.0 μs
0	0	1	$(2^9 + 16)/f_X$	52.8 μs	33.0 μs
0	1	0	$(2^{10} + 16)/f_X$	104 μs	65.0 μs
0	1	1	$(2^{11} + 16)/f_X$	206 μs	129 μs
1	0	0	$(2^{13} + 16)/f_X$	820 μs	513 μs
1	0	1	$(2^{15} + 16)/f_X$	3.27 ms	2.05 ms
1	1	0	$(2^{17} + 16)/f_X$	13.1 ms	8.19 ms
1	1	1	$(2^{18} + 16)/f_X$	26.2 ms	16.4 ms

Caution 1. To set the STOP mode when the X1 clock is used as the CPU clock, set the OSTS register before executing the STOP instruction.

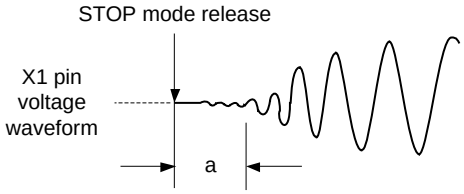
Caution 2. Change the setting of the OSTS register before setting the MSTOP bit of the clock operation status control register (CSC) to 0.

Caution 3. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

Caution 4. The oscillation stabilization time counter counts up to the oscillation stabilization time set by the OSTS register.
In the following cases, set the oscillation stabilization time of the OSTS register to the value greater than the count value which is to be checked by the OSTC register after the oscillation starts.

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

Caution 5. The X1 clock oscillation stabilization time does not include the time until clock oscillation starts ("a" below).



Remark f_X: X1 clock oscillation frequency

5.3.6 Peripheral enable register 0, 1 (PER0, PER1)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise.

To use the peripheral functions below, which are controlled by this register, set (1) the bit corresponding to each function before specifying the initial settings of the peripheral functions.

- 12-bit Interval timer
- A/D converter
- Comparator
- IICA Serial interface IICA
- Serial array unit n
- Timer array unit
- Real-time clock 2
- Capacitive touch sensing unit (CTSU)

Remark n = 0, 1

The PER0 and PER1 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 5-7. Format of Peripheral Enable Register 0 (PER0) (1/2)

Address: F00F0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PER0	TMKAEN	CMPEN	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

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 12-bit interval timer is in the reset status.
1	Enables input clock supply. • SFR used by the 12-bit interval timer 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 comparator is in the reset status.
1	Enables input clock supply. • SFR used by the comparator can be read and written.

Figure 5-7. Format of Peripheral Enable Register 0 (PER0) (2/2)

Address: F00F0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PER0	TMKAEN	CMPEN	ADCEN	IICA0EN	SAU1EN	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. • The A/D converter is in the reset status.
1	Enables input clock supply. • SFR used by the A/D converter can be read and written.

IICA0EN	Control of serial interface IICA input clock supply
0	Stops input clock supply. • SFR used by the serial interface IICA cannot be written. • The serial interface IICA is in the reset status.
1	Enables input clock supply. • SFR used by the serial interface IICA can be read and written.

SAU1EN	Control of serial array unit 1 input clock supply
0	Stops input clock supply. • SFR used by the serial array unit 1 cannot be written. • The serial array unit 1 is in the reset status.
1	Enables input clock supply. • SFR used by the serial array unit 1 can be read and written.

SAU0EN	Control of serial array unit 0 input clock supply
0	Stops input clock supply. • SFR used by the serial array unit 0 cannot be written. • The serial array unit 0 is in the reset status.
1	Enables input clock supply. • SFR used by the serial array unit 0 can be read and written.

TAU0EN	Control of timer array unit input clock supply
0	Stops input clock supply. • SFR used by the timer array unit cannot be written. • The timer array unit is in the reset status.
1	Enables input clock supply. • SFR used by the timer array unit can be read and written.

Caution 1. Be sure to clear bit 1 to 0.

Caution 2. Be sure to clear the following bits to 0.
10-pin and 16-pin products: bit 3

Caution 3. Do not switch the target bits of the PER0 register while the operation of the peripheral functions is enabled. Switch the setting by the PER0 register while the peripheral functions that are assigned to the PER0 register are being stopped.

Figure 5-8. Format of Peripheral Enable Register 1 (PER1)

- 16-, 20-, and 24-pin products and 32-pin products

Address: F007AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PER1	RTCWEN	0	0	0	0	0	CTSUEN	0

RTCWEN	Control of real-time clock 2 input clock supply
0	Stops input clock supply (stops f_{CLK} supply). • SFR used by the real-time clock 2 cannot be written. • Operable in real-time
1	Enables input clock supply. • SFR used by the real-time clock 2 can be read and written.

- 10-, 16-, 20-, and 24-pin products and 32-pin products

CTSUEN	Control of capacitive touch sensing unit input clock supply
0	Stops input clock supply. • SFR used by the capacitive touch sensing unit cannot be written. • The capacitive touch sensing unit is in the reset status.
1	Enables input clock supply. • SFR used by the capacitive touch sensing unit can be read and written.

Caution 1. Be sure to clear the following bits to 0.
10-pin products: bits 0, 2 to 7
16-, 20-, 24-, and 32-pin products: bits 0, 2 to 6

Caution 2. When the real-time clock 2 is used, first set the RTCWEN bit to 1, while oscillation of the input clock (f_{RTC}) is stable. If RTCWEN = 0, writing to the registers controlling the real-time clock 2 is ignored.

5.3.7 Operation speed mode control register (OSMC)

This register is used to reduce power consumption by stopping unnecessary clock functions.

If the RTCLPC bit is set to 1, power consumption can be reduced, because clock supply to the peripheral functions, except the real-time clock 2 and 12-bit interval timer, is stopped in STOP mode or HALT mode while subsystem clock is selected as CPU clock.

In addition, the OSMC register can be used to select the count clock of the real-time clock 2 and 12-bit interval timer.

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

Reset signal generation clears this register to 00H.

Figure 5-9. Format of Operation Speed Mode Control Register (OSMC)

Address: F00F3H After reset: 00H R/W

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

RTCLPC	Setting in STOP mode and in HALT mode while the CPU is operating with subsystem clock
0	Enables supply of the subsystem clock to peripheral functions (See **Table 17-1**, **Table 17-2**, and **Table 17-3** for the peripheral functions whose operations are enabled).
1	Stops supply of the subsystem clock to peripheral functions other than real-time clock 2 and 12-bit interval timer.

WUTMMCK0	Selection of the operation clock for real-time clock 2 and 12-bit interval timer
0	Subsystem clock
1	Low-speed on-chip oscillator clock (f_{IL})

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

This register is used to change the frequency of the high-speed on-chip oscillator which is set by an 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 5-10. 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	High-speed on-chip oscillator clock frequency selection
0	0	1	16 MHz
0	1	0	8 MHz
0	1	1	4 MHz
1	0	0	2 MHz
1	0	1	1 MHz
Other than above			Setting prohibited

Caution 1. Set the HOCODIV register with the high-speed on-chip oscillator clock (f_{IH}) selected as the CPU/peripheral hardware clock (f_{CLK}).

Caution 2. 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**

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

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

With self-measurement of the high-speed on-chip oscillator frequency via a timer using high-accuracy external clock input (timer array unit), 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 5-11. Format of High-Speed On-Chip Oscillator Trimming Register (HIOTRM)

Address: F00A0H After reset: undefined[Note 1] 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 1. The value after reset is the value adjusted at shipment.

Remark The HIOTRM register holds a six-bit value used to adjust the high-speed on-chip oscillator with an increment of 1 corresponding to an increase of frequency by about 0.05%.

5.4 System Clock Oscillator

5.4.1 X1 oscillator (16-, 20-, and 24-pin products and 32-pin products)

The X1 oscillator oscillates with a crystal resonator or ceramic resonator (1 to 12 MHz) connected to the X1 pin and X2 pin. 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

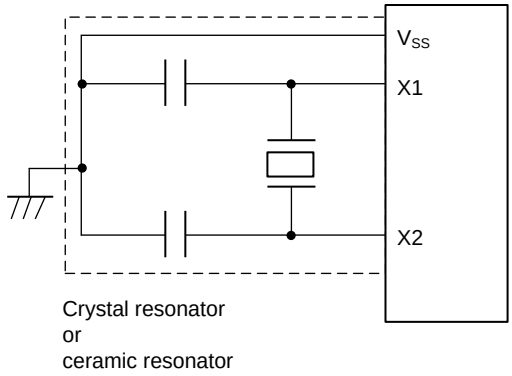
When the X1 oscillator is not used, set the input port mode (EXCLK, OSCSEL = 0, 0).

When the pins are not used as input port pins, either, see **Table 2-2 Connections of Unused Pins**.

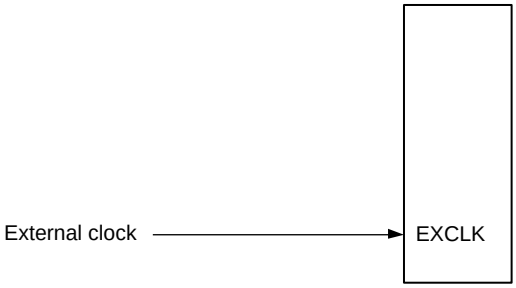
Figure 5-12 shows an example of the external circuit of the X1 oscillator.

Figure 5-12. Example of External Circuit of X1 Oscillator

(a) Crystal or ceramic oscillation



(b) External clock input



(Cautions are listed on the next page.)

5.4.2 XT1 oscillator (16-, 20-, and 24-pin products and 32-pin products)

The XT1 oscillator oscillates with a crystal resonator (32.768 kHz (typ.) connected to the XT1 pin and XT2 pin.

To use the XT1 oscillator, set bit 4 (OSCSELS) of the clock operation mode control register (CMC) to 1.

An external clock can also be input. In this case, input the clock signal to the EXCLKS pin. To use the XT1 oscillator, set bits 5 and 4 (EXCLKS, OSCSELS) of the clock operation mode control register (CMC) as follows.

- Crystal oscillation: EXCLKS, OSCSELS = 0, 1
- External clock input: EXCLKS, OSCSELS = 1, 1

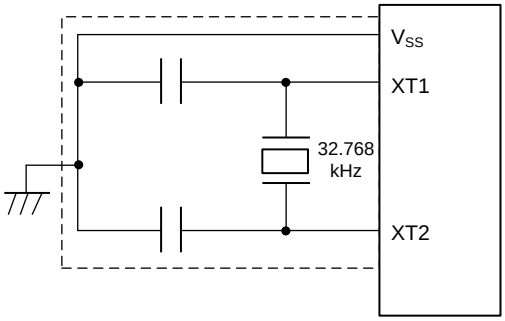
When the XT1 oscillator is not used, set the input port mode (EXCLKS, OSCSELS = 0, 0).

When the pins are not used as input port pins, either, see **Table 2-2 Connections of Unused Pins**.

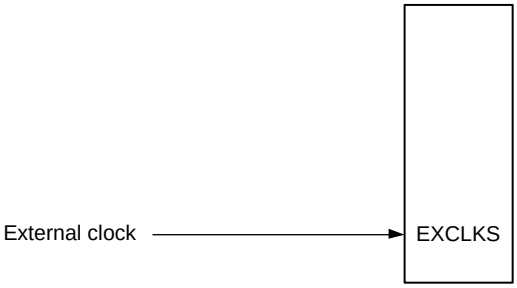
Figure 5-13 shows an example of the external circuit of the XT1 oscillator.

Figure 5-13. Example of External Circuit of XT1 Oscillator

(a) Crystal oscillation



(b) External clock



Caution When using the X1 oscillator and XT1 oscillator, wire as follows in the area enclosed by the broken lines in Figure 5-12 and Figure 5-13 to avoid an adverse effect from wiring capacitance.

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

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

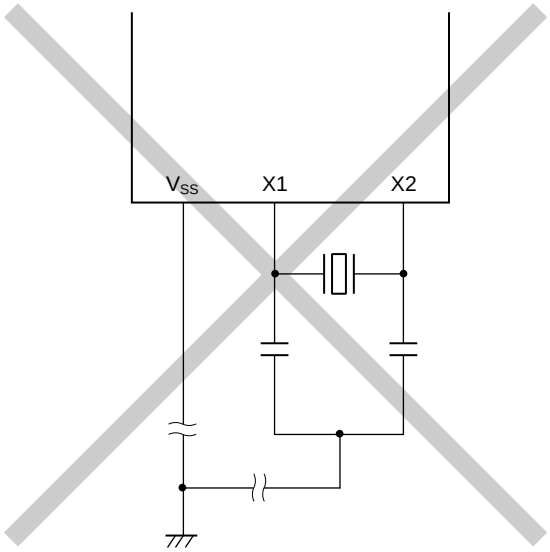
- **Pins and circuit boards include parasitic capacitance. Therefore, perform oscillation evaluation using a circuit board to be actually used and confirm that there are no problems.**
- **Before using the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) as the mode of the XT1 oscillator, evaluate the resonators described in 5.7 Resonator and Oscillator Constants.**
- **Make the wiring between the XT1 and XT2 pins and the resonators as short as possible, and minimize the parasitic capacitance and wiring resistance. Note this particularly when the ultra-low power consumption oscillation (AMPHS1, AMPHS0 = 1, 0) is selected.**

- **Configure the circuit of the circuit board, using material with little parasitic capacitance and wiring resistance.**
- **Place a ground pattern that has the same potential as V_{SS} as much as possible near the XT1 oscillator.**
- **Be sure that the signal lines between the XT1 and XT2 pins, and the resonators do not cross with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows.**
- **The impedance between the XT1 and XT2 pins may drop and oscillation may be disturbed due to moisture absorption of the circuit board in a high-humidity environment or dew condensation on the board. When using the circuit board in such an environment, take measures to damp-proof the circuit board, such as by coating.**
- **When coating the circuit board, use material that does not cause capacitance or leakage between the XT1 and XT2 pins.**

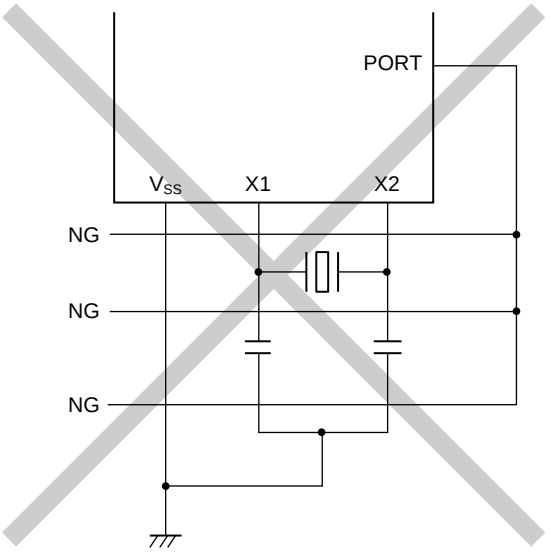
Figure 5-14 shows examples of incorrect resonator connection.

Figure 5-14. Examples of Incorrect Resonator Connection (1/2)

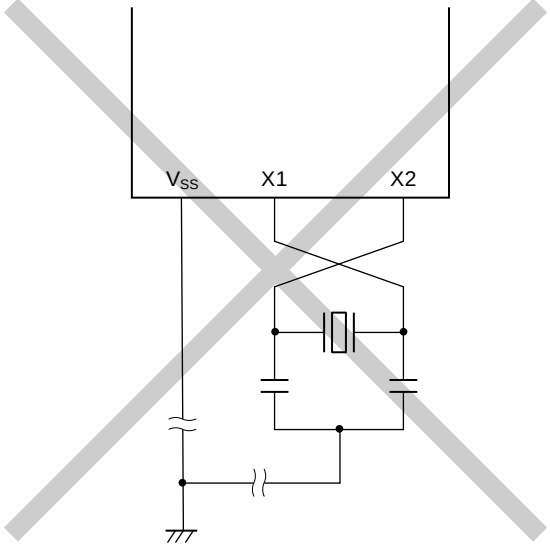
(a) Too long wiring



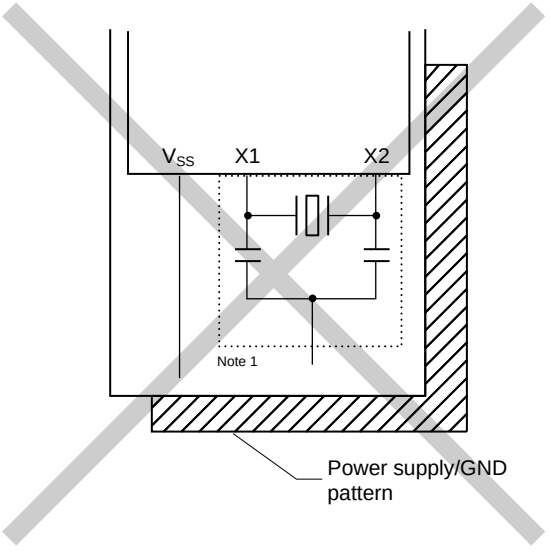
(b) Crossed signal line



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



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

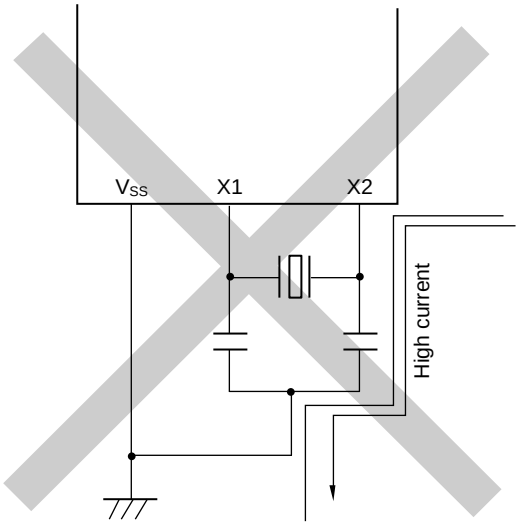


Note 1. Do not place a power supply/GND pattern under the wiring section (section indicated by a broken line in the figure) of the X1 pin and X2 pin and the resonators in a multi-layer board or double-sided board.
Do not configure a layout that will cause capacitance elements and affect the oscillation characteristics.

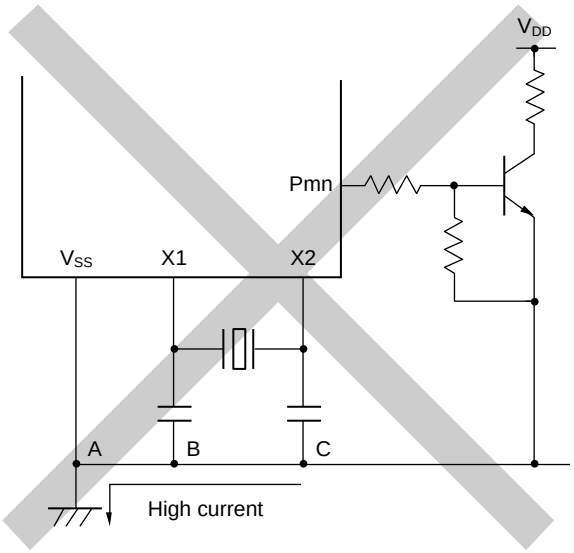
Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. In addition, insert a resistor in series to the XT2 side.

Figure 5-14. Examples of Incorrect Resonator Connection (2/2)

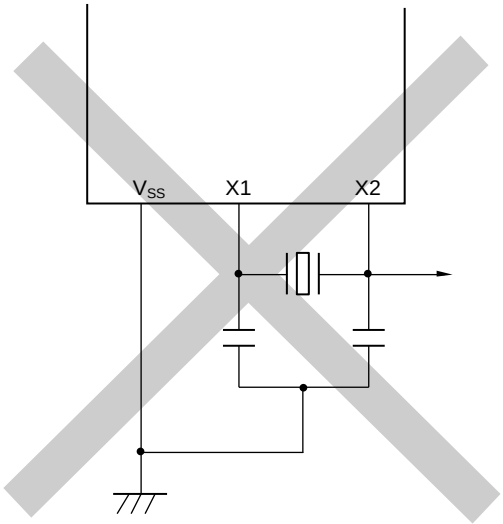
(e) Wiring near high alternating current



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



(g) Signals are fetched



Caution **When X2 and X1 pins are wired in parallel, the crosstalk noise of X2 pin may increase with X1 pin, resulting in malfunctioning.**

Remark When using the subsystem clock, replace X1 and X2 pins with XT1 and XT2 pins, respectively. In addition, insert a resistor in series to the XT2 side.

5.4.3 High-speed on-chip oscillator

The high-speed on-chip oscillator is incorporated. The frequency can be selected from among 16, 8, 4, 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.

5.4.4 Low-speed on-chip oscillator

The low-speed on-chip oscillator is incorporated.

The low-speed on-chip oscillator clock is used only as the watchdog timer, real-time clock 2, and 12-bit interval timer clock. The low-speed on-chip oscillator clock cannot be used as the CPU clock.

The low-speed on-chip oscillator runs while the watchdog timer is operating or when bit 4 (WUTMMCK0) in the operation speed mode control register (OSMC) is set to 1.

The low-speed on-chip oscillator is stopped when the watchdog timer is stopped and WUTMMCK0 is set to 0.

5.5 Clock Generator Operation

The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode (see **Figure 5-1**).

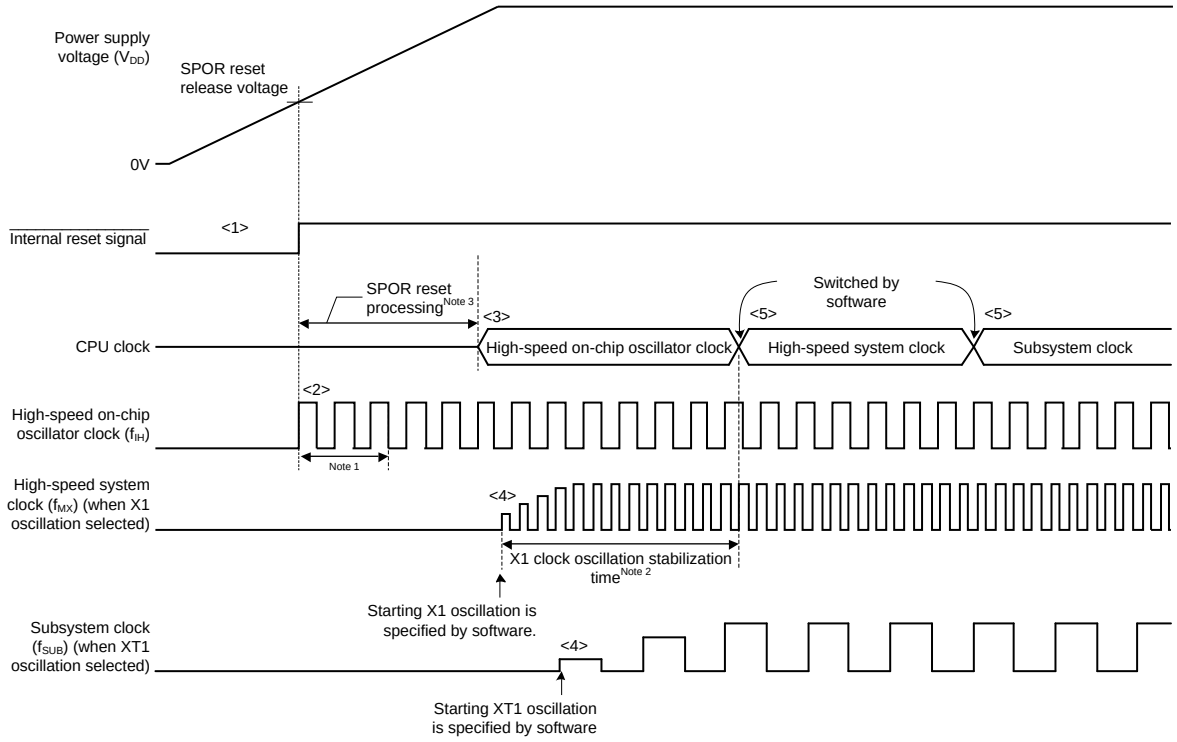
- Main system clock (f_{MAIN})
  - High-speed system clock[Note 1] (f_{MX})
    X1 clock[Note 1] (f_X)
    External main system clock[Note 1] (f_{EX})
  - High-speed on-chip oscillator clock (f_{IH})
- Subsystem clock[Note 1] (f_{SUB})
  - XT1 clock[Note 1] (f_{XT})
  - External subsystem clock[Note 1] (f_{EXS})
- 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.

When the power supply voltage is turned on, the clock generator operation is shown in **Figure 5-15**.

Note 1. 16-, 20-, and 24-pin products and 32-pin products

Figure 5-15. Clock Generator Operation When Power Supply Voltage Is Turned On



<1> When the power is turned on, an internal reset signal is generated by the selectable power-on-reset (SPOR) circuit.

<2> When the power supply voltage exceeds detection voltage of the SPOR circuit, the reset is released and the high-speed on-chip oscillator automatically starts oscillation.

<3> The CPU starts operation on the high-speed on-chip oscillator clock after waiting for the voltage to stabilize and an SPOR reset processing have been performed after reset release.

<4> Set the start of oscillation of the X1 clock or XT1 clock via software (see **5.6.2 Example of setting X1 oscillation clock** and **5.6.3 Example of setting XT1 oscillation clock**).

<5> When switching the CPU clock to the X1 clock or the XT1 clock, wait for the clock oscillation to stabilize, and then switch the clock via software (see **5.6.2 Example of setting X1 oscillation clock** and **5.6.3 Example of setting XT1 oscillation clock**).

Note 1. The 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 SPOR reset processing time, see **CHAPTER 19 SELECTABLE POWER-ON-RESET CIRCUIT**.

Caution When an external clock input from the EXCLK pin is in use, oscillation stabilization time is unnecessary.

5.6 Controlling Clock

5.6.1 Example of setting high-speed on-chip oscillator

After a reset release, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock. The frequency of the high-speed on-chip oscillator can be selected by using FRQSEL0 to FRQSEL2 of the option byte (000C2H). This frequency can be changed with the high-speed on-chip oscillator frequency select register (HOCODIV).

[Option byte setting]

Address: 000C2H

Option byte (000C2H)

7	6	5	4	3	2	1	0
1	1	1	1	1	FRQSEL2	FRQSEL1	FRQSEL0

FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator
0	0	1	16 MHz
0	1	0	8 MHz
0	1	1	4 MHz
1	0	0	2 MHz
1	0	1	1 MHz
Other than above			Setting prohibited

[High-speed on-chip oscillator frequency select register (HOCODIV) setting]

Address: F00A8H

7	6	5	4	3	2	1	0
0	0	0	0	0	HOCODIV 2	HOCODIV 1	HOCODIV 0

HOCODIV 2	HOCODIV 1	HOCODIV 0	Selected frequency
0	0	1	16 MHz
0	1	0	8 MHz
0	1	1	4 MHz
1	0	0	2 MHz
1	0	1	1 MHz
Other than above			Setting prohibited

Caution 1. Set the HOCODIV register within the operable voltage range before and after the frequency change.

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**

5.6.2 Example of setting X1 oscillation clock

After a reset release, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock. To subsequently change the clock to the X1 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 clock to f_{CLK} by using the system clock control register (CKC).

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

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

CMC

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

AMPH bit: Set this bit to 0 if the X1 clock is 10 MHz or less.

<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 104 µs is set based on a 10 MHz resonator.

OSTS

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

<3> Clear (0) the MSTOP bit of the CSC register to start oscillating the X1 oscillator.

CSC

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

<4> Use the OSTC register to wait for oscillation of the X1 oscillator to stabilize.

Example: Wait until the bits reach the following values when a wait of at least 104 µs is set based on a 10 MHz resonator.

OSTC

7	6	5	4	3	2	1	0
MOST8	MOST9	MOST10	MOST11	MOST13	MOST15	MOST17	MOST18
1	1	1	0	0	0	0	0

<5> Use the MCM0 bit of the CKC register to specify the X1 clock as the CPU/peripheral hardware clock.

CKC

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

5.6.3 Example of setting XT1 oscillation clock

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

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

<1> To run only the real-time clock and 12-bit interval timer on the subsystem clock (ultra-low current consumption) when in the STOP mode or HALT mode during CPU operation on the subsystem clock, set the RTCLPC bit to 1.

	7	6	5	4	3	2	1	0
OSMC	RTCLPC			WUTMMCK0				
	0/1	0	0	0	0	0	0	0

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

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

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

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

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

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

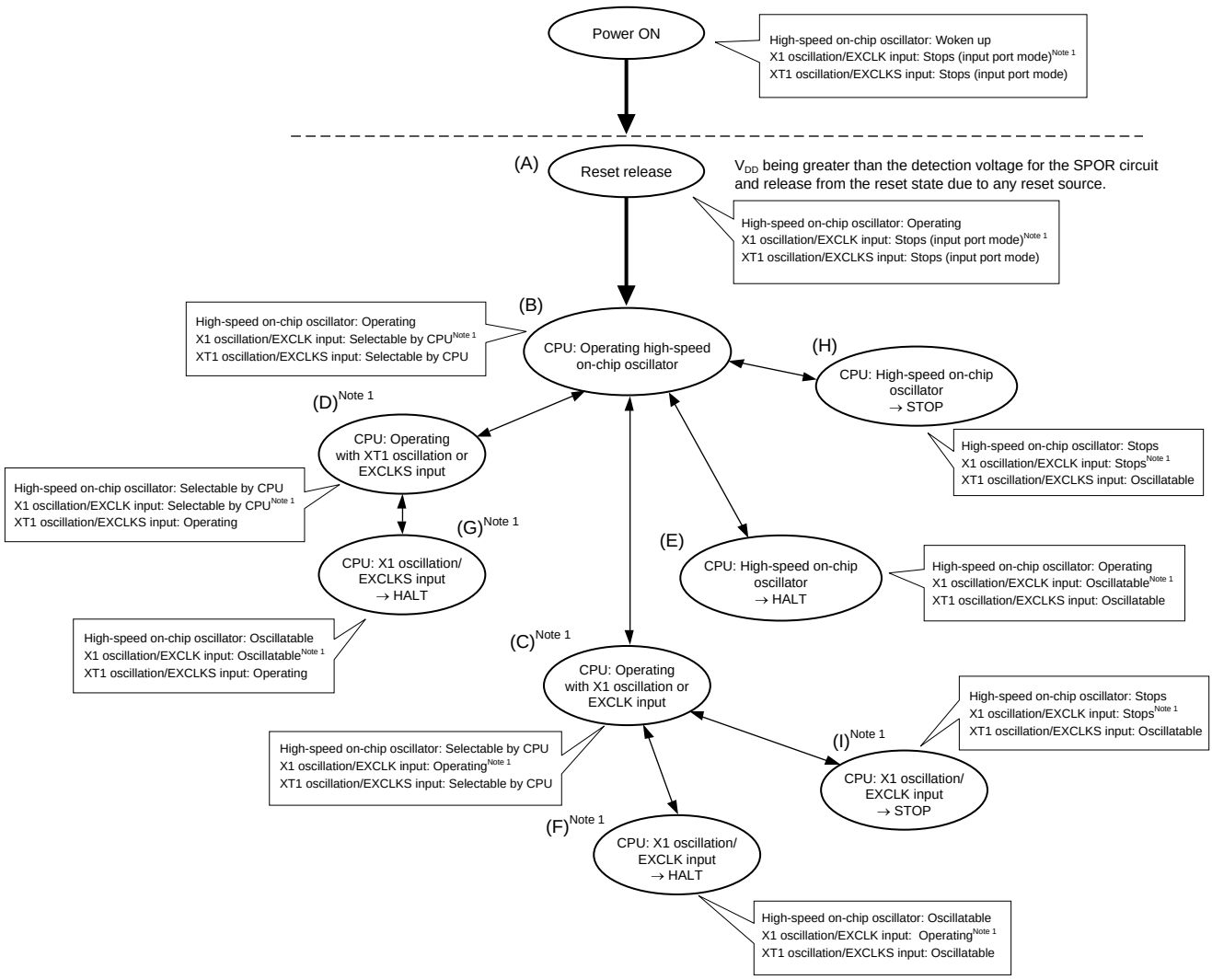
<5> Use the CSS bit of the CKC register to specify the XT1 oscillation clock as the CPU/peripheral hardware clock.

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

5.6.4 CPU clock status transition diagram

Figure 5-16 shows the CPU clock status transition diagram of this product.

Figure 5-16. CPU Clock Status Transition Diagram



Note 1. 16-, 20-, and 24-pin products and 32-pin products

Table 5-3 shows transition of the CPU clock and examples of setting the SFR registers.

Table 5-3. CPU Clock Transition and SFR Register Setting Examples for 16-, 20-, and 24-Pin Products and 32-Pin Products (1/2)

(1) CPU clock changing to high-speed on-chip oscillator clock (B) after a reset release (A)

Status Transition	SFR Register Setting
(A) → (B)	SFR registers do not have to be set (default status after a reset release).

(2) CPU clock changing to high-speed system clock (C) after a reset release (A) (The CPU operates with the high-speed on-chip oscillator clock immediately after a reset release (B).)

(Setting sequence of SFR registers) ⟶

Setting Flag of SFR Register / Status Transition	CMC Register Note 1			OSTS Register	CSC Register	OSTC Register	CKC Register
	EXCLK	OSCSEL	AMPH		MSTOP		MCM0
(A) → (B) → (C) (X1 clock: 1 MHz ≤ f_X ≤ 10 MHz)	0	1	0	Note 2	0	Must be checked	1
(A) → (B) → (C) (X1 clock: 10 MHz < f_X ≤ 12 MHz)	0	1	1	Note 2	0	Must be checked	1
(A) → (B) → (C) (External main system clock)	1	1	×	Note 2	0	Must be checked	1

Note 1. The clock operation mode control register (CMC) can be written only once by an 8-bit memory manipulation instruction after reset release. Set XTSEL to 0.

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 26 ELECTRICAL SPECIFICATIONS (T_A = −40 to +85°C) and CHAPTER 27 ELECTRICAL SPECIFICATIONS) (T_A = −40 to +105°C, T_A = −40 to +125°C).

(3) CPU clock changing to subsystem clock (D) after a reset release (A) (The CPU operates with the high-speed on-chip oscillator clock immediately after a reset release (B).)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register[Note 1]				OSTS Register	Waiting for Oscillation Stabilization	CKC register
	EXCLKS	OSCSELS	AMPHS1	AMPHS0	XTSTOP		CSS
(A) → (B) → (D) (XT1 clock)	0	1	0/1	0/1	0	Necessary	1
(A) → (B) → (D) (external sub clock)	1	1	×	×	0	Necessary	1

Note 1. The clock operation mode control register (CMC) can be written only once by an 8-bit memory manipulation instruction after a reset release. Set XTSEL to 1.

Remark 1. ×: Don't care

Remark 2. (A) to (I) in **Table 5-3** correspond to (A) to (I) in **Figure 5-16**.

(4) CPU clock changing from high-speed on-chip oscillator clock (B) to high-speed system clock (C)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register[Note 1]			OSTS Register	CSC Register	OSTC Register	CKC Register
	EXCLK	OSCSEL	AMPH		MSTOP		MCM0
(B) → (C) (X1 clock: 1 MHz ≤ f_X ≤ 10 MHz)	0	1	0	Note 2	0	Must be checked	1
(B) → (C) (X1 clock: 10 MHz < f_X ≤ 12 MHz)	0	1	1	Note 2	0	Must be checked	1
(B) → (C) (external main clock)	1	1	×	Note 2	0	Must be checked	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 written only once by an 8-bit memory manipulation instruction after a reset release.
This setting is not necessary if it has already been set. Set XTSEL to 0.

Note 2. Set the oscillation stabilization time of the oscillation stabilization time select register (OSTS) 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 26 ELECTRICAL SPECIFICATIONS (T_A = −40 to +85°C) and CHAPTER 27 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C, T_A = −40 to +125°C)).

(5) CPU clock changing from high-speed on-chip oscillator clock (B) to subsystem clock (D)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register[Note 1]			OSTS register	Waiting for Oscillation Stabilization	CKC register
	EXCLKS	OSCSELS	AMPHS1,0	XTSTOP		CSS
(B) → (D) (XT1 clock)	0	1	00: Low power consumption oscillation 01: Normal oscillation 10: Ultra-low power consumption oscillation	0	Necessary	1
(B) → (D) (external sub clock)	1	1	×	0	Necessary	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 written only once by an 8-bit memory manipulation instruction after a reset release.
This setting is not necessary if it has already been set. Set XTSEL to 0.

Remark 1. ×: Don't care

Remark 2. (A) to (I) in **Table 5-3** correspond to (A) to (I) in **Figure 5-16**.

Table 5-3. CPU Clock Transition and SFR Register Setting Examples for 16-, 20-, and 24-Pin Products and 32-Pin Products (2/2)

(6) CPU clock changing from high-speed system clock (C) to high-speed on-chip oscillator clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CSC Register	Oscillation accuracy stabilization time	CKC Register
	HIOSTOP		MCM0
(C) → (B)	0	27 μs (typ.)	0

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

Remark The clock supply stop time varies depending on the temperature conditions and STOP mode period.

(7) CPU clock changing from subsystem clock (D) to high-speed on-chip oscillator clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CSC register	Oscillation accuracy stabilization time	CKC Register
	HIOSTOP		CSS
(D) → (B)	0	27μs (typ.)	0

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

Remark 1. (A) to (I) in **Table 5-3** correspond to (A) to (I) in **Figure 5-16**.

Remark 2. The clock supply stop time varies depending on the temperature conditions and STOP mode period.

(8) HALT mode (E) set while CPU is operating with high-speed on-chip oscillator clock (B)
HALT mode (F) set while CPU is operating with high-speed system clock (C)
HALT mode (G) set while CPU is operating with subsystem clock (D)

Status Transition	Setting
(B) → (E) (C) → (F) (D) → (G)	Executing HALT instruction

Remark (A) to (I) in **Table 5-3** correspond to (A) to (I) in **Figure 5-16**.

(9) STOP mode (H) set while CPU is operating with high-speed on-chip oscillator clock (B)
STOP mode (I) set while CPU is operating with high-speed system clock (C)

(Setting sequence of SFR registers) →

Status Transition		Setting		
(B) → (H)		Stopping peripheral functions that cannot operate in STOP mode	—	Executing STOP instruction
(C)→ (I)	In X1 oscillation		Sets the OSTS register	
	External clock		—	

Remark (A) to (I) in **Table 5-3** correspond to (A) to (I) in **Figure 5-16**.

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

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

Table 5-4. Changing CPU Clock

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
High-speed on-chip oscillator clock	X1 clock	Stabilization of X1 oscillation ● OSCSEL = 1, EXCLK = 0, MSTOP = 0, XTSEL = 0 ● After elapse of oscillation stabilization time	The operating current can be reduced by stopping the high-speed on-chip oscillator (HIOSTOP = 1) after checking that the CPU clock is changed.
	External main system clock	An external clock input from the EXCLK pin must be enabled. ● OSCSEL = 1, EXCLK = 1, MSTOP = 0, XTSEL = 0	
	XT1 clock	Stabilization of XT1 oscillation ● OSCSELS = 1, EXCLKS = 0, XTSTOP = 0, XTSEL = 1 ● After elapse of oscillation stabilization time	
	External subsystem clock	Enabling input of external clock from the EXCLKS pin ● OSCSELS = 1, EXCLKS = 1, XTSTOP = 0, XTSEL = 1	
X1 clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator ● HIOSTOP = 0 ● After elapse of oscillation accuracy stabilization time	X1 oscillation can be stopped (MSTOP = 1) after checking that the CPU clock is changed.
	External main system clock	Change is not possible	—
External main system clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator ● HIOSTOP = 0 ● After elapse of oscillation accuracy stabilization time	External main system clock input can be disabled (MSTOP = 1) after checking that the CPU clock is changed.
	X1 clock	Change is not possible	—

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

By setting bits 4 and 6 (MCM0, CSS) of the system clock control register (CKC), the CPU clock can be switched (between the main system clock and the subsystem clock), and main system clock can be switched (between the main system clock and the subsystem clock) and (between the high-speed on-chip oscillator clock and the high-speed system 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 **Table 5-5** to **Table 5-7**).

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 high-speed on-chip oscillator clock can be ascertained using bit 5 (MCS) of the CKC register.

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

Table 5-5. Maximum Time Required for System Clock Switchover

Clock A	Switching directions	Clock B	Remark
f_{IH}	←→	f_{MX}	See **Table 5-6**
f_{MAIN}	←→	f_{SUB}	See **Table 5-7**

Table 5-6. Maximum Number of Clocks Required for Main System Clock Switchover ($f_{IH} \leftrightarrow f_{MX}$)

Set Value Before Switchover		Set Value After Switchover	
MCM0		MCM0	
		0 ($f_{MAIN} = f_{IH}$)	1 ($f_{MAIN} = f_{MX}$)
0 ($f_{MAIN} = f_{IH}$)	$f_{MX} \geq f_{IH}$	—	$1 + f_{IH}/f_{MX}$
	$f_{MX} < f_{IH}$	—	$2f_{IH}/f_{MX}$ clock
1 ($f_{MAIN} = f_{MX}$)	$f_{MX} \geq f_{IH}$	$2f_{MX}/f_{IH}$ clock	—
	$f_{MX} < f_{IH}$	$1 + f_{MX}/f_{IH}$	—

Table 5-7. Maximum Number of Clocks Required for $f_{MAIN} \leftrightarrow f_{SUB}$

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

Remark 1. The number of clocks listed in **Table 5-6** and **Table 5-7** is the number of CPU clocks before switchover.

Remark 2. Calculate the number of clocks by rounding to the nearest whole number.
Example) When switching the main system clock from the high-speed system clock to the high-speed on-chip oscillator clock (@ oscillation with f_{IH} = 8 MHz selected, f_{MX} = 10 MHz)
$2f_{MX}/f_{IH} = 2\ (10/8) = 2.5 \rightarrow 3$ clocks

5.6.7 Conditions before clock oscillation is stopped

The following lists the register flag settings for stopping the clock oscillation (disabling external clock input) and conditions before the clock oscillation is stopped. Before stopping the clock oscillation, check the conditions before the clock oscillation is stopped.

Table 5-8. Conditions Before the Clock Oscillation Is Stopped and Flag Settings

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

5.7 Resonator and Oscillator Constants

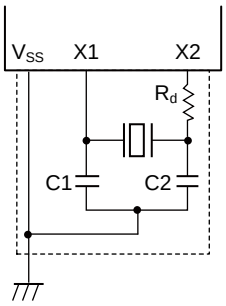
For the resonators for which the operation is verified and their oscillator constants, refer to the target product page of the Renesas Electronics website.

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.

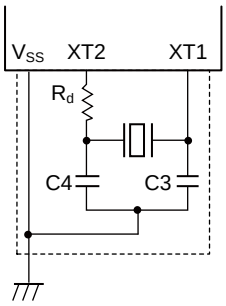
Caution 2. **The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the RL78 microcontroller so that the internal operation conditions are within the specifications of the DC and AC characteristics.**

Figure 5-17. External Oscillation Circuit Example

(a) X1 oscillation



(b) XT1 oscillation



Remark 16-, 20-, and 24-pin products and 32-pin products

CHAPTER 6 TIMER ARRAY UNIT

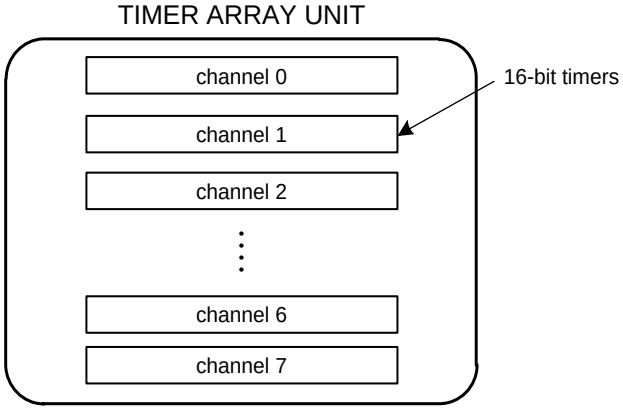
Units	Channels	32, 24, 20, 16, 10-pin
Unit 0	Channel 0	✓
	Channel 1	✓
	Channel 2	✓
	Channel 3	✓
	Channel 4	✓
	Channel 5	✓
	Channel 6	✓
	Channel 7	✓

Caution 1. The presence or absence of timer I/O pins depends on the product. See Table 6-2 Timer I/O Pins provided in Each Product for details.

Caution 2. Most of the following descriptions in this chapter use the 32-pin products as an example.

The timer array unit has eight 16-bit timers.

Each 16-bit timer is called a channel and can be used as an independent timer. In addition, two or more “channels” can be used to create a high-accuracy timer.



For details about each function, see the table below.

Independent channel operation function	Simultaneous channel operation function
• Interval timer (→ refer to **6.8.1 Operation as interval timer/square wave output**.) • Square wave output (→ refer to **6.8.1 Operation as interval timer/square wave output**.) • External event counter (→ refer to **6.8.2 Operation as external event counter**.) • Divider[Note 1] (→ refer to **6.8.3 Operation as frequency divider (channels 0 and 3 only)**.) • Input pulse interval measurement (→ refer to **6.8.4 Operation as input pulse interval measurement**.) • Measurement of high-/low-level width of input signal (→ refer to **6.8.5 Operation as input signal high-/low-level width measurement**.) • Delay counter (→ refer to **6.8.6 Operation as delay counter**.)	• One-shot pulse output (→ refer to **6.9.1 Operation as one-shot pulse output function**.) • PWM output (→ refer to **6.9.2 Operation as PWM function**.) • Multiple PWM output (→ refer to **6.9.3 Operation as multiple PWM output function**.) • Two-channel input with one-shot pulse output function (→ refer to **6.9.4 Operation as two-channel input with one-shot pulse output function**.)

Note 1. Channels 0 and 3 only.

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)

Interlinked operation of channel 1 with the serial array unit operating as UART0 can be obtained by setting the ISC register. The input pulse interval measurement mode can then be used to measure the width at the baud rate of the other party in communications and make the required adjustments in response.

6.1 Functions of Timer Array Unit

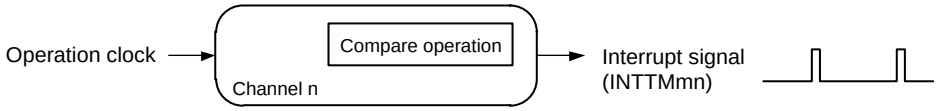
Timer array unit has the following functions.

6.1.1 Independent channel operation function

By operating a channel independently, it can be used for the following purposes without being affected by the operation mode of other channels.

1) Interval timer

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

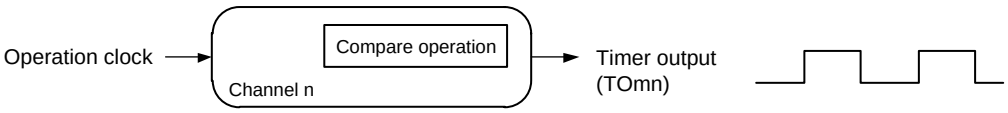


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Remark 2. The presence or absence of timer I/O pins of channel 0 to 7 depends on the product. See **Table 6-2 Timer I/O Pins provided in Each Product** for details.

2) Square wave output

A toggle operation is performed each time an INTTMmn interrupt is generated and a square wave with a duty factor of 50% is output from a timer output pin (TOmn).

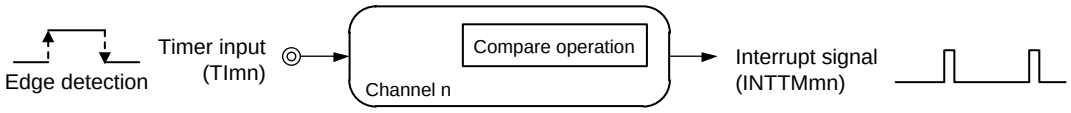


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Remark 2. The presence or absence of timer I/O pins of channel 0 to 7 depends on the product. See **Table 6-2 Timer I/O Pins provided in Each Product** for details.

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.

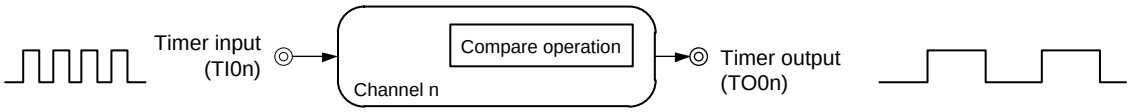


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Remark 2. The presence or absence of timer I/O pins of channel 0 to 7 depends on the product. See **Table 6-2 Timer I/O Pins provided in Each Product** for details.

4) Divider function (channels 0 and 3 only)

A clock input from a timer input pin (TI0n) is divided and output from an output pin (TO0n).

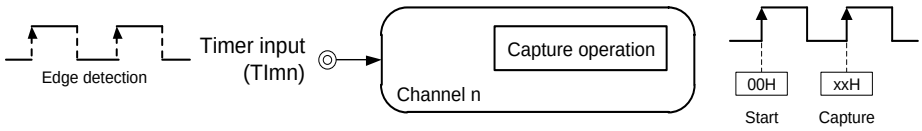


Remark 1. n: Channel number (n = 0, 3)

Remark 2. The presence or absence of timer I/O pins of channels 0 and 3 depends on the product. See **Table 6-2 Timer I/O Pins provided in Each Product** for details.

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.

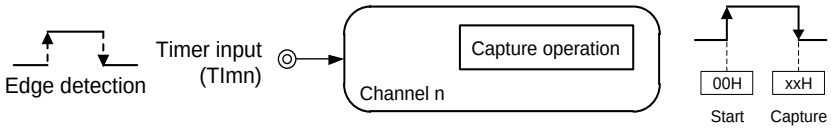


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Remark 2. The presence or absence of timer I/O pins of channel 0 to 7 depends on the product. See **Table 6-2 Timer I/O Pins provided in Each Product** for details.

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.

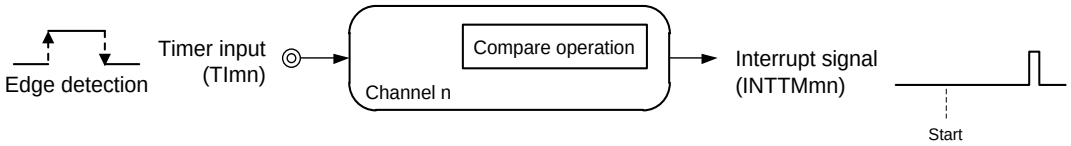


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Remark 2. The presence or absence of timer I/O pins of channel 0 to 7 depends on the product. See **Table 6-2 Timer I/O Pins provided in Each Product** for details.

7) Delay counter

Counting is started at the valid edge of the signal input to the timer input pin (TImn), and an interrupt is generated after any delay period.



Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

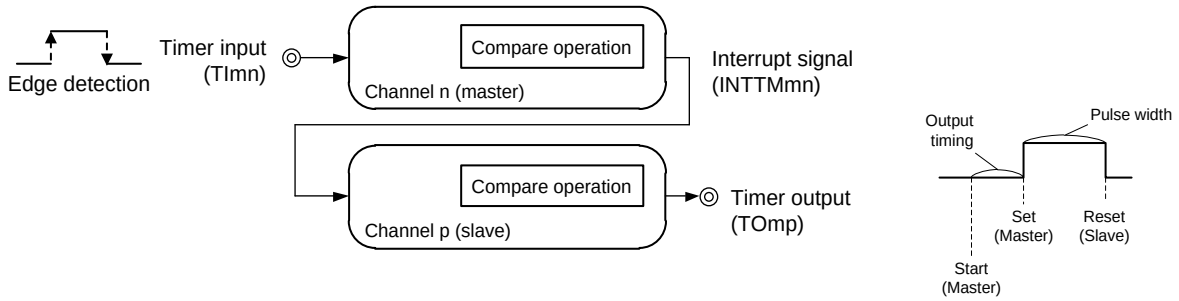
Remark 2. The presence or absence of timer I/O pins of channel 0 to 7 depends on the product. See **Table 6-2 Timer I/O Pins provided in Each Product** for details.

6.1.2 Simultaneous channel operation function

By using the combination of a master channel (a reference timer mainly controlling the cycle) and slave channels (timers operating according to the master channel), channels can be used for the following purposes.

1) One-shot pulse output

Two channels are used as a set to generate a one-shot pulse with a specified output timing and a specified pulse width.

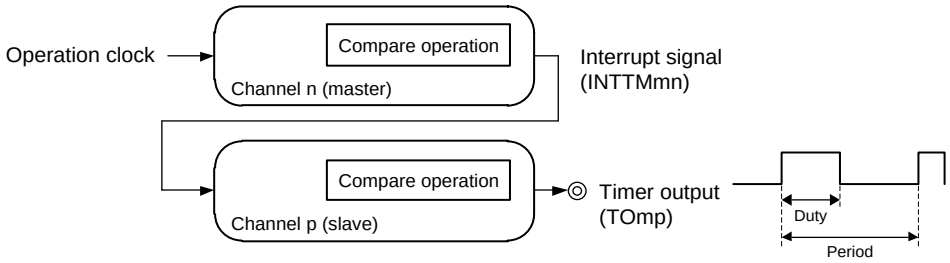


Caution For details about the rules of simultaneous channel operation function, see 6.4.1 Basic rules of simultaneous channel operation function.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)
p: Slave channel number (n < p ≤ 7)

2) PWM (Pulse Width Modulation) output

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

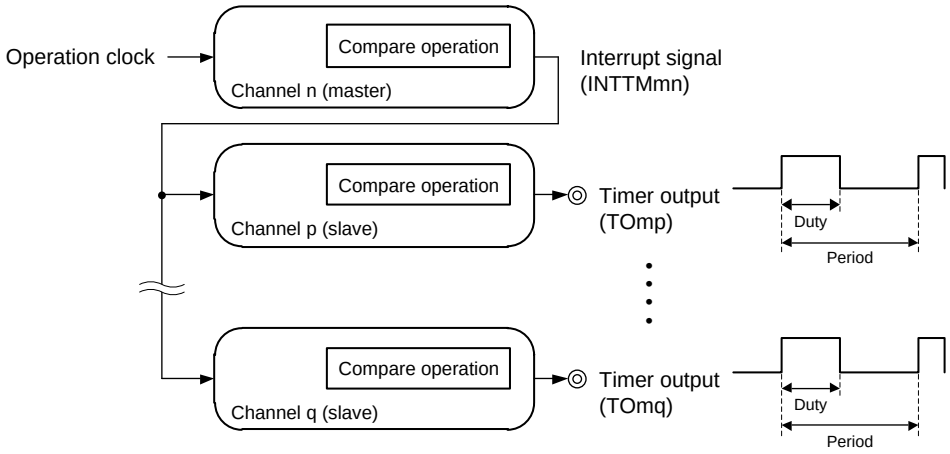


Caution For details about the rules of simultaneous channel operation function, see 6.4.1 Basic rules of simultaneous channel operation function.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)
p: Slave channel number (n < p ≤ 7)

3) Multiple PWM (Pulse Width Modulation) output

By extending the PWM function and using one master channel and two or more slave channels, up to seven types of PWM signals that have a specific period and a specified duty factor can be generated.

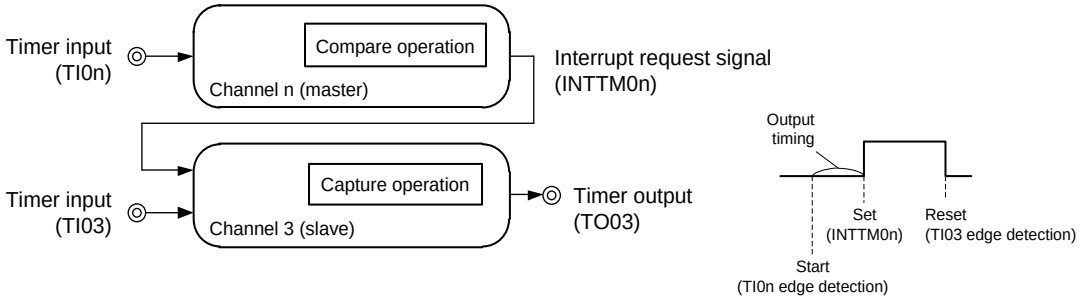


Caution For details about the rules of simultaneous channel operation function, see 6.4.1 Basic rules of simultaneous channel operation function.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)
p, q: Slave channel number (n < p < q ≤ 7)

4) 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 (TI0n, TI03) input.



Caution There are several rules for using the simultaneous channel operation function.
For details, see 6.4.1 Basic rules of simultaneous channel operation function.

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

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

The 8-bit timer operation function makes it possible to use a 16-bit timer channel in a configuration consisting of two 8-bit timer channels. This function can only be used for channels 1 and 3.

Caution There are several rules for using 8-bit timer operation function.
For details, see 6.4.2 Basic rules of 8-bit timer operation function (channels 1 and 3 only).

6.2 Configuration of Timer Array Unit

Timer array unit includes the following hardware.

Table 6-1. Configuration of Timer Array Unit

Item	Configuration
Timer/counter	Timer count register mn (TCRmn)
Register	Timer data register mn (TDRmn)
Timer input	TI00 to TI07[Note 1]
Timer output	TO00 to TO07[Note 1], output controller
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Timer clock select register m (TPSm) • Timer channel enable status register m (TEm) • Timer channel start register m (TSm) • Timer channel stop register m (TTm) • Timer output enable register m (TOEm) • Timer output register m (TOm) • Timer output level register m (TOLm) • Timer output mode register m (TOMm)
	<Registers of each channel> • Timer mode register mn (TMRmn) • Timer status register mn (TSRmn) • Input switch control register (ISC) • Noise filter enable registers 1 (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 7 depends on the product. See **Table 6-2 Timer I/O Pins provided in Each Product**.

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 **4.5.3 Register setting examples for used port and alternate functions**.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

The presence or absence of timer I/O pins in each timer array unit channel depends on the product.

Table 6-2. Timer I/O Pins provided in Each Product

Timer array unit channels		I/O Pins of Each Product				
		32-pin	24-pin	20-pin	16-pin	10-pin
Unit 0	Channel 0	P137/TI00 P03/TO00/(TI00) P20/(TI00)/(TO00) P42/(TI00) P21/(TO00)	P137/TI00 P03/TO00/(TI00) P20/(TI00)/(TO00) P21/(TO00)		P137/TI00 P03/TO00/(TI00)	
	Channel 1	P02/TI01/TO01 P01/(TI01/TO01) P04/(TI01/TO01) P40/(TI01/TO01)				
	Channel 2	P01/TI02/TO02 P00/(TI02/TO02) P02/(TI02/TO02) P05/(TI02/TO02) P41/(TI02/TO02)				P01/TI02/TO02 P00/(TI02/TO02)
	Channel 3	P41/TI03/TO03 P06/(TI03/TO03) P07/(TO03) P10/(TI03/TO03) P11/(TI03/TO03) P16/(TI03/TO03) P20/(TI03/TO03)	P41/TI03/TO03 P06/(TI03/TO03) P07/(TO03) P10/(TI03/TO03) P11/(TI03/TO03) P20/(TI03/TO03)	P41/TI03/TO03 P06/(TI03/TO03) P07/(TO03) P20/(TI03/TO03)	P41/TI03/TO03 P06/(TI03/TO03) P07/(TO03)	—
	Channel 4	P07/TI04/TO04 P17/(TI04/TO04) P23/(TI04/TO04)	P07/TI04/TO04 P23/(TI04/TO04)		P07/TI04/TO04	—
	Channel 5	P122/TI05/TO05 P03/(TI05/TO05)				—
	Channel 6	P04/TI06/TO06 P22/(TI06/TO06)			P04/TI06/TO06	—
	Channel 7	P121/TI07/TO07 P05/(TI07/TO07)				—

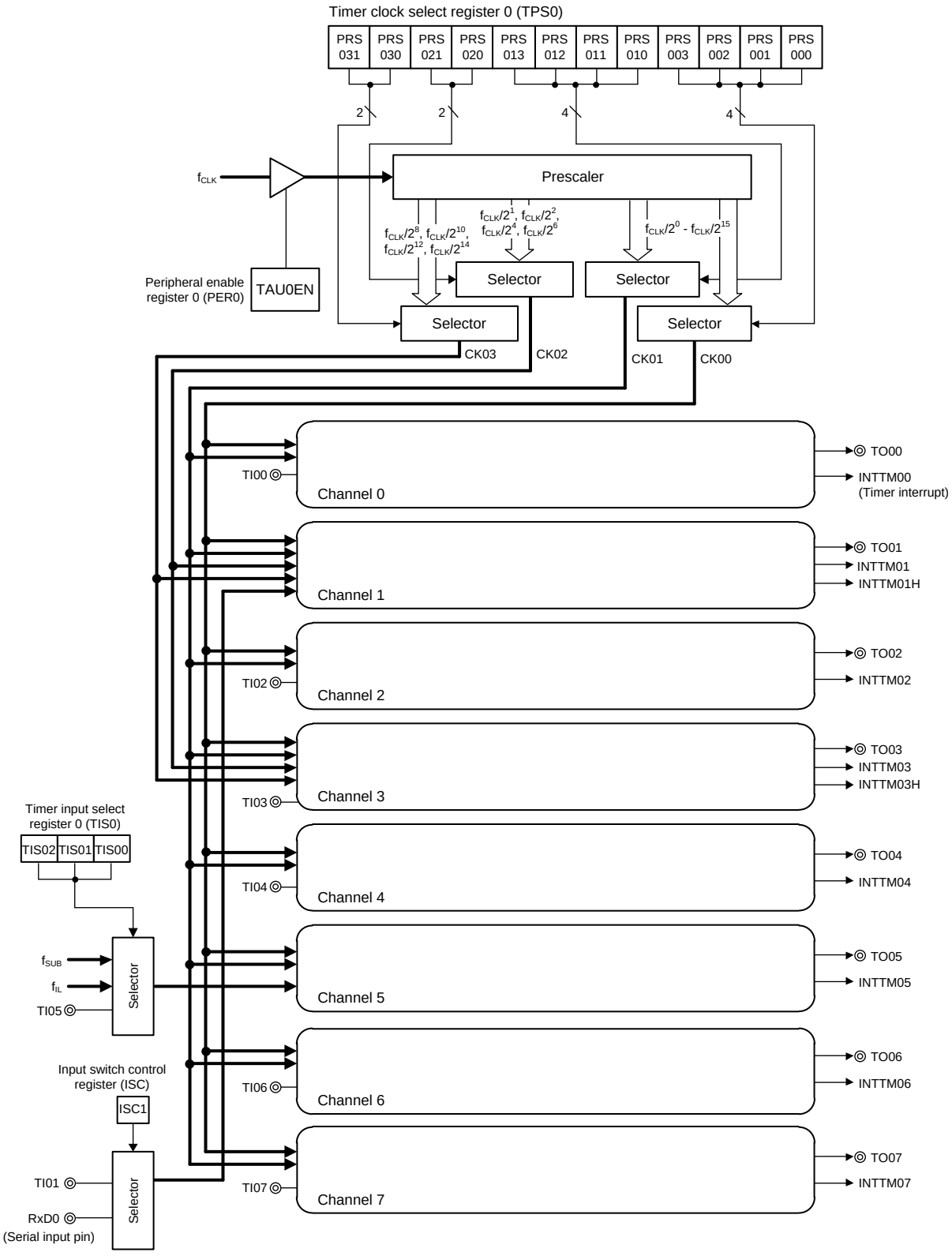
Remark 1. When timer input and timer output are shared by the same pin, either only timer input or only timer output can be used.

Remark 2. —: There is no timer I/O pin, but the channel is available. (However, the channel can only be used as an interval timer.)
×: The channel is not available.

Remark 3. Pins in the parentheses indicate an alternate port when setting the peripheral I/O redirection register 0 to 6 (PIOR0 to PIOR6). For details, see **4.3.6 Peripheral I/O redirection registers 0 to 6 (PIOR0 to PIOR6)**.

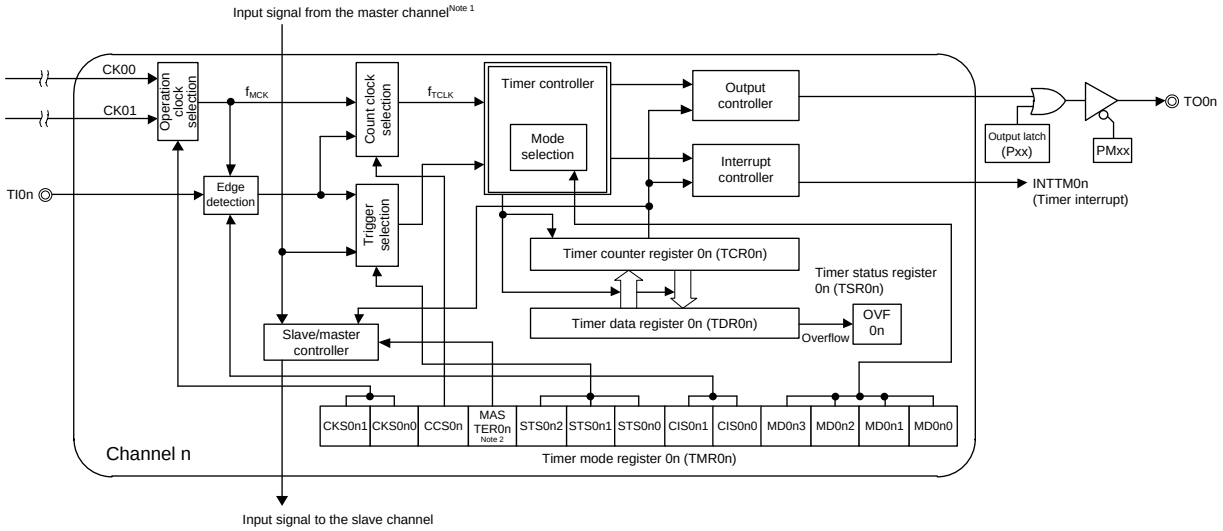
Figure 6-1 shows the block diagrams of the timer array unit.

Figure 6-1. Entire Block Diagram of Timer Array Unit



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

Figure 6-2. Internal Block Diagram of Channels 0, 2, 4, 6 of Timer Array Unit

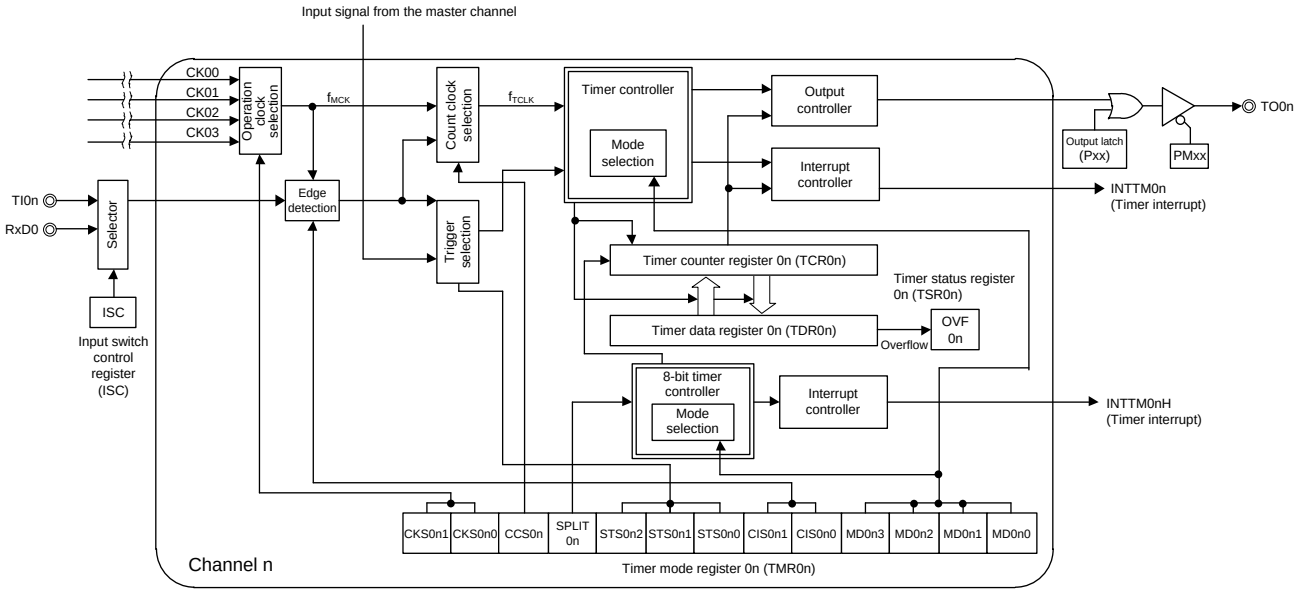


Note 1. Channels 2, 4, and 6 only

Note 2. n = 2, 4, 6 only

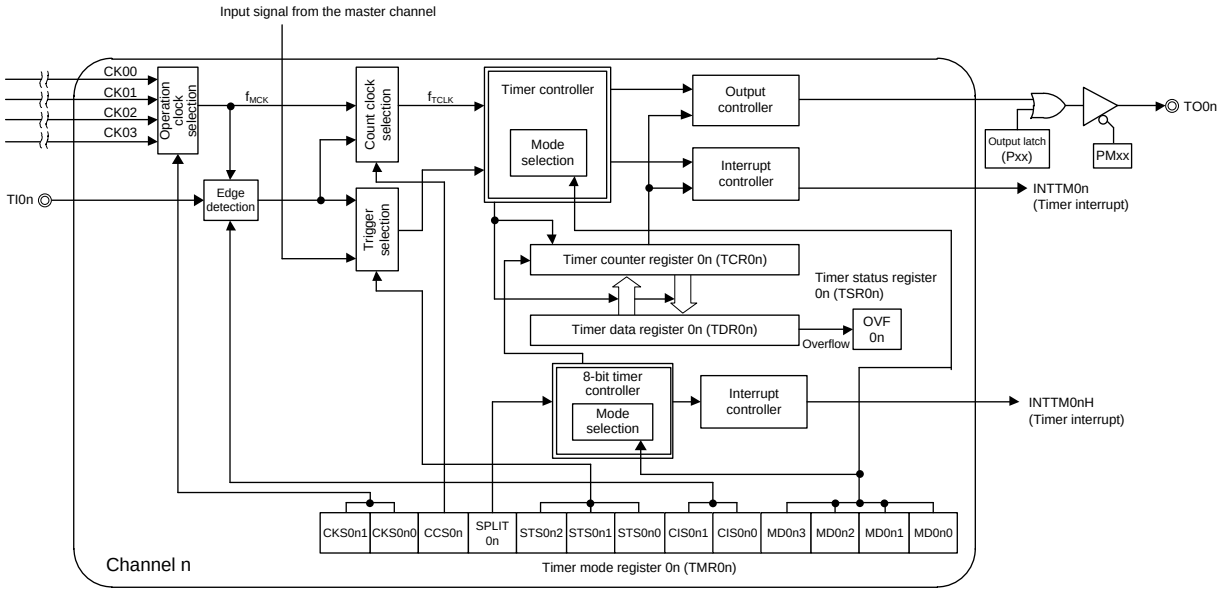
Remark n = 0, 2, 4, 6

Figure 6-3. Internal Block Diagram of Channel 1 of Timer Array Unit



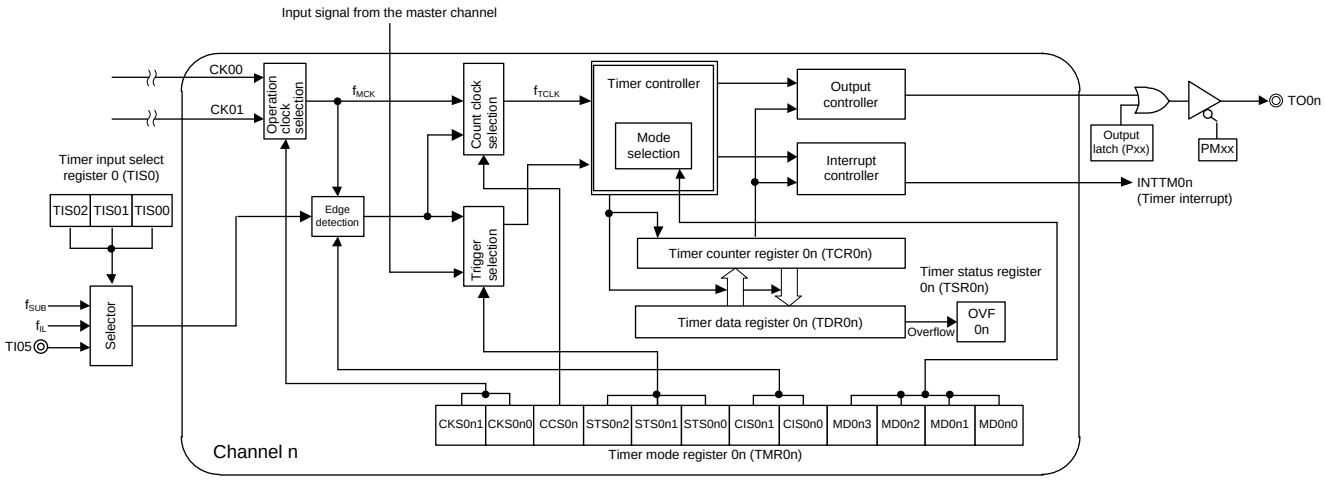
Remark n = 1

Figure 6-4. Internal Block Diagram of Channel 3 of Timer Array Unit



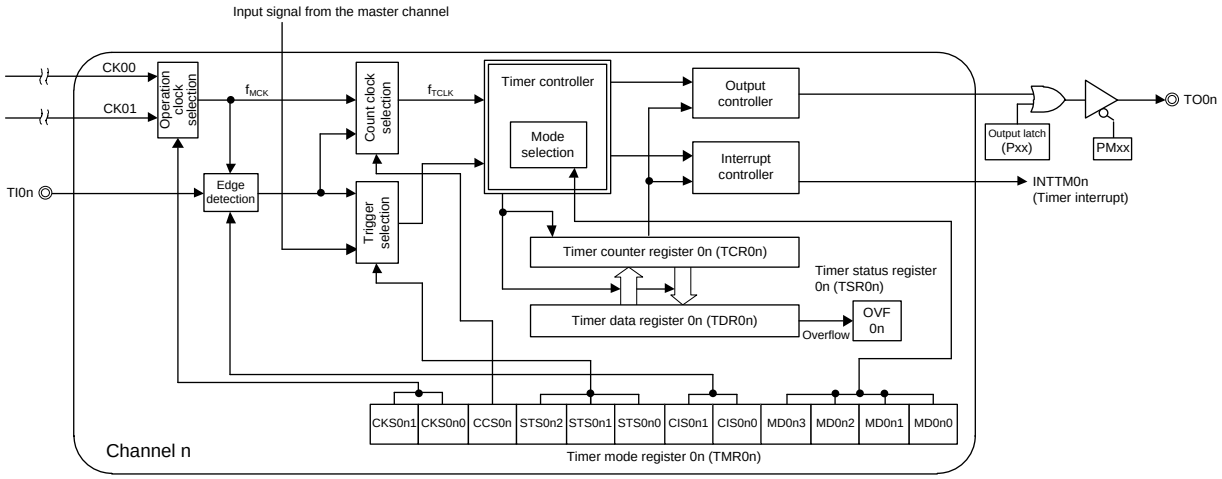
Remark n = 3

Figure 6-5. Internal Block Diagram of Channel 5 of Timer Array Unit



Remark n = 5

Figure 6-6. Internal Block Diagram of Channel 7 of Timer Array Unit



Remark n = 7

6.2.1 Timer count register mn (TCRmn)

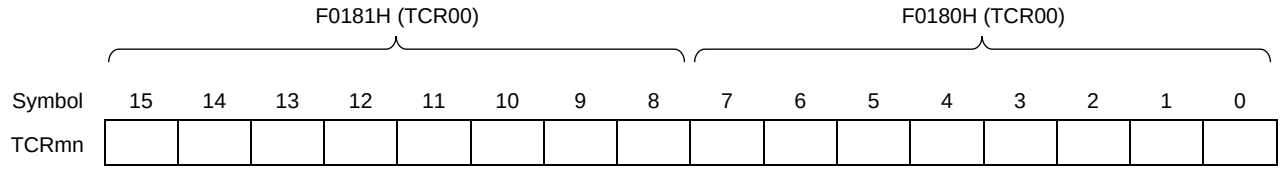
The TCRmn register is a 16-bit read-only register and is used to count clocks.

The value of this counter is incremented or decremented in synchronization with the rising edge of a count clock.

Whether the counter is incremented or decremented depends on the operation mode that is selected by the MDmn3 to MDmn0 bits of timer mode register mn (TMRmn) (refer to **6.3.3 Timer mode register mn (TMRmn)**).

Figure 6-7. Format of Timer Count Register mn (TCRmn)

Address: F0180H, F0181H (TCR00) to F018EH, F018FH (TCR07), F01C0H, F01C1H (TCR10) to F01CEH, F01CFH (TCR17)
After reset: FFFFH R



Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

The count value can be read by reading timer count register mn (TCRmn).

The count value is set to FFFFH in the following cases.

- When the reset signal is generated
- When the TAUmEN bit of peripheral enable register 0 (PER0) is cleared
- When counting of the slave channel has been completed in the PWM output mode
- When counting of the slave channel has been completed in the delay count mode
- When counting of the master/slave channel has been completed in the one-shot pulse output mode
- When counting of the slave channel has been completed in the multiple PWM output mode

The count value is cleared to 0000H in the following cases.

- When the start trigger is input in the capture mode
- When capturing has been completed in the capture mode

Caution The count value is not captured to timer data register mn (TDRmn) even when the TCRmn register is read.

The TCRmn register read value differs as follows according to operation mode changes and the operating status.

Table 6-3. Timer Count Register mn (TCRmn) Read Value in Various Operation Modes

Operation Mode	Count Mode	Timer count register mn (TCRmn) Read Value[Note 1]			
		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 1. 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 7)

6.2.2 Timer data register mn (TDRmn)

This is a 16-bit register from which a capture function and a compare function can be selected. The capture or compare function can be switched by selecting an operation mode by using the MDmn3 to MDmn0 bits of timer mode register mn (TMRmn).

The value of the TDRmn register can be changed at any time.

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

In addition, for the TDRm1 and TDRm3 registers, while in the 8-bit timer mode (when the SPLITm1, SPLITm3 bits of timer mode registers m1 and m3 (TMRm1, TMRm3) are 1), it is possible to read and write the data in 8-bit units, with TDRm1H and TDRm3H used as the higher 8 bits, and TDRm1L and TDRm3L used as the lower 8 bits.

Reset signal generation clears this register to 0000H.

Figure 6-8. Format of Timer Data Register mn (TDRmn) (n = 0, 2, 4 to 7)

Address: FFF18H, FFF19H (TDR00), FFF64H, FFF65H (TDR02), FFF68H, FFF69H (TDR04) to FFF6EH, FFF6FH (TDR07)
After reset: 0000H R/W

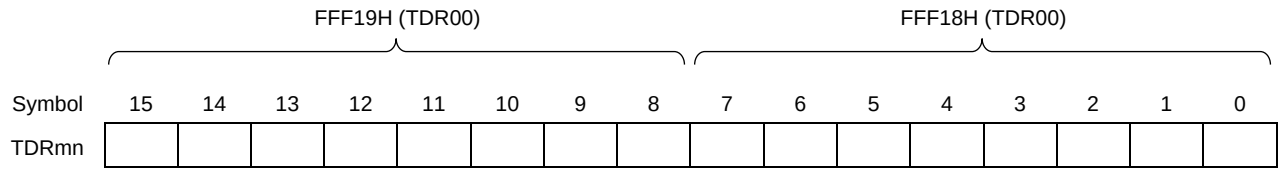
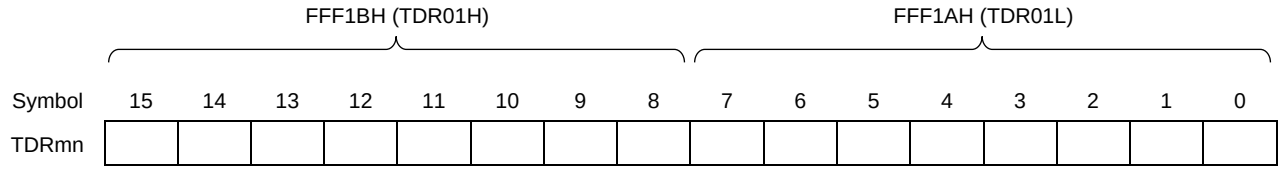


Figure 6-9. Format of Timer Data Register mn (TDRmn) (n = 1, 3)

Address: FFF1AH, FFF1BH (TDR01), FFF66H, FFF67H (TDR03) 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 7)

6.3 Registers Controlling Timer Array Unit

Timer array unit is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Timer clock select register m (TPSm)
- Timer mode register mn (TMRmn)
- Timer status register mn (TSRmn)
- Timer channel enable status register m (TEm)
- Timer channel start register m (TSm)
- Timer channel stop register m (TTm)
- Timer output enable register m (TOEm)
- Timer input select register 0 (TIS0)
- Timer output register m (TOm)
- Timer output level register m (TOLm)
- Timer output mode register m (TOMm)
- Input switch control register (ISC)
- Noise filter enable registers 1 (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 7)

6.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the timer array unit 0 is used, be sure to set bit 0 (TAU0EN) of this register to 1.

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 6-10. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PER0	TMKAEN	CMPEN	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

TAU0EN	Control of timer array unit 0 input clock
0	Stops supply of input clock. • SFR used by the timer array unit 0 cannot be written. • The timer array unit 0 is in the reset status.
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, the values of the registers which control the timer array unit are cleared to their initial values and writing to them is ignored (except for timer input select register 0 (TIS0), input switch control register (ISC), noise filter enable register 1 (NFEN1), port mode control register 0, 2 (PMC0, PMC2), port mode register 0, 1, 2, 4, 6, 12 (PM0, PM1, PM2, PM4, PM6, PM12), and port register 0, 1, 2, 4, 6, 12, 13 (P0, P1, P2, P4, P6, P12, P13)).

- **Timer clock select register m (TPSm)**
- **Timer mode register mn (TMRmn)**
- **Timer status register mn (TSRmn)**
- **Timer channel enable status register m (TEm)**
- **Timer channel start register m (TSm)**
- **Timer channel stop register m (TTm)**
- **Timer output enable register m (TOEm)**
- **Timer output register m (TOm)**
- **Timer output level register m (TOLm)**
- **Timer output mode register m (TOMm)**

Caution 2. Be sure to clear bit 1 to 0.

6.3.2 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 7):

All channels for which CKm0 is selected as the operation clock (CKSmn1, CKSmn0 = 0, 0) are stopped (TEmn = 0).

If the PRSm10 to PRSm13 bits can be rewritten (n = 0 to 7):

All channels for which CKm1 is selected as the operation clock (CKSmn1, CKSmn0 = 0, 1) are stopped (TEmn = 0).

If the PRSm20 and PRSm21 bits can be rewritten (n = 1, 3):

All channels for which CKm2 is selected as the 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 6-11. 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

PRSmk3	PRSmk2	PRSmk1	PRSmk0	Selection of operation clock (CKmk)[Note 1] (k = 0, 1)					
					f_{CLK} (MHz)				
					1	2	4	8	16
0	0	0	0	f_{CLK}	1 MHz	2 MHz	4 MHz	8 MHz	16 MHz
0	0	0	1	$f_{CLK}/2$	500 kHz	1 MHz	2 MHz	4 MHz	8 MHz
0	0	1	0	$f_{CLK}/2^2$	250 kHz	500 kHz	1 MHz	2 MHz	4 MHz
0	0	1	1	$f_{CLK}/2^3$	125 kHz	250 kHz	500 MHz	1 MHz	2 MHz
0	1	0	0	$f_{CLK}/2^4$	62.5 kHz	125 kHz	250kHz	500 kHz	1 MHz
0	1	0	1	$f_{CLK}/2^5$	31.3 kHz	62.5 kHz	125kHz	250 kHz	500 kHz
0	1	1	0	$f_{CLK}/2^6$	15.6 kHz	31.3 kHz	62.5 kHz	125 kHz	250 kHz
0	1	1	1	$f_{CLK}/2^7$	7.81 kHz	15.6 kHz	31.3 kHz	62.5 kHz	125 kHz
1	0	0	0	$f_{CLK}/2^8$	3.91 kHz	7.81 kHz	15.6 kHz	31.3 kHz	62.5 kHz
1	0	0	1	$f_{CLK}/2^9$	1.95 kHz	3.91 kHz	7.81 kHz	15.6 kHz	31.3 kHz
1	0	1	0	$f_{CLK}/2^{10}$	977 Hz	1.95 kHz	3.91 kHz	7.81 kHz	15.6 kHz
1	0	1	1	$f_{CLK}/2^{11}$	488 Hz	977 Hz	1.95 kHz	3.91 kHz	7.81 kHz
1	1	0	0	$f_{CLK}/2^{12}$	244 Hz	488 Hz	977 Hz	1.95 kHz	3.91 kHz
1	1	0	1	$f_{CLK}/2^{13}$	122 Hz	244 Hz	488 kHz	977 Hz	1.95 kHz
1	1	1	0	$f_{CLK}/2^{14}$	61 Hz	122 Hz	244 Hz	488 Hz	977 Hz
1	1	1	1	$f_{CLK}/2^{15}$	30.5 Hz	61 Hz	122 Hz	244 Hz	488 Hz

Note 1. When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop timer array unit (TTm = 00FFH).

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 (m = 0, n = 0 to 7), interrupt requests output from timer array units cannot be used.

Remark 1. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 2. Waveform of the clock to be selected in the TPS0 register which becomes high level for one period of f_{CLK} from its rising edge. For details, see **6.5.1 Count clock (fTCLK)**.

Figure 6-11. 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

PRSm21	PRSm20	Selection of operation clock (CKm2)[Note 1]					
			f_{CLK} (MHz)				
			1	2	4	8	16
0	0	$f_{CLK}/2$	500 kHz	1 MHz	2 MHz	4 MHz	8 MHz
0	1	$f_{CLK}/2^2$	250 kHz	500 kHz	1 MHz	2 MHz	4 MHz
1	0	$f_{CLK}/2^4$	62.5 kHz	125 kHz	250 kHz	500 kHz	1 MHz
1	1	$f_{CLK}/2^6$	15.6 kHz	31.3 kHz	62.5 kHz	125 kHz	250 kHz

PRSm31	PRSm30	Selection of operation clock (CKm3)[Note 1]					
			f_{CLK} (MHz)				
			1	2	4	8	16
0	0	$f_{CLK}/2^8$	3.91 kHz	7.81 kHz	15.6 kHz	31.3 kHz	62.5 kHz
0	1	$f_{CLK}/2^{10}$	977 Hz	1.95 kHz	3.91 kHz	7.81 kHz	15.6 kHz
1	0	$f_{CLK}/2^{12}$	244 Hz	488 Hz	977 Hz	1.95 kHz	3.91 kHz
1	1	$f_{CLK}/2^{14}$	61 Hz	122 Hz	244 Hz	488 Hz	977 Hz

Note 1. When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop timer array unit (TTm = 00FFH).
The timer array unit must also be stopped if the operation 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 6-4** can be achieved by using the interval timer function.

Table 6-4. Interval Times Available for Operation clock CKSm2 or CKSm3

Clock		Interval time[Note 1] (f_{CLK} = 16 MHz)			
		10 µs	100 µs	1 ms	10 ms
CKm2	$f_{CLK}/2$	✓	—	—	—
	$f_{CLK}/2^2$	✓	—	—	—
	$f_{CLK}/2^4$	✓	✓	—	—
	$f_{CLK}/2^6$	✓	✓	—	—
CKm3	$f_{CLK}/2^8$	—	✓	✓	—
	$f_{CLK}/2^{10}$	—	—	✓	✓
	$f_{CLK}/2^{12}$	—	—	✓	✓
	$f_{CLK}/2^{14}$	—	—	—	✓

Note 1. The margin is within 5%.

Remark 1. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 2. For details of a signal of $f_{CLK}/2^j$ selected with the TPSm register, see **6.5.1 Count clock (fTCLK)**.

6.3.3 Timer mode register mn (TMRmn)

The TMRmn register sets an operation mode of channel n. This register is used to select the operation clock (f_{MCK}), select the count clock, select the master/slave, select the 16 or 8-bit timer (only for channels 1 and 3), specify the start trigger and capture trigger, select the valid edge of the timer input, and specify the operation mode (interval, capture, event counter, one-count, or capture and one-count).

Rewriting the TMRmn register is prohibited when the register is in operation (when TEmn = 1). However, bits 7 and 6 (CISmn1, CISmn0) can be rewritten even while the register is operating with some functions (when TEmn = 1) (for details, see **6.8 Independent Channel Operation Function of Timer Array Unit** and **6.9 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, TMRm4, TMRm6: MASTERmn bit (n = 2, 4, 6)
TMRm1, TMRm3: SPLITmn bit (n = 1, 3)
TMRm0, TMRm5, TMRm7: Fixed to 0

Figure 6-12. Format of Timer Mode Register mn (TMRmn) (1/5)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2, 4, 6)	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, 5, 7)	CKSmn1	CKSmn0	0	CCSmn	0[Note 1]	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

CKSmn1	CKSmn0	Selection of operation clock (f_{MCK}) of channel
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 channel 1, valid edge of input signal selected by ISC • In channel 5, valid edge of input signal selected by TIS0
Count clock (f_{TCLK}) is used for the counter, output controller, and interrupt controller.	

Note 1. Bit 11 is a read-only bit and fixed to 0. Writing to this bit 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 operation clock (f_{MCK}) specified by using the CKSmn0 and CKSmn1 bits or the valid edge of the signal input from the TImn pin is selected as the count clock (f_{TCLK}).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6-12. Format of Timer Mode Register mn (TMRmn) (2/5)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2, 4, 6)	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, 5, 7)	CKSmn1	CKSmn0	0	CCSmn	0[Note 1]	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

(Bit 11 of TMRmn (n = 2, 4, 6))

MASTERmn	Selection between using channel n independently or simultaneously with another channel (as a slave or master)
0	Operates in independent channel operation function or as slave channel in simultaneous channel operation function.
1	Operates as master channel in simultaneous channel operation function.
Only the channel 2, 4, 6 can be set as a master channel (MASTERmn = 1). Be sure to use channel 0, 5, 7 are 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	When the channel is used as a slave channel in two-channel input with one-shot pulse output function: The interrupt request signal of the master channel (INTTM0n) is used as the start trigger. A valid edge of the TI03 pin input of the slave channel is used as the end trigger.
Other than above			Setting prohibited

Note 1. Bit 11 is a read-only bit and fixed to 0. Writing to this bit is ignored.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6-12. Format of Timer Mode Register mn (TMRmn) (3/5)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2, 4, 6)	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, 5, 7)	CKSmn1	CKSmn0	0	CCSmn	0[Note 1]	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 1. Bit 11 is a read-only bit and fixed to 0. Writing to this bit is ignored.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6-12. Format of Timer Mode Register mn (TMRmn) (4/5)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2, 4, 6)	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, 5, 7)	CKSmn1	CKSmn0	0	CCSmn	0[Note 1]	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

MDmn3	MDmn2	MDmn1	Setting of operation mode of channel n	Corresponding function	Corresponding function
0	0	0	Interval timer mode	Interval timer/ Square wave output/ Divider function/ PWM output (master)	Down count
0	1	0	Capture mode	Input pulse interval measurement/ Two-channel input with one-shot pulse output function (slave)	Up count
0	1	1	Event counter mode	External event counter	Down count
1	0	0	One-count mode	Delay counter/ One-shot pulse output/ Two-channel input with one-shot pulse output function (master) PWM output (slave)	Down count
1	1	0	Capture & one-count mode	Measurement of high-/low-level width of input signal	Up count
Other than above			Setting prohibited		
The operation of each mode varies depending on MDmn0 bit (see the table below).					

Note 1. Bit 11 is a read-only bit and fixed to 0. Writing to this bit is ignored.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6-12. Format of Timer Mode Register mn (TMRmn) (5/5)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2, 4, 6)	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, 5, 7)	CKSmn1	CKSmn0	0	CCSmn	0[Note 1]	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Operation mode (Value set by the MDmn3 to MDmn1 bits (see the table above))	MDmn0	Setting of starting counting and interrupt
• Interval timer mode (0, 0, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• Capture mode (0, 1, 0)	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.
Other than above		Setting prohibited

Note 1. Bit 11 is a read-only bit and fixed to 0. Writing to this bit is ignored.

Note 2. In one-count mode, interrupt output (INTTMmn) when starting a count operation and TOmn output are not controlled.

Note 3. If the start trigger (TSmn = 1) is issued during operation, the counter is initialized, and recounting is started (does not occur the interrupt request).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.3.4 Timer status register mn (TSRmn)

The TSRmn register indicates the overflow status of the counter of channel n.

The TSRmn register is valid only in the capture mode (MDmn3 to MDmn1 = 010B) and capture & one-count mode (MDmn3 to MDmn1 = 110B). See **Table 6-5** for the operation of the OVF bit in each operation mode and set/clear conditions.

The TSRmn register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the TSRmn register can be read with an 8-bit memory manipulation instruction with TSRmnL.

Reset signal generation clears this register to 0000H.

Figure 6-13. Format of Timer Status Register mn (TSRmn)

Address: F01A0H, F01A1H (TSR00) to F01AEH, F01AFH (TSR07) 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 7)

Table 6-5. OVF Bit Operation and Set/Clear Conditions in Each Operation Mode

Timer operation mode	OVF bit	Set/clear conditions
• Capture mode • Capture & one-count mode	clear	When no overflow has occurred upon capturing
	set	When an overflow has occurred upon capturing
• Interval timer mode • Event counter mode • One-count mode	clear	—
	set	(Use prohibited)

Remark The OVF bit does not change immediately after the counter has overflowed, but changes upon the subsequent capture.

6.3.5 Timer channel enable status register m (TEm)

The TEm register is used to display whether the timer operation of each channel is enabled or stopped.

Each bit of the TEm register corresponds to each bit of the timer channel start register m (TSm) and the timer channel stop register m (TTm). When a bit of the TSm register is set to 1, the corresponding bit of this register is set to 1. When a bit of the TTm register is set to 1, the corresponding bit of this register is cleared to 0.

The TEm register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the TEm register can be set with a 1-bit or 8-bit memory manipulation instruction with TEmL.

Reset signal generation clears this register to 0000H.

Figure 6-14. 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	TEm7	TEm6	TEm5	TEm4	TEm3	TEm2	TEm1	TEm0

TEHm3	Indication of whether operation of the higher 8-bit timer is enabled or stopped when channel 3 is in the 8-bit timer mode
0	Operation is stopped.
1	Operation is enabled.

TEHm1	Indication of whether operation of the higher 8-bit timer is enabled or stopped when channel 1 is in the 8-bit timer mode
0	Operation is stopped.
1	Operation is enabled.

TEmn	Indication of operation enable/stop status of channel n
0	Operation is stopped.
1	Operation is enabled.
This bit displays whether operation of the lower 8-bit timer for TEm1 and TEm3 is enabled or stopped when channel 1 or 3 is in the 8-bit timer mode.	

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.3.6 Timer channel start register m (TSm)

The TSm register is a trigger register that is used to initialize timer count register mn (TCRmn) and start the counting operation of each channel.

When a bit of this register is set to 1, the corresponding bit of timer channel enable status register m (TEm) is set to 1. The TSmn, TSHm1, TSHm3 bits are immediately cleared when operation is enabled (TEmn, TEHm1, TEHm3 = 1), because they are trigger bits.

The TSm register can be set by a 16-bit memory manipulation instruction.

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 6-15. 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	TSm7	TSm6	TSm5	TSm4	TSm3	TSm2	TSm1	TSm0

TSHm3	Trigger to enable operation (start operation) of the higher 8-bit timer when channel 3 is in the 8-bit timer mode
0	No trigger operation
1	The TEHm3 bit is set to 1 and the count operation becomes enabled. The TCRm3 register count operation start in the interval timer mode in the count operation enabled state (see **Table 6-6** in **6.5.2 Start timing of counter**).

TSHm1	Trigger to enable operation (start operation) of the higher 8-bit timer when channel 1 is in the 8-bit timer mode
0	No trigger operation
1	The TEHm1 bit is set to 1 and the count operation becomes enabled. The TCRm1 register count operation start in the interval timer mode in the count operation enabled state (see **Table 6-6** in **6.5.2 Start timing of counter**).

TSmn	Operation enable (start) trigger of channel n
0	No trigger operation
1	The TEmn bit is set to 1 and the count operation becomes enabled. The TCRmn register count operation start in the count operation enabled state varies depending on each operation mode (see **Table 6-6** in **6.5.2 Start timing of counter**). 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.

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

Caution 1. **Be sure to clear bits 15 to 12, 10, and 8 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 (TNFENnm = 1):

Four cycles of the operation clock (f_{MCK})

When the TImn pin noise filter is disabled (TNFENnm = 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 7)

6.3.7 Timer channel stop register m (TTm)

The TTm register is a trigger register that is used to stop the counting operation of each channel.

When a bit of this register is set to 1, the corresponding bit of timer channel enable status register m (TEm) is cleared to 0. The TTmn, TTHm1, TTHm3 bits are immediately cleared when operation is stopped (TEmn, 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 6-16. 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	TTm7	TTm6	TTm5	TTm4	TTm3	TTm2	TTm1	TTm0

TTHm3	Trigger to stop operation of the higher 8-bit timer when channel 3 is in the 8-bit timer mode
0	No trigger operation
1	TEHm3 bit is cleared to 0 and the count operation is stopped.

TTHm1	Trigger to stop operation of the higher 8-bit timer when channel 1 is in the 8-bit timer mode
0	No trigger operation
1	TEHm1 bit is cleared to 0 and the count operation is stopped.

TTmn	Operation stop trigger of channel n
0	No trigger operation
1	TEmn bit is cleared to 0 and the count operation is stopped. The TTm1 and TTm3 bits respectively trigger stoppage of operation of the lower 8-bit timers in channels 1 and 3 when these channels are in the 8-bit timer mode.

Caution Be sure to clear bits 15 to 12, 10, and 8 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 7)

6.3.8 Timer input select register 0 (TIS0)

The TIS0 register is used to select the timer input of channel 5 of unit 0.

The TIS0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6-17. 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	0	0	0	0	TIS02	TIS01	TIS00

TIS02	TIS01	TIS00	Selection of timer input used with channel 5
0	0	0	Input signal of timer input pin (TI05)
0	0	1	
0	1	0	
0	1	1	
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
1	0	1	Subsystem clock (f_{SUB})
Other than the above			Setting prohibited

Caution The high-level width and low-level width of the selected timer input must be $1/f_{MCK}$ + 10 ns or more. Therefore, 1 cannot be set to the TIS02 bit when f_{SUB} is selected for f_{CLK} (CSS of the CKC register = 1).

6.3.9 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 6-18. 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	TOEm7	TOEm6	TOEm5	TOEm4	TOEm3	TOEm2	TOEm1	TOEm0

TOEmn	Timer output enable/disable of channel n
0	Disable output of timer. Without reflecting on TOmn bit timer operation, to fixed the output. Writing to the TOmn bit is enabled and the level set in the TOmn bit is output from the TOmn pin.
1	Enable output of timer. Reflected in the TOmn bit timer operation, to generate the output waveform. Writing to the TOmn bit is disabled (writing is ignored).

Caution **Be sure to clear bits 15 to 8 to 0.**

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.3.10 Timer output register m (TOm)

The TOm register is a buffer register of timer output of each channel.

The value of each bit in this register is output from the timer output pin (TOmn) of each channel.

The TOmn bit on this register can be rewritten by software only when timer output is disabled (TOEmn = 0). When timer output is enabled (TOEmn = 1), rewriting this register by software is ignored, and the value is changed only by the timer operation.

To use the TO0n alternate pin as a port function pin, set the corresponding TOmn bit to “0”.

The TOm register can be set by a 16-bit memory manipulation instruction.

The 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 6-19. 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	TOm7	TOm6	TOm5	TOm4	TOm3	TOm2	TOm1	TOm0

TOmn	Timer output of channel n
0	Timer output value is “0”.
1	Timer output value is “1”.

Caution Be sure to clear bits 15 to 8 to 0.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.3.11 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 simultaneous channel operation function (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 6-20. 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	TOLm7	TOLm6	TOLm5	TOLm4	TOLm3	TOLm2	TOLm1	0

TOLmn	Control of timer output level of channel n
0	Positive logic output (active-high)
1	Negative logic output (active-low)

Caution Be sure to clear bits 15 to 8, 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 7)

6.3.12 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, multiple PWM output, or two-channel input with one-shot pulse 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 6-21. 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	TOMm7	TOMm6	TOMm5	TOMm4	TOMm3	TOMm2	TOMm1	0

TOMmn	Control of timer output mode of channel n
0	Master channel output mode (to produce toggle output by timer interrupt request signal (INTTMmn))
1	Slave channel output mode (output is set by the timer interrupt request signal (INTTMmn) of the master channel, and reset by the timer interrupt request signal (INTTMmp) of the slave channel)

Caution Be sure to clear bits 15 to 8, and 0 to 0.

Remark m: Unit number (m = 0)
n: Channel number
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
p: Slave channel number
n < p ≤ 7
(For details of the relation between the master channel and slave channel, refer to **6.4.1 Basic rules of simultaneous channel operation function**.)

6.3.13 Input switch control register (ISC)

The ISC register is used to implement UART0 baud rate correction by using channel 1 in association with the serial array unit.

When the ISC1 bit is set to 1, the input signal of the serial data input (RxD0) pin is selected as a timer input.

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

Reset signal generation clears this register to 00H.

Figure 6-22. Format of Input Switch Control Register (ISC)

Address: F0073H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	ISC1	ISC0

ISC1	Switching channel 1 input of timer array unit
0	Uses the input signal of the TI01 pin as a timer input (normal operation).
1	Uses the input signal of the RxD0 pin as a timer input (detects the wakeup signal and measures the pulse width for baud rate correction).

ISC0	Switching external interrupt (INTP0) input
0	Uses the input signal of the INTP0 pin as an external interrupt (normal operation).
1	Uses the input signal of the RxD0 pin as an external interrupt (wakeup signal detection).

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

6.3.14 Noise filter enable registers 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 operation clock (f_{MCK}) for the target channel, whether the signal keeps the same value for two clock cycles is detected. When the noise filter is disabled, the input signal is only synchronized with the operation clock (f_{MCK}) for the target channel[Note 1].

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 1. For details, see **6.5.1(2) When valid edge of input signal via the TImn pin is selected (CCSmn = 1)**, **6.5.2 Start timing of counter**, and **6.7 Timer Input (TImn) Control**.

Figure 6-23. Format of Noise Filter Enable Registers 1 (NFEN1) (1/2)

Address: F0071H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN1	TNFEN07	TNFEN06	TNFEN05	TNFEN04	TNFEN03	TNFEN02	TNFEN01	TNFEN00

TNFEN07	Enable/disable using noise filter of TI07 pin
0	Noise filter OFF
1	Noise filter ON

TNFEN06	Enable/disable using noise filter of TI06 pin
0	Noise filter OFF
1	Noise filter ON

TNFEN05	Enable/disable using noise filter of TI05 pin
0	Noise filter OFF
1	Noise filter ON

TNFEN04	Enable/disable using noise filter of TI04 pin
0	Noise filter OFF
1	Noise filter ON

TNFEN03	Enable/disable using noise filter of TI03 pin
0	Noise filter OFF
1	Noise filter ON

Remark The presence or absence of timer I/O pins of channel 0 to 7 depends on the product. See **Table 6-2 Timer I/O Pins provided in Each Product**.

Figure 6-23. Format of Noise Filter Enable Registers 1 (NFEN1) (2/2)

Address: F0071H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN1	TNFEN07	TNFEN06	TNFEN05	TNFEN04	TNFEN03	TNFEN02	TNFEN01	TNFEN00

TNFEN02	Enable/disable using noise filter of TI02 pin
0	Noise filter OFF
1	Noise filter ON

TNFEN01	Enable/disable using noise filter of TI01 pin[Note 1]
0	Noise filter OFF
1	Noise filter ON

TNFEN00	Enable/disable using noise filter of TI00 pin
0	Noise filter OFF
1	Noise filter ON

Note 1. The applicable pin can be switched by setting the ISC1 bit of the ISC register.
ISC1 = 0: Whether or not to use the noise filter of the TI01 pin can be selected.
ISC1 = 1: Whether or not to use the noise filter of the RxD0 pin can be selected.

Remark The presence or absence of timer I/O pins of channel 0 to 7 depends on the product. See **Table 6-2 Timer I/O Pins provided in Each Product**.

6.3.15 Registers controlling port functions of pins to be used for timer I/O

Using port pins for the timer array unit functions requires setting of the registers that control the port functions multiplexed on the target pins (port mode register (PMxx), port register (Pxx), and port mode control register (PMCxx)). For details, see **4.3.1 Port mode registers 0, 1, 2, 4, 6, 12 (PM0, PM1, PM2, PM4, PM6, PM12)**, **4.3.2 Port registers 0, 1, 2, 4, 6, 12, 13 (P0, P1, P2, P4, P6, P12, P13)**, and **4.3.5 Port mode control registers 0, 2 (PMC0, PMC2)**.

The port mode register (PMxx), port register (Pxx), and port mode control register (PMCxx) to be set depend on the product. For details, see **4.5.3 Register setting examples for used port and alternate functions**.

When using the ports (such as P03/TO00) to be shared with the timer output pin for timer output, set the port mode control register (PMCxx) bit, port mode register (PMxx) bit and port register (Pxx) bit corresponding to each port to 0.

Example) When using P03/TO00 for timer output

Set the PMC03 bit of port mode control register 0 to 0.

Set the PM03 bit of port mode register 0 to 0.

Set the P03 bit of port register 0 to 0.

When using the ports (such as P04/TI01) to be shared with the timer input pin for timer input, set the port mode register (PMxx) bit corresponding to each port to 1. And set the port mode control register (PMCxx) bit corresponding to each port to 0. At this time, the port register (Pxx) bit may be 0 or 1.

Example) When using P04/TI01 for timer input

Set the PMC04 bit of port mode control register 0 to 0.

Set the PM04 bit of port mode register 0 to 1.

Set the P04 bit of port register 0 to 0 or 1.

6.4 Basic Rules of Timer Array Unit

6.4.1 Basic rules of simultaneous channel operation function

When simultaneously using multiple channels, namely, a combination of a master channel (a reference timer mainly counting the cycle) and slave channels (timers operating according to the master channel), the following rules apply.

(1) Only an even channel (channel 0, 2, 4, etc.) can be set as a master channel.

(2) Any channel, except channel 0, can be set as a slave channel.

(3) The slave channel must be lower than the master channel.
Example) If channel 2 is set as a master channel,
channel 3 or those that follow (channels 3, 4, 5, etc.) can be set as a slave channel.

(4) Two or more slave channels can be set for one master channel.

(5) When two or more master channels are to be used, slave channels with a master channel between them may not be set.
Example) If channels 0 and 4 are set as master channels,
channels 1 to 3 can be set as the slave channels of master channel 0.
channels 5 to 7 cannot be set as the slave channels of master channel 0.

(6) The operation 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 to the lower channels.

(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 operation clocks of master channels and slave channels have to be synchronized.

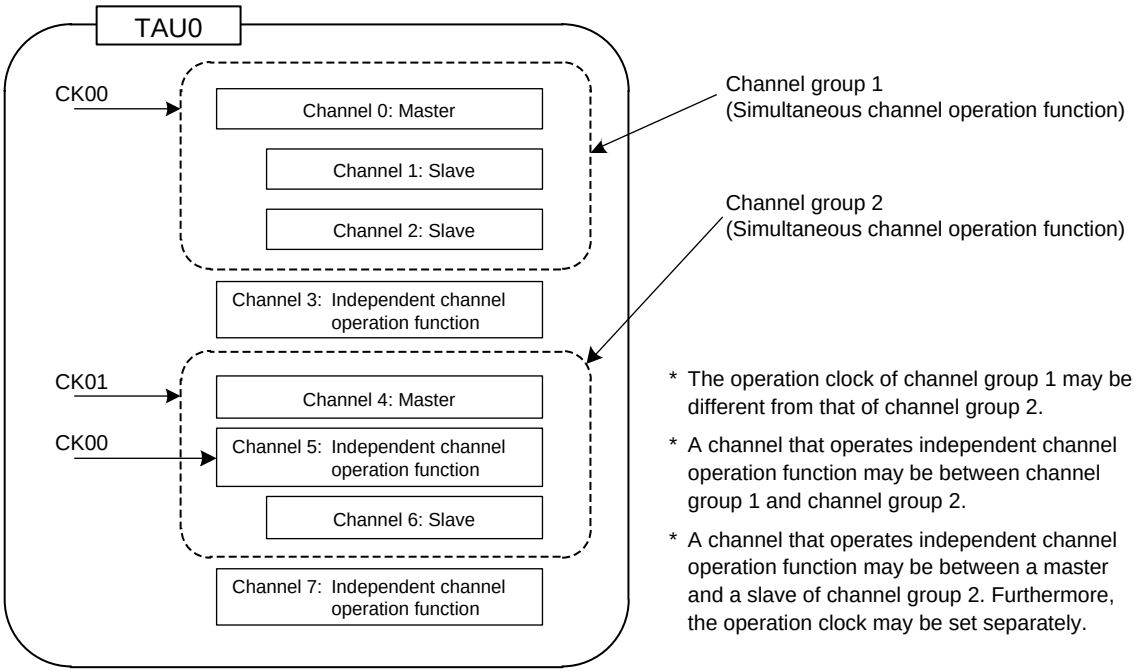
(14) Timer mode register m0 (TMRm0) has no master bit (it is fixed as “0”). However, as channel 0 is the highest channel, it can be used as a master channel during simultaneous operation.

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 6.4.1 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 7)

Example)



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

The 8-bit timer operation function makes it possible to use a 16-bit timer channel in a configuration consisting of two 8-bit timer channels.

This function can only be used for channels 1 and 3, and there are several rules for using it.

The basic rules for this function are as follows:

(1) The 8-bit timer operation function applies only to channels 1 and 3.

(2) When using 8-bit timers, set the SPLITmn bit of timer mode register mn (TMRmn) to 1.

(3) The higher 8 bits can be operated as the interval timer function.

(4) At the start of operation, the higher 8 bits output INTTMm1H/INTTMm3H (an interrupt) (which is the same operation performed when MDmn0 is set to 1).

(5) The operation clock of the higher 8 bits is selected according to the CKSmn1 and CKSmn0 bits of the lower-bit TMRmn register.

(6) For the higher 8 bits, the TSHm1/TSHm3 bit is manipulated to start channel operation and the TTHm1/TTHm3 bit is manipulated to stop channel operation. The channel status can be checked using the TEHm1/TEHm3 bit.

(7) The lower 8 bits operate according to the TMRmn register settings. The following three functions support operation of the lower 8 bits:
- Interval timer function/Square Wave Output 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 and TTm1/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)

6.5 Operation of Counter

6.5.1 Count clock (f_{TCLK})

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

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

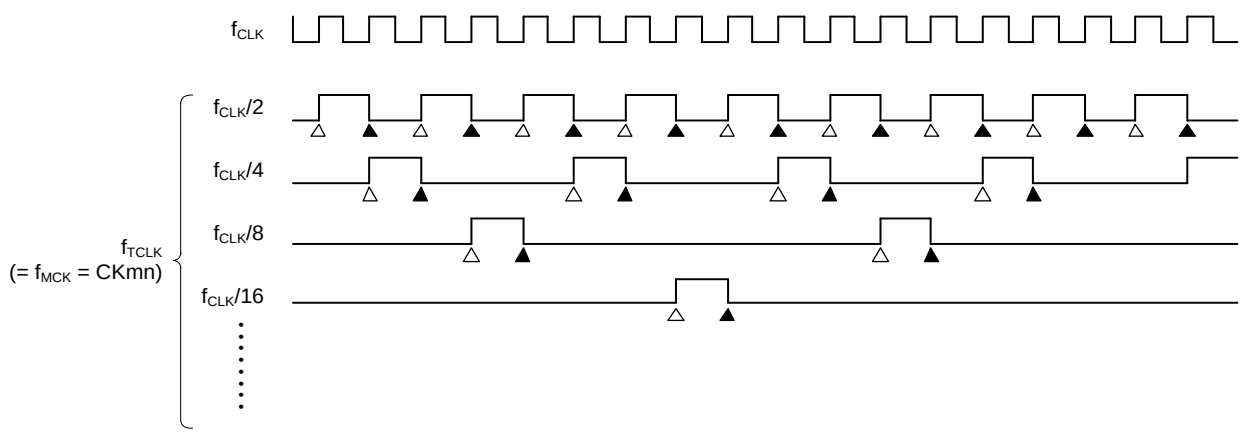
Because the timer array unit is designed to operate in synchronization with f_{CLK}, the timings of the count clock (f_{TCLK}) are shown below.

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

The count clock (f_{TCLK}) is 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 6-24. 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

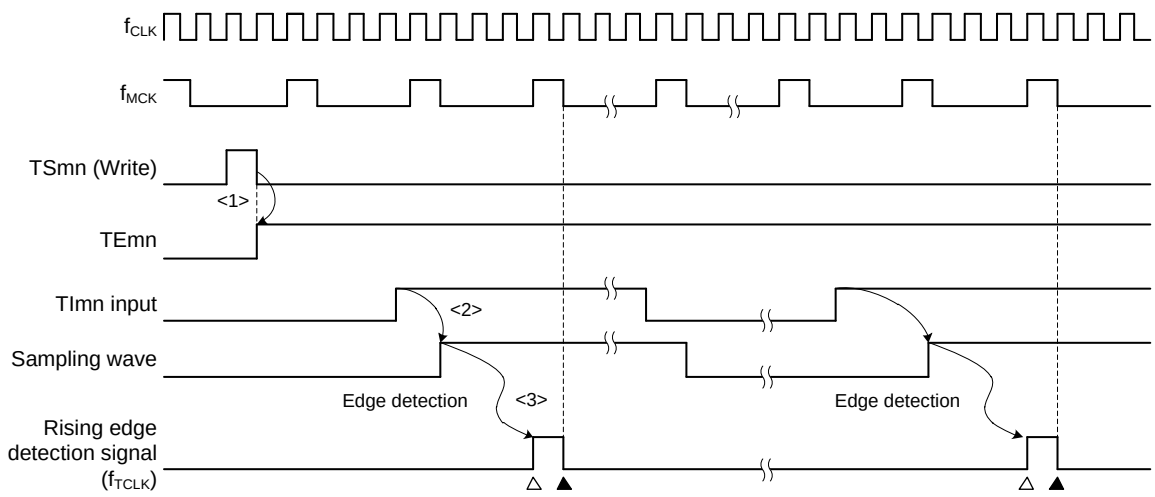
Remark 2. f_{CLK}: CPU/peripheral hardware clock

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

The count clock (f_{TCLK}) becomes the signal that detects valid edge of input signal via the TImn pin and synchronizes next rising f_{MCK}. The count clock (f_{TCLK}) is delayed for 1 to 2 period of f_{MCK} from the input signal via the TImn pin (when a noise filter is used, the delay becomes 3 to 4 clock).

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

Figure 6-25. Timing of f_{CLK} and count clock (f_{TCLK}) (When CCSmn = 1, noise filter unused)



<1> Setting TSmn bit to 1 enables the timer to be started and to become wait state for valid edge of input signal via the TImn pin.

<2> The rise of input signal via the TImn pin is sampled by f_{MCK}.

<3> The edge is detected by the rising of the sampled signal and the detection signal (count clock) is output.

Remark 1. △: Rising edge of the count clock
▲: Synchronization, increment/decrement of counter

Remark 2. f_{CLK}: CPU/peripheral hardware clock
f_{MCK}: Operation clock of channel n

Remark 3. The waveform of the input signal via TImn pin of the input pulse interval measurement, the measurement of high/low width of input signal, the delay counter, and the one-shot pulse output are the same.

6.5.2 Start timing of counter

Timer count register mn (TCRmn) becomes enabled to operation by setting of TSmn bit of timer channel start register m (TSm).

Operations from count operation enabled state to timer count register mn (TCRmn) count start is shown in **Table 6-6**.

Table 6-6. Operations from Count Operation Enabled State to Timer count Register mn (TCRmn) Count Start

Timer operation mode	Operation when TSmn = 1 is set
Interval timer mode	No operation is carried out from start trigger detection (TSmn=1) until count clock generation. The first count clock loads the value of the TDRmn register to the TCRmn register and the subsequent count clock performs count down operation (see **6.5.3(1) Operation of interval timer mode**).
Event counter mode	Writing 1 to the TSmn bit loads the value of the TDRmn register to the TCRmn register. If detect edge of TImn input. The subsequent count clock performs count down operation (see **6.5.3(2) Operation of event counter mode**).
Capture mode	No operation is carried out from start trigger detection (TSmn=1) until count clock generation. The first count clock loads 0000H to the TCRmn register and the subsequent count clock performs count up operation (see **6.5.3(3) Operation of capture mode (input pulse interval measurement)**).
One-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads the value of the TDRmn register to the TCRmn register and the subsequent count clock performs count down operation (see **6.5.3(4) One-count mode operation**).
Capture & one-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to the TCRmn register and the subsequent count clock performs count up operation (see **6.5.3(5) Operation of capture & one-count mode (high-level width measurement)**).

6.5.3 Operation of counter

Here, the counter operation in each mode is explained.

(1) Operation of interval timer mode

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit. Timer count register mn (TCRmn) holds the initial value until count clock generation.

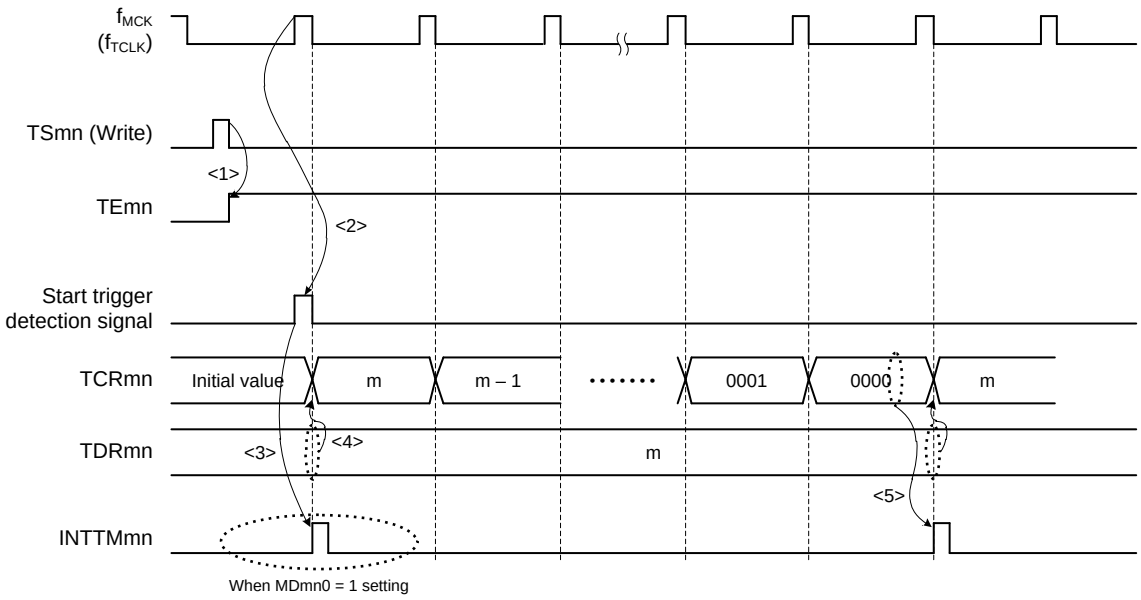
<2> A start trigger is generated at the first count clock (f_{MCK}) after operation is enabled.

<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 at the next count clock (f_{MCK}) and the value of timer data register mn (TDRmn) is loaded to the TCRmn register and counting keeps on.

Figure 6-26. 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 f_{MCK}, the start trigger detection signal, and INTTMmn become active between one clock in synchronization with f_{CLK}.

(2) Operation of event counter mode

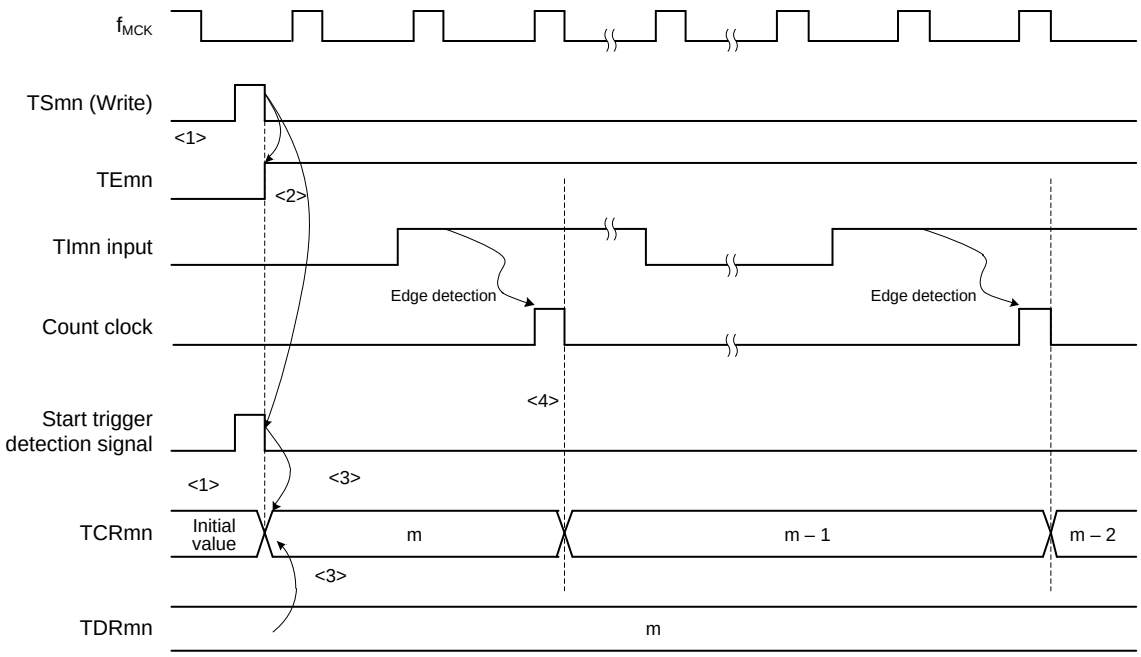
<1> Timer count register mn (TCRmn) holds its initial value while operation is stopped (TEmn = 0).

<2> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.

<3> As soon as 1 has been written to the TSmn bit and 1 has been set to the TEmn bit, the value of timer data register mn (TDRmn) is loaded to the TCRmn register to start counting.

<4> After that, the TCRmn register value is counted down according to the count clock of the valid edge of the TImn input.

Figure 6-27. Operation Timing (In Event Counter Mode)



Remark **Figure 6-27** shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (f_{MCK}).

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

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

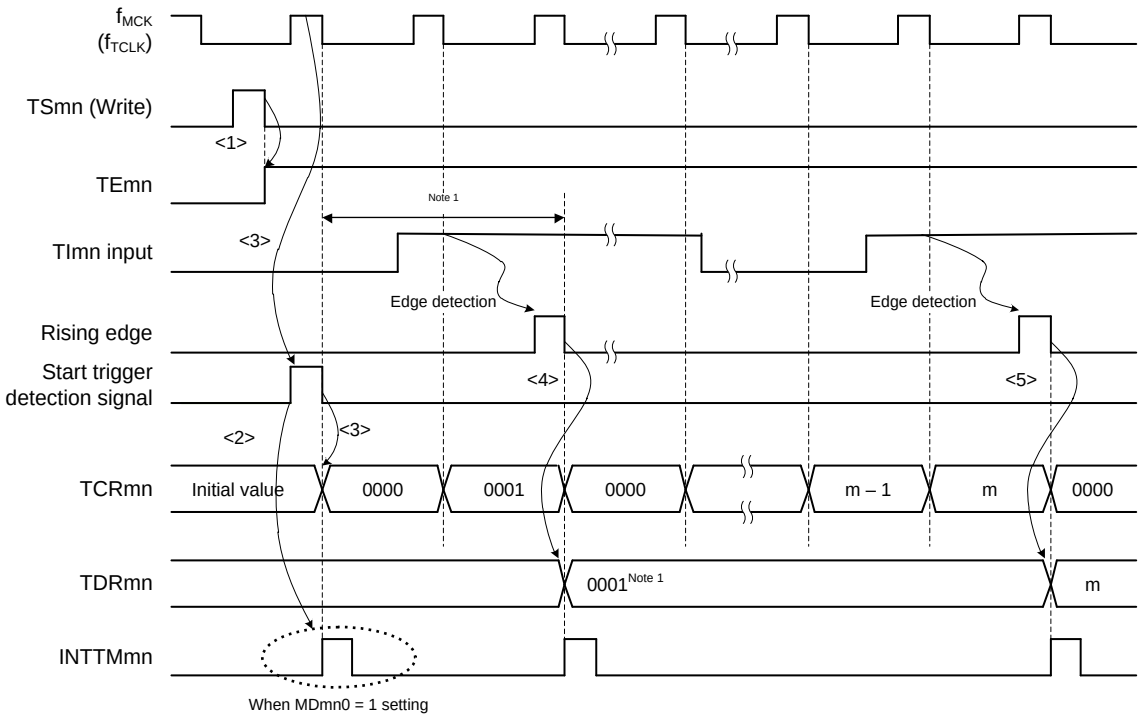
<2> Timer count register mn (TCRmn) holds the initial value until count clock generation.

<3> A start trigger is generated at the first count clock (f_{MCK}) 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 an INTTMmn interrupt is generated. However, this capture value is nomeaning. 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 the TDRmn register and an INTTMmn interrupt is generated.

Figure 6-28. Operation Timing (In Capture Mode: Input Pulse Interval Measurement)



Note 1. If a clock has been input to TImn (the trigger exists) when capturing starts, counting starts when a trigger is detected, even if no edge is detected. Therefore, the first captured value (<4>) does not determine a pulse interval (in the above figure, 0001 just indicates two clock cycles but does not determine the pulse interval) and so the user can ignore it.

Caution In the first cycle operation of count clock after writing the TSmn bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.

Remark **Figure 6-28** shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (f_{MCK}).

(4) One-count mode operation

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

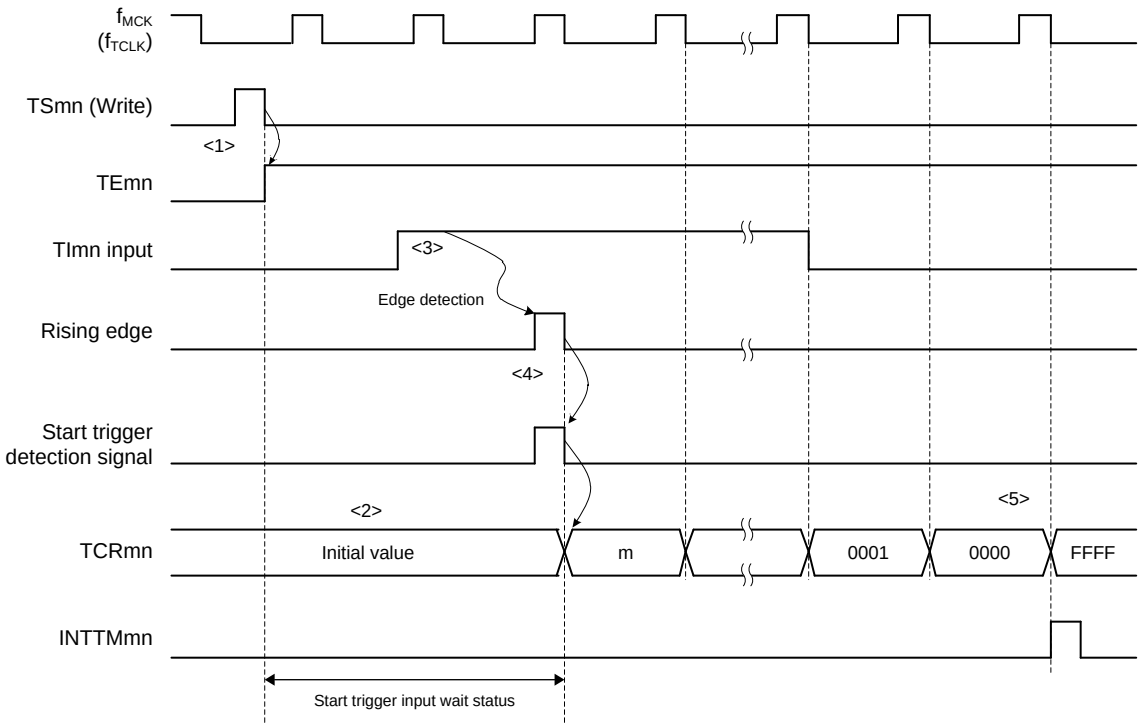
<2> Timer count register mn (TCRmn) holds the initial value until start trigger generation.

<3> Rising edge of the TImn input is detected.

<4> On start trigger detection, the value of timer data register m (TDRmn) is loaded to the TCRmn register and count starts.

<5> When the TCRmn register counts down and its count value is 0000H, the interrupt request signal (INTTMmn) is generated and the value of the TCRmn register becomes FFFFH and counting stops.

Figure 6-29. Operation Timing (In One-count Mode)



Remark **Figure 6-29** shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (f_{MCK}).

(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).

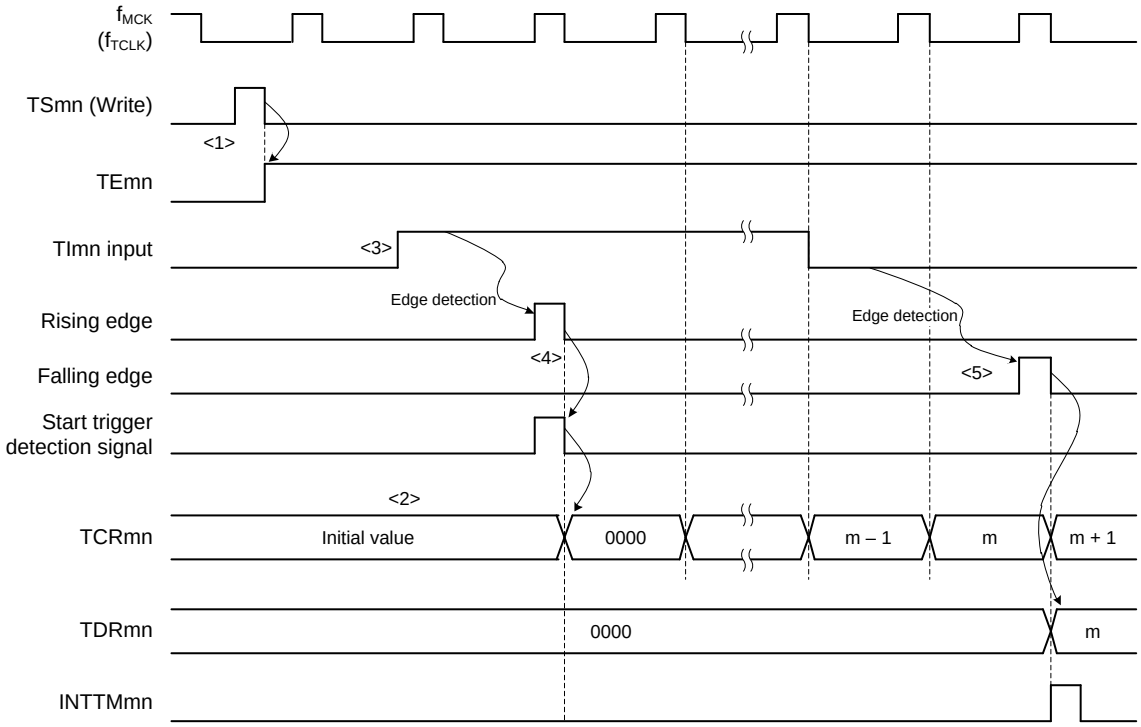
<2> Timer count register mn (TCRmn) holds the initial value until start trigger generation.

<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 an INTTMmn interrupt is generated.

Figure 6-30. Operation Timing (In Capture & One-count Mode: High-level Width Measurement)

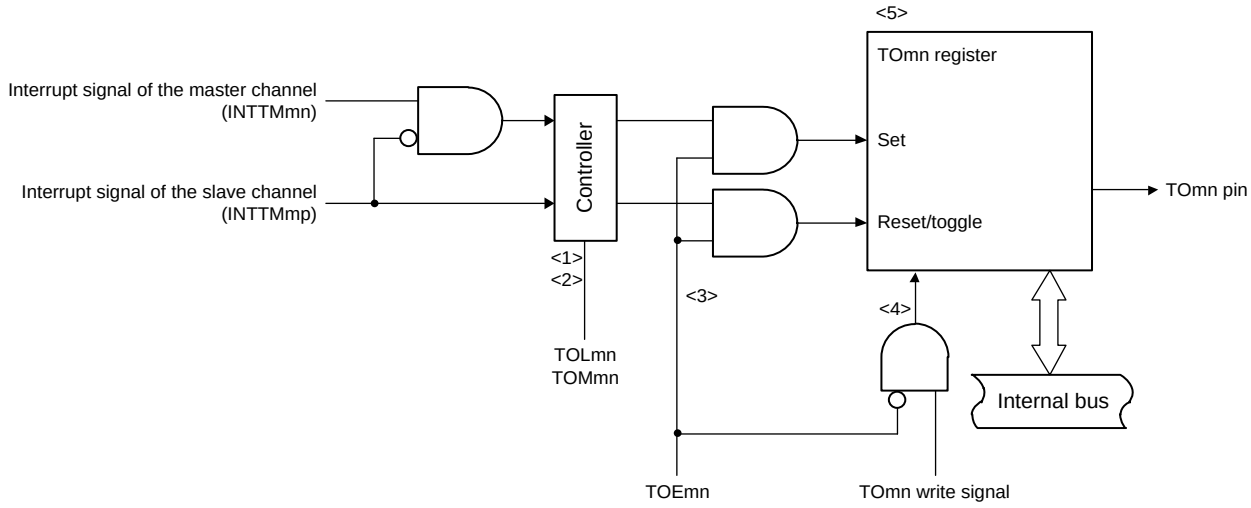


Remark **Figure 6-30** shows the timing when the noise filter is not used. By making the noise filter on-state, the edge detection becomes 2 f_{MCK} cycles (it sums up to 3 to 4 cycles) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (f_{MCK}).

6.6 Channel Output (TOmn pin) Control

6.6.1 TOmn pin output circuit configuration

Figure 6-31. 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 INTTMmp (slave channel timer interrupt) is transmitted to timer output register m (TOm).

<2> When TOMmn = 1 (slave channel output mode), both INTTMmn (master channel timer interrupt) and INTTMmp (slave channel timer interrupt) are transmitted to the TOm register.
At this time, the TOLm register becomes valid and the signals are controlled as follows:
When TOLmn = 0: Positive logic output (INTTMmn → set, INTTMmp → reset)
When TOLmn = 1: Negative logic output (INTTMmn → reset, INTTMmp → set)

When INTTMmn and INTTMmp are simultaneously generated, (0% output of PWM), INTTMmp (reset signal) takes priority, and INTTMmn (set signal) is masked.

<3> While timer output is enabled (TOEmn = 1), INTTMmn (master channel timer interrupt) and INTTMmp (slave channel timer interrupt) are transmitted to the TOm register. Writing to the TOm register (TOmn write signal) becomes invalid.
When TOEmn = 1, the TOmn pin output never changes with signals other than interrupt signals.
To initialize the TOmn pin output level, it is necessary to set timer operation is stopped (TOEmn = 0) and to write a value to the TOm register.

<4> While timer output is disabled (TOEmn = 0), writing to the TOmn bit to the target channel (TOmn write signal) becomes valid. When timer output is disabled (TOEmn = 0), neither INTTMmn (master channel timer interrupt) nor INTTMmp (slave channel timer interrupt) is transmitted to the TOm register.

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

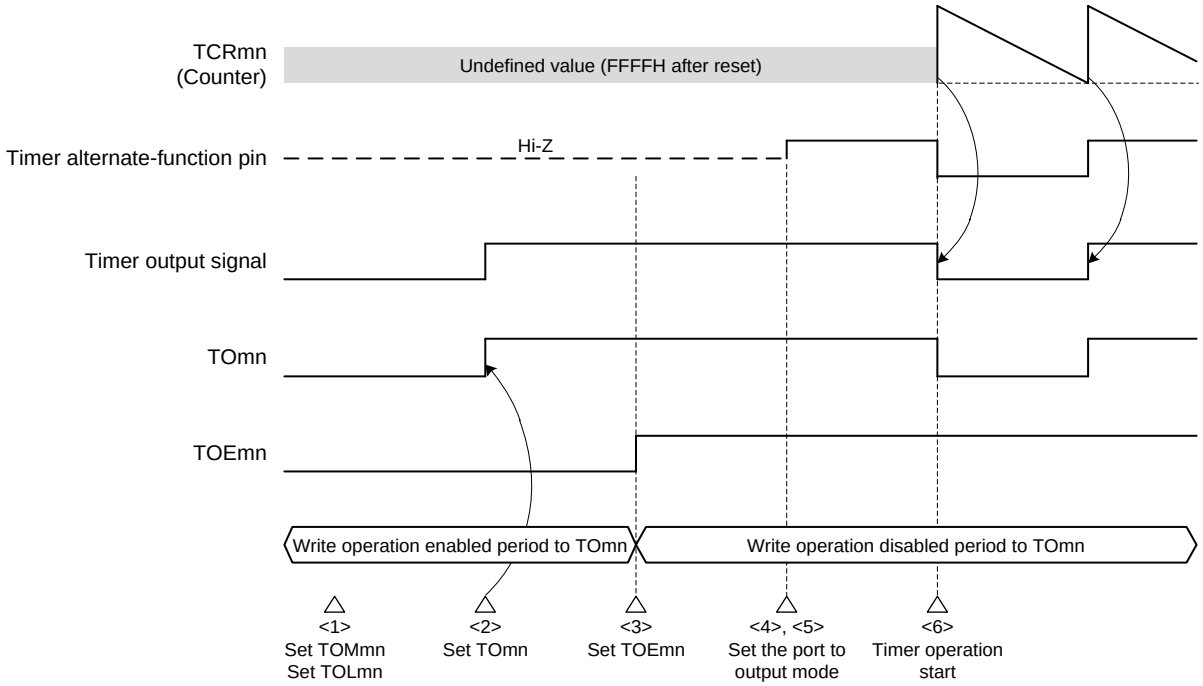
(Remark is listed on the next page.)

Remark m: Unit number (m = 0)
n: Channel number
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
p: Slave channel number
n < p ≤ 7

6.6.2 TOmn pin output setting

The following figure shows the procedure and status transition of the TOmn output pin from initial setting to timer operation start.

Figure 6-32. 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 **6.3.15 Registers controlling port functions of pins to be used for timer I/O**).

<5> The port I/O setting is set to output (see **6.3.15 Registers controlling port functions of pins to be used for timer I/O**).

<6> The timer operation is enabled (TSmn = 1).

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.6.3 Cautions on channel output operation

(1) Changing values set in the registers TOm, TOEm, and TOLm during timer operation

Since the timer operations (operations of timer 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 in **6.7 Timer Input (TImn) Control** and **6.8 Independent Channel Operation Function of Timer Array Unit**.

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 7)

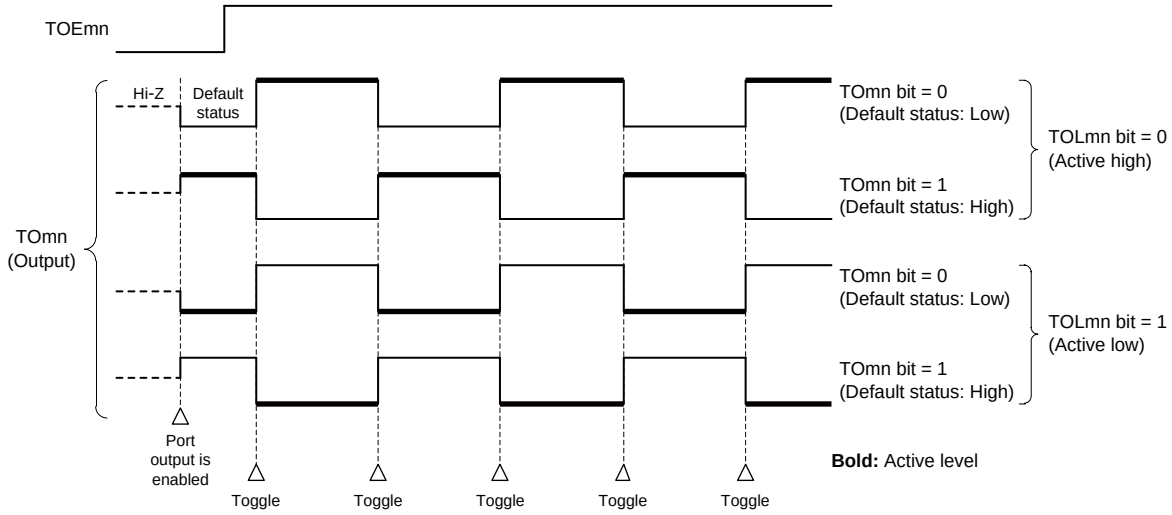
(2) Default level of TOmn pin and output level after timer operation start

The change in the output level of the TOmn pin when timer output register m (TOm) is written while timer output is disabled (TOEmn = 0), the initial level is changed, and then timer output is enabled (TOEmn = 1) before port output is enabled, is shown below.

(a) When operation starts with master channel output mode (TOMmn = 0) setting

The setting of timer output level register m (TOLm) is invalid when master channel output mode (TOMmn = 0). When the timer operation starts after setting the default level, the toggle signal is generated and the output level of the TOmn pin is reversed.

Figure 6-33. TOmn Pin Output Status at Toggle Output (TOMmn = 0)



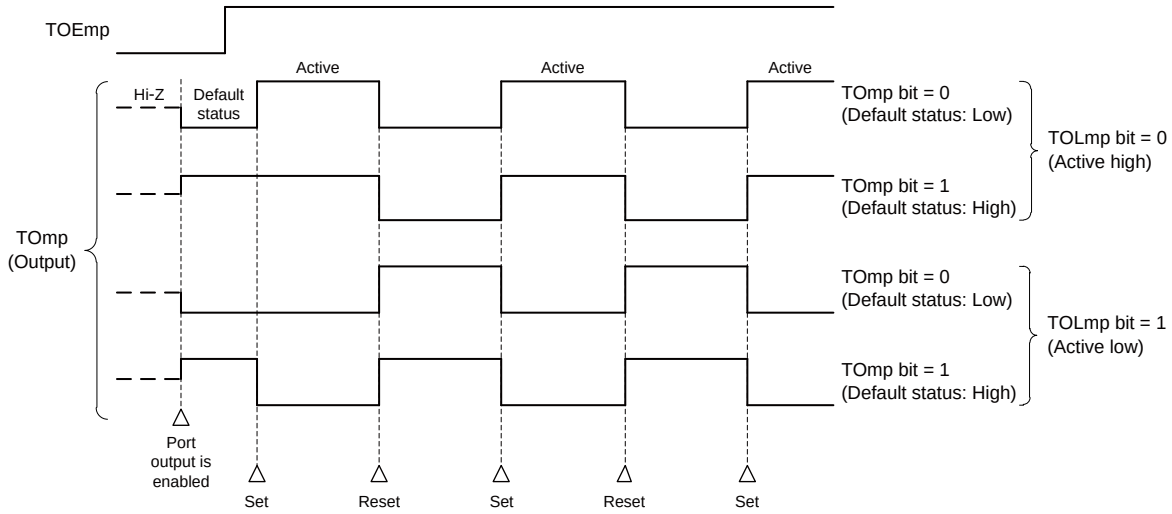
Remark 1. Toggle: Reverse TOmn pin output status

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

(b) When operation starts with slave channel output mode (TOMmp = 1) setting (PWM output))

When slave channel output mode (TOMmp = 1), the active level is determined by timer output level register m (TOLm) setting.

Figure 6-34. TOmp Pin Output Status at PWM Output (TOMmp = 1)



Remark 1. Set: The output signal of the TOmp pin changes from inactive level to active level.
Reset: The output signal of the TOmp pin changes from active level to inactive level.

Remark 2. m: Unit number (m = 0), p: Channel number (p = 1 to 7)

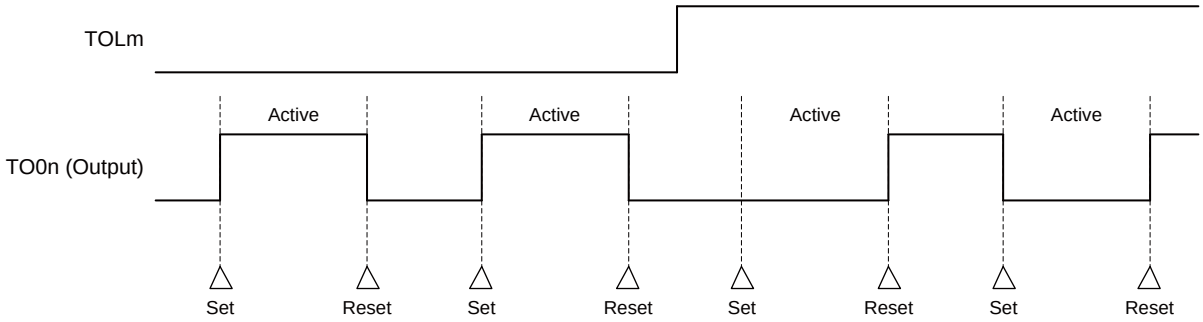
(3) Operation of TOmn pin in slave channel output mode (TOMmn = 1)

(a) When timer output level register m (TOLm) setting has been changed during timer operation

When the TOLm register setting has been changed during timer operation, the setting becomes valid at the generation timing of the TOmn pin change condition. Rewriting the TOLm register does not change the output level of the TOmn pin.

The operation when TOMmn is set to 1 and the value of the TOLm register is changed while the timer is operating (TEmn = 1) is shown below.

Figure 6-35. Operation when TOLm Register Has Been Changed Contents 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 7)

(b) Set/reset timing

To realize 0%/100% output at PWM output, the TOmn pin/TOmn bit set timing at master channel timer interrupt (INTTMmn) generation is delayed by 1 count clock by the slave channel.

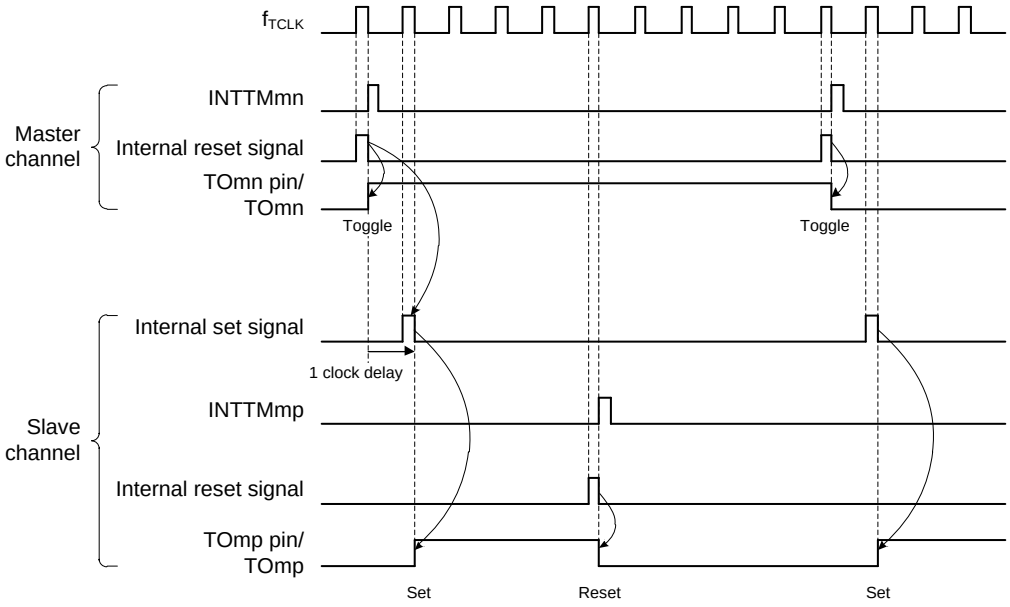
If the set condition and reset condition are generated at the same time, a higher priority is given to the latter.

Figure 6-36 shows the set/reset operating status where the master/slave channels are set as follows.

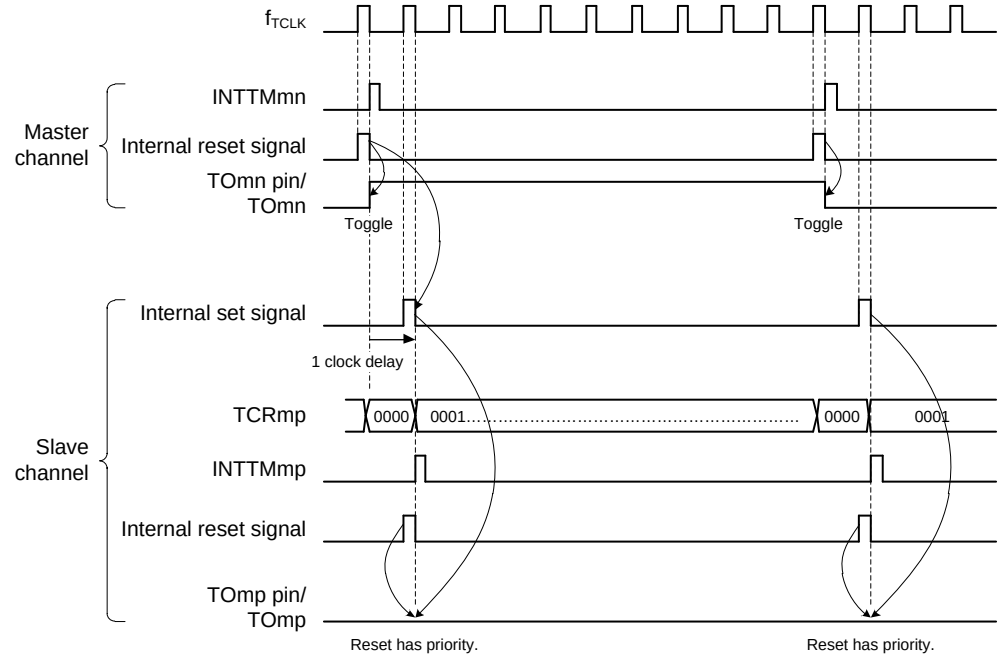
Master channel: TOEmn = 1, TOMmn = 0, TOLmn = 0
Slave channel: TOEmp = 1, TOMmp = 1, TOLmp = 0

Figure 6-36. Set/Reset Timing Operating Status

(1) Basic operation timing



(2) Operation timing when 0% duty



(Remarks are listed on the next page.)

Remark 1. Internal reset signal: TOmn pin reset/toggle signal
Internal set signal: TOmn pin set signal

Remark 2. m: Unit number (m = 0)
n: Channel number
n = 0 to 7 (n = 0, 2, 4, 6 for master channel)
p: Slave channel number
n < p ≤ 7

6.6.4 Collective manipulation of TOmn bit

In timer output register m (TOm), the setting bits (TOmn) for all the channels are located in one register in the same way as timer channel start register m (TSm). Therefore, the TOmn bit of all the channels can be manipulated collectively. Only the desired bits can also be manipulated by enabling writing only to the TOmn bits (TOEmn = 0) that correspond to the relevant bits of the channel used to perform output (TOmn).

Figure 6-37. Example of TO0n Bit Collective Manipulation

Before writing

TO0	0	0	0	0	0	0	0	0	TO07 0	TO06 0	TO05 1	TO04 0	TO03 0	TO02 0	TO01 1	TO00 0
TOE0	0	0	0	0	0	0	0	0	TOE07 0	TOE06 0	TOE05 1	TOE04 0	TOE03 1	TOE02 1	TOE01 1	TOE00 1

Data to be written

0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1

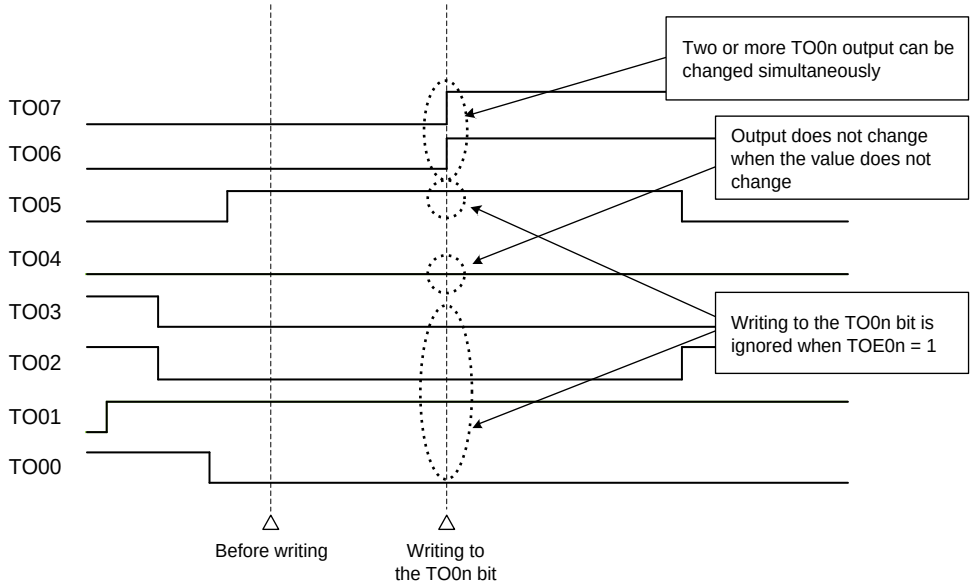
After writing

TO0	0	0	0	0	0	0	0	0	TO07 1	TO06 1	TO05 1	TO04 0	TO03 0	TO02 0	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 6-38. TO0n Pin Status by Collective Manipulation of TO0n Bit



Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.6.5 Timer interrupt and TOmn pin output at count 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.

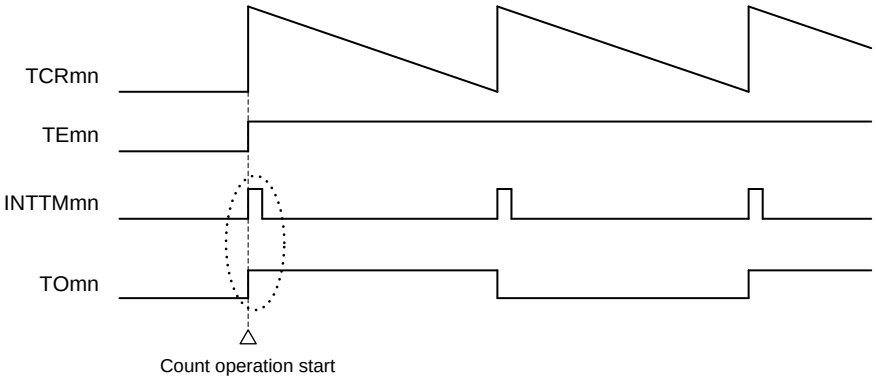
When MDmn0 is set to 1, the count operation start timing can be known by the timer interrupt (INTTMmn) generation.

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

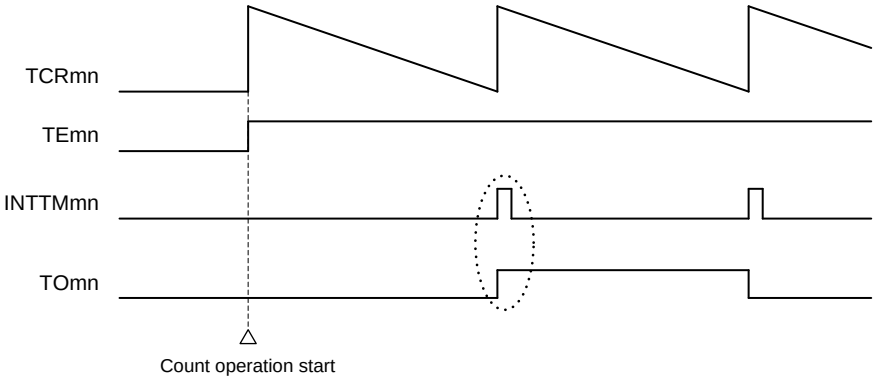
Figure 6-39 shows operation examples when the interval timer mode (TOEmn = 1, TOMmn = 0) is set.

Figure 6-39. Operation examples of timer interrupt at count operation start and TOmn output

(a) When MDmn0 is set to 1



(b) When MDmn0 is set to 0



When MDmn0 is set to 1, a timer interrupt (INTTMmn) is output at count operation start, and TOmn performs a toggle operation.

When MDmn0 is set to 0, a timer interrupt (INTTMmn) is not output at count operation start, and TOmn does not change either. After counting one cycle, INTTMmn is output and TOmn performs a toggle operation.

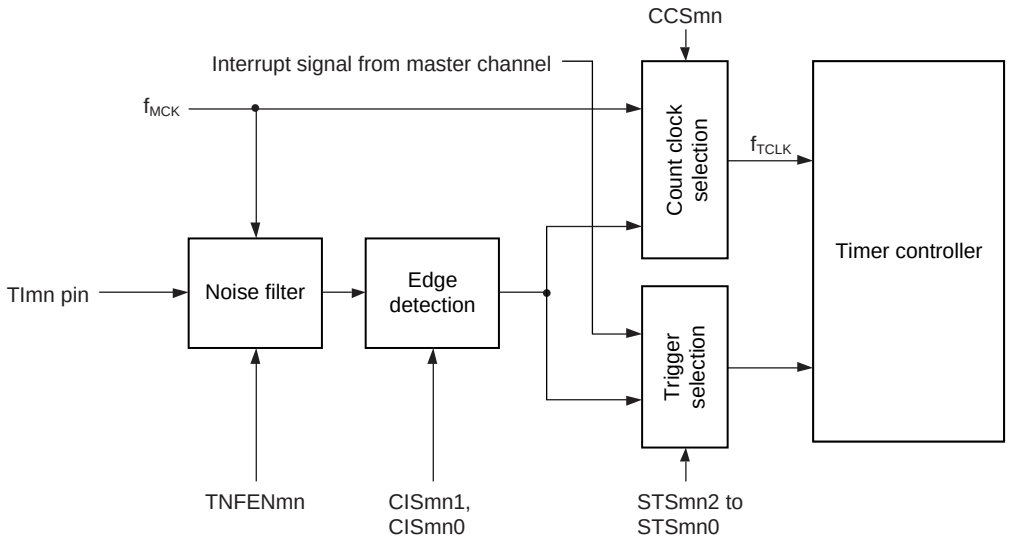
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.7 Timer Input (TImn) Control

6.7.1 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.

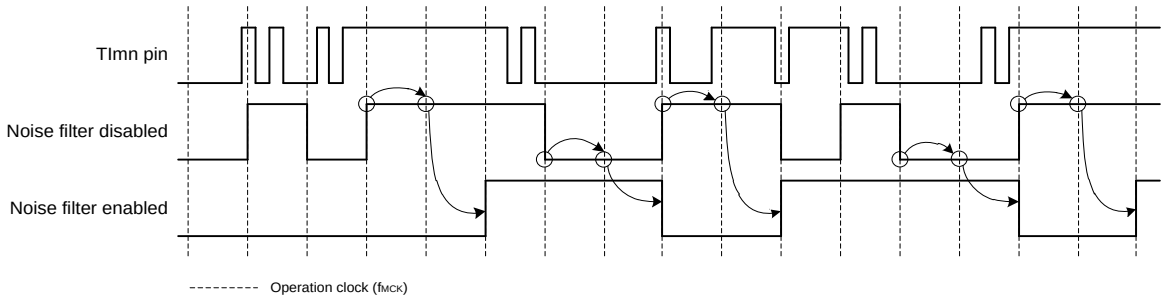
Figure 6-40. Input Circuit Configuration



6.7.2 Noise filter

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

Figure 6-41. Sampling Waveforms through TImn Input Pin with Noise Filter Enabled and Disabled



6.7.3 Cautions on channel input operation

When a timer input pin is set as unused, the operation 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 operation 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 operation clock (f_{MCK}), and then set the operation enable trigger bit in the timer channel start register (TSm).

6.8 Independent Channel Operation Function of Timer Array Unit

6.8.1 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.

$$\text{Generation period of INTTMmn (timer interrupt)} = \text{Period of count clock} \times (\text{Set value of TDRmn} + 1)$$

(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.

$$\text{Period of square wave output from TOmn} = \text{Period of count clock} \times (\text{Set value of TDRmn} + 1) \times 2$$

$$\text{Frequency of square wave output from TOmn} = \text{Frequency of count clock}/\{(\text{Set value of TDRmn} + 1) \times 2\}$$

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.

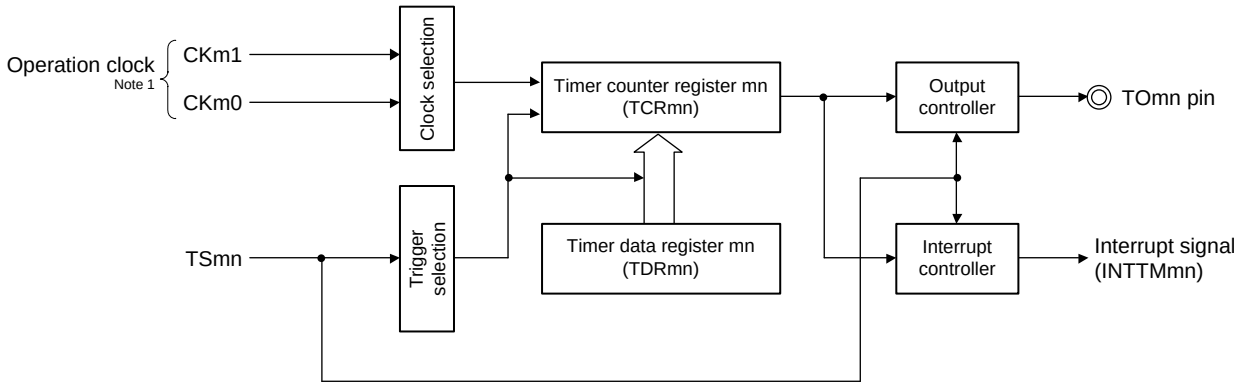
After that, the TCRmn register count down in synchronization with the count clock.

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.

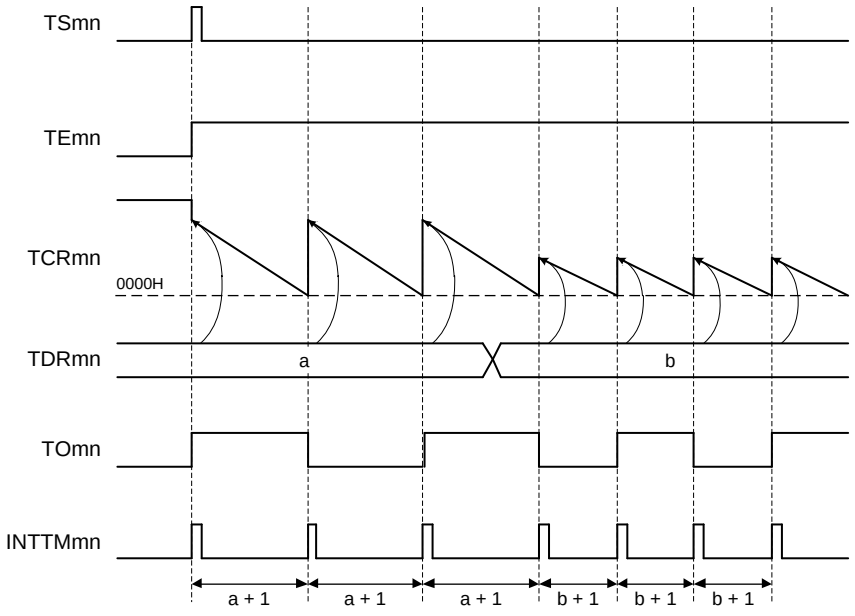
Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6-42. Block Diagram of Operation as Interval Timer/Square Wave Output



Note 1. When channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Figure 6-43. Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MDmn0 = 1)

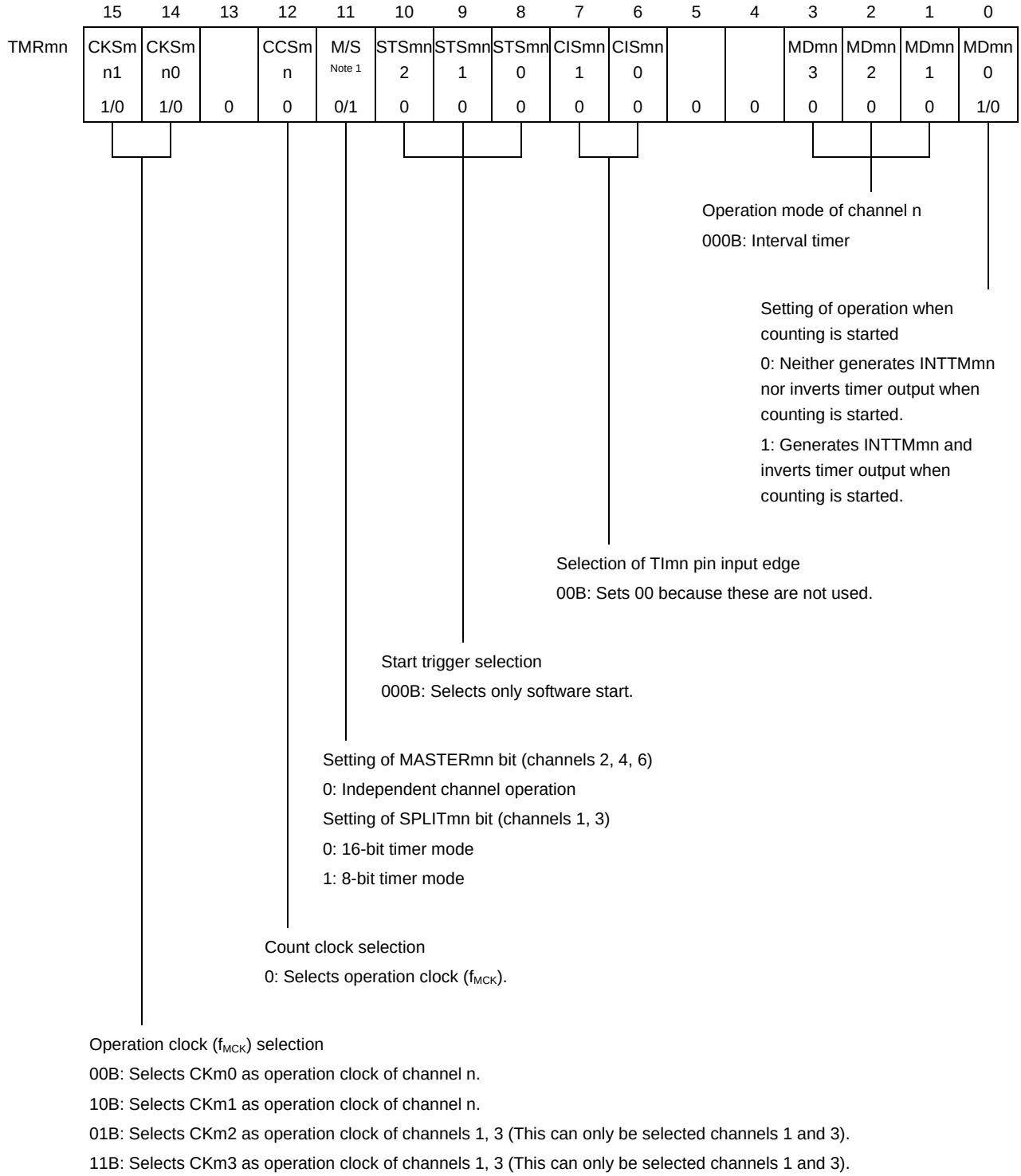


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Remark 2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)
TOmn: TOmn pin output signal

Figure 6-44. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (1/2)

(a) Timer mode register mn (TMRmn)



Note 1. TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6-44. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (2/2)

(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.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6-45. Operation Procedure of Interval Timer/Square Wave Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets timer mode register mn (TMRmn) (determines operation mode of channel). Sets interval (period) value to timer data register mn (TDRmn).	Channel stops operating. (Clock is supplied and some power is consumed.)
	To use the TOmn output Clears the TOMmn bit of timer output mode register m (TOMm) to 0 (master channel output mode). Clears the TOLmn bit to 0. Sets the TOmn bit and determines default level of the TOmn output. →	The TOmn pin goes into Hi-Z output state. The TOmn default setting level is output when the port mode register is in the output mode and the port register is 0.
	Sets the TOEmn bit to 1 and enables operation of TOmn. →	TOmn does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmn pin outputs the TOmn set level.
Operation start	(Sets the TOEmn bit to 1 only if using TOmn output and resuming operation.) Sets the TSmn (TSHm1, TSHm3) bit to 1. → The TSmn (TSHm1, TSHm3) bit automatically returns to 0 because it is a trigger bit.	TEmn (TEHm1, TEHm3) = 1, and count operation starts. Value of the TDRmn register is loaded to timer count register mn (TCRmn). INTTMmn is generated and TOmn performs toggle operation if the MDmn0 bit of the TMRmn register is 1.
During operation	Set values of the TMRmn register, TOMmn, and TOLmn bits cannot be changed. Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TOm and TOEm registers can be changed.	Counter (TCRmn) counts down. When count value reaches 0000H, the value of the TDRmn register is loaded to the TCRmn register again and the count operation is continued. By detecting TCRmn = 0000H, INTTMmn is generated and TOmn performs toggle operation. After that, the above operation is repeated.
Operation stop	The TTmn (TTHm1, TTHm3) bit is set to 1. → The TTmn (TTHm1, TTHm3) bit automatically returns to 0 because it is a trigger bit.	TEmn (TEHm1, TEHm3), and count operation stops. The TCRmn register holds count value and stops. The TOmn output is not initialized but holds current status.
	The TOEmn bit is cleared to 0 and value is set to the TOmn bit. →	The TOmn pin outputs the TOmn bit set level.

Operation is resumed.

(Remark is listed on the next page.)

Figure 6-45. 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. →	The TOmn pin output level is held by port function.
	When holding the TOmn pin output level is not necessary Setting not required.	
	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmn bit is cleared to 0 and the TOmn pin is set to port mode.)

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.8.2 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.

$$\text{Specified number of counts} = \text{Set value of TDRmn} + 1$$

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) of timer channel start register m (TSm) to 1.

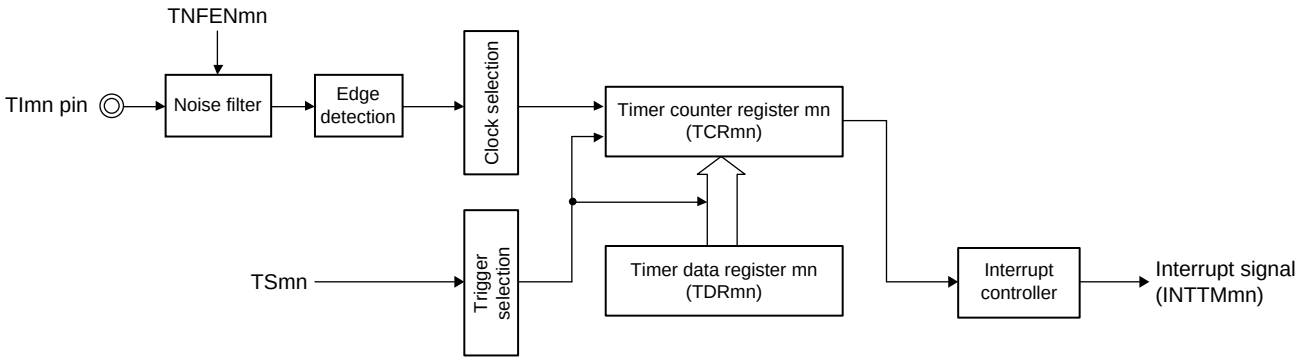
The TCRmn register counts down each time the valid input edge of the TImn pin has been detected. When TCRmn = 0000H, the TCRmn register loads the value of the TDRmn register again, and outputs INTTMmn.

After that, the above operation is repeated.

An irregular waveform that depends on external events is output from the TOmn pin. Stop the output by setting the TOEmn bit of timer output enable register m (TOEm) to 0.

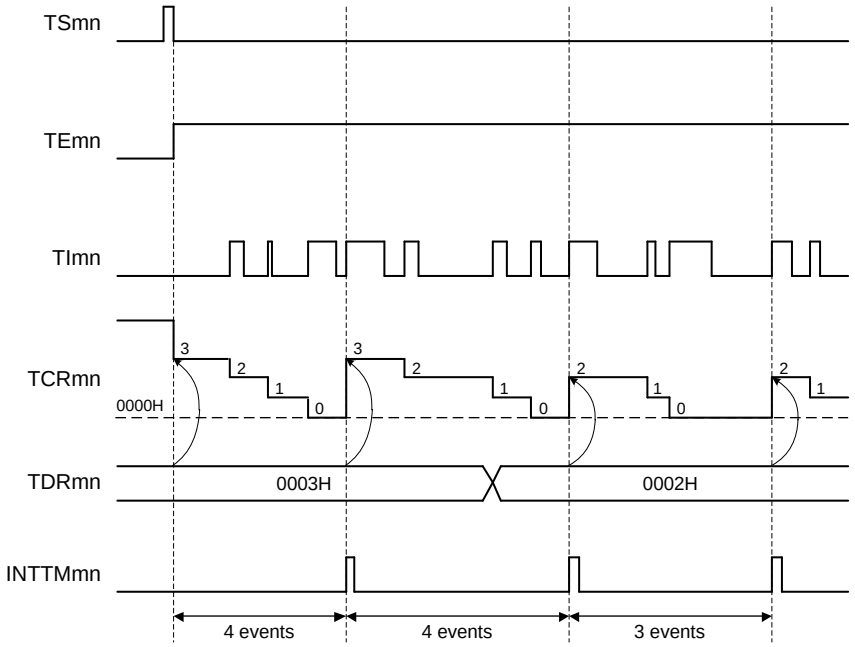
The TDRmn register can be rewritten at any time. The new value of the TDRmn register becomes valid during the next count period.

Figure 6-46. Block Diagram of Operation as External Event Counter



Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6-47. Example of Basic Timing of Operation as External Event Counter

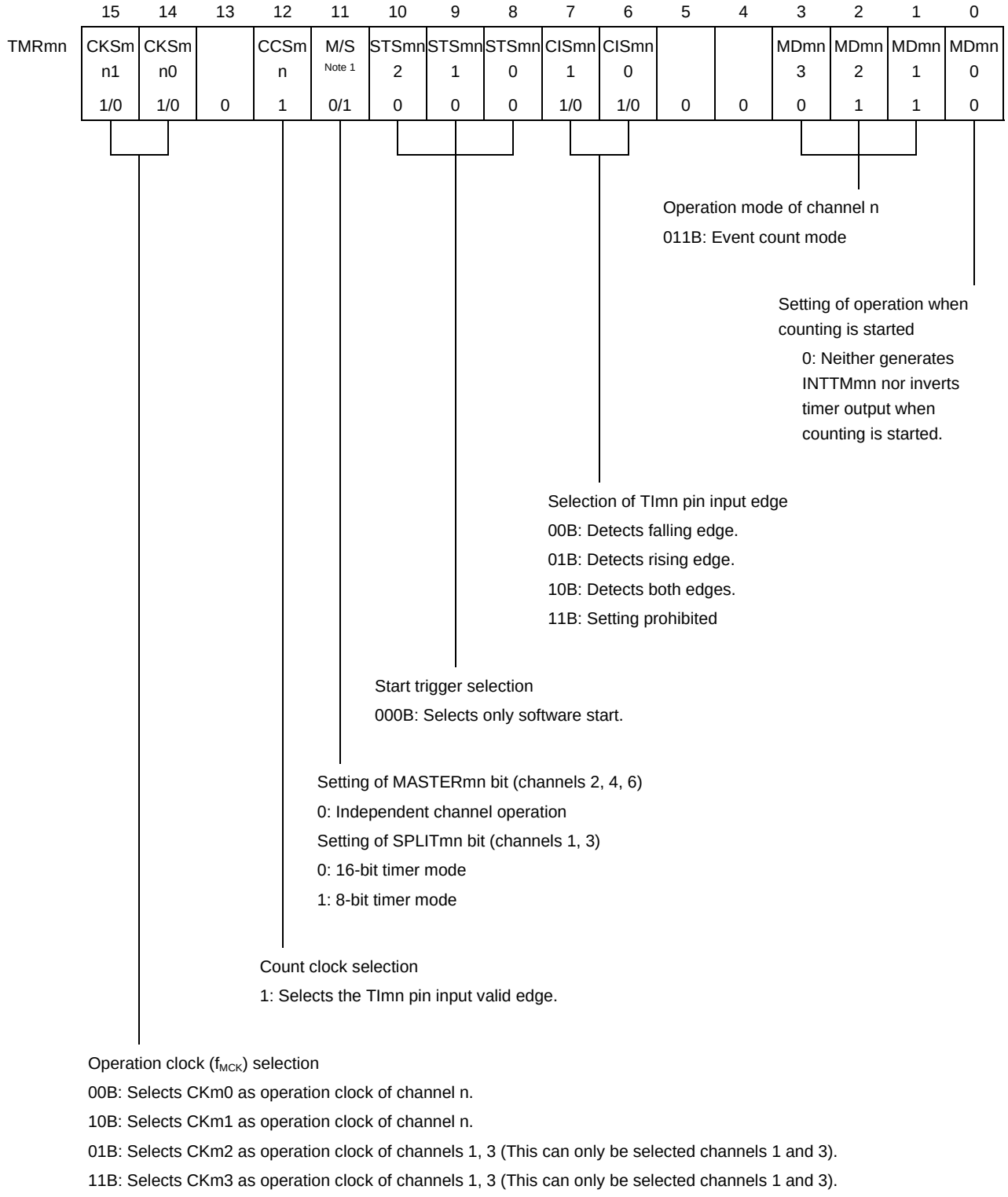


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Remark 2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TImn: TImn pin input signal
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)

Figure 6-48. Example of Set Contents of Registers in External Event Counter Mode (1/2)

(a) Timer mode register mn (TMRmn)



Note 1. TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6-48. Example of Set Contents of Registers in External Event Counter Mode (2/2)

(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn 0	0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

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

(d) Timer output level register m (TOLm)

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

(e) Timer output mode register m (TOMm)

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

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6-49. Operation Procedure When External Event Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable 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. →	Power-off status 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 7)

6.8.3 Operation as frequency divider (channels 0 and 3 only)

The timer array unit can be used as a frequency divider that divides a clock input to the TI0n pin and outputs the result from the TO0n pin.

The divided clock frequency output from TO0n can be calculated by the following expression.

- When rising edge/falling edge is selected:

$$\text{Divided clock frequency} = \text{Input clock frequency}/\{(\text{Set value of TDR0n} + 1) \times 2\}$$

- When both edges are selected:

$$\text{Divided clock frequency} \approx \text{Input clock frequency}/(\text{Set value of TDR0n} + 1)$$

Timer count register 0n (TCR0n) operates as a down counter in the interval timer mode.

After the channel start trigger bit (TS0n) of timer channel start register 0 (TS0) is set to 1, the TCR0n register loads the value of timer data register 0n (TDR0n) when the TI0n valid edge is detected. If the MD0n0 bit of timer mode register 0n (TMR0n) is 0 at this time, INTTM0n is not output and TO0n is not toggled. If the MD0n0 bit of timer mode register 0n (TMR0n) is 1, INTTM0n is output and TO0n is toggled.

After that, the TCR0n register counts down at the valid edge of the TI0n pin. When TCR0n = 0000H, it toggles TO0n. At the same time, the TCR0n register loads the value of the TDR0n register again, and continues counting.

If detection of both the edges of the TI0n pin is selected, the duty factor error of the input clock affects the divided clock period of the TO0n output.

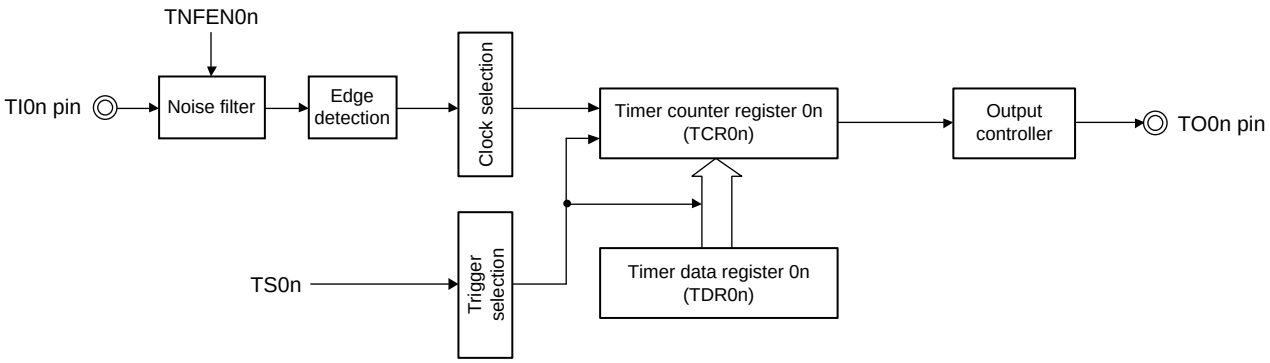
The period of the TO0n output clock includes a sampling error of one period of the operation clock.

$$\text{Clock period of TO0n output} = \text{Ideal TO0n output clock period} \pm \text{Operation clock period (error)}$$

The TDR0n register can be rewritten at any time. The new value of the TDR0n register becomes valid during the next count period.

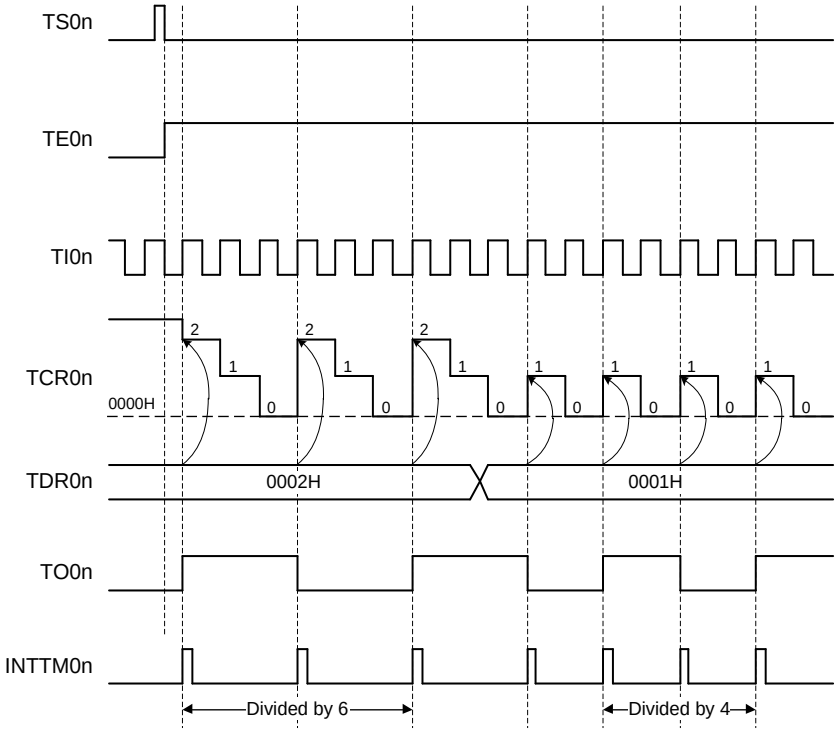
Remark n: Channel number (n = 0, 3)

Figure 6-50. Block Diagram of Operation as Frequency Divider



Remark n: Channel number (n = 0, 3)

Figure 6-51. Example of Basic Timing of Operation as Frequency Divider (MD0n0 = 1)

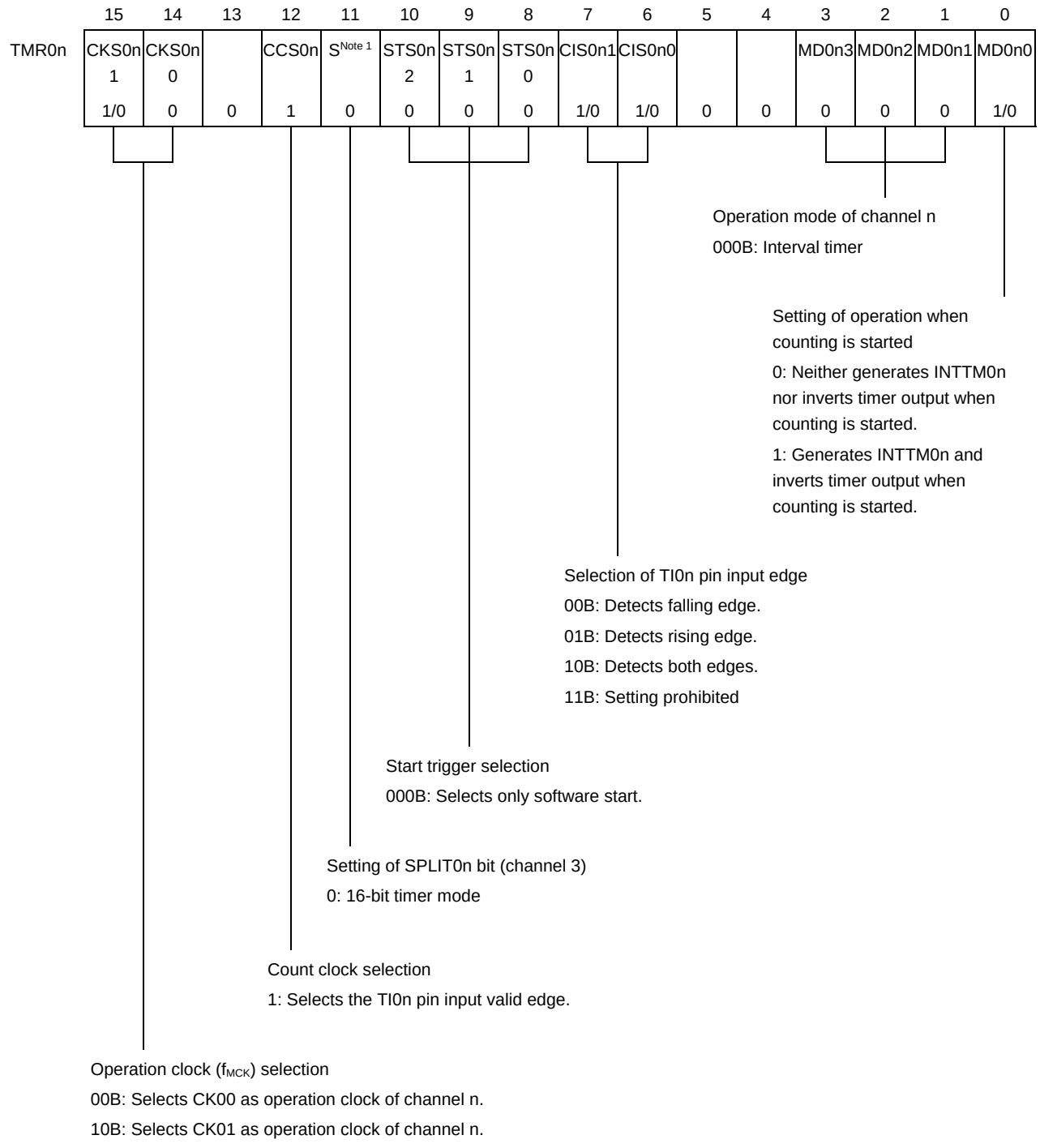


Remark 1. n: Channel number (n = 0, 3)

Remark 2. TS0n: Bit n of timer channel start register 0 (TS0)
TE0n: Bit n of timer channel enable status register 0 (TE0)
TI0n: TI0n pin input signal
TCR0n: Timer count register 0n (TCR0n)
TDR0n: Timer data register 0n (TDR0n)
TO0n: TO0n pin output signal

Figure 6-52. Example of Set Contents of Registers During Operation as Frequency Divider (1/2)

(a) Timer mode register 0n (TMR0n)



Note 1. Channel 3 only

Remark n: Channel number (n = 0, 3)

Figure 6-52. Example of Set Contents of Registers During Operation as Frequency Divider (2/2)

(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n	0: Outputs 0 from TO0n. 1: Outputs 1 from TO0n.
	1/0	

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n	0: Stops the TO0n output operation by counting operation. 1: Enables the TO0n output operation by counting operation.
	1/0	

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n	0: Cleared to 0 when master channel output mode (TOM0n = 0)
	0	

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n	0: Sets master channel output mode.
	0	

Remark n: Channel number (n = 0, 3)

Figure 6-53. Operation Procedure When Frequency Divider Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register 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 0n (TMR0n) (determines operation mode of channel and selects the detection edge). Sets interval (period) value to timer data register 0n (TDR0n).	Channel stops operating. (Clock is supplied and some power is consumed.)
	Clears the TOM0n bit of timer output mode register 0 (TOM0) to 0 (master channel output mode). Clears the TOL0n bit to 0. Sets the TO0n bit and determines default level of the TO0n output. →	The TO0n pin goes into Hi-Z output state. The TO0n default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOE0n bit to 1 and enables operation of TO0n. →	TO0n does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TO0n pin outputs the TO0n set level.

Remark n: Channel number (n = 0, 3)

Figure 6-53. Operation Procedure When Frequency Divider Function Is Used (2/2)

	Software Operation	Hardware Status
Operation start	Sets the TOE0n bit to 1 (only when operation is resumed). Sets the TS0n bit to 1. →	TE0n = 1, and count operation starts. Value of the TDR0n register is loaded to timer count register 0n (TCR0n). INTTM0n is generated and TO0n performs toggle operation if the MD0n0 bit of the TMR0n register is 1.
During operation	Set value of the TDR0n register can be changed. The TCR0n register can always be read. The TSR0n register is not used. Set values of the TO0 and TOE0 registers can be changed. Set values of the TMR0n register, TOM0n, and TOL0n bits cannot be changed.	Counter (TCR0n) counts down. When count value reaches 0000H, the value of the TDR0n register is loaded to the TCR0n register again, and the count operation is continued. By detecting TCR0n = 0000H, INTTM0n is generated and TO0n performs toggle operation. After that, the above operation is repeated.
Operation stop	The TT0n bit is set to 1. → The TT0n bit automatically returns to 0 because it is a trigger bit.	TE0n = 0, and count operation stops. The TCR0n register holds count value and stops. The TO0n output is not initialized but holds current status.
	The TOE0n bit is cleared to 0 and value is set to the TO0n bit. →	The TO0n pin outputs the TO0n set level.
TAU stop	To hold the TO0n pin output level Clears the TO0n bit to 0 after the value to be held is set to the port register. → When holding the TO0n pin output level is not necessary Setting not required.	The TO0n pin output level is held by port function.
	The TAU0EN bit of the PER0 register is cleared to 0.	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0n bit is cleared to 0 and the TO0n pin is set to port mode)

Operation is resumed.

Remark n: Channel number (n = 0, 3)

6.8.4 Operation as input pulse interval measurement

The count value can be captured on detection of a valid edge of TImn pin input and the interval of the pulse input to TImn pin can be measured. In addition, the count value can be captured by setting TSmn to 1 by software during the period of TEmn = 1.

For the UART0 baud rate correction, set bit 1 (ISC1) of the input switch control register (ISC) to 1.

In the following descriptions, read TI0n as RXD0. When the ISC1 bit is set to 1, the input signal of the serial data input (RXD0) pin is selected as a timer input (TI01). The width at the baud rate (transfer rate) of the other party in communications can be measured by using the input pulse interval measurement mode with the input edge signal of the start bit as a trigger.

The input pulse interval can be calculated by the following expression.

$$\text{TImn input pulse interval} = \text{Period of count clock} \times ((10000\text{H} \times \text{TSRmn: OVF}) + (\text{Capture value of TDRmn} + 1))$$

Caution The TImn pin input is sampled using the operation clock selected with the CKSmn bit of timer mode register mn (TMRmn), so an error of one cycle of the operation clock 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.

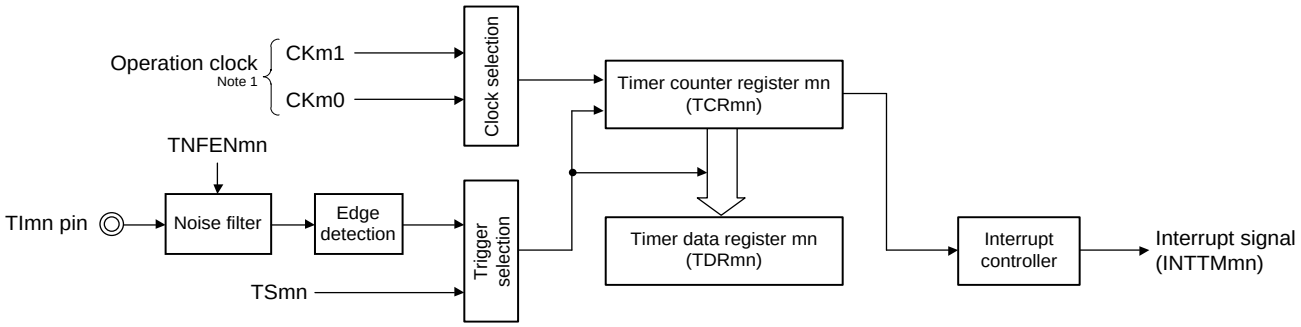
When the TImn pin input valid edge is detected, the count value of the TCRmn register is transferred (captured) to timer data register mn (TDRmn) and, at the same time, the TCRmn register is cleared to 0000H, and the INTTMmn is output. If the counter overflows at this time, the OVF bit of timer status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

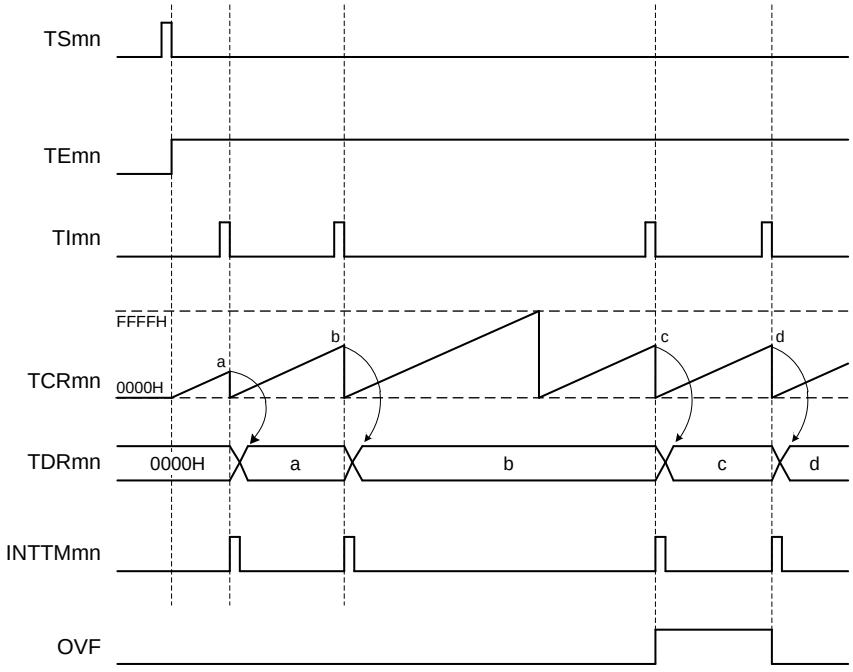
Figure 6-54. Block Diagram of Operation as Input Pulse Interval Measurement



Note 1. 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 7)

Figure 6-55. Example of Basic Timing of Operation as Input Pulse Interval Measurement (MDmn0 = 0)

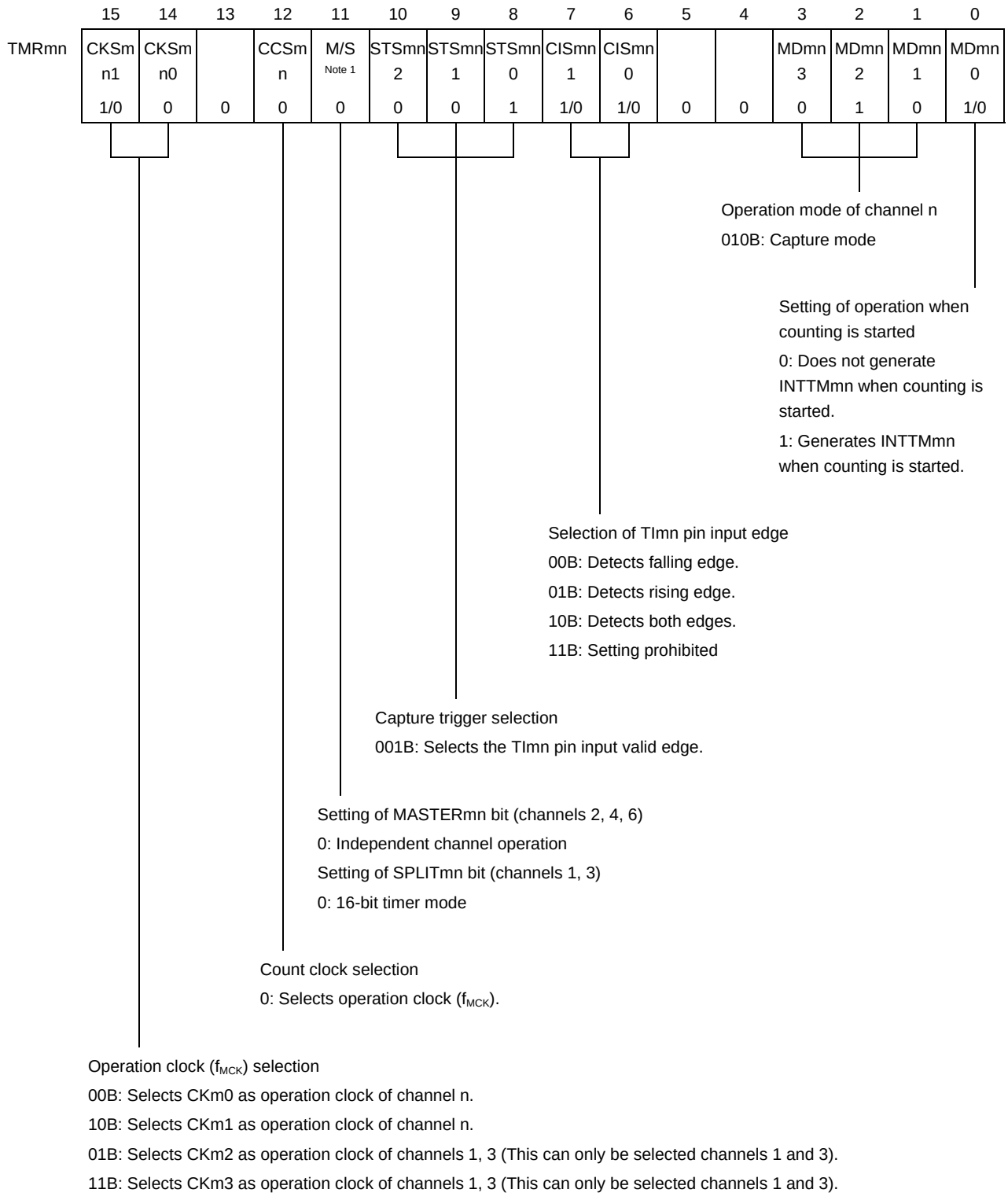


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Remark 2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TImn: TImn pin input signal
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)
OVF: Bit 0 of timer status register mn (TSRmn)

Figure 6-56. Example of Set Contents of Registers to Measure Input Pulse Interval (1/2)

(a) Timer mode register mn (TMRmn)



Note 1. TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6-56. Example of Set Contents of Registers to Measure Input Pulse Interval (2/2)

(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 TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

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

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6-57. Operation Procedure When Input Pulse Interval Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register m to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable 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 the 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. →	Power-off status 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 7)

6.8.5 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.

$$\text{Signal width of TImn input} = \text{Period of count clock} \times ((10000\text{H} \times \text{TSRmn: OVF}) + (\text{Capture value of TDRmn} + 1))$$

Caution The TImn pin input is sampled using the operation clock selected with the CKSmn bit of timer mode register mn (TMRmn), so an error of one cycle of the operation clock occurs.

Timer count register mn (TCRmn) operates as an up counter in the capture & one-count mode.

When the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, the TEmn bit is set to 1 and the TImn pin start edge detection wait status is set.

When the TImn pin input start edge (rising edge of the TImn pin input when the high-level width is to be measured) is detected, the counter counts up from 0000H in synchronization with the count clock. When the valid capture edge (falling edge of the TImn pin input when the high-level width is to be measured) is detected later, the count value is transferred to timer data register mn (TDRmn) and, at the same time, INTTMmn is output. If the counter overflows at this time, the OVF bit of timer status register mn (TSRmn) is set to 1. If the counter does not overflow, the OVF bit is cleared. The TCRmn register stops at the value “value transferred to the TDRmn register + 1”, and the TImn pin start edge detection wait status is set. After that, the above operation is repeated.

As soon as the count value has been captured to the TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSRmn register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.

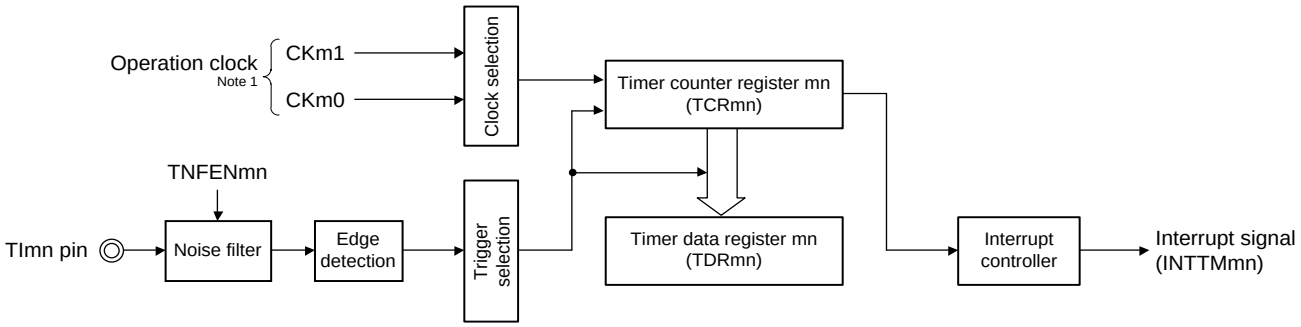
Whether the high-level width or low-level width of the TImn pin is to be measured can be selected by using the CISmn1 and CISmn0 bits of the TMRmn register.

Because this function is used to measure the signal width of the TImn pin input, the TSmn bit cannot be set to 1 while the TEmn bit is 1.

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

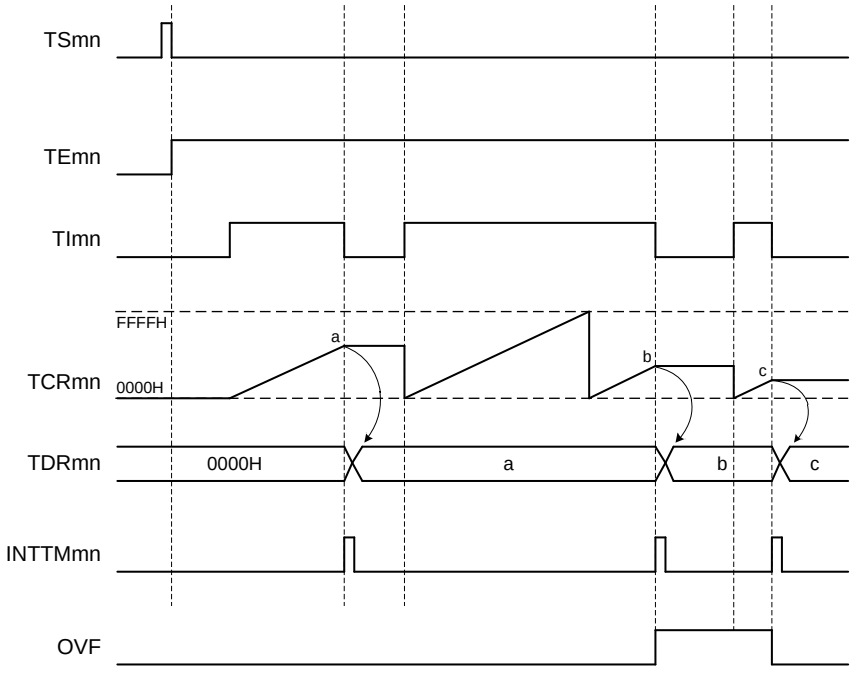
CISmn1, CISmn0 of TMRmn register = 11B: High-level width is measured.

Figure 6-58. Block Diagram of Operation as Input Signal High-/Low-Level Width Measurement



Note 1. For channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Figure 6-59. Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement

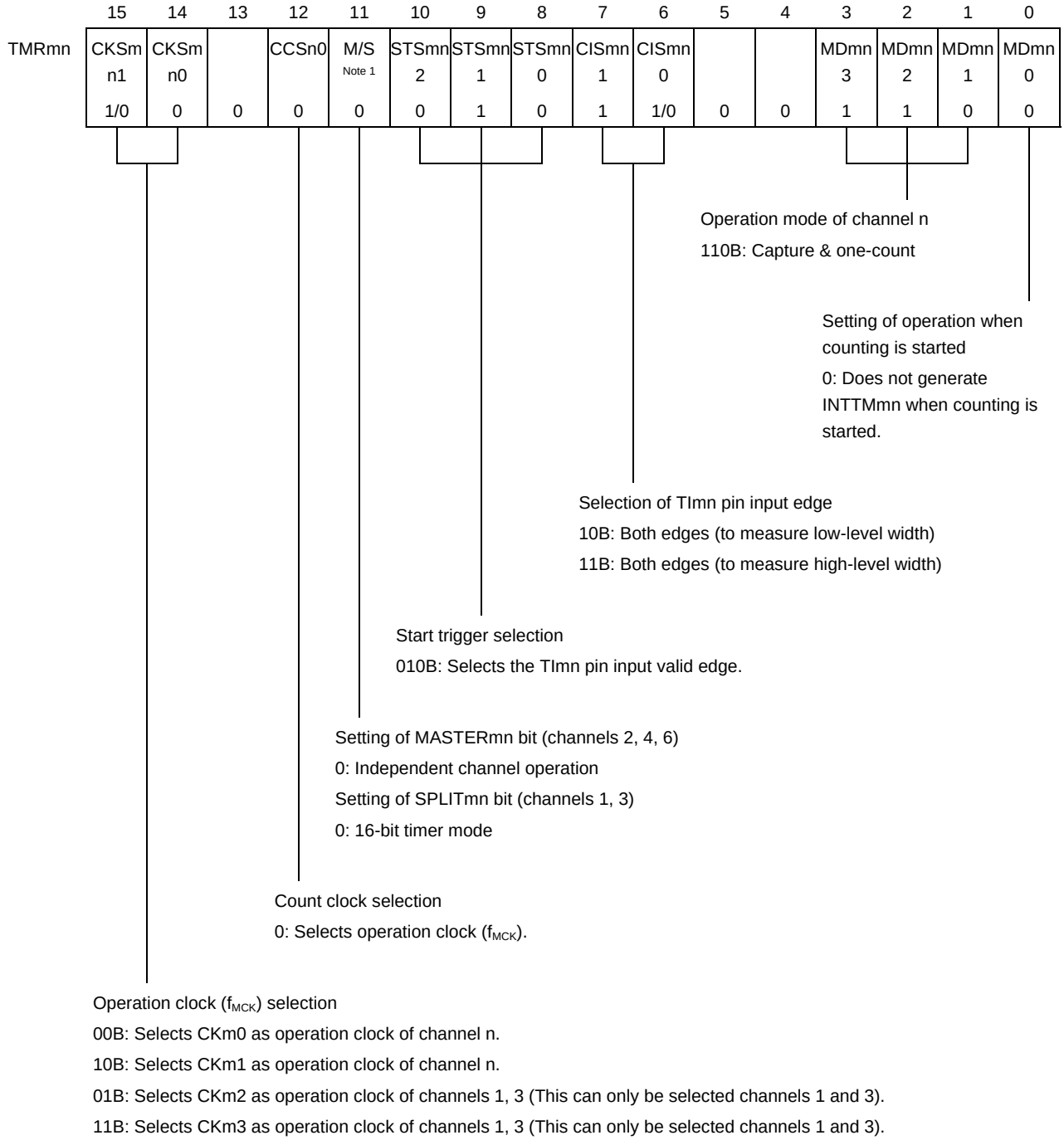


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Remark 2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TImn: TImn pin input signal
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)
OVF: Bit 0 of timer status register mn (TSRmn)

Figure 6-60. Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width (1/2)

(a) Timer mode register mn (TMRmn)



Note 1. TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6-60. Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width (2/2)

(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 TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

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

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6-61. Operation Procedure When Input Signal High-/Low-Level Width Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable 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	The TDRmn register can always be read. The TCRmn register can always be read. The TSRmn register can always be read. Set values of the TMRmn register, TOMmn, TOLmn, TOmn, and TOEmn bits cannot be changed.	When the TImn pin start edge is detected, the counter (TCRmn) counts up from 0000H. If a capture edge of the TImn pin is detected, the count value is transferred to timer data register mn (TDRmn) and INTTMmn is generated. If an overflow occurs at this time, the OVF bit of timer status register mn (TSRmn) is set; if an overflow does not occur, the OVF bit is cleared. The TCRmn register stops the count operation until the next TImn pin start edge is detected. After that, the above operation is repeated.
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops. The OVF bit of the TSRmn register is also held.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

6.8.6 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 is also possible to start counting down and generate INTTMmn (a timer interrupt) at any interval by setting TSmn to 1 by software while TEmn = 1.

The interrupt generation period can be calculated by the following expression.

$$\text{Generation period of INTTMmn (timer interrupt)} = \text{Period of count clock} \times (\text{Set value of TDRmn} + 1)$$

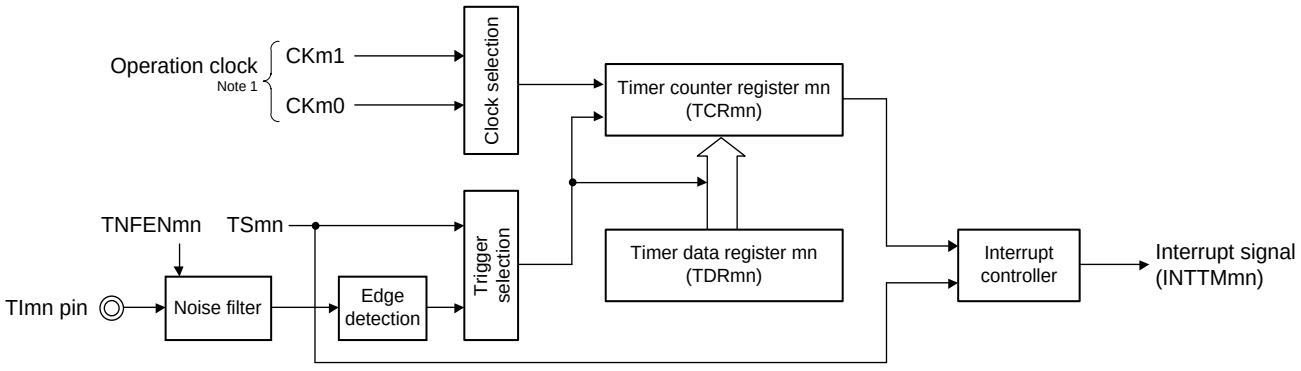
Timer count register mn (TCRmn) operates as a down counter in the one-count mode.

When the channel start trigger bit (TSmn, TSHm1, TSHm3) of timer channel start register m (TSm) is set to 1, the TEmn, TEHm1, TEHm3 bits are set to 1 and the TImn pin input valid edge detection wait status is set.

Timer count register mn (TCRmn) starts operating upon TImn pin input valid edge detection and loads the value of timer data register mn (TDRmn). The TCRmn register counts down from the value of the TDRmn register it has loaded, in synchronization with the count clock. When TCRmn = 0000H, it outputs INTTMmn and stops counting until the next TImn pin input valid edge is detected.

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

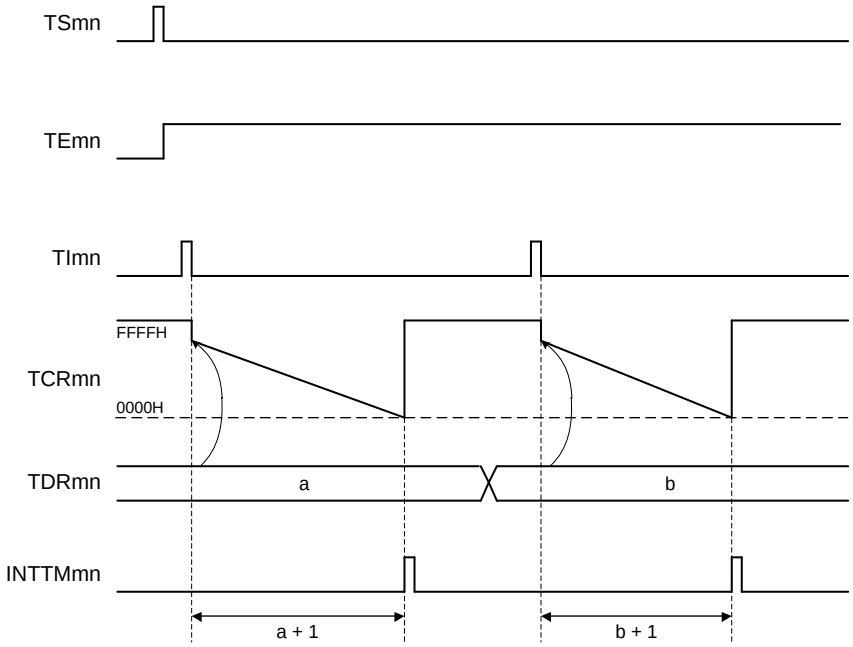
Figure 6-62. Block Diagram of Operation as Delay Counter



Note 1. For 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 7)

Figure 6-63. Example of Basic Timing of Operation as Delay Counter

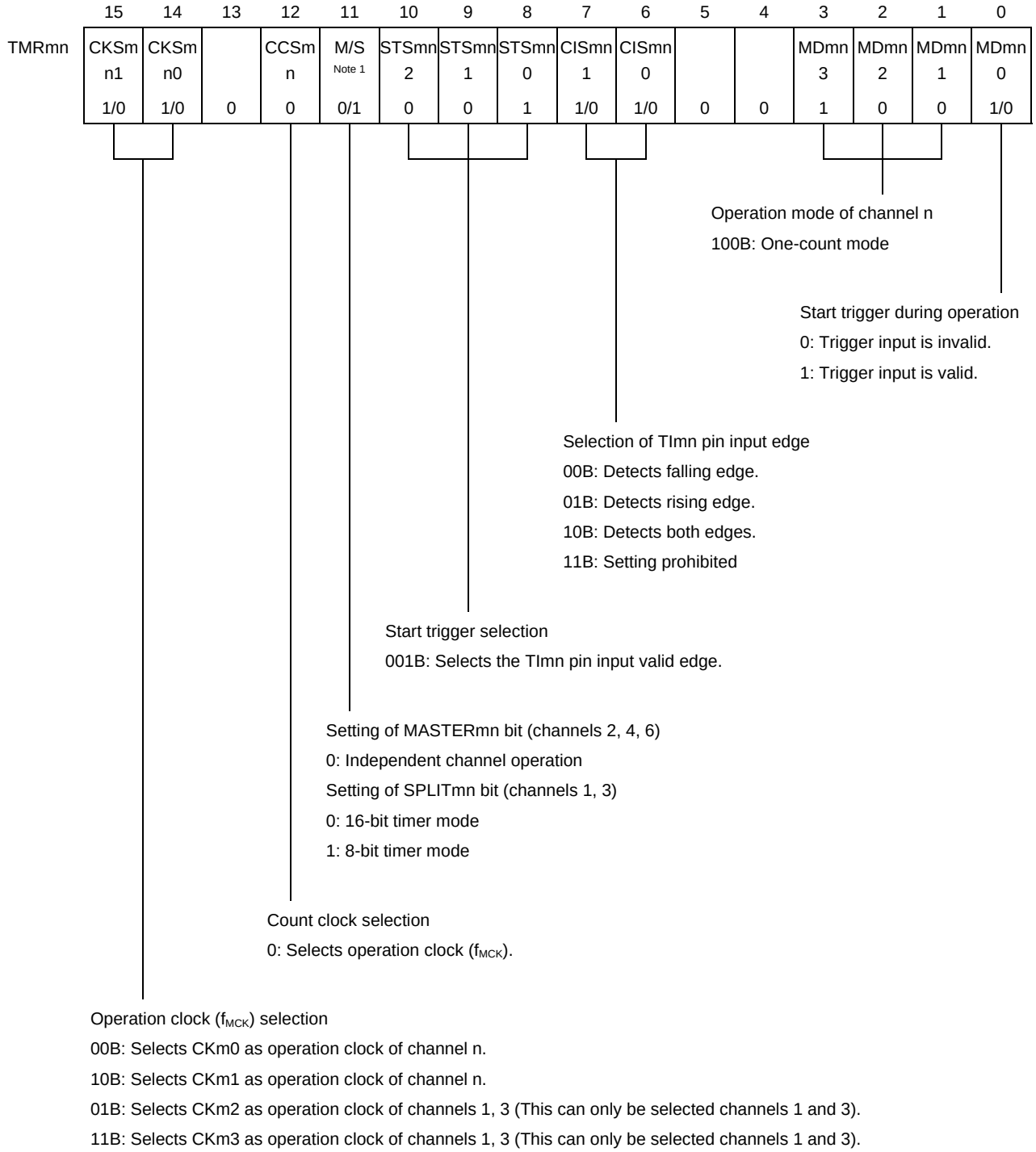


Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Remark 2. TSmn: Bit n of timer channel start register m (TSm)
TEmn: Bit n of timer channel enable status register m (TEm)
TImn: TImn pin input signal
TCRmn: Timer count register mn (TCRmn)
TDRmn: Timer data register mn (TDRmn)

Figure 6-64. Example of Set Contents of Registers to Delay Counter (1/2)

(a) Timer mode register mn (TMRmn)



Note 1. TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmn bit
TMRm0, TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6-64. Example of Set Contents of Registers to Delay Counter (2/2)

(b) Timer output register m (TOm)

	Bit n	
TOm	TOmn	0: Outputs 0 from TOmn.
	0	

(c) Timer output enable register m (TOEm)

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

(d) Timer output level register m (TOLm)

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

(e) Timer output mode register m (TOMm)

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

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 7)

Figure 6-65. Operation Procedure When Delay Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable 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. →	Power-off status 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 7)

6.9 Simultaneous Channel Operation Function of Timer Array Unit

6.9.1 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.

$$\text{Delay time} = \{\text{Set value of TDRmn (master)} + 2\} \times \text{Count clock period}$$

$$\text{Pulse width} = \{\text{Set value of TDRmp (slave)}\} \times \text{Count clock period}$$

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.

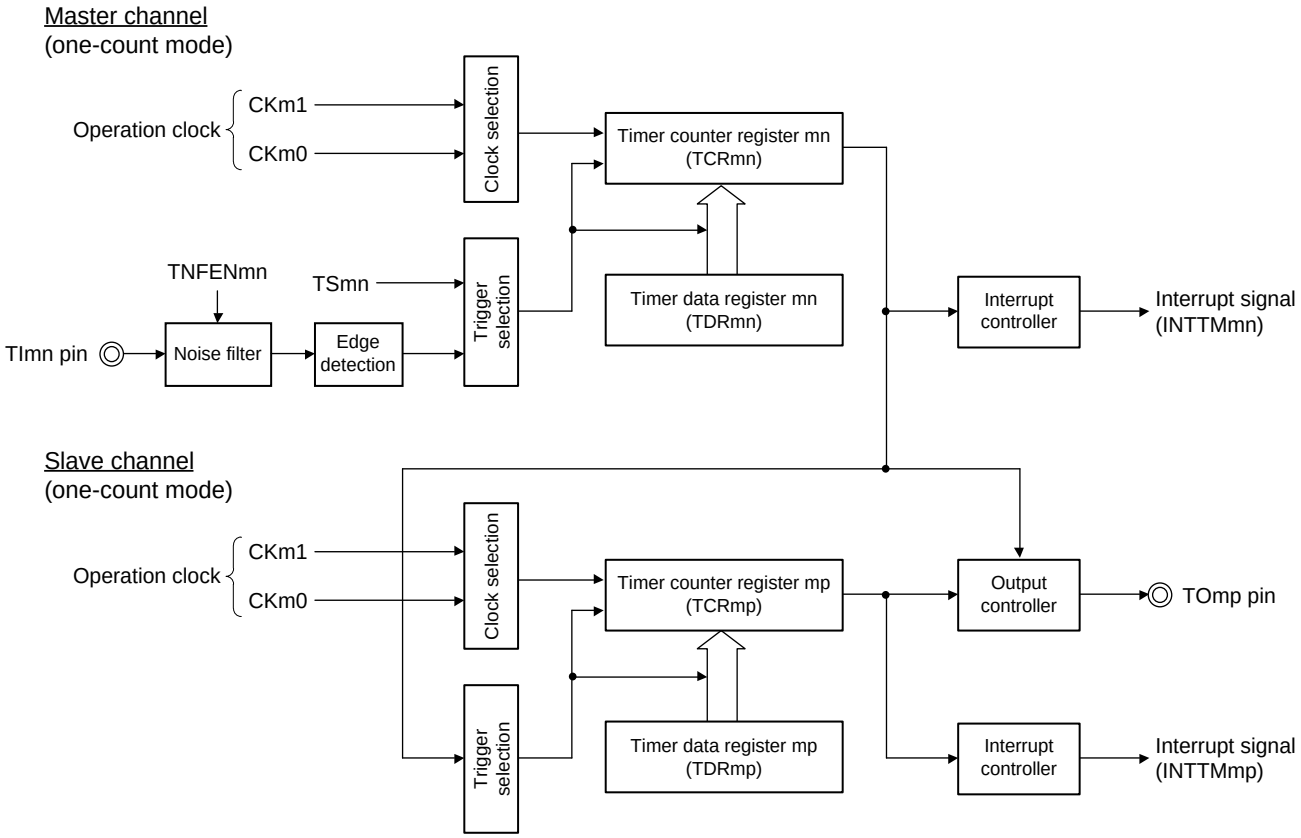
The slave channel operates in the one-count mode and counts the pulse width. The TCRmp register of the slave channel starts operation using INTTMmn of the master channel as a start trigger, and loads the value of the TDRmp register. The TCRmp register counts down from the value of The TDRmp register it has loaded, in synchronization with the count clock. When count value = 0000H, it outputs INTTMmp and stops counting until the next start trigger (INTTMmn of the master channel) is detected. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.

Instead of using the TImn pin input, a one-shot pulse can also be output using the software operation (TSmn = 1) as a start trigger.

Caution **The timing of loading of timer data register mn (TDRmn) of the master channel is different from that of the TDRmp register of the slave channel. If the TDRmn and TDRmp registers are rewritten during operation, therefore, an illegal waveform is output. Rewrite the TDRmn register after INTTMmn is generated and the TDRmp register after INTTMmp is generated.**

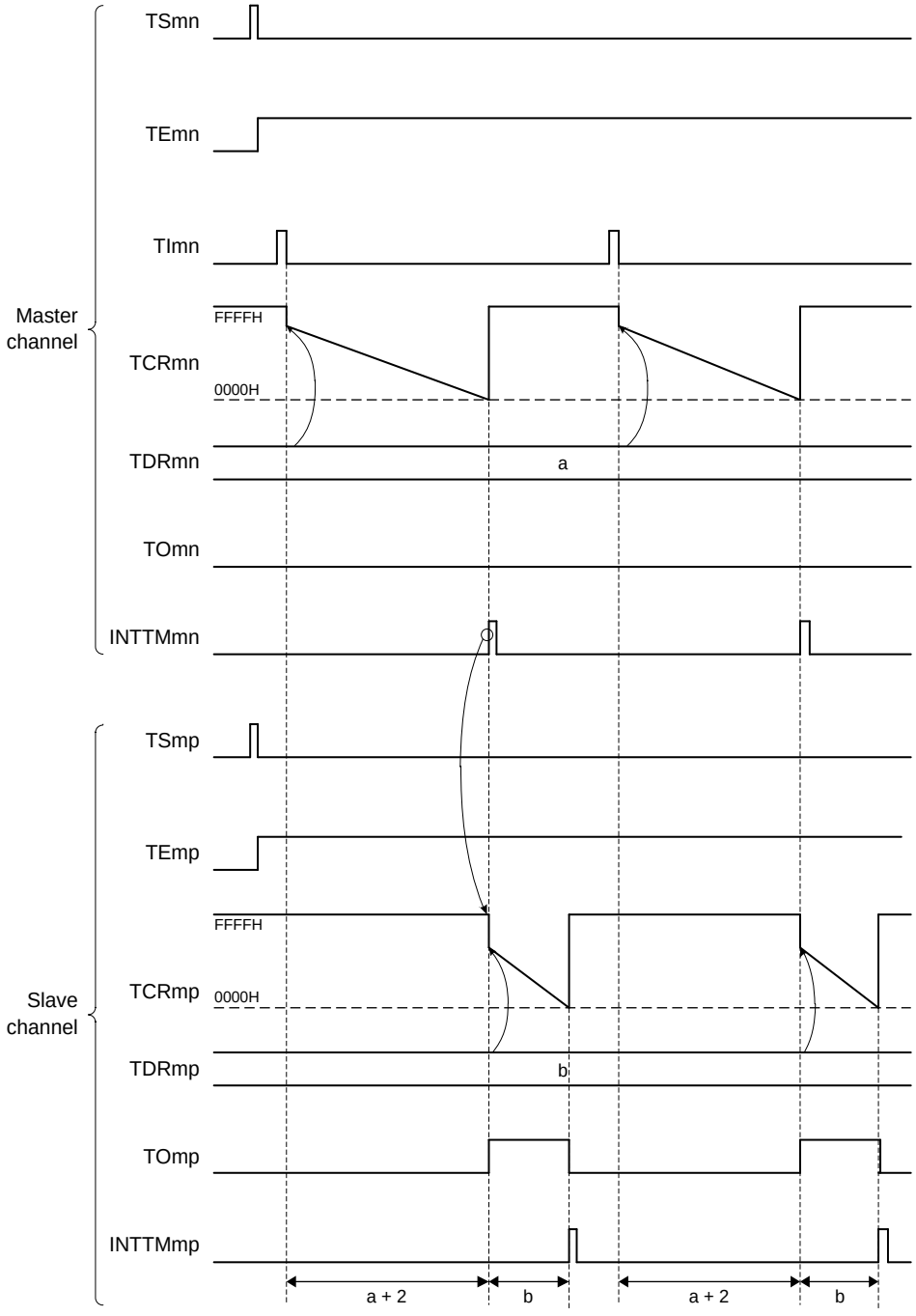
Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6-66. Block Diagram of Operation as One-Shot Pulse Output Function



Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6-67. Example of Basic Timing of Operation as One-Shot Pulse Output Function

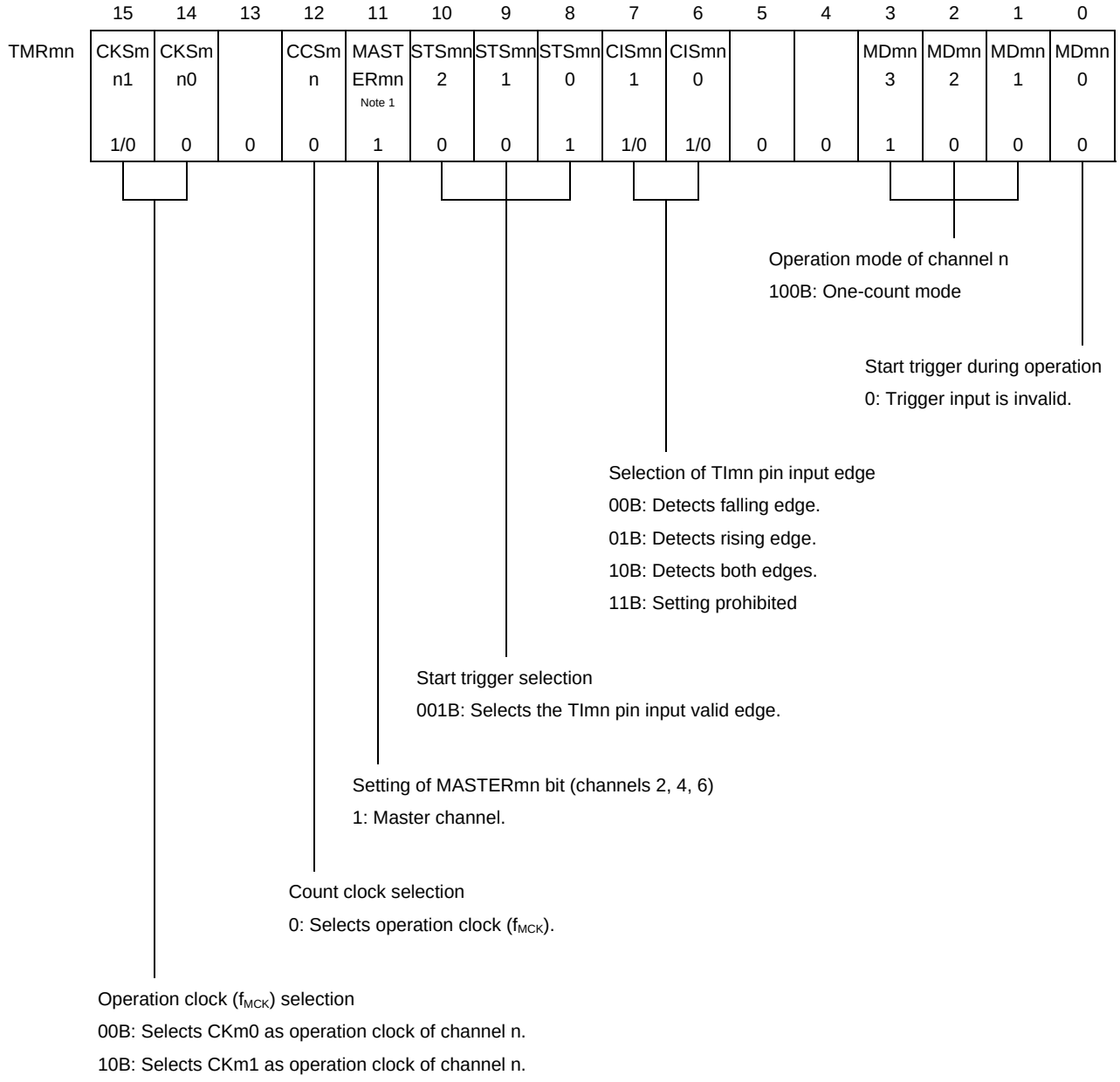


Remark 1. m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

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 6-68. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Master Channel) (1/2)

(a) Timer mode register mn (TMRmn)



Note 1. TMRm2, TMRm4, TMRm6: MASTERmn = 1
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)

Figure 6-68. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used

(Master Channel) (2/2)

(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 TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

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

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)

Figure 6-69. xample of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Slave Channel) (1/2)

(a) Timer mode register mp (TMRmp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmp	CKSmp1	CKSmp0		CCSmp	M/S Note 1	STSmp2	STSmp1	STSmp0	CISmp1	CISmp0			MDmp3	MDmp2	MDmp1	MDmp0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0

Operation mode of channel p
100B: One-count mode

Start trigger during operation
0: Trigger input is invalid.

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

Start trigger selection
100B: Selects INTTMmn of master channel.

Setting of MASTERmp bit (channels 2, 4, 6)
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.

Note 1. TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmp bit
TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6-69. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Slave Channel) (2/2)

(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.

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6-70. Operation Procedure of One-Shot Pulse Output Function (1/3)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets the corresponding bit of the noise filter enable register 1 (NFEN1) to 1. Sets timer mode register mn, mp (TMRmn, TMRmp) of two channels to be used (determines operation mode of channels). An output delay is set to timer data register mn (TDRmn) of the master channel, and a pulse width is set to the TDRmp register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOMmp bit of timer output mode register m (TOMm) is set to 1 (slave channel output mode). Sets the TOLmp bit. Sets the TOmp bit and determines default level of the TOmp output. →	The TOmp pin goes into Hi-Z output state. The TOmp default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOEmp bit to 1 and enables operation of TOmp. → Clears the port register and port mode register to 0. →	TOmp does not change because channel stops operating. The TOmp pin outputs the TOmp set level.

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6-70. Operation Procedure of One-Shot Pulse Output Function (2/3)

	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 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.
	The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	
	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 1].	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 count operation stops until the next start trigger detection. The slave channel, triggered by INTTMmn of the master channel, loads the value of the TDRmp register to the TCRmp register, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of 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. →	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.

Operation is resumed.

Note 1. Do not set the TSmn bit of the slave channel to 1.

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6-70. Operation Procedure of One-Shot Pulse Output Function (3/3)

	Software Operation	Hardware Status
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. →	The TOmp pin output level is held by port function.
	When holding the TOmp pin output level is not necessary Setting not required.	
	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmp bit is cleared to 0 and the TOmp pin is set to port mode)

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

6.9.2 Operation as PWM function

Two channels can be used as a set to generate a pulse of any period and duty factor.

The period and duty factor of the output pulse can be calculated by the following expressions.

$$\text{Pulse period} = \{\text{Set value of TDRmn (master)} + 1\} \times \text{Count clock period}$$

$$\text{Duty factor}[\%] = \{\text{Set value of TDRmp (slave)}\}/\{\text{Set value of TDRmn (master)} + 1\} \times 100$$

$$0\%\ \text{output: Set value of TDRmp (slave)} = 0000\text{H}$$

$$100\%\ \text{output: Set value of TDRmp (slave)} \geq \{\text{Set value of TDRmn (master)} + 1\}$$

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.

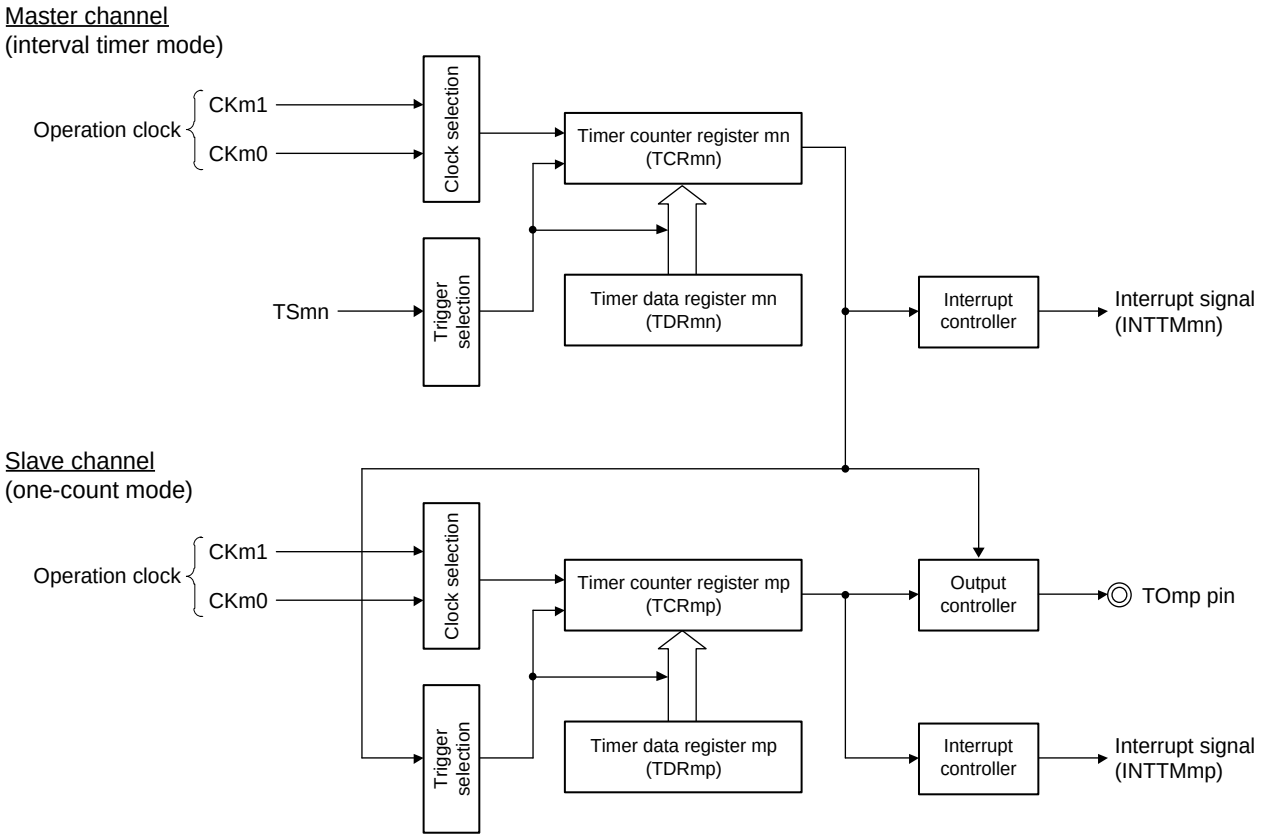
If two channels are used to output a PWM waveform, the period until the slave channel counts down to 0000H is the PWM output (TOmp) duty.

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.

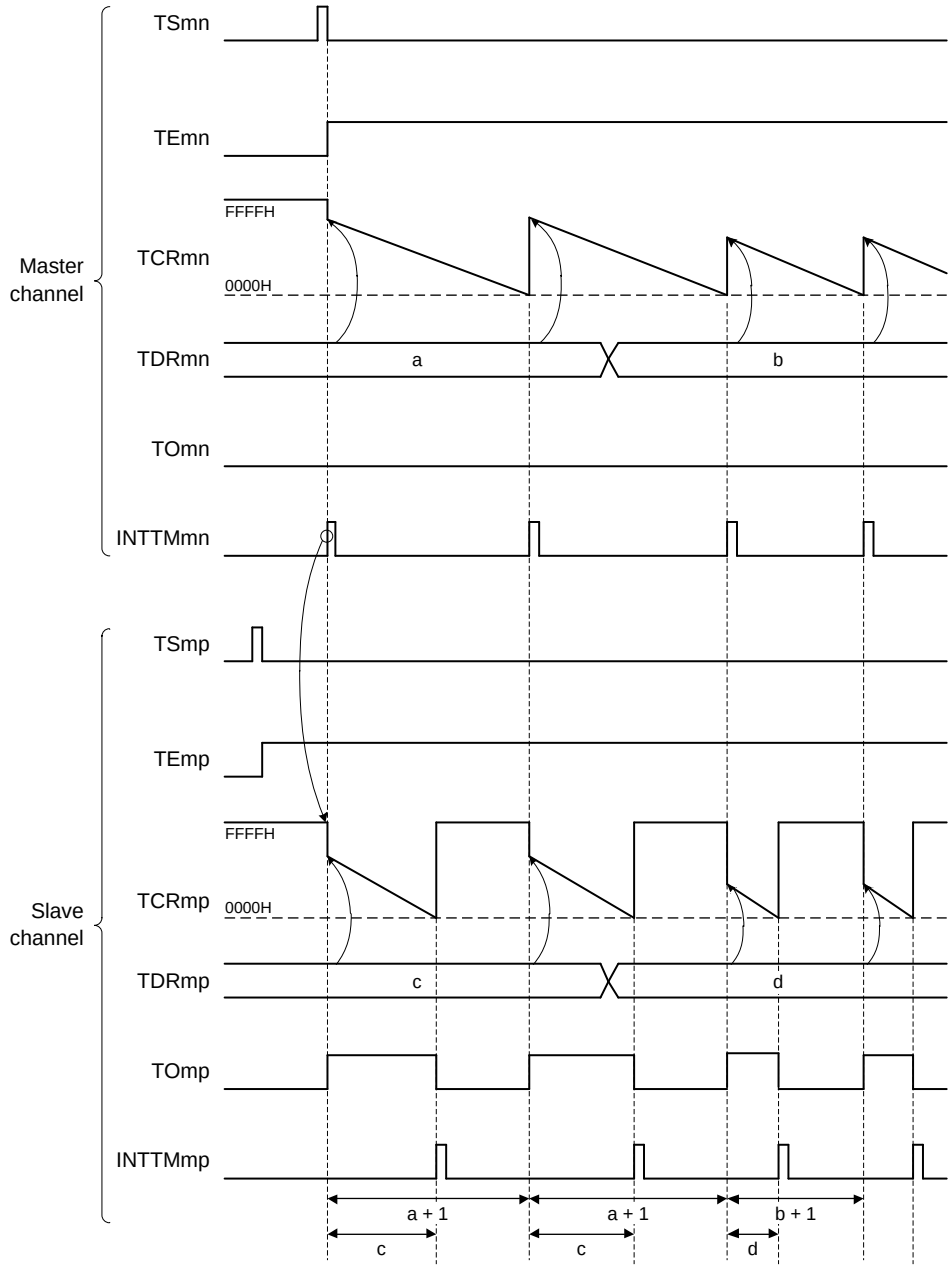
Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6-71. Block Diagram of Operation as PWM Function



Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6-72. Example of Basic Timing of Operation as PWM Function

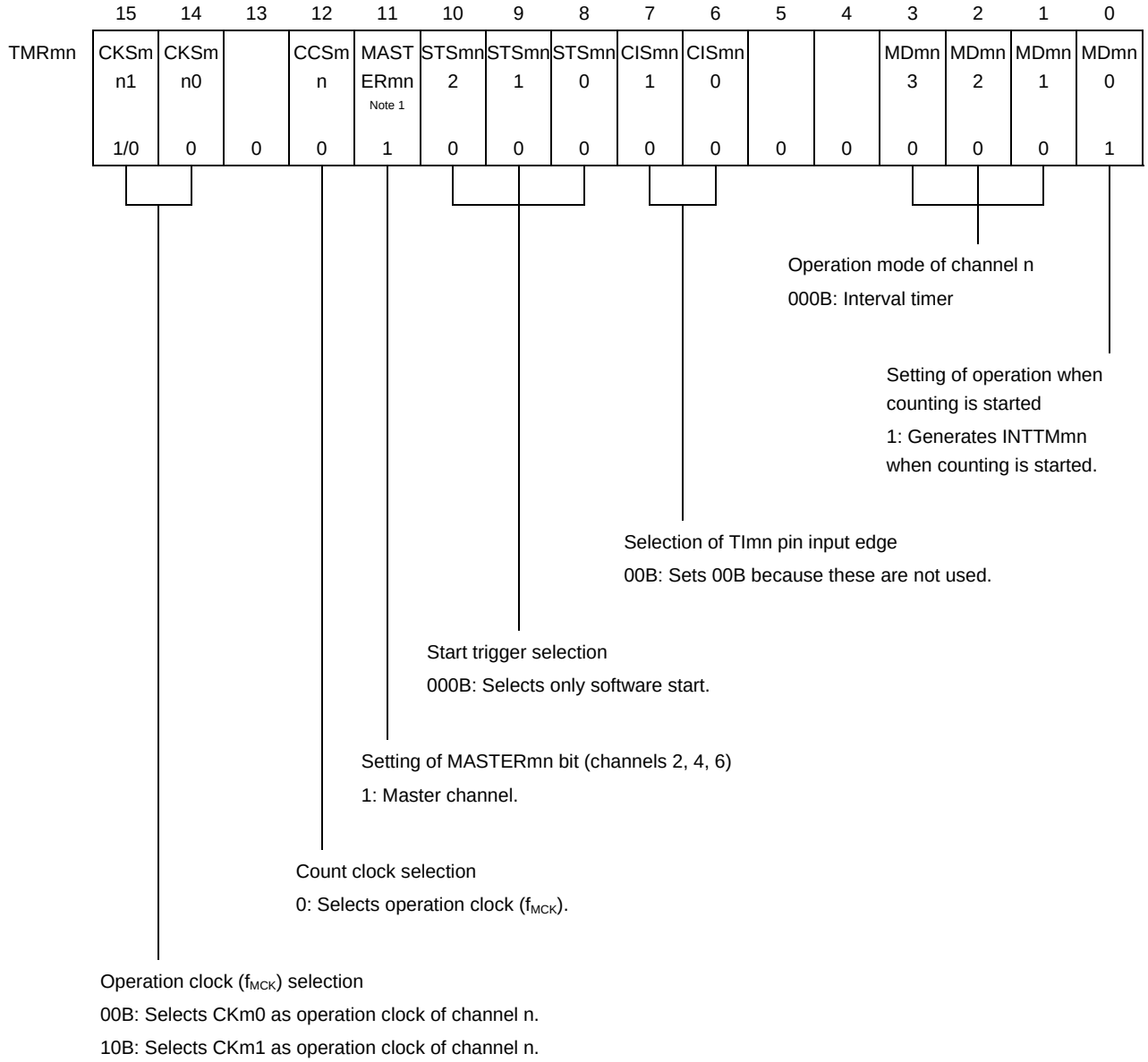


Remark 1. m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

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 6-73. Example of Set Contents of Registers When PWM Function (Master Channel) Is Used (1/2)

(a) Timer mode register mn (TMRmn)



Note 1. TMRm2, TMRm4, TMRm6: MASTERmn = 1
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)

Figure 6-73. Example of Set Contents of Registers When PWM Function (Master Channel) Is Used (2/2)

(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 TOMmn = 0 (master channel output mode).

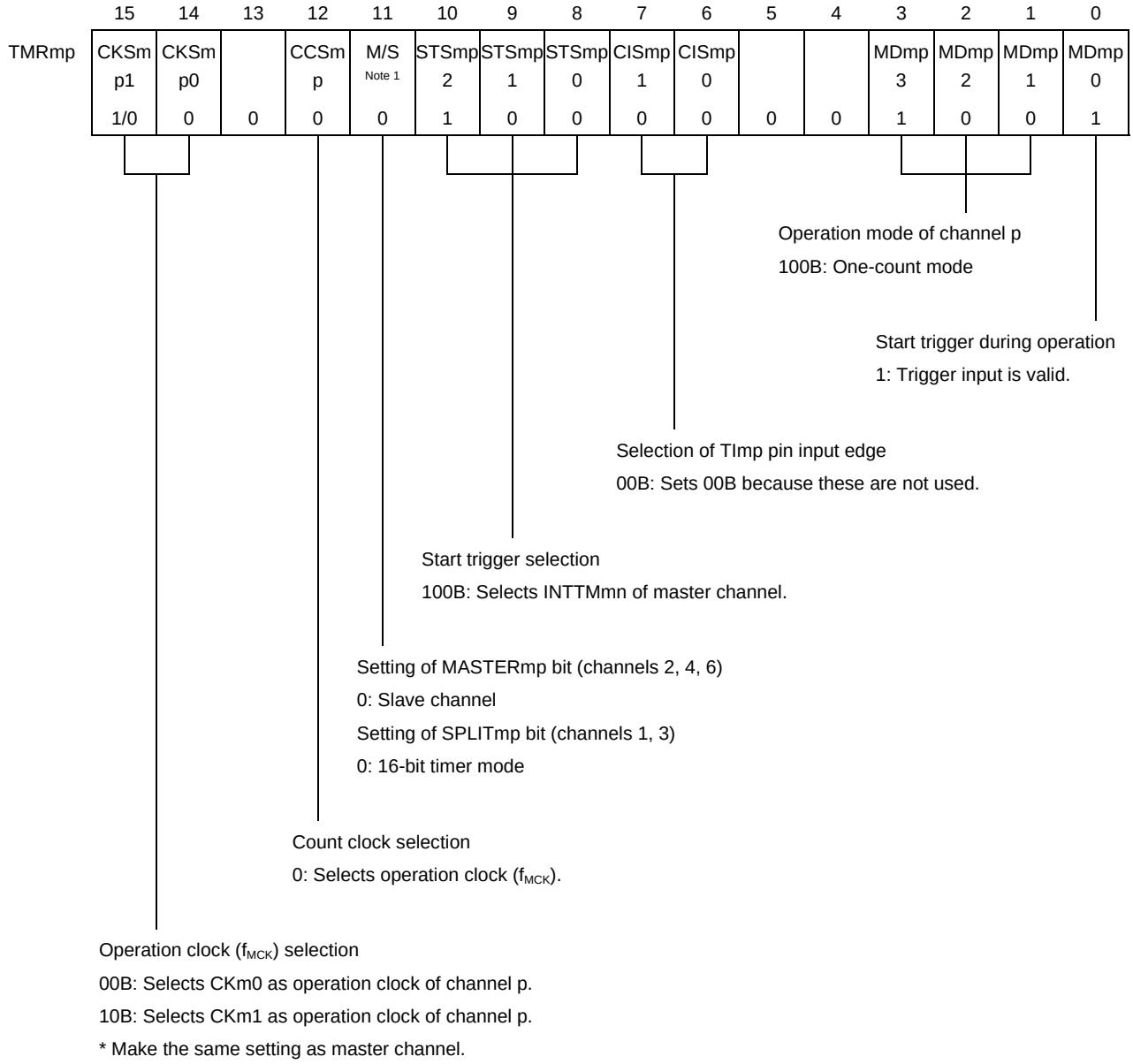
(e) Timer output mode register m (TOMm)

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

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)

Figure 6-74. Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used (1/2)

(a) Timer mode register mp (TMRmp)



Note 1. TMRm2, TMRm4, TMRm6: MASTERmn bit
TMRm1, TMRm3: SPLITmp bit
TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6-74. Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used (2/2)

(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.

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

Figure 6-75. Operation Procedure When PWM Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. ⟶	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets timer mode register mn, mp (TMRmn, TMRmp) of two channels to be used (determines operation mode of channels). An interval (period) value is set to timer data register mn (TDRmn) of the master channel, and a duty factor is set to the TDRmp register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOMmp bit of timer output mode register m (TOMm) is set to 1 (slave channel output mode). Sets the TOLmp bit. Sets the TOmp bit and determines default level of the TOmp output. ⟶	The TOmp pin goes into Hi-Z output state. The TOmp default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOEmp bit to 1 and enables operation of TOmp. ⟶ Clears the port register and port mode register to 0. ⟶	TOmp does not change because channel stops operating. The TOmp pin outputs the TOmp set level.

(Remark is listed on the next page.)

Figure 6-75. 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. →	TEmn = 1, TEmp = 1
	The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	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.	Master channel loads the value of the TDRmn register to timer count register mn (TCRmn), and the counter starts counting 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. →	TEmn, TEmp = 0, and count operation stops.
	The TTmn and TTmp bits automatically return to 0 because they are trigger bits.	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. →	The TOmp pin output level is held by port function.
	When holding the TOmp pin output level is not necessary Setting not required.	
	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status
		All circuits are initialized and SFR of each channel is also initialized. (The TOmp bit is cleared to 0 and the TOmp pin is set to port mode)

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4, 6)
p: Slave channel number (n < p ≤ 7)

6.9.3 Operation as multiple PWM output function

By extending the PWM function and using multiple slave channels, many PWM waveforms with different duty values can be output.

For example, when using two slave channels, the period and duty factor of an output pulse can be calculated by the following expressions.

$$\text{Pulse period} = \{\text{Set value of TDRmn (master)} + 1\} \times \text{Count clock period}$$

$$\text{Duty factor 1}[\%] = \{\text{Set value of TDRmp (slave 1)}\}/\{\text{Set value of TDRmn (master)} + 1\} \times 100$$

$$\text{Duty factor 2}[\%] = \{\text{Set value of TDRmq (slave 2)}\}/\{\text{Set value of TDRmn (master)} + 1\} \times 100$$

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.

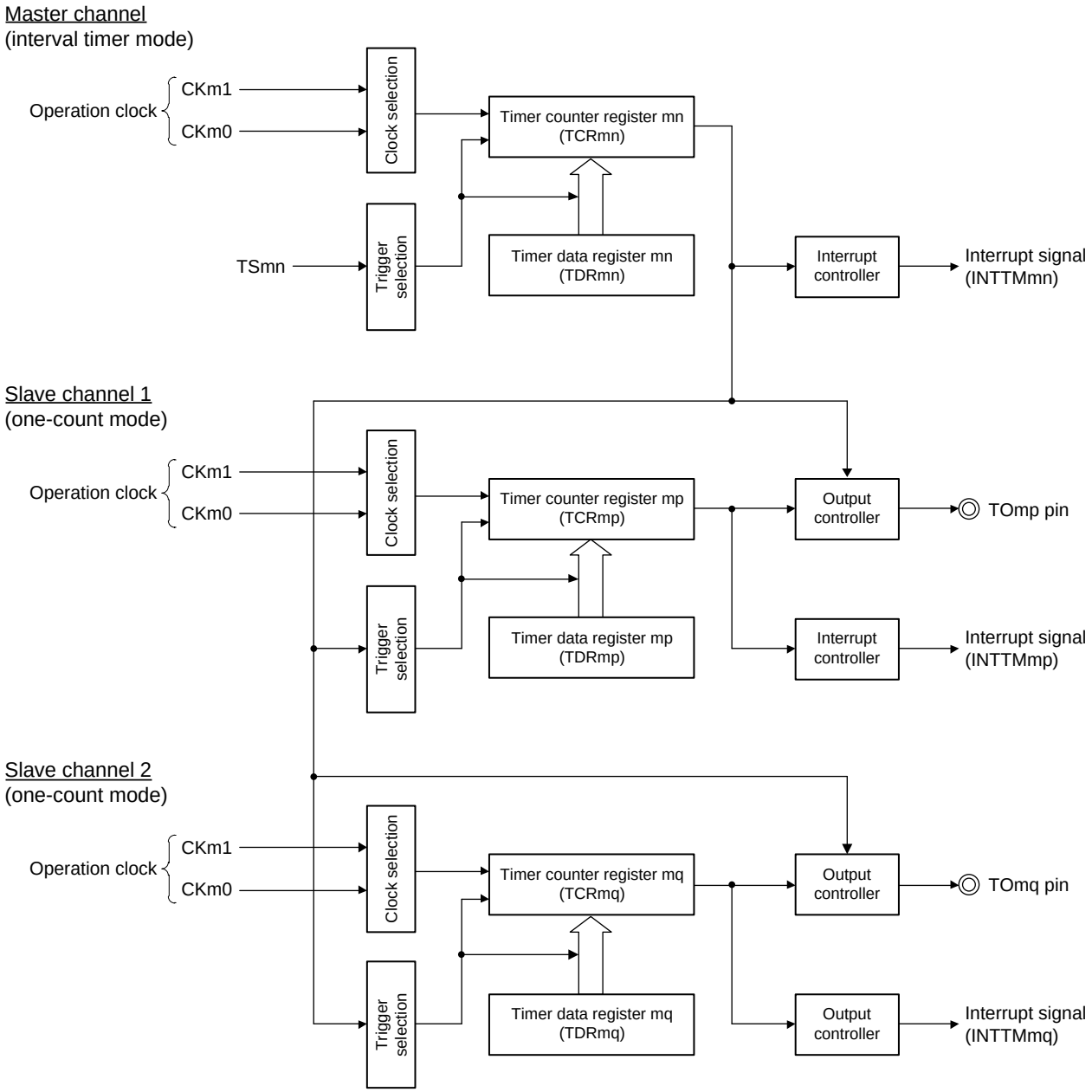
In the same way as the TCRmp register of the slave channel 1, the TCRmq register of the slave channel 2 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmq pin. The TCRmq register loads the value of the TDRmq register, using INTTMmn of the master channel as a 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 seven types of PWM signals can be output at the same time.

Caution **To rewrite both timer data register mn (TDRmn) of the master channel and the TDRmp register of the slave channel 1, write access is necessary at least twice. Since the values of the TDRmn and TDRmp registers are loaded to the TCRmn and TCRmp registers after INTTMmn is generated from the master channel, if rewriting is performed separately before and after generation of INTTMmn from the master channel, the TOmp pin cannot output the expected waveform. To rewrite both the TDRmn register of the master and the TDRmp register of the slave, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel (This applies also to the TDRmq register of the slave channel 2).**

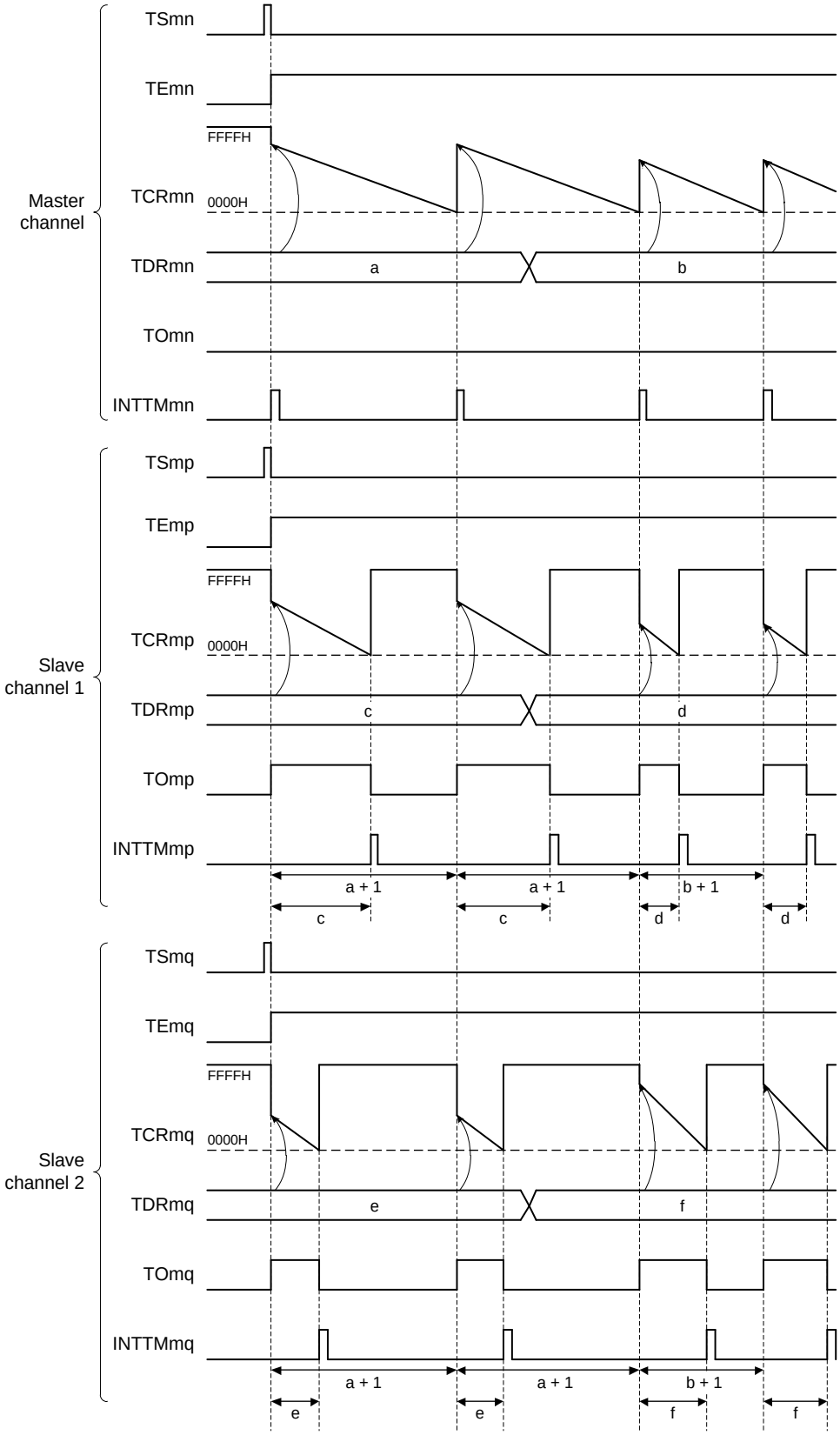
Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4)
p: Slave channel number, q: Slave channel number
$n < p < q \leq 7$ (Where p and q are integers greater than n)

Figure 6-76. Block Diagram of Operation as Multiple PWM Output Function (Output Two Types of PWMs)



Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 7 (Where p and q are integers greater than n)

Figure 6-77. Example of Basic Timing of Operation as Multiple PWM Output Function (Output Two Types of PWMs)



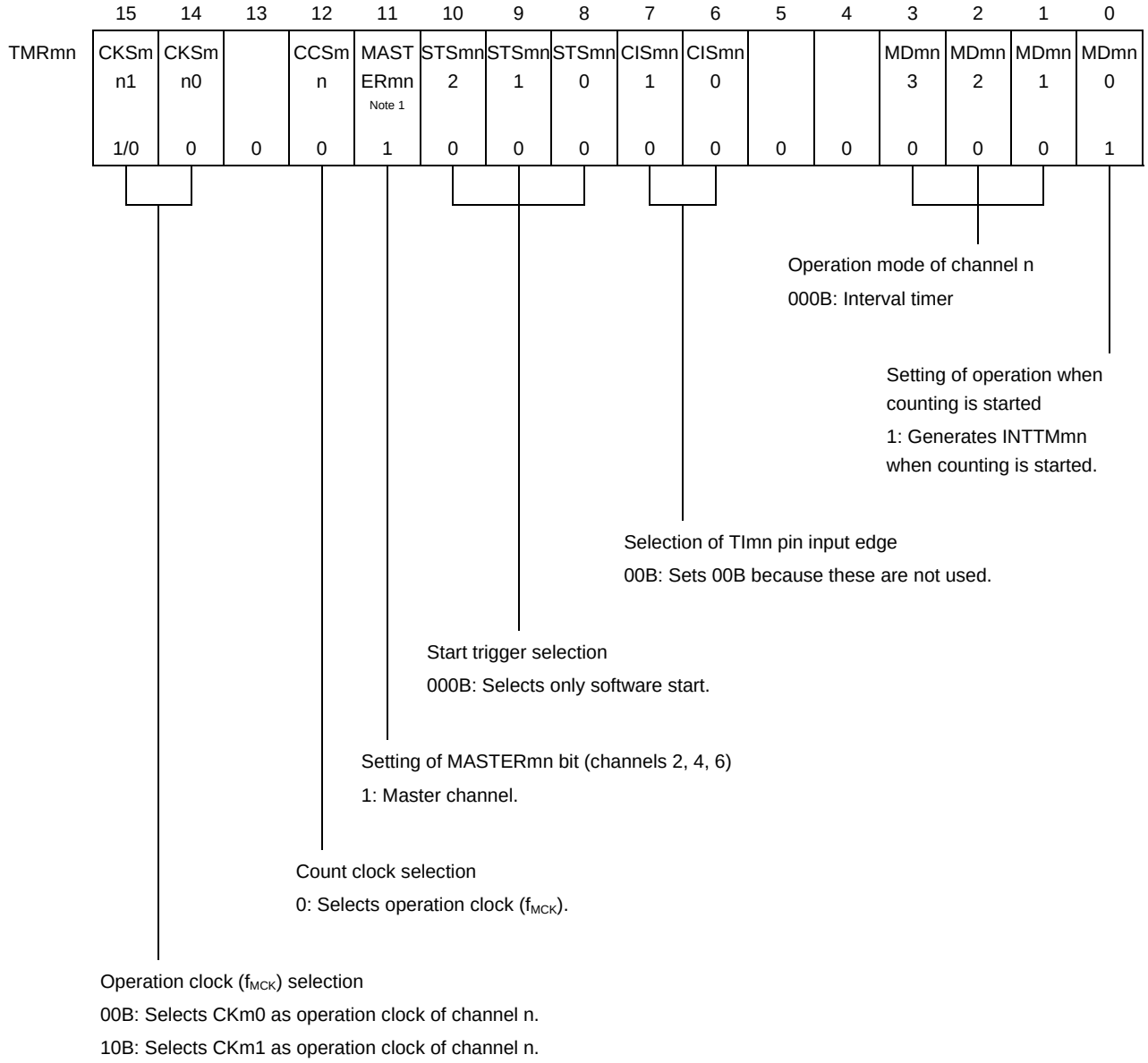
(Remarks are listed on the next page.)

Remark 1. m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 7 (Where p and q are integers 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 6-78. Example of Set Contents of Registers When Multiple PWM Output Function (Master Channel) Is Used (1/2)

(a) Timer mode register mn (TMRmn)



Note 1. TMRm2, TMRm4, TMRm6: MASTERmn = 1
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4)

Figure 6-78. Example of Set Contents of Registers When Multiple PWM Output Function (Master Channel) Is Used (2/2)

(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 TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

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

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4)

Figure 6-79. Example of Set Contents of Registers When Multiple PWM Output Function (Slave Channel) Is Used (Output Two Types of PWMs) (1/2)

(a) Timer mode register mp, mq (TMRmp, TMRmq)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmp	CKSmp1	CKSmp0		CCSmp	M/S Note 1	STSmp2	STSmp1	STSmp0	CISmp1	CISmp0			MDmp3	MDmp2	MDmp1	MDmp0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

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

Operation mode of channel p, q
100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

Selection of TImp and TImq pins input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMmn of master channel.

Setting of MASTERmp and MASTERmq bits (channels 2, 4, 6)
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.

Note 1. TMRm2, TMRm4, TMRm6: MASTERmp, MASTERmq bit
TMRm1, TMRm3: SPLITmp, SPLITmq bit
TMRm5, TMRm7: Fixed to 0

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 7 (Where p and q are integers greater than n)

Figure 6-79. Example of Set Contents of Registers When Multiple PWM Output Function (Slave Channel) Is Used (Output Two Types of PWMs) (2/2)

(b) Timer output register m (TOm)

	Bit q	Bit p	
TOm	TOmq	TOmp	0: Outputs 0 from TOmp or TOmq. 1: Outputs 1 from TOmp or TOmq.
	1/0	1/0	

(c) Timer output enable register m (TOEm)

	Bit q	Bit p	
TOEm	TOEmq	TOEmp	0: Stops the TOmp or TOmq output operation by counting operation. 1: Enables the TOmp or TOmq output operation by counting operation.
	1/0	1/0	

(d) Timer output level register m (TOLm)

	Bit q	Bit p	
TOLm	TOLmq	TOLmp	0: Positive logic output (active-high) 1: Negative logic output (active-low)
	1/0	1/0	

(e) Timer output mode register m (TOMm)

	Bit q	Bit p	
TOMm	TOMmq	TOMmp	1: Sets the slave channel output mode.
	1	1	

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 7 (Where p and q are integers greater than n)

Figure 6-80. Operation Procedure When Multiple PWM Output Function Is Used (Output Two Types of PWMs) (1/3)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. ⟶	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 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 channel. The TOMmp and TOMmq bits of timer output mode register m (TOMm) are set to 1 (slave channel output mode). Sets the TOLmp and TOLmq bits. Sets the TOmp and TOmq bits and determines default level of the TOmp and TOmq outputs. ⟶	The TOmp and TOmq pins go into Hi-Z output state. The TOmp and TOmq default setting levels are output when the port mode register is in output mode and the port register is 0.
	Sets the TOEmp and TOEmq bits to 1 and enables operation of TOmp and TOmq. ⟶	TOmp and TOmq do not change because channels stop operating.
	Clears the port register and port mode register to 0. ⟶	The TOmp and TOmq pins output the TOmp and TOmq set levels.

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 7 (Where p and q are a consecutive integer greater than n)

Figure 6-80. Operation Procedure When Multiple PWM Output Function Is Used (Output Two Types of PWMs) (2/3)

	Software Operation	Hardware Status
Operation start	(Sets the TOEmp and TOEmq (slave) bits to 1 only when resuming operation.) The TSmn bit (master), and TSmp and TSmq (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. →	TEmn = 1, TEmp, TEmq = 1
	The TSmn, TSmp, and TSmq bits automatically return to 0 because they are trigger bits.	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.	Master channel loads the value of the TDRmn register to timer count register mn (TCRmn), and the counter starts counting 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 (master), TTmp, and TTmq (slave) bits are set to 1 at the same time. →	TEmn, TEmp, TEmq = 0, and count operation stops.
	The TTmn, TTmp, and TTmq bits automatically return to 0 because they are trigger bits.	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.

Operation is resumed.

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 7 (Where p and q are a consecutive integer greater than n)

Figure 6-80. Operation Procedure When Multiple PWM Output Function Is Used (Output Two Types of PWMs) (3/3)

	Software Operation	Hardware Status
TAU stop	To hold the TOmp and TOmq pin output levels Clears the TOmp, TOmq bits to 0 after the value to be held is set to the port register. →	The TOmp and TOmq pin output levels are held by port function.
	When holding the TOmp and TOmq pin output levels are not necessary Setting not required.	
	The TAUmEN bit of the PER0 register is cleared to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOmp and TOmq bits are cleared to 0 and the TOmp and TOmq pins are set to port mode.)

Remark m: Unit number (m = 0), n: Master channel number (n = 0, 2, 4)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 7 (Where p and q are a consecutive integer greater than n)

6.9.4 Operation as two-channel input with one-shot pulse output function

By using signal input to two pins (TI0n and TI0p), a one-shot pulse having any delay pulse width can be generated.

The delay (output delay time) and one-shot pulse width can be calculated by the following expressions.

$$\text{Delay time} = \{\text{Set value of TDR0n (master)} + 2\} \times \text{count clock period}$$

$$\text{One-shot pulse active-level width} = \text{count clock period} \times ((10000\text{H} + \text{TSR0p: OVF}) + (\text{capture value of TDR0p (slave)} + 1))$$

Caution The TI0n and TI0p pin inputs are each sampled using the operation clock (f_{MCK}) selected with the CKS0n1 bit of the timer mode register 0n (TMR0n), so an error of one cycle of the operation 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 TI0n pin input used as the start trigger. Upon detection of a start trigger (valid edge of TI0n pin input), the master channel loads the value of timer data register 0n (TDR0n) to the timer count register 0n (TCR0n), and performs counting down in synchronization with the count clock (f_{TCLK}). When TCR0n = 0000H, the master channel outputs INTTM0n and outputs the active level from the TO0p 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 TI0p pin input used as the end trigger.

Upon detection of an end trigger (valid edge of TI0p pin input), the slave channel transfers (captures) the count value of the TCR0p register to the TDR0p register, and clears it to 0000H. Simultaneously, the slave channel outputs INTTM0p and the inactive level from the TO0p pin. Here, if the counter overflow has occurred, the OVF bit in the timer status register 0p (TSR0p) is set; if not, the OVF bit is cleared. After this, the same steps are repeated.

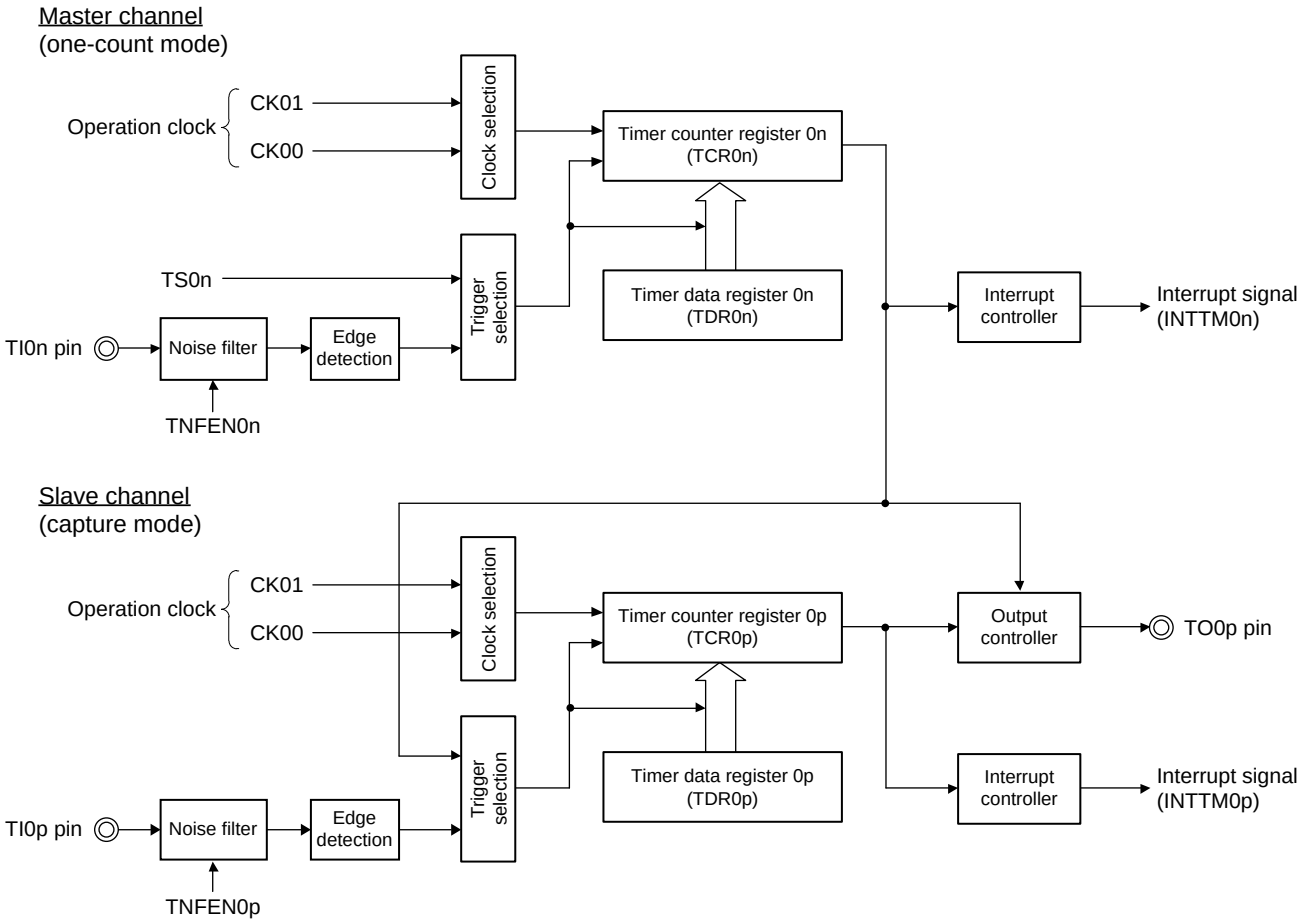
When the count value is captured to the TDR0p register, the OVF bit in the TSR0p register is updated depending on the overflow status during the active level period, which allows the overflow status of the captured value to be checked.

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 TSR0p 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 TI0n pin input, the software operation (TS0n = 1) can be used as a start trigger for the master channel startup detection.

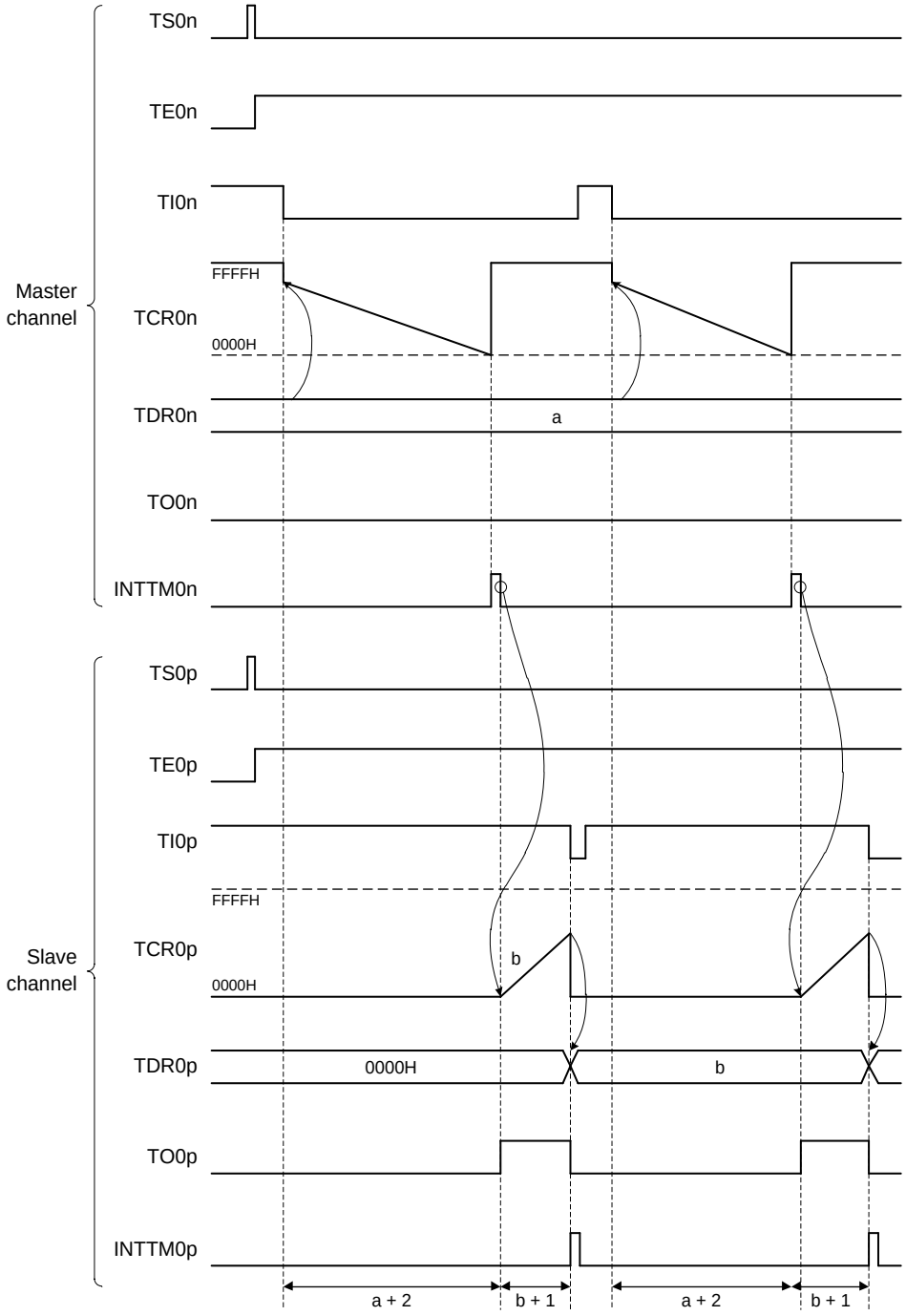
Remark n: Master channel number (n = 0, 2), p: Slave channel number (p = 3)

Figure 6-81. Block Diagram of Operation for Two-channel Input with One-shot Pulse Output Function



Remark n: Master channel number (n = 0, 2)
p: Slave channel number (p = 3)

Figure 6-82. Example of Basic Timing of Operation for Two-channel Input with One-shot Pulse Output Function

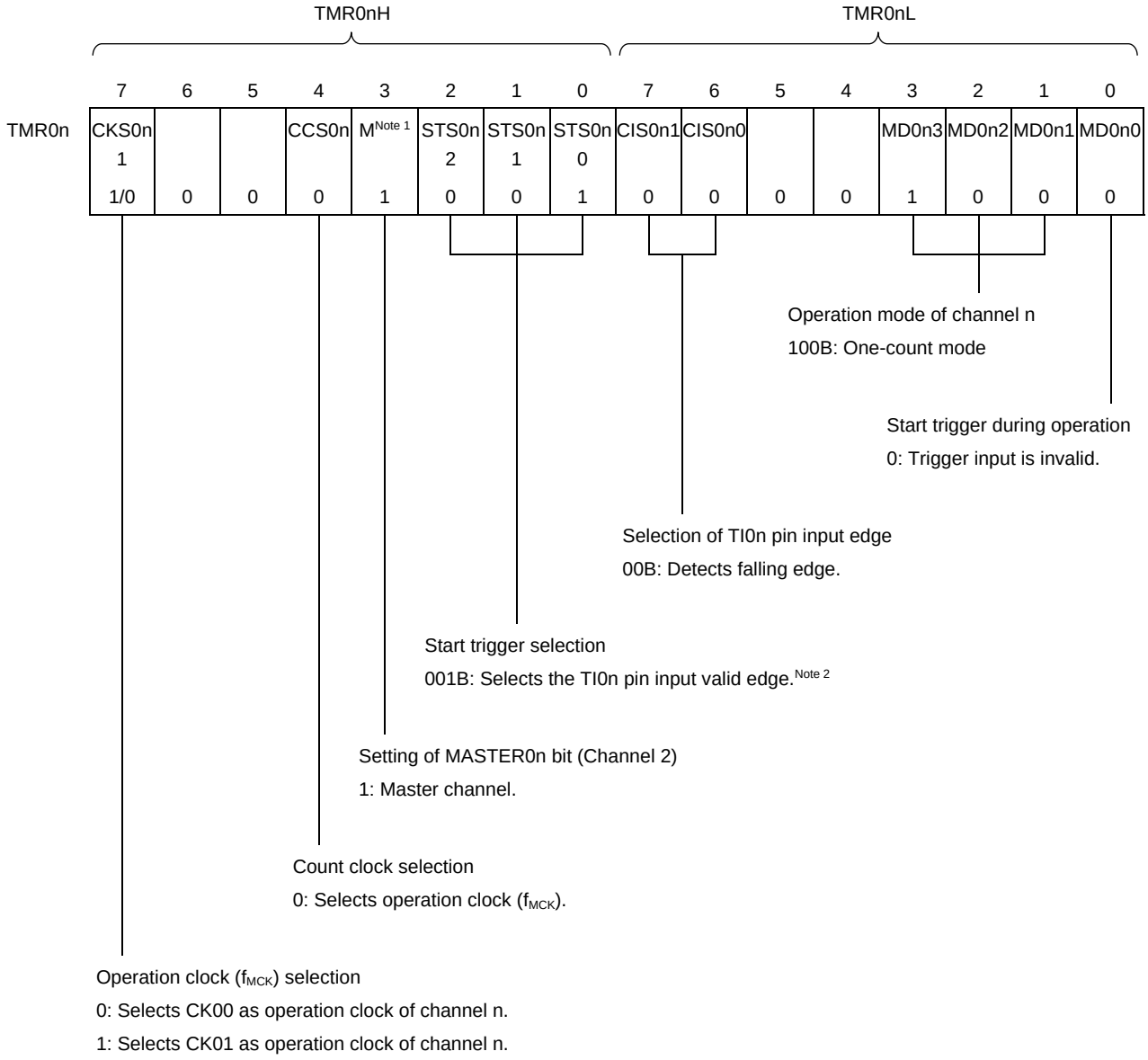


Remark 1. n: Master channel number (n = 0, 2)
p: Slave channel number (p = 3)

Remark 2. TS0n, TS0p: Bit n, p of timer channel start register 0 (TS0)
TE0n, TE0p: Bit n, p of timer channel enable status register 0 (TE0)
TI0n, TI0p: TI0n and TI0p pins input signal
TCR0n, TCR0p: Timer count registers 0n, 0p (TCR0n, TCR0p)
TDR0n, TDR0p: Timer data registers 0n, 0p (TDR0n, TDR0p)
TO0n, TO0p: TO0n and TO0p pins output signal

Figure 6-83. Example of Set Contents of Registers for Two-channel Input with One-shot Pulse Output Function (Master Channel) (1/2)

(a) Timer mode register 0n (TMR0nH, TMR0nL)



Note 1. TMR02: MASTER02 bit
TMR00: 0 fixed

Note 2. A software operation (TS0n = 1) can be used as a start trigger, instead of using the TI0n pin input.

Remark n: Master channel number (n = 0, 2)

Figure 6-83. Example of Set Contents of Registers for Two-channel Input with One-shot Pulse Output Function (Master Channel) (2/2)

(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n	0: Outputs 0 from TO0n.
	0	

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n	0: Stops the TO0n output operation by counting operation.
	0	

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n	0: Setting is invalid because master channel output mode is set (TOM0n = 0).
	0	

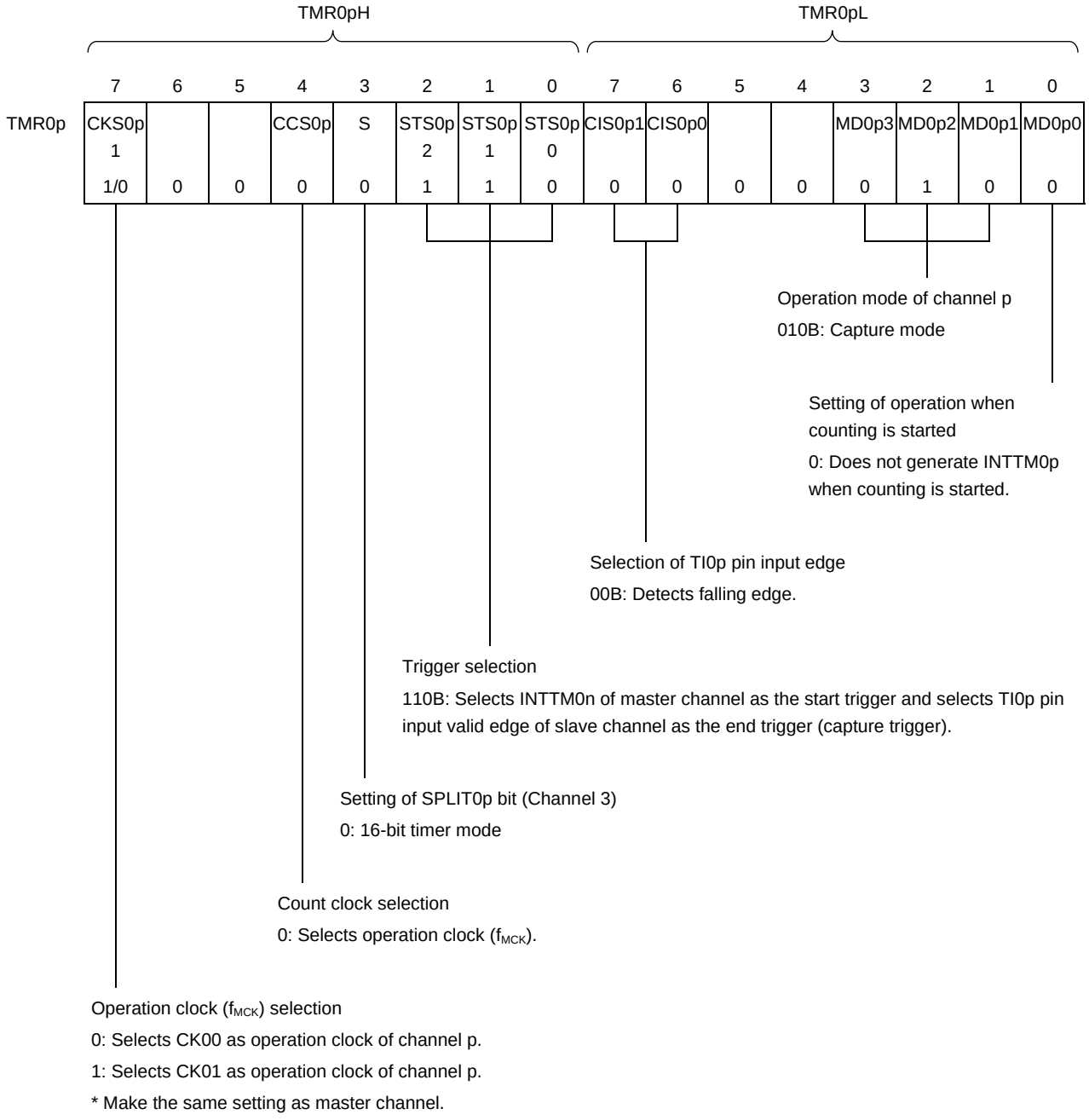
(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n	0: Sets master channel output mode.
	0	

Remark n: Master channel number (n = 0, 2)

Figure 6-84. Example of Set Contents of Registers for Two-channel Input with One-shot Pulse Output Function (Slave Channel) (1/2)

(a) Timer mode register 0p (TMR0pH, TMR0pL)



Remark n: Master channel number (n = 0, 2)
p: Slave channel number (p = 3)

Figure 6-84. Example of Set Contents of Registers for Two-channel Input with One-shot Pulse Output Function (Slave Channel) (2/2)

(b) Timer output register 0 (TO0)

	Bit p	
TO0	TO0p 1/0	0: Outputs 0 from TO0p. 1: Outputs 1 from TO0p.

(c) Timer output enable register 0 (TOE0)

	Bit p	
TOE0	TOE0p 1/0	0: Stops the TO0p output operation by counting operation (the level set in the TO0p bit is output from the TO0p pin). 1: Enables the TO0p output operation by counting operation (output from the TO0p pin is toggled).

(d) Timer output level register 0 (TOL0)

	Bit p	
TOL0	TOL0p 1/0	0: Positive logic output (active-high) 1: Negative logic output (active-low)

(e) Timer output mode register 0 (TOM0)

	Bit p	
TOM0	TOM0p 1	1: Sets the slave channel output mode.

Remark p: Slave channel number (p = 3)

Figure 6-85. Operation Procedure of 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 TAU0EN 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 0 (TPS0). Determines operation clock (CK00 and CK01) for each channel.	
Channel default setting	Sets noise filter enable register 1 (NFEN1). Sets timer mode register 0n, p (TMR0n, TMR0p) (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 0n (TDR0n) (for the access procedure to the TDR0nH and TDR0nL registers, see **6.2.2 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 TOL0 register to 0. Clears the target bit of the timer output enable register 0 (TOE0) to 0. Sets slave channel. Sets the target bit of timer output mode register 0 (TOM0) to 1 (slave channel output mode). Sets the target bit of the TOL0 register. Sets the TO0p bit and determines default level of the TO0p output. Sets the TOE0p bit to 1 and enables operation of TO0p. →	The TO0p pin goes into Hi-Z state. (The port mode register is set to input mode.) TO0p does not change because channel stops operating (The TO0p pin is not affected even if the TO0p bit is modified).
	Clears the port register and port mode register to 0 (output mode is set). →	The level set in the TO0p bit is output from the TO0p pin.

Remark n: Master channel number (n = 0, 2)
p: Slave channel number (p = 3)

Figure 6-85. Operation Procedure of Two-channel Input with One-shot Pulse Output Function (2/2)

	Software Operation	Hardware Status
Operation start	Sets the TOE0p bit of the slave channel to 1 to enable TO0p operation (only when operation is resumed). Sets the target bits of the TS0 register (master and slave) to 1 at the same time. →	The target bits of the TE0 register are set to 1 and the master channel enters the TI0n pin input valid edge detection wait status.
	The target bits of the TS0 register automatically return to 0 because they are trigger bits.	
	Count operation starts on detection of the next start triggers: → • The TI0n pin input valid edge is detected. • The TS0n bit is set to 1 by software.	Value of the TDR0n register is loaded to the timer count register 0n (TCR0n) of the master channel, and count down operation starts.
During operation	Changes master channel setting. The TCR0n register can always be read (for the access procedure to the TCR0nH and TCR0nL registers, see **6.2.1 Timer count register mn (TCRmn)**). The set values of only the CIS0n1 and CIS0n0 bits of the TMR0n register can be changed. The set values in the target bits of the TDR0n, TO0, TOE0, TOM0, and TOL0 registers cannot be changed. Changes slave channel setting. The TDR0p register can always be read. The TCR0p register can always be read. The TSR0p register can always be read. The set values of only the CIS0p1 and CIS0p0 bits of the TMR0p register can be changed. The set values in the target bits of the TO0p, TOE0p, TOM0, and TOL0 registers cannot be changed.	The master channel counter (TCR0n) performs count down operation. When the count value reaches TCR0n = 0000H, INTTM0n is generated, and the counter stops at TCR0n = FFFFH until the next start trigger is detected (the TI0n pin input valid edge is detected or TS0n bit is set to 1). The slave channel, triggered by INTTM0n of the master channel, clears the timer counter register 0p (TCR0p) to 0000H. The counter (TCR0p) starts counting up from 0000H, and when the TI0n pin input valid edge is detected, the count value is transferred to the timer data register 0p (TDR0p) (capture) and the TCR0p register is cleared to 0000H. At this time, INTTM0p is generated, which sets the TO0p 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 TE0 register are cleared to 0, and count operation stops. The TCR0n and TCR0p registers hold count value and stop. The TO0p output is not initialized but holds current status.
	The target bits of the TT0 register automatically return to 0 because they are trigger bits.	
	Clears the TOE0p bit of slave channel to 0 and sets a value to the TO0p bit. →	The level set in the TO0p bit is output from the TO0p pin.
TAU stop	To hold the TO0p pin output level Clears the TO0p bit to 0 after the value to be held (output latch) is set to the port register. →	The TO0p pin output level is held by port function.
	Clears the TAU0EN 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 n: Master channel number (n = 0, 2)
p: Slave channel number (p = 3)

6.10 Cautions When Using Timer Array Unit

6.10.1 Cautions when using timer output

Depending on the product, a timer output and other alternate functions may be assigned to some pins. In such case, the outputs of the other alternate functions must be set to their initial states.

For details, see **4.5 Register Settings When Using Alternate Function**.

CHAPTER 7 REAL-TIME CLOCK 2

7.1 Functions of Real-time Clock 2

Real-time clock 2 (RTC2) has the following functions.

- Counters of year, month, day of the week, date, hour, minute, and second, that can count up to 99 years (with leap year correction function)
- Constant-period interrupt function (period: 0.5 seconds, 1 second, 1 minute, 1 hour, 1 day, 1 month)
- Alarm interrupt function (alarm: day of the week, hour, and minute)
- Pin output function of 1 Hz[Note 1]

The real-time clock 2 interrupt signal (INTRTC) can be utilized for wakeup from STOP mode.

Note 1. This function is not available for 10-pin products.

Caution The year, month, day of the week, day, hour, minute and second can only be counted when a subsystem clock (f_{SUB} = 32.768 kHz) is selected as the operation clock of real-time clock 2. When the low-speed oscillation clock (f_{IL} = 15 kHz) is selected, only the constant-period interrupt function is available.
However, the constant-period interrupt interval when f_{IL} is selected will be calculated with the constant-period (the value selected with RTCC0 register) × f_{SUB}/f_{IL}.

7.2 Configuration of Real-time Clock 2

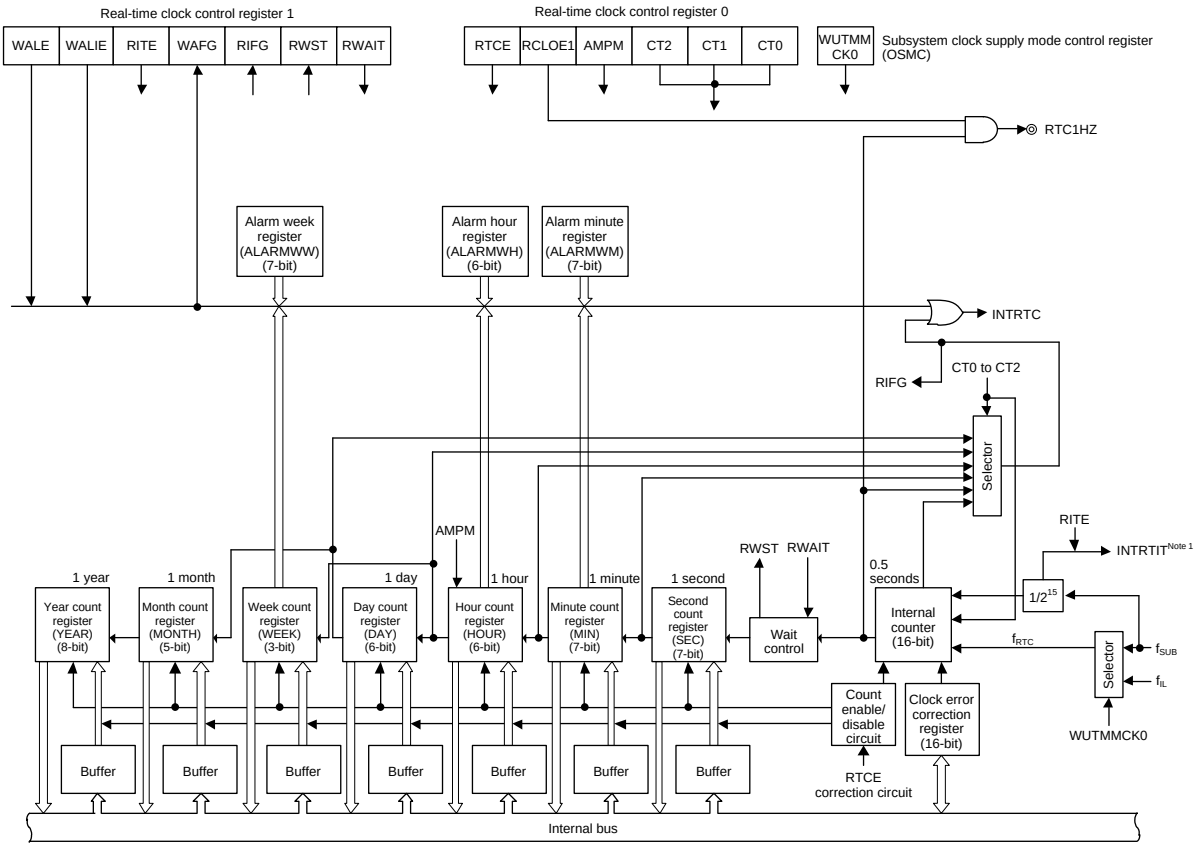
Real-time clock 2 includes the following hardware.

Table 7-1. Configuration of Real-time Clock 2

Item	Configuration
Counter	Counter (16-bit)
Control registers	Peripheral enable register 1 (PER1)
	Subsystem clock supply mode control register (OSMC)
	Real-time clock control register 0 (RTCC0)
	Internal reset status register by data retention power supply voltage (PORSR)
	Real-time clock control register 1 (RTCC1)
	Second count register (SEC)
	Minute count register (MIN)
	Hour count register (HOUR)
	Day count register (DAY)
	Week count register (WEEK)
	Month count register (MONTH)
	Year count register (YEAR)
	Clock error correction register (SUBCUD)
	Alarm minute register (ALARMWM)
	Alarm hour register (ALARMWH)
	Alarm week register (ALARMWW)

Figure 7-1 shows the real-time clock 2 diagram.

Figure 7-1. Real-time Clock 2 Diagram



Note 1. An interrupt that indicates the timing to get the correction value from the clock error correction register (SUBCUD). The fetch timing is 1 second (f_{SUB} base) interval.

7.3 Registers Controlling Real-time Clock 2

Real-time clock 2 is controlled by the following registers.

- Peripheral enable register 1 (PER1)
- Subsystem clock supply mode control register (OSMC)
- Internal reset status register by data retention power supply voltage (PORSR)
- Real-time clock control register 0 (RTCC0)
- Real-time clock control register 1 (RTCC1)
- Second count register (SEC)
- Minute count register (MIN)
- Hour count register (HOUR)
- Day count register (DAY)
- Week count register (WEEK)
- Month count register (MONTH)
- Year count register (YEAR)
- Clock error correction register (SUBCUD)
- Alarm minute register (ALARMWM)
- Alarm hour register (ALARMWH)
- Alarm week register (ALARMWW)
- Port mode registers 0/1/4 (PM0/PM1/PM4)
- Port registers 0/1/4 (P0/P1/P4)

The following shows the register states depending on reset sources.

Reset Source	System Registers[Note 1]	Calendar Registers[Note 2]
Internal reset by data retention power supply voltage	Reset	Not reset
External reset	Retained	Retained
WDT	Retained	Retained
TRAP	Retained	Retained
SPOR	Retained	Retained
Other internal reset	Retained	Retained

Note 1. RTCC0, RTCC1, SUBCUD

Note 2. SEC, MIN, HOUR, DAY, WEEK, MONTH, YEAR, ALARMWM, ALARMWH, ALARMWW, (counter)

Reset generation does not reset the SEC, MIN, HOUR, WEEK, DAY, MONTH, YEAR, ALARMWM, ALARMWH, and ALARMWW registers. Initialize all the registers after power on.

7.3.1 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 real-time clock 2 registers are manipulated, be sure to set bit 7 (RTCWEN) 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 7-2. Format of Peripheral Enable Register 1 (PER1)

Address: F007AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PER1	RTCWEN	0	0	0	0	0	CTSUEN	0

RTCWEN	Control of internal clock supply to real-time clock 2
0	Stops input clock supply. • SFR used by real-time clock 2 cannot be written. • Real-time clock 2 can operate.
1	Enables input clock supply. • SFR used by real-time clock 2 can be read/written. • Real-time clock 2 can operate.

Caution 1. The clock error correction register (SUBCUD) becomes read/write enabled when RTCWEN in the peripheral enable register 1 (PER1) is set to 1.

Caution 2. When using real-time clock 2, first set the RTCWEN bit to 1 and then set the following registers, while oscillation of the count clock (f_{RTC}) is stable. If RTCWEN = 0, writing to the control registers of real-time clock 2 is ignored, and read values are the values set when RTCWEN = 1 (except for the subsystem clock supply mode control register (OSMC), internal reset status register by data retention power supply voltage (PORSR), port mode registers 0, 1, 4 (PM0, PM1, PM4), and port registers 0, 1, 4 (P0, P1, P4).

- **Real-time clock control register 0 (RTCC0)**
- **Real-time clock control register 1 (RTCC1)**
- **Second count register (SEC)**
- **Minute count register (MIN)**
- **Hour count register (HOUR)**
- **Day count register (DAY)**
- **Week count register (WEEK)**
- **Month count register (MONTH)**
- **Year count register (YEAR)**
- **Clock error correction register (SUBCUD)**
- **Alarm minute register (ALARMWM)**
- **Alarm hour register (ALARMWH)**
- **Alarm week register (ALARMWW)**

Caution 3. Be sure to set bits 0, and 2 to 6 to 0.

7.3.2 Subsystem clock supply mode control register (OSMC)

This register is used to reduce power consumption by stopping unnecessary clock functions.

If the RTCLPC bit is set to 1, power consumption can be reduced, because clock supply to the peripheral functions other than real-time clock 2 and 12-bit interval timer is stopped in STOP mode or in HALT mode while the subsystem clock is selected as the CPU clock.

In addition, the OSMC register is used to select the operation clock of real-time clock 2 and 12-bit interval timer.

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

Reset signal generation clears this register to 00H.

Figure 7-3. Format of Subsystem Clock Supply Mode Control Register (OSMC)

Address: F00F3H After reset: 00H R/W

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

RTCLPC	In STOP mode and in HALT mode while the CPU operates using the subsystem clock
0	Enables subsystem clock supply to peripheral functions. (For peripheral functions for which operation is enabled, see **CHAPTER 17 STANDBY FUNCTION**).
1	Stops subsystem clock supply to peripheral functions other than real-time clock 2 and 12-bit interval timer.

WUTMMCK0	Selection of operation clock of real-time clock 2 and 12-bit interval timer
0	Subsystem clock (f_{SUB})
1	Low-speed on-chip oscillator clock (f_{IL})

Caution 1. If the subsystem clock is oscillating, be sure to select the subsystem clock (WUTMMCK0 bit = 0).

Caution 2. When WUTMMCK0 is set to 1, the low-speed on-chip oscillator clock oscillates.

Caution 3. When WUTMMCK0 is set to 1, only the constant-period interrupt function of real-time clock 2 can be used. The year, month, day of the week, day, hour, minute, and second counters and the 1-Hz output function of real-time clock 2 cannot be used. The interval of the constant-period interrupt is calculated by constant period (value selected by using the RTCC0 register) × f_{SUB}/f_{IL}.

Caution 4. The subsystem clock and low-speed on-chip oscillator clock can only be switched by using the WUTMMCK0 bit if real-time clock 2 and 12-bit interval timer are all stopped.

7.3.3 Internal reset status register by data retention power supply voltage (PORSR)

The PORSR register checks the occurrence of an internal reset by the data retention power supply voltage.

Writing 1 to bit 0 (PORF) of the PORSR register is valid; writing 0 is invalid.

To check the occurrence of an internal reset by the data retention power supply voltage, write 1 to the PORF bit in advance.

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

Internal reset generated by the data retention power supply voltage clears this register to 00H.

Caution 1. The PORSR register is initialized only by an internal reset with the data retention power supply voltage and retains its value on other reset sources.

Caution 2. When PORF is 1, it is guaranteed that an internal reset by the data retention power supply voltage has not occurred, but it is not guaranteed that the RAM value is retained.

Figure 7-4. Format of Internal Reset Status Register by Data Retention Power Supply Voltage (PORSR)

Address: F00F9H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PORSR	0	0	0	0	0	0	0	PORF

PORF	Check the occurrence of an internal reset by the data retention power supply voltage
0	Indicates that 1 has not been written or that an internal reset by the data retention power supply voltage has occurred.
1	Indicates that an internal reset by the data retention power supply voltage has not occurred.

7.3.4 Real-time clock control register 0 (RTCC0)

The RTCC0 register is an 8-bit register that is used to start or stop the real-time clock 2 operation, control the RTC1HZ pin, set the 12- or 24-hour system, and set the constant-period interrupt function.

RTCC0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Internal reset generated by the data retention power supply voltage of the internal reset circuit clears this register to 00H.

Figure 7-5. Format of Real-time Clock Control Register 0 (RTCC0) (1/2)

Address: FFF9DH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RTCC0	RTCE	0	RCLOE1	0	AMPM	CT2	CT1	CT0

RTCE[Note 1]	Real-time clock 2 operation control
0	Stops counter operation.
1	Starts counter operation.

RCLOE1[Note 2]	RTC1HZ pin output control
0	Disables output of the RTC1HZ pin (1 Hz)
1	Enables output of the RTC1HZ pin (1 Hz)
Output of 1 Hz is not output because the clock counter does not operate when RTCE = 0.	

Note 1. When shifting to STOP mode immediately after setting RTCE to 1, use the procedure shown in **Figure 7-19 Procedure for Shifting to HALT/STOP Mode after Setting RTCE = 1**.

Note 2. When the RCLOE1 bit is set while the clock counter operates (RTCE = 1), a glitch may be output to the 1-Hz output pin (RTC1HZ).

Caution Be sure to clear bits 4 and 6 to 0.

Figure 7-5. Format of Real-time Clock Control Register 0 (RTCC0) (2/2)

Address: FFF9DH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RTCC0	RTCE	0	RCLOE1	0	AMPM	CT2	CT1	CT0

Table 7-2. Relationship Between RTCE and RCLOE1 Settings and Status

Register Settings		Status	
RTCE	RCLOE1	Real-time clock 2	RTC1HZ pin output
0	×	Counting stopped	No output
1	0	Count operation	No output
	1	Count operation	1-Hz output

AMPM	12-/24-hour system select
0	12-hour system (a.m. and p.m. are displayed.)
1	24-hour system
When changing the value of the AMPM bit while the clock counter operates (RTCE = 1), set RWAIT (bit 0 of RTCC1) and then set the hour counter (HOUR) again. When the AMPM value is 0, the 12-hour system is displayed. When the value is 1, the 24-hour system is displayed. **Table 7-3** shows the displayed time digits.	

CT2	CT1	CT0	Constant-period interrupt (INTRTC) selection
0	0	0	Does not use constant-period interrupt function.
0	0	1	Once per 0.5 s (synchronized with second count up)
0	1	0	Once per 1 s (same time as second count up)
0	1	1	Once per 1 m (second 00 of every minute)
1	0	0	Once per 1 hour (minute 00 and second 00 of every hour)
1	0	1	Once per 1 day (hour 00, minute 00, and second 00 of every day)
1	1	×	Once per 1 month (Day 1, hour 00 a.m., minute 00, and second 00 of every month)
When changing the values of the CT2 to CT0 bits while the counter operates (RTCE = 1), rewrite the values of the CT2 to CT0 bits after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, after rewriting the values of the CT2 to CT0 bits, enable interrupt servicing after clearing the RIFG and RTCIF flags.			

Caution Be sure to clear bits 4 and 6 to 0.

Remark ×: Don't care

7.3.5 Real-time clock control register 1 (RTCC1)

The RTCC1 register is an 8-bit register that is used to control the alarm interrupt function and the wait time of the counter.

RTCC1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Internal reset generated by the data retention power supply voltage of the internal reset circuit clears this register to 00H.

Figure 7-6. Format of Real-time Clock Control Register 1 (RTCC1) (1/3)

Address: FFF9EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RTCC1	WALE	WALIE	RITE	WAFG	RIFG	0	RWST	RWAIT

WALE	Alarm operation control
0	Match operation is invalid.
1	Match operation is valid.[Note 1]
When setting a value to the WALE bit while the counter operates (RTCE = 1) and WALIE = 1, rewrite the WALE bit after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG and RTCIF flags after rewriting the WALE bit. When setting each alarm register (WALIE flag of real-time clock control register 1 (RTCC1), the alarm minute register (ALARMWM), the alarm hour register (ALARMWH), and the alarm week register (ALARMWW)), set match operation to be invalid (“0”) for the WALE bit.	

WALIE	Control of alarm interrupt (INTRTC) function operation
0	Does not generate interrupt on matching of alarm.
1	Generates interrupt on matching of alarm. [Note 1]

Note 1. When the detection of matching for an alarm or the alarm interrupt is to be used, set the fixed-cycle interrupt to “once per second”, and, within 1 second of the generation of the INTRTC interrupt, set the RWAIT bit to 1 and read or write counter values. If the RWAIT bit is set to 1 and counter values are read or written with any given timing, matching for an alarm may not occur and the interrupt request may also not be generated. For details on the procedures for reading and writing counter values, see **7.4.3 Reading real-time clock 2 counter**, and **7.4.4 Writing to real-time clock 2 counter**.

Caution If writing is performed to RTCC1 with a 1-bit manipulation instruction, the RIFG and WAFG flags may be cleared. Therefore, to perform writing to RTCC1, be sure to use an 8-bit manipulation instruction. To prevent the RIFG and WAFG flags from being cleared during writing, set the corresponding bit to 1 (to disable writing). If the RIFG and WAFG flags are not used and the value may be changed, RTCC1 may be written by using a 1-bit manipulation instruction.

Figure 7-6. Format of Real-time Clock Control Register 1 (RTCC1) (2/3)

Address: FFF9EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RTCC1	WALE	WALIE	RITE	WAFG	RIFG	0	RWST	RWAIT

RITE	Control of correction timing signal interrupt (INTRTIT) function operation
0	Disables the correction timing signal interrupt.
1	Enables the correction timing signal interrupt.

WAFG	Alarm detection status flag
0	Alarm mismatch
1	Detection of matching of alarm
This is a status flag that indicates detection of matching with the alarm. It is valid only when WALE = 1 and is set to “1” one clock (32.768 kHz) after matching of the alarm is detected. This flag is cleared when 0 is written to it. Writing 1 to it is invalid.	

RIFG	Constant-period interrupt status flag
0	Constant-period interrupt is not generated.
1	Constant-period interrupt is generated.
This flag indicates the status of generation of the constant-period interrupt. When the constant-period interrupt is generated, it is set to 1. This flag is cleared when 0 is written to it. Writing 1 to it is invalid.	

Caution If writing is performed to RTCC1 with a 1-bit manipulation instruction, the RIFG and WAFG flags may be cleared. Therefore, to perform writing to RTCC1, be sure to use an 8-bit manipulation instruction. To prevent the RIFG and WAFG flags from being cleared during writing, set the corresponding bit to 1 (to disable writing). If the RIFG and WAFG flags are not used and the value may be changed, RTCC1 may be written by using a 1-bit manipulation instruction.

Figure 7-6. Format of Real-time Clock Control Register 1 (RTCC1) (3/3)

Address: FFF9EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RTCC1	WALE	WALIE	RITE	WAFG	RIFG	0	RWST	RWAIT

RWST	Wait status flag of real-time clock 2
0	Counter is operating.
1	Mode to read or write counter value.
This status flag indicates whether the setting of the RWAIT bit is valid. Before reading or writing the counter value, confirm that the value of this flag is 1. Even if the RWAIT bit is set to 0, the RWST bit is not set to 0 while writing to the counter. After writing is completed, the RWST bit is set to 0.	

RWAIT	Wait control of real-time clock 2[Note 1]
0	Sets counter operation.
1	Stops SEC to YEAR counters. Mode to read or write counter value.
This bit controls the operation of the counter. Be sure to write 1 to it to read or write the counter value. As the counter (16-bit) is continuing to run, complete reading or writing within one second and turn back to 0. When RWAIT = 1, it takes up to one cycle of f_{RTC} until the counter value can be read or written (RWST = 1).[Note 2, Note 3] When the internal counter (16-bit) overflowed while RWAIT = 1, it keeps the event of overflow until RWAIT = 0, then counts up. However, when it wrote a value to second count register, it will not keep the overflow event.	

Note 1. When the detection of matching for an alarm or the alarm interrupt is to be used, set the fixed-cycle interrupt to “once per second”, and, within 1 second of the generation of the INTRTC interrupt, set the RWAIT bit to 1 and read or write counter values. If the RWAIT bit is set to 1 and counter values are read or written with any given timing, matching for an alarm may not occur and the interrupt request may also not be generated. For details on the procedures for reading and writing counter values, see **7.4.3 Reading real-time clock 2 counter**, and **7.4.4 Writing to real-time clock 2 counter**.

Note 2. When setting RWAIT=1 during 1 operating clock (f_{RTC}), after setting RTCE = 1, it may take two clock time of the operation clock (f_{RTC}), until RWST bit is set to 1.

Note 3. When setting RWAIT = 1 during 1 operating clock (f_{RTC}), after returning from a stand-by (HALT mode and STOP mode), it may take two clock time of the operation clock (f_{RTC}), until RWST bit is set to 1.

Caution If writing is performed to RTCC1 with a 1-bit manipulation instruction, the RIFG and WAFG flags may be cleared. Therefore, to perform writing to RTCC1, be sure to use an 8-bit manipulation instruction. To prevent the RIFG and WAFG flags from being cleared during writing, set the corresponding bit to 1 (to disable writing). If the RIFG and WAFG flags are not used and the value may be changed, RTCC1 may be written by using a 1-bit manipulation instruction.

Remark 1. Constant-period interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the constant-period interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

Remark 2. The internal counter (16 bits) is cleared when the second count register (SEC) is written.

7.3.6 Second count register (SEC)

The SEC register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of seconds.

It is a decimal counter that counts up when the counter (16-bit) overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks of f_{RTC} later. Set a decimal value of 00 to 59 to this register in BCD code.

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

Reset signal generation does not clear this register to its default value.

Figure 7-7. Format of Second Count Register (SEC)

Address: FFF92H After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1

Remark The internal counter (16 bits) is cleared when the second count register (SEC) is written.

Caution When reading or writing to SEC while the clock counter operates (RTCE = 1), be sure to use the flows shown in 7.4.3 Reading real-time clock 2 counter and 7.4.4 Writing to real-time clock 2 counter.

7.3.7 Minute count register (MIN)

The MIN register is an 8-bit register that takes a value of 0 to 59 (decimal) and indicates the count value of minutes.

It is a decimal counter that counts up when the second counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks of f_{RTC} later. Even if the second count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 00 to 59 to this register in BCD code.

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

Reset signal generation does not clear this register to its default value.

Figure 7-8. Format of Minute Count Register (MIN)

Address: FFF93H After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1

Caution When reading or writing to MIN while the clock counter operates (RTCE = 1), be sure to use the flows shown in 7.4.3 Reading real-time clock 2 counter and 7.4.4 Writing to real-time clock 2 counter.

7.3.8 Hour count register (HOUR)

The HOUR register is an 8-bit register that takes a value of 00 to 23 or 01 to 12 and 21 to 32 (decimal) and indicates the count value of hours.

It is a decimal counter that counts up when the minute counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks of f_{RTC} later. Even if the minute count register overflows while this register is being written, this register ignores the overflow and is set to the value written.

Specify a decimal value of 00 to 23, 01 to 12, or 21 to 32 by using BCD code according to the time system specified using bit 3 (AMPM) of real-time clock control register 0 (RTCC0).

If the AMPM bit value is changed, the values of the HOUR register change according to the specified time system.

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

Reset signal generation does not clear this register to its default value.

Figure 7-9. Format of Hour Count Register (HOUR)

Address: FFF94H After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
HOUR	0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1

Caution 1. Bit 5 (HOUR20) of the HOUR register indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected)

Caution 2. When reading or writing to HOUR while the clock counter operates (RTCE = 1), be sure to use the flows shown in 7.4.3 Reading real-time clock 2 counter and 7.4.4 Writing to real-time clock 2 counter.

Table 7-3 shows the relationship between the setting value of the AMPM bit, the hour count register (HOUR) value, and time.

Table 7-3. Displayed Time Digits

24-Hour Display (AMPM = 1)		12-Hour Display (AMPM = 1)	
Time	HOUR Register	Time	HOUR Register
0	00H	12 a.m.	12H
1	01H	1 a.m.	01H
2	02H	2 a.m.	02H
3	03H	3 a.m.	03H
4	04H	4 a.m.	04H
5	05H	5 a.m.	05H
6	06H	6 a.m.	06H
7	07H	7 a.m.	07H
8	08H	8 a.m.	08H
9	09H	9 a.m.	09H
10	10H	10 a.m.	10H
11	11H	11 a.m.	11H
12	12H	12 p.m.	32H
13	13H	1 p.m.	21H
14	14H	2 p.m.	22H
15	15H	3 p.m.	23H
16	16H	4 p.m.	24H
17	17H	5 p.m.	25H
18	18H	6 p.m.	26H
19	19H	7 p.m.	27H
20	20H	8 p.m.	28H
21	21H	9 p.m.	29H
22	22H	10 p.m.	30H
23	23H	11 p.m.	31H

The HOUR register value is set to 12-hour display when the AMPM bit is 0 and to 24-hour display when the AMPM bit is 1.

In 12-hour display, the fifth bit of the HOUR register displays 0 for AM and 1 for PM.

7.3.9 Date count register (DAY)

The DAY register is an 8-bit register that takes a value of 1 to 31 (decimal) and indicates the count value of days.

It is a decimal counter that count ups when the hour counter overflows.

This counter counts as follows.

[DAY count values]

- 01 to 31 (January, March, May, July, August, October, December)
- 01 to 30 (April, June, September, November)
- 01 to 29 (February, leap year)
- 01 to 28 (February, normal year)

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks of f_{RTC} later. Even if the hour count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 01 to 31 to this register in BCD code.

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

Reset signal generation does not clear this register to its default value.

Figure 7-10. Format of Day-of-week Count Register (DAY)

Address: FFF96H After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1

Caution When reading or writing to DAY while the clock counter operates (RTCE = 1), be sure to use the flows shown in 7.4.3 Reading real-time clock 2 counter and 7.4.4 Writing to real-time clock 2 counter.

7.3.10 Day-of-week count register (WEEK)

The WEEK register is an 8-bit register that takes a value of 0 to 6 (decimal) and indicates the count value of weekdays.

It is a decimal counter that counts up when a carry to the date counter occurs.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks of f_{RTC} later.

Set a decimal value of 00 to 06 to this register in BCD code.

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

Reset signal generation does not clear this register to its default value.

Figure 7-11. Format of Date Count Register (WEEK)

Address: FFF95H After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
WEEK	0	0	0	0	0	WEEK4	WEEK2	WEEK1

Caution 1. The value corresponding to the month count register (MONTH) or the day count register (DAY) is not stored in the week count register (WEEK) automatically.
After reset release, set the week count register as follow.

Day	WEEK
Sunday	00H
Monday	01H
Tuesday	02H
Wednesday	03H
Thursday	04H
Friday	05H
Saturday	06H

Caution 2. When reading or writing to WEEK while the clock counter operates (RTCE = 1), be sure to use the flows shown in 7.4.3 Reading real-time clock 2 counter and 7.4.4 Writing to real-time clock 2 counter.

7.3.11 Month count register (MONTH)

The MONTH register is an 8-bit register that takes a value of 1 to 12 (decimal) and indicates the count value of months.

It is a decimal counter that count ups when the date counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks of f_{RTC} later. Even if the day count register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 01 to 12 to this register in BCD code.

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

Reset signal generation does not clear this register to its default value.

Figure 7-12. Format of Month Count Register (MONTH)

Address: FFF97H After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

Caution When reading or writing to MONTH while the clock counter operates (RTCE = 1), be sure to use the flows shown in 7.4.3 Reading real-time clock 2 counter and 7.4.4 Writing to real-time clock 2 counter.

7.3.12 Year count register (YEAR)

The YEAR register is an 8-bit register that takes a value of 0 to 99 (decimal) and indicates the count value of years.

It is a decimal counter that counts up when the month count register (MONTH) overflows.

Values 00, 04, 08, …, 92, and 96 indicate a leap year.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks of f_{RTC} later. Even if the MONTH register overflows while this register is being written, this register ignores the overflow and is set to the value written. Set a decimal value of 00 to 99 to this register in BCD code.

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

Reset signal generation does not clear this register to its default value.

Figure 7-13. Format of Year Count Register (YEAR)

Address: FFF98H After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

Caution When reading or writing to YEAR while the clock counter operates (RTCE = 1), be sure to use the flows shown in 7.4.3 Reading real-time clock 2 counter and 7.4.4 Writing to real-time clock 2 counter.

7.3.13 Clock error correction register (SUBCUD)

This register is used to correct the clock with a minimum resolution and accuracy of 0.96 ppm when it is slow or fast by changing the counter value every second.

F8 to F0 of SUBCUD are 9-bit fixed-point (two's complement) register. For details, see **Table 7-5 Clock Error Correction Values**.

The SUBCUD register can be set by a 16-bit memory manipulation instruction.

Internal reset generated by the data retention power supply voltage of the internal reset circuit clears this register to 0020H.

Figure 7-14. Format of Clock Error Correction Register (SUBCUD)

Address: F0310H After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SUBCUD	F15	0	0	0	0	0	0	F8	F7	F6	F5	F4	F3	F2	F1	F0

F15	Clock error correction enable
0	Stops clock error correction.
1	Enables clock error correction.

The range of value that can be corrected by using the clock error correction register (SUBCUD) is shown in **Table 7-4**.

Table 7-4. Correctable Range of Crystal Resonator Oscillation Frequency Deviation

Item	Value
Correctable range	−274.6 ppm to +212.6 ppm
Maximum quantization error	±0.48 ppm
Minimum resolution	0.96 ppm

Table 7-5. Clock Error Correction Values

SUBCUD										Target Correction Values
F15	F8	F7	F6	F5	F4	F3	F2	F1	F0	
1	1	0	0	0	0	0	0	0	0	−274.6 ppm
	1	0	0	0	0	0	0	0	1	−273.7 ppm
	1	0	0	0	0	0	0	1	0	−272.7 ppm
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	1	1	1	1	1	1	1	0	1	−33.3 ppm
	1	1	1	1	1	1	1	1	0	−32.4 ppm
	1	1	1	1	1	1	1	1	1	−31.4 ppm
	0	0	0	0	0	0	0	0	0	−30.5 ppm
	0	0	0	0	0	0	0	0	1	−29.6 ppm
	0	0	0	0	0	0	0	1	0	−28.6 ppm
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	0	0	0	0	1	1	1	1	1	−0.95 ppm
	0	0	0	1	0	0	0	0	0	0 ppm
	0	0	0	1	0	0	0	0	1	0.95 ppm
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	0	1	1	1	1	1	1	0	1	210.7 ppm
	0	1	1	1	1	1	1	1	0	211.7 ppm
	0	1	1	1	1	1	1	1	1	212.6 ppm
0	×	×	×	×	×	×	×	×	×	Clock error correction stopped

The F8 to F0 value of the SUBCUD register is calculated from the target correction value by using the following expression.

$$\mathrm{SUBCUD}[8:0] = \left(\frac{\text{Target correction value [ppm]} \times 2^{15}}{10^6}\right)_{\text{2's complement (9−bit fixed−point format)}} + 0001.00000\mathrm{B}$$

Caution The target correction value is the oscillation frequency deviation (unit: [ppm]) of the crystal resonator. For calculating the correction value, see 7.4.8 Example of clock error correction of real-time clock 2.

Examples 1) When target correction value = 18.3 [ppm]

$$\mathrm{SUBCUD}[8:0] = \left(\frac{18.3 \times 2^{15}}{10^6}\right)_{\text{2's complement (9−bit fixed−point format)}} + 0001.00000\mathrm{B}$$

$$= (0.59965)_{\text{2's complement (9−bit fixed−point format)}} + 0001.00000\mathrm{B}$$

$$= 0000.10011\mathrm{B} + 0001.00000\mathrm{B}$$

$$= 0001.10011\mathrm{B}$$

Examples 2) When target correction value = 18.3 [ppm]

$$\mathrm{SUBCUD}[8:0] = \left(\frac{-18.3 \times 2^{15}}{10^{6}}\right)_{\text{2's complement (9-bit fixed-point format)}} + 0001.00000\mathrm{B}$$

$$= (-0.59965)_{\text{2's complement (9-bit fixed-point format)}} + 0001.00000\mathrm{B}$$

$$= 1111.01101\mathrm{B} + 0001.00000\mathrm{B}$$

$$= 0000.01101\mathrm{B}$$

7.3.14 Alarm minute register (ALARMWM)

This register is used to set the minute of an alarm.

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

Reset signal generation does not clear this register to its default value.

Figure 7-15. Format of Alarm Minute Register (ALARMWM)

Address: FFF9AH After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

Caution Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

7.3.15 Alarm hour register (ALARMWH)

This register is used to set the hour of an alarm.

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

Reset signal generation does not clear this register to its default value.

Figure 7-16. Format of Alarm Hour Register (ALARMWH)

Address: FFF9BH After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWH	0	0	WH20	WH10	WH8	WH4	WH2	WH1

Caution 1. Set a decimal value of 00 to 23 or 01 to 12 and 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Caution 2. Bit 5 (WH20) of the ALARMWH register indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

7.3.16 Alarm day-of-week register (ALARMWW)

This register is used to set the day of the week of an alarm.

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

Reset signal generation does not clear this register to its default value.

Figure 7-17. Format of Alarm Day-of-Week Register (ALARMWW)

Address: FFF9CH After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

Table 7-6 shows an example of setting the alarm.

Table 7-6. Setting Alarm

Time of Alarm	Day of the Week							12-Hour Display				24-Hour Display			
	Sun. WW0	Mon. WW1	Tue. WW2	Wed. WW3	Thu. WW4	Fri. WW5	Sat. WW6	Hour 10	Hour 1	Min. 10	Min. 1	Hour 10	Hour 1	Min. 10	Min. 1
Every day, 0:00 a.m.	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
Every day, 1:30 a.m.	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
Every day, 11:59 a.m.	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
Monday through Friday, 0:00 p.m.	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
Sunday, 1:30 p.m.	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
Monday, Wednesday, Friday, 11:59 p.m.	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

7.3.17 Registers controlling port functions of real-time clock 2 output pins

When using real-time clock 2, set the registers that control the port functions multiplexed on the target channel (port mode register (PMxx), and port register (Pxx)). For details, see **4.3.1 Port mode registers 0, 1, 2, 4, 6, 12 (PM0, PM1, PM2, PM4, PM6, PM12)** and **4.3.2 Port registers 0, 1, 2, 4, 6, 12, 13 (P0, P1, P2, P4, P6, P12, P13)**.

When using the ports (such as P00, P11, P13, and P41) to be shared with the real-time clock 2 output pins for real-time clock 2, set port mode register (PMxx) bit and port register (Pxx) bit corresponding to each port to 0.

Example) When using P00/RTC1HZ for real-time clock 2 output

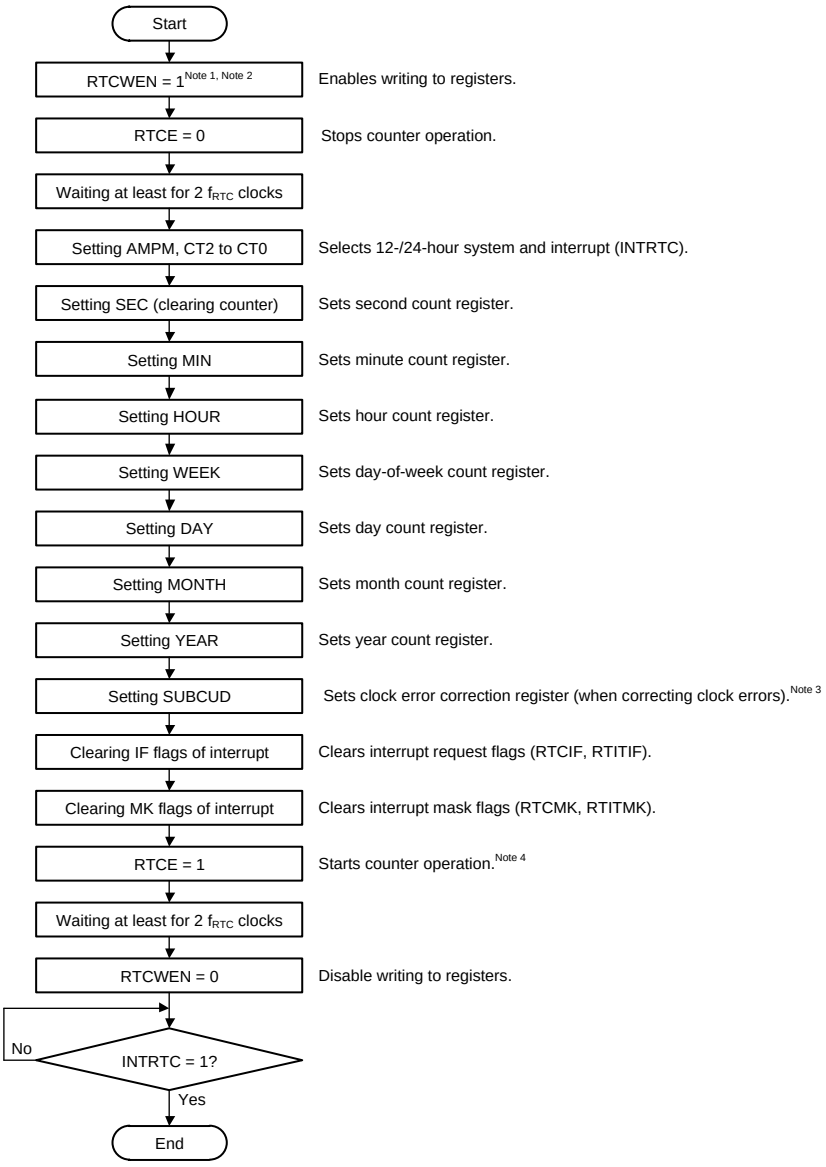
Set the PM00 bit of port mode register (PM0) to 0.

Set the P00 bit of port register (P0) to 0.

7.4 Real-time Clock 2 Operation

7.4.1 Starting operation of real-time clock 2

Figure 7-18. Procedure for Starting Operation of Real-time Clock 2



Note 1. Set RTCWEN to 0, except when accessing the RTC register, in order to prevent error when writing to the clock counter.

Note 2. While oscillation of the count clock (f_{RTC}) is stable, first set the RTCWEN bit to 1.

Note 3. Set up the SUBCUD register only if the clock error must be corrected. For details about how to calculate the correction value, see **7.4.8 Example of clock error correction of real-time clock 2**.

Note 4. Confirm the procedure described in **7.4.2 Shifting to HALT/STOP mode after starting operation** when shifting to HALT/STOP mode without waiting for INTRTC = 1 after RTCE = 1.

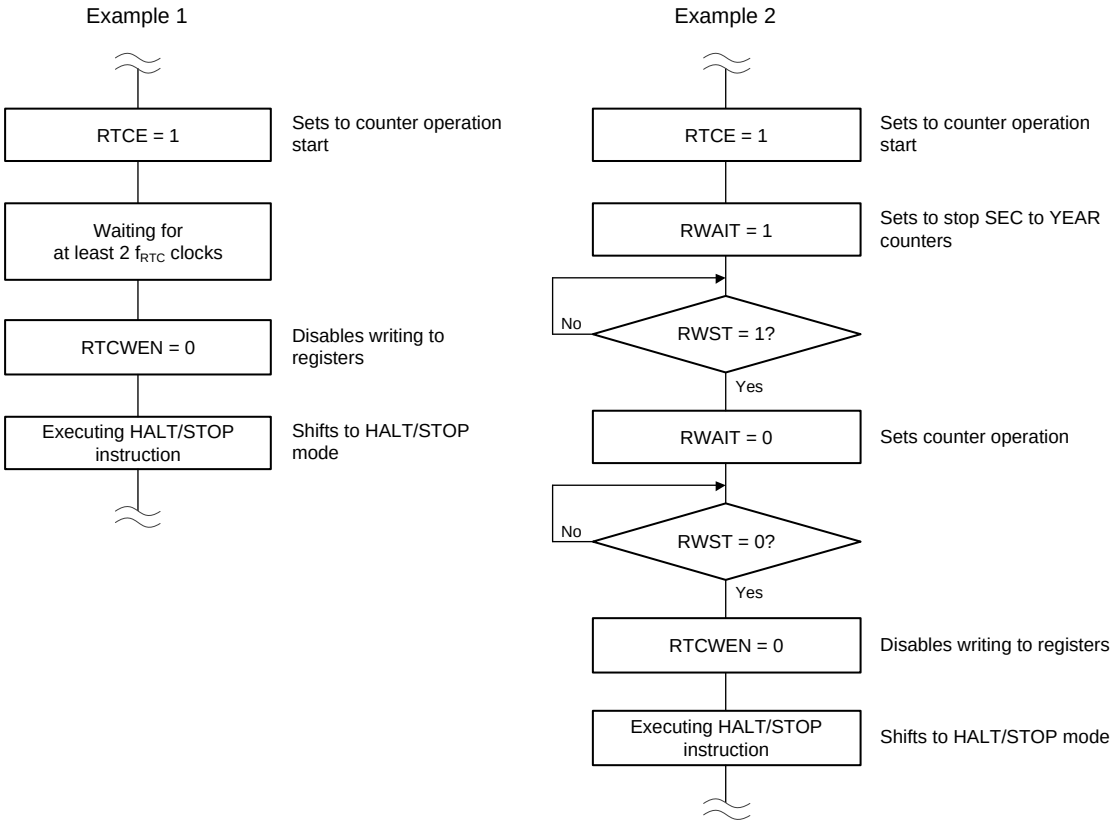
7.4.2 Shifting to HALT/STOP mode after starting operation

Perform one of the following processing when shifting to STOP mode immediately after setting the RTCE bit to 1.

However, after setting the RTCE bit to 1, this processing is not required when shifting to STOP mode after the first INTRTC interrupt has occurred.

- Shifting to HALT/STOP mode when at least two input clocks of the count clock (f_{RTC}) have elapsed after setting the RTCE bit to 1 (see Example 1 of **Figure 7-19**).
- Checking by polling the RWST bit to become 1, after setting the RTCE bit to 1 and then setting the RWAIT bit to 1. Afterward, setting the RWAIT bit to 0 and shifting to HALT/STOP mode after checking again by polling that the RWST bit has become 0 (see Example 2 of **Figure 7-19**).

Figure 7-19. Procedure for Shifting to HALT/STOP Mode after Setting RTCE = 1



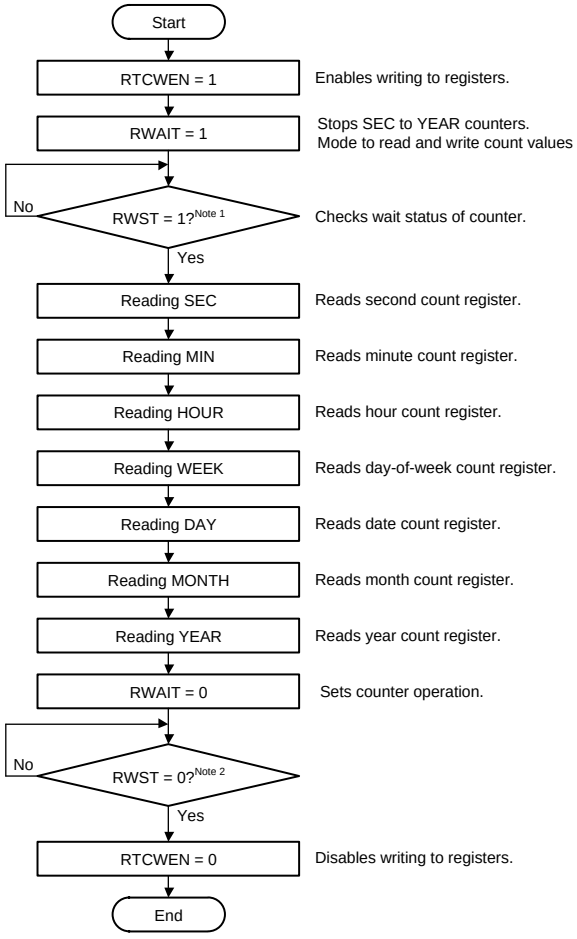
7.4.3 Reading real-time clock 2 counter

Read the counter after setting RWAIT to 1.

Set RWAIT to 0 after completion of reading the counter.

When using the alarm interrupt, read the counter according to the procedure shown in **Figure 7-21**.

Figure 7-20. Procedure for Reading Real-time Clock 2



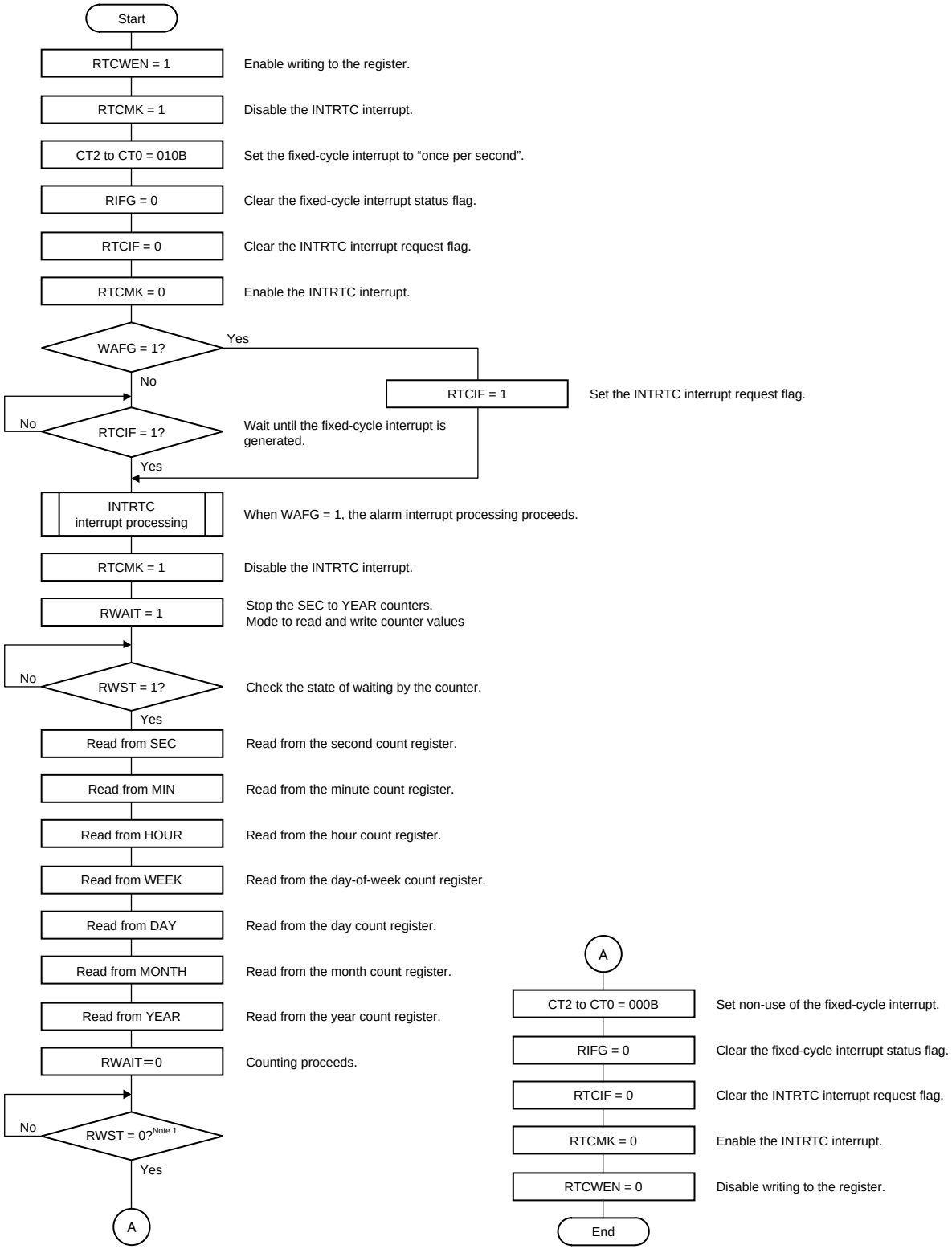
Note 1. When the counter is stopped (RTCE = 0), RWST is not set to 1.

Note 2. Be sure to confirm that RWST = 0 before shifting to STOP mode.

Caution Complete setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.

Remark SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR may be read in any sequence. All the registers do not have to be set and only some registers may be read.

Figure 7-21. Procedure for Reading from Real-time Clock 2 (when the Alarm Interrupt is in Use)



(Note 1, Cautions, and Remarks are listed on the next page.)

Note 1. Be sure to confirm that RWST = 0 before shifting to STOP mode.

Caution Complete setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.

Remark SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR may be read in any sequence. All the registers do not have to be set and only some registers may be read.

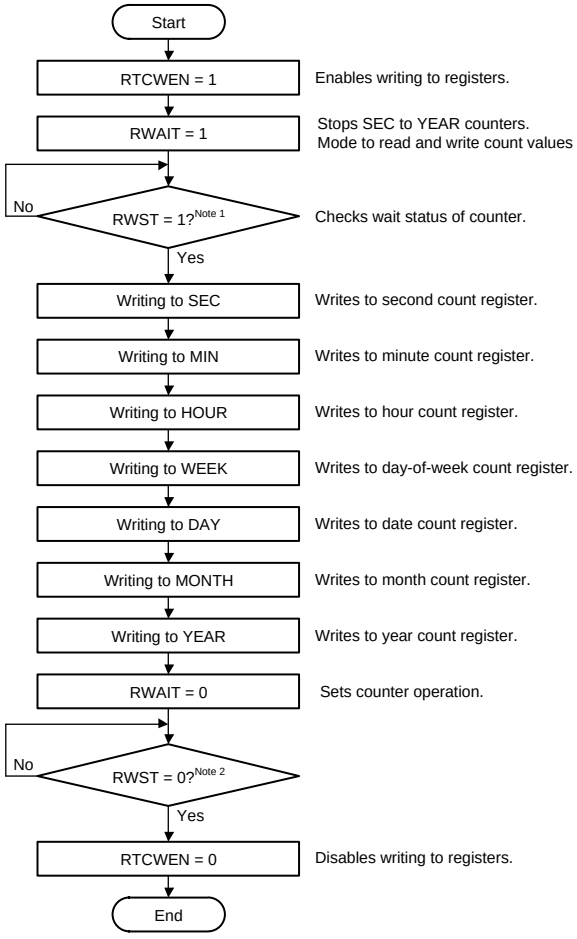
7.4.4 Writing to real-time clock 2 counter

Write to the counter after setting RWAIT to 1.

Set RWAIT to 0 after completion of writing the counter.

When using the alarm interrupt, write to the counter according to the procedure shown in **Figure 7-23**.

Figure 7-22. Procedure for Writing Real-time Clock 2 Counter



Note 1. When the counter is stopped (RTCE = 0), RWST is not set to 1.

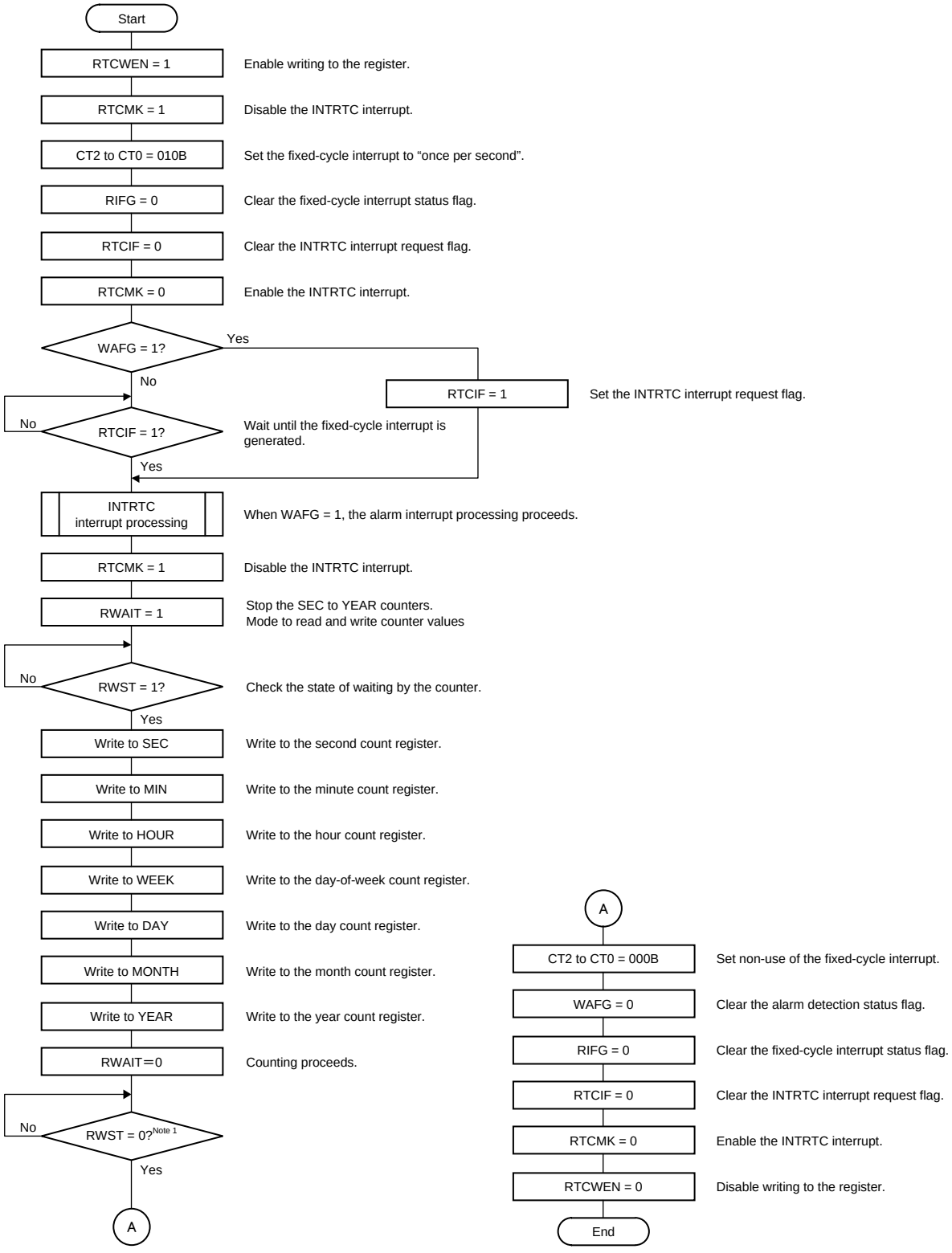
Note 2. Be sure to confirm that RWST = 0 before shifting to STOP mode.

Caution 1. Complete setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.

Caution 2. When changing the values of the SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR registers while the counter operates (RTCE = 1), rewrite the values of the MIN register after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG, and RTCIF flags after rewriting the MIN register.

Remark SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR may be read in any sequence. All the registers do not have to be set and only some registers may be written.

Figure 7-23. Procedure for Writing to Real-time Clock 2 (when the Alarm Interrupt is in Use)



(Note 1, Caution 1, Caution 2, and Remarks are listed on the next page.)

Note 1. Be sure to confirm that RWST = 0 before shifting to STOP mode.

Caution 1. Complete setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.

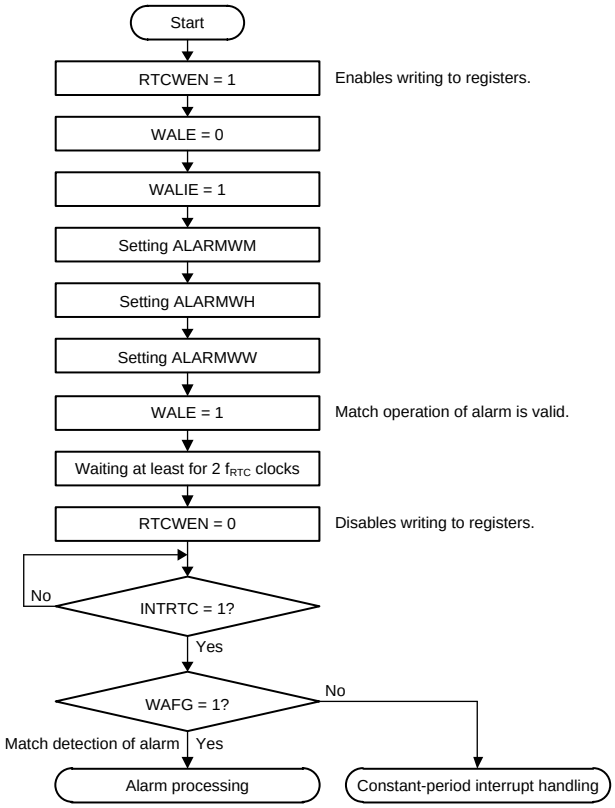
Caution 2. When changing the values of the SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR registers while the counter operates (RTCE = 1), rewrite the values of the MIN register after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG, and RTCIF flags after rewriting the MIN register.

Remark SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR may be read in any sequence. All the registers do not have to be set and only some registers may be written.

7.4.5 Setting alarm of real-time clock 2

Set the alarm time after setting WALE to 0 (to disable alarm operation).

Figure 7-24. Alarm Setting Procedure

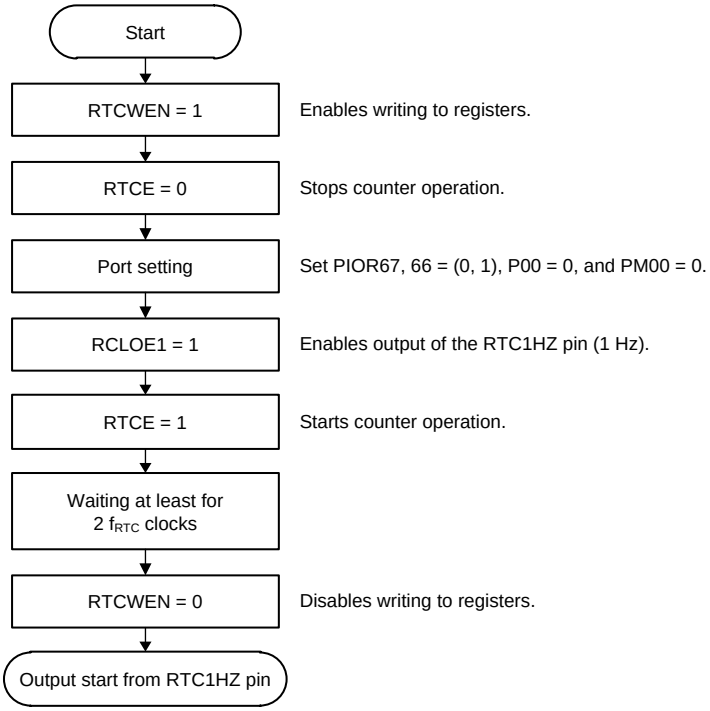


Remark 1. ALARMWM, ALARMWH, and ALARMWW may be written in any sequence.

Remark 2. Constant-period interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the constant-period interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

7.4.6 1-Hz output of real-time clock 2

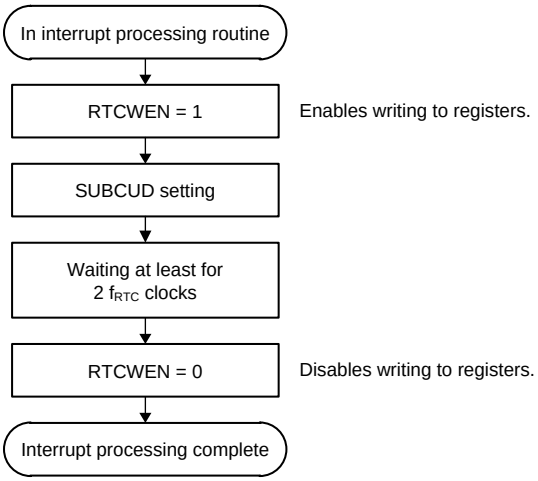
Figure 7-25. 1-Hz Output Setting Procedure (When Outputting from P00 with 32-Pin Products)



7.4.7 Clock error correction register setting procedure

RTC correction may not be successful if there is a conflict between the clock error correction register (SUBCUD) rewrite and correction timing. In order to prevent conflict between the correction timing and rewrite of the SUBCUD register, be sure to complete rewrite of the SUBCUD register before the next correction timing occurs (within approx. 0.5 seconds), which is calculated starting from the correction timing interrupt (INTRTIT) or periodic interrupt (INTRTC) that is synchronized with the correction timing.

- Set the clock error correction register after setting RTCWEN to 1. Set RTCWEN to 0 after completion of register setting.



7.4.8 Example of clock error correction of real-time clock 2

The clock can be corrected every second with a minimum resolution and accuracy of 0.96 ppm when it is slow or fast, by setting a value to the clock error correction register (SUBCUD).

The following shows how to calculate the target correction value, and how to calculate the F8 to F0 values of the clock error correction register from the target correction value.

● **Calculating the target correction value**

(Output frequency of the RTC1HZ pin using)

[Measuring the oscillation frequency]

The oscillation frequency of each product is measured by outputting 1 Hz from the RTC1HZ pin when the F15 bit of the clock error correction register (SUBCUD) is cleared to 0 (stops the clock error correction).

Remark See **7.4.6 1-Hz output of real-time clock 2** for the procedure of outputting about 1 Hz from the RTC1HZ pin.

[Calculating the target correction value]

(When the output frequency from the RTC1HZ pin is 0.9999817 Hz)

$$\text{Oscillation frequency} = 32768 \times 0.9999817 \fallingdotseq 32767.40 \text{ Hz}$$

Assume the target frequency to be 32768 Hz. Then the target correction value is calculated as follows.

$$\text{Target correction value} = \left(\text{Oscillation frequency} - \text{Target frequency}\right) \div \text{Target frequency}$$

$$= (32767.40 - 32768.00) \div 32768.00$$

$$\fallingdotseq -18.3 \text{ ppm}$$

Remark 1. The oscillation frequency is the frequency of the input clock (f_{RTC}). It can be calculated from the output frequency of the RTC1HZ pin × 32768 when clock error correction is not operating.

Remark 2. The target correction value is the oscillation frequency deviation (unit: [ppm]) of the crystal resonator.

Remark 3. The target frequency is the frequency resulting after clock error correction performed.

● **Calculating the F8 to F0 value of the clock error correction register (SBUCUD)**

The F8 to F0 values in the SUBCUD register is calculated from the target correction value by using the following expression.

Calculate the F8 to F0 values in the SUBCUD register from the target correction value by using the following expression.

$$\mathrm{SUBCUD}[8:0] = \left(\frac{\text{Target correction value [ppm]} \times 2^{15}}{10^{6}}\right)_{\text{2's complement (9−bit fixed−point format)}} + 0001.00000\mathrm{B}$$

Examples 1) When target correction value = −18.3 [ppm]

$$\mathrm{SUBCUD}[8:0] = \left(\frac{-18.3 \times 2^{15}}{10^{6}}\right)_{\text{2's complement (9−bit fixed−point format)}} + 0001.00000\mathrm{B}$$

$$= (-0.59965)_{\text{2's complement (9−bit fixed−point format)}} + 0001.00000\mathrm{B}$$

$$= 1111.01101\mathrm{B} + 0001.00000\mathrm{B}$$

$$= 0000.01101\mathrm{B}$$

Examples 2) When target correction value = 94.0 [ppm]

$$\mathrm{SUBCUD}[8:0] = \left(\frac{94.0 \times 2^{15}}{10^{6}}\right)_{\text{2's complement (9−bit fixed−point format)}} + 0001.00000\mathrm{B}$$

$$= (+3.08019)_{\text{2's complement (9−bit fixed−point format)}} + 0001.00000\mathrm{B}$$

$$= 0011.00011\mathrm{B} + 0001.00000\mathrm{B}$$

$$= 0100.00011\mathrm{B}$$

CHAPTER 8 12-BIT INTERVAL TIMER

8.1 Functions of 12-bit Interval Timer

An interrupt request signal (INTIT) is generated at any previously specified time interval. It can be utilized as the trigger for waking up from STOP mode and HALT mode.

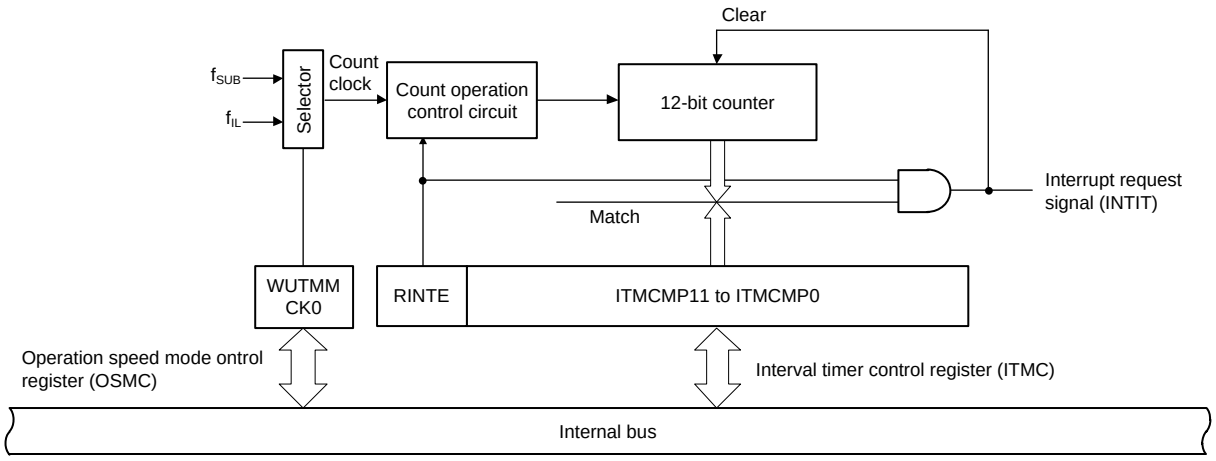
8.2 Configuration of 12-bit Interval Timer

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 0 (PER0)
	Operation speed mode control register (OSMC)
	Interval timer control register (ITMC)

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



8.3 Registers Controlling 12-bit Interval Timer

The 12-bit interval timer is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Operation speed mode control register (OSMC)
- Interval timer control register (ITMC)

8.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise.

When using the 12-bit interval timer, be sure to set bit 7 (TMKAEN) to 1 at first.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 8-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PER0	TMKAEN	CMPEN	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

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 12-bit interval timer is in the reset status.
1	Enables input clock supply. • SFR used by the 12-bit interval timer can be read and written.

Caution 1. Set the WUTMMCK0 bit of the OSMC register to 1 to determine the clock source for counting before supplying an input clock signal to the 12-bit interval timer (TMKAEN = 1).

Caution 2. When setting the 12-bit interval timer, be sure to first set the TMKAEN bit to 1 and then set the following register, while oscillation of the count clock is stable.
If TMKAEN = 0, writing to the 12-bit interval timer is ignored, and all read values are default values (except for the operation speed mode control register (OSMC)).

- **Interval timer control register (ITMC)**

Caution 3. Be sure to clear the following bits to 0.
10-pin and 16-pin products: Bits 1 and 3
20-, 24-, and 32-pin products: Bit 1

8.3.2 Operation speed mode control register (OSMC)

The WUTMMCK0 bit can be used to control supply of the 12-bit interval timer count clock.

Set the WUTMMCK0 bit to 1 before operating the 12-bit interval timer. Do not clear the WUTMMCK0 bit to 0 before counter operation has stopped.

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

Reset signal generation clears this register to 00H.

Figure 8-3. Format of Operation Speed Mode Control Register (OSMC)

Address: F00F3H After reset: 00H R/W

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

RTCLPC	Setting in STOP mode or HALT mode while subsystem clock is selected as CPU clock
0	Enables supply of subsystem clock to peripheral functions. (See **CHAPTER 17 STANDBY FUNCTION** for peripheral functions whose operations are enabled.)
1	Stops supply of the subsystem clock to peripheral functions other than high-precision RTC and 12-bit interval timer.

WUTMMCK0	Supply of the count clock for high-precision RTC and 12-bit interval timer
0	Supplies the subsystem clock (f_{SUB}).
1	Low-speed on-chip oscillator clock (f_{IL}) supply

8.3.3 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 sets this register to 0FFFH.

Figure 8-4. Format of Interval Timer Control Register (ITMC)

Address: FFF90H After reset: 0FFFH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ITMC	RINTE	0	0	0	ITCMP11	ITCMP10	ITCMP9	ITCMP8	ITCMP7	ITCMP6	ITCMP5	ITCMP4	ITCMP3	ITCMP2	ITCMP1	ITCMP0

RINTE	12-bit interval timer operation control
0	Count operation stopped (count clear)
1	Count operation started

ITCMP11 to ITCMP0	Specification of the 12-bit interval timer compare value
001H	These bits generate an interrupt at the fixed cycle (count clock cycles x (ITCMP setting + 1)).
. . .	
FFFH	
000H	Setting prohibited

Example interrupt cycles when 001H or FFFH is specified for ITCMP11 to ITCMP0

- ITCMP11 to ITCMP0 = 001H, count clock: when f_{IL} = 15 kHz
  1/15 [kHz] × (1 + 1) ≅ 0.1333 [ms] = 133.3 [μs]
- ITCMP11 to ITCMP0 = FFFH, count clock: when f_{IL} = 15 kHz
  1/15 [kHz] × (4095 + 1) ≅ 273 [ms]

<R> **Caution 1. Set the ITMK flag to 1 to disable processing of the INTIT interrupt before stopping the counter (by clearing the RINTE bit to 0). To restart counter operation (by setting the RINTE bit to 1), clear the ITIF flag to 0 and then clear the ITMK flag to enable INTIT interrupt processing.**

Caution 2. The value read from the RINTE bit is applied one count clock cycle after setting the RINTE bit to 1.

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 the counting operation is stopped (RINTE = 0). However, it is possible to change the settings of the ITCMP11 to ITCMP0 bits at the same time as when changing the setting of the RINTE bit 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 a 12-bit interval timer that repeatedly generates interrupt requests (INTIT).

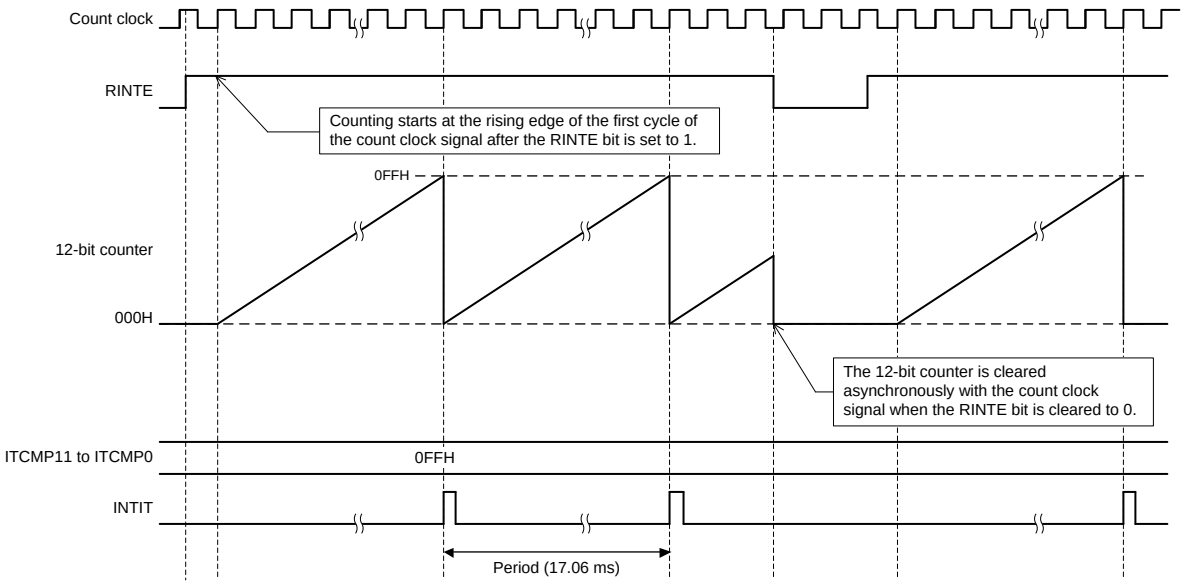
When the RINTE bit is set to 1, the 12-bit counter starts counting.

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 shown in **Figure 8-5**.

Figure 8-5. 12-bit Interval Timer Operation Timing

(ITCMP11 to ITCMP0 = 0FFH, count clock: f_{IL} = 15 kHz)

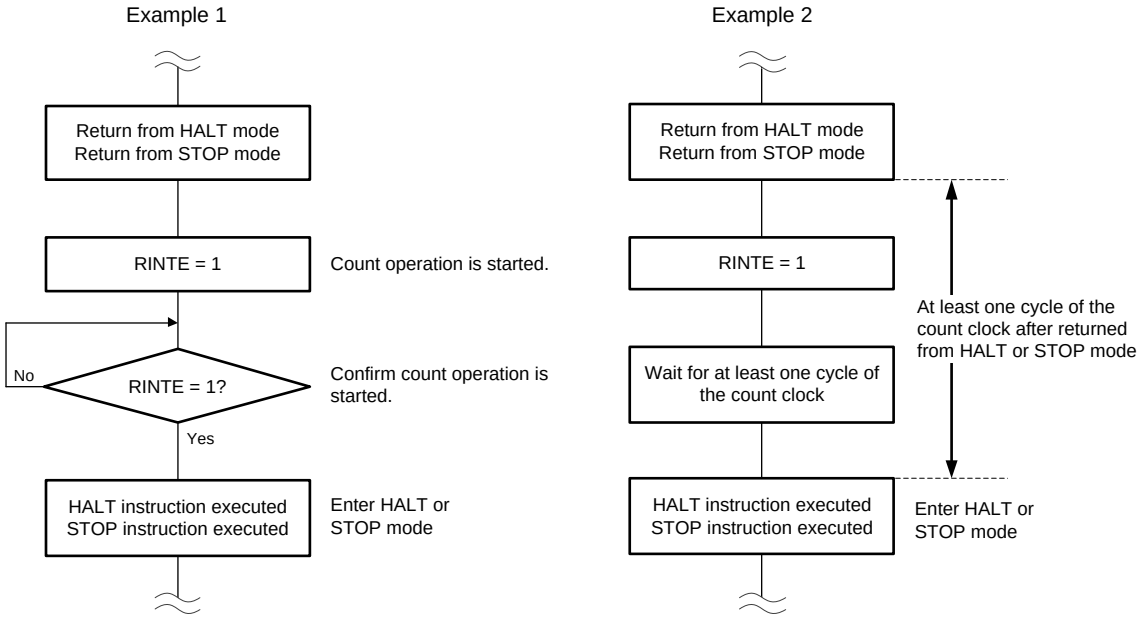


8.4.2 Start of count operation and re-enter to HALT/STOP mode after returned from HALT/STOP mode

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 8-6**).
- 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 8-6**).

Figure 8-6. Procedure of entering to HALT or STOP mode after setting RINTE to 1



CHAPTER 9 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

9.1 Functions of Clock Output/Buzzer Output Controller

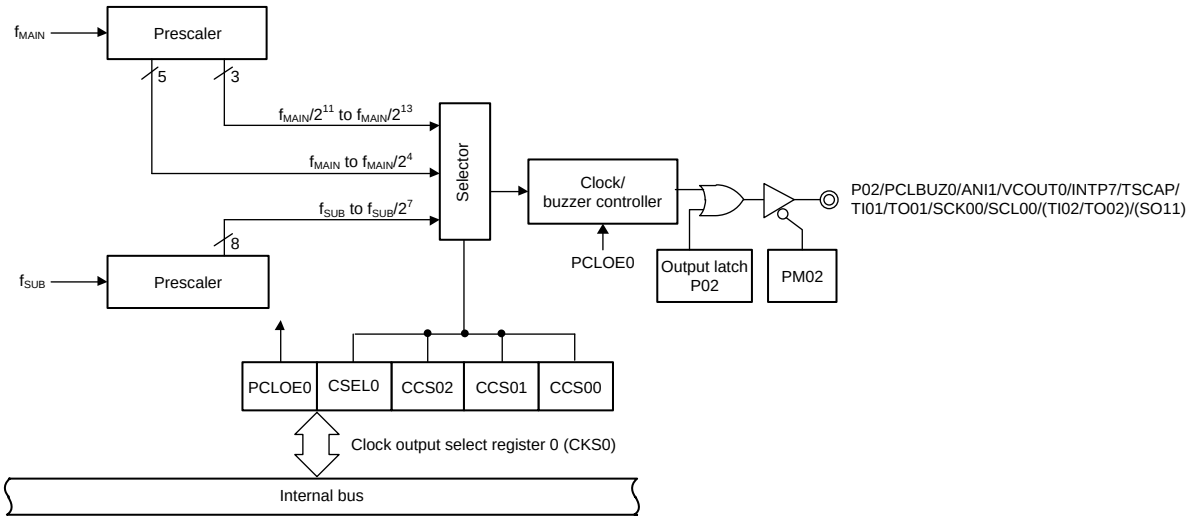
The clock output controller is intended for clock output for supply to peripheral ICs. Buzzer output is a function to output a square wave of buzzer frequency.

One pin can be used to output a clock or buzzer sound.

The PCLBUZ0 pin outputs a clock selected by clock output select register 0 (CKS0).

Figure 9-1 shows a block diagram of the clock output/buzzer output controller.

Figure 9-1. Block Diagram of Clock Output/Buzzer Output Controller



Caution 1. **For output frequencies available from the PCLBUZ0 pin, refer to 26.4 AC Characteristics and 27.4 AC Characteristics.**

Caution 2. **It is not possible to output the subsystem clock (f_{SUB}) from the PCLBUZ0 pin while the RTCLPC bit of the subsystem clock supply mode control register (OSMC), which controls the supply of the subsystem clock, is set to 1 and moreover while HALT mode is set with the subsystem clock (f_{SUB}) selected as CPU clock.**

Caution 3. **Do not select f_{SUB} as output clock for the clock output/buzzer output when the WUTMMCK0 bit of the OSMC register has been set to 1.**

Remark The clock output/buzzer output pins in the above diagram show the information of 24-pin and 32-pin products with PIOR62 = 0, PIOR61 = 0, and PIOR60 = 0.

9.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

Table 9-1. Configuration of Clock Output/Buzzer Output Controller

Item	Configuration
Control registers	Clock output select register 0 (CKS0) Port mode registers 0, 1, 4 (PM0, PM1, PM4) Port registers 0, 1, 4 (P0, P1, P4) Port mode control register 0 (PMC0) Peripheral I/O redirection register 6 (PIOR6)

9.3 Registers Controlling Clock Output/Buzzer Output Controller

The following registers are used to control the clock output/buzzer output controller.

- Clock output select register 0 (CKS0)
- Port mode registers 0, 1, 4 (PM0, PM1, PM4)
- Port registers 0, 1, 4 (P0, P1, P4)
- Port mode control register 0 (PMC0)
- Peripheral I/O redirection register 6 (PIOR6)

9.3.1 Clock output select register 0 (CKS0)

This register enables or disables output from the pin for clock output or buzzer frequency output (PCLBUZ0) and sets the output clock.

The CKS0 register is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 9-2. Format of Clock Output Select Register 0 (CKS0)

Address: FFFA5H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CKS0	PCLOE0	0	0	0	CSEL0	CCS02	CCS01	CCS00

PCLOE0	PCLBUZ0 pin output enable/disable specification
0	Output disable (default)
1	Output enable

CSEL0	CCS02	CCS01	CCS00	PCLBUZ0 pin output clock selection					
					f_{MAIN} (MHz)				
					1	5	8	10	16
0	0	0	0	f_{MAIN}	1 MHz	5 MHz Note 1	8 MHz Note 1	10 MHz Note 1	Setting prohibited Note 1
0	0	0	1	$f_{MAIN}/2$	500 kHz	2.5 MHz	4 MHz	5 MHz Note 1	8 MHz Note 1
0	0	1	0	$f_{MAIN}/2^2$	250 kHz	1.25 MHz	2 MHz	2.5 MHz	4 MHz
0	0	1	1	$f_{MAIN}/2^3$	125 kHz	625 kHz	1 MHz	1.25 MHz	2 MHz
0	1	0	0	$f_{MAIN}/2^4$	62.5 kHz	312.5 kHz	500 kHz	625 kHz	1 MHz
0	1	0	1	$f_{MAIN}/2^{11}$	488 Hz	2.44 kHz	3.91 kHz	4.88 kHz	7.81 kHz
0	1	1	0	$f_{MAIN}/2^{12}$	244 Hz	1.22 kHz	1.95 kHz	2.44 kHz	3.91 kHz
0	1	1	1	$f_{MAIN}/2^{13}$	122 Hz	610 Hz	977 Hz	1.22 kHz	1.95 kHz
1	0	0	0	f_{SUB}	32.768 kHz				
1	0	0	1	$f_{SUB}/2$	16.384 kHz				
1	0	1	0	$f_{SUB}/2^2$	8.192 kHz				
1	0	1	1	$f_{SUB}/2^3$	4.096 kHz				
1	1	0	0	$f_{SUB}/2^4$	2.048 kHz				
1	1	0	1	$f_{SUB}/2^{11}$	1.024 kHz				
1	1	1	0	$f_{SUB}/2^{12}$	512 Hz				
1	1	1	1	$f_{SUB}/2^{13}$	256 Hz				

Note 1. The available output clock varies depending on the operating voltage range. For detail, refer to **26.4 AC Characteristics** and **27.4 AC Characteristics**.

Caution 1. Change the output clock after disabling the PCLBUZ0 pin output (PCLOE0 = 0).

Caution 2. To shift to STOP mode, execute the STOP instruction after at least 1.5 clock cycles of the PCLBUZ0 pin output clock have elapsed following disabling of the PCLBUZ0 pin output (PCLOE0 = 0).

Caution 3. **To shift to STOP mode when the main system clock is selected (CSELn = 0), set PCLOEn = 0 before executing the STOP instruction. When the subsystem clock is selected (CSELn = 1), PCLOEn = 1 can be set because the clock can be output while the RTCLPC bit of the subsystem clock supply mode control (OSMC) register (the bit which controls the supply of the subsystem clock) is set to 0 and moreover while STOP mode is set.**

Caution 4. **It is not possible to output the subsystem clock (f_{SUB}) from the PCLBUZn pin while the RTCLPC bit of the subsystem clock supply mode control register (OSMC), which controls the supply of the subsystem clock, is set to 1 and moreover while HALT mode is set with the subsystem clock (f_{SUB}) selected as CPU clock.**

Remark f_{MAIN}: Main system clock frequency
f_{SUB}: Subsystem clock frequency

9.3.2 Registers controlling port functions of clock output/buzzer output pin

Using the port pin for the clock output/buzzer output controller requires setting of the registers that control the port function multiplexed on the clock output/buzzer output pin (PCLBUZ0 pin): (port mode registers 0, 1, 4 (PM0, PM1, PM4), port registers 0, 1, 4 (P0, P1, P4), port mode control register 0 (PMC0), peripheral I/O redirection register 6 (PIOR6)).

For details on the registers that control the port functions, see **4.3.1 Port mode registers 0, 1, 2, 4, 6, 12 (PM0, PM1, PM2, PM4, PM6, PM12)**, **4.3.2 Port registers 0, 1, 2, 4, 6, 12, 13 (P0, P1, P2, P4, P6, P12, P13)**, **4.3.5 Port mode control registers 0, 2 (PMC0, PMC2)**, and **4.3.6 Peripheral I/O redirection registers 0 to 6 (PIOR0 to PIOR6)**.

When you intend to use the PCLBUZ0 pin, set the corresponding bits in the port mode register (PM0) and port mode control register 0 (PMC0) to 0 and the corresponding bits in the port register (P0) and port output mode register (POM0) to 0.

For details, see **4.5.3 Register setting examples for used port and alternate functions**.

The table below lists the clock output/buzzer output pins set by the PIOR6 register.

● 24- and 32-Pin Products

PIOR62	PIOR61	PIOR60	Clock Output/Buzzer Output Pin
0	0	0	P02 (initial value)
0	0	1	P40
0	1	0	P06
0	1	1	P10
1	0	0	P11

● 16- and 20-Pin Products

PIOR61	PIOR60	Clock Output/Buzzer Output Pin
0	0	P02 (initial value)
0	1	P40
1	0	P06

● 10-Pin Products

PIOR60	Clock Output/Buzzer Output Pin
0	P02 (initial value)
1	P40

9.4 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).

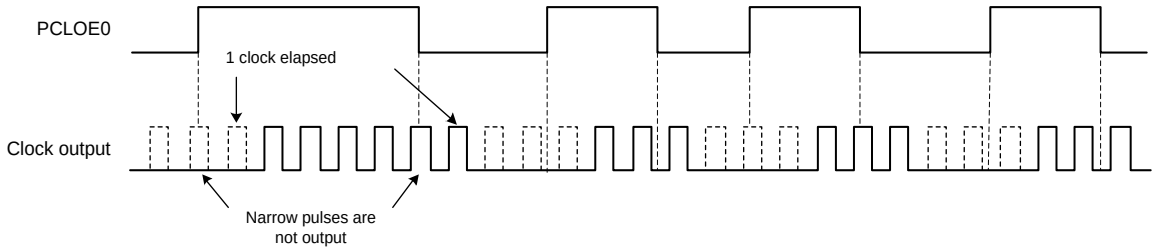
9.4.1 Operation as output pin

The PCLBUZ0 pin is output as the following procedure.

1. Set the corresponding bits in the port mode register (PM0/PM1/PM4), port register (P0/P1/P4), and port mode control register 0 (PMC0) for the port pin on which the PCLBUZ0 function is multiplexed to 0.
2. Select the output frequency with bits 0 to 3 (CCS00 to CCS02, CSEL0) of the clock output select register (CKS0) for the PCLBUZ0 pin (output is disabled).
3. Set bit 7 (PCLOE0) of the CKS0 register to 1 to enable clock/buzzer output.

Remark The controller used for clock output starts or stops clock output one clock after enabling or disabling of clock output (by the PCLOE bit) is switched. At this time, pulses with a narrow width are not output.
Figure 9-3 shows enabling or stopping of output by the PCLOE0 bit and the timing of clock output.

Figure 9-3. Timing of Clock Output from PCLBUZ0 Pin



Remark **When the main system clock is selected for the PCLBUZn output (CSELn = 0), if STOP mode is entered within 1.5 clock cycles of the PCLBUZ0 pin output clock after disabling the PCLBUZ0 pin output (PCLOE0 = 0), the width of the PCLBUZ0 pin output pulse becomes shorter. In such cases, only execute the STOP instruction after at least 1.5 clock cycles of the PCLBUZ0 pin output clock have elapsed following disabling of the PCLBUZ0 pin output.**

CHAPTER 10 WATCHDOG TIMER

10.1 Functions of Watchdog Timer

The counting operation of the watchdog timer is set by the user option byte (000C0H).

The watchdog timer operates on the low-speed on-chip oscillator clock.

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

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 18 RESET FUNCTION**.

When 75% of the overflow time is reached, an interval interrupt is generated.

10.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 10-1. Configuration of Watchdog Timer

Item	Configuration
Control register	Watchdog timer enable register (WDTE)

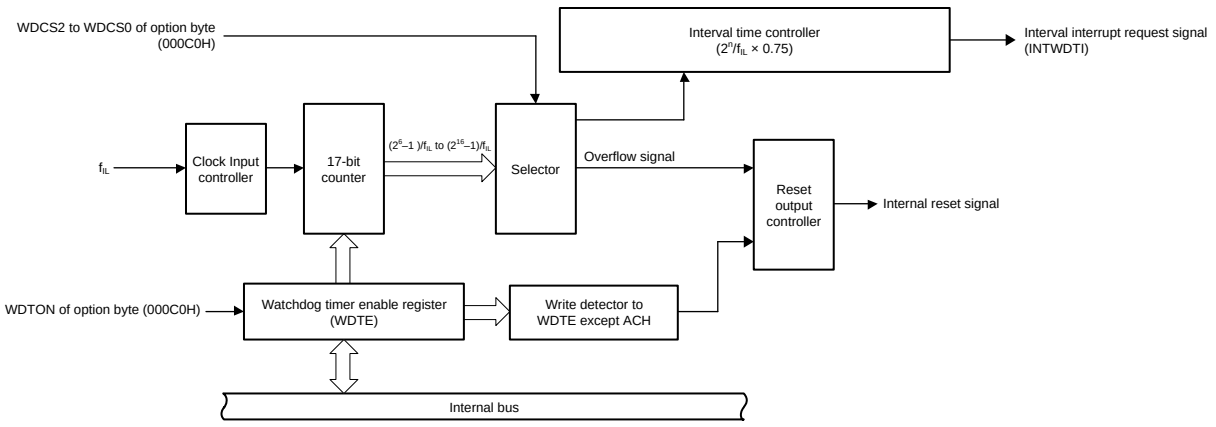
How the counter operation is controlled and the overflow time are set by the option byte.

Table 10-2. Setting of Option Bytes and Watchdog Timer

Setting of Watchdog Timer	Option Byte (000C0H)
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 21 OPTION BYTE**.

Figure 10-1. Block Diagram of Watchdog Timer



10.3 Register Controlling Watchdog Timer

The watchdog timer is controlled by the watchdog timer enable register (WDTE).

10.3.1 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 1AH or 9AH[Note 1].

Figure 10-2. Format of Watchdog Timer Enable Register (WDTE)

Address: FFFABH After reset: 1AH/9AH[Note 1] R/W

Symbol	7	6	5	4	3	2	1	0
WDTE								

WDTON Bit Setting Value	WDTE Register Reset Value
0 (watchdog timer count operation disabled)	1AH
1 (watchdog timer count operation enabled)	9AH

Note 1. The WDTE register reset value differs depending on the WDTON bit setting value of the option byte (000C0H). To operate the watchdog timer, set the WDTON bit to 1.

Caution 1. If a value other than “ACH” is written to the WDTE register, an internal reset signal is generated.

Caution 2. If a 1-bit memory manipulation instruction is executed for the WDTE register, an internal reset signal is generated.

Caution 3. The value read from the WDTE register is 1AH/9AH (this differs from the written value (ACH)).

10.4 Operation of Watchdog Timer

10.4.1 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 21 OPTION BYTE**).

WDTON	Watchdog Timer Counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

- Set an overflow time by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H) (for details, see **10.4.2 Setting time of watchdog timer** and **CHAPTER 21 OPTION BYTE**).

<2> After a reset release, the watchdog timer starts counting.

<3> By writing “ACH” to the watchdog timer enable register (WDTE) after the watchdog timer starts counting and before the overflow time set by the option byte, the watchdog timer is cleared and starts counting again.

<4> If the overflow time expires without “ACH” written to the WDTE register, an internal reset signal is generated. An internal reset signal is also generated in the following cases.

- If a 1-bit manipulation instruction is executed on the WDTE register
- If data other than “ACH” is written to the WDTE register

Caution 1. If the watchdog timer is cleared by writing “ACH” to the WDTE register, the actual overflow time may become shorter than the overflow time set by the option byte by up to one clock cycle of f_{IL}.

Caution 2. Clearing the watchdog timer is effective immediately before the counter value overflows.

Caution 3. 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: Watchdog timer operation stops.
WDSTBYON = 1: Watchdog timer operation continues.

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 clock[Note 1] 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. Accordingly, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 clock after the STOP mode release by an interval interrupt and the watchdog timer is to be cleared.

Caution 4. Setting WDTON = 0 and WDSTBYON = 1 is prohibited.

Note 1. Other than 10-pin products

10.4.2 Setting time of watchdog timer

Set the overflow time and interval interrupt 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) before the overflow time. When 75% of the overflow time is reached, an interval interrupt is generated.

The following overflow time and interval interrupt time can be set.

Table 10-3. Setting of Overflow Time and Interval Interrupt Time

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer (f_{IL} = 17.25 kHz (MAX.))	Interval Interrupt Time of Watchdog Timer (f_{IL} = 17.25 kHz (MAX.))
0	0	0	$(2^6-1)/f_{IL}$ (3.65 ms)	$2^6/f_{IL} \times 0.75$ (2.78 ms)
0	0	1	$(2^7-1)/f_{IL}$ (7.36 ms)	$2^7/f_{IL} \times 0.75$ (5.56 ms)
0	1	0	$(2^8-1)/f_{IL}$ (14.7 ms)	$2^8/f_{IL} \times 0.75$ (11.1 ms)
0	1	1	$(2^9-1)/f_{IL}$ (29.6 ms)	$2^9/f_{IL} \times 0.75$ (22.2 ms)
1	0	0	$(2^{11}-1)/f_{IL}$ (118 ms)	$2^{11}/f_{IL} \times 0.75$ (89.0 ms)
1	0	1	$(2^{13}-1)/f_{IL}$ (474 ms)	$2^{13}/f_{IL} \times 0.75$ (356 ms)
1	1	0	$(2^{14}-1)/f_{IL}$ (949 ms)	$2^{14}/f_{IL} \times 0.75$ (712 ms)
1	1	1	$(2^{16}-1)/f_{IL}$ (3799 ms)	$2^{16}/f_{IL} \times 0.75$ (2849 ms)

Caution 1. When operating with the X1 clock[Note 1] 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 clock[Note 1] after the STOP mode release by an interval interrupt and the watchdog timer is to be cleared.

Caution 2. 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.

Caution 3. The watchdog timer always generates an interval interrupt when the specified time is reached unless this is specifically disabled. If the interval interrupt from the watchdog timer is not to be used, be sure to disable the interrupt by setting the WDTIMK bit to 1.

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

Note 1. Other than 10-pin products

CHAPTER 11 A/D CONVERTER

The number of analog input channels of the A/D converter differs depending on the product.

- 10-pin products: 4 channels (ANI0 to ANI3), internal reference voltage[Note 1] (0.815 V (typ.)), temperature sensor output voltage, CTSU TSCAP voltage
- 16-pin products: 7 channels (ANI0 to ANI6), internal reference voltage[Note 1] (0.815 V (typ.)), temperature sensor output voltage, CTSU TSCAP voltage
- 20-pin products: 11 channels (ANI0 to ANI10), internal reference voltage[Note 1] (0.815 V (typ.)), temperature sensor output voltage, CTSU TSCAP voltage
- 24-pin products: 11 channels (ANI0 to ANI10), internal reference voltage[Note 1] (0.815 V (typ.)), temperature sensor output voltage, CTSU TSCAP voltage
- 32-pin products: 11 channels (ANI0 to ANI10), internal reference voltage[Note 1] (0.815 V (typ.)), temperature sensor output voltage, CTSU TSCAP voltage

Note 1. The internal reference voltage cannot be used for the A/D converter and the comparator simultaneously. When the internal reference voltage is selected as the target of conversion by the A/D converter, do not set the internal reference voltage as the reference voltage of the comparator.

11.1 Function of A/D Converter

The A/D converter is used to convert analog input signals into digital values, and is configured to control up to 11 channels of A/D converter analog inputs. The internal reference voltage, temperature sensor output voltage, or CTSU TSCAP voltage is selectable. 10-bit or 8-bit resolution can also be selected by using the ADTYP bit of A/D converter mode register 2 (ADM2).

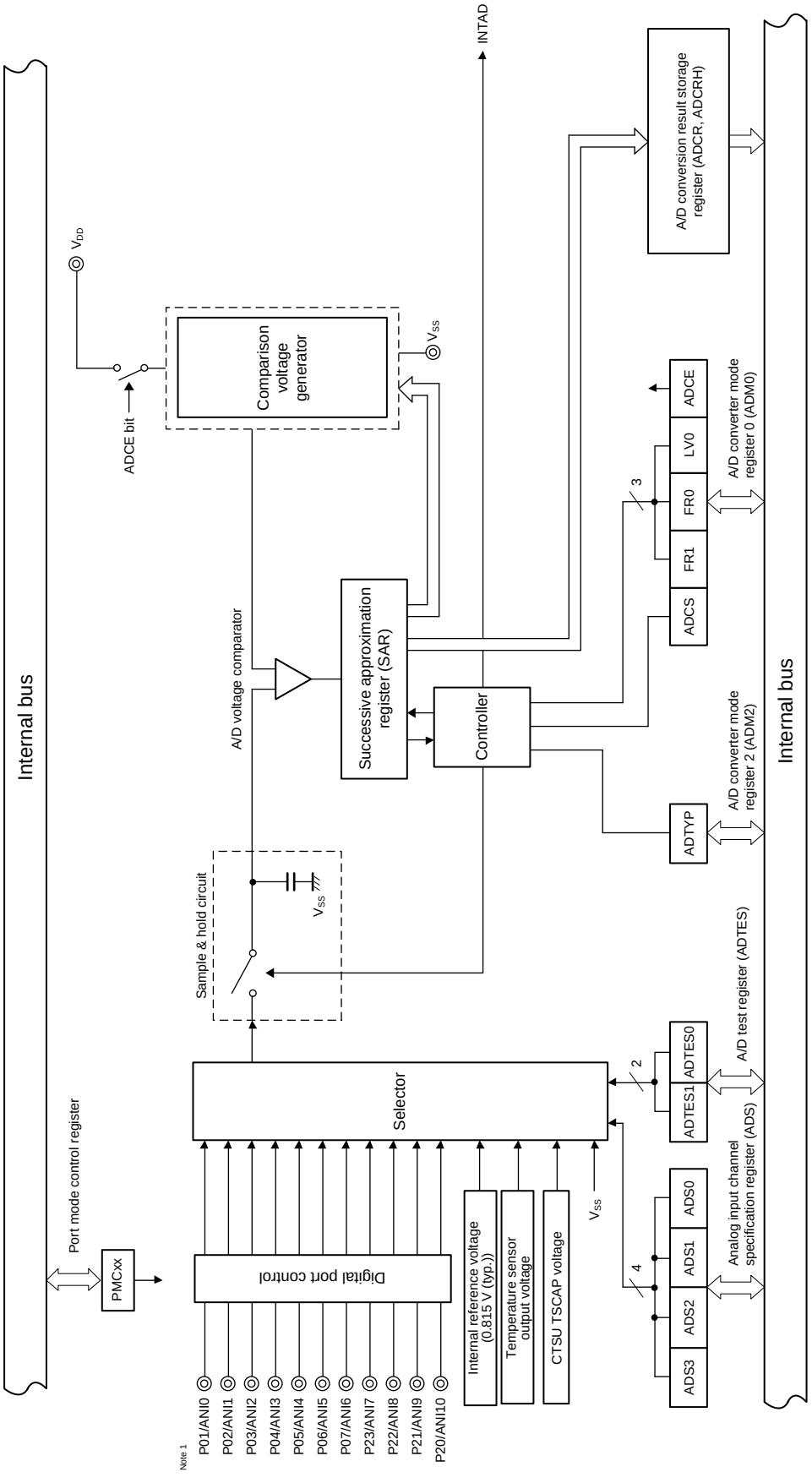
The A/D converter has the following function.

■ 10-bit/8-bit resolution A/D conversion

Following the selection of one analog input channel from among ANI0 to ANI10, software initiates A/D conversion with 10-bit or 8-bit resolution. An A/D conversion end interrupt request signal (INTAD) is generated on completion of A/D conversion.

The range of operating voltage for the A/D converter is from 2.4 V to 5.5 V.

Figure 11-1. Block Diagram of A/D Converter



Note 1. ANI0 to ANI3 for 10-pin products; ANI0 to ANI6 for 16-pin products

11.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

1) ANI0 to ANI10[Note 1]

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.

In addition to the voltages on analog input pins from ANI0 to ANI10, the internal reference voltage (0.815 (typ.)), temperature sensor output voltage, or CTSU TSCAP voltage can be selected as the target of conversion by the A/D converter.

Note 1. ANI0 to ANI3 for 10-pin products; ANI0 to ANI6 for 16-pin products

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

The 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 V_{DD}) 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 V_{DD}), 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 V_{DD})

Bit 9 = 1: (3/4 V_{DD})

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.

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 used to set voltage tap data whose values from the comparison voltage generator match the voltage values of the analog input pins, one bit at a time starting from the most significant bit (MSB).

If data is set in the SAR register all the way to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register (conversion results) are held in the A/D conversion result register (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)

Each time A/D conversion is completed, the conversion result is loaded from the successive approximation register and the ten higher-order bits of the A/D conversion result are held in this register (the six lower-order bits are fixed to 0).

7) 8-bit A/D conversion result register (ADCRH)

Each time A/D conversion is completed, the conversion result is loaded from the successive approximation register and the eight higher-order bits of the A/D conversion result are held in this register.

8) Controller

This circuit controls the conversion time of an analog input signal that is to be converted into a digital signal, as well as starting and stopping of the conversion operation. When A/D conversion has been completed, this controller generates an A/D conversion end interrupt request signal (INTAD).

11.3 Registers Controlling A/D Converter

The A/D converter is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- A/D converter mode register 0 (ADM0)
- 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)
- A/D test register (ADTES)
- Port mode registers 0, 2 (PM0, PM2)
- Port mode control registers 0, 2 (PMC0, PMC2)

11.3.1 Peripheral enable register 0 (PER0)

The PER0 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise.

If the A/D converter is to be used, be sure to set 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 11-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PER0	TMKAEN	CMPEN	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

ADCEN	Control of A/D converter input clock supply
0	Stops supply of an input clock. • The SFRs used by the A/D converter cannot be written. • The A/D converter is in the reset state.
1	Enables supply of an input clock. • The SFRs used by the A/D converter can be read/written.

Caution 1. When setting the A/D converter, make sure that ADCEN = 1 before setting the following registers. 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 and 2 (PM0, PM2) and port mode control registers 0 and 2 (PMC0, PMC2)).

- **A/D converter mode register 0 (ADM0)**
- **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)**
- **A/D test register (ADTES)**

Caution 2. Be sure to clear the following bits to 0.
10-pin and 16-pin products: Bits 1 and 3
20-pin, 24-pin, and 32-pin products: Bit 1

11.3.2 A/D converter mode register 0 (ADM0)

This register sets the time for converting analog input to digital data, and starts and stops conversion operation.

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 11-3. Format of A/D Converter Mode Register 0 (ADM0)

Address: FFF30H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADM0	ADCS	0	0	FR1[Note 1]	FR0[Note 1]	0	LV0[Note 1]	ADCE

ADCS	A/D conversion operation control
0	Stops conversion operation (conversion stopped/standby state)
1	Enables conversion operation (conversion operation state)
<Clear conditions> • 0 is written to ADCS. • The bit is automatically cleared to 0 when A/D conversion ends. <Set condition> • 1 is written to ADCS when ADCE = 1.	

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 FR1, FR0, and LV0 bits and A/D conversion, see **Table 11-2 10-Bit Resolution A/D Conversion Time Selection** or **Table 11-3 8-Bit Resolution A/D Conversion Time Selection**.

Note 2. The operation of the A/D voltage comparator is controlled by the ADCS and ADCE bits, and it takes 0.1 μs from the start of operation for the operation to stabilize. Therefore, when the ADCS bit is set to 1 after 0.1 μs or more have elapsed from the time ADCE bit is set to 1, the conversion result at that time has priority over the first conversion result. If the ADCS bit is set to 1 to perform A/D conversion without waiting for at least 0.1 μs, ignore data of the conversion.

Caution 1. Only rewrite the FR1, FR0, and LV0 bits while in the conversion standby state (ADCS = 0, ADCE = 1) or conversion is stopped (ADCS = 0, ADCE = 0). Rewriting the values of the FR1, FR0, and LV0 bits, and ADCS bits by an 8-bit manipulation instruction at the same time is prohibited.

Caution 2. Setting ADCS = 1 and ADCE = 0 is prohibited. When 1 is written to the ADCS bit while conversion is stopped (ADCE = 0, ADCS = 0), the ADCS bit is not set to 1.

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 follow the procedure described in 11.7 A/D Converter Setup Flowchart.

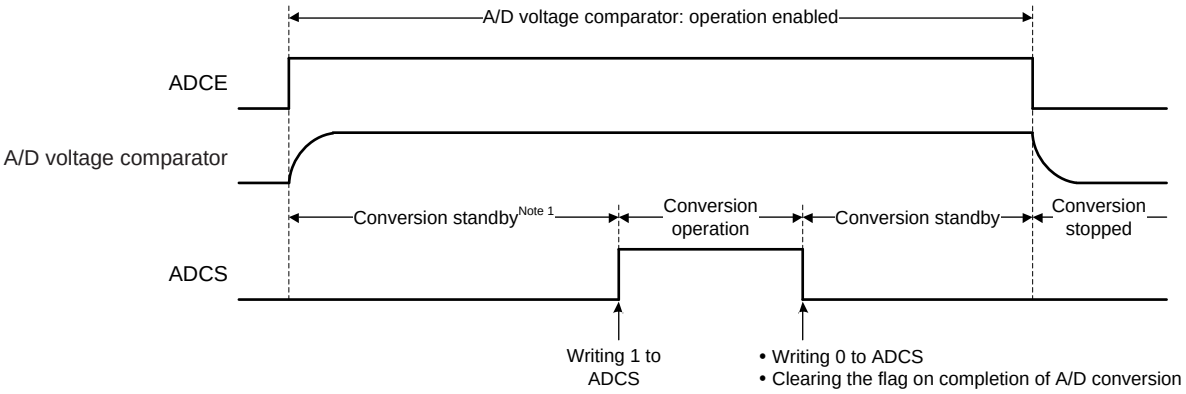
Caution 4. Be sure to clear bits 2, 5, and 6 to 0.

Caution 5. Setting the ADCS bit to 1 again during conversion (ADCS = 1) is prohibited. When re-conversion by the same channel is required, stop the conversion operation once (ADCS = 0), and then restart A/D conversion (ADCS = 1).

Table 11-1. 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

Figure 11-4. Timing Chart when A/D Voltage Comparator Is Used



Note 1. It requires at least 0.1 μs to stabilize the internal circuit until the A/D conversion operation is started (ADCS = 1) after the operation of the A/D voltage comparator is enabled (ADCE = 1). If the ADCS bit is set to 1 without waiting for at least 0.1 μs, ignore data of the first conversion.

Table 11-2. 10-Bit Resolution A/D Conversion Time Selection

A/D Converter Mode Register 0 (ADM0)			Conversion Clock	Number of Conversion Clocks	Conversion Time	Conversion Time Selection (μs)				
FR1	FR0	LV0 Note 2				f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 10 MHz	f_{CLK} = 16 MHz
0	0	0	$f_{CLK}/8$	23 f_{AD} (number of sampling clocks: 9 f_{AD})	$184/f_{CLK}$	Setting prohibited	Setting prohibited	23.0	18.4	11.5
0	1		$f_{CLK}/4$		$92/f_{CLK}$		23.0	11.5	9.2	5.75
1	0		$f_{CLK}/2$		$46/f_{CLK}$		11.5	5.75	Setting prohibited	Setting prohibited
1	1		f_{CLK}		$23/f_{CLK}$	23.0	5.75	Setting prohibited		
0	0	1 Note 1	$f_{CLK}/8$	17 f_{AD} (number of sampling clocks: 3 f_{AD})	$136/f_{CLK}$	Setting prohibited	Setting prohibited	17.0	13.6	8.5
0	1		$f_{CLK}/4$		$68/f_{CLK}$		17.0	8.5	6.8	4.25
1	0		$f_{CLK}/2$		$34/f_{CLK}$		8.5	4.25	Setting prohibited	Setting prohibited
1	1		f_{CLK}		$17/f_{CLK}$	17.0	4.25	Setting prohibited		

Note 1. Setting is prohibited when 2.4 V ≤ V_{DD} < 2.7 V. Can be selected when 2.7 V ≤ V_{DD} ≤ 5.5 V.

Note 2. When the internal reference voltage, temperature sensor output voltage, or CTSU TSCAP voltage is selected as the target of conversion by the A/D converter, be sure to clear the LV0 bit to 0.

Table 11-3. 8-Bit Resolution A/D Conversion Time Selection

A/D Converter Mode Register 0 (ADM0)			Conversion Clock	Number of Conversion Clocks	Conversion Time	Conversion Time Selection (μs)				
FR1	FR0	LV0 Note 2				f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 10 MHz	f_{CLK} = 16 MHz
0	0	0	$f_{CLK}/8$	21 f_{AD} (number of sampling clocks: 9 f_{AD})	$168/f_{CLK}$	Setting prohibited	Setting prohibited	21.0	16.8	10.5
0	1		$f_{CLK}/4$		$84/f_{CLK}$		21.0	10.5	8.4	5.25
1	0		$f_{CLK}/2$		$42/f_{CLK}$		10.5	5.25	Setting prohibited	Setting prohibited
1	1		f_{CLK}		$21/f_{CLK}$	21.0	5.25	Setting prohibited		
0	0	1 Note 1	$f_{CLK}/8$	15 f_{AD} (number of sampling clocks: 3 f_{AD})	$120/f_{CLK}$	Setting prohibited	Setting prohibited	15.0	12.0	7.5
0	1		$f_{CLK}/4$		$60/f_{CLK}$		15.0	7.5	6.0	3.75
1	0		$f_{CLK}/2$		$30/f_{CLK}$		7.5	3.75	Setting prohibited	Setting prohibited
1	1		f_{CLK}		$15/f_{CLK}$	15.0	Setting prohibited	Setting prohibited		

Note 1. Setting is prohibited when 2.4 V ≤ V_{DD} < 2.7 V. Can be selected when 2.7 V ≤ V_{DD} ≤ 5.5 V.

Note 2. When the internal reference voltage, temperature sensor output voltage, or CTSU TSCAP voltage is selected as the target of conversion by the A/D converter, be sure to clear the LV0 bit to 0.

Caution 1. The A/D conversion time must also be within the range of the conversion time (t_{CONV}) indicated in 26.6.1 A/D converter characteristics or 27.6.1 A/D converter characteristics.

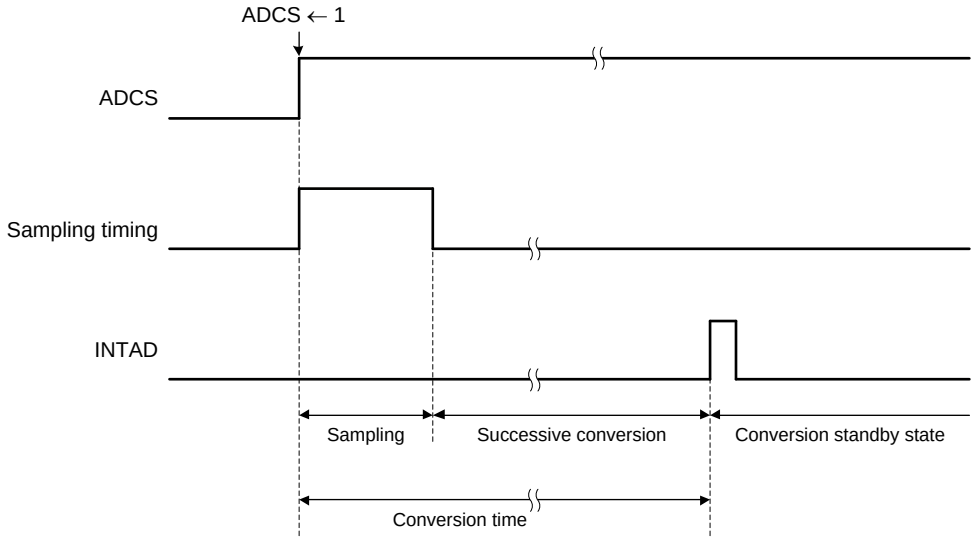
Caution 2. **When the internal reference voltage is selected as the target of conversion by the A/D converter, the internal reference voltage cannot be used as the reference voltage of the comparator.**

Caution 3. **Only rewrite the FR1, FR0, and LV0 bits while in the conversion standby state (ADCS = 0, ADCE = 1) or conversion is stopped (ADCS = 0, ADCE = 0). Rewriting the values of the FR1, FR0, and LV0 bits, and ADCS bits by an 8-bit manipulation instruction at the same time is prohibited.**

Caution 4. **The above conversion time does not include clock frequency errors. Select the conversion time, taking clock frequency errors into consideration.**

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Figure 11-5. A/D Converter Sampling and A/D Conversion Timing



11.3.3 A/D converter mode register 2 (ADM2)

This register is used to set the resolution of the A/D converter.

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 11-6. Format of A/D Converter Mode Register 2 (ADM2)

Address: F0010H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	[0]
ADM2	0	0	0	0	0	0	0	ADTYP

ADTYP	Selection of the A/D conversion resolution
0	10-bit resolution
1	8-bit resolution

Caution Only rewrite the value of the ADM2 register while conversion is stopped (ADCS = 0, ADCE = 0).

11.3.4 10-bit A/D conversion result register (ADCR)

This is a 16-bit register which holds the result of A/D conversion. The six lower-order bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR). The eight higher-order bits of the conversion result are stored in FFF1FH and the two lower-order bits are stored in the two higher-order bits of FFF1EH.

The ADCR register can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 11-7. Storing of the A/D Conversion Result in the Case of 10-bit Resolution

Address: FFF1FH, FFF1EH After reset: 0000H R

	FFF1FH								FFF1EH							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCR	ADCR 9	ADCR 8	ADCR 7	ADCR 6	ADCR 5	ADCR 4	ADCR 3	ADCR 2	ADCR 1	ADCR 0	0	0	0	0	0	0

Caution 1. When writing to A/D converter mode register 0 (ADM0) and the analog input channel specification register (ADS), the contents of the ADCR/ADCRH register may become undefined. After conversion ends, read the conversion result before writing to the ADM0 and ADS registers. Using timing other than the above may cause an incorrect conversion result to be read.

Caution 2. When the ADCR register is read while 8-bit resolution A/D conversion is selected (when the ADTYP bit of A/D converter mode register 2 (ADM2) is 1), 0 is read from the two lower-order bits (ADCR1, ADCR0). Note that, when the ADCR register is read before completion of A/D conversion while 8-bit resolution A/D conversion is selected, 0 may not be read from the two lower-order bits (ADCR1, ADCR0).

Caution 3. When the ADCR register is accessed in 16-bit units, the ten higher-order bits of the conversion result are read in order from bit 15 of the ADCR register.

11.3.5 8-bit A/D conversion result register (ADCRH)

This is an 8-bit register which holds the result of A/D conversion. When A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR). In the case of 10-bit resolution, the eight higher-order bits are stored.

The ADCRH register can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 11-8. 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	ADCR9	ADCR8	ADCR7	ADCR6	ADCR5	ADCR4	ADCR3	ADCR2

Caution When writing to A/D converter mode register 0 (ADM0) and the analog input channel specification register (ADS), the contents of the ADCR/ADCRH registers may become undefined. After conversion ends, read the conversion result before writing to the ADM0 and ADS registers. Using timing other than the above may cause an incorrect conversion result to be read.

11.3.6 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 11-9. 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	0	0	0	0	ADS3	ADS2	ADS1	ADS0

ADS3	ADS2	ADS1	ADS0	Target of A/D conversion	Analog input pin
0	0	0	0	ANI0	P01/ANI0 pin
0	0	0	1	ANI1	P02/ANI1 pin
0	0	1	0	ANI2	P03/ANI2 pin
0	0	1	1	ANI3	P04/ANI3 pin
0	1	0	0	ANI4	P05/ANI4 pin
0	1	0	1	ANI5	P06/ANI5 pin
0	1	1	0	ANI6	P07/ANI6 pin
0	1	1	1	ANI7	P23/ANI7 pin
1	0	0	0	ANI8	P22/ANI8 pin
1	0	0	1	ANI9	P21/ANI9 pin
1	0	1	0	ANI10	P20/ANI10 pin
1	0	1	1	CTSU TSCAP voltage[Note 1]	—
1	1	0	0	Temperature sensor output voltage[Note 1]	—
1	1	0	1	Internal reference voltage (0.815 V (typ.))[Note 1]	—
Other than the above				Setting prohibited	

Note 1. When the internal reference voltage, temperature sensor output voltage, or CTSU TSCAP voltage is selected as the target of conversion by the A/D converter, be sure to clear the LV0 bit in A/D converter mode register 0 (ADM0) to 0.

Caution 1. Only rewrite the ADS register while in the conversion standby state (ADCS = 0, ADCE = 1) or conversion is stopped (ADCS = 0, ADCE = 0).

Caution 2. For the ports which are used as analog input ports, select input mode with port mode registers 0 and 2 (PM0, PM2) and analog input with port mode control registers 0 and 2 (PMC0, PMC2). Do not use the ADS register to set the pins which are set as digital I/O with port mode control registers (PMC0, PMC2).

Caution 3. The internal reference voltage cannot be used for the A/D converter and the comparator simultaneously. When the internal reference voltage is selected as the target of conversion by the A/D converter (ADS3 to ADS0 = 1101B), it cannot be set as the reference voltage of the comparator.

Caution 4. Be sure to clear bits 4 to 7 to 0.

11.3.7 A/D test register (ADTES)

This register is used to select V_{SS} as the analog input to be A/D converted. When the internal reference voltage (0.815 V (typ.)), temperature sensor output voltage, or CTSU TSCAP voltage is selected as the target of A/D conversion, the sampling capacitor must be discharged before A/D conversion of the internal reference voltage, temperature sensor output voltage, or CTSU TSCAP voltage proceeds. Perform A/D conversion once by clearing the ADTES0 bit of the ADTES register to 0 and setting the ADTES1 bit to 1.

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

Reset signal generation clears this register to 00H.

Figure 11-10. 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	Selection of target of A/D conversion
0	0	ANIx/CTSU TSCAP voltage/temperature sensor output voltage/internal reference voltage (0.815 V (typ.)) (This is specified using the analog input channel specification register (ADS).)
0	1	Setting prohibited
1	0	V_{SS} (discharging the sampling capacitor)
1	1	V_{DD}

Caution When A/D converting the internal reference voltage (0.815 V (typ.)), temperature sensor output voltage, or CTSU TSCAP voltage, follow the procedure described in 11.7.2 Setting up the internal reference voltage, temperature sensor output voltage, or CTSU TSCAP voltage for A/D conversion to discharge the sampling capacitor once.

Remark Be sure to clear bits 2 to 7 to 0.

11.3.8 Registers controlling port function of analog input pins

Set up the registers for controlling the functions of the ports shared with the analog input pins of the A/D converter (port mode registers 0 and 2 (PM0, PM2) and port mode control registers 0 and 2 (PMC0, PMC2)). For details, see **4.3.1 Port mode registers 0, 1, 2, 4, 6, 12 (PM0, PM1, PM2, PM4, PM6, PM12)** and **4.3.5 Port mode control registers 0, 2 (PMC0, PMC2)**.

For an example of settings when used as the analog input pins of the A/D converter, see **4.5.3 Register setting examples for used port and alternate functions**.

When using the ANI0 to ANI10 pins as analog input of the A/D converter, set the corresponding bits in port mode registers 0 and 2 (PM0, PM2) and port mode control register 0 and 2 (PMC0, PMC2) to 1.

11.4 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 ends.

<3> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) V_{DD} 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) V_{DD}, the MSB bit of the SAR register remains set. If the analog input is smaller than (1/2) V_{DD}, the MSB bit is reset.

<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) V_{DD}

Bit 9 = 0: (1/4) V_{DD}

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

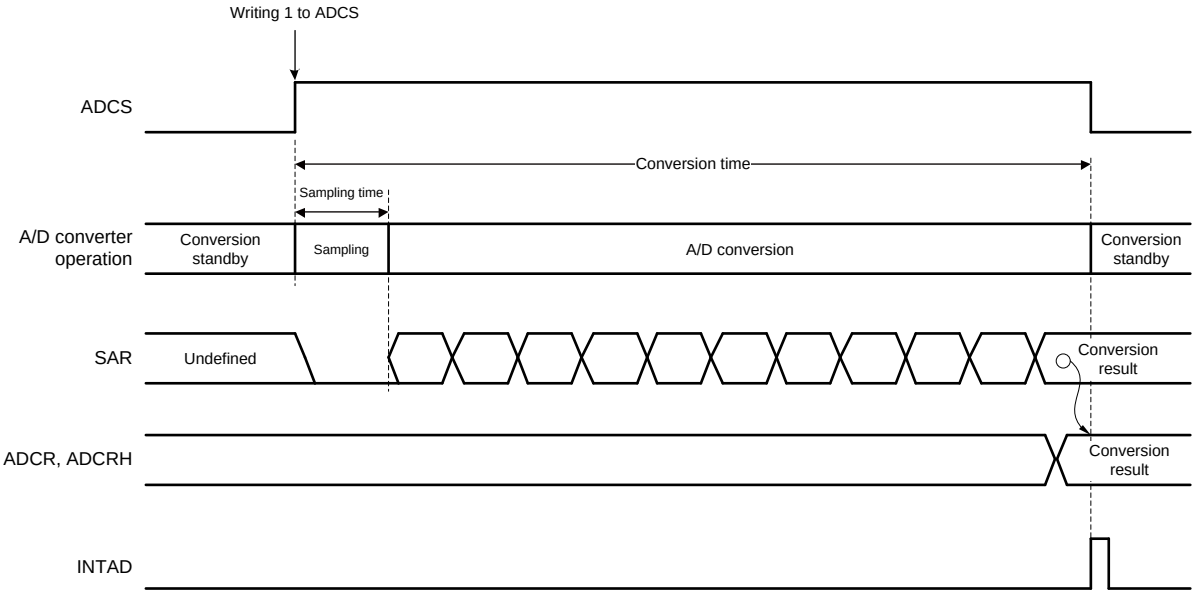
<6> Comparison is continued in this way up to bit 0 of the SAR register.

<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.
At the same time, the A/D conversion end interrupt request (INTAD) is generated.
After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the system enters the A/D conversion standby state.

Remark 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

Figure 11-11. Conversion Operation of A/D Converter (Software Trigger Mode)



A/D conversion is performed once when the bit 7 (ADCS) of A/D converter mode register 0 (ADM0) is set to 1 by software. The ADCS bit is automatically cleared to 0 after A/D conversion ends.

Reset signal generation clears the A/D conversion result register (ADCR, ADCRH) to 0000H or 00H.

11.5 Input Voltage and Conversion Results

The relationship between the analog input voltage input to the analog input (ANI0 to ANI10, internal reference voltage) and the theoretical A/D conversion result (stored in the 10-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$\mathrm{SAR} = \mathrm{INT}\left(\frac{\mathrm{V_{AIN}}}{\mathrm{V_{DD}}} \times 1024 + 0.5\right)$$

$$\mathrm{ADCR} = \mathrm{SAR} \times 64$$

or

$$\left(\frac{\mathrm{ADCR}}{64} - 0.5\right) \times \frac{\mathrm{V_{DD}}}{1024} \leq \mathrm{V_{AIN}} < \left(\frac{\mathrm{ADCR}}{64} + 0.5\right) \times \frac{\mathrm{V_{DD}}}{1024}$$

where, INT ( ): Function which returns integer part of value in parentheses

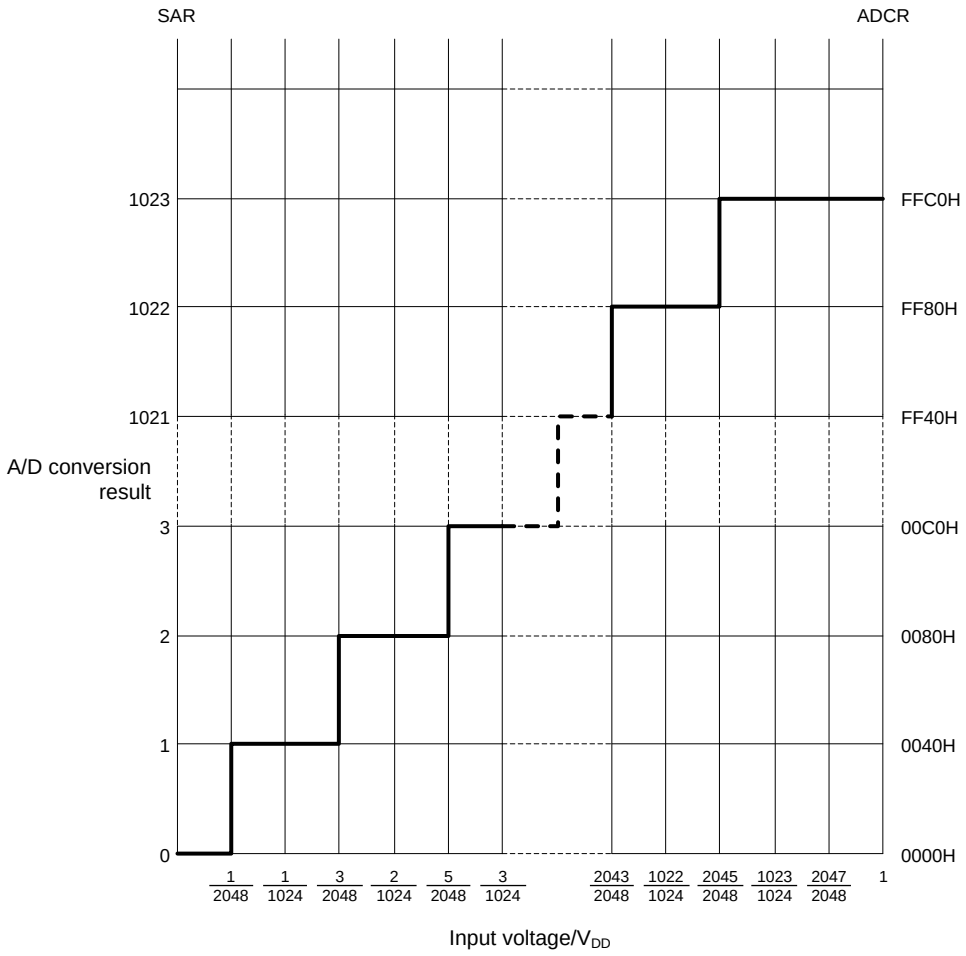
V_{AIN}: Analog input voltage

ADCR: A/D conversion result register (ADCR) value

SAR: Successive approximation register

Figure 11-12 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 11-12. Relationship between Analog Input Voltage and A/D Conversion Result



11.6 A/D Converter Operation Modes

The operation of the A/D converter is described below. In addition, the setting procedure is described in **11.7 A/D Converter Setup Flowchart**.

<1> While conversion is stopped, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the conversion standby state.

<2> After software counts the stabilization wait time (0.1 μs), the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion of the analog input specified by the analog input channel specification register (ADS).

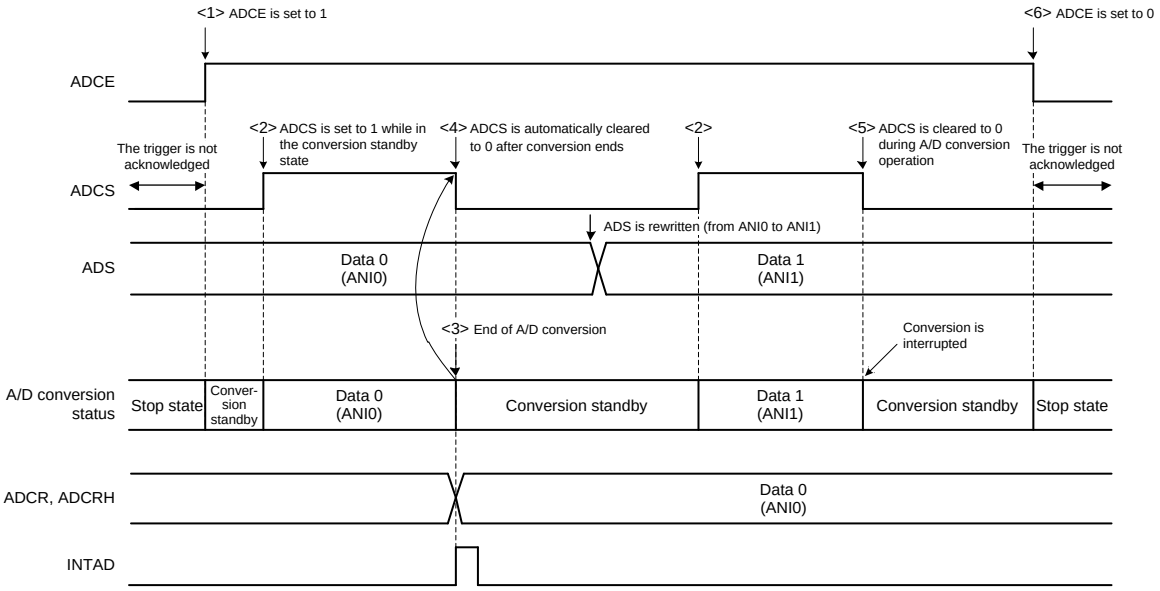
<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the system enters the conversion standby state.

<5> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the conversion standby state.

<6> When ADCE is cleared to 0 while in the conversion standby state, the A/D converter enters the stop state. Setting ADCS = 1 and ADCE = 0 is prohibited. Setting ADCS to 1 while conversion is stopped (ADCS = 0, ADCE = 0) is ignored and A/D conversion does not start.

Figure 11-13. Example of Operation Timing

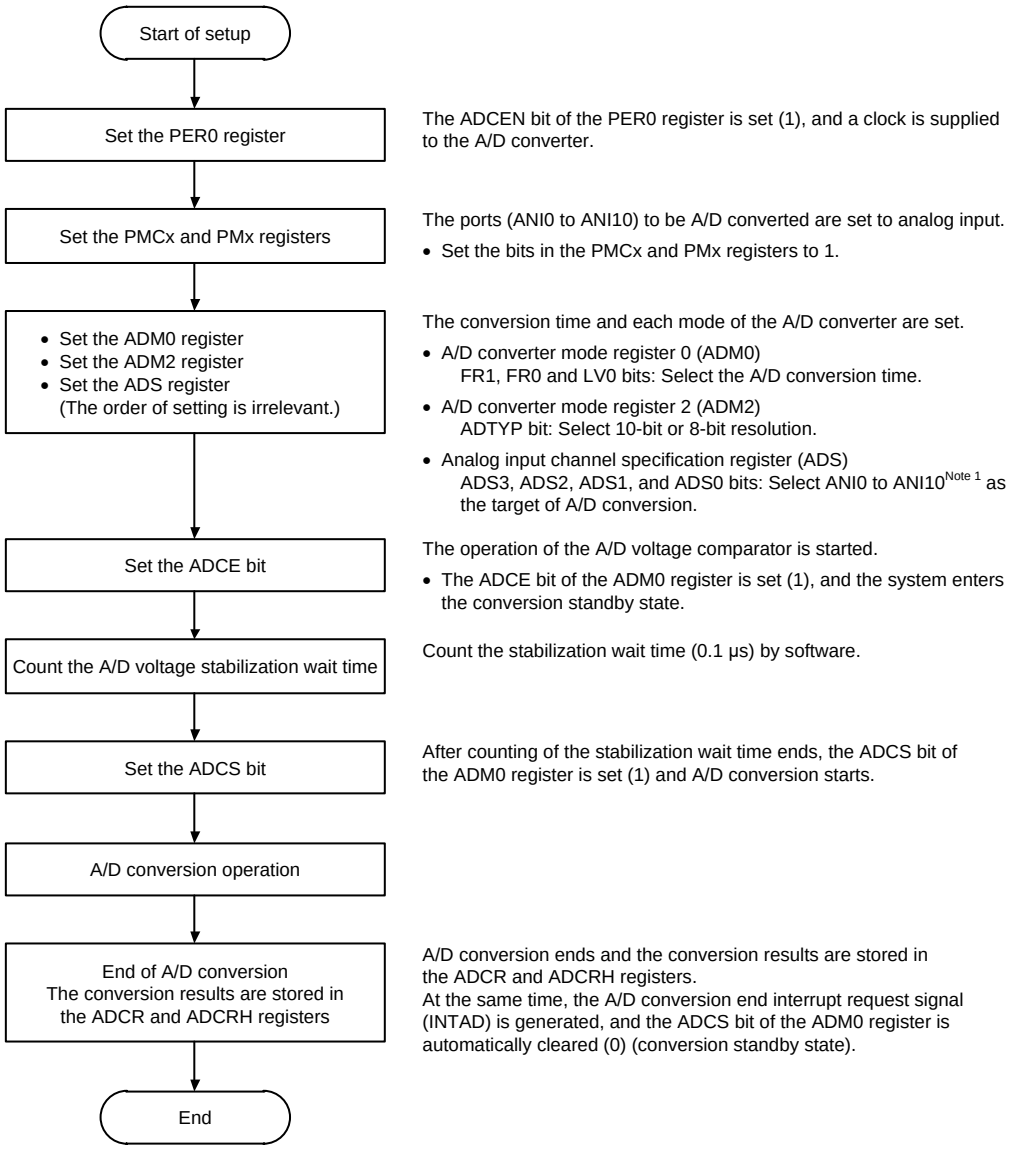


11.7 A/D Converter Setup Flowchart

The A/D converter setup flowchart is described below.

11.7.1 Setting up ANI0 to ANI10 for A/D conversion

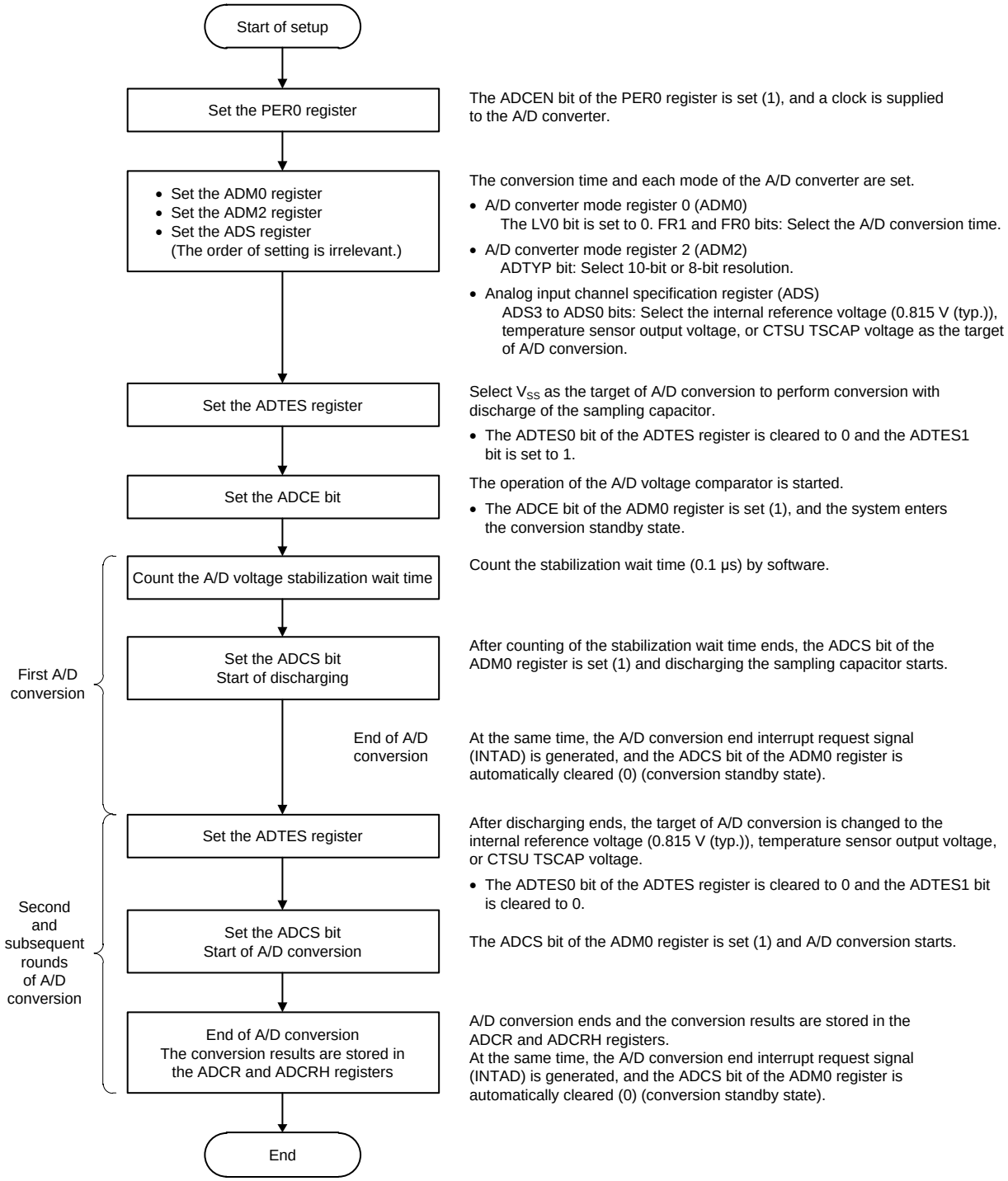
Figure 11-14. Setting up ANI0 to ANI10 for A/D Conversion



Note 1. ANI0 to ANI3 for 10-pin products,
ANI0 to ANI6 for 16-pin products

11.7.2 Setting up the internal reference voltage, temperature sensor output voltage, or CTSU TSCAP voltage for A/D conversion

Figure 11-15. Setting up Internal Reference Voltage, Temperature Sensor Output Voltage, or CTSU TSCAP Voltage for A/D Conversion



11.8 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

11.8.1 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 relative to the full scale is expressed by %FSR (Full Scale Range).

1LSB is as follows when the resolution is 10 bits.

$1LSB = 1/2^{10} = 1/1024$

$= 0.098\%FSR$

The accuracy is determined by overall error, regardless of the resolution.

11.8.2 Overall error

This shows the maximum error value between the actual measurement 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.

11.8.3 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 11-16. Overall Error

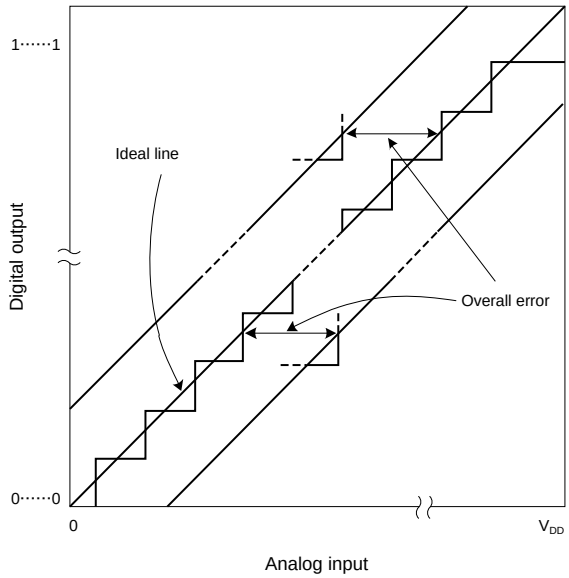
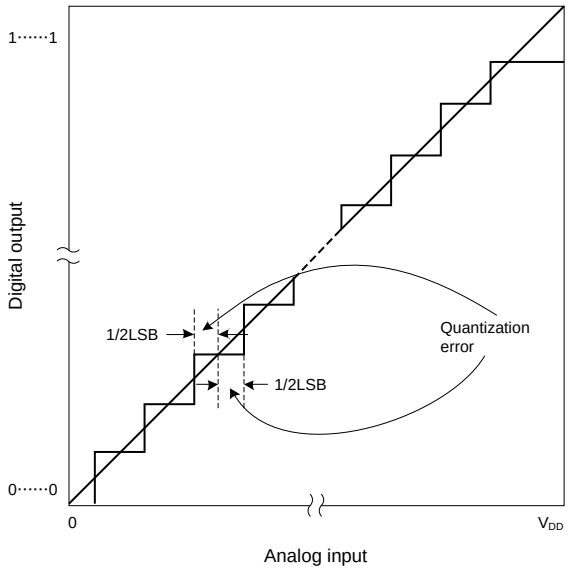


Figure 11-17. Quantization Error



11.8.4 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.

11.8.5 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.

11.8.6 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.

11.8.7 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 11-18. Zero-Scale Error

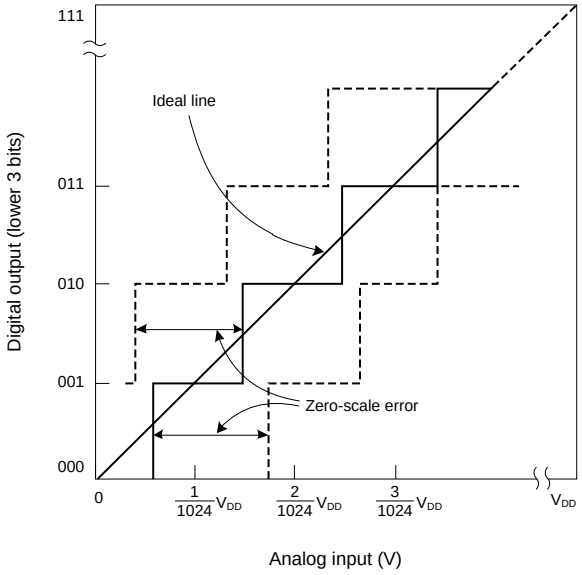


Figure 11-19. Full-Scale Error

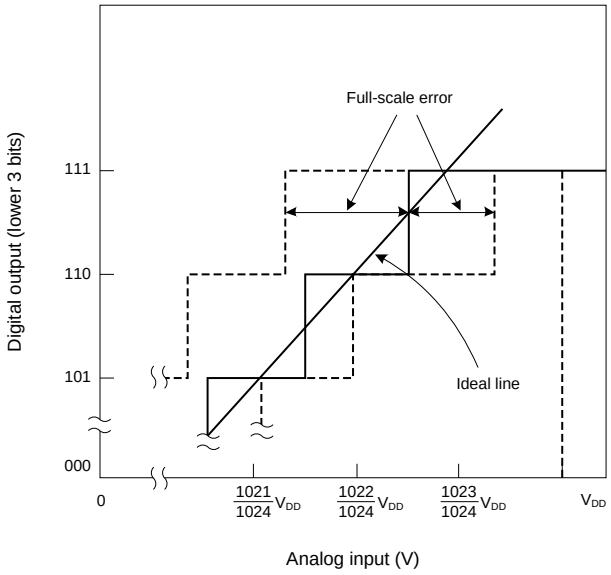


Figure 11-20. Integral Linearity Error

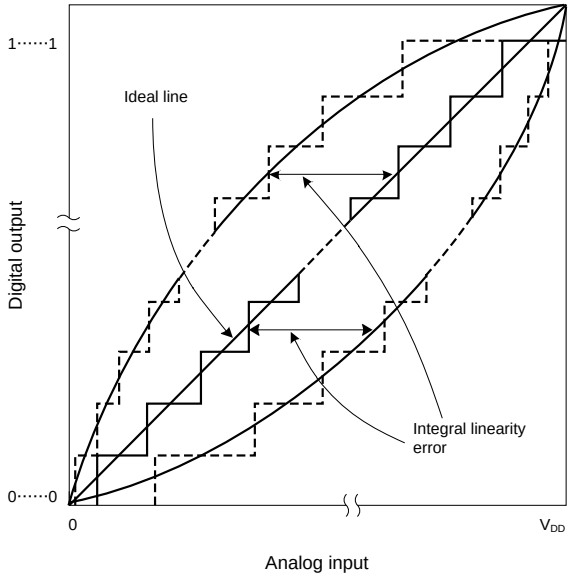
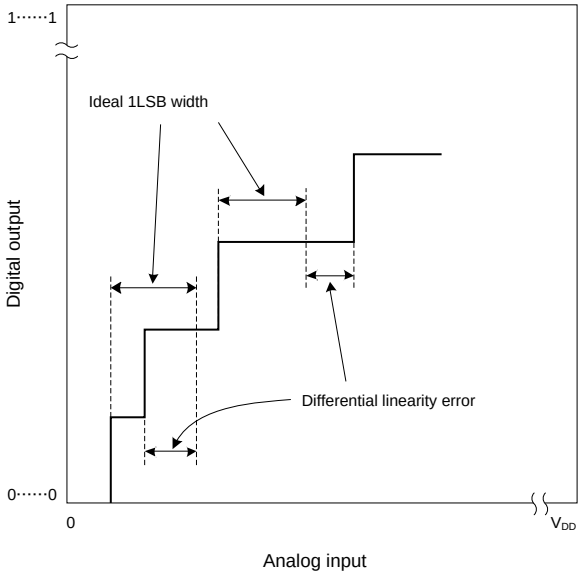


Figure 11-21. Differential Linearity Error



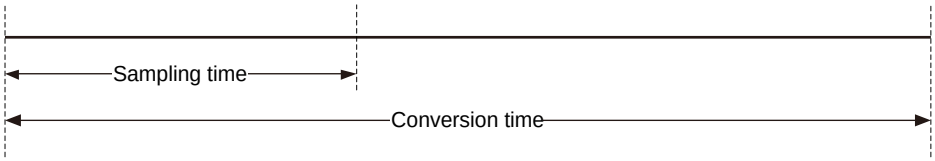
11.8.8 Conversion time

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.

11.8.9 Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.



11.9 Notes on A/D Converter

11.9.1 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.

11.9.2 Input voltage on ANI0 to ANI10 pins

Observe the rated range of the ANI0 to ANI10 pins input voltage. If a voltage exceeding V_{DD} or a voltage lower than V_{SS} (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.

11.9.3 Conflicting operations

(1) If writing to the A/D conversion result register (ADCR, ADCRH) at the end of conversion and reading of the ADCR or ADCRH register by software operation are in contention, the latter is given priority.
After the read operation, the new conversion result is written to the ADCR or ADCRH register.

(2) If writing to the ADCR or ADCRH register at the end of conversion and writing to A/D converter mode register 0 (ADM0) are in contention, the latter is given priority. Writing to the ADCR or ADCRH register is not performed, nor is the A/D conversion end interrupt signal (INTAD) generated.

11.9.4 Noise countermeasures

To maintain the 10-bit resolution, attention must be paid to noise on V_{DD} and the ANI0 to ANI10 pins.

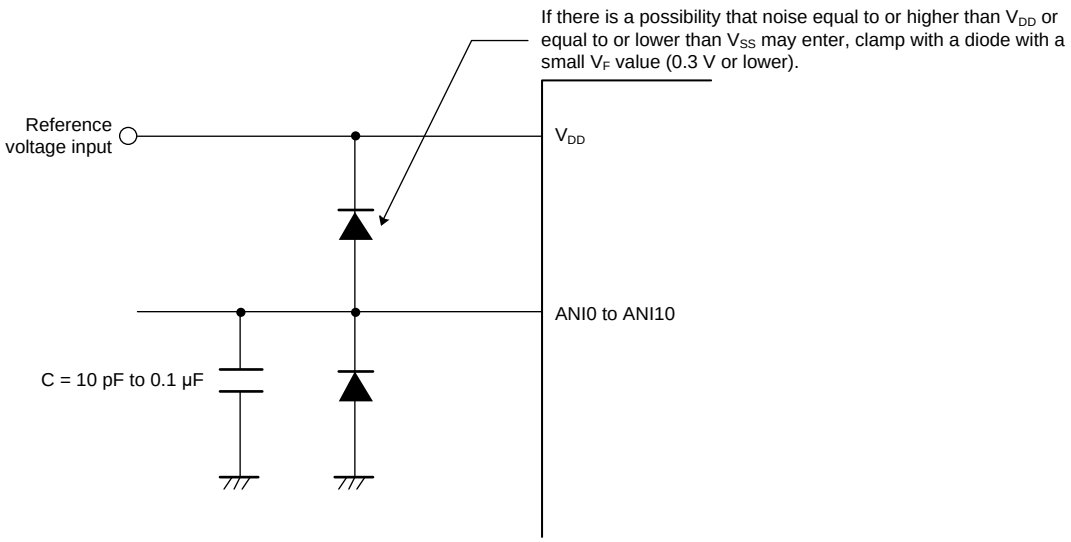
(1) Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.

(2) The higher the output impedance of the analog input source, the greater the influence. To reduce noise, we recommend connecting C externally as shown in **Figure 11-22**.

(3) Do not switch other pins during conversion.

(4) The accuracy is improved if the HALT mode is set immediately after the start of conversion.

Figure 11-22. Analog Input Pin Connection



11.9.5 Analog input (ANIn) pins

(1) The analog input pins (ANI0 to ANI10[Note 1]) are also used as input port pins (P01 to P07[Note 2] and P20 to P23[Note 2]). When A/D conversion is performed with any of the ANI0 to ANI10 pins selected, do not change output values to alternate port P01 to P07[Note 2] and P20 to P23[Note 2] while conversion is in progress; otherwise the conversion accuracy 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.

Note 1. ANI0 to ANI3 for 10-pin products,
ANI0 to ANI6 for 16-pin products

Note 2. P01 to P04 for 10-pin products,
P01 to P07 for 16-pin products

11.9.6 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, keep the output impedance of the analog input source at no greater than 1 kΩ. If the output impedance is high, we recommend connecting a capacitor of about 0.1 μF between the ANI0 to ANI10[Note 1] pins and the ground (see **Figure 11-22**).

Note 1. ANI0 to ANI3 for 10-pin products,
ANI0 to ANI6 for 16-pin products

11.9.7 Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the analog input channel specification register (ADS) is changed. When A/D conversion is to be stopped and then resumed, clear the ADIF flag before resuming the A/D conversion.

11.9.8 Conversion results just after A/D conversion start

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 0.1 μs after the ADCE bit was set to 1. Take measures such as polling the A/D conversion end interrupt request signal (INTAD) and removing the first conversion result.

11.9.9 A/D conversion result register (ADCR, ADCRH) read operation

When a write operation is performed to A/D converter mode register 0 (ADM0), the analog input channel specification register (ADS), and port mode control register (PMCx), the contents of the ADCR and ADCRH registers may become undefined and a correct conversion result may not be read. After conversion ends, read the conversion result before writing to the ADM0, ADS, or PMCx register.

11.9.10 Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 11-23. Internal Equivalent Circuit of ANIn Pin

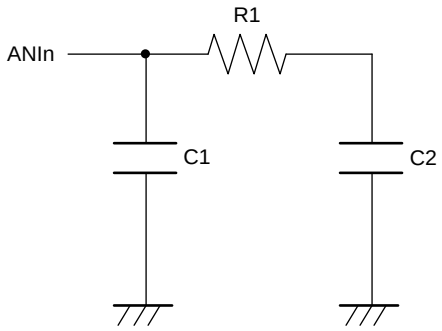


Table 11-4. Resistance and Capacitance Values of Equivalent Circuit

V_{DD}	Pin	R1 (kΩ)	C1 (pF)	C2 (pF)
2.4 V ≤ V_{DD} ≤ 5.5 V	ANI0 to ANI10[Note 1]	40	8	1.7
2.4 V ≤ V_{DD} < 2.7 V	ANI0 to ANI10[Note 1]	200		

Note 1. ANI0 to ANI3 for 10-pin products,
ANI0 to ANI6 for 16-pin products

Remark The resistance and capacitance values are not guaranteed values.

11.9.11 Starting the A/D converter

Start the A/D converter after the V_{DD} voltage stabilizes.

CHAPTER 12 COMPARATOR

Caution 10-pin products have only one comparator channel. 16-, 20-, 24-, and 32-pin products have two comparator channels. This chapter mainly describes the comparator in the case of 32-pin products.

12.1 Comparator Functions

The comparator has the following functions.

- The comparator response speed can be selected.
  High-speed mode: Decreased response delay time, with increased power consumption.
  Low-speed mode: Increased response delay time, with decreased power consumption.
- The comparator reference voltage is selectable as the externally input reference voltage or the internal reference voltage[Note 1] (0.815 V (typ.)).
- The integrated digital filter for noise elimination allows selecting the noise elimination width.
- The inverted/non-inverted comparator output can be output from the VCOUTn pin.
- An interrupt (INTCMPn) can be generated upon detection of the effective edge of the comparator output signal.

Note 1. The internal reference voltage cannot be used for the A/D converter and comparator simultaneously. When the internal reference voltage is selected as the reference voltage of the comparator, do not select the internal reference voltage as the target for conversion by the A/D converter.

Remark n = 0, 1

12.2 Comparator Configuration

The comparator consists of the following hardware:

1) IVCMPn Pin

The comparator analog input pin. An analog signal to be compared by the comparator is input to this pin.

Remark n = 0, 1

2) IVREFn Pin

An input pin to supply the reference voltage externally. The reference voltage of the comparator and analog input that is input to the IVCMPn pin are compared.

In addition to the voltage supplied to the IVREFn pin externally, the internal reference voltage (0.815 V (typ.)) can be selected for the comparator reference voltage.

For details on setting the comparator reference voltage, see **12.3.2 Comparator Mode Setting Register (COMPMDR)**.

Remark n = 0, 1

3) VCOUTn Pin

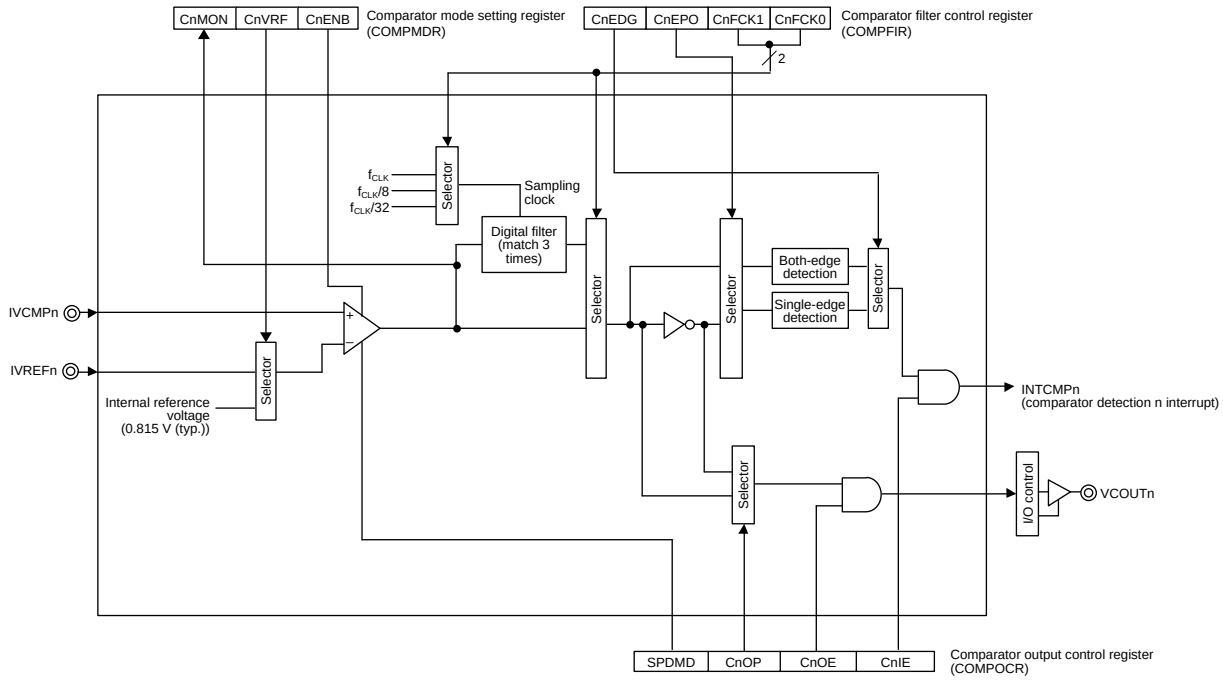
A pin to output the comparator comparison results. The inverted/non-inverted comparator output can be output from the VCOUTn pin.

For use of the VCOUTn pin as a comparator output, see **12.4.3 Comparator n Output (n = 0, 1)**.

Remark n = 0, 1

Figure 12-1 shows the block diagram of the comparator.

Figure 12-1. Block Diagram of the Comparator



Remark n = 0, 1

12.3 Registers Controlling the Comparator

The following lists the registers to control the comparator.

- Peripheral enable register 0 (PER0)
- Comparator mode setting register (COMPMDR)
- Comparator filter control register (COMPFIR)
- Comparator output control register (COMPOCR)
- Port mode control register 0 (PMC0)
- Port mode register 0 (PM0)
- Port register 0 (P0)
- Port mode register 4 (PM4)
- Port register 4 (P4)
- Port mode register 12 (PM12)
- Port register 12 (P12)

12.3.1 Peripheral Enable Register 0 (PER0)

This register enables or disables clock supply to 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 6 (CMPEN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 12-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PER0	TMKAEN	CMPEN	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

CMPEN	Control of comparator input clock
0	Stops input clock supply. • SFR used by the comparator cannot be written. • The comparator is in the reset status.
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 before setting the registers shown below. If CMPEN = 0, control registers of the comparator are cleared to their initial values and writing to them is ignored (except for port mode registers 0, 4, and 12 (PM0, PM4, PM12), port registers 0, 4, and 12 (P0, P4, P12), and port mode control register 0 (PMC0)).

- **Comparator mode setting register (COMPMDR)**
- **Comparator filter control register (COMPFIR)**
- **Comparator output control register (COMPOCR)**

Caution 2. Be sure to clear the following bits to 0.
10-pin and 16-pin products: Bits 1 and 3
20-pin, 24-pin, and 32-pin products: Bit 1

12.3.2 Comparator Mode Setting Register (COMPMDR)

This register selects the comparator reference voltage, starts/stops the comparison operation, and indicates the comparison result state.

The COMPMDR register can be set by a 1-bit or 8-bit memory manipulation instruction. Note that the CnMON bit (n = 1, 0) can only be read.

Reset signal generation clears this register to 00H.

Figure 12-3. Format of Comparator Mode Setting Register (COMPMDR)

Address: FFF60H After reset: 00H R/W[Note 1]

Symbol	7	6	5	4	3	2	1	0
COMPMDR	C1MON	C1VRF	0	C1ENB	C0MON	C0VRF	0	C0ENB

C1MON[Note 2]	Comparator 1 monitor flag
0	IVCMP1 < comparator 1 reference voltage
1	IVCMP1 > comparator 1 reference voltage

C1VRF	Comparator 1 reference voltage selection
0	Supplied from the IVREF1 pin
1	Supplied from the internal reference voltage (0.815 V (typ.))[Note 3]

C1ENB	Comparator 1 operation control
0	Comparator 1 operation disabled
1	Comparator 1 operation enabled

C0MON[Note 4]	Comparator 0 monitor flag
0	IVCMP0 < comparator 0 reference voltage
1	IVCMP0 > comparator 0 reference voltage

C0VRF	Comparator 0 reference voltage selection
0	Supplied from the IVREF0 pin
1	Supplied from the internal reference voltage (0.815 V (typ.))[Note 5]

C0ENB	Comparator 0 operation control
0	Comparator 0 operation disabled
1	Comparator 0 operation enabled

Note 1. Bits 3 and 7 are read-only bits.

Note 2. After the comparator 1 operation is enabled (C1ENB = 1), the IVREF1 pin state can be read from the C1MON bit setting. When the comparator 1 operation is then disabled (C1ENB = 0), the C1MON bit value is undefined.

Note 3. When the internal reference voltage (0.815 V (typ.)) is selected as the comparator 1 reference voltage, the internal reference voltage cannot be selected for the A/D converter.

Note 4. After the comparator 0 operation is enabled (C0ENB = 1), the IVREF0 pin state can be read from the C0MON bit setting. When the comparator 0 operation is then disabled (C0ENB = 0), the C0MON bit value is undefined.

Note 5. When the internal reference voltage (0.815 V (typ.)) is selected as the comparator 0 reference voltage, the internal reference voltage cannot be selected for the A/D converter.

12.3.3 Comparator Filter Control Register (COMPFIR)

This register selects the effective edge for the comparator interrupt signal, and enables or disables the digital filter.

If noise elimination is required, set the CnFCK1 and CnFCK0 bits (n = 1, 0) so that the digital filter is enabled. When the digital filter is enabled, the comparator output is checked if its level remains the same for three consecutive digital filter sampling clock cycles.

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

Reset signal generation clears this register to 00H.

Figure 12-4. Format of Comparator Filter Control Register (COMPFIR)

Address: FFF61H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
COMPFIR	C1EDG	C1EPO	C1FCK1	C1FCK0	C0EDG	C0EPO	C0FCK1	C0FCK0

C1EDG	C1EPO	Effective edge selection for comparator 1 interrupt signal[Note 1]
0	0	Rising edge
0	1	Falling edge
1	×	Both rising and falling edges

C1FCK1	C1FCK0	Comparator 1 digital filter enable/disable[Note 1, Note 2, Note 3]
0	0	Digital filter disabled
0	1	Digital filter enabled, sampling clock: f_{CLK}
1	0	Digital filter enabled, sampling clock: $f_{CLK}/8$
1	1	Digital filter enabled, sampling clock: $f_{CLK}/32$

C0EDG	C0EPO	Effective edge selection for comparator 0 interrupt signal[Note 4]
0	0	Rising edge
0	1	Falling edge
1	×	Both rising and falling edges

C0FCK1	C0FCK0	Comparator 0 digital filter enable/disable[Note 4, Note 5, Note 6]
0	0	Digital filter disabled
0	1	Digital filter enabled, sampling clock: f_{CLK}
1	0	Digital filter enabled, sampling clock: $f_{CLK}/8$
1	1	Digital filter enabled, sampling clock: $f_{CLK}/32$

Note 1. If the C1EDG, C1EPO, and C1FCK1 or C1FCK0 bits are changed while operation of the comparator 1 is enabled, a comparator detection 1 interrupt (INTCMP1) may be generated. Change these bits only after clearing the C1IE bit in the COMPOCR register to 0 to disable an interrupt request.
Also, be sure to clear bit 2 (CMPIF1) in the interrupt request flag register 1H (IF1H) to 0 after changing these bits.

Note 2. If the value of the C1FCK1 or C1FCK0 bit is changed, a wait period of four cycles of the sampling clock is required to update the digital filter. To use the comparator detection 1 interrupt (INTCMP1), set the C1IE bit in the COMPOCR register to 1 after this wait period.

Note 3. To use the comparator in STOP mode, disable the digital filter (C1FCK1 and C1FCK0 = 00B).

Note 4. If the C0EDG, C0EPO, and C0FCK1 or C0FCK0 bits are changed while operation of the comparator 0 is enabled, a comparator detection 0 interrupt (INTCMP0) may be generated. Change these bits only after clearing the C0IE bit in the COMPOCR register to 0 to disable an interrupt request.
Also, be sure to clear bit 1 (CMPIF0) in the interrupt request flag register 1H (IF1H) to 0 after changing these bits.

Note 5. If the value of the C0FCK1 or C0FCK0 bit is changed, a wait period of four cycles of the sampling clock is required to update the digital filter. To use the comparator detection 0 interrupt (INTCMP0), set the C0IE bit in the COMPOCR register to 1 after this wait period.

Note 6. To use the comparator in STOP mode, disable the digital filter (C0FCK1 and C0FCK0 = 00B).

Remark ×: Don’t care

12.3.4 Comparator Output Control Register (COMPOCR)

This register selects the comparator response speed, controls the VCOUTn output, and enables or disables the interrupt request signal.

Remark n = 0, 1

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

Reset signal generation clears this register to 00H.

Figure 12-5. Format of Comparator Output Control Register (COMPOCR)

Address: FFF62H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
COMPOCR	SPDMD	C1OP	C1OE	C1IE	0	C0OP	C0OE	C0IE

SPDMD[Note 1]	Comparator speed selection
0	Low-speed mode
1	High-speed mode

C1OP	VCOUT1 output polarity selection
0	Non-inverted comparator 1 output is output from the VCOUT1 pin.
1	Inverted comparator 1 output is output from the VCOUT1 pin.

C1OE	VCOUT1 pin output enable/disable
0	Comparator 1 VCOUT1 pin output disabled
1	Comparator 1 VCOUT1 pin output enabled

C1IE	Comparator 1 interrupt request enable/disable
0	Comparator 1 interrupt request disabled
1	Comparator 1 interrupt request enabled

C0OP	VCOUT0 output polarity selection
0	Non-inverted comparator 0 output is output from the VCOUT0 pin.
1	Inverted comparator 0 output is output from the VCOUT0 pin.

C0OE	VCOUT0 pin output enable/disable
0	Comparator 0 VCOUT0 pin output disabled
1	Comparator 0 VCOUT0 pin output enabled

C0IE	Comparator 0 interrupt request enable/disable
0	Comparator 0 interrupt request disabled
1	Comparator 0 interrupt request enabled

Note 1. When rewriting the SPDMD bit, be sure to clear the CnENB bit (n = 0, 1) in the COMPMDR register to 0 in advance.

12.3.5 Registers Controlling Port Functions of Comparator I/O Pins

The port mode register 0 (PM0), port mode register 4 (PM4), port mode register 12 (PM12), port register 0 (P0), port register 4 (P4), port register 12 (P12), and port mode control register 0 (PMC0) should be appropriately set to control the functions of the port pins that are also used for input and output of the comparator. For details, see **4.3.1 Port mode registers 0, 1, 2, 4, 6, 12 (PM0, PM1, PM2, PM4, PM6, PM12)**, **4.3.2 Port registers 0, 1, 2, 4, 6, 12, 13 (P0, P1, P2, P4, P6, P12, P13)**, and **4.3.5 Port mode control registers 0, 2 (PMC0, PMC2)**. For an example of settings when using a port pin for input and output of the comparator, see **4.5.3 Register setting examples for used port and alternate functions**.

When using the IVCMP0 and IVREF0 pins as analog inputs of the comparator, set the corresponding bits in the port mode register (PM0) and port mode control register (PMC0) to 1.

When using the VCOUT0 pin as a comparator output, set the bits in the port mode register (PM0), port register (P0), and port mode control register (PMC0) to 0. For details on the VCOUT0 pin output setting, follow the setting procedure in **12.4.3 Comparator n Output (n = 0, 1)**.

When using the IVCMP1 and IVREF1 pins as analog inputs of the comparator, set the corresponding bits in the port mode register (PM0) and port mode control register (PMC0) to 1.

When using the VCOUT1 pin as a comparator output, set the bits in the port mode register (PM0), port register (P0), and port mode control register (PMC0) to 0. For details on the VCOUT1 pin output setting, follow the setting procedure in **12.4.3 Comparator n Output (n = 0, 1)**.

12.4 Comparator n Operation (n = 0, 1)

The CnMON bit in the COMPMDR register is set to 1 when the analog input voltage on the IVCMPn (n = 0, 1) pin is higher than the reference voltage. When lower, the CnMON bit is set to 0.

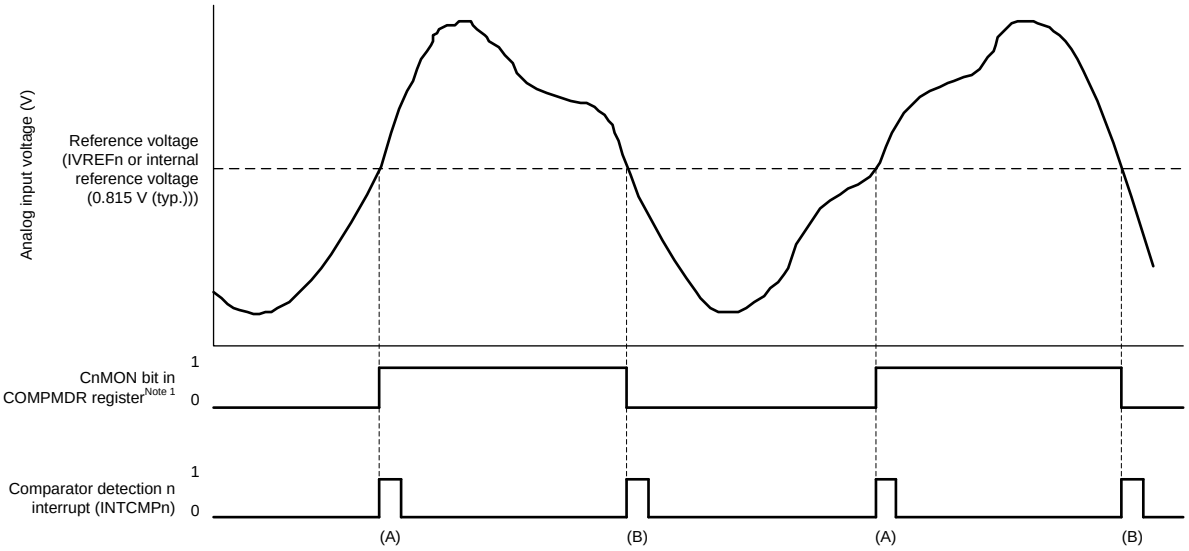
When using the comparator detection n interrupt (INTCMPn), set the CnIE bit in the COMPOCR register to 1 (interrupt request enabled). If the comparison result changes at this time, a comparator n interrupt request signal is generated. For details on the comparator 0 interrupt, refer to **12.4.2 Comparator n Interrupt Operation (n = 0, 1)**.

Remark n = 0, 1

Figure 12-6 shows an example of the comparator n operation (no digital filter (CnFCK1 and CnFCK0 in COMPFIR = 00B), both-edge detection on an interrupt (CnEDG = 1).

Remark n = 0, 1

Figure 12-6. Example of Comparator n (n = 0, 1) Operation (No Digital Filter, Both-Edge Detection on Interrupt)



Note 1. The output delay time depends on the comparator operating mode. For details, see **26.6.2 Comparator characteristics** and **27.6.2 Comparator characteristics**.

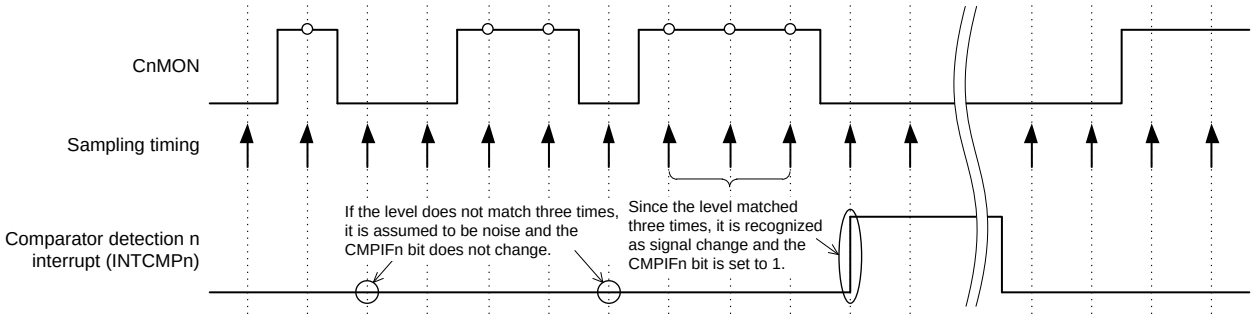
Caution When the rising edge is specified as an effective edge for an interrupt (CnEDG = 0 and CnEPO = 0), INTCMPn only changes at (A).
When the falling edge is specified as an effective edge for an interrupt (CnEDG = 0 and CnEPO = 1), INTCMPn only changes at (B).

12.4.1 Comparator n Digital Filter Operation (n = 0, 1)

Comparator n incorporates a digital filter. The sampling clock is selected by the CnFCK1 and CnFCK0 bits in the COMPFIR register. The comparator n output signal is sampled every sampling clock, and when the level of the output signal matches three times, the digital filter output changes at the next sampling clock.

Figure 12-7 shows the comparator n (n = 0, 1) digital filter and interrupt operation example.

Figure 12-7. Comparator n (n = 0, 1) Digital Filter and Interrupt Operation Example



Remark The operation example in **Figure 12-7** applies when the digital filter is enabled (the CnFCK1 and CnFCK0 bits in the COMPFIR register = 01B, 10B, or 11B).

12.4.2 Comparator n Interrupt Operation (n = 0, 1)

When using the comparator n interrupt, set the CnIE bit in the COMPOCR register to 1 (interrupt request enabled). The condition for interrupt request generation can be set by the COMPFIR register. The comparator outputs can also be passed through the digital filter.

For details on the register settings, refer to **12.3.3 Comparator Filter Control Register (COMPFIR)** and **12.3.4 Comparator Output Control Register (COMPOCR)**.

12.4.3 Comparator n Output (n = 0, 1)

The comparison result from the comparator can be output from the VCOUTn pin. The CnOP and CnOE bits in the COMPOCR register are used to set the output polarity (inverted or non-inverted output) of the VCOUTn pin and enable or disable the VCOUTn pin output, respectively. For details on the register settings, refer to **12.3.4 Comparator Output Control Register (COMPOCR)**.

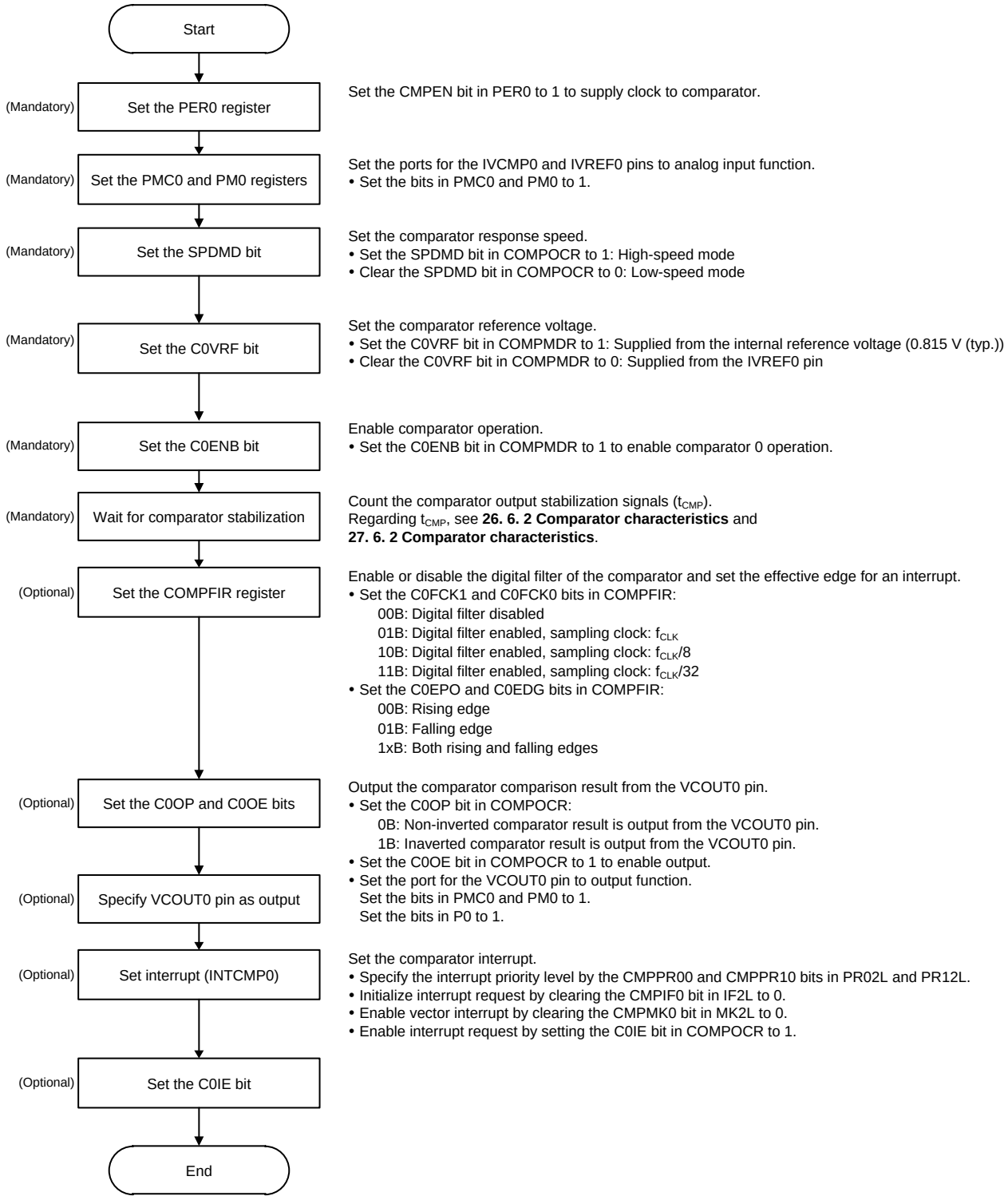
To output the comparator comparison result from the VCOUTn pin, follow the procedure shown in **Figure 12-8 Procedure for Enabling Comparator Operation**.

12.5 Comparator Setting Flowcharts

The following shows the comparator setting flowcharts.

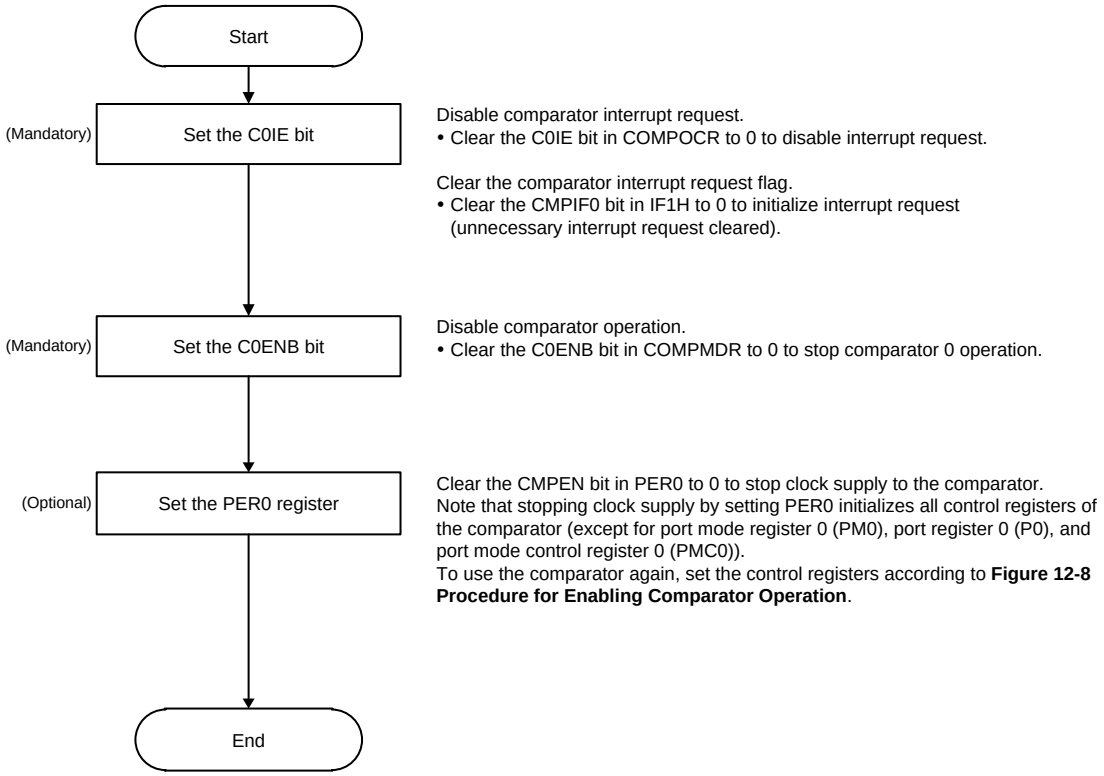
12.5.1 Enabling Comparator Operation (in the Case of CMP0)

Figure 12-8. Procedure for Enabling Comparator Operation



12.5.2 Disabling Comparator Operation (in the Case of CMP0)

Figure 12-9. Procedure for Disabling Comparator Operation



CHAPTER 13 SERIAL ARRAY UNIT

A single serial array unit has up to four serial channels. Each channel can achieve 3-wire serial (simplified SPI or CSI[Note 1]), UART, and simplified I²C communication.

Function assignment of each channel supported by the RL78/G16 is as shown below.

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.

● 10-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	—	—
	3	—	—	—

● 16-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11

● 20-, 24-, and 32-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	—		—

When “UART0” is used for channels 0 and 1 of the unit 0, CSI00 cannot be used, but CSI11, UART1, or IIC11 can be used for channels 2 and 3.

Caution Most of the following descriptions in this section use the units and channels of the 32-pin products as an example.

13.1 Functions of Serial Array Unit

Each serial interface supported by the RL78/G16 has the following features.

13.1.1 Simplified SPI (CSI00, CSI11, CSI20)

Data is transmitted or received in synchronization with the serial clock (SCK) output from the master.

3-wire serial communication is clocked communication performed by using three communication lines: one for the serial clock (SCK), one for transmitting serial data (SO), one for receiving serial data (SI).

For details about the settings, see **13.5 Operation of Simplified SPI (CSI00, CSI11, CSI20)** 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 1]
  During master communication: Max. $f_{CLK}/4$
  During slave communication: Max. $f_{MCK}/6$

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

Note 1. Set up the transfer rate within a range satisfying the SCK cycle time (t_{KCY}). For details, see **CHAPTER 26 ELECTRICAL SPECIFICATIONS (T_A = −40 to +85°C)** and **CHAPTER 27 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C, T_A = −40 to +125°C)**.

13.1.2 UART (UART0 to UART2)

This is a start-stop synchronization communication function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel).

For details about the settings, see **13.6 Operation of UART (UART0 to UART02) Communication**.

[Data transmission/reception]

- Data length of 7, 8, or 9 bits[Note 1]
- MSB/LSB first selectable
- Level setting of transmit/receive data and select of reverse
- Parity bit appending and parity check functions
- Stop bit appending

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

Note 1. Only UART0 can be specified for the 9-bit data length.

The ISC register can be used to set up the input signal on the RxD0 pin of UART0 as an external interrupt input or as a timer input for the timer array unit. The input pulse interval measurement mode of the timer array unit can then be used to measure the width at the baud rate of the other party in communications and make the required adjustments in response.

13.1.3 Simplified I²C (IIC00, IIC11, IIC20)

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This simplified I²C is designed for single communication with a device such as EEPROM, flash memory, or A/D converter, and therefore, it functions only as a master.

Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.

For details about the settings, see **13.7 Operation of Simplified I2C (IIC00, IIC11, IIC20) Communication**.

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function[Note 1] and ACK detection function
- Data length of 8 bits
  (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
- Manual generation of start condition and stop condition

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- ACK error, or overrun error

***[Functions not supported by simplified I²C]**

- Slave transmission, slave reception
- Arbitration loss detection function
- Wait detection functions

Note 1. When receiving the last data, 0 is written to the SOEmn bit of the serial output enable register m (SOEm) and serial communication data output is stopped, disabling ACK output. See **13.7.3(2) Processing flow**.

Remark 1. To use an I²C bus of full function, see **CHAPTER 14 SERIAL INTERFACE IICA**.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2, 3), mn = 00, 02, 03, 10

13.2 Configuration of Serial Array Unit

The serial array unit includes the following registers, and input and output pins.

Table 13-1. Configuration of Serial Array Unit

Item	Configuration
Shift register	8 or 9 bits[Note 1]
Buffer register	Lower 8 bits or 9 bits of serial data register mn (SDRmn)[Note 1, Note 2]
Serial clock I/O	SCK00, SCK11, SCK20 pins (for simplified SPI), SCL00, SCL00, SCL11, SCL20 pins (for simplified I^2C)
Serial data input	SI00, SI11, SI20 pins (for simplified SPI), RxD0, RxD1, RxD2 pins (for UART)
Serial data output	SO00, SO11, SO20 pins (for simplified SPI), TxD0, TxD1, TxD2 pins (for UART)
Serial data I/O	SDA00, SDA11, SDA20 pins (for simplified I^2C)
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Serial clock select register m (SPSm) • Serial channel enable status register m (SEm) • Serial channel start register m (SSm) • Serial channel stop register m (STm) • Serial output enable register m (SOEm) • Serial output register m (SOm) • Serial output level register m (SOLm) • 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 output mode registers 0, 1, 2, 4 (POM0, POM1, POM2, POM4) • Port mode control registers 0, 2 (PMC0, PMC2) • Port mode registers 0, 1, 2, 4, 6 (PM0, PM1, PM2, PM4, PM6) • Port registers 0, 1, 2, 4, 6 (P0, P1, P2, P4, P6)

Note 1. The number of bits used as the shift register and buffer register differs depending on the unit and channel.

- mn = 00, 01: lower 9 bits
- Other than above: lower 8 bits

Note 2. The lower 8 bits of serial data register mn (SDRmn) can be read or written as the following SFR, depending on the communication mode.

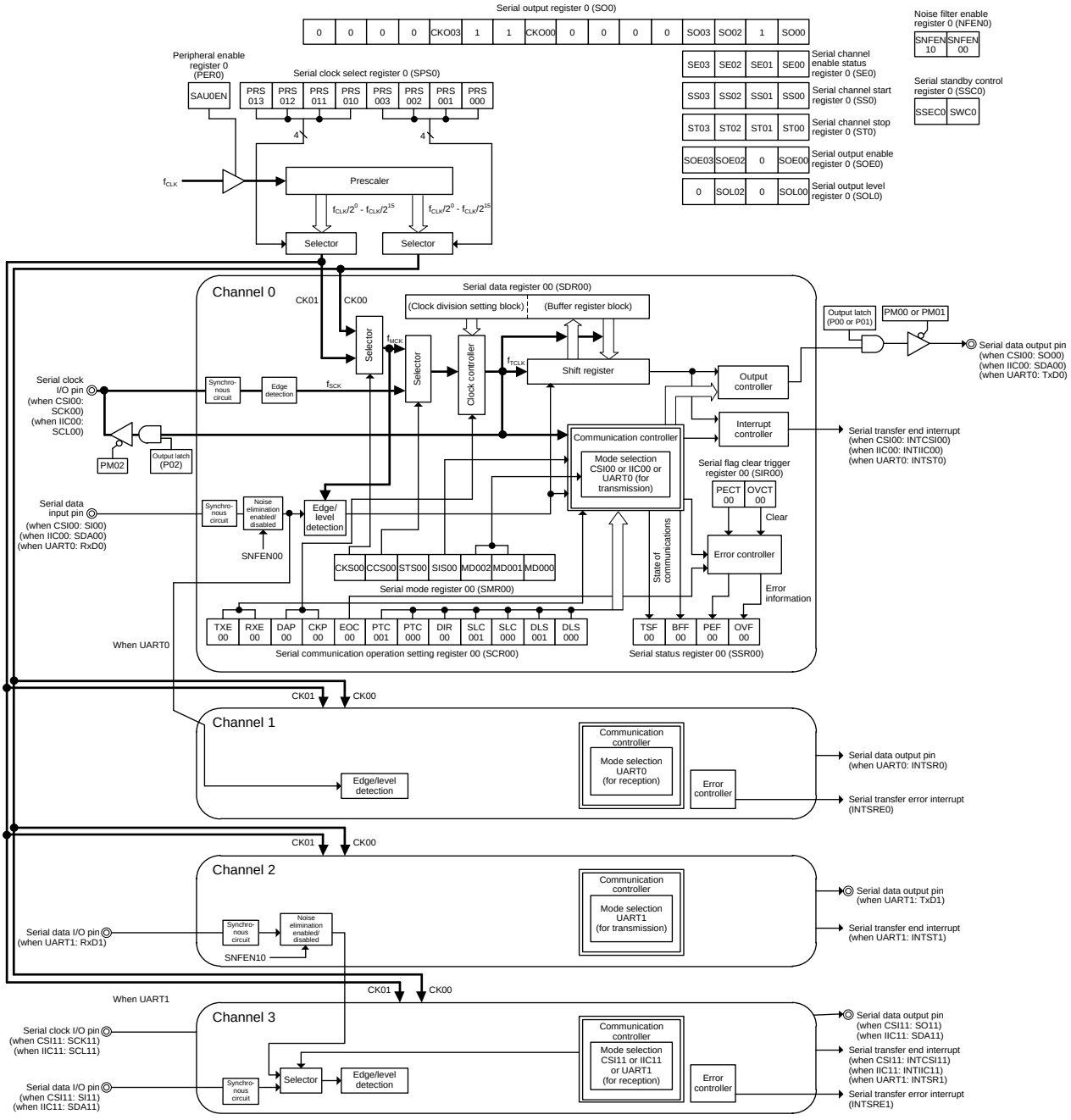
- CSIp communication … SIOp (CSIp data register)
- UARTq reception … RXDq (UARTq receive data register)
- UARTq transmission … TXDq (UARTq transmit data register)

- IICr communication … SIOr (IICr data register)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 11, 20)
q: UART number (q = 0 to 2), r: IIC number (r = 00, 11, 20)

Figure 13-1 shows the block diagram of serial array unit 0.

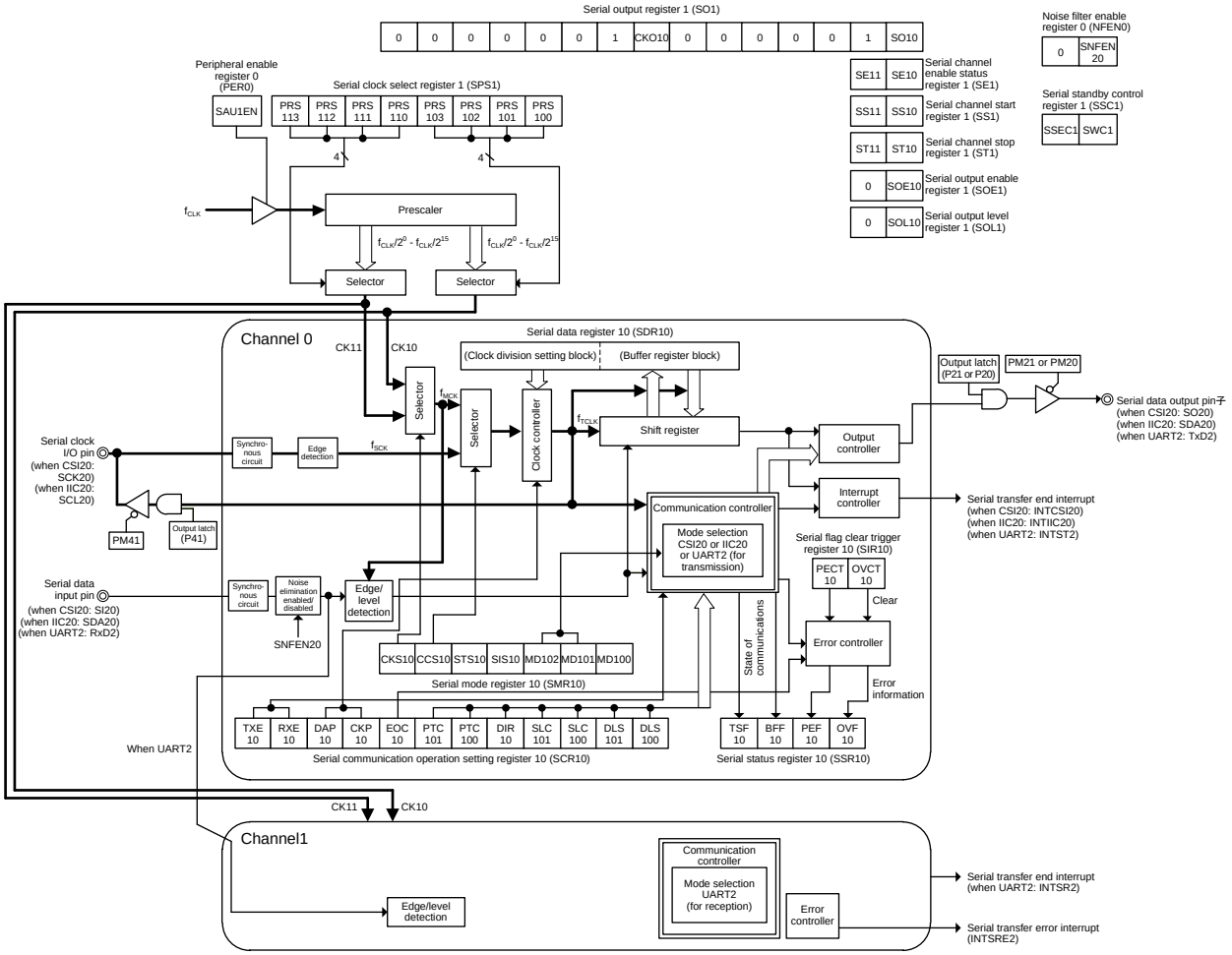
Figure 13-1. Block Diagram of Serial Array Unit 0



Remark The serial pins in the above diagram are those when PIOR21, PIOR20 = 00B in 32-pin products.

Figure 13-2 shows the block diagram of serial array unit 1.

Figure 13-2. Block Diagram of Serial Array Unit 1



Remark The serial pins in the above diagram are those when PIOR26, PIOR25 = 00B in 32-pin products.

13.2.1 Shift Register

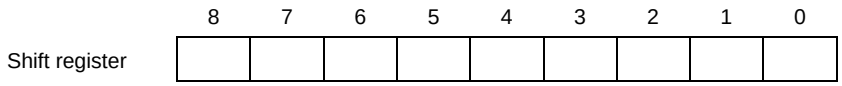
This is a 9-bit register that converts parallel data into serial data or vice versa.

In case of the UART communication of nine bits of data, nine bits (bits 0 to 8) are used.[Note 1]

During reception, it converts data input to the serial pin into parallel data. When data is transmitted, the value set to this register is output as serial data from the serial output pin.

The shift register cannot be directly manipulated by program.

To read or write the shift register, use the lower 8 or 9 bits of serial data register mn (SDRmn).



Note 1. Only UART0 can be specified for the 9-bit data length.

13.2.2 Lower 8 or 9 bits of the serial data register mn (SDRmn)

The SDRmn is the transmit/receive data register (16 bits) of channel n.

Bits 8 to 0 (lower 9 bits)[Note 1] or bits 7 to 0 (lower 8 bits) function as a transmit/receive buffer register, and bits 15 to 9 are used as a register that sets the division ratio of the operation clock (f_{MCK}).

When data is received, parallel data converted by the shift register is stored in the lower 8 or 9 bits. When data is to be transmitted, set transmit data to be transferred to the shift register to the lower 8 or 9 bits.

The data stored in the lower 8 or 9 bits of this register is as follows, depending on the setting of bits 0 and 1 (DLSmn0, DLSmn1) of serial communication operation setting register mn (SCRmn), regardless of the output sequence of the data.

- 7-bit data length (stored in bits 0 to 6 of the SDRmn register)
- 8-bit data length (stored in bits 0 to 7 of the SDRmn register)
- 9-bit data length (stored in bits 0 to 8 of the SDRmn register)[Note 1]

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

The lower 8 or 9 bits of the SDRmn register can be read or written[Note 2] as the following SFR, depending on the communication mode.

- CSIp communication … SIOp (CSIp data register)
- UARTq reception … RXDq (UARTq receive data register)
- UARTq transmission … TXDq (UARTq transmit data register)
- IICr communication … SIOr (IICr data register)

The value of each SDRmn register is 0000H following a reset.

Note 1. Only UART0 can be specified for the 9-bit data length.

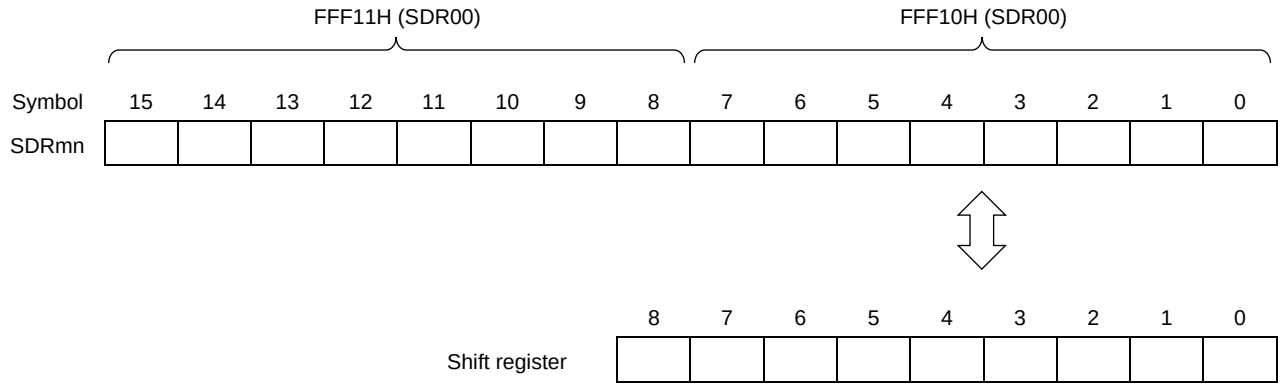
Note 2. When operation is stopped (SEmn = 0), do not rewrite SDRmn[7:0] by an 8-bit memory manipulation instruction (SDRmn[15:9] are all cleared to 0).

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, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 11, 20)
q: UART number (q = 0 to 2), r: IIC number (r = 00, 11, 20)

Figure 13-3. Format of Serial Data Register mn (SDRmn) (mn = 00, 01)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01) After reset: 0000H R/W

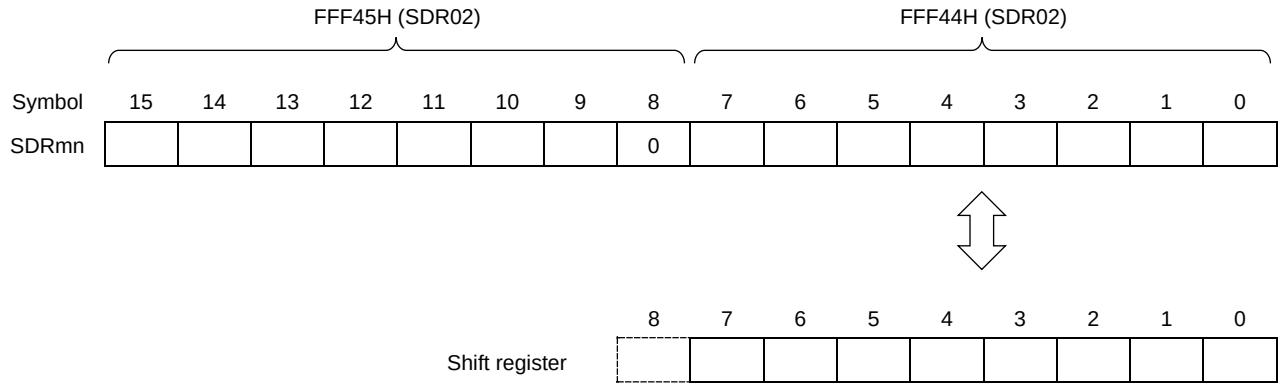


Remark For the function of the higher 7 bits of the SDRmn register, see **13.3 Registers to Control the Serial Array Unit**.

Figure 13-4. Format of Serial Data Register mn (SDRmn) (mn = 02, 03, 10, 11)

Address: FFF44H, FFF45H (SDR02)[Note 1], FFF46H, FFF47H (SDR03)[Note 1], FFF48H, FFF49H (SDR10)[Note 2], FFF4AH, FFF4BH (SDR11)[Note 2]

After reset: 0000H R/W



Note 1. 16- to 32-pin products

Note 2. 20- to 32-pin products

Caution Be sure to set bit 8 to 0.

Remark For the function of the higher 7 bits of the SDRmn register, see **13.3 Registers to Control the Serial Array Unit**.

13.3 Registers to Control the Serial Array Unit

The following registers are used to control the serial array unit.

- Peripheral enable register 0 (PER0)
- Serial clock select register m (SPSm)
- Serial mode register mn (SMRmn)
- Serial communication operation setting register mn (SCRmn)
- Serial data register mn (SDRmn)
- Serial flag clear trigger register mn (SIRmn)
- Serial status register mn (SSRmn)
- Serial channel start register m (SSm)
- Serial channel stop register m (STm)
- Serial channel enable status register m (SEm)
- Serial output enable register m (SOEm)
- Serial output level register m (SOLm)
- Serial output register m (SOm)
- Input switch control register (ISC)
- Noise filter enable register 0 (NFEN0)
- Port output mode registers 0, 1, 2, 4 (POM0, POM1, POM2, POM4)
- Port mode control registers 0, 2 (PMC0, PMC2)
- Port mode registers 0, 1, 2, 4, 6 (PM0, PM1, PM2, PM4, PM6)
- Port registers 0, 1, 2, 4, 6 (P0, P1, P2, P4, P6)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

13.3.1 Peripheral enable register 0 (PER0)

The PER0 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise.

If serial array unit 0 is to be used, be sure to set bit 2 (SAU0EN) of this register to 1.

If serial array unit 1 is to be used, be sure to set bit 3 (SAU1EN) of this register to 1.

The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of the PER0 register is 00H following a reset.

Figure 13-5. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PER0	TMKAEN	CMPEN	ADCEN	IICA0EN	SAU1EN[Note 1]	SAU0EN	0	TAU0EN

SAUmEN	Control of supply of an input clock to serial array unit m
0	Stops supply of an input clock. • The SFRs used by serial array unit m cannot be written. • Serial array unit m is in the reset state.
1	Enables supply of an input clock. • The SFRs used by serial array unit m can be read and written.

Note 1. 20- to 32-pin products only

Caution 1. When setting serial array unit m, make sure that the setting of the SAUmEN bit is 1 before setting the following registers. If SAUmEN = 0, the values of the registers which control serial array unit m are cleared to their initial values, and writing to them is ignored (except for the input switch control register (ISC), noise filter enable register 0 (NFEN0), port output mode registers 0, 1, 2, 4 (POM0, POM1, POM2, POM4), port mode control registers 0, 2 (PMC0, PMC2), port mode registers 0, 1, 2, 4, 6 (PM0, PM1, PM2, PM4, PM6), and port registers 0, 1, 2, 4, 6 (P0, P1, P2, P4, P6)).

- **Serial clock select register m (SPSm)**
- **Serial mode register mn (SMRmn)**
- **Serial communication operation setting register mn (SCRmn)**
- **Serial data register mn (SDRmn)**
- **Serial flag clear trigger register mn (SIRmn)**
- **Serial status register mn (SSRmn)**
- **Serial channel start register m (SSm)**
- **Serial channel stop register m (STm)**
- **Serial channel enable status register m (SEm)**
- **Serial output enable register m (SOEm)**
- **Serial output level register m (SOLm)**
- **Serial output register m (SOm)**

Caution 2. Be sure to clear the following bits to 0.
10- and 16-pin products: Bit 1, 3
20- to 32-pin products: Bit 1

13.3.2 Serial clock select register m (SPSm)

The SPSm 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.

The value of each SPSm register is 0000H following a reset.

Figure 13-6. Format of Serial Clock Select Register m (SPSm)

Address: F0126H, F0127H (SPS0), F0166H, F0167H (SPS1)[Note 1] 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	PRSm13	PRSm12	PRSm11	PRSm10	PRSm03	PRSm02	PRSm01	PRSm00

PRSmk3	PRSmk2	PRSmk1	PRSmk0	Selection of operation clock (CKmk)[Note 2]					
					f_{CLK} (MHz)				
					1	2	4	8	16
0	0	0	0	f_{CLK}	1 MHz	2 MHz	4 MHz	8 MHz	16 MHz
0	0	0	1	$f_{CLK}/2$	500 kHz	1 MHz	2 MHz	4 MHz	8 MHz
0	0	1	0	$f_{CLK}/2^2$	250 kHz	500 kHz	1 MHz	2 MHz	4 MHz
0	0	1	1	$f_{CLK}/2^3$	125 kHz	250 kHz	500 MHz	1 MHz	2 MHz
0	1	0	0	$f_{CLK}/2^4$	62.5 kHz	125 kHz	250 kHz	500 kHz	1 MHz
0	1	0	1	$f_{CLK}/2^5$	31.25 kHz	62.5 kHz	125 kHz	250 kHz	500 kHz
0	1	1	0	$f_{CLK}/2^6$	15.63 kHz	31.25 kHz	62.5 kHz	125 kHz	250 kHz
0	1	1	1	$f_{CLK}/2^7$	7.81 kHz	15.63 kHz	31.25 kHz	62.5 kHz	125 kHz
1	0	0	0	$f_{CLK}/2^8$	3.91 kHz	7.81 kHz	15.63 kHz	31.25 kHz	62.5 kHz
1	0	0	1	$f_{CLK}/2^9$	1.95 kHz	3.91 kHz	7.81 kHz	15.63 kHz	31.25 kHz
1	0	1	0	$f_{CLK}/2^{10}$	977 Hz	1.95 kHz	3.91 kHz	7.81 kHz	15.63 kHz
1	0	1	1	$f_{CLK}/2^{11}$	488 Hz	977 Hz	1.95 kHz	3.91 kHz	7.81 kHz
1	1	0	0	$f_{CLK}/2^{12}$	244 Hz	488 Hz	977 Hz	1.95 kHz	3.91 kHz
1	1	0	1	$f_{CLK}/2^{13}$	122 Hz	244 Hz	488 Hz	977 Hz	1.95 kHz
1	1	1	0	$f_{CLK}/2^{14}$	61 Hz	122 Hz	244 Hz	488 Hz	977 Hz
1	1	1	1	$f_{CLK}/2^{15}$	31 Hz	61 Hz	122 Hz	244 Hz	488 Hz

Note 1. 20- to 32-pin products

Note 2. 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, 1)

Remark 3. k: Channel number (k = 0, 1)

13.3.3 Serial mode register mn (SMRmn)

The SMRmn register is used to set an operation mode of channel n. It is also used to select an operation clock (f_{MCK}), specify whether the serial clock (f_{SCK}) may be input or not, set a start trigger, the operating mode (as simplified SPI or 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.

The value of each SMRmn register is 0020H following a reset.

Figure 13-7. Format of Serial Mode Register mn (SMRmn) (1/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03)Note 1, F0150H, F0151H (SMR10), F0152H, F0153H (SMR11)Note 1

After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn	0	0	0	0	0	STSmn Note 2	0	SISmn0Note 2	1	0	0	MDmn2	MDmn1	MDmn0

CKSmn	Selection of operation clock (f_{MCK}) of channel n
0	Operation clock CKm0 set by the SPSm register
1	Operation clock CKm1 set by the SPSm register
Operation clock (f_{MCK}) is used by the edge detector. In addition, depending on the setting of the CCSmn bit and the higher 7 bits of the SDRmn register, a transfer clock (f_{TCLK}) is generated.	

CCSmn	Selection of transfer clock (f_{TCLK}) of channel n
0	Divided operation clock f_{MCK} specified by the CKSmn bit
1	Clock input f_{SCK} from the SCKp pin (slave transfer in simplified SPI or CSI mode)
Transfer clock f_{TCLK} is used for the shift register, communication controller, output controller, interrupt controller, and error controller. When CCSmn = 0, the division ratio of operation clock (f_{MCK}) is set by the higher 7 bits of the SDRmn register.	

STSmn Note 2	Selection of start trigger source
0	Only software trigger is valid (selected for simplified SPI or 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 1. SMR00, SMR01: All products
SMR02, SMR03: 16- to 32-pin products
SMR10, SMR11: 20- to 32-pin products

Note 2. The SMR01, SMR03, and SMR11 registers 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, SMR02, and SMR10 registers) to 0. Be sure to set bit 5 to 1.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 11, 20)
q: UART number (q = 0 to 2), r: IIC number (r = 00, 11, 20), mn = 00 to 03, 10, 11

Figure 13-7. Format of Serial Mode Register mn (SMRmn) (2/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03)Note 1, F0150H, F0151H (SMR10), F0152H, F0153H (SMR11)Note 1

After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn	0	0	0	0	0	STSmn Note 2	0	SISmn0Note 2	1	0	0	MDmn2	MDmn1	MDmn0

SISmn0 Note 2	Controls inversion of level of receive data of channel n in UART mode
0	Falling edge is detected as the start bit. The input communication data is captured as is.
1	Rising edge is detected as the start bit. The input communication data is inverted and captured.

MDmn2	MDmn1	Setting of operation mode of channel n
0	0	Simplified SPI (CSI) mode
0	1	UART mode
1	0	Simplified I^2C mode
1	1	Setting prohibited

MDmn0	Selection of interrupt source of channel n
0	Transfer end interrupt
1	Buffer empty interrupt (Occurs when data is transferred from the SDRmn register to the shift register.)
For continuous transmission, set this bit to 1 and write the next transmit data when SDRmn data has run out.	

Note 1. SMR00, SMR01: All products
SMR02, SMR03: 16- to 32-pin products
SMR10, SMR11: 20- to 32-pin products

Note 2. The SMR01, SMR03, and SMR11 registers 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, SMR02, and SMR10 registers) to 0. Be sure to set bit 5 to 1.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 11, 20)
q: UART number (q = 0 to 2), r: IIC number (r = 00, 11, 20), mn = 00 to 03, 10, 11

13.3.4 Serial communication operation setting register mn (SCRmn)

The SCRmn is a communication operation setting register of channel n. It is used to set a data transmission/reception mode, phase of data and clock, whether an error signal is to be masked or not, parity bit, start bit, stop bit, and data length.

Rewriting the SCRmn register is prohibited when the register is in operation (when SEmn = 1).

The SCRmn register can be set by a 16-bit memory manipulation instruction.

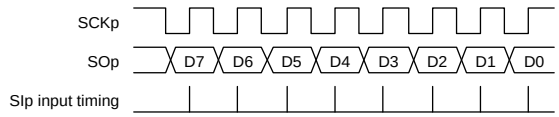
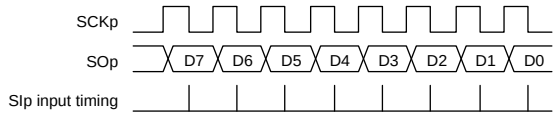
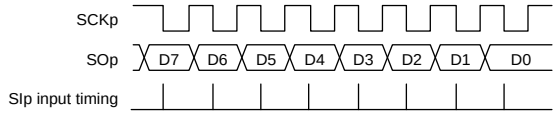
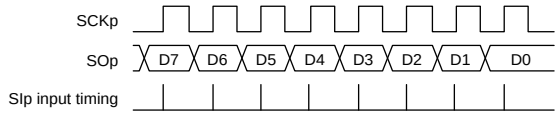
The value of each SCRmn register is 0087H following a reset.

Figure 13-8. Format of Serial Communication Operation Setting Register mn (SCRmn) (1/2)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03)Note 1, F0158H, F0159H (SCR10), F015AH,F015BH (SCR11)Note 1
After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn	0	EOCmn	PTCmn1	PTCmn0	DIRmn	0	SLCmn1Note 2	SLCmn0	0	1	DLSmn1Note 3	DLSmn0

TXEmn	RXEmn	Setting of operation mode of channel n
0	0	Disable communication.
0	1	Reception only
1	0	Transmission only
1	1	Transmission/reception

DAPmn	CKPmn	Selection of data and clock phase in simplified SPI (CSI) mode	Type
0	0		1
0	1		2
1	0		3
1	1		4
Be sure to set DAPmn, CKPmn = 0, 0 in the UART mode and simplified I^2C mode.			

EOCmn	Mask control of error interrupt signal (INTSREx (x = 0 to 3))
0	Disables generation of error interrupt INTSREx (INTSRx is generated).
1	Enables generation of error interrupt INTSREx (INTSRx is not generated if an error occurs).
Set EOCmn = 0 in the simplified SPI (CSI) mode, simplified I^2C mode, and during UART transmission.Note 4	

Note 1. SCR00, SCR01: All products
SCR02, SCR03: 16- to 32-pin products
SCR10, SCR11: 20- to 32-pin products

Note 2. The SCR00, SCR02, and SCR10 registers only.

Note 3. This bit is fixed to 1 for the other registers. This bit is fixed to 1 for the other registers.

Note 4. 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, SCR03, and SCR11 registers to 0). Be sure to set bit 2 to 1.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 11, 20)

Figure 13-8. Format of Serial Communication Operation Setting Register mn (SCRmn) (2/2)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03)[Note 1], F0158H, F0159H (SCR10), F015AH,F015BH (SCR11)[Note 1]

After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn	0	EOCmn	PTCmn1	PTCmn0	DIRmn	0	SLCmn1[Note 2]	SLCmn0	0	1	DLSmn1[Note 3]	DLSmn0

PTCmn1	PTCmn0	Setting of parity bit in UART mode	
		Transmission	Reception
0	0	Does not output the parity bit.	Receives without parity
0	1	Outputs 0 parity.[Note 4]	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.			

DIRmn	Selection of data transfer sequence in simplified SPI (CSI) and UART modes
0	Inputs/outputs data with MSB first.
1	Inputs/outputs data with LSB first.
Be sure to clear DIRmn = 0 in the simplified I^2C mode.	

SLCmn1[Note 2]	SLCmn0	Setting of stop bit in UART mode
0	0	No stop bit
0	1	Stop bit length = 1 bit
1	0	Stop bit length = 2 bits (mn = 00, 02, 10 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[Note 3]	DLSmn0	Setting of data length in simplified SPI (CSI) and UART modes
0	1	9-bit data length (stored in bits 0 to 8 of the SDRmn register) (settable in UART mode only)
1	0	7-bit data length (stored in bits 0 to 6 of the SDRmn register)
1	1	8-bit data length (stored in bits 0 to 7 of the SDRmn register)
Other than above		Setting prohibited
Be sure to set DLSmn1, DLSmn0 = 1, 1 in the simplified I^2C mode.		

Note 1. SCR00, SCR01: All products
SCR02, SCR03: 16- to 32-pin products
SCR10, SCR11: 20- to 32-pin products

Note 2. The SCR00, SCR02, and SCR10 registers only.

Note 3. The SCR00 and SCR01 registers only. This bit is fixed to 1 for the other registers.

Note 4. 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, SCR03, and SCR11 registers to 0). Be sure to set bit 2 to 1.**

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 11, 20)

13.3.5 Serial data register mn (SDRmn)

The SDRmn is the transmit/receive data register (16 bits) of channel n.

Bits 8 to 0 (lower 9 bits) of SDR00 and SDR01 or bits 7 to 0 (lower 8 bits) of SDR02[Note 1], SDR03[Note 1], SDR10[Note 2], and SDR11[Note 2] function as a transmit/receive buffer register, and bits 15 to 9 (higher 7 bits) are used as a register that sets the division ratio of the operation clock (f_{MCK}).

If the CCSmn bit of serial mode register mn (SMRmn) is cleared to 0, the clock set by dividing the operation clock by bits 15 to 9 (higher 7 bits) of the SDRmn register is used as the transfer clock.

If the CCSmn bit of serial mode register mn (SMRmn) is set to 1, set bits 15 to 9 (upper 7 bits) of SDR00 and SDR01 to 0000000B. The input clock f_{SCK} (slave transfer in simplified SPI or CSI mode) from the SCKp pin is used as the transfer clock.

The lower 8 or 9 bits of the SDRmn register function as a transmit/receive buffer register. During reception, the parallel data converted by the shift register is stored in the lower 8 or 9 bits, and during transmission, the data to be transmitted to the shift register is set to the lower 8 or 9 bits.

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

However, the higher 7 bits can only be written or read when the operation is stopped (SEmn = 0). During operation (SEmn = 1), a value is written only to the lower 8 or 9 bits of the SDRmn register. When the SDRmn register is read during operation, the higher 7 bits are always read as 0.

The value of each SDRmn register is 0000H following a reset.

Figure 13-9. Format of Serial Data Register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01) After reset: 0000H R/W

	FFF11H (SDR00)								FFF10H (SDR00)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn																

Address: FFF44H, FFF45H (SDR02)[Note 1], FFF46H, FFF47H (SDR03)[Note 1], FFF48H, FFF49H (SDR10)[Note 2], FFF4AH, FFF4BH (SDR11)[Note 2]

After reset: 0000H R/W

	FFF45H (SDR02)								FFF44H (SDR02)							
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn								0								

(Note 1, Note 2, Cautions, and Remarks are listed on the next page.)

SDRmn[15:9]							Transfer clock setting by dividing the operation 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. 16- to 32-pin products

Note 2. 20- to 32-pin products

Caution 1. Be sure to clear bit 8 of the SDR02 and SDR03 registers of the 16- to 32-pin products and the SDR10 and SDR11 registers of the 20- to 32-pin products to "0".

Caution 2. Setting SDRmn[15:9] to 0000000B or 0000001B is prohibited when UART is used.

Caution 3. Setting SDRmn[15:9] to 0000000B is prohibited when simplified I^2C is used. Set SDRmn[15:9] to 0000001B or greater.

Caution 4. 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 or 9 bits of the SDRmn register, see **13.2 Configuration of Serial Array Unit**.

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

13.3.6 Serial flag clear trigger register mn (SIRmn)

The SIRmn is a trigger register that is used to clear each error flag of channel n.

When each bit (FECTmn, PECTmn, OVCTmn) of this register is set to 1, the corresponding bit (FEFmn, PEFmn, OVFmn) of serial status register mn is cleared to 0. Because the SIRmn is a trigger register, it is cleared immediately when the corresponding bit of the SSRmn register is cleared.

The SIRmn register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SIRmn register can be set with an 8-bit memory manipulation instruction with SIRmnL.

The value of each SIRmn register is 0000H following a reset.

Figure 13-10. Format of Serial Flag Clear Trigger Register mn (SIRmn)

Address: F0108H, F0109H (SIR00) to F010EH, F010FH (SIR03)[Note 1], F0148H, F0149H (SIR10), F014AH, F014BH (SIR11)[Note 1]
After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	FECTmn[Note 2]	PECTmn	OVCTmn

FECTmn[Note 2]	Clear trigger of framing error flag of channel n
0	Not cleared
1	Clears the FEFmn bit of the SSRmn register to 0.

PECTmn	Clear trigger of parity error flag of channel n
0	Not cleared
1	Clears the PEFmn bit of the SSRmn register to 0.

OVCTmn	Clear trigger of overrun error flag of channel n
0	Not cleared
1	Clears the OVFmn bit of the SSRmn register to 0.

Note 1. SIR00, SIR01: All products
SIR02, SIR03: 16- to 32-pin products
SIR10, SIR11: 20- to 32-pin products

Note 2. The SIR01, SIR03, and SIR11 registers only.

Caution Be sure to clear bits 15 to 3 (or bits 15 to 2 for the SIR00, SIR02, and SIR10 registers) to 0

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Remark 2. When the SIRmn register is read, 0000H is always read.

13.3.7 Serial status register mn (SSRmn)

The SSRmn register indicates the state of communications and occurrence of errors for channel n. The errors indicated by this register are a framing error, parity error, and overrun error.

The SSRmn register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSRmn register can be read with an 8-bit memory manipulation instruction with SSRmnL.

The value of each SSRmn register is 0000H following a reset.

Figure 13-11. Format of Serial Status Register mn (SSRmn) (1/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03)[Note 1], F0140H, F0141H (SSR10), F0142H, F0143H (SSR11)[Note 1]

After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSFmn	BFFmn	0	0	FEFmn Note 2	PEFmn	OVFmn

TSFmn	Flag indicating the state of communications for channel n
0	Communication is stopped or suspended.
1	Communication is in progress.

<Clear conditions>

- The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is suspended).
- Communication ends.

<Set condition>

- Communication starts.

BFFmn	Flag indicating the state of the buffer register for channel n
0	Valid data is not stored in the SDRmn register.
1	Valid data is stored in the SDRmn register.

<Clear conditions>

- Transferring transmit data from the SDRmn register to the shift register ends during transmission.
- Reading receive data from the SDRmn register ends during reception.
- The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is enabled).

<Set conditions>

- Transmit data is written to the SDRmn register while the TXEmn bit of the SCRmn register is set to 1 (transmission or transmission/reception mode in each communication mode).
- Receive data is stored in the SDRmn register while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission/reception mode in each communication mode).
- A reception error occurs.

Note 1. SSR00, SSR01: All products
SSR02, SSR03: 16- to 32-pin products
SSR10, SSR11: 20- to 32-pin products

Note 2. The SSR01, SSR03, and SSR11 registers only.

Remark m: Unit number (m = 0, 1), 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)[Note 1], F0140H, F0141H (SSR10), F0142H, F0143H (SSR11)[Note 1]

After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSFmn	BFFmn	0	0	FEFmn Note 2	PEFmn	OVFmn

FEFmn Note 2	Framing error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception).
<Clear condition> ● 1 is written to the FECTmn bit of the SIRmn register. <Set condition> ● A stop bit is not detected when UART reception ends.	

PEFmn	Parity/ACK error detection flag of channel n
0	No error occurs.
1	Parity error occurs (during UART reception) or ACK is not detected (during I^2C transmission).
<Clear condition> ● 1 is written to the PECTmn bit of the SIRmn register. <Set conditions> ● The parity of the transmit data and the parity bit do not match when UART reception ends (parity error). ● No ACK signal is returned from the slave at the ACK reception timing during I^2C transmission (ACK is not detected).	

OVFmn	Overrun error detection flag of channel n
0	No error occurs.
1	An error occurs
<Clear condition> ● 1 is written to the OVCTmn bit of the SIRmn register. <Set conditions> ● Even though receive data is stored in the SDRmn register, that data is not read and transmit data or the next receive data is written while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission/reception mode in each communication mode). ● Transmit data is not ready for slave transmission or transmission/reception in simplified SPI (CSI) mode.	

Note 1. SSR00, SSR01: All products
SSR02, SSR03: 16- to 32-pin products
SSR10, SSR11: 20- to 32-pin products

Note 2. The SSR01, SSR03, and SSR11 registers only.

Caution If data is written to the SDRmn register when BFFmn = 1, the transmit/receive data stored in the register is discarded and an overrun error (OVFmn = 1) is detected.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

13.3.8 Serial channel start register m (SSm)

The SSm is a trigger register that is used to enable starting communication/count by each channel.

When 1 is written to a bit (SSmn) of this register, the corresponding bit (SEmn) of serial channel enable status register m (SEm) is set to 1 (operation is enabled). Because the SSmn bit is a trigger bit, it is cleared immediately when SEmn = 1.

The SSm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SSm register can be set with an 1-bit or 8-bit memory manipulation instruction with SSmL.

The value of each SSm register is 0000H following a reset.

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 Note 1	SS02 Note 1	SS01	SS00

Address: F0162H, F0163H (SS1)[Note 2] After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SS11	SS10

SSmn	Operation start trigger of channel n
0	No trigger operation
1	Set the SEmn bit to 1 to place the channel in the communications waiting state.[Note 3]

Note 1. 16- to 32-pin products only

Note 2. 20- to 32-pin products only

Note 3. Setting an SSmn bit to 1 during communications stops communications through channel n and places the channel in the waiting state. At this time, the values of the control registers and shift register, the states of the SCKmn and SOmn pins, and the values of the FEFmn, PEFmn, and OVFmn flags are retained.

Caution 1. Be sure to clear bits 15 to 2 of the SS0 register of the 10-pin products, bits 15 to 4 of the SS0 register of the 16- to 32-pin products, and bits 15 to 2 of the SS1 register of the 20- to 32-pin products 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 at least 4 f_{MCK} clock cycles have elapsed.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Remark 2. When the SSm register is read, 0000H is always read.

13.3.9 Serial channel stop register m (STm)

The STm is a trigger register that is used to enable stopping communication/count by each channel.

When 1 is written to a bit (STmn) of this register, the corresponding bit (SEmn) of serial channel enable status register m (SEm) is cleared to 0 (operation is stopped). Because the STmn bit is a trigger bit, it is cleared immediately when SEmn = 0.

The STm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the STm register can be set with a 1-bit or 8-bit memory manipulation instruction with STmL.

The value of each STm register is 0000H following a reset.

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 Note 1	ST02 Note 1	ST01	ST00

Address: F0164H, F0165H (SS1)Note 2 After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ST11	ST10

STmn	Operation stop trigger of channel n
0	No trigger operation
1	Clears the SEmn bit to 0 and stops the communication operationNote 3

Note 1. 16- to 32-pin products only

Note 2. 20- to 32-pin products only

Note 3. The values of the control registers and shift register, the states of the SCKmn and SOmn pins, and the values of the FEFmn, PEFmn, and OVFmn flags are retained.

Caution Be sure to clear bits 15 to 2 of the ST0 register of the 10-pin products, bits 15 to 4 of the ST0 register of the 16- to 32-pin products, and bits 15 to 2 of the ST1 register of the 20- to 32-pin products to 0.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Remark 2. When the STm register is read, 0000H is always read.

13.3.10 Serial channel enable status register m (SEm)

The SEm register indicates whether data transmission/reception operation of each channel is enabled or stopped.

When 1 is written to a bit of serial channel start register m (SSm), the corresponding bit of this register is set to 1. When 1 is written to a bit of serial channel stop register m (STm), the corresponding bit is cleared to 0.

For channel n whose operation is enabled, the value of the CKOmn bit of serial output register m (SOm) to be described later cannot be rewritten by software, and a value reflected by a communication operation is output from the serial clock pin.

For channel n whose operation is stopped, the value of the CKOmn bit of the SOm register can be set by software and is output from the serial clock pin. In this way, any waveform, such as that of a start condition/stop condition, can be created by software.

The SEm register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the SEm register can be read with a 1-bit or 8-bit memory manipulation instruction with SEmL.

The value of each SEm register is 0000H following a reset.

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 Note 1	SE02 Note 1	SE01	SE00

Address: F0160H, F0161H (SE1)Note 2 After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SE11	SE10

SEmn	Indication of whether operation of channel n is enabled or stopped
0	Operation stops.
1	Operation is enabled.

Note 1. 16- to 32-pin products only

Note 2. 20- to 32-pin products only

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

13.3.11 Serial output enable register m (SOEm)

The SOEm register is used to enable or stop output of the serial communication operation of each channel.

For channel n whose serial output is enabled, the value of the SOmn bit of serial output register m (SOm) to be described later cannot be rewritten by software, and a value reflected by a communication operation is output from the serial data output pin.

For channel n, whose serial output is stopped, the SOmn bit value of the SOm register can be set by software, and that value can be output from the serial data output pin. In this way, any waveform, such as that of a start condition/stop condition, can be created by software.

The SOEm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SOEm register can be set with a 1-bit or 8-bit memory manipulation instruction with SOEmL.

The value of each SOEm register is 0000H following a reset.

Figure 13-15. Format of Serial Output Enable Register m (SOEm)

Address: F012AH, F012BH (SOE0) 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	SOE03 Note 1	SOE02 Note 1	0	SOE00

Address: F016AH, F016BH (SOE1)[Note 2] After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE10

SOEmn	Serial output enable/stop of channel n
0	Stops output by serial communication operation.
1	Enables output by serial communication operation.

Note 1. 16- to 32-pin products only

Note 2. 20- to 32-pin products only

Caution Be sure to clear bits 15 to 1 of the SOE0 register of the 10-pin products, bits 15 to 4 and 1 of the SOE0 register of the 16- to 32-pin products, and bits 15 to 1 of the SOE1 register of the 20- to 32-pin products to 0.

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

13.3.12 Serial output register m (SOm)

The SOm is a buffer register for serial output of each channel.

The value of the SOmn bit of this register is output from the serial data output pin of channel n.

The value of the CKOmn bit of this register is output from the serial clock output pin of channel n.

The SOmn bit of this register can be rewritten by software only when serial output is disabled (SOEmn = 0). When serial output is enabled (SOEmn = 1), rewriting by software is ignored, and the value of the register can be changed only by a serial communication operation.

The CKOmn bit of this register can be rewritten by software only when the channel operation is stopped (SEmn = 0). While channel operation is enabled (SEmn = 1), rewriting by software is ignored, and the value of the CKOmn bit can be changed only by a serial communication operation.

To use the pin for serial interface as a port function pin, set the corresponding CKOmn and SOmn bits to 1.

The SOm register can be set by a 16-bit memory manipulation instruction.

The values of the SO0 and SO1 registers are set to 0F0FH and 0303H, respectively, following a reset.

Figure 13-16. Format of Serial Output Register m (SOm)

Address: F0128H, F0129H (SO0) 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	CKO03 Note 1	1	1	CKO00	0	0	0	0	SO03 Note 1	SO02 Note 1	1	SO00

Address: F0168H, F0169H (SO1)Note 2 After reset: 0303H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	0	0	1	CKO10	0	0	0	0	0	0	1	SO10

CKOmn	Serial clock output of channel n
0	Serial clock output value is 0.
1	Serial clock output value is 1.

SOmn	Serial data output of channel n
0	Serial data output value is 0.
1	Serial data output value is 1.

Note 1. 16- to 32-pin products only

Note 2. 20- to 32-pin products only

Caution Be sure to clear bits 15 to 12 and 7 to 4 of the SO0 register of the 10-pin products to 0, and set bits 11 to 9 and 3 to 1 of the SO0 register of the 10-pin products to 1. Be sure to clear bits 15 to 12 and 7 to 4 of the SO0 register of the 16-, 20-, 24-, and 32-pin products to 0, and set bits 10, 9, and 1 of the SO0 register of the 16-, 20-, 24-, and 32-pin products to 1. Be sure to clear bits 15 to 10 and 7 to 2 of the SO1 register of the 20-, 24-, and 32-pin products to 0, and set bits 9 and 1 of the SO1 register of the 20-, 24-, and 32-pin products to 1.

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

13.3.13 Serial output level register m (SOLm)

The SOLm register is used to set inversion of the data output level of each channel.

This register can be set only in the UART mode. Be sure to set 0 for the bit corresponding the channel used in the simplified SPI (CSI) mode or simplified I^2C mode.

Inverting channel n by using this register is reflected on pin output only when serial output is enabled (SOEmn = 1). When serial output is disabled (SOEmn = 0), the value of the SOmn bit is output as is.

Rewriting the SOLm register is prohibited when the register is in operation (when SEmn = 1).

The SOLm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SOLm register can be set with an 8-bit memory manipulation instruction with SOLmL.

The value of each SOLm register is 0000H following a reset.

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 Note 1	0	SOL00

Address: F0174H, F0175H (SOL1)Note 2 After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL10

SOLmn	Selects inversion of the level of the transmit data of channel n in UART mode
0	Communication data is output as is.
1	Communication data is inverted and output.

Note 1. 16- to 32-pin products only

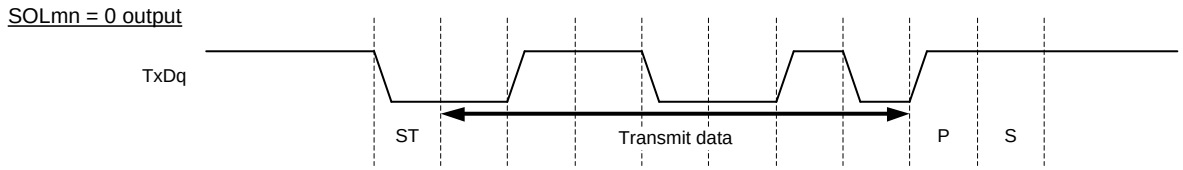
Note 2. 20- to 32-pin products only

Caution Be sure to clear bits 15 to 1 of the SOL0 register of the 10-pin products, bits 15 to 3 and 1 of the SOL0 register of the 16- to 32-pin products, and bits 15 to 1 of the SOL1 register of the 20- to 32-pin products to 0.

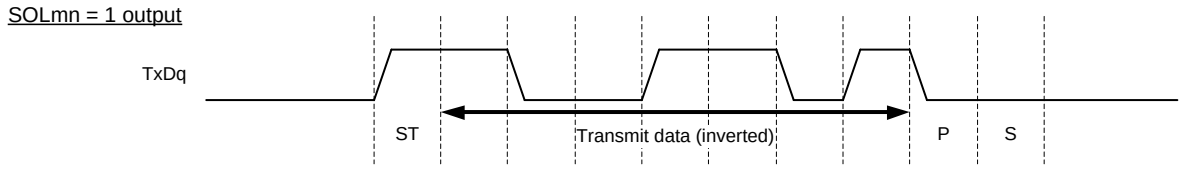
Figure 13-18 shows examples in which the level of transmit data is reversed during UART transmission.

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, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10

13.3.14 Input switch control register (ISC)

The ISC1 and ISC0 bits in the ISC register are used to handle the combination of the external interrupt and the timer array unit at the time of baud rate correction of UART0.

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

The value of the ISC register is 00H following a reset.

Figure 13-19. Format of Input Switch Control Register (ISC)

Address: F0073H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	0	0	0	ISC1	ISC0

ISC1	Switching the channel 1 of the timer array unit
0	Select the input signal on TI01 pin as the timer input (normal operation)
1	Select the input signal on RxD0 pin as the timer input (Detection of the wake-up signal and pulse-width-measurement for baud rate correction)

ISC0	Switching the external interrupt (INTP0)
0	Select the input signal on INTP0 pin as the external interrupt input (normal operation)
1	Select the input signal on RxD0 pin as the external interrupt input (detection of the wake-up signal)

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

13.3.15 Noise filter enable register 0 (NFEN0)

The NFEN0 register is used to set whether the noise filter can be used for the input signal from the serial data input pin to each channel.

Disable the noise filter of the pin used for simplified SPI (CSI) or simplified I^2C communication, by clearing the corresponding bit of this register to 0.

Enable the noise filter of the pin used for UART communication, by setting the corresponding bit of this register to 1. When the noise filter is enabled, after synchronization is performed with the operation clock (f_{MCK}) of the target channel, 2-clock match detection is performed. When the noise filter is disabled, only synchronization is performed with the operation clock (f_{MCK}) of the target channel.

The NFEN0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

The value of the NFEN0 register is 00H following a reset.

Figure 13-20. Format of Noise Filter Enable Register 0 (NFEN0)

Address: F0070H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN0	0	0	0	SNFEN20	0	SNFEN10	0	SNFEN00

SNFEN20	Use of noise filter of RxD2 pin
0	Noise filter OFF
1	Noise filter ON
Set the SNFEN20 bit to 1 to use the RxD2 pin. Clear SNFEN20 to 0 to use the other than RxD2 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 SNFEN10 to 0 to use the other than RxD1 pin.	

SNFEN00	Use of noise filter of RxD0 pin
0	Noise filter OFF
1	Noise filter ON
Set the SNFEN00 bit to 1 to use the RxD0 pin. Clear SNFEN00 to 0 to use the other than RxD0 pin.	

Caution Be sure to clear bits 7 to 1 of the 10-pin products, bits 7 to 3 and 1 of the 16- to 32-pin products, and bits 7 to 5, 3, and 1 of the 20- to 32-pin products to 0.

13.3.16 Registers controlling port functions of serial input/output pins

Using the serial array unit requires setting of the registers that control the port functions multiplexed on the target channel (port mode register (PMxx), port register (Pxx), port output mode register (POMxx), port mode control register (PMCxx)).

For details, see **4.3.1 Port mode registers 0, 1, 2, 4, 6, 12 (PM0, PM1, PM2, PM4, PM6, PM12)**, **4.3.2 Port registers 0, 1, 2, 4, 6, 12, 13 (P0, P1, P2, P4, P6, P12, P13)**, **4.3.4 Port input mode registers 0, 1, 2, 4 (POM0, POM1, POM2, POM4)**, and **4.3.5 Port mode control registers 0, 2 (PMC0, PMC2)**.

Specifically, using a port pin with a multiplexed serial data or serial clock output function (e.g. P04/ANI3/IVREF0/INTP3/TS04/TI06/TO06/TxD1) for serial data or serial clock output, requires setting the corresponding bits in the port mode control register (PMCxx) and port mode register (PMxx) to 0, and the corresponding bit in the port register (Pxx) to 1.

When using the port pin in N-ch open-drain output (V_{DD} tolerance) mode, set the corresponding bit in the port output mode register (POMxx) to 1.

Example) When P04/ANI3/IVREF0/INTP3/TS04/TI06/TO06/TxD1/(TI01/TO01)/(SI00/RxD0/SDA00)/(SO00/TxD0) is to be used for serial data output

Set the PMC04 bit of port mode control register 0 to 0.
Set the PM04 bit of port mode register 0 to 0.
Set the P04 bit of port register 0 to 1.

Specifically, using a port pin with a multiplexed serial data or serial clock input function (e.g. P01/TOOLRxD/ANI0/INTP5/TS00/TI02/TO02/SI00/RxD0/SDA00) for serial data or serial clock 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. In this case, the corresponding bit in the port register (Pxx) can be set to 0 or 1.

Example) When P01/TOOLRxD/ANI0/INTP5/TS00/TI02/TO02/SI00/RxD0/SDA00/(TI01/TO01)/(SI11/SDA11)/(SO11)/(SDAA0) is to be used for serial data input

Set the PMC01 bit of port mode control register 0 to 0.
Set the PM01 bit of port mode register 0 to 1.
Set the P01 bit of port register 0 to 0 or 1.

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 the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise.

To stop the operation of serial array unit 0, set bit 2 (SAU0EN) to 0. To stop the operation of serial array unit 1, set bit 3 (SAU1EN) to 0.

Figure 13-21. 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.

Symbol	7	6	5	4	3	2	1	0
PER0	TMKAEN	CMPEN	ADCEN	IICA0EN	SAU1EN[Note 1]	SAU0EN		TAU0EN
	×	×	×	×	0/1	0/1	0	×

Control of SAUm input clock
0: Stops supply of input clock
1: Supplies input clock

Note 1. This is not provided in the 10- and 16-pin products.

Caution 1. If SAUmEN = 0, writing to the registers which control serial array unit m is ignored, and, even if the register is read, only the initial value is read.
Note that this does not apply to the following registers.

- **Input switch control register (ISC)**
- **Noise filter enable register 0 (NFEN0)**
- **Port output mode registers 0, 1, 2, 4 (POM0, POM1, POM2, POM4)**
- **Port mode control registers 0, 2 (PMC0, PMC2)**
- **Port mode registers 0, 1, 2, 4, 6 (PM0, PM1, PM2, PM4, PM6)**
- **Port registers 0, 1, 2, 4, 6 (P0, P1, P2, P4, P6)**

Caution 2. Be sure to clear the following bits to 0.

- **10- and 16-pin products: Bits 1 and 3**
- **20- to 32-pin products: Bit 1**

Remark : Setting disabled (set to the initial value)
×: 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 13-22. Each Register Setting When Stopping the Operation by Channels (1/2)

(a) Serial channel stop register m (STm) … The STm 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													STm3 Note 1	STm2 Note 1	STm1	STm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1	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													SEm3 Note 1	SEm2 Note 1	SEm1	SEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0/1	0/1

0: Operation stops

*The SEm is a read-only status register, whose operation is stopped by using the STm register.
With a channel whose operation is stopped, the value of the CKOmn bit of the SOm register can be set by software.

(c) Serial output enable register m (SOEm) … This register is used to enable or stop output of the serial communication operation of each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm													SOEm 3 Note 1	SOEm 2 Note 1		SOEm 0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0/1	0	0/1

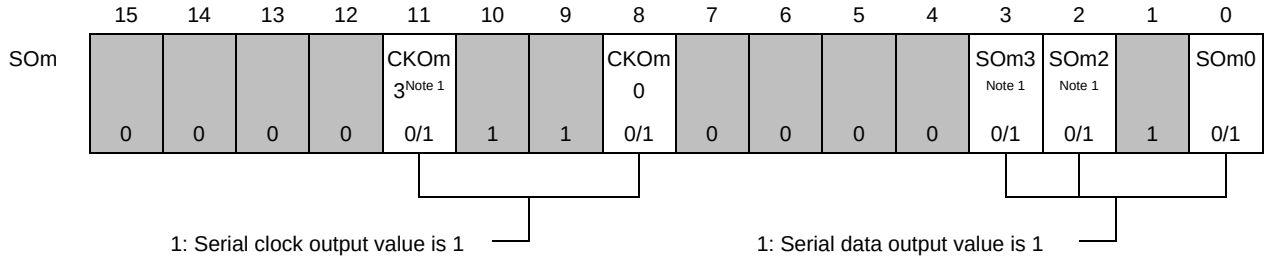
0: Stops output by serial communication operation

*For channel n, whose serial output is stopped, the SOmn bit value of the SOm register can be set by software.

(Note 1 and Remarks are listed on the next page.)

Figure 13-22. Each Register Setting When Stopping the Operation by Channels (2/2)

(d) Serial output register m (SOm) … The SOm is a buffer register for serial output of each channel.



*When using pins corresponding to each channel as port function pins, set the corresponding CKOmn, SOmn bits to 1.

Note 1. 16-, 20-, 24-, and 32-pin products only of serial array unit 0.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2, 3), mn = 00, 02, 03, 10

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, CSI11, CSI20) 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 1]

  During master communication: Max. $f_{CLK}/4$
  During slave communication: Max. $f_{MCK}/6$

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

Note 1. Set up the transfer rate within a range satisfying the SCK cycle time (t_{KCY}). For details, see **CHAPTER 26 ELECTRICAL SPECIFICATIONS (T_A = −40 to +85°C)** and **CHAPTER 27 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C, T_A = −40 to +125°C)**.

The channels supporting simplified SPI (CSI00, CSI11, CSI20) are channels 0 and 3 of SAU0 and channel 0 of SAU1.

- 10-pin products

Unit	Channel	Used as simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	—	—
	3	—		—

- 16-pin products

Unit	Channel	Used as simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11

- 20-, 24-, and 32-pin products

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	—		—

Simplified SPI (CSI00, CSI11, CSI20) performs the following six 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**).

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	CSI11	CSI20
Target channel	Channel 0 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1
Pins used	SCK00, SO00	SCK11, SO11	SCK20, SO20
Interrupt	INTCSI00	INTCSI11	INTCSI20
	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 1]	Max. f_{CLK} /4 [Hz] Min. f_{CLK}/ (2 × 2^{15} × 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 cycle before the start of the serial clock operation.		
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse		
Data direction	MSB or LSB first		

Note 1. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **CHAPTER 26 ELECTRICAL SPECIFICATIONS (T_A = −40 to +85°C)** and **CHAPTER 27 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C, T_A = −40 to +125°C)**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), mn = 00, 03, 10

(1) Register setting

Figure 13-23. Example of Contents of Registers for Master Transmission of simplified SPI (CSI00, CSI11, CSI20) (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	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	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	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	1	0	0/1	0/1	0	0	0	0	0/1	0	0	0	0	1	1	0/1

Selection of the data and clock phase
(For details about the setting, see **13.3 Registers to Control the Serial Array Unit**.)

Selection of data transfer sequence
- 0: Inputs/outputs data with MSB first
- 1: Inputs/outputs data with LSB first.

Setting of data length
- 0: 7-bit data length
- 1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting (Operation clock (f_{MCK}) division setting)							0	Transmit data (Transmit data setting)							

SIOp

(d) Serial output register m (SOm) … Set only the bit of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm					CKOm3			CKOm0					SOm3	SOm2		SOm0
	0	0	0	0	0/1	1	1	0/1	0	0	0	0	0/1	×	1	0/1

Communication starts when a bit is 1 if the clock phase is non-reverse (the CKPmn bit of the SCRmn = 0). If the clock phase is reversed (CKPmn = 1), communication starts when a bit is 0.

(Remarks are listed on the next page.)

Figure 13-23. Example of Contents of Registers for Master Transmission of simplified SPI (CSI00, CSI11, CSI20) (2/2)

(e) Serial output enable register m (SOEm) … Set only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm													SOEm3	SOEm2		SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	×	0	0/1

(f) Serial channel start register m (SSm) … Set only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	×	×	0/1

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), p: CSI number (p = 00, 11, 20), mn = 00, 03, 10

Remark 2. ☐ : Setting is fixed in the simplified SPI (CSI) master 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 13-24. Initial Setting Procedure for Master Transmission

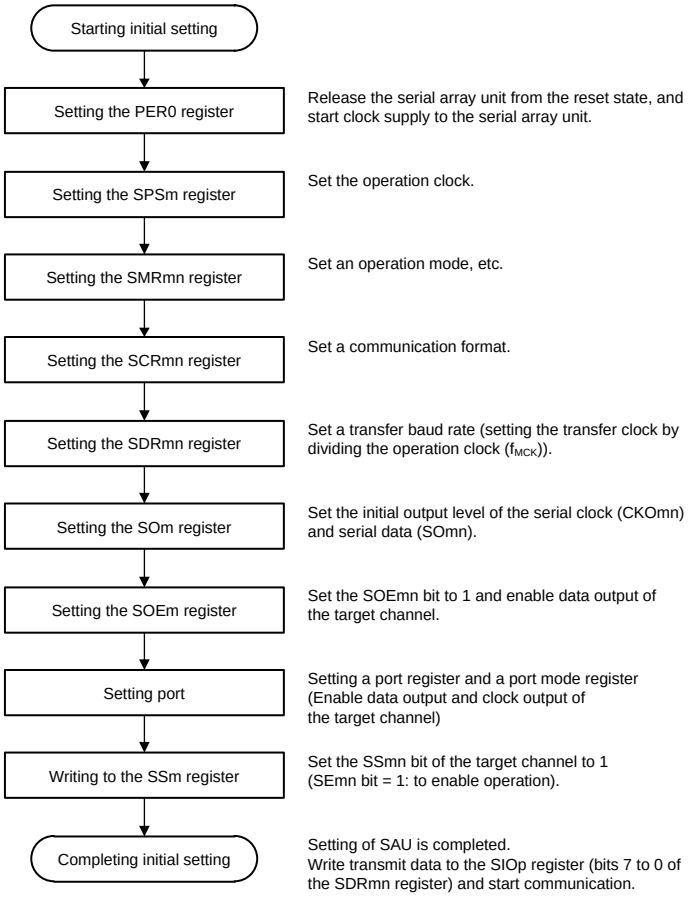


Figure 13-25. Procedure for Stopping Master Transmission

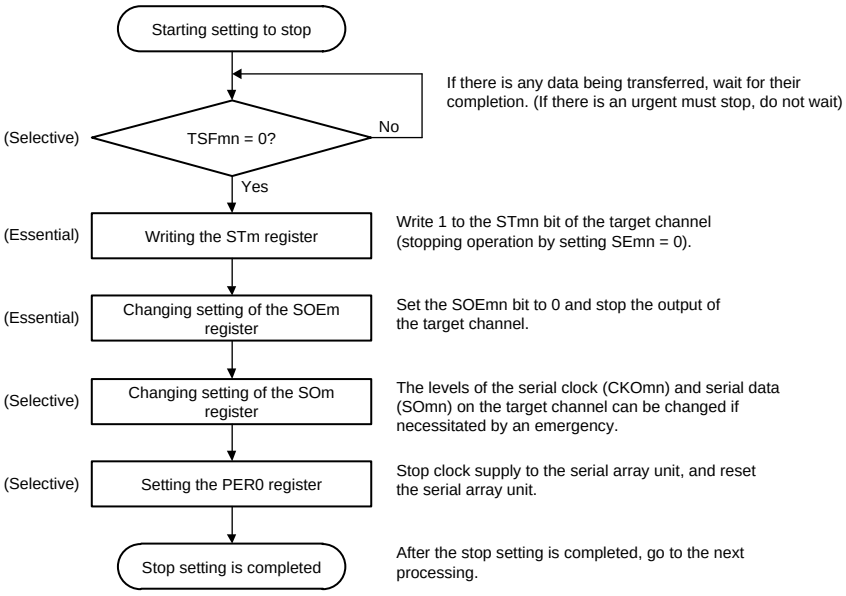
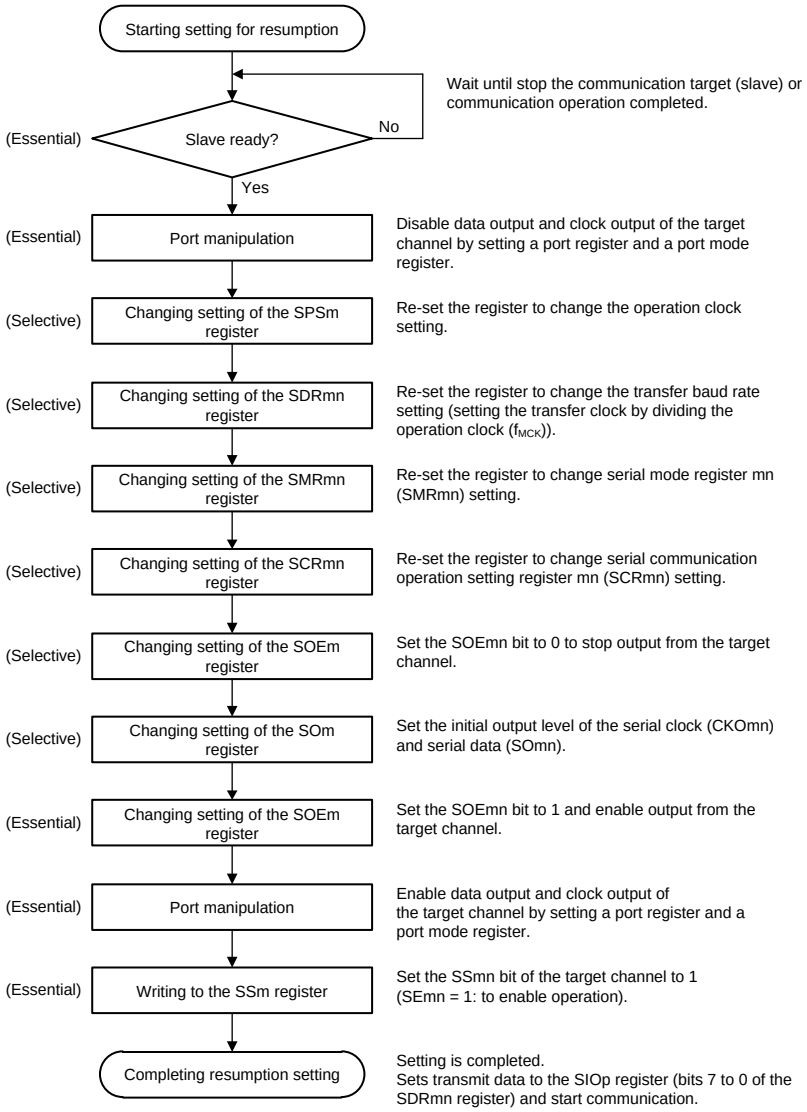


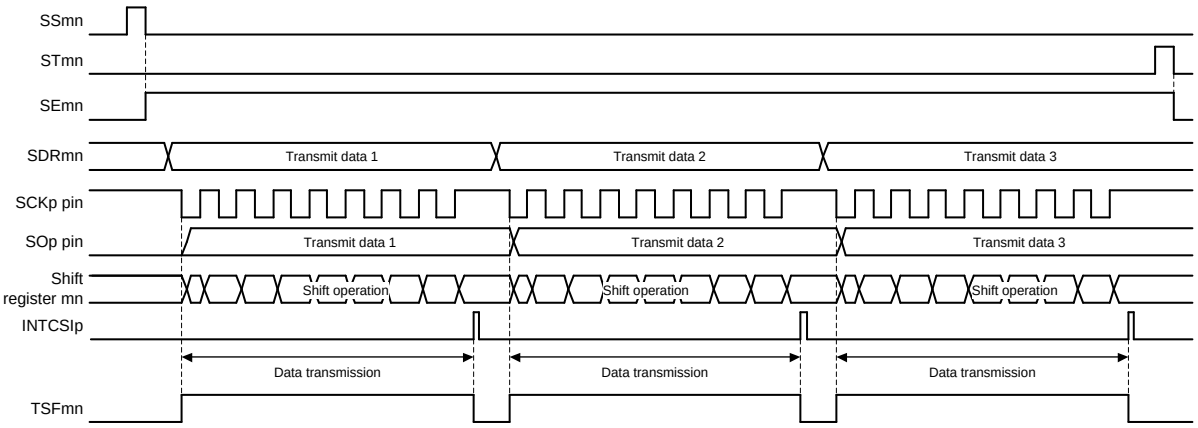
Figure 13-26. Procedure for Resuming Master Transmission



Remark If PER0 is rewritten while stopping the master transmission and the clock supply is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

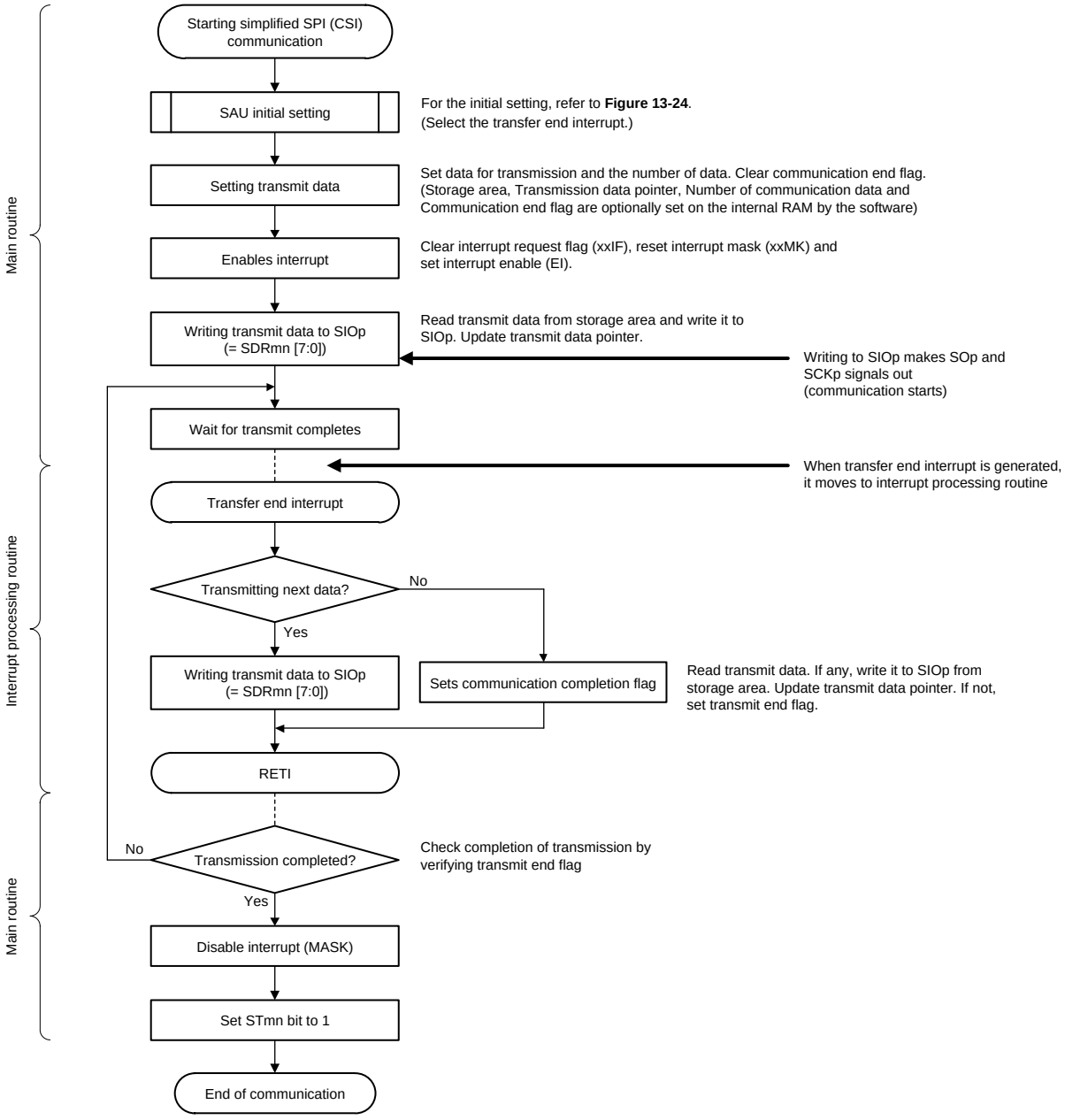
(3) Processing flow (in single-transmission mode)

Figure 13-27. Timing Chart of Master Transmission (in Single-Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)



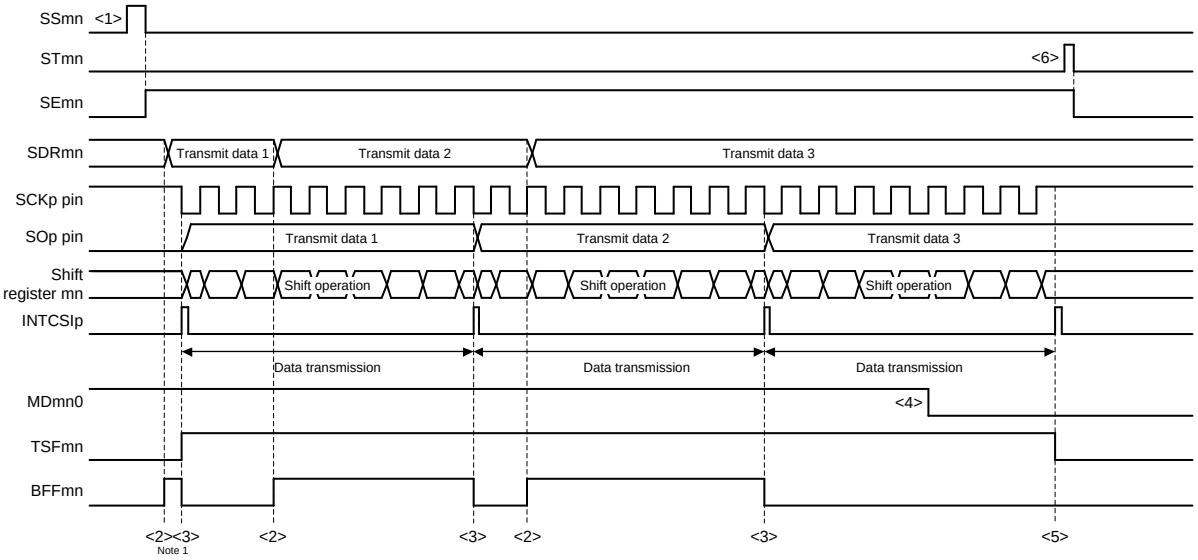
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), p: CSI number (p = 00, 11, 20), mn = 00, 03,10

Figure 13-28. Flowchart of Master Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 13-29. Timing Chart of Master Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)

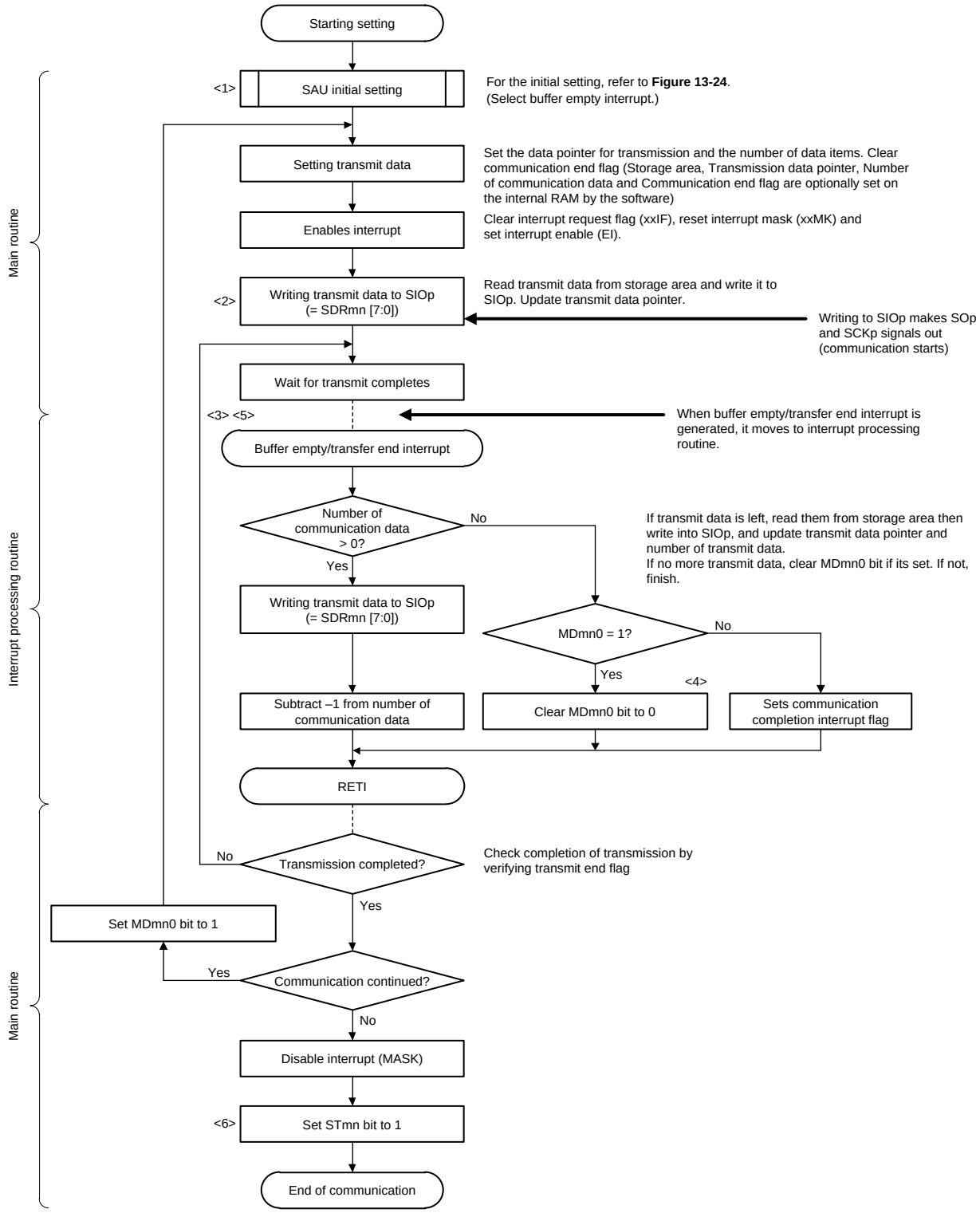


Note 1. If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

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 m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), p: CSI number (p = 00, 11, 20), mn = 00, 03,10

Figure 13-30. Flowchart of Master Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 13-29 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 another device.

Simplified SPI	CSI00	CSI11	CSI20
Target channel	Channel 0 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1
Pins used	SCK00, SI00	SCK11, SI11	SCK20, SI20
Interrupt	INTCSI00	INTCSI11	INTCSI20
	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 1]	Max. $f_{CLK}/4$ [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 cycle before the start of the serial clock operation.		
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse		
Data direction	MSB or LSB first		

Note 1. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **CHAPTER 26 ELECTRICAL SPECIFICATIONS (T_A = −40 to +85°C)** and **CHAPTER 27 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C, T_A = −40 to +125°C)**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), p: CSI number (p = 00, 11, 20), mn = 00, 03, 10

(1) Register setting

Figure 13-31. Example of Contents of Registers for Master Reception of simplified SPI (CSI00, CSI11, CSI20) (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	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	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	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	0	1	0/1	0/1	0	0	0	0	0/1	0	0	0	0	1	1	0/1

Selection of the data and clock phase
(For details about the setting, see **13.3 Registers to Control the Serial Array Unit**.)

Selection of data transfer sequence
- 0: Inputs/outputs data with MSB first
- 1: Inputs/outputs data with LSB first

Setting of data length
- 0: 7-bit data length
- 1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting (Operation clock (f_{MCK}) division setting)							0	Receive data (Write FFH as dummy data.)							

SIOp

(d) Serial output register m (SOm) … Set only the bit of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm					CKOm3			CKOm0					SOm3	SOm2		SOm0
	0	0	0	0	0/1	1	1	0/1	0	0	0	0	×	×	1	×

Communication starts when a bit is 1 if the clock phase is non-reverse (the CKPmn bit of the SCRmn = 0). If the clock phase is reversed (CKPmn = 1), communication starts when a bit is 0.

(Remarks are listed on the next page.)

Figure 13-31. Example of Contents of Registers for Master Reception of simplified SPI (CSI00, CSI11, CSI20) (2/2)

(e) Serial output enable register m (SOEm) … This register is not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm													SOEm3	SOEm2		SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	×	×	0	×

(f) Serial channel start register m (SSm) … Set only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	×	×	0/1

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), p: CSI number (p = 00, 11, 20), mn = 00, 03, 10

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 13-32. Initial Setting Procedure for Master Reception

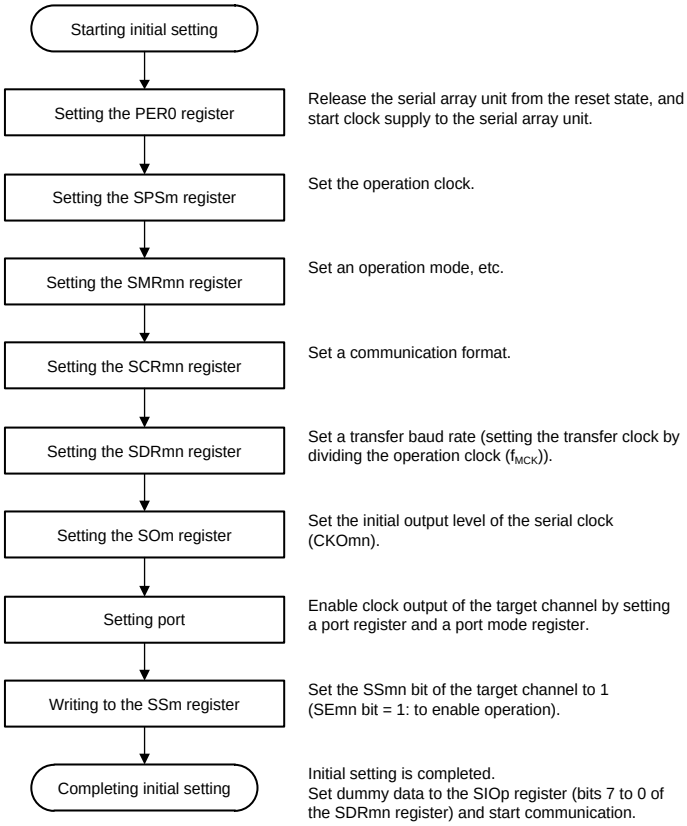


Figure 13-33. Procedure for Stopping Master Reception

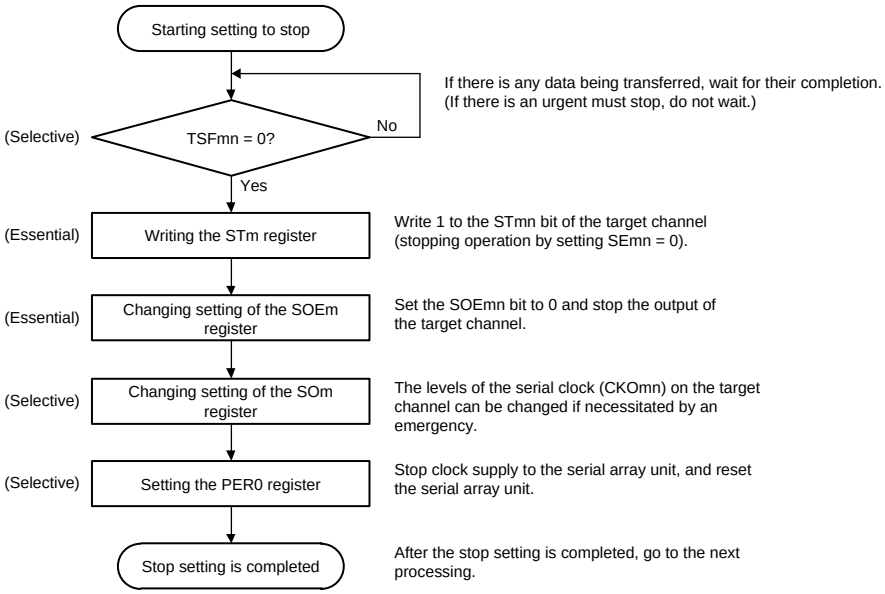
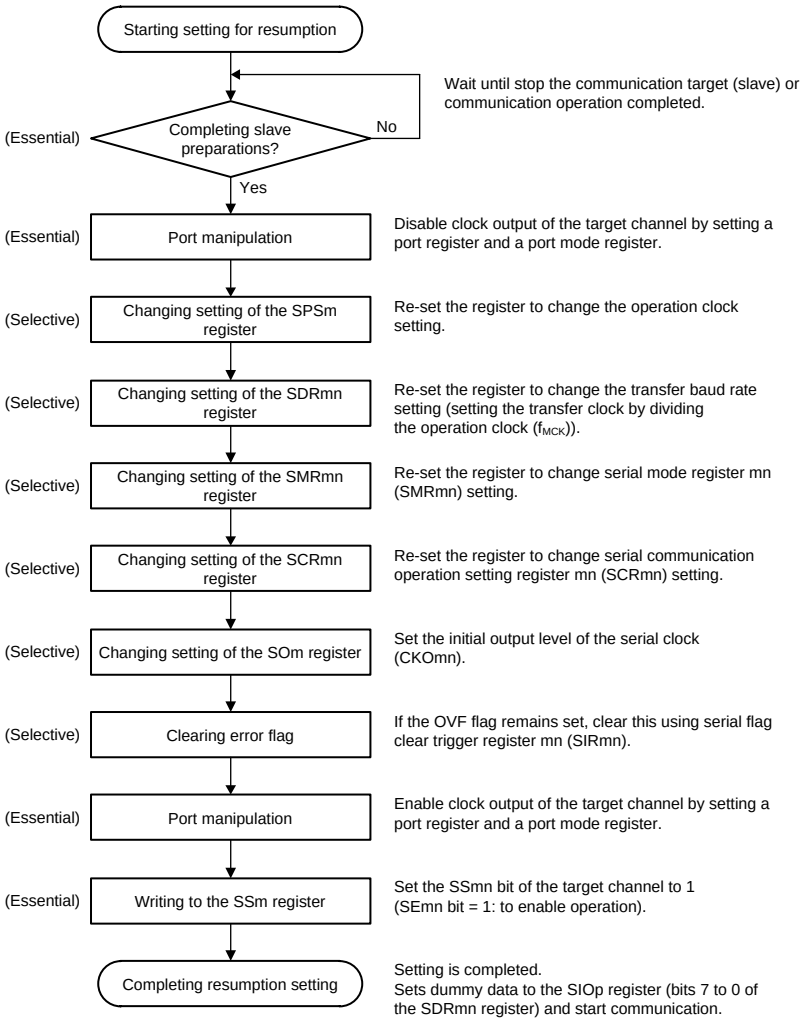


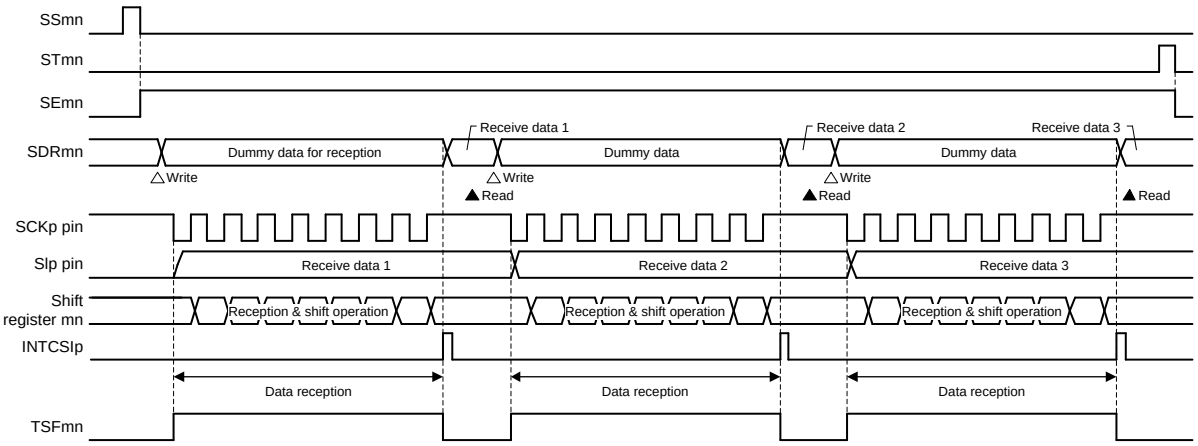
Figure 13-34. Procedure for Resuming Master Reception



Remark If PER0 is rewritten while stopping the communication to reset the serial array unit, wait until the communication target (slave) stops or communication finishes, and then perform initialization instead of restarting the communication.

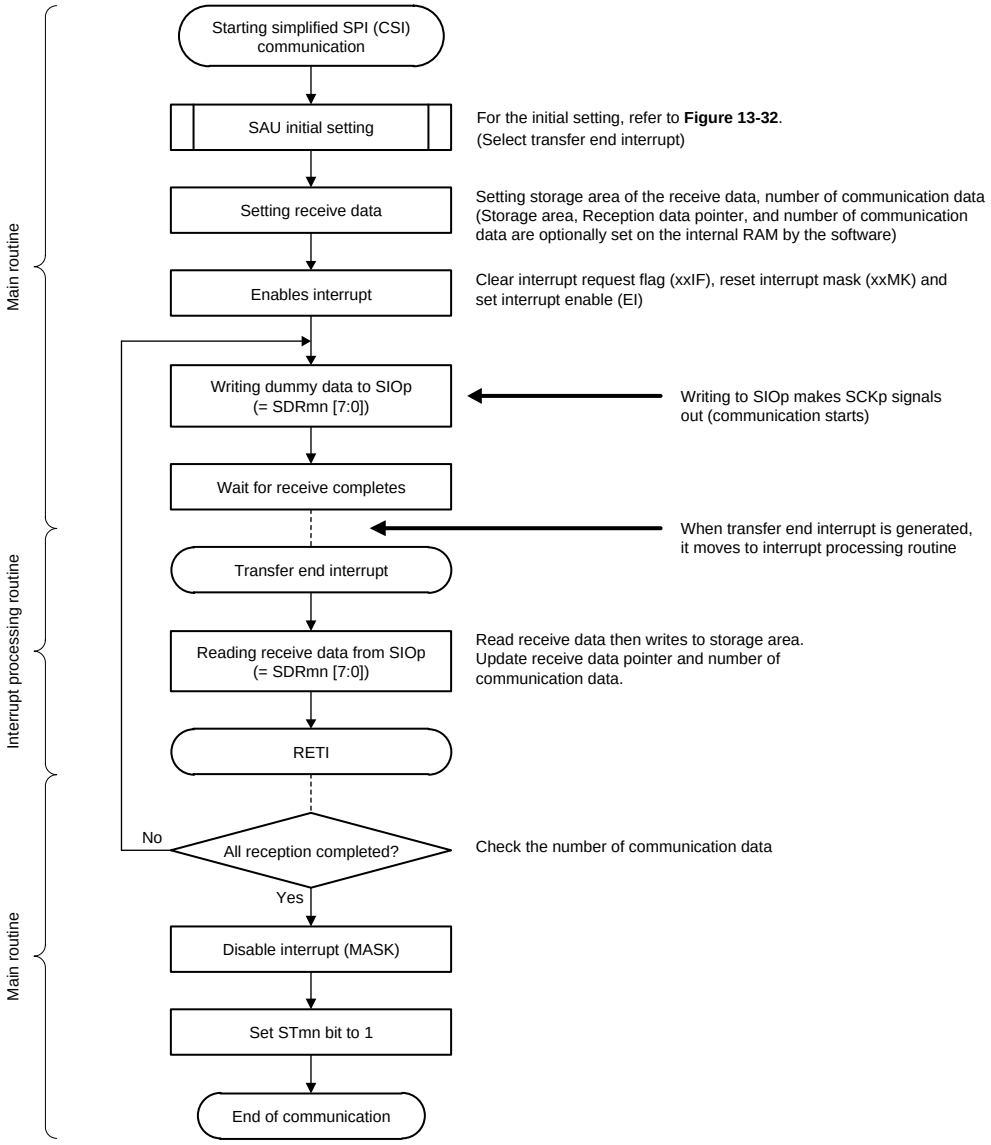
(3) Processing flow (in single-reception mode)

Figure 13-35. Timing Chart of Master Reception (in Single-Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



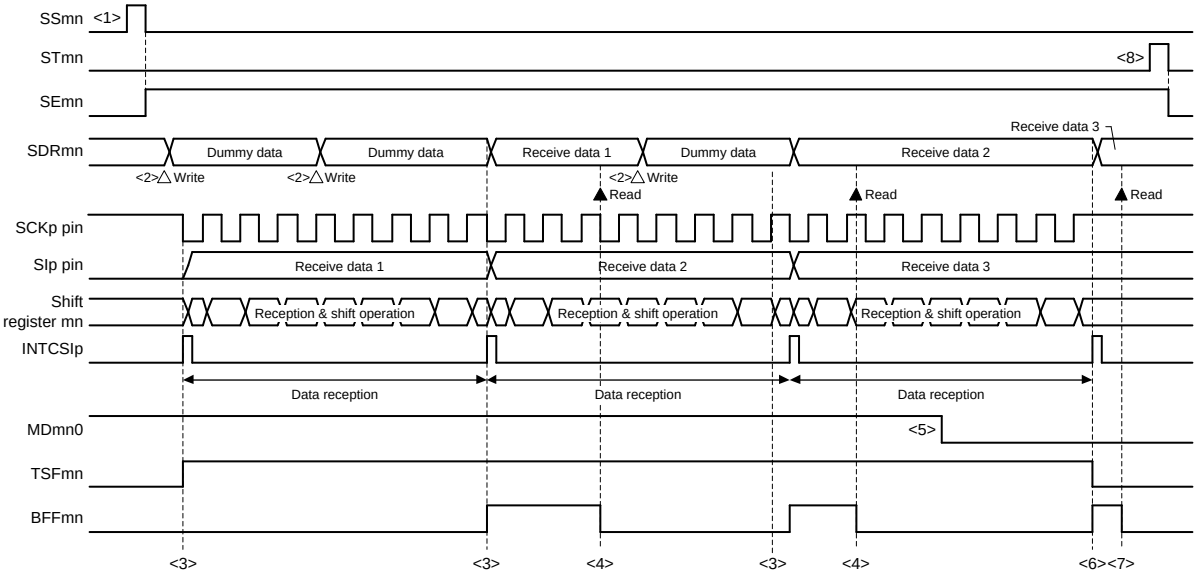
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), p: CSI number (p = 00, 11, 20), mn = 00, 03, 10

Figure 13-36. Flowchart of Master Reception (in Single-Reception Mode)



(4) Processing flow (in continuous reception mode)

Figure 13-37. Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)

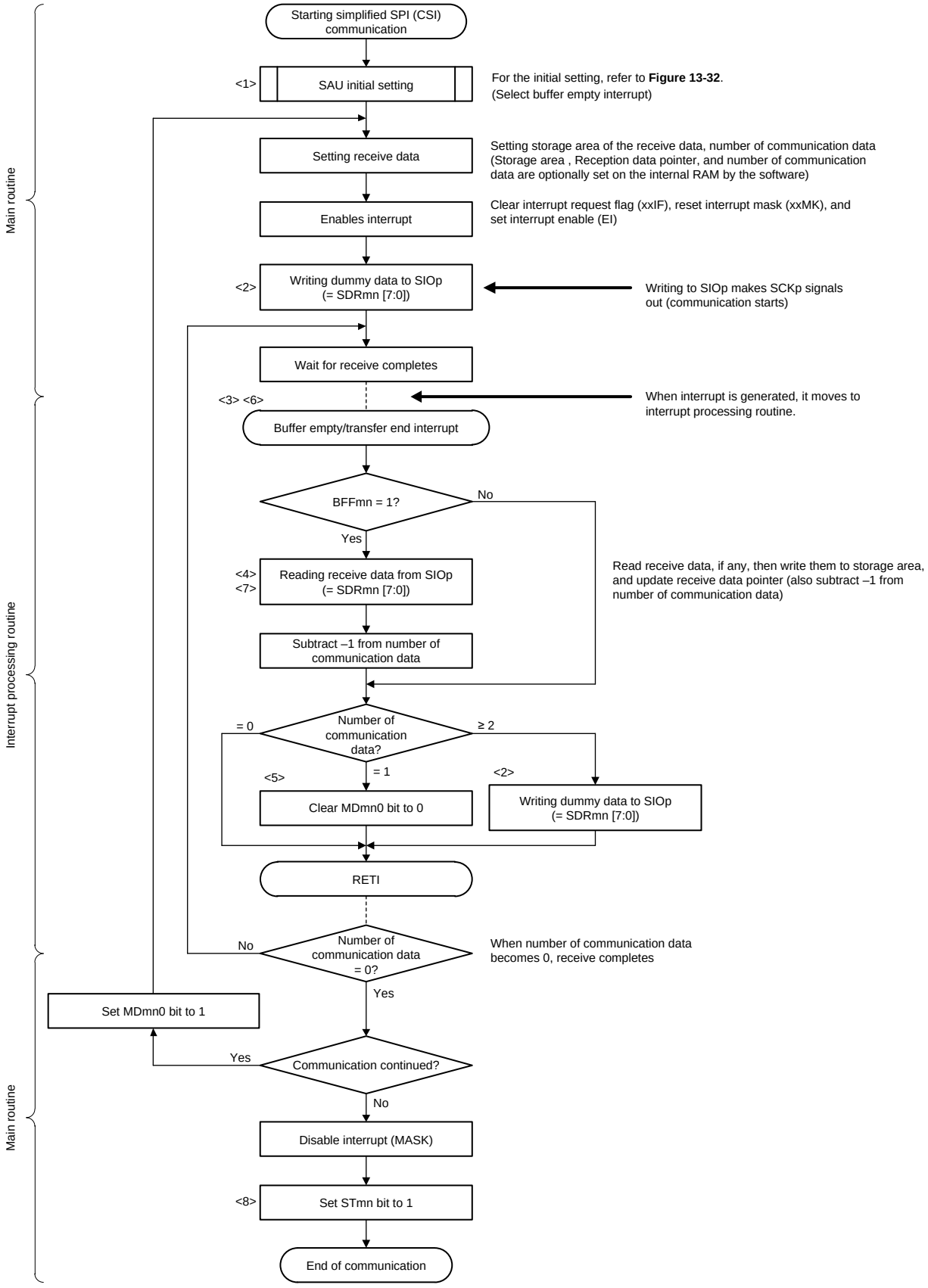


Caution The MDmn0 bit can be rewritten even during operation.
However, rewrite it before receive of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last receive data.

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 13-38 Flowchart of Master Reception (in Continuous Reception Mode)**.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), p: CSI number (p = 00, 11, 20), mn = 00, 03, 10

Figure 13-38. Flowchart of Master Reception (in Continuous Reception Mode)



(Remark is listed on the next page.)

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 13-37 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	CSI11	CSI20
Target channel	Channel 0 of SAU0	Channel 3 of SAU0	Channel 3 of SAU0
Pins used	SCK00, SI00, SO00	SCK11, SI11, SO11	SCK20, SI20, SO20
Interrupt	INTCSI00	INTCSI11	INTCSI20
	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 1]	Max. $f_{CLK}/4$ [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 cycle before the start of the serial clock operation.		
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse		
Data direction	MSB or LSB first		

Note 1. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **CHAPTER 26 ELECTRICAL SPECIFICATIONS (T_A = −40 to +85°C)** and **CHAPTER 27 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C, T_A = −40 to +125°C)**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), p: CSI number (p = 00, 11, 20), mn = 00, 03,10

(1) Register setting

Figure 13-39. Example of Contents of Registers for Master Transmission/Reception of simplified SPI (CSI00, CSI11, CSI20) (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	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	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	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	1	1	0/1	0/1	0	0	0	0	0/1	0	0	0	0	1	1	0/1

Selection of the data and clock phase
(For details about the setting, see **13.3 Registers to Control the Serial Array Unit**.)

Selection of data transfer sequence
- 0: Inputs/outputs data with MSB first
- 1: Inputs/outputs data with LSB first

Setting of data length
- 0: 7-bit data length
- 1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting (Operation clock (f_{MCK}) division setting)							0	Transmit data setting/receive data register							

SIOp

(d) Serial output register m (SOm) … Set only the bit of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm					CKOm3			CKOm0					SOm3	SOm2		SOm0
	0	0	0	0	0/1	1	1	0/1	0	0	0	0	0/1	×	1	0/1

Communication starts when a bit is 1 if the clock phase is non-reverse (the CKPmn bit of the SCRmn = 0). If the clock phase is reversed (CKPmn = 1), communication starts when a bit is 0.

(Remarks are listed on the next page.)

Figure 13-39. Example of Contents of Registers for Master Transmission/Reception of simplified SPI (CSI00, CSI11, CSI20) (2/2)

(e) Serial output enable register m (SOEm) … Set only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm													SOEm3	SOEm2		SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	×	0	0/1

(f) Serial channel start register m (SSm) … Set only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	×	×	0/1

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), p: CSI number (p = 00, 11, 20), mn = 00, 03, 10

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 13-40. Initial Setting Procedure for Master Transmission/Reception

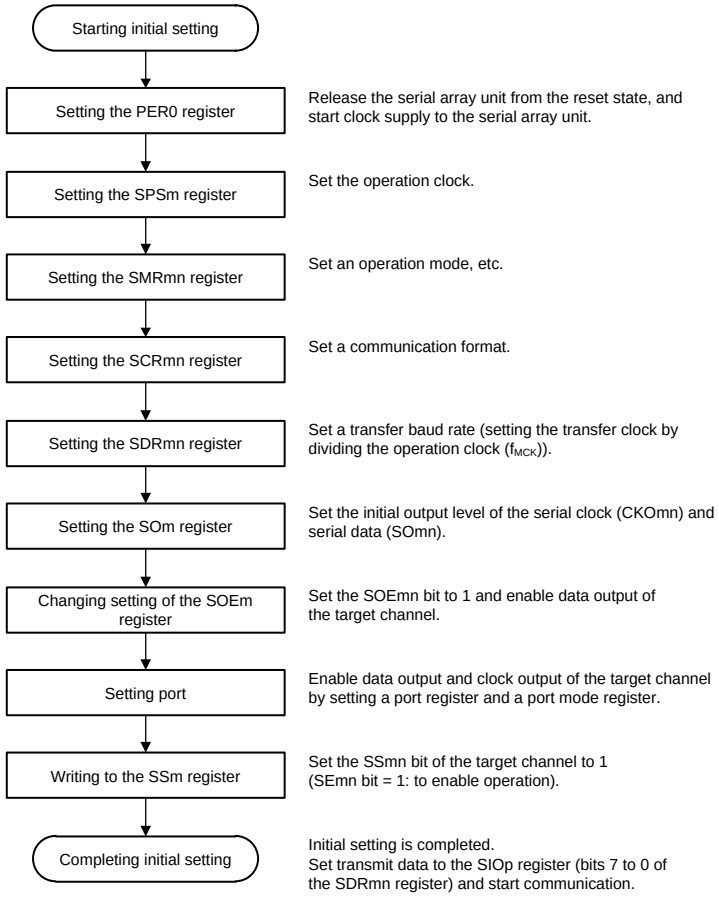


Figure 13-41. Procedure for Stopping Master Transmission/Reception

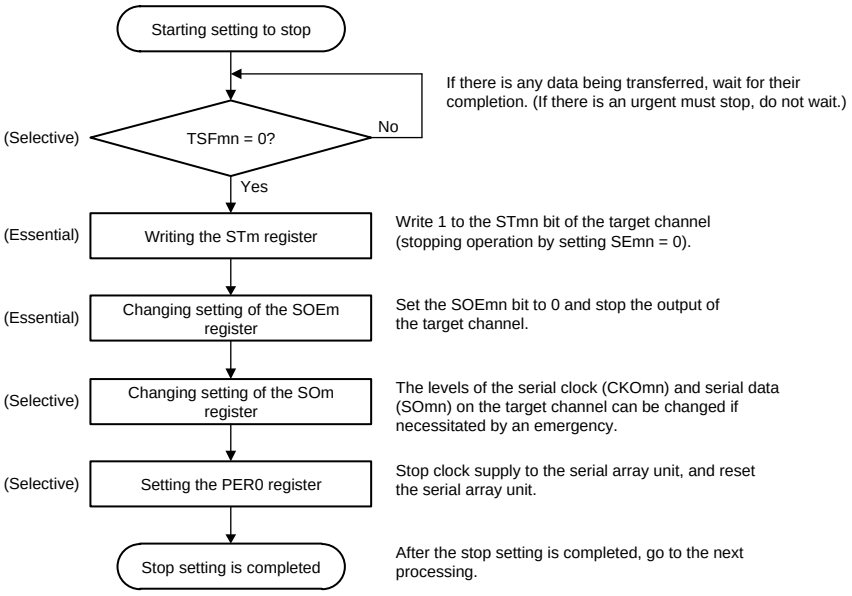
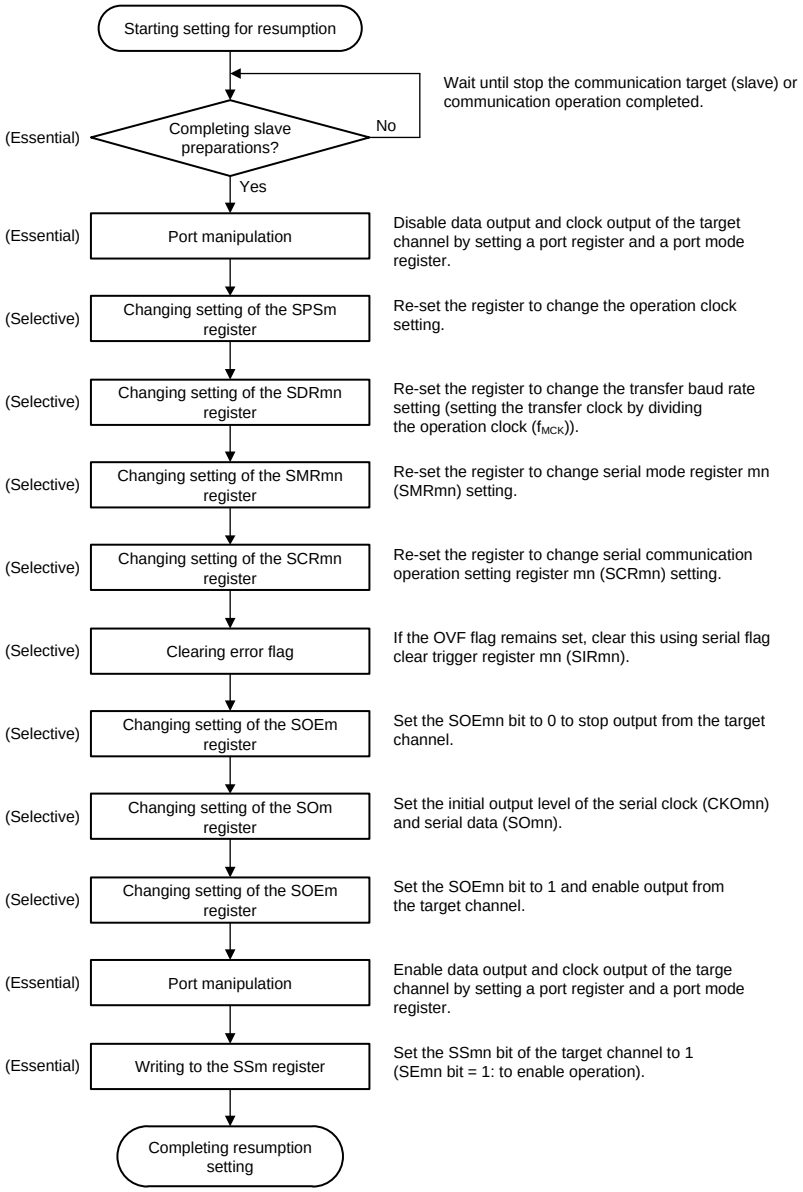
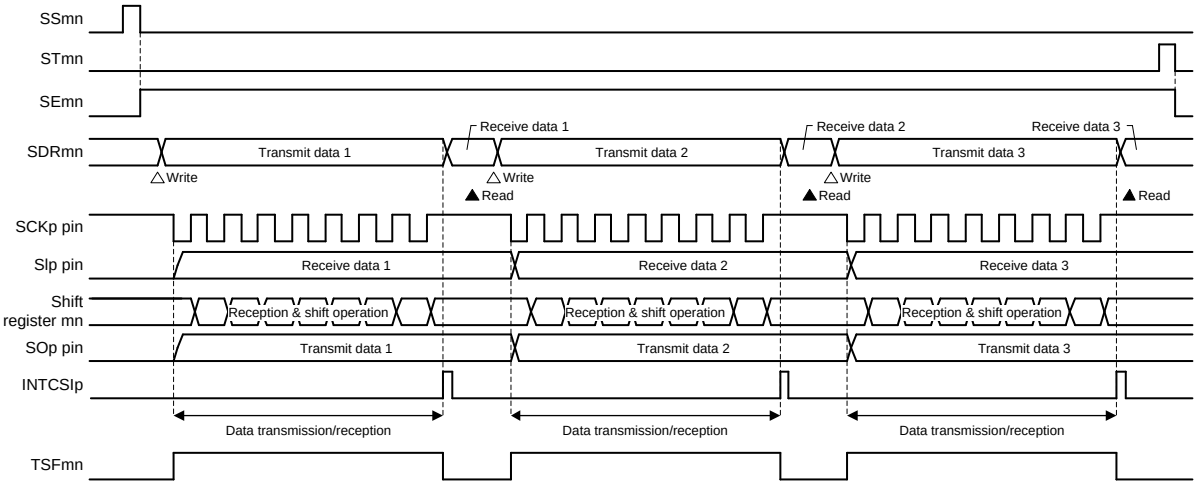


Figure 13-42. Procedure for Resuming Master Transmission/Reception



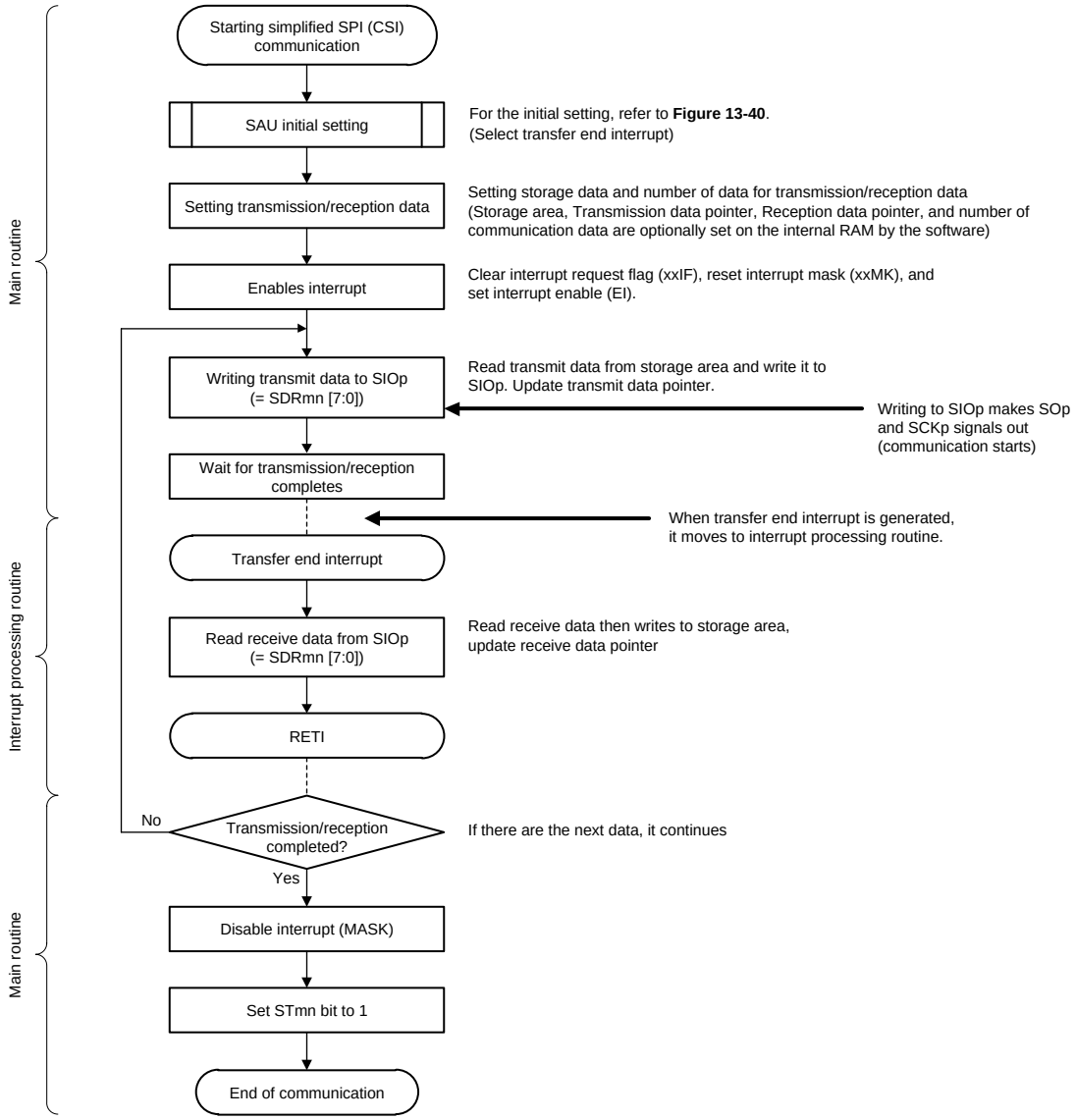
(3) Processing flow (in single-transmission/reception mode)

Figure 13-43. Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



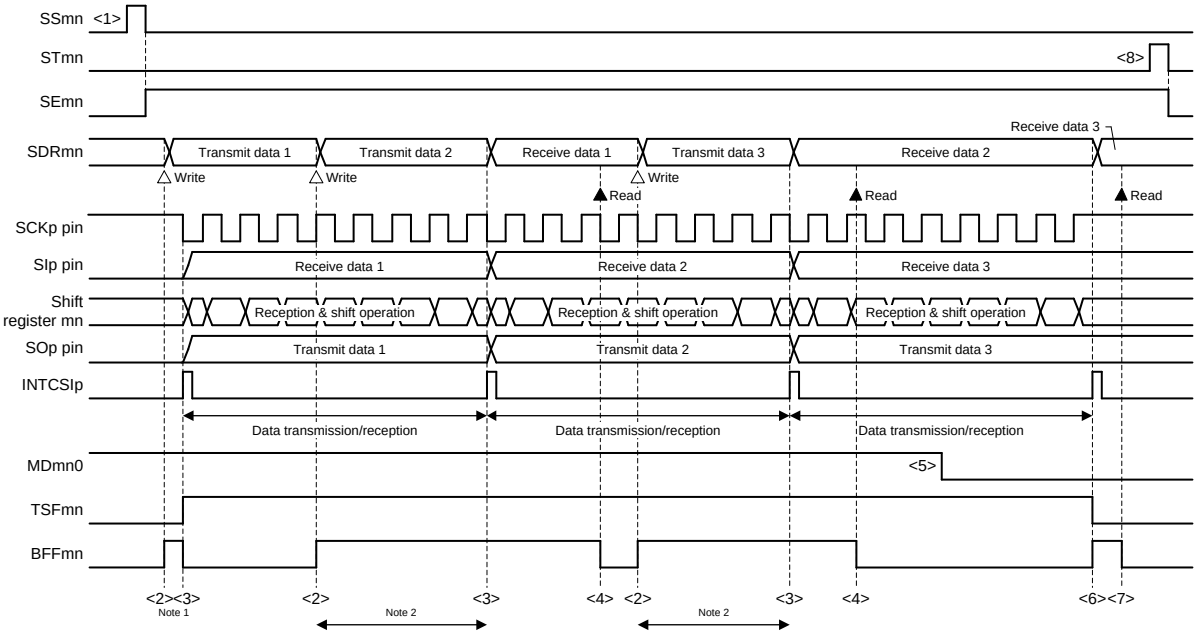
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), p: CSI number (p = 00, 11, 20), mn = 00, 03, 10

Figure 13-44. Flowchart of Master Transmission/Reception (in Single-Transmission/Reception Mode)



(4) Processing flow (in continuous transmission/reception mode)

Figure 13-45. Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Note 1. If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

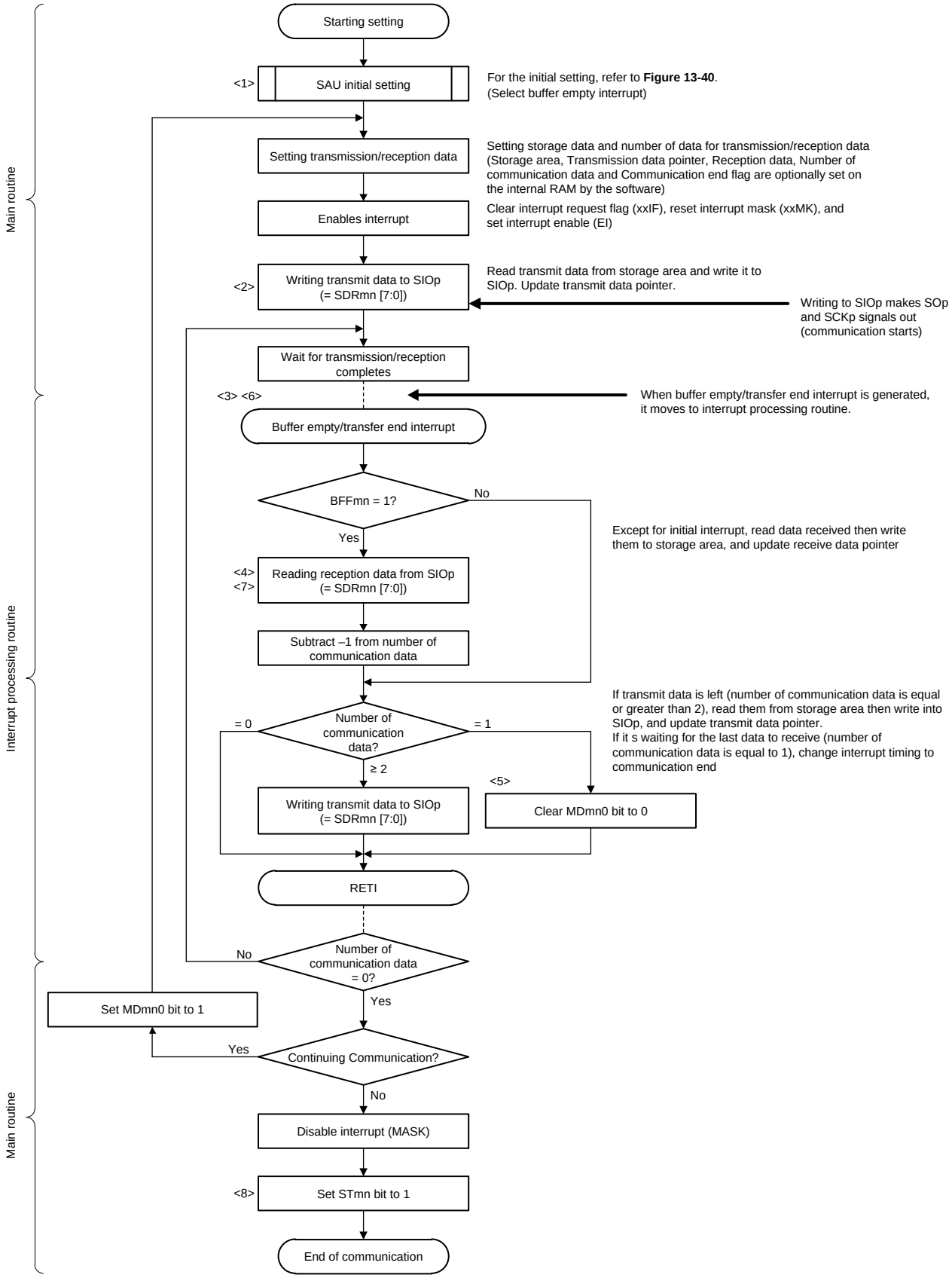
Note 2. The transmit data can be read by reading the SDRmn register during this period. Reading this register does not affect the transfer operation.

Caution The MDmn0 bit of serial mode register mn (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 13-46 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), p: CSI number (p = 00, 11, 20), mn = 00, 03, 10

Figure 13-46. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



(Remark is listed on the next page.)

Remark <1> to <8> in the figure correspond to <1> to <8> in **Figure 13-45 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	CSI11	CSI20
Target channel	Channel 0 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1
Pins used	SCK00, SO00	SCK11, SO11	SCK20, SO20
Interrupt	INTCSI00	INTCSI11	INTCSI20
	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]Note 1, Note 2		
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data output starts from the start of the operation of the serial clock. • DAPmn = 1: Data output starts half a clock cycle before the start of the serial clock operation.		
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse		
Data direction	MSB or LSB first		

Note 1. Because the external serial clock input to the SCK00, SCK11, and SCK20 pins is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz].

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **CHAPTER 26 ELECTRICAL SPECIFICATIONS (T_A = −40 to +85°C)** and **CHAPTER 27 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C, T_A = −40 to +125°C)**.

Remark 1. f_{MCK}: Operation clock frequency of target channel
f_{SCK}: Serial clock frequency

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), mn = 00, 03, 10

(1) Register setting

Figure 13-47. Example of Contents of Registers for Slave Transmission of simplified SPI (CSI00, CSI11, CSI20) (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	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0/1

Operation clock (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	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	1	0	0/1	0/1	0	0	0	0	0/1	0	0	0	0	1	1	0/1

Selection of the data and clock phase
(For details about the setting, see **13.3 Registers to Control the 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) … Set only the bit of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm					CKOm3			CKOm0					SOm3	SOm2		SOm0
	0	0	0	0	×	1	1	×	0	0	0	0	0/1	×	1	0/1

(Remarks are listed on the next page.)

Figure 13-47. Example of Contents of Registers for Slave Transmission of simplified SPI (CSI00, CSI11, CSI20) (2/2)

(e) Serial output enable register m (SOEm) … Set only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm													SOEm3	SOEm2		SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	×	0	0/1

(f) Serial channel start register m (SSm) … Set only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	×	×	0/1

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), p: CSI number (p = 00, 11, 2), mn = 00, 03, 10

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 13-48. Initial Setting Procedure for Slave Transmission

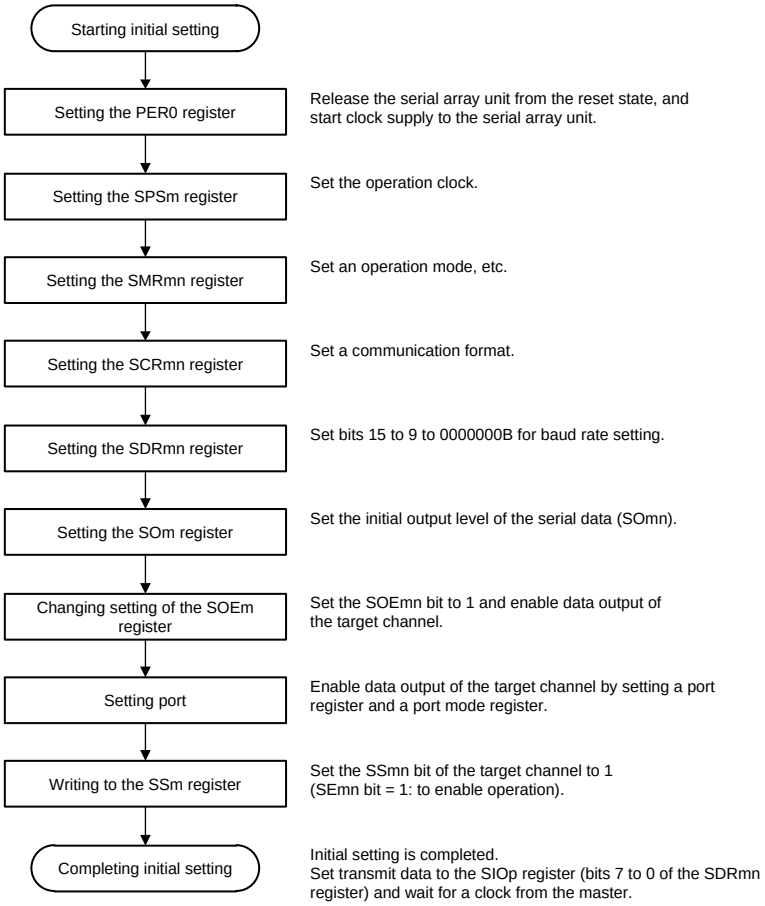


Figure 13-49. Procedure for Stopping Slave Transmission

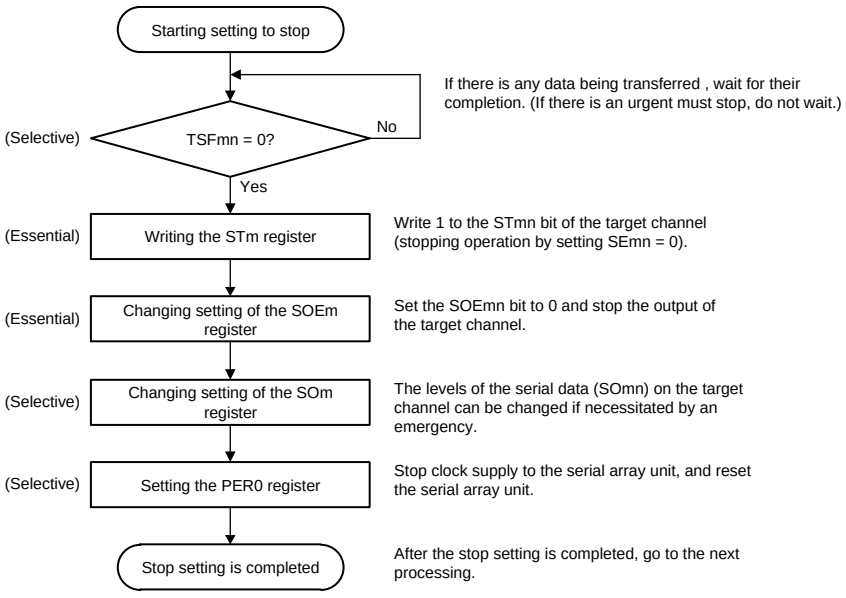
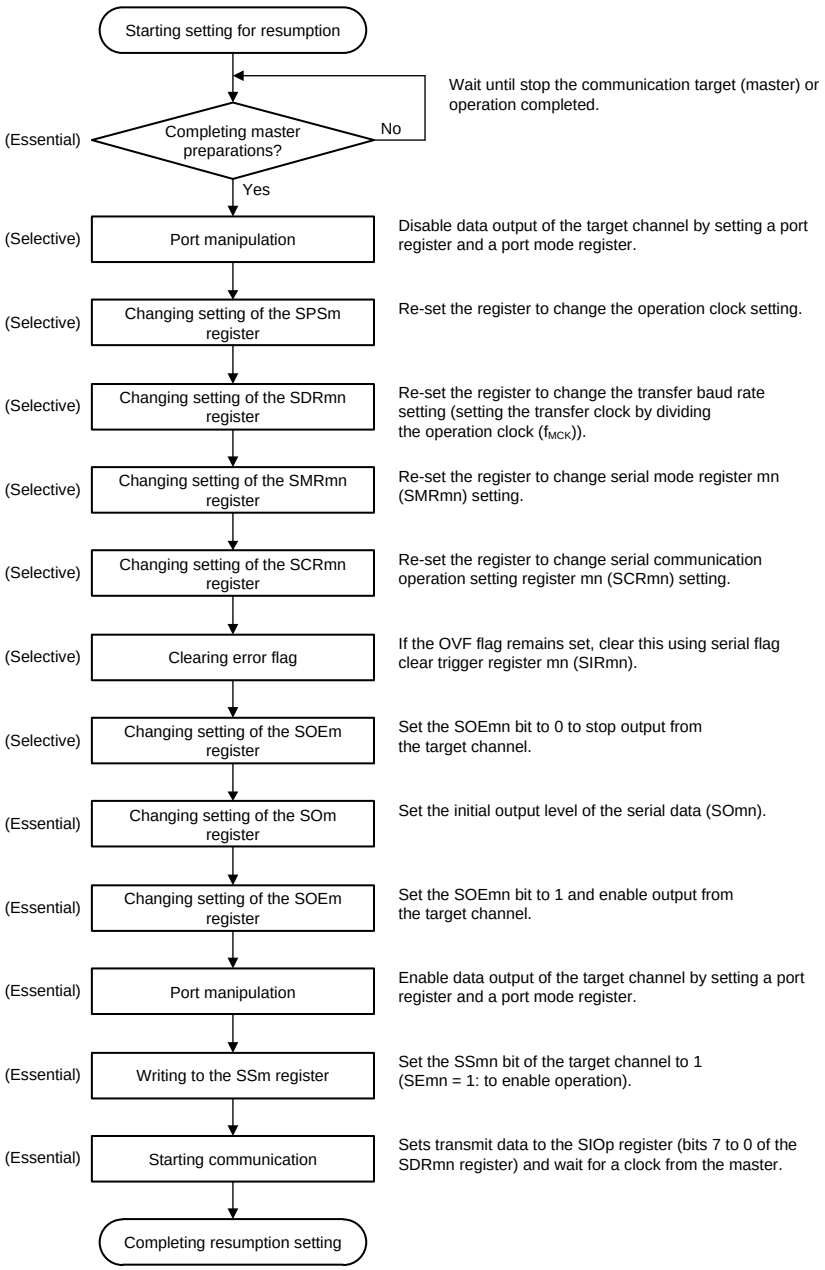


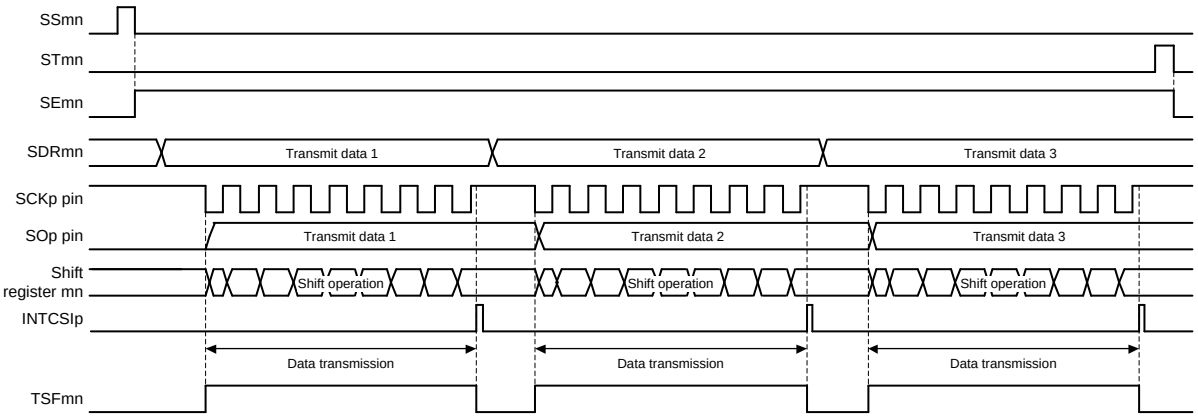
Figure 13-50. Procedure for Resuming Slave Transmission



Remark If PER0 is rewritten while stopping the communication to reset the serial array unit, wait until the communication target (master) stops or communication finishes, and then perform initialization instead of restarting the communication.

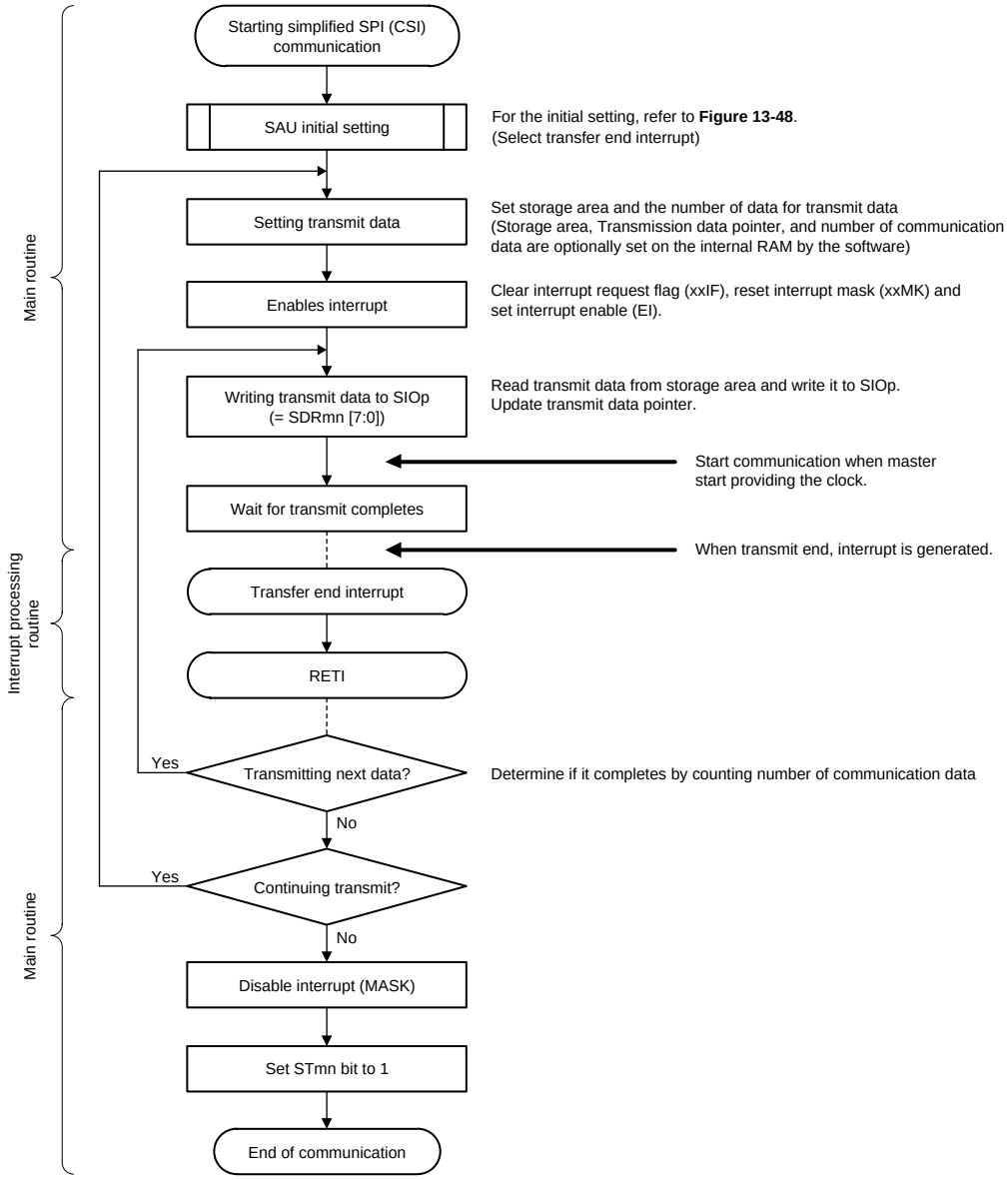
(3) Processing flow (in single-transmission mode)

Figure 13-51. Timing Chart of Slave Transmission (in Single-Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)



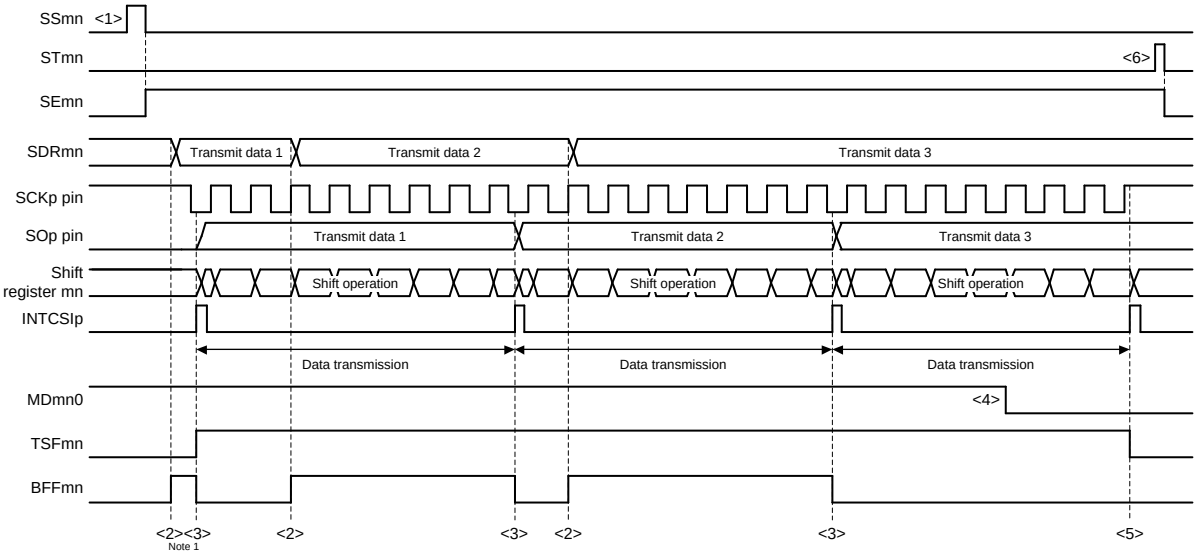
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), p: CSI number (p = 00, 11, 20), mn = 00, 03, 10

Figure 13-52. Flowchart of Slave Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 13-53. Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)

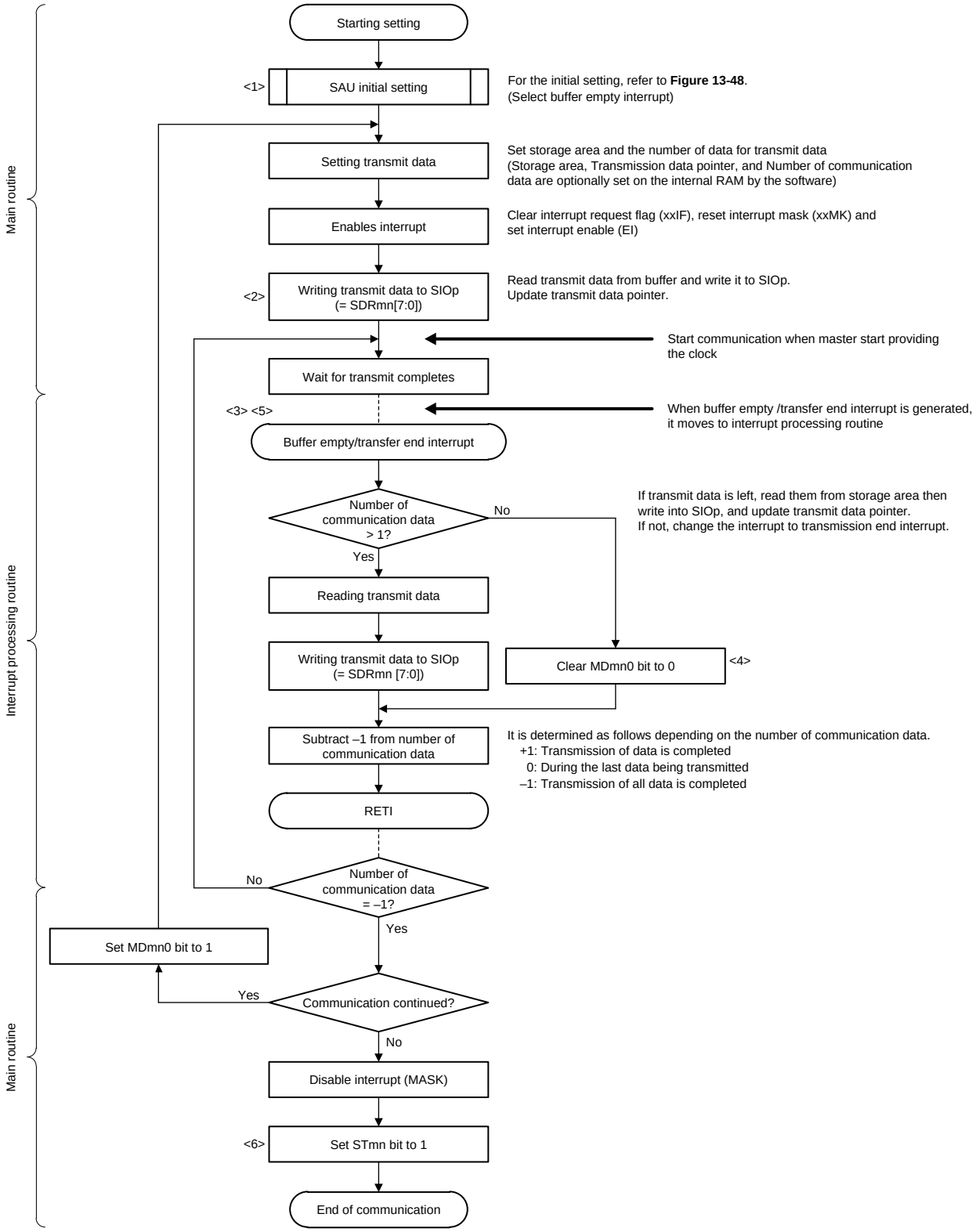


Note 1. If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), p: CSI number (p = 00, 11, 20), mn = 00, 03, 10

Figure 13-54. Flowchart of Slave Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 13-53 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	CSI11	CSI20
Target channel	Channel 0 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1
Pins used	SCK00, SI00	SCK11, SI11	SCK20, SI20
Interrupt	INTCSI00	INTCSI11	INTCSI20
	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]Note 1, Note 2		
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data input starts from the start of the operation of the serial clock. • DAPmn = 1: Data input starts half a clock cycle before the start of the serial clock operation.		
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse		
Data direction	MSB or LSB first		

Note 1. Because the external serial clock input to the SCK00, SCK11, and SCK20 pins is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz].

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **CHAPTER 26 ELECTRICAL SPECIFICATIONS (T_A = −40 to +85°C)** and **CHAPTER 27 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C, T_A = −40 to +125°C)**.

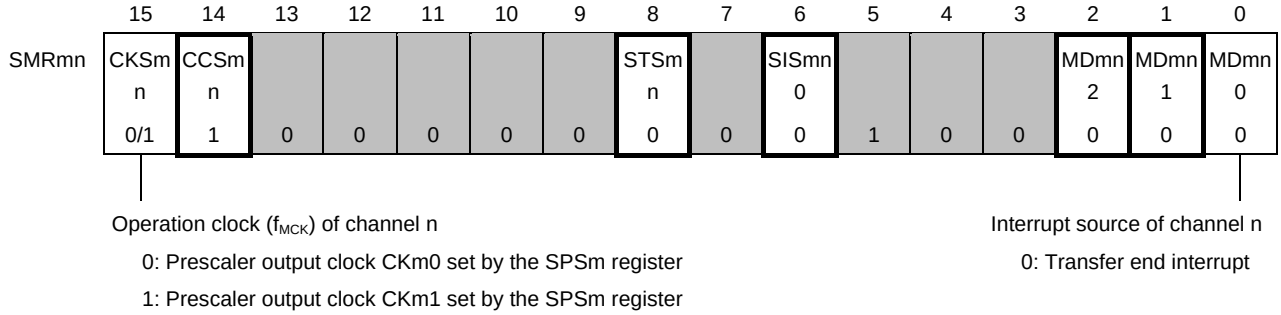
Remark 1. f_{MCK}: Operation clock frequency of target channel
f_{SCK}: Serial clock frequency

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), mn = 00, 03, 10

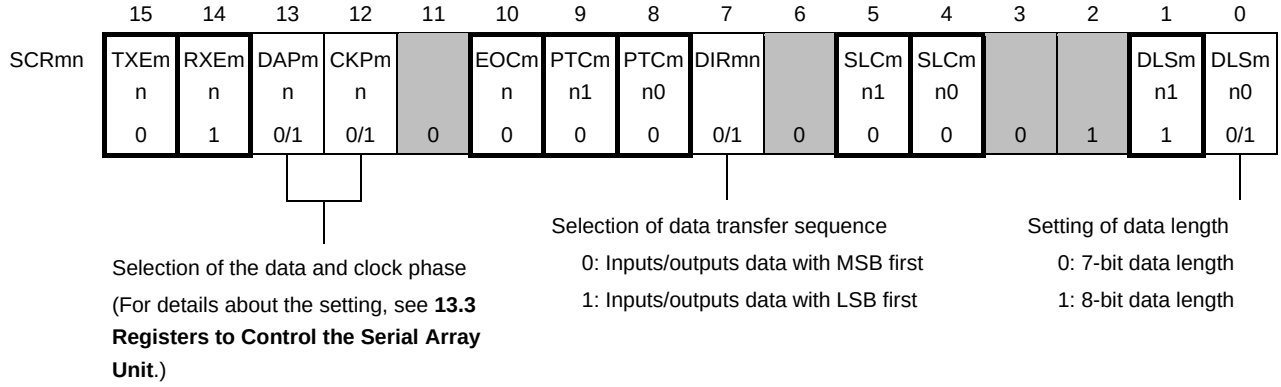
(1) Register setting

Figure 13-55. Example of Contents of Registers for Slave Reception of simplified SPI (CSI00, CSI11, CSI20) (1/2)

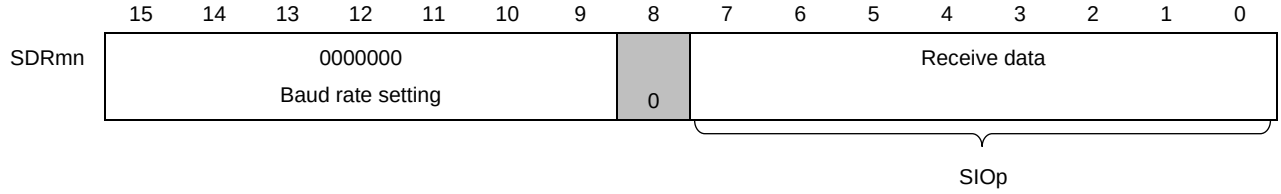
(a) Serial mode register mn (SMRmn)



(b) Serial communication operation setting register mn (SCRmn)



(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



(d) Serial output register m (SOm) … This register is not used in this mode.

(Remarks are listed on the next page.)

Figure 13-55. Example of Contents of Registers for Slave Reception of simplified SPI (CSI00, CSI11, CSI20) (2/2)

(e) Serial output enable register m (SOEm) … This register is not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm													SOEm3	SOEm2		SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	×	×	0	×

(f) Serial channel start register m (SSm) … Set only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	×	×	0/1

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), p: CSI number (p = 00, 11, 20), mn = 00, 03, 10

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 13-56. Initial Setting Procedure for Slave Reception

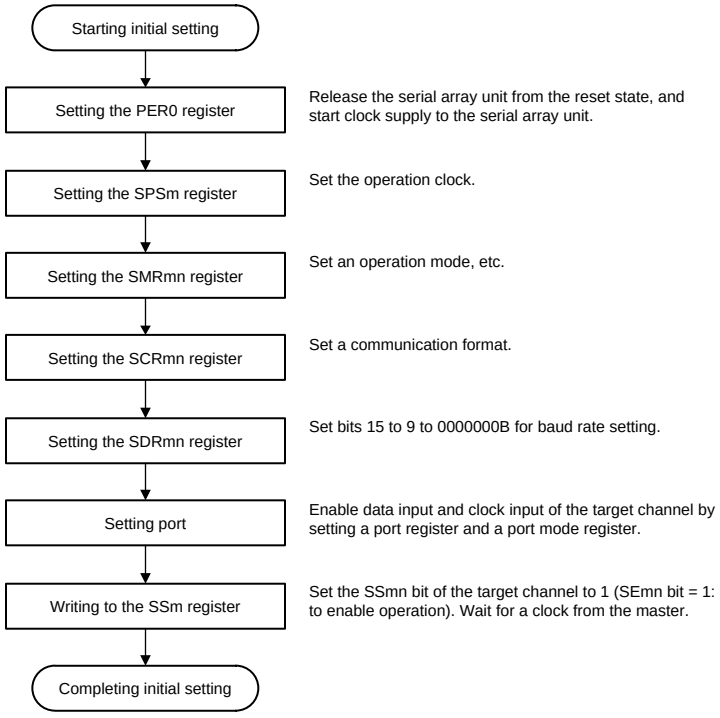


Figure 13-57. Procedure for Stopping Slave Reception

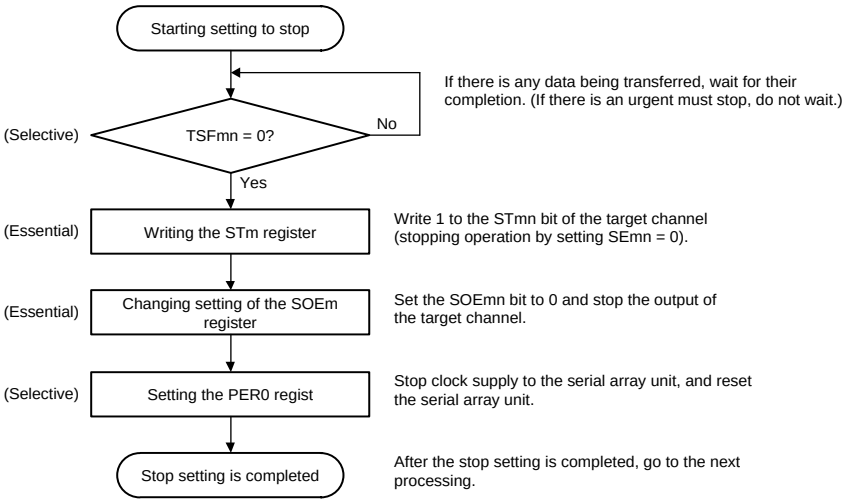
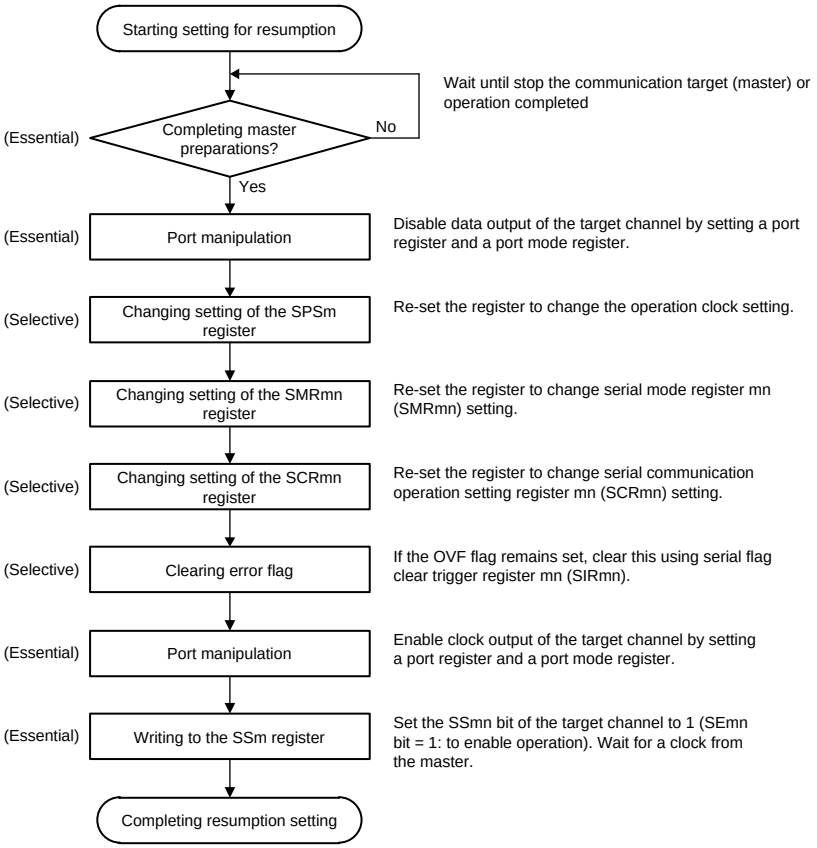


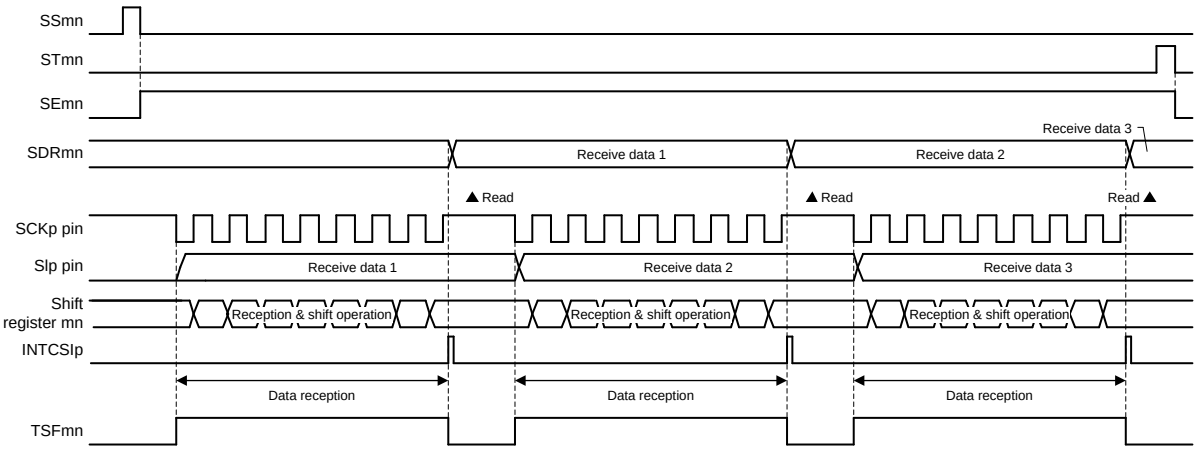
Figure 13-58. Procedure for Resuming Slave Reception



Remark If PER0 is rewritten while stopping the communication to reset the serial array unit, wait until the communication target (master) stops or communication finishes, and then perform initialization instead of restarting the communication.

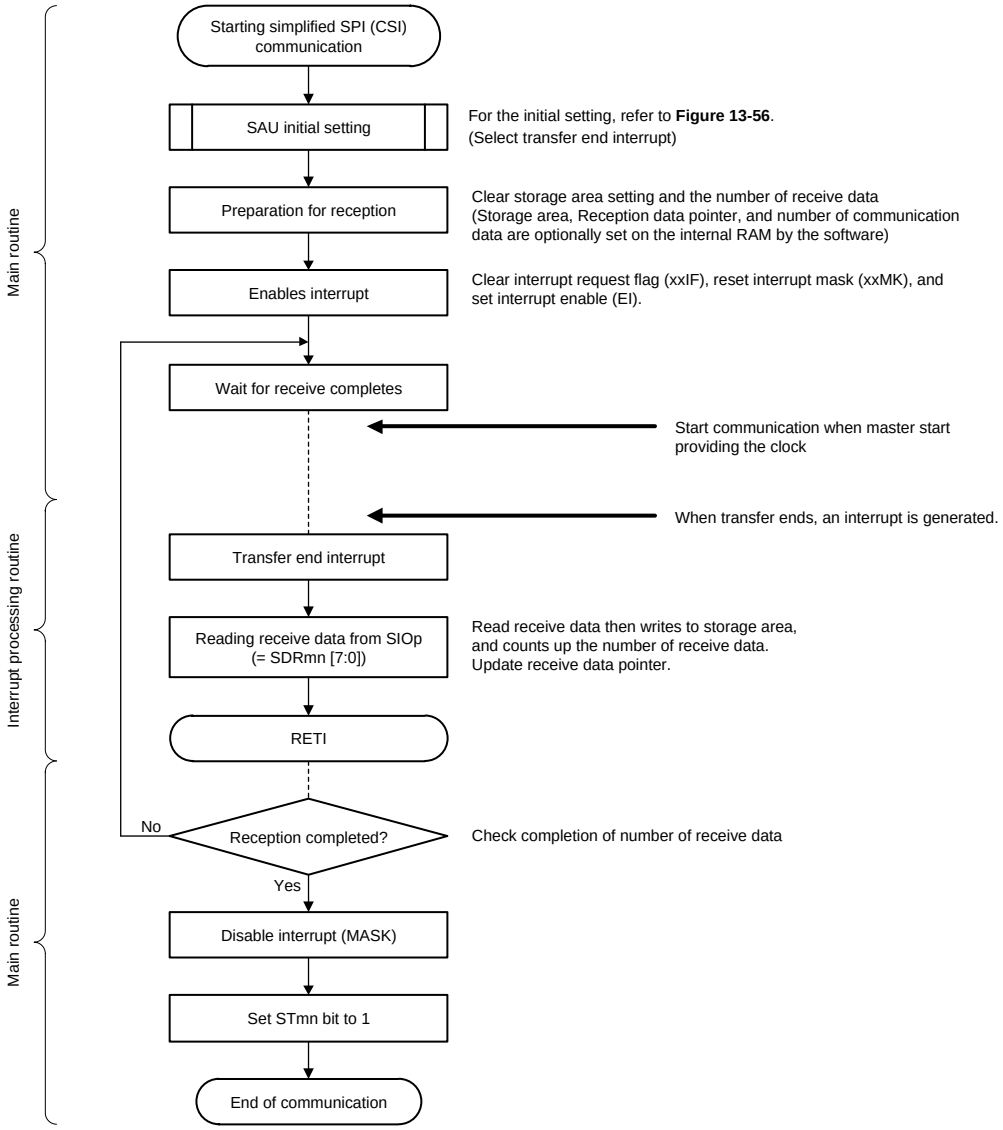
(3) Processing flow (in single-reception mode)

Figure 13-59. Timing Chart of Slave Reception (in Single-Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), p: CSI number (p = 00, 11, 20), mn = 00, 03, 10

Figure 13-60. Flowchart of Slave Reception (in Single-Reception Mode)



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	CSI11	CSI20
Target channel	Channel 0 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1
Pins used	SCK00, SI00, SO00	SCK11, SI11, SO11	SCK20, SI20, SO20
Interrupt	INTCSI00	INTCSI11	INTCSI20
	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]Note 1, Note 2		
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data I/O starts at the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock cycle before the start of the serial clock operation.		
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse		
Data direction	MSB or LSB first		

Note 1. Because the external serial clock input to the SCK00, SCK11, and SCK20 pins is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [Hz].

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **CHAPTER 26 ELECTRICAL SPECIFICATIONS (T_A = −40 to +85°C)** and **CHAPTER 27 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C, T_A = −40 to +125°C)**.

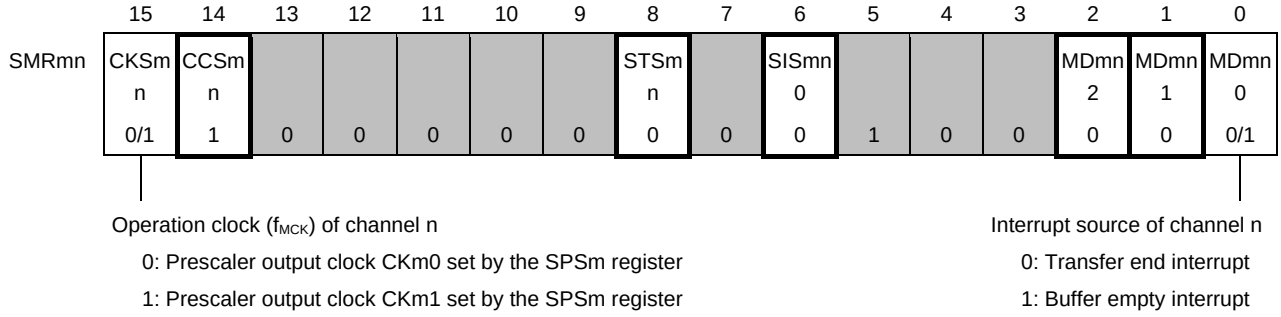
Remark 1. f_{MCK}: Operation clock frequency of target channel
f_{SCK}: Serial clock frequency

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), mn = 00, 03, 10

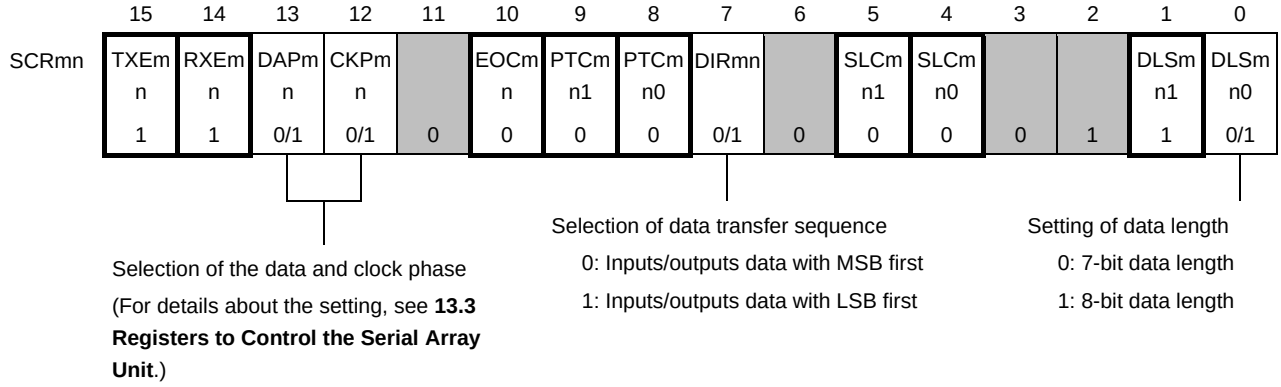
(1) Register setting

Figure 13-61. Example of Contents of Registers for Slave Transmission/Reception of simplified SPI (CSI00, CSI11, CSI20) (1/2)

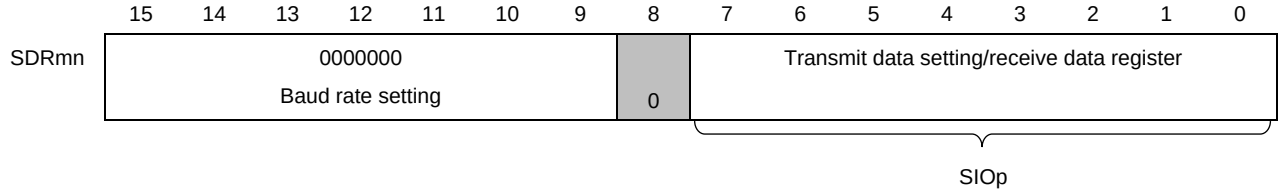
(a) Serial mode register mn (SMRmn)



(b) Serial communication operation setting register mn (SCRmn)



(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



(d) Serial output register m (SOm) … Set only the bit of the target channel.

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

Figure 13-61. Example of Contents of Registers for Slave Transmission/Reception of simplified SPI (CSI00, CSI11, CSI20) (2/2)

(e) Serial output enable register m (SOEm) … Set only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm													SOEm3	SOEm2		SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	×	0	0/1

(f) Serial channel start register m (SSm) … Set only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	×	×	0/1

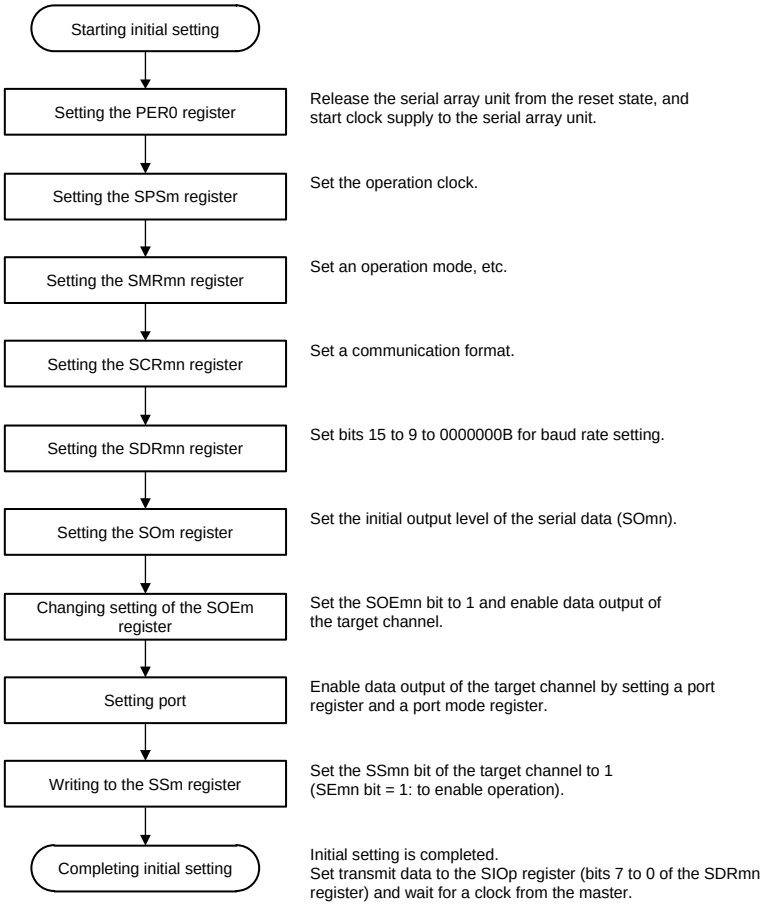
Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), p: CSI number (p = 00, 11, 20), mn = 00, 03, 10

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 13-62. Initial Setting Procedure for Slave Transmission/Reception



Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

Figure 13-63. Procedure for Stopping Slave Transmission/Reception

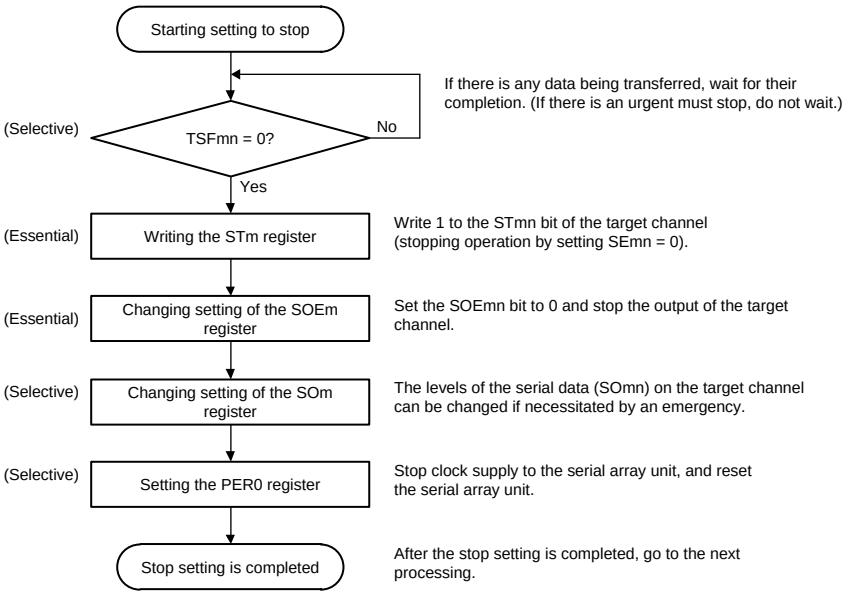
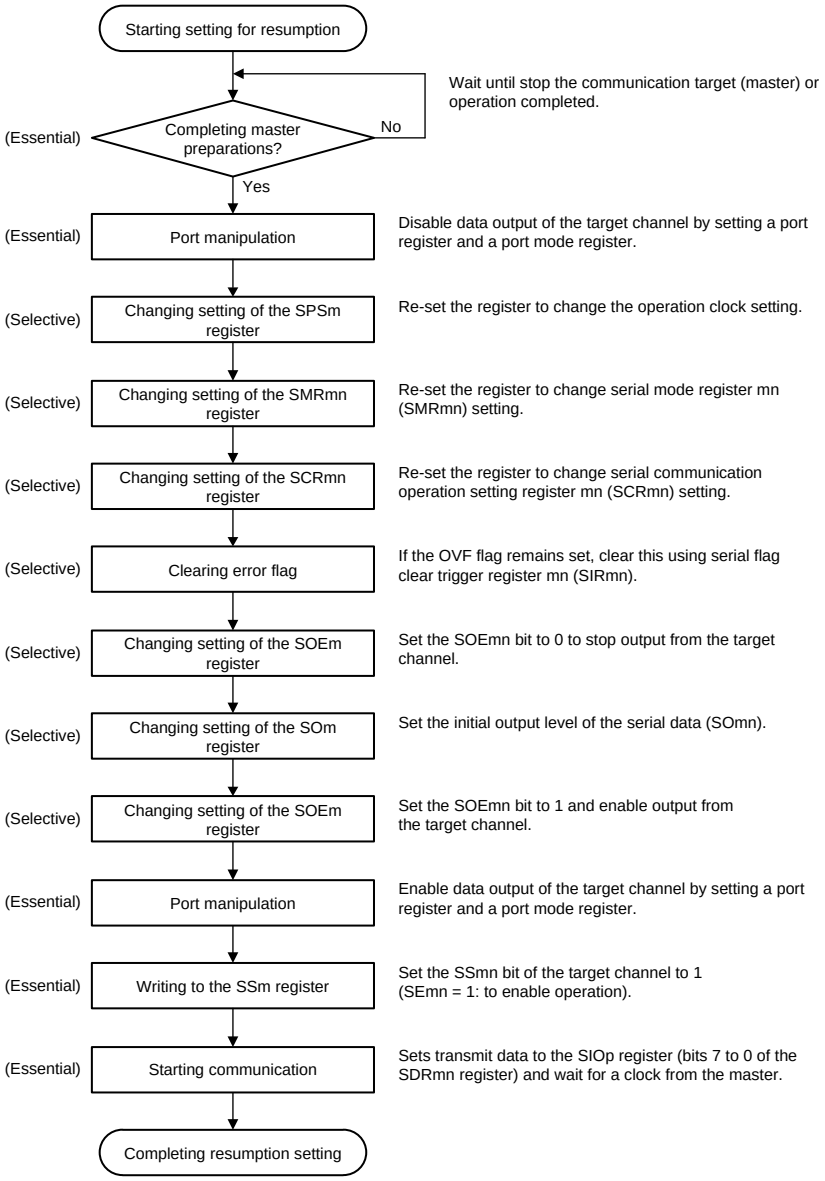


Figure 13-64. Procedure for Resuming Slave Transmission/Reception

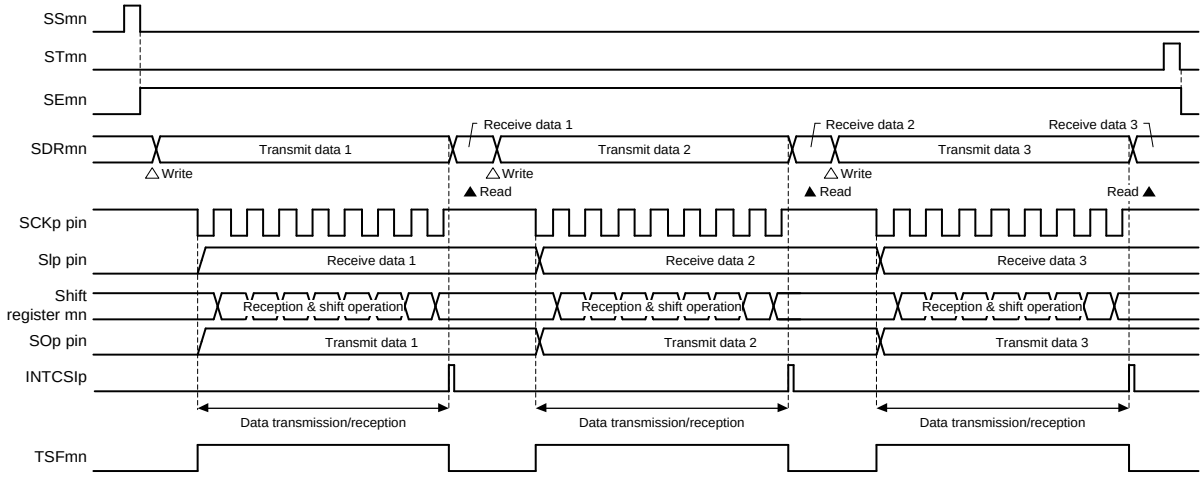


Caution 1. Be sure to set transmit data to the SIOp register before the clock from the master is started.

Caution 2. If PER0 is rewritten while stopping the communication to reset the serial array unit, wait until the communication target (master) stops or communication finishes, and then perform initialization instead of restarting the communication.

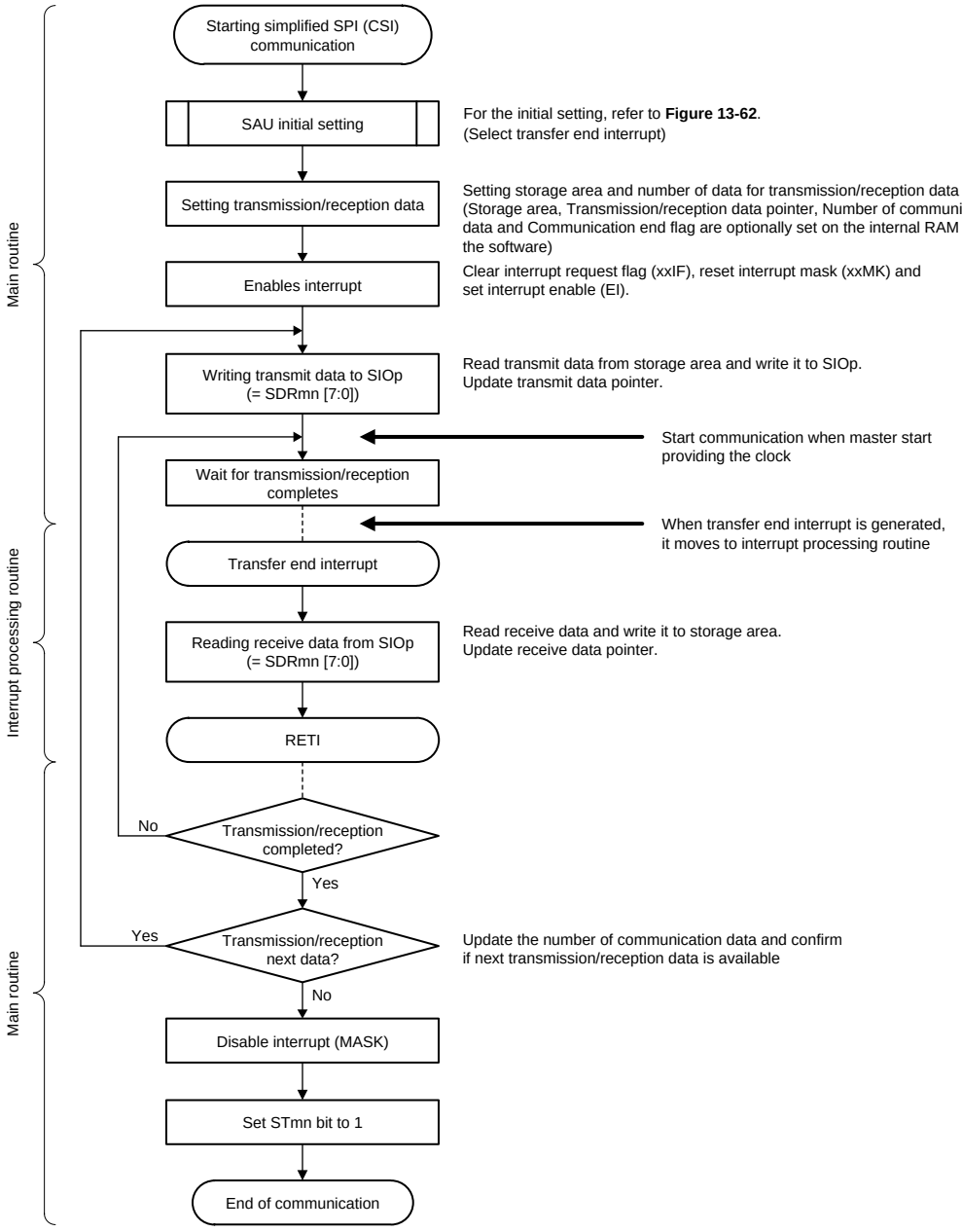
(3) Processing flow (in single-transmission/reception mode)

Figure 13-65. Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), p: CSI number (p = 00, 11, 20), mn = 00, 03, 10

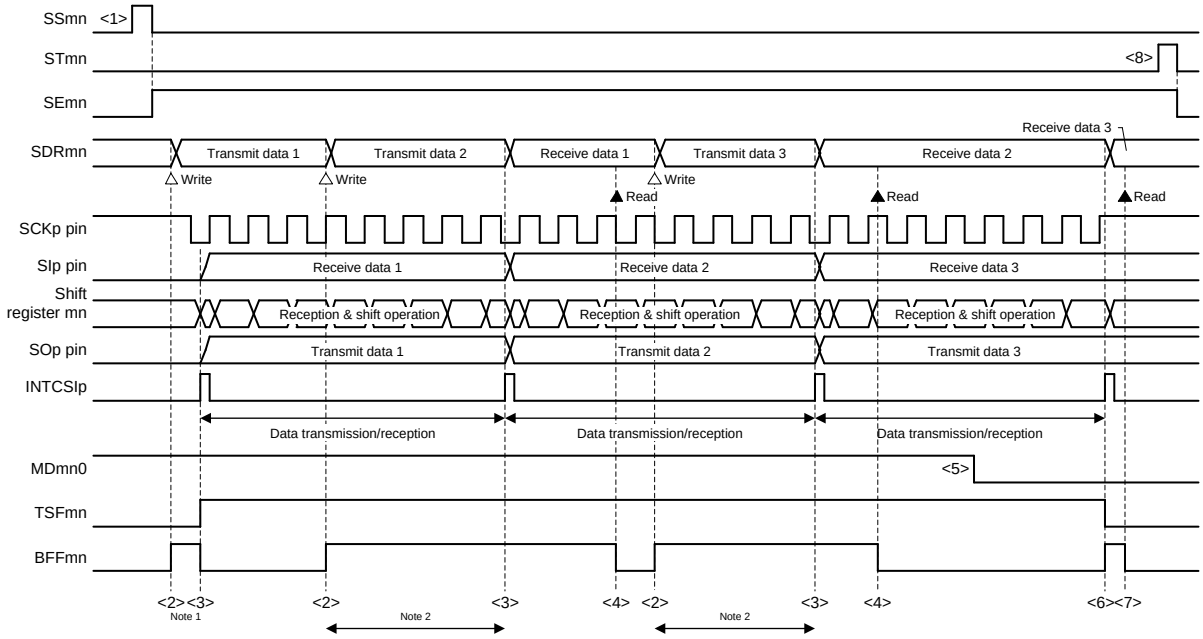
Figure 13-66. Flowchart of Slave Transmission/Reception (in Single-Transmission/Reception Mode)



Caution Be sure to set transmit data to the SIOp register before the clock from the master is started.

(4) Processing flow (in continuous transmission/reception mode)

Figure 13-67. 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.

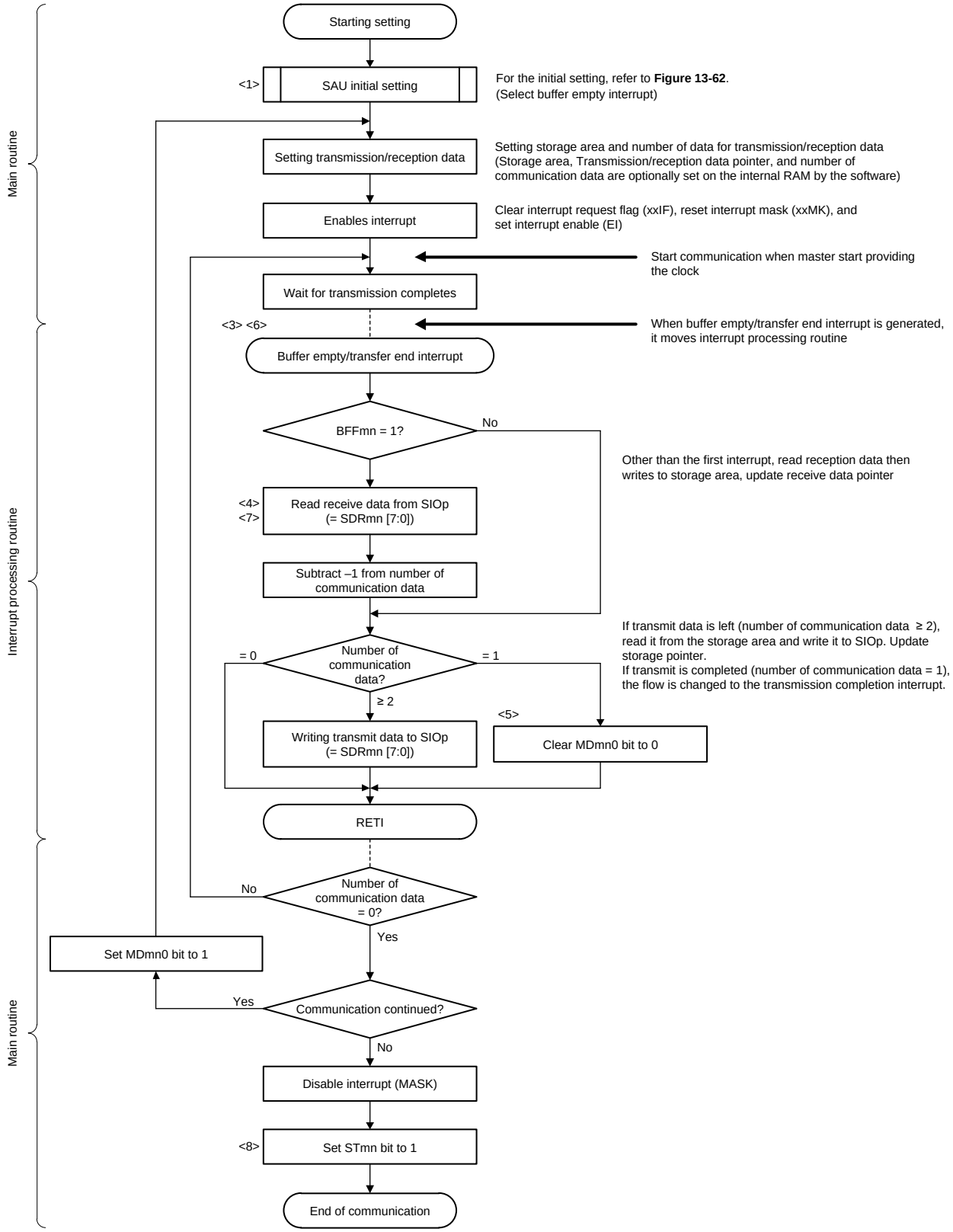
Note 2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

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 13-68 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), p: CSI number (p = 00, 11, 20), mn = 00, 03, 10

Figure 13-68. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)



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

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 13-67 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)**.

13.5.7 Calculating Transfer Clock Frequency

The transfer clock frequency for simplified SPI (CSI00, CSI11, CSI20) communication can be calculated by the following expressions.

(1) Master

$$(\text{Transfer clock frequency}) = \{\text{Operation clock } (f_{MCK}) \text{ frequency of target channel}\} \div (\text{SDRmn}[15:9] + 1) \div 2 [\text{Hz}]$$

(2) Slave

$$(\text{Transfer clock frequency}) = \{\text{Frequency of serial clock (SCK) supplied by master}\}^{\text{Note 1}} \quad [\text{Hz}]$$

Note 1. 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 1]	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 16 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	16 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	8 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	4 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	2 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	500 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	250 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	125 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	62.5 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	31.25 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	15.63 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	7.81 kHz
	X	X	X	X	1	1	0	0	$f_{CLK}/2^{12}$	3.91 kHz
	X	X	X	X	1	1	0	1	$f_{CLK}/2^{13}$	1.95 kHz
	X	X	X	X	1	1	1	0	$f_{CLK}/2^{14}$	977 Hz
	X	X	X	X	1	1	1	1	$f_{CLK}/2^{15}$	488 Hz
1	0	0	0	0	X	X	X	X	f_{CLK}	16 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	8 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	4 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	2 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	500 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	250 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	125 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	62.5 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	31.25 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	15.63 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	7.81 kHz
	1	1	0	0	X	X	X	X	$f_{CLK}/2^{12}$	3.91 kHz
	1	1	0	1	X	X	X	X	$f_{CLK}/2^{13}$	1.95 kHz
	1	1	1	0	X	X	X	X	$f_{CLK}/2^{14}$	977 Hz
	1	1	1	1	X	X	X	X	$f_{CLK}/2^{15}$	488 Hz
Other than above									Setting prohibited	

(Note 1 and Remarks are listed on the next page.)

Note 1. 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. X: Don't care

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), mn = 00, 03,10

13.5.8 Procedure for Processing Errors that Occurred During Simplified SPI (CSI00, CSI11, CSI20) Communication

The procedure for processing errors that occurred during simplified SPI (CSI00, CSI11, CSI20) communication is described in **Figure 13-69**.

Figure 13-69. Processing Procedure in Case of Overrun Error

Software Manipulation	State of the Hardware	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		The error type is identified and the read value is used to clear the error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	The error flag is cleared.	The error only during reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), mn = 00, 03,10

13.6 Operation of UART (UART0 to UART02) Communication

This is a start-stop synchronization communication function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex asynchronous communication UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel).

[Data transmission/reception]

- Data length of 7, 8, or 9 bits
- MSB/LSB first selectable
- Level setting of transmit/receive data (selecting whether to reverse the level)
- Parity bit appending and parity check functions
- Stop bit appending, stop bit check function

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

The ISC register can be used to set up the input signal on the RxD0 pin of UART0 as an external interrupt input or as a timer input for the timer array unit. The input pulse interval measurement mode of the timer array unit can then be used to measure the width at the baud rate of the other party in communications and make the required adjustments in response.

UART0 uses channels 0 and 1 of SAU0.

UART1 uses channels 2 and 3 of SAU0.

UART2 uses channels 0 and 1 of SAU1.

- 10-pin products

Unit	Channel	Used as simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	—	—
	3	—		—

- 16-pin products

Unit	Channel	Used as simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11

- 20-, 24-, and 32-pin products

Unit	Channel	Used as simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	—		—

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. At this time, however, channel 2, 3, or other channels of the same unit can be used for a function other than UART0, such as CSI11, UART1, and IIC11.

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 two types of communication operations.

- UART transmission (See **13.6.1**)
- UART reception (See **13.6.2**)

13.6.1 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	UART2
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1
Pins used	TxD0	TxD1	TxD2
Interrupt	INTST0	INTST1	INTST2
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.		
Error detection flag	None		
Transfer data length	7, 8, or 9 bits[Note 1]		
Transfer rate[Note 2]	Max. f_{MCK}/6 [bps] (SDRmn[15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]		
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)		
Parity bit	The following selectable • No parity bit • Appending 0 parity • Appending even parity • Appending odd parity		
Stop bit	The following selectable • Appending 1 bit • Appending 2 bit		
Data direction	MSB or LSB first		

Note 1. Only UART0 can be specified for the 9-bit data length.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **CHAPTER 26 ELECTRICAL SPECIFICATIONS (T_A = −40 to +85°C)** and **CHAPTER 27 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C, T_A = −40 to +125°C)**.

Remark 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10

(1) Register setting

Figure 13-70. Example of Contents of Registers for UART Transmission of UART (UART0 to UART2) (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	CCSmn												MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	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	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	1	0	0	0	0	0	0/1	0/1	0/1	0	0/1	0/1	0	1	0/1[Note 1]	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: Outputs data with MSB first
- 1: 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) … Set only the bit of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOLm														SOLm2		SOLm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0/1[Note 4]	0	0/1

- 0: Non-reverse (normal) transmission
- 1: Reverse transmission

(Note 1, Note 2, Note 4, and Remarks are listed on the next page.)

Figure 13-70. Example of Contents of Registers for UART Transmission of UART (UART0 to UART2) (2/2)

(e) Serial output register m (SOm) … Set only the bit of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm					CKOm3			CKOm0					SOm3	SOm2		SOm0
	0	0	0	0	×Note 4	1	1	×	0	0	0	0	×Note 4	0/1 Note 3, Note 4	1	0/1Note 3

0: Serial data output value is 0
1: Serial data output value is 1

(f) Serial output enable register m (SOEm) … Set only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm													SOEm3	SOEm2		SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	×Note 4	0/1Note 4	0	0/1

(g) Serial channel start register m (SSm) … Set only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	×Note 4	0/1Note 4	×	0/1

Note 1. The SCR00 and SCR01 registers only. This bit is fixed to 1 for the other registers.

Note 2. When UART0 performs 9-bit communication, bits 0 to 8 of the SDRm0 register are used as the transmission data specification area. Only UART0 can be specified for the 9-bit data length.

Note 3. 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.

Note 4. Serial array unit 0 only.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 2), mn = 00, 02, 10

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 13-71. Initial Setting Procedure for UART Transmission

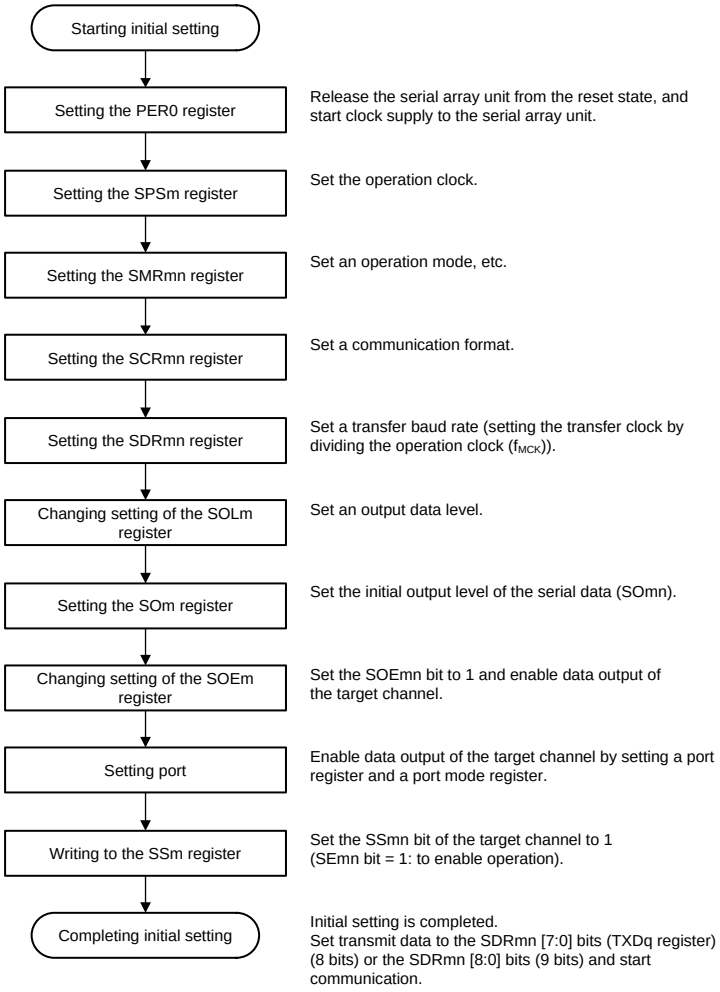


Figure 13-72. Procedure for Stopping UART Transmission

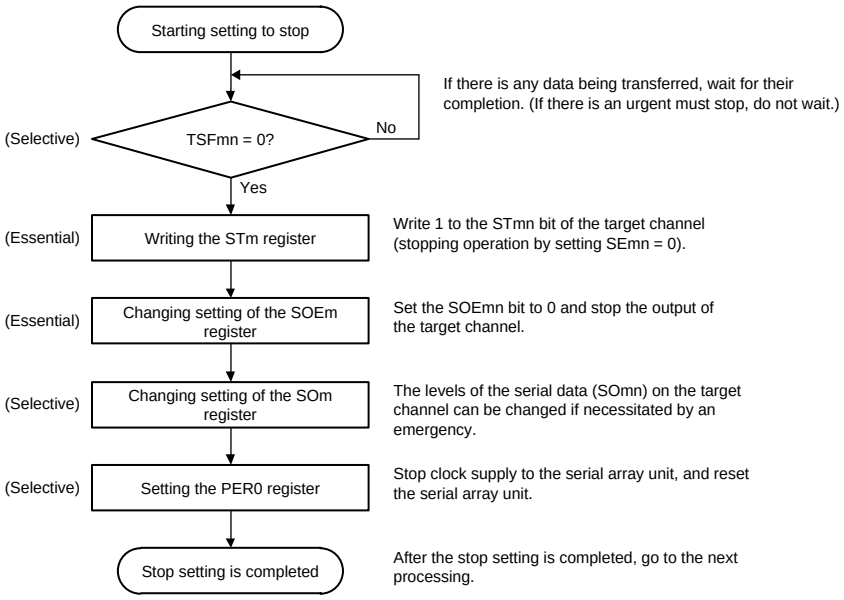
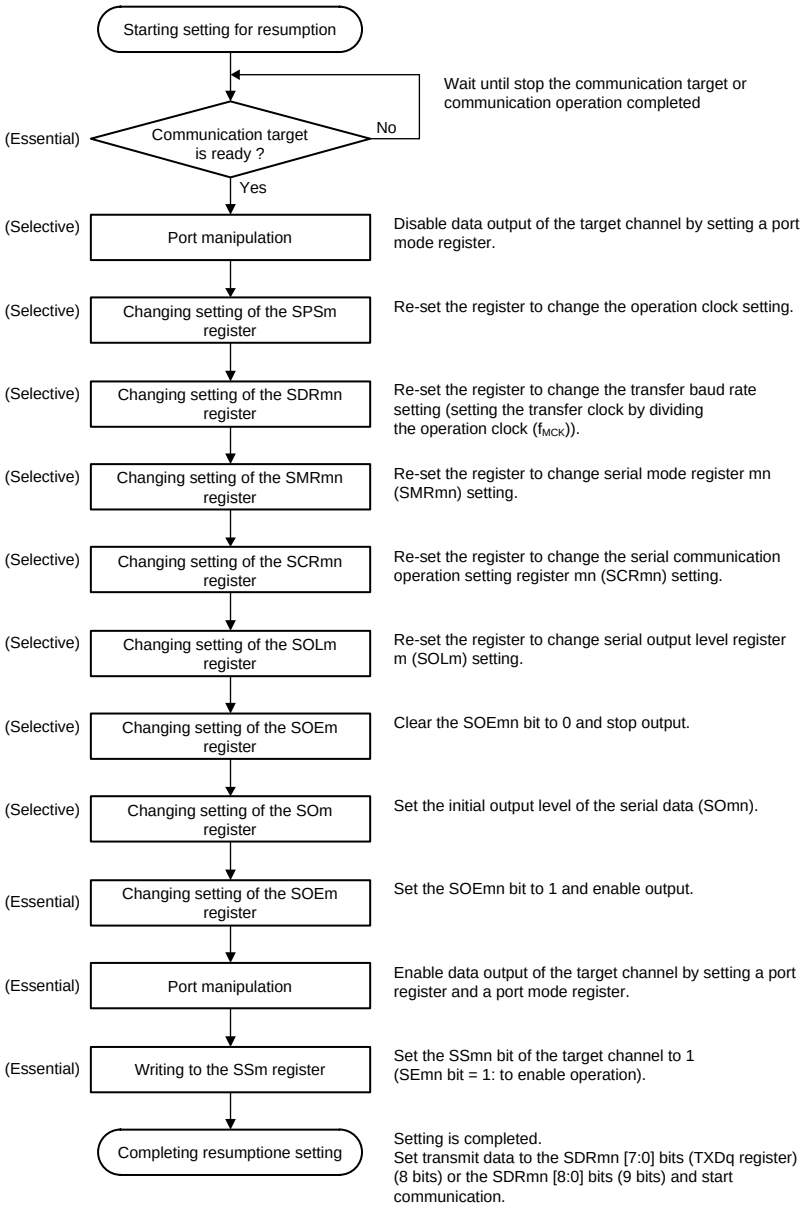


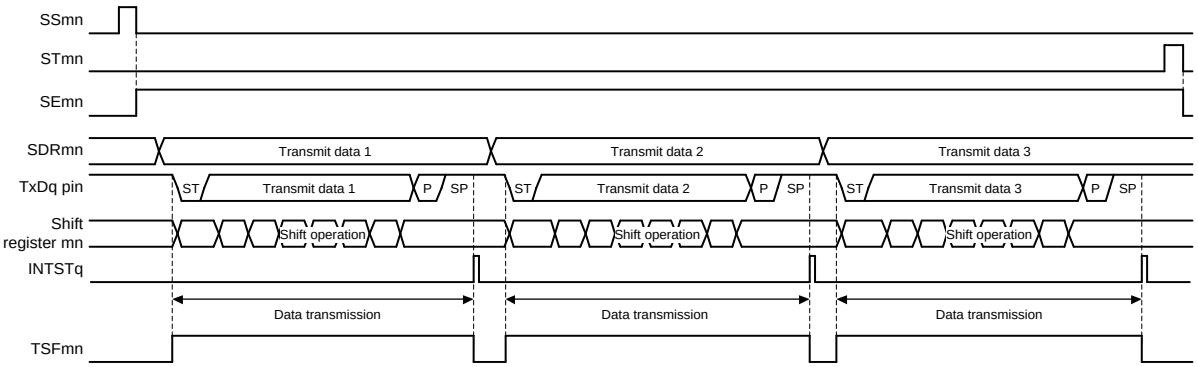
Figure 13-73. Procedure for Resuming UART Transmission



Remark If PER0 is rewritten while stopping the communication to reset the serial array unit, wait until the communication target stops or communication finishes, and then perform initialization instead of restarting the communication.

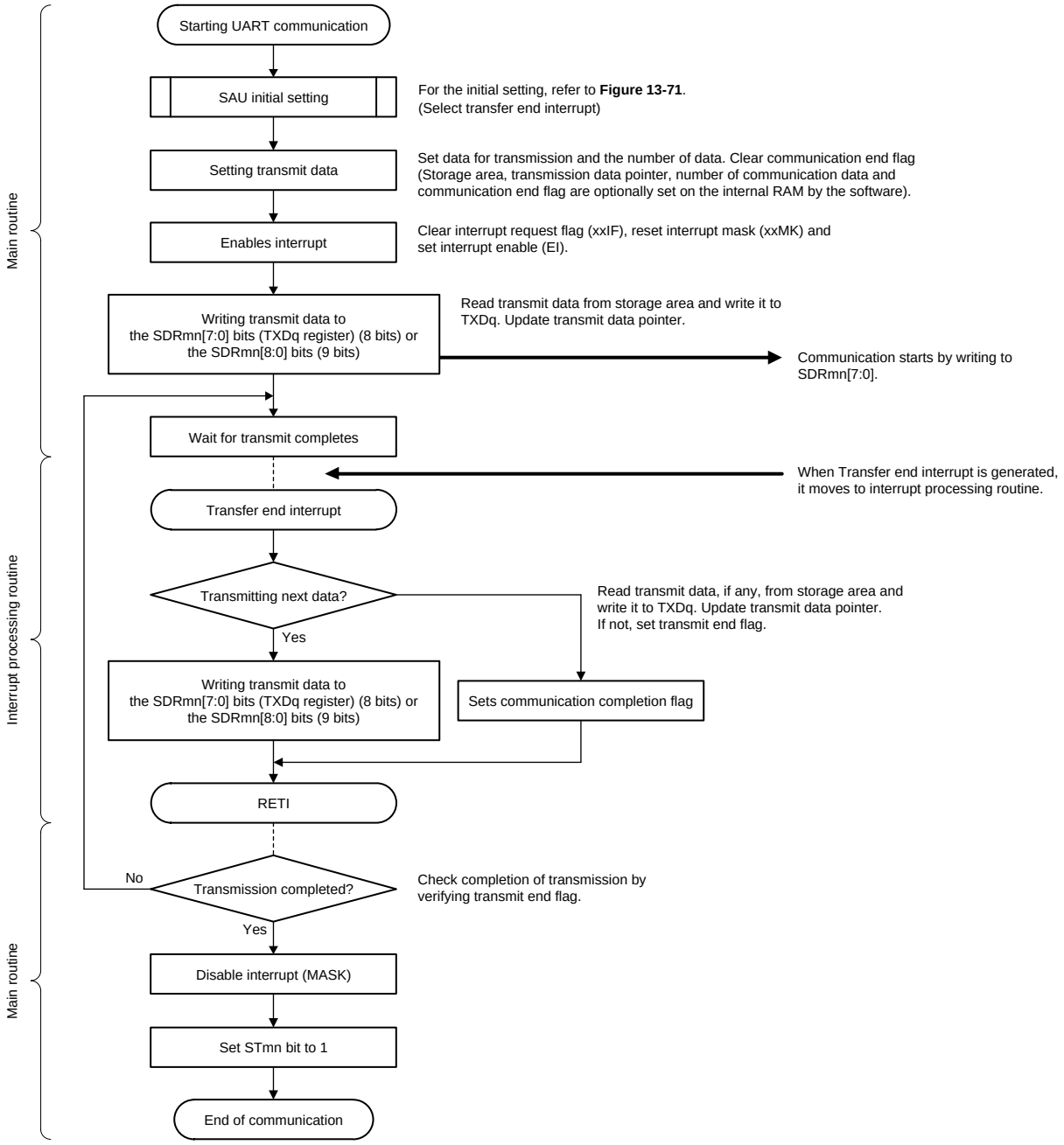
(3) Processing flow (in single-transmission mode)

Figure 13-74. Timing Chart of UART Transmission (in Single-Transmission Mode)



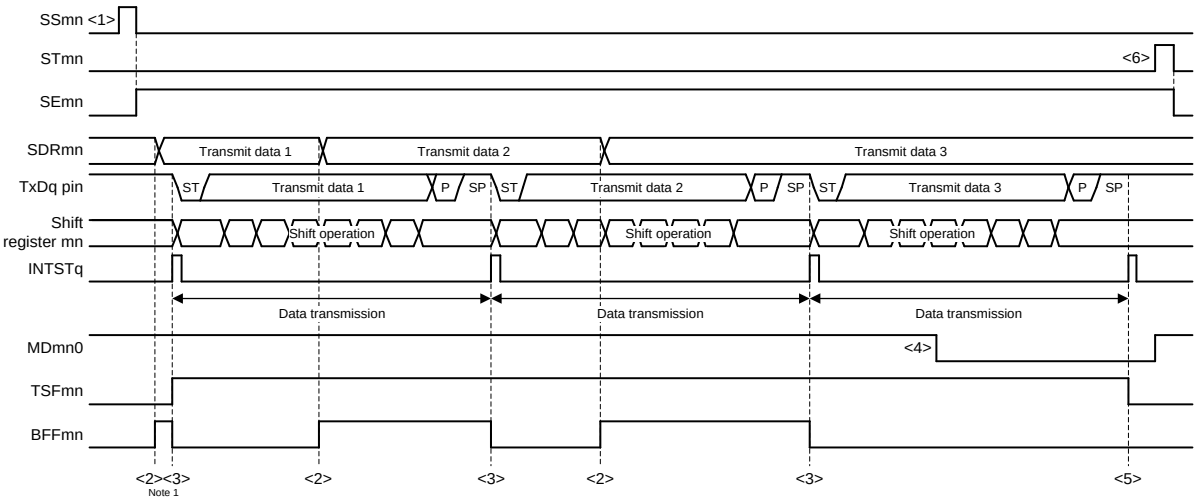
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 2), mn = 00, 02, 10

Figure 13-75. Flowchart of UART Transmission (in Single-Transmission Mode)



(4) Processing flow (in continuous transmission mode)

Figure 13-76. Timing Chart of UART Transmission (in Continuous Transmission Mode)

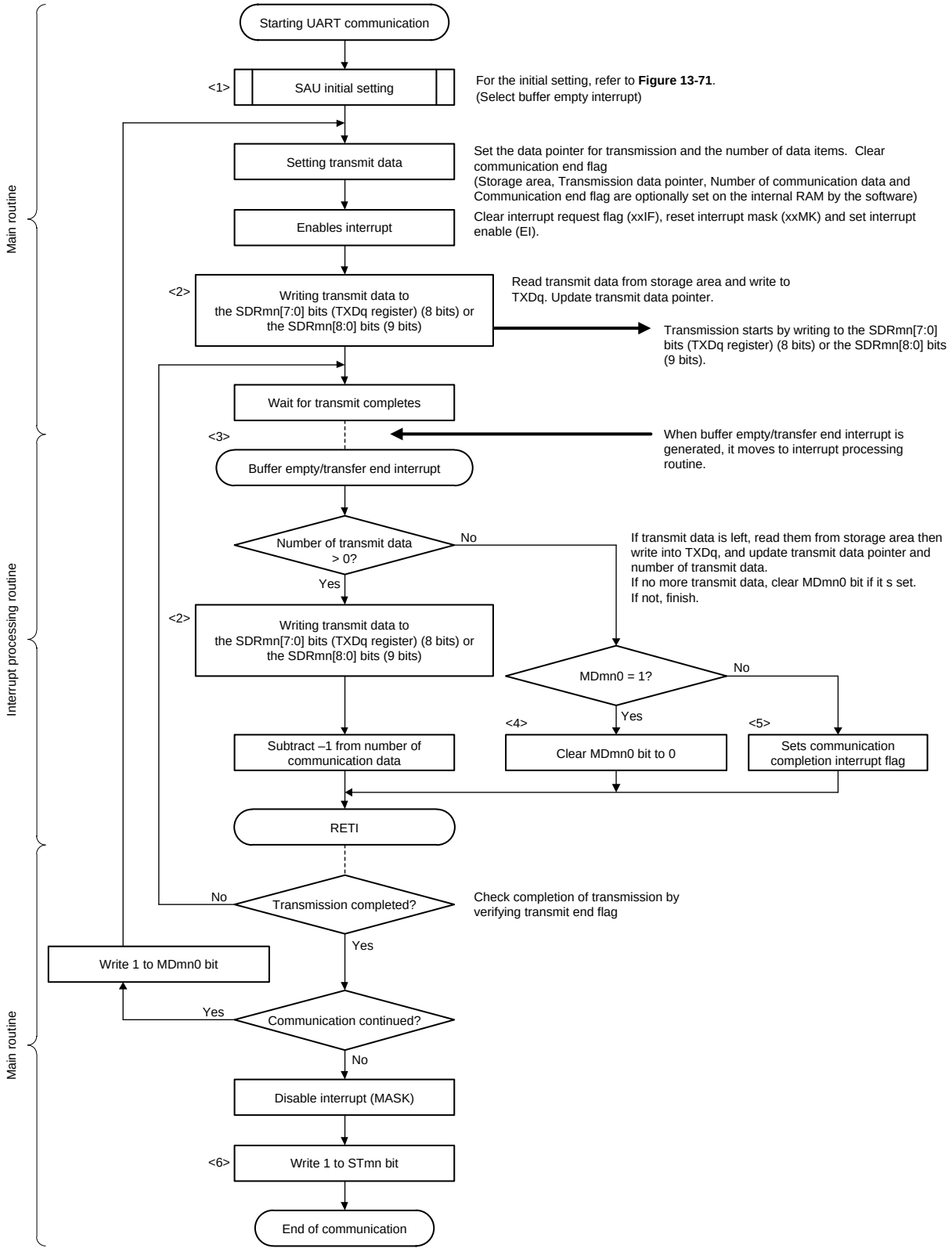


Note 1. If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

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 m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 2), mn = 00, 02, 10

Figure 13-77. Flowchart of UART Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in **Figure 13-76 Timing Chart of UART Transmission (in Continuous Transmission Mode)**.

13.6.2 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	UART2
Target channel	Channel 1 of SAU0	Channel 3 of SAU0	Channel 1 of SAU1
Pins used	RxD0	RxD1	RxD2
Interrupt	INTSR0	INTSR1	INTSR2
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)		
Error interrupt	INTSRE0	INTSRE1	INTSRE2
Error detection flag	• Framing error detection flag (FEFmn) • Parity error detection flag (PEFmn) • Overrun error detection flag (OVFmn)		
Transfer data length	7, 8, or 9 bits[Note 1]		
Transfer rate[Note 2]	Max. $f_{MCK}/6$ [bps] (SDRmn[15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]		
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)		
Parity bit	The following selectable • No parity bit (no parity check) • No parity judgment (0 parity) • Even parity check • Odd parity check		
Stop bit	Appending 1 bit		
Data direction	MSB or LSB first		

Note 1. Only UART0 can be specified for the 9-bit data length.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **CHAPTER 26 ELECTRICAL SPECIFICATIONS (T_A = −40 to +85°C)** and **CHAPTER 27 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C, T_A = −40 to +125°C)**.

Remark 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

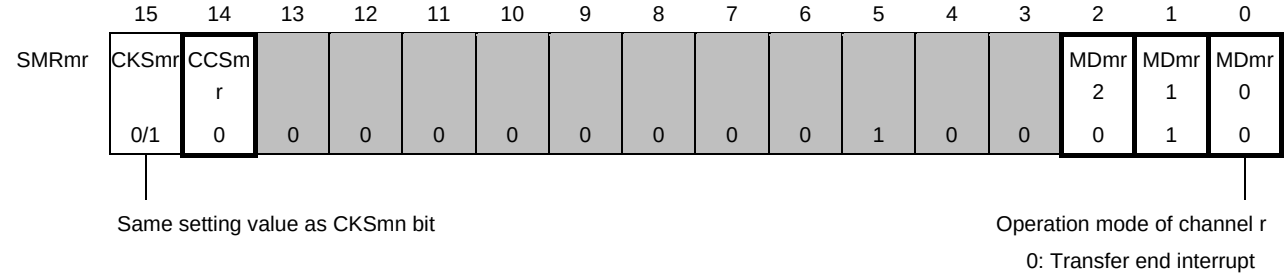
Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11

(1) Register setting

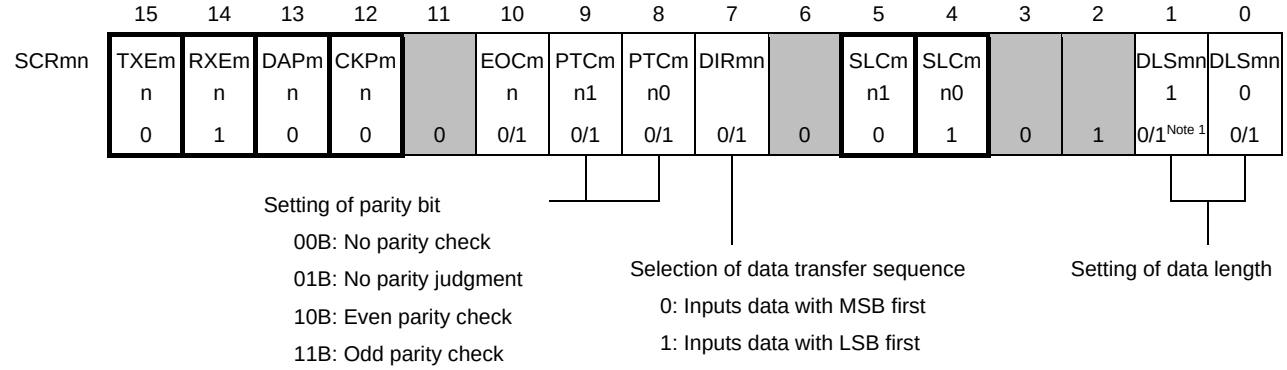
Figure 13-78. Example of Contents of Registers for UART Reception of UART (UART0 to UART2) (1/2)

(a) Serial mode register mn (SMRmn)

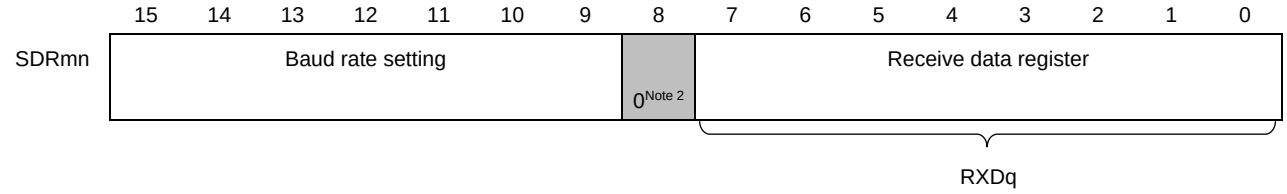
(b) Serial mode register mr (SMRmr)



(c) Serial communication operation setting register mn (SCRmn)



(d) Serial data register mn (SDRmn) (lower 8 bits: RXDq)



(Note 1 to Note 3, Caution, and Remarks are listed on the next page.)

Figure 13-78. Example of Contents of Registers for UART Reception of UART (UART0 to UART2) (2/2)

(e) Serial output register m (SOm) … This register is not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm					CKOm3			CKOm0					SOm3	SOm2		SOm0
	0	0	0	0	×Note 3	1	1	×Note 3	0	0	0	0	×Note 3	×Note 3	1	×

(f) Serial output enable register m (SOEm) … This register is not used in this mode.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm													SOEm3	SOEm2		SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	×Note 3	×Note 3	0	×

(g) Serial channel start register m (SSm) … Set only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1Note 3	×Note 3	0/1	×

Note 1. The SCR00 and SCR01 registers only. This bit is fixed to 1 for the other registers.

Note 2. When UART performs 9-bit communication, bits 0 to 8 of the SDRm1 register are used as the reception data specification area. Only UART0 can be specified for the 9-bit data length.

Note 3. Serial array unit 0 only.

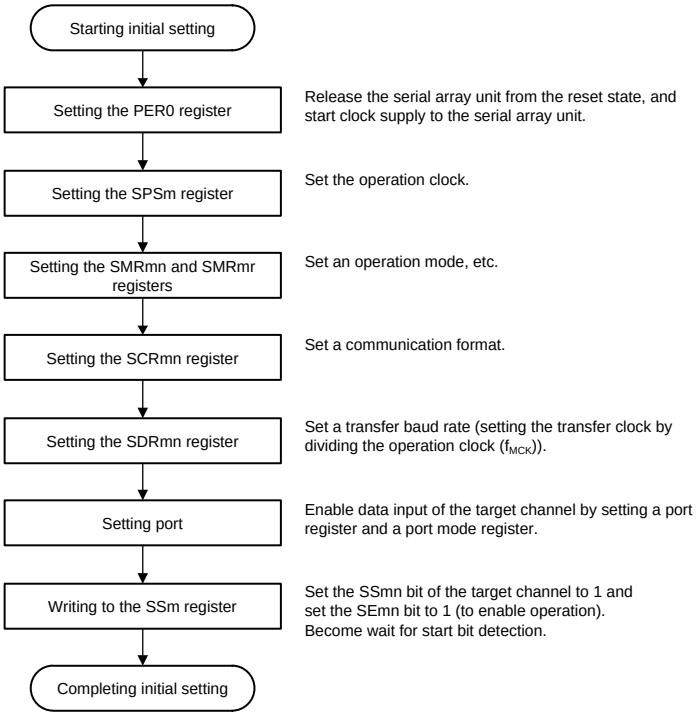
Caution For the UART reception, be sure to set the SMRmr register of channel r to UART transmission mode that is to be paired with channel n.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), r: Channel number (r = n – 1), q: UART number (q = 0 to 2), mn = 01, 03, 11

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 13-79. Initial Setting Procedure for UART Reception



Caution Set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after at least 4 f_{MCK} clock cycles have elapsed.

Figure 13-80. Procedure for Stopping UART Reception

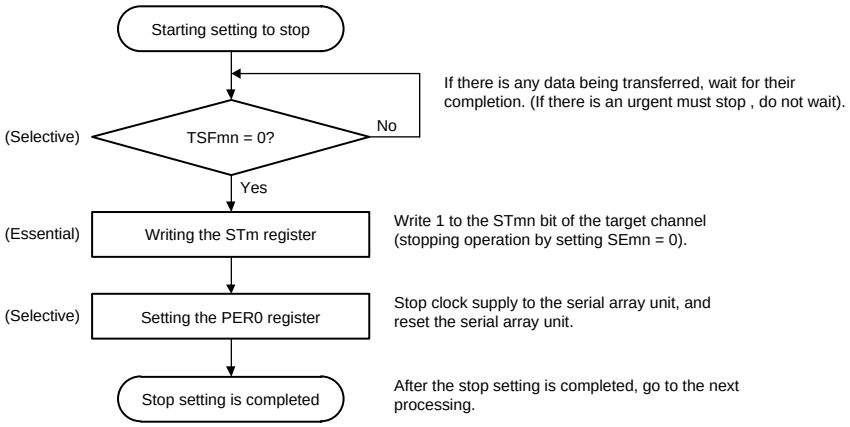
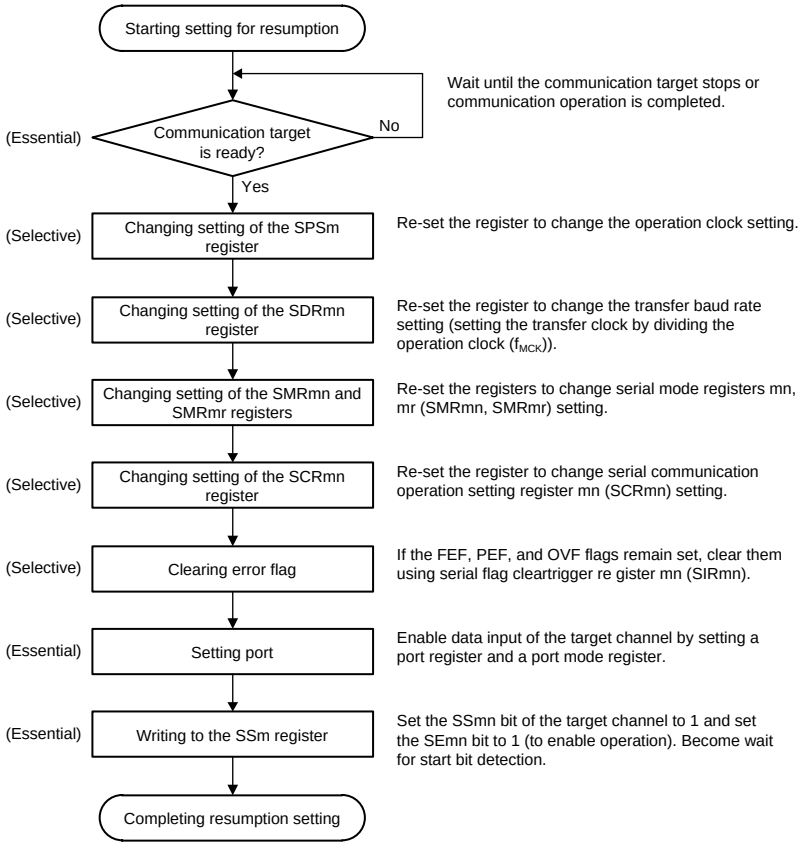


Figure 13-81. Procedure for Resuming UART Reception

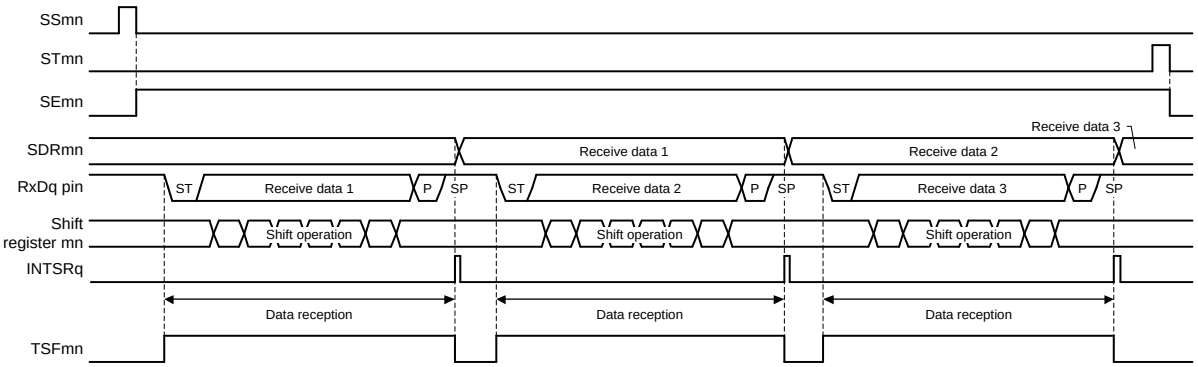


Caution Set the RXEmn bit of SCRmn register to 1, and then be sure to set SSmn to 1 after at least 4 f_{MCK} clocks have elapsed.

Remark If PER0 is rewritten while stopping the communication to reset the serial array unit, wait until the communication target stops or communication finishes, and then perform initialization instead of restarting the communication.

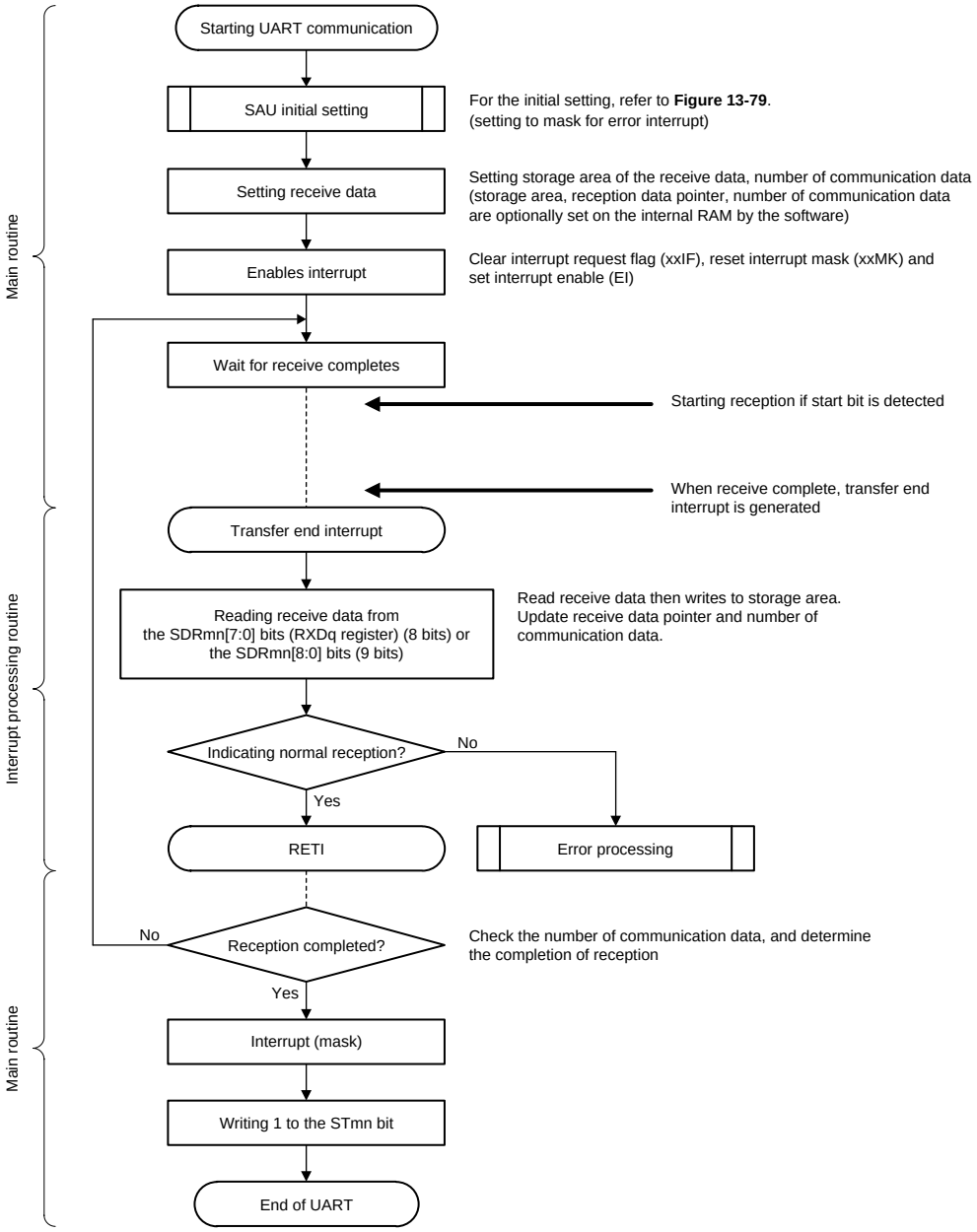
(3) Processing flow

Figure 13-82. Timing Chart of UART Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), r: Channel number (r = n – 1), q: UART number (q = 0 to 2), mn = 01, 03, 11

Figure 13-83. Flowchart of UART Reception



13.6.3 Calculating Baud Rate

(1) Baud rate calculation expression

The baud rate for UART (UART0 to UART2) communication can be calculated by the following expressions.

$$(\text{Baud rate}) = \{\text{Operation clock } (f_{MCK}) \text{ frequency of target channel}\} \div (\text{SDRmn}[15:9] + 1) \div 2[\text{bps}]$$

Caution Setting serial data register mn (SDRmn) SDRmn[15:9] = (0000000B, 0000001B) is prohibited.

Remark 1. When UART is used, the value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000010B to 1111111B) and therefore is 2 to 127.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 13-3. Selection of Operation Clock For UART

SMRmn Register	SPSm Register								Operation Clock (f_{MCK})[Note 1]	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 16 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	16 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	8 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	4 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	2 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	500 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	250 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	125 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	62.5 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	31.25 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	15.63 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	7.81 kHz
	X	X	X	X	1	1	0	0	$f_{CLK}/2^{12}$	3.91 kHz
	X	X	X	X	1	1	0	1	$f_{CLK}/2^{13}$	1.95 kHz
	X	X	X	X	1	1	1	0	$f_{CLK}/2^{14}$	977 Hz
	X	X	X	X	1	1	1	1	$f_{CLK}/2^{15}$	488 Hz
1	0	0	0	0	X	X	X	X	f_{CLK}	16 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	8 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	4 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	2 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	500 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	250 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	125 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	62.5 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	31.25 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	15.63 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	7.81 kHz
	1	1	0	0	X	X	X	X	$f_{CLK}/2^{12}$	3.91 kHz
	1	1	0	1	X	X	X	X	$f_{CLK}/2^{13}$	1.95 kHz
	1	1	1	0	X	X	X	X	$f_{CLK}/2^{14}$	977 Hz
	1	1	1	1	X	X	X	X	$f_{CLK}/2^{15}$	488 Hz
Other than above									Setting prohibited	

(Note 1 and Remarks are listed on the next page.)

Note 1. 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. X: Don't care

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

(2) Baud rate error during transmission

The baud rate error of UART (UART0 to UART2) 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.

$$(\text{Baud rate error}) = (\text{Calculated baud rate value}) \div (\text{Target baud rate}) \times 100 - 100[\%]$$

Here is an example of setting a UART baud rate at f_{CLK} = 16 MHz.

UART Baud Rate (Target Baud Rate)	f_{CLK} = 16 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Baud Rate	Error from Target Baud Rate
300 bps	$f_{CLK}/2^9$	51	300.48 bps	+0.16%
600 bps	$f_{CLK}/2^8$	51	600.96 bps	+0.16%
1200 bps	$f_{CLK}/2^7$	51	1201.92 bps	+0.16%
2400 bps	$f_{CLK}/2^6$	51	2403.85 bps	+0.16%
4800 bps	$f_{CLK}/2^5$	51	4807.69 bps	+0.16%
9600 bps	$f_{CLK}/2^4$	51	9615.38 bps	+0.16%
19200 bps	$f_{CLK}/2^3$	51	19230.8 bps	+0.16%
31250 bps	$f_{CLK}/2^3$	31	31250.0 bps	±0.0%
38400 bps	$f_{CLK}/2^2$	51	38461.5 bps	+0.16%
76800 bps	$f_{CLK}/2$	51	76923.1 bps	+0.16%
153600 bps	f_{CLK}	51	153846 bps	+0.16%
312500 bps	f_{CLK}	25	307692.3 bps	−1.54%

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 10

(3) Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UART0 to UART2) 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.

$$(\text{Maximum receivable baud rate}) = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times Brate$$

$$(\text{Minimum receivable baud rate}) = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times Brate$$

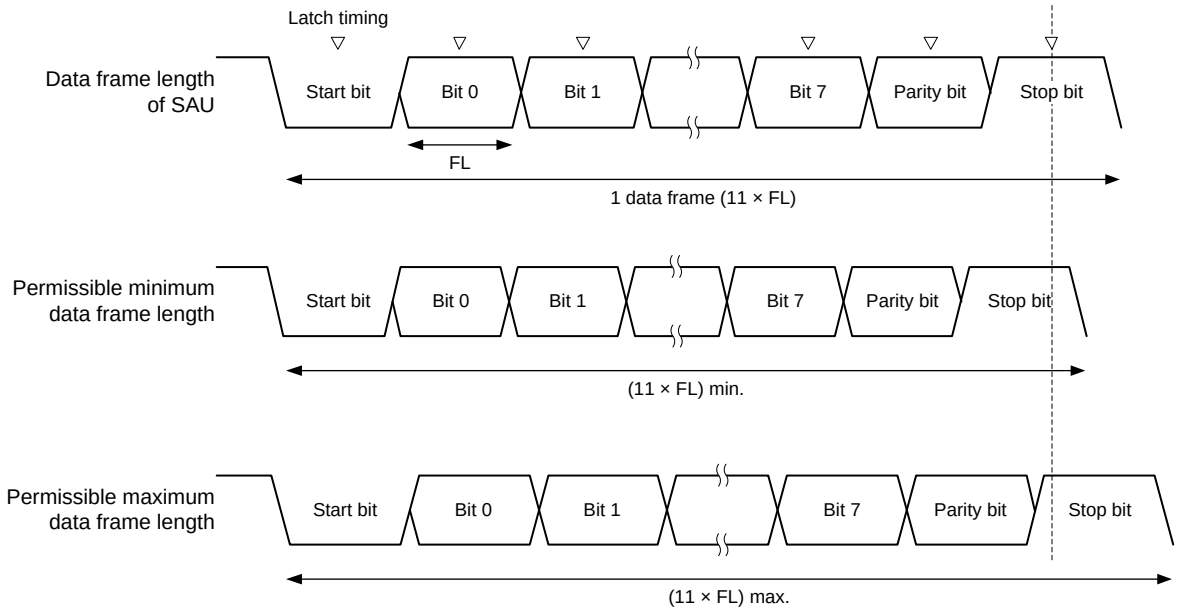
Brate: Calculated baud rate value at the reception side (See **13.6.3(1) Baud rate calculation expression**.)

k: SDRmn[15:9] + 1

Nfr: 1 data frame length [bits] = (Start bit) + (Data length) + (Parity bit) + (Stop bit)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), mn = 01, 03, 11

Figure 13-84. Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in **Figure 13-84**, 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.

13.6.4 Procedure for Processing Errors that Occurred During UART (UART0 to UART2) Communication

The procedure for processing errors that occurred during UART (UART0 to UART2) communication is described in **Figure 13-85** and **Figure 13-86**.

Figure 13-85. Processing Procedure in Case of Parity Error or Overrun Error

Software Manipulation	State of the Hardware	Remark
Reads serial data register mn (SDRmn).	→ The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		The error type is identified and the read value is used to clear the error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn).	→ The error flag is cleared.	Only the error generated during reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.

Figure 13-86. Processing Procedure in Case of Framing Error

Software Manipulation	State of the Hardware	Remark
Reads serial data register mn (SDRmn).	→ The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		The error type is identified and the read value is used to clear the error flag.
Writes serial flag clear trigger register mn (SIRmn).	→ The error flag is cleared.	Only the error generated during reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets the STmn bit of serial channel stop register m (STm) to 1.	→ The SEmn bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operation.	
Synchronization with other party of communication		Synchronization with the other party of communication is re-established and communication is resumed because it is considered that a framing error has occurred because the start bit has been shifted.
Sets the SSmn bit of serial channel start register m (SSm) to 1.	→ The SEmn bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate.	

Remark m: Unit number (m = 0, 1), n: Channel number (n = 1 to 3), mn = 00 to 03, 10, 11

13.7 Operation of Simplified I²C (IIC00, IIC11, IIC20) Communication

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This simplified I²C is designed for single communication with a device such as EEPROM, flash memory, or A/D converter, and therefore, it functions only as a master.

Operate the control registers by software for setting the start and stop conditions while observing the specifications of the I²C bus line.

[Data transmission/reception]

- Master transmission, master reception (only master function with a single master)
- ACK output function[Note 1], 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)
- Wait detection functions

Note 1. 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 **13.7.3(2) Processing flow** for details.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), mn = 00, 03, 10

The channel supporting simplified I²C (IIC00, IIC11, IIC20) is channels 0 and 3 of SAU0 and channel 0 of SAU1.

● 10-pin products

Unit	Channel	Used as simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	—	—
	3	—		—

● 16-pin products

Unit	Channel	Used as simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11

● 20-, 24-, and 32-pin products

Unit	Channel	Used as simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	IIC00
	1	—		—
	2	—	UART1	—
	3	CSI11		IIC11
1	0	CSI20	UART2	IIC20
	1	—		—

Simplified I²C (IIC00, IIC11, IIC20) performs the following four types of communication operations.

- Address field transmission (See **13.7.1**.)
- Data transmission (See **13.7.2**.)
- Data reception (See **13.7.3**.)
- Stop condition generation (See **13.7.4**.)

13.7.1 Address Field Transmission

Address field transmission is a transmission operation that first executes in I^2C communication to identify the target for transfer (slave). After a start condition is generated, an address (7 bits) and a transfer direction (1 bit) are transmitted in one frame.

Simplified I^2C	IIC00	IIC11	IIC20
Target channel	Channel 0 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1
Pins used	SCL00, SDA00Note 1	SCL11, SDA11Note 1	SCL20, SDA20Note 1
Interrupt	INTIIC00	INTIIC11	INTIIC20
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)		
Error detection flag	ACK error detection flag (PEFmn)		
Transfer data length	8 bits (transmitted with specifying the higher 7 bits as address and the least significant bit as R/W control)		
Transfer rateNote 2	Max. $f_{MCK}/4$ [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)		
Data level	Non-reverse output (default: high level)		
Parity bit	No parity bit		
Stop bit	Appending 1 bit (for ACK transmission/reception timing)		
Data direction	MSB first		

Note 1. To perform communication via simplified I^2C, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) with the port output mode register (POMxx). See **4.3 Registers Controlling Port Function** and **4.5 Register Settings When Using Alternate Function** for details.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **CHAPTER 26 ELECTRICAL SPECIFICATIONS (T_A = −40 to +85°C)** and **CHAPTER 27 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C, T_A = −40 to +125°C)**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), mn = 00, 03, 10

(1) Register setting

Figure 13-87. Example of Contents of Registers for Address Field Transmission of Simplified I²C (IIC00, IIC11, IIC20) (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	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	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

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	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	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								Transmit data setting (address + R/W)							
								0								

SIOr

(d) Serial output register m (SOm)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm					CKOm3			CKOm0					SOm3			SOm0
	0	0	0	0	0/1	1	1	0/1	0	0	0	0	0/1	1	1	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													SOEm3			SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	0	0	0/1

SOEmn = 0 until the start condition is generated, and SOEmn = 1 after generation.

(Remarks are listed on the next page.)

Figure 13-87. Example of Contents of Registers for Address Field Transmission of Simplified I²C (IIC00, IIC11, IIC20)

(2/2)

(f) Serial channel start register m (SSm) … Set only the bit of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 ×	SSm1 ×	SSm0 0/1

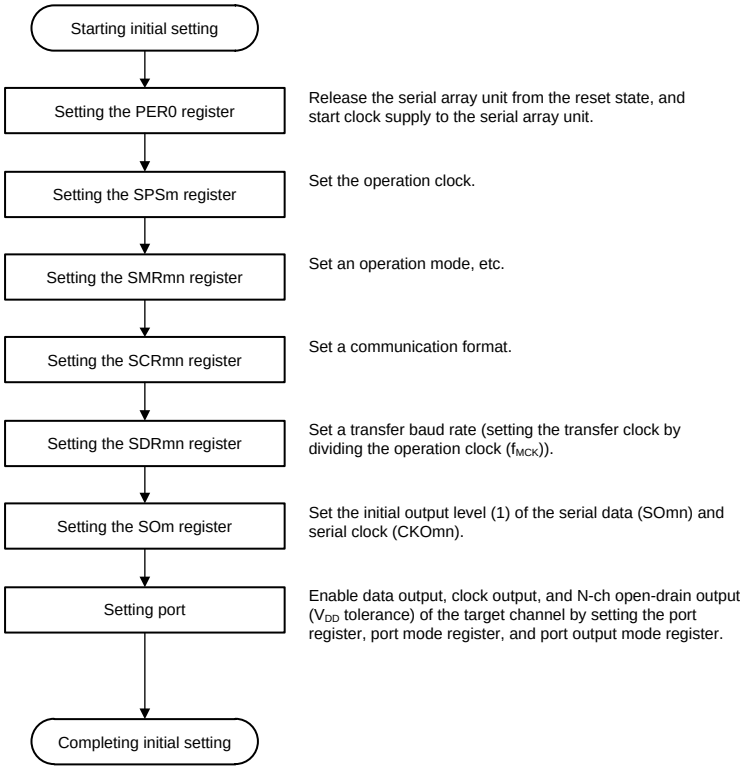
SSmn = 0 until the start condition is generated, and SSmn = 1 after generation.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), r: IIC number (r = 00, 11, 20), mn = 00, 03,10

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.

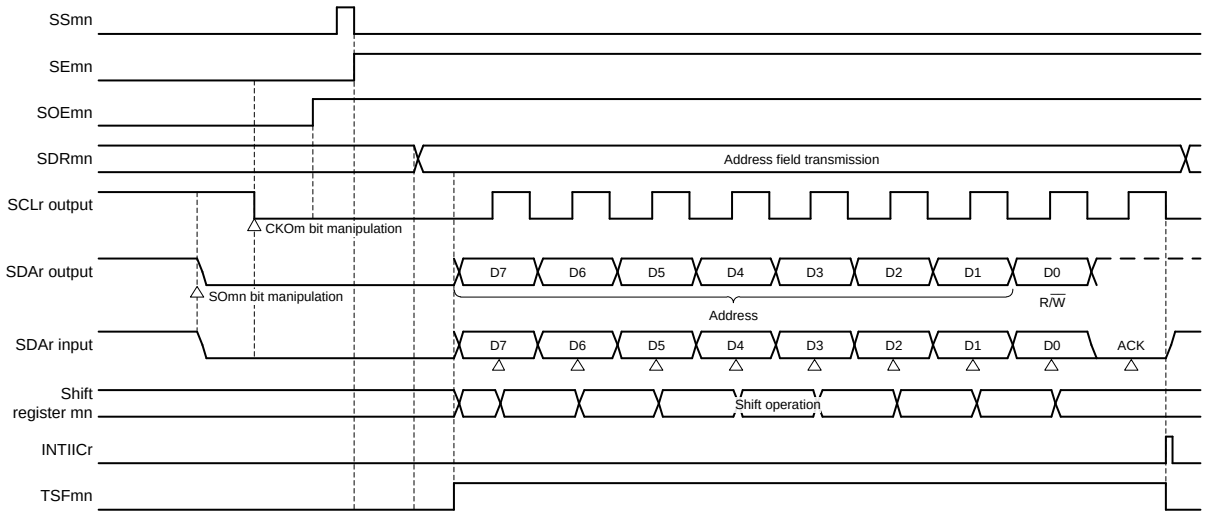
(2) Operation procedure

Figure 13-88. Initial Setting Procedure for Simplified I²C Address Field Transmission



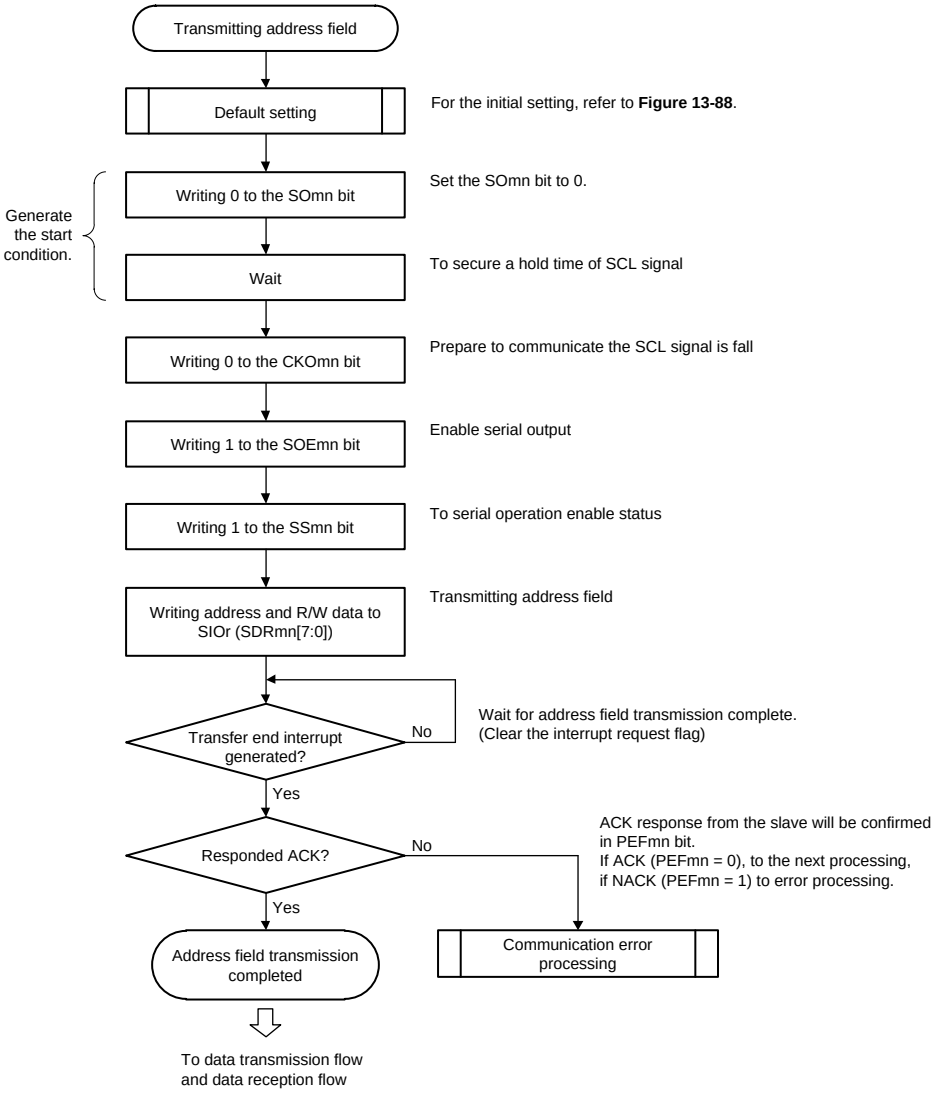
(3) Processing flow

Figure 13-89. Timing Chart of Address Field Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), r: IIC number (r = 00, 11, 20), mn = 00, 03, 10

Figure 13-90. Flowchart of Simplified I²C Address Field Transmission



13.7.2 Data Transmission

Data transmission is an operation to transmit data to the target for transfer (slave) after transmission of an address field. After all data are transmitted to the slave, a stop condition is generated and the bus is released.

Simplified I^2C	IIC00	IIC11	IIC20
Target channel	Channel 0 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1
Pins used	SCL00, SDA00Note 1	SCL11, SDA11Note 1	SCL20, SDA20Note 1
Interrupt	INTIIC00	INTIIC11	INTIIC20
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)		
Error detection flag	ACK error detection flag (PEFmn)		
Transfer data length	8 bits		
Transfer rateNote 2	Max. $f_{MCK}/4$ [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)		
Data level	Non-reverse output (default: high level)		
Parity bit	No parity bit		
Stop bit	Appending 1 bit (for ACK reception timing)		
Data direction	MSB first		

Note 1. To perform communication via simplified I^2C, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) with the port output mode register (POMxx). For details, see **4.3 Registers Controlling Port Function** and **4.5 Register Settings When Using Alternate Function**.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **CHAPTER 26 ELECTRICAL SPECIFICATIONS (T_A = −40 to +85°C)** and **CHAPTER 27 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C, T_A = −40 to +125°C)**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), mn = 00, 03,10

(1) Register setting

Figure 13-91. Example of Contents of Registers for Data Transmission of Simplified I^2C (IIC00, IIC11, IIC20)

(a) Serial mode register mn (SMRmn) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0

(b) Serial communication operation setting register mn (SCRmn) … Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	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 1]							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					CKOm3			CKOm0					SOm3	SOm2		SOm0
	0	0	0	0	0/1[Note 2]	1	1	0/1[Note 2]	0	0	0	0	0/1[Note 2]	×	1	0/1[Note 2]

(e) Serial output enable register m (SOEm) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm													SOEm3	SOEm2		SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	1	1	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													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	×	×	0/1

(Note 1, Note 2, and Remarks are listed on the next page.)

Note 1. Because the setting is completed by address field transmission, setting is not required.

Note 2. The value varies depending on the communication data during communication operation.

Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), r: IIC number (r = 00, 11, 20), mn = 00, 03,10

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 13-92. Timing Chart of Data Transmission

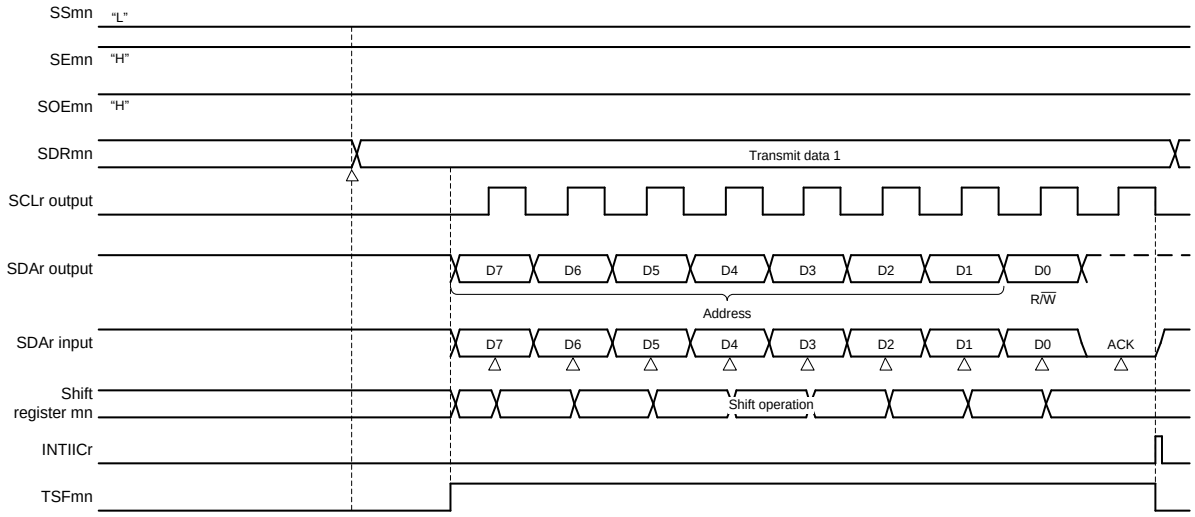
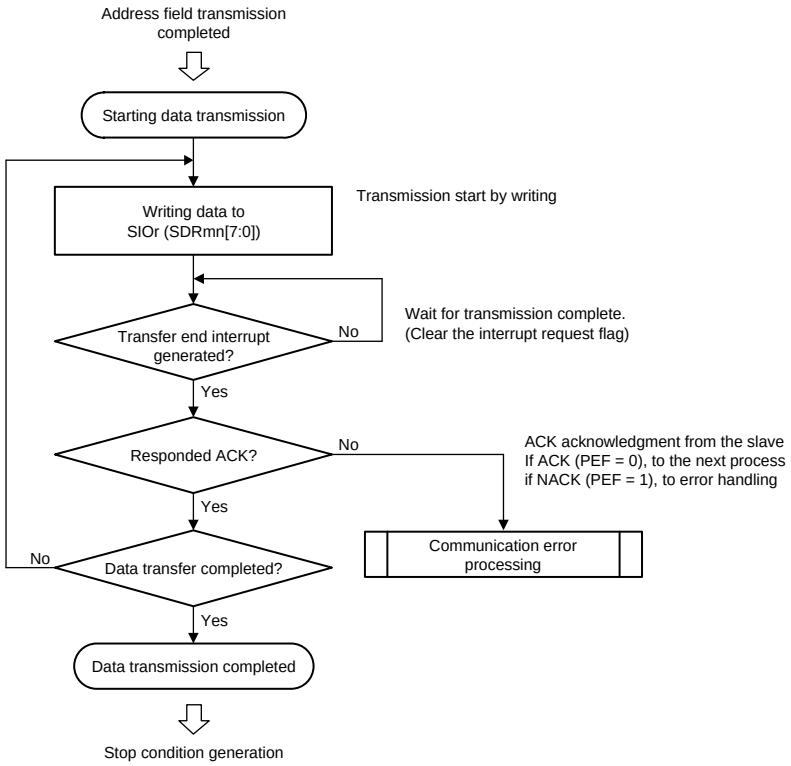


Figure 13-93. Flowchart of Simplified I²C Data Transmission



13.7.3 Data Reception

Data reception is an operation to receive data from the target for transfer (slave) after transmission of an address field. After all data are received from the slave, a stop condition is generated and the bus is released.

Simplified I^2C	IIC00	IIC11	IIC20
Target channel	Channel 0 of SAU0	Channel 3 of SAU0	Channel 0 of SAU1
Pins used	SCL00, SDA00Note 1	SCL11, SDA11Note 1	SCL20, SDA20Note 1
Interrupt	INTIIC00	INTIIC11	INTIIC20
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)		
Error detection flag	Overrun error detection flag (OVFmn) only		
Transfer data length	8 bits		
Transfer rateNote 2	Max. $f_{MCK}/4$ [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)		
Data level	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^2C, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) with the port output mode register (POMxx). For details, see **4.3 Registers Controlling Port Function** and **4.5 Register Settings When Using Alternate Function**.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics specified in the electrical characteristics. For details, see **CHAPTER 26 ELECTRICAL SPECIFICATIONS (T_A = −40 to +85°C)** and **CHAPTER 27 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C, T_A = −40 to +125°C)**.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), mn = 00, 03, 10

(1) Register setting

Figure 13-94. Example of Contents of Registers for Data Reception of Simplified I^2C (IIC00, IIC10, IIC11, IIC20)

(a) Serial mode register mn (SMRmn) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn						STSmn		SISmn0				MDmn2	MDmn1	MDmn0
	0/1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0

(b) Serial communication operation setting register mn (SCRmn) … Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn	RXEmn	DAPmn	CKPmn		EOCmn	PTCmn1	PTCmn0	DIRmn		SLCmn1	SLCmn0			DLSmn1	DLSmn0
	0	1	0	0	0	0	0	0	0	0	0	1	0	1	1	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 1]							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					CKOm3			CKOm0					SOm3	SOm2		SOm0
	0	0	0	0	0/1[Note 2]	1	1	0/1[Note 2]	0	0	0	0	0/1[Note 2]	×	1	0/1[Note 2]

(e) Serial output enable register m (SOEm) … Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm													SOEm3	SOEm2		SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	×	0	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													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	0/1	×	×	0/1

(Note 1, Note 2, and Remarks are listed on the next page.)

Note 1. Because the setting is completed by address field transmission, setting is not required.

Note 2. The value varies depending on the communication data during communication operation.

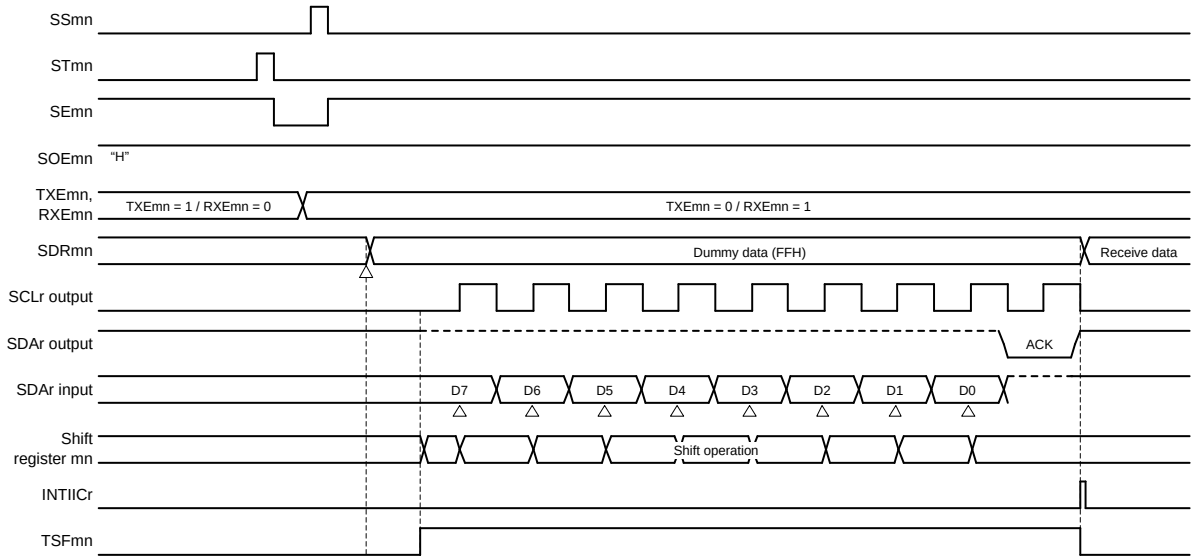
Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), r: IIC number (r = 00, 11, 20), mn = 00, 03, 10

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.

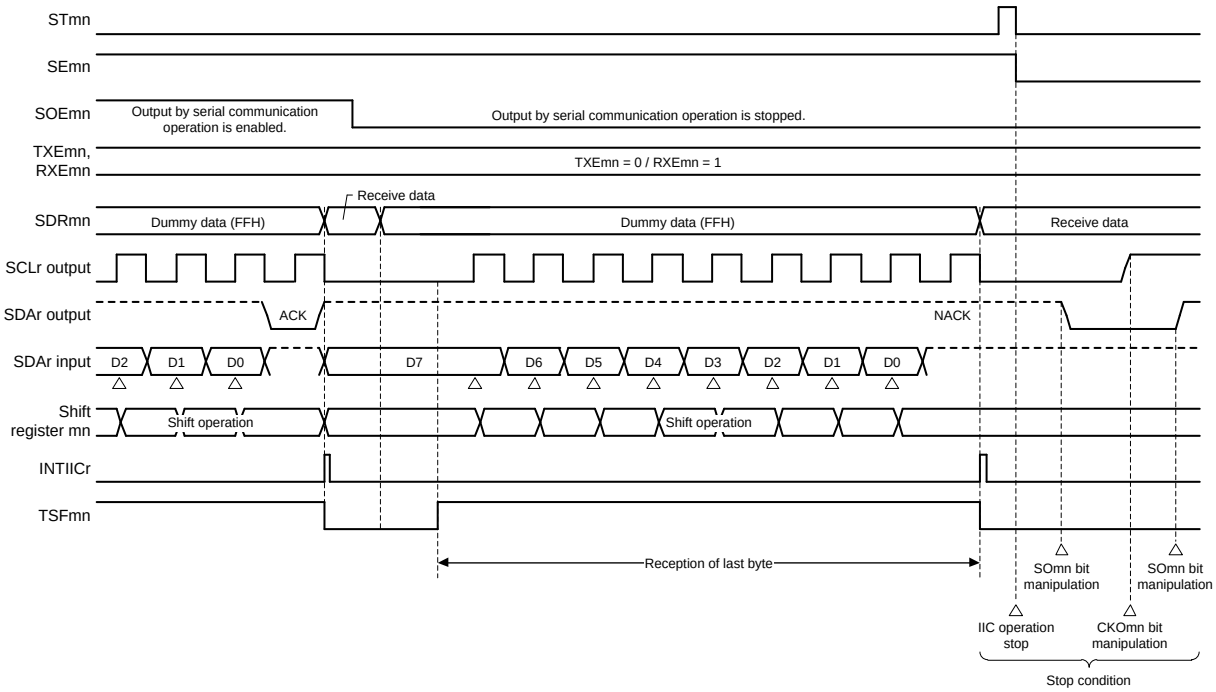
(2) Processing flow

Figure 13-95. Timing Chart of Data Reception

(a) When starting data reception

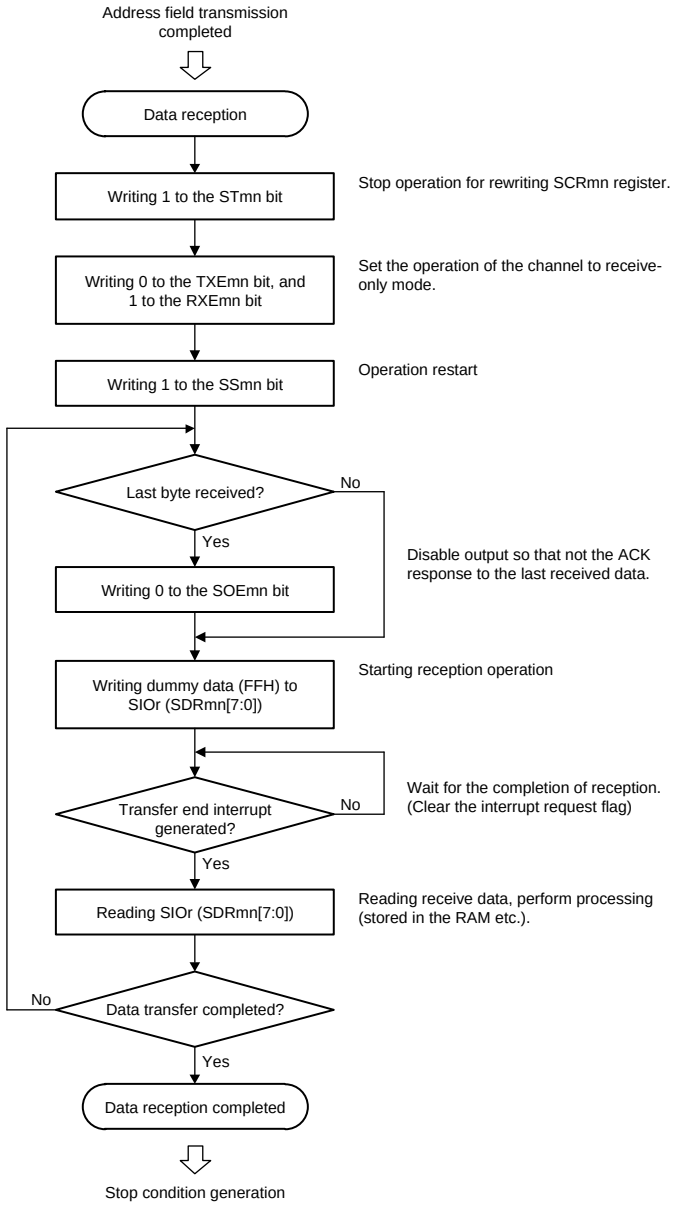


(b) When receiving last data



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), r: IIC number (r = 00, 11, 20), mn = 00, 03, 10

Figure 13-96. Flowchart of Data Reception



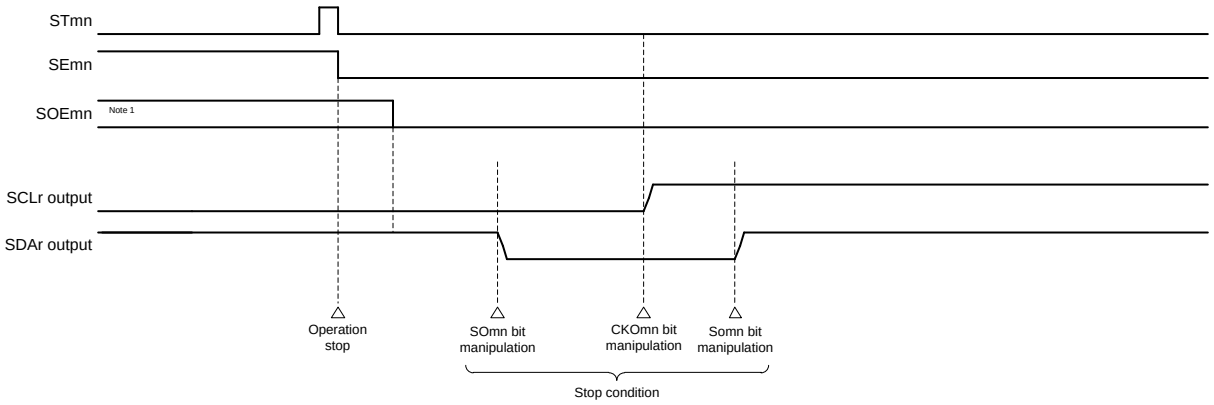
Caution ACK is not output when the last data is received (NACK). Communication is then completed by setting 1 in the STmn bit of serial channel stop register m (STm) to stop operation and generating a stop condition.

13.7.4 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.

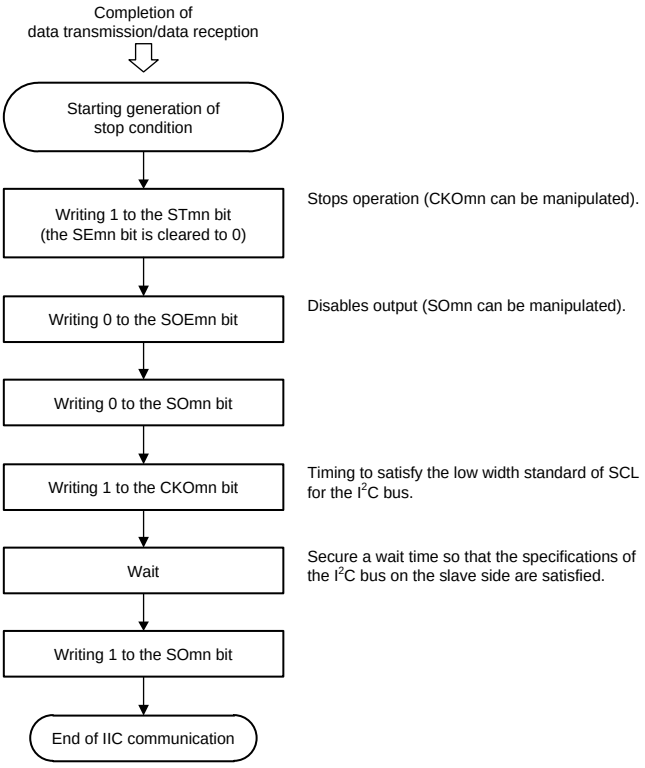
(1) Processing flow

Figure 13-97. Timing Chart of Stop Condition Generation



Note 1. During a receive operation, the SOEmn bit of serial output enable register m (SOEm) is cleared to 0 before receiving the last data.

Figure 13-98. Flowchart of Stop Condition Generation



13.7.5 Calculating Transfer Rate

The transfer rate for simplified I²C (IIC00, IIC11, IIC20) communication can be calculated by the following expressions.

$$(\text{Transfer rate}) = \{\text{Operation clock } (f_{MCK}) \text{ frequency of target channel}\} \div (\text{SDRmn}[15:9] + 1) \div 2$$

Caution SDRmn[15:9] must not be set to 00000000B. Set SDRmn[15:9] to 0000001B or greater.
The duty ratio of the SCL signal output by the simplified I²C is 50%. The I²C bus specifications define that the low-level width of the SCL signal is longer than the high-level width. If 400 kbps (fast mode) or 1 Mbps (fast mode plus) is specified, therefore, the low-level width of the SCL output signa becomes shorter than the value specified in the I²C bus specifications. Make sure that the SDRmn[15:9] value satisfies the I²C bus specifications.

Remark 1. The value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000001B to 1111111B) and therefore is 1 to 127.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), mn = 00, 03, 10

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 13-4. Selection of Operation Clock For Simplified I^2C

SMRmn Register	SPSm Register								Operation Clock (f_{MCK})[Note 1]	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 16 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	16 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	8 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	4 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	2 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	500k Hz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	250 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	125 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	62.5 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	31.25 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	15.63 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	7.81 kHz
1	0	0	0	0	X	X	X	X	f_{CLK}	16 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	8 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	4 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	2 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	500 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	250 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	125 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	62.5 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	31.25 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	15.63 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	7.81 kHz
Other than above									Setting prohibited	

Note 1. 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. X: Don't care

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), mn = 00, 03, 10

Here is an example of setting an I^2C transfer rate where $f_{MCK} = f_{CLK} = 16$ MHz.

I^2C Transfer Mode (Desired Transfer Rate)	f_{CLK} = 16 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Transfer Rate	Error from Desired Transfer Rate
100 kHz	$f_{CLK}/2$	39	100 kHz	0.0%
400 kHz	f_{CLK}	20	380 kHz	5.0%Note 1

Note 1. The error cannot be set to about 0% because the duty ratio of the SCL signal is 50%.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), mn = 00, 03, 10

13.7.6 Procedure for Processing Errors that Occurred during Simplified I²C (IIC00, IIC11, IIC20) Communication

The procedure for processing errors that occurred during simplified I²C (IIC00, IIC11, IIC20) communication is described in **Figure 13-99** and **Figure 13-100**.

Figure 13-99. Processing Procedure in Case of Overrun Error

Software Manipulation	State of the Hardware	Remark
Reads serial data register mn (SDRmn).	→ The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		The error type is identified and the read value is used to clear the error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn).	→ The error flag is cleared.	Only the error during reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.

Figure 13-100. Processing Procedure in Case of ACK Error in Simplified I²C Mode

Software Manipulation	State of the Hardware	Remark
Reads serial status register mn (SSRmn).		The error type is identified and the read value is used to clear the error flag.
Writes serial flag clear trigger register mn (SIRmn).	→ The error flag is cleared.	Only the error during reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets the STmn bit of serial channel stop register m (STm) to 1.	→ The SEmn bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operation.	The slave is not ready for reception because ACK is not returned. Therefore, a stop condition is created, the bus is released, and communication is started again from the start condition. Or, a restart condition is generated and transmission can be redone from address transmission.
Creates a stop condition.		
Creates a start condition.		
Sets the SSmn bit of serial channel start register m (SSm) to 1.	→ The SEmn bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate.	

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 3), r: IIC number (r = 00, 11, 20), mn = 00, 03, 10

CHAPTER 14 SERIAL INTERFACE IICA

14.1 Functions of Serial Interface IICA

The serial interface IICA has the following three modes.

1) Operation stop mode

This mode is used when serial transfers are not performed. The operating power can be reduced in this mode.

2) I^2C bus mode (multi-master supported)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCLA0) line and a serial data bus (SDAA0) line.

It complies with the I^2C bus format and the master can generate “start condition”, “address”, “transfer direction specification”, “data”, and “stop condition” to the slave on the serial data bus. The slave automatically detects these received states and data by hardware. This function can simplify the part of application program that controls the I^2C bus.

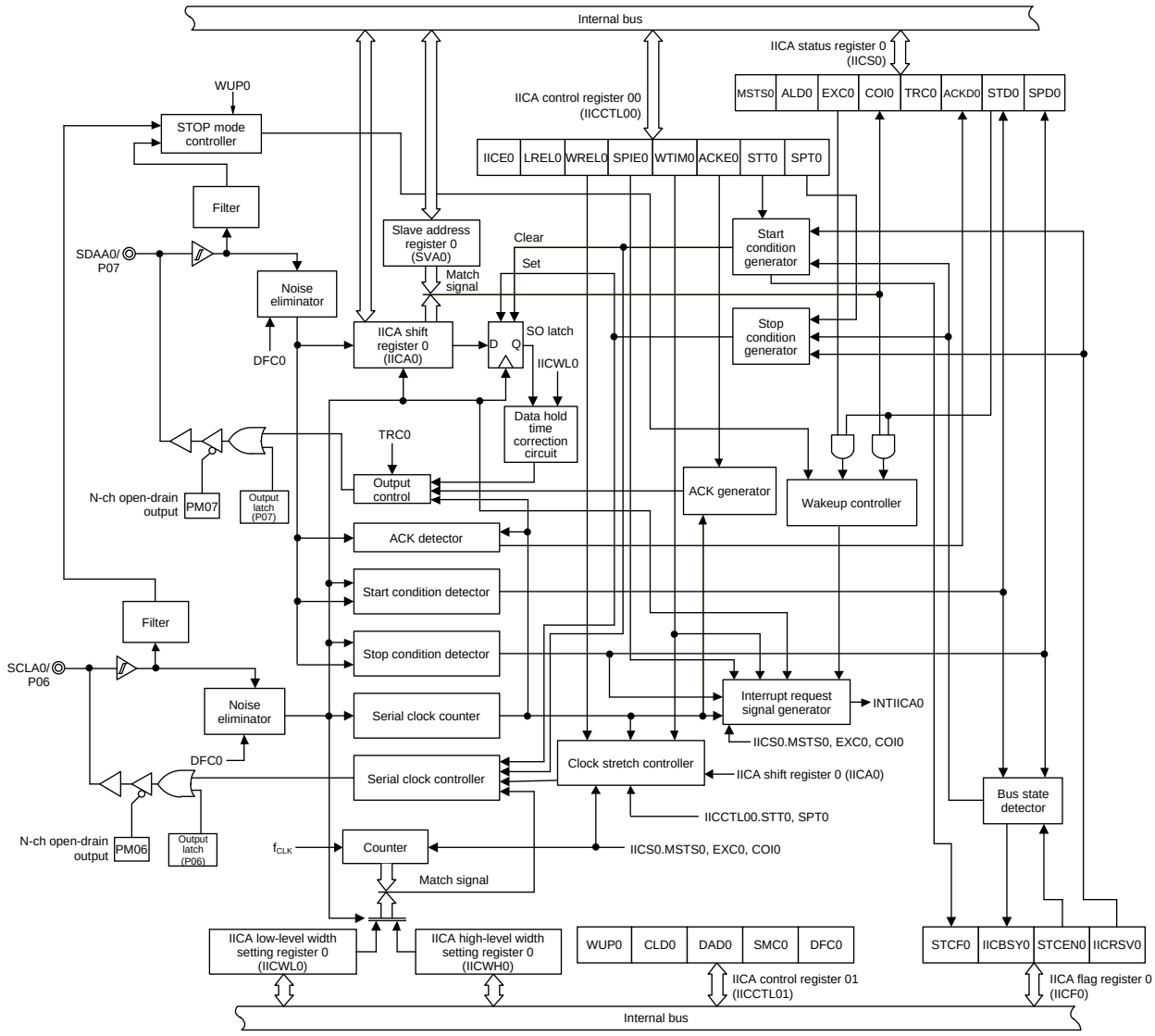
Since the SCLA0 and SDAA0 pins are used for open drain outputs, serial interface IICA requires pull-up resistors for the serial clock line and the serial data bus line.

3) Wakeup mode

The STOP mode can be released by generating an interrupt request signal (INTIICA0) when an extension code from the master or the local address has been received while in STOP mode. This can be set by using the WUP0 bit of IICA control register 01 (IICCTL01).

Figure 14-1 shows a block diagram of serial interface IICA.

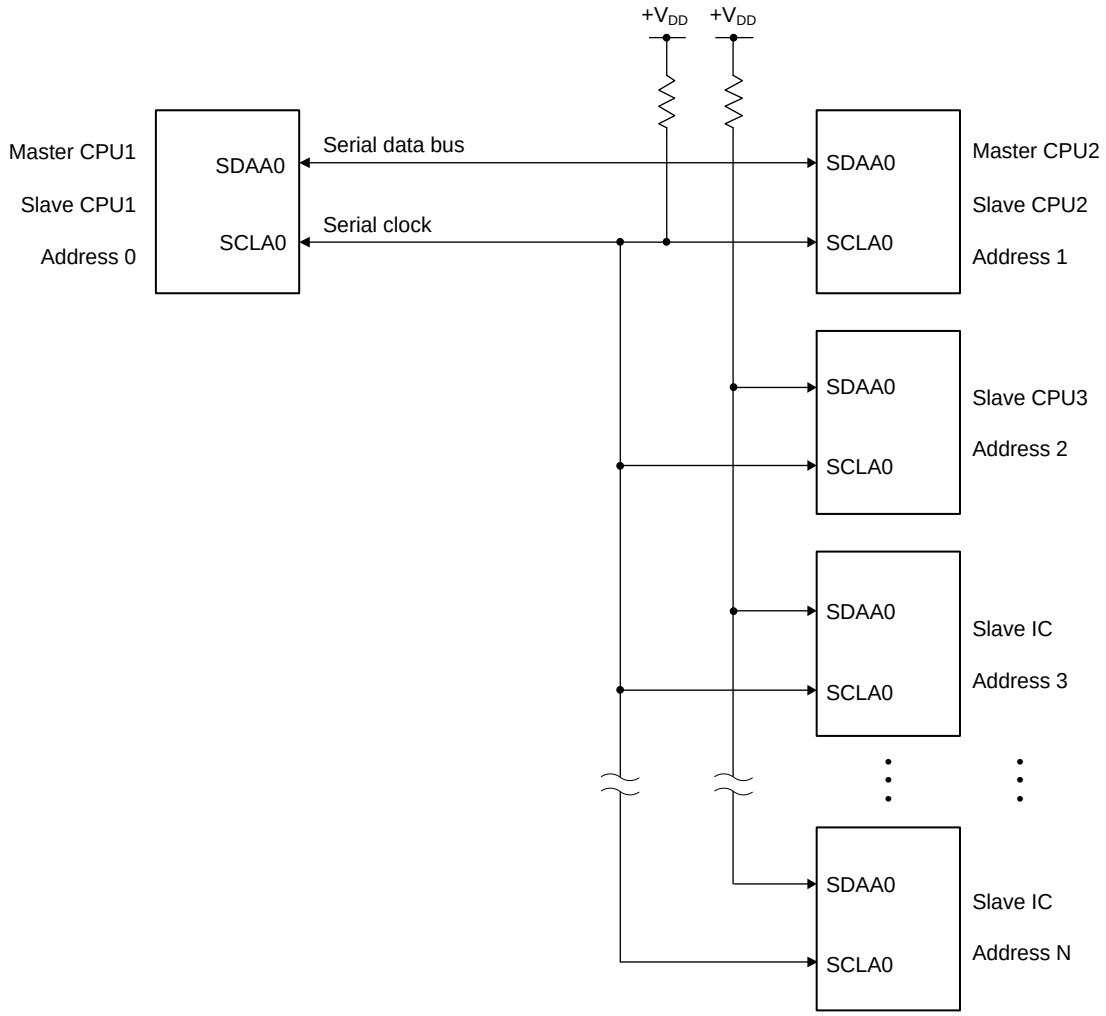
Figure 14-1. Block Diagram of Serial Interface IICA



Remark The IICA pins in this figure are when PIOR32 and PIOR33 = 0 for 16- and 20-pin products.

Figure 14-2 shows a serial bus configuration example.

Figure 14-2. Serial Bus Configuration Example Using I^2C Bus



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 0 (IICA0) Slave address register 0 (SVA0)
Control registers	Peripheral enable register 0 (PER0) IICA control register 00 (IICCTL00) IICA status register 0 (IICS0) IICA flag register 0 (IICF0) IICA control register 01 (IICCTL01) IICA low-level width setting register 0 (IICWL0) IICA high-level width setting register 0 (IICWH0) Port mode registers 0, 1, 6 (PM0, PM1, PM6) Port registers 0, 1, 6 (P0, P1, P6) Port output mode registers 0, 1 (POM0, POM1) Port mode control register 0 (PMC0)

(1) IICA shift register 0 (IICA0)

The IICA0 register is used to convert 8-bit serial data to 8-bit parallel data and vice versa in synchronization with the serial clock. The IICA0 register can be used for both transmission and reception.

The actual transmit and receive operations can be controlled by writing to and reading from the IICA0 register.

Writing to the IICA0 register during the clock stretch period releases clock stretching and starts data transfer.

The IICA0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 14-3. Format of IICA Shift Register 0 (IICA0)

Address: FFF50H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IICA0								

Caution 1. Do not write data to the IICA0 register during data transfer.

Caution 2. Write to or read from the IICA0 register only during the clock stretch period. Access to the IICA0 register in the communication state other than during the clock stretch period is prohibited. When the device serves as the master, however, the IICA0 register can be written only once after the communication trigger bit (STT0) is set to 1.

Caution 3. When communication is reserved, write data to the IICA0 register after the interrupt triggered by a stop condition is detected.

(2) Slave address register 0 (SVA0)

This register holds seven bits (A6, A5, A4, A3, A2, A1, and A0) of the local address when in slave mode.

The SVA0 register can be set by an 8-bit memory manipulation instruction.

However, rewriting to this register is prohibited while STD0 = 1 (the start condition is detected).

Reset signal generation clears this register to 00H.

Figure 14-4. Format of Slave Address Register 0 (SVA0)

Address: F0234H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SVA0	A6	A5	A4	A3	A2	A1	A0	0Note 1

Note 1. Be sure to clear bit 0 to 0.

(3) SO latch

The SO latch is used to retain the output level of the SDAA0 pin.

(4) Wakeup controller

This circuit generates an interrupt request (INTIICA0) when the received address matches the address value set in slave address register 0 (SVA0) or when an extension code is received.

(5) Serial clock counter

This counter counts the serial clock cycles that are output or input during transmit/receive operations and is used to verify that 8-bit data has been transmitted or received.

(6) Interrupt request signal generator

This circuit controls the generation of an interrupt request signal (INTIICA0).

An I^2C interrupt request is generated by the following two triggers.

- Falling edge of 8th or 9th clock of the serial clock (set by the WTIM0 bit)
- Interrupt request generated when a stop condition is detected (set by the SPIE0 bit)

Remark WTIM0 bit: Bit 3 of IICA control register 00 (IICCTL00)
SPIE0 bit: Bit 4 of IICA control register 00 (IICCTL00)

(7) Serial clock controller

In master mode, this circuit generates the clock for output to the SCLA0 pin from a sampling clock.

(8) Clock stretch controller

This circuit controls the timing of clock stretching.

(9) ACK generator, stop condition detector, start condition detector, and ACK detector

These circuits generate and detect each state.

(10) Data hold time correction circuit

This circuit generates the hold time for data after the falling edge of the serial clock.

(11) Start condition generator

This circuit generates a start condition when the STT0 bit is set to 1.

However, while communication reservation is disabled (IICRSV0 bit = 1) and the bus is not released (IICBSY0 bit = 1), start condition requests are ignored and the STCF0 bit is set to 1.

(12) Stop condition generator

This circuit generates a stop condition when the SPT0 bit is set to 1.

(13) Bus state detector

This circuit detects whether or not the bus is released by detecting a start condition or stop condition.

However, as the bus state cannot be detected immediately after the operation, use the STCEN0 bit to set the initial state of this circuit.

Remark STT0 bit: Bit 1 of IICA control register 00 (IICCTL00)
SPT0 bit: Bit 0 of IICA control register 00 (IICCTL00)
IICRSV0 bit: Bit 0 of IICA flag register 0 (IICF0)
IICBSY0 bit: Bit 6 of IICA flag register 0 (IICF0)
STCF0 bit: Bit 7 of IICA flag register 0 (IICF0)
STCEN0 bit: Bit 1 of IICA flag register 0 (IICF0)

14.3 Registers Controlling Serial Interface IICA

Serial interface IICA is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- IICA control register 00 (IICCTL00)
- IICA flag register 0 (IICF0)
- IICA status register 0 (IICS0)
- IICA control register 01 (IICCTL01)
- IICA low-level width setting register 0 (IICWL0)
- IICA high-level width setting register 0 (IICWH0)
- Port mode registers 0, 1, 6 (PM0, PM1, PM6)
- Port registers 0, 1, 6 (P0, P1, P6)
- Port output mode registers 0, 1 (POM0, POM1)
- Port mode control register 0 (PMC0)

14.3.1 Peripheral enable register 0 (PER0)

The PER0 register is used to enable or disable the supply of a clock signal to various on-chip peripheral modules. Clock supply to an on-chip peripheral module that is not to be used can be stopped to decrease power consumption and noise.

When serial interface IICA is used, be sure to set bit 4 (IICA0EN) 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	TMKAEN	CMPEN	ADCEN	IICA0EN	SAU1EN	SAU0EN	0	TAU0EN

IICA0EN	Control of serial interface IICA input clock supply
0	Stops supply of an input clock. • The SFRs used by serial interface IICA cannot be written. • Serial interface IICA is in the reset state.
1	Enables supply of an input clock. • The SFRs used by serial interface IICA can be read/written.

Caution 1. When setting serial interface IICA, make sure that the setting of the IICA0EN bit is 1 before setting the following registers. If IICA0EN = 0, the values of the registers which control the serial interface IICA are cleared to their initial values, and writing to them is ignored (except for port mode registers 0, 1, 6 (PM0, PM1, PM6), port registers 0, 1, 6 (P0, P1, P6), port output mode registers 0, 1 (POM0, POM1), and port mode control register 0 (PMC0)).

- **IICA control register 00 (IICCTL00)**
- **IICA flag register 0 (IICF0)**
- **IICA status register 0 (IICS0)**
- **IICA control register 01 (IICCTL01)**
- **IICA low-level width setting register 0 (IICWL0)**
- **IICA high-level width setting register 0 (IICWH0)**

Caution 2. Be sure to clear the following bits to 0.
10- and 16-pin products: Bits 1 and 3
20-, 24-, and 32-pin products: Bit 1

14.3.2 IICA control register 00 (IICCTL00)

This register is used to enable or stop I^2C operations, set the timing of clock stretching, and set other I^2C operations.

The IICCTL00 register can be set by a 1-bit or 8-bit memory manipulation instruction. Note that the SPIE0, WTIM0, and ACKE0 bits must be set while the setting of IICE0 is 0 or during the clock stretch period. These bits can be set at the same time when the IICE0 bit is set from 0 to 1.

Reset signal generation clears this register to 00H.

Figure 14-6. Format of IICA Control Register 00 (IICCTL00) (1/4)

Address: F0230H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IICCTL00	IICE0	LREL0	WREL0	SPIE0	WTIM0	ACKE0	STT0	SPT0

IICE0	I^2C operation enable
0	Stops operation. Resets IICA status register 0 (IICS0)[Note 1]. Also stops internal operation.
1	Enables operation.
Be sure to set this bit (1) while the SCLA0 and SDAA0 lines are at the high level.	
Condition for clearing (IICE0 = 0)	Condition for setting (IICE0 = 1)
• Cleared by instruction • Reset	• Set by instruction

LREL0 Note 2, Note 3	Exit from communications
0	Normal operation
1	The device exits from the current communications and enters standby mode. This setting is automatically cleared to 0 after having been executed. This bit is used when an extension code not related to the local station is received. The SCLA0 and SDAA0 lines go into the high impedance state. The following flags of IICA control register 00 (IICCTL00) and IICA status register 0 (IICS0) are cleared to 0. STT0, SPT0, MSTS0, EXC0 COI0, TRC0, ACKD0, STD0
The standby mode following exit from communications remains in effect until the following communication participation conditions are met. • After a stop condition is detected, startup is in master mode. • An address match or extension code reception after the start condition	
Condition for clearing (LREL0 = 0)	Condition for setting (LREL0 = 1)
• Automatically cleared after execution • Reset	• Set by instruction

Figure 14-6. Format of IICA Control Register 00 (IICCTL00) (2/4)

WREL0 Note 2, Note 3	Release clock stretching
0	Clock stretching is not released.
1	Clock stretching is released. This bit is automatically cleared to 0 after clock stretching has been released.
When the WREL0 bit is set (clock stretching is released) during the clock stretch period at the 9th clock pulse in the transmission state (TRC0 = 1), the SDAA0 line goes into the high impedance state (TRC0 = 0).	
Condition for clearing (WREL0 = 0)	Condition for setting (WREL0 = 1)
• Automatically cleared after execution • Reset	• Set by instruction

SPIE0Note 4	Enable/disable generation of interrupt request when stop condition is detected
0	Disable
1	Enable
If the WUP0 bit of IICA control register 01 (IICCTL01) is 1, no stop condition interrupt will be generated even if SPIE0 = 1.	
Condition for clearing (SPIE0 = 0)	Condition for setting (SPIE0 = 1)
• Cleared by instruction • Reset	• Set by instruction

WTIM0Note 4	Control of clock stretching and interrupt request generation
0	An interrupt request is generated on the falling edge of the 8th clock. Master mode: After the output of 8 clock pulses, the clock is stretched while the clock output is at the low level. Slave mode: After the input of 8 clock pulses, the clock is set to the low level to stretch the master's clock.
1	An interrupt request is generated on the falling edge of the 9th clock. Master mode: After the output of 9 clock pulses, the clock is stretched while the clock output is at the low level. Slave mode: After the input of 9 clock pulses, the clock is set to the low level to stretch the master's clock.
An interrupt is generated on the falling edge of the 9th clock during address transfer independently of the setting of this bit. The setting of this bit is valid after the completion of address transfer. In master mode, clock stretching is applied at the falling edge of the 9th clock during address transfer. For a slave that has received its local address, clock stretching is applied at the falling edge of the 9th clock after an acknowledge (ACK) has been issued. However, if the slave has received an extension code, clock stretching is applied at the falling edge of the 8th clock.	
Condition for clearing (WTIM0 = 0)	Condition for setting (WTIM0 = 1)
• Cleared by instruction • Reset	• Set by instruction

Figure 14-6. Format of IICA Control Register 00 (IICCTL00) (3/4)

ACKE0 Note 4, Note 5	Acknowledgment control
0	Disables acknowledgment.
1	Enables acknowledgment. During the 9th clock period, the SDAA0 line is set to low level.

Condition for clearing (ACKE0 = 0)	Condition for setting (ACKE0 = 1)
• Cleared by instruction • Reset	• Set by instruction

STT0 Note 2, Note 6	Start condition trigger
0	A start condition is not generated.
1	When the bus is released (in standby state, when IICBSY0 = 0): If this bit is set (1), a start condition is generated (startup as the master). When a third party is communicating: • When communication reservation is enabled (IICRSV0 = 0) Functions as the start condition reservation flag. When set to 1, automatically generates a start condition after the bus has been released. • When communication reservation is disabled (IICRSV0 = 1) The STT0 bit is cleared even if it is set (1), and the STT0 clear flag (STCF0) is set (1). No start condition is generated. In the clock stretch state (in master mode): Releases clock stretching to generate a restart condition.

Cautions concerning set timing

- For master reception: Setting this bit to 1 during transfer is prohibited. It can be set to 1 only during the clock stretch period after the ACKE0 bit has been cleared to 0 and the 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 following the output of the 9th clock.
- Setting this bit to 1 at the same time as the stop condition trigger (SPT0) is prohibited.
- Once the STT0 bit is set to 1, setting it to 1 again before the clear condition is met is prohibited.

Condition for clearing (STT0 = 0)	Condition for setting (STT0 = 1)
• Setting the STT0 bit to 1 while communication reservation is prohibited • A loss in arbitration • A start condition is generated by the master • Cleared by LREL0 = 1 (exit from communications) • When IICE0 = 0 (operation stop) • Reset	• Set by instruction

Figure 14-6. Format of IICA Control Register 00 (IICCTL00) (4/4)

SPT0[Note 7]	Stop condition trigger
0	A stop condition is not generated.
1	A stop condition is generated (termination of transfer as a master).

Cautions concerning set timing

- For master reception: Setting this bit to 1 during transfer is prohibited.
  It can be set to 1 only during the clock stretch period after the ACKE0 bit has been cleared to 0 and the slave has been notified of final reception.
- For master transmission: A stop condition cannot be generated normally during the acknowledge period. Therefore, set it to 1 during the clock stretch period following the output of the 9th clock.
- Setting this bit to 1 at the same time as the start condition trigger (STT0) is prohibited.
- Set the SPT0 bit to 1 only when in master mode.
- Note that if the SPT0 bit is set to 1 during the clock stretch period following the output of 8 clock pulses while WTIM0 = 0, a stop condition is generated during the high-level period of the 9th clock after clock stretching has been released. The WTIM0 bit should be changed from 0 to 1 during the clock stretch period following the output of 8 clock pulses, and the SPT0 bit should be set to 1 during the clock stretch period following the 9th clock output.
- Once the SPT0 bit is set to 1, setting it to 1 again before the clear condition is met is prohibited.

Condition for clearing (STT0 = 0)	Condition for setting (STT0 = 1)
• A loss in arbitration • Automatically cleared after the detection of the stop condition • Cleared by LREL0 = 1 (exit from communications) • When IICE0 = 0 (operation stop) • Reset	• Set by instruction

Note 1. The IICA status register 0 (IICA0), the STCF0 and IICBSY0 bits of IICA flag register 0 (IICF0), and the CLD0 and DAD0 bits of IICA control register 01 (IICCTL01) are reset.

Note 2. The signal of this bit is invalid while IICE0 = 0.

Note 3. Reading the LREL0 and WREL0 bits always returns 0.

Note 4. The signal of this bit is invalid while IICE0 = 0. Set this bit during that period.

Note 5. The set value is invalid during address transfer and if the code is not an extension code.
When the device serves as a slave and the addresses match, an acknowledgment is generated regardless of the set value.

Note 6. The STT0 bit is always read as 0.

Note 7. The SPT0 bit is always read as 0.

Caution 1. If the operation of I^2C is enabled (IICE0 = 1) when the SCLA0 line is at the high level, the SDAA0 line is at the low level, and the digital filter is turned on (DFC0 of the IICCTL01 register = 1), a start condition will be inadvertently detected immediately. In this case, set (1) the LREL0 bit by using a 1-bit memory manipulation instruction immediately after enabling operation of I^2C (IICE0 = 1).

Caution 2. While bit 3 (TRC0) of IICA status register 0 (IICS0) is 1 (transmission state), if bit 5 (WREL0) of IICA control register 00 (IICCTL00) is set to 1 at the 9th clock to release clock stretching, the TRC0 bit is cleared (reception state) and the SDAA0 line is set to the high impedance state. Release clock stretching while the TRC0 bit is 1 (transmission state) by writing to IICA shift register 0.

Remark IICRSV0: Bit 0 of IICA flag register 0 (IICF0)
STCF0: Bit 7 of IICA flag register 0 (IICF0)

14.3.3 IICA status register 0 (IICS0)

This register indicates the state of the I^2C.

The IICS0 register can only be read by a 1-bit or 8-bit memory manipulation instruction while the setting of STT0 is 1 or during the clock stretch period.

Reset signal generation clears this register to 00H.

Caution Reading the IICS0 register while the address match wakeup function is enabled (WUP0 = 1) in STOP mode is prohibited. When the WUP0 bit is changed from 1 to 0 (wakeup operation is stopped), regardless of the INTIICA0 interrupt request signal, the change in the state is not reflected until the next start condition or stop condition is detected. To use the wakeup function, therefore, enable the generation of an interrupt on detection of a stop condition (SPIE0 = 1) and read the IICS0 register after the interrupt has been detected.

Remark STT0: Bit 1 of IICA control register 00 (IICCTL00)
WUP0: Bit 7 of IICA control register 01 (IICCTL01)

Figure 14-7. Format of IICA Status Register 0 (IICS0) (1/3)

Address: FFF51H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
IICS0	MSTS0	ALD0	EXC0	COI0	TRC0	ACKD0	STD0	SPD0

MSTS0	Master state check flag
0	Slave state or communication standby state
1	Master communication state

Condition for clearing (MSTS0 = 0)	Condition for setting (MSTS0 = 1)
• When a stop condition is detected • When ALD0 = 1 (arbitration loss) • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is generated

ALD0	Detection of arbitration loss
0	Indicates that no arbitration has occurred or a win in arbitration.
1	Indicates a loss in arbitration. The MSTS0 bit is cleared.

Condition for clearing (ALD0 = 0)	Condition for setting (ALD0 = 1)
• Automatically cleared after the IICS0 register has been read[Note 1] • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• A loss in arbitration

Figure 14-7. Format of IICA Status Register 0 (IICS0) (2/3)

EXC0	Detection of extension code reception
0	Extension code was not received.
1	Extension code was received.

Condition for clearing (EXC0 = 0)	Condition for setting (EXC0 = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When the higher four bits of the received address data is either “0000” or “1111” (set at the rising edge of the 8th clock).

COI0	Address match detection
0	The addresses do not match.
1	The addresses match.

Condition for clearing (COI0 = 0)	Condition for setting (COI0 = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When the received address matches the local address (slave address register 0 (SVA0)) (set at the rising edge of the 8th clock).

TRC0	Transmission/reception state detection
0	Reception state (not in the transmission state). The SDAA0 line is set to high impedance.
1	Transmission state. Enable the output of the value in the SO0 latch to the SDAA0 line (valid after the falling edge of the 9th clock of the first byte).

Condition for clearing (TRC0 = 0)	Condition for setting (TRC0 = 1)
<Both master and slave> • When a stop condition is detected • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Cleared by WREL0 = 1 (clock stretching released)Note 2 • When the ALD0 bit changes from 0 to 1 (arbitration loss) • Reset • When not participating in communications (MSTS0, EXC0, COI0 = 0) <Master> • When 1 is output to the LSB (transfer direction specification bit) of the first byte <Slave> • When a start condition is detected • When 0 is input to the LSB (transfer direction specification bit) of the first byte	<Master> • When a start condition is generated • When 0 (master transmission) is output to the LSB (transfer direction specification bit) of the first byte (during address transfer) <Slave> • When 1 (slave transmission) is input to the LSB (transfer direction specification bit) of the first byte from the master (during address transfer)

Figure 14-7. Format of IICA Status Register 0 (IICS0) (3/3)

ACKD0	Acknowledge (ACK) detection
0	Acknowledge was not detected.
1	Acknowledge was detected.

Condition for clearing (ACKD0 = 0)	Condition for setting (ACKD0 = 1)
• When a stop condition is detected • At the rising edge of the first clock of the next byte • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When the SDAA0 line is at the low level at the rising edge of the 9th clock of the SCLA0 line

STD0	Start condition detection
0	A start condition was not detected.
1	A start condition was detected. This indicates that the address transfer period is in effect.

Condition for clearing (STD0 = 0)	Condition for setting (STD0 = 1)
• When a stop condition is detected • At the rising edge of the first clock of the next byte following address transfer • Cleared by LREL0 = 1 (exit from communications) • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When a start condition is detected

SPD0	Stop condition detection
0	A stop condition was not detected.
1	A stop condition was detected. Communication by the master is terminated and the bus is released.

Condition for clearing (SPD0 = 0)	Condition for setting (SPD0 = 1)
• At the rising edge of the first clock of the address transfer byte following setting of this bit and detection of a start condition • When the WUP0 bit changes from 1 to 0 • When the IICE0 bit changes from 1 to 0 (operation stop) • Reset	• When a stop condition is detected

Note 1. The ALD0 bit is also cleared when a 1-bit memory manipulation instruction is executed for another bit in the IICS0 register. Therefore, when using the ALD0 bit, read the data of this bit before the data of the other bits.

Note 2. While bit 3 (TRC0) of IICA status register 0 (IICS0) is 1 (transmission state), if bit 5 (WREL0) of IICA control register 00 (IICCTL00) is set to 1 at the 9th clock to release clock stretching, the TRC0 bit is cleared (reception state) and the SDAA0 line is set to the high impedance state. Release clock stretching while the TRC0 bit is 1 (transmission state) by writing to IICA shift register 0.

Remark LREL0: Bit 6 of IICA control register 00 (IICCTL00)
IICE0: Bit 7 of IICA control register 00 (IICCTL00)

14.3.4 IICA flag register 0 (IICF0)

This register sets the operation mode of I^2C and indicates the state of the I^2C bus.

The IICF0 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the STT0 clear flag (STCF0) and I^2C bus status flag (IICBSY0) bits are read-only.

The IICRSV0 bit can be used to enable or disable communication reservation.

The STCEN0 bit can be used to set the initial value of the IICBSY0 bit.

The IICRSV0 and STCEN0 bits can only be written while operation of I^2C is disabled (bit 7 (IICE0) of IICA control register 00 (IICCTL00) = 0). The IICF0 register is only readable after the operation is enabled.

Reset signal generation clears this register to 00H.

Figure 14-8. Format of IICA Flag Register 0 (IICF0) (1/2)

Address: FFF52H After reset: 00H R/W[Note 1]

Symbol	7	6	5	4	3	2	1	0
IICF0	STCF0	IICBSY0	0	0	0	0	STCEN0	IICRSV0

STCF0	STT0 clear flag
0	A start condition is generated.
1	A start condition cannot be generated and the STT0 flag is cleared.

Condition for clearing (STCF0 = 0)	Condition for setting (STCF0 = 1)
• Cleared by STT0 = 1 • When IICE0 = 0 (operation stop) • Reset	• A start condition cannot be generated when communication reservation is disabled (IICRSV0 = 1) and the STT0 bit is cleared to 0.

IICBSY0	I^2C bus status flag
0	Bus released state (the initial state of communication when STCEN0 = 1)
1	Bus communication state (the initial state of communication when STCEN0 = 0)

Condition for clearing (IICBSY0 = 0)	Condition for setting (IICBSY0 = 1)
• When a stop condition is detected • When IICE0 = 0 (operation stop) • Reset	• When a start condition is detected • Setting of the IICE0 bit when STCEN0 = 0

STCEN0	Initial start enable trigger
0	After operation is enabled (IICE0 = 1), enable generation of a start condition upon detection of a stop condition.
1	After operation is enabled (IICE0 = 1), enable generation of a start condition without detecting a stop condition.

Condition for clearing (STCEN0 = 0)	Condition for setting (STCEN0 = 1)
• Cleared by instruction • When a start condition is detected • Reset	• Set by instruction

Figure 14-8. Format of IICA Flag Register 0 (IICF0) (2/2)

IICRSV0	Communication reservation function disable bit
0	Enable communication reservation.
1	Disable communication reservation.

Condition for clearing (IICRSV0 = 0)	Condition for setting (IICRSV0 = 1)
• Cleared by instruction • Reset	• Set by instruction

Note 1. Bits 7 and 6 are read-only.

Caution 1. Write to the STCEN0 bit only when the operation is stopped (IICE0 = 0).

Caution 2. When STCEN0 = 1, the bus released state (IICBSY0 = 0) is recognized regardless of the actual bus state. Therefore, when generating a first start condition (STT0 = 1), it is necessary to verify that third party communications are not being conducted in order to prevent other communications from being destroyed.

Caution 3. Write to the IICRSV0 bit only when the operation is stopped (IICE0 = 0).

Remark STT0: Bit 1 of IICA control register 00 (IICCTL00)
IICE0: Bit 7 of IICA control register 00 (IICCTL00)

14.3.5 IICA control register 01 (IICCTL01)

This register is used to set the operation mode of I^2C and detect the states of the SCLA0 and SDAA0 pins.

The IICCTL01 register can be set by a 1-bit or 8-bit memory manipulation instruction. However, the CLD0 and DAD0 bits are read-only.

Set the IICCTL01 register, except the WUP0 bit, while operation of I^2C is disabled (bit 7 (IICE0) of IICA control register 00 (IICCTL00) = 0).

Reset signal generation clears this register to 00H.

Figure 14-9. Format of IICA Control Register 01 (IICCTL01) (1/2)

Address: F0231H After reset: 00H R/W[Note 1]

Symbol	7	6	5	4	3	2	1	0
IICCTL01	WUP0	0	CLD0	DAD0	SMC0	DFC0	0	0

WUP0	Control of address match wakeup
0	Stops operation of address match wakeup while in STOP mode.
1	Enables operation of address match wakeup while in STOP mode.
To shift to STOP mode when WUP0 = 1, execute the STOP instruction at least three cycles of f_{CLK} after setting (1) the WUP0 bit (see **Figure 14-21 Flow when Setting WUP0 =** 1). Clear (0) the WUP0 bit after an address match or reception of an extension code. Clearing (0) the WUP0 bit allows participation in subsequent communications (releasing clock stretching and writing transmit data must follow clearing (0) the WUP0 bit). The interrupt timing on an address match or extension code reception while WUP0 = 1 is the same as the interrupt timing when WUP0 = 0 (a delay difference equivalent to the sampling error by the clock is generated). Furthermore, when WUP0 = 1, a stop condition interrupt is not generated even if the SPIE0 bit is set to 1.	
Condition for clearing (WUP0 = 0)	Condition for setting (WUP0 = 1)
• Cleared by instruction (after address match or extension code reception)	• Set by instruction (when MSTS0, EXC0, and COI0 = 0 and STD0 = 0 (non-participation in communications)[Note 2]

Figure 14-9. Format of IICA Control Register 01 (IICCTL01) (2/2)

CLD0	SCLA0 pin level detection (valid only when IICE0 = 1)
0	The SCLA0 pin was detected to be at the low level.
1	The SCLA0 pin was detected to be at the high level.

Condition for clearing (CLD0 = 0)	Condition for setting (CLD0 = 1)
• When the SCLA0 pin is at the low level • When IICE0 = 0 (operation stop) • Reset	• When the SCLA0 pin is at the high level

DAD0	SDAA0 pin level detection (valid only when IICE0 = 1)
0	The SDAA0 pin was detected be at the low level.
1	The SDAA0 pin was detected to be at high level.

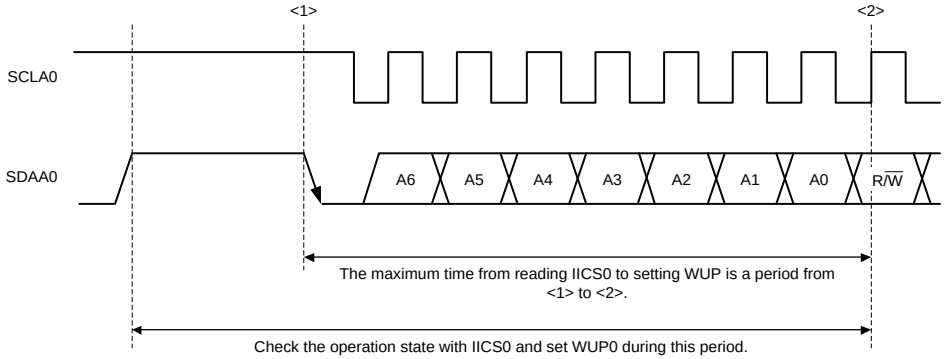
Condition for clearing (DAD0 = 0)	Condition for setting (DAD0 = 1)
• When the SDAA0 pin is at the low level • When IICE0 = 0 (operation stop) • Reset	• When the SDAA0 pin is at the high level

SMC0	Operation mode switching
0	Operates in standard mode (maximum transfer rate: 100 kbps).
1	Operates in fast mode (maximum transfer rate: 400 kbps)

DFC0	Digital filter operation control
0	Digital filter off
1	Digital filter on
Use the digital filter only in fast mode. The digital filter is used for noise elimination. Set (1) or clearing (0) the DFC0 bit does not vary the transfer clock.	

Note 1. Bits 5 and 4 are read-only.

Note 2. The WUP0 bit must be set after checking the state of IICA status register 0 (IICS0) during the period shown below.



Caution When setting the transfer clock, take care with the minimum operation frequency of f_{CLK}. The minimum operation frequency of f_{CLK} for serial interface IICA is determined according to the mode.

Normal mode: f_{CLK} = 1 MHz (min.)
Fast mode: f_{CLK} = 3.5 MHz (min.)

Remark IICE0: Bit 7 of IICA control register 00 (IICCTL00)

14.3.6 IICA low-level width setting register 0 (IICWL0)

This register is used to set the low-level width (tLOW) of the SCLA0 pin signal that is output by serial interface IICA and to control the SDAA0 pin signal.

The IICWL0 register can be set by an 8-bit memory manipulation instruction.

Set the IICWL0 register while operation of I^2C is disabled (bit 7 (IICE0) of IICA control register 00 (IICCTL00) = 0).

Reset signal generation set this register to FFH.

For details about setting the IICWL0 register, see **14.4.2 Setting transfer clock by using IICWL0 and IICWH0 registers**.

The data hold time is one-quarter of the time set by the IICWL0 register.

Figure 14-10. Format of IICA Low-Level Width Setting Register 0 (IICWL0)

Address: F0232H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWL0								

14.3.7 IICA high-level width setting register 0 (IICWH0)

This register is used to set the high-level width of the SCLA0 pin signal that is output by serial interface IICA and to control the SDAA0 pin signal.

The IICWH0 register can be set by an 8-bit memory manipulation instruction.

Set the IICWH0 register while operation of I^2C is disabled (bit 7 (IICE0) of IICA control register 00 (IICCTL00) = 0).

Reset signal generation set this register to FFH.

Figure 14-11. Format of IICA High-Level Width Setting Register 0 (IICWH0)

Address: F0233H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWH0								

Remark For the procedure of setting the transfer clock on the master side, see **14.4.2(1) Setting transfer clock on master side**. For the procedure of setting the IICWL0 and IICWH0 registers on the slave side, see **14.4.2(2) Setting IICWL0 and IICWH0 registers on slave side**.

14.3.8 Registers controlling port functions of IICA serial input/output pins

When the IICA is to be used, set the registers that control the port functions multiplexed on the IICA serial I/O pins (SCLA0 and SDAA0 pins): port mode register (PM0), port register (P0), port output mode register (POM0), and port mode control register (PMC0).

For details on the registers that control the port functions, see **4.3.1 Port mode registers 0, 1, 2, 4, 6, 12 (PM0, PM1, PM2, PM4, PM6, PM12)**, **4.3.2 Port registers 0, 1, 2, 4, 6, 12, 13 (P0, P1, P2, P4, P6, P12, P13)**, **4.3.4 Port input mode registers 0, 1, 2, 4 (POM0, POM1, POM2, POM4)**, and **4.3.5 Port mode control registers 0, 2 (PMC0, PMC2)**.

When you intend to use the clock I/O pin (SCLA0) and serial data I/O pin (SDAA0) of IICA0, set the corresponding bits in port mode register (PM0) and port mode control register (PMC0) to 0 and the corresponding bits in port register (P0) and port output mode register (POM0) to 1. For details, see **4.5.3 Register setting examples for used port and alternate functions**.

Since these pins are used as N-ch open-drain outputs (with a withstand voltage of V_{DD}), use a resistor to pull them up to the power-supply voltage of the external device.

14.4 I^2C Bus Mode Functions

14.4.1 Pin configuration

The serial clock pin (SCLA0) and the serial data bus pin (SDAA0) are configured as follows.

(1) SCLA0

This pin is used for serial clock input and output.

This pin is an N-ch open-drain output for both master and slave. Input is Schmitt input.

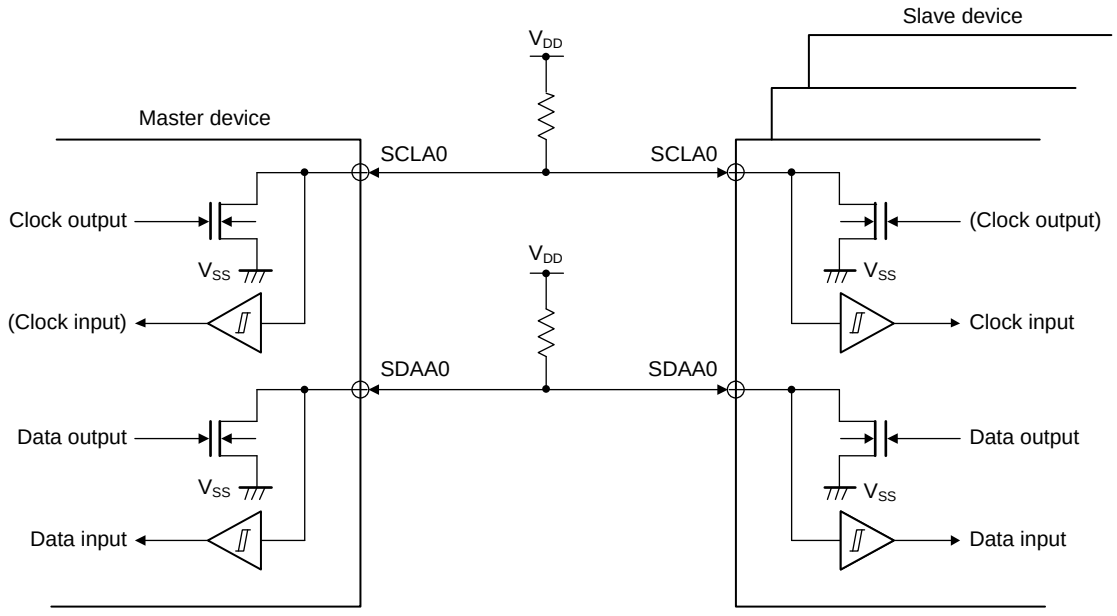
(2) SDAA0

This pin is used for serial data input and output.

This pin is an N-ch open-drain output for both master and slave. 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 14-12. Pin Configuration Diagram



14.4.2 Setting transfer clock by using IICWL0 and IICWH0 registers

(1) Setting transfer clock on master side

$$\text{Transfer clock} = \frac{f_{CLK}}{IICWL + IICWH + f_{CLK}(t_R + t_F)}$$

At this time, the optimal setting values of the IICWL0 and IICWH0 registers are as follows.

(The fractional parts of all setting values are rounded up.)

- In fast mode

$$IICWL0 = \frac{0.52}{\text{Transfer clock}} \times f_{CLK}$$

$$IICWH0 = \left(\frac{0.48}{\text{Transfer clock}} - t_R - t_F\right) \times f_{CLK}$$

- In normal mode

$$IICWL0 = \frac{0.47}{\text{Transfer clock}} \times f_{CLK}$$

$$IICWH0 = \left(\frac{0.53}{\text{Transfer clock}} - t_R - t_F\right) \times f_{CLK}$$

(2) Setting IICWL0 and IICWH0 registers on slave side

(The fractional parts of all setting values are rounded up.)

- In fast mode

$$\mathrm{IICWL0} = 1.3\ \mu\mathrm{s} \times \mathrm{f_{CLK}}$$

$$\mathrm{IICWH0} = (1.2\ \mu\mathrm{s} - \mathrm{t_R} - \mathrm{t_F}) \times \mathrm{f_{CLK}}$$

- In normal mode

$$\mathrm{IICWL0} = 4.7\ \mu\mathrm{s} \times \mathrm{f_{CLK}}$$

$$\mathrm{IICWH0} = (5.3\ \mu\mathrm{s} - \mathrm{t_R} - \mathrm{t_F}) \times \mathrm{f_{CLK}}$$

Caution When setting the transfer clock, take care with the minimum operation frequency of f_{CLK}. The minimum operation frequency of f_{CLK} for serial interface IICA is determined according to the mode.
Fast mode: f_{CLK} = 3.5 MHz (min.)
Normal mode: f_{CLK} = 1 MHz (min.)

Remark 1. Calculate the rise time (t_R) and fall time (t_F) of the SDAA0 and SCLA0 signals separately, because they differ depending on the pull-up resistance and wiring capacity.

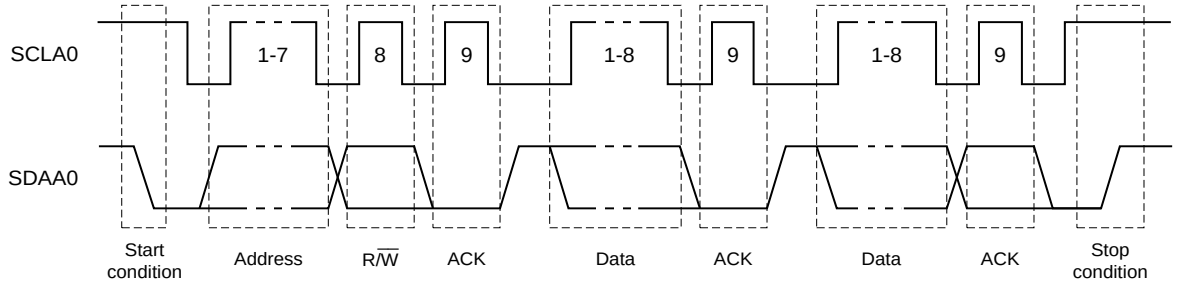
Remark 2. IICWL0: IICA low-level width setting register 0
IICWH0: ICA high-level width setting register 0
t_F: SDAA0 and SCLA0 signal fall time
t_R: SDAA0 and SCLA0 signal rise time
f_{CLK}: CPU/peripheral hardware clock frequency

14.5 I^2C Bus Definitions and Control Methods

The following describes the serial data communication format of the I^2C bus and the signals to be used.

Figure 14-13 shows the timing of transfer of the “start condition”, “address”, “data”, and “stop condition” which are generated on the serial data bus of the I^2C bus.

Figure 14-13. I^2C Bus Serial Data Transfer Timing



The master generates a start condition, slave address, and stop condition.

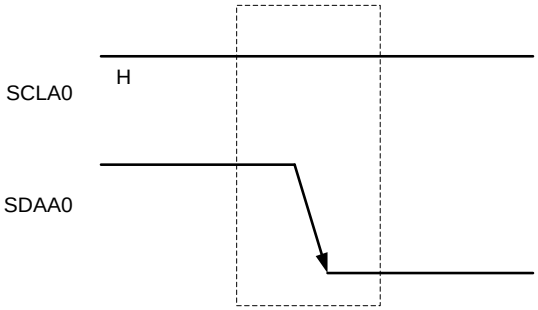
The acknowledge (ACK) can be generated by either the master or slave (normally, it is output by the receiver of 8-bit data).

The serial clock (SCLA0) is continuously output by the master. However, for the slave, a low-level period of the SCLA0 pin can be extended and clock stretching can be inserted.

14.5.1 Start condition

A start condition is generated when the SDAA0 pin changes from the high to the low level while the SCLA0 pin is at the high level. A start condition is a signal that the master generates to the slave when starting a serial transfer. When the device is used as a slave, a start condition can be detected.

Figure 14-14. Start Condition



A start condition is output when bit 1 (STT0) of IICA control register 00 (IICCTL00) is set (1) after a stop condition has been detected (SPD0: Bit 0 of IICA status register 0 (IICS0) = 1). When a start condition is detected, bit 1 (STD0) of the IICS0 register is set (1).

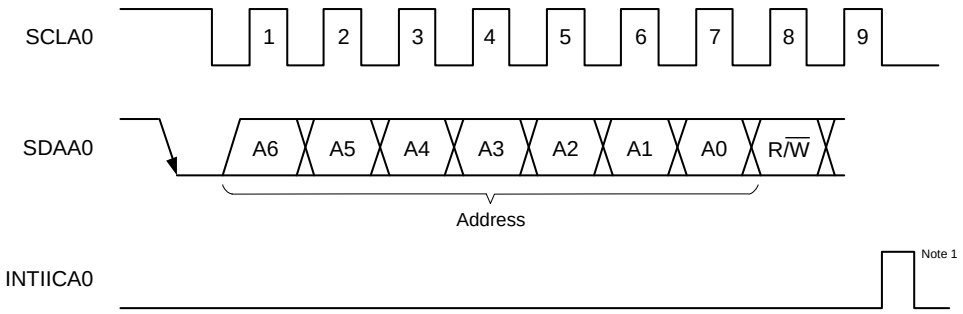
14.5.2 Address

7-bit data that follows a start condition is defined as address.

An address is a 7-bit data segment that is output in order for the master to select a certain slave from among multiple slaves connected to the bus line. Therefore, slaves on the bus line must have different addresses.

A slave detects this condition by the hardware and checks whether the 7-bit data matches the value of slave address register 0 (SVA0). If the 7-bit data matches the value of the SVA0 register at this time, that slave is selected and it communicates with the master until the master generates a start condition or stop condition.

Figure 14-15. Address



Note 1. INTIICA0 is not generated if data other than the local address or extension code is received in slave operation.

The address is output when 8 bits consisting of the slave address and the transfer direction described in **14.5.3 Transfer direction specification** are written to IICA shift register 0 (IICA0). The received address is written to the IICA0 register.

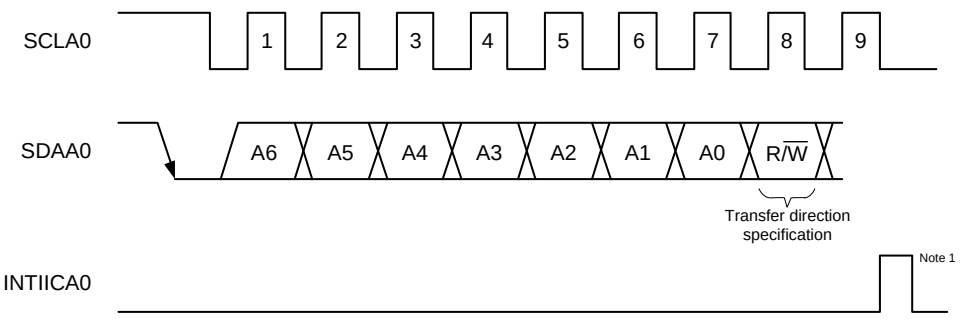
The slave address is assigned to the higher 7 bits of the IICA0 register.

14.5.3 Transfer direction specification

Following the 7-bit address, the master sends 1-bit data to specify the transfer direction.

When this transfer direction specification bit is 0, it indicates that the master is transmitting data to a slave. When the transfer direction specification bit is 1, it indicates that the master is receiving data from a slave.

Figure 14-16. Transfer Direction Specification



Note 1. INTIICA0 is not generated if data other than the local address or extension code is received in slave operation.

14.5.4 Acknowledge (ACK)

ACK is used to check the state of serial data on the transmission and reception sides.

The reception side returns ACK each time it receives 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. The detection of ACK can be checked by using bit 2 (ACKD0) of IICA status register 0 (IICS0).

When the master receives the last data item, it does not return ACK but generates a stop condition. If a slave does not return ACK in reception, the master outputs a stop condition or restart condition and stops transmission. If ACK is not returned, the following causes can be considered.

<1> Reception was not performed normally.

<2> The final data item was received.

<3> The reception side specified by the address does not exist.

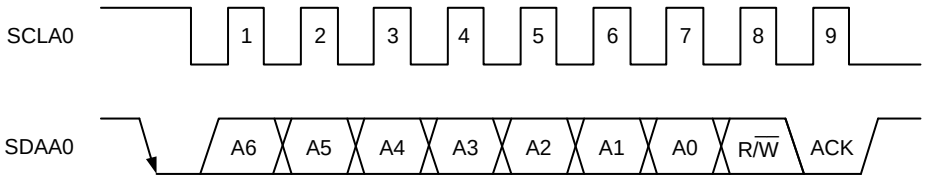
To generate ACK, the reception side sets the SDAA0 line to the low level at the 9th clock (normal reception).

Automatic generation of ACK is enabled by setting bit 2 (ACKE0) of IICA control register 00 (IICCTL00) to 1. Bit 3 (TRC0) of the IICS0 register is set by the 8th bit data that follows 7-bit address information. Usually, set the ACKE0 bit to 1 for reception (TRC0 = 0).

When the slave can no longer receive data during reception (TRC0 = 0) or does not require the next data item, the slave must clear the ACKE0 bit to 0 to inform the master that it cannot receive any more data.

When the master does not require the next data item during reception (TRC0 = 0), it must clear the ACKE0 bit to 0 so that ACK is not generated. In this way, the master informs the slave (transmission side) of the end of data (transmission will be stopped).

Figure 14-17. ACK



When the local address is received, ACK is automatically generated, regardless of the value of the ACKE0 bit. When an address other than the local address is received, ACK is not generated (NACK).

When an extension code is received, ACK is generated if the ACKE0 bit is set to 1 in advance.

How ACK is generated in data reception differs with the setting of the clock stretch timing as follows.

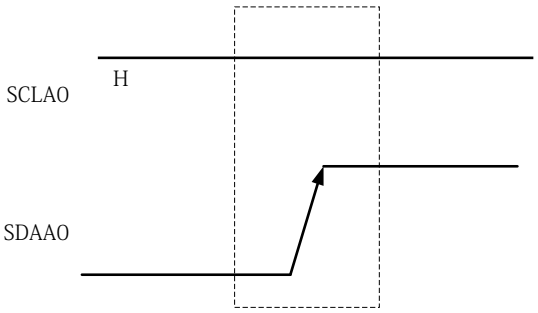
- When the falling edge of the 8th clock is set (bit 3 (WTIM0) of the IICCTL00 register = 0):
  By setting the ACKE0 bit to 1 before releasing clock stretching, ACK is generated at the falling edge of the 8th clock of the SCLA0 pin.
- When the falling edge of the 9th clock is set (bit 3 (WTIM0) of the IICCTL00 register = 1):
  ACK is generated by setting the ACKE0 bit to 1 in advance.

14.5.5 Stop condition

A stop condition is generated when the SDAA0 pin changes from the low to the high level while the SCLA0 pin is at the high level.

A stop condition is a signal that the master generates to the slave when serial transfer has been completed. A stop condition can be detected when the device is used as a slave.

Figure 14-18. Stop Condition



A stop condition is generated when bit 0 (SPT0) of IICA control register 00 (IICCTL00) is set to 1. When the stop condition is detected, bit 0 (SPD0) of IICA status register 0 (IICS0) is set to 1 and INTIICA0 is generated when bit 4 (SPIE0) of the IICCTL00 register is set to 1.

14.5.6 Clock stretching

Clock stretching is used to notify the other party in communications that a master or slave is preparing to transmit or receive data (i.e., in the clock stretch state).

By setting the SCLA0 pin to the low level, the other party is notified of the clock stretch state. When both the master and slave are released from the clock stretch state, the next data transfer can be started.

Figure 14-19. Clock Stretching (1/2)

(1) When clock stretching occurs at the falling edge of the 9th clock for the master and at the falling edge of the 8th clock for the slave (master: transmission, slave: reception, and ACKE0 = 1)

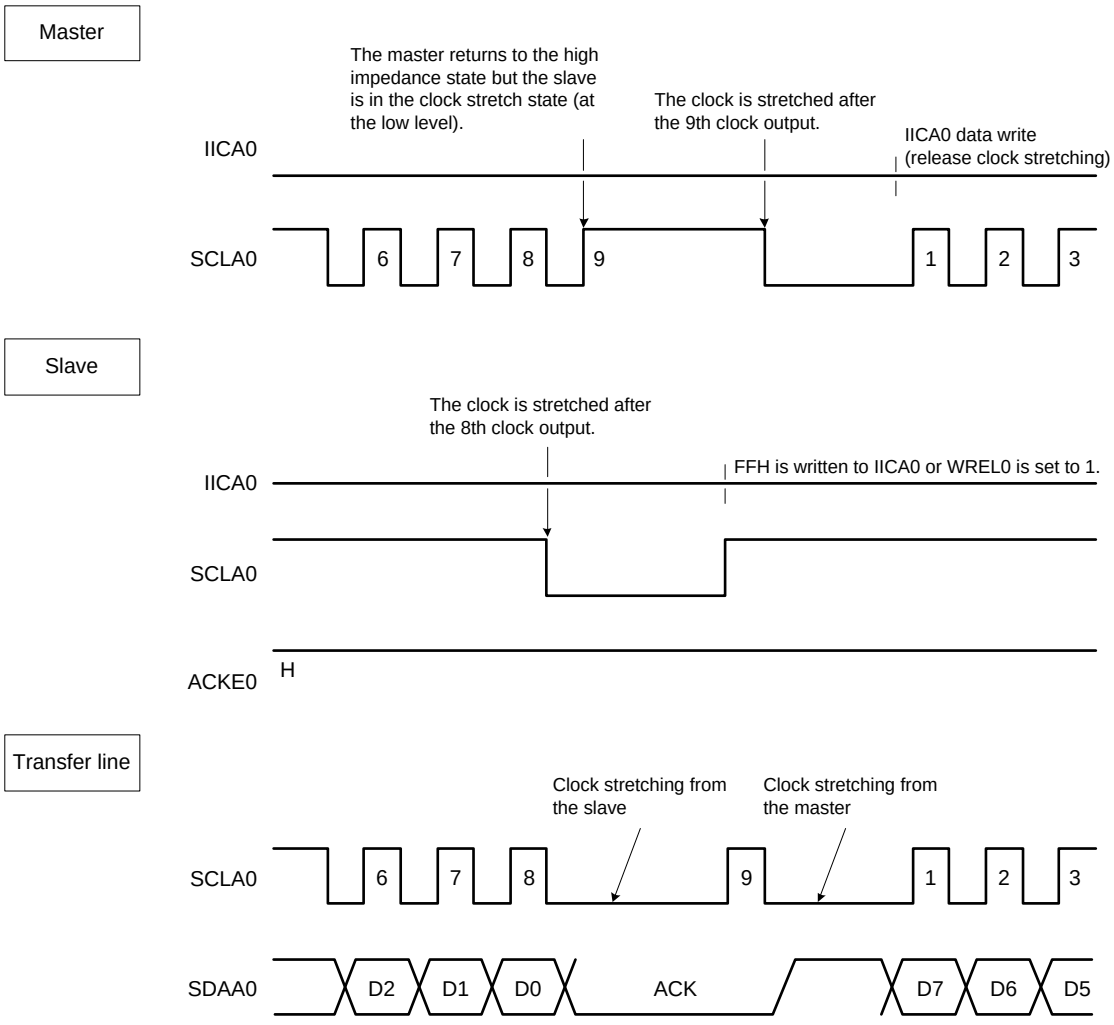
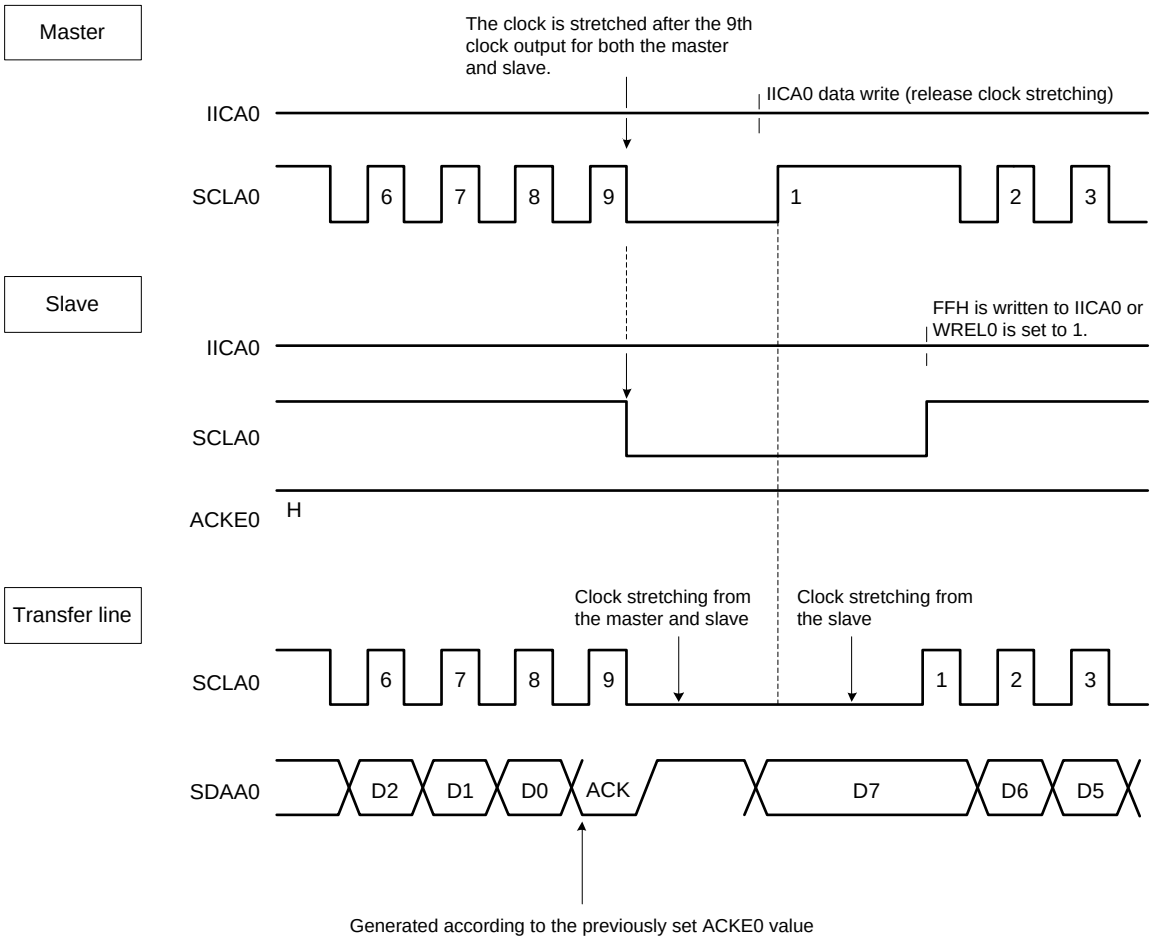


Figure 14-19. Clock Stretching (2/2)

(2) When clock stretching occurs at the falling edge of the 9th clock for both master and slave

(master: transmission, slave: reception, and ACKE0 = 1)



Remark ACKE0: Bit 2 of IICA control register 00 (IICCTL00)
WREL0: Bit 5 of IICA control register 00 (IICCTL00)

Clock stretching is automatically generated depending on the setting of bit 3 (WTIM0) of IICA control register 00 (IICCTL00).

Normally, the reception side releases clock stretching when bit 5 (WREL0) of the IICCTL00 register is set to 1 or FFH is written to IICA shift register 0 (IICA0), and the transmission side releases clock stretching when data is written to the IICA0 register.

The master can also release clock stretching through either of the following methods.

- By setting bit 1 (STT0) of the IICCTL00 register to 1
- By setting bit 0 (SPT0) of the IICCTL00 register to 1

14.5.7 Releasing clock stretching

The I^2C interface usually releases clock stretching by the following processing.

- Writing data to IICA shift register 0 (IICA0)
- Setting bit 5 (WREL0) of IICA control register 00 (IICCTL00) (releasing clock stretching)
- Setting bit 1 (STT0) of the IICCTL00 register (generating a start condition)Note 1
- Setting bit 0 (SPT0) of the IICCTL00 register (generating a stop condition)Note 1

Note 1. For the master in I^2C communications only

When the above processing for releasing clock stretching is executed, I^2C releases clock stretching and communications are resumed.

To release clock stretching and transmit data (including the address), write the data to the IICA0 register.

To receive data after clock stretching has been released, or to complete data transmission, set bit 5 (WREL0) of the IICCTL00 register to 1.

To generate a restart condition after clock stretching has been released, set bit 1 (STT0) of the IICCTL00 register to 1.

To generate a stop condition after clock stretching has been released, set bit 0 (SPT0) of the IICCTL00 register to 1.

Execute the release processing only once for each period of the clock stretch state.

If, for example, data is written to the IICA0 register after clock stretching has been released by setting the WREL0 bit to 1, an incorrect value may be output to the SDAA0 line because the timing for changing the SDAA0 line conflicts with the timing for writing to the IICA0 register.

In addition to the above, when communications are aborted, clock stretching can be released because clearing the IICE0 bit to 0 stops communications.

If the I^2C bus has deadlocked due to noise, clock stretching can be released because the device exits from communications when bit 6 (LREL0) of the IICCTL00 register is set to 1.

Caution If the processing for releasing clock stretching is executed while WUP0 = 1, clock stretching will not be released.

14.5.8 Interrupt request (INTIICA0) generation timing and clock stretching control

The setting of bit 3 (WTIM0) of IICA control register 00 (IICCTL00) generates INTIICA0 and controls clock stretching at the timing shown in **Table 14-2**.

Table 14-2. INTIICA0 Generation Timing and Clock Stretching Control

WTIM0	In Slave Operation			In Master Operation		
	Address	Data Reception	Data Transmission	Address	Data Reception	Data Transmission
0	9Note 1, Note 2	8Note 2	8Note 2	9	8	8
1	9Note 1, Note 2	9Note 2	9Note 2	9	9	9

Note 1. The slave's INTIICA0 signal and clock stretching occur at the falling edge of the 9th clock only on matches with the address set in slave address register 0 (SVA0).
At this point, ACK is generated regardless of the value set in bit 2 (ACKE0) of the IICCTL00 register. The slave that has received an extension code generates INTIICA0 at the falling edge of the 8th clock.
However, if an address mismatch occurs after restart, INTIICA0 is generated at the falling edge of the 9th clock, but clock stretching does not occur.

Note 2. If the received address does not match the setting of slave address register 0 (SVA0) and the extension code is not received, neither INTIICA0 nor clock stretching occurs.

Remark The numbers in the table indicate the numbers of clock cycles of the serial clock. Interrupt requests and control of clock stretching are both synchronized with the falling edge of the serial clock.

(1) In address transmission/reception

- Slave operation: The timing of the interrupt and clock stretching is determined by the conditions described in Note 1 and Note 2 above, regardless of the setting of the WTIM0 bit.
- Master operation: The interrupt and clock stretching occur at the falling edge of the 9th clock, regardless of the setting of the WTIM0 bit.

(2) In data reception

- Master/slave operation: The timing of the interrupt and clock stretching is determined by the setting of the WTIM0 bit.

(3) In data transmission

- Master/slave operation: The timing of the interrupt and clock stretching is determined by the setting of the WTIM0 bit.

(4) Releasing clock stretching

There are four methods for releasing clock stretching as follows.

- Writing data to the IICA shift register 0 (IICA0)
- Setting bit 5 (WREL0) of IICA control register 00 (IICCTL00) (releasing clock stretching)
- Setting bit 1 (STT0) of the IICCTL00 register (generating a start condition)Note 1
- Setting bit 0 (SPT0) of the IICCTL00 register (generating a stop condition)Note 1

Note 1. Master only

When the clock stretch timing has been set to the falling edge of the 8th clock cycle (WTIM0 = 0), whether or not ACK is to be generated must be determined before releasing clock stretching.

(5) Stop condition detection

INTIICA0 is generated when a stop condition is detected (only when SPIE0 = 1).

14.5.9 Address match detection method

In I^2C bus mode, the master can select a particular slave device by transmitting the corresponding slave address.

An address match can be detected automatically by the hardware. An INTIICA0 interrupt request is only generated when the address set in slave address register 0 (SVA0) matches the slave address sent by the master or when an extension code is received.

14.5.10 Error detection

In I^2C bus mode, the state of the serial data bus (SDAA0) during transmission is captured in IICA shift register 0 (IICA0) of the transmitting device, so a transmission error can be detected by comparing the IICA data before the start of transmission and after the end of transmission. A transmission error is judged to having occurred when the two data values do not match.

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 (EXC0) is set to 1 for extension code reception and an interrupt request (INTIICA0) is generated at the falling edge of the 8th clock.
The local address stored in slave address register 0 (SVA0) is not affected.

(2) If 11110xx0 is transferred from the master in 10-bit address transfer while the SVA0 register is set to 11110xx0, the settings are as follows. Note that an interrupt request (INTIICA0) occurs at the falling edge of the 8th clock.

- Higher four bits of data match: EXC0 = 1
- Seven bits of data match: COI0 = 1

Remark EXC0: Bit 5 of IICA status register 0 (IICS0)
COI0: Bit 4 of IICA status register 0 (IICS0)

(3) Since the processing after the interrupt request occurs differs according to the data that follows the extension code, such processing is performed by software.
If the extension code is received in slave operation, the slave is participating in communications even with an address mismatch.
For example, after the extension code is received, if you do not wish to operate the target device as a slave, set bit 6 (LREL0) of IICA control register 00 (IICCTL00) 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 (for address authentication)
1111_0xx	1	10-bit slave address specification (when read command is issued after address match)

Remark See the I^2C bus specifications issued by NXP Semiconductors for details of extension codes other than those described above.

14.5.12 Arbitration

When several masters simultaneously generate a start condition (when the STT0 bit is set to 1 before the STD0 bit is set to 1), master communication is performed while adjusting the clock cycles until the data differs. This kind of operation is called arbitration.

A master which lost in arbitration sets the arbitration loss flag (ALD0) in the IICA status register 0 (IICS0) to 1 at the timing it lost in arbitration and sets both the SCLA0 and SDAA0 lines to high impedance to release the bus.

A loss in arbitration is detected by checking ALD0 = 1 by software at the timing of the next interrupt request (the 8th or 9th clock cycle, when a stop condition is detected, etc.).

For details of interrupt request timing, see **14.5.8 Interrupt request (INTIICA0) generation timing and clock stretching control**.

Remark STD0: Bit 1 of IICA status register 0 (IICS0)
STT0: Bit 1 of IICA control register 00 (IICCTL00)

Figure 14-20. Arbitration Timing Example

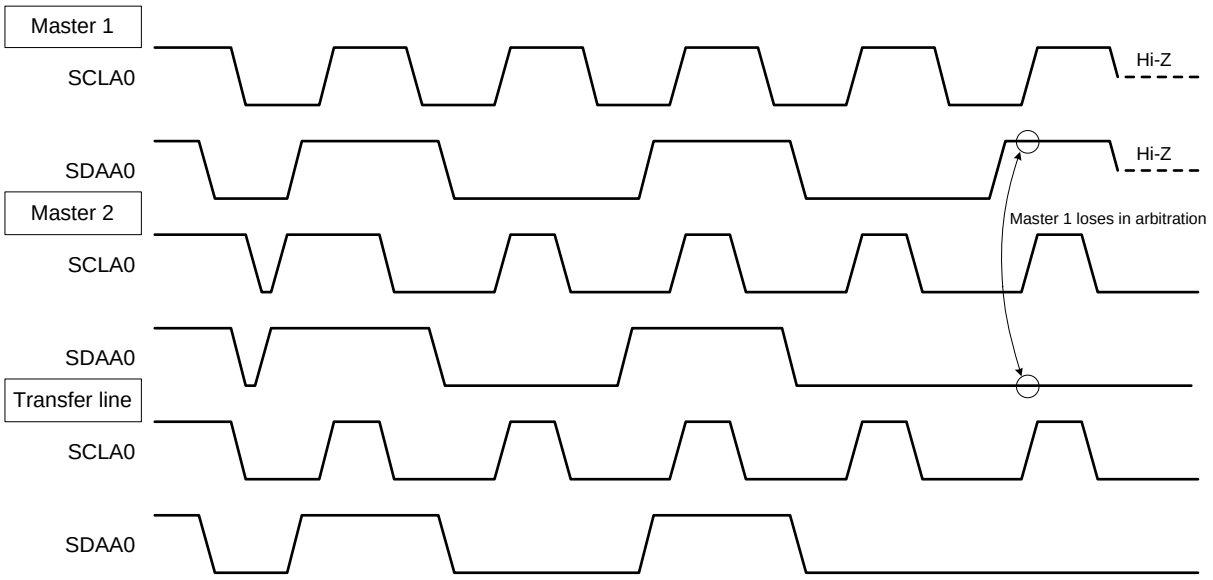


Table 14-4. State when Arbitration Occurred and Interrupt Request Generation Timing

State when Arbitration Occurred	Interrupt Request Generation Timing
During address transmission	Falling edge of 8th or 9th 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	
Restart condition is detected during data transfer	
Stop condition is detected during data transfer	When stop condition is generated (when SPIE0 = 1)[Note 2]
Data is at low level while attempting to generate a restart condition	Falling edge of 8th or 9th clock following byte transfer[Note 1]
Stop condition is detected while attempting to generate restart condition	When stop condition is generated (when SPIE0 = 1)[Note 2]
Data is at low level while attempting to generate a stop condition	Falling edge of 8th or 9th clock following byte transfer[Note 1]
When SCLA0 is at low level while attempting to generate a restart condition	

Note 1. When the WTIM0 bit (bit 3 of IICA control register 00 (IICCTL00)) = 1, an interrupt request occurs at the falling edge of the 9th clock. When WTIM0 = 0, the extension code's slave address is received, an interrupt request occurs at the falling edge of the 8th clock.

Note 2. If there is a possibility that arbitration will occur, set SPIE0 = 1 for master operation.

Remark SPIE0: Bit 4 of IICA control register 00 (IICCTL00)

14.5.13 Wakeup function

This is a slave function of I^2C to generate an interrupt request signal (INTIICA0) when the local address and an extension code are received.

When the addresses do not match, an unnecessary INTIICA0 signal is not generated to allow efficient processing.

When a start condition is detected, the wakeup standby state is entered. Even a master that has generated a start condition enters the wakeup standby state while transmitting an address because the master may become a slave due to an arbitration loss.

To use the wakeup function while in the STOP mode, set the WUP0 bit to 1. The address can be received regardless of the operation clock. An interrupt request signal (INTIICA0) is also generated when the local address and an extension code are received. Operation returns to normal operation by using an instruction to clear the WUP0 bit to 0 after this interrupt has been generated.

Figure 14-21 shows the flow when setting WUP0 = 1 and **Figure 14-22** shows the flow when setting WUP0 = 0 upon an address match.

Figure 14-21. Flow when Setting WUP0 = 1

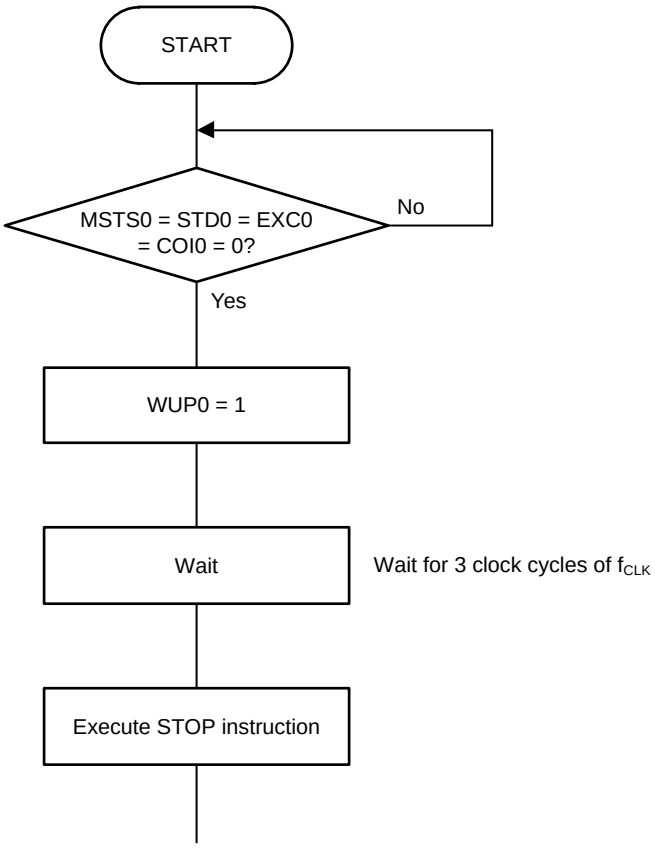
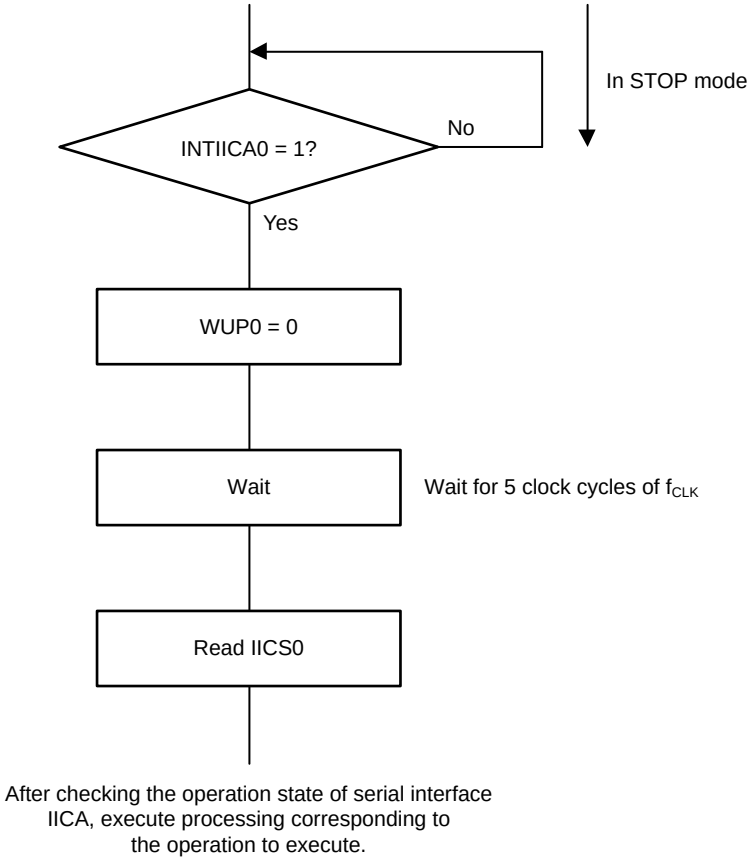


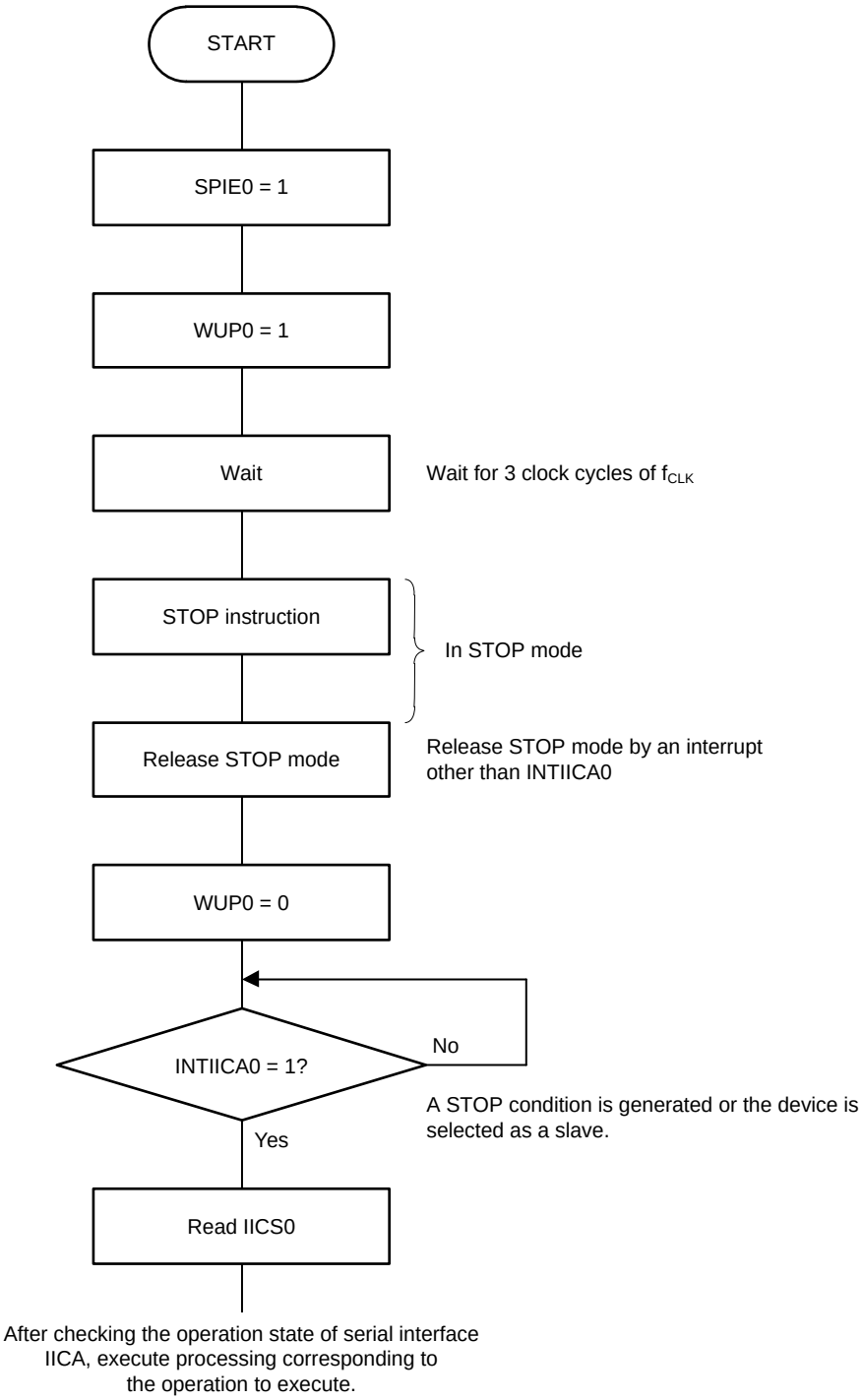
Figure 14-22. Flow when Setting WUP0 = 0 upon Address Match (Including Extension Code Reception)



Follow the flow below to perform the processing to release the STOP mode other than by an interrupt request signal (INTIICA0) from serial interface IICA.

- When operating next I^2C communication as a master:
  - Flow shown in **Figure 14-23**
- When operating next I^2C communication as a slave:
  - When recovered by the INTIICA0 interrupt: Same as the flow in **Figure 14-22**.
  - When recovered by an interrupt other than the INTIICA0 interrupt:
    Continue operation with WUP0 set to 1 until the INTIICA0 interrupt is generated.

Figure 14-23. When Operating as Master after Release from STOP Mode by Interrupt Other than INTIICA0



14.5.14 Communication reservation

(1) When communication reservation is enabled (bit 0 (IICRSV0) of IICA flag register 0 (IICF0) = 0)

To proceed with master communications next while not currently participating in the bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes for not participating in the bus.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled (ACK was not returned and the bus was released by setting bit 6 (LREL0) of IICA control register 00 (IICCTL00) to 1 and exiting from communication)

If bit 1 (STT0) of the IICCTL00 register is set to 1 while not participating in the bus, a start condition is automatically generated and the wait state is entered after the bus has been released (when a stop condition is detected).

If the address is written to IICA shift register 0 (IICA0) after bit 4 (SPIE0) of the IICCTL00 register is set to 1, and release of the bus is detected (detection of the stop condition) by generation of an interrupt request signal (INTIICA0), the device automatically starts communication as the master. Data written to the IICA0 register before detecting the stop condition is invalid.

When the STT0 bit has been set to 1, the operation mode (operation as start condition or as communication reservation) is determined according to the state of the bus.

- If the bus has been released ····· start condition generation
- If the bus has not been released (standby mode) ··· communication reservation

Check whether the communication reservation operates or not by using the MSTS0 bit (bit 7 of IICA status register 0 (IICS0)) after the STT0 bit is set to 1 and the wait time elapses.

Use software to secure the wait time calculated by the following expression.

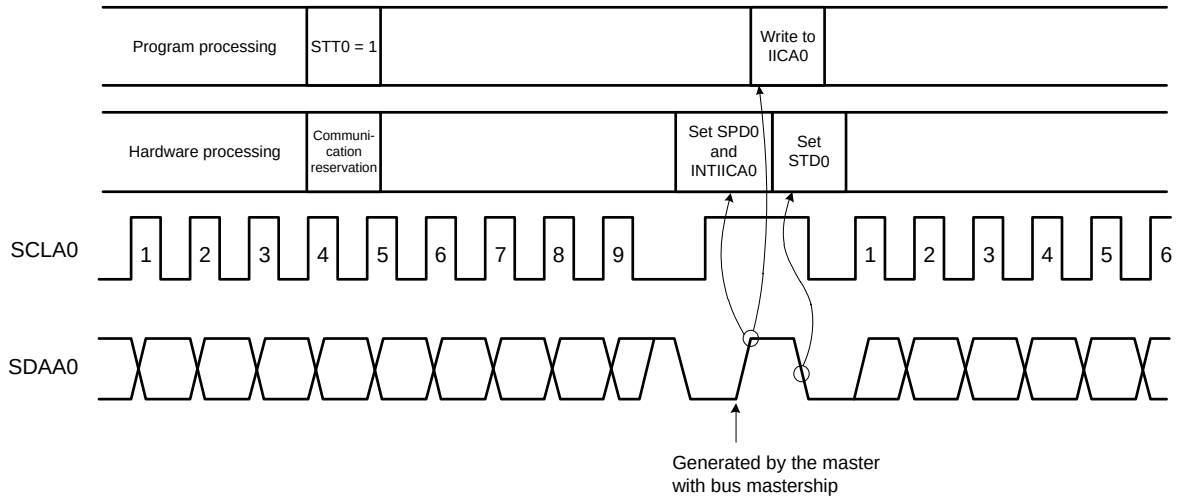
Wait time from setting STT0 = 1 to checking the MSTS0 flag:

$$(\text{IICWL0 setting value} + \text{IICWH0 setting value} + 4)/f_{CLK} + t_F \times 2$$

Remark IICWL0: IICA low-level width setting register 0
IICWH0: IICA high-level width setting register 0
t_F: SDAA0 and SCLA0 signal falling time
f_{CLK}: CPU/peripheral hardware clock frequency

Figure 14-24 shows the communication reservation timing.

Figure 14-24. Communication Reservation Timing



Remark IICA0: IICA shift register 0
STT0: Bit 1 of IICA control register 00 (IICCTL00)
STD0: Bit 1 of IICA status register 0 (IICS0)
SPD0: Bit 0 of IICA status register 0 (IICS0)

Communication reservations are accepted at the timing shown in **Figure 14-25**. After bit 1 (STD0) of IICA status register 0 (IICS0) is set to 1, a communication reservation can be made by setting bit 1 (STT0) of IICA control register 00 (IICCTL00) to 1 before a stop condition is detected.

Figure 14-25. Timing for Accepting Communication Reservations

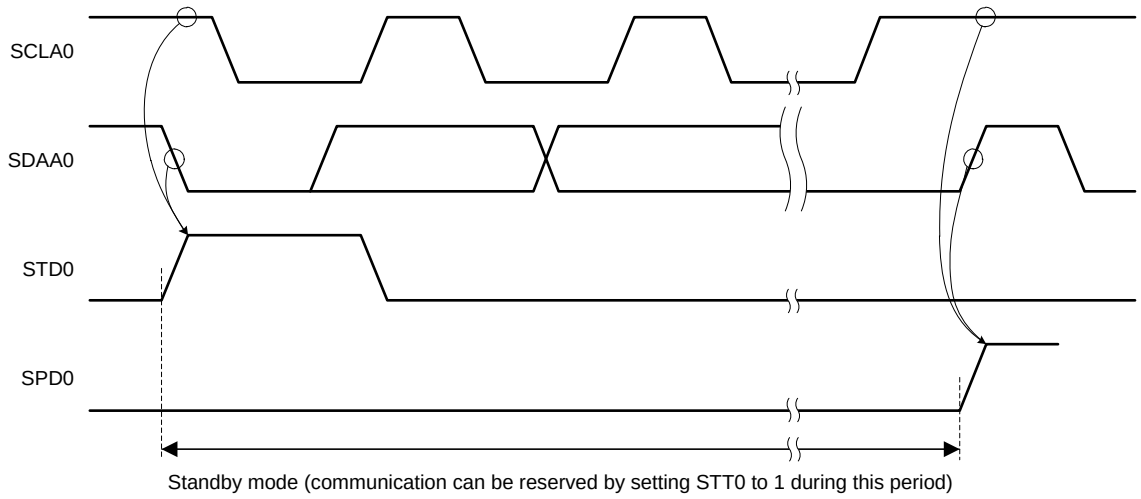
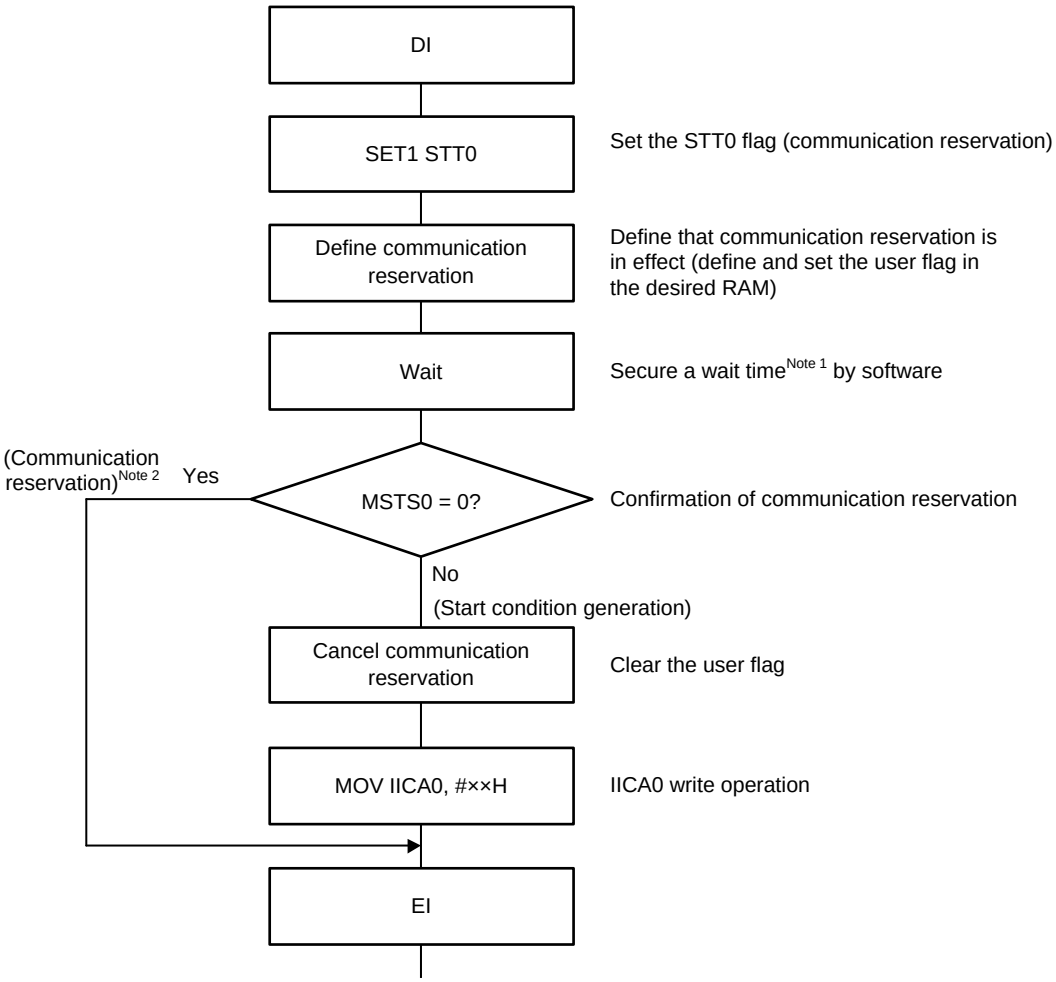


Figure 14-26 shows the procedure for communication reservation.

Figure 14-26. Communication Reservation Procedure



Note 1. A wait time is as follows.
(IICWL0 setting value + IICWH0 setting value + 4)/f_{CLK} + t_F × 2

Note 2. In communication reservation operation, writing to IICA shift register 0 (IICA0) is executed in response to a stop condition interrupt request.

Remark STT0: Bit 1 of IICA control register 00 (IICCTL00)
MSTS0: Bit 7 of IICA status register 0 (IICS0)
IICA0: IICA shift register 0
IICWL0: IICA low-level width setting register 0
IICWH0: IICA high-level width setting register 0
t_F: SDAA0 and SCLA0 signal falling time
f_{CLK}: CPU/peripheral hardware clock frequency

(2) When communication reservation is disabled (bit 0 (IICRSV0) of IICA flag register 0 (IICF0) = 1)

When bit 1 (STT0) of IICA control register 00 (IICCTL00) is set to 1 while the bus is not participating in this communication during communication, this request is rejected and a start condition is not generated. Non-participation of the bus in this case includes the following two states.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled (ACK was not returned and the bus was released by setting bit 6 (LREL0) of the IICCTL00 register to 1 and exiting from communication)

Whether the start condition was generated or rejected can be checked by reading STCF0 (bit 7 of the IICF0 register). Since up to 5 clock cycles are taken until the STCF0 bit is set to 1 after setting STT0 = 1, secure this time by software.

14.5.15 Cautions

(1) When STCEN0 = 0

Immediately after I^2C operation is enabled (IICE0 = 1), the bus communication state (IICBSY0 = 1) is recognized regardless of the actual bus state. When performing master communication from the state where a stop condition is not detected, first generate a stop condition to release the bus, then perform master communication.

When using multiple masters, it is not possible to perform master communication while the bus is not released (a stop condition is not detected).

Follow the following sequence to generate a stop condition.

<1> Set IICA control register 01 (IICCTL01).

<2> Set bit 7 (IICE0) of IICA control register 00 (IICCTL00) to 1.

<3> Set bit 0 (SPT0) of the IICCTL00 register to 1.

(2) When STCEN0 = 1

Immediately after I^2C operation is enabled (IICE0 = 1), the bus released state (IICBSY0 = 0) is recognized regardless of the actual bus state. To generate the first start condition (STT0 = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

(3) If I^2C communication with the other party is already in progress

If I^2C operation is enabled and the device participates in communication already in progress when the SDAA0 pin is at the low level and the SCLA0 pin is at the high level, the macro of I^2C recognizes that the SDAA0 pin has changed from the high to the low level (start condition detection). If the value on the bus at this time can be recognized as an extension code, ACK is returned, but this interferes with I^2C communications with the other party. To avoid this, start the I^2C in the following sequence.

<1> Clear bit 4 (SPIE0) of the IICCTL00 register to 0 to disable generation of an interrupt request signal (INTIICA0) in response to the detection of a stop condition.

<2> Set bit 7 (IICE0) of the IICCTL00 register to 1 to enable operation of the I^2C.

<3> Wait for detection of a start condition.

<4> Set bit 6 (LREL0) of the IICCTL00 register to 1 before ACK is returned (in 4 to 72 clock cycles after setting the IICE0 bit to 1) to forcibly disable detection.

(4) After setting the STT0 and SPT0 bits (bits 1 and 0 of the IICCTL00 register), re-setting these bits before they are cleared to 0 is prohibited.

(5) When transmission is reserved, set the SPIE0 bit (bit 4 of the IICCTL00 register) to 1 so that an interrupt request is generated when a stop condition is detected. Transfer is started when communication data is written to IICA shift register 0 (IICA0) after the interrupt request has been generated. If the interrupt is not generated in response to the detection of a stop condition, the device stops in the wait state because the interrupt request is not generated when communication is started. However, it is not necessary to set the SPIE0 bit to 1 when detecting the MSTS0 bit (bit 7 of IICA status register 0 (IICS0)) by software.

14.5.16 Communication operations

The following describes three operation procedures as flows.

1) Master operation in single master system

The flow when the device is used as a master in the single master system is described.

This flow is broadly divided into the initial settings and communication processing. Execute the initial settings at startup. If communication with the slave is required, make necessary preparation for communication and then execute communication processing.

2) Master operation in multi-master system

In the I^2C bus multi-master system, whether the bus is released or used cannot be judged only by the I^2C bus specifications at the stage when a device participates in communications. Here, when the data and clock are at the high level for a certain period (1 frame), the device participates in communications in the bus released state.

This flow is broadly divided into the initial settings, communication waiting, and communication processing. The processing when the device loses in arbitration and is specified as a slave is omitted here, and only the processing as a master is shown. Execute the initial settings at startup to participate in communications. Then, wait for a communication request as a master or wait for specification as a slave. The actual communication is performed in the communication processing, and it supports transmission/reception with the slave and the arbitration with other masters.

3) Slave operation

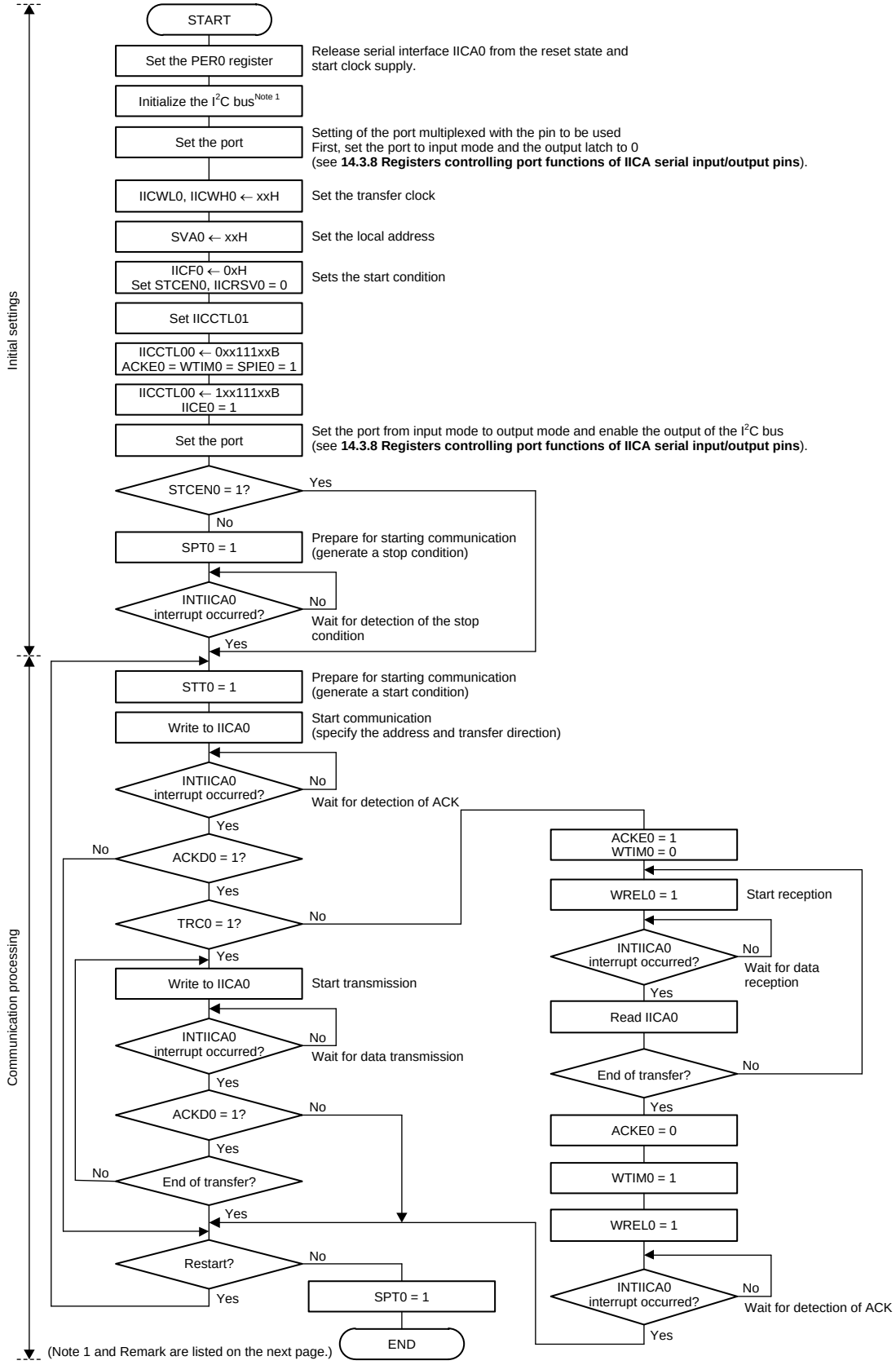
An example when the device is used as a slave of the I^2C bus is explained below.

When used as a slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIICA0 interrupt occurrence (communication waiting). When an INTIICA0 interrupt occurs, the communication state is judged and the result is passed to the main processing as a flag.

By checking each flag, the required communication processing is performed.

(1) Master operation in single master system

Figure 14-27. Master Operation in Single Master System

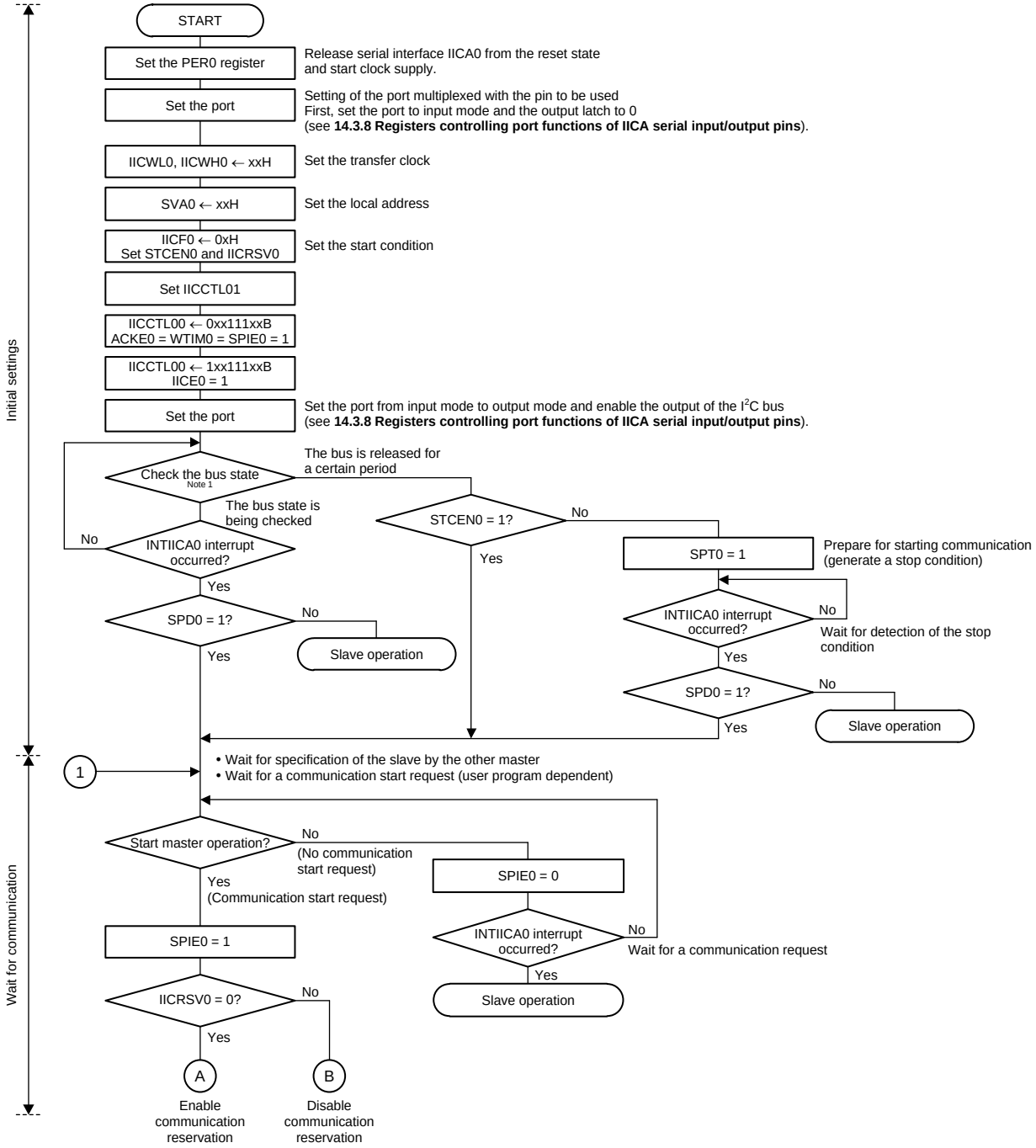


Note 1. Release the I^2C bus (SCLA0 and SDAA0 pins = high level) in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDAA0 pin, set the SCLA0 pin to the output port, and output a clock pulse from the output port until the SDAA0 pin is constantly at the high level.

Remark Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

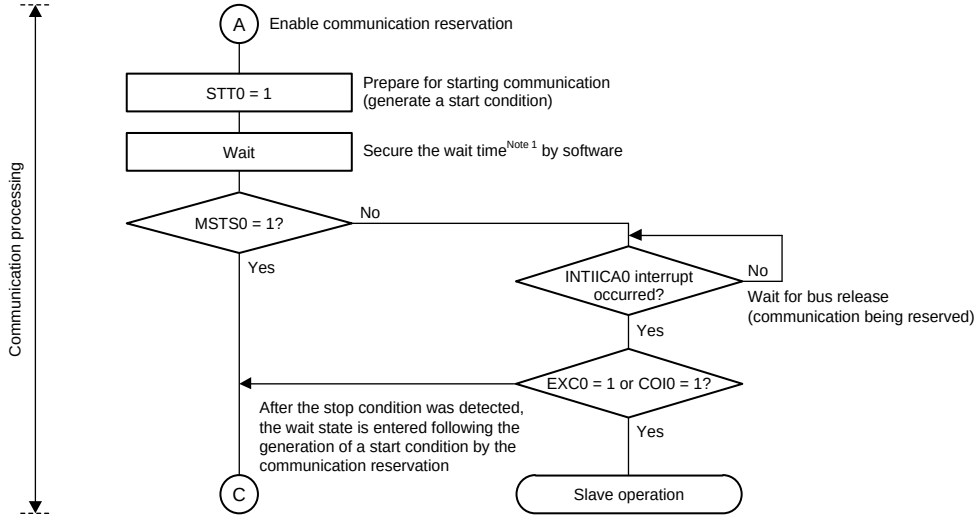
(2) Master operation in multi-master system

Figure 14-28. Master operation in Multi-master System (1/3)

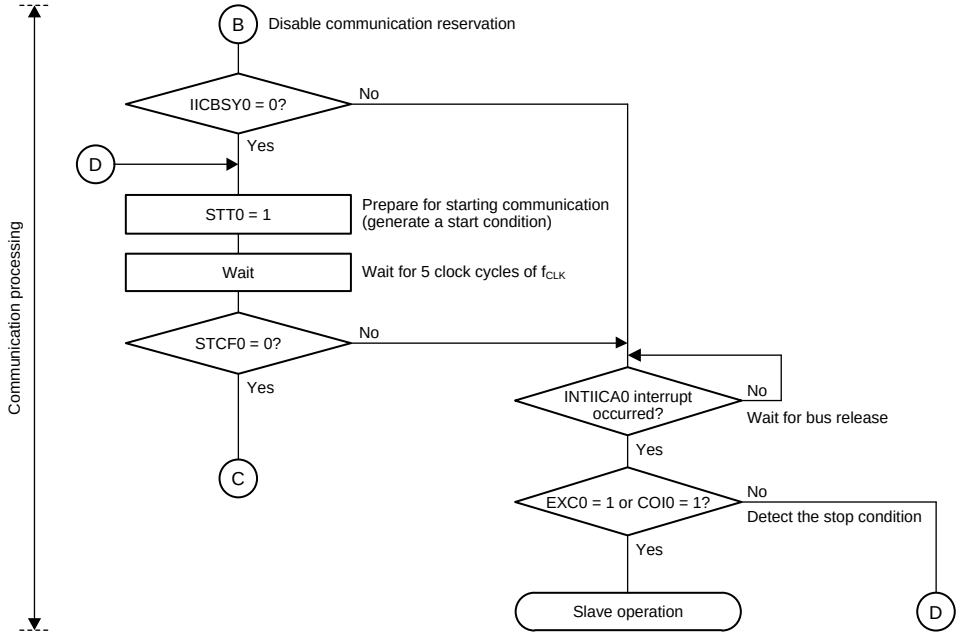


Note 1. Confirm that the bus is released (CLD0 bit = 1, DAD0 bit = 1) for a certain period (for example, for a period of one frame). If the SDAA0 pin is constantly at the low level, decide whether to release the I^2C bus (SCLA0 and SDAA0 pins = high level) in conformance with the specifications of the product that is communicating.

Figure 14-28. Master Operation in Multi-master System (2/3)

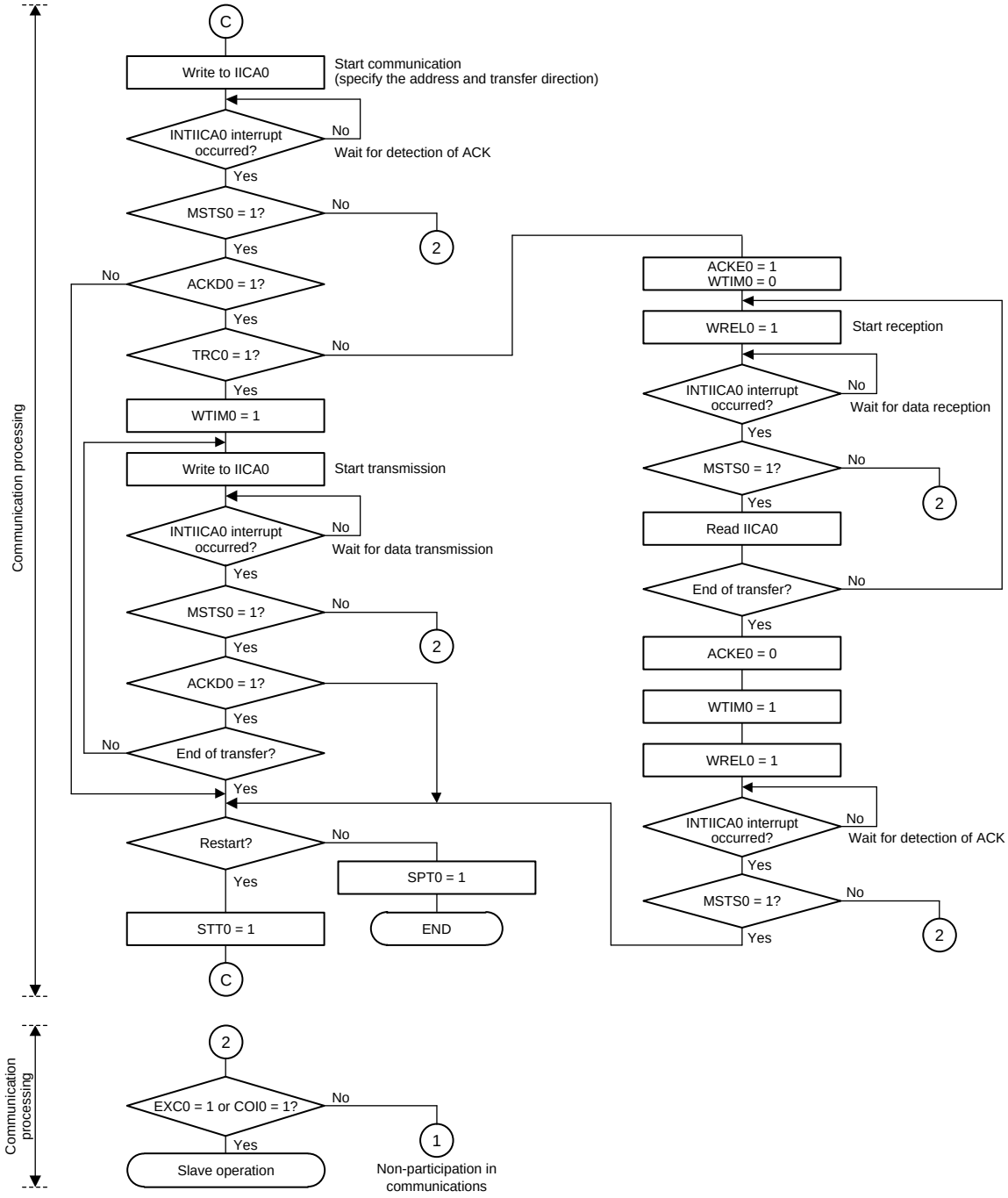


Note 1. A wait time is as follows.
(IICWL0 setting value + IICWH0 setting value + 4)/f_{CLK} + f_F × 2



Remark IICWL0: IICA low-level width setting register 0
IICWH0: IICA high-level width setting register 0
t_F: SDAA0 and SCLA0 signal falling time
f_{CLK}: CPU/peripheral hardware clock frequency

Figure 14-28. 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 the multi-master system, read the MSTS0 bit to check the arbitration result each time interrupt INTIICA0 is generated.

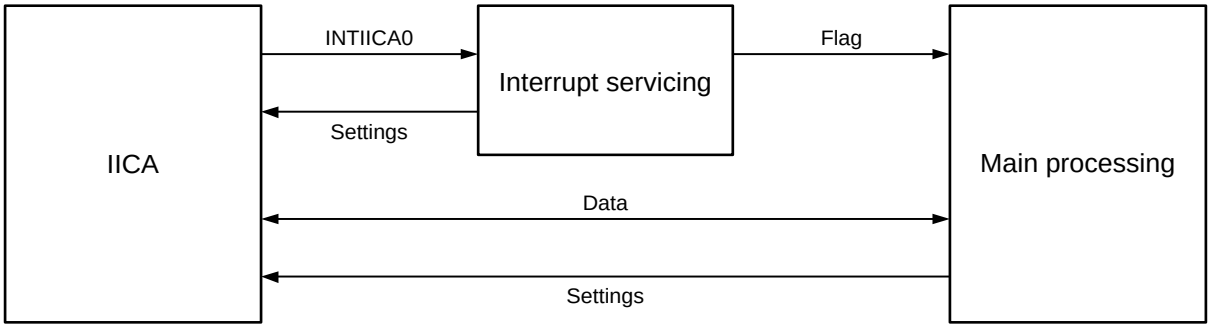
Remark 3. To use the device as a slave in the multi-master system, check the status by reading IICA status register 0 (IICS0) and IICA flag register 0 (IICF0) each time interrupt INTIICA0 is generated, and determine next processing to be performed.

(3) Slave operation

The processing procedure of slave operation is as follows.

Basically, slave operation is event-driven. This requires processing by the INTIICA0 interrupt (processing that requires substantially changing the operation state such as detection of a stop condition during communication).

In the following explanation, it is assumed that the extension code is not supported for data communication. It is also assumed that the INTIICA0 interrupt servicing only performs processing for state transition and actual data communication is performed by the main processing.



Therefore, data communication processing is performed by preparing the following three flags and passing them to the main processing instead of INTIICA0.

<1> Communication mode flag
This flag indicates the following two communication states.

- Clear mode: State in which data communication is not performed
- Communication mode: State in which data communication is performed (from valid address detection to stop condition detection, no detection of ACK from master, address mismatch)

<2> Ready flag
This flag indicates that data communication is enabled. Its function is the same as the INTIICA0 interrupt in ordinary data communications. This flag is set by interrupt servicing and cleared by the main processing. Clear this flag by interrupt servicing when communication is started. However, the ready flag is not set by interrupt servicing when the first data is transmitted. Therefore, the first data is transmitted without the flag being cleared (an address match is interpreted as a request for the next data).

<3> Communication direction flag
This flag indicates the direction of communication. Its value is the same as the TRC0 bit.

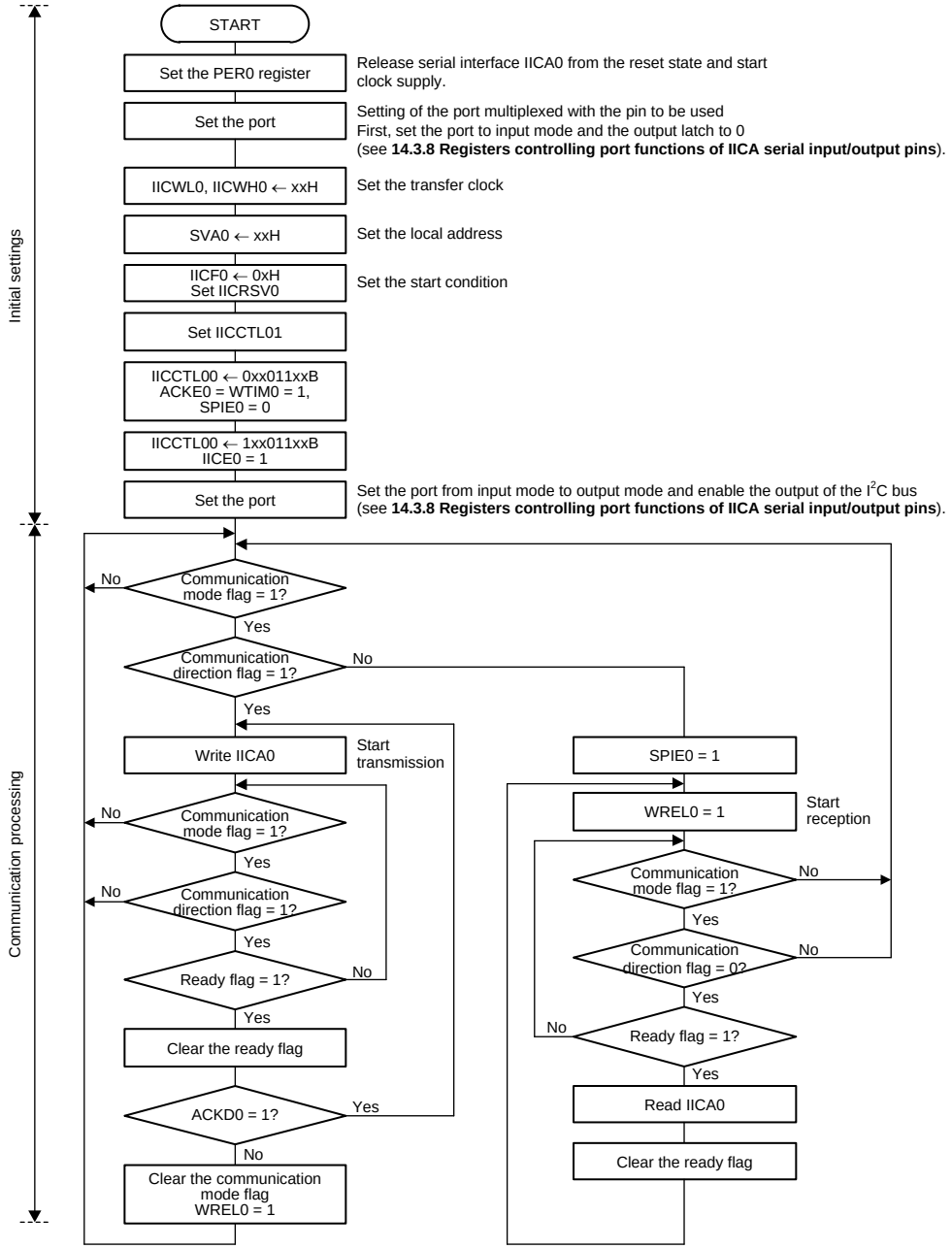
Next, the main processing of slave operation is explained.

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 in response to an interrupt. Here, check the state by using the flags).

The transmission operation is repeated until the master no longer returns ACK. If ACK is not returned from the master, communication is completed.

In reception, the necessary amount of data is received. When communication is completed, ACK is not returned for the next data. After that, the master generates a stop condition or restart condition. The device exits the communication state in this way.

Figure 14-29. Slave Operation Procedure (1)



Remark Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

An example of the processing procedure of the slave with the INTIICA0 interrupt is explained below (processing is performed assuming that no extension code is used). The INTIICA0 interrupt checks the status, and the following operations are performed.

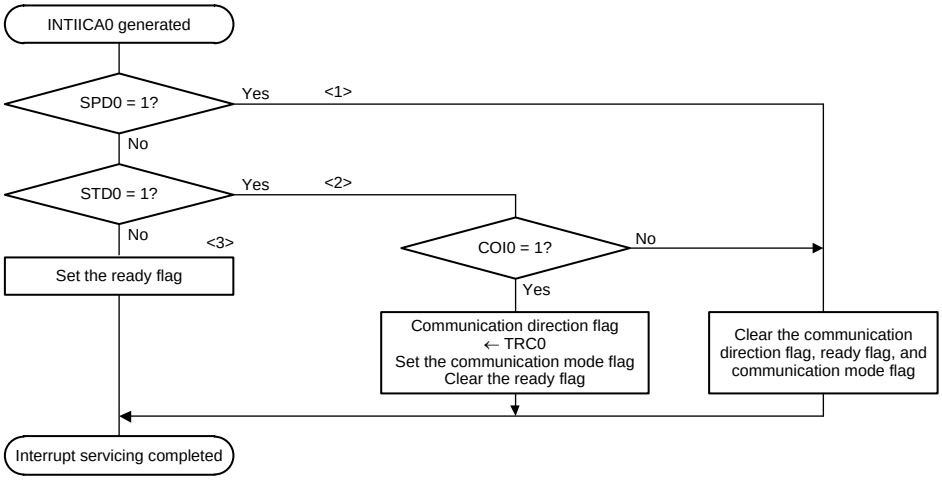
<1> Communication is stopped if the stop condition is issued.

<2> If the start condition is issued, the address is checked and communication is completed if the address is not matched.
If the address is matched, the communication mode is set, the wait state is released, and processing returns from the interrupt (the ready flag is cleared).

<3> For data transmission/reception, only the ready flag is set. Processing returns from the interrupt while the I^2C bus remains in the wait state.

Remark <1> to <3> above correspond to <1> to <3> in **Figure 14-30 Slave Operation Procedure (2)**.

Figure 14-30. Slave Operation Procedure (2)



14.5.17 I²C interrupt request (INTIICA0) generation timing

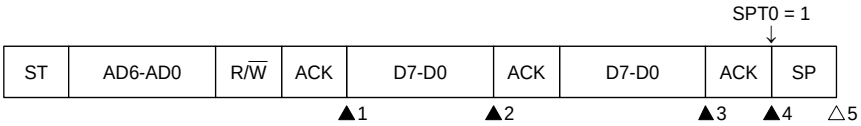
The timing of data transmission/reception and generation of interrupt request signal INTIICA0 and the value of IICA status register 0 (IICS0) with the INTIICA0 signal timing are shown below.

Remark ST: Start condition
AD6 to AD0: Address
R/$\overline{W}$: Transfer direction specification
ACK: Acknowledge
D7 to D0: Data
SP: Stop condition

(1) Master operation

(a) Start ~ Address ~ Data ~ Data ~ Stop (transmission/reception)

(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B

▲2: IICS0 = 1000×000B

▲3: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)[Note 1]

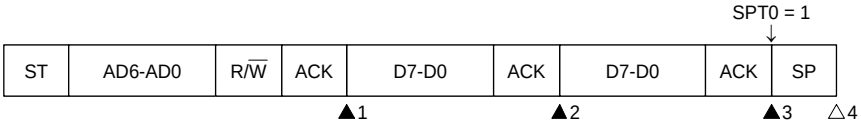
▲4: IICS0 = 1000××00B (Sets the SPT0 bit to 1)

△5: IICS0 = 00000001B

Note 1. To generate a stop condition, set the WTIM0 bit to 1 and change the timing for generating the INTIICA0 interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



▲1: IICS0 = 1000×110B

▲2: IICS0 = 1000×100B

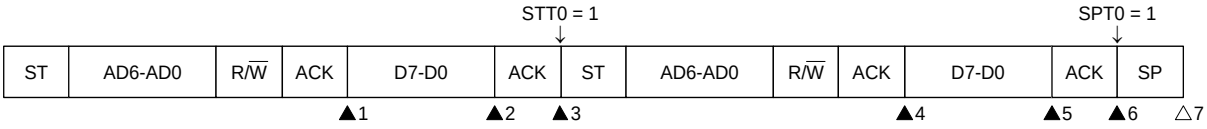
▲3: IICS0 = 1000××00B (Sets the SPT0 bit to 1)

△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B

▲2: IICS0 = 1000×000B (Sets the WTIM0 bit to 1[Note 1])

▲3: IICS0 = 1000××00B (Clears the WTIM0 bit to 0[Note 2], sets the STT0 bit to 1)

▲4: IICS0 = 1000×110B

▲5: IICS0 = 1000×000B (Sets the WTIM0 bit to 1[Note 3])

▲6: IICS0 = 1000××00B (Sets the SPT0 bit to 1)

△7: IICS0 = 00000001B

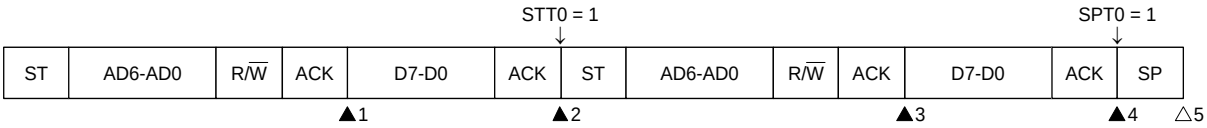
Note 1. To generate a start condition, set the WTIM0 bit to 1 and change the timing for generating the INTIICA0 interrupt request signal.

Note 2. Clear the WTIM0 bit to 0 to restore the original setting.

Note 3. To generate a stop condition, set the WTIM0 bit to 1 and change the timing for generating the INTIICA0 interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(ii) When WTIM0 = 1



▲1: IICS0 = 1000×110B

▲2: IICS0 = 1000××00B (Sets the STT0 bit to 1)

▲3: IICS0 = 1000×110B

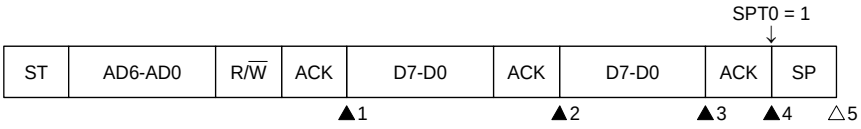
▲4: IICS0 = 1000××00B (Sets the SPT0 bit to 1)

△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

(i) When WTIM0 = 0



▲1: IICS0 = 1010x110B

▲2: IICS0 = 1010x000B

▲3: IICS0 = 1010x000B (Sets the WTIM0 bit to 1[Note 1])

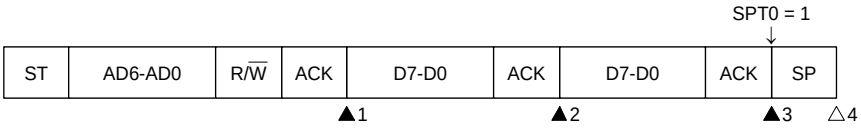
▲4: IICS0 = 1010xx00B (Sets the SPT0 bit to 1)

△5: IICS0 = 00000001B

Note 1. To generate a start condition, set the WTIM0 bit to 1 and change the timing for generating the INTIICA0 interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



▲1: IICS0 = 1010×110B

▲2: IICS0 = 1010×100B

▲3: IICS0 = 1010××00B (Sets the SPT0 bit to 1)

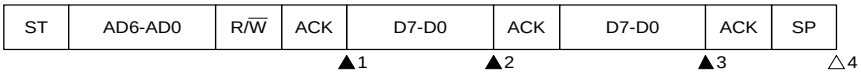
△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(2) Slave operation (slave address data reception)

(a) Start ~ Address ~ Data ~ Data ~ Stop

(i) When WTIM0 = 0



▲1: IICS0 = 0001×110B

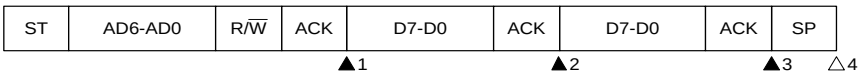
▲2: IICS0 = 0001×000B

▲3: IICS0 = 0001×000B

△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



▲1: IICS0 = 0001×110B

▲2: IICS0 = 0001×100B

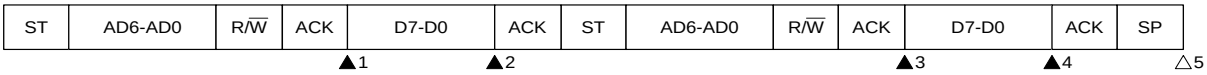
▲3: IICS0 = 0001××00B

△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0 = 0 (a match with SVA0 after restart)



▲1: IICS0 = 0001×110B

▲2: IICS0 = 0001×000B

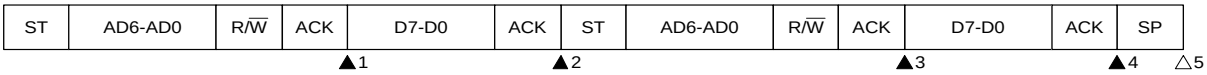
▲3: IICS0 = 0001×110B

▲4: IICS0 = 0001×000B

△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (a match with SVA0 after restart)



▲1: IICS0 = 0001×110B

▲2: IICS0 = 0001××00B

▲3: IICS0 = 0001×110B

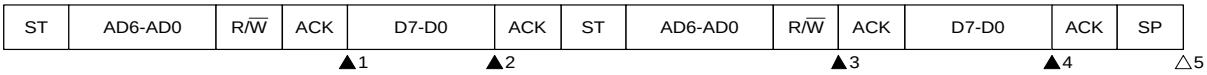
▲4: IICS0 = 0001××00B

△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

(i) When WTIM0 = 0 (an address mismatch after restart (extension code))



▲1: IICS0 = 0001×110B

▲2: IICS0 = 0001×000B

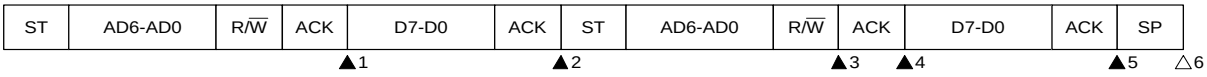
▲3: IICS0 = 0010×010B

▲4: IICS0 = 0010×000B

△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (an address mismatch after restart (extension code))



▲1: IICS0 = 0001×110B

▲2: IICS0 = 0001××00B

▲3: IICS0 = 0010×010B

▲4: IICS0 = 0010×110B

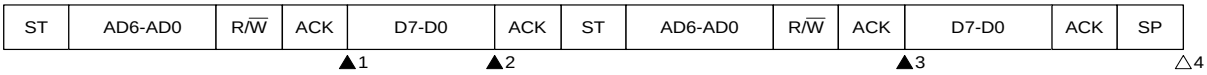
▲5: IICS0 = 0010××00B

△6: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0 = 0 (an address mismatch after restart (other than extension code))



▲1: IICS0 = 0001×110B

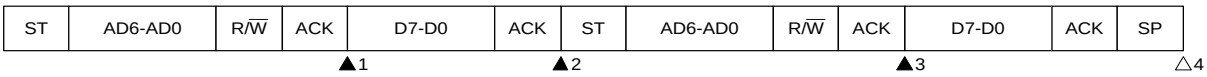
▲2: IICS0 = 0001×000B

▲3: IICS0 = 00000×10B

△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(ii) When WTIM0 = 1 (an address mismatch after restart (other than extension code))



▲1: IICS0 = 0001×110B

▲2: IICS0 = 0001××00B

▲3: IICS0 = 00000×10B

△4: IICS0 = 00000001B

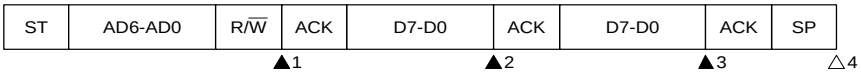
Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(3) Slave operation (when receiving extension code)

The device always participates in communications when it receives an extension code.

(a) Start ~ Code ~ Data ~ Data ~ Stop

(i) When WTIM0 = 0



▲1: IICS0 = 0010×010B

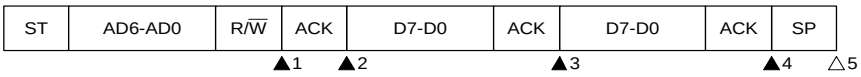
▲2: IICS0 = 0010×000B

▲3: IICS0 = 0010×000B

△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



▲1: IICS0 = 0010×010B

▲2: IICS0 = 0010×110B

▲3: IICS0 = 0010×100B

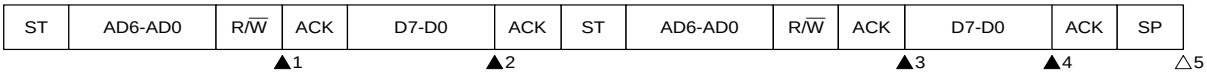
▲4: IICS0 = 0010××00B

△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0 = 0 (a match with SVA0 after restart)



▲1: IICS0 = 0010×010B

▲2: IICS0 = 0010×000B

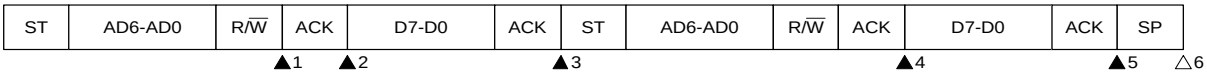
▲3: IICS0 = 0001×110B

▲4: IICS0 = 0001×000B

△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (a match with SVA0 after restart)



▲1: IICS0 = 0010×010B

▲2: IICS0 = 0010×110B

▲3: IICS0 = 0010××00B

▲4: IICS0 = 0001×110B

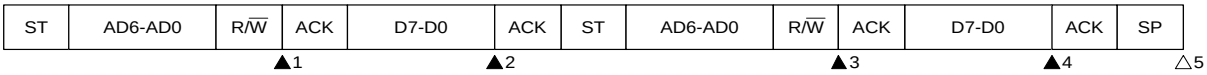
▲5: IICS0 = 0001××00B

△6: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

(i) When WTIM0 = 0 (extension code reception after restart)



▲1: IICS0 = 0010×010B

▲2: IICS0 = 0010×000B

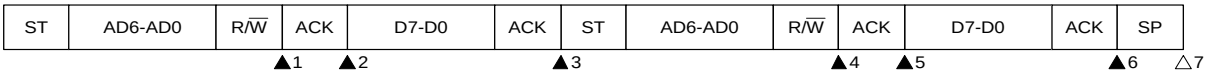
▲3: IICS0 = 0010×010B

▲4: IICS0 = 0010×000B

△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (extension code reception after restart)



▲1: IICS0 = 0010×010B

▲2: IICS0 = 0010×110B

▲3: IICS0 = 0010××00B

▲4: IICS0 = 0010×010B

▲5: IICS0 = 0010×110B

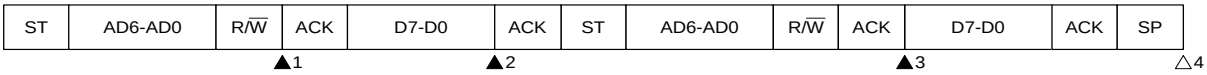
▲6: IICS0 = 0010××00B

△7: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM0 = 0 (an address mismatch after restart (other than extension code))



▲1: IICS0 = 0010×010B

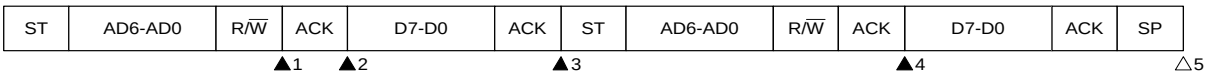
▲2: IICS0 = 0010×000B

▲3: IICS0 = 00000×10B

△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1 (an address mismatch after restart (other than extension code))



▲1: IICS0 = 0010×010B

▲2: IICS0 = 0010×110B

▲3: IICS0 = 0010××00B

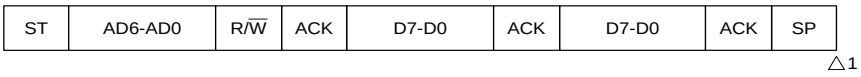
▲4: IICS0 = 00000×10B

△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(4) Operation when not participating in communications

(a) Start ~ Code ~ Data ~ Data ~ Stop



△1: IICS0 = 00000001B

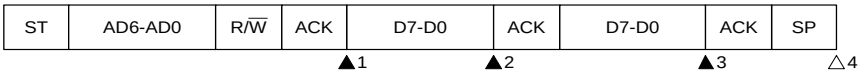
Remark △: Generated only when SPIE0 = 1

(5) Arbitration loss operation (operation as slave after arbitration loss)

To use the device as a master in the multi-master system, read the MSTS0 bit to check the arbitration result each time interrupt request signal INTIICA0 is generated.

(a) When lost in arbitration during transmission of slave address data

(i) When WTIM0 = 0



▲1: IICS0 = 0101×110B

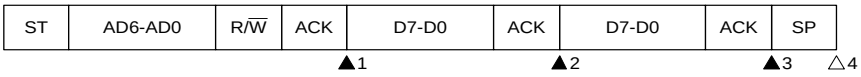
▲2: IICS0 = 0001×000B

▲3: IICS0 = 0001×000B

△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



▲1: IICS0 = 0101×110B

▲2: IICS0 = 0001×100B

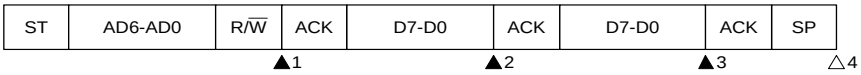
▲3: IICS0 = 0001××00B

△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(b) When lost in arbitration during transmission of extension code

(i) When WTIM0 = 0



▲1: IICS0 = 0110×010B

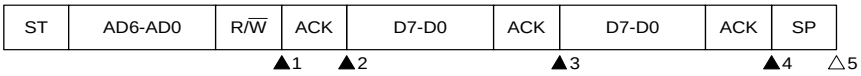
▲2: IICS0 = 0010×000B

▲3: IICS0 = 0010×000B

△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



▲1: IICS0 = 0110×010B

▲2: IICS0 = 0010×110B

▲3: IICS0 = 0010×100B

▲4: IICS0 = 0010××00B

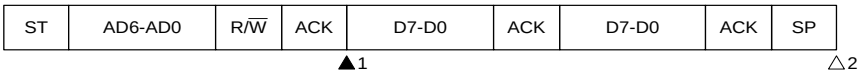
△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(6) Arbitration loss operation (not participating in communications after arbitration loss)

When the device is used as a master in the multi-master system, read the MSTS0 bit to check the arbitration result each time interrupt request signal INTIICA0 is generated.

(a) When lost in arbitration during transmission of slave address data (when WTIM0 = 1)

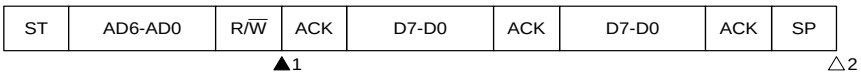


▲1: IICS0 = 01000110B

△2: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(b) When lost in arbitration during transmission of extension code



▲1: IICS0 = 0110×010B

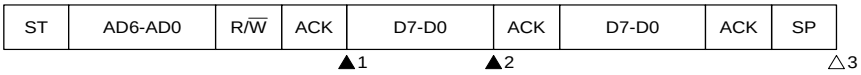
Sets LREL0 = 1 by software

△2: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don’t care

(c) When lost in arbitration during data transfer

(i) When WTIM0 = 0



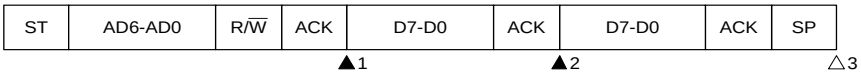
▲1: IICS0 = 10001110B

▲2: IICS0 = 01000000B

△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

When WTIM0 = 1



▲1: IICS0 = 10001110B

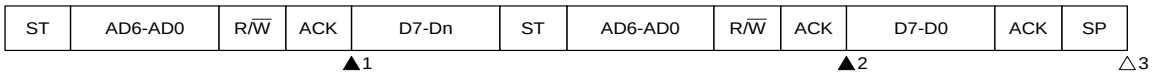
▲2: IICS0 = 01000100B

△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1

(d) When lost in the restart condition during data transfer

(i) Other than extension code (e.g., a mismatch with SVA0)



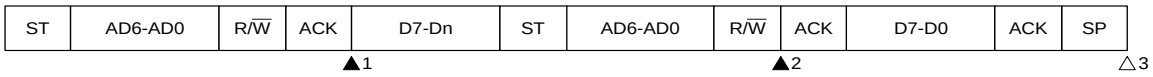
▲1: IICS0 = 1000×110B

▲2: IICS0 = 01000110B

△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care
n = 6 to 0

(ii) Extension code



▲1: IICS0 = 1000×110B

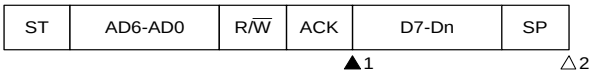
▲2: IICS0 = 01100010B

Sets LREL0 = 1 by software

△3: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care
n = 6 to 0

(e) When lost in the stop condition during data transfer



▲1: IICS0 = 10000110B

△2: IICS0 = 01000001B

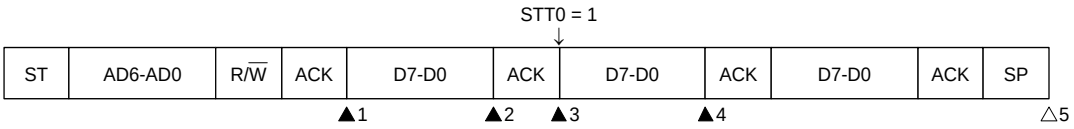
Remark ▲: Always generated

△: Generated only when SPIE0 = 1

n = 6 to 0

(f) When lost in arbitration due to the data being at the low level when attempting to generate a restart condition

(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B

▲2: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)

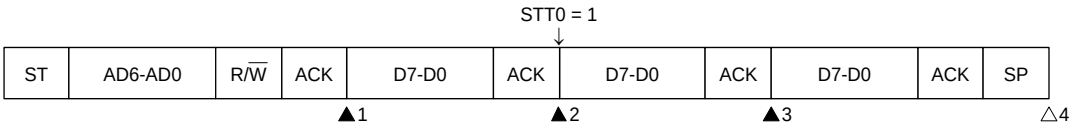
▲3: IICS0 = 1000×100B (Clears the WTIM0 bit to 0)

▲4: IICS0 = 01000000B

△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



▲1: IICS0 = 1000×110B

▲2: IICS0 = 1000×100B (Sets the STT0 bit to 1)

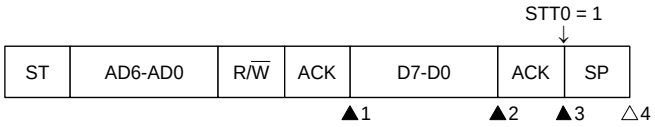
▲3: IICS0 = 01000100B

△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(g) When lost in arbitration at the stop condition when attempting to generate a restart condition

(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B

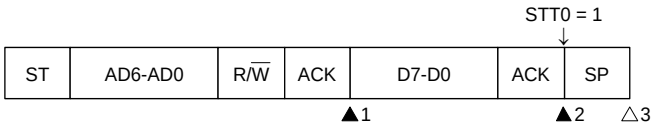
▲2: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)

▲3: IICS0 = 1000××00B (Sets the STT0 bit to 1)

△4: IICS0 = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



▲1: IICS0 = 1000×110B

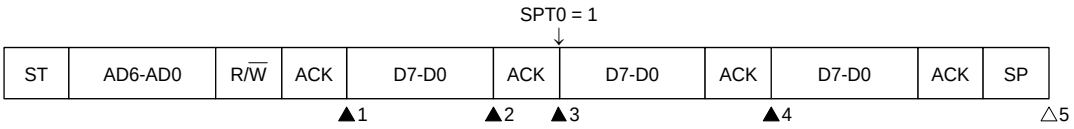
▲2: IICS0 = 1000××00B (Sets the STT0 bit to 1)

△3: IICS0 = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(h) When lost in arbitration due to the data being at the low level when attempting to generate a stop condition

(i) When WTIM0 = 0



▲1: IICS0 = 1000×110B

▲2: IICS0 = 1000×000B (Sets the WTIM0 bit to 1)

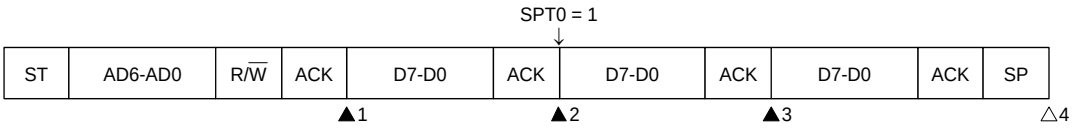
▲3: IICS0 = 1000×100B (Clears the WTIM0 bit to 0)

▲4: IICS0 = 01000100B

△5: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

(ii) When WTIM0 = 1



▲1: IICS0 = 1000×110B

▲2: IICS0 = 1000×100B (Sets the SPT0 bit to 1)

▲3: IICS0 = 01000100B

△4: IICS0 = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE0 = 1
×: Don't care

14.6 Timing Charts

In the I^2C bus mode, the master outputs an address on the serial bus to select a target slave device from among several slave devices.

The master transmits the TRC0 bit (bit 3 of IICA status register 0 (IICS0)) that indicates the data transfer direction following the slave address and starts serial communication with the slave.

Figure 14-31 and **Figure 14-32** show timing charts of data communication.

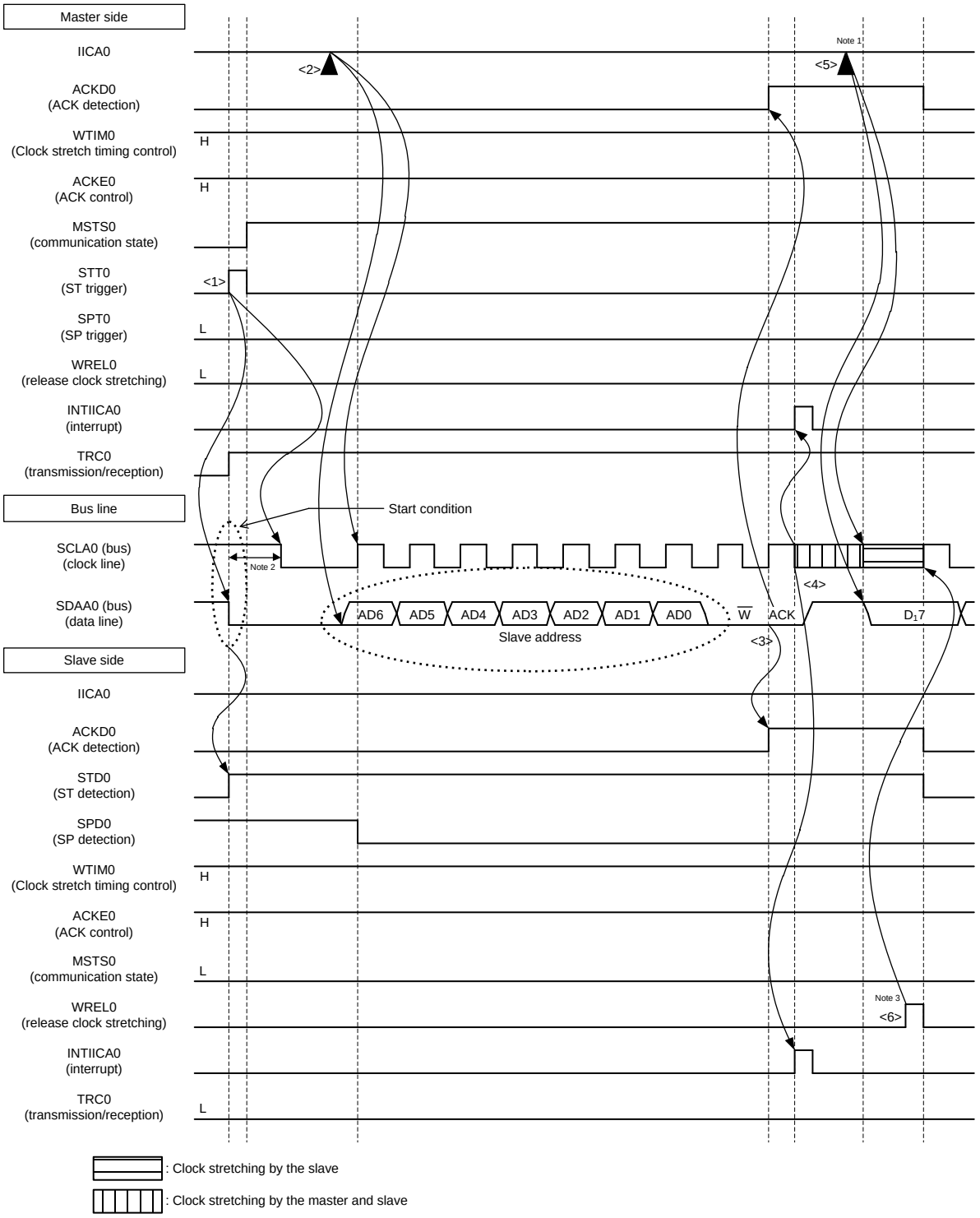
Shift operation of IICA shift register 0 (IICA0) proceeds in synchronization with the falling edge of the serial clock (SCLA0), and the transmit data is transferred to the SO latch and output from the SDAA0 pin with the MSB first.

The data input to the SDAA0 pin is captured in IICA0 at the rising edge of SCLA0.

Figure 14-31. Example of Master to Slave Communications

(9th Cycle Clock Stretching is Selected for Both Master and Slave) (1/4)

(1) Start condition ~ address ~ data



Note 1. To release clock stretching in transmission by the master, write data to the IICA0 register instead of setting the WREL0 bit.

Note 2. Make sure that the time between the fall of the SDAA0 pin signal and the fall of the SCLA0 pin signal is at least 4.0 μs when standard mode is set and at least 0.6 μs when fast mode is set.

Note 3. To release clock stretching in reception by the slave, write FFH to IICA0 or set the WREL0 bit.

Explanation of <1> to <6> in **Figure 14-31 (1) Start condition ~ address ~ data** is given below.

<1> When the start condition trigger is set by the master (STT0 = 1), the bus data line (SDAA0) goes low and a start condition (SDAA0 = 0, SCLA0 = 1) is generated. After that, when the start condition is detected, the master enters the master communication state (MSTS0 = 1). It is ready for communications when the bus clock line goes low (SCLA0 = 0) after the hold time has elapsed.

<2> When the master writes the address + W (transmission) to IICA shift register 0 (IICA0), the slave address is transmitted.

<3> In the slave, if the address received matches its local address (SVA0 value)[Note 1], an ACK is sent to the master by the hardware. The ACK is detected by the master (ACKD0 = 1) at the rising edge of the 9th clock.

<4> The master issues an interrupt (INTIICA0: address transmission end interrupt) at the falling edge of the 9th clock. The slave with the matching address applies clock stretching (SCLA0 = 0) and issues an interrupt (INTIICA0: address match interrupt)[Note 1].

<5> The master writes transmit data to the IICA0 register and releases clock stretching by the master.

<6> If the slave releases clock stretching (WREL0 = 1), the master starts to transfer data to the slave.

Note 1. If the transmitted address does not match the address of the slave, the slave does not return an ACK to the master (NACK: SDAA0 = 1). The slave also neither issue the INTIICA0 interrupt (address match interrupt) nor apply clock stretching.
The master, however, issues the INTIICA0 interrupt (address transmission end interrupt) in response to an ACK or NACK.

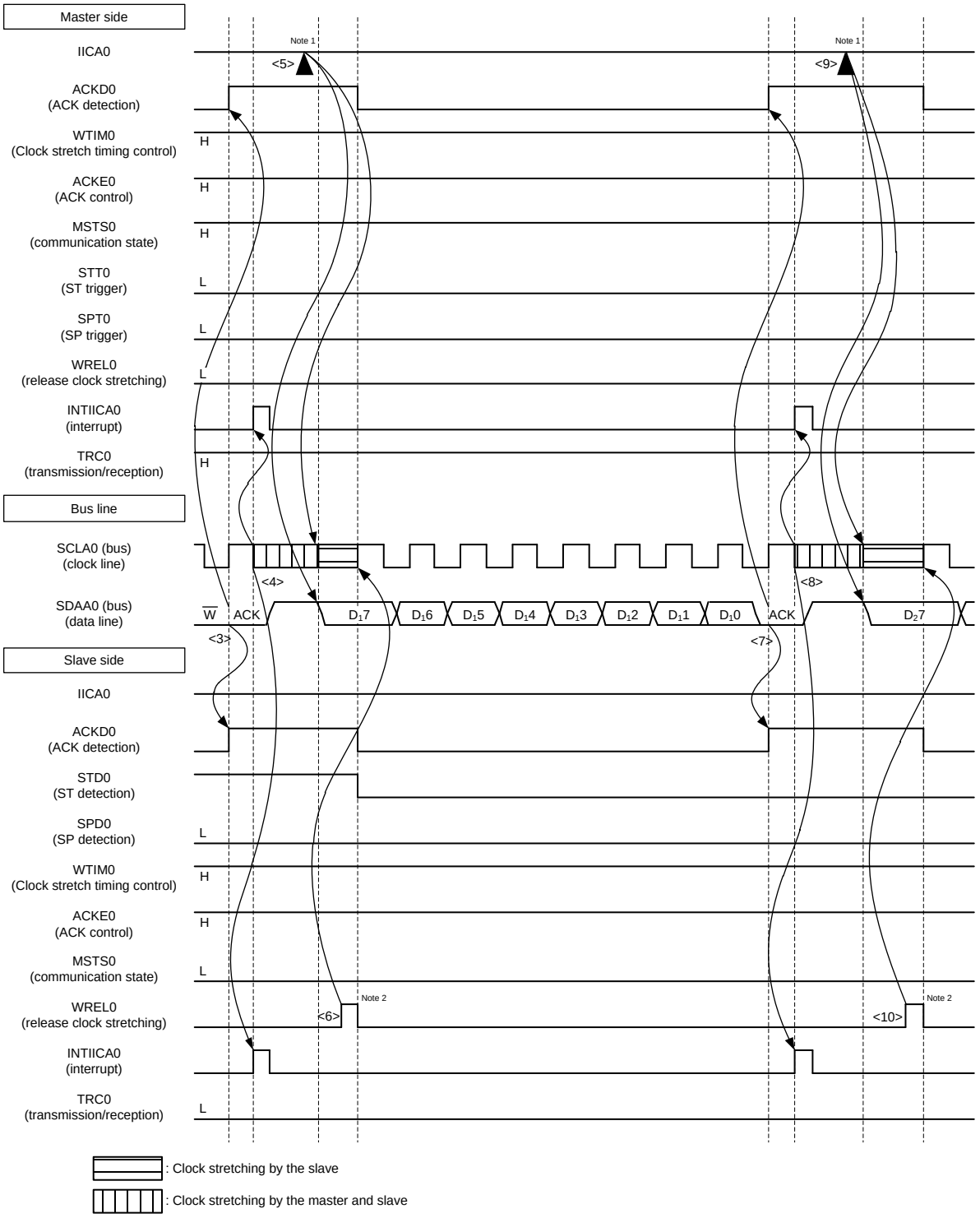
Remark <1> to <15> in **Figure 14-31** represent a series of operation procedures for data communications via the I^2C bus.

- **Figure 14-31 (1) Start condition ~ address ~ data** shows steps <1> to <6>.
- **Figure 14-31 (2) Address ~ data ~ data** shows steps <3> to <10>.
- **Figure 14-31 (3) Data ~ data ~ stop condition** shows steps <7> to <15>.

Figure 14-31. Example of Master to Slave Communications

(9th Cycle Clock Stretching is Selected for Both Master and Slave) (2/4)

(2) Address ~ data ~ data



Note 1. To release clock stretching in transmission by the master, write data to the IICA0 register instead of setting the WREL0 bit.

Note 2. To release clock stretching in reception by the slave, write FFH to IICA0 or set the WREL0 bit.

Explanation of <3> to <10> in **Figure 14-31 (2) Address ~ data ~ data** is given below.

<3> In the slave, if the address received matches its local address (SVA0 value)[Note 1], an ACK is sent to the master by the hardware. The ACK is detected by the master (ACKD0 = 1) at the rising edge of the 9th clock.

<4> The master issues an interrupt (INTIICA0: address transmission end interrupt) at the falling edge of the 9th clock. The slave with the matching address applies clock stretching (SCLA0 = 0) and issues an interrupt (INTIICA0: address match interrupt)[Note 1].

<5> The master writes transmit data to the IICA shift register 0 (IICA0) and releases clock stretching by the master.

<6> If the slave releases clock stretching (WREL0 = 1), the master starts to transfer data to the slave.

<7> After data transfer is completed, because ACKE0 = 1 for the slave, an ACK is sent to the master by the hardware. The ACK is detected by the master (ACKD0 = 1) at the rising edge of the 9th clock.

<8> The master and slave apply clock stretching (SCLA0 = 0) at the falling edge of the 9th clock, and both the master and slave issue an interrupt (INTIICA0: transfer end interrupt).

<9> The master writes transmit data to the IICA0 register and releases clock stretching by the master.

<10> When the slave reads the received data and releases clock stretching (WREL0 = 1), the master starts to transmit data to the slave.

Note 1. If the transmitted address does not match the address of the slave, the slave does not return an ACK to the master (NACK: SDAA0 = 1). The slave also neither issue the INTIICA0 interrupt (address match interrupt) nor apply clock stretching.
The master, however, issues the INTIICA0 interrupt (address transmission end interrupt) in response to an ACK or NACK.

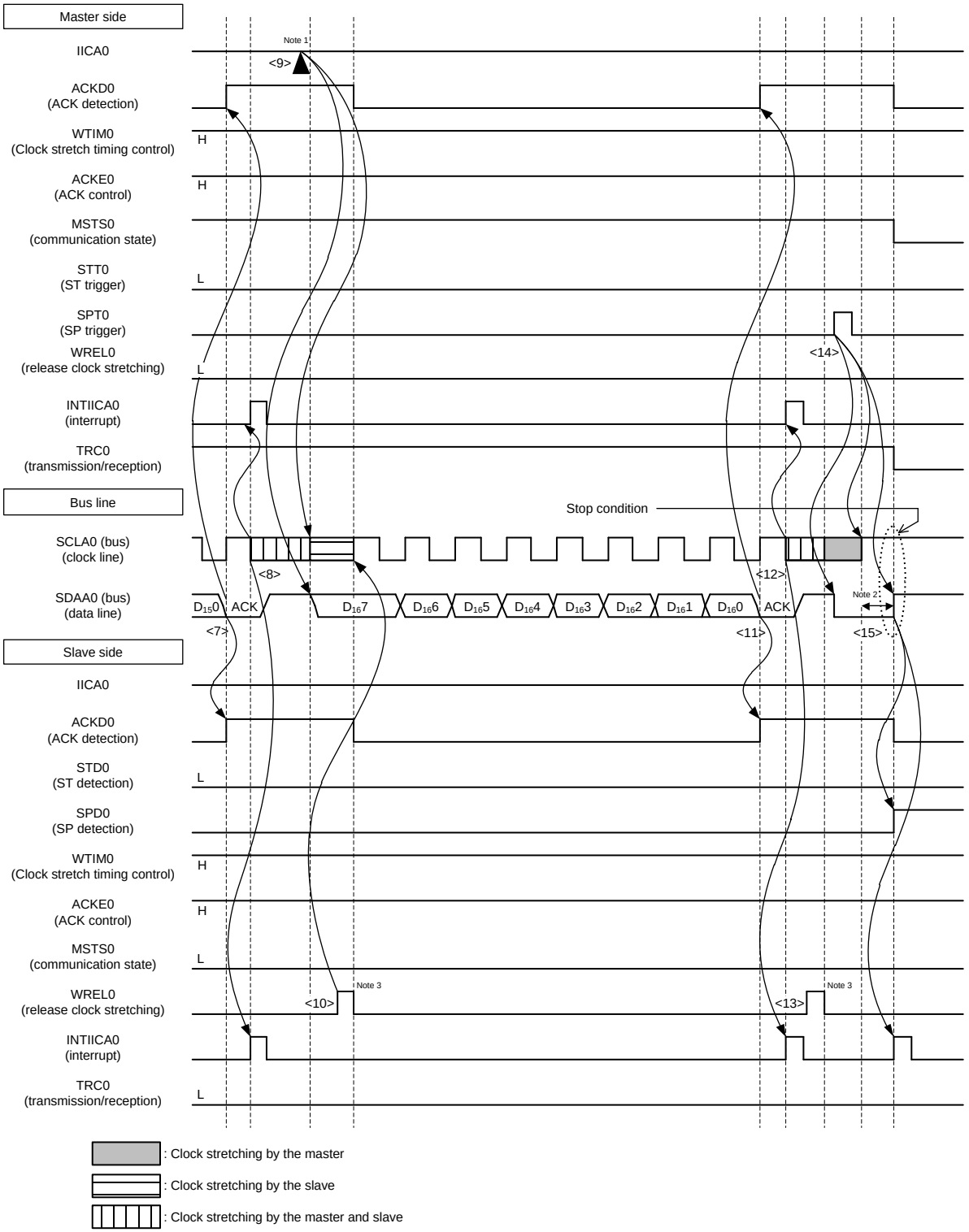
Remark <1> to <15> in **Figure 14-31** represent a series of operation procedures for data communications via the I^2C bus.

- **Figure 14-31 (1) Start condition ~ address ~ data** shows steps <1> to <6>.
- **Figure 14-31 (2) Address ~ data ~ data** shows steps <3> to <10>.
- **Figure 14-31 (3) Data ~ data ~ stop condition** shows steps <7> to <15>.

Figure 14-31. Example of Master to Slave Communications

(9th Cycle Clock Stretching is Selected for Both Master and Slave) (3/4)

(3) Data ~ data ~ stop condition



Note 1. To release clock stretching in transmission by the master, write data to the IICA0 register instead of setting the WREL0 bit.

Note 2. Make sure that the time between the rise of the SCLA0 pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0 μs when standard mode is set and at least 0.6 μs when fast mode is set.

Note 3. To release clock stretching in reception by the slave, write FFH to IICA0 or set the WREL0 bit.

Explanation of <7> to <15> in **Figure 14-31 (3) Data ~ data ~ stop condition** is given below.

<7> After data transfer is completed, because ACKE0 = 1 for the slave, an ACK is sent to the master by the hardware. The ACK is detected by the master (ACKD0 = 1) at the rising edge of the 9th clock.

<8> The master and slave apply clock stretching (SCLA0 = 0) at the falling edge of the 9th clock, and both the master and slave issue an interrupt (INTIICA0: transfer end interrupt).

<9> The master writes transmit data to the IICA shift register 0 (IICA0) and releases clock stretching by the master.

<10> When the slave reads the received data and releases clock stretching (WREL0 = 1), the master starts to transmit data to the slave.

<11> After data transfer is completed, an ACK is sent to the master by the hardware of the slave (ACKE0 =1). The ACK is detected by the master (ACKD0 = 1) at the rising edge of the 9th clock.

<12> The master and slave apply clock stretching (SCLA0 = 0) at the falling edge of the 9th clock, and both the master and slave issue an interrupt (INTIICA0: transfer end interrupt).

<13> The slave reads the received data and releases clock stretching (WREL0 = 1).

<14> When the master sets a stop condition trigger (SPT0 = 1), the bus data line is cleared (SDAA0 = 0) and the bus clock line is set (SCLA0 = 1). After the stop condition setup time has elapsed, a stop condition (SDAA0 changes from 0 to 1 with SCLA0 =1) is generated by the bus data line being set (SDAA0 = 1).

<15> The slave detects the stop condition and issues an interrupt (INTIICA0: stop condition interrupt).

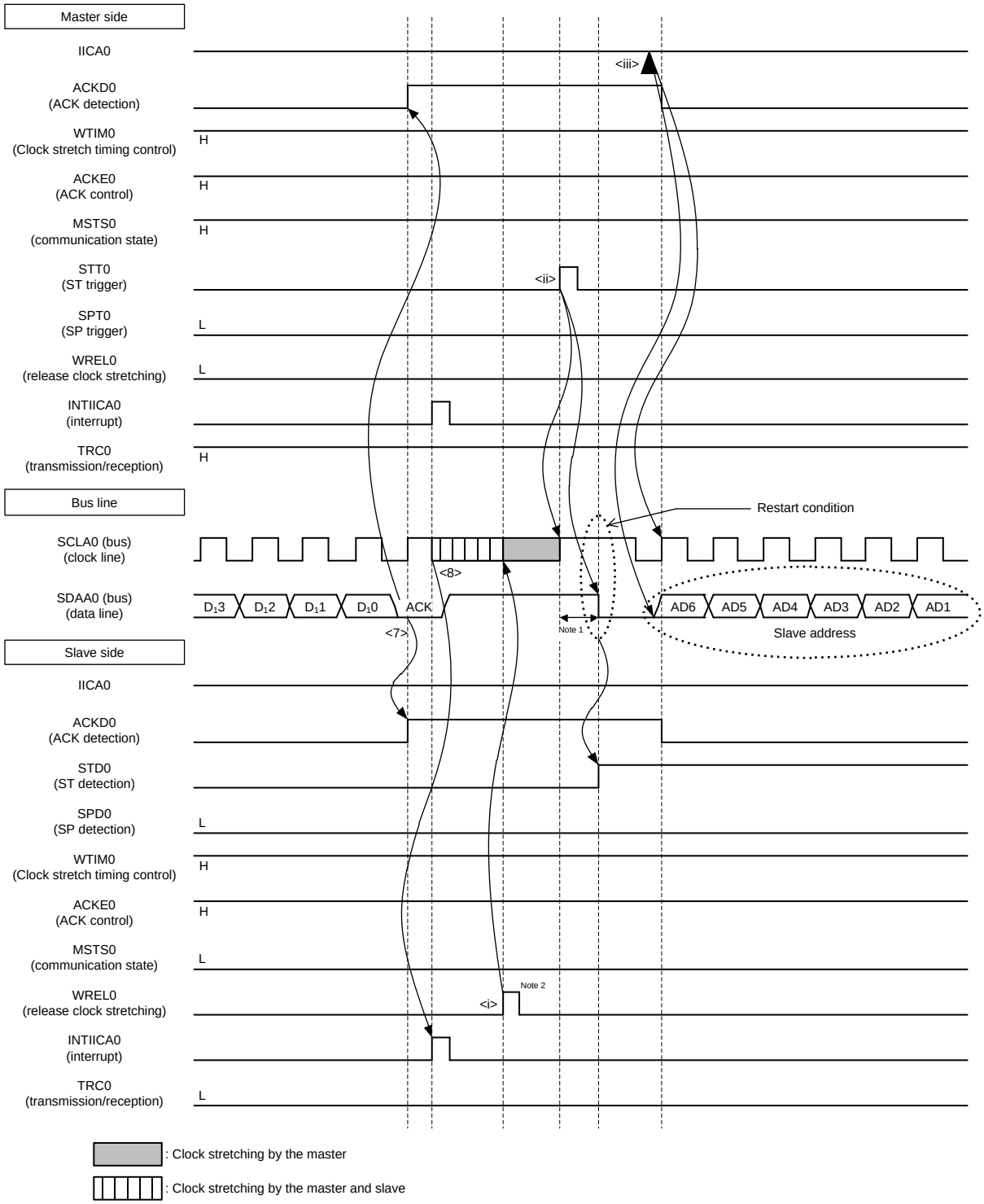
Remark <1> to <15> in **Figure 14-31** represent a series of operation procedures for data communications via the I^2C bus.

- **Figure 14-31 (1) Start condition ~ address ~ data** shows steps <1> to <6>.
- **Figure 14-31 (2) Address ~ data ~ data** shows steps <3> to <10>.
- **Figure 14-31 (3) Data ~ data ~ stop condition** shows steps <7> to <15>.

Figure 14-31. Example of Master to Slave Communications

(9th Cycle Clock Stretching is Selected for Both Master and Slave) (4/4)

(4) Data ~ restart condition ~ address



Note 1. Make sure that the time between the rise of the SCLA0 pin signal and the generation of the start condition after a restart condition has been issued is at least 4.7 μs when standard mode is set and at least 0.6 μs when fast mode is set.

Note 2. To release clock stretching in reception by the slave, write FFH to IICA0 or set the WREL0 bit.

The following describes the operations in **Figure 14-31 (4) Data ~ restart condition ~ address**. After the operations of steps <7> and <8>, the operations of steps <i> to <iii> are performed. As a result, processing returns to the data transmission procedure in step <3>.

<7> After data transfer is completed, because ACKE0 = 1 for the slave, an ACK is sent to the master by the hardware. The ACK is detected by the master (ACKD0 = 1) at the rising edge of the 9th clock.

<8> The master and slave apply clock stretching (SCLA0 = 0) at the falling edge of the 9th clock, and both the master and slave issue an interrupt (INTIICA0: transfer end interrupt).

<i> The slave reads the received data and releases clock stretching (WREL0 = 1).

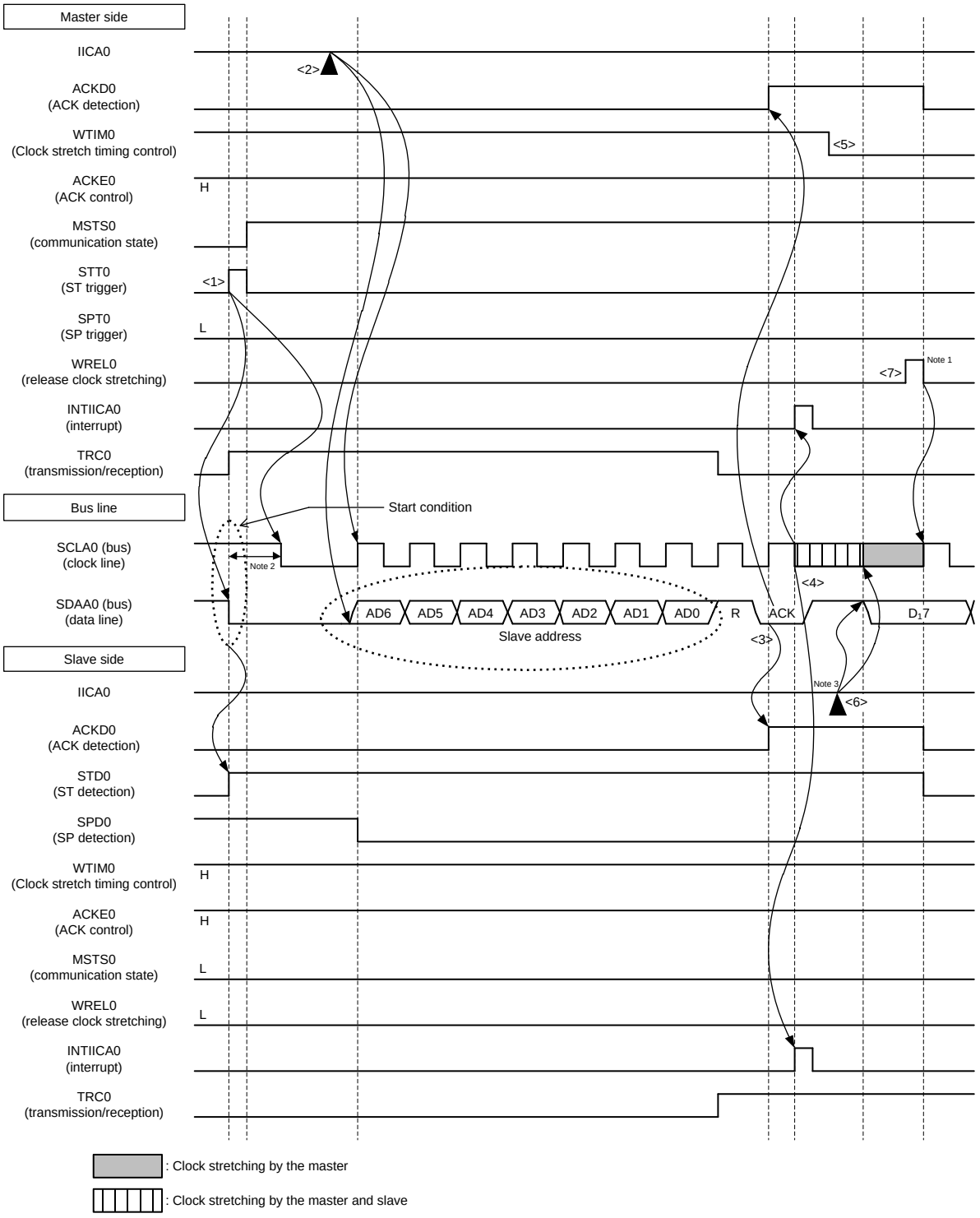
<ii> When the start condition trigger is set again by the master (STT0 = 1), the bus clock line goes high (SCLA0 = 1), the bus data line goes low (SDAA0 = 0) after the restart condition setup time has elapsed, and a start condition (SDAA0 changes from 1 to 0 with SCLA0 = 1) is generated. After that, when the start condition is detected, the master is ready for communications when the bus clock line goes low (SCLA0 = 0) after the hold time has elapsed.

<iii> When the master writes the address + R/W (transmission) to the IICA shift register 0 (IICA0), the slave address is transmitted.

Figure 14-32. Example of Slave to Master Communications

(8th Cycle Clock Stretching is Selected for Master, 9th Cycle Clock Stretching is Selected for Slave) (1/3)

(1) Start condition ~ address ~ data



Note 1. To release clock stretching in reception by the master, write FFH to IICA0 or set the WREL0 bit.

Note 2. Make sure that the time between the fall of the SDAA0 pin signal and the fall of the SCLA0 pin signal is at least 4.0 μs when standard mode is set and at least 0.6 μs when fast mode is set.

Note 3. To release clock stretching in transmission by the slave, write data to the IICA0 register instead of setting the WREL0 bit.

Explanation of <1> to <7> in **Figure 14-32 (1) Start condition ~ address ~ data** is given below.

<1> When the start condition trigger is set by the master (STT0 = 1), the bus data line (SDAA0) goes low and a start condition (SDAA0 changes from 1 to 0 with SCLA0 = 1) is generated. After that, when the start condition is detected, the master enters the master communication state (MSTS0 = 1). It is ready for communications when the bus clock line goes low (SCLA0 = 0) after the hold time has elapsed.

<2> When the master writes the address + R (reception) to IICA shift register 0 (IICA0), the slave address is transmitted.

<3> In the slave, if the address received matches its local address (SVA0 value)[Note 1], an ACK is sent to the master by the hardware. The ACK is detected by the master (ACKD0 = 1) at the rising edge of the 9th clock.

<4> The master issues an interrupt (INTIICA0: address transmission end interrupt) at the falling edge of the 9th clock. The slave with the matching address applies clock stretching (SCLA0 = 0) and issues an interrupt (INTIICA0: address match interrupt)[Note 1].

<5> The timing of clock stretching by the master is changed to the 8th clock (WTIM0 = 0).

<6> The slave writes transmit data to the IICA0 register and releases clock stretching by the slave.

<7> The master releases clock stretching (WREL0 = 1) and starts data transfer with the slave.

Note 1. If the transmitted address does not match the address of the slave, the slave does not return an ACK to the master (NACK: SDAA0 = 1). The slave also neither issue the INTIICA0 interrupt (address match interrupt) nor apply clock stretching.
The master, however, issues the INTIICA0 interrupt (address transmission end interrupt) in response to an ACK or NACK.

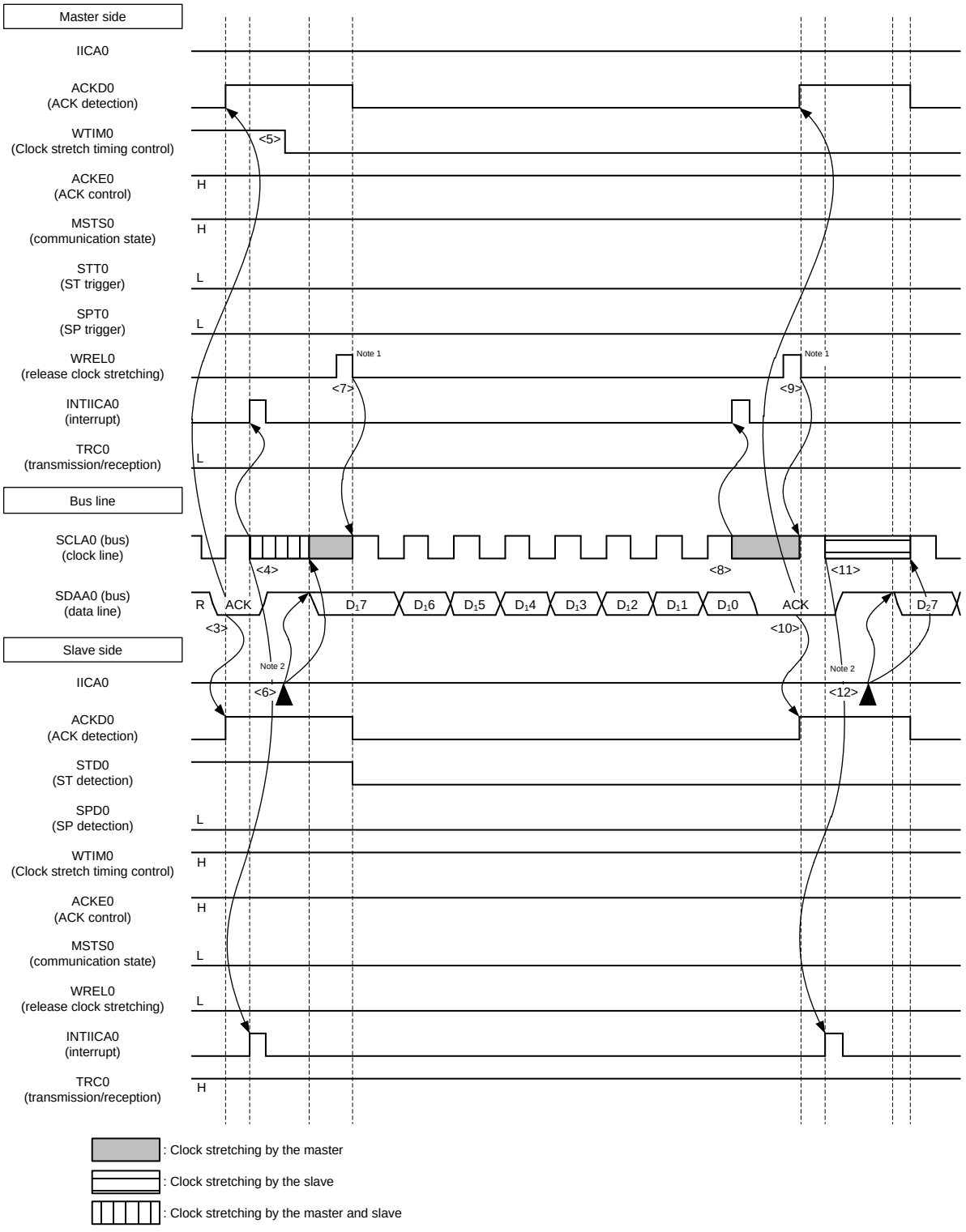
Remark <1> to <19> in **Figure 14-32** represent a series of operation procedures for data communications via the I^2C bus.

- **Figure 14-32 (1) Start condition ~ address ~ data** shows steps <1> to <7>.
- **Figure 14-32 (2) Address ~ data ~ data** shows steps <3> to <12>.
- **Figure 14-32 (3) Data ~ data ~ stop condition** shows steps <8> to <19>.

Figure 14-32. Example of Slave to Master Communications

(8th Cycle Clock Stretching is Selected for Master, 9th Cycle Clock Stretching is Selected for Slave) (2/3)

(2) Address ~ data ~ data



Note 1. To release clock stretching in reception by the master, write FFH to IICA0 or set the WREL0 bit.

Note 2. To release clock stretching in transmission by the slave, write data to the IICA0 register instead of setting the WREL0 bit.

Explanation of <3> to <12> in **Figure 14-32 (2) Address ~ data ~ data** is given below.

<3> In the slave, if the address received matches its local address (SVA0 value)[Note 1], an ACK is sent to the master by the hardware. The ACK is detected by the master (ACKD0 = 1) at the rising edge of the 9th clock.

<4> The master issues an interrupt (INTIICA0: address transmission end interrupt) at the falling edge of the 9th clock. The slave with the matching address applies clock stretching (SCLA0 = 0) and issues an interrupt (INTIICA0: address match interrupt)[Note 1].

<5> The timing of clock stretching by the master is changed to the 8th clock (WTIM0 = 0).

<6> The slave writes transmit data to the IICA shift register 0 (IICA0) and releases clock stretching by the slave.

<7> The master releases clock stretching (WREL0 = 1) and starts data transfer with the slave.

<8> The master applies clock stretching (SCLA0 = 0) at the falling edge of the 8th clock and issues an interrupt (INTIICA0: transfer end interrupt). Because ACKE0 = 0 for the master, an ACK is sent to the slave by the hardware.

<9> The master reads the received data and releases clock stretching (WREL0 = 1).

<10> The ACK is detected by the slave (ACKD0 = 1) at the rising edge of the 9th clock.

<11> The slave applies clock stretching (SCLA0 = 0) at the falling edge of the 9th clock and issues an interrupt (INTIICA0: transfer end interrupt).

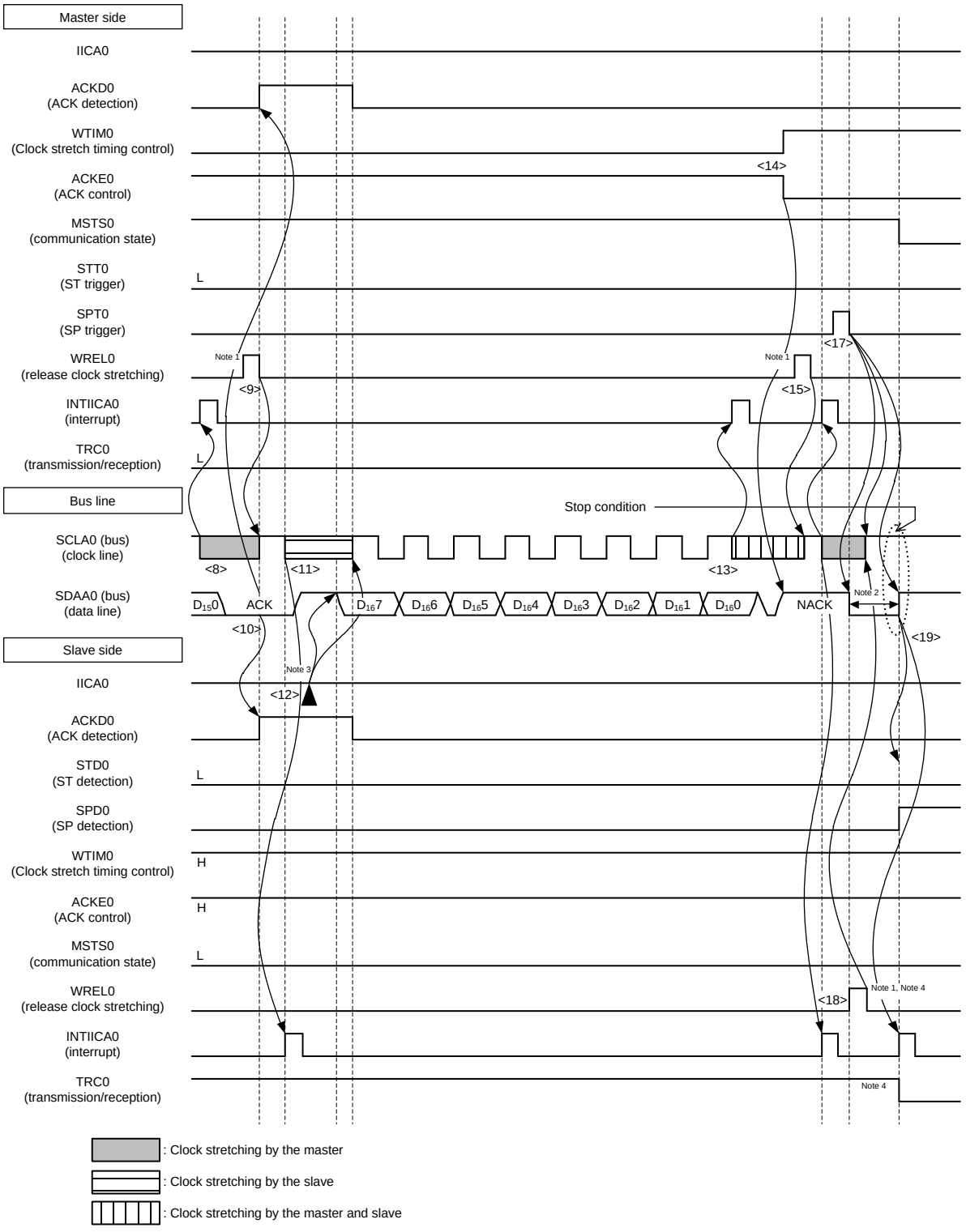
<12> When the slave writes transmit data to the IICA0 register, clock switching by the slave is released. The slave then starts to transfer data to the master.

Note 1. If the transmitted address does not match the address of the slave, the slave does not return an ACK to the master (NACK: SDAA0 = 1). The slave also neither issue the INTIICA0 interrupt (address match interrupt) nor apply clock stretching.
The master, however, issues the INTIICA0 interrupt (address transmission end interrupt) in response to an ACK or NACK.

Remark <1> to <19> in **Figure 14-32** represent a series of operation procedures for data communications via the I^2C bus.

- **Figure 14-32 (1) Start condition ~ address ~ data** shows steps <1> to <7>.
- **Figure 14-32 (2) Address ~ data ~ data** shows steps <3> to <12>.
- **Figure 14-32 (3) Data ~ data ~ stop condition** shows steps <8> to <19>.

Figure 14-32. Example of Slave to Master Communications
(8th Cycle Clock Stretching is Changed to 9th Cycle Clock Stretching for Master,
9th Cycle Clock Stretching is Selected for Slave) (3/3)
(3) Data ~ data ~ stop condition



Note 1. To release clock stretching, write FFH to IICA0 or set the WREL0 bit.

Note 2. Make sure that the time between the rise of the SCLA0 pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0 μs when standard mode is set and at least 0.6 μs when fast mode is set.

Note 3. To release clock stretching in transmission by the slave, write data to the IICA0 register instead of setting the WREL0 bit.

Note 4. If clock stretching in transmission by the slave is released by setting the WREL0 bit, the TRC0 bit will be cleared.

Explanation of <8> to <19> in **Figure 14-32 (3) Data ~ data ~ stop condition** is given below.

<8> The master applies clock stretching (SCLA0 = 0) at the falling edge of the 8th clock and issues an interrupt (INTIICA0: transfer end interrupt). Because ACKE0 = 0 for the master, an ACK is sent to the slave by the hardware.

<9> The master reads the received data and releases clock stretching (WREL0 = 1).

<10> The ACK is detected by the slave (ACKD0 = 1) at the rising edge of the 9th clock.

<11> The slave applies clock stretching (SCLA0 = 0) at the falling edge of the 9th clock and issues an interrupt (INTIICA0: transfer end interrupt).

<12> When the slave writes transmit data to the IICA0 register, clock switching by the slave is released. The slave then starts to transfer data to the master.

<13> The master issues an interrupt (INTIICA0: transfer end interrupt) at the falling edge of the 8th clock and applies clock stretching (SCLA0 = 0). Because ACK control (ACKE0 = 1) is used, the bus data line is at the low level (SDAA0 = 0) at this stage.

<14> The master sets NACK as the response (ACKE0 = 0) and changes the timing of clock stretching to the 9th clock (WTIM0 = 1).

<15> If the master releases clock stretching (WREL0 = 1), the slave detects the NACK (ACK = 0) at the rising edge of the 9th clock.

<16> The master and slave apply clock stretching (SCLA0 = 0) at the falling edge of the 9th clock, and both the master and slave issue an interrupt (INTIICA0: transfer end interrupt).

<17> When the master issues a stop condition (SPT0 = 1), the bus data line is cleared (SDAA0 = 0) and the master releases clock stretching. The master then waits until the bus clock line is set (SCLA0 = 1).

<18> The slave acknowledges the NACK, halts transmission, and releases clock stretching (WREL0 = 1) to end communications. Once the slave releases clock stretching, the bus clock line is set (SCLA0 = 1).

<19> When the master confirms that the bus clock line has been set (SCLA0 = 1), it sets the bus data line (SDAA0 = 1) and issues a stop condition (SDAA0 changes from 0 to 1 with SCLA0 =1) after the stop condition setup time has elapsed. The slave detects this stop condition and issues an interrupt (INTIICA0: stop condition interrupt).

Remark <1> to <19> in **Figure 14-32** represent a series of operation procedures for data communications via the I^2C bus.

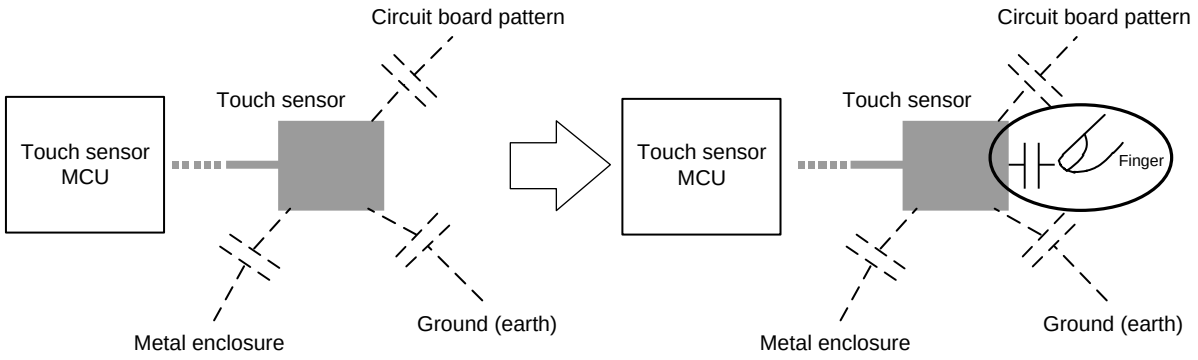
- **Figure 14-32 (1) Start condition ~ address ~ data** shows steps <1> to <7>.
- **Figure 14-32 (2) Address ~ data ~ data** shows steps <3> to <12>.
- **Figure 14-32 (3) Data ~ data ~ stop condition** shows steps <8> to <19>.

CHAPTER 15 CAPACITIVE TOUCH SENSING UNIT (CTSUb)

The capacitive touch sensing unit (CTSU) measures the electrostatic capacitance of the touch sensor. Changes in the electrostatic capacitance are determined by software, which enables the CTSU to detect whether a finger is in contact with the touch sensor. The electrode surface of the touch sensor is usually enclosed with a dielectric so that a finger does not come into contact with the electrode.

As shown in **Figure 15-1**, electrostatic capacitance (parasitic capacitance) exists between the electrode and the surrounding conductors. Because the human body is an electrical conductor, when a finger is placed close to the electrode, the value of electrostatic capacitance increases.

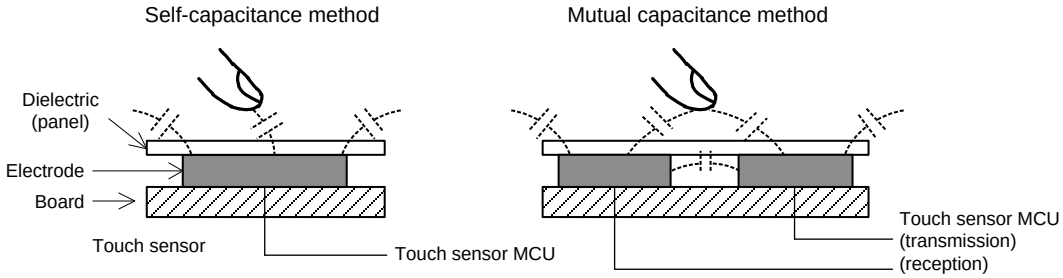
Figure 15-1. Increased Electrostatic Capacitance Due to Presence of Finger



Electrostatic capacitance is detected by the following methods: Self-capacitance and mutual capacitance.

In the self-capacitance method, the CTSU detects electrostatic capacitance generated between a finger and a single electrode. In the mutual capacitance method, two electrodes are used as a transmit electrode and a receive electrode, and the CTSU detects the change in the electrostatic capacitance generated between the two when a finger is placed close to them.

Figure 15-2. Self-Capacitance Method and Mutual Capacitance Method



Electrostatic capacitance is measured by counting a clock signal whose frequency changes according to the amount of charged/discharged current, for a specified period.

For details on the measurement principles of the CTSU, refer to **15.4.1 Principles of measurement operation**.

For development and implementation of touch applications, refer to the application notes below.

- Capacitive Sensor Microcontrollers CTSU Capacitive Touch Introduction Guide (R30AN0424)
- RL78 Family Using the standalone version of QE to Develop Capacitive Touch Applications (R01AN6574)
- RL78 Family CTSU Module Software Integration System (R11AN0484)
- RL78 Family TOUCH Module Software Integration System (R11AN0485)

15.1 Overview

Table 15-1 lists the specifications of the CTSU.

Table 15-1. CTSU Specifications

Item		Description
Operating clock[Note 1]		f_{CLK}, $f_{CLK}/2$, or $f_{CLK}/4$
Pins	TS00 to TS14	Electrostatic capacitance measurement pins (15 channels)
	TSCAP	LPF (low-pass filter) connection pin We recommend connecting a 10-nF capacitor.
Measurement modes	Self-capacitance single scan mode	Electrostatic capacitance on a channel is measured by the self-capacitance method.
	Self-capacitance multi-scan mode	Electrostatic capacitance on multiple channels is measured successively by the self-capacitance method.
	Mutual capacitance full scan mode	Electrostatic capacitance on multiple channels is measured successively by the mutual capacitance method.
Noise prevention		Synchronous noise prevention, high-pass noise prevention
Measurement start conditions		• Software trigger • External trigger (an interval interrupt signal from the 12-bit interval timer)

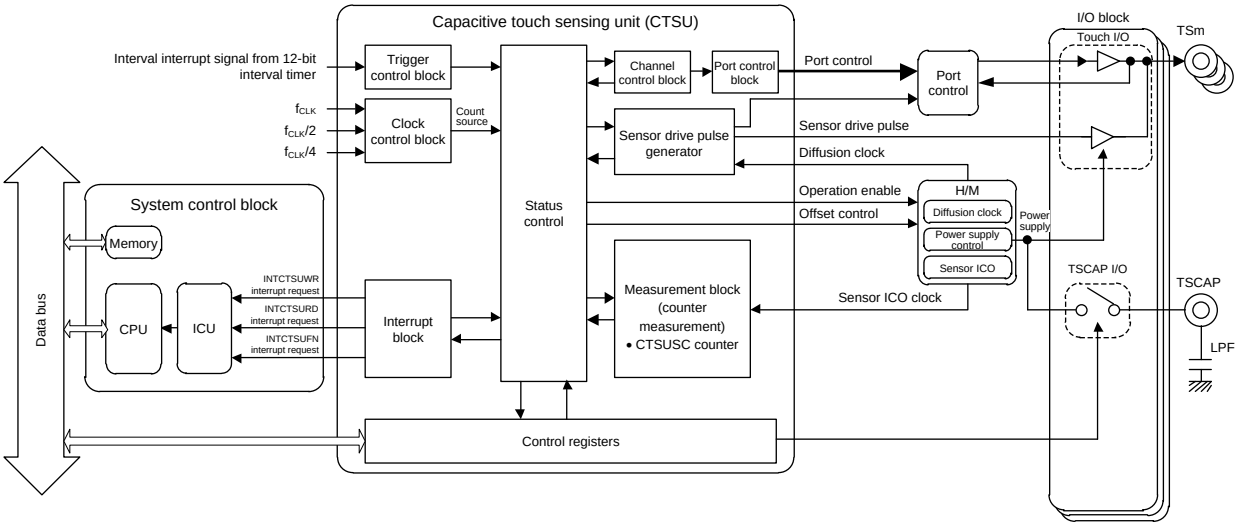
<R> Note 1. The measurement accuracy of the capacitive sensing unit (CTSUb) depends on the accuracy of the operating clock. Do not select the subsystem clock (f_{SUB}) as the CPU/peripheral hardware clock (f_{CLK}).

15.2 CTSU Configuration

The CTSU consists of the status control block, trigger control block, clock control block, channel control block, port control block, sensor drive pulse generator, measurement block, interrupt block, and control registers.

Figure 15-3 shows the CTSU block diagram, and **Table 15-2** lists the CTSU pin configuration.

Figure 15-3. CTSU Block Diagram



<R> **Remark** ICO: Intensity of Current controlled Oscillator
ICU: Interrupt Control Unit
m = 00 to 14

Table 15-2. CTSU Pin Configuration

Pin Name	I/O	Function
TS00 to TS14	I/O	Electrostatic capacitive measurement pins (touch pins)
TSCAP	—	LPF connection pin

15.3 Registers Controlling CTSU

Registers controlling the CTSU are shown below.

- Peripheral enable register 1 (PER1)
- CTSU control register 0 (CTSUCR0)
- CTSU control register 1 (CTSUCR1)
- CTSU synchronous noise reduction setting register (CTSUSDPRS)
- CTSU sensor stabilization wait control register (CTSUSST)
- CTSU measurement channel register 0 (CTSUMCH0)
- CTSU measurement channel register 1 (CTSUMCH1)
- CTSU channel enable control register 0 (CTSUCHAC0)
- CTSU channel enable control register 1 (CTSUCHAC1)
- CTSU channel transmit/receive control register 0 (CTSUCHTRC0)
- CTSU channel transmit/receive control register 1 (CTSUCHTRC1)
- CTSU high-pass noise reduction control register (CTSUDCLKC)
- CTSU status register (CTSUST)
- CTSU high-pass noise reduction spectrum diffusion control register (CTSUSSC)
- CTSU sensor offset register 0 (CTSUSO0)
- CTSU sensor offset register 1 (CTSUSO1)
- CTSU sensor counter (CTSUSC)
- CTSU reference counter (CTSURC)
- CTSU error status register (CTSUERRS)
- Touch pin function select register 0 (TSSEL0)
- Touch pin function select register 1 (TSSEL1)
- TSCAP pin setting register (VTSEL)
- CTSU trimming register (RTRIM)
- CTSU trimming result register (CTSUTRESULT)

15.3.1 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise.

When the CTSU is used, be sure to set bit 1 (CTSUEN) of this register to 1.

The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 15-4. Format of Peripheral Enable Register 1 (PER1)

Address: F007AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PER1	RTCWEN	0	0	0	0	0	CTSUEN	0

CTSUEN	Control of CTSU input clock supply
0	Stops input clock supply. • SFR used by the CTSU cannot be written. • The CTSU is in the reset status.
1	Enables input clock supply. • SFR used by the CTSU can be read/written.

Caution When using the CTSU, be sure to first set the CTSUEN bit to 1 and then set the following registers. If the CTSUEN bit = 0, the values of the registers controlling the CTSU are cleared to their initial values and writing to them is ignored (except for touch pin function select registers 0 and 1 (TSSEL0 and TSSEL1), TSCAP pin setting register (VTSEL), CTSU trimming register (RTRIM), and CTSU trimming result register (CTSUTRESULT)).

- **CTSU control register 0 (CTSUCR0)**
- **CTSU control register 1 (CTSUCR1)**
- **CTSU synchronous noise reduction setting register (CTSUSDPRS)**
- **CTSU sensor stabilization wait control register (CTSUSST)**
- **CTSU measurement channel register 0 (CTSUMCH0)**
- **CTSU measurement channel register 1 (CTSUMCH1)**
- **CTSU channel enable control register 0 (CTSUCHAC0)**
- **CTSU channel enable control register 1 (CTSUCHAC1)**
- **CTSU channel transmit/receive control register 0 (CTSUCHTRC0)**
- **CTSU channel transmit/receive control register 1 (CTSUCHTRC1)**
- **CTSU high-pass noise reduction control register (CTSUDCLKC)**
- **CTSU status register (CTSUST)**
- **CTSU high-pass noise reduction spectrum diffusion control register (CTSUSSC)**
- **CTSU sensor offset register 0 (CTSUSO0)**
- **CTSU sensor offset register 1 (CTSUSO1)**
- **CTSU sensor counter (CTSUSC)**
- **CTSU error status register (CTSUERRS)**

15.3.2 CTSU control register 0 (CTSUCR0)

The CTSUCR0 register is used to select the transmission power supply and operation start trigger, enable or disable suspension, and start or stop the measurement operation for the CTSU.

The CTSUCR0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 15-5. Format of CTSU Control Register 0 (CTSUCR0) (1/3)

Address: F0380H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CTSUCR0	0	0	0	CTSUINIT	CTSUIOC	CTSUSNZ	CTSUCAP	CTSUSTRT

<R>

CTSUINIT	CTSU control block initialization[Note 1, Note 2]
0	—
1	The CTSU control block and registers are initialized.
This bit is read as 0. To forcibly stop the current operation, set the CTSUSTRT bit to 0 and the CTSUINIT bit to 1 simultaneously. In this case, the operation is stopped and the internal control registers are initialized.	

CTSUIOC	CTSU transmit pin control
0	Low-level output from the TS pins
1	High-level output from the TS pins
This bit selects the level output from the TS pins when the CTSUTSOD bit in the CTSUERRS register is set to 1. This bit is ignored when the CTSUTSOD bit is set to 0.	

Figure 15-5. Format of CTSU Control Register 0 (CTSUCR0) (2/3)

Address: F0380H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CTSUCR0	0	0	0	CTSUINIT	CTSUIOC	CTSUSNZ	CTSUCAP	CTSUSTRT

CTSUSNZ	CTSU suspension enable[Note 3]
0	Suspension is disabled.
1	Suspension is enabled.

<R>

Setting this bit to 1 drives the CTSU into the suspended state, which decreases power consumption during the wait state for measurement. The suspended state refers to the state in which the capacitor of the external low-pass filter connected to the TSCAP pin is not being charged.

The CTSU state changes as follows depending on the register setting.

<State control of the CTSU hard macro>

VDC: CTSU measurement power

CTSUPON Bit in CTSUCR1 Register	CTSUSNZ Bit	CTSUCAP Bit	CTSUSTRT Bit	External Trigger	CTSU State
0	0	0	0	—	Stopped
1	0	0	0	—	Waiting for the start of measurement (VDC = ON)
1	0	0	1	—	Measuring in normal operating mode (VDC = ON)
1	1	1	0	—	Preparing for setting measurement by an external trigger (VDC = OFF)
1	1	1	1	Not detected (waiting)	Suspended (waiting for an external trigger) (VDC = OFF)
1	1	1	1	Detection of rising edges (operating)	Measuring in normal operating mode (VDC = ON)[Note 4]
1	1	0	0	—	SW suspended (VDC = OFF)
Other than the above					Setting prohibited

(1) Suspended state

While the CTSU is in the wait state for an external trigger by setting the CTSUSTRT bit to 1 after selecting an external trigger (the CTSUCAP bit = 1) and enabling suspension (the CTSUSNZ bit = 1), the CPU can be placed in STOP mode.

When a rising edge of the external trigger is detected during STOP mode, the CTSU issues a clock request to the clock generating block and makes a transition to normal operating mode to start measurement.

(2) SW suspended state

The SW suspended state refers to the state of suspension initiated when the software trigger has been selected (the CTSUCA bit = 0) and suspension has been enabled (the CTSUSNZ = 1).

This state is used when placing the CTSU hard macro in the suspended state to decrease power consumption by software. In the SW suspended state, placing the CPU in STOP mode is also possible. For return from STOP state, an interrupt is used.

To start measurement from the SW suspended state, set the CTSUSNZ bit to 0 and then wait for at least 64 cycles of the base clock (e.g.: at least 128 µs when the base clock is at 0.5 MHz) before setting the CTSUSTRT bit to 1.

To resume the SW suspended state after the end of measurement, set the CTSUSNZ bit to 1.

Figure 15-5. Format of CTSU Control Register 0 (CTSUCR0) (3/3)

Address: F0380H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CTSUCR0	0	0	0	CTSUINIT	CTSUIOC	CTSUSNZ	CTSUCAP	CTSUSTRT

CTSUCAP	CTSU measurement operation start trigger selection Note 3
0	Software trigger
1	External trigger (an interval interrupt signal from the 12-bit interval timer)

CTSUSTRT	CTSU measurement operation start Note 1
0	Measurement operation stops.
1	Measurement operation starts.

When the CTSUCAP bit is 0 (software trigger), measurement is started by writing 1 to the CTSUSTRT bit, and the CTSUSTRT bit becomes 0 when measurement is finished.

When the CTSUCAP bit is 1 (external trigger), the CTSU waits for an external trigger by writing 1 to the CTSUSTRT bit, and measurement is started at the rising edge of the external trigger. When measurement is finished, the CTSU waits for the next external trigger and operation continues.

If 1 is written to the CTSUSTRT bit when it is 1, writing is ignored and operation continues.

To forcibly stop operation (forced stop) when the CTSUSTRT bit is 1, set the CTSUSTRT bit to 0 and the CTSUINIT bit to 1 simultaneously.

The CTSU states are listed below.

CTSUSTRT Bit	CTSUCAP Bit	CTSU State
0	0	Stopped
0	1	Stopped
1	0	During measurement
1	1	During measurement or wait for an external trigger Note 5

Note 1. Do not write 1 to the CTSUINIT bit at the same time as setting the CTSUSTRT bit to 1 (CTSU operation starts).

Note 2. The CTSUSC, CTSUMCH0, CTSUMCH1, and CTSUST registers are initialized.

Note 3. The CTSUCAP and CTSUSNZ bits should be set while the CTSUSTRT bit is 0. These bits can be set at the same time as the CTSUSTRT bit is set to 1.

Note 4. When a trigger occurs during STOP mode, measurement is performed in normal measurement mode.

Note 5. The state can be read from the CTSUSTC[2:0] flags in the CTSUST register.
During measurement: CTSUSTC[2:0] flags in the CTSUST register ≠ 000B
Wait for an external trigger: CTSUSTC[2:0] flags in the CTSUST register = 000B

15.3.3 CTSU control register 1 (CTSUCR1)

The CTSUCR1 register is used to set the measurement mode, operating clock, and power supply capacity.

The CTSUCR1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 15-6. Format of CTSU Control Register 1 (CTSUCR1)

Address: F0381H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CTSUCR1	CTSUMD1	CTSUMD0	CTSUCLK1	CTSUCLK0	CTSUATUNE1	0	CTSUCSW	CTSUPON

CTSUMD1	CTSUMD0	CTSU measurement mode selection
0	0	Self-capacitance single scan mode
0	1	Self-capacitance multi-scan mode
1	0	Setting prohibited
1	1	Mutual capacitance full scan mode

CTSUCLK1	CTSUCLK0	CTSU operating clock selection
0	0	f_{CLK}
0	1	$f_{CLK}/2$ (f_{CLK} divided by 2)
1	0	$f_{CLK}/4$ (f_{CLK} divided by 4)
1	1	Setting prohibited

CTSUATUNE1	CTSU power supply capacity adjustment[Note 1]
0	Normal output
1	High-current output

CTSUCSW	CTSU LPF capacitance charging control[Note 2]
0	Capacitance switch turned off
1	Capacitance switch turned on
This bit controls charging of the LPF capacitor connected to the TSCAP pin (turning on/off of the capacitance switch). After the capacitance switch is turned on, wait until the capacitance connected to the TSCAP pin is charged for the specified time before starting measurement (the CTSUSTRT bit in the CTSUCR0 register = 1). Prior to charging the capacitance, use an I/O port to output a low level to the TSCAP pin, and discharge the LPF capacitance that has been already charged.	

CTSUPON	CTSU power supply enable[Note 2]
0	Powered off
1	Powered on

Note 1. Normally, the value of this bit should be set to 0.

Note 2. Set the CTSUPON and CTSUCSW bits to the same value at the same time.

Caution The CTSUCR1 register should be set when the CTSUSTRT bit in the CTSUCR0 register is 0.

15.3.4 CTSU synchronous noise reduction setting register (CTSUSDPRS)

The CTSUSDPRS register is used to set the base period, pulse count, and measurement time, and turn on or off the high-pass noise reduction function.

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

Reset signal generation clears this register to 00H.

Figure 15-7. Format of CTSU Synchronous Noise Reduction Setting Register (CTSUSDPRS)

Address: F0382H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CTSUSDPRS	0	CTSUSOFF	CTSUPRMODE1	CTSUPRMODE0	CTSUPRRATIO3	CTSUPRRATIO2	CTSUPRRATIO1	CTSUPRRATIO0

CTSUSOFF	CTSU high-pass noise reduction function off setting
0	High-pass noise reduction function turned on
0	High-pass noise reduction function turned off
This bit turns on or off the function for reducing high-pass noise. Set this bit to 1 when turning off the high-pass noise reduction function.	

CTSUPRMODE1	CTSUPRMODE0	CTSU base period and pulse count setting
0	0	510 pulses
0	1	126 pulses
1	0	62 pulses (recommended setting value)
1	1	Setting prohibited
These bits select the number of base pulses during measurement.		

CTSUPRRATIO[3:0]	CTSU measurement time and pulse count adjustment
These bits are used to determine the measurement time and the number of measurement pulses. Set these bits to 3 (0011B), which is the recommended setting. These are calculated from the following formula in accord with the setting of the CTSUPRMODE[1:0] bits, which determines the number of base pulses. Number of measurement pulses = number of base pulses × (CTSUPRRATIO[3:0] bits + 1) Measurement time = (number of base pulses × (CTSUPRRATIO[3:0] bits + 1) + (number of base pulses − 2) × 0.25) × base clock cycle × CTSU measurement count Note 1	

Note 1. For details on the base clock cycle, refer to **15.3.16 CTSU sensor offset register 1 (CTSUSO1)**.

Caution The CTSUSDPRS register should be set when the CTSUSTRT bit in the CTSUCR0 register is 0.

15.3.5 CTSU sensor stabilization wait control register (CTSUSST)

The CTSUSST register is used to set the CTSU sensor stabilization wait time.

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

Reset signal generation clears this register to 00H.

Figure 15-8. Format of CTSU Sensor Stabilization Wait Control Register (CTSUSST)

Address: F0383H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CTSUSST	CTSUSST7	CTSUSST6	CTSUSST5	CTSUSST4	CTSUSST3	CTSUSST2	CTSUSST1	CTSUSST0

CTSUSST[7:0]	CTSU sensor stabilization wait control
These bits set the stabilization wait time for the TSCAP pin voltage. The value of these bits should be fixed to 00010000B.	

Caution 1. The CTSUSST register should be set when the CTSUSTRT bit in the CTSUCR0 register is 0.

Caution 2. If these bits are not set, the TSCAP voltage becomes unstable at the start of measurement, and the CTSU is unable to obtain correct touch measurement results.

15.3.6 CTSU measurement channel register 0 (CTSUMCH0)

The CTSUMCH0 register is used to set the channel to be measured in self-capacitance single scan mode, and indicate the channel that is being measured or received channel in other modes.

The CTSUMCH0 register can be set by an 8-bit memory manipulation instruction.

Writing 1 to the CTSUINIT bit in the CTSUCR0 register initializes this register.

Reset signal generation sets this register to 1FH.

Figure 15-9. Format of CTSU Measurement Channel Register 0 (CTSUMCH0)

Address: F0384H After reset: 1FH R/W[Note 1]

Symbol	7	6	5	4	3	2	1	0
CTSUMCH0	0	0	0	CTSUMCH04	CTSUMCH03	CTSUMCH02	CTSUMCH01	CTSUMCH00

● In self-capacitance single scan mode

CTSUMCH0[4:0]	CTSU measurement target selection[Note 2]
00000	TS00
00001	TS01
⋮	⋮
01101	TS13
01110	TS14
Other than the above	Setting prohibited

● In other measurement modes

CTSUMCH0[4:0]	Channel that is being measured by CTSU or received channel
00000	TS00
00001	TS01
⋮	⋮
01101	TS13
01110	TS14
11111	Measurement is stopped

Note 1. Writing to these bits is enabled only in self-capacitance single scan mode (the CTSUMD[1:0] bits in the CTSUCR1 register = 00B).

Note 2. Set only enabled channels (00000B to 01110B) when setting channels in self-capacitance single scan mode. In other modes, writing to these bits has no effect.

Caution The CTSUMCH0 register should be set when the CTSUSTRT bit in the CTSUCR0 register is 0.

15.3.7 CTSU measurement channel register 1 (CTSUMCH1)

The CTSUMCH1 register is used to indicate the transmit channel that is being measured in the mutual capacitance full scan mode. The value of the bits in this register is 11111B while measurement is stopped or in self-capacitance single scan mode and multi-scan mode.

The CTSUMCH1 register can be read by an 8-bit memory manipulation instruction.

Writing 1 to the CTSUINIT bit in the CTSUCR0 register initializes this register.

Reset signal generation sets this register to 1FH.

Figure 15-10. Format of CTSU Measurement Channel Register 1 (CTSUMCH1)

Address: F0385H After reset: 1FH R

Symbol	7	6	5	4	3	2	1	0
CTSUMCH1	0	0	0	CTSUMCH14	CTSUMCH13	CTSUMCH12	CTSUMCH11	CTSUMCH10

<R>

CTSUMCH1[4:0]	CTSU transmit channel flag
00000	Transmit channel that is being measured: TS00
00001	Transmit channel that is being measured: TS01
⋮	⋮
01101	Transmit channel that is being measured: TS13
01110	Transmit channel that is being measured: TS14
11111	Measurement is stopped

15.3.8 CTSU channel enable control register 0 (CTSUCHAC0)

The CTSUCHAC0 register is used to enable or disable the TS pins (TS00 to TS07) of the CTSU.

The CTSUCHAC0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 15-11. Format of CTSU Channel Enable Control Register 0 (CTSUCHAC0) (1/2)

Address: F0386H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CTSUCHAC0	CTSUCHAC07	CTSUCHAC06	CTSUCHAC05	CTSUCHAC04	CTSUCHAC03	CTSUCHAC02	CTSUCHAC01	CTSUCHAC00

CTSUCHAC07	Control over enabling or disabling of channel 7 (TS07) of CTSU[Note 2]
0	Not measurement target
1	Measurement target

CTSUCHAC06	Control over enabling or disabling of channel 6 (TS06) of CTSU[Note 2]
0	Not measurement target
1	Measurement target

CTSUCHAC05	Control over enabling or disabling of channel 5 (TS05) of CTSU[Note 2]
0	Not measurement target
1	Measurement target

CTSUCHAC04	Control over enabling or disabling of channel 4 (TS04) of CTSU
0	Not measurement target
1	Measurement target

CTSUCHAC03	Control over enabling or disabling of channel 3 (TS03) of CTSU
0	Not measurement target
1	Measurement target

CTSUCHAC02	Control over enabling or disabling of channel 2 (TS02) of CTSU[Note 1]
0	Not measurement target
1	Measurement target

Figure 15-11. Format of CTSU Channel Enable Control Register 0 (CTSUCHAC0) (2/2)

Address: F0386H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CTSUCHAC0	CTSUCHAC07	CTSUCHAC06	CTSUCHAC05	CTSUCHAC04	CTSUCHAC03	CTSUCHAC02	CTSUCHAC01	CTSUCHAC00

CTSUCHAC01	Control over enabling or disabling of channel 1 (TS01) of CTSU[Note 1]
0	Not measurement target
1	Measurement target

CTSUCHAC00	Control over enabling or disabling of channel 0 (TS00) of CTSU
0	Not measurement target
1	Measurement target

Note 1. 32-pin products only.

Note 2. 16-, 20-, 24-, and 32-pin products only.

Caution The CTSUCHAC0 register should be set when the CTSUSTRT bit in the CTSUCR0 register is 0.

15.3.9 CTSU channel enable control register 1 (CTSUCHAC1)

The CTSUCHAC1 register is used to enable or disable the TS pins (TS08 to TS14) of the CTSU.

The CTSUCHAC1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 15-12. Format of CTSU Channel Enable Control Register 1 (CTSUCHAC1) (1/2)

Address: F0387H After reset: 00H R/W

Symbol	7[Note 1]	6	5	4	3	2	1	0
CTSUCHAC1	0	CTSUCHAC16	CTSUCHAC15	CTSUCHAC14	CTSUCHAC13	CTSUCHAC12	CTSUCHAC11	CTSUCHAC10

CTSUCHAC16	Control over enabling or disabling of channel 14 (TS14) of CTSU[Note 2]
0	Not measurement target
1	Measurement target

CTSUCHAC15	Control over enabling or disabling of channel 13 (TS13) of CTSU[Note 4]
0	Not measurement target
1	Measurement target

CTSUCHAC14	Control over enabling or disabling of channel 12 (TS12) of CTSU[Note 2]
0	Not measurement target
1	Measurement target

CTSUCHAC13	Control over enabling or disabling of channel 11 (TS11) of CTSU[Note 3]
0	Not measurement target
1	Measurement target

CTSUCHAC12	Control over enabling or disabling of channel 10 (TS10) of CTSU[Note 3]
0	Not measurement target
1	Measurement target

CTSUCHAC11	Control over enabling or disabling of channel 9 (TS09) of CTSU[Note 3]
0	Not measurement target
1	Measurement target

Figure 15-12. Format of CTSU Channel Enable Control Register 1 (CTSUCHAC1) (2/2)

Address: F0387H After reset: 00H R/W

Symbol	7[Note 1]	6	5	4	3	2	1	0
CTSUCHAC1	0	CTSUCHAC16	CTSUCHAC15	CTSUCHAC14	CTSUCHAC13	CTSUCHAC12	CTSUCHAC11	CTSUCHAC10

CTSUCHAC10	Control over enabling or disabling of channel 8 (TS08) of CTSU[Note 3]
0	Not measurement target
1	Measurement target

Note 1. Do not set bit 7 to 1.

Note 2. 32-pin products only.

Note 3. 20-, 24-, and 32-pin products only.

Note 4. 16-, 20-, 24-, and 32-pin products only.

Caution The CTSUCHAC1 register should be set when the CTSUSTRT bit in the CTSUCR0 register is 0.

15.3.10 CTSU channel transmit/receive control register 0 (CTSUCHTRC0)

The CTSUCHTRC0 register is used to set transmission or reception for the TS pins (TS00 to TS07) in mutual capacitance full scan mode.

The CTSUCHTRC0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 15-13. Format of CTSU Channel Transmit/Receive Control Register 0 (CTSUCHTRC0) (1/2)

Address: F038BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CTSUCHTRC0	CTSUCHTRC07	CTSUCHTRC06	CTSUCHTRC05	CTSUCHTRC04	CTSUCHTRC03	CTSUCHTRC02	CTSUCHTRC01	CTSUCHTRC00

CTSUCHTRC07	Control over transmission and reception for channel 7 (TS07) of CTSU[Note 1, Note 3]
0	Reception
1	Transmission

CTSUCHTRC06	Control over transmission and reception for channel 6 (TS06) of CTSU[Note 1, Note 3]
0	Reception
1	Transmission

CTSUCHTRC05	Control over transmission and reception for channel 5 (TS05) of CTSU[Note 1, Note 3]
0	Reception
1	Transmission

CTSUCHTRC04	Control over transmission and reception for channel 4 (TS04) of CTSU[Note 1]
0	Reception
1	Transmission

CTSUCHTRC03	Control over transmission and reception for channel 3 (TS03) of CTSU[Note 1]
0	Reception
1	Transmission

CTSUCHTRC02	Control over transmission and reception for channel 2 (TS02) of CTSU[Note 1, Note 2]
0	Reception
1	Transmission

Figure 15-13. Format of CTSU Channel Transmit/Receive Control Register 0 (CTSUCHTRC0) (2/2)

Address: F038BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CTSUCHTRC0	CTSUCHTRC07	CTSUCHTRC06	CTSUCHTRC05	CTSUCHTRC04	CTSUCHTRC03	CTSUCHTRC02	CTSUCHTRC01	CTSUCHTRC00

CTSUCHTRC01	Control over transmission and reception for channel 1 (TS01) of CTSU[Note 1, Note 2]
0	Reception
1	Transmission

CTSUCHTRC00	Control over transmission and reception for channel 0 (TS00) of CTSU[Note 1]
0	Reception
1	Transmission

Note 1. Set this bit to 0 in self capacitance single scan mode and multi-scan mode.

Note 2. 32-pin products only.

Note 3. 16-, 20-, 24- and 32-pin products only.

Caution The CTSUCHTRC0 register should be set when the CTSUSTRT bit in the CTSUCR0 register is 0.

15.3.11 CTSU channel transmit/receive control register 1 (CTSUCHTRC1)

The CTSUCHTRC1 register is used to set reception or transmission for the TS pins (TS08 to TS15) in mutual capacitance full scan mode.

The CTSUCHTRC1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 15-14. Format of CTSU Channel Transmit/Receive Control Register 1 (CTSUCHTRC1) (1/2)

Address: F038CH After reset: 00H R/W

Symbol	7[Note 1]	6	5	4	3	2	1	0
CTSUCHTRC1	0	CTSUCHTRC16	CTSUCHTRC15	CTSUCHTRC14	CTSUCHTRC13	CTSUCHTRC12	CTSUCHTRC11	CTSUCHTRC10

CTSUCHTRC16	Control over transmission and reception for channel 14 (TS14) of CTSU[Note 2, Note 3]
0	Reception
1	Transmission

CTSUCHTRC15	Control over transmission and reception for channel 13 (TS13) of CTSU[Note 2, Note 5]
0	Reception
1	Transmission

CTSUCHTRC14	Control over transmission and reception for channel 12 (TS12) of CTSU[Note 2, Note 3]
0	Reception
1	Transmission

CTSUCHTRC13	Control over transmission and reception for channel 11 (TS11) of CTSU[Note 2, Note 4]
0	Reception
1	Transmission

CTSUCHTRC12	Control over transmission and reception for channel 10 (TS10) of CTSU[Note 2, Note 4]
0	Reception
1	Transmission

CTSUCHTRC11	Control over transmission and reception for channel 9 (TS09) of CTSU[Note 2, Note 4]
0	Reception
1	Transmission

Figure 15-14. Format of CTSU Channel Transmit/Receive Control Register 1 (CTSUCHTRC1) (2/2)

Address: F038CH After reset: 00H R/W

Symbol	7 Note 1	6	5	4	3	2	1	0
CTSUCHTRC1	0	CTSUCHTRC16	CTSUCHTRC15	CTSUCHTRC14	CTSUCHTRC13	CTSUCHTRC12	CTSUCHTRC11	CTSUCHTRC10

CTSUCHTRC10	Control over transmission and reception for channel 8 (TS08) of CTSU Note 2, Note 4
0	Reception
1	Transmission

Note 1. Do not set bit 7 to 1.

Note 2. Set this bit to 0 in self capacitance single scan mode and multi-scan mode.

Note 3. 32-pin products only.

Note 4. 20-, 24-, and 32-pin products only.

Note 5. 16-, 20-, 24- and 32-pin products only.

Caution The CTSUCHTRC1 register should be set when the CTSUSTRT bit in the CTSUCR0 register is 0.

15.3.12 CTSU high-pass noise reduction control register (CTSUDCLKC)

The CTSUDCLKC register is used to set the mode of the spectrum diffusion clock for high-pass noise reduction and control the amount of diffusion.

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

Reset signal generation clears this register to 00H.

Figure 15-15. Format of CTSU High-Pass Noise Reduction Control Register (CTSUDCLKC)

Address: F0390H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CTSUDCLKC	0	0	CTSUSSCNT1	CTSUSSCNT0	0	0	CTSUSSMOD1	CTSUSSMOD0

CTSUSSCNT1	CTSUSSCNT0	CTSU diffusion clock control
1	1	Using the high-pass noise reduction function[Note 1]
Other than the above		Setting prohibited

CTSUSSMOD1	CTSUSSMOD0	CTSU diffusion clock mode selection
0	0	Using the high-pass noise reduction function[Note 2]
Other than the above		Setting prohibited

Note 1. If these bits are not set, touch measurement may not be correctly performed.

Note 2. If these bits are not set, the effect of high-pass noise reduction cannot be correctly obtained.

Caution The CTSUDCLKC register should be set when the CTSUSTRT bit in the CTSUCR0 register is 0.

15.3.13 CTSU status register (CTSUST)

The CTSUST register is used to indicate the current measurement status, whether the measurement result stored in the counter has been read, whether the counter has overflowed, and the mutual capacitance measurement status.

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

Writing 1 to the CTSUINIT bit in the CTSUCR0 register initializes this register.

Reset signal generation clears this register to 00H.

Figure 15-16. Format of CTSU Status Register (CTSUST)

Address: F0391H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
CTSUST	CTSUPS	0	CTSUSOVF	CTSUDTSR	0	CTSUSTC2	CTSUSTC1	CTSUSTC0

CTSUPS	CTSU mutual capacitance measurement status flag[Note 1, Note 2]
0	First measurement
1	Second measurement
This flag indicates whether the measurement is the first or second of two measurements for each channel in mutual capacitance full scan mode (the CTSUMD[1:0] bits in the CTSUCR1 register = 11B).	

CTSUSOVF	CTSU sensor counter overflow flag[Note 3, Note 4]
0	No overflow
1	An overflow
This flag indicates whether the sensor counter has overflowed. FFFFh can be read as the measurement result (CTSUSC counter) when an overflow has occurred.	

CTSUDTSR	CTSU data transfer status flag[Note 2]
0	Measurement result has been read
1	Measurement result has not been read
This flag indicates whether the measurement result stored in the sensor counter and the reference counter has been read. This flag is set to 1 when measurement is completed; 0 when the reference counter is read by software.	

CTSUSTC2	CTSUSTC1	CTSUSTC0	CTSU measurement status counter[Note 2]
0	0	0	Status0
0	0	1	Status1
0	1	0	Status2
0	1	1	Status3
1	0	0	Status4
1	0	1	Status5

Note 1. This flag indicates 0 while measurement is stopped or in other measurement modes.

Note 2. Read-only bit

Note 3. Even if an overflow occurs, measurement processing continues until the set period. No interrupt is generated even when an overflow occurs. To determine the channel on which the overflow has occurred, read the measurement result of each channel after measurement is completed (after a measurement end interrupt is generated).

Note 4. This flag is cleared when 0 is written after 1 is read by software.

15.3.14 CTSU high-pass noise reduction spectrum diffusion control register (CTSUSSC)

The CTSUSSC register is used to specify the spectrum diffusion frequency division setting according to the base clock frequency division setting.

The CTSUSSC register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 15-17. Format of CTSU High-Pass Noise Reduction Spectrum Diffusion Control Register (CTSUSSC)

Address: F0392H, F0393H After reset: 0000H R/W

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

CTSUSSDIV[3:0]	CTSU spectrum diffusion frequency division setting
These bits specify the spectrum diffusion frequency division setting. See the relationship between base clock frequencies and CTSUSSDIV[3:0] bits settings in **Table 15-3**, for setting the value of these bits.	

Table 15-3. Relationship between Base Clock Frequencies and CTSUSSDIV[3:0] Bits Settings

Base Clock Frequency fb (MHz)	CTSUSSDIV[3:0] Bits Setting
4.00 ≤ fb	0000
2.00 ≤ fb < 4.00	0001
1.33 ≤ fb < 2.00	0010
1.00 ≤ fb < 1.33	0011
0.80 ≤ fb < 1.00	0100
0.67 ≤ fb < 0.80	0101
0.57 ≤ fb < 0.67	0110
0.50 ≤ fb < 0.57	0111
0.44 ≤ fb < 0.50	1000
0.40 ≤ fb < 0.44	1001
0.36 ≤ fb < 0.40	1010
0.33 ≤ fb < 0.36	1011
0.31 ≤ fb < 0.33	1100
0.29 ≤ fb < 0.31	1101
0.27 ≤ fb < 0.29	1110
fb < 0.27	1111

15.3.15 CTSU sensor offset register 0 (CTSUSO0)

The CTSUSO0 register is used to adjust the offset of the sensor and set the measurement count for the CTSU.

The CTSUSO0 register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 15-18. Format of CTSU Sensor Offset Register 0 (CTSUSO0)

Address: F0394H, F0395H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTSUSO0	CTSUSNUM[5:0]						CTSUSO[9:0]									

CTSUSNUM[5:0]	CTSU measurement count setting[Note 1]
These bits set how many times the number of measurement pulses specified by the CTSUPRRATIO[3:0] bits and CTSUPRMODE[1:0] bits of the CTSUSDPRS register is repeated in the measurement time. The number of measurement pulses is repeated (CTSUSNUM[5:0] bits + 1) times.	

CTSUSO[9:0]	CTSU sensor offset adjustment[Note 1]
0000000000	Current offset amount is 0
0000000001	Current offset amount is 1
0000000010	Current offset amount is 2
⋮	⋮
1111111110	Current offset amount is 1022
1111111111	Current offset amount is maximum
These control bits adjust the input current offset of the sensor ICO. These bits are used to offset the sensor ICO input current generated from electrostatic capacitance while the electrode is not being touched during touch measurement, thus preventing overflow of the CTSU sensor counter.	

Note 1. Make settings for the TS pin that is to be measured next after an INTCTSUWR interrupt is generated.

15.3.16 CTSU sensor offset register 1 (CTSUSO1)

The CTSUSO1 register is used to adjust the gain and select the base clock frequency.

The CTSUSO1 register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 15-19. Format of CTSU Sensor Offset Register 1 (CTSUSO1) (1/2)

Address: F0396H, F0397H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTSUSO1	0	CTSUICOG [1:0]		CTSUSDPA [4:0]					0							

CTSUICOG1	CTSUICOG0	CTSU ICO gain adjustment
0	0	100% gain
0	1	66% gain
1	0	50% gain
1	1	40% gain

These bits adjust the output frequency gain of the sensor ICO. Normally, the value of these bits should be set to 00B for the maximum gain.

If changes in the capacitance between when the electrode is touched and when it is not touched greatly exceed the dynamic range of the sensor ICO, set the gain adjustment bits to adjust the gain appropriately.

Figure 15-19. Format of CTSU Sensor Offset Register 1 (CTSUSO1) (2/2)

Address: F0396H, F0397H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTSUSO1	0	CTSUICOG [1:0]		CTSUSDPA [4:0]					0							

CTSUSDPA[4:0]	CTSU Base Clock Setting
00000	Operating clock divided by 2Note 1
00001	Operating clock divided by 4
00010	Operating clock divided by 6
00011	Operating clock divided b 8
00100	Operating clock divided by 10
00101	Operating clock divided by 12
00110	Operating clock divided by 14
00111	Operating clock divided by 16
01000	Operating clock divided by 18
01001	Operating clock divided by 20
01010	Operating clock divided by 22
01011	Operating clock divided by 24
01100	Operating clock divided by 26
01101	Operating clock divided by 28
01110	Operating clock divided by 30
01111	Operating clock divided by 32
10000	Operating clock divided by 34
10001	Operating clock divided by 36
10010	Operating clock divided by 38
10011	Operating clock divided by 40
10100	Operating clock divided by 42
10101	Operating clock divided by 44
10110	Operating clock divided by 46
10111	Operating clock divided by 48
11000	Operating clock divided by 50
11001	Operating clock divided by 52
11010	Operating clock divided by 54
11011	Operating clock divided by 56
11100	Operating clock divided by 58
11101	Operating clock divided by 60
11110	Operating clock divided by 62
11111	Operating clock divided by 64
These bits are used to generate a base clock used as the source for the sensor drive pulse by dividing the operating clock.	

Note 1. The CTSUSDPA[4:0] bits should not be set to 00000B while the high-pass noise reduction function is turned off (the CTSUSOFF bit in the CTSUSDPRS register = 1) in mutual capacitance full scan mode (the CTSUMD[1:0] bits in the CTSUCR1 register = 11B).

Caution Write first to the CTSUSSC register, then the CTSUSO0 register, and then the CTSUSO1 register after an INTCTSUWR interrupt is generated. Write operation to the CTSUSO1 register causes a transition to status 3. Thus, set all the bits in a single setting when writing to the CTSUSO1 register.

15.3.17 CTSU sensor counter (CTSUSC)

The CTSUSC register is a read-only register configured as an up-counter that counts cycles of the sensor ICO clock.

The CTSUSC register can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 15-20. Format of CTSU Sensor Counter (CTSUSC)

Address: F0398H, F0399H After reset: 0000H R

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

CTSUSC[15:0]	CTSU sensor counter
Read the CTSUSC counter after an INTCTSURD interrupt is generated. These bits indicate FFFFH when an overflow occurs. These bits are cleared immediately before the CTSU measurement status counter value changes to status 4 (the CTSUSTC[2:0] flags in the CTSUST register change to 100B) in the next measurement. These bits are also cleared by setting the CTSUINIT bit of the CTSUCR0 register.	

15.3.18 CTSU reference counter (CTSURC)

The CTSURC register is a read-only register used for reading the measurement results.

The CTSURC register can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 15-21. Format of CTSU Reference Counter (CTSURC)

Address: F039AH, F039BH After reset: 0000H R

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

CTSURC[15:0]	CTSU reference counter
Read first from the CTSUSC counter and then the CTSURC counter after an INTCTSURD interrupt is generated. These bits indicate FFFFH when an overflow occurs. Even when the stabilization time specified for status 3 has elapsed, if the CTSURC counter is not read, status 3 continues until the counter is read. After these bits are read, they are cleared immediately before the CTSU measurement status counter value changes to status 4 (the CTSUSTC[2:0] flags in the CTSUST register change to 100B) in the next measurement. These bits are also cleared using the CTSUINIT bit of the CTSUCR0 register.	

15.3.19 CTSU error status register (CTSUERRS)

The CTSUERRS register is used to monitor the abnormality of the TSCAP voltage.

The CTSUERRS register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 15-22. Format of CTSU Error Status Register (CTSUERRS) (1/3)

Address: F039CH, F039DH After reset: 0000H R/W[Note 1]

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTSUERRS	CTSUICOMP	0	CTSUALMES	0	0	CTSUR03EN	CTSUR02EN	CTSUR01EN	CTSUTSOC	CTSUCLKSEL1	0	0	CTSUDRV	CTSUTSOD	CTSUSPMD[1:0]	

CTSUICOMP	TSCAP voltage error monitoring
0	Normal TSCAP voltage
1	Abnormal TSCAP voltage

<R>

This bit indicates the abnormal state of the TSCAP voltage. It monitors the TSCAP voltage and is set to 1 if the voltage becomes abnormal.

If the offset current amount set by the CTSUSO0 register exceeds the sensor ICO input current during touch measurement, the TSCAP voltage becomes abnormal and touch measurement cannot be correctly performed.

If the TSCAP voltage becomes abnormal, the sensor ICO counter value will be undefined.

This bit is cleared by writing 0 to the CTSUPON bit of the CTSUCR1 register and turning off the power supply.

CTSUALMES	Selection of multiple electrode connection (MEC) function
0	Capacitance measurement mode
1	Multiple electrode connection (MEC) function

This bit is used to select the multiple electrode connection (MEC) function. When measuring electrostatic capacitance on each TSm pin, set this bit to 0.

When the CTSUALMES bit is set to 1 (selecting multiple electrode connection), channels (TSm pins) set by the CTSUCHACxx bit are the target channels for simultaneous measurement.

Make settings such that at least one of the channels (TSm pins) set by the CTSUCHACxx bit matches the channel set by the CTSUMCH0 register.

Multiple electrode connection is only available when the CTSUMD1 bit = 0 (self-capacitance scan mode).

Do not use this function while the CTSUMD1 bit = 1 (mutual capacitance scan mode).

CTSUR03EN	CTSU self-test load resistance R03 setting
0	Capacitance measurement mode
1	Self-test load resistance 60 kΩ is set

This bit is used to conduct CTSU self-test.

It sets self-test load resistance R03.

Figure 15-22. Format of CTSU Error Status Register (CTSUERRS) (2/3)

Address: F039CH, F039DH After reset: 0000H R/W[Note 1]

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTSUERRS	CTSUICOMP	0	CTSUALMES	0	0	CTSUR03EN	CTSUR02EN	CTSUR01EN	CTSUTSOC	CTSUCLKSEL1	0	0	CTSUDRV	CTSUTSOD	CTSUSPMD[1:0]	

CTSUR02EN	CTSU self-test load resistance R02 setting
0	Capacitance measurement mode
1	Self-test load resistance 30 kΩ is set
This bit is used to conduct CTSU self-test. It sets self-test load resistance R02.	

CTSUR01EN	CTSU self-test load resistance R01 setting
0	Capacitance measurement mode
1	Self-test load resistance 15 kΩ is set
This bit is used to conduct CTSU self-test. It sets self-test load resistance R01.	

CTSUTSOC	Calibration setting bit 2
0	Capacitance measurement mode
1	Calibration setting 2
The CTSUTSOC bit is used to calibrate the CTSU. When measuring electrostatic capacitance, set this bit to 0.	

CTSUCLKSEL1	Calibration setting bit 3
0	Capacitance measurement mode
1	Calibration setting 3
The CTSUCLKSEL1 bit is used to calibrate the CTSU. When measuring electrostatic capacitance, set this bit to 0.	

CTSUDRV	Calibration setting bit 1
0	Capacitance measurement mode
1	Calibration setting 1
The CTSUDRV bit is used to calibrate the CTSU. Set this bit to 0 to save power except when calibrating the CTSU.	

CTSUTSOD	TS pin fixed output bit
0	Capacitance measurement mode
1	TS pin fixed output (high/low-level output)
The CTSUTSOD bit is used to calibrate the CTSU. When measuring electrostatic capacitance, set this bit to 0. When the CTSUTSOD bit is set to 1, the level specified by the CTSUIOC bit in the CTSUCR0 register is output from the TSm pin.	

Figure 15-22. Format of CTSU Error Status Register (CTSUERRS) (3/3)

Address: F039CH, F039DH After reset: 0000H R/W[Note 1]

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTSUERRS	CTSUICOMP	0	CTSUALMES	0	0	CTSUR03EN	CTSUR02EN	CTSUR01EN	CTSUTSOC	CTSUCLKSEL1	0	0	CTSUDRV	CTSUTSOD	CTSUSPMD[1:0]	

CTSUSPMD[1:0]		Calibration mode bit
0	0	Capacitance measurement mode
0	1	Setting prohibited
1	0	Calibration mode
1	1	Setting prohibited
The CTSUSPMD[1:0] bits are used to calibrate the CTSU. When measuring electrostatic capacitance, set these bits to 00B.		

Note 1. Bit 15 (CTSUICOMP) is read-only.

15.3.20 Touch pin function select registers 0, 1 (TSSEL0, TSSEL1)

These registers select whether the touch pin function or another multiplexed function is used with the P01, P03 to P07, P16, P17, P20 to P23, and P41 to P43 pins.

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

Reset signal generation clears these registers to 00H.

Figure 15-23. Format of Touch Pin Function Select Registers 0, 1 (TSSEL0, TSSEL1)

Address: F030AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TSSEL0	TSSEL07	TSSEL06	TSSEL05	TSSEL04	TSSEL03	TSSEL02	TSSEL01	TSSEL00

Address: F030BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TSSEL1	0	TSSEL14	TSSEL13	TSSEL12	TSSEL11	TSSEL10	TSSEL09	TSSEL08

TSSELxx (xx = 0 to 14)	Selection of a function other than the touch pin function (multiplexed function) or the touch pin function for the Pmn pin (m = 0, 1, 2, 4; n = 0 to 7)
0	Use the Pmn pin for a function other than the touch pin function (multiplexed function).
1	Use the Pmn pin for the touch pin function.

Remark To use the Pmn pins as touch pins (TSSELxx = 1), be sure to set the PUmn bit of the PUm register, POMmn bit of the POMm register, and PIMmn bit of the PIMm register to “0”.

- 10-pin products: Only TSSEL00, 03, and 04 are valid.
- 16-pin products: Only TSSEL00, 03, 04, 05, 06, 07, and 13 are valid.
- 20- and 24-pin products: TSSEL00, 03, 04, 05, 06, 07, 08, 09, 10, 11, and 13 are valid.

15.3.21 TSCAP pin setting register (VTSEL)

When the touch pin function is in use (when the TSSELxx bit is set to 1), the setting of the VTSEL register is effective. This register disables or enables input to the P02 pin.

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

Reset signal generation clears this register to 00H.

Figure 15-24. Format of TSCAP Pin Setting Register (VTSEL)

Address: F030DH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
VTSEL	0	0	0	0	0	0	0	VTSEL0

VTSEL0	Disabling or enabling of input to the P02 pin
0	When the touch pin function is in use, input to the P02 pin is disabled.
1	When the touch pin function is in use, input to the P02 pin is enabled.

15.3.22 CTSU trimming register (RTRIM)

The RTRIM register is written with the reference resistance setting that was adjusted under specific conditions for each chip at the time of shipment.

When writing to this register, only do so after setting the CTSUSPMD[1:0] bits of the CTSUERRS register to 10B (calibration mode). When this register is reset, it returns to the value written at the time of shipment.

Do not write to this register while the value of the CTSUSPMD[1:0] bits is 00B (capacitance measurement mode).

Figure 15-25. Format of CTSU Trimming Register (RTRIM)

Address: F0730H After reset: Specific value set for each chip at the time of shipment

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

15.3.23 CTSU trimming result register (CTSUTRESULT)

This register holds the variation coefficient of the reference load resistance. It is written with the value set at the time of shipment. Do not write to this register.

Figure 15-26. Format of CTSU Trimming Result Register (CTSUTRESULT)

Address: F0731H After reset: Initial value set at the time of shipment

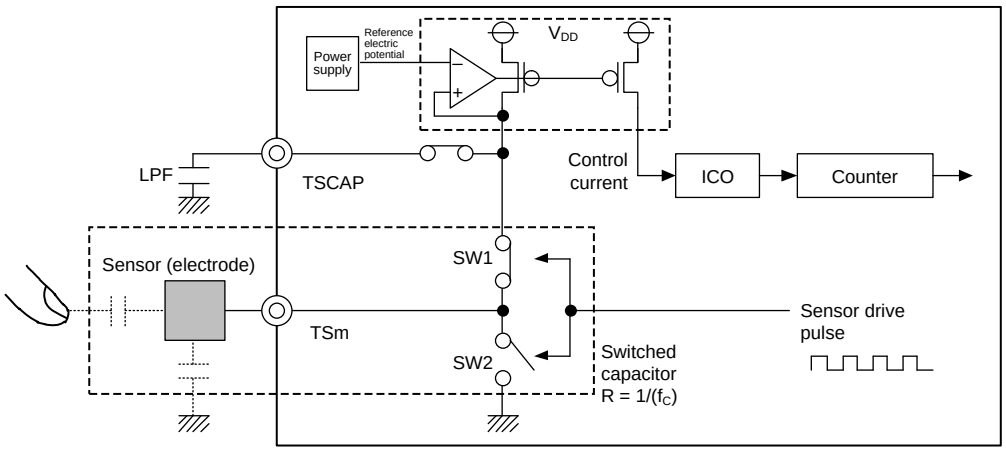
Symbol	7	6	5	4	3	2	1	0
CTSUTRESULT	CTSUTRESULT[7:0]							

15.4 Operation

<R> 15.4.1 Principles of measurement operation

Figure 15-27 shows the measurement circuit.

Figure 15-27. Measurement Circuit



Remark m = 00 to 14

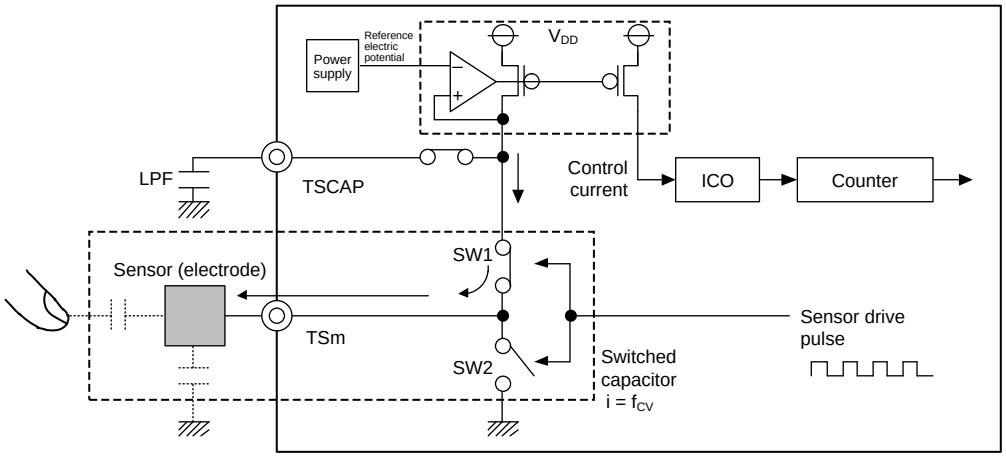
The electrostatic capacitance measurement operation principles of the CTSU current frequency conversion method are explained using **Figure 15-28** to **Figure 15-30**.

(1) The electrostatic capacitance of the electrode is charged by turning SW1 on and SW2 off (**Figure 15-28**).

(2) The charged capacitance is discharged by turning SW1 off and SW2 on (**Figure 15-29**).

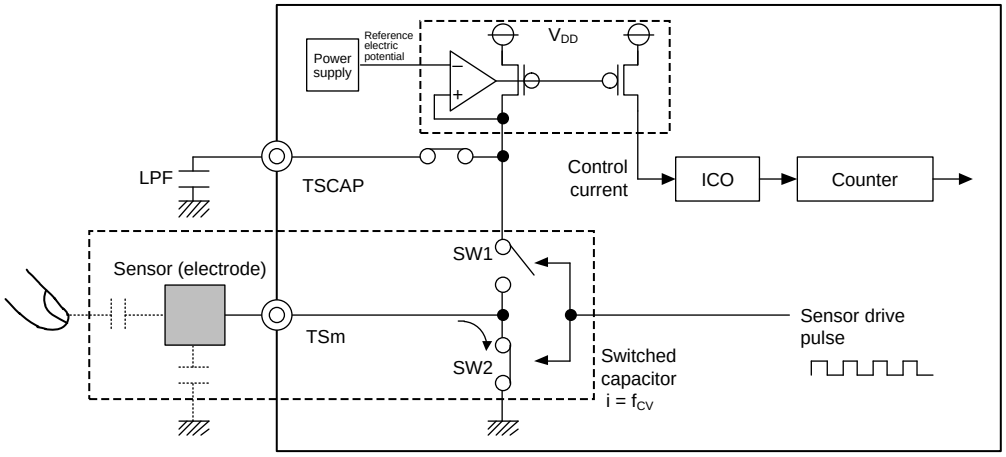
Current flows to the switched capacitor by switching between charging and discharging in steps (1) and (2). At this time, the value of electrostatic capacitance varies depending on whether a finger is in close proximity, so the flowing current changes. A clock is generated by supplying the control current, which is proportional to the amount of the current flowing through the switched capacitor, from the circuit that generates the TSCAP power supply to the current controlled oscillator (ICO). The counter is used to measure the clock frequency which changes depending on whether a finger is in close proximity, and the value read from the counter is used by software to determine contact with a finger (**Figure 15-30**).

Figure 15-28. Charging Operation



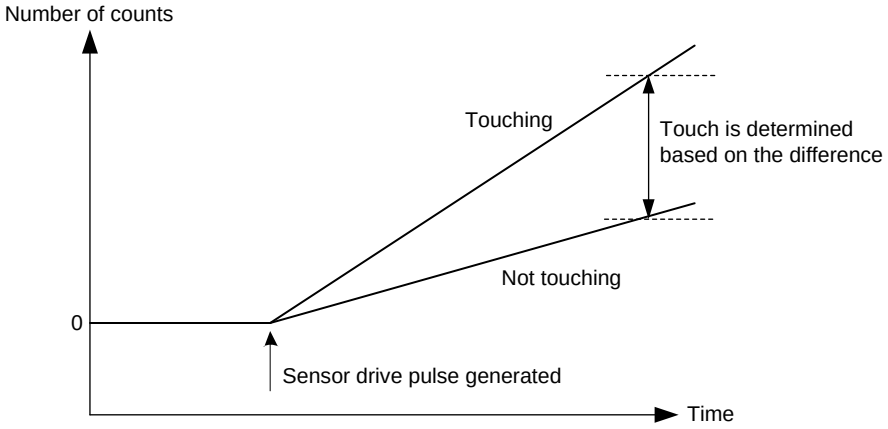
Remark m = 00 to 14

Figure 15-29. Discharging Operation



Remark m = 00 to 14

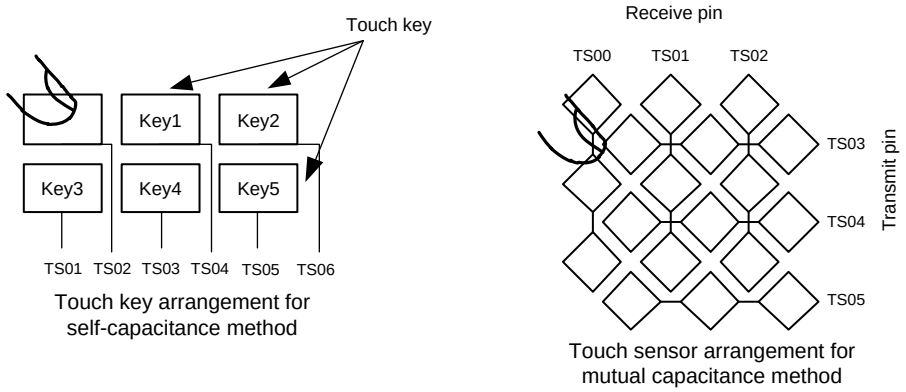
Figure 15-30. Change in Measured Value When Finger is Touching and Not Touching



15.4.2 Measurement modes

The CTSU supports self-capacitance and mutual capacitance methods. **Figure 15-31** illustrates these methods.

<R> Figure 15-31. Overview of Self-Capacitance Method and Mutual Capacitance Method



In the self-capacitance method, a single touch pin is allocated to a single touch key to measure individual electrostatic capacitance when a finger is in close proximity. In this method, single scan and multi-scan can be used as measurement modes.

In the mutual capacitance method, the capacitance between two opposite electrodes (transmit and receive pins) is measured.

(1) Initial setting flowchart

Figure 15-32 shows the flowchart for CTSU initial setting.

<R> Figure 15-32. CTSU Initial Setting Flowchart

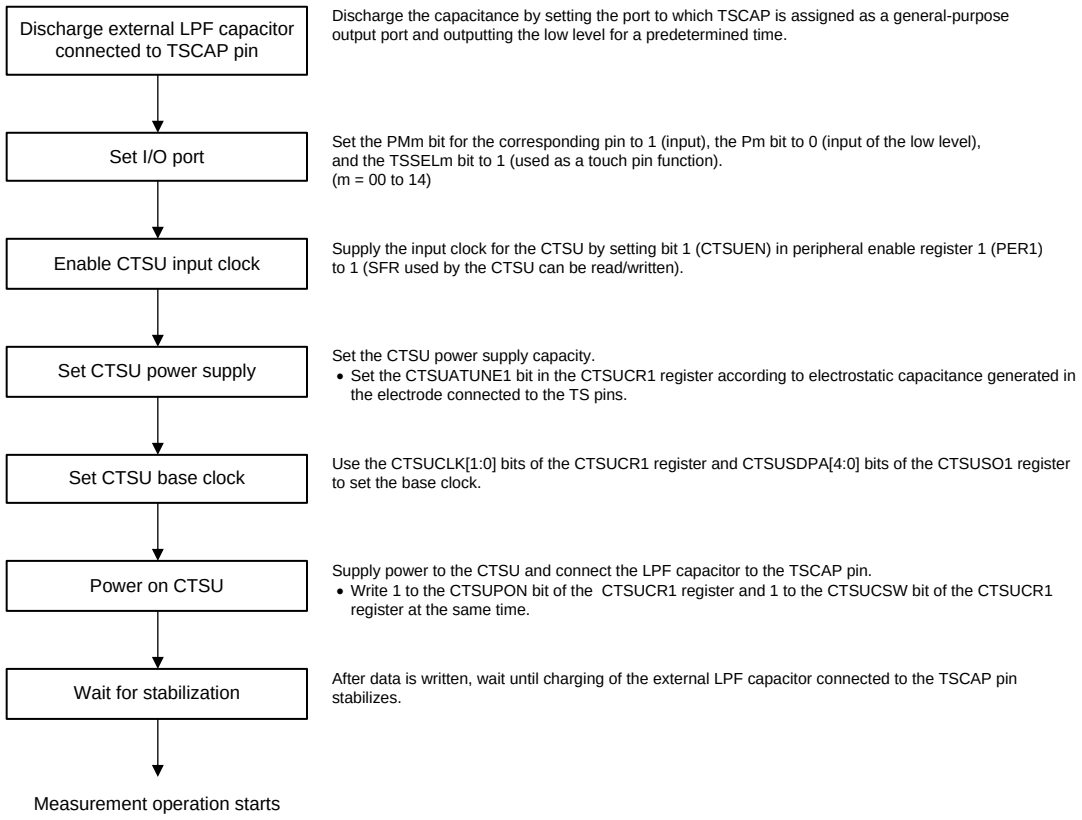
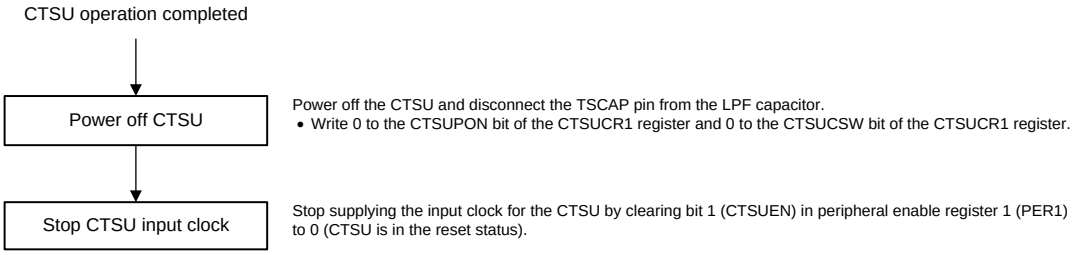


Figure 15-33 shows the flowchart for stopping operation of the CTSU and placing it in the reset state.

Figure 15-33. CTSU Stopping Flowchart



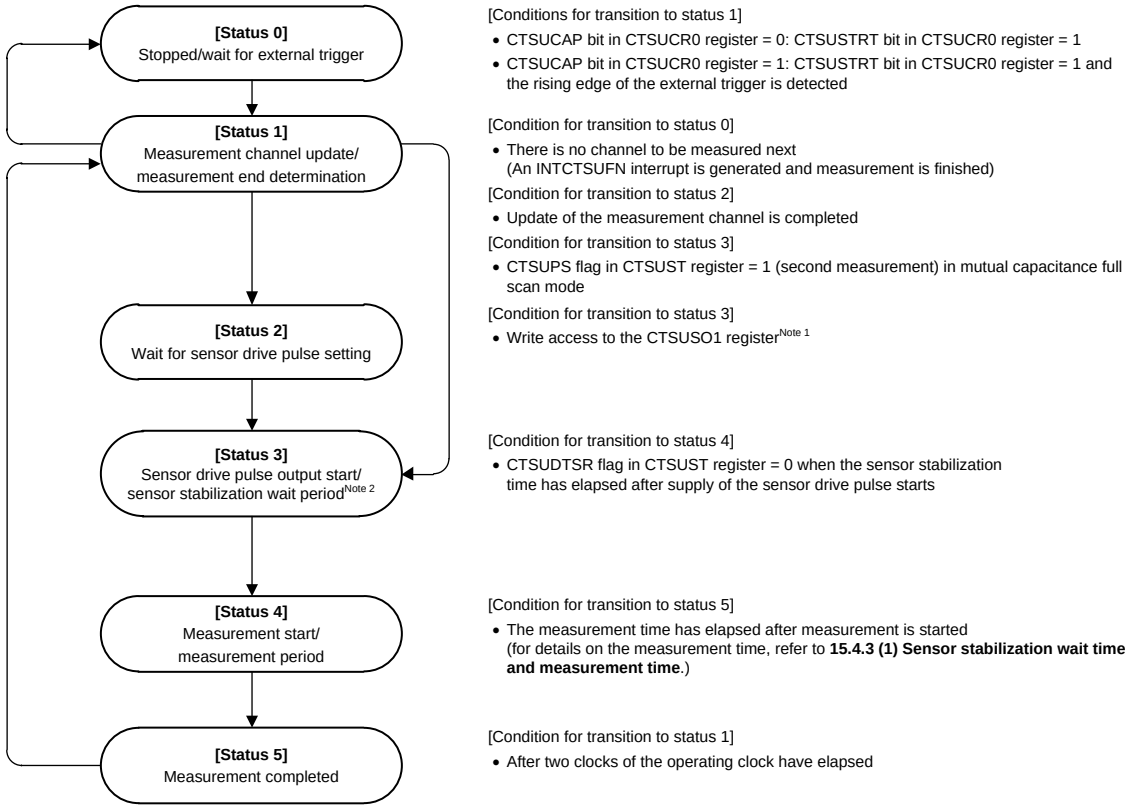
When restarting operation after it has been stopped, follow the initial setting flowchart in **Figure 15-32**.

(2) Status counter

The measurement status counter of the CTSU status register (CTSUST) indicates the current measurement status.

The measurement status is common to all three modes. **Figure 15-34** shows status operation transitions.

Figure 15-34. Status Operation Transitions



Note 1. When using the ICU to set the registers in the INTCTSUWR interrupt handling, write to the CTSUSO1 register last.

Note 2. If the CTSUDTSR flag in the CTSUST register is 1, wait until the previous measurement result is transferred.

The status of the status counter transitions to status 0 when all of the specified measurement channels are measured.

The CTSUSTRT bit in the CTSUCR0 register is cleared to 0 by hardware when a software trigger is used.

When an external trigger is used, the value 1 is retained, and the CTSU waits for the next trigger.

When operation is forcibly stopped (by writing 0 to the CTSUSTRT bit of the CTSUCR0 register and 1 to the CTSUINIT bit of the CTSUCR0 register at the same time) during measurement or the wait state for the trigger, the status transitions to status 0 and measurement is stopped forcibly.

If there is no channel to be measured by setting the CTSUMCH0, CTSUCHACn (n = 0, 1), and CTSUCHTRCn (n = 0, 1) registers, an INTCTSUFN interrupt is generated immediately after a transition to status 1, and then the status transitions to status 0.

The following are the cases when there is no channel to be measured.

- A measurement target channel is not specified by the CTSUCHACn register.
- In self-capacitance single scan mode, the channel specified in the CTSUMCH0 register is not a measurement target in the CTSUCHACn register.
- In mutual capacitance full scan mode, there is no transmit channel or receive channel which is specified by the CTSUCHACn and CTSUCHTRCn registers.

(3) Self-capacitance single scan mode operation

In self-capacitance single scan mode, electrostatic capacitance on a channel is measured. **Figure 15-35** shows the software flowchart and an operation example, and **Figure 15-36** shows the timing chart.

<R> Figure 15-35. Software Flowchart and Operation Example of Self-Capacitance Single Scan Mode

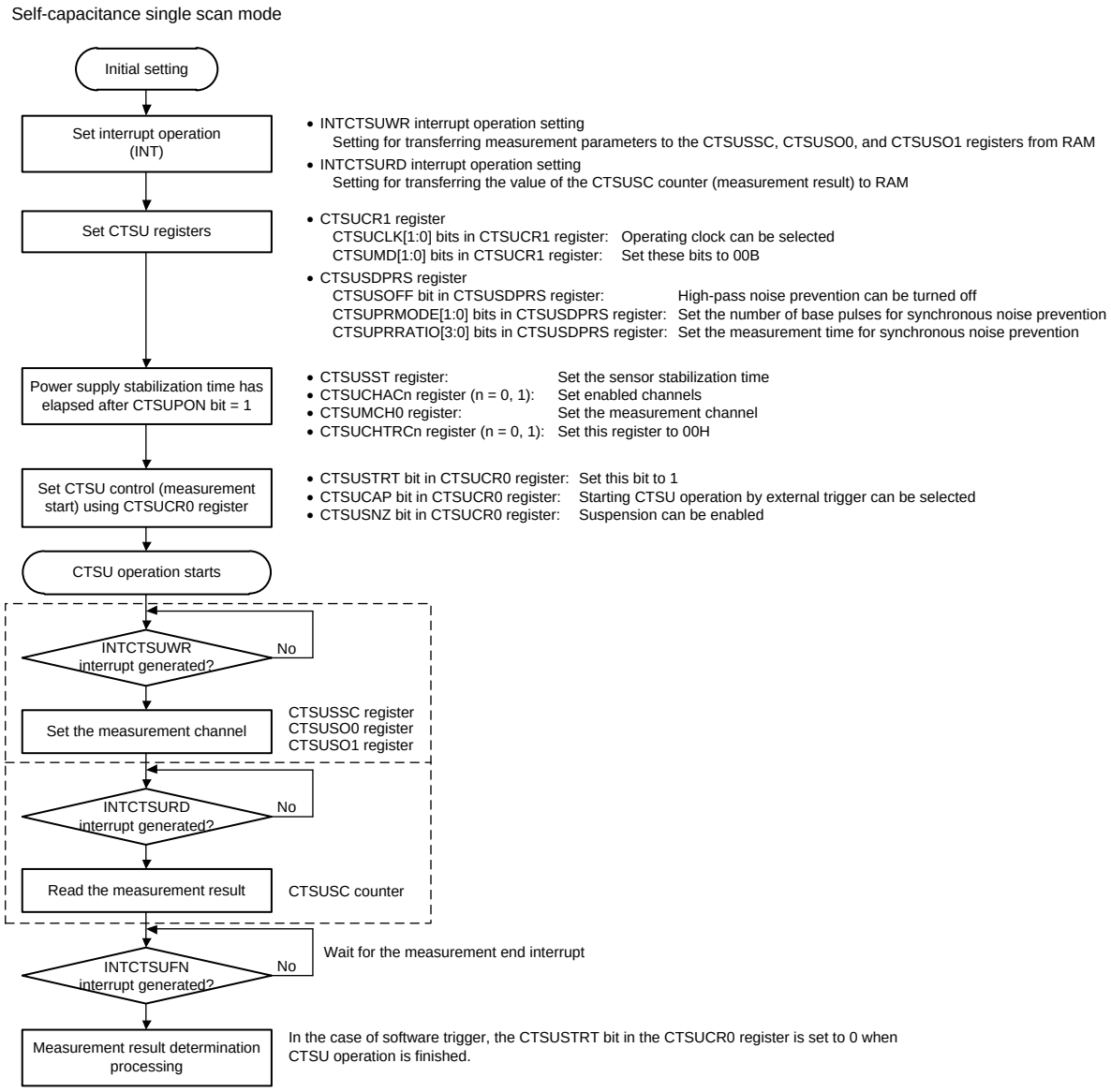
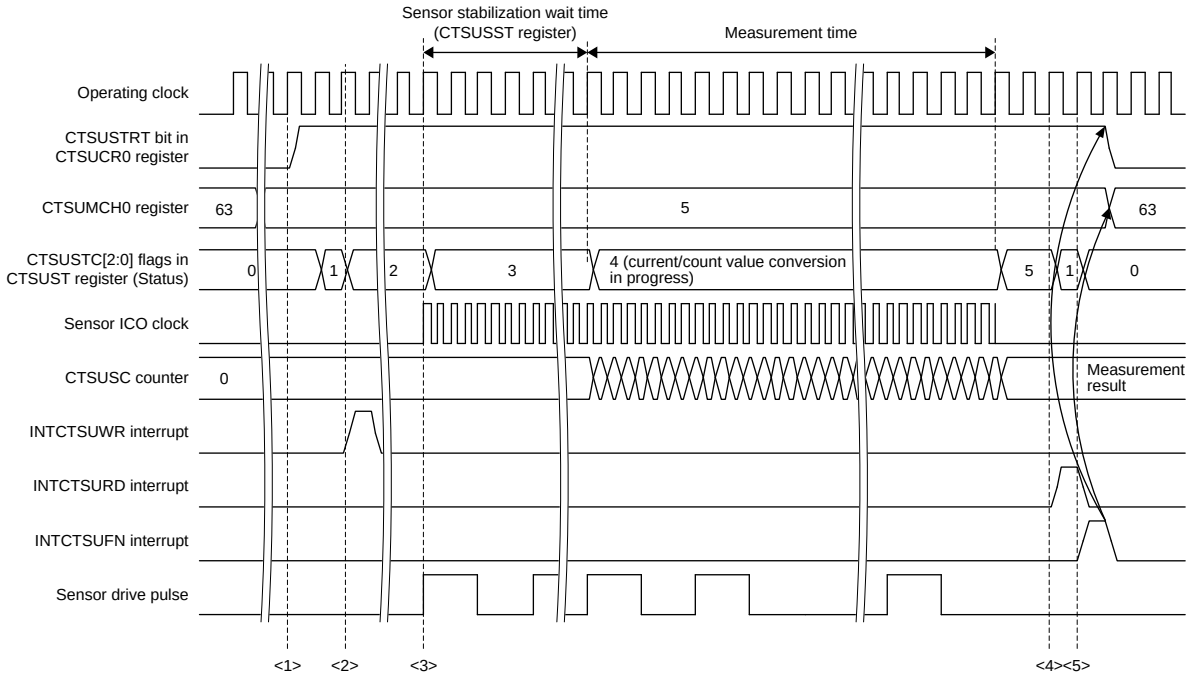


Figure 15-36. Timing Chart of Self-Capacitance Single Scan Mode (Measurement Start Condition is Software Trigger)



The following describes operation shown in the timing chart in **Figure 15-36**.

<1> After various settings are made, operation is started by writing 1 to the CTSUSTRT bit of the CTSUCR0 register.

<2> After a channel to be measured is determined according to the preset conditions, a request for setting the corresponding channel (INTCTSUWR) is output.

<3> Upon completion of setting the measurement channel (writing to CTSUSSC, CTSUSO0, and CTSUSO1 registers), the sensor drive pulse is output and the sensor ICO clock operates.

<4> After the sensor stabilization wait time and the measurement time have elapsed and measurement is finished, a measurement result read request (INTCTSURD) is output.

<5> A measurement end interrupt (INTCTSUFN) is output and measurement is finished (transition to status 0).

Table 15-4 lists the touch pin states in self-capacitance single scan mode.

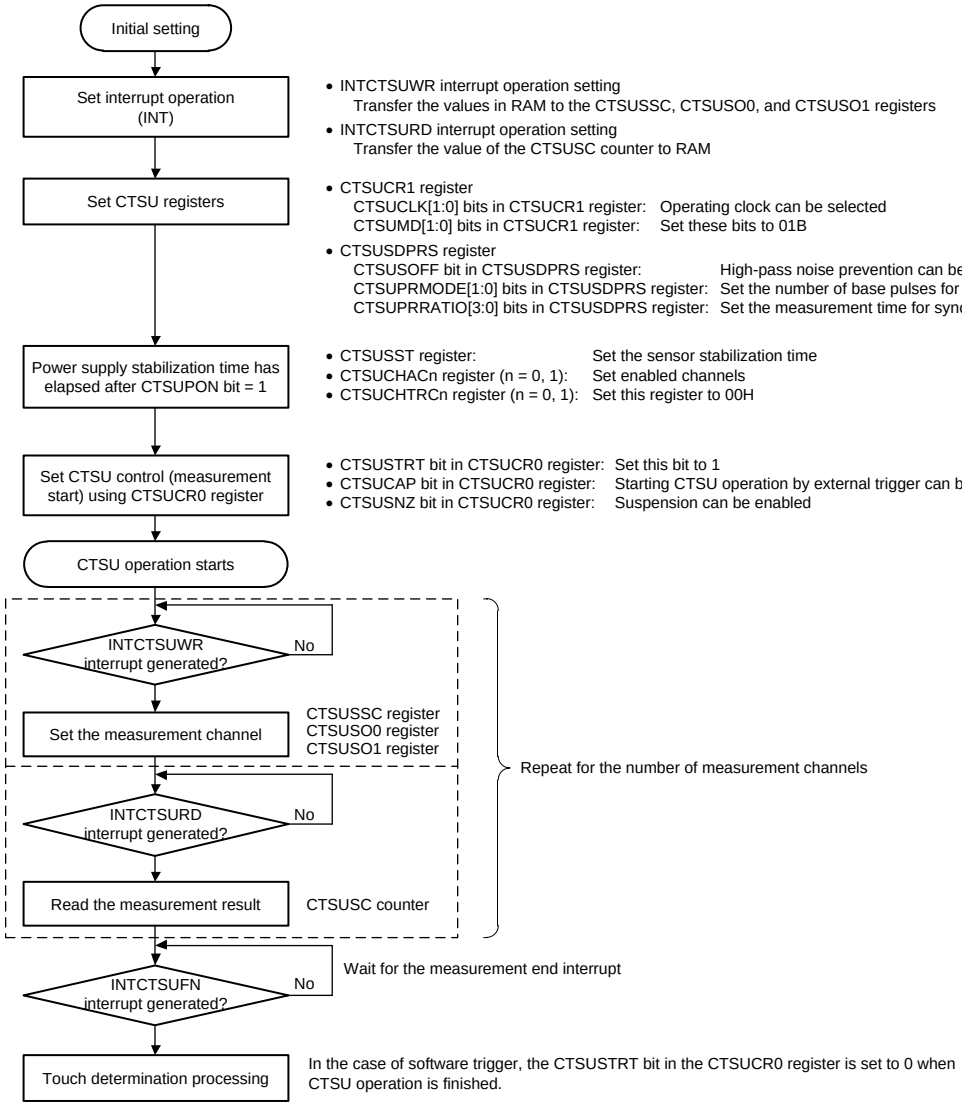
Table 15-4. Touch Pin States in Self-Capacitance Single Scan Mode

Status	Touch Pin	
	Measurement Channel	Non-measurement Channel
0	Low	Low
1	Low	Low
2	Low	Low
3	Pulse	Low
4	Pulse	Low
5	Low	Low

(4) Self-capacitance multi-scan mode operation

In self-capacitance multi-scan mode, electrostatic capacitance is measured sequentially on all channels specified as measurement targets by the CTSUCHACn registers (n = 0 to 2) in ascending order. **Figure 15-37** shows the software flowchart and an operation example, and **Figure 15-38** shows the timing chart.

Figure 15-37. Software Flow and Operation Example of Self-Capacitance Multi-Scan Mode



Channel measurement sequence in self-capacitance multi-scan mode

<Setting>

- Select self-capacitance multi-scan mode (CTSUMD[1:0] bits in CTSUCR1 register = 01B)
- Set channels 0, 3, 5, and 6 as enabled channels (CTSUCHACn[7:0] bits in CTSUCHACn register (n = 0) = 01101001B)

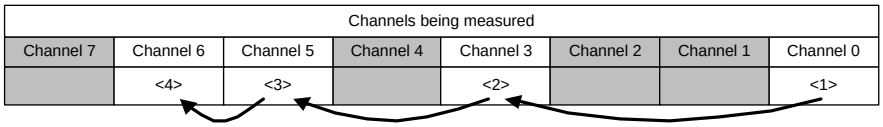
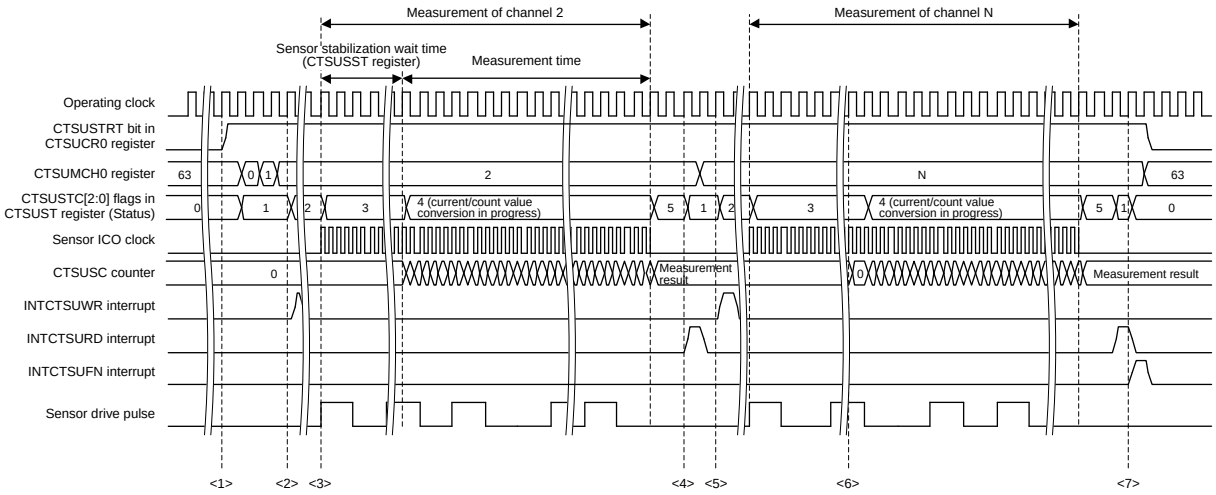


Figure 15-38. Timing Chart of Self-Capacitance Multi-Scan Mode (Measurement Start Condition is Software Trigger)



The following describes operation shown in the timing chart in **Figure 15-38**.

<1> After various settings are made, operation is started by writing 1 to the CTSUSTRT bit of the CTSUCR0 register.

<2> After a channel to be measured is determined according to the preset conditions, a request for setting the corresponding channel (INTCTSUWR) is output.

<3> Upon completion of setting the measurement channel (writing to CTSUSSC, CTSUSO0, and CTSUSO1 registers), the sensor drive pulse is output and the sensor ICO clock operates.

<4> After the sensor stabilization wait time and the measurement time have elapsed and measurement is finished, a measurement result read request (INTCTSURD) is output.

<5> After a channel to be measured next is determined, a measurement channel setting request (INTCTSUWR) is output.

<6> After the stabilization wait time has elapsed and when the previous measurement is read, the result is cleared and measurement is started.

<7> Upon completion of all measurement channels, a measurement end interrupt (INTCTSUFN) is output and measurement is finished (transition to status 0).

Table 15-5 lists the touch pin states in self-capacitance multi-scan mode.

Table 15-5. Touch Pin States in Self-Capacitance Multi-Scan Mode

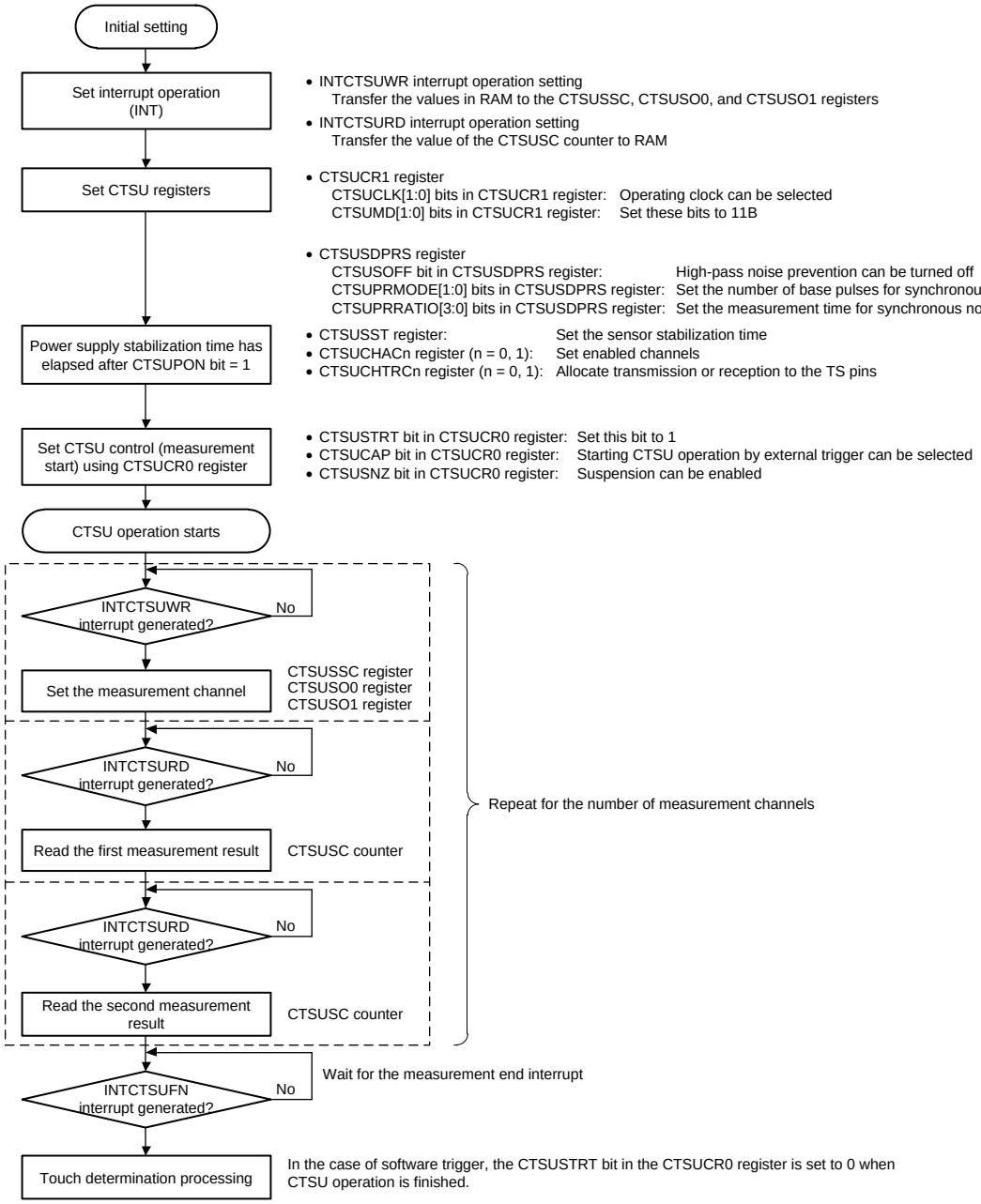
Status	Touch Pin	
	Measurement Channel	Non-measurement Channel
0	Low	Low
1	Low	Low
2	Low	Low
3	Pulse	Low
4	Pulse	Low
5	Low	Low

(5) Mutual capacitance full scan mode operation

In mutual capacitance full scan mode, measurement is performed during the high-level period of the sensor drive pulse on the receive channel by applying the edge to the target transmit channel to be measured. A single measurement target is measured twice, at the rising and falling edges. The difference between the data of these two measurements is used to determine whether or not the electrode is touched, thus achieving higher touch sensitivity.

Electrostatic capacitance is measured sequentially on channels set to transmission or reception by the CTSUCHTRCn registers (n = 0 to 2) and specified as measurement targets by the CTSUCHACn registers (n = 0 to 2). Electrostatic capacitance is measured by combining signals from the measurement target pins that are allocated to transmission or reception. **Figure 15-39** shows the software flowchart and an operation example, and **Figure 15-40** shows the timing chart.

Figure 15-39. Software Flowchart and Operation Example of Mutual Capacitance Full Scan Mode



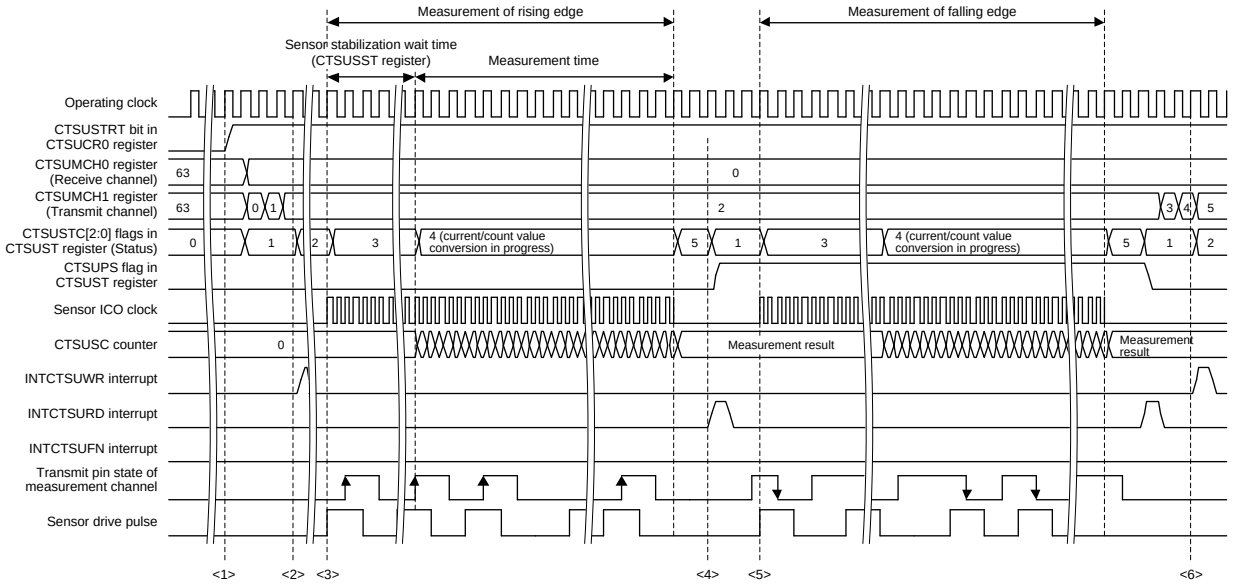
Channel measurement sequence in mutual capacitance full scan mode

<Setting>

- Select mutual capacitance full scan mode (CTSUMD[1:0] bits in CTSUCR1 register = 11B)
- Set channels 0, 3, 5, and 6 as enabled channels (CTSUCHACn[7:0] bits in CTSUCHACn register (n = 0) = 01101001B)
- Set channels 0 to 3 as receive channels and channels 4 to 7 as transmit channels (CTSUCHTRCn[7:0] bits in CTSUCHTRCn register (n = 0) = 1111000B)

		Receive channels			
		Channel 3	Channel 2	Channel 1	Channel 0
Transmit channels	Channel 4				
	Channel 5	<3>			<1>
	Channel 6	<4>			<2>
	Channel 7				

Figure 15-40. Timing Chart of Mutual Capacitance Full Scan Mode (Measurement Start Condition is Software Trigger)



The following describes operation shown in the timing chart in **Figure 15-40**.

<1> After various settings are made, operation is started by writing 1 to the CTSUSTRT bit of the CTSUCR0 register.

<2> After a channel to be measured is determined according to the preset conditions, a request for setting the corresponding channel (INTCTSUWR) is output.

<3> Upon completion of setting the measurement channel (writing to CTSUSSC, CTSUSO0, and CTSUSO1 registers), the sensor drive pulse is output and the sensor ICO clock operates. At the same time, a pulse which is handled as the rising edge is output to the transmit pin on the measurement channel during the high-level period of the sensor drive pulse.

<4> After the sensor stabilization wait time and the measurement time have elapsed and measurement is finished, a measurement result read request (INTCTSURD) is output.

<5> A pulse which is handled as the falling edge is measured during the high-level period of the sensor drive pulse in the channel.

<6> After the channel is measured twice, a channel to be measured next is determined and measured in the similar way.

<7> Upon completion of all measurement channels, a measurement end interrupt (INTCTSUFN) is output and measurement is finished (transition to status 0).

The mutual capacitance measurement status flag (the CTSUPS flag in the CTSUST register) is changed when status 5 transitions to status 1.

Table 15-6 lists the touch pin states in mutual capacitance full scan mode.

Table 15-6. Touch Pin States in Mutual Capacitance Full Scan Mode

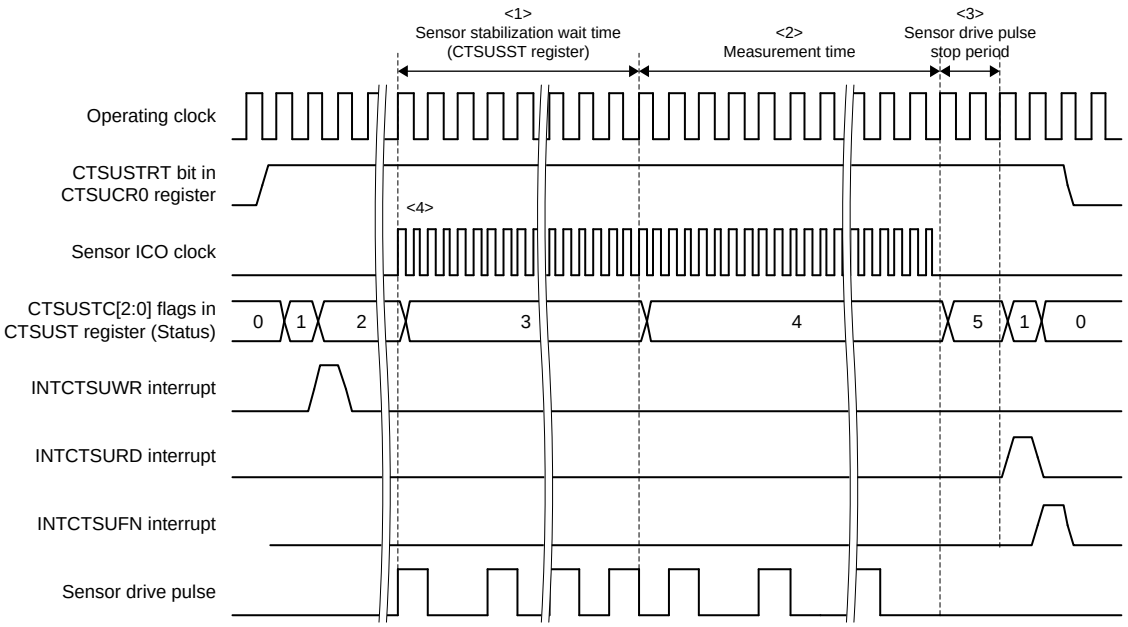
Status	Receive Channel Touch Pin		Transmit Channel Touch Pin		Remarks
	Measurement Channel	Non-measurement Channel	Measurement Channel	Non-measurement Channel	
0	Low	Low	Low	Low	
1	Low	Low	Low/High	Low	
2	Low	Low	Low	Low	
3	Pulse	Low	Pulse	Low	Pulse of the phase same as that of the receive channel at the first measurement. Pulse of the phase opposite to that of the receive channel at the second measurement.
4	Pulse	Low	Pulse	Low	
5	Low	Low	Low	Low	

15.4.3 Items common to multiple modes

(1) Sensor stabilization wait time and measurement time

Figure 15-41 shows the timing chart of the sensor stabilization wait time and measurement time.

Figure 15-41. Sensor Stabilization Wait Time and Measurement Time



<1> In response to the INTCTSUWR interrupt request, output of the sensor drive pulse is started by write access to the CTSUSO1 register. Then, wait for the stabilization time set in the CTSUSST register.

<2> When the sensor stabilization time has elapsed and the CTSUDTSR flag in the CTSUST register is set to 0, measurement is started at transition to status 4. The measurement time is determined by the base clock cycle and the settings of the CTSUPRMODE[1:0] and CTSUPRRATIO[3:0] bits in the CTSUSDPRS register and the CTSUSNUM[5:0] bits in the CTSUSO0 register. When the measurement time has elapsed, measurement of the corresponding channel is finished.

<3> After the measurement time has elapsed, the status transitions to status 1 after two operating clock cycles and an INTCTSURD interrupt is generated, so read the data from the CTSUSC counter. At this time, the sensor drive pulse is output at the low level. When measurement of all specified channels is completed, the CTSUSTRT bit in the CTSUCR0 register becomes 0.

<4> The sensor ICO clock oscillates while the value of the CTSUSTC[2:0] flags in the CTSUST register is 011B (status 3) or 100B (status 4).

(2) Interrupts

There are three types of interrupts for the CTSU:

- Write request interrupt for setting registers for each channel (INTCTSUWR)
- Measurement data transfer request interrupt (INTCTSURD)
- Measurement end interrupt (INTCTSUFN)

a) Write request interrupt for setting registers for each channel (INTCTSUWR)

Store the setting data for each measurement channel in the RAM, and set INT transfer corresponding to the INTCTSUWR interrupt in advance. The INTCTSUWR interrupt is output when status 1 transitions to status 2. Write the setting data of the corresponding channel from the RAM to the CTSUSSC, CTSUSO0, and CTSUSO1 registers. Since write access to the CTSUSO1 register controls a transition to the next status, be sure to set this register last.

The registers (CTSUSSC, CTSUSO0, and CTSUSO1 registers) to be set are allocated at sequential addresses. Set the operation at interrupt generation as shown below:

- Transfer destination address: Address of the CTSUSSC register
- Handling at the transfer destination address: Transfer 2-byte data three times by a single interrupt. (The address of the start byte is fixed.)
- Transfer source address: CTSUSSC register data storage address for the minimum channel in the setting data stored in the RAM
- Handling at the transfer source address: Transfer 2-byte data three times by a single interrupt. (The address of the first byte is continued from the previous interrupt handling.)
- Number of transfers by an interrupt: Specify the number of measurements.

b) Measurement data transfer request interrupt (INTCTSURD)

Set ICU transfer corresponding to the INTCTSURD interrupt in advance. After measurement for one channel is completed, the INTCTSURD interrupt is output on the transition from status 5 to status 1. Read the measurement result from the CTSUSC counter.

The measurement result register (CTSUSC counter) as a transfer source is allocated at sequential addresses. Set the operation at interrupt generation as shown below:

- Transfer source address: Address of the CTSUSC counter
- Handling at the transfer source address: Transfer 2-byte data twice by a single interrupt. (The start address is fixed.)
- Transfer destination address: CTSUSC counter data storage address for the minimum channel in the setting data stored in the RAM
- Handling at the transfer destination address: Transfer 2-byte data twice by a single interrupt. (The start address is continued from the previous interrupt handling.)
- Number of transfers by an interrupt: Specify the number of measurements.

c) Measurement end interrupt (INTCTSUFN)

When all channels are measured, an interrupt is generated when status 1 transitions to status 0. Use software to confirm the overflow flag (CTSUSOVF flag in the CTSUST register) and read the measurement results to determine whether or not the electrode is touched.

Interrupt requests are accepted or disabled in the interrupt control block.

(3) Measurement start conditions

There are two types of measurement start conditions for the CTSU:

- Software trigger

Setting the CTSUCAP bit in the CTSUCR0 register to 0 selects a software trigger as the trigger to start measurement by the CTSU. In this case, measurement by the CTSU starts when the CTSUSTRT bit in the CTSUCR0 register is set to 1.

- External trigger (an interval interrupt signal from the 12-bit interval timer)

Setting the CTSUCAP bit in the CTSUCR0 register to 1 selects an external trigger (an interval interrupt signal input from the 12-bit interval timer as the trigger to start measurement by the CTSU. Start measurement by the CTSU after setting the 12-bit interval timer for the required external trigger. After the CTSUSTRT bit in the CTSUCR0 register is set to 1, measurement starts in response to rising edges of the selected external trigger.

If a further external trigger is input while measurement is in progress, the input is ignored and measurement continues. A next external event acting as the trigger becomes possible one cycle of the operating clock after the INTCTSUFN interrupt.

To use an external trigger, set the 12-bit interval timer by following the procedure below.

1. Initialize the CTSU.
2. Set the 12-bit interval timer while the CTSUSTRT bit in the CTSUCR0 register is 0 (stopping measurement).
3. Set the CTSUCAP bit in the CTSUCR0 register to 1 and then set the CTSUSTRT bit to 1.
   Enabling suspension (the CTSUSNZ bit = 1) drives the CTSU hardware into the suspended state, which decreases power consumption during the wait state. In this case, set the CTSUSTRT bit to 1 after setting the CTSUCAP and CTSUSNZ bits in the CTSUCR0 register to 1.
4. Activate the 12-bit interval timer.
   After activation of the timer, the CPU can be placed in STOP mode.

(4) Intermittent operation using the software trigger (the CTSUCAP bit = 0)

In the system that does not use an external trigger (the CTSUCAP bit = 1), the interval of measurement is generated by an interrupt, etc. and touch measurement operation is started by software (the CTSUSTRT bit = 1).

By enabling suspension (the CTSUSNZ bit = 1) in the waiting state for the start of touch measurement, the CTSU hard macro is placed in the suspended state, which decreases power consumption during the wait state.

If the interval of measurement is long and lower power consumption is required, follow the procedure below to start measurement.

- Configuration

Set the 12-bit interval timer as the source for returning the system from standby.

- Setting the CTSU before standby of the system

Place the system in the standby state while the CTSUCAP bit is 0, the CTSUSNZ bit is 1, and the CTSUSTRT bit is 0. In the SW suspended mode, placing the CPU in STOP mode is also possible.

The CTSU hard macro makes a transition to the suspended state when the CTSUSNZ bit is set to 1.

- Starting measurement after return of the system from standby

To start measurement, follow the procedure below after return of the system from standby.

1. Set the CTSUSNZ bit to 0 to release the CTSU hard macro from the suspended state.
2. Wait for at least 64 cycles of the base clock.
   [Example] When the base clock is at 0.5 MHz, wait for at least 128 µs.
   (2 µs × 64 cycles = 128 µs)
3. Start measurement by a software trigger.

15.5 Usage Notes

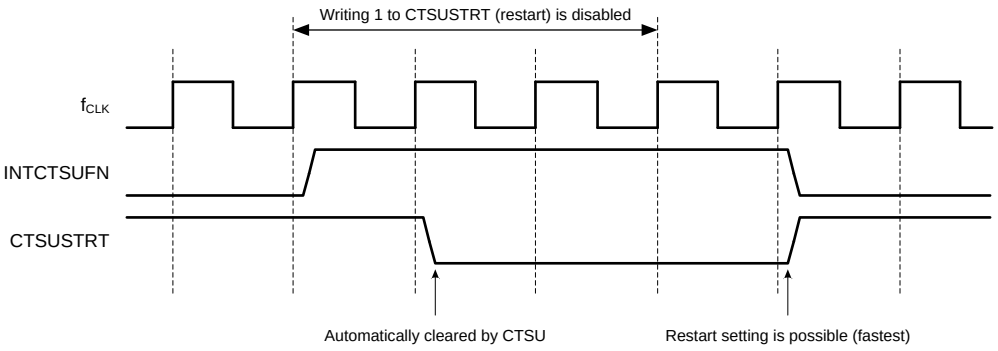
1) Measurement result data (CTSUSC counter)

Read access to this counter during measurement is prohibited. If it is accessed, an incorrect value may be read due to asynchronous operation.

2) Software trigger

When 10B (f_{CLK}/4) is selected by the CTSUCLK[1:0] bits in the CTSUCR1 register, to restart measurement by writing 1 to the CTSUSTRT bit in the CTSUCR0 register after measurement has been completed, wait for at least three cycles to elapse after an interrupt is generated, and then write to the CTSUSTRT bit in the CTSUCR0 register.

Figure 15-42. Notes on Restarting Measurement



3) External trigger (an interval interrupt signal from the 12-bit interval timer)

- If an external trigger is input during the measurement time, measurement is not started. A next external event acting as the trigger becomes possible one cycle of the operating clock after the INTCTSUFN interrupt.
- To stop external trigger mode, write 0 to the CTSUSTRT bit of the CTSUCR0 register and 1 to the CTSUINIT bit of the CTSUCR0 register at the same time (forced stop).

4) Notes on forcibly stopping operation

To forcibly stop the current operation, write 0 to the CTSUSTRT bit of the CTSUCR0 register and 1 to the CTSUINIT bit of the CTSUCR0 register at the same time. After this setting, the operation is stopped and the internal control registers are initialized.

When the CTSUINIT bit of the CTSUCR0 register is used for initialization, the following registers are initialized in addition to the initialization of the internal measurement state

- CTSUMCH0 register
- CTSUMCH1 register
- CTSUST register
- CTSUSC counter

<R> If operation is forcibly stopped, an interrupt request may be generated depending on the internal state. After operation is forcibly stopped, perform the processing for stopping/disabling the interrupt.

5) TSCAP pin

The TSCAP pin requires an external decoupling capacitor to stabilize CTSU internal voltage. The traces between the TSCAP pin and the capacitor, and the capacitor and ground should be as short and wide as physically possible.

The capacitor connected to the TSCAP pin should be fully discharged using I/O port control to output a low level, before turning on the switch (the CTSUCSW bit in the CTSUCR1 register = 1) to establish a connection.

6) Notes during measurement operation (CTSUSTRT bit in CTSUCR0 register = 1)

<R> During measurement operation (the CTSUSTRT bit in the CTSUCR0 register = 1), do not change or stop the CPU/peripheral hardware clock (f_{CLK}). Also, do not change the port settings of the touch pins (TS and TSCAP pins).

If control settings non-compliant to these restrictions are made, after operation is forcibly stopped (the CTSUSTRT bit in the CTSUCR0 register = 0 and the CTSUINIT bit in the CTSUCR0 register = 1), write 0 to the CTSUPON bit of the CTSUCR1 register and 0 to the CTSUCSW bit of the CTSUCR1 register at the same time, and set the CTSUSNZ bit of the CTSUCR0 register to 0. Then, restart from the initial setting flow shown in **Figure 15-32**.

<R> 7) Points to note on noise countermeasures

Fluctuations in the V_{DD} power supply due to noise or other factors may lead to variation in the results of measurement. In addition, the application of ripple noise to the V_{DD} power supply may change (decrease) the control current of the capacitive sensing unit and in turn decrease the measured values of the electrostatic capacitance, depending on the frequency band of the ripple noise. For details, refer to the application note, **Capacitive Touch Ripple Noise Prevention Guide (R30AN0453)**.

Take note of these points when designing the power supply circuit.

<R> **8) Evaluation of detection by the capacitive sensing unit**

The final stages of product development require operating the system under conditions close to those at the time of product shipment to judge the validity of the results of detection by the touch sensor. “QE for Capacitive Touch” (a development assistance tool for capacitive touch sensors) is provided for use in evaluation. Monitor the states in the measurement of capacitance and evaluate them thoroughly. If the expected results are not obtained, use the “QE for Capacitive Touch” tool to adjust and re-evaluate the CapTouch parameters (mainly the touch thresholds).

<R> **9) Memory occupancy**

The amount of memory required to run the capacitive sensing unit depends on the configuration of the buttons. For details, refer to the following application notes.

- RL78 Family CTSU Module Software Integration System (R11AN0484)
- RL78 Family TOUCH Module Software Integration System (R11AN0485)

CHAPTER 16 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.

		32-pin	24-pin	20-pin	16-pin	10-pin
Maskable interrupts	External	10	10	8	8	8
	Internal	30	30	30	26	17

16.1 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, PR10L, PR10H, PR11L, PR11H, PR02L, PR12L). Multiple interrupt servicing can be applied to low-priority interrupts when high-priority interrupts are generated. If two or more interrupt requests, each having the same priority, are simultaneously generated, then they are processed according to the default priority of vectored interrupt servicing. For default priority, see **Table 16-1**.

A standby release signal is generated and STOP and HALT 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.

16.2 Interrupt Sources and Configuration

Interrupt sources include maskable interrupts and software interrupts. In addition, they also have up to seven reset sources (see **Table 16-1**). 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 16-1. Interrupt Source List (1/2)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type[Note 2]	32-pin	24-pin	20-pin	16-pin	10-pin
		Name	Trigger								
Maskable	0	INTWDTI	Watchdog timer interval (75% of the overflow time + 3/(4 × f_{IL}))	Internal	00004H	(A)	✓	✓	✓	✓	✓
	1	INTP0	Pin input edge detection	External	00006H	(B)	✓	✓	✓	✓	✓
	2	INTP1			00008H		✓	✓	✓	✓	✓
	3	INTP2			0000AH		✓	✓	✓	✓	✓
	4	INTP3			0000CH		✓	✓	✓	✓	✓
	5	INTP4			0000EH		✓	✓	✓	✓	✓
	6	INTP5			00010H		✓	✓	✓	✓	✓
	7	INTST0/ INTCSI00/ INTIIC00	UART0 transmission transfer end, buffer empty interrupt/CSI00 transfer end, buffer empty interrupt/IIC00 transfer end	Internal	00012H	(A)	✓	✓	✓	✓	✓
	8	INTSR0	UART0 reception transfer end		00014H		✓	✓	✓	✓	✓
	9	INTSRE0	UART0 reception communication error occurrence		00016H		✓	✓	✓	✓	✓
	10	INTTM01H	End of counting or capture by timer channel 01 (at higher 8-bit timer operation)		00018H		✓	✓	✓	✓	✓
	11	INTTM00	End of counting or capture by timer channel 00		0001AH		✓	✓	✓	✓	✓
	12	INTTM01	End of counting or capture by timer channel 01 (at 16-bit/lower 8-bit timer operation)		0001CH		✓	✓	✓	✓	✓
	13	INTST1	UART1 transmission transfer end, buffer empty interrupt		0001EH		✓	✓	✓	✓	—
	14	INTSR1/ INTCSI11/ INTIIC11	UART1 reception transfer end/CSI11 transfer end, buffer empty interrupt/IIC11 transfer end		00020H		✓	✓	✓	✓	—
	15	INTSRE1	UART1 reception communication error occurrence		00022H		✓	✓	✓	✓	—
	16	INTST2/ INTCSI20/ INTIIC20	UART2 transmission transfer end, buffer empty interrupt/CSI20 transfer end, buffer empty interrupt/IIC20 transfer end		00024H		✓	✓	✓	—	—
	17	INTSR2	UART2 reception transfer end		00026H		✓	✓	✓	—	—
	18	INTSRE2	UART2 reception communication error occurrence		00028H		✓	✓	✓	—	—
	19	INTTM03H	End of counting or capture by timer channel 03 (at higher 8-bit timer operation)		0002AH		✓	✓	✓	✓	✓ Note 4
	20	INTIICA0	End of IICA0 communication		0002CH		✓	✓	✓	✓	✓
	21	INTTM02	End of counting or capture by timer channel 02		0002EH		✓	✓	✓	✓	✓
	22	INTTM03	End of counting or capture by timer channel 03 (at 16-bit/lower 8-bit timer operation)		00030H		✓	✓	✓	✓	✓ Note 4
	23	INTAD	End of A/D conversion		00032H		✓	✓	✓	✓	✓
	24	INTRTC	Fixed-cycle signal of real-time clock 2/alarm match detection		00034H		✓	✓	✓	✓	—
	25	INTRTIT	RTC correction timing		00036H		✓	✓	✓	✓	—
	26	INTIT	12-bit interval timer interval signal detection		00038H		✓	✓	✓	✓	✓
	27	INTTM04	End of counting or capture by timer channel 04		0003AH		✓	✓	✓	✓	✓ Note 4
	28	INTTM05	End of counting or capture by timer channel 05		0003CH		✓	✓	✓	✓	✓ Note 4

Table 16-1. Interrupt Source List (2/2)

Interrupt Type	Default Priority[Note 1]	Interrupt Source		Internal/External	Vector Table Address	Basic Configuration Type[Note 2]	32-pin	24-pin	20-pin	16-pin	10-pin
		Name	Trigger								
Maskable	29	INTTM06	End of counting or capture by timer channel 06	Internal	0003EH	(A)	✓	✓	✓	✓	✓ Note 4
	30	INTTM07	End of counting or capture by timer channel 07		00040H		✓	✓	✓	✓	✓ Note 4
	31	INTP6	Pin input edge detection	External	00042H	(B)	✓	✓	✓	✓	✓
	32	INTP7	Pin input edge detection		00044H		✓	✓	✓	✓	✓
	33	INTP8	Pin input edge detection		00046H		✓	✓	—	—	—
	34	INTP9	Pin input edge detection		00048H		✓	✓	—	—	—
	35	INTCMP0	Valid edge detection by comparator 0	Internal	0004AH	(A)	✓	✓	✓	✓	✓
	36	INTCMP1	Valid edge detection by comparator 1		0004CH		✓	✓	✓	✓	—
	37	INTCTSUWR	End of writing to the setting register for each CTSU channel		0004EH		✓	✓	✓	✓	✓
	38	INTCTSURD	End of transfer of CTSU measurement data		00050H		✓	✓	✓	✓	✓
	39	INTCTSUFN	End of CTSU measurement		00052H		✓	✓	✓	✓	✓
Software	—	BRK	Execution of BRK instruction	—	0007EH	(C)	✓	✓	✓	✓	✓
Reset	—	RESET	$\overline{\text{RESET}}$ pin input	—	00000H	—	✓	✓	✓	✓	✓
		SPOR	Selectable power-on-reset				✓	✓	✓	✓	✓
		WDT	Overflow of watchdog timer				✓	✓	✓	✓	✓
		TRAP	Execution of illegal instruction[Note 3]				✓	✓	✓	✓	✓
		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. 0 indicates the highest priority and 39 indicates the lowest priority.

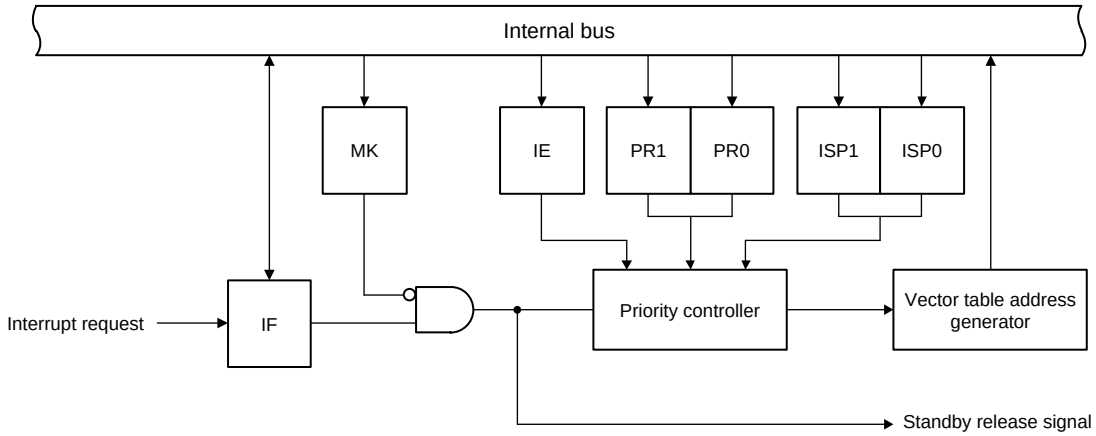
Note 2. Basic configuration types (A) to (C) correspond to (A) to (C) in **Figure 16-1**.

Note 3. A reset is generated when the instruction code in FFH is executed.
No reset is generated if an illegal instruction is executed during emulation by the on-chip debug emulator.

Note 4. Completion of counting by a channel of the array unit is only possible.

Figure 16-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal maskable interrupt



(B) External maskable interrupt (INTPn)

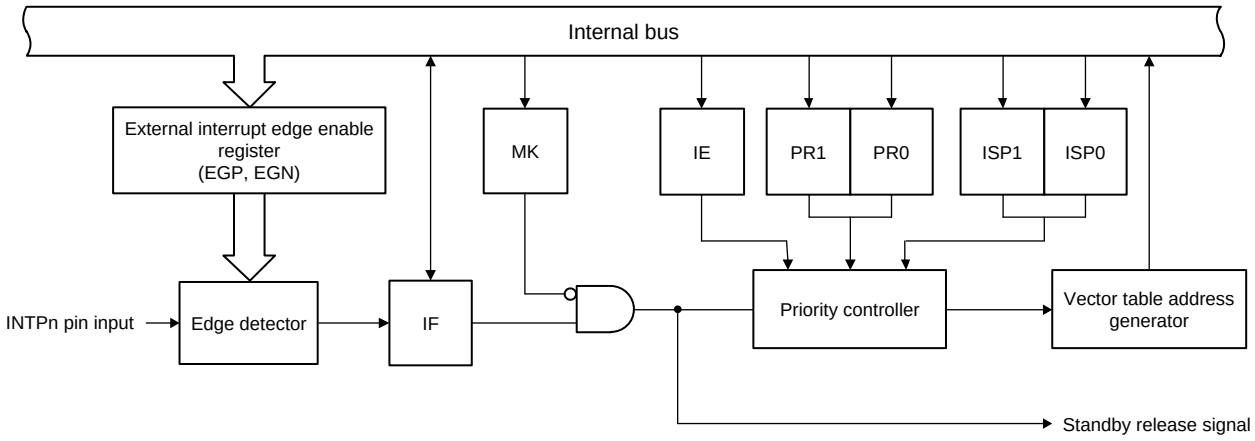
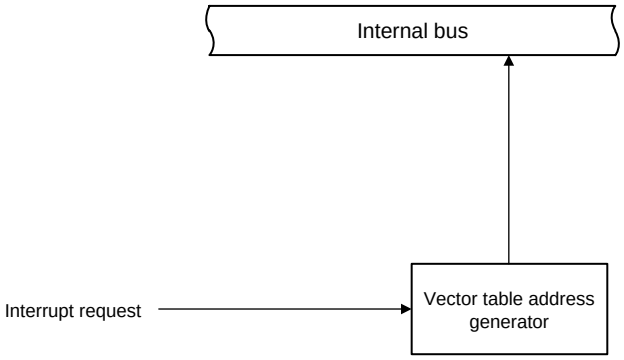


Figure 16-1. Basic Configuration of Interrupt Function (2/2)

(C) 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 10-, 16-, and 20-pin: n = 0 to 7,
24- and 32-pin: n = 0 to 9

16.3 Registers Controlling Interrupt Functions

The following 6 types of registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L)
- Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L)
- Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR10L, PR10H, PR11L, PR11H, PR12L)
- External interrupt rising edge enable register (EGP0, EGP1)
- External interrupt falling edge enable register (EGN0, EGN1)
- Program status word (PSW)

Table 16-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 16-2. Flags Corresponding to Interrupt Request Sources (1/2)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		32-pin	24-pin	20-pin	16-pin	10-pin
		Register		Register		Register					
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L	✓	✓	✓	✓	✓
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		✓	✓	✓	✓	✓
INTST0[Note 1]	STIF0[Note 1]		STMK0[Note 1]		STPR00, STPR10[Note 1]		✓	✓	✓	✓	✓
INTCSI00[Note 1]	CSIIF00[Note 1]		CSIMK00[Note 1]		CSIPR000, CSIPR100[Note 1]		✓	✓	✓	✓	✓
INTIIC00[Note 1]	IICIF00[Note 1]		IICMK00[Note 1]		IICPR000, IICPR100[Note 1]		✓	✓	✓	✓	✓
INTSR0	SRIF0	IF0H	SRMK0	MK0H	SRPR00, SRPR10	PR00H, PR10H	✓	✓	✓	✓	✓
INTSRE0	SREIF0		SREMK0		SREPR00, SREPR10		✓	✓	✓	✓	✓
INTTM01H	TMIF01H		TMMK01H		TMPR001H, TMPR101H		✓	✓	✓	✓	✓
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100		✓	✓	✓	✓	✓
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101		✓	✓	✓	✓	✓
INTST1	STIF1		STMK1		STPR01, STPR11		✓	✓	✓	✓	—
INTSR1	SRIF1		SRMK1		SRPR01, SRPR11		✓	✓	✓	✓	—
INTCSI11	CSIIF11		CSIMK11		CSIPR011, CSIPR111		✓	✓	✓	✓	—
INTIIC11	IICIF11		IICMK11		IICPR011, IICPR111		✓	✓	✓	✓	—
INTSRE1	SREIF1		SREMK1		SREPR01, SREPR11		✓	✓	✓	✓	—

Table 16-2. Flags Corresponding to Interrupt Request Sources (2/2)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		32-pin	24-pin	20-pin	16-pin	10-pin
		Register		Register		Register					
INTST2Note 2	STIF2Note 2	IF1L	STMK2Note 2	MK1L	STPR02, STPR12Note 2	PR01L, PR11L	✓	✓	✓	—	—
INTCSI20Note 2	CSIIF20Note 2		CSIMK20Note 2		CSIPR020, CSIPR120Note 2		✓	✓	✓	—	—
INTIIC20Note 2	IICIF20Note 2		IICMK20Note 2		IICPR020, IICPR120Note 2		✓	✓	✓	—	—
INTSR2	SRIF2		SRMK2		SRPR02, SRPR12		✓	✓	✓	—	—
INTSRE2	SREIF2		SREMK2		SREPR02, SREPR12		✓	✓	✓	—	—
INTTM03H	TMIF03H		TMMK03H		TMPR003H, TMPR103H		✓	✓	✓	✓	✓
INTIICA0	IICAIF0		IICAMK0		IICAPR00, IICAPR10		✓	✓	✓	✓	✓
INTTM02	TMIF02		TMMK02		TMPR002, TMPR102		✓	✓	✓	✓	✓
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103		✓	✓	✓	✓	✓
INTAD	ADIF		ADMK		ADPR0, ADPR1		✓	✓	✓	✓	✓
INTRTC	RTCIF	IF1H	RTCMK	MK1H	RTCPR0, RTCPR1	PR01H, PR11H	✓	✓	✓	✓	—
INTRTIT	RTITIF		RTITMK		RTITPR0, RTITPR1		✓	✓	✓	✓	—
INTIT	ITIF		ITMK		ITPR0, ITPR1		✓	✓	✓	✓	✓
INTTM04	TMIF04		TMMK04		TMPR004, TMPR104		✓	✓	✓	✓	✓
INTTM05	TMIF05		TMMK05		TMPR005, TMPR105		✓	✓	✓	✓	✓
INTTM06	TMIF06		TMMK06		TMPR006, TMPR106		✓	✓	✓	✓	✓
INTTM07	TMIF07		TMMK07		TMPR007, TMPR107		✓	✓	✓	✓	✓
INTP6	PIF6		PMK6		PPR06,PPR16		✓	✓	✓	✓	✓
INTP7	PIF7	1F2L	PMK7	MK2L	PPR07,PPR17	PR02L, PR12L	✓	✓	✓	✓	✓
INTP8	PIF8		PMK8		PPR08, PPR18		✓	✓	—	—	—
INTP9	PIF9		PMK9		PPR09, PPR19		✓	✓	—	—	—
INTCMP0	CMPIF0		CMPMK0		CMPPR00, CMPPR10		✓	✓	✓	✓	✓
INTCMP1	CMPIF1		CMPMK1		CMPPR01, CMPPR11		✓	✓	✓	✓	—
INTCTSUWR	CTSUWRIF		CTSUWRMK		CTSUWRPR0, CTSUWRPR1		✓	✓	✓	✓	✓
INTCTSURD	CTSURDIF		CTSURDMK		CTSURDPR0, CTSURDPR1		✓	✓	✓	✓	✓
INTCTSUFN	CTSUFNIF		CTSUFNMK		CTSUFNPR0, CTSUFNPR1		✓	✓	✓	✓	✓

Note 1. If one of the interrupt sources INTST0, INTCSI00, and INTIIC00 is generated, bit 7 of the IF0L register is set to 1. Bit 7 of the MK0L, PR00L, and PR10L registers supports these three interrupt sources.

Note 2. If one of the interrupt sources INTST2, INTCSI20, INTIIC20 is generated, bit 0 of the IF1L register is set to 1. Bit 0 of the MK1L, PR01L, and PR11L registers supports these three interrupt sources.

16.3.1 Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L)

The interrupt request flags are set to 1 when the corresponding interrupt request is generated or an instruction is executed. They are cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon reset signal generation.

When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is entered.

The IF0L, IF0H, IF1L, IF1H, and IF2L registers can be set by a 1-bit or 8-bit memory manipulation instruction. When the IF0L and IF0H registers, and the IF1L and IF1H registers are combined to form 16-bit registers IF0 and IF1, they can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks

Figure 16-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L)

Address: FFFE0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF0L	STIF0 CSIIF00 IICIF00	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	WDTIIF

Address: FFFE1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF0H	SREIF1	SRIF1 CSIIF11 IICIF11	STIF1	TMIF01	TMIF00	TMIF01H	SREIF0	SRIF0

Address: FFFE2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF1L	ADIF	TMIF03	TMIF02	IICAIF0	TMIF03H	SREIF2	SRIF2	STIF2 CSIIF20 IICIF20

Address: FFFE3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF1H	PIF6	TMIF07	TMIF06	TMIF05	TMIF04	ITIF	RTITIF	RTCIF

Address: FFFD0H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IF2L	CTSUFNIF	CTSURDIF	CTSUWRIF	CMPIF1	CMPIF0	PIF9	PIF8	PIF7

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	An interrupt request signal is generated and the interrupt is requested.

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 16-2. 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.

16.3.2 Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt.

The MK0L, MK0H, MK1L, MK1H, and MK2L registers can be set by a 1-bit or 8-bit memory manipulation instruction.

When the MK0L and MK0H registers, and the MK1L and MK1H registers are combined to form 16-bit registers MK0 and MK1, they can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets these registers to F

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 16-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L)

Address: FFFE4H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK0L	STMK0 CSIMK00 IICMK00	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	WDTIMK

Address: FFFE5H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK0H	SREMK1	SRMK1 CSIMK11 IICMK11	STMK1	TMMK01	TMMK00	TMMK01H	SREMK0	SRMK0

Address: FFFE6H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK1L	ADMK	TMMK03	TMMK02	IICAMK0	TMMK03H	SREMK2	SRMK2	STMK2 CSIMK20 IICMK20

Address: FFFE7H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK1H	PMK6	TMMK07	TMMK06	TMMK05	TMMK04	ITMK	RTITMK	RTCMK

Address: FFFD4H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
MK2L	CTSUFNMK	CTSURDMK	CTSUWRMK	CMPMK1	CMPMK0	PMK9	PMK8	PMK7

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 16-2. Be sure to set bits that are not available to the initial value.

16.3.3 Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR10L, PR10H, PR11L, PR11H, PR12L)

The priority specification flag registers are used to set the corresponding maskable interrupt priority level.

A priority level is set by using the PR0xy and PR1xy registers in combination (xy = 0L, 0H, 1L, 1H, 2L).

The PR00L, PR00H, PR01L, PR01H, PR02L, 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, 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 16-4. Format of Priority Specification Flag Registers

(PR00L, PR00H, PR01L, PR01H, PR02L, PR10L, PR10H, PR11L, PR11H, PR12L)

Address: FFFE8H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR00L	STPR00 CSIPR000 IICPR000	PPR05	PPR04	PPR03	PPR02	PPR01	PPR00	WDTIPR0

Address: FFFECH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR10L	STPR10 CSIPR100 IICPR100	PPR15	PPR14	PPR13	PPR12	PPR11	PPR10	WDTIPR1

Address: FFFE9H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR00H	SREPR01	SRPR011 CSIPR011 IICPR011	STPR01	TMPR001	TMPR000	TMPR001H	SREPR00	SRPR00

Address: FFFEDH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR10H	SREPR11	SRPR111 CSIPR111 IICPR111	STPR11	TMPR101	TMPR100	TMPR101H	SREPR10	SRPR10

Address: FFFEAH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR01L	ADPR0	TMPR003	TMPR002	IICAPR00	TMPR003H	SREPR02	SRPR02	STPR02 CSIPR020 IICPR020

Address: FFFEEH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR11L	ADPR1	TMPR103	TMPR102	IICAPR10	TMPR103H	SREPR12	SRPR12	STPR12 CSIPR120 IICPR120

Address: FFFEBH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR01H	PPR06	TMPR007	TMPR006	TMPR005	TMPR004	ITPR0	RTITPR0	RTCPR0

Address: FFFEFH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR11H	PPR16	TMPR107	TMPR106	TMPR105	TMPR104	ITPR1	RTITPR1	RTCPR1

Address: FFFD8H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR02L	CTSUFNPR0	CTSURDPR0	CTSUWRPR0	CMPPPR01	CMPPR00	PPR09	PPR08	PPR07

Address: FFFDCH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PR12L	CTSUFNPR1	CTSURDPR1	CTSUWRPR1	CMPPPR11	CMPPR10	PPR19	PPR18	PPR17

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 16-2. Be sure to set bits that are not available to the initial value.

16.3.4 External interrupt rising edge enable register (EGP0, EGP1), external interrupt falling edge enable register (EGN0, EGN1)

These registers specify the valid edge for INTP0 to INTP9.

The EGP0 EGP1, EGN0, and EGN1 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 16-5. Format of External Interrupt Rising Edge Enable Register (EGP0, EGP1) and External Interrupt Falling Edge Enable Register (EGN0, EGN1)

Address: FFF38H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	EGP07	EGP06	EGP05	EGP04	EGP03	EGP02	EGP01	EGP00

Address: FFF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	EGN07	EGN06	EGN05	EGN04	EGN03	EGN02	EGN01	EGN00

Address: FFF3AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP1	0	0	0	0	0	0	EGP09	EGP08

Address: FFF3BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN1	0	0	0	0	0	0	EGN09	EGN08

EGP0n	EGN0n	INTPn pin valid edge selection (n = 0 to 9)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 16-3 shows the ports corresponding to the EGP0 and EGP1 bits, and EGN0 and EGN1 bits.

Table 16-3. Interrupt Request Signals Corresponding to EGP0 and EGP1 Bits, and EGN0 and EGN1 Bits

Detection Enable Bit		Interrupt Request Signal	24-, 32-pin	10-, 16-, 20-pin
EGP00	EGN00	INTP0	✓	✓
EGP01	EGN01	INTP1	✓	✓
EGP02	EGN02	INTP2	✓	✓
EGP03	EGN03	INTP3	✓	✓
EGP04	EGN04	INTP4	✓	✓
EGP05	EGN05	INTP5	✓	✓
EGP06	EGN06	INTP6	✓	✓
EGP07	EGN07	INTP7	✓	✓
EGP08	EGN08	INTP8	✓	—
EGP09	EGN09	INTP9	✓	—

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 Function**.

Remark 2. n = 0 to 9

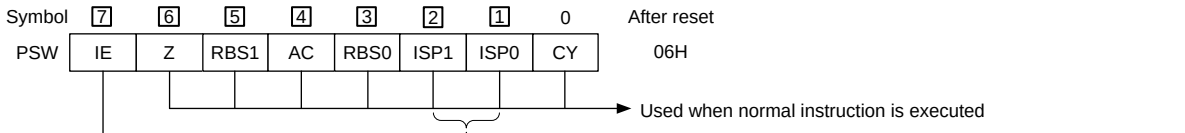
16.3.5 Program status word (PSW)

The program status word is a register used to hold the instruction execution result and the current status for an interrupt request. The IE flag that sets maskable interrupt enable/disable and the ISP0 and ISP1 flags that control multiple interrupt servicing are mapped to the PSW.

Besides 8-bit read/write, this register can carry out operations using bit manipulation instructions and dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged, if the BRK instruction is executed, the contents of the PSW are automatically saved into a stack and the IE flag is reset to 0. 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 16-6. Configuration of Program Status Word



ISP1	ISP0	Priority of interrupt currently being serviced
0	0	Enables interrupts of level 0 (while an interrupt of level 1 or 0 is being serviced)
0	1	Enables interrupts of levels 0 and 1 (while an interrupt of level 2 is being serviced)
1	0	Enables interrupts of levels 0 to 2 (while an interrupt of level 3 is being serviced)
1	1	Enables all interrupts (wait for acknowledgment of an interrupt)

IE	Interrupt request acknowledgment enable/disable
0	Disabled
1	Enabled

16.4 Interrupt Servicing Operations

16.4.1 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 16-4**.

For the interrupt request acknowledgment timing, see **Figure 16-8** and **Figure 16-9**.

Table 16-4. Time from Generation of Maskable Interrupt until Servicing

	Minimum Time	Maximum Time[Note 1]
Servicing time	9 clocks	16 clocks

Note 1. Maximum time does not apply when an instruction from the internal RAM area is executed.

Remark 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specification flag is acknowledged first. If two or more interrupts requests have the same priority level, the request with the highest default priority is acknowledged first.

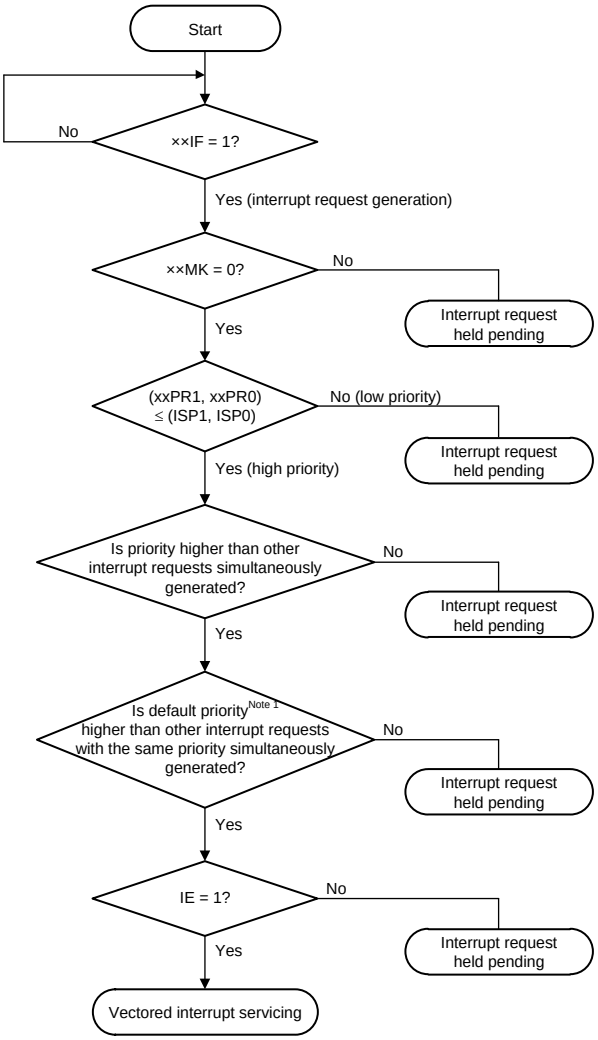
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 16-7 shows the interrupt request acknowledgment algorithm.

If a maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag is reset (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.

Returning from an interrupt is possible by using the RETI instruction.

Figure 16-7. Interrupt Request Acknowledgment Processing Algorithm



xxIF: Interrupt request flag

xxMK: Interrupt mask flag

xxPR0: Priority specification flag 0

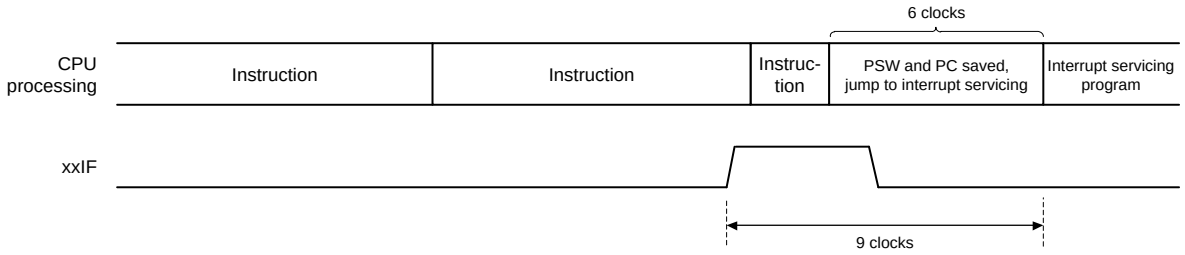
xxPR1: Priority specification flag 1

IE: Flag that controls acknowledgment of maskable interrupt request (1 = Enable, 0 = Disable)

ISP0, ISP1: Flag that indicates the priority level of the interrupt currently being serviced (see **Figure 16-6**)

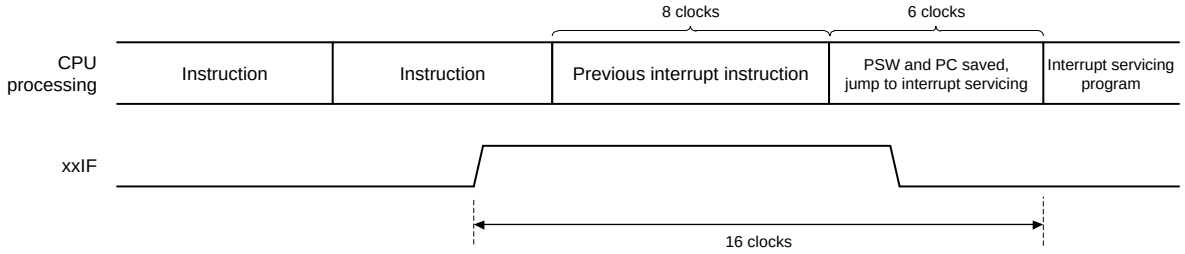
Note 1. For the default priority, refer to **Table 16-1 Interrupt Source List**.

Figure 16-8. Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

Figure 16-9. Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

16.4.2 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.

Returning from a software interrupt is possible by using the RETB instruction.

Caution The RETI instruction cannot be used for return from the software interrupt.

16.4.3 Multiple interrupt servicing

Multiple interrupt servicing occurs when another interrupt request is acknowledged during execution of an interrupt.

Multiple interrupt servicing does not occur unless the interrupt request acknowledgment enabled state is selected (IE = 1). When an interrupt request is acknowledged, interrupt request acknowledgment becomes disabled (IE = 0). Therefore, to enable multiple interrupt servicing, it is necessary to set (1) the IE flag with the EI instruction during interrupt servicing to enable interrupt acknowledgment.

Moreover, even if interrupts are enabled, multiple interrupt servicing may not be enabled, this being subject to interrupt priority control. Two types of priority control are available: default priority control and programmable priority control. Programmable priority control is used for multiple interrupt servicing.

In the interrupt enabled state, if an interrupt request with a priority higher than that of the interrupt currently being serviced is generated, it is acknowledged for multiple interrupt servicing. If an interrupt with a priority equal to or lower than that of the interrupt currently being serviced is generated during interrupt servicing, it is not acknowledged for multiple interrupt servicing. However, when setting the IE flag to 1 during the interruption at level 0, other level 0 interruptions can be allowed.

Interrupt requests that are not enabled because interrupts are in the interrupt disabled state or because they have a lower priority are held pending. When servicing of the current interrupt ends, the pending interrupt request is acknowledged following execution of at least one main processing instruction execution.

Table 16-5 shows relationship between interrupt requests enabled for multiple interrupt servicing and **Figure 16-10** shows multiple interrupt servicing examples.

Table 16-5. 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

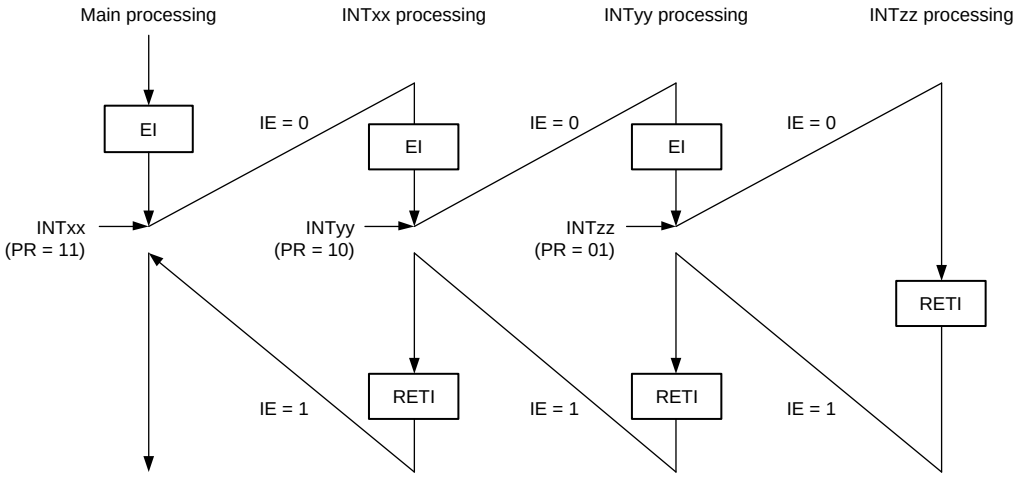
Remark 2. ×: Multiple interrupt servicing disabled

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 are enabled).
IE = 0: Interrupt request acknowledgment is disabled.
IE = 1: Interrupt request acknowledgment is enabled.

Remark 4. PR is a flag contained in the PR00L, PR00H, PR01L, PR01H, PR02L, PR10L, PR10H, PR11L, PR11H, and PR12L registers.
PR = 00: Specify level 0 with xxPR1x = 0, xxPR0x = 0 (higher priority level)
PR = 01: Specify level 1 with xxPR1x = 0, xxPR0x = 1
PR = 10: Specify level 2 with xxPR1x = 1, xxPR0x = 0
PR = 11: Specify level 3 with xxPR1x = 1, xxPR0x = 1 (lower priority level)

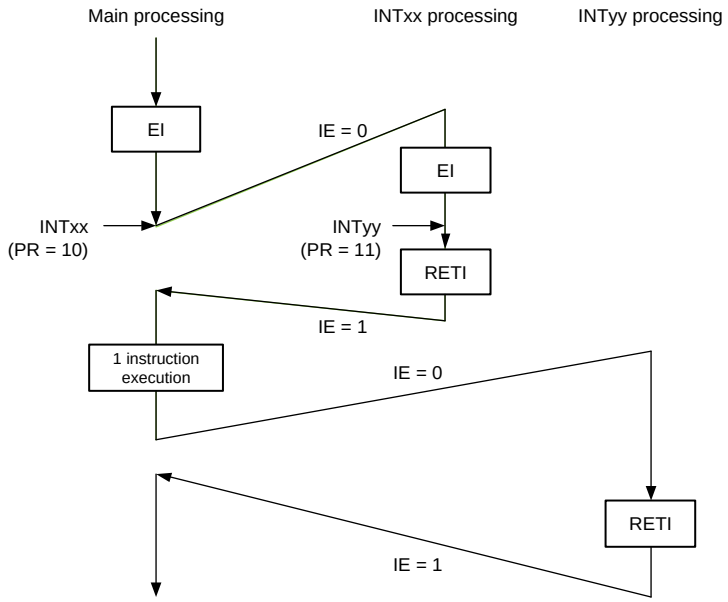
Figure 16-10. Examples of Multiple Interrupt Servicing (1/2)

Example 1. Multiple interrupt servicing occurs twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and multiple interrupt servicing takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledgment.

Example 2. Multiple interrupt servicing does not occur due to priority control



Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

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

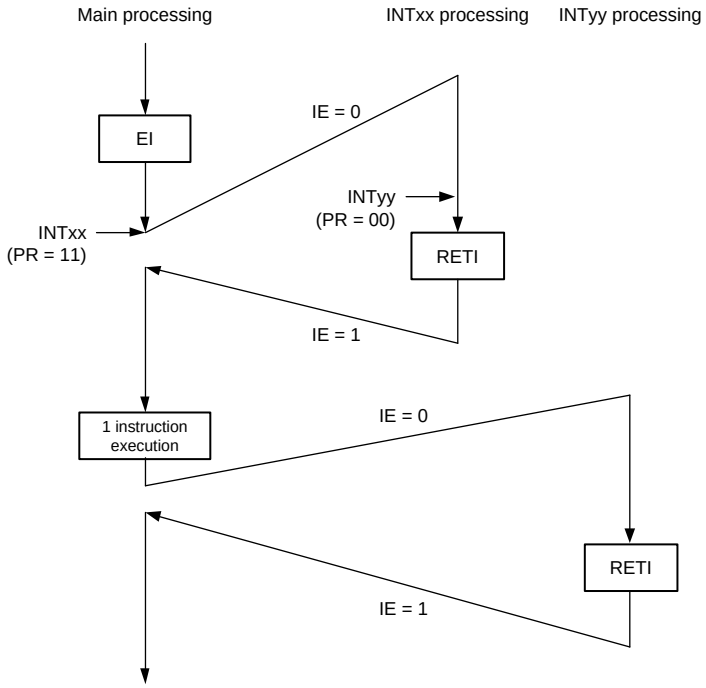
PR = 11: Specify level 3 with xxPR1x = 1, xxPR0x = 1 (lower priority level)

IE = 0: Interrupt request acknowledgment is disabled

IE = 1: Interrupt request acknowledgment is enabled.

Figure 16-10. Examples of Multiple Interrupt Servicing (2/2)

Example 3. Multiple interrupt servicing does not occur because interrupts are not enabled



Interrupts are not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

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.

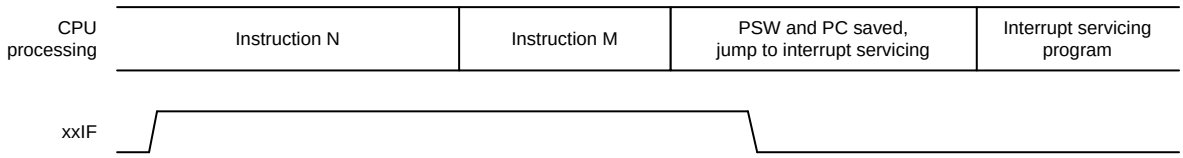
16.4.4 Interrupt request pending

There are instructions where, even if an interrupt request is issued while the instructions are being executed, interrupt request acknowledgment is held pending until the end of execution of the next instruction. These instructions (instructions that hold interrupt requests pending) are listed below.

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, $addr20
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- SKH
- SKNH
- Write instructions for the IF0L, IF0H, IF1L, IF1H, IF2L, MK0L, MK0H, MK1L, MK1H, MK2L, PR00L, PR00H, PR01L, PR01H, PR02L, PR10L, PR10H, PR11L, PR11H, and PR12L registers

Figure 16-11 shows the timing at which interrupt requests are held pending.

Figure 16-11. Interrupt Request Pending



Remark Instruction N: Instruction to hold interrupt requests pending
Instruction M: Instruction other than the instruction to hold interrupt requests pending

CHAPTER 17 STANDBY FUNCTION

17.1 Overview

The standby function reduces the operating current of the system, and the following two 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, 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 and high-speed on-chip oscillator stop, stopping the whole system, thereby considerably reducing the CPU operating current.

Because this mode can be cleared by an interrupt request, it enables intermittent operations to be carried out. However, because a wait time is required to secure the oscillation stabilization time after the STOP mode is released when the X1 clock is selected, select the HALT mode if it is necessary to start processing immediately upon interrupt request generation.

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 status are also held.

<R> **Caution 1. When shifting to the STOP mode, be sure to stop the peripheral hardware operating with the main system clock before executing STOP instruction[Note 1]. When the CPU is operating with the subsystem clock, do not set it to the STOP mode.**

Caution 2. 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 3. 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 21 OPTION BYTE.

Note 1. Other than 10-pin products.

17.2 Registers controlling standby function

The standby function is controlled by the following registers.

For details of each register, see **CHAPTER 5 CLOCK GENERATOR**.

- Operation speed mode control register (OSMC)
- Oscillation stabilization time counter status register (OSTC)Note 1
- Oscillation stabilization time select register (OSTS)Note 1

Note 1. Other than 10-pin products.

17.3 Standby Function Operation

17.3.1 HALT mode

(1) HALT mode

The HALT mode is set by executing the HALT instruction.

The HALT mode can be set regardless of whether the CPU clock before the setting was the high-speed system clock (other than 10-pin products), the high-speed on-chip oscillator clock, or the subsystem clock. The operating status 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 17-1. Operating Status in HALT Mode (1) (1/2)

HALT Mode Setting / Item			When HALT Instruction is Executed While CPU is Operating on Main System Clock		
			When CPU is Operating on High-speed On-chip Oscillator Clock (f_{IH})	When CPU is Operating on X1 Clock (f_X)	When CPU is Operating on External Main System Clock (f_{EX})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{IH}	Operation continues (cannot be stopped)	Operation disabled	
		f_X	Operation disabled	Operation continues (cannot be stopped)	Cannot operate
		f_{EX}		Cannot operate	Operation continues (cannot be stopped)
	Subsystem clock Note 2	f_{XT}	Status before HALT mode was set is retained	Operation disabled	Operation disabled
		f_{EXS}			
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and bit 4 (WUTMMCK0) of operation speed mode control register (OSMC) • WUTMMCK0 = 1: Oscillates • WUTMMCK0 = 0 and WDTON = 0: Stops • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 1: Oscillates • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 0: Stops		
CPU			Operation stopped		
Code flash memory					
Data flash memory					
RAM					
Illegal-memory access detection function					
Port (latch)			Status before HALT mode was set is retained		
Timer array unit			Operable		
Real-time clock 2					
12-bit interval timer					
Watchdog timer			Set by bit 0 (WDSTBYON) of option byte (000C0H) WDSTBYON = 0: Operation stopped WDSTBYON = 1: Operation continues		
Clock output/buzzer output			Operable		
A/D converter					
Comparator					
Serial array unit (SAU)					
Serial interface (IICA)					
Capacitive touch sensing unit (CTSU)					
Selectable power-on-reset function					
External interrupt					
Internal reset by data retention power supply voltage			Operating		

Table 17-1. Operating Status in HALT Mode (1) (2/2)

HALT Mode Setting / Item	When HALT Instruction is Executed While CPU is Operating on Main System Clock		
	When CPU is Operating on High-speed On-chip Oscillator Clock (f_{IH})	When CPU is Operating on X1 Clock (f_X)	When CPU is Operating on External Main System Clock (f_{EX})
CRC operation function (General-purpose CRC)	Operation stopped		
RAM parity error detection function			
RAM guard function			
SFR guard function			
Illegal-memory access detection 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_X: X1 clock[Note 1]
f_{EX}: External main system clock[Note 1]
f_{XT}: XT1 clock[Note 1]
f_{EXS}: External subsystem clock[Note 1]

Note 1. Other than 10-pin products.

Note 2. X1 oscillator and XT1 oscillator cannot operate at the same time because X1 and X2 pins of the X1 oscillator are shared with XT1 and XT2 pins of the XT1 oscillator.

Table 17-2. Operating Status in HALT Mode (2) (1/2)

HALT Mode Setting / Item			When HALT Instruction is Executed While CPU is Operating on Subsystem Clock	
			When CPU is Operating on XT1 Clock (f_{XT})	When CPU is Operating on External Subsystem Clock (f_{EXS})
System clock			Clock supply to the CPU is stopped	
	Main system clock	f_{IH}	Operation disabled	
		f_X		
		f_{EX}		
	Subsystem clock Note 2	f_{XT}	Operation continues (cannot be stopped)	Cannot operate
		f_{EXS}	Cannot operate	Operation continues (cannot be stopped)
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and bit 4 (WUTMMCK0) of operation speed mode control register (OSMC) • WUTMMCK0 = 1: Oscillates • WUTMMCK0 = 0 and WDTON = 0: Stops • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 1: Oscillates • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 0: Stops	
CPU			Operation stopped	
Code flash memory				
Data flash memory				
RAM			Operation stopped	
Port (latch)			Status before HALT mode was set is retained	
Timer array unit			Operable when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0)	
Real-time clock 2			Operable	
12-bit interval timer				
Watchdog timer			Set by bit 0 (WDSTBYON) of option byte (000C0H) WDSTBYON = 0: Operation stopped WDSTBYON = 1: Operation continues	
Clock output/buzzer output			Operable when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0)	
A/D converter			Operation disabled	
Comparator			Operable when the RTCLPC bit is 0 and when the external input (IVREFn) is selected as the reference voltage of the comparator (operation is disabled when other than the external input (IVREFn) is selected and the RTCLPC bit is not 0)	
Serial array unit (SAU)			Operable when the RTCLPC bit is 0 (operation is disabled when the RTCLPC bit is not 0)	
Serial interface (IICA)			Operation disabled	
<R> Capacitive touch sensing unit (CTSU)				
External interrupt			Operable	
Selectable power-on-reset function				
Internal reset by data retention power supply voltage			Operating	

Table 17-2. Operating Status in HALT Mode (2) (2/2)

HALT Mode Setting / Item	When HALT Instruction is Executed While CPU is Operating on Subsystem Clock	
	When CPU is Operating on XT1 Clock (f_{XT})	When CPU is Operating on External Subsystem Clock (f_{EXS})
CRC operation function (General-purpose CRC)	Operation stopped	
RAM parity error detection function		
RAM guard function		
SFR guard function		
Illegal-memory access detection 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_X: X1 clock[Note 1]
f_{EX}: External main system clock[Note 1]
f_{XT}: X1 clock[Note 1]
f_{EXS}: External subsystem clock[Note 1]

Note 1. Other than 10-pin products

Note 2. X1 oscillator and XT1 oscillator cannot operate at the same time because X1 and X2 pins of the X1 oscillator are shared with XT1 and XT2 pins of the XT1 oscillator.

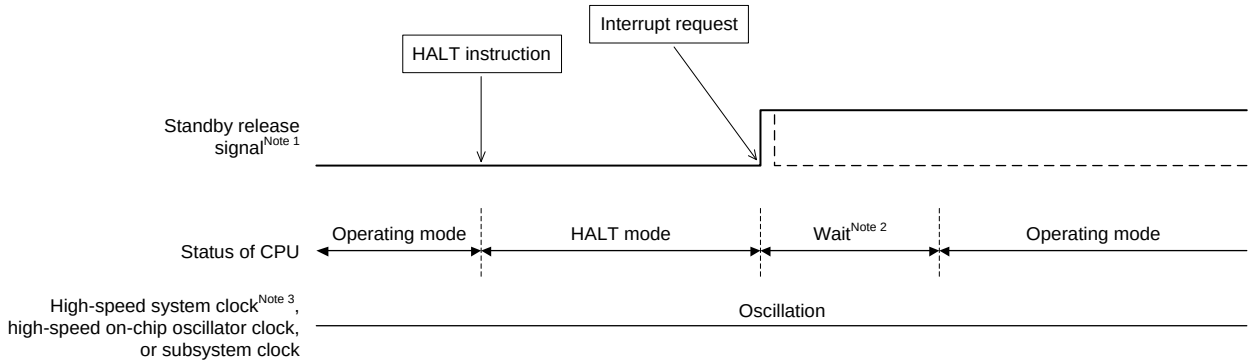
(2) HALT mode release

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 17-1. HALT Mode Release by Interrupt Request Generation



Note 1. For details of the standby release signal, see **Figure 16-1 Basic Configuration of Interrupt Function**.

Note 2. Wait time for HALT mode release:

- When vectored interrupt servicing is carried out:
  Main system clock: 24 or 25 clocks
  Subsystem clock (RTCLPC = 0): 10 or 11 clocks
  Subsystem clock (RTCLPC = 1): 11 or 12 clocks
- When vectored interrupt servicing is not carried out:
  Main system clock: 18 or 19 clocks
  Subsystem clock (RTCLPC = 0): 4 or 5 clocks
  Subsystem clock (RTCLPC = 1): 5 or 6 clocks

Note 3. Other than 10-pin products.

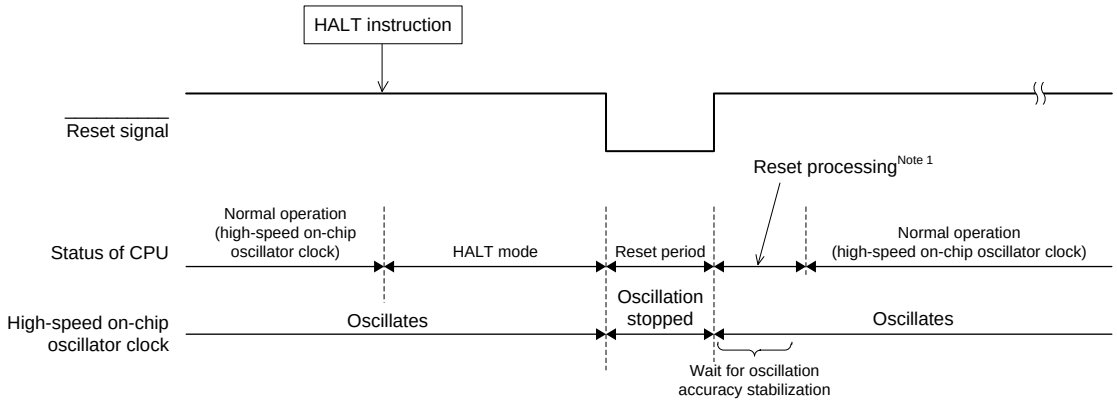
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

b) HALT mode release by reset signal generation

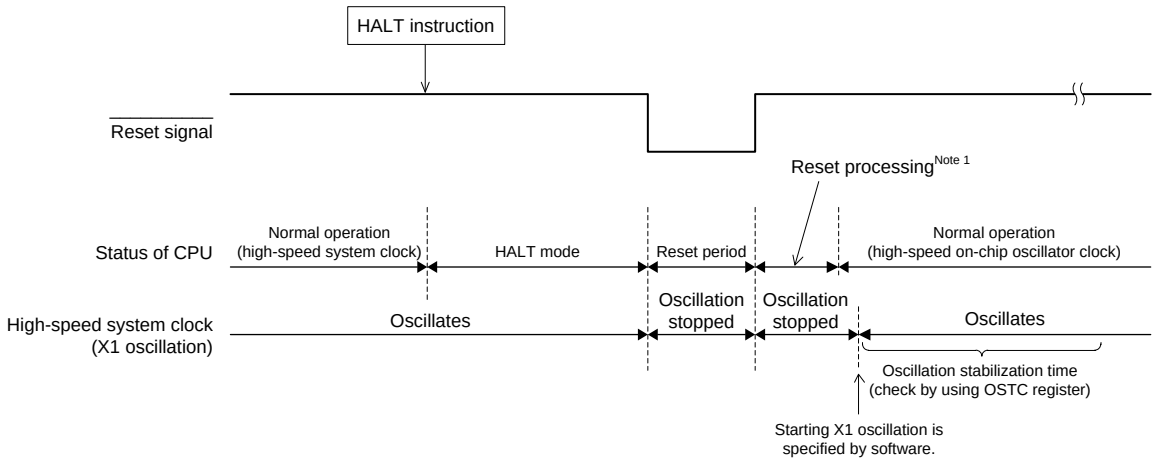
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 17-2. HALT Mode Release by Reset Signal Generation

(1) When high-speed on-chip oscillator clock is used as CPU clock



(2) When high-speed system clock is used as CPU clock (Other than 10-pin products)



Note 1. For the reset processing time, see **CHAPTER 18 RESET FUNCTION**. For the reset processing time of the SPOR circuit, see **CHAPTER 19 SELECTABLE POWER-ON-RESET CIRCUIT**.

17.3.2 STOP mode

(1) STOP mode setting and operating status

The STOP mode is set by executing the STOP instruction. It can be set only when the CPU clock before the setting was the 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 status in the STOP mode are shown below.

Table 17-3. Operating Status in STOP Mode (1/2)

STOP Mode Setting / Item			When STOP Instruction is Executed While CPU is Operating at Main System Clock		
			When CPU is Operating on High-speed On-chip Oscillator Clock (f_{IH})	When CPU is Operating on X1 Clock (f_X)	When CPU is Operating on External Main System Clock (f_{EX})
System clock			Clock supply to the CPU is stopped		
	Main system clock Note 2	f_{IH}	Stopped		
		f_X			
		f_{EX}			
	Subsystem clock Note 2	f_{XT}	Status before STOP mode was set is retained		
		f_{EXS}			
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and bit 4 (WUTMMCK0) of operation speed mode control register (OSMC) • WUTMMCK0 = 1: Oscillates • WUTMMCK0 = 0 and WDTON = 0: Stops • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 1: Oscillates • WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 0: Stops		
CPU			Operation stopped		
Code flash memory					
Data flash memory					
RAM					
Illegal-memory access detection function					
Port (latch)			Status before STOP mode was set is retained		
Timer array unit			Operation disabled		
Real-time clock 2			Operable		
12-bit interval timer					

Table 17-3. Operating Status in STOP Mode (2/2)

STOP Mode Setting / Item	When STOP Instruction is Executed While CPU is Operating at Main System Clock		
	When CPU is Operating on High-speed On-chip Oscillator Clock (f_{IH})	When CPU is Operating on X1 Clock (f_X)	When CPU is Operating on External Main System Clock (f_{EX})
Watchdog timer	Set by bit 0 (WDSTBYON) of option byte (000C0H) WDSTBYON = 0: Operation stopped WDSTBYON = 1: Operation continues		
Clock output/buzzer output	Operates when the subsystem clock is selected as the clock source for counting and the RTCLPC bit is 0 (operation is disabled when a clock other than the subsystem clock is selected and the RTCLPC bit is not 0).		
A/D converter	Operation disabled		
Comparator	Operable (only when the digital filter is not in use)		
Serial array unit (SAU)	Operation disabled		
Serial interface (IICA)	Wakeup by address match operable		
Capacitive touch sensing unit (CTSU)	Operation stopped		
Selectable power-on-reset function	Operable		
External interrupt			
Internal reset by data retention power supply voltage	Operating		
CRC operation function (General-purpose CRC)	Operation stopped		
RAM parity error detection function			
RAM guard function			
SFR guard function			
Illegal-memory access detection 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_X: X1 clock[Note 1]
f_{EX}: External main system cloc[Note 1]
f_{XT}: XT1 clock[Note 1]
f_{EXS}: External subsystem clock[Note 1]

Note 1. Other than 10-pin products.

Note 2. X1 oscillator and XT1 oscillator cannot operate at the same time because X1 and X2 pins of the X1 oscillator are shared with XT1 and XT2 pins of the XT1 oscillator.

(2) STOP mode release

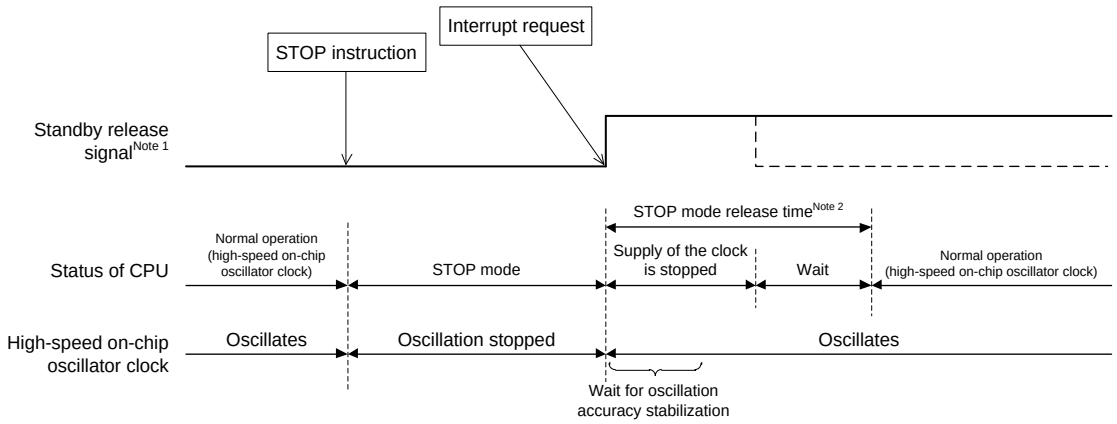
The STOP mode can be released by the following two sources.

a) STOP mode 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 17-3. STOP Mode Release by Interrupt Request Generation (1/3)

(1) When high-speed on-chip oscillator clock is used as CPU clock



Note 1. For details of the standby release signal, see **Figure 16-1 Basic Configuration of Interrupt Function**.

Note 2. STOP mode release time: Supply of the clock is stopped: 27 μs (TYP.)

[Wait]

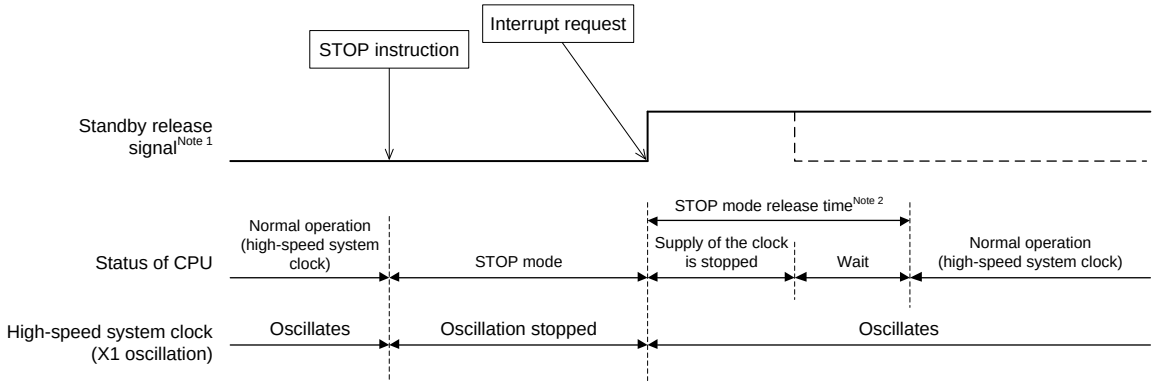
- When vectored interrupt servicing is carried out: 8 clocks
- When vectored interrupt servicing is not carried out: 2 clocks

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 17-3. STOP Mode Release by Interrupt Request Generation (2/3)

(2) When high-speed system clock (X1 oscillation) is used as CPU clock (Other than 10-pin products)



Note 1. For details of the standby release signal, see **Figure 16-1 Basic Configuration of Interrupt Function**.

Note 2. STOP mode release time: Whichever is longer, 27 μs (TYP.) or the oscillation stabilization time (set by OSTS)

[Wait]

- When vectored interrupt servicing is carried out: 10 or 11 clocks
- When vectored interrupt servicing is not carried out: 4 or 5 clocks

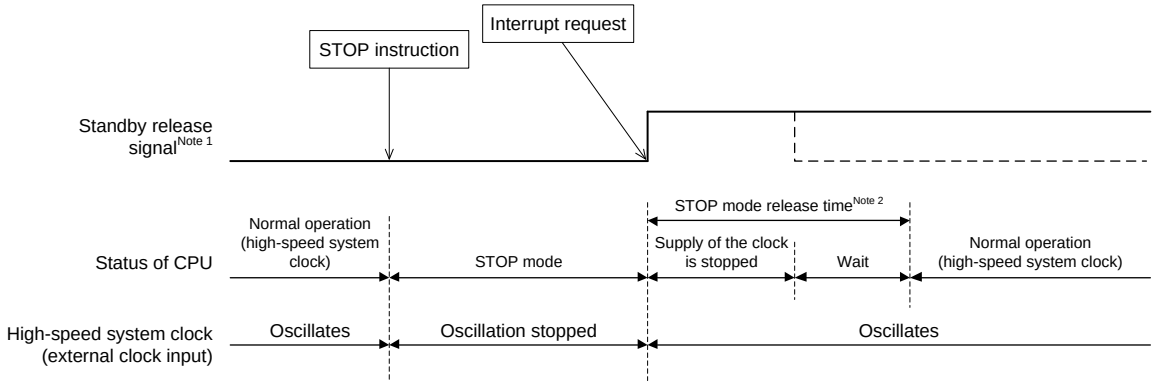
Caution To reduce the oscillation stabilization time after release from the STOP mode while CPU operates based on the high-speed system clock (X1 oscillation), switch the clock to the high-speed on-chip oscillator clock temporarily before executing the STOP instruction.

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.

Figure 17-3. STOP Mode Release by Interrupt Request Generation (3/3)

(3) When high-speed system clock (external clock input) is used as CPU clock (Other than 10-pin products)



Note 1. For details of the standby release signal, see **Figure 16-1 Basic Configuration of Interrupt Function**.

Note 2. STOP mode release time
Supply of the clock is stopped: 27 μs

[Wait]

- When vectored interrupt servicing is carried out: 8 clocks
- When vectored interrupt servicing is not carried out: 2 clocks

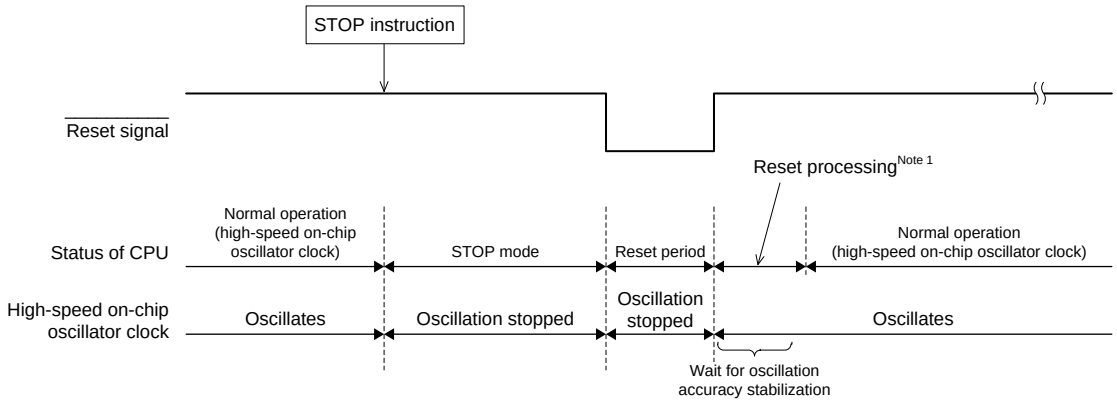
Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

b) STOP mode release by reset signal generation

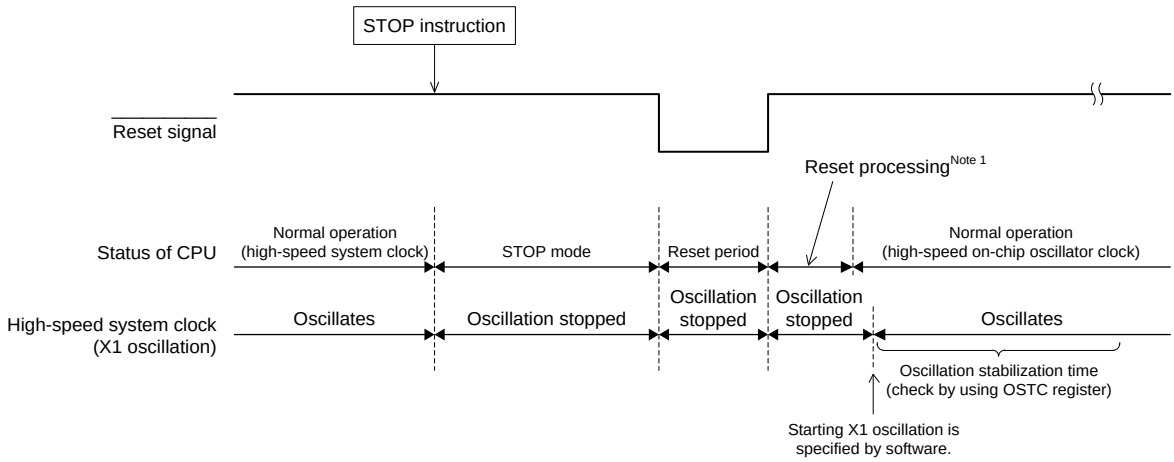
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 17-4. STOP Mode Release by Reset Signal Generation

(1) When high-speed on-chip oscillator clock is used as CPU clock



(2) When high-speed system clock is used as CPU clock (Other than 10-pin products)



Note 1. For the reset processing time, see **CHAPTER 18 RESET FUNCTION**. For the reset processing time of the SPOR circuit, see **CHAPTER 19 SELECTABLE POWER-ON-RESET CIRCUIT**.

CHAPTER 18 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 selectable power-on-reset (SPOR) circuit

(4) Internal reset by execution of illegal instruction[Note 1]

(5) Internal reset by data retention power supply voltage

(6) Internal reset by illegal-memory access

(7) Internal reset by the RAM parity error

External and internal resets start program execution from the address stored at 0000H and 0001H when the reset signal is generated.

Note 1. The illegal instruction is generated when instruction code FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the on-chip debug emulator.

Caution 1. For an external reset, set the PORTSELB bit of the user option byte (000C1H) to 1 so that the P125 pin operates as $\overline{\text{RESET}}$, and 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, and then apply power supply. The $\overline{\text{RESET}}$ pin must be kept low for at least 10 μs during the period in which the supply voltage is within the operating range shown in 26.4 AC Characteristics and 27.4 AC Characteristics before inputting a high level to the $\overline{\text{RESET}}$ pin.)

Caution 2. During reset input, the X1clock[Note 2], high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock stop oscillating, and external main system clock[Note 2] input is invalid.
Only during an internal reset by data retention power supply voltage, the X1 clock and external subsystem clock stop oscillating or become invalid.

Caution 3. The port pin becomes the following status because each SFR and 2nd SFR are initialized after reset.

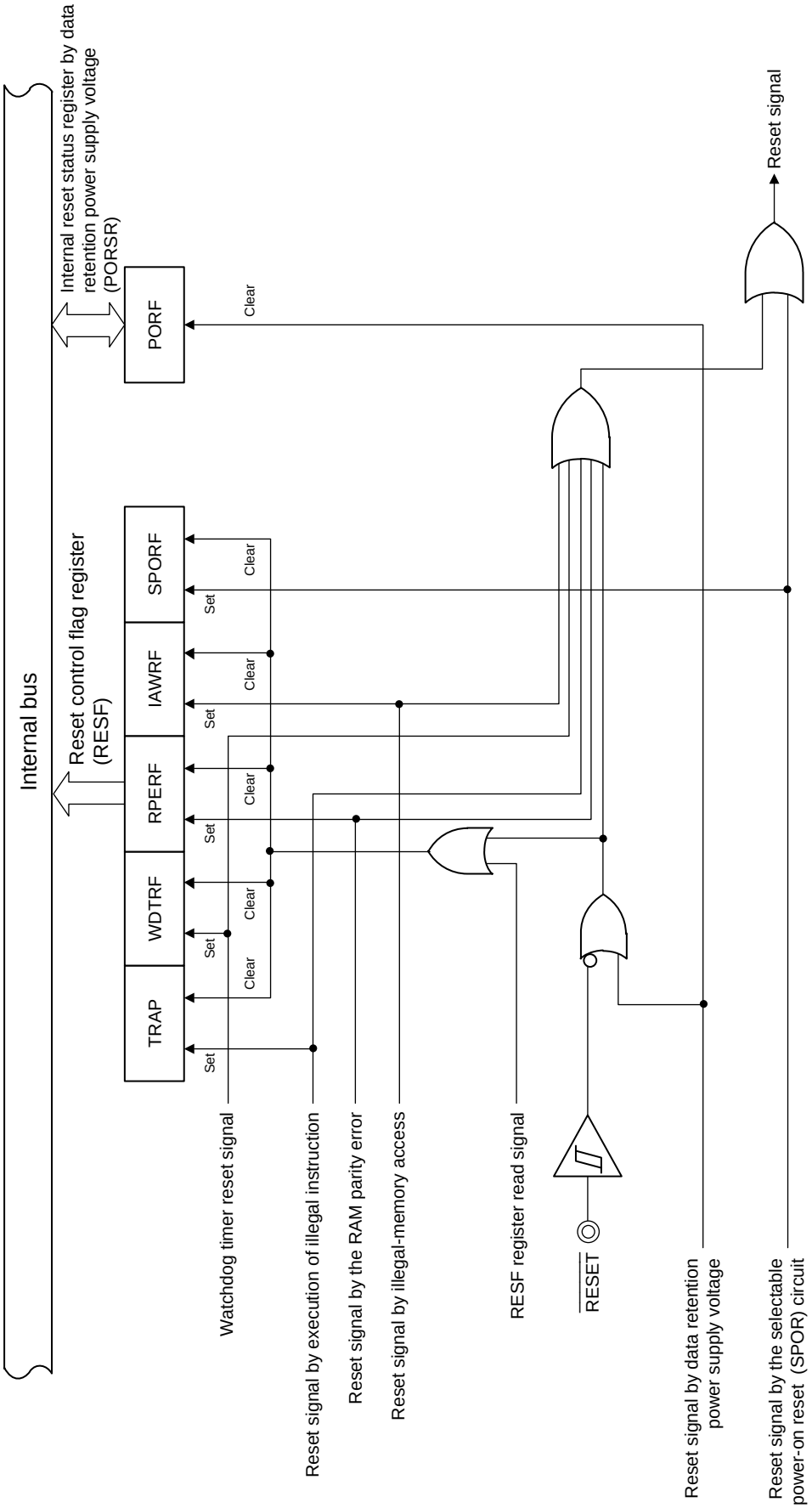
- **P40**
  High-impedance during external reset period or reset period by the data retention power supply voltage. High level during other types of reset or after receiving a reset (connected to the internal pull-up resistor).
- **P125**
  Low level during external reset period (low level input to $\overline{\text{RESET}}$ pin). High level during other types of reset period or after receiving a reset (connected to the internal pull-up resistor).
- **Ports other than P40 and P125**
  High-impedance during reset period or after receiving a reset.

Caution 4. The registers below are only initialized by an internal reset by data retention power voltage.

- **Registers related to RTC2**
- **The EXCLKS, OSCSELS, XTSEL, AMPHS1, and AMPHS0 bits of the CMC register of clock generator**

Note 2. Other than 10-pin products.

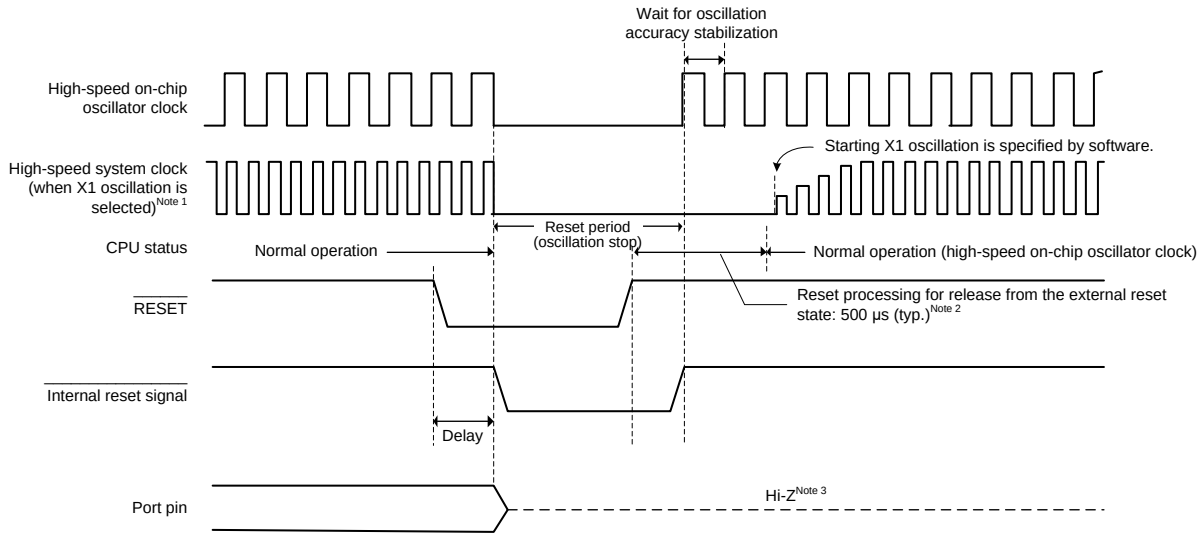
Figure 18-1. Block Diagram of Reset Function



18.1 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 18-2. Timing of Reset by $\overline{\text{RESET}}$ Input



Note 1. Other than 10-pin products.

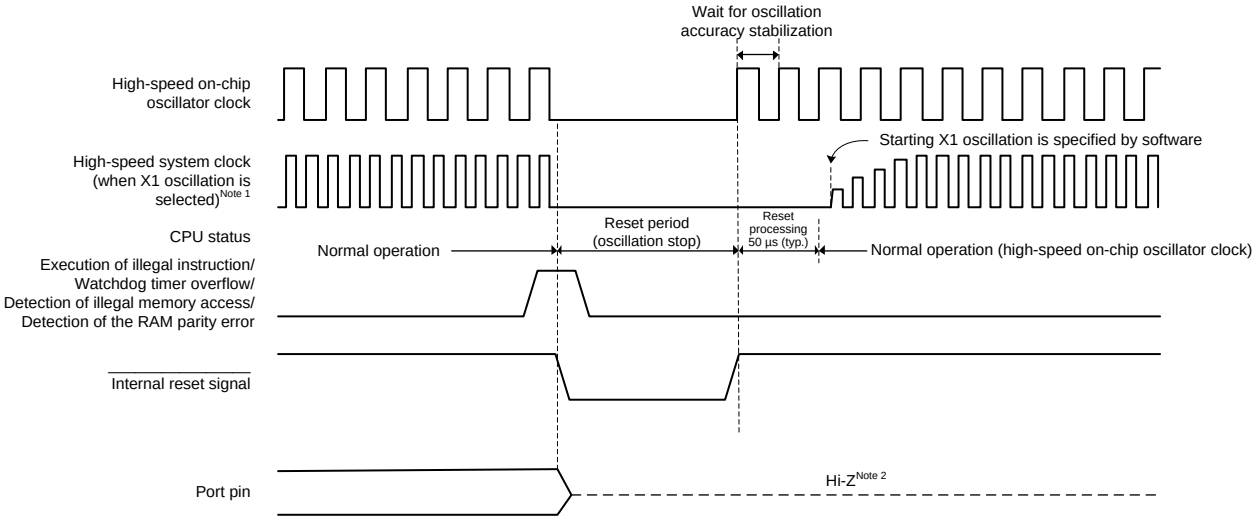
Note 2. After power is supplied, an SPOR reset processing time of 3.01 ms (MAX.) is required before reset processing starts after release of the external reset.

Note 3. Status of port pin P40 is as follows.

- High-impedance during external reset period or reset period by the data retention power supply voltage
- High level after receiving a reset (connected to the internal pull-up resistor)

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 the RAM parity error, or detection of illegal memory access. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts.

Figure 18-3. Timing of Reset Due to Watchdog Timer Overflow, Execution of Illegal Instruction, Detection of the RAM Parity Error, or Detection of Illegal Memory Access



Note 1. Other than 10-pin products.

Note 2. Statuses of port pins P40 and P125 pins are as follows.

- High level during reset period or after receiving a reset (connected to the internal pull-up resistor).

Remark For the reset timing due to the voltage detection by the selectable power-on-reset (SPOR) circuit, see **CHAPTER 19 SELECTABLE POWER-ON-RESET CIRCUIT**.

18.2 States of Operation During Reset Periods

Table 18-1 shows the states of operation during reset periods. **Table 18-2** shows the state of the hardware after acceptance of a reset.

Table 18-1. States of Operation During Reset Period

Item			During Reset Period
System clock			Clock supply to the CPU is stopped.
	Main system clock	f_{IH}	Operation stopped
		f_X Note 1	Operation stopped (the X1 and X2 pins are input port mode)
		f_{EX} Note 1	Clock input invalid (the pin is input port mode)
	Subsystem clock	f_{XT} Note 1	Operable (operation is stopped in the case of an internal reset by data retention power supply voltage, and the XT1 and XT2 pins are input port mode)
		f_{EXS} Note 1	Operable (operation is stopped in the case of an internal reset by data retention power supply voltage, and the EXCLKS 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 2
Real-time clock 2			Other than an internal reset by data retention power supply voltage: Operable An internal reset by data retention power supply voltage: Only calendar registers retain data
Timer array unit			Operation stopped
12-bit interval timer			
Watchdog timer			
Clock output/buzzer output			
A/D converter			
Comparator			
Serial array unit (SAU)			
Serial interface (IICA)			
Capacitive touch sensing unit (CTSU)			
Internal reset by data retention power supply voltage			Detection operation possible
Selectable power-on-reset function			Detection operation possible
External interrupt			Operation stopped
Illegal-memory access detection function			
CRC operation function (General-purpose CRC)			
RAM parity error detection function			
RAM guard function			
SFR guard function			

Note 1. Other than 10-pin products. X1 oscillator and XT1 oscillator cannot operate at the same time because X1 and X2 pins of the X1 oscillator are shared with XT1 and XT2 pins of the XT1 oscillator.

Note 2. Statuses of P40 and P125 pins are as follows

- P40
  High-impedance during external reset period or reset period by the data retention power supply voltage.
  High level during other types of reset or after receiving a reset (connected to the internal pull-up resistor).
- P125
  Low level during external reset period (low level input to $\overline{\text{RESET}}$ pin). High level during other types of reset period or after receiving a reset (connected to the internal pull-up resistor).

Remark f_{IH}: Highspeed on-chip oscillator clock
f_X: X1 clock
f_{EX}: External main system clock
f_{XT}: XT1 oscillation clock
f_{EXS}: External subsystem clock
f_{IL}: Low-speed on-chip oscillator clock

Table 18-2. State of Hardware After Receiving a Reset Signal

Hardware		After Reset Acknowledgment[Note 1]
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 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

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**.

18.3 Register for Confirming Reset Source

18.3.1 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.

The external reset, a reset by the data retention lower limit voltage, and reading the RESF register clear TRAP, WDTRF, RPERF, IAWRF, and SPORF flags.

Figure 18-4. 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	SPORF

TRAP	Internal reset request by execution of illegal instruction[Note 2]
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

WDTRF	Internal reset request by watchdog timer (WDT)
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

RPERF	Internal reset request by RAM parity
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.

SPORF	Internal reset request by selectable power-on reset (SPOR) circuit
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.

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 on-chip debug emulator.

Caution 1. Do not read data by a 1-bit memory manipulation instruction.

Caution 2. When enabling the 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 the RAM parity error resets (RPERDIS = 0). For details, see 20.3.2 RAM parity error detection function.

The status of the RESF register when a reset request is generated is shown in **Table 18-3**.

Table 18-3. RESF Register Status When Reset Request Is Generated

Reset Source / Flag	$\overline{\text{RESET}}$ Input	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by RAM Parity Error	Reset by Illegal-Memory Access	Reset by SPOR	Reset by Data Retention Lower Limit Voltage
TRAP	Cleared (0)	Set (1)	Held	Held	Held	Held	Cleared (0)
WDTRF		Held	Set (1)	Held	Held	Held	
RPERF		Held	Held	Set (1)	Held	Held	
IAWRF		Held	Held	Held	Set (1)	Held	
SPORF		Held	Held	Held	Held	Set (1)	

18.3.2 Internal Reset Status Register by Data Retention Power Supply Voltage (PORSR)

The PORSR register is used to check the occurrence of an internal reset by data retention power supply voltage.

Writing 1 to bit 0 (PORF) of the PORSR register enables this function. Writing 0 disables this function.

Write 1 to the PORF bit in advance to enable checking of the occurrence of an internal reset by data retention power supply voltage.

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

Occurrence of an internal reset signal by data retention power supply voltage clears this register to 00H.

Caution 1. The PORSR register is reset only by an internal reset by data retention power supply voltage; it retains the value when a reset caused by another factor occurs.

Caution 2. If the PORF bit is set to 1, it guarantees that no internal reset by data retention power supply voltage has occurred, but it does not guarantee that the RAM value is retained.

Figure 18-5. Format of Internal Reset Status Register by Data Retention Power Supply Voltage (PORSR)

Address: F00F9H After reset: 00H R/W

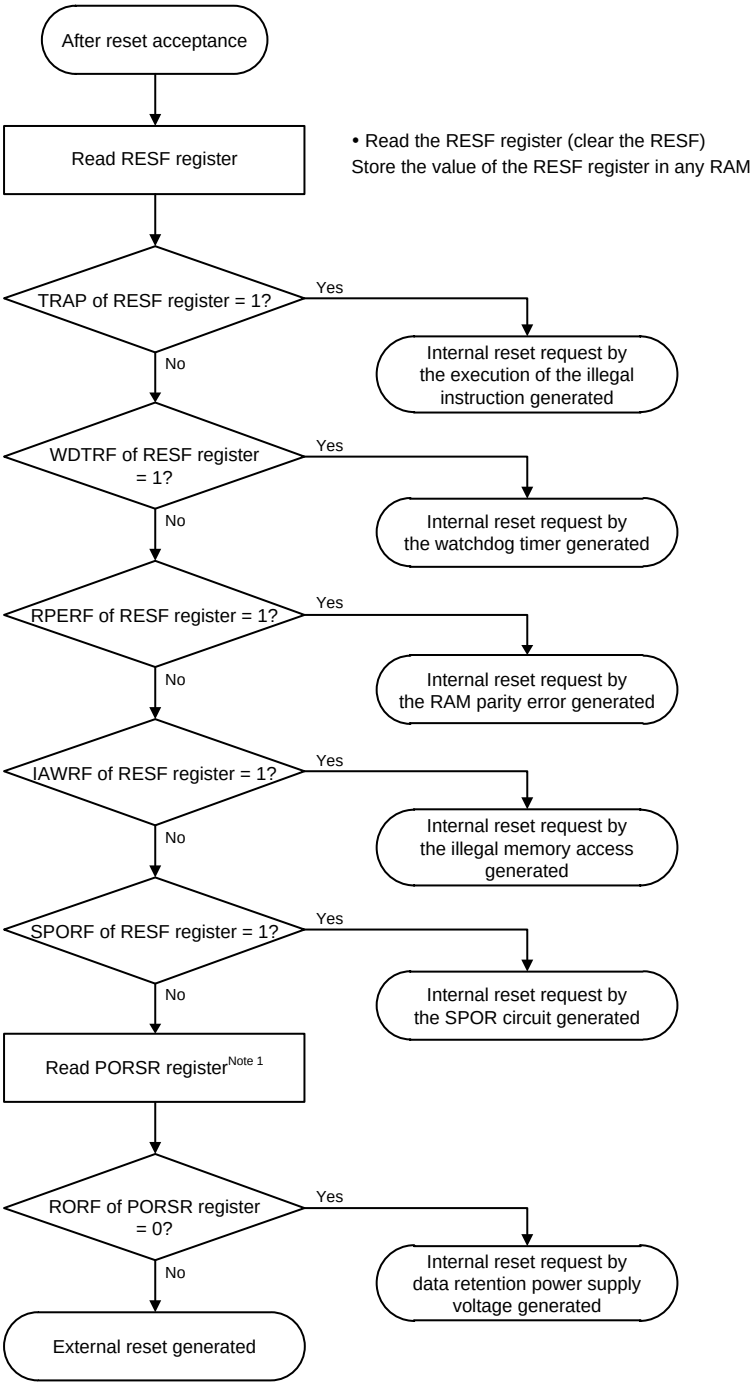
Symbol	7	6	5	4	3	2	1	0
PORSR	0	0	0	0	0	0	0	PORF

PORF	Checking Occurrence of an Internal Reset by Data Retention Power Supply Voltage
0	A value 1 has not been written, or an internal reset by data retention power supply voltage has occurred.
1	No internal reset by data retention power supply voltage has occurred.

The RESF register is automatically cleared when it is read by an 8-bit memory manipulation instruction.

Figure 18-6 shows the procedure for checking a reset source.

Figure 18-6. Example of Procedure for Checking Reset Source



* The flow described above is an example of the procedure for checking.

Note 1. Before receiving a reset signal, write 1 to bit 0 (PORF) of the PORSR register.

CHAPTER 19 SELECTABLE POWER-ON-RESET CIRCUIT

19.1 Functions of Selectable Power-on-reset Circuit

The selectable power-on-reset (SPOR) circuit 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_{SPOR}) ($V_{DD} \geq V_{SPOR}$).
- The SPOR circuit compares the supply voltage (V_{DD}) with the detection voltage (V_{SPDR}), and generates an internal reset signal when $V_{DD} < V_{SPDR}$.
- The detection level for the power supply detection voltage (V_{SPOR}, V_{SPDR}) can be selected by using the option byte (000C1H) as one of 4 levels (for details, see **21.2 Format of User Option Byte**).

Bit 0 (SPORF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of the RESF register, see **CHAPTER 18 RESET FUNCTION**.

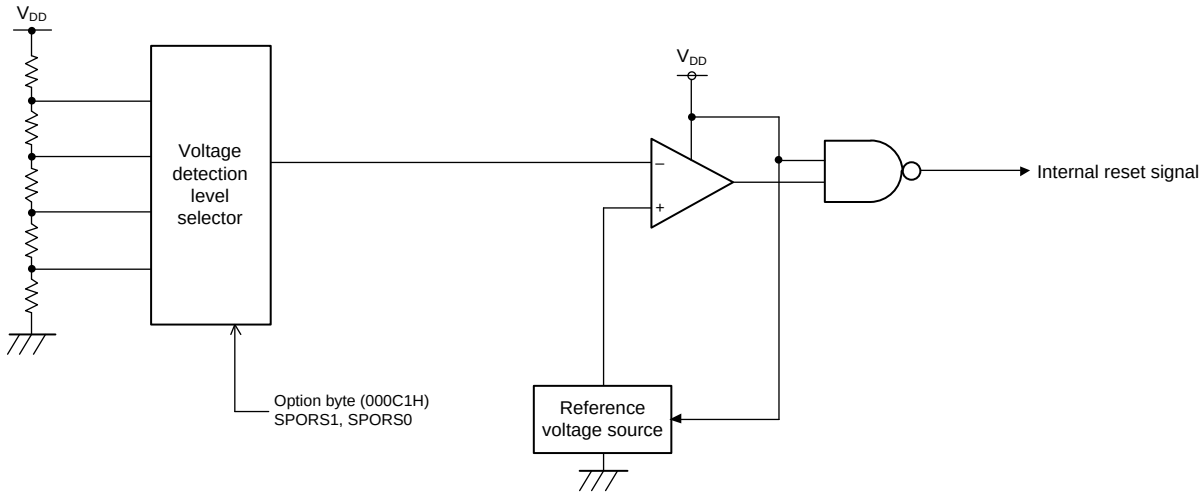
Caution **The values of all flags in the reset control flag register (RESF) are retained until V_{DD} reaches data retention lower limit voltage.**

Remark V_{SPOR}: SPOR power supply rise detection voltage
V_{SPDR}: SPOR power supply fall detection voltage
For details, see **26.6.4 SPOR circuit characteristics** and **27.6.4 SPOR circuit characteristics**.

19.2 Configuration of Selectable Power-on-reset Circuit

The block diagram of the selectable power-on-reset circuit is shown in **Figure 19-1**.

Figure 19-1. Block Diagram of Selectable Power-on-reset Circuit



19.3 Operation of Selectable Power-on-reset Circuit

Specify the voltage detection level by using the option byte 000C1H.

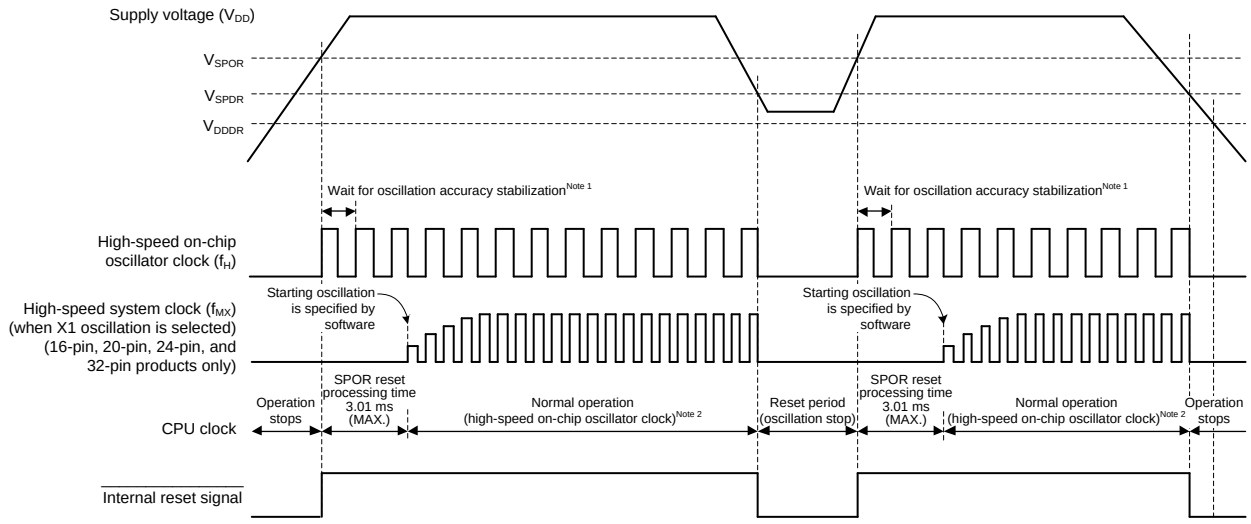
The internal reset signal is generated at power on.

The internal reset status is retained until the supply voltage (V_{DD}) exceeds the voltage detection level (V_{SPOR}). The internal reset is cleared when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{SPOR}).

The internal reset is generated when the supply voltage (V_{DD}) drops lower than the voltage detection level (V_{SPDR}).

Figure 19-2 shows the timing of generation of the internal reset signal by the selectable power-on-reset circuit.

Figure 19-2. Timing of Internal Reset Signal Generation



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 can be selected as the CPU clock (16-pin, 20-pin, 24-pin, and 32-pin products only).
To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time.

Remark V_{SPOR}: SPOR power supply rise detection voltage
V_{SPDR}: SPOR power supply fall detection voltage
V_{DDDR}: Data retention lower limit voltage

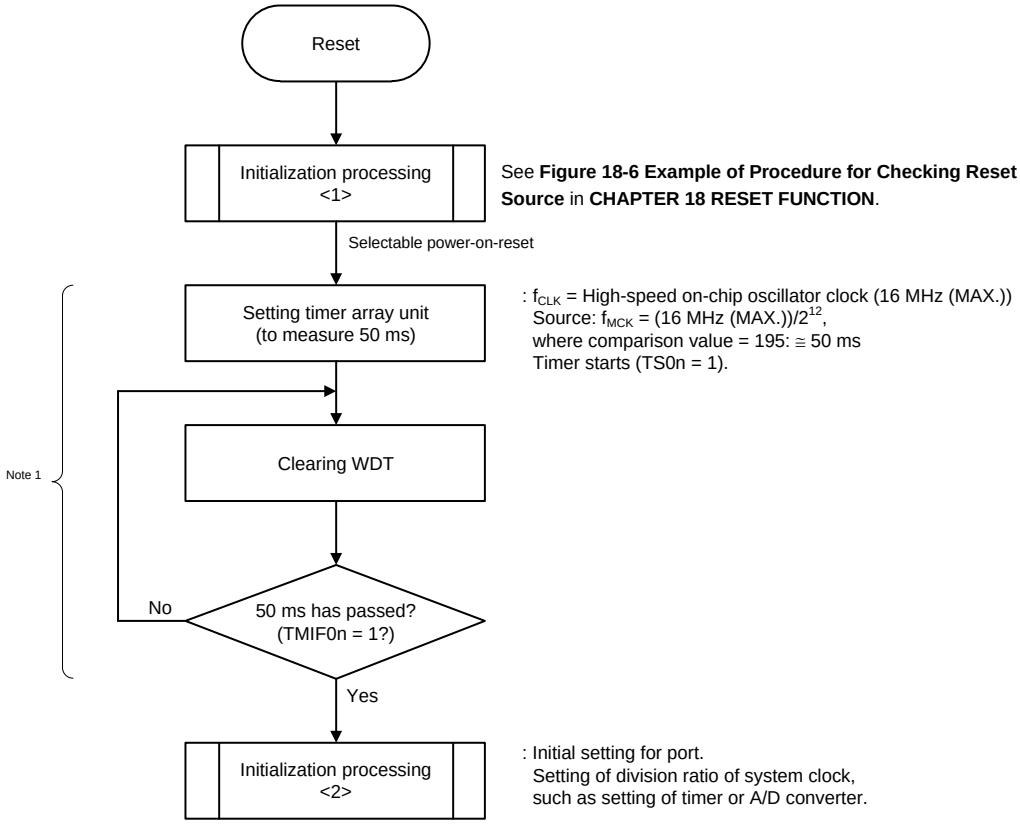
19.4 Cautions for Selectable Power-on-reset Circuit

In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the SPOR detection voltage (V_{SPOR}, V_{SPDR}), 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 timer and etc., and then initialize the ports.

Figure 19-3. Example of Software Processing When Supply Voltage Fluctuation is 50 ms or Less in Vicinity of the Voltage Detection Level



Note 1. If reset is generated again during this period, initialization processing <2> is not started.

Remark n: Channel number
For 10-pin products, n = 0, 1, and 2; for 16-pin products, n = 0 to 5, and 7; and for 20-pin, 24-pin, and 32-pin products, n = 0 to 7.

CHAPTER 20 SAFETY FUNCTIONS

20.1 Overview of Safety Functions

The RL78/G16 provides the following safety functions to comply with the IEC60730 safety standards.

These safety functions enable the microcontroller to self-diagnose abnormalities and safely stop operating if an abnormality is detected.

1) Flash memory CRC operation (general-purpose CRC)

This detects data errors in the flash memory by performing CRC operations.

- General-purpose CRC:
  This can be used for checking various data in addition to the code flash memory area while the CPU is running.

2) RAM parity error detection

This detects parity errors when reading RAM data.

3) RAM guard function

This prevents rewriting of data in RAM due to incorrect CPU operations.

4) SFR guard function

This prevents rewriting of data in the SFRs due to incorrect CPU operations.

5) Invalid memory access detection

This detects unauthorized access to invalid memory areas (such as areas where no memory is allocated and areas to which access is restricted).

6) Frequency detection

This function allows a self-check of the CPU/peripheral hardware clock frequencies using the timer array unit.

7) Testing of the A/D converter

This test checks whether or not the A/D converter is operating normally by converting the A/D converter's positive and negative reference voltages, analog input channels (ANI), temperature sensor output voltage, and internal reference voltage.

Remark For usage examples of the safety functions complying with the IEC60730 safety standards, refer to RL78 Family IEC60730/60335 self test library application notes.

20.2 Registers Used for Safety Functions

The following registers are used for the respective safety functions.

Register Name	Safety Function
• CRC input register (CRCIN) • CRC data register (CRCD)	CRC operation (general-purpose CRC)
• RAM parity error control register (RPECTL)	RAM parity error detection
• Invalid memory access detection control register (IAWCTL)	RAM guard function
	SFR guard function
	Invalid memory access detection
• Timer input select register 0 (TIS0)	Frequency detection
• A/D test register (ADTES)	Testing of the A/D converter

The content of each register is described in **20.3 Operation of Safety Functions**.

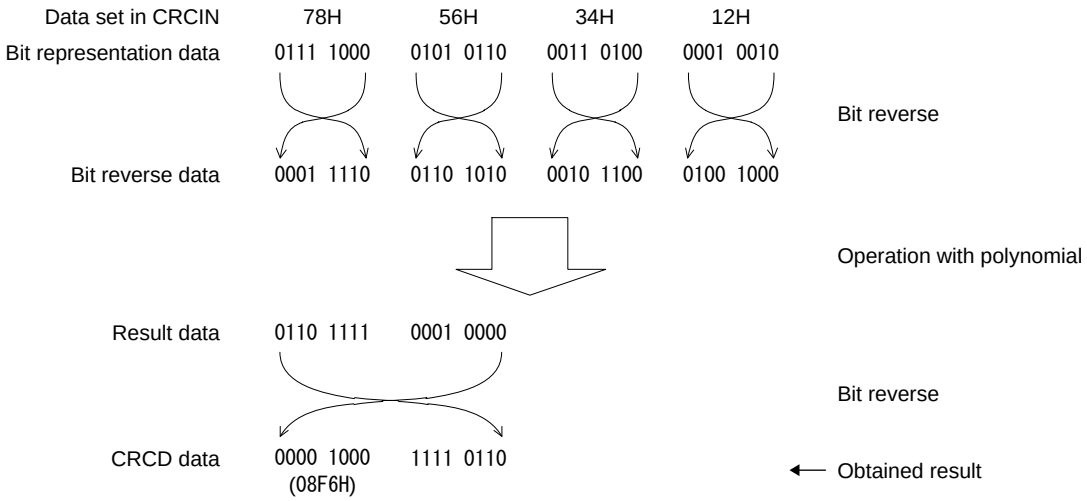
20.3 Operation of Safety Functions

20.3.1 CRC operation (general-purpose CRC)

The general-purpose CRC handles CRC operation as a peripheral module while the CPU is operating. The general-purpose CRC can be used for checking various data in addition to the code flash memory area. The data to be checked can be specified by using software (a user-created program). The CRC calculation function in the HALT mode can be used only during DMA transfer.

The general-purpose CRC can operate either in the main system clock operation mode or in the subsystem clock operation mode.

The CRC generator polynomial used is "$X^{16} + X^{12} + X^5 + 1$" of CRC-16-CCITT. The data to be input is inverted in bit order and then calculated to allow for LSB-first communication. For example, if the data 12345678H is sent from the LSB, values are written to the CRCIN register in the order of 78H, 56H, 34H, and 12H, enabling a value of 08F6H to be obtained from the CRCD register. This is the result obtained by executing a CRC operation on the bit rows shown below, which consist of the data 12345678H inverted in bit order.



Caution 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.

20.3.1.1 CRC input register (CRCIN)

The CRCIN register is an 8-bit register to set the CRC operation data of the general-purpose CRC.

The possible setting range is 00H to FFH.

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

Reset signal generation clears this register to 00H.

Figure 20-1. 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

20.3.1.2 CRC data register (CRCD)

This register holds the CRC operation result of the general-purpose CRC.

The setting range is 0000H to FFFFH.

After one clock cycle of CPU/peripheral hardware clock (f_{CLK}) has elapsed from the time the CRCIN register was written, the CRC operation result is stored in 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 20-2. Format of CRC Data Register (CRCD)

Address: F02FAH After reset: 0000H R/W

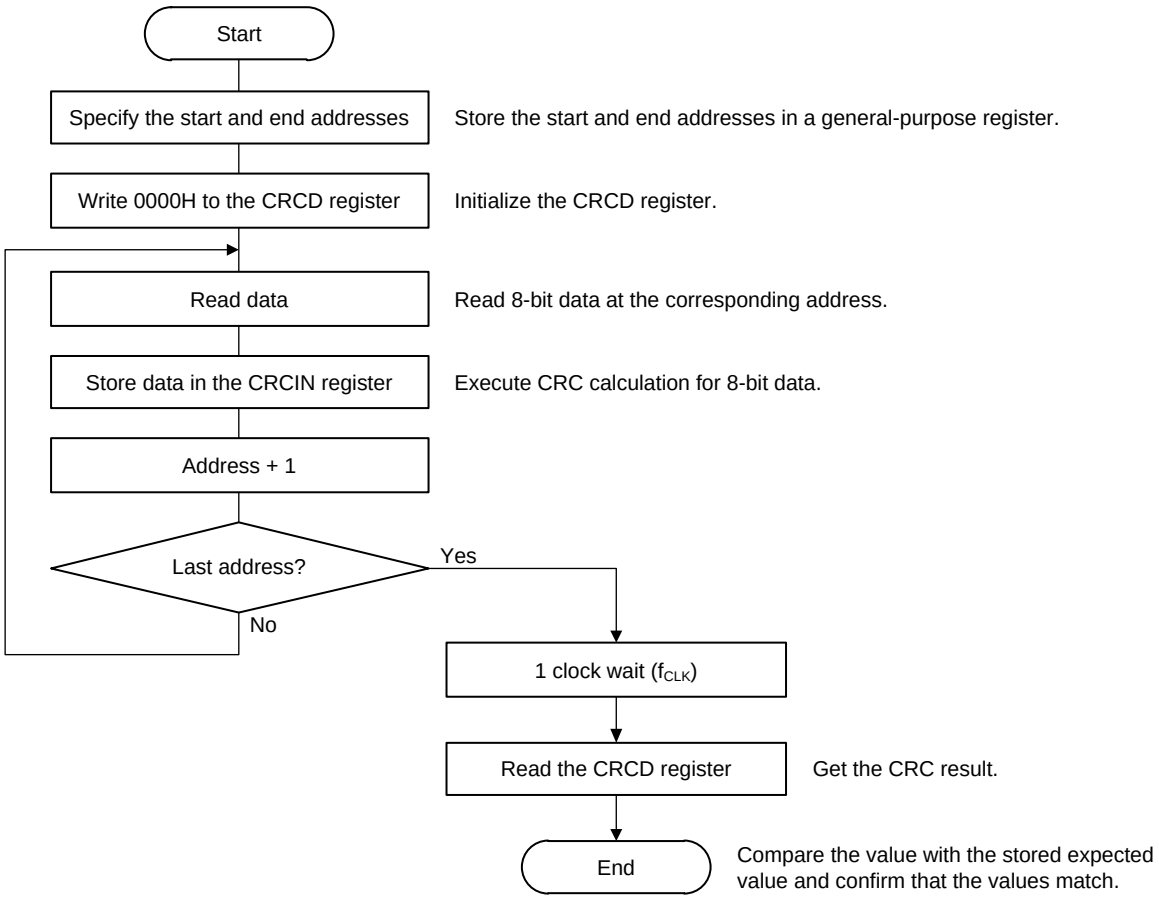
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCD																

Caution 1. Read the value written to CRCD register before writing to CRCIN register.

Caution 2. If conflict between writing a value and storing the operation result in the CRCD register occurs, the writing is ignored.

<Operation flow>

Figure 20-3. Flowchart of CRC Operation (General-Purpose CRC)



20.3.2 RAM parity error detection

The IEC60730 standard mandates the checking of RAM data. A single-bit parity bit is therefore added to all 8-bit data in the RL78/G16's RAM. By using this RAM parity error detection, 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.

20.3.2.1 RAM parity error control register (RPECTL)

This register is used to control the 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 20-4. 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	Enables parity error resets.
1	Disables parity error resets.

RPEF	Parity error status flag
0	No parity error has occurred.
1	A parity error has occurred.

Caution The parity bit is appended when data 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.

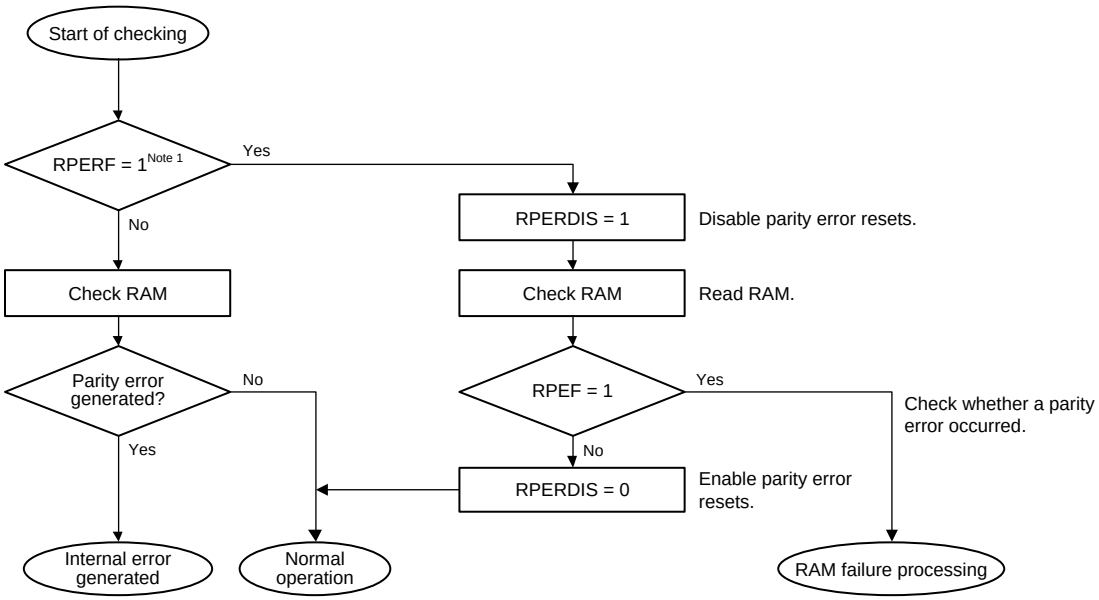
Remark 1. Parity error resets are enabled by default (RPERDIS = 0).

Remark 2. Even if parity error resets are disabled (RPERDIS = 1), the RPEF flag will be set (1) if a parity error occurs. If parity error resets are enabled (RPERDIS = 0) with RPEF set to 1, a parity error reset is generated when the RPERDIS bit is cleared to 0.

Remark 3. The RPEF flag in the RPECTL register is set (1) when a parity error occurs 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. The general-purpose registers are not included for RAM parity error detection.

Figure 20-5. Flowchart of RAM Parity Checking



Note 1. To check the state following an internal reset triggered by a RAM parity error, see **CHAPTER 18 RESET FUNCTION**.

20.3.3 RAM guard function

The RL78/G16 has functionality to protect data in the specified memory space.

Enabling this function disables writing to the specified area of the RAM. Reading from the specified area is possible as usual.

20.3.3.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access, and protection of the RAM and SFRs.

Use the GRAM1 and GRAM0 bits to protect the RAM.

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

Reset signal generation clears this register to 00H.

Figure 20-6. 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	0	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GRAM1	GRAM0	Protected area in RAM[Note 1]
0	0	Disabled. Writing to the RAM is allowed.
0	1	128 bytes from the base address of the RAM
1	0	256 bytes from the base address of the RAM
1	1	512 bytes from the base address of the RAM

Note 1. The base address of the RAM differs depending on the size of the RAM in the product.

20.3.4 SFR guard function

The RL78/G16 provides functionality to protect the data in the control registers for use with the ports, interrupts, clock control, and RAM parity error detection.

Enabling this function disables writing to the protected area of the SFRs. Reading from the protected area is possible as usual.

20.3.4.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and the protection of the RAM and SFRs.

Use the GPORT, GINT, and GCSC bits to protect the SFRs.

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

Reset signal generation clears this register to 00H.

Figure 20-7. 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	0	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GPORT	Protection of port control registers
0	Disabled. Reading from and writing to the port control registers are possible.
1	Enabled. Writing to the port control registers is not allowed. Reading is possible. [Protected SFRs] PMxx, PUxx, POMxx, PMCxx, PIOR0 to PIOR6, TSSEL0, 1, VTSEL Note 1

GINT	Protection of interrupt control registers
0	Disabled. Reading from and writing to the interrupt control registers are allowed.
1	Enabled. Writing to the interrupt control registers is not allowed. Reading is possible. [Protected SFRs] IFxx, MKxx, PRxx, EGPx, EGNx

GCSC	Protection of clock and RAM parity error detection control registers
0	Disabled. Reading from and writing to the clock and RAM parity error detection control registers are allowed.
1	Enabled. Writing to the clock and RAM parity error detection control registers is not allowed. Reading is possible. [Protected SFRs] CMC, CSC, OSTS, CKC, PERx, OSMC, RPECTL

Note 1. The port registers (Pxx) are not protected.

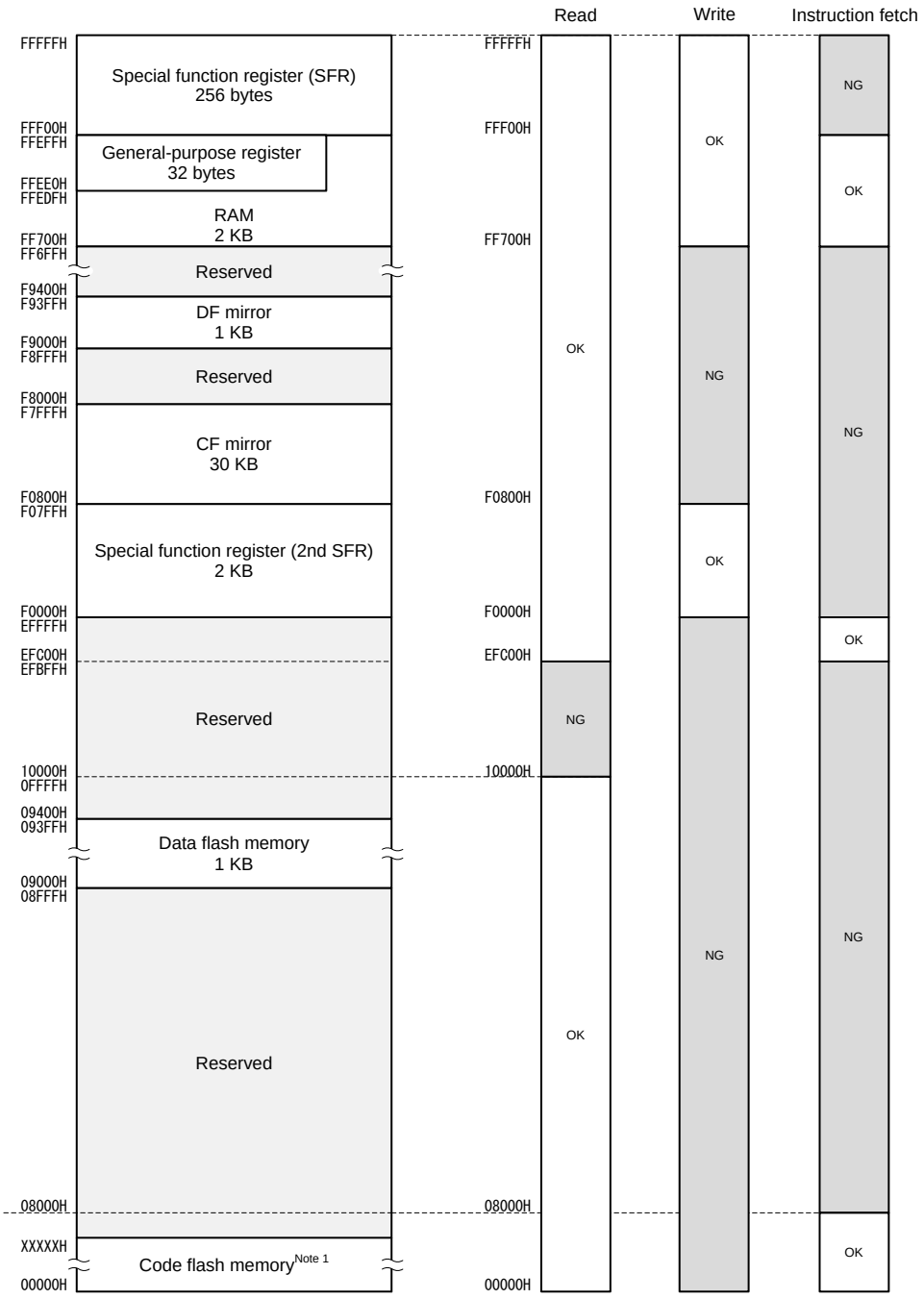
20.3.5 Invalid memory access detection

The IEC60730 standard mandates checking that the CPU and interrupts are operating correctly

The RL78/G16 provides functionality to trigger a reset when an invalid memory area is accessed.

Access to the areas indicated as “Not allowed” in **Figure 20-8** is detected as invalid.

Figure 20-8. Invalid Access Areas



Note 1. The following table lists the capacity and address of the code flash memory, and the lowest address of the area to be detected as invalid when accessed of each product.

Products	Code flash memory (00000H to xxxxxH)	Lowest address of the area to be detected as invalid for reading	Lowest address of the area to be detected as invalid for instruction fetching
R5F121xA (x = 1, 4, 6, 7, B)	16384 × 8 bits (00000H to 03FFFH)	10000H	08000H
R5F121xC (x = 1, 4, 6, 7, B)	32768 × 8 bits (00000H to 07FFFH)	10000H	08000H

20.3.6 Frequency detection

The IEC60730 standard mandates checking that the oscillation frequency is correct.

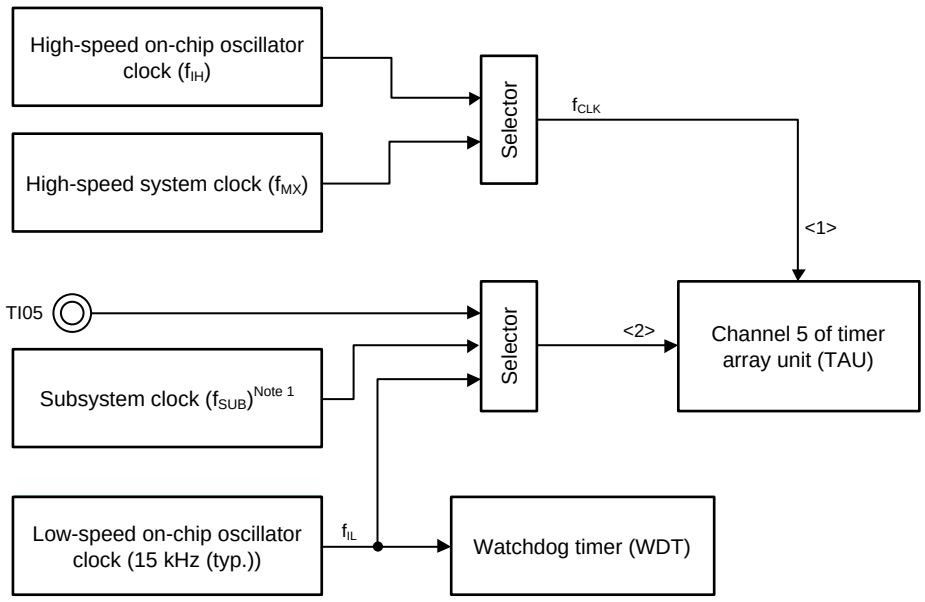
By using the CPU/peripheral hardware clock frequency (f_{CLK}) to measure the pulse width of the input signal to channel 5 of the timer array unit (TAU), whether the proportional relationship between the two clock frequencies is correct can be determined. Note that, however, if one or both clock operations are completely stopped, the proportional relationship between the clocks cannot be determined.

<Clock frequencies to be compared>

<1> CPU/peripheral hardware clock frequency (f_{CLK}):
- — High-speed on-chip oscillator clock (f_{IH})
- — High-speed system clock (f_{MX})

<2> Input on channel 5 of the timer array unit:
- — Timer input on channel 5 (TI05)
- — Low-speed on-chip oscillator clock (f_{IL}: 15 kHz (typ.))
- — Subsystem clock (f_{SUB})Note 1

Figure 20-9. Configuration of Frequency Detection



Note 1. Can only be selected in the products incorporating the subsystem clock.

Caution f_{MX} and f_{SUB} cannot be set at the same time.

If the results of input pulse interval measurement are abnormal, the clock frequency is considered to be abnormal.

For details on the input pulse interval measurement, see **6.8.4 Operation as input pulse interval measurement**.

20.3.6.1 Timer input select register 0 (TIS0)

The TIS0 register is used to select the timer input on channel 5 of 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 20-10. Format of Timer Input Select Register 0 (TIS0)

Address: F0074H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	0	0	TIS02	TIS01	TIS00

TIS02	TIS01	TIS00	Selection of timer input used with channel 5
0	0	0	Input signal of timer input pin (TI05)
0	0	1	
0	1	0	
0	1	1	
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
1	0	1	Subsystem clock (f_{SUB})
Other than the above			Setting prohibited

20.3.7 Testing of the A/D converter

The IEC60730 standard mandates testing of the A/D converter. This test checks whether or not the A/D converter is operating normally by converting the A/D converter's positive and negative reference voltages, analog input channels (ANI), temperature sensor output voltage, and internal reference voltage. For details on the method of checking, refer to the application note (R01AN0955) Safety Function (A/D test).

Use the following procedure to check the analog multiplexer.

<1> Select the ANIx pin for A/D conversion using the ADTES register (ADTES1 = 0, ADTES0 = 0).

<2> Perform A/D conversion for the ANIx pin (conversion result 1-1).

<3> Select the A/D converter's negative reference voltage for A/D conversion using the ADTES register (ADTES1 = 1, ADTES0 = 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 for A/D conversion using the ADTES register (ADTES1 = 0, ADTES0 = 0).

<6> Perform A/D conversion for the ANIx pin (conversion result 1-2).

<7> Select the A/D converter's positive reference voltage for A/D conversion using the ADTES register (ADTES1 = 1, ADTES0 = 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 for A/D conversion using the ADTES register (ADTES1 = 0, ADTES0 = 0).

<10> Perform A/D conversion for the ANIx pin (conversion result 1-3).

<11> Check that the conversion results 1-1, 1-2, and 1-3 are equal.

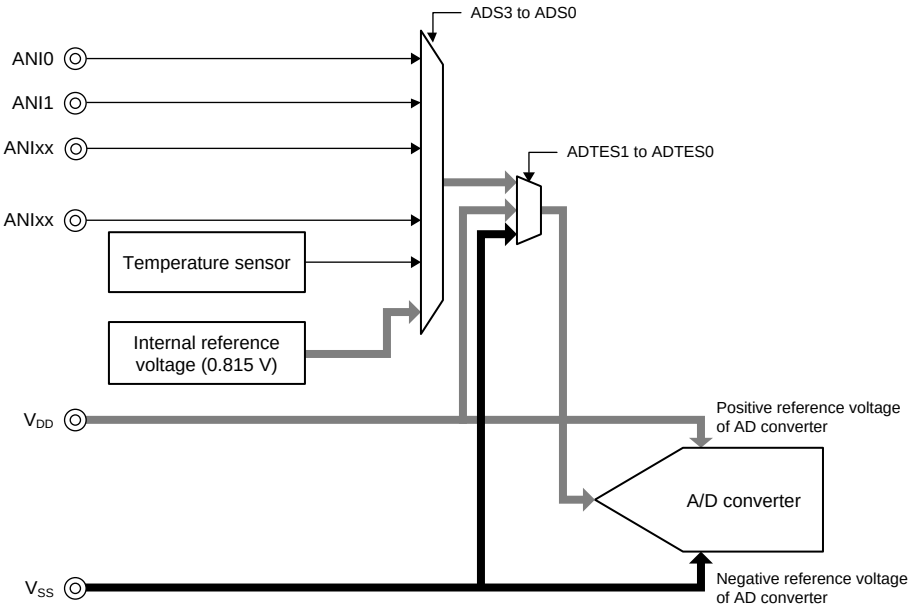
<12> Check that the A/D conversion result 2-1 is all zero and conversion result 2-2 is all one.

This procedure allows confirming that the analog multiplexer is selected and all wiring is connected

Remark 1. If the analog input voltage is variable during A/D conversion in steps <1> to <10> above, use another method to check the analog multiplexer.

Remark 2. The results of conversion might include an error. Consider an appropriate level of error in comparison of the results of conversion.

Figure 20-11. Configuration of Testing of A/D Converter



20.3.7.1 A/D test register (ADTES)

This register is used to select V_{SS} as the analog input to be A/D converted. When the internal reference voltage (0.815 V (typ.)), temperature sensor output voltage, or CTSU TSCAP voltage is selected as the target for A/D conversion, the sampling capacitor must be discharged before A/D conversion of the selected voltage proceeds. Perform A/D conversion once by setting the ADTES1 bit of the ADTES register to 1.

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

Reset signal generation clears this register to 00H.

Figure 20-12. 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	ANIx, CTSU TSCAP voltage, temperature sensor output voltage, internal reference voltage (0.815 V (typ.)) (This is specified using the analog input channel specification register (ADS).)
1	0	V_{SS} (discharging the sampling capacitor)
1	1	V_{DD}
Other than the above		Setting prohibited

Caution When A/D converting the internal reference voltage (0.815 V (typ.)), temperature sensor voltage, and CTSU TSCAP voltage, follow the procedure described in 11.7.2 Setting up the internal reference voltage, temperature sensor output voltage, or CTSU TSCAP voltage for A/D conversion to discharge the sampling capacitor once.

Remark Be sure to clear bits 2 to 7 to 0.

20.3.7.2 Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted.

Set the A/D test register (ADTES) to 00H when measuring the ANIxx, temperature sensor output voltage, or internal reference voltage (0.85 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 20-13. Format of Analog Input Channel Specification Register (ADS)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	0	0	0	0	ADS3	ADS2	ADS1	ADS0

● Select mode (ADMD = 0)

ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	0	0	ANI0	P01/ANI0 pin
0	0	0	1	ANI1	P02/ANI1 pin
0	0	1	0	ANI2	P03/ANI2 pin
0	0	1	1	ANI3	P04/ANI3 pin
0	1	0	0	ANI4	P05/ANI4 pin
0	1	0	1	ANI5	P06/ANI5 pin
0	1	1	0	ANI6	P07/ANI6 pin
0	1	1	1	ANI7	P23/ANI7 pin
1	0	0	0	ANI8	P22/ANI8 pin
1	0	0	1	ANI9	P21/ANI9 pin
1	0	1	0	ANI10	P20/ANI10 pin
1	0	1	1	Touch TSCAP voltage[Note 1]	
1	1	0	0	Temperature sensor output voltage[Note 1]	
1	1	0	1	Internal reference voltage (0.815 V (typ.))[Note 1]	
Other than the above				Setting prohibited	

Note 1. When the internal reference voltage, temperature sensor output voltage, or CTSU TSCAP voltage is selected as the target for conversion by the A/D converter, be sure to clear the LV0 bit of A/D converter mode register 0 (ADM0) to 0.

Caution 1. Only rewrite the value of the ADS register while in the conversion standby state (ADCS = 0, ADCE = 1) or conversion is stopped (ADCS = 0, ADCE = 0).

Caution 2. For the ports which are used as analog input ports, select input mode with port mode registers 0 and 2 (PM0, PM2) and analog input with port mode control registers 0 and 2 (PM0, PM2). Do not use the ADS register to set the pins which should be set as digital I/O with port mode control registers (PMC0, PMC2).

Caution 3. **The internal reference voltage cannot be used for the A/D converter and comparator simultaneously. When the internal reference voltage is selected as the target for conversion by the A/D converter (ADS3 to ADS0 = 1101B), it cannot be set as the reference voltage of the comparator.**

Caution 4. **Be sure to clear bits 4 to 7 to 0.**

CHAPTER 21 OPTION BYTE

21.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory 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.

Caution The option bytes should always be set regardless of whether each function is used.

21.1.1 User option byte (000C0H to 000C2H)

1) 000C0H

- Setting of watchdog timer operation
  - Enabling or disabling of counter operation
  - Enabling or disabling of counter operation in the HALT or STOP mode

- Time setting of watchdog timer
  - Setting of overflow time of watchdog timer
  - Setting of interval interrupt time of watchdog timer

2) 000C1H

- Setting of SPOR detection level (V_{SPOR})
- Controlling of P125/$\overline{\text{RESET}}$/(INTP0)/INTP1/(VCOUT0)/(VCOUT1)/(SI11)[Note 1] pin
  - Select P125/INTP0/INTP1/(VCOUT0)/(VCOUT1)/(SI11)[Note 1] or $\overline{\text{RESET}}$.

Note 1. For 20-pin or more pins products

3) 000C2H

- Setting of the frequency of the high-speed on-chip oscillator
  - Select from 1 to 16 MHz.

21.1.2 On-chip debug option byte (000C3H)

- Control of on-chip debug operation
  - On-chip debug operation is disabled or enabled.

21.2 Format of User Option Byte

Figure 21-1. Format of User Option Byte (000C0H)

Address: 000C0H

7	6	5	4	3	2	1	0
1	1	1	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

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.))	Watchdog timer interval interrupt time (f_{IL} = 17.25 kHz (MAX.))
0	0	0	$(2^6-1)/f_{IL}$ (3.65 ms)	$2^6/f_{IL} \times 0.75$ (2.78 ms)
0	0	1	$(2^7-1)/f_{IL}$ (7.36 ms)	$2^7/f_{IL} \times 0.75$ (5.56 ms)
0	1	0	$(2^8-1)/f_{IL}$ (14.7 ms)	$2^8/f_{IL} \times 0.75$ (11.1 ms)
0	1	1	$(2^9-1)/f_{IL}$ (29.6 ms)	$2^9/f_{IL} \times 0.75$ (22.2 ms)
1	0	0	$(2^{11}-1)/f_{IL}$ (118 ms)	$2^{11}/f_{IL} \times 0.75$ (89.0 ms)
1	0	1	$(2^{13}-1)/f_{IL}$ (474 ms)	$2^{13}/f_{IL} \times 0.75$ (356 ms)
1	1	0	$(2^{14}-1)/f_{IL}$ (949 ms)	$2^{14}/f_{IL} \times 0.75$ (712 ms)
1	1	1	$(2^{16}-1)/f_{IL}$ (3799 ms)	$2^{16}/f_{IL} \times 0.75$ (2849 ms)

WDSTBYON	Operation control of watchdog timer counter (HALT/STOP mode)
0	Counter operation stopped in HALT/STOP mode
1	Counter operation enabled in HALT/STOP mode

Caution 1. Be sure to write 1 to bits 7 to 5.

Caution 2. Setting WDTON = 0 and WDSTBYON = 1 is prohibited.

Caution 3. The watchdog timer always generates an interval interrupt when the specified time is reached unless this is specifically disabled. If the interval interrupt from the watchdog timer is not to be used, be sure to disable the interrupt by setting the WDTIMK bit to 1.

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

Figure 21-2. Format of User Option Byte (000C1H)

Address: 000C1H

7	6	5	4	3	2	1	0
1	1	1	PORTSELB	SPORS1	SPORS0	1	1

- Setting of SPOR detection voltage

Detection voltage (V_{SPOR})		Option byte setting value	
Rising edge	Falling edge	SPORS1	SPORS0
4.28 V	4.20 V	0	0
2.90 V	2.84 V	0	1
2.57 V	2.52 V	1	0
2.16 V	2.11 V	1	1

- P125/$\overline{\text{RESET}}$/INTP0/INTP1/(VCOUT0)/(VCOUT1)/(SI11)[Note 1] pin control

PORTSELB	P125/$\overline{\text{RESET}}$/INTP0/INTP1/(VCOUT0)/(VCOUT1)/(SI11)[Note 1] pin control
0	Port function (P125/INTP0/INTP1/(VCOUT0)/(VCOUT1)/(SI11)[Note 1]
1	$\overline{\text{RESET}}$ input (internal pull-up resistor can be always connected.)

Note 1. For 20-pin or more pins products

Caution 1. Be sure to write 1 to bits 7 to 5, 1, and 0.

Caution 2. Set the detection voltage (V_{SPOR}) to be within the operating voltage range.
The operating voltage range is as follows.
For CPU operating frequencies from 1 MHz to 16 MHz: V_{DD} = 2.4 to 5.5 V

Remark 1. For details on the SPOR circuit, see **CHAPTER 19 SELECTABLE POWER-ON-RESET CIRCUIT**.

Remark 2. The detection voltage is a typical value. For details, see **26.6.4 SPOR circuit characteristics** and **27.6.4 SPOR circuit characteristics**.

Figure 21-3. Format of User Option Byte (000C2H)

Address: 000C2H

7	6	5	4	3	2	1	0
1	1	1	1	1	FRQSEL2	FRQSEL1	FRQSEL0

FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator	
			Operating frequency (f_{MAIN})	Operating voltage range (V_{DD})
0	0	1	16 MHz	2.4 V to 5.5 V
0	1	0	8 MHz	
0	1	1	4 MHz	
1	0	0	2 MHz	
1	0	1	1 MHz	
Other than above			Setting prohibited	

Caution Be sure to write 1 to bits 7 to 3.

21.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

Figure 21-4. Format of On-chip Debug Option Byte (000C3H)

Address: 000C3H

7	6	5	4	3	2	1	0
OCDENSET	0	0	0	0	1	0	1

OCDENSET	Control of on-chip debug operation
0	Disables on-chip debug operation.
1	Enables on-chip debugging.[Note 1]

Note 1. Does not erase data of flash memory in case of failures in authenticating on-chip debug security ID.

Caution Bit 7 (OCDENSET) can only be specified a value.
Be sure to set 0000101B to bits 6 to 0.

Remark The value on bits 3 and 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.

21.4 Setting of Option Byte

The user option byte and on-chip debug option byte can be set using the link option in addition to describing to 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              F7H         ;   Enables watchdog timer operation,
                                    ;   Overflow time of watchdog timer is 2^9/f_IL,
                                    ;   Stops watchdog timer operation during HALT/STOP mode
        DB              E7H         ;   Select 2.90 V for rising and 2.84 V for falling for V_SPOR
                                    ;   Use the port function (P125/INTP0/INTP1/(VCOUT0)/(VCOUT1)/(SI11))
        DB              FDH             Select 1 MHz as the frequency of the high-speed on-chip oscillator clock
        DB              85H         ;   Enables on-chip debug operation
```

Caution To specify the option byte by using assembly language, use OPT_BYTE as the relocation attribute name of the CSEG pseudo instruction.

CHAPTER 22 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.

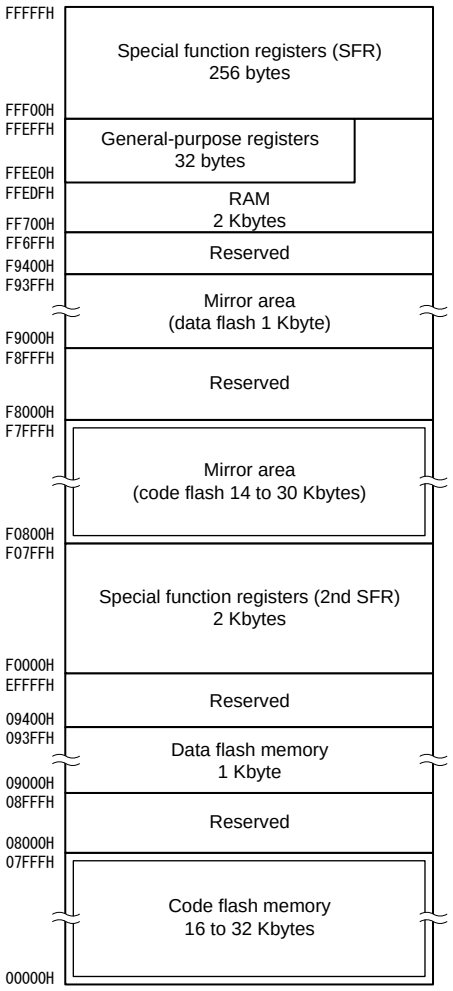


Table 22-1. Overview of Flash Memory

Flash capacity	Code flash: 16 to 32 Kbytes Data flash: 1 Kbyte
Block size	Code flash: 1 Kbyte (blocks 0 to 31) Data flash: 512 bytes (blocks 0 and 1)
Unit for writing and block erasure	[Writing] Code flash/data flash: 32 bits [Block erasure] Code flash: 1 Kbyte Data flash: 512 bytes

The following methods for programming the flash memory are available.

- The code flash memory and data 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 **22.1 Serial Programming Using Flash Memory Programmer**)

  Data can be written to the flash memory on-board or off-board by using a dedicated flash memory programmer.
- Serial programming using external device (UART communication) (see **22.2 Writing to Flash Memory by Using External Device (that Incorporates UART)**)

  Data can be written to the flash memory on-board through UART communication with an external device (microcontroller or ASIC).
- Self-programming (see **22.6 Self-Programming**)

  The user application can execute self-programming of the code flash memory or data flash memory by using the flash self-programming code.

Caution The data flash memory can be rewritten by using the flash self-programming code, but cannot be rewritten in background operation during user program execution.

22.1 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.

- PR5PG-FP6
- E2 or E2 Lite on-chip debugging emulator

Data can be written to the flash memory on-board or off-board, by using a dedicated flash memory programmer.

1) On-board programming

The contents of the flash memory can be rewritten after the RL78 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 before the RL78 microcontroller is mounted on the target system.

Table 22-2. Wiring between RL78/G16 and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer				Pin Name	Pin No.					
					10-pin	16-pin		20-pin	24-pin	32-pin
Signal Name		I/O	Pin Function		SSOP	SSOP	HWQFN	SSOP	HWQFN	QFN/QFP
PG-FP6	E2 or E2 Lite on-chip debugging emulator									
—	TOOL0	I/O	Transmit/ receive signal	TOOL0/ P40	1	2	16	4	24	2
SI/RxD	—	I/O	Transmit/ receive signal							
—	RESET_OUT	Output	Reset signal	$\overline{\text{RESET}}$	2	3	1	5	1	3
$\overline{\text{RESET}}$	—	Output								
V_{DD} Note 1		I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	5	8	6	10	10	8
GND		—	Ground	V_{SS}	4	7	5	9	9	7
FLMD1	EMV_{DD}	—	Driving power for TOOL0 pin	V_{DD}	5	8	6	10	10	8

Note 1. The signal name for the PG-FP6 is V_{CC}.

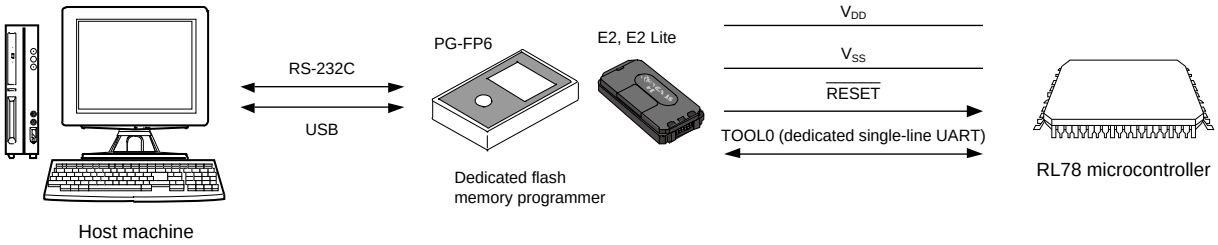
Remark Pins that are not indicated in the above table can be left open when using the flash memory programmer for flash programming.

About a connection between RL78 microcontroller and a connector, refer to the user's manual of each programmer.
About a connection with E2 or E2 Lite, see **23.1 Connecting E2, E2 Lite On-chip Debugging Emulator**.

22.1.1 Programming environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 22-1. Environment for Writing Program to Flash Memory



A host machine that controls the dedicated flash memory programmer is necessary.

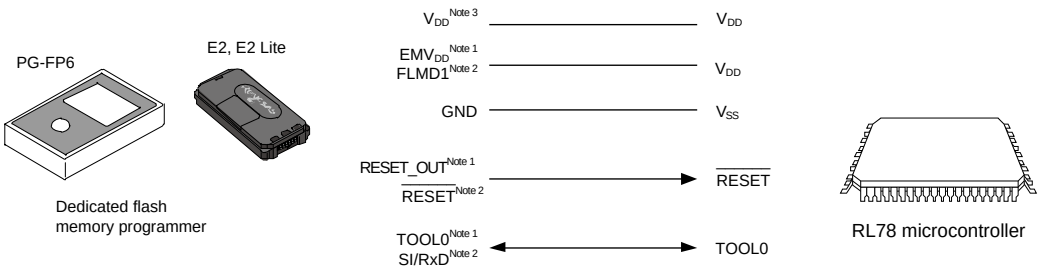
To interface between the dedicated flash memory programmer and the RL78 microcontroller, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART.

22.1.2 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: Fixed to 115200 bps

Figure 22-2. Communication with Dedicated Flash Memory Programmer



Note 1. When using E2 or E2 Lite on-chip debugging emulator.

Note 2. When using PG-FP6.

Note 3. The signal name for the PG-FP6 is V_{CC}.

The dedicated flash memory programmer generates the following signals for the RL78 microcontroller. See the manual for the PG-FP6 or, E2 or E2 Lite on-chip debugging emulator for details.

Table 22-3. Pin Connection

Dedicated Flash Memory Programmer				RL78 Microcontroller
Signal Name		I/O	Pin Function	Pin Name[Note 1]
PG-FP6	E2 or E2 Lite on-chip debugging emulator			
V_{DD}[Note 2]		I/O	V_{DD} voltage generation/ power monitoring	V_{DD}
GND		—	Ground	V_{SS}
FLMD1	EMV_{DD}		Driving power for TOOL0 pin	V_{DD}
$\overline{RESET}$	—	Output	Reset signal	$\overline{RESET}$
—	RESET_OUT	Output		
—	TOOL0	I/O	Transmit/receive signal	TOOL0
SI/RxD	—	I/O	Transmit/receive signal	

Note 1. Pins to be connected differ with the product. For details, see **Table 22-1**.

Note 2. The signal name for the PG-FP6 is V_{CC}.

22.2 Writing to Flash Memory by Using External Device (that Incorporates UART)

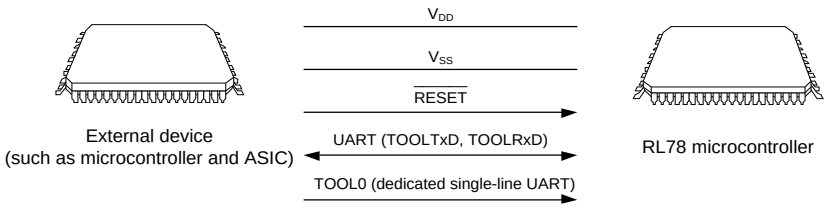
On-board data writing to the internal flash memory is possible by using the RL78 microcontroller and an external device (a microcontroller or ASIC) connected to a UART.

On the development of flash memory programmer by user, refer to the RL78 Microcontrollers (RL78 Protocol B) Serial Programming Edition Application Note (R01AN6332).

22.2.1 Programming environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 22-3. Environment for Writing Program to Flash Memory



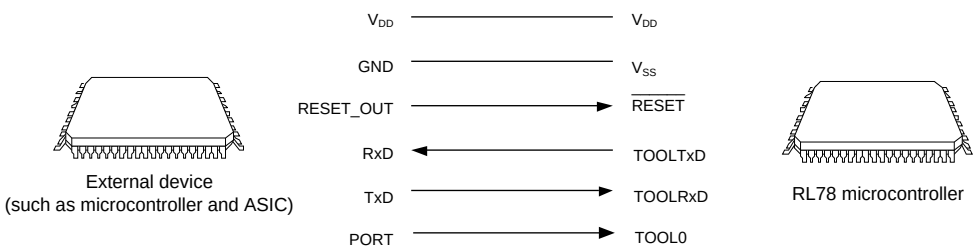
Processing to write data to or erase data from the RL78 microcontroller by using an external device is performed on-board. Off-board writing is not possible.

22.2.2 Communication mode

Communication between the external device and the RL78 microcontroller is established by serial communication using the TOOL0 pin via the dedicated UART of the RL78 microcontroller.

Transfer rate: Fixed to 115200 bps

Figure 22-4. Communication with External Device



The external device generates the following signals for the RL78 microcontroller.

Table 22-4. 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}
RESET_OUT	Output	Reset signal output	$\overline{\text{RESET}}$
RxD	Input	Receive signal	TOOLTxD
TxD	Output	Transmit signal	TOOLRxD
PORT	Output	Mode signal	TOOL0

22.3 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 the flash memory programming mode, see **22.4.2 Flash memory programming mode**.

22.3.1 P40/TOOL0 pin

In the flash memory programming mode, connect this pin to the dedicated flash memory programmer via an external 1kΩ 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 pin 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 **26.10 Timing of Entry to Flash Memory Programming Mode** and **27.10 Timing of Entry to Flash Memory Programming Mode**).

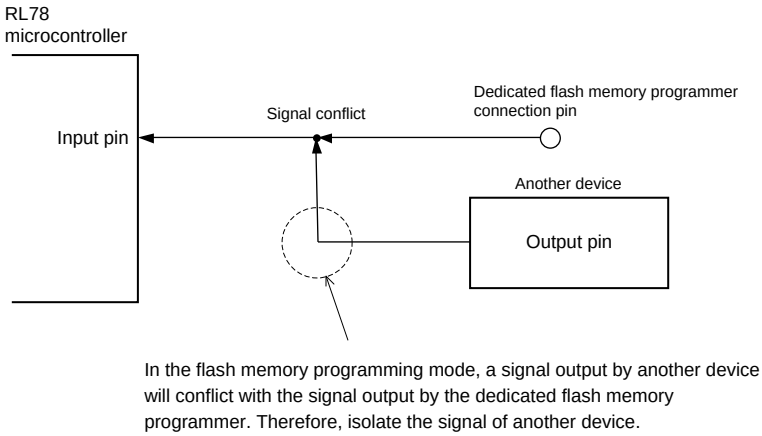
Remark 2. The SAU and IICA pins are not used for communication between the RL78 microcontroller and dedicated flash memory programmer, because single-line UART (TOOL0 pin) is used.

22.3.2 $\overline{\text{RESET}}$ 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 22-5. Signal Conflict ($\overline{\text{RESET}}$ Pin)



22.3.3 Port pins

When the flash memory programming mode is set, all the pins not used for flash memory programming enter the same status as that immediately after reset. If external devices connected to the ports do not recognize the port status immediately after reset, the port pin must be connected to either V_{DD} or V_{SS} via a resistor.

22.3.4 X1 and X2 pins (16-pin, 20-pin, 24-pin, and 32-pin products)

Connect X1 and X2 pins 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.

22.3.5 Power supply

To use the supply voltage output of the flash memory programmer, connect the V_{DD} pin to V_{DD}[Note 1] 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.

Note that the operating voltage during flash memory programming must be in the range from 2.4 V to 5.5 V. If the on-board supply voltage is less than 2.4 V, satisfy the requirement for operating voltage (2.4 V to 5.5 V) by, for example, switching to the voltage from a dedicated flash memory programmer, and isolate the on-board supply voltage.

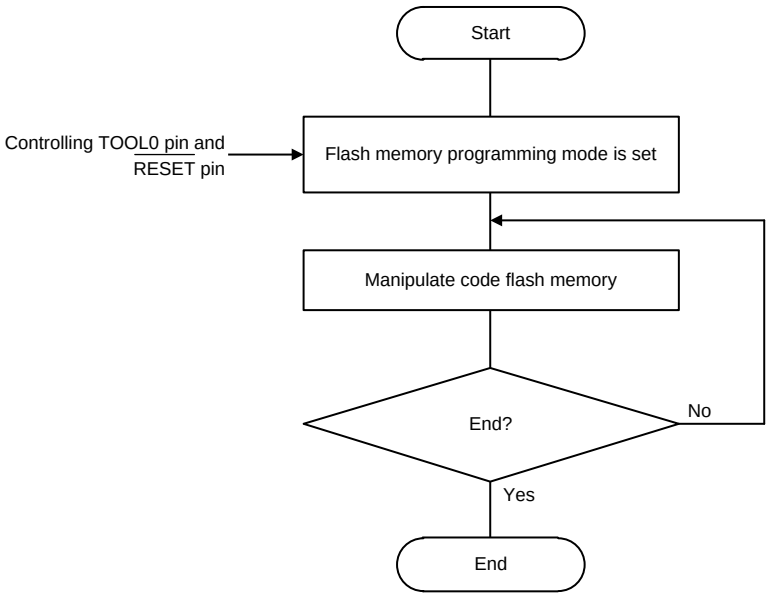
Note 1. The signal name for the PG-FP6 is V_{CC}.

22.4 Serial Programming Method

22.4.1 Serial programming procedure

The following figure illustrates a flow for rewriting the code flash memory through serial programming.

Figure 22-6. Code Flash Memory Manipulation Procedure



For the flash memory programming mode, see **22.4.2 Flash memory programming mode**.

22.4.2 Flash memory programming mode

To rewrite the contents of the code flash memory by serial programming, the flash memory programming mode must be entered.

<Serial programming 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. The operating voltage during the flash memory programming mode is 2.4 V to 5.5 V.

<Serial programming using an external device (UART communication)>

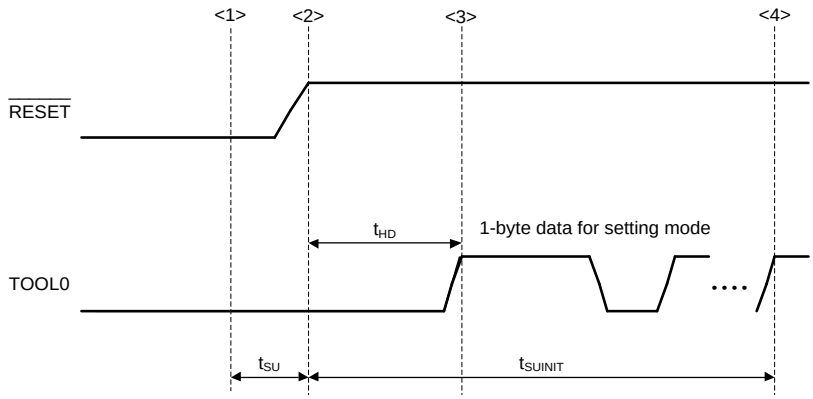
Set the TOOL0 pin to the low level, and then cancel the reset (see **Table 22-5**). After that, enter flash memory programming mode according to the procedures <1> to <4> shown in **Figure 22-7**.

The operating voltage during the flash memory programming mode is 2.4 V to 5.5 V.

Table 22-5. 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 22-7. Setting of Flash Memory Programming Mode



<1> The low level is input to the TOOL0 pin.

<2> The external reset ends (SPOR reset must end before the external reset ends.).

<3> The TOOL0 pin is set to the high level.

<4> Setting of entry to the flash memory programming mode by UART reception.

Remark t_{SUINIT}: The segment shows that it is necessary to finish specifying the initial communication settings within 100 ms from when the resets end.
t_{SU}: How long from when the TOOL0 pin is placed at the low level until an external reset ends
t_{HD}: How long to keep the TOOL0 pin at the low level from when the external reset ends

For details, see **26.10 Timing of Entry to Flash Memory Programming Mode** and **27.10 Timing of Entry to Flash Memory Programming Mode**.

22.4.3 Selecting communication mode

Communication modes of the RL78 microcontroller are as follows.

Table 22-6. Communication Modes

Communication Mode	Standard Setting[Note 1]				Pins Used
	Port	Speed[Note 2]	Frequency	Multiply Rate	
1-line UART (when flash memory programmer is used, or when external device is used)	UART	115200 bps	—	—	TOOL0
Dedicated UART (when external device is used)	UART	115200 bps	—	—	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.

22.4.4 Communication commands

The RL78 microcontroller executes serial programming through the commands listed in **Table 22-7**.

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.

Table 22-7. Flash Memory Control Commands

Classification	Command Name	Function
CRC checking	CRC check (code flash memory)	Calculate the checksum of the code flash memory.
	CRC check (data flash memory)	Calculate the checksum of the data flash memory.
Writing after erasure	Write after erase (code flash memory)	Write data after erasing data in the code flash memory.
	Write after erase (data flash memory)	Write data after erasing data in the data flash memory.

22.5 Processing Time for Each Command When PG-FP5 Is in Use (Reference Values)

The following tables show the processing time for each command (reference value) when PG-FP5 is used as a dedicated flash memory programmer.

Table 22-8. Processing Time for Each Command When PG-FP5 Is in Use (Reference Values)

Command of PG-FP5	Code Flash		Data Flash
	16 KB	32 KB	1 KB
	R5F121BAxFP, R5F121BAxFNA, R5F1217AxNA, R5F1216AxSP, R5F1214AxSP, R5F1214AxNA, R5F1211AxSP (x = M,G,A)	R5F121BCxFP, R5F121BCxFNA, R5F1217CxNA, R5F1216CxSP, R5F1214CxSP, R5F1214CxNA, R5F1211CxSP (x = M,G,A)	All products
Write after erase	2.5 s	3.0 s	1.0 s
CRC check	1.5 s	2.0 s	0.5 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: 115,200 bps

22.6 Self-Programming

The RL78 microcontroller supports a self-programming function that can be used to rewrite the code flash memory via a user program. Because this function allows a user application to rewrite the code flash memory by using the flash self-programming library, it can be used to upgrade the program in the field.

Caution 1. **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.**
Since the CPU is stopped while rewriting the flash memory, an interrupt cannot be accepted during this period.

Caution 2. **The high-speed on-chip oscillator should be kept operating during self-programming. If it is kept stopping, the high-speed on-chip oscillator clock should be operated (HIOSTOP = 0). The flash self-programming code should be executed after 30 μs have elapsed.**

22.6.1 Registers controlling self-programming

- Flash address pointer registers (FLAPH, FLAPL)
- Flash end address specification registers (FLSEDH, FLSEDL)
- Flash write buffer registers (FLWHH, FLWHL, FLWLH, FLWLL)
- Flash programming mode control register (FLPMC)
- Flash memory sequencer initial setting register (FSSET)
- Flash memory sequencer control register (FSSQ)
- Flash memory sequencer status registers (FSASTH, FSASTL)

22.6.1.1 Flash address pointer registers H and L (FLAPH, FLAPL)

The FLAPH and FLAPL registers specify the address where programming of the flash memory is to start.

The FLAPH and FLAPL registers can be set by an 8-bit memory manipulation instruction.

These registers are set to 00H following a reset.

Figure 22-8. Format of Flash Address Pointer Registers H and L (FLAPH, FLAPL)

Address: F00C3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
FLAPH	0	FLAP14	FLAP13	FLAP12	FLAP11	FLAP10	FLAP9	FLAP8

Address: F00C2H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
FLAPL	FLAP7	FLAP6	FLAP5	FLAP4	FLAP3	FLAP2	FLAP1	FLAP0

22.6.1.2 Flash end address specification registers H and L (FLSEDH, FLSEDL)

The FLSEDH and FLSEDL registers specify the address where programming of the flash memory is to end.

The FLSEDH and FLSEDL registers can be set by an 8-bit memory manipulation instruction.

These registers are set to 00H following a reset.

Figure 22-9. Format of Flash End Address Specification Registers H and L (FLSEDH, FLSEDL)

Address: F00C5H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
FLSEDH	0	EWA14	EWA13	EWA12	EWA11	EWA10	EWA9	EWA8

Address: F00C4H After reset: 00H R/W

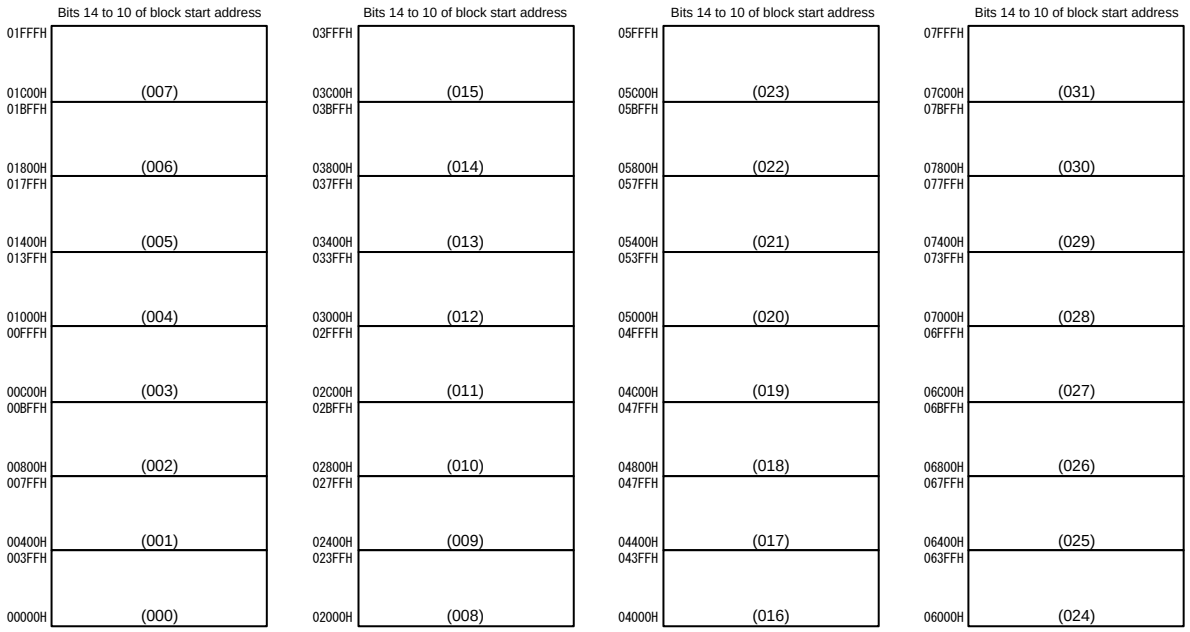
Symbol	7	6	5	4	3	2	1	0
FLSEDL	EWA7	EWA6	EWA5	EWA4	EWA3	EWA2	0	0

Table 22-9. Method of Setting the FLAPH/L and FLSEDH/L Registers

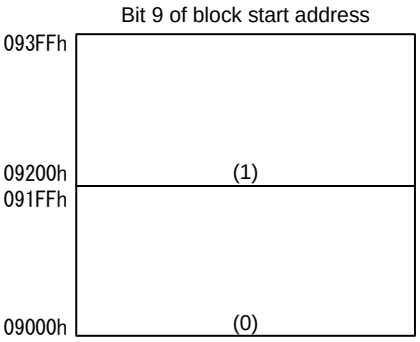
Commands exclusively for use with the flash memory sequencer		Settings of the FLAPH/L and FLSEDH/L Registers	
FSSQ	Write	Code flash	FLAPH/L registers: Bits 14 to 0 = bits 14 to 0 of the address from which writing is to proceed FLSEDH/L registers: Bits 14 to 0 = All 0s (can be unset)
		Data flash	FLAPH/L registers: Bits 14 to 10 = All 0s Bits 9 to 0 = Bits 9 to 0 of the address from which writing is to proceed FLSEDH/L registers: Bits 14 to 0 = All 0s (can be unset)
	Block erase	Code flash	FLAPH/L registers: Bits 14 to 10 = bits 14 to 10 of the block start address Bits 9 to 0 = All 0s FLSEDH/L registers: Bits 14 to 10 = bits 14 to 10 of the block start address Bits 9 to 2 = All 1s
		Data flash	FLAPH/L registers: Bits 14 to 10 = All 0s Bit 9 = bit 9 of the block start address Bits 8 to 0 = All 0s FLSEDH/L registers: Bits 14 to 10 = All 0s Bit 9 = bit 9 of the block start address Bits 8 to 2 = All 1s

Caution Set the FLAPH/L registers and the FLSEDH/L registers so that the following condition is met.
FLAPH/L setting ≤ FLSEDH/L setting

Block configuration of code flash memory



Block configuration of data flash memory



22.6.1.3 Flash write buffer registers HH, HL, LH, and LL (FLWHH, FLWHL, FLWLH, FLWLL)

The FLWHH, FLWHL, FLWLH, and FLWLL registers hold data to be written during programming of the flash memory.

The FLWHH, FLWHL, FLWLH, and FLWLL registers can be set by an 8-bit memory manipulation instruction.

These registers are set to 00H following a reset.

Figure 22-10. Format of Flash Write Buffer Registers HH, HL, LH, and LL (FLWHH, FLWHL, FLWLH, FLWLL)

Address: F00CBH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
FLWHH	FLW31	FLW30	FLW29	FLW28	FLW27	FLW26	FLW25	FLW24

Address: F00CAH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
FLWHL	FLW23	FLW22	FLW21	FLW20	FLW19	FLW18	FLW17	FLW16

Address: F00C9H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
FLWLH	FLW15	FLW14	FLW13	FLW12	FLW11	FLW10	FLW9	FLW8

Address: F00C8H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
FLWLL	FLW7	FLW6	FLW5	FLW4	FLW3	FLW2	FLW1	FLW0

22.6.1.4 Flash programming mode control register (FLPMC)

The FLPMC register sets the flash memory to the self-programming mode.

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

This register is set to 08H following a reset.

Figure 22-11. Format of Flash Programming Mode Control Register (FLPMC)

Address: F00C0H After reset: 08H R/W

Symbol	7	6	5	4	3	2	1	0
FLPMC	0	0	SELDFL	0	FWEDIS	0	FLSPM	0

SELDFL	Flash programming area selection
0	Selects the code flash area.
1	Selects the data flash area.

FWEDIS	Software control over enabling or disabling erasure and programming of the flash memory[Note 1]
0	Enables programming and erasure.
1	Disables programming and erasure.

FLSPM	Flash programming mode selection[Note 1]
0	The flash memory is in the read mode (normal mode).
1	The flash memory is in the self-programming mode.

Note 1. Be sure to keep the value of this bit at 0 until erasure or programming of the flash memory is completed.

22.6.1.5 Flash memory sequencer initial setting register (FSSET)

The FSSET register sets the operating frequency of the flash memory sequencer.

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

This register is set to 00H following a reset.

Figure 22-12. Format of Flash Memory Sequencer Initial Setting Register (FSSET)

Address: F00BEH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
FSSET	0	0	0	FSET4	FSET3	FSET2	FSET1	FSET0

FSET4-0	Setting of the operating frequency of the flash memory sequencer
—	Set the operating frequency of the flash memory sequencer. For the correspondence between the operating frequency of the flash memory sequencer and the setting of the FSET4-0 bits, see **Table 22-10**.

Caution Set the value corresponding to that obtained by rounding the CPU operating frequency up to the nearest whole number in the FSET4-0 bits. For example, when the CPU operating frequency is 4.5 MHz, set the bits for 5 MHz.
Note that frequencies that are not whole numbers, such as 1.5 MHz, are not available as CPU operating frequencies below 4 MHz.

Table 22-10. Correspondence between the Operating Frequency of the Flash Memory Sequencer and the Setting of the FSET4-0 Bits

Operating frequency [MHz]	Setting of the FSET4-0 bits	Operating frequency [MHz]	Setting of the FSET4-0 bits	Operating frequency [MHz]	Setting of the FSET4-0 bits
16	01111B	10	01001B	4	00011B
15	01110B	9	01000B	3	00010B
14	01101B	8	00111B	2	00001B
13	01100B	7	00110B	1	00000B
12	01011B	6	00101B	—	—
11	01010B	5	00100B	—	—

22.6.1.6 Flash memory sequencer control register (FSSQ)

The FSSQ register defines the commands to be used when the flash memory sequencer is activated.

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

This register is set to 00H following a reset.

Figure 22-13. Format of Flash Memory Sequencer Control Register (FSSQ)

Address: F00C1H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
FSSQ	SQST	0	0	0	0	SQMD2	SQMD1	SQMD0

SQST	Flash memory sequencer operation control
0	Stops operation.
1	Starts operation.

SQMD2-0	Flash memory sequencer command selection
000b	Initial value (commands not selected)
001b	Write Writes data specified in the FLWHH, FLWHL, FLWLH, and FLWLL registers to the address specified in the FLAPH and FLAPL registers. Unit for writing (code flash area): 1 word (4 bytes) (when the SELDFL bit is set to 0) Unit for writing (data flash area): 1 word (4 bytes) (when the SELDFL bit is set to 1)
100b	Block erasure Erases blocks in the range from the block start address specified in the FLAPH and FLAPL registers to the block end address specified in the FLSEDH and FLSEDL registers. Unit for block erasing (code flash area): 1 block (1 Kbyte) (when the SELDFL bit is set to 0) Unit for block eraing (data flash area): 1 block (512 bytes) (when the SELDFL bit is set to 1)
Other than above	Setting prohibited

22.6.1.7 Flash memory sequencer status registers H and L (FSASTH, FSASTL)

The FSASTH and FSASTL registers indicate the results of the operations of the flash memory sequencer.

The FSASTH and FSASTL registers can be read by an 8-bit memory manipulation instruction.

Figure 22-14. Format of Flash Memory Sequencer Status Registers H and L (FSASTH, FSASTL)

Address: F00C7H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
FSASTH	0	SQEND	0	0	0	0	0	0

Address: F00C6H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
FSASTL	0	0	0	SEQER	0	0	WRER	ERER

SQEND	Flash memory sequencer operation status flag
0	Operation is in progress.
1	Operation has ended.
<Clearing condition> The SQST bit being cleared.	

SEQER	Flash memory sequencer error flag
0	No error has occurred.
1	An error has occurred.
<Clearing condition> Next activation of the flash memory sequencer	

WRER	Write command error flag
0	No error has occurred.
1	An error has occurred.
<Clearing condition> Activation of the next command action Forcible termination of a command during writing leads to the values read being undefined.	

ERER	Block erase command error flag
0	No error has occurred.
1	An error has occurred.
<Clearing condition> Next activation of the flash memory sequencer Forcible termination of the command during block erasure leads to values read being undefined.	

22.6.2 Procedure for executing self-programming of code/data flash memory

The following figure illustrates a flow for rewriting the code/data flash memory by using the flash self-programming code.

For details of the registers to be used for execution of self-programming, see **22.6.1 Registers controlling self-programming**.

Figure 22-15. Flash Memory Self-Programming Execution Procedure

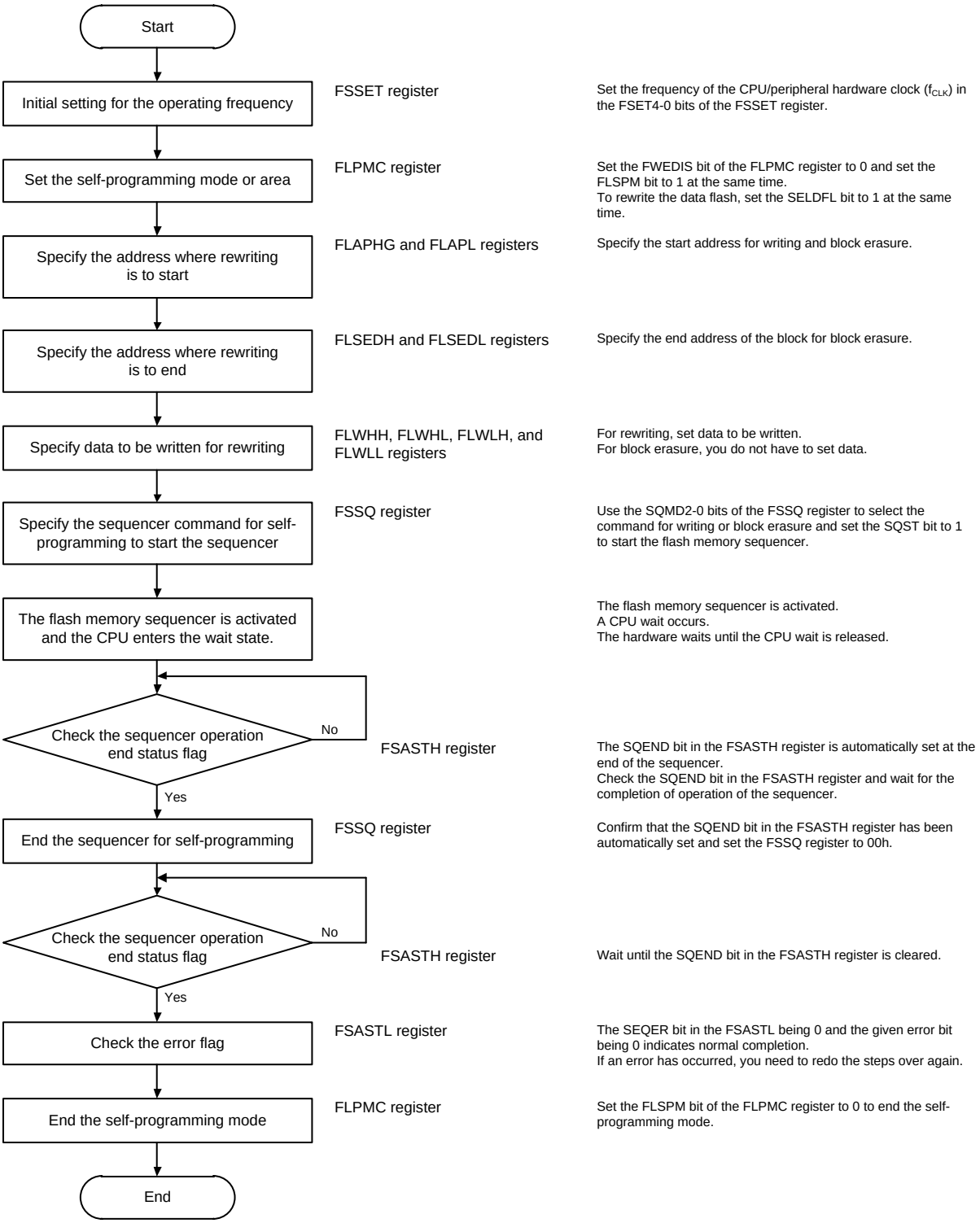
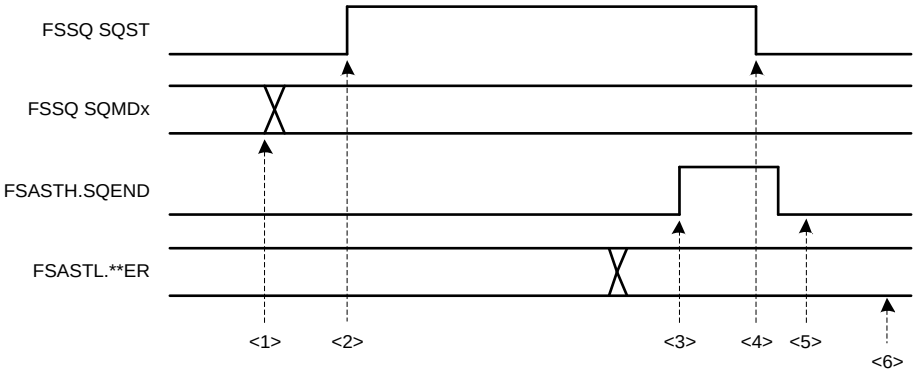


Figure 22-16. Flash Memory Sequencer Starting and Ending Processing



<1> Set up operation.

<2> Set the SQST bit (the sequencer starts operating and the CPU enters the wait state).

<3> The CPU wait state is released.

<4> Clear the SQST bit (the sequencer stops operating).

<5> Check the sequencer operation end status flag.

<6> Check the error flag.

22.6.3 Notes on self-programming

(1) Allocate the self-programming code for rewriting the code/data flash area to the code flash area. Self-programming by fetching from the RAM is prohibited. Additionally, rewriting the boot area and the block for storing the self-programming code is prohibited.

(2) Prohibit an interrupt before setting the self-programming mode. To prohibit an interrupt, clear (0) the IE flag by the DI instruction in the same way as in the normal operation mode.

(3) When using the flash memory sequencer to rewrite the code/data flash memory, set the value corresponding to the CPU operating frequency in the FSET4-0 bits of the FSSET register before proceeding. Note that if rewriting is attempted while the value corresponding to the CPU operating frequency is not correct, operation is undefined and written data are not guaranteed. Even if the values in the flash memory are as expected immediately after rewriting, retaining the values for any specified period is not guaranteed.

(4) The high-speed on-chip oscillator should be kept operating before executing self-programming. If it is stopped, it should be made to operate again (HIOSTOP = 0), and the flash self-programming code should be re-executed after 30 μs have elapsed.

(5) Do not execute other settings or instructions which are not related to the self-programming procedure during the self-programming execution flow shown in **22.6.2 Procedure for executing self-programming of code/data flash memory**.

(6) The CPU is stopped during rewriting through self-programming. The code flash or data flash memory cannot be accessed while it is being rewritten.

22.7 Data Flash

22.7.1 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 self-programming code.
- 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 (512-byte) units.
- The data flash can be accessed in 8-bit or 16-bit units.
- The data flash can be directly read by CPU instructions.
- Instructions cannot be executed from the code flash memory while rewriting the data flash memory since the CPU is stopped during this period (that is, background operation (BGO) is not supported).
- Because the data flash memory is an area exclusively used for data, it cannot be used to execute instructions.
- The data flash memory cannot be accessed while rewriting the code flash memory since the CPU is stopped during this period.
- The high-speed on-chip oscillator should be kept operating while rewriting the data flash memory. If it is stopped, it should be made to operate again (HIOSTOP = 0), and the flash self-programming code should be re-executed after 30 μs have elapsed.

22.7.2 Procedure for accessing data flash memory

The data flash memory is accessible at any time after a reset ends. Reading the data flash memory by CPU instructions does not require initial settings of the registers. For the procedure for rewriting the data flash memory, see **22.6.2 Procedure for executing self-programming of code/data flash memory**.

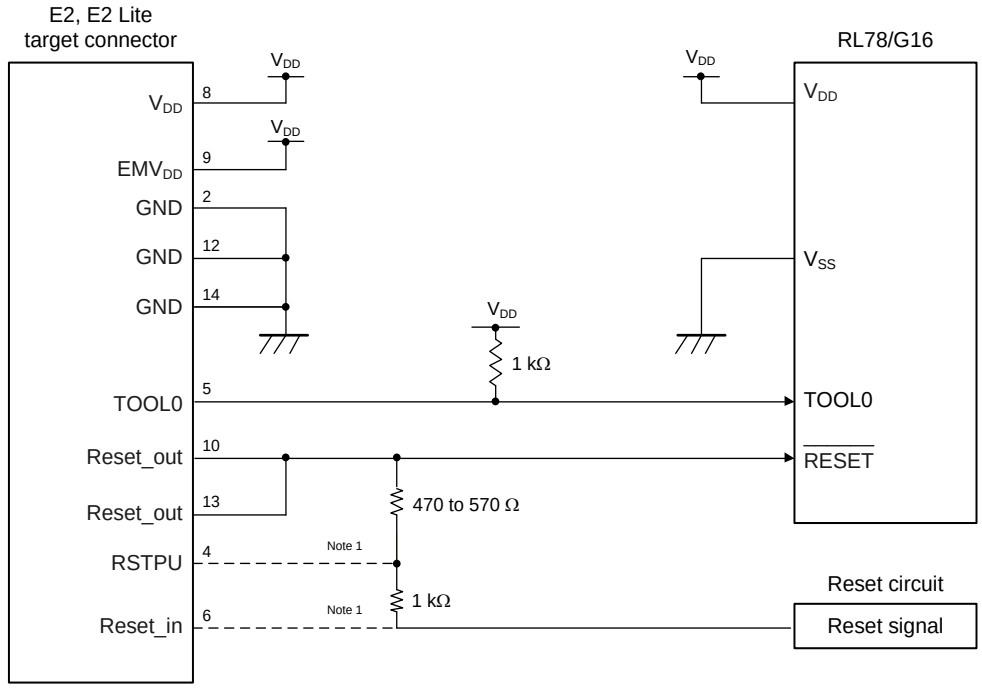
CHAPTER 23 ON-CHIP DEBUG FUNCTION

23.1 Connecting E2, E2 Lite On-chip Debugging Emulator

The RL78 microcontroller uses the V_{DD}, $\overline{RESET}$, TOOL0, and V_{SS} pins to communicate with the host machine via an E2, E2 Lite on-chip debugging emulator. Serial communication is performed by using a single-line UART that uses the TOOL0 pin.

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.

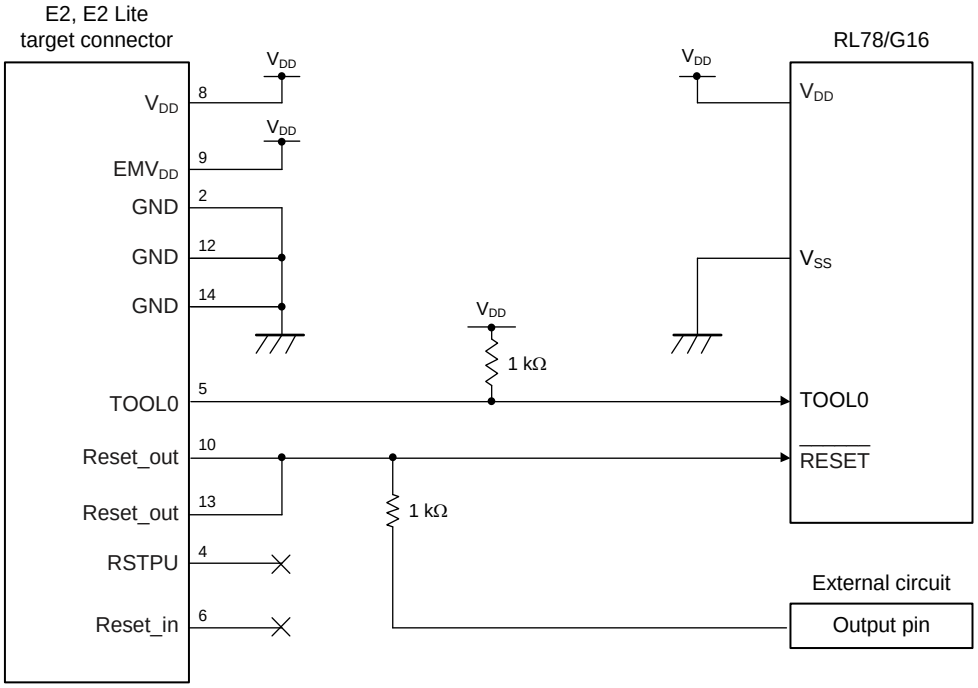
Figure 23-1. Connection Example of E2, E2 Lite On-chip Debugging Emulator



Note 1. Connecting is not necessary during flash programming.

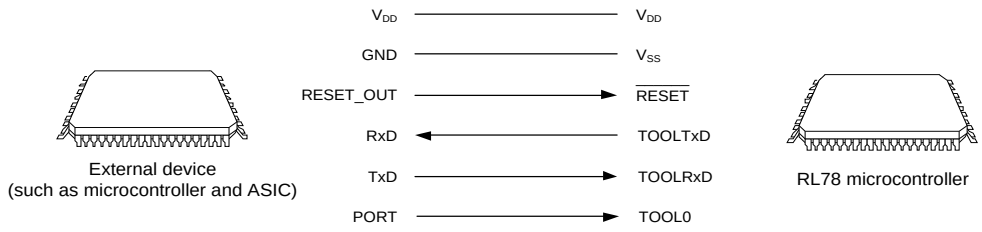
For the target system which uses the multi-use feature of $\overline{\text{RESET}}$ pin, its connection to an external circuit should be isolated.

Figure 23-2. Connection Example of E2, E2 Lite On-chip Debugging Emulator and RL78 microcontroller
(When using to the alternative function of $\overline{\text{RESET}}$ pin)



23.2 Connecting External Device (that Incorporates UART)

The V_{DD}, $\overline{RESET}$, TOOL0, V_{SS}, TOOLTxD, and TOOLRxD pins are used to communicate with the host machine on board via the RL78 microcontroller and an external device (a microcontroller or ASIC) connected to a UART. Communication between the external device and the RL78 microcontroller is established by serial communication via the dedicated UART using the TOOLTxD and TOOLRxD pins of the RL78 microcontroller.



23.3 On-Chip Debug Security ID

The RL78 microcontroller has an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 21 OPTION BYTE**) and an on-chip debug security ID setting area at 000C4H to 000CDH, to prevent third parties from reading memory content.

Table 23-1. On-Chip Debug Security ID

Address	On-Chip Debug Security ID
000C4H to 000CDH	Any ID code of 10 bytes

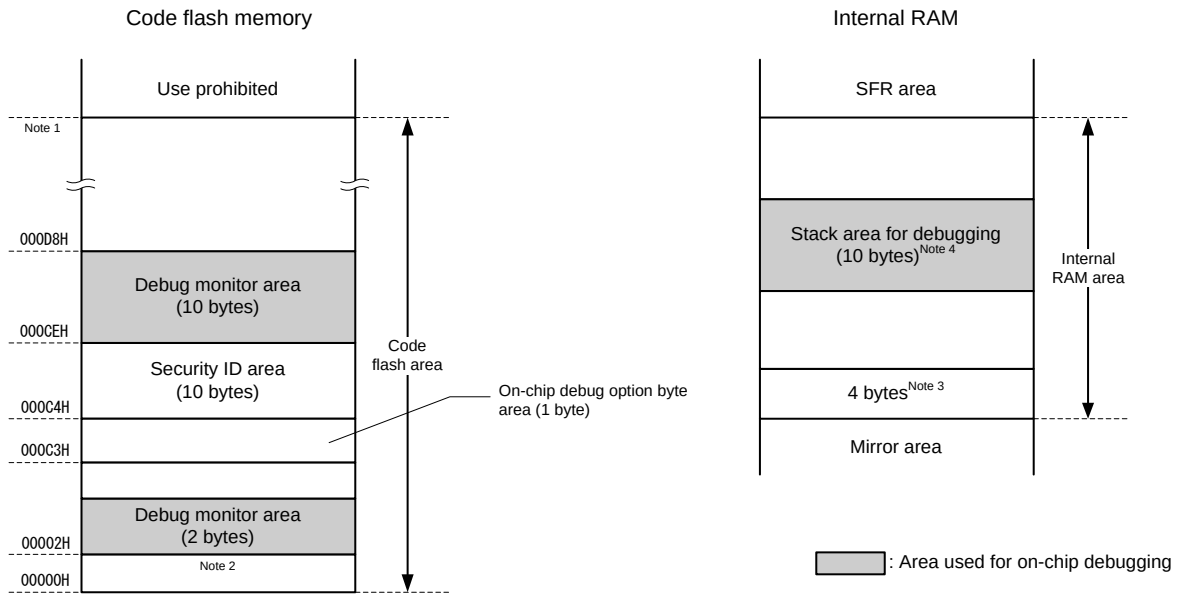
23.4 Securing of User Resources

To perform communication between the RL78 microcontroller and E2, E2 Lite on-chip debugging emulator, as well as each debug function, the securing of memory space must be done beforehand. If Renesas Electronics assembler or compiler is used, the items can be set by using link options.

1) Securement of memory space

The shaded portions in **Figure 23-3** are the areas reserved for placing the debug monitor program, so user programs or data cannot be allocated in these spaces. When using the on-chip debug function, these spaces must be secured so as not to be used by the user program. Moreover, this area must not be rewritten by the user program.

Figure 23-3. Memory Spaces Where Debug Monitor Programs Are Allocated



Note 1. Address differs depending on products as follows.

Products	Address
R5F121BA, R5F1217A, R5F1216A, R5F1214A, R5F1211A	03FFFH
R5F121BC, R5F1217C, R5F1216C, R5F1214C, R5F1211C	07FFFH

Note 2. In debugging, reset vector is rewritten to address allocated to a monitor program.

Note 3. When the pseudo-RRM function and the pseudo-DMM function are used, four bytes in the RAM area are consumed.
The RAM area is set by the build tool, when the pseudo-RRM function and the pseudo-DMM function are used.
If the RAM area has not been set, the first four bytes of the RAM area will be used.
(For details on the setting, refer to the user's manual of the build tool.)
When the pseudo-RRM function and pseudo-DMM function are not used, this area can be used as the internal RAM.

Note 4. Since this area is allocated immediately before the stack area, the address of this area varies depending on the stack increase and decrease. That is, 10 extra bytes are consumed for the stack area used.

CHAPTER 24 BCD CORRECTION CIRCUIT

24.1 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).

24.2 Registers Used by BCD Correction Circuit

The BCD correction circuit uses the following registers.

- BCD correction result register (BCDADJ)

24.2.1 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 24-1. Format of BCD Correction Result Register (BCDADJ)

Address: F00FEH After reset: undefined R

Symbol	7	6	5	4	3	2	1	0
BCDADJ								

24.3 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 25 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 (R01US0015E)**.

25.1 Conventions Used in Operation List

25.1.1 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 **Table 25-1**, R0, R1, R2, etc.) can be used for description.

Table 25-1. Operand Identifiers and Specification Methods

Identifier	Description Method	
r	X(R0), A(R1), C(R2), B(R3), E(R4), D(R5), L(R6), H(R7)	
rp	AX(RP0), BC(RP1), DE(RP2), HL(RP3)	
sfr	Special-function register symbols (SFR symbols) FFF00H to FFFFFH	
sfrp	Special-function register symbols (16-bit manipulatable SFR symbols. Even addresses only[Note 1]) FFF00H to FFFFFH	
saddr	FFE20H to FFF1FH	Immediate data or labels
saddrp	FFE20H to FFF1FH	Immediate data or labels (even addresses only[Note 1])
addr20	00000H to FFFFFH	Immediate data or labels
addr16	0000H to FFFFH	Immediate data or labels (even addresses only for 16-bit data transfer instructions[Note 1])
addr5	0080H to 00BFH	Immediate data or labels (even addresses only[Note 1])
word	16-bit immediate data or label	
byte	8-bit immediate data or label	
bit	3-bit immediate data or label	
RBn	RB0 to RB3	

Note 1. Bit 0 = 0 when an odd address is specified.

Remark The special function registers can be described to operand sfr as symbols. See **Table 3-5 SFR List** for the symbols of the special function registers.
The extended special function registers can be described to operand !addr16 as symbols. See **Table 3-6 Extended SFR (2nd SFR) List** for the symbols of the extended special function registers.

25.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

Table 25-2. 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)

25.1.3 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 25-3. 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

25.1.4 PREFIX instruction

Instructions with “ES:” have a PREFIX operation code as a prefix to extend the accessible data area to the 1 MB space (00000H to FFFFFH), by adding the ES register value to the 64 KB space from F0000H to FFFFFH. When a PREFIX operation code is attached as a prefix to the target instruction, only one instruction immediately after the PREFIX operation code is executed as the addresses with the ES register value added.

An interrupt and DMA transfer are not acknowledged between a PREFIX instruction code and the following one instruction.

Table 25-4. Use Example of PREFIX Operation Code

Instruction	Opcode				
	1	2	3	4	5
MOV !addr16, #byte	CFH	!addr16		#byte	—
MOV ES:!addr16, #byte	11H	CFH	!addr16		#byte
MOV A, [HL]	8BH	—	—	—	—
MOV A, ES:[HL]	11H	8BH	—	—	—

Caution Set the ES register value with MOV ES, A, etc., before executing the PREFIX instruction.

25.2 Operation List

Table 25-5. Operation List (1/13)

Instruction Group	Mnemonic	Operand	Bytes	Clock		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			
		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)			

Table 25-5. Operation List (2/13)

Instruction Group	Mnemonic	Operand	Bytes	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	[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			
		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)			

Table 25-5. Operation List (3/13)

Instruction Group	Mnemonic	Operand	Bytes	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	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)			
		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	×		×

Table 25-5. Operation List (4/13)

Instruction Group	Mnemonic	Operand	Bytes	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
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 4]	1	1	—	AX ← rp			
		rp, AX[Note 4]	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			
		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			
		BC, !addr16	3	1	4	BC ← (addr16)			

Table 25-5. Operation List (5/13)

Instruction Group	Mnemonic	Operand	Bytes	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	MOVW	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 4]	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 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)	×	×	×
	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	×	×	×

Table 25-5. Operation List (6/13)

Instruction Group	Mnemonic	Operand	Bytes	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	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)	×	×	×
	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)	×		

Table 25-5. Operation List (7/13)

Instruction Group	Mnemonic	Operand	Bytes	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	AND	A, [HL+C]	2	1	4	A ← A ∧ (HL + C)	×		
		A, ES:[HL+C]	3	2	5	A ← A ∧ ((ES:HL) + C)	×		
	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)	×		

Table 25-5. Operation List (8/13)

Instruction Group	Mnemonic	Operand	Bytes	Clock		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)	×	×	×
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)	×	×	×

Table 25-5. Operation List (9/13)

Instruction Group	Mnemonic	Operand	Bytes	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	SUBW	AX, #word	3	1	—	AX, CY ← AX − word	×	×	×
		AX, BC	1	1	—	AX, CY ← AX − BC	×	×	×
		AX, DE	1	1	—	AX, CY ← AX − DE	×	×	×
		AX, HL	1	1	—	AX, CY ← AX − HL	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX − (addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX − (ES:addr16)	×	×	×
		AX, saddrp	2	1	—	AX, CY ← AX − (saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX − (HL + byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX − ((ES:HL) + byte)	×	×	×
	CMPW	AX, #word	3	1	—	AX − word	×	×	×
		AX, BC	1	1	—	AX − BC	×	×	×
		AX, DE	1	1	—	AX − DE	×	×	×
		AX, HL	1	1	—	AX − HL	×	×	×
		AX, !addr16	3	1	4	AX − (addr16)	×	×	×
		AX, ES:!addr16	4	2	5	AX − (ES:addr16)	×	×	×
		AX, saddrp	2	1	—	AX − (saddrp)	×	×	×
		AX, [HL+byte]	3	1	4	AX − (HL + byte)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX − ((ES:HL) + byte)	×	×	×
Multiply	MULU	X	1	1	—	AX ← A × X			
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			

Table 25-5. Operation List (10/13)

Instruction Group	Mnemonic	Operand	Bytes	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
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			×
Rotate	ROR	A, 1	2	1	—	(CY, A_7 ← A_0, A_{m-1} ← A_m) × 1			×
	ROL	A, 1	2	1	—	(CY, A_0 ← A_7, A_{m+1} ← A_m) × 1			×
	RORC	A, 1	2	1	—	(CY ← A_0, A_7 ← CY, A_{m-1} ← A_m) × 1			×
	ROLC	A, 1	2	1	—	(CY ← A_7, A_0 ← CY, A_{m+1} ← A_m) × 1			×
	ROLWC	AX,1	2	1	—	(CY ← AX_{15}, AX_0 ← CY, AX_{m+1} ← AX_m) × 1			×
		BC,1	2	1	—	(CY ← BC_{15}, BC_0 ← CY, BC_{m+1} ← BC_m) × 1			×
Bit manipulate	MOV1	CY, A.bit	2	1	—	CY ← A.bit			×
		A.bit, CY	2	1	—	A.bit ← CY			
		CY, PSW.bit	3	1	—	CY ← PSW.bit			×
		PSW.bit, CY	3	4	—	PSW.bit ← CY	×	×	
		CY, saddr.bit	3	1	—	CY ← (saddr).bit			×
		saddr.bit, CY	3	2	—	(saddr).bit ← CY			
		CY, sfr.bit	3	1	—	CY ← sfr.bit			×
		sfr.bit, CY	3	2	—	sfr.bit ← CY			
		CY,[HL].bit	2	1	4	CY ← (HL).bit			×
		[HL].bit, CY	2	2	—	(HL).bit ← CY			
		CY, ES:[HL].bit	3	2	5	CY ← (ES, HL).bit			×
		ES:[HL].bit, CY	3	3	—	(ES, HL).bit ← CY			
	AND1	CY, A.bit	2	1	—	CY ← CY ∧ A.bit			×
		CY, PSW.bit	3	1	—	CY ← CY ∧ PSW.bit			×
		CY, saddr.bit	3	1	—	CY ← CY ∧ (saddr).bit			×
		CY, sfr.bit	3	1	—	CY ← CY ∧ sfr.bit			×
		CY,[HL].bit	2	1	4	CY ← CY ∧ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∧ (ES, HL).bit			×
	OR1	CY, A.bit	2	1	—	CY ← CY ∨ A.bit			×
		CY, PSW.bit	3	1	—	CY ← CY ∨ PSW.bit			×
		CY, saddr.bit	3	1	—	CY ← CY ∨ (saddr).bit			×
		CY, sfr.bit	3	1	—	CY ← CY ∨ sfr.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ∨ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∨ (ES, HL).bit			×

Table 25-5. Operation List (11/13)

Instruction Group	Mnemonic	Operand	Bytes	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Bit manipulate	XOR1	CY, A.bit	2	1	—	CY ← CY ∀ A.bit			×
		CY, PSW.bit	3	1	—	CY ← CY ∀ PSW.bit			×
		CY, saddr.bit	3	1	—	CY ← CY ∀ (saddr).bit			×
		CY, sfr.bit	3	1	—	CY ← CY ∀ sfr.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ∀ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∀ (ES, HL).bit			×
	SET1	A.bit	2	1	—	A.bit ← 1			
		PSW.bit	3	4	—	PSW.bit ← 1	×	×	×
		!addr16.bit	4	2	—	(addr16).bit ← 1			
		ES:!addr16.bit	5	3	—	(ES, addr16).bit ← 1			
		saddr.bit	3	2	—	(saddr).bit ← 1			
		sfr.bit	3	2	—	sfr.bit ← 1			
		[HL].bit	2	2	—	(HL).bit ← 1			
		ES:[HL].bit	3	3	—	(ES, HL).bit ← 1			
	CLR1	A.bit	2	1	—	A.bit ← 0			
		PSW.bit	3	4	—	PSW.bit ← 0	×	×	×
		!addr16.bit	4	2	—	(addr16).bit ← 0			
		ES:!addr16.bit	5	3	—	(ES, addr16).bit ← 0			
		saddr.bit	3	2	—	(saddr).bit ← 0			
		sfr.bit	3	2	—	sfr.bit ← 0			
		[HL].bit	2	2	—	(HL).bit ← 0			
		ES:[HL].bit	3	3	—	(ES, HL).bit ← 0			
	SET1	CY	2	1	—	CY ← 1			1
	CLR1	CY	2	1	—	CY ← 0			0
	NOT1	CY	2	1	—	CY ← $\overline{CY}$			×
Call/return	CALL	rp	2	3	—	(SP − 2) ← (PC + 2)$_S$, (SP − 3) ← (PC + 2)$_H$, (SP − 4) ← (PC + 2)$_L$, PC ← CS, rp, SP ← SP − 4			
		$!addr20	3	3	—	(SP − 2) ← (PC + 3)$_S$, (SP − 3) ← (PC + 3)$_H$, (SP − 4) ← (PC + 3)$_L$, PC ← PC + 3 + jdisp16, SP ← SP − 4			
		!addr16	3	3	—	(SP − 2) ← (PC + 3)$_S$, (SP − 3) ← (PC + 3)$_H$, (SP − 4) ← (PC + 3)$_L$, PC ← 0000, addr16, SP ← SP − 4			
		!!addr20	4	3	—	(SP − 2) ← (PC + 4)$_S$, (SP − 3) ← (PC + 4)$_H$, (SP − 4) ← (PC + 4)$_L$, PC ← addr20, SP ← SP − 4			
	CALLT	[addr5]	2	5	—	(SP − 2) ← (PC + 2)$_S$, (SP − 3) ← (PC + 2)$_H$, (SP − 4) ← (PC + 2)$_L$, PCS ← 0000, PC$_H$ ← (0000, addr5 + 1), PC$_L$ ← (0000, addr5), SP ← SP − 4			
	BRK	—	2	5	—	(SP − 1) ← PSW, (SP − 2) ← (PC + 2)$_S$, (SP − 3) ← (PC + 2)$_H$, (SP − 4) ← (PC + 2)$_L$, PC$_S$ ← 0000, PC$_H$ ← (0007FH), PC$_L$ ← (0007EH), SP ← SP − 4, IE ← 0			
	RET	—	1	6	—	PC$_L$ ← (SP), PC$_H$ ← (SP + 1), PC$_S$ ← (SP + 2), SP ← SP + 4			

Table 25-5. Operation List (12/13)

Instruction Group	Mnemonic	Operand	Bytes	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Call/return	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
Stack manipulate	PUSH	PSW	2	1	—	$(SP - 1) \leftarrow PSW$, $(SP - 2) \leftarrow 00H$, $SP \leftarrow SP - 2$			
		rp	1	1	—	$(SP - 1) \leftarrow rp_H$, $(SP - 2) \leftarrow rp_L$, $SP \leftarrow SP - 2$			
	POP	PSW	2	3	—	$PSW \leftarrow (SP + 1)$, $SP \leftarrow SP + 2$	R	R	R
		rp	1	1	—	$rp_L \leftarrow (SP)$, $rp_H \leftarrow (SP + 1)$, $SP \leftarrow SP + 2$			
	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 5]	—	PC ← PC + 2 + jdisp8 if CY = 1			
	BNC	$addr20	2	2/4[Note 5]	—	PC ← PC + 2 + jdisp8 if CY = 0			
	BZ	$addr20	2	2/4[Note 5]	—	PC ← PC + 2 + jdisp8 if Z = 1			
	BNZ	$addr20	2	2/4[Note 5]	—	PC ← PC + 2 + jdisp8 if Z = 0			
	BH	$addr20	3	2/4[Note 5]	—	PC ← PC + 3 + jdisp8 if $(Z \vee CY) = 0$			
	BNH	$addr20	3	2/4[Note 5]	—	PC ← PC + 3 + jdisp8 if $(Z \vee CY) = 1$			
	BT	saddr.bit, $addr20	4	3/5[Note 5]	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 1			
		sfr.bit, $addr20	4	3/5[Note 5]	—	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
		A.bit, $addr20	3	3/5[Note 5]	—	PC ← PC + 3 + jdisp8 if A.bit = 1			
		PSW.bit, $addr20	4	3/5[Note 5]	—	PC ← PC + 4 + jdisp8 if PSW.bit = 1			
		[HL].bit, $addr20	3	3/5[Note 5]	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 1			
		ES:[HL].bit, $addr20	4	4/6[Note 5]	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1			
	BF	saddr.bit, $addr20	4	3/5[Note 5]	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 0			
		sfr.bit, $addr20	4	3/5[Note 5]	—	PC ← PC + 4 + jdisp8 if sfr.bit = 0			
		A.bit, $addr20	3	3/5[Note 5]	—	PC ← PC + 3 + jdisp8 if A.bit = 0			
		PSW.bit, $addr20	4	3/5[Note 5]	—	PC ← PC + 4 + jdisp8 if PSW.bit = 0			
		[HL].bit, $addr20	3	3/5[Note 5]	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 0			
		ES:[HL].bit, $addr20	4	4/6[Note 5]	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 0			

Table 25-5. Operation List (13/13)

Instruction Group	Mnemonic	Operand	Bytes	Clock		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Conditional branch	BTCLR	saddr.bit, $addr20	4	3/5[Note 5]	—	PC ← PC + 4 + jdisp8 if (saddr).bit = 1 then reset (saddr).bit			
		sfr.bit, $addr20	4	3/5[Note 5]	—	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit			
		A.bit, $addr20	3	3/5[Note 5]	—	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit			
		PSW.bit, $addr20	4	3/5[Note 5]	—	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	×	×	×
		[HL].bit, $addr20	3	3/5[Note 5]	—	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit			
		ES:[HL].bit, $addr20	4	4/6[Note 5]	—	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 6]	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. Except r = A

Note 4. Except rp = AX

Note 5. This indicates the number of clocks “when condition is not met/when condition is met”.

Note 6. n indicates the register bank number (n = 0 to 3).

Remark 1. These numbers of clock cycles apply when the program is in the internal ROM (flash memory) area. When the instruction is fetched from the internal RAM area, the number is, at most, the double of the number given here plus 3 further clock cycles.

Remark 2. cnt indicates the bit shift count.

CHAPTER 26 ELECTRICAL SPECIFICATIONS (T_A = −40 to +85°C)

This chapter describes the electrical specifications of A: Consumer applications (T_A = −40 to +85°C), G: Industrial applications (T_A = −40 to +105°C), and M: Industrial applications (T_A = −40 to +125°C) when they are used in the range of T_A = −40 to +85°C.

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 Function for the port functions and 2.2.1 Functions for each product for the other functions.**

Remark There are differences in the high-speed on-chip oscillator clock accuracy between G: Industrial applications and M: Industrial applications, and A: Consumer applications.

Classification	A: Consumer applications	G: Industrial applications	M: Industrial applications
High-speed on-chip oscillator clock accuracy	±2.0% when T_A = −40 to +85°C	±1.5% when T_A = +85 to +105°C ±1.0% when T_A = −20 to +85°C ±1.5% when T_A = −40 to −20°C	±1.5% when T_A = +85 to +125°C ±1.0% when T_A = −20 to +85°C ±1.5% when T_A = −40 to −20°C

26.1 Absolute Maximum Ratings

[T_A = 25°C]

Item	Symbol	Condition		Rating	Unit
Supply voltage	V_{DD}			−0.5 to +6.5	V
Input voltage	V_{I1}			−0.3 to V_{DD} + 0.3[Note 1]	V
Output voltage	V_{O1}			−0.3 to V_{DD} + 0.3	V
Output current, high	I_{OH1}	Per pin		−40	mA
		Total of all pins	P06, P07, P20 to P23, P40 to P43, P121, P122, P125	−70	mA
		−170mA	P00 to P05, P10 to P17	−100	mA
Output current, low	I_{OL1}	Per pin		40	mA
		Total of all pins	P06, P07, P20 to P23, P40 to P43, P121, P122, P125	100	mA
		170mA	P00 to P05, P10 to P17, P60, P61	100	mA
Operating ambient temperature	T_A			−40 to +85	°C
Storage temperature	T_{stg}			−65 to +150	°C

Note 1. This must be no greater than 6.5 V.

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any item. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark 1. The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

Remark 2. The reference voltage is V_{SS}.

26.2 Oscillator Characteristics

26.2.1 X1 and XT1 oscillator characteristics

[T_A = −40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Resonator	Condition	MIN.	TYP.	MAX.	Unit
X1 clock oscillation frequency (fx)Note 1	Ceramic resonator/ crystal resonator	2.4 V ≤ V_{DD} ≤ 5.5 V	1		12	MHz
XT1 clock oscillation frequency (f_{XT})Note 1	Crystal resonator		32	32.768	35	kHz

Note 1. Indicates only permissible oscillator frequency ranges. Refer to **26.4 AC Characteristics** for instruction execution time. For actual applications, request evaluation by the manufacturer of the oscillator circuit mounted on a board so you can use appropriate values.

Caution Since the CPU is started by the high-speed on-chip oscillator clock after release from the reset state, the user should use the oscillation stabilization time counter status register (OSTC) to check the X1 clock oscillation stabilization time. Specify the values for the oscillation stabilization time in the OSTC register and the oscillation stabilization time select register (OSTS) after having sufficiently evaluated the oscillation stabilization time with the resonator to be used.

Remark When using the X1 or XT1 oscillator, refer to **5.4 System Clock Oscillator**.

26.2.2 On-chip oscillator characteristics

[T_A = −40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator clock frequencyNote 1, Note 2	f_{IH}		1		16	MHz
High-speed on-chip oscillator clock frequency accuracy		T_A = −40 to +85°C	−2.0		+2.0	%
Low-speed on-chip oscillator clock frequency	f_{IL}			15		kHz
Low-speed on-chip oscillator clock frequency accuracy			−15		+15	%

Note 1. The high-speed on-chip oscillator frequency is selected by bits 0 to 2 of option byte (00C2H) and bits 0 to 2 of the HOCODIV register.

Note 2. The listed values only indicate the characteristics of the oscillators. Refer to **26.4 AC Characteristics** for instruction execution time.

26.3 DC Characteristics

26.3.1 Pin characteristics

[T_A = −40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

(1/2)

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Output current, high Note 1	I_{OH1}	Per pin for P00 to P07, P10 to P17, P20 to P23, P40 to P43, P121, P122, P125				−10.0 Note 2	mA
		Total of P06, P07, P20 to P23, P40 to P43, P121, P122, P125 (when duty ≤ 70% Note 3)	4.0 V ≤ V_{DD} ≤ 5.5 V			−65.0 Note 5	mA
			2.7 V ≤ V_{DD} < 4.0 V			−14.0	mA
			2.4 V ≤ V_{DD} < 2.7 V			−10.5	mA
		Total of P00 to P05, P10 to P17 (when duty ≤ 70% Note 3)	4.0 V ≤ V_{DD} ≤ 5.5 V			−65.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			−12.0	mA
			2.4 V ≤ V_{DD} < 2.7 V			−9.0	mA
		Total of all pins (when duty ≤ 70% Note 3)				−105.0	mA
Output current, low Note 4	I_{OL1}	Per pin for P00 to P07, P10 to P17, P20 to P23, P40 to P43, P121, P122, P125				20.0 Note 2	mA
		Per pin for P60, P61				15.0 Note 2	mA
		Total of P06, P07, P20 to P23, P40 to P43, P121, P122, P125 (when duty ≤ 70% Note 3)	4.0 V ≤ V_{DD} ≤ 5.5 V			85.0 Note 6	mA
			2.7 V ≤ V_{DD} < 4.0 V			21.0	mA
			2.4 V ≤ V_{DD} < 2.7 V			4.2	mA
		Total of P00 to P05, P10 to P17, P60, P61 (when duty ≤ 70% Note 3)	4.0 V ≤ V_{DD} ≤ 5.5 V			85.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			18.0	mA
			2.4 V ≤ V_{DD} < 2.7 V			3.6	mA
		Total of all pins (when duty ≤ 70% Note 3)				145.0	mA

Note 1. Device operation is guaranteed at the listed currents even if current is flowing from the V_{DD} pin to an output pin.

Note 2. The value for maximum total current must not be exceeded.

Note 3. The listed currents apply when the duty cycle is no greater than 70%. Use the following formula to calculate the output current when the duty cycle is greater than 70%, where n is the duty cycle.

- Total output current from the listed pins = (I_{OH} × 0.7)/(n × 0.01)
  Example when n = 80% and I_{OH} = –10.0 mA
  Total output current from the listed pins = (–10.0 × 0.7)/(80 × 0.01) ≅ –8.7 mA
- Total output current from the listed pins = (I_{OL} × 0.7)/(n × 0.01)
  Example when n = 80% and I_{OL} = 10.0 mA
  Total output current from the listed pins = (10.0 × 0.7)/(80 × 0.01) ≅ 8.7 mA

Note that the duty cycle has no effect on the current that is allowed to flow into a single pin. A current higher than the absolute maximum rating must not flow into a single pin.

Note 4. Device operation is guaranteed at the listed currents even if current is flowing from an output pin to the V_{SS} pin.

Note 5. When the multiplexed TSm pin is used as a mutual-capacitance transmit pin, the value is –30 mA.

Note 6. When the multiplexed TSm pin is used as a mutual-capacitance transmit pin, the value is 40 mA.

Caution P00, P01, P03 to P07, P14, P16, P17, P20, P22, and P41 do not output high level in N-ch open-drain mode.

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

[T_A = −40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

(2/2)

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}			0.8 V_{DD}		V_{DD}	V
	V_{IH2}	P60, P61		0.7 V_{DD}		6.0	V
Input voltage, low	V_{IL1}			0		0.2 V_{DD}	V
	V_{IL2}	P60, P61		0		0.3 V_{DD}	V
Output voltage, high Note 1	V_{OH1}	4.0 V ≤ V_{DD} ≤ 5.5 V	I_{OH} = −10 mA	V_{DD} − 1.5			V
			I_{OH} = −3.0 mA	V_{DD} − 0.7			V
		2.7 V ≤ V_{DD} ≤ 5.5 V	I_{OH} = −2.0 mA	V_{DD} − 0.6			V
		2.4 V ≤ V_{DD} ≤ 5.5 V	I_{OH} = −1.5 mA	V_{DD} − 0.5			V
Output voltage, low Note 2	V_{OL1}	4.0 V ≤ V_{DD} ≤ 5.5 V	I_{OL} = 20 mA			1.3	V
			I_{OL} = 8.5 mA			0.7	V
		2.7 V ≤ V_{DD} ≤ 5.5 V	I_{OL} = 3.0 mA			0.6	V
			I_{OL} = 1.5 mA			0.4	V
		2.4 V ≤ V_{DD} ≤ 5.5 V	I_{OL} = 0.6 mA			0.4	V
		P60, P61 4.0 V ≤ V_{DD} ≤ 5.5 V	I_{OL} = 15 mA			2.0	V
			I_{OL} = 5 mA			0.4	V
		P60, P61 2.7 V ≤ V_{DD} ≤ 5.5 V	I_{OL} = 3.0 mA			0.4	V
			I_{OL} = 2.0 mA			0.4	V
		P60, P61 2.4 V ≤ V_{DD} ≤ 5.5 V	I_{OL} = 1.0 mA			0.4	V
Input leakage current, high	I_{LIH1}	P00 to P07, P10 to P17, P20 to P23, P40 to P43, P60, P61, P125, P137 V_I = V_{DD}				1	μA
	I_{LIH2}	P121, P122 (X1, X2, XT1, XT2, EXCLK, EXCLKS) V_I = V_{DD}	In input port or external clock input			1	μA
			In resonator connection			10	μA
Input leakage current, low	I_{LIL1}	P00 to P07, P10 to P17, P20 to P23, P40 to P43, P60, P61, P125, P137 V_I = V_{SS}				−1	μA
	I_{LIL2}	P121, P122 (X1, X2, XT1, XT2, EXCLK, EXCLKS) V_I = V_{SS}	In input port or external clock input			−1	μA
			In resonator connection			−10	μA
On-chip pull-up resistance	R_U	Except P60 and P61 V_I = V_{SS}		10	20	100	kΩ

Note 1. The value under the condition which satisfies the high-level output current (I_{OH1}).

Note 2. The value under the condition which satisfies the low-level output current (I_{OL1}).

Caution The maximum value of V_{IH} of P00, P01, P03 to P07, P14, P16, P17, P20, P22, and P41 is V_{DD} even in N-ch open-drain mode. These pins do not output high level in N-ch open-drain mode.

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

26.3.2 Supply current characteristics

[T_A = −40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Symbol	Condition				MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD1}	Operating mode	Basic operation	f_{IH} = 16 MHz[Note 4]	V_{DD} = 3.0 V, 5.0 V		0.97		mA
			Normal operation	f_{IH} = 16 MHz[Note 4]	V_{DD} = 3.0 V, 5.0 V		2.11	2.76	mA
				f_{IH} = 4 MHz[Note 4]	V_{DD} = 3.0 V, 5.0 V		1.22	1.64	mA
				f_{EX} = 16 MHz[Note 5, Note 6] V_{DD} = 3.0 V, 5.0 V	Square wave input		1.97	2.62	mA
				f_X = 12 MHz[Note 5, Note 6] V_{DD} = 3.0 V, 5.0 V	Resonator connection		1.79	2.49	mA
				f_{MX} = 4 MHz[Note 5, Note 6] V_{DD} = 3.0 V, 5.0 V	Square wave input		1.07	1.49	mA
					Resonator connection		1.12	1.55	mA
				f_{SUB} = 32.768 kHz[Note 7] T_A = −40°C	Square wave input		3.65	5.80	µA
					Resonator connection		3.70	6.00	µA
				f_{SUB} = 32.768 kHz[Note 7] T_A = +25°C	Square wave input		3.90	5.80	µA
					Resonator connection		4.18	6.00	µA
				f_{SUB} = 32.768 kHz[Note 7] T_A = +50°C	Square wave input		4.04	6.20	µA
					Resonator connection		4.37	6.40	µA
				f_{SUB} = 32.768 kHz[Note 7] T_A = +70°C	Square wave input		4.20	6.50	µA
					Resonator connection		4.56	6.70	µA
				f_{SUB} = 32.768 kHz[Note 7] T_A = +85°C	Square wave input		4.40	7.80	µA
					Resonator connection		4.80	8.00	µA
	I_{DD2}[Note 2]	HALT mode		f_{IH} = 16 MHz[Note 4]	V_{DD} = 3.0 V, 5.0 V		385	800	µA
				f_{IH} = 4 MHz[Note 4]	V_{DD} = 3.0 V, 5.0 V		334	630	µA
				f_{EX} = 16 MHz[Note 5, Note 6] V_{DD} = 3.0 V, 5.0 V	Square wave input		229	638	µA
				f_X = 12 MHz[Note 5, Note 6] V_{DD} = 3.0 V, 5.0 V	Resonator connection		351	902	µA
				f_{MX} = 4 MHz[Note 5, Note 6] V_{DD} = 3.0 V, 5.0 V	Square wave input		167	452	µA
					Resonator connection		226	599	µA
				f_{SUB} = 32.768 kHz[Note 8] T_A = −40°C	Square wave input		0.69	1.45	µA
					Resonator connection		0.75	1.65	µA
				f_{SUB} = 32.768 kHz[Note 8] T_A = +25°C	Square wave input		0.75	1.45	µA
					Resonator connection		1.04	1.65	µA
				f_{SUB} = 32.768 kHz[Note 8] T_A = +50°C	Square wave input		0.84	1.74	µA
					Resonator connection		1.20	1.94	µA
				f_{SUB} = 32.768 kHz[Note 8] T_A = +70°C	Square wave input		0.97	2.20	µA
					Resonator connection		1.33	2.40	µA
				f_{SUB} = 32.768 kHz[Note 8] T_A = +85°C	Square wave input		1.13	3.10	µA
					Resonator connection		1.51	3.30	µA
	I_{DD3}[Note 3]	STOP mode[Note 9]		V_{DD} = 3.0 V			0.62	2.80	µA

Note 1. The listed currents are the total currents flowing into V_{DD}, including the input leakage currents flowing when the level of the input pin is fixed to V_{DD} or V_{SS}.
Regarding the values for main system clock operation, the TYP. value does not include the peripheral operating current. The MAX. value includes the peripheral operating current, but does not include those flowing into the A/D converter, comparator, capacitive touch sensing unit (CTSU), I/O port, and on-chip pull-

up/pull-down resistors.
Regarding the values for subsystem clock operation, the TYP. and MAX. values do not include the peripheral operating current. However, in HALT mode, the current flowing into the RTC is included.
Regarding the values in STOP mode, the TYP. and MAX. values do not include the peripheral operating current.

Note 2. When the HALT instruction is executed from the flash memory.

Note 3. The listed currents do not include the current flowing into real-time clock 2, the clock output/buzzer output controller, 12-bit interval timer, and watchdog timer.

Note 4. When the high-speed system clock and subsystem clock are stopped.

Note 5. When the high-speed on-chip oscillator and subsystem clock are stopped.

Note 6. 16-pin, 20-pin, 24-pin, and 32-pin products only.

Note 7. When the high-speed on-chip oscillator and high-speed system clock are stopped. When AMPHS1 = 1 (ultra-low power consumption oscillation). However, not including the current flowing into real-time clock 2, the 12-bit interval timer, watchdog timer, and capacitive touch sensing unit.

Note 8. When the high-speed on-chip oscillator and high-speed system clock are stopped. When RTCLPC = 1 and AMPHS1 = 1 (ultra-low power consumption oscillation).

Note 9. Regarding the value for current to operate the subsystem clock in STOP mode, refer to that in HALT mode.

Remark 1. f_{IH}: High-speed on-chip oscillator clock frequency

Remark 2. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

Remark 3. Except for subsystem clock operation, the temperature condition of the TYP. value is T_A = 25°C.

Remark 4. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

Peripheral Functions

[T_A = −40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Low-speed on-chip oscillator operating current	I_{FIL} Note 1				0.30		μA
RTC2 operating current	I_{RTC} Note 1, Note 2, Note 8	f_{SUB} = 32.768 kHz			0.02		μA
12-bit interval timer operating current	I_{TMKA} Note 1, Note 2, Note 3				0.02		μA
Watchdog timer operating current	I_{WDT} Note 1, Note 4				0.02		μA
A/D converter operating current	I_{ADC} Note 1, Note 5	In conversion at maximum speed	V_{DD} = 5.0 V		1.30	1.90	mA
			V_{DD} = 3.0 V		0.50		mA
Temperature sensor operating current	I_{TMPS} Note 1				75.0		μA
Comparator operating current	I_{CMP} Note 1, Note 6	In high-speed mode	V_{DD} = 5.0 V		6.50		μA
		In low-speed mode	V_{DD} = 5.0 V		1.70		μA
Internal reference voltage operating current	I_{VREG} Note 1				10		μA
Self-programming operating current	I_{FSP} Note 1, Note 7				2.0	12.20	mA

Note 1. The current flowing into V_{DD}.

Note 2. When the high-speed on-chip oscillator and high-speed system clock are stopped.

Note 3. This current only flows into the 12-bit interval timer. It does not include the operating current of the low-speed on-chip oscillator and XT1 oscillator. The supply current of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{FIL} and I_{TMKA} when the 12-bit interval timer is in operation.

Note 4. This current only flows into the watchdog timer. It does not include 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_{FIL} and I_{WDT} when the watchdog timer is in operation.

Note 5. This current only flows into 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 is operating or in the HALT mode.

Note 6. This current only flows into a single comparator. The supply current of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{CMP} when the comparator is in operation.

Note 7. This current only flows during self-programming.

Note 8. This current only flows into real-time clock 2. It does not include the operating current of the low-speed on-chip oscillator and XT1 oscillator. The supply current of the RL78 microcontrollers is the sum of the values of either I_{DD1} or I_{DD2}, and I_{RTC}, when real-time clock 2 operates in operation mode or HALT mode. When the low-speed on-chip oscillator is selected, I_{FIL} should be added. I_{DD2} subsystem clock operation includes the operational current of real-time clock 2.

Remark The temperature condition of the TYP. value is T_A = 25°C.

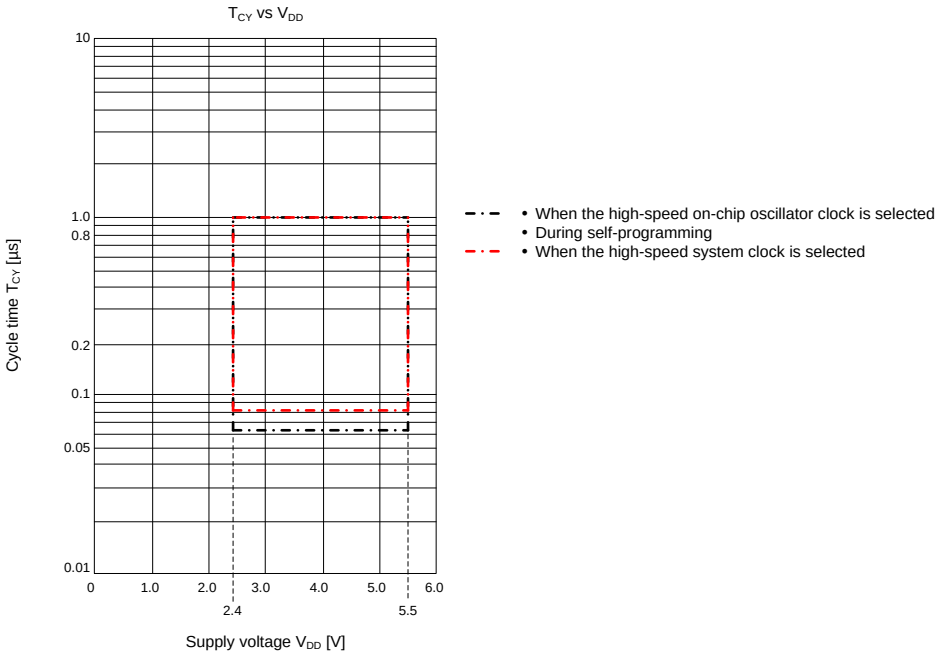
26.4 AC Characteristics

[T_A = −40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

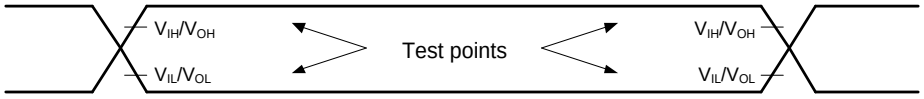
Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	When high-speed on-chip oscillator clock (f_{IH}) is selected	2.4 V ≤ V_{DD} ≤ 5.5 V	0.0625		1.0	µs
		When high-speed system clock (f_{MX}) is selected	2.4 V ≤ V_{DD} ≤ 5.5 V	0.0833		1.0	µs
		Subsystem clock (f_{SUB}) operation	2.4 V ≤ V_{DD} ≤ 5.5 V	28.5	30.5	31.3	µs
		In the self-programming mode	2.4 V ≤ V_{DD} ≤ 5.5 V	0.0625		1.0	µs
External system clock frequency	f_{EX}	2.4 V ≤ V_{DD} ≤ 5.5 V		1.0		16	MHz
External system clock input high-level width, low-level width	t_{EXH}, t_{EXL}	2.4 V ≤ V_{DD} ≤ 5.5 V		30			ns
TI00 to TI07 input high-level width, low-level width	t_{TIH}, t_{TIL}	Noise filter is not used		$1/f_{MCK}$ + 10			ns
TO00 to TO07 output frequency	f_{TO}	4.0 V ≤ V_{DD} ≤ 5.5 V				8	MHz
		2.7 V ≤ V_{DD} < 4.0 V				5	MHz
		2.4 V ≤ V_{DD} < 2.7 V				4	MHz
PCLBUZ0 output frequency	f_{PCL}	4.0 V ≤ V_{DD} ≤ 5.5 V				10	MHz
		2.7 V ≤ V_{DD} < 4.0 V				5	MHz
		2.4 V ≤ V_{DD} < 2.7 V				4	MHz
$\overline{\text{RESET}}$ low-level width	t_{RSL}			10			µs

Remark f_{MCK}: Timer array unit operating clock frequency
(Operation clock to be set by timer clock select register 0 (TPS0) and the CKS0n1 bit of timer mode register 0n (TMR0n). n: Channel number (n = 0 to 7).)

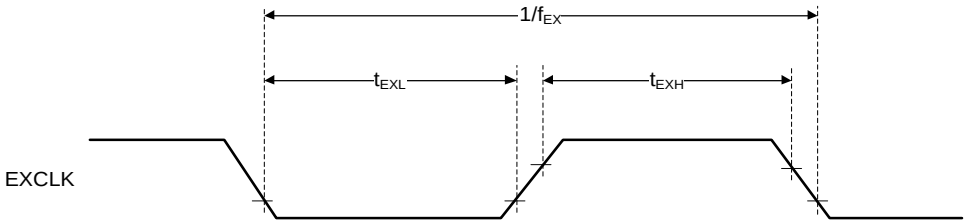
Minimum Instruction Execution Time during Main System Clock Operation



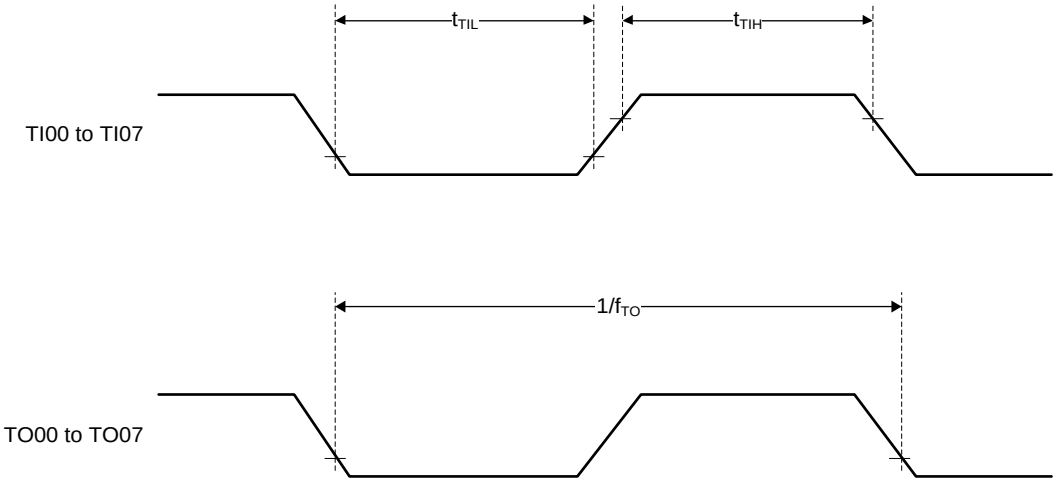
At AC Timing



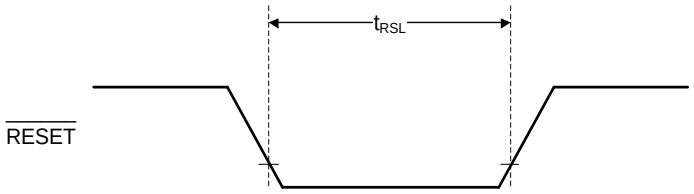
External System Clock Timing



TI/TO Timing

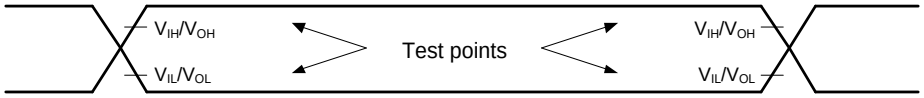


$\overline{\text{RESET}}$ Input Timing



26.5 Serial Interface Characteristics

AC Timing Test Points



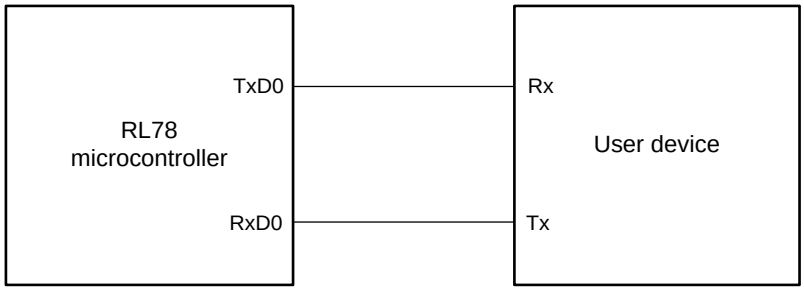
26.5.1 Serial array unit

(1) UART mode

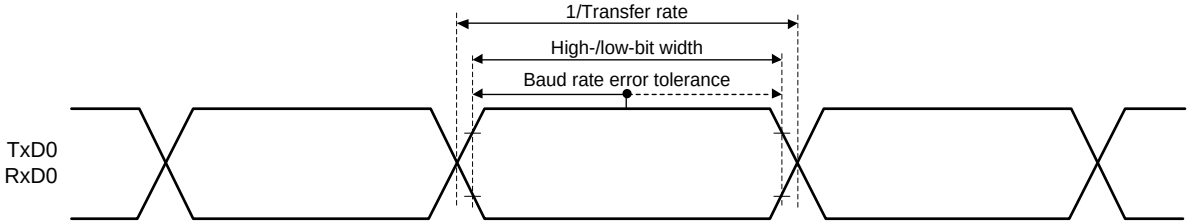
[T_A = −40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Transfer rate					f_{MCK}/6	bps
		Theoretical value of the maximum transfer rate f_{CLK} = f_{MCK} = 16 MHz			2.6	Mbps

UART mode connection diagram



UART mode bit width (reference)



Remark f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 11, 20))

(2) Simplified SPI (CSI) mode (master mode, SCKp... internal clock output)

[T_A = −40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
SCKp cycle time	t_{KCY1}	$t_{KCY1} \geq 4/f_{CLK}$	2.4 V ≤ V_{DD} ≤ 5.5 V	250			ns
SCKp high-/low-level width	t_{KH1}, t_{KL1}	2.7 V ≤ V_{DD} ≤ 5.5 V		$t_{KCY1}/2$ − 18			ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		$t_{KCY1}/2$ − 38			ns
SIp setup time (to SCKp ↑)[Note 1]	t_{SIK1}	2.7 V ≤ V_{DD} ≤ 5.5 V		47			ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		75			ns
SIp hold time (from SCKp ↑)[Note 1]	t_{KSI1}			19			ns
Delay time from SCKp ↓ to SOp output[Note 2]	t_{KSO1}	C = 30 pF[Note 3, Note 4]				25	ns

Note 1. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The SIp setup time becomes “to SCKp ↓” and the SIp hold time becomes “from SCKp ↓” when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

Note 2. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The delay time to SOp output becomes “from SCKp ↓” when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

Note 3. C is the load capacitance of the SCKp and SOp output lines.

Note 4. External loads on P60 and P61 in the case of the SOp output line: R = 1 kΩ, C = 30 pF

(3) Simplified SPI (CSI) mode (slave mode, SCKp... external clock input)

[T_A = −40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5V, V_{SS} = 0V]

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
SCKp cycle time	t_{KCY2}	2.7 V ≤ V_{DD} ≤ 5.5 V		$6/f_{MCK}$			ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		$6/f_{MCK}$ and also 500			ns
SCKp high-/low-level width	t_{KH2},t_{KL2}	2.4 V ≤ V_{DD} ≤ 5.5 V		$t_{KCY2}/2$ − 18			ns
SIp setup time (to SCKp ↑)[Note 1]	t_{SIK2}	2.7 V ≤ V_{DD} ≤ 5.5 V		$1/f_{MCK}$ + 20			ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		$1/f_{MCK}$ + 30			ns
SIp hold time (from SCKp ↑)[Note 1]	t_{KSI2}	2.4 V ≤ V_{DD} ≤ 5.5 V		$1/f_{MCK}$ + 31			ns
Delay time from SCKp ↓ to SOp output[Note 2]	t_{KSO2}	C = 30 pF Note 3, Note 4	2.7 V ≤ V_{DD} ≤ 5.5 V			$2/f_{MCK}$ + 50	ns
			2.4 V ≤ V_{DD} ≤ 5.5 V			$2/f_{MCK}$ + 75	ns

Note 1. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The SIp setup time becomes “to SCKp ↓” and the SIp hold time becomes “from SCKp ↓” when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

Note 2. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The delay time to SOp output becomes “from SCKp ↓” when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

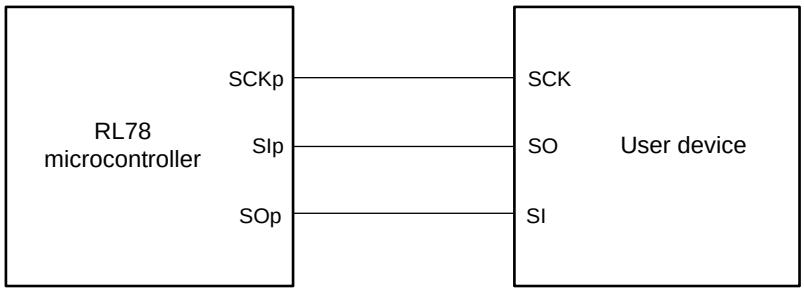
Note 3. C is the load capacitance of the SOp output lines.

Note 4. External loads on P60 and P61 in the case of the SOp output line: R = 1 kΩ, C = 30 pF

Remark 1. p: CSI number (p = 00, 11, 20), n: Channel number (n = 0, 1)

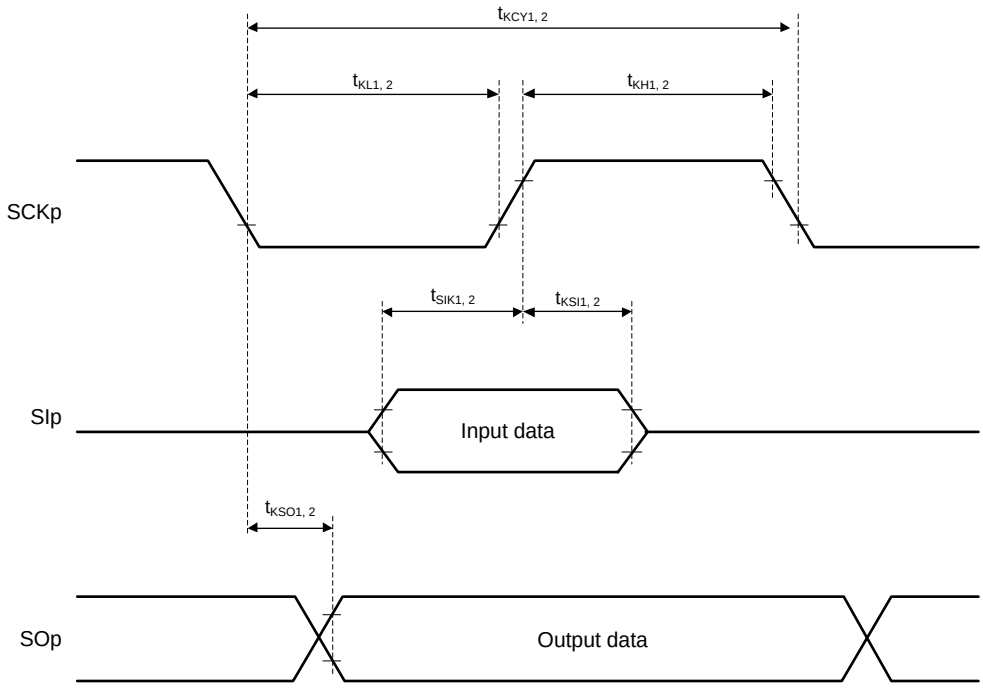
Remark 2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by serial clock select register 0 (SPS0) and the CKS0n bit of serial mode register 0n (SMR0n). n: Channel number (n = 0, 1))

Simplified SPI (CSI) mode connection diagram



Simplified SPI (CSI) mode serial transfer timing

(When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1)



Remark p: CSI number (p = 00, 11, 20), n: Channel number (n = 0, 1)

(4) Simplified I²C mode

[T_A = −40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

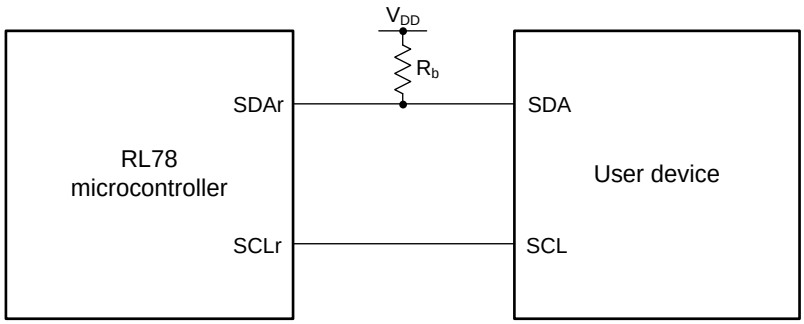
Item	Symbol	Condition	MIN.	MAX.	Unit
SCLr clock frequency	f_{SCL}	C_b = 100 pF, R_b = 3 kΩ		400 Note 1	kHz
Hold time when SCLr = “L”	t_{LOW}	C_b = 100 pF, R_b = 3 kΩ	1150		ns
Hold time when SCLr = “H”	t_{HIGH}	C_b = 100 pF, R_b = 3 kΩ	1150		ns
Data setup time (reception)	$t_{SU:DAT}$	C_b = 100 pF, R_b = 3 kΩ	$1/f_{MCK}$ + 145 Note 2		ns
Data hold time (transmission)	$t_{HD:DAT}$	C_b = 100 pF, R_b = 3 kΩ	0	355	ns

Note 1. The value must also be no greater than $f_{MCK}/4$.

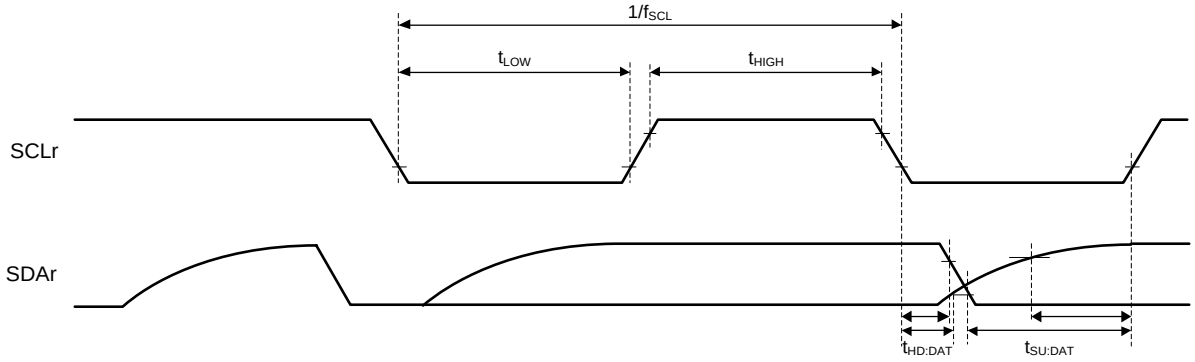
Note 2. Set f_{MCK} so that it will not exceed the hold time when SCLr = “L” or SCLr = “H”.

Caution Select the N-ch open drain output (V_{DD} tolerance) mode for the SDAr pin by using port output mode register 0, 1, 2, 4, or 6 (POM0, 1, 2, 4, or 6).

Simplified I²C mode connection diagram



Simplified I^2C mode serial transfer timing



Remark 1. R_b[Ω]: Communication line (SDAr) pull-up resistance, C_b[F]: Communication line (SCLr, SDAr) load capacitance

Remark 2. r: IIC number (r = 00, 11, 20)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by serial clock select register 0 (SPS0) and the CKS0n bit of serial mode register 0n (SMR0n). n: Channel number (n = 0, 1))

26.5.2 Serial interface IICA

[T_A = −40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Symbol	Condition	Standard Mode		Fast Mode		Unit
			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 or restart condition is detected.

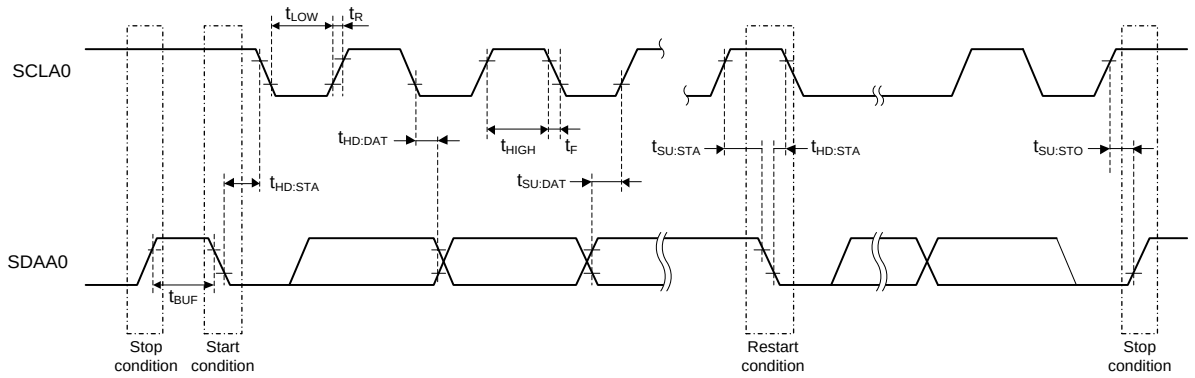
Note 2. The maximum value (MAX.) of $t_{HD:DAT}$ applies to normal transfer and a wait is inserted at the ACK (acknowledge) timing.

Remark The maximum value of C_b (communication line capacitance) and the value of R_b (communication line pull-up resistance) at that time in each mode are as follows.

Standard mode: C_b = 400 pF, R_b = 2.7 kΩ

Fast mode: C_b = 200 pF, R_b = 1.7 kΩ

IICA serial transfer timing



26.6 Analog Characteristics

26.6.1 A/D converter characteristics

Targets: ANI0 to ANI10, internal reference voltage, temperature sensor output voltage, and CTSU TSCAP voltage

[T_A = −40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error Note 1, Note 2, Note 3	AINL	10-bit resolution	V_{DD} = 5 V		±1.7	±3.1	LSB
			V_{DD} = 3 V		±2.3	±4.5	LSB
Conversion time	t_{CONV}	10-bit resolution Targets: ANI0 to ANI10	2.7 V ≤ V_{DD} ≤ 5.5 V	4.25		17	µs
			2.4 V ≤ V_{DD} ≤ 5.5 V Note 5	5.75		23	µs
		10-bit resolution Target: Internal reference voltage Note 6 Target: Temperature sensor output voltage Note 6 Target: CTSU TSCAP voltage Note 6	2.4 V ≤ V_{DD} ≤ 5.5 V	5.75		23	µs
Zero-scale error Note 1, Note 2, Note 3, Note 4	E_{ZS}	10-bit resolution	V_{DD} = 5 V			±0.19	%FSR
			V_{DD} = 3 V			±0.39	%FSR
Full-scale error Note 1, Note 2, Note 3, Note 4	E_{FS}	10-bit resolution	V_{DD} = 5 V			±0.29	%FSR
			V_{DD} = 3 V			±0.42	%FSR
Integral linearity error Note 1, Note 2, Note 3	ILE	10-bit resolution	V_{DD} = 5 V			±1.8	LSB
			V_{DD} = 3 V			±1.7	LSB
Differential linearity error Note 1, Note 2, Note 3	DLE	10-bit resolution	V_{DD} = 5 V			±1.4	LSB
			V_{DD} = 3 V			±1.5	LSB
Analog input voltage	V_{AIN}	Targets: ANI0 to ANI10		0		V_{DD}	V
		Target: Internal reference voltage Note 6		V_{REG} Note 7			V
		Target: Temperature sensor output voltage Note 6		V_{TMPS25} Note 7			V
		Target: CTSU TSCAP voltage (2.4 V ≤ V_{DD} ≤ 5.5 V)		V_{TSCAP}			V

Note 1. The TYP. value is an average value at T_A = 25°C. The MAX. value is an average value ±3σ at normal distribution.

Note 2. These values are the results of characteristic evaluation and are not checked for shipment.

Note 3. A quantization error (±1/2 LSB) is not included.

Note 4. Expressed as a ratio (%FSR) relative to the full-scale value.

Note 5. Be sure to set the LV0 bit in A/D converter mode register 0 (ADM0) to 0 when conversion is done in the operating voltage range of 2.4 V ≤ V_{DD} < 2.7 V.

Note 6. Be sure to set the LV0 bit in A/D converter mode register 0 (ADM0) to 0 when the internal reference voltage, temperature sensor output voltage, or CTSU TSCAP voltage is selected as the target for conversion.

Note 7. Refer to **26.6.3 Temperature sensor/internal reference** voltage characteristics.

Caution 1. **Arrange wiring and insert the capacitor so that no noise appears on the power supply/ground line.**

Caution 2. **Do not allow any pulses that rapidly change such as digital signals to be input/output to/from the pins adjacent to the conversion pin during A/D conversion.**

Caution 3. **Note that the internal reference voltage cannot be used as the reference voltage of the comparator when the internal reference voltage is selected as the target for A/D conversion.**

26.6.2 Comparator characteristics

[T_A = −40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Input voltage range	I_{VREF}	IVREFn pin input (CnVRF bit = 0)		0		V_{DD} − 1.4	V
		Internal reference voltage (CnVRF bit = 1) Note 1		V_{REG} Note 2			V
	I_{VCMP}	IVCMPn pin input		−0.3		V_{DD} + 0.3	V
Output delay	t_d	V_{DD} = 3.0 V, input slew rate > 50 mV/μs	High-speed mode			0.5	μs
			Low-speed mode		2.0		μs
Operation stabilization wait time	t_{CMP}			100			μs

Note 1. When the internal reference voltage is selected as the reference voltage of the comparator, the internal reference voltage cannot be used as the target for A/D conversion.

Note 2. Refer to **26.6.3 Temperature sensor/internal reference voltage characteristics**.

Remark n: Channel number (n = 0, 1)

26.6.3 Temperature sensor/internal reference voltage characteristics

[T_A = −40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Temperature sensor output voltage	V_{TMPS25}			1.05		V
Internal reference voltage	V_{REG}		0.74	0.815	0.89	V
Temperature coefficient	F_{VTMPS}	Temperature dependence of the temperature sensor voltage		−3.6		mV/°C
Operation stabilization wait time	t_{AMP}	A/D converter is used (ADS register = 0DH)	5			μs

Caution **The internal reference voltage cannot be simultaneously used by the A/D converter and the comparator; only one of them must be selected.**

26.6.4 SPOR circuit characteristics

[T_A = −40 to +85°C, V_{SS} = 0 V]

Item		Symbol	Condition	MIN.	TYP.	MAX.	Unit
Detection voltage	Power supply voltage level	V_{SPOR0}	Power supply rising	4.08	4.28	4.45	V
		V_{SPDR0}	Power supply falling	4.00	4.20	4.37	V
		V_{SPOR1}	Power supply rising	2.76	2.90	3.02	V
		V_{SPDR1}	Power supply falling	2.70	2.84	2.96	V
		V_{SPOR2}	Power supply rising	2.44	2.57	2.68	V
		V_{SPDR2}	Power supply falling	2.40	2.52	2.62	V
		V_{SPOR3}	Power supply rising		2.16		V
		V_{SPDR3}	Power supply falling		2.11		V
Minimum pulse width[Note 1]		T_{SPW}		300			µs

Note 1. Time required for the reset operation by the SPOR circuit when V_{DD} falls below V_{SPDR}.

Caution Make sure to keep the internal reset state by the SPOR or an external reset until the power supply voltage (V_{DD}) reaches the operating voltage range shown in 26.4 AC Characteristics.

26.6.5 Power supply voltage rising slope characteristics

[T_A = −40 to +85°C, V_{SS} = 0 V]

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Power supply voltage rising slope	S_{VDD}				54	V/ms

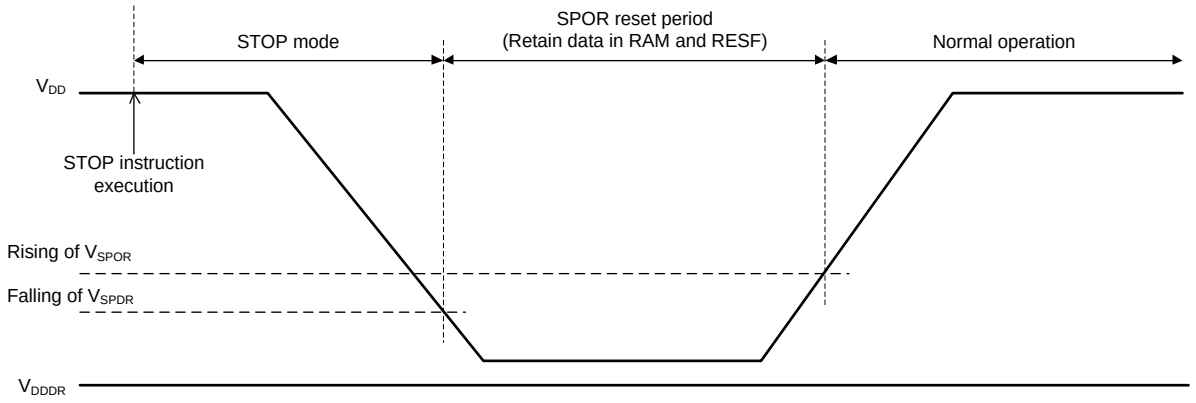
Caution Make sure to keep the internal reset state by the SPOR or an external reset until the power supply voltage (V_{DD}) reaches the operating voltage range shown in 26.4 AC Characteristics.

26.7 RAM Data Retention Characteristics

[T_A = −40 to +85°C, V_{SS} = 0 V]

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Data retention power supply voltage	V_{DDDR}		1.9		5.5	V

Caution Data in RAM is retained until the power supply voltage falls below the MIN. value of the data retention power supply voltage (V_{DDDR}). Note that data in the RESF register might not be cleared even if the power supply voltage falls below the MIN. value of the data retention power supply voltage (V_{DDDR}).



26.8 Flash Memory Programming Characteristics

[T_A = −40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Number of code flash rewrites Note 1, Note 2	C_{erwr}	Retained for 20 years	T_A = +85°C	1000			Times
Number of data flash rewrites Note 1, Note 2		Retained for 1 year	T_A = +25°C		1,000,000		Times
		Retained for 5 years	T_A = +85°C	100,000			Times
		Retained for 20 years	T_A = +85°C	10,000			Times

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. These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas Electronics.

Code flash/data flash self-programming time

[T_A = −40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Symbol	f_{CLK} = 1 MHz			f_{CLK} = 16 MHz			Unit
		MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	
Writing (4 bytes)	t_{P4}		104	905		53.8	504.9	µs
Block erasure (1 KB)	t_{E1K}		7.9	262.3		5.5	214.1	ms

Caution The listed values do not include the time until the operations of the flash memory start following execution of an instruction by software.

26.9 Dedicated Flash Memory Programmer Communication (UART)

[T_A = −40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

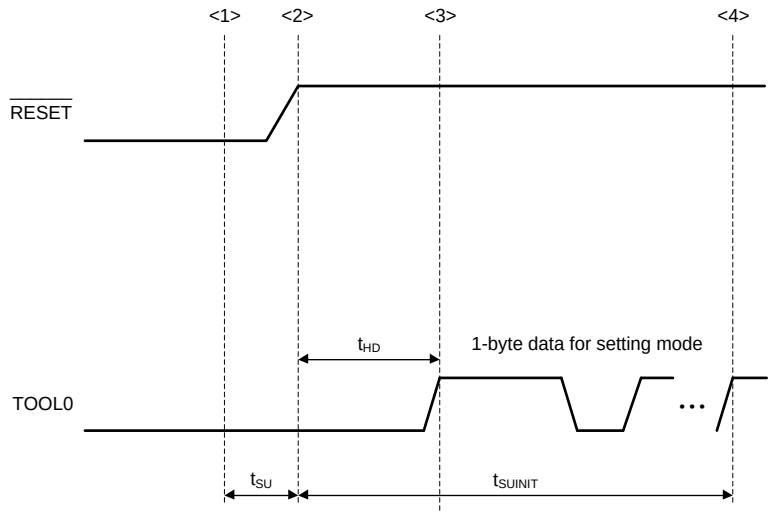
Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Transfer rate				115,200		bps

Remark The transfer rate during flash memory programming is fixed to 115,200 bps.

26.10 Timing of Entry to Flash Memory Programming Mode

[T_A = −40 to +85°C, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Time to complete the communication for the initial setting after the external reset is released	t_{SUINIT}	The SPOR reset must be released before the external reset is released.			100	ms
Time to release the external reset after the TOOL0 pin is set to the low level	t_{SU}	The SPOR reset must be released before the external reset is released.	10			μs
Time to hold the TOOL0 pin at the low level after the external reset is released	t_{HD}	The SPOR reset must be released before the external reset is released.	1			ms



<1> The low level is input to the TOOL0 pin.

<2> The external reset is released (the SPOR reset must have been released before that).

<3> The TOOL0 pin is released from the low level.

<4> Setting of entry to the flash memory programming mode by UART reception is completed.

Remark t_{SUINIT}: During this period, the communications for the initial setting must be completed within 100 ms after release from the reset.

t_{SU}: Time to release the external reset after the TOOL0 pin is set to the low level

t_{HD}: Time to hold the TOOL0 pin at the low level after the external reset is released

CHAPTER 27 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C, T_A = −40 to +125°C)

This chapter describes the electrical specifications of the following target products.

Target product G: Industrial applications T_A = −40 to +105°C

Target product M: Industrial applications T_A = −40 to +125°C

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 Function for the port functions and 2.2.1 Functions for each product for the other functions.

Remark When the products are used in the range of T_A = −40 to +85°C, refer to **CHAPTER 26 Electrical Specifications (T_A = −40 to +85°C)**. However, there are differences in the high-speed on-chip oscillator clock accuracy between G: Industrial applications and M: Industrial applications, and A: Consumer applications.

Classification	A: Consumer applications	G: Industrial applications	M: Industrial applications
High-speed on-chip oscillator clock accuracy	±2.0% when T_A = −40 to +85°C	±1.5% when T_A = +85 to +105°C ±1.0% when T_A = −20 to +85°C ±1.5% when T_A = −40 to −20°C	±1.5% when T_A = +85 to +125°C ±1.0% when T_A = −20 to +85°C ±1.5% when T_A = −40 to −20°C

27.1 Absolute Maximum Ratings

[T_A = 25°C]

Item	Symbol	Condition		Rating	Unit
Supply voltage	V_{DD}			−0.5 to +6.5	V
Input voltage	V_{I1}			−0.3 to V_{DD} + 0.3[Note 1]	V
Output voltage	V_{O1}			−0.3 to V_{DD} + 0.3	V
Output current, high	I_{OH1}	Per pin		−40	mA
		Total of all pins	P06, P07, P20 to P23, P40 to P43, P121, P122, P125	−70	mA
		−170mA	P00 to P05, P10 to P17	−100	mA
Output current, low	I_{OL1}	Per pin		40	mA
		Total of all pins	P06, P07, P20 to P23, P40 to P43, P121, P122, P125	100	mA
		170mA	P00 to P05, P10 to P17, P60, P61	100	mA
Operating ambient temperature	T_A	G products		−40 to +105	°C
		M products		−40 to +125	°C
Storage temperature	T_{stg}			−65 to +150	°C

Note 1. This must be no greater than 6.5 V.

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any item. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark 1. The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

Remark 2. The reference voltage is V_{SS}.

27.2 Oscillator Characteristics

27.2.1 X1 and XT1 oscillator characteristics

[T_A = −40 to +105°C: G products, T_A = −40 to +125°C: M products, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Resonator	Condition	MIN.	TYP.	MAX.	Unit
X1 clock oscillation frequency (fx)Note 1	Ceramic resonator/ crystal resonator	2.4 V ≤ V_{DD} ≤ 5.5 V	1		12	MHz
XT1 clock oscillation frequency (f_{XT})Note 1	Crystal resonator		32	32.768	35	kHz

Note 1. Indicates only permissible oscillator frequency ranges. Refer to **27.4 AC Characteristics** for instruction execution time. For actual applications, request evaluation by the manufacturer of the oscillator circuit mounted on a board so you can use appropriate values.

Caution Since the CPU is started by the high-speed on-chip oscillator clock after release from the reset state, the user should use the oscillation stabilization time counter status register (OSTC) to check the X1 clock oscillation stabilization time. Specify the values for the oscillation stabilization time in the OSTC register and the oscillation stabilization time select register (OSTS) after having sufficiently evaluated the oscillation stabilization time with the resonator to be used.

Remark When using the X1 or XT1 oscillator, refer to **5.4 System Clock Oscillator**.

27.2.2 On-chip oscillator characteristics

[T_A = −40 to +105°C: G products, T_A = −40 to +125°C: M products, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator clock frequencyNote 1, Note 2	f_{IH}		1		16	MHz
High-speed on-chip oscillator clock frequency accuracy		T_A = +85 to +125°C	−1.5		+1.5	%
		T_A = −20 to +85°C	−1.0		+1.0	%
		T_A = −40 to −20°C	−1.5		+1.5	%
Low-speed on-chip oscillator clock frequency	f_{IL}			15		kHz
Low-speed on-chip oscillator clock frequency accuracy			−15		+15	%

Note 1. The high-speed on-chip oscillator frequency is selected by bits 0 to 2 of option byte (00C2H) and bits 0 to 2 of the HOCODIV register.

Note 2. The listed values only indicate the characteristics of the oscillators. Refer to **27.4 AC Characteristics** for instruction execution time.

27.3 DC Characteristics

27.3.1 Pin characteristics

[T_A = −40 to +105°C: G products, T_A = −40 to +125°C: M products, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

(1/2)

Item	Symbol	Condition			MIN.	TYP.	MAX.	Unit
Output current, high[Note 1]	I_{OH1}	Per pin for P00 to P07, P10 to P17, P20 to P23, P40 to P43, P121, P122, P125					−3.0[Note 2]	mA
		Total of P06, P07, P20 to P23, P40 to P43, P121, P122, P125 (when duty ≤ 70%[Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V				−25.0	mA
			2.4 V ≤ V_{DD} < 4.0 V				−7.0	mA
		Total of P00 to P05, P10 to P17 (when duty ≤ 70%[Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V				−24.0	mA
			2.4 V ≤ V_{DD} < 4.0 V				−6.0	mA
		Total of all pins (when duty ≤ 70%[Note 3])					−40.0	mA
Output current, low[Note 4]	I_{OL1}	Per pin for P00 to P07, P10 to P17, P20 to P23, P40 to P43, P121, P122, P125					8.5[Note 2]	mA
		Per pin for P60, P61					15.0[Note 2]	mA
		Total of P06, P07, P20 to P23, P40 to P43, P121, P122, P125 (when duty ≤ 70%[Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V	T_A = −40 to +105°C			50.0[Note 5]	mA
				T_A = −40 to +125°C			40.0	mA
			2.7 V ≤ V_{DD} < 4.0 V				10.5	mA
			2.4 V ≤ V_{DD} < 2.7 V				4.2	mA
		Total of P00 to P05, P10 to P17, P60, P61 (when duty ≤ 70%[Note 3])	4.0 V ≤ V_{DD} ≤ 5.5 V	T_A = −40 to +105°C			50.0[Note 5]	mA
				T_A = −40 to +125°C			40.0	mA
			2.7 V ≤ V_{DD} < 4.0 V				9.0	mA
			2.4 V ≤ V_{DD} < 2.7 V				3.6	mA
		Total of all pins (when duty ≤ 70%[Note 3])		T_A = −40 to +105°C			80.0	mA
				T_A = −40 to +125°C			60.0	mA

Note 1. Device operation is guaranteed at the listed currents even if current is flowing from the V_{DD} pin to an output pin.

Note 2. The value for maximum total current must not be exceeded.

Note 3. The listed currents apply when the duty cycle is no greater than 70%. Use the following formula to calculate the output current when the duty cycle is greater than 70%, where n is the duty cycle.

- Total output current from the listed pins = (I_{OH} × 0.7)/(n × 0.01)
  Example when n = 80% and I_{OH} = –10.0 mA
  Total output current from the listed pins = (–10.0 × 0.7)/(80 × 0.01) ≅ –8.7 mA
- Total output current from the listed pins = (I_{OL} × 0.7)/(n × 0.01)
  Example when n = 80% and I_{OL} = 10.0 mA
  Total output current from the listed pins = (10.0 × 0.7)/(80 × 0.01) ≅ 8.7 mA

Note that the duty cycle has no effect on the current that is allowed to flow into a single pin. A current higher than the absolute maximum rating must not flow into a single pin.

Note 4. Device operation is guaranteed at the listed currents even if current is flowing from an output pin to the V_{SS} pin.

Note 5. When the multiplexed TSm pin is used as a mutual-capacitance transmit pin, the value is 40 mA.

Caution P00, P01, P03 to P07, P14, P16, P17, P20, P22, and P41 do not output high level in N-ch open-drain mode.

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

[T_A = −40 to +105°C: G products, T_A = −40 to +125°C: M products, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

(2/2)

Item	Symbol	Condition			MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}				0.8 V_{DD}		V_{DD}	V
	V_{IH2}	P60, P61			0.7 V_{DD}		6.0	V
Input voltage, low	V_{IL1}				0		0.2 V_{DD}	V
	V_{IL2}	P60, P61			0		0.3 V_{DD}	V
Output voltage, high Note 1	V_{OH1}	4.0 V ≤ V_{DD} ≤ 5.5 V		I_{OH} = −3.0 mA	V_{DD} − 0.7			V
		2.4 V ≤ V_{DD} ≤ 5.5 V		I_{OH} = −1.0 mA	V_{DD} − 0.5			V
Output voltage, low Note 2	V_{OL1}	4.0 V ≤ V_{DD} ≤ 5.5 V		I_{OL} = 8.5 mA			0.7	V
		2.7 V ≤ V_{DD} ≤ 5.5 V		I_{OL} = 1.5 mA			0.5	V
		2.4 V ≤ V_{DD} ≤ 5.5 V		I_{OL} = 0.6 mA			0.4	V
		P60, P61	4.0 V ≤ V_{DD} ≤ 5.5 V	I_{OL} = 15 mA			2.0	V
				I_{OL} = 5 mA			0.4	V
			2.7 V ≤ V_{DD} ≤ 5.5 V	I_{OL} = 3.0 mA			0.4	V
				I_{OL} = 2.0 mA			0.4	V
			2.4 V ≤ V_{DD} ≤ 5.5 V	I_{OL} = 1.0 mA			0.4	V
Input leakage current, high	I_{LIH1}	P00 to P07, P10 to P17, P20 to P23, P40 to P43, P60, P61, P125, P137 V_I=V_{DD}					1	μA
	I_{LIH2}	P121, P122 (X1, X2, XT1, XT2, EXCLK, EXCLKS) V_I = V_{DD}		In input port or external clock input			1	μA
				In resonator connection			10	μA
Input leakage current, low	I_{LIL1}	P00 to P07, P10 to P17, P20 to P23, P40 to P43, P60, P61, P125, P137 V_I = V_{SS}					−1	μA
	I_{LIL2}	P121, P122 (X1, X2, XT1, XT2, EXCLK, EXCLKS) V_I = V_{SS}		In input port or external clock input			−1	μA
				In resonator connection			−10	μA
On-chip pull-up resistance	R_U	Except P60 and P61 V_I = V_{SS}			10	20	100	kΩ

Note 1. The value under the condition which satisfies the high-level output current (I_{OH1}).

Note 2. The value under the condition which satisfies the low-level output current (I_{OL1}).

Caution The maximum value of V_{IH} of P00, P01, P03 to P07, P14, P16, P17, P20, P22, and P41 is V_{DD} even in N-ch open-drain mode. These pins do not output high level in N-ch open-drain mode.

Remark The characteristics of functions multiplexed on a given pin are the same as those for the port pin unless otherwise specified.

27.3.2 Supply current characteristics

[T_A = −40 to +105°C: G products, T_A = −40 to +125°C: M products, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Symbol	Condition				MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD1}	Operating mode	Basic operation	f_{IH} = 16 MHz[Note 4]	V_{DD} = 3.0 V, 5.0 V		0.97		mA
			Normal operation	f_{IH} = 16 MHz[Note 4]	V_{DD} = 3.0 V, 5.0 V		2.11	2.78	mA
				f_{IH} = 4 MHz[Note 4]	V_{DD} = 3.0 V, 5.0 V		1.22	1.65	mA
				f_{EX} = 16 MHz[Note 5, Note 6] V_{DD} = 3.0 V, 5.0 V	Square wave input		1.97	2.64	mA
				f_X = 12 MHz[Note 5, Note 6] V_{DD} = 3.0 V, 5.0 V	Resonator connection		1.79	2.51	mA
				f_{MX} = 4 MHz[Note 5, Note 6] V_{DD} = 3.0 V, 5.0 V	Square wave input		1.07	1.5	mA
					Resonator connection		1.12	1.56	mA
				f_{SUB} = 32.768 kHz[Note 7] T_A = −40°C	Square wave input		3.65	5.80	µA
					Resonator connection		3.70	6.00	µA
				f_{SUB} = 32.768 kHz[Note 7] T_A = +25°C	Square wave input		3.90	5.80	µA
					Resonator connection		4.18	6.00	µA
				f_{SUB} = 32.768 kHz[Note 7] T_A = +50°C	Square wave input		4.04	6.20	µA
					Resonator connection		4.37	6.40	µA
				f_{SUB} = 32.768 kHz[Note 7] T_A = +70°C	Square wave input		4.20	6.50	µA
					Resonator connection		4.56	6.70	µA
				f_{SUB} = 32.768 kHz[Note 7] T_A = +85°C	Square wave input		4.40	7.80	µA
					Resonator connection		4.80	8.00	µA
				f_{SUB} = 32.768 kHz[Note 7] T_A = +105°C	Square wave input		4.92	9.12	µA
					Resonator connection		5.36	9.32	µA
				f_{SUB} = 32.768 kHz[Note 7] T_A = +125°C	Square wave input		6.14	15.37	µA
					Resonator connection		6.60	15.57	µA

Item	Symbol	Condition			MIN.	TYP.	MAX.	Unit
Supply current[Note 1]	I_{DD2}[Note 2]	HALT mode	f_{IH} = 16 MHz[Note 4]	V_{DD} = 3.0 V, 5.0 V		385	824	μA
			f_{IH} = 4 MHz[Note 4]	V_{DD} = 3.0 V, 5.0 V		334	637	μA
			f_{EX} = 16 MHz[Note 5, Note 6] V_{DD} = 3.0 V, 5.0 V	Square wave input		229	669	μA
			f_X = 12 MHz[Note 5, Note 6] V_{DD} = 3.0 V, 5.0 V	Resonator connection		351	960	μA
			f_{MX} = 4 MHz[Note 5, Note 6] V_{DD} = 3.0 V, 5.0 V	Square wave input		167	459	μA
				Resonator connection		226	620	μA
			f_{SUB} = 32.768 kHz[Note 8] T_A = −40°C	Square wave input		0.69	1.45	μA
				Resonator connection		0.75	1.65	μA
			f_{SUB} = 32.768 kHz[Note 8] T_A = +25°C	Square wave input		0.75	1.45	μA
				Resonator connection		1.04	1.65	μA
			f_{SUB} = 32.768 kHz[Note 8] T_A = +50°C	Square wave input		0.84	1.74	μA
				Resonator connection		1.20	1.94	μA
			f_{SUB} = 32.768 kHz[Note 8] T_A = +70°C	Square wave input		0.97	2.20	μA
				Resonator connection		1.33	2.40	μA
			f_{SUB} = 32.768 kHz[Note 8] T_A = +85°C	Square wave input		1.13	3.10	μA
				Resonator connection		1.51	3.30	μA
			f_{SUB} = 32.768 kHz[Note 8] T_A = +105°C	Square wave input		1.58	8.92	μA
				Resonator connection		1.99	9.12	μA
			f_{SUB} = 32.768 kHz[Note 8] T_A = +125°C	Square wave input		2.68	10.67	μA
				Resonator connection		3.12	10.87	μA
	I_{DD3}[Note 3]	STOP mode[Note 9]	V_{DD} = 3.0 V	T_A = +105°C		0.62	4.12	μA
				T_A = +125°C		0.62	10.37	μA

Note 1. The listed currents are the total currents flowing into V_{DD}, including the input leakage currents flowing when the level of the input pin is fixed to V_{DD} or V_{SS}.
Regarding the values for main system clock operation, the TYP. value does not include the peripheral operating current. The MAX. value includes the peripheral operating current, but does not include those flowing into the A/D converter, comparator, capacitive touch sensing unit (CTSU), I/O port, and on-chip pull-up/pull-down resistors.
Regarding the values for subsystem clock operation, the TYP. and MAX. values do not include the peripheral operating current. However, in HALT mode, the current flowing into the RTC is included.
Regarding the values in STOP mode, the TYP. and MAX. values do not include the peripheral operating current.

Note 2. When the HALT instruction is executed from the flash memory.

Note 3. The listed currents do not include the current flowing into real-time clock 2, the clock output/buzzer output controller, 12-bit interval timer, and watchdog timer.

Note 4. When the high-speed system clock and subsystem clock are stopped.

Note 5. When the high-speed on-chip oscillator and subsystem clock are stopped.

Note 6. 16-pin, 20-pin, 24-pin, and 32-pin products only.

Note 7. When the high-speed on-chip oscillator and high-speed system clock are stopped. When AMPHS1 = 1 (ultra-low power consumption oscillation). However, not including the current flowing into real-time clock 2, the 12-bit interval timer, watchdog timer, and capacitive touch sensing unit.

Note 8. When the high-speed on-chip oscillator and high-speed system clock are stopped. When RTCLPC = 1 and AMPHS1 = 1 (ultra-low power consumption oscillation).

Note 9. Regarding the value for current to operate the subsystem clock in STOP mode, refer to that in HALT mode.

Remark 1. f_{IH}: High-speed on-chip oscillator clock frequency

Remark 2. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

Remark 3. Except for subsystem clock operation, the temperature condition of the TYP. value is T_A = 25°C.

Remark 4. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

Peripheral Functions

[T_A = −40 to +105°C: G products, T_A = −40 to +125°C: M products, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Low-speed on-chip oscillator operating current	I_{FIL} Note 1				0.30		μA
RTC2 operating current	I_{RTC} Note 1, Note 2, Note 8	f_{SUB} = 32.768 kHz			0.02		μA
12-bit interval timer operating current	I_{TMKA} Note 1, Note 2, Note 3				0.02		μA
Watchdog timer operating current	I_{WDT} Note 1, Note 4				0.02		μA
A/D converter operating current	I_{ADC} Note 1, Note 5	In conversion at maximum speed	V_{DD} = 5.0 V		1.30	1.90	mA
			V_{DD} = 3.0 V		0.50		mA
Temperature sensor operating current	I_{TMPS} Note 1				75.0		μA
Comparator operating current	I_{CMP} Note 1, Note 6	In high-speed mode	V_{DD} = 5.0 V		6.50		μA
		In low-speed mode	V_{DD} = 5.0 V		1.70		μA
Internal reference voltage operating current	I_{VREG} Note 1				10		μA
Self-programming operating current	I_{FSP} Note 1, Note 7				2.0	12.20	mA

Note 1. The current flowing into V_{DD}.

Note 2. When the high-speed on-chip oscillator and high-speed system clock are stopped.

Note 3. This current only flows into the 12-bit interval timer. It does not include the operating current of the low-speed on-chip oscillator and XT1 oscillator. The supply current of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{FIL} and I_{TMKA} when the 12-bit interval timer is in operation.

Note 4. This current only flows into the watchdog timer. It does not include 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_{FIL} and I_{WDT} when the watchdog timer is in operation.

Note 5. This current only flows into 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 is operating or in the HALT mode.

Note 6. This current only flows into a single comparator. The supply current of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{CMP} when the comparator is in operation.

Note 7. This current only flows during self-programming.

Note 8. This current only flows into real-time clock 2. It does not include the operating current of the low-speed on-chip oscillator and XT1 oscillator. The supply current of the RL78 microcontrollers is the sum of the values of either I_{DD1} or I_{DD2}, and I_{RTC}, when real-time clock 2 operates in operation mode or HALT mode. When the low-speed on-chip oscillator is selected, I_{FIL} should be added. I_{DD2} subsystem clock operation includes the operational current of real-time clock 2.

Remark The temperature condition of the TYP. value is T_A = 25°C.

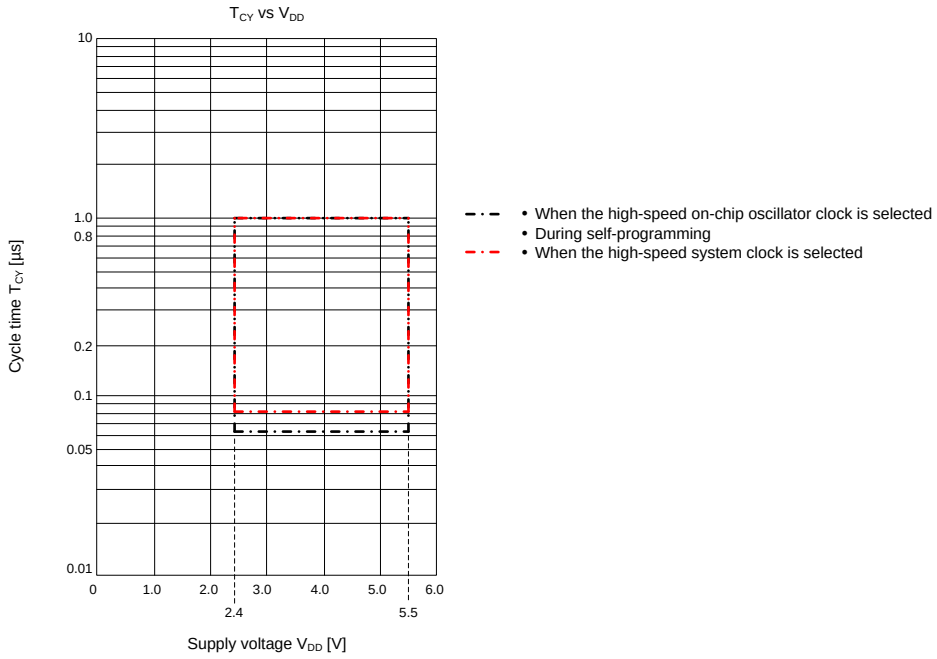
27.4 AC Characteristics

[T_A = −40 to +105°C: G products, T_A = −40 to +125°C: M products, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

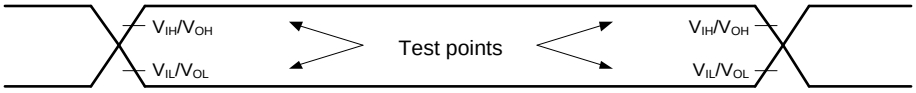
Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	When high-speed on-chip oscillator clock (f_{IH}) is selected	2.4 V ≤ V_{DD} ≤ 5.5 V	0.0625		1.0	µs
		When high-speed system clock (f_{MX}) is selected	2.4 V ≤ V_{DD} ≤ 5.5 V	0.0833		1.0	µs
		Subsystem clock (f_{SUB}) operation	2.4 V ≤ V_{DD} ≤ 5.5 V	28.5	30.5	31.3	µs
		In the self-programming mode	2.4 V ≤ V_{DD} ≤ 5.5 V	0.0625		1.0	µs
External system clock frequency	f_{EX}	2.4 V ≤ V_{DD} ≤ 5.5 V		1.0		16	MHz
External system clock input high-level width, low-level width	t_{EXH}, t_{EXL}	2.4 V ≤ V_{DD} ≤ 5.5 V		30			ns
TI00 to TI07 input high-level width, low-level width	t_{TIH}, t_{TIL}	Noise filter is not used		$1/f_{MCK}$ + 10			ns
TO00 to TO07 output frequency	f_{TO}	4.0 V ≤ V_{DD} ≤ 5.5 V				8	MHz
		2.7 V ≤ V_{DD} < 4.0 V				5	MHz
		2.4 V ≤ V_{DD} < 2.7 V				4	MHz
PCLBUZ0 output frequency	f_{PCL}	4.0 V ≤ V_{DD} ≤ 5.5 V				10	MHz
		2.7 V ≤ V_{DD} < 4.0 V				5	MHz
		2.4 V ≤ V_{DD} < 2.7 V				4	MHz
$\overline{\text{RESET}}$ low-level width	t_{RSL}			10			µs

Remark f_{MCK}: Timer array unit operating clock frequency
(Operation clock to be set by timer clock select register 0 (TPS0) and the CKS0n1 bit of timer mode register 0n (TMR0n). n: Channel number (n = 0 to 7).)

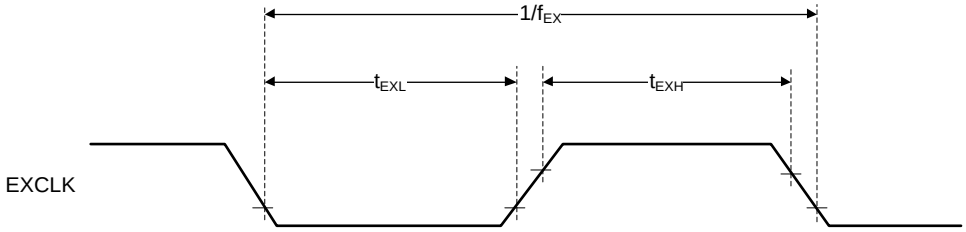
Minimum Instruction Execution Time during Main System Clock Operation



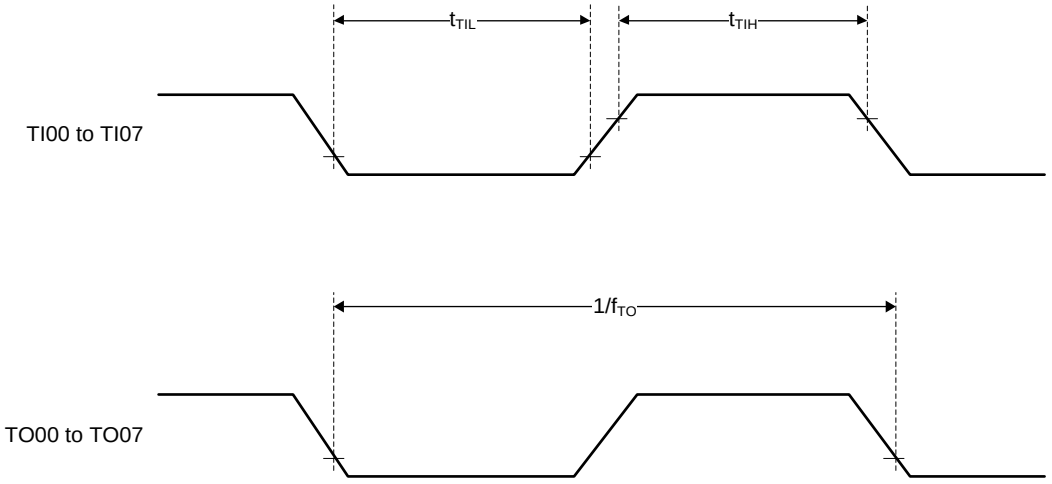
At AC Timing



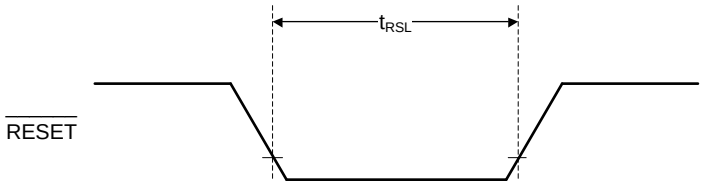
External System Clock Timing



TI/TO Timing

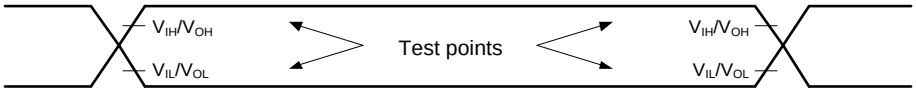


$\overline{\text{RESET}}$ Input Timing



27.5 Serial Interface Characteristics

AC Timing Test Points



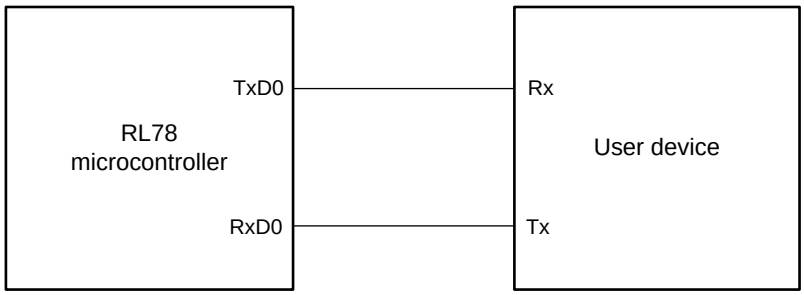
27.5.1 Serial array unit

(1) UART mode

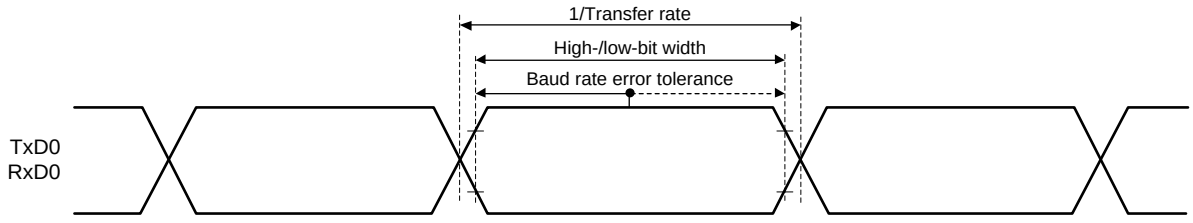
[T_A = −40 to +105°C: G products, T_A = −40 to +125°C: M products, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Transfer rate					f_{MCK}/12	bps
		Theoretical value of the maximum transfer rate f_{CLK} = f_{MCK} = 16 MHz			1.3	Mbps

UART mode connection diagram



UART mode bit width (reference)



Remark f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by serial clock select register m (SPSm) and the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 11, 20))

(2) Simplified SPI (CSI) mode (master mode, SCKp... internal clock output)

[T_A = −40 to +105°C: G products, T_A = −40 to +125°C: M products, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
SCKp cycle time	t_{KCY1}	$t_{KCY1} \geq 4/f_{CLK}$	2.7 V ≤ V_{DD} ≤ 5.5 V	250			ns
			2.4 V ≤ V_{DD} ≤ 5.5 V	500			ns
SCKp high-/low-level width	t_{KH1}, t_{KL1}	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		113			ns
SIp hold time (from SCKp ↑)Note 1	t_{KSI1}			38			ns
Delay time from SCKp ↓ to SOp outputNote 2	t_{KSO1}	C = 30 pFNote 3, Note 4				66	ns

Note 1. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The SIp setup time becomes “to SCKp ↓” and the SIp hold time becomes “from SCKp ↓” when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

Note 2. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The delay time to SOp output becomes “from SCKp ↓” when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

Note 3. C is the load capacitance of the SCKp and SOp output lines.

Note 4. External loads on P60 and P61 in the case of the SOp output line: R = 1 kΩ, C = 30 pF

(3) Simplified SPI (CSI) mode (slave mode, SCKp... external clock input)

[T_A = −40 to +105°C: G products, T_A = −40 to +125°C: M products, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
SCKp cycle time	t_{KCY2}	2.7 V ≤ V_{DD} ≤ 5.5 V		$12/f_{MCK}$			ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		$12/f_{MCK}$ and also 1000			ns
SCKp high-/low-level width	t_{KH2},t_{KL2}	2.7 V ≤ V_{DD} ≤ 5.5 V		$t_{KCY2}/2$ − 16			ns
		2.4 V ≤ V_{DD} ≤ 5.5 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} ≤ 5.5 V		$1/f_{MCK}$ + 60			ns
SIp hold time (from SCKp ↑)Note 1	t_{KSI2}	2.4 V ≤ V_{DD} ≤ 5.5 V		$1/f_{MCK}$ + 62			ns
Delay time from SCKp ↓ to SOp outputNote 2	t_{KSO2}	C = 30 pF Note 3, Note 4	2.7 V ≤ V_{DD} ≤ 5.5 V			$2/f_{MCK}$ + 66	ns
			2.4V ≤ V_{DD} ≤ 5.5V			$2/f_{MCK}$ + 113	ns

Note 1. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The SIp setup time becomes “to SCKp ↓” and the SIp hold time becomes “from SCKp ↓” when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

Note 2. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1. The delay time to SOp output becomes “from SCKp ↓” when DAP0n = 0 and CKP0n = 1, or DAP0n = 1 and CKP0n = 0.

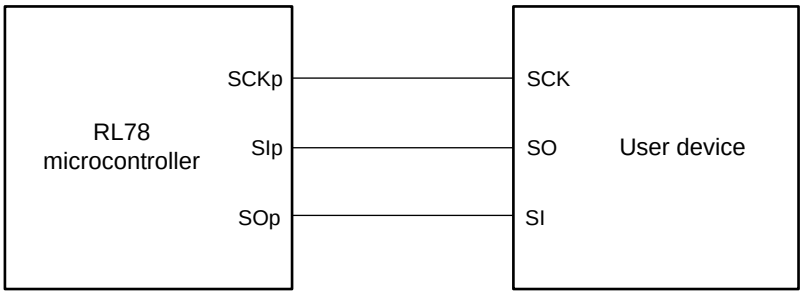
Note 3. C is the load capacitance of the SOp output lines.

Note 4. External loads on P60 and P61 in the case of the SOp output line: R = 1 kΩ, C = 30 pF

Remark 1. p: CSI number (p = 00, 11, 20), n: Channel number (n = 0, 1)

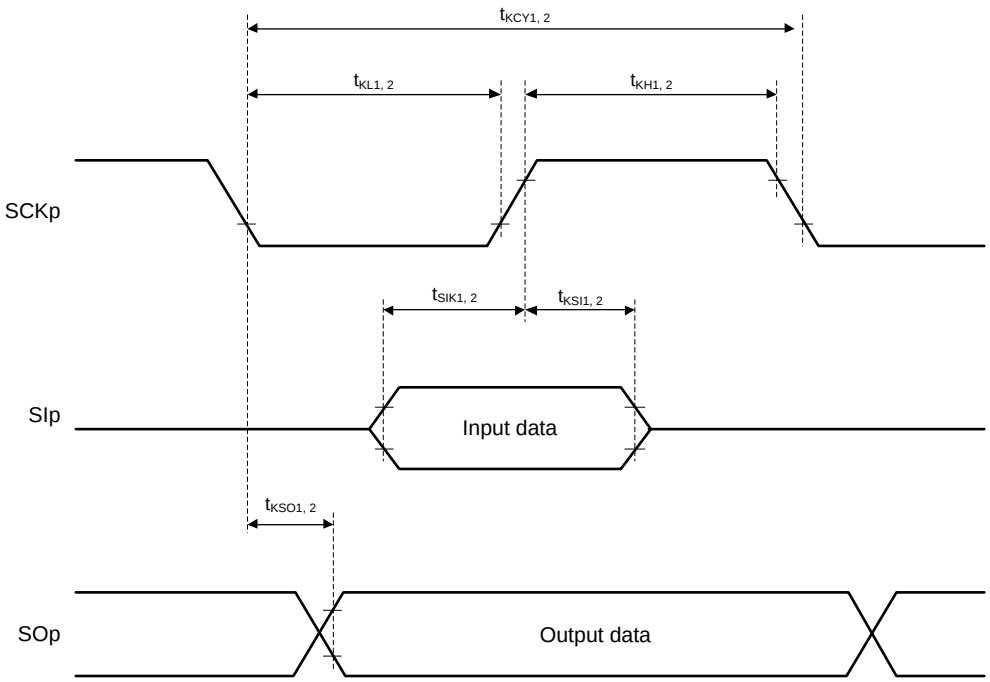
Remark 2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by serial clock select register 0 (SPS0) and the CKS0n bit of serial mode register 0n (SMR0n). n: Channel number (n = 0, 1))

Simplified SPI (CSI) mode connection diagram



Simplified SPI (CSI) mode serial transfer timing

(When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1)



Remark p: CSI number (p = 00, 11, 20), n: Channel number (n = 0, 1)

(4) Simplified I²C mode

[T_A = −40 to +105°C: G products, T_A = −40 to +125°C: M products, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

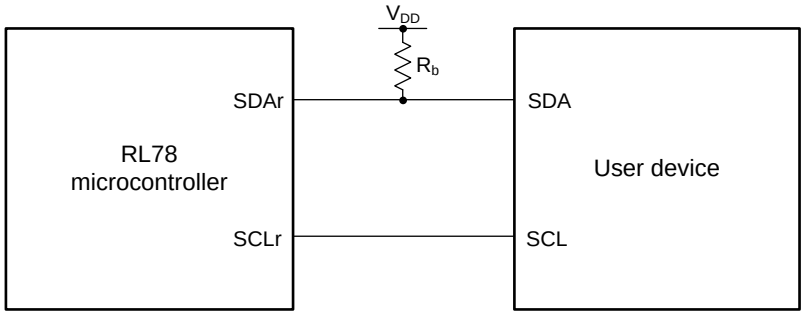
Item	Symbol	Condition	MIN.	MAX.	Unit
SCLr clock frequency	f_{SCL}	2.7 V ≤ V_{DD} ≤ 5.5 V C_b = 100 pF, R_b = 3 kΩ		400[Note 1]	kHz
		2.4 V ≤ V_{DD} ≤ 5.5 V C_b = 100 pF, R_b = 3 kΩ		100[Note 1]	kHz
Hold time when SCLr = “L”	t_{LOW}	2.7 V ≤ V_{DD} ≤ 5.5 V C_b = 100 pF, R_b = 3 kΩ	1200		ns
		2.4 V ≤ V_{DD} ≤ 5.5 V C_b = 100 pF, R_b = 3 kΩ	4600		ns
Hold time when SCLr = “H”	t_{HIGH}	2.7 V ≤ V_{DD} ≤ 5.5 V C_b = 100 pF, R_b = 3 kΩ	1200		ns
		2.4 V ≤ V_{DD} ≤ 5.5 V C_b = 100 pF, R_b = 3 kΩ	4600		ns
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ V_{DD} ≤ 5.5 V C_b = 100 pF, R_b = 3 kΩ	1/f_{MCK} + 220[Note 2]		ns
		2.4 V ≤ V_{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 ≤ V_{DD} ≤ 5.5 V C_b = 100 pF, R_b = 3 kΩ	0	770	ns
		2.4 V ≤ V_{DD} ≤ 5.5 V C_b = 100 pF, R_b = 3 kΩ	0	1420	ns

Note 1. The value must also be no greater than f_{MCK}/4.

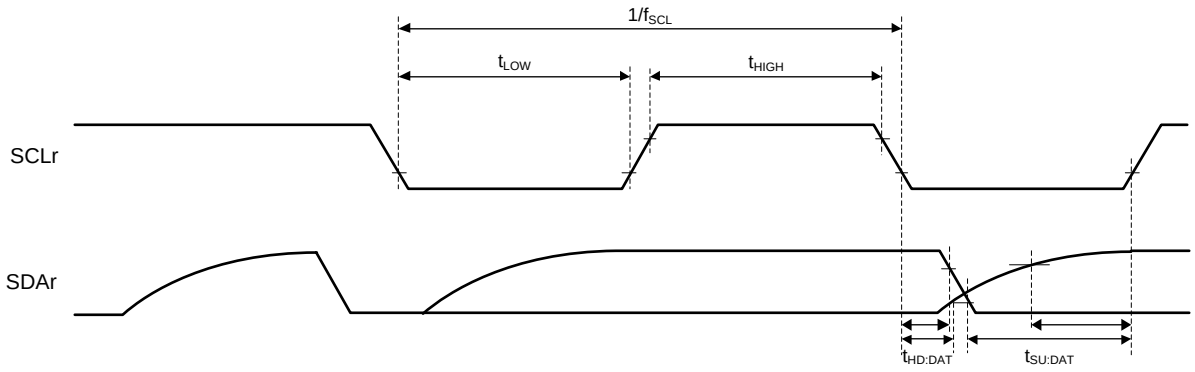
Note 2. Set f_{MCK} so that it will not exceed the hold time when SCLr = “L” or SCLr = “H”.

Caution Select the N-ch open drain output (V_{DD} tolerance) mode for the SDAr pin by using port output mode register 0, 1, 2, 4, or 6 (POM0, 1, 2, 4, or 6).

Simplified I²C mode connection diagram



Simplified I^2C mode serial transfer timing



Remark 1. R_b[Ω]: Communication line (SDAr) pull-up resistance, C_b[F]: Communication line (SCLr, SDAr) load capacitance

Remark 2. r: IIC number (r = 00, 11, 20)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by serial clock select register 0 (SPS0) and the CKS0n bit of serial mode register 0n (SMR0n). n: Channel number (n = 0, 1))

27.5.2 Serial interface IICA

[T_A = −40 to +105°C: G products, T_A = −40 to +125°C: M products, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Symbol	Condition	Standard Mode		Fast Mode		Unit
			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 or restart condition is detected.

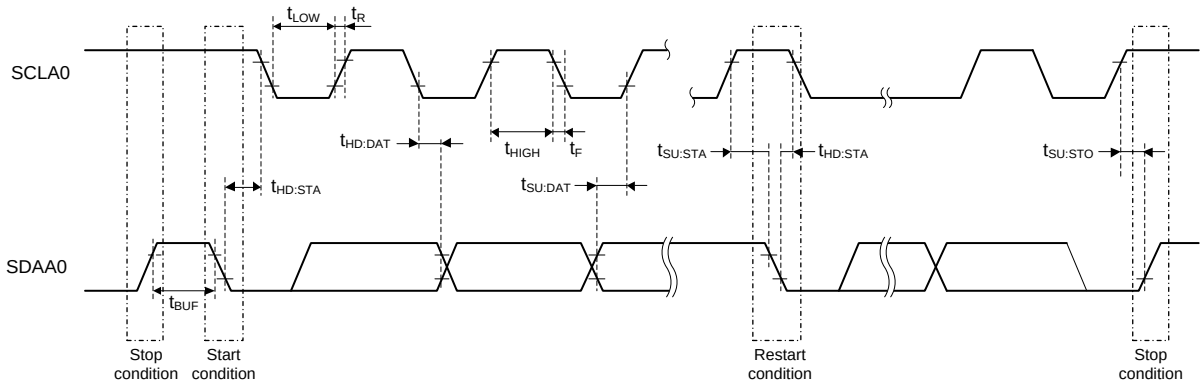
Note 2. The maximum value (MAX.) of $t_{HD:DAT}$ applies to normal transfer and a wait is inserted at the ACK (acknowledge) timing.

Remark The maximum value of C_b (communication line capacitance) and the value of R_b (communication line pull-up resistance) at that time in each mode are as follows.

Standard mode: C_b = 400 pF, R_b = 2.7 kΩ

Fast mode: C_b = 200 pF, R_b = 1.7 kΩ

IICA serial transfer timing



27.6 Analog Characteristics

27.6.1 A/D converter characteristics

Targets: ANI0 to ANI10, internal reference voltage, temperature sensor output voltage, and CTSU TSCAP voltage

[T_A = −40 to +105°C: G products, T_A = −40 to +125°C: M products, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error Note 1, Note 2, Note 3	AINL	10-bit resolution	V_{DD} = 5 V		±1.7	±3.1	LSB
			V_{DD} = 3 V		±2.3	±4.5	LSB
Conversion time	t_{CONV}	10-bit resolution Targets: ANI0 to ANI10	2.7 V ≤ V_{DD} ≤ 5.5 V	4.25		17	µs
			2.4 V ≤ V_{DD} ≤ 5.5 V Note 5	5.75		23	µs
		10-bit resolution Target: Internal reference voltage Note 6 Target: Temperature sensor output voltage Note 6 Target: CTSU TSCAP voltage Note 6	2.4 V ≤ V_{DD} ≤ 5.5 V	5.75		23	µs
Zero-scale error Note 1, Note 2, Note 3, Note 4	E_{ZS}	10-bit resolution	V_{DD} = 5 V			±0.19	%FSR
			V_{DD} = 3 V			±0.39	%FSR
Full-scale error Note 1, Note 2, Note 3, Note 4	E_{FS}	10-bit resolution	V_{DD} = 5 V			±0.29	%FSR
			V_{DD} = 3 V			±0.42	%FSR
Integral linearity error Note 1, Note 2, Note 3	ILE	10-bit resolution	V_{DD} = 5 V			±1.8	LSB
			V_{DD} = 3 V			±1.7	LSB
Differential linearity error Note 1, Note 2, Note 3	DLE	10-bit resolution	V_{DD} = 5 V			±1.4	LSB
			V_{DD} = 3 V			±1.5	LSB
Analog input voltage	V_{AIN}	Targets: ANI0 to ANI10		0		V_{DD}	V
		Target: Internal reference voltage Note 6		V_{REG} Note 7			V
		Target: Temperature sensor output voltage Note 6		V_{TMPS25} Note 7			V
		Target: CTSU TSCAP voltage (2.4 V ≤ V_{DD} ≤ 5.5 V)		V_{TSCAP}			V

Note 1. The TYP. value is an average value at T_A = 25°C. The MAX. value is an average value ±3σ at normal distribution.

Note 2. These values are the results of characteristic evaluation and are not checked for shipment.

Note 3. A quantization error (±1/2 LSB) is not included.

Note 4. Expressed as a ratio (%FSR) relative to the full-scale value.

Note 5. Be sure to set the LV0 bit in A/D converter mode register 0 (ADM0) to 0 when conversion is done in the operating voltage range of 2.4 V ≤ V_{DD} < 2.7 V.

Note 6. Be sure to set the LV0 bit in A/D converter mode register 0 (ADM0) to 0 when the internal reference voltage, temperature sensor output voltage, or CTSU TSCAP voltage is selected as the target for conversion.

Note 7. Refer to **27.6.3 Temperature sensor/internal reference** voltage characteristics.

Caution 1. **Arrange wiring and insert the capacitor so that no noise appears on the power supply/ground line.**

Caution 2. **Do not allow any pulses that rapidly change such as digital signals to be input/output to/from the pins adjacent to the conversion pin during A/D conversion.**

Caution 3. **Note that the internal reference voltage cannot be used as the reference voltage of the comparator when the internal reference voltage is selected as the target for A/D conversion.**

27.6.2 Comparator characteristics

[T_A = −40 to +105°C: G products, T_A = −40 to +125°C: M products, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Input voltage range	I_{VREF}	IVREFn pin input (CnVRF bit = 0)		0		V_{DD} − 1.4	V
		Internal reference voltage (CnVRF bit = 1) Note 1		V_{REG} Note 2			V
	I_{VCMP}	IVCMPn pin input		−0.3		V_{DD} + 0.3	V
Output delay	t_d	V_{DD} = 3.0 V, input slew rate > 50 mV/μs	High-speed mode			0.5	μs
			Low-speed mode		2.0		μs
Operation stabilization wait time	t_{CMP}			100			μs

Note 1. When the internal reference voltage is selected as the reference voltage of the comparator, the internal reference voltage cannot be used as the target for A/D conversion.

Note 2. Refer to **27.6.3 Temperature sensor/internal reference voltage characteristics.**

Remark n: Channel number (n = 0, 1)

27.6.3 Temperature sensor/internal reference voltage characteristics

[T_A = −40 to +105°C: G products, T_A = −40 to +125°C: M products, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Temperature sensor output voltage	V_{TMPS25}			1.05		V
Internal reference voltage	V_{REG}		0.74	0.815	0.89	V
Temperature coefficient	F_{VTMPS}	Temperature dependence of the temperature sensor voltage		−3.6		mV/°C
Operation stabilization wait time	t_{AMP}	A/D converter is used (ADS register = 0DH)	5			μs

Caution **The internal reference voltage cannot be simultaneously used by the A/D converter and the comparator; only one of them must be selected.**

27.6.4 SPOR circuit characteristics

[T_A = −40 to +105°C: G products, T_A = −40 to +125°C: M products, V_{SS} = 0 V]

Item		Symbol	Condition	MIN.	TYP.	MAX.	Unit
Detection voltage	Power supply voltage level	V_{SPOR0}	Power supply rising	4.08	4.28	4.45	V
		V_{SPDR0}	Power supply falling	4.00	4.20	4.37	V
		V_{SPOR1}	Power supply rising	2.76	2.90	3.02	V
		V_{SPDR1}	Power supply falling	2.70	2.84	2.96	V
		V_{SPOR2}	Power supply rising	2.44	2.57	2.68	V
		V_{SPDR2}	Power supply falling	2.40	2.52	2.62	V
		V_{SPOR3}	Power supply rising		2.16		V
		V_{SPDR3}	Power supply falling		2.11		V
Minimum pulse width[Note 1]		T_{SPW}		300			µs

Note 1. Time required for the reset operation by the SPOR circuit when V_{DD} falls below V_{SPDR}.

Caution Make sure to keep the internal reset state by the SPOR or an external reset until the power supply voltage (V_{DD}) reaches the operating voltage range shown in 27.4 AC Characteristics.

27.6.5 Power supply voltage rising slope characteristics

[T_A = −40 to +105°C: G products, T_A = −40 to +125°C: M products, V_{SS} = 0 V]

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Power supply voltage rising slope	S_{VDD}				54	V/ms

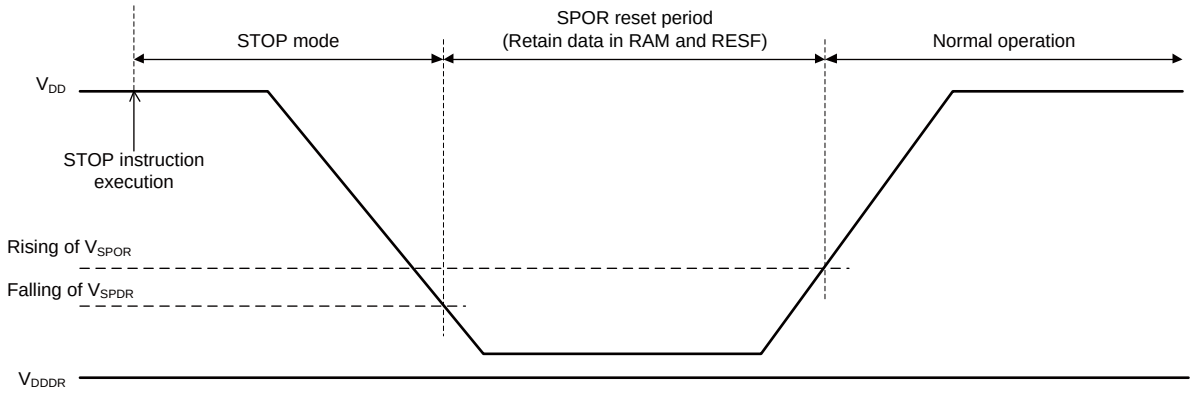
Caution Make sure to keep the internal reset state by the SPOR or an external reset until the power supply voltage (V_{DD}) reaches the operating voltage range shown in 27.4 AC Characteristics.

27.7 RAM Data Retention Characteristics

[T_A = −40 to +105°C: G products, T_A = −40 to +125°C: M products, V_{SS} = 0 V]

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Data retention power supply voltage	V_{DDDR}		1.9		5.5	V

Caution Data in RAM is retained until the power supply voltage falls below the MIN. value of the data retention power supply voltage (V_{DDDR}). Note that data in the RESF register might not be cleared even if the power supply voltage falls below the MIN. value of the data retention power supply voltage (V_{DDDR}).



27.8 Flash Memory Programming Characteristics

[T_A = −40 to +105°C: G products, T_A = −40 to +125°C: M products, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Symbol	Condition		MIN.	TYP.	MAX.	Unit
Number of code flash rewrites Note 1, Note 2	C_{erwr}	Retained for 20 years	T_A = +85°C Note 3	1000			Times
Number of data flash rewrites Note 1, Note 2		Retained for 1 year	T_A = +25°C		1,000,000		Times
		Retained for 5 years	T_A = +85°C Note 3	100,000			Times
		Retained for 20 years	T_A = +85°C Note 3	10,000			Times

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. These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas Electronics.

Note 3. This temperature is the average value at which data are retained.

Code flash/data flash self-programming time

[T_A = −40 to +105°C: G products, T_A = −40 to +125°C: M products, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Symbol	f_{CLK} = 1 MHz			f_{CLK} = 16 MHz			Unit
		MIN.	TYP.	MAX.	MIN.	TYP.	MAX.	
Writing (4 bytes)	t_{P4}		104	905		53.8	504.9	µs
Block erasure (1 KB)	t_{E1K}		7.9	262.3		5.5	214.1	ms

Caution The listed values do not include the time until the operations of the flash memory start following execution of an instruction by software.

27.9 Dedicated Flash Memory Programmer Communication (UART)

[T_A = −40 to +105°C: G products, T_A = −40 to +125°C: M products, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

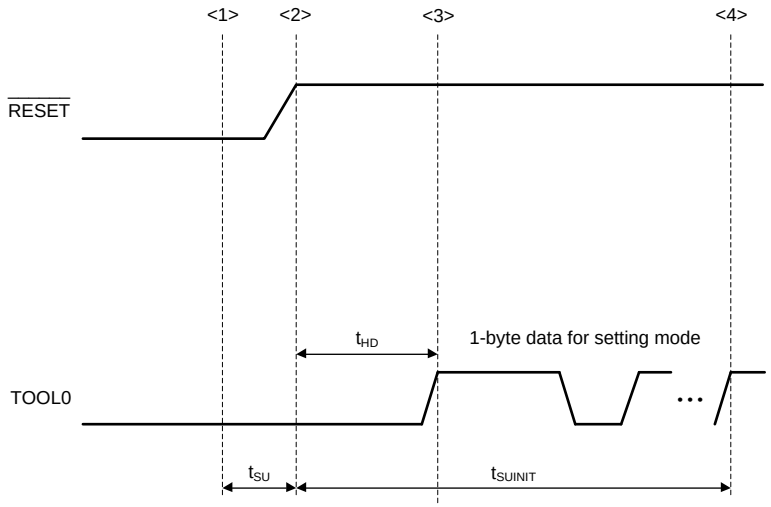
Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Transfer rate				115,200		bps

Remark The transfer rate during flash memory programming is fixed to 115,200 bps.

27.10 Timing of Entry to Flash Memory Programming Mode

[T_A = −40 to +105°C: G products, T_A = −40 to +125°C: M products, 2.4 V ≤ V_{DD} ≤ 5.5 V, V_{SS} = 0 V]

Item	Symbol	Condition	MIN.	TYP.	MAX.	Unit
Time to complete the communication for the initial setting after the external reset is released	t_{SUINIT}	The SPOR reset must be released before the external reset is released.			100	ms
Time to release the external reset after the TOOL0 pin is set to the low level	t_{SU}	The SPOR reset must be released before the external reset is released.	10			μs
Time to hold the TOOL0 pin at the low level after the external reset is released	t_{HD}	The SPOR reset must be released before the external reset is released.	1			ms



<1> The low level is input to the TOOL0 pin.

<2> The external reset is released (the SPOR reset must have been released before that).

<3> The TOOL0 pin is released from the low level.

<4> Setting of entry to the flash memory programming mode by UART reception is completed.

Remark t_{SUINIT}: During this period, the communications for the initial setting must be completed within 100 ms after release from the reset.

t_{SU}: Time to release the external reset after the TOOL0 pin is set to the low level

t_{HD}: Time to hold the TOOL0 pin at the low level after the external reset is released

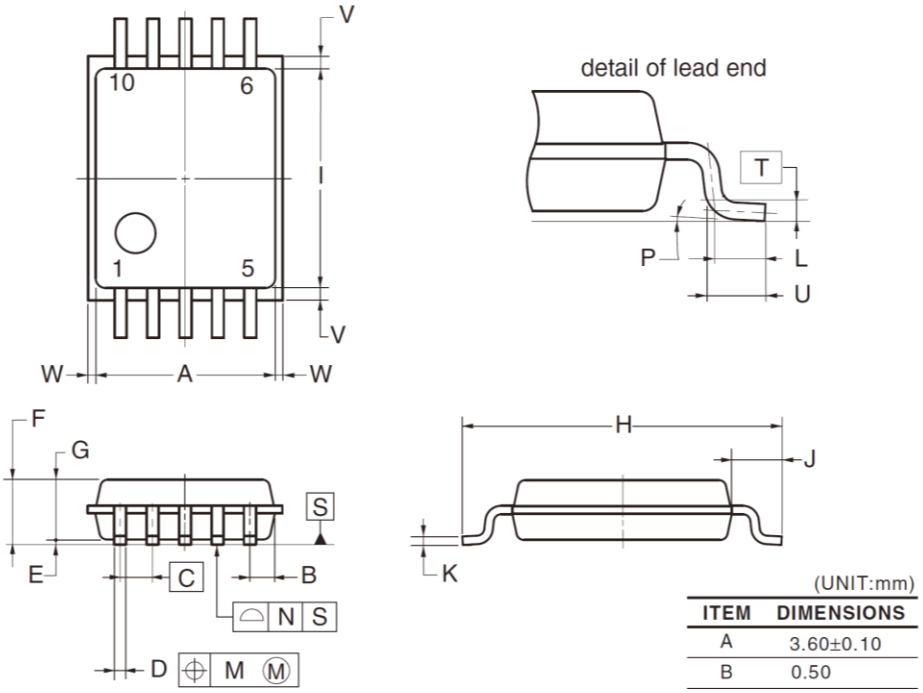
CHAPTER 28 PACKAGE DRAWINGS

28.1 10-pin products

R5F1211CMSP, R5F1211CGSP, R5F1211CASP

R5F1211AMSP, R5F1211AGSP, R5F1211AASP

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	$3^{\circ}{}^{+5^{\circ}}_{-3^{\circ}}$
T	0.25 (T.P.)
U	0.60±0.15
V	0.25 MAX.
W	0.15 MAX.

NOTE

Each lead centerline is located within 0.13 mm of its true position (T.P.) at maximum material condition.

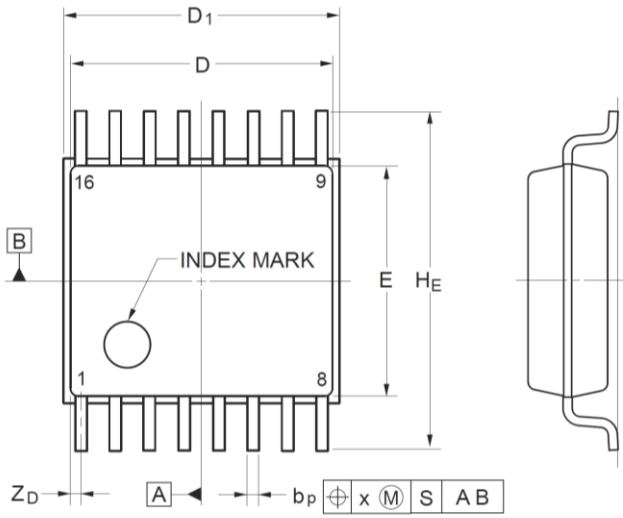
28.2 16-pin products

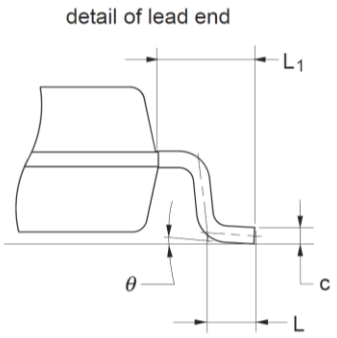
R5F1214CMSP, R5F1214CGSP, R5F1214CASP

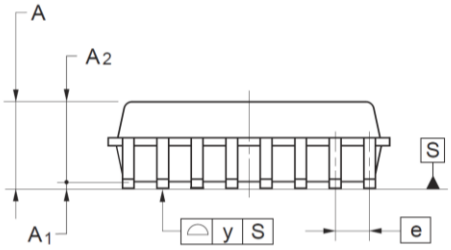
R5F1214AMSP, R5F1214AGSP, R5F1214AASP

JEITA Package Code	RENESAS Code	Previous Code	MASS (Typ) [g]
P-SSOP16-4.4x5-0.65	PRSP0016JC-B	P16MA-65-FAB-1	0.08

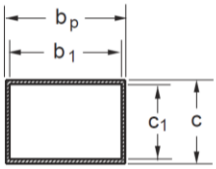
Unit: mm







Terminal cross section

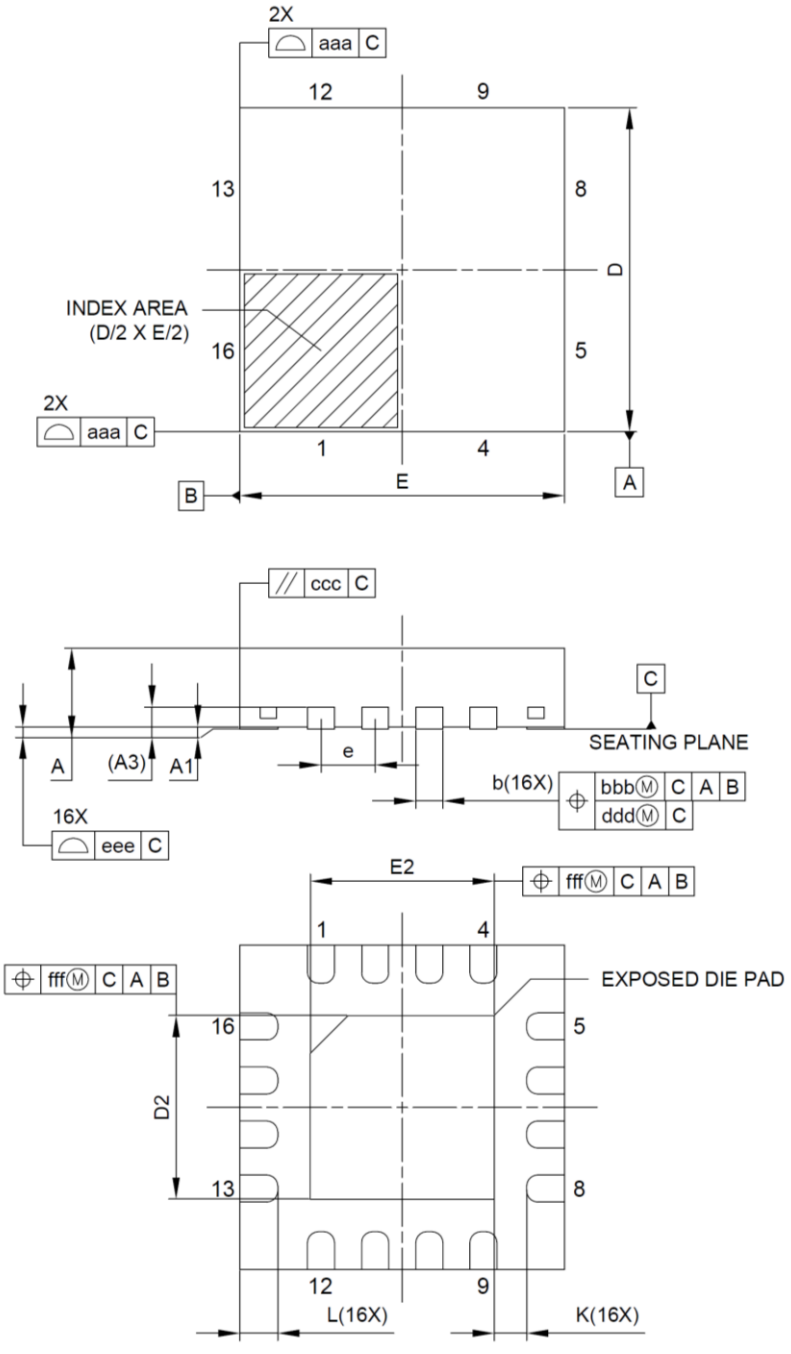


Reference Symbol	Dimensions in millimeters		
	Min	Nom	Max
D	4.85	5.00	5.15
D_1	5.05	5.20	5.35
E	4.20	4.40	4.60
A_2	—	1.50	—
A_1	0.075	0.125	0.175
A	—	—	1.725
b_p	0.17	0.24	0.32
b_1	—	0.22	—
c	0.14	0.17	0.20
c_1	—	0.15	—
θ	0°	—	8°
H_E	6.20	6.40	6.60
e	—	0.65	—
x	—	—	0.13
y	—	—	0.10
Z_D	—	0.225	—
L	0.35	0.50	0.65
L_1	—	1.00	—

R5F1214CMNA, R5F1214CGNA, R5F1214CANA

R5F1214AMNA, R5F1214AGNA, R5F1214AANA

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN016-3x3-0.50	PWQN0016KD-A	0.02



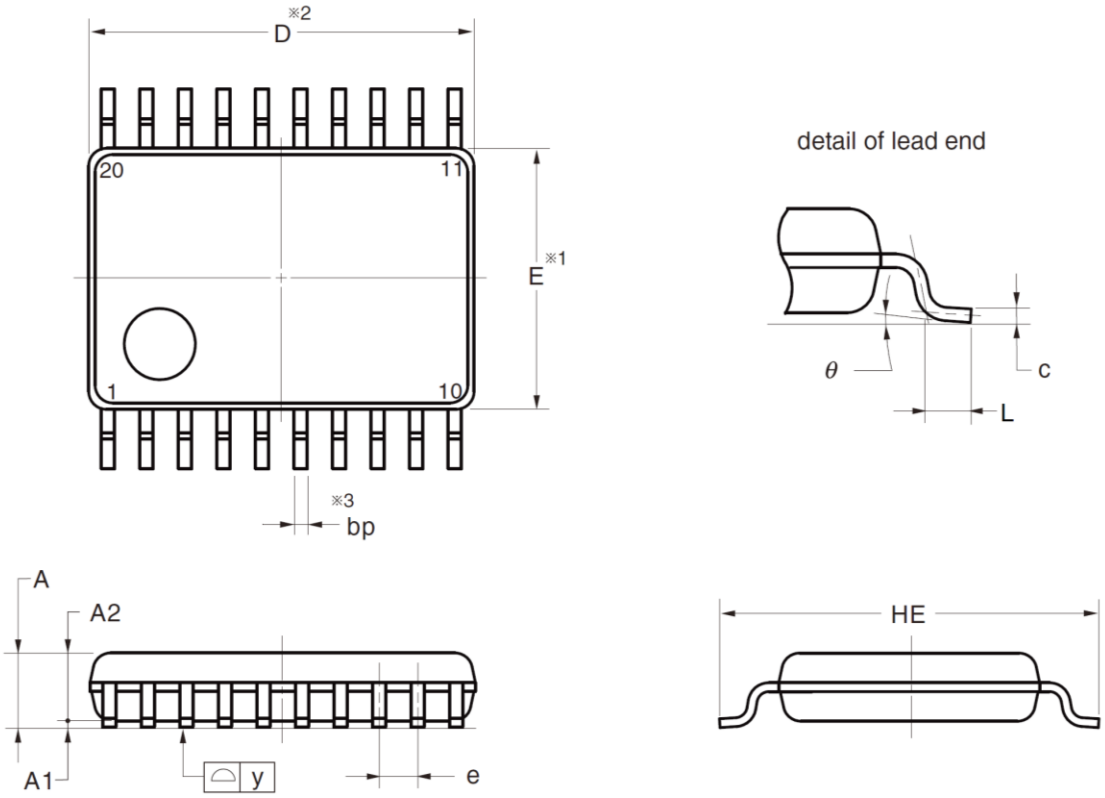
Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A1	0.00	0.02	0.05
A3		0.203 REF.	
b	0.20	0.25	0.30
D		3.00 BSC	
E		3.00 BSC	
e		0.50 BSC	
L	0.30	0.35	0.40
K	0.20	—	—
D2	1.65	1.70	1.75
E2	1.65	1.70	1.75
aaa		0.15	
bbb		0.10	
ccc		0.10	
ddd		0.05	
eee		0.08	
fff		0.10	

28.3 20-pin products

R5F1216CMSP, R5F1216CGSP, R5F1216CASP

R5F1216AMSP, R5F1216AGSP, R5F1216AASP

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°

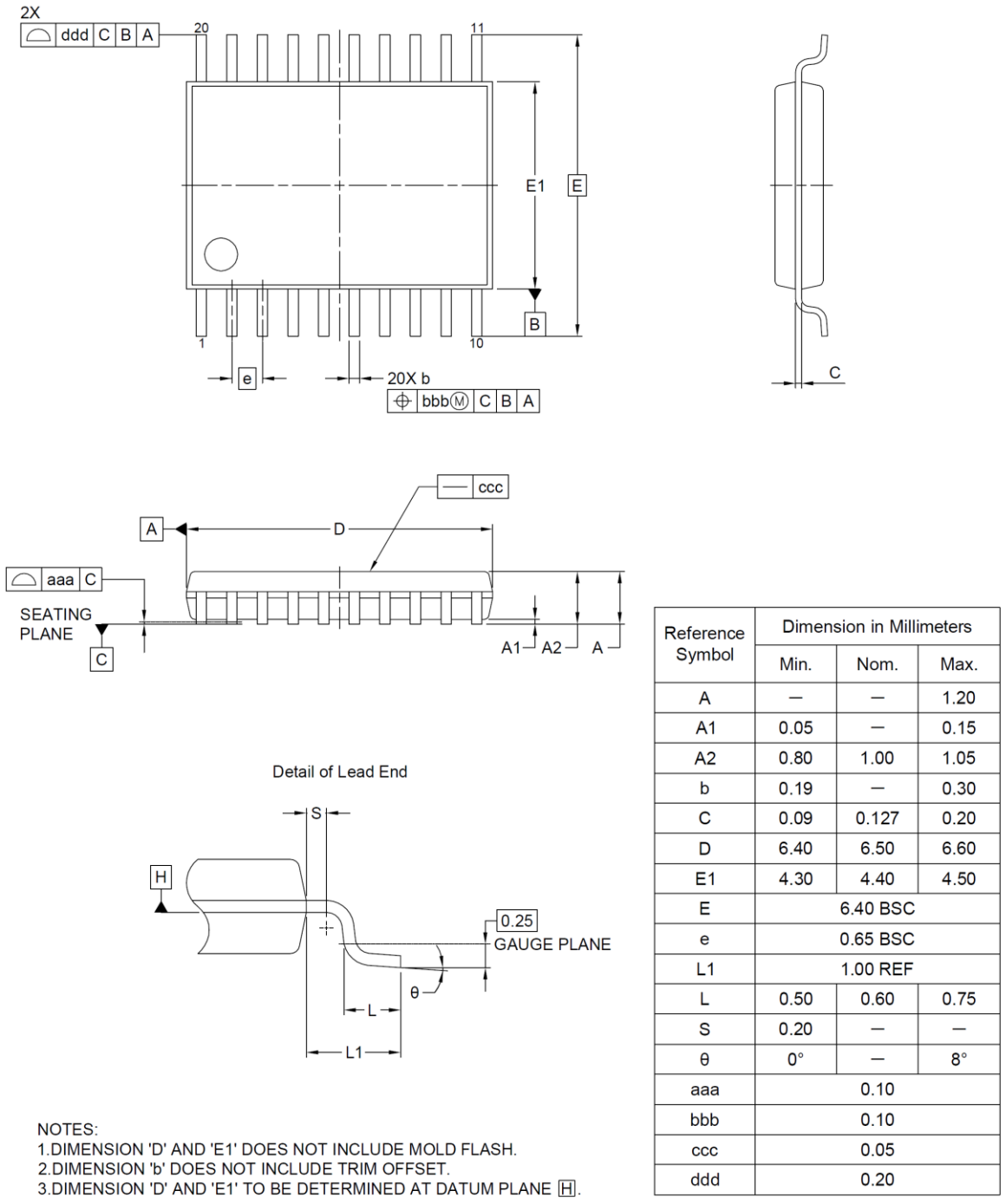
NOTE

1.Dimensions "※1" and "※2" do not include mold flash.

2.Dimension "※3" does not include trim offset.

<R> R5F1216CMSM, R5F1216CASM

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-TSSOP20-4.40x6.50-0.65	PTSP0020JI-A	0.08



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	1.20
A1	0.05	—	0.15
A2	0.80	1.00	1.05
b	0.19	—	0.30
C	0.09	0.127	0.20
D	6.40	6.50	6.60
E1	4.30	4.40	4.50
E	6.40 BSC		
e	0.65 BSC		
L1	1.00 REF		
L	0.50	0.60	0.75
S	0.20	—	—
θ	0°	—	8°
aaa	0.10		
bbb	0.10		
ccc	0.05		
ddd	0.20		

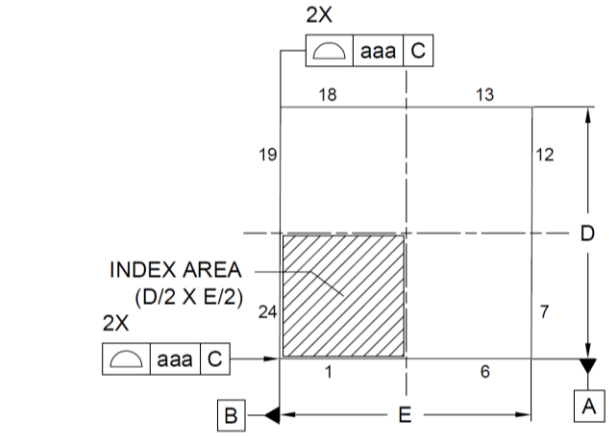
NOTES:
1.DIMENSION 'D' AND 'E1' DOES NOT INCLUDE MOLD FLASH.
2.DIMENSION 'b' DOES NOT INCLUDE TRIM OFFSET.
3.DIMENSION 'D' AND 'E1' TO BE DETERMINED AT DATUM PLANE H.

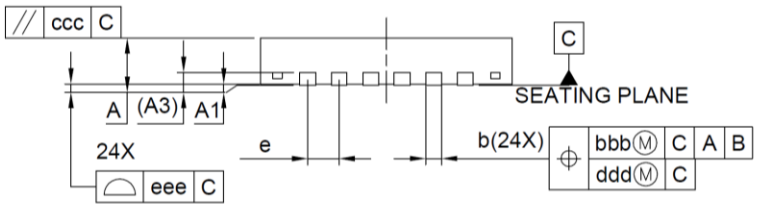
28.4 24-pin products

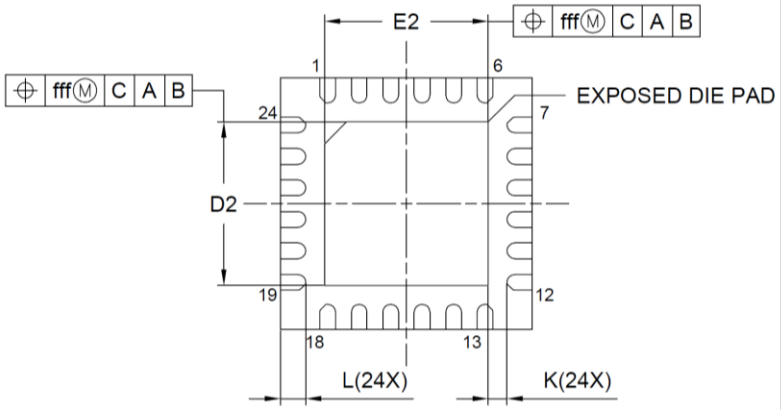
R5F1217CMNA, R5F1217CGNA, R5F1217CANA

R5F1217AMNA, R5F1217AGNA, R5F1217AANA

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN024-4x4-0.50	PWQN0024KF-A	0.04







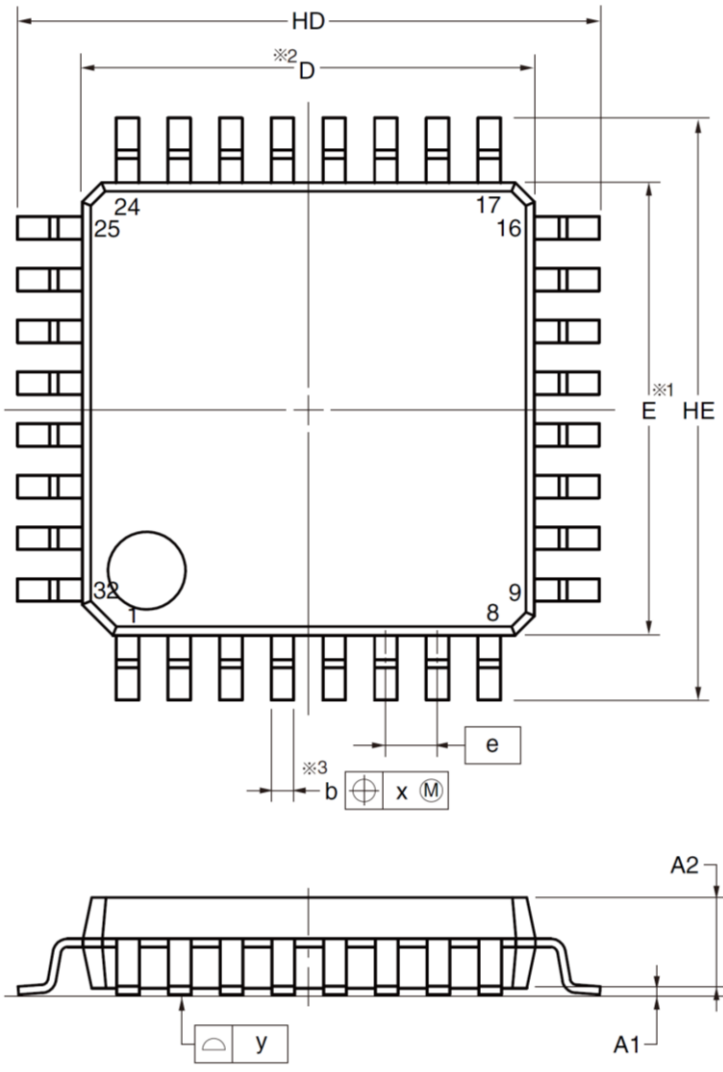
Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A1	0.00	0.02	0.05
A3		0.203 REF.	
b	0.18	0.25	0.30
D		4.00 BSC	
E		4.00 BSC	
e		0.50 BSC	
L	0.35	0.40	0.45
K	0.20	—	—
D2	2.55	2.60	2.65
E2	2.55	2.60	2.65
aaa		0.15	
bbb		0.10	
ccc		0.10	
ddd		0.05	
eee		0.08	
fff		0.10	

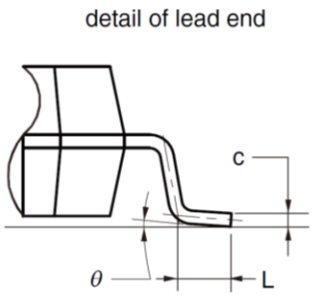
28.5 32-pin products

R5F121BCMFP, R5F121BCGFP, R5F121BCAFP

R5F121BAMFP, R5F121BAGFP, R5F121BAAFP

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LQFP32-7x7-0.80	PLQP0032GB-A	P32GA-80-GBT-1	0.2





(UNIT:mm)

ITEM	DIMENSIONS
D	7.00±0.10
E	7.00±0.10
HD	9.00±0.20
HE	9.00±0.20
A	1.70 MAX.
A1	0.10±0.10
A2	1.40
b	0.37±0.05
c	0.145±0.055
L	0.50±0.20
θ	0° to 8°
e	0.80
x	0.20
y	0.10

NOTE

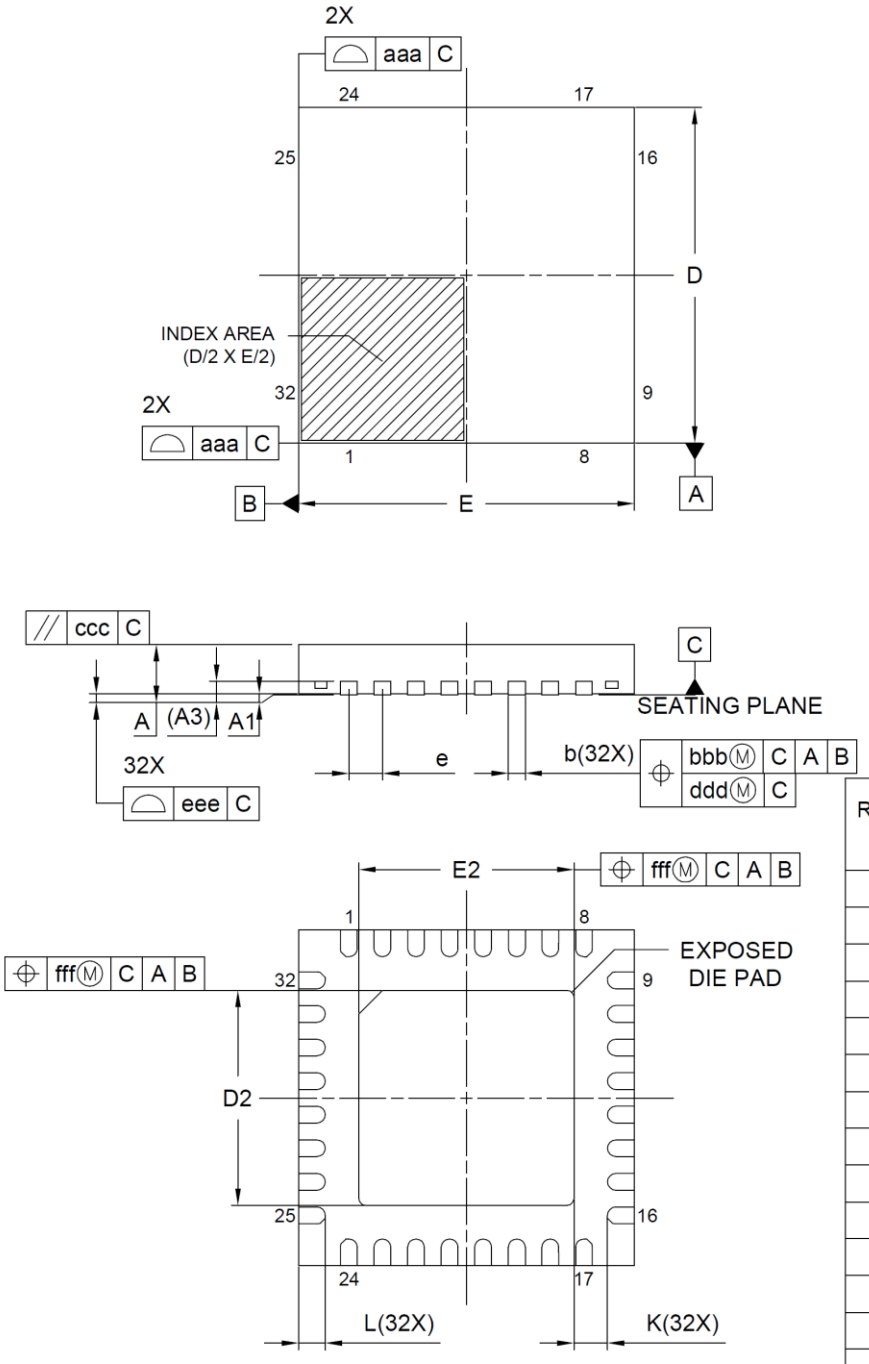
1.Dimensions "※1" and "※2" do not include mold flash.

2.Dimension "※3" does not include trim offset.

R5F121BCMNA, R5F121BCGNA, R5F121BCANA

R5F121BAMNA, R5F121BAGNA, R5F121BAANA

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN032-5x5-0.50	PWQN0032KE-A	0.06



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A_1	0.00	0.02	0.05
A_3		0.203 REF.	
b	0.18	0.25	0.30
D		5.00 BSC	
E		5.00 BSC	
e		0.50 BSC	
L	0.35	0.40	0.45
K	0.20	—	—
D_2	3.15	3.20	3.25
E_2	3.15	3.20	3.25
aaa		0.15	
bbb		0.10	
ccc		0.10	
ddd		0.05	
eee		0.08	
fff		0.10	

APPENDIX A REVISION HISTORY

A.1 Major Revisions in This Edition

Page	Description	Classification
CHAPTER 1 OUTLINE		
p.24	1.1 Features: The mutual capacitance method of the capacitive touch sensing unit (CTSUb), modified	(c)
p.25	Figure 1-1. Part Number, Memory Size, and Package of RL78/G16: Part No., packaging specifications, package type, and Note 1, modified	(c) (d)
p.26	Table 1-1. List of Ordering Part Numbers: TSSOP package, added	(d)
p.31, p.32	1.3.3 20-pin products: TSSOP, added	(d)
CHAPTER 2 PIN FUNCTIONS		
p.60	2.2.2 Pins for each product (pins other than port pins): ANI0 to ANI10, TSCAP, and $\overline{\text{RESET}}$, modified	(a) (c)
CHAPTER 4 PORT FUNCTIONS		
p.154, p.155	Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function: (SCLA0) for P00 and (SCLA0) for P01, modified	(a)
CHAPTER 5 CLOCK GENERATOR		
p.174	Figure 5-1. Block Diagram of Clock Generator, modified	(a)
CHAPTER 8 12-BIT INTERVAL TIMER		
p.395	8.3.3 Interval timer control register (ITMC): Caution 1, modified	(a)
CHAPTER 15 CAPACITIVE TOUCH SENSING UNIT (CTSUb)		
p.699	Table 15-1. CTSU Specifications: Note 1, added	(c)
p.700	Figure 15-3. CTSU Block Diagram: Remark, modified	(a)
p.703	Figure 15-5. Format of CTSU Control Register 0 (CTSUCR0) (1/3): CTSU control block initialization, modified	(c)
p.704	Figure 15-5. Format of CTSU Control Register 0 (CTSUCR0) (2/3): CTSU suspension enable, modified	(a)
p.711	Figure 15-10. Format of CTSU Measurement Channel Register 1 (CTSUMCH1): The CTSU transmit channel flag, modified	(a)
p.730	Figure 15-22. Format of CTSU Error Status Register (CTSUERRS) (1/3): The TSCAP voltage error monitoring, modified	(c)
p.736, p.737	15.4.1 Principles of measurement operation: “switched capacitor filter” modified to “switched capacitor”. Remark, modified. “ICO” modified to “current controlled oscillator (ICO)”.	(a) (c)
p.739	Figure 15-31. Overview of Self-Capacitance Method and Mutual Capacitance Method: The pin name, modified	(a)
p.740	Figure 15-32. CTSU Initial Setting Flowchart: The description, modified	(a) (c)
p.743	Figure 15-35. Software Flowchart and Operation Example of Self-Capacitance Single Scan Mode: The description, modified. “Touch determination processing” modified to “Measurement result determination processing”	(c)
p.759, p.760	15.5 Usage Notes: The term “interrupt” in item 4), modified to “interrupt controller”. The main text, modified in item 6). Items 7), 8), and 9), added.	(c)

Page	Description	Classification
CHAPTER 17 STANDBY FUNCTION		
p.786	17.1 Overview: Caution 1, modified	(a) (c)
p.790	Table 17-2. Operating Status in HALT Mode (2) (1/2): Capacitive touch sensing unit (CTSU), modified	(a)
CHAPTER 28 PACKAGE DRAWINGS		
p.947	28.3 20-pin products: PTSP0020JI-A package drawing, added	(d)

Remark “Classification” in the above table classifies revisions as follows.

(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note,

(d): Addition/change of package, part number, or management division,

(e): Addition/change of related documents

A.2 Revision History of Preceding Editions

Here is the revision history of the preceding editions. Chapter indicates the chapter of each edition.

(1/7)

Edition	Description	Chapter
Rev.0.90	1.1 Features: Capacitive touch sensing unit (CTSUb), modified. The ROM, RAM capacities (data flash, RAM), modified.	CHAPTER 1 OUTLINE
	Figure 1-1. Part Number, Memory Size, and Package of RL78/G16: Packaging specifications and Package type, modified	
	Table 1-1. List of Ordering Part Numbers: The ordering part number (product name, packaging specifications), modified. RENESAS code, added.	
	RESET modified to $\overline{\text{RESET}}$	
	1.3.3 20-pin products: The multiplexed functions of 14-pin timer, modified	
	1.3.5 32-pin products: 32-pin plastic LQFP (7 × 7 mm, 0.8-mm pitch), added	
	1.6 Outline of Functions: High-speed system clock for 16-pin to 32-pin in the table, modified	
	2.1.1 10-pin products to 2.1.5 32-pin products: The order of pin names in Alternate Function, modified. The pin types of P121, P122 and P42, P43, modified.	CHAPTER 2 PIN FUNCTIONS
	2.1.3 20-pin products: The alternate function of P03 in the table, modified	
	RESET modified to $\overline{\text{RESET}}$	
	2.2.1 Functions for each product: The SCK00 function for 10-pin products in the table, modified	
	2.2.2 Pins for each product (pins other than port pins): VCOUT0 and VCOUT1 in the table, modified	
	2.4 Block Diagrams of Pins: The order of pin block diagrams in Figure 2-2 to Figure 2-10, modified	
	Figure 2-2. Pin Block Diagram for Pin Type 3-2-1: The figure, modified	
	Figure 2-5. Pin Block Diagram for Pin Type 7-2-3: The figure title, modified	
	Figure 2-11. Pin Block Diagram for Pin Type 7-33-9: The figure, added	
	Table 3-3. Vector Table: 10-pin setting for 00034H and 00036H addresses, and 20-pin setting for 00046H and 00048H addresses, modified	CHAPTER 3 CPU ARCHITECTURE
	Table 3-5. SFR List (2/3): R/W for the comparator filter control register, modified	
	Table 3-6. Extended SFR (2nd SFR) List (1/6): R/W for the flash sequencer status register L, modified	
	Table 3-6. Extended SFR (2nd SFR) List (1/6): R/W for the flash sequence status register H, modified	
	Table 3-6. Extended SFR (2nd SFR) List (5/6): The manipulable bit range (1-bit) for F030DH, F0386H to F0391H addresses, modified	
	Table 3-6. Extended SFR (2nd SFR) List (6/6): R/W for the CTSU error status register, modified. Note 4, added.	
	Figure 4-3. Format of Pull-up Resistor Option Registers 0, 1, 2, 4, 12 (PU0, PU1, PU2, PU4, PU12): The figure title, modified	CHAPTER 4 PORT FUNCTIONS
	Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function (3/14): TSSELt for the P02 pin, modified	
	Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function (8/14): TSSELt for the P15 pin, modified	

(2/7)

Edition	Description	Chapter
Rev.0.90	5.3 Registers Controlling Clock Generator: 2); Peripheral enable register 1 (PER1), modified	CHAPTER 5 CLOCK GENERATOR
	Figure 5-14. Examples of Incorrect Resonator Connection (2/2): (g) Signals are fetched; The figure, modified	
	5.6.3 Example of setting XT1 oscillation clock: <1>; The register, modified	
	Independent channel operation function: Note 1, modified	CHAPTER 6 TIMER ARRAY UNIT
	6.1.1 Independent channel operation function, 1) Interval timer to 6) Measurement of high-/low-level width of input signal: Remark 1 and Remark 2, added	
	6.1.1 Independent channel operation function, 4) Divider function (channels 0 and 3 only): Channel 3, added. The channel number, changed to index (n).	
	Figure 6-22. Format of Input Switch Control Register (ISC): RXD0 → RxD0, modified	
	6.8.3 Operation as frequency divider (channels 0 and 3 only): Channel 3, added. The channel number, changed to index (n). Remark, added.	
	Figure 6-52. Example of Set Contents of Registers During Operation as Frequency Divider (1/2): Setting of SPLIT0n bit (channel 3), added. Note 1, added.	
	Figure 7-6. Format of Real-time Clock Control Register 1 (RTCC1): Note 1, added	CHAPTER 7 REAL-TIME CLOCK 2
	Figure 7-18. Procedure for Starting Operation of Real-time Clock 2: Interrupt request flags and interrupt mask flags, modified	
	7.4.3 Reading real-time clock 2 counter: The description, modified	
	Figure 7-21. Procedure for Reading from Real-time Clock 2 (when the Alarm Interrupt is in Use): Note 1, Cautions, and Remarks, added	
	7.4.4 Writing to real-time clock 2 counter: The description, modified	
	Figure 7-23. Procedure for Writing to Real-time Clock 2 (when the Alarm Interrupt is in Use): Note 1, Caution 1, Caution 2, and Remarks, added	
	13.1.3 Simplified I2C (IIC00, IIC11, IIC20): Remark 2, modified	CHAPTER 13 SERIAL ARRAY UNIT
	RxDq → RXDq, TxDq → TXDq, modified	
	Figure 13-6. Format of Serial Clock Select Register m (SPSm): The operation clock value of $f_{CLK}/2^{13}$, modified	
	13.3.11 Serial output enable register m (SOEm): Remark, modified	
	13.3.12 Serial output register m (SOm): Remark, modified	
	Figure 13-22. Each Register Setting When Stopping the Operation by Channels (2/2): Remark 1, modified	
	14.5.4 Acknowledge (ACK): The description on the setting of the clock stretch timing, modified	CHAPTER 14 SERIAL INTERFACE IICA
	Figure 14-19. Clock Stretching: The titles for (1) and (2), modified	
	14.5.8 Interrupt request (INTIICA0) generation timing and clock stretching control: The description of (4), modified	
	Figure 14-31. Example of Master to Slave Communications: WTIM0, modified	
	Figure 14-32. Example of Slave to Master Communications: WTIM0, modified	
	Table 15-1. CTSU Specifications: The description on the TSCAP pin, modified	CHAPTER 15 CAPACITIVE TOUCH SENSING UNIT (CTSUb)
	Figure 15-22. Format of CTSU Error Status Register (CTSUERRS) (1/3): The description on the CTSUALMES bit, modified	
	Table 16-1. Interrupt Source List: The trigger for INTWDTI, modified. POR for reset, deleted.	CHAPTER 16 INTERRUPT FUNCTIONS

(3/7)

Edition	Description	Chapter
Rev.0.90	17.1 Overview: Caution 3, modified	CHAPTER 17 STANDBY FUNCTION
	Figure 22-2. Communication with Dedicated Flash Memory Programmer: Note (Note 1), added to EMVDD	CHAPTER 22 FLASH MEMORY
	22.2 Writing to Flash Memory by Using External Device (that Incorporates UART): The description, modified	
	Figure 22-4. Communication with External Device: The signal name of the external device, modified	
	Table 22-7. Flash Memory Control Commands: Writing after erasure, added	
	Figure 22-8. Format of Flash Address Pointer Registers H and L (FLAPH, FLAPL): Bits 6 and 5 of the FLAPH register and bits of the FLAPL register, modified	
	Figure 22-9. Format of Flash End Address Specification Registers H and L (FLSEDH, FLSEDL): Bits 6 and 5 of the FLSEDH register, modified	
	Table 22-9. Method of Setting the FLAPH/L and FLSEDH/L Registers: The description of FLAPH/L and FLSEDH/L register settings, modified	
	22.6.1.2 Flash end address specification registers H and L (FLSEDH, FLSEDL): Block configuration of code flash memory, modified	
	22.6.3 Notes on self-programming: The description in (5), modified	
	23.2 Connecting External Device (that Incorporates UART): The signal name of the external device in the figure, modified	CHAPTER 23 ON-CHIP DEBUG FUNCTION
	Figure 23-3. Memory Spaces Where Debug Monitor Programs Are Allocated: The address value of internal RASM, deleted. Note 3, modified.	
	26.3.1 Pin characteristics: Notes 5 and 6, added	CHAPTER 26 ELECTRICAL SPECIFICATIONS (T_A = −40 to +85°C) (Target)
	26.3.2 Supply current characteristics: f_{MX} = 12 MHz in the table modified to f_{EX} = 16 MHz and f_X = 12 MHz. The TYP. and MAX. values in the table, modified. Notes 1 and 8, modified.	
	26.3.2 Supply current characteristics Peripheral Functions: The CTSU operating current in the table, deleted. The TYP. and MAX. values of self-programming operating current, modified.	
	26.6.1 A/D converter characteristics: The conditions for the conversion time and the analog input voltage in the table, modified	
	26.6.6 CTSU characteristics, deleted	
	26.8 Flash Memory Programming Characteristics: The T_A value, modified	
	26.9 Dedicated Flash Memory Programmer Communication (UART): The T_A value, modified	
	27.2.2 On-chip oscillator characteristics: The TYP. and MAX. values for the high-speed on-chip oscillator clock frequency accuracy (T_A=+85 to +125°C), modified	CHAPTER 27 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C, T_A = −40 to +125°C) (Target)
	27.3.1 Pin characteristics: Note 5, added	
	27.3.2 Supply current characteristics: f_{MX} = 12 MHz in the table modified to f_{EX} = 16 MHz and f_X = 12 MHz. The TYP. and MAX. values in the table, modified. Notes 1 and 8, modified.	
	27.3.2 Supply current characteristics Peripheral Functions: The CTSU operating current in the table, deleted. The MAX. value for A/D converter operating current, modified. The TYP. and MAX. values of self-programming operating current, modified.	
	27.6.1 A/D converter characteristics: The conditions for the conversion time and the analog input voltage in the table, modified	

(4/7)

Edition	Description	Chapter
Rev.0.90	27.6.3 Temperature sensor/internal reference voltage characteristics: The MIN. and MAX. values of the internal reference voltage in the table, ,modified	CHAPTER 27 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C, T_A = −40 to +125°C) (Target)
	27.6.6 CTSU characteristics, deleted	
	28.5 32-pin products: The package drawing, modified	CHAPTER 28 PACKAGE DRAWINGS
Rev.1.00	Figure 1-1. Part Number, Memory Size, and Package of RL78/G16: Packaging specifications (#10), modified	CHAPTER 1 OUTLINE
	Table 1-1. List of Ordering Part Numbers: The product name of the 16-pin plastic SSOP, modified	
	RESET modified to $\overline{\text{RESET}}$	CHAPTER 2 PIN FUNCTIONS
	3.1 Overview: The section, added	CHAPTER 3 CPU ARCHITECTURE
	Table 3-6. Extended SFR (2nd SFR) List (6/6): The symbol for the CTSU trimming register, modified	
	Figure 4-7. Format of Touch Pin Function Select Registers 0, 1 (TSSEL0, TSSEL1): The style, modified	CHAPTER 4 PORT FUNCTIONS
	Table 4-7. Concept of Basic Settings: Output Settings of Unused Alternate Function (Output Function for SAU), modified	
	Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function (1/14 to 3/14): Alternate Function Output (SAU Output Function), modified	
	Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function (4/14): PIORr and Alternate Function Output (SAU Output Function), modified. Pm of P03 ((TO05)), modified.	
	Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function (5/14 to 7/14): PIORr and Alternate Function Output (SAU Output Function), modified	
	Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function (8/14): Alternate Function Output (SAU Output Function), modified	
	Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function (9/14 to 12/14): PIORr and Alternate Function Output (SAU Output Function), modified	
	Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function (13/14, 14/14): PIORr, modified	
	15.3 Registers Controlling CTSU: The symbol for the CTSU trimming register, modified	CHAPTER 15 CAPACITIVE TOUCH SENSING UNIT (CTSUb)
	15.3.3 CTSU control register 1 (CTSUCR1): The description, modified	
	Figure 15-6. Format of CTSU Control Register 1 (CTSUCR1): Bit 2, modified. The CTSUATUNE0 bit, deleted. Note 2, deleted.	
	Figure 15-22. Format of CTSU Error Status Register (CTSUERRS): The description on the CTSUDRV bit, modified	
	Figure 15-26. Format of CTSU Trimming Result Register (CTSUTRESULT): Bits, modified	
	Figure 15-32. CTSU Initial Setting Flowchart: The processing in “Set CTSU power supply”, modified	
	20.1 Overview of Safety Functions: Remark, modified	CHAPTER 20 SAFETY FUNCTIONS

(5/7)

Edition	Description	Chapter
Rev.1.00	CHAPTER 26 ELECTRICAL SPECIFICATIONS (T_A = −40 to +85°C): The section title, modified	CHAPTER 26 ELECTRICAL SPECIFICATIONS (T_A = −40 to +85°C)
	26.8 Flash Memory Programming Characteristics: The condition was added to the code flash/data flash self-programming time	
	26.10 Timing of Entry to Flash Memory Programming Mode: The condition, added	
	CHAPTER 27 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C, T_A = −40 to +125°C): The section title, modified	CHAPTER 27 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C, T_A = −40 to +125°C)
	27.8 Flash Memory Programming Characteristics: The condition in the table, modified. Note 3, added. The condition was added to the code flash/data flash self-programming time.	
	27.10 Timing of Entry to Flash Memory Programming Mode: The condition, added	
Rev.1.10	1.1 Features: Features of the high-speed on-chip oscillator, modified	CHAPTER 1 OUTLINE
	1.1 Features: Features of the capacitive touch sensing unit (CTSUb), modified	
	Figure 1-1. Part Number, Memory Size, and Package of RL78/G16: #06 and #07 were added to the packaging specifications	
	Table 1-1. List of Ordering Part Numbers: Packaging Specifications, modified	
	1.3.1 10-pin products: The pin connection diagram of the 10-pin plastic LSSOP, modified	
	1.3.2 16-pin products: The pin connection diagrams of the 16-pin plastic SSOP and the 16-pin plastic HWQFN, modified. Remark 3, added.	
	1.3.3 20-pin products: The title of the 20-pin plastic LSSOP (4.4 × 6.5 mm, 0.65-mm pitch), modified. The pin connection diagram, modified.	
	Table 1-4. Multiplexed Functions of 20-pin Products: The heading of Pin No., modified	
	1.3.4 24-pin products: The pin connection diagram of the 24-pin plastic HWQFN, modified. Remark 3, added.	
	Table 1-5. Multiplexed Functions of 24-pin Products: The heading of Pin No., modified	
	1.3.5 32-pin products: The pin connection diagram of the 32-pin plastic HWQFN, added. The pin connection diagram of the 32-pin plastic LQFP, modified. Remark 3, added.	
	Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function (1/14, 2/14, 4/14 to 6/14, 8/14 to 11/14): POMp, modified	CHAPTER 4 PORT FUNCTIONS
	Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function (2/14 to 5/14, 9/14): PMCq, modified	
	7.3 Registers Controlling Real-time Clock 2: The reset source in the table, modified	CHAPTER 7 REAL-TIME CLOCK 2

(6/7)

Edition	Description	Chapter
Rev.1.10	CHAPTER 15 CAPACITIVE TOUCH SENSING UNIT (CTSUb): Referenced application notes, added	CHAPTER 15 CAPACITIVE TOUCH SENSING UNIT (CTSUb)
	Figure 15-5. Format of CTSU Control Register 0 (CTSUCR0): The description on the CTSUSNZ bit, modified. Note 4, added.	
	15.3.8 CTSU channel enable control register 0 (CTSUCHAC0): The description, modified	
	15.3.9 CTSU channel enable control register 1 (CTSUCHAC1): The description, modified	
	15.3.10 CTSU channel transmit/receive control register 0 (CTSUCHTRC0): The description, modified	
	15.3.11 CTSU channel transmit/receive control register 1 (CTSUCHTRC1): The description, modified	
	15.3.12 CTSU high-pass noise reduction control register (CTSUDCLKC): The description, modified	
	15.3.13 CTSU status register (CTSUST): The description, modified	
	15.3.21 TSCAP pin setting register (VTSEL): The description, modified	
	15.4.3 Items common to multiple modes (3) Measurement start conditions: Procedures 3 and 4 of the external trigger, modified	
	15.4.3 Items common to multiple modes (4) Intermittent operation using the software trigger (the CTSUCAP bit = 0): The title and description, modified	
	20.3.4 SFR guard function: The description, modified	CHAPTER 20 SAFETY FUNCTIONS
	Figure 20-7. Format of Invalid Memory Access Detection Control Register (IAWCTL): The description on the GCSC bit, modified	
	Figure 20-8. Invalid Access Areas: The figure, modified. Note 1, modified.	
	Figure 20-11. Configuration of Testing of A/D Converter: Note 1, deleted	
	CHAPTER 26 ELECTRICAL SPECIFICATIONS (T_A = −40 to +85°C): The description, modified. Remark, added.	CHAPTER 26 ELECTRICAL SPECIFICATIONS (T_A = −40 to +85°C)
	26.3.2 Supply current characteristics: The unit in the list (condition: f_{SUB} = 32.768 kHz), modified	
	CHAPTER 27 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C, T_A = −40 to +125°C): Remark, added	CHAPTER 27 ELECTRICAL SPECIFICATIONS (T_A = −40 to +105°C, T_A = −40 to +125°C)
	27.3.2 Supply current characteristics: The unit in the list (condition: f_{SUB} = 32.768 kHz), modified	
Rev.1.20	Figure 1-1. Part Number, Memory Size, and Package of RL78/G16: ROM number, added. Packaging Specifications, modified	CHAPTER 1 OUTLINE
	Table 1-1. List of Ordering Part Numbers: Packaging Specifications, modified	
	Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function (4/14): Pm of P03 (RxD1), modified	CHAPTER 4 PORT FUNCTIONS
	Table 4-8. Setting Examples of Registers and Output Latches When Using Pin Function (6/14): PMn and Pm of P10 ((RxD1)), modified	
	Figure 17-3. STOP Mode Release by Interrupt Request Generation (1/3): Note 2, modified	CHAPTER 17 STANDBY FUNCTION
	Figure 17-3. STOP Mode Release by Interrupt Request Generation (2/3): Note 2, modified	
	Figure 17-3. STOP Mode Release by Interrupt Request Generation (3/3): Note 2, modified	

(7/7)

Edition	Description	Chapter
Rev.1.20	Figure 18-2. Timing of Reset by $\overline{\text{RESET}}$ Input: Note 2, modified	CHAPTER 18 RESET FUNCTION
	Figure 19-2. Timing of Internal Reset Signal Generation: SPOR reset processing time, modified	CHAPTER 19 SELECTABLE POWER-ON-RESET CIRCUIT
	20.1 Overview of Safety Functions: The description, modified	CHAPTER 20 SAFETY FUNCTIONS
	20.3.1 CRC operation (general-purpose CRC): The description, modified	
	20.3.3 RAM guard function: The description, modified	
	20.3.4 SFR guard function: The description, modified	
	Figure 23-3. Memory Spaces Where Debug Monitor Programs Are Allocated: The area of 4 bytes in the internal RAM, modified	CHAPTER 23 ON-CHIP DEBUG FUNCTION

RL78/G16 User's Manual: Hardware

Publication Date: Rev.0.50 Jul 29, 2022
Rev.1.30 May 19, 2025

Published by: Renesas Electronics Corporation

RL78/G16

Renesas Electronics Corporation

R01UH0980EJ0130