User's Manual

16

RL78/I1E

User's Manual: Hardware

16-Bit Single-Chip Microcontrollers

Renesas Electronics
www.renesas.com

Rev. 1.30 Mar 2024

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

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

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

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

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

Trademarks

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

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

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

1. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on
   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

How to Use This Manual

Readers

This manual is intended for user engineers who wish to understand the functions of the RL78/I1E and design and develop application systems and programs for these devices.

Purpose

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

Organization

The RL78/I1E manual is separated into two parts: this manual and the software edition (common to the RL78 family).

RL78/I1E User's Manual Hardware (This Manual)	**RL78 Family User's Manual Software**
• Pin functions • Internal block functions • Interrupts • Other on-chip peripheral functions • Electrical specifications	• CPU functions • Instruction set • Explanation of each instruction

How to Read This Manual

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

• To gain a general understanding of functions:

→ Read this manual in the order of the **CONTENTS**. The mark "<R>" shows major revised points. The revised points can be easily searched by copying an "<R>" in the PDF file and specifying it in the "Find what:" field.

• How to interpret the register format:

→ For a bit number enclosed in angle brackets, the bit name is defined as a reserved word in the assembler, and is defined as an sfr variable using the #pragma sfr directive in the compiler.

• To know details of the RL78/I1E 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/I1E User's Manual Hardware	This manual
RL78 Family User's Manual Software	R01US0015E

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

Document Name		Document No.
PG-FP5 Flash Memory Programmer User's Manual		—
	RL78, 78K, V850, RX100, RX200, RX600 (Except RX64x), R8C, SH	R20UT2923E
	Common	R20UT2922E
	Setup Manual	R20UT0930E

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 MPUs & MCUs RL78 Family	R01CP0003E
Semiconductor Package Mount Manual	Note
Semiconductor Reliability Handbook	R51ZZ0001E

Note See the “Semiconductor Device Mount Manual” website (http://www.renesas.com/products/package/index.jsp).

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

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

CONTENTS

CHAPTER 1 OUTLINE

1.1 Features

Ultra-low power consumption technology

- V_{DD} = 2.4 to 5.5 V
- HALT mode
- STOP mode
- SNOOZE mode

RL78 CPU core

- CISC architecture with 3-stage pipeline
- Minimum instruction execution time: Can be changed from high speed (0.03125 μs: @ 32 MHz operation with high-speed on-chip oscillator or PLL clock)^Note to ultra-low speed (1 μs: @ 1 MHz operation with high-speed on-chip oscillator or PLL clock)
- Multiply/divide/multiply & accumulate instructions are supported.
- Address space: 1 MB
- General-purpose registers: (8-bit register × 8) × 4 banks
- On-chip RAM: 8 KB

Note For industrial applications (M; T_A = –40 to +125°C): 0.04167 μs @ 24 MHz operation with high-speed on-chip oscillator or PLL clock

Code flash memory

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

Data flash memory

- Data flash memory: 4 KB
- Back ground operation (BGO): Instructions can be executed from the program memory while rewriting the data flash memory.
- Number of rewrites: 1,000,000 times (TYP.)
- Voltage of rewrites: V_{DD} = 2.4 to 5.5 V

High-speed on-chip oscillator

- Select from 32 MHz, 24 MHz, 16 MHz, 12 MHz, 8 MHz, 6 MHz, 4 MHz, 3 MHz, 2 MHz, and 1 MHz
- High accuracy: ±2.0% (V_{DD} = 2.4 to 5.5 V, T_A = –40 to +105°C)
  ±3.0% (V_{DD} = 2.4 to 5.5 V, T_A = –40 to +125°C)

Operating ambient temperature

- 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 power-on-reset (POR) circuit
- On-chip voltage detector (LVD) (Select interrupt and reset from 7 levels)

Data transfer controller (DTC)

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

Event link controller (ELC)

- Event signals of 16 types can be linked to the specified peripheral function.

Serial interfaces

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

Timer

- 16-bit timer: 8 channels
  (Timer Array Unit (TAU): 6 channels, timer RJ: 1 channel, timer RG: 1 channel)
- Interval timer: 1 channel
- Real-time clock: 1 channel (calendar for 99 years, alarm function, and clock correction function)
- Watchdog timer: 1 channel (operable with the dedicated low-speed on-chip oscillator)

Analog front-end (AFE) power supply

- Sensor power supply (SBIAS) output: 0.5 V to 2.2 V

24-bit ΔΣ A/D converter with programmable gain instrumentation amplifier

- 24-bit second-order ΔΣ A/D converter (AV_{DD} = 2.7 to 5.5 V)
- SNDR: 85 dB (TYP.)
- Output data rate: 488 sps to 15.625 ksps in normal mode
  61 sps to 1.953 ksps in low power mode
- Programmable gain instrumentation amplifier input: 3 or 4 channels
  (differential input mode or single-ended input mode can be specified for each input channel)
- DAC for offset adjustment
- Variable gain: x1 to x64
- On-chip temperature sensor

10-bit A/D converter

- 8-bit/10-bit successive approximation A/D converter (AV_{DD} = 2.7 to 5.5 V)
- Analog input: 8 or 10 channels, sensor power supply (SBIAS), and internal reference voltage
- Internal reference voltage (1.45 V)

Configurable amplifier

- Matrix configuration that consists of 3 operational amplifier channels and a configurable switch (AV_{DD} = 2.7 to 5.5 V)
- Can be used as a 2- or 3-channel general operational amplifier
- Operational amplifier output: 3 channels
- General-purpose Analog I/O ports: 5 or 6 channels
- Offset voltage calibration

D/A converter

- 12-bit R-2R resistor ladder type D/A converter (AV_{DD} = 2.7 to 5.5 V)
- Analog output: 1 channel (via configurable amplifier)

I/O port

- CMOS I/O: 10 to 14 (N-ch open drain I/O [withstanding voltage of V_{DD}]: 6, CMOS I/O: 7 to 11, CMOS input: 3)
- Can be set to TTL input buffer and on-chip pull-up resistor
- Different potential interface: Can connect to a 2.5/3 V device
- On-chip clock output/buzzer output controller

Others

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

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

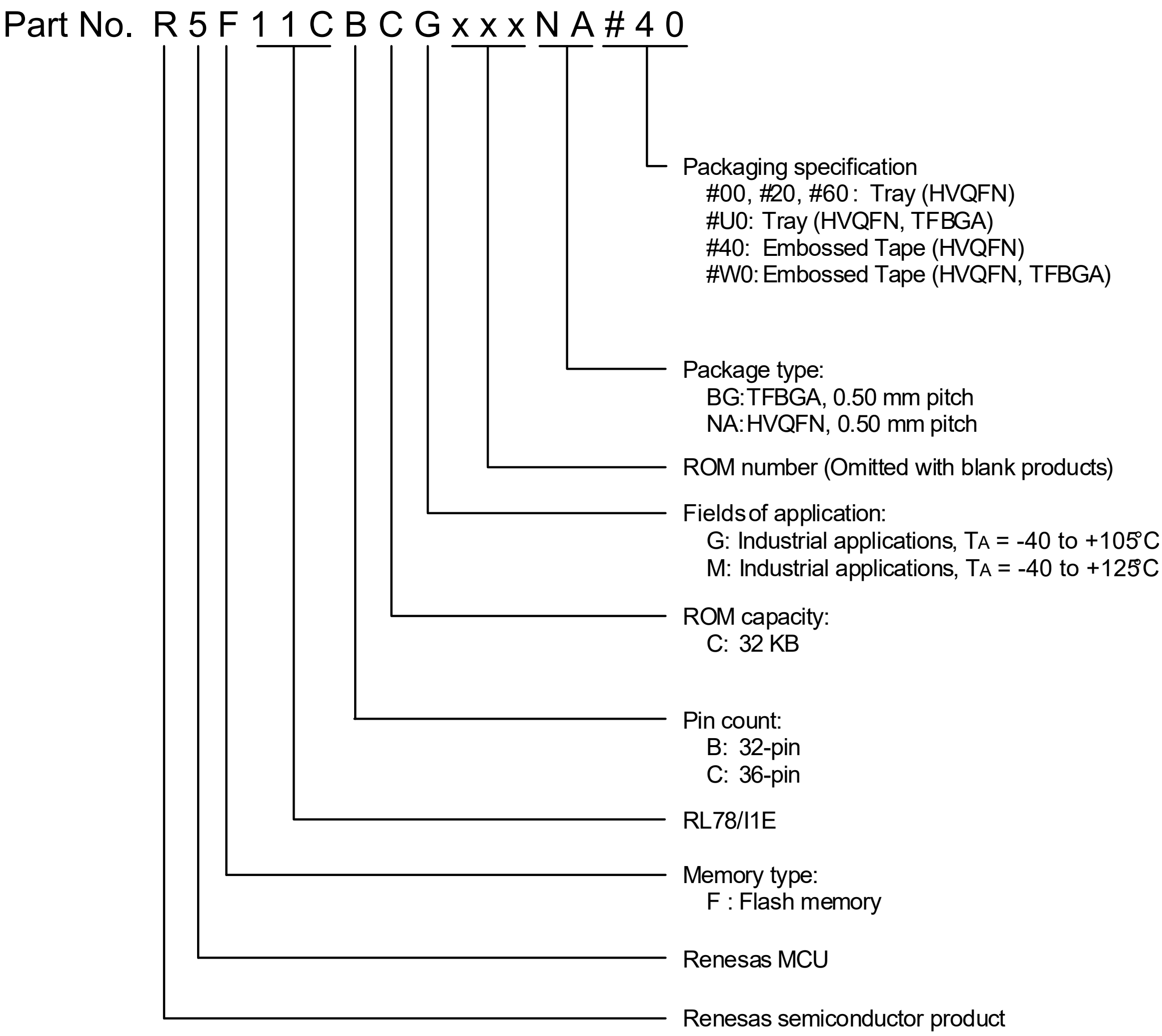
○ ROM, RAM capacities

Flash ROM	Data flash	RAM	RL78/I1E	
			32 pins	36 pins
32 KB	4 KB	8 KB	R5F11CBC	R5F11CCC

1.2 Ordering Information

Figure 1 - 1 Part Number, Memory Size, and Package of RL78/I1E

<R>



<R>

Pin count	Package	Fields of Application Note	Ordering Part Number
32 pins	32-pin plastic HVQFN (5 × 5 mm, 0.5 mm pitch)	G	R5F11CBCGNA#20 R5F11CBCGNA#40 R5F11CBCGNA#00 R5F11CBCGNA#60
		M	R5F11CBCMNA#U0 R5F11CBCMNA#W0
36 pins	36-pin plastic TFBGA (4 × 4 mm, 0.5 mm pitch)	G	R5F11CCCGBG#U0 R5F11CCCGBG#W0
		M	R5F11CCCMBG#U0 R5F11CCCMBG#W0

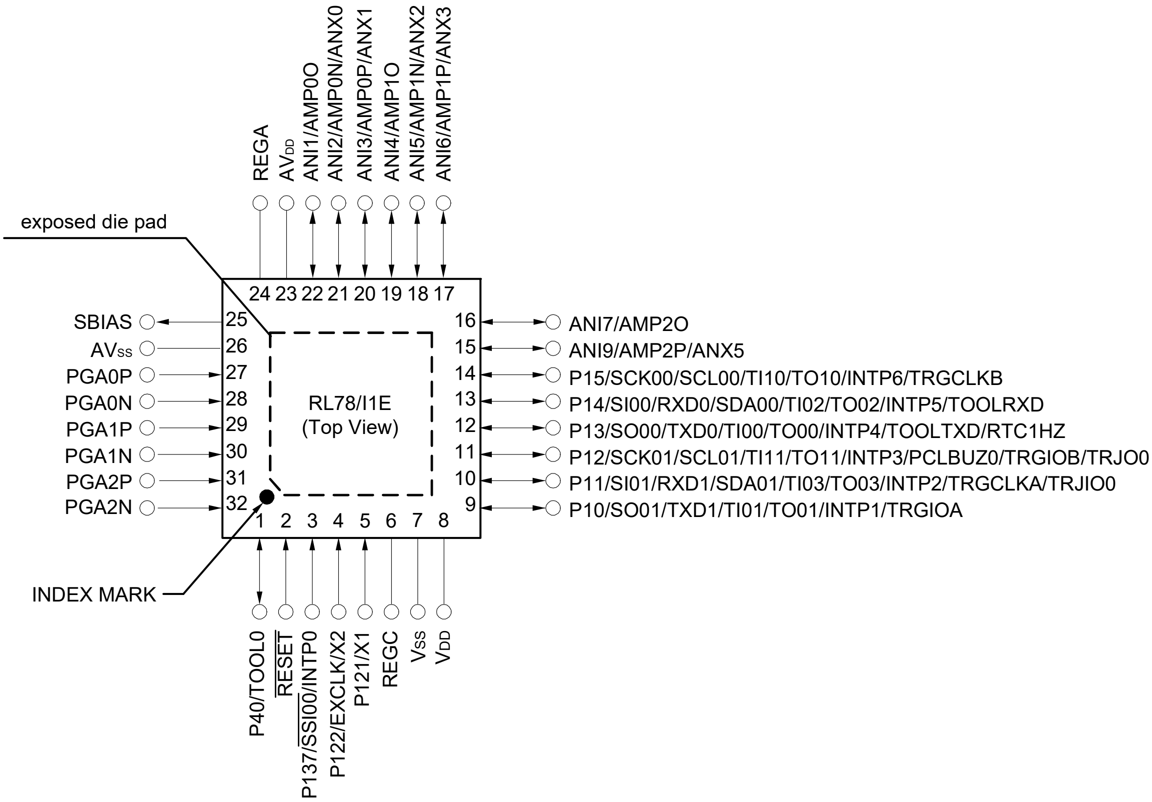
Note For the fields of application, refer to **Figure 1 - 1 Part Number, Memory Size, and Package of RL78/I1E**.

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 32-pin products

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

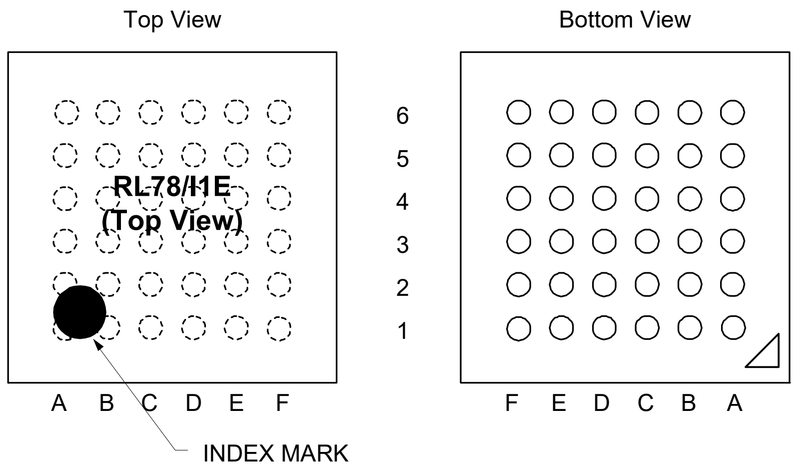


Caution 1. Connect the REGC pin to the Vss pin via a capacitor (0.47 to 1 µF).
Caution 2. Connect the REGA pin to the AVss pin via a capacitor (0.22 µF).
Caution 3. Make the AVss pin the same potential as the Vss pin.
Caution 4. Make the AVDD pin the same potential as the VDD pin.
Caution 5. Connect the SBIAS pin to the AVss pin via a capacitor (0.22 µF).

Remark 1. It is recommended to connect an exposed die pad to Vss.

1.3.2 36-pin products

• 36-pin plastic TFBGA (4 × 4 mm, 0.5 mm pitch)



	A	B	C	D	E	F	
6	PGA2P	PGA1N	PGA1P	PGA0P	PGA3P	AVSS	6
5	PGA2N	P40/TOOL0	PGA0N	PGA3N	REGA	SBIAS	5
4	RESET	P137/SSI00/ INTP0	P11/SI01/RXD1/ SDA01/TI03/ TO03/INTP2/ TRGCLKA/ TRJIO0	P12/SCK01/ SCL01/TI11/ TO11/INTP3/ PCLBUZ0/ TRGIOB/TRJO0	ANI0	AVDD	4
3	P122/EXCLK/X2	P15/SCK00/ SCL00/TI10/ TO10/INTP6/ TRGCLKB	P10/SO01/TXD1/ TI01/TO01/ INTP1/TRGIOA	ANI3/AMP0P/ ANX1	ANI2/AMP0N/ ANX0	ANI1/AMP0O	3
2	P121/X1	REGC	P14/SI00/RXD0/ SDA00/TI02/ TO02/INTP5/ TOOLRXD	P41/ANI6/ AMP1P/ANX3	P42/ANI5/ AMP1N/ANX2	ANI4/AMP1O	2
1	VDD	VSS	P13/SO00/TXD0/ TI00/TO00/INTP4/ TOOLTXD/ RTC1HZ	P16/INTP7/ANI9/ AMP2P/ANX5	P17/ANI8/ AMP2N/ANX4	ANI7/AMP2O	1
	A	B	C	D	E	F	

Caution 1. Connect the REGC pin to the VSS pin via a capacitor (0.47 to 1 μF).
Caution 2. Connect the REGA pin to the AVSS pin via a capacitor (0.22 μF).
Caution 3. Make the AVSS pin the same potential as the VSS pin.
Caution 4. Make the AVDD pin the same potential as the VDD pin.
Caution 5. Connect the SBIAS pin to the AVSS pin via a capacitor (0.22 μF).

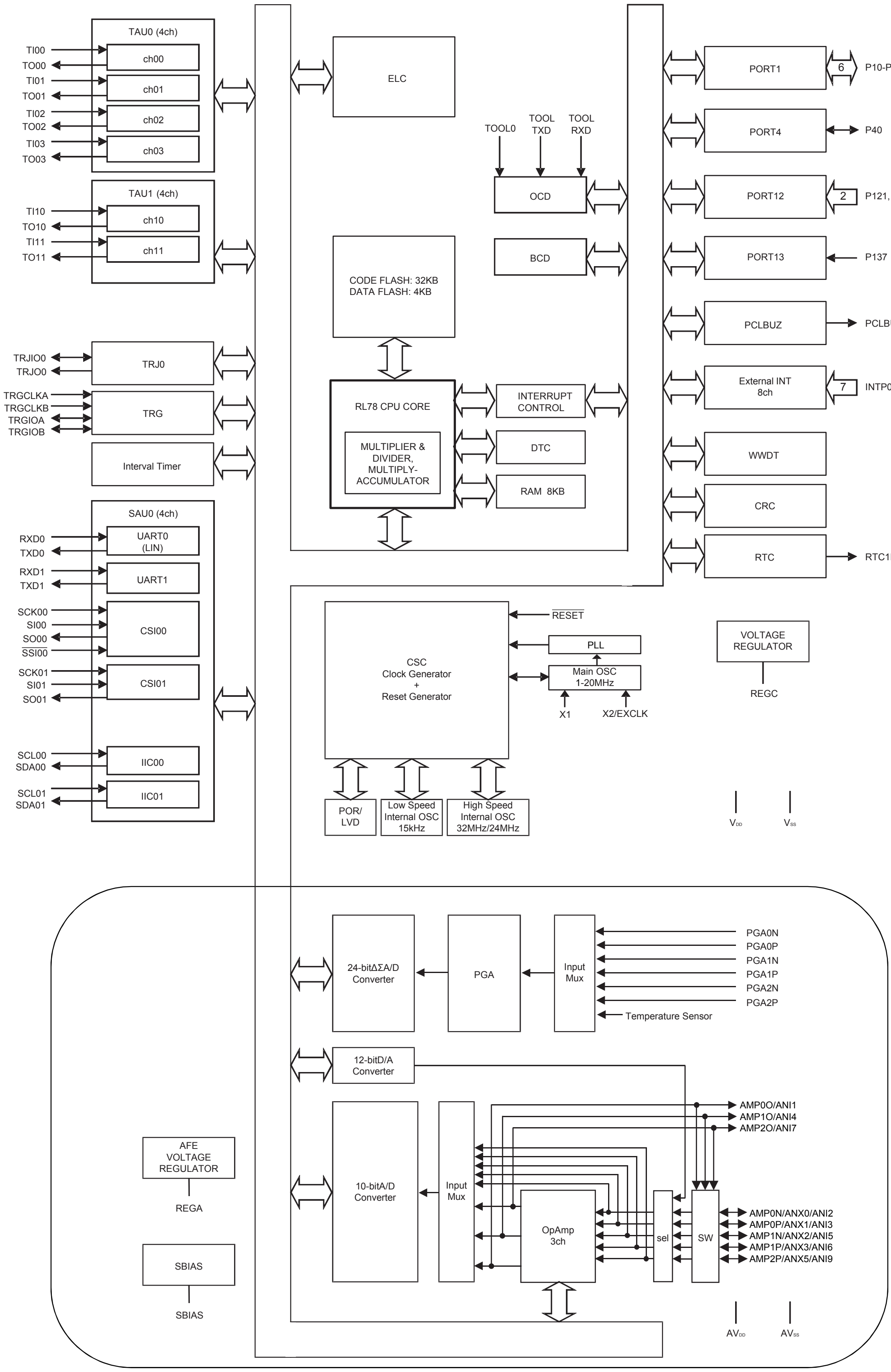
1.4 Pin Identification

ANI0 to ANI9:	Analog input
AMP0P to AMP2P:	Operational amplifier positive input
AMP0N to AMP2N:	Operational amplifier negative input
AMP0O to AMP2O:	Operational amplifier output
ANX0 to ANX5:	General-purpose analog ports for operational amplifier
AV_{DD}:	Power supply for analog
AV_{SS}:	Ground for analog
EXCLK:	External clock input (main system clock)
INTP0 to INTP7:	External interrupt input
P10 to P17:	Port 1
P40 to P42:	Port 4
P121, P122:	Port 12
P137:	Port 13
PCLBUZ0:	Programmable clock output/ buzzer output
PGA0N to PGA3N:	PGA negative analog input
PGA0P to PGA3P:	PGA positive analog input

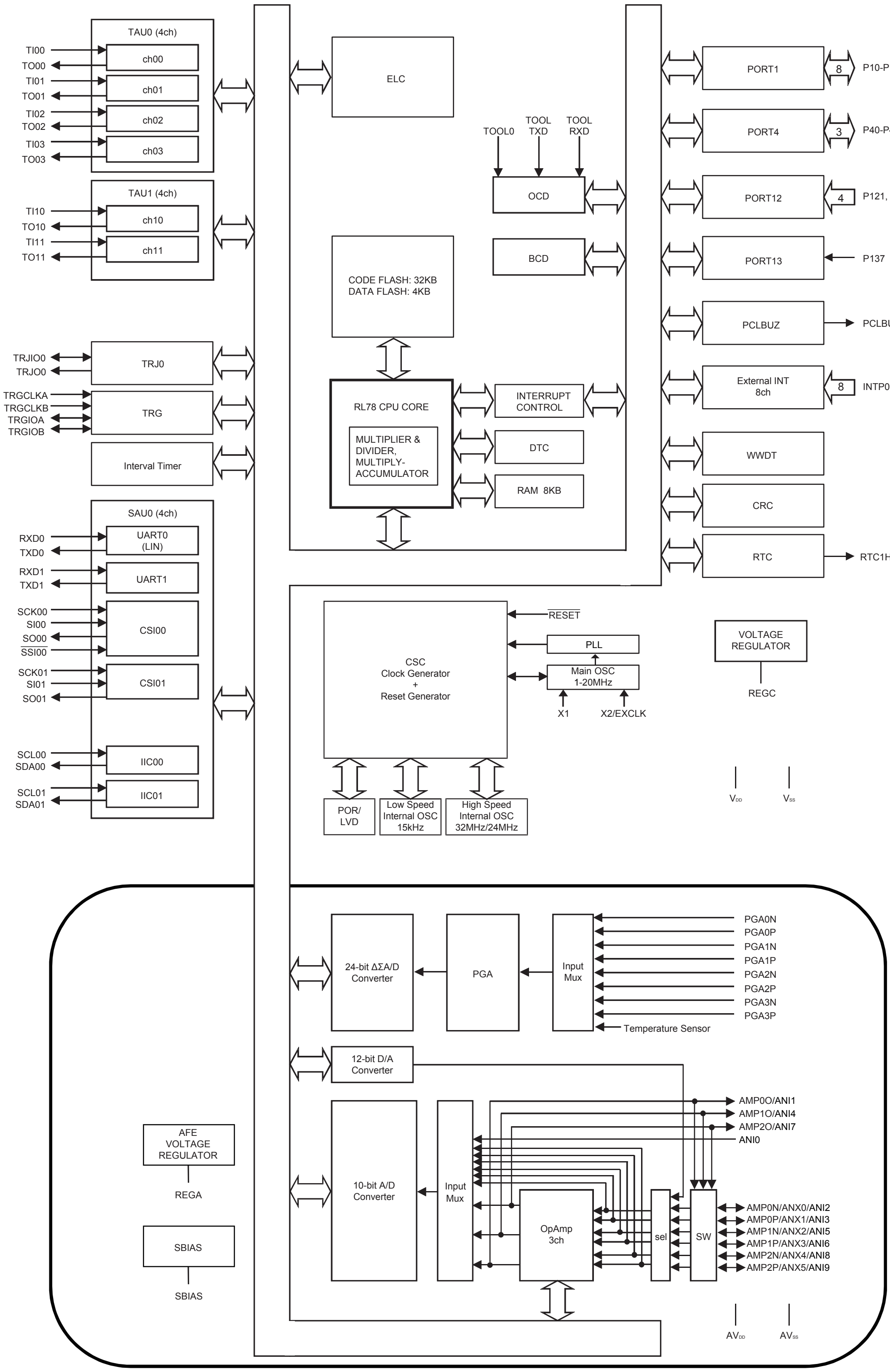
$\overline{\text{RESET}}$:	Reset
REGA:	Regulator capacitance for analog
REGC:	Regulator capacitance
RTC1HZ:	Real-time clock correction
RxD0, RxD1:	Receive data
SBIAS:	Bias output for MEMS sensor
SCK00, SCK01:	Serial clock input/output
SCL00, SCL01:	Serial clock output
SI00, SI01:	Serial data input
SO00, SO01:	Serial data output
TI00 to TI03, TI10, TI11:	Timer input
TO00 to TO03, TO10, TO11, TRJO0:	Timer output
TOOL0:	Data input/output for tools
TOOLRxD, TOOLTxD:	Data input/output for external devices
TRGCLKA, TRGCLKB:	Timer external clock input
TRGIOA, TRGIOB, TRJIO0:	Timer input/output
TxD0, TxD1:	Transmit data
V_{DD}:	Power supply
V_{SS}:	Ground
X1, X2:	Crystal oscillator (main system clock)

1.5 Block Diagram

1.5.1 32-pin products



1.5.2 36-pin products



1.6 Outline of Functions

[32-pin, 36-pin products]

(1/2)

Item		32-pin	36-pin
		R5F11CBC	R5F11CCC
Code flash memory		32 KB	
Data flash memory		4 KB	
RAM		8 KB	
Address space		1 MB	
Main system clock	High-speed system clock	X1 (crystal/ceramic) oscillation, external main system clock input (EXCLK) 1 to 20 MHz: V_{DD} = 2.7 to 5.5 V, 1 to 16 MHz: V_{DD} = 2.4 to 2.7 V	
	High-speed on-chip oscillator clock (f_{IH})	1 to 32 MHz (V_{DD} = 2.7 to 5.5 V)Note 1 1 to 16 MHz (V_{DD} = 2.4 to 5.5 V)	
	PLL clock (f_{PLL} divided by 2, 4, or 8)	3 to 32 MHz (V_{DD} = 2.7 to 5.5 V)Note 2 3 to 16 MHz (V_{DD} = 2.4 to 5.5 V)	
General-purpose register		8 bits × 32 registers (8 bits × 8 registers × 4 banks)	
Minimum instruction execution time		0.03125 μs (high-speed on-chip oscillator clock: f_{IH} = 32 MHz operation)Note 3	
		0.03125 μs (PLL clock: f_{PLL} = 64 MHz, f_{IH} = 32 MHz operation)Note 4	
		0.05 μs (high-speed system clock: f_{MX} = 20 MHz operation)	
Instruction set		• Data transfer (8/16 bits) • Adder and subtractor/logical operation (8/16 bits) • Multiplication (8 bits × 8 bits, 16 bits × 16 bits), division (16 bits ÷ 16 bits, 32 bits ÷ 32 bits) • Multiplication and Accumulation (16 bits × 16 bits + 32 bits) • Rotate, barrel shift, and bit manipulation (Set, reset, test, and Boolean operation), etc.	
I/O port	Total	10	14
	CMOS I/O	7	11
	CMOS input	3	3
Timer	16-bit timer	8 channels (TAU: 6 channels, Timer RJ: 1 channel, Timer RG: 1 channel)	
	Watchdog timer	1 channel	
	Real-time clock (RTC)	1 channel	
	Interval timer	1 channel	
	Timer output	Timer outputs: 10 channels PWM outputs: 9 channels	
	RTC output	1	
Clock output/buzzer output		1	
		2.44 kHz, 4.88 kHz, 9.76 kHz, 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz (Main system clock: f_{MAIN} = 20 MHz operation)	
8/10-bit A/D converter		8 channels	10 channels
Serial interface		Simplified SPI (CSI): 2 channels/UART: 2 channels (UART supporting LIN-bus: 1 channel)/simplified I^2C: 2 channels	

Note 1. 1 to 24 MHz (V_{DD} = 2.7 to 5.5 V) for M products (industrial applications, T_A = –40 to +125°C)

Note 2. 3 to 24 MHz (V_{DD} = 2.7 to 5.5 V) for M products (industrial applications, T_A = –40 to +125°C)

Note 3. 0.04167 μs (high-speed on-chip oscillator clock: f_{IH} = 24 MHz operation) for M products (industrial applications, T_A = –40 to +125°C)

Note 4. 0.04167 μs (PLL clock: f_{PLL} = 64 MHz, f_{IH} = 24 MHz operation) for M products (industrial applications, T_A = –40 to +125°C

(2/2)

Item		32-pin	36-pin
		R5F11CBC	R5F11CCC
Data transfer controller (DTC)		22 sources	
Event link controller (ELC)		Event input: 16 Event trigger output: 7	
Vectored interrupt sources	Internal	23	23
	External	7	8
ΔΣ A/D converter	24-bit	3 channels	4 channels
	AFE temperature sensor	1 channel	
Operational amplifier	3-pin	3 channels Note 1	3 channels
	General-purpose port	5 channels	6 channels
D/A converter	12-bit	1 channel	
Reset		• Reset by $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on-reset • Internal reset by voltage detector • Internal reset by illegal instruction execution Note 2 • Internal reset by RAM parity error • Internal reset by illegal-memory access	
Power-on-reset circuit		• Power-on-reset: 1.56 ±0.03 V • Power-down-reset: 1.55 ±0.03 V	
Voltage detector		• At rise: 2.55 V to 4.64 V (7 steps) • At fall: 2.61 V to 4.74 V (7 steps)	
On-chip debug function		Provided	
Power supply voltage		V_{DD} = 2.4 to 5.5 V	
Operating ambient temperature		T_A = –40 to +105°C (G: Industrial applications), T_A = –40 to +125°C (M: Industrial applications)	

Note 1. When each of the 3 channels is in use as an independent amplifier, at least one channel must be in a voltage follower configuration.

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

CHAPTER 2 PIN FUNCTIONS

2.1 Port Functions

The relationship between these power supplies and the pins is shown below.

Table 2 - 1 Pin I/O Buffer Power Supplies

(1) 32-pin, 36-pin products

Power Supply	Corresponding Pins
V_{DD}	All pins
AV_{DD}	AFE pins

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

2.1.1 32-pin products

Function Name	Pin Type	I/O	After Reset	Alternate Function	Function
P10	7-1-4	I/O	Input port	SO01/TXD1/TI01/TO01/INTP1/TRGIOA	Port 1. 6-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting at input port. Input of P11, P12, P14 and P15 can be set to TTL input buffer. Output of P10 to P15 can be set to N-ch open-drain output (V_{DD} tolerance).
P11	8-1-4			SI01/RXD1/SDA01/TI03/TO03/INTP2/ TRGCLKA/TRJIO0	
P12	8-1-4			SCK01/SCL01/TI11/TO11/INTP3/PCLBUZ0/ TRGIOB/TRJO0	
P13	7-1-4			SO00/TXD0/TI00/TO00/INTP4/TOOLTXD/ RTC1HZ	
P14	8-1-4			SI00/RXD0/SDA00/TI02/TO02/INTP5/ TOOLRXD	
P15	8-1-4			SCK00/SCL00/TI10/TO10/INTP6/TRGCLKB	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 1-bit I/O port. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting at input port.
P121	2-2-1	Input	Input port	X1	Port 12. 2-bit input-only port.
P122	2-2-1			EXCLK/X2	
P137	2-1-2	Input	Input port	$\overline{SSI00}$/INTP0	Port 13. 1-bit input-only port.

2.1.2 36-pin products

Function Name	Pin Type	I/O	After Reset	Alternate Function	Function
P10	7-1-4	I/O	Input port	SO01/TXD1/TI01/TO01/INTP1/TRGIOA	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. Input of P11, P12, P14 and P15 can be set to TTL input buffer. Output of P10 to P15 can be set to N-ch open-drain output (V_{DD} tolerance). P16 and P17 can be set to analog input or analog output.Note
P11	8-1-4			SI01/RXD1/SDA01/TI03/TO03/INTP2/ TRGCLKA/TRJIO0	
P12	8-1-4			SCK01/SCL01/TI11/TO11/INTP3/PCLBUZ0/ TRGIOB/TRJO0	
P13	7-1-4			SO00/TXD0/TI00/TO00/INTP4/TOOLTXD/ RTC1HZ	
P14	8-1-4			SI00/RXD0/SDA00/TI02/TO02/INTP5/ TOOLRXD	
P15	8-1-4			SCK00/SCL00/TI10/TO10/INTP6/TRGCLKB	
P16	6-3-2		Analog function	INTP7/ANI9/AMP2P/ANX5	
P17	6-3-2			ANI8/AMP2N/ANX4	
P40	7-1-3	I/O	Input port	TOOL0	Port 4. 3-bit I/O port. Input/output can be specified. Use of an on-chip pull-up resistor can be specified by a software setting at input of P40. P41 and P42 can be set to analog input or analog output. Note
P41	6-3-2		Analog function	ANI6/AMP1P/ANX3	
P42	6-3-2			ANI5/AMP1N/ANX2	
P121	2-2-1	Input	Input port	X1	Port 12. 2-bit input-only port.
P122	2-2-1			EXCLK/X2	
P137	2-1-2	Input	Input port	$\overline{\text{SSI00}}$/INTP0	Port 13. 1-bit input-only port.

Note Each pin can be specified as either digital or analog by setting port mode control register x (PMCx) (Can be specified in 1-bit units).

2.2 Functions Other Than Port Pins

2.2.1 Alternate functions other than AFE

(1/2)

Pin Function	I/O	Function	Pin Count	
			32-pin	36-pin
INTP0	Input	External interrupt request input	√	√
INTP1	Input		√	√
INTP2	Input		√	√
INTP3	Input		√	√
INTP4	Input		√	√
INTP5	Input		√	√
INTP6	Input		√	√
INTP7	Input		—	√
TI00	Input	External count clock/capture trigger input to 16-bit timer 00	√	√
TI01	Input	External count clock/capture trigger input to 16-bit timer 01 (can be used in 8-bit mode)	√	√
TI02	Input	External count clock/capture trigger input to 16-bit timer 02	√	√
TI03	Input	External count clock/capture trigger input to 16-bit timer 03 (can be used in 8-bit mode)	√	√
TI10	Input	External count clock/capture trigger input to 16-bit timer 10	√	√
TI11	Input	External count clock/capture trigger input to 16-bit timer 11	√	√
TO00	Output	16-bit timer 00 output	√	√
TO01	Output	16-bit timer 01 output (can be used in 8-bit mode)	√	√
TO02	Output	16-bit timer 02 output	√	√
TO03	Output	16-bit timer 03 output (can be used in 8-bit mode)	√	√
TO10	Output	16-bit timer 10 output	√	√
TO11	Output	16-bit timer 11 output	√	√
SI00	Input	Serial data input to serial interface CSI00	√	√
SI01	Input	Serial data input to serial interface CSI01	√	√
SO00	Output	Serial data output from serial interface CSI00	√	√
SO01	Output	Serial data output from serial interface CSI01	√	√
SCK00	I/O	Clock I/O to/from serial interface CSI00	√	√
SCK01	I/O	Clock I/O to/from serial interface CSI01	√	√
SSI00	Input	Chip select input for serial interface CSI00	√	√
TXD0	Output	Serial data output from serial interface UART0	√	√
TXD1	Output	Serial data output from serial interface UART1	√	√
RXD0	Input	Serial data input to serial interface UART0	√	√
RXD1	Input	Serial data input to serial interface UART1	√	√
SCL00	Output	Clock output from serial interface IIC00	√	√
SCL01	Output	Clock output from serial interface IIC01	√	√
SDA00	I/O	Serial data I/O to/from serial interface IIC00	√	√
SDA01	I/O	Serial data I/O to/from serial interface IIC01	√	√
PCLBUZ0	Output	Clock output/buzzer output 0	√	√
RTC1HZ	Output	Real-time clock correction clock (1 Hz) output	√	√

(2/2)

Pin Function	I/O	Function	Pin Count	
			32-pin	36-pin
TRGIOA	I/O	Timer RG I/O	√	√
TRGCLKA	Input	External clock input to timer RG	√	√
TRGIOB	I/O	Timer RG I/O	√	√
TRGCLKB	Input	External clock input to timer RG	√	√
TRJIO0	I/O	Timer RJ I/O	√	√
TRJO0	Output	Timer RJ output	√	√
EXCLK	Input	External clock input for main system clock	√	√
X1	—	Connecting a resonator for main system clock	√	√
X2	—		√	√
RESET	Input	Active-low system reset input Connect to V_{DD} directly or via a resistor when external reset is not used.	√	√
REGC	—	Pin for connecting capacitor for stabilizing regulator output for internal operation Connect to V_{SS} directly or via a capacitor (0.47 to 1 μF). Use a capacitor whose characteristics are as desirable as possible because this capacitor is used to stable the internal voltage.	√	√
V_{DD}	—	Positive power supply	√	√
V_{SS}	—	Ground potential	√	√
TOOLTxD	Output	UART serial data transmission pin for connecting an external device for flash memory programming	√	√
TOOLRxD	Input	UART serial data reception pin for connecting an external device for flash memory programming	√	√
TOOL0	I/O	Flash memory programmer/debugger data I/O	√	√

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

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

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

For details, see 30.4 Programming Method.

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

2.2.2 AFE pin functions

Pin Function	I/O	Function	Pin Count	
			32-pin	36-pin
ANI0	Input	10-bit A/D converter analog input 0	—	√
ANI1	Input	10-bit A/D converter analog input 1	√	√
ANI2	Input	10-bit A/D converter analog input 2	√	√
ANI3	Input	10-bit A/D converter analog input 3	√	√
ANI4	Input	10-bit A/D converter analog input 4	√	√
ANI5	Input	10-bit A/D converter analog input 5	√	√
ANI6	Input	10-bit A/D converter analog input 6	√	√
ANI7	Input	10-bit A/D converter analog input 7	√	√
ANI8	Input	10-bit A/D converter analog input 8	—	√
ANI9	Input	10-bit A/D converter analog input 9	√	√
AMP0P	Input	Operational amplifier 0 positive input	√	√
AMP1P	Input	Operational amplifier 1 positive input	√	√
AMP2P	Input	Operational amplifier 2 positive input	√	√
AMP0N	Input	Operational amplifier 0 negative input	√	√
AMP1N	Input	Operational amplifier 1 negative input	√	√
AMP2N	Input	Operational amplifier 2 negative input	—	√
AMP0O	Output	Operational amplifier 0 output	√	√
AMP1O	Output	Operational amplifier 1 output	√	√
AMP2O	Output	Operational amplifier 2 output	√	√
ANX0	I/O	General-purpose analog I/O port 0 for operational amplifiers 0, 1, 2	√	√
ANX1	I/O	General-purpose analog I/O port 1 for operational amplifiers 0, 1, 2	√	√
ANX2	I/O	General-purpose analog I/O port 2 for operational amplifiers 1, 2	√	√
ANX3	I/O	General-purpose analog I/O port 3 for operational amplifiers 1, 2	√	√
ANX4	I/O	General-purpose analog I/O port 4 for operational amplifiers 2	—	√
ANX5	I/O	General-purpose analog I/O port 5 for operational amplifiers 2	√	√
PGA0P	Input	PGA positive analog input 0	√	√
PGA1P	Input	PGA positive analog input 1	√	√
PGA2P	Input	PGA positive analog input 2	√	√
PGA3P	Input	PGA positive analog input 3	—	√
PGA0N	Input	PGA negative analog input 0	√	√
PGA1N	Input	PGA negative analog input 1	√	√
PGA2N	Input	PGA negative analog input 2	√	√
PGA3N	Input	PGA negative analog input 3	—	√
SBIAS	Output	Bias output for MEMS sensor	√	√
REGA	—	Pin for connecting capacitor for stabilizing regulator output for internal operation (analog power supply)	√	√
AV_{DD}	—	Analog positive power supply	√	√
AV_{SS}	—	Analog ground potential	√	√

2.3 Connection of Unused Pins

Table 2 - 3 shows the Connection of Unused Pins.

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

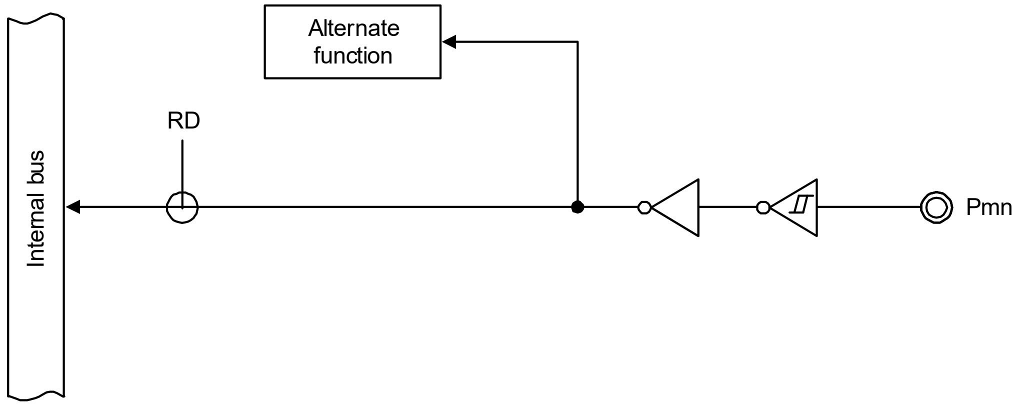
Table 2 - 3 Connection of Unused Pins

Pin Name	I/O	Recommended Connection of Unused Pins
P10 to P15	I/O	Input: Independently connect to VDD or VSS via a resistor. Output: Leave open.
P16/INTP7/ANI9/AMP2P/ANX5, P17/ANI8/AMP2N/ANX4		Set to analog I/O (PMCxx = 1) and leave open.
P40/TOOL0		Input: Independently connect to VDD or leave open. Output: Leave open.
P41/ANI6/AMP1P/ANX3, P42/ANI5/AMP1N/ANX2		Set to analog I/O (PMCxx = 1) and leave open.
P121, P122	Input	Independently connect to VDD or VSS via a resistor.
P137	Input	Independently connect to VDD or VSS via a resistor.
$\overline{\text{RESET}}$	Input	Directly connect to VDD or via a resistor.
REGC	—	Connect to VSS via a capacitor (0.47 to 1 µF).
ANI0	Input	Connect to AVSS.
ANI1/AMP0O, ANI2/AMP0N/ANX0, ANI3/AMP0P/ANX1, ANI4/AMP1O, ANI7/AMP2O	I/O	Leave open.
PGA0N to PGA3N	Input	Connect to AVSS.
PGA0P to PGA3P	Input	Connect to AVSS.
REGA	—	Connect to AVSS via a capacitor (0.22 µF).
SBIAS	Output	Connect to AVSS via a capacitor (0.22 µF).

2.4 Pin Block Diagrams

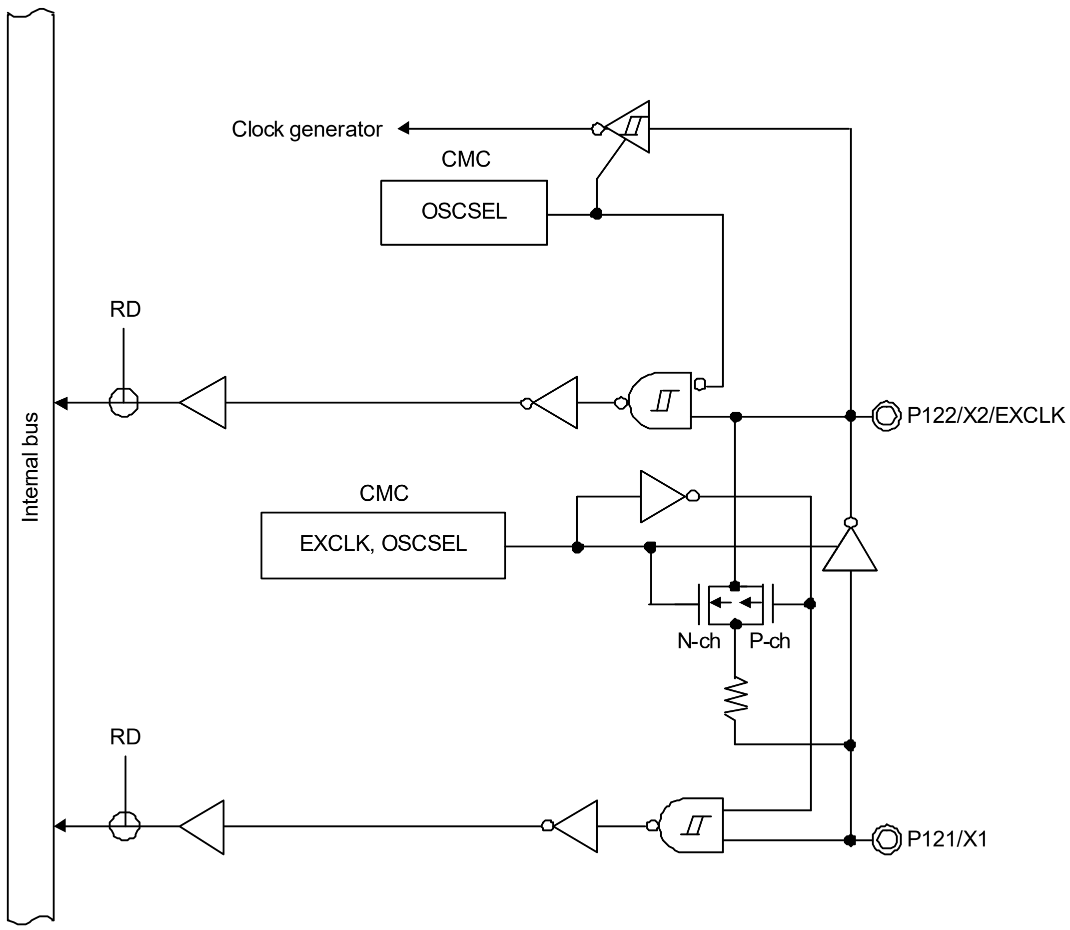
For the pin types listed in 2.1.1 32-pin products and 2.1.2 36-pin products, pin block diagrams are shown in Figures 2 - 1 to 2 - 6.

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



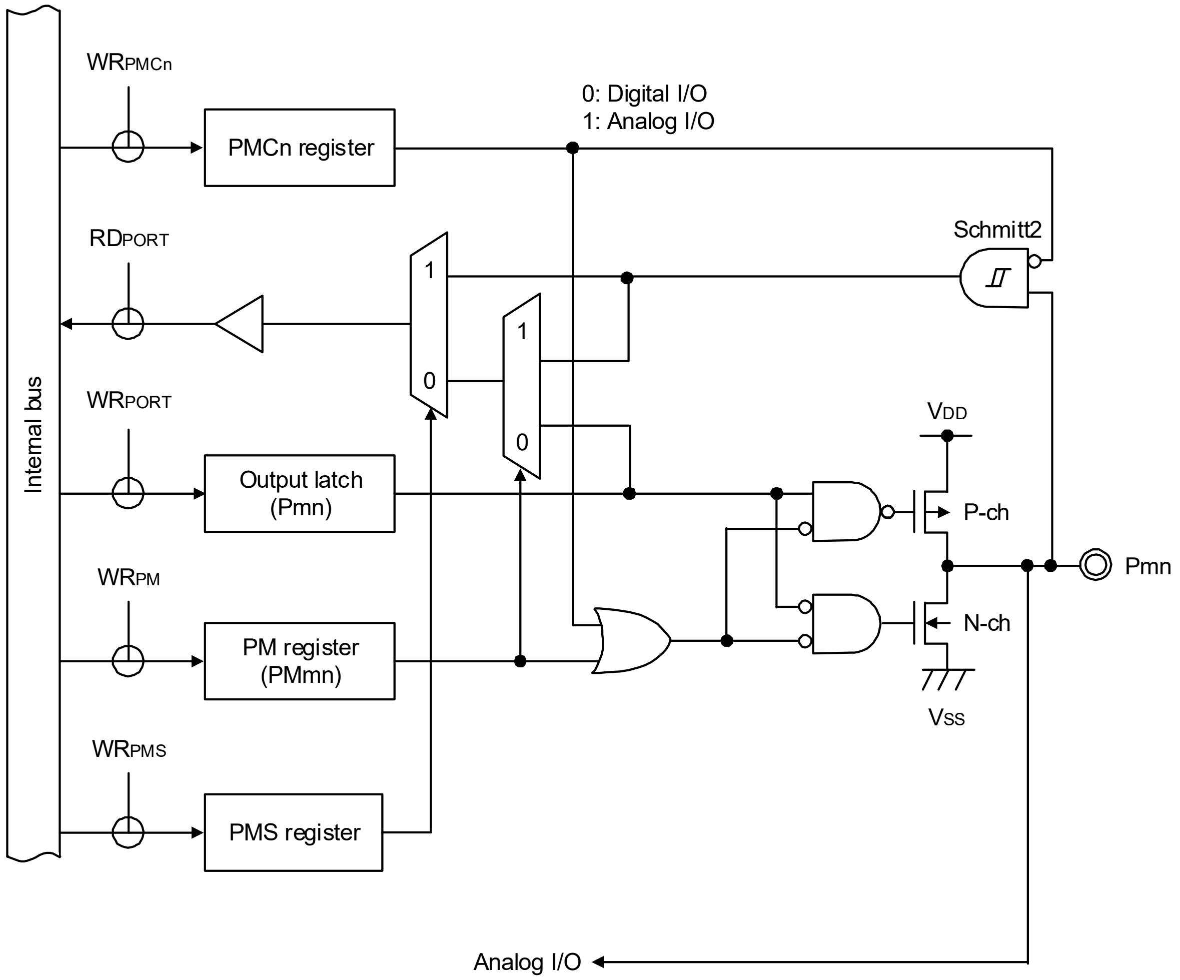
Remark Refer to **2.1 Port Functions** for alternate functions.

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



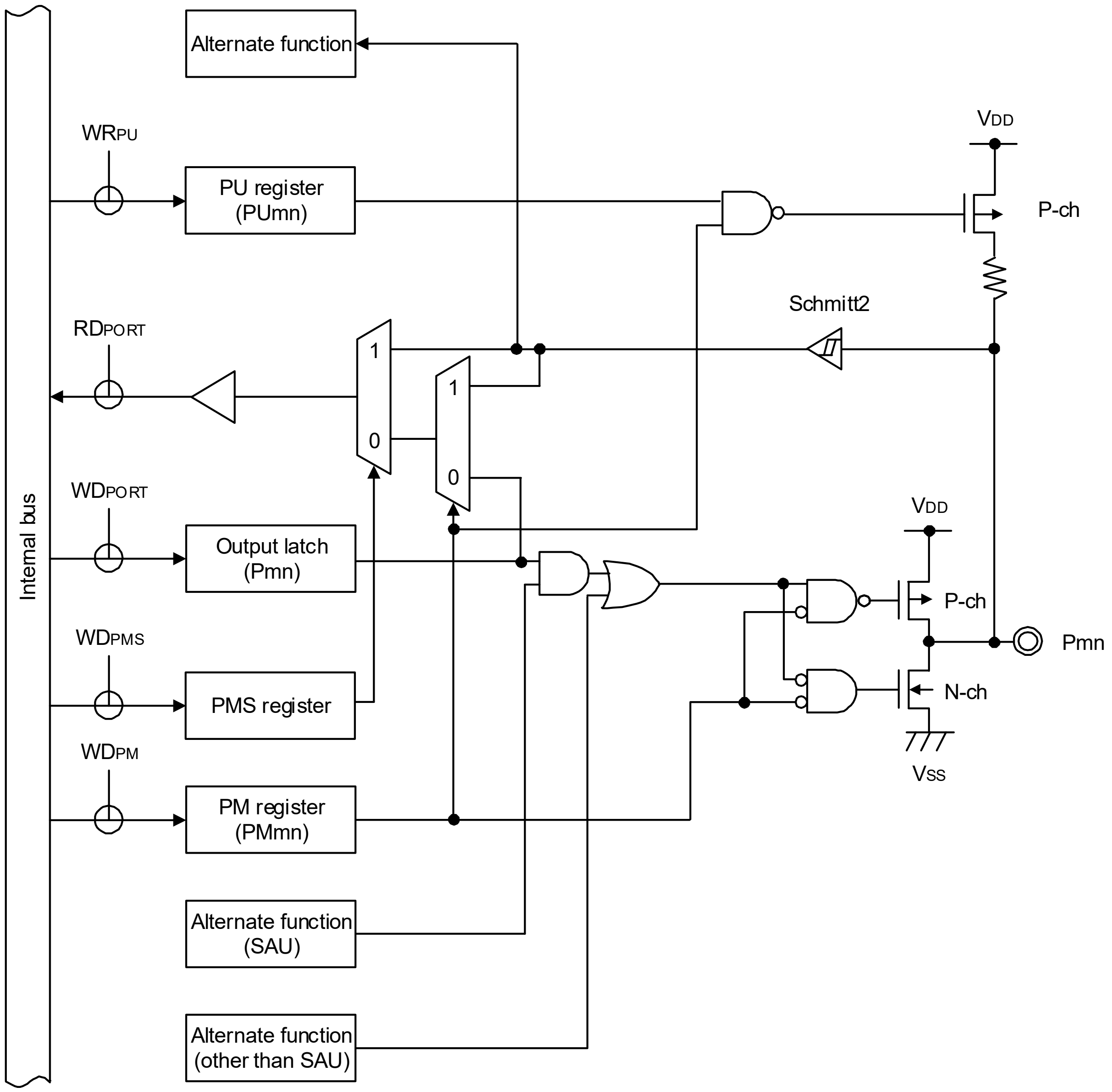
Remark Refer to **2.1 Port Functions** for alternate functions.

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



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

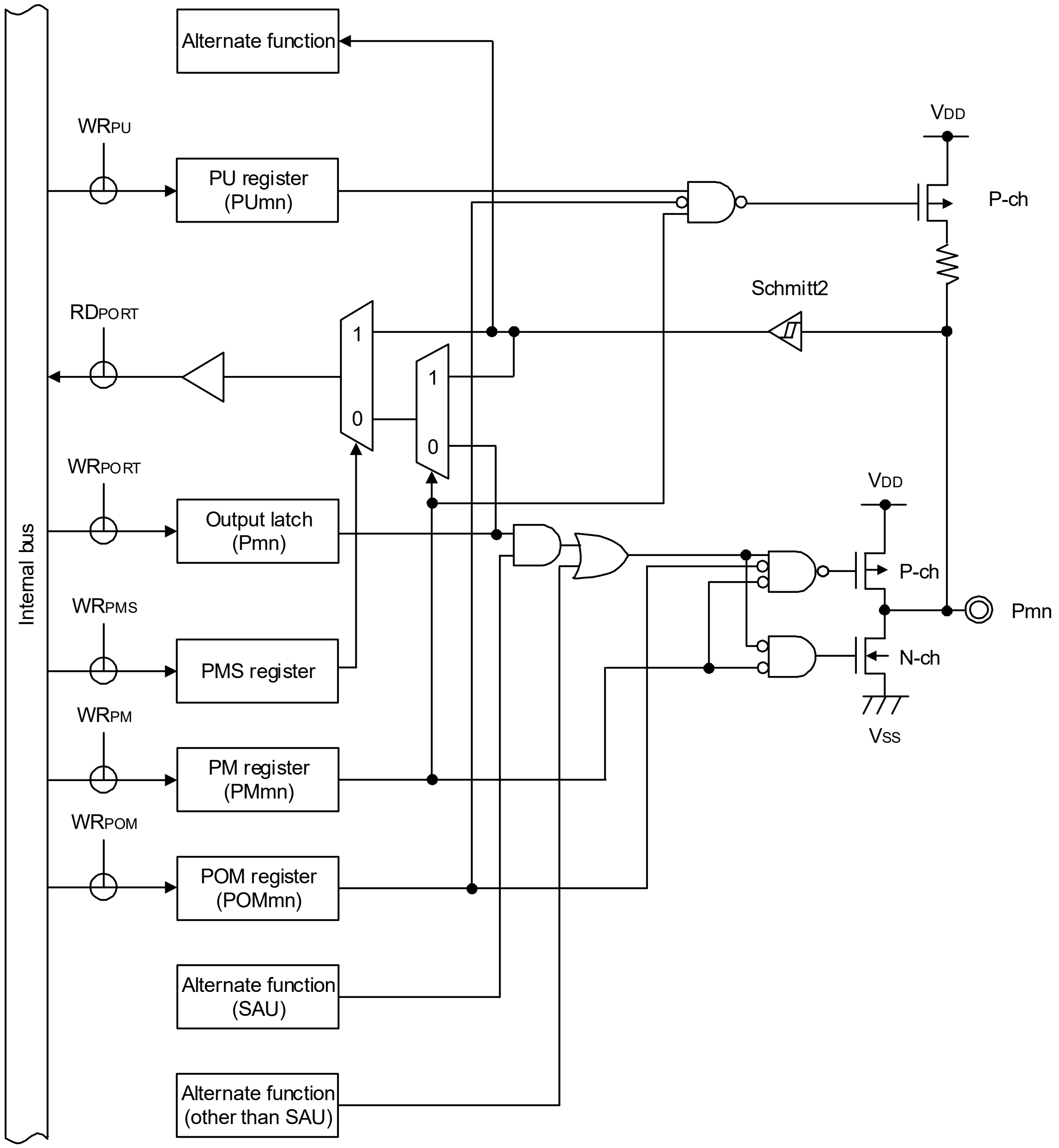
Figure 2 - 4 Pin Block Diagram of Pin Type 7-1-3



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

Remark 2. SAU: Serial array unit

Figure 2 - 5 Pin Block Diagram of Pin Type 7-1-4

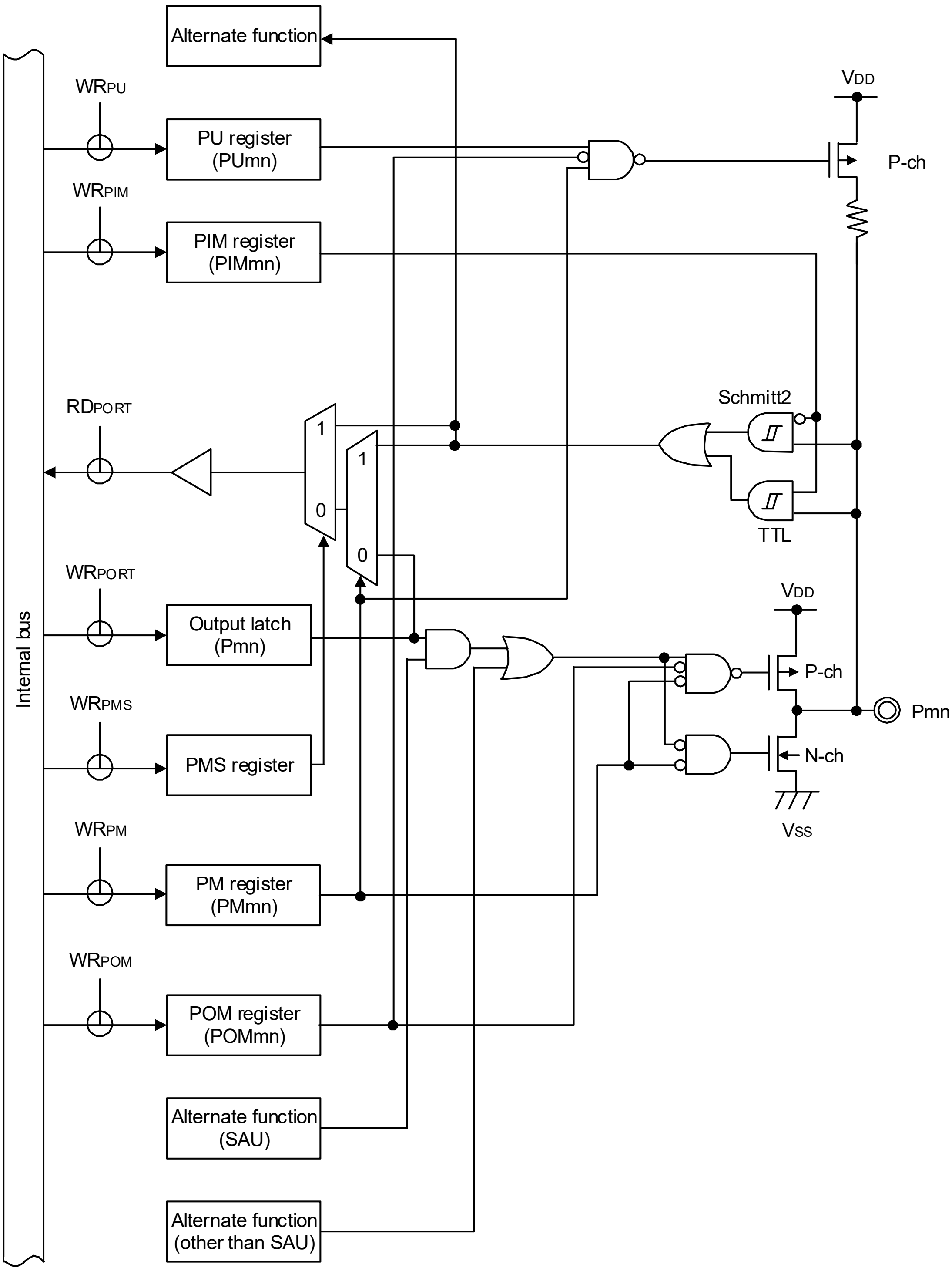


Caution **A through current may flow through if the pin is in the intermediate potential, because the input buffer is also turned on when the pin is in N-ch open-drain output mode by port output mode register (POMx).**

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

Remark 2. SAU: Serial array unit

Figure 2 - 6 Pin Block Diagram of Pin Type 8-1-4



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

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

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

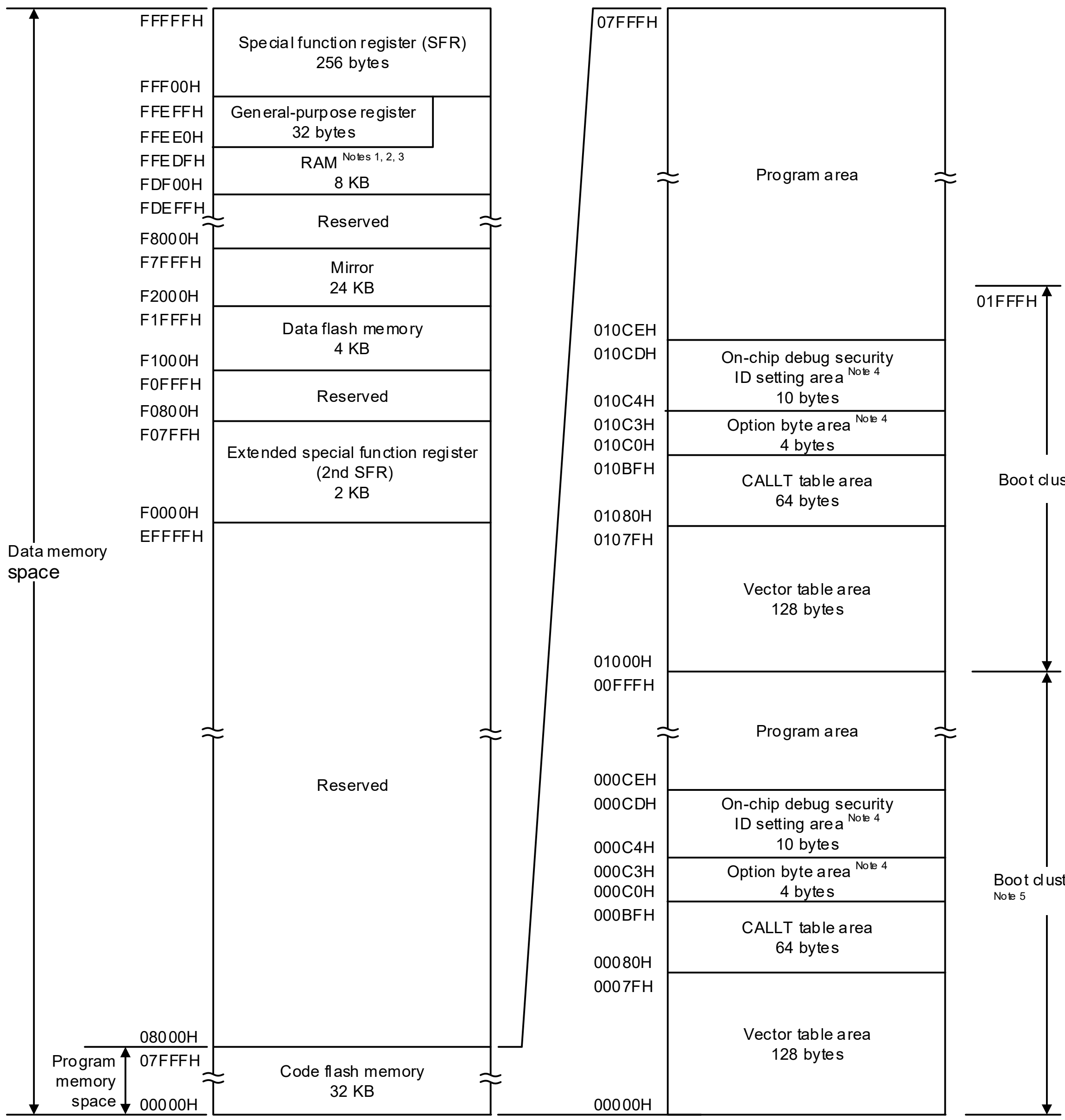
Remark 2. SAU: Serial array unit

CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

Products in the RL78/I1E can access a 1 MB address space. Figure 3 - 1 shows the memory maps.

Figure 3 - 1 Memory Map (R5F11CxC (x = B, C))



Note 1. Do not allocate the stack area, data buffers for use by the flash library, arguments of library functions, branch destinations in the processing of vectored interrupts, or destinations or sources for DTC transfer to the area from FFE20H to FFEDFH when performing self-programming or rewriting of the data flash memory.
The RAM area used by the flash library starts at FDF00H. For the RAM areas used by the flash library, see **Self RAM list of Flash Self-Programming Library for RL78 Family (R20UT2944)**.

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

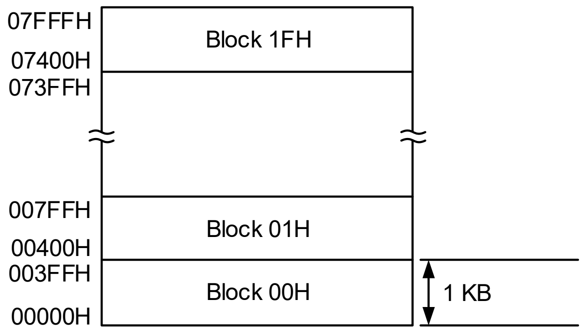
Note 3. The area from FE300H to FE6FFH must not be used by the user when using the on-chip debugging trace function.

Note 4. When boot swap is not used: Set the option bytes to 000C0H to 000C3H, and the on-chip debug security IDs to 000C4H to 000CDH.
When boot swap is used: Set the option bytes to 000C0H to 000C3H and 010C0H to 010C3H, and the on-chip debug security IDs to 000C4H to 000CDH and 010C4H to 010CDH.

Note 5. Writing boot cluster 0 can be prohibited depending on the setting of security (see **30.7 Security Settings**).

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

Remark The flash memory is divided into blocks (one block = 1 KB). For the address values and block numbers, see **Tables 3 - 1**.



(R5F11CxC (x = B, C))

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

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

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

Remark R5F11CxC (x = B, C): Block numbers 00H to 1FH

3.1.1 Internal program memory space

The internal program memory space stores the program and table data.
The RL78/I1E products incorporate internal ROM (flash memory), as shown below.

Table 3 - 2 Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
R5F11CxC (x = B, C)	Flash memory	32768 × 8 bits (00000H to 07FFFH)

The internal program memory space is divided into the following areas.

(1) Vector table area

The 128-byte area 00000H to 0007FH is reserved as a vector table area. The program start addresses for branch upon reset or generation of each interrupt request are stored in the vector table area. Furthermore, the interrupt jump address is a 64 K address of 00000H to 0FFFFH, because the vector code is assumed to be 2 bytes.

Of the 16-bit address, the lower 8 bits are stored at even addresses and the higher 8 bits are stored at odd addresses.

To use the boot swap function, set a vector table also at 01000H to 0107FH.

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

Table 3 - 3 Vector Table

Vector Table Address	Interrupt Source	32-pin	36-pin
0000H	RESET, POR, LVD, WDT, TRAP, IAW, RPE	√	√
0004H	INTWDTI	√	√
0006H	INTLVI	√	√
0008H	INTP0	√	√
000AH	INTP1	√	√
000CH	INTP2	√	√
000EH	INTP3	√	√
0010H	INTP4	√	√
0012H	INTP5	√	√
001EH	INTST0/INTCSI00/INTIIC00	√	√
0020H	INTSR0/INTCSI01/INTIIC01	√	√
0022H	INTSRE0	√	√
	INTTM01H	√	√
0024H	INTST1	√	√
0026H	INTSR1	√	√
0028H	INTSRE1	√	√
	INTTM03H	√	√
002CH	INTTM00	√	√
002EH	INTTM01	√	√
0030H	INTTM02	√	√
0032H	INTTM03	√	√
0034H	INTAD	√	√
0036H	INTRTC	√	√
0038H	INTIT	√	√
0040H	INTTRJ0	√	√
0042H	INTTM10	√	√
0044H	INTTM11	√	√
004AH	INTP6	√	√
004CH	INTP7	—	√
004EH	INTDSAD	√	√
0050H	INTDSADS	√	√
005AH	INTTRG	√	√
0062H	INTFL	√	√
007EH	BRK	√	√

(2) CALLT instruction table area
The 64-byte area 00080H to 000BFH can store the subroutine entry address of a 2-byte call instruction (CALLT). Set the subroutine entry address to a value in a range of 00000H to 0FFFFH (because an address code is 2 bytes).
To use the boot swap function, set a CALLT instruction table also at 01080H to 010BFH.

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

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

3.1.2 Mirror area

The RL78/I1E mirrors the code flash area of 00000H to 0FFFFH, to F0000H to FFFFFH.

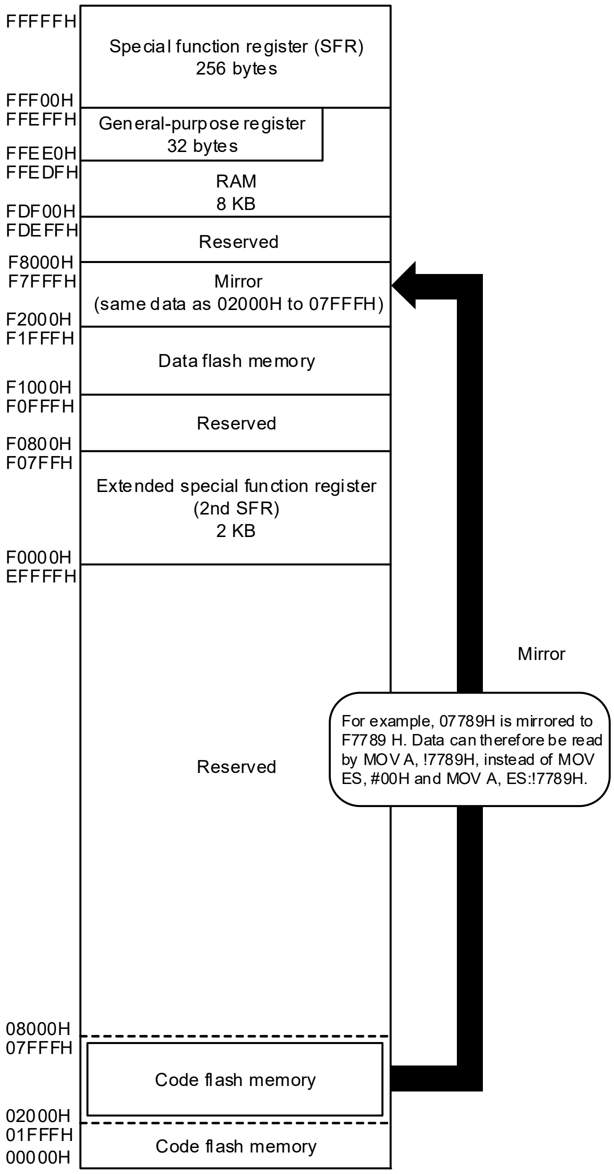
By reading data from F0000H to FFFFFH, an instruction that does not have the ES register as an operand can be used, and thus the contents of the code flash can be read with the shorter code. However, the code flash area is not mirrored to the special function register (SFR), extended special function register (2nd SFR), RAM, data flash memory, and use prohibited areas.

See **3.1 Memory Space** for the mirror area of each product.

The mirror area can only be read and no instruction can be fetched from this area.

The following show examples.

Example R5F11CxC (x = B, C)



The PMC register is described below.

- Processor mode control register (PMC)
  This register sets the flash memory space for mirroring to area from F0000H to FFFFFH.
  The PMC register can be set by a 1-bit or 8-bit memory manipulation instruction.
  Reset signal generation sets this register to 00H.

Figure 3 - 2 Format of Configuration of Processor mode control register (PMC)

Address: FFFFEH After reset: 00H R/W

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

MAA	Selection of flash memory space for mirroring to area from F0000H to FFFFFH
0	00000H to 07FFFH is mirrored to F0000H to F7FFFH
1	Setting prohibited

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

3.1.3 Internal data memory space

The RL78/I1E products incorporate the following RAMs.

Table 3 - 4 Internal RAM Capacity

Part Number	Internal RAM
R5F11CxC (x = B, C)	8192 × 8 bits (FDF00H to FFEFFH)

The internal RAM can be used as a data area and a program area where instructions are fetched (it is prohibited to use the general-purpose register area for fetching instructions). Four general-purpose register banks consisting of eight 8-bit registers per bank are assigned to the 32-byte area of FFEE0H to FFEFFH of the internal RAM area.

The internal RAM is used as stack memory.

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

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

Caution 3. Use of FDF00H to FE309H in the internal RAM areas is prohibited when performing self-programming and rewriting the data flash memory, because these areas are used by libraries.

Caution 4. FE300H to FE6FFH in the internal RAM area cannot be used as stack memory when using the on-chip debugging trace function.

3.1.4 Special function register (SFR) area

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

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

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

On-chip peripheral hardware special function registers (2nd SFRs) are allocated in the area F0000H to F07FFH (see **Tables 3 - 8** to **3 - 14** in **3.2.5 Extended special function registers (2nd SFRs: 2nd Special Function Registers)**).

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

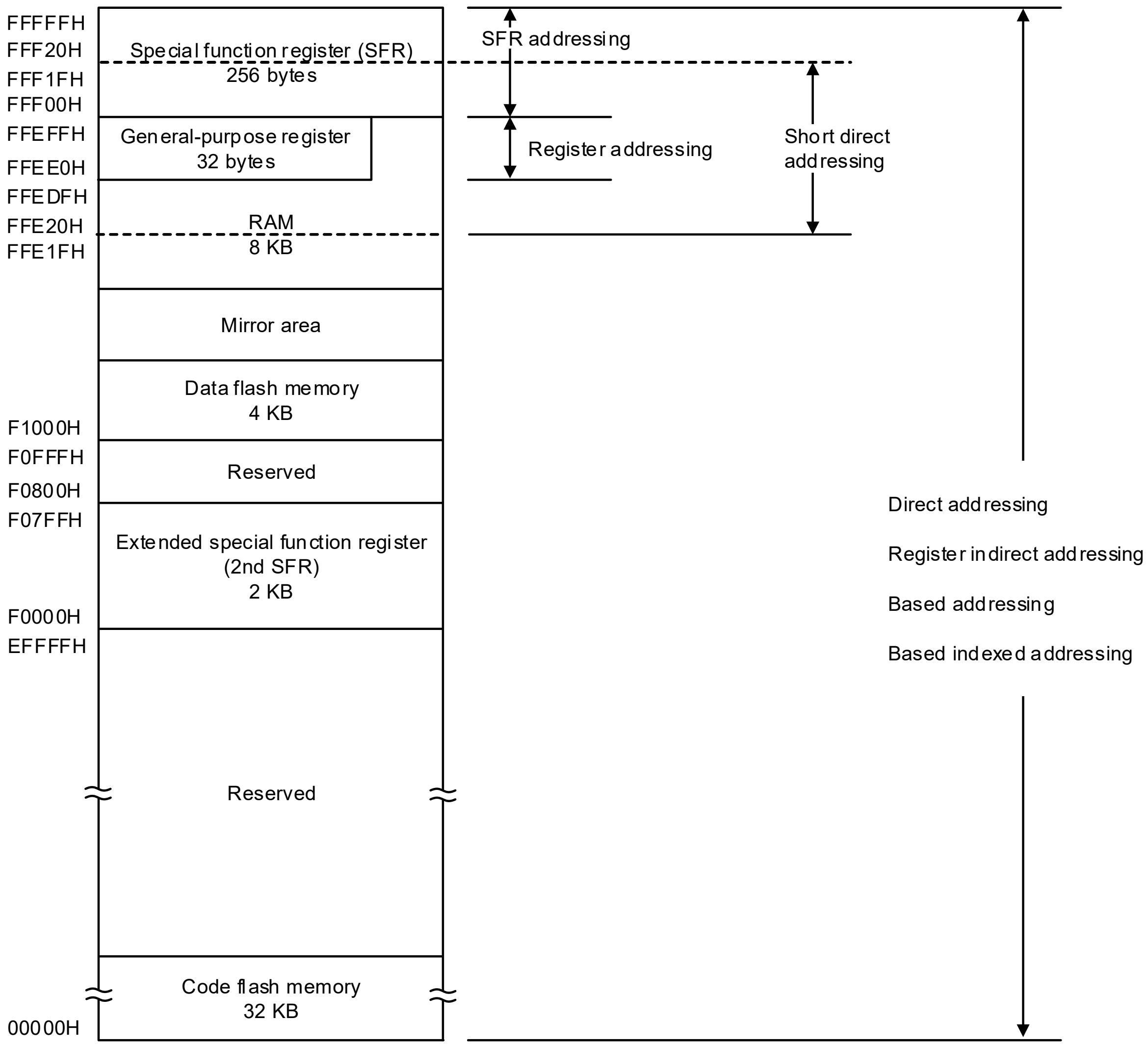
Caution 2. When accessing timer RJ counter register 0 (TRJ0) allocated in F0500H of the extended SFR (2nd SFR), the CPU does not proceed to the next instruction processing but enters the wait state for CPU processing. For this reason, if this wait state occurs, the number of instruction execution clocks is increased by the number of wait clocks. The number of wait clocks for access to timer RJ counter register 0 (TRJ0) is one clock for both writing and reading.

3.1.6 Data memory addressing

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

Several addressing modes are provided for addressing the memory relevant to the execution of instructions for the RL78/I1E, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of the special function registers (SFR) and general-purpose registers are available for use. Figure 3 - 3 shows correspondence between data memory and addressing. For details of each addressing.

Figure 3 - 3 Correspondence Between Data Memory and Addressing



3.2 Processor Registers

The RL78/I1E products incorporate the following processor registers.

3.2.1 Control registers

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

(1) Program counter (PC)

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

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

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

Figure 3 - 4 Format of Program Counter

	19 … 0
PC	

(2) Program status word (PSW)

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

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

Figure 3 - 5 Format of Program Status Word

	7							0
PSW	IE	Z	RBS1	AC	RBS0	ISP1	ISP0	CY

(a) Interrupt enable flag (IE)

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

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

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

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

(b) Zero flag (Z)
When the operation result is zero 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)
This flag manages the priority of acknowledgeable maskable vectored interrupts. Vectored interrupt requests specified lower than the value of ISP0 and ISP1 flags by the priority specification flag registers (PRn0L, PRn0H, PRn1L, PRn1H, PRn2L, PRn2H) (see **22.3.3**) cannot be acknowledged. Actual vectored interrupt requests acknowledgment is controlled by the interrupt enable flag (IE).

Remark n = 0, 1

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

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

Figure 3 - 6 Format of Stack Pointer

	15															0
SP	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SP7	SP6	SP5	SP4	SP3	SP2	SP1	0

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

Caution 1. Since reset signal generation makes the SP contents undefined, be sure to initialize the SP before using the stack.

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

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

Caution 4. Use of FDF00H to FE309H in the internal RAM areas is prohibited when performing self-programming and rewriting the data flash memory, because these areas are used by libraries.

Caution 5. FE300H to FE6FFH in the internal RAM area cannot be used as stack memory when using the on-chip debugging trace function.

3.2.2 General-purpose registers

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

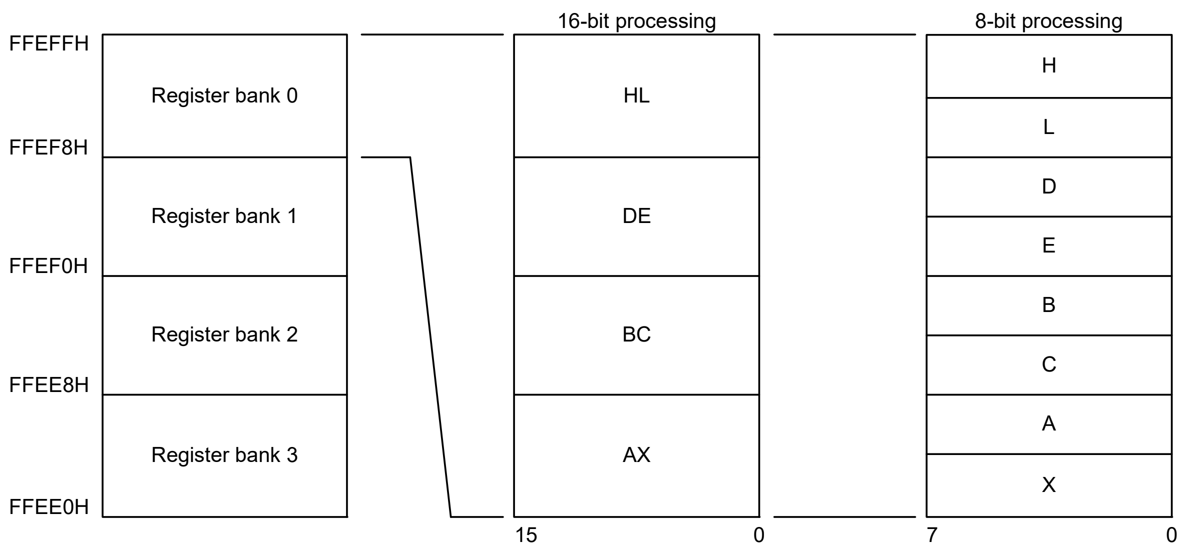
Each register can be used as an 8-bit register, and two 8-bit registers can also be used in a pair as a 16-bit register (AX, BC, DE, and HL).

Register banks to be used for instruction execution are set by the CPU control instruction (SEL RBn). Because of the 4-register bank configuration, an efficient program can be created by switching between a register for normal processing and a register for interrupt processing for each bank.

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

Figure 3 - 7 Configuration of General-Purpose Registers

(a) Function name



3.2.3 ES and CS registers

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

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

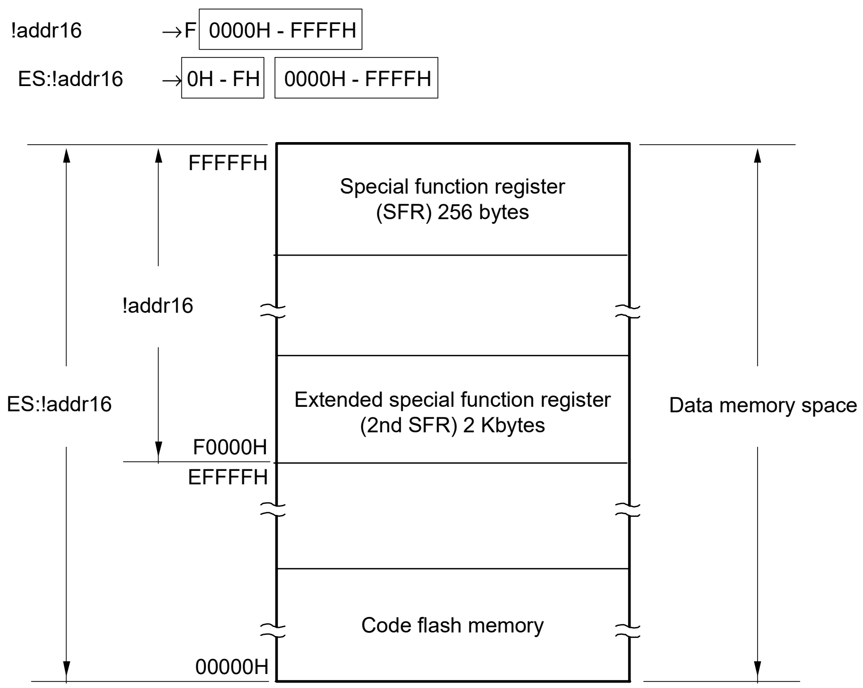
Figure 3 - 8 Configuration of ES and CS Registers

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

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

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

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



3.2.4 Special function registers (SFRs)

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

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

Tables 3 - 5 to 3 - 7 give lists of the SFRs. The meanings of items in the table are as follows.

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

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

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

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF01H	Port register 1	P1		R/W	√	√	—	00H
FFF04H	Port register 4	P4		R/W	√	√	—	00H
FFF0CH	Port register 12	P12		R/W	√	√	—	Undefined
FFF0DH	Port register 13	P13		R/W	√	√	—	Undefined
FFF10H	Serial data register 00	TXD0/ SIO00	SDR00	R/W	—	√	√	0000H
FFF11H		—			—	—		
FFF12H	Serial data register 01	RXD0/ SIO01	SDR01	R/W	—	√	√	0000H
FFF13H		—			—	—		
FFF18H	Timer data register 00	TDR00		R/W	—	—	√	0000H
FFF19H								
FFF1AH	Timer data register 01	TDR01L	TDR01	R/W	—	√	√	00H
FFF1BH		TDR01H			—	√		00H
FFF1EH	10-bit A/D conversion result register	ADCR		R	—	—	√	0000H
FFF1FH	8-bit A/D conversion result register	ADCRH		R	—	√	—	00H
FFF21H	Port mode register 1	PM1		R/W	√	√	—	FFH
FFF24H	Port mode register 4	PM4		R/W	√	√	—	FFH
FFF30H	A/D converter mode register 0	ADM0		R/W	√	√	—	00H
FFF31H	Analog input channel specification register	ADS		R/W	√	√	—	00H
FFF32H	A/D converter mode register 1	ADM1		R/W	√	√	—	00H
FFF38H	External interrupt rising edge enable register 0	EGP0		R/W	√	√	—	00H
FFF39H	External interrupt falling edge enable register 0	EGN0		R/W	√	√	—	00H
FFF44H	Serial data register 02	TXD1	SDR02	R/W	—	√	√	0000H
FFF45H		—			—	—		
FFF46H	Serial data register 03	RXD1	SDR03	R/W	—	√	√	0000H
FFF47H		—			—	—		
FFF54H	16-bit watch error correction register	SUBCUDW		R/W	—	—	√	0000H
FFF55H								
FFF60H	Timer RD general register C	TRGGRC		R/W	—	—	√	FFFFH
FFF61H								
FFF62H	Timer RD general register D	TRGGRD		R/W	—	—	√	FFFFH
FFF63H								
FFF64H	Timer data register 02	TDR02		R/W	—	—	√	0000H
FFF65H								
FFF66H	Timer data register 03	TDR03L	TDR03	R/W	—	√	√	00H
FFF67H		TDR03H			—	√		00H
FFF70H	Timer data register 10	TDR10		R/W	—	—	√	0000H
FFF71H								

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF72H	Timer data register 11	TDR11L	TDR11	R/W	—	√	√	00H
FFF73H		TDR11H			—	√		00H
FFF90H	Interval timer control register	ITMC		R/W	—	—	√	7FFFH
FFF91H								
FFF92H	Second count register	SEC		R/W	—	√	—	00H
FFF93H	Minute count register	MIN		R/W	—	√	—	00H
FFF94H	Hour count register	HOUR		R/W	—	√	—	12H Note 1
FFF95H	Week count register	WEEK		R/W	—	√	—	00H
FFF96H	Day count register	DAY		R/W	—	√	—	01H
FFF97H	Month count register	MONTH		R/W	—	√	—	01H
FFF98H	Year count register	YEAR		R/W	—	√	—	00H
FFF99H	Watch error correction register	SUBCUD		R/W	—	√	—	00H
FFF9AH	Alarm minute register	ALARMWM		R/W	—	√	—	00H
FFF9BH	Alarm hour register	ALARMWH		R/W	—	√	—	12H
FFF9CH	Alarm week register	ALARMWW		R/W	—	√	—	00H
FFF9DH	Real-time clock control register 0	RTCC0		R/W	√	√	—	00H
FFF9EH	Real-time clock control register 1	RTCC1		R/W	√	√	—	00H
FFFA0H	Clock operation mode control register	CMC		R/W	—	√	—	00H
FFFA1H	Clock operation status control register	CSC		R/W	√	√	—	C0H
FFFA2H	Oscillation stabilization time counter status register	OSTC		R	√	√	—	00H
FFFA3H	Oscillation stabilization time select register	OSTS		R/W	—	√	—	07H
FFFA4H	System clock control register	CKC		R/W	√	√	—	00H
FFFA5H	Clock output select register 0	CKS0		R/W	√	√	—	00H
FFFA8H	Reset control flag register	RESF		R	—	√	—	Undefined Note 2
FFFA9H	Voltage detection register	LVIM		R/W	√	√	—	00H Note 2
FFFAAH	Voltage detection level register	LVIS		R/W	√	√	—	00H/01H/81H Note 2
FFFABH	Watchdog timer enable register	WDTE		R/W	—	√	—	9AH/1AH Note 3
FFFACH	CRC input register	CRCIN		R/W	—	√	—	00H
FFFD0H	Interrupt request flag register 2L	IF2L	IF2	R/W	√	√	√	00H
FFFD1H	Interrupt request flag register 2H	IF2H		R/W	√	√		00H
FFFD4H	Interrupt mask flag register 2L	MK2L	MK2	R/W	√	√	√	FFH
FFFD5H	Interrupt mask flag register 2H	MK2H		R/W	√	√		FFH
FFFD8H	Priority specification flag register 02L	PR02L	PR02	R/W	√	√	√	FFH
FFFD9H	Priority specification flag register 02H	PR02H		R/W	√	√		FFH

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulatable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFFDCH	Priority specification flag register 12L	PR12L	PR12	R/W	√	√	√	FFH
FFFDDH	Priority specification flag register 12H	PR12H		R/W	√	√		FFH
FFFE0H	Interrupt request flag register 0L	IF0L	IF0	R/W	√	√	√	00H
FFFE1H	Interrupt request flag register 0H	IF0H		R/W	√	√		00H
FFFE2H	Interrupt request flag register 1L	IF1L	IF1	R/W	√	√	√	00H
FFFE3H	Interrupt request flag register 1H	IF1H		R/W	√	√		00H
FFFE4H	Interrupt mask flag register 0L	MK0L	MK0	R/W	√	√	√	FFH
FFFE5H	Interrupt mask flag register 0H	MK0H		R/W	√	√		FFH
FFFE6H	Interrupt mask flag register 1L	MK1L	MK1	R/W	√	√	√	FFH
FFFE7H	Interrupt mask flag register 1H	MK1H		R/W	√	√		FFH
FFFE8H	Priority specification flag register 00L	PR00L	PR00	R/W	√	√	√	FFH
FFFE9H	Priority specification flag register 00H	PR00H		R/W	√	√		FFH
FFFEAH	Priority specification flag register 01L	PR01L	PR01	R/W	√	√	√	FFH
FFFEBH	Priority specification flag register 01H	PR01H		R/W	√	√		FFH
FFFECH	Priority specification flag register 10L	PR10L	PR10	R/W	√	√	√	FFH
FFFEDH	Priority specification flag register 10H	PR10H		R/W	√	√		FFH
FFFEEH	Priority specification flag register 11L	PR11L	PR11	R/W	√	√	√	FFH
FFFEFH	Priority specification flag register 11H	PR11H		R/W	√	√		FFH
FFFF0H	Multiply and accumulation register (L)	MACRL		R/W	—	—	√	0000H
FFFF1H								
FFFF2H	Multiply and accumulation register (H)	MACRH		R/W	—	—	√	0000H
FFFF3H								
FFFFEH	Processor mode control register	PMC		R/W	√	√	—	00H

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

Note 2. These values vary depending on the reset source.

<table>
<tr><th colspan="2">Reset Source
Register</th><th>RESET Input</th><th>Reset by POR</th><th>Reset by Execution of Illegal Instruction</th><th>Reset by WDT</th><th>Reset by RAM parity error</th><th>Reset by illegal-memory access</th><th>Reset by LVD</th></tr>
<tr><td rowspan="5">RESF</td><td>TRAP</td><td colspan="2" rowspan="5">Cleared (0)</td><td>Set (1)</td><td colspan="3">Held</td><td rowspan="4">Held</td></tr>
<tr><td>WDTRF</td><td>Held</td><td>Set (1)</td><td colspan="2">Held</td></tr>
<tr><td>RPERF</td><td colspan="2">Held</td><td>Set (1)</td><td>Held</td></tr>
<tr><td>IAWRF</td><td colspan="3">Held</td><td>Set (1)</td></tr>
<tr><td>LVIRF</td><td colspan="4">Held</td><td>Set (1)</td></tr>
<tr><td rowspan="3">LVIM</td><td>LVISEN</td><td colspan="6">Cleared (0)</td><td rowspan="4">Held</td></tr>
<tr><td>LVIOMSK</td><td colspan="6" rowspan="2">Held</td></tr>
<tr><td>LVIF</td></tr>
<tr><td colspan="2">LVIS</td><td colspan="6">Cleared (00H/01H/81H)</td></tr>
</table>

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 **Tables 3 - 8** to **3 - 13 Extended SFR (2nd SFR) List**.

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

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

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

Tables 3 - 8 to 3 - 14 give lists of the extended SFRs. The meanings of items in the table are as follows.

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

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

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

Table 3 - 8 Extended Special Function Register (2nd SFR) List (1/7)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulatable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0010H	A/D converter mode register 2	ADM2	R/W	√	√	—	00H
F0011H	Conversion result comparison upper limit setting register	ADUL	R/W	—	√	—	FFH
F0012H	Conversion result comparison lower limit setting register	ADLL	R/W	—	√	—	00H
F0013H	A/D test register	ADTES	R/W	—	√	—	00H
F0031H	Pull-up resistor option register 1	PU1	R/W	√	√	—	00H
F0034H	Pull-up resistor option register 4	PU4	R/W	√	√	—	01H
F0041H	Port input mode register 1	PIM1	R/W	√	√	—	00H
F0051H	Port output mode register 1	POM1	R/W	√	√	—	00H
F0061H	Port mode control register 1	PMC1	R/W	√	√	—	FFH
F0064H	Port mode control register 4	PMC4	R/W	√	√	—	FFH
F0070H	Noise filter enable register 0	NFEN0	R/W	√	√	—	00H
F0071H	Noise filter enable register 1	NFEN1	R/W	√	√	—	00H
F0072H	Noise filter enable register 2	NFEN2	R/W	√	√	—	00H
F0073H	Input switch control register	ISC	R/W	√	√	—	00H
F0074H	Timer input select register 0	TIS0	R/W	—	√	—	00H
F0078H	Invalid memory access detection control register	IAWCTL	R/W	—	√	—	00H
F007AH	Peripheral enable register 1	PER1	R/W	√	√	—	00H
F007BH	Port mode select register	PMS	R/W	√	√	—	00H
F0090H	Data flash control register	DFLCTL	R/W	√	√	—	00H
F00A8H	High-speed on-chip oscillator frequency select register	HOCODIV	R/W	—	√	—	Undefined Note Note
F00ACH	Configurable amplifier 0 trimming code register	AMP0TRM	R	—	√	—	00H
F00ADH	Configurable amplifier 1 trimming code register	AMP1TRM	R	—	√	—	00H
F00AEH	Configurable amplifier 2 trimming code register	AMP2TRM	R	—	√	—	00H
F00F0H	Peripheral enable register 0	PER0	R/W	√	√	—	00H
F00F3H	Subsystem clock supply mode control register	OSMC	R/W	—	√	—	00H
F00F5H	RAM parity error control register	RPECTL	R/W	√	√	—	00H
F00FEH	BCD correction result register	BCDADJ	R	—	√	—	Undefined

Note The value specified by using FRQSEL2 to FRQSEL0 of the option byte 000C2H is set.

Table 3 - 9 Extended Special Function Register (2nd SFR) List (2/7)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulatable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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								
F0116H	Serial mode register 03	SMR03		R/W	—	—	√	0020H
F0117H								
F0118H	Serial communication operation setting register 00	SCR00		R/W	—	—	√	0087H
F0119H								
F011AH	Serial communication operation setting register 01	SCR01		R/W	—	—	√	0087H
F011BH								
F011CH	Serial communication operation setting register 02	SCR02		R/W	—	—	√	0087H
F011DH								
F011EH	Serial communication operation setting register 03	SCR03		R/W	—	—	√	0087H
F011FH								
F0120H	Serial channel enable status register 0	SE0L	SE0	R	√	√	√	0000H
F0121H		—			—	—		
F0122H	Serial channel start register 0	SS0L	SS0	R/W	√	√	√	0000H
F0123H		—			—	—		
F0124H	Serial channel stop register 0	ST0L	ST0	R/W	√	√	√	0000H
F0125H		—			—	—		
F0126H	Serial clock select register 0	SPS0L	SPS0	R/W	—	√	√	0000H
F0127H		—			—	—		

Table 3 - 10 Extended Special Function Register (2nd SFR) List (3/7)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulatable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0128H	Serial output register 0	SO0		R/W	—	—	√	0F0FH
F0129H								
F012AH	Serial output enable register 0	SOE0L	SOE0	R/W	√	√	√	0000H
F012BH		—			—	—		
F0134H	Serial output level register 0	SOL0L	SOL0	R/W	—	√	√	0000H
F0135H		—			—	—		
F0138H	Serial standby control register 0	SSC0L	SSC0	R/W	—	√	√	0000H
F0139H		—			—	—		
F0180H	Timer counter register 00	TCR00		R	—	—	√	FFFFH
F0181H								
F0182H	Timer counter register 01	TCR01		R	—	—	√	FFFFH
F0183H								
F0184H	Timer counter register 02	TCR02		R	—	—	√	FFFFH
F0185H								
F0186H	Timer counter register 03	TCR03		R	—	—	√	FFFFH
F0187H								
F0190H	Timer mode register 00	TMR00		R/W	—	—	√	0000H
F0191H								
F0192H	Timer mode register 01	TMR01		R/W	—	—	√	0000H
F0193H								
F0194H	Timer mode register 02	TMR02		R/W	—	—	√	0000H
F0195H								
F0196H	Timer mode register 03	TMR03		R/W	—	—	√	0000H
F0197H								
F01A0H	Timer status register 00	TSR00L	TSR00	R	—	√	√	0000H
F01A1H		—			—	—		
F01A2H	Timer status register 01	TSR01L	TSR01	R	—	√	√	0000H
F01A3H		—			—	—		
F01A4H	Timer status register 02	TSR02L	TSR02	R	—	√	√	0000H
F01A5H		—			—	—		
F01A6H	Timer status register 03	TSR03L	TSR03	R	—	√	√	0000H
F01A7H		—			—	—		
F01B0H	Timer channel enable status register 0	TE0L	TE0	R	√	√	√	0000H
F01B1H		—			—	—		
F01B2H	Timer channel start register 0	TS0L	TS0	R/W	√	√	√	0000H
F01B3H		—			—	—		
F01B4H	Timer channel stop register 0	TT0L	TT0	R/W	√	√	√	0000H
F01B5H		—			—	—		
F01B6H	Timer clock select register 0	TPS0		R/W	—	—	√	0000H
F01B7H								

Table 3 - 11 Extended Special Function Register (2nd SFR) List (4/7)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulatable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F01B8H	Timer output register 0	TO0L	TO0	R/W	—	√	√	0000H
F01B9H		—			—	—		
F01BAH	Timer output enable register 0	TOE0L	TOE0	R/W	√	√	√	0000H
F01BBH		—			—	—		
F01BCH	Timer output level register 0	TOL0L	TOL0	R/W	—	√	√	0000H
F01BDH		—			—	—		
F01BEH	Timer output mode register 0	TOM0L	TOM0	R/W	—	√	√	0000H
F01BFH		—			—	—		
F01C0H	Timer counter register 10	TCR10		R	—	—	√	FFFFH
F01C1H								
F01C2H	Timer counter register 11	TCR11		R	—	—	√	FFFFH
F01C3H								
F01D0H	Timer mode register 10	TMR10		R/W	—	—	√	0000H
F01D1H								
F01D2H	Timer mode register 11	TMR11		R/W	—	—	√	0000H
F01D3H								
F01E0H	Timer status register 10	TSR10L	TSR10	R	—	√	√	0000H
F01E1H		—			—	—		
F01E2H	Timer status register 11	TSR11L	TSR11	R	—	√	√	0000H
F01E3H		—			—	—		
F01F0H	Timer channel enable status register 1	TE1L	TE1	R	√	√	√	0000H
F01F1H		—			—	—		
F01F2H	Timer channel start register 1	TS1L	TS1	R/W	√	√	√	0000H
F01F3H		—			—	—		
F01F4H	Timer channel stop register 1	TT1L	TT1	R/W	√	√	√	0000H
F01F5H		—			—	—		
F01F6H	Timer clock select register 1	TPS1		R/W	—	—	√	0000H
F01F7H								
F01F8H	Timer output register 1	TO1L	TO1	R/W	—	√	√	0000H
F01F9H		—			—	—		
F01FAH	Timer output enable register 1	TOE1L	TOE1	R/W	√	√	√	0000H
F01FBH		—			—	—		
F01FCH	Timer output level register 1	TOL1L	TOL1	R/W	—	√	√	0000H
F01FDH		—			—	—		
F01FEH	Timer output mode register 1	TOM1L	TOM1	R/W	—	√	√	0000H
F01FFH		—			—	—		

Table 3 - 12 Extended Special Function Register (2nd SFR) List (5/7)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulatable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0240H	Timer RJ control register 0	TRJCR0	R/W	—	√	—	00H
F0241H	Timer RJ I/O control register 0	TRJIOC0	R/W	√	√	—	00H
F0242H	Timer RJ mode register 0	TRJMR0	R/W	√	√	—	00H
F0243H	Timer RJ event pin select register 0	TRJISR0	R/W	√	√	—	00H
F0250H	Timer RG mode register	TRGMR	R/W	√	√	—	00H
F0251H	Timer RG count control register	TRGCNTC	R/W	√	√	—	00H
F0252H	Timer RG control register	TRGCR	R/W	√	√	—	00H
F0253H	Timer RG interrupt enable register	TRGIER	R/W	√	√	—	00H
F0254H	Timer RG status register	TRGSR	R/W	√	√	—	00H
F0255H	Timer RG I/O control register	TRGIOR	R/W	√	√	—	00H
F0256H	Timer RG counter	TRG	RW	—	—	√	0000H
F0257H							
F0258H	Timer RG general register A	TRGGRA	RW	—	—	√	FFFFH
F0259H							
F025AH	Timer RG general register B	TRGGRB	RW	—	—	√	FFFFH
F025BH							
F02D8H	RTC clock select register	RTCCL	R/W	√	√	—	00H
F02DEH	Peripheral clock control register	PCKC	R/W	√	√	—	00H
F02E0H	DTC base address register	DTCBAR	R/W	√	√	—	FDH
F02E5H	PLL control register	DSCCTL	R/W	√	√	—	00H
F02E6H	Main clock control register	MCKC	R/W	√	√	—	00H
F02E8H	DTC activation enable register 0	DTCEN0	R/W	√	√	—	00H
F02E9H	DTC activation enable register 1	DTCEN1	R/W	√	√	—	00H
F02EAH	DTC activation enable register 2	DTCEN2	R/W	√	√	—	00H
F02F0H	Flash memory CRC control register	CRC0CTL	R/W	√	√	—	00H
F02F2H	Flash memory CRC operation result register	PGCRCL	R/W	—	—	√	0000H
F02FAH	CRC data register	CRCD	R/W	—	—	√	0000H
F0300H	Event output destination select register 00	ELSELR00	R/W	—	√	—	00H
F0301H	Event output destination select register 01	ELSELR01	R/W	—	√	—	00H
F0302H	Event output destination select register 02	ELSELR02	R/W	—	√	—	00H
F0303H	Event output destination select register 03	ELSELR03	R/W	—	√	—	00H
F0304H	Event output destination select register 04	ELSELR04	R/W	—	√	—	00H
F0305H	Event output destination select register 05	ELSELR05	R/W	—	√	—	00H

Table 3 - 13 Extended Special Function Register (2nd SFR) List (6/7)

Address	Extended Special Function Register (2nd SFR) Name	Symbol		R/W	Manipulatable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0306H	Event output destination select register 06	ELSELR06		R/W	—	√	—	00H
F0307H	Event output destination select register 07	ELSELR07		R/W	—	√	—	00H
F0308H	Event output destination select register 08	ELSELR08		R/W	—	√	—	00H
F0309H	Event output destination select register 09	ELSELR09		R/W	—	√	—	00H
F030AH	Event output destination select register 10	ELSELR10		R/W	—	√	—	00H
F030BH	Event output destination select register 11	ELSELR11		R/W	—	√	—	00H
F030CH	Event output destination select register 12	ELSELR12		R/W	—	√	—	00H
F030DH	Event output destination select register 13	ELSELR13		R/W	—	√	—	00H
F030EH	Event output destination select register 14	ELSELR14		R/W	—	√	—	00H
F030FH	Event output destination select register 15	ELSELR15		R/W	—	√	—	00H
F0440H	Analog front-end power selection register	AFEPWS		R/W	√	√	—	00H
F0441H	Analog front-end power detection register	AFEPWD		R	√	√	—	00H
F0442H	Analog front-end clock selection register	AFECKS		R/W	—	√	—	00H
F0443H	Sensor reference voltage setting register	VSBIAS		R/W	—	√	—	10H
F0450H	ΔΣ A/D converter conversion result register C	DSADCRC	DSADCR0	R	—	√	√	0000H
F0451H	ΔΣ A/D converter conversion result register L	DSADCRL		R	—	√		
F0452H	ΔΣ A/D converter conversion result register M	DSADCRM	DSADCR1	R	—	√	√	0000H
F0453H	ΔΣ A/D converter conversion result register H	DSADCRH		R	—	√		
F0454H	ΔΣ A/D converter mean value register C	DSADMVC	DSADMV0	R	—	√	√	0000H
F0455H	ΔΣ A/D converter mean value register L	DSADMVL		R	—	√		
F0456H	ΔΣ A/D converter mean value register M	DSADMVM	DSADMV1	R	—	√	√	0000H
F0457H	ΔΣ A/D converter mean value register H	DSADMVH		R	—	√		
F0458H	ΔΣ A/D converter mode register	DSADMR		R/W	—	√	—	00H
F0459H	ΔΣ A/D converter control register	DSADCTL		R/W	√	√	—	00H
F045AH	Input multiplexer 0 setting register 0	PGA0CTL0		R/W	—	√	—	40H
F045BH	Input multiplexer 0 setting register 1	PGA0CTL1		R/W	—	√	—	10H
F045CH	Input multiplexer 0 setting register 2	PGA0CTL2		R/W	—	√	—	01H
F045DH	Input multiplexer 0 setting register 3	PGA0CTL3		R/W	—	√	—	00H
F045EH	Input multiplexer 1 setting register 0	PGA1CTL0		R/W	—	√	—	40H
F045FH	Input multiplexer 1 setting register 1	PGA1CTL1		R/W	—	√	—	10H
F0460H	Input multiplexer 1 setting register 2	PGA1CTL2		R/W	—	√	—	01H
F0461H	Input multiplexer 1 setting register 3	PGA1CTL3		R/W	—	√	—	00H
F0462H	Input multiplexer 2 setting register 0	PGA2CTL0		R/W	—	√	—	40H

Table 3 - 14 Extended Special Function Register (2nd SFR) List (7/7)

Address	Extended Special Function Register (2nd SFR) Name	Symbol	R/W	Manipulatable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0463H	Input multiplexer 2 setting register 1	PGA2CTL1	R/W	—	√	—	10H
F0464H	Input multiplexer 2 setting register 2	PGA2CTL2	R/W	—	√	—	01H
F0465H	Input multiplexer 2 setting register 3	PGA2CTL3	R/W	—	√	—	00H
F0466H	Input multiplexer 3 setting register 0	PGA3CTL0	R/W	—	√	—	40H
F0467H	Input multiplexer 3 setting register 1	PGA3CTL1	R/W	—	√	—	10H
F0468H	Input multiplexer 3 setting register 2	PGA3CTL2	R/W	—	√	—	01H
F0469H	Input multiplexer 3 setting register 3	PGA3CTL3	R/W	—	√	—	00H
F046AH	Input multiplexer 4 setting register 0	PGA4CTL0	R/W	—	√	—	40H
F046BH	Input multiplexer 4 setting register 1	PGA4CTL1	R/W	—	√	—	00H
F046CH	Input multiplexer 4 setting register 2	PGA4CTL2	R/W	—	√	—	01H
F046DH	Input multiplexer 4 setting register 3	PGA4CTL3	R/W	—	√	—	00H
F046EH	Disconnection detection setting register	PGABOD	R/W	—	√	—	00H
F0470H	Configurable amplifier 0 output selection register	AMP0S0	R/W	—	√	—	00H
F0471H	Configurable amplifier 0 negative input selection register	AMP0S1	R/W	√	√	—	01H
F0472H	Configurable amplifier 0 positive input selection register	AMP0S2	R/W	√	√	—	02H
F0473H	Configurable amplifier 0 mode register	AMP0MR	R/W	—	√	—	00H
F0474H	Configurable amplifier 1 output selection register	AMP1S0	R/W	—	√	—	00H
F0475H	Configurable amplifier 1 negative input selection register	AMP1S1	R/W	√	√	—	04H
F0476H	Configurable amplifier 1 positive input selection register	AMP1S2	R/W	√	√	—	08H
F0477H	Configurable amplifier 1 mode register	AMP1MR	R/W	—	√	—	00H
F0478H	Configurable amplifier 2 output selection register	AMP2S0	R/W	—	√	—	00H
F0479H	Configurable amplifier 2 negative input selection register	AMP2S1	R/W	√	√	—	10H
F047AH	Configurable amplifier 2 positive input selection register	AMP2S2	R/W	√	√	—	20H
F047BH	Configurable amplifier 2 mode register	AMP2MR	R/W	—	√	—	00H
F047CH	Configurable amplifier 0 trimming register	AMP0CAL	R/W	—	√	—	00H
F047DH	Configurable amplifier 1 trimming register	AMP1CAL	R/W	—	√	—	00H
F047EH	Configurable amplifier 2 trimming register	AMP2CAL	R/W	—	√	—	00H
F0480H	D/A converter mode register 0	DACM0	R/W	—	√	—	00H
F0481H	D/A converter mode register 1	DACM1	R/W	—	√	—	00H
F0482H	D/A converter data register	DACDL / DACD	R/W	—	√	√	0000H
F0483H		— / DACD		—	—		
F0500H	Timer RJ counter register 0	TRJ0	R/W	—	—	√	FFFFH

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

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

The RL78/I1E 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 (PM1, PM4) Port registers (P1, P4, P12, P13) Pull-up resistor option registers (PU1, PU4) Port input mode registers (PIM1) Port output mode registers (POM1) Port mode control registers (PMC1, PMC4)
Ports	• 32-pin products 10 in total (CMOS I/O: 7 (N-ch open drain I/O [V_{DD} tolerance]: 6), CMOS input: 3 • 36-pin products 14 in total (CMOS I/O: 11 (N-ch open drain I/O [V_{DD} tolerance]: 6), CMOS input: 3
Pull-up resistors	• 32-pin products: 7 in total • 36-pin products: 7 in total

4.2.1 Port 1

Port 1 is an I/O port with an output latch. Port 1 can be set to the input mode or output mode in 1-bit units by using port mode register 1 (PM1). When the P10 to P15 pins are used as input ports, use of an on-chip pull-up resistor can be specified in 1-bit units by using pull-up resistor option register 1 (PU1).
Input to the P11, P12, P14, and P15 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units by using port input mode register 1 (PIM1).
Output from the P10 to P15 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units by using port output mode register 1 (POM1).
To use the P16 and P17 pins as digital I/O pins, set them to digital I/O by using port mode control register 1 (PMC1). (This can be specified in 1-bit units.)
Port 1 can also be used for digital functions such as timer I/O, external interrupt request input, serial interface data I/O, clock I/O, and transmission/reception using the programming UART, and for analog functions such as analog input to A/D converters and analog output to the configurable amplifier.
To use the P16 and P17 pins as analog I/O pins, set them to analog I/O by using port mode control register 1 (PMC1). (This can be specified in 1-bit units.)
Reset signal generation sets the P10 to P15 pins to be input ports, and the P16 and P17 pins to be used for analog functions.

4.2.2 Port 4

Port 4 is an I/O port with an output latch. Port 4 can be set to the input mode or output mode in 1-bit units by using port mode register 4 (PM4). When the P40 pin is used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by using pull-up resistor option register 4 (PU4).
To use the P41 and P42 pins as digital I/O pins, set them to digital I/O by using port mode control register 4 (PMC4). (This can be specified in 1-bit units.)
Port 4 can also be used for digital functions such as data I/O for flash memory programmers and debuggers, and for analog functions such as analog input to A/D converters and analog output to the configurable amplifier.
To use the P41 and P42 pins as analog I/O pins, set them to analog I/O by using port mode control register 4 (PMC4). (This can be specified in 1-bit units.)
Reset signal generation sets the P40 pin to be an input port, and the P41 and P42 pins to be analog function ports.

4.2.3 Port 12

P121 and P122 are input-only ports.
This port can also be used for connecting a resonator for the main system clock and for an external clock input for the main system clock.

4.2.4 Port 13

P137 is a 1-bit input-only port.
This port can also be used for external interrupt request input and a serial interface chip select input.

4.3 Registers Controlling Port Function

Port functions are controlled by the following registers.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Pull-up resistor option registers (PUxx)
- Port input mode registers (PIMxx)
- Port output mode registers (POMxx)
- Port mode control registers (PMCxx)

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

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

Port		Bit name						32- pin	36 pin
		PMxx register	Pxx register	PUxx register	PIMxx register	POMxx register	PMCxx register		
Port 1	0	PM10	P10	PU10	—	POM10	—	√	√
	1	PM11	P11	PU11	PIM11	POM11	—	√	√
	2	PM12	P12	PU12	PIM12	POM12	—	√	√
	3	PM13	P13	PU13	—	POM13	—	√	√
	4	PM14	P14	PU14	PIM14	POM14	—	√	√
	5	PM15	P15	PU15	PIM15	POM15	—	√	√
	6	PM16	P16	—	—	—	PMC16	—	√
	7	PM17	P17	—	—	—	PMC17	—	√
Port 4	0	PM40	P40	PU40	—	—	—	√	√
	1	PM41	P41	—	—	—	PMC41	—	√
	2	PM42	P42	—	—	—	PMC42	—	√
	3	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—
Port 12	0	—	—	—	—	—	—	—	—
	1	—	P121	—	—	—	—	√	√
	2	—	P122	—	—	—	—	√	√
	3	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—
	7	—	—	—	—	—	—	—	—
Port 13	0	—	—	—	—	—	—	—	—
	1	—	—	—	—	—	—	—	—
	2	—	—	—	—	—	—	—	—
	3	—	—	—	—	—	—	—	—
	4	—	—	—	—	—	—	—	—
	5	—	—	—	—	—	—	—	—
	6	—	—	—	—	—	—	—	—
	7	—	P137	—	—	—	—	√	√

4.3.1 Port mode registers (PMxx)

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

Figure 4 - 1 Format of Port mode register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM4	1	1	1	1	1	PM42	PM41	PM40	FFF24H	FFH	R/W

PMmn	Selection of Pmn pin I/O mode (m = 1 or 4, n = 0 to 7)
0	Output mode (the pin functions as an output port (output buffer on))
1	Input mode (the pin functions as an input port (output buffer off))

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

4.3.2 Port registers (Pxx)

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

Note When P16, P17, P41, and P42 are set to the analog function, if a port is read in input mode, the read value is always 0, not the pin level.

Figure 4 - 2 Format of Port register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P1	P17	P16	P15	P14	P13	P12	P11	P10	FFF01H	00H (output latch)	R/W
P4	0	0	0	0	0	P42	P41	P40	FFF04H	00H (output latch)	R/W
P12	0	0	0	0	0	P122	P121	0	FFF0CH	Undefined	R/W Note
P13	P137	0	0	0	0	0	0	0	FFF0DH	Undefined	R/W Note

Pmn	m = 1, 4, 12, or 13, n = 0 to 7	
	Output data control (in output mode)	Input data read (in input mode)
0	Output 0	Input low level
1	Output 1	Input high level

Note P121, P121, and P137 are read-only.

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

4.3.3 Pull-up resistor option registers (PUxx)

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

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

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

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

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU1	0	0	PU15	PU14	PU13	PU12	PU11	PU10	F0031H	00H	R/W
PU4	0	0	0	0	0	0	0	PU40	F0034H	01H	R/W

PUmn	Selection of Pmn pin on-chip pull-up resistor (m = 1 or 4, n = 0 to 5)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

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

4.3.4 Port input mode registers (PIMxx)

These registers set the input buffer in 1-bit units.

TTL input buffer can be selected during serial communication with an external device of the different potential.

Port input mode registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 4 - 4 Format of Port input mode register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM1	0	0	PIM15	PIM14	0	PIM12	PIM11	0	F0041H	00H	R/W

PIM1n	P1n pin input buffer selection (n = 1, 2, 4, or 5)
0	Normal input buffer
1	TTL input buffer

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

4.3.5 Port output mode registers (POMxx)

These registers set the output mode in 1-bit units.
N-ch open-drain output (VDD tolerance) mode can be selected during serial communication with an external device of the different potential, and for the SDA00 and SDA01 pins during simplified I²C communication with an external device of the same potential.
In addition, POMxx register is set with PUxx register, whether or not to use the on-chip pull-up resistor.
These registers can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears these registers to 00H.

Caution An on-chip pull-up resistor is not connected to a bit for which N-ch open drain output (VDD tolerance) mode (PON1n = 1) is set.

Figure 4 - 5 Format of Port output mode register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
POM1	0	0	POM15	POM14	POM13	POM12	POM11	POM10	F0051H	00H	R/W

POM1n	P1n pin output mode selection (n = 0 to 5)
0	Normal output mode
1	N-ch open-drain output (VDD tolerance) mode

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

4.3.6 Port mode control registers (PMCxx)

These registers set the P16, P17, P41, and P42 digital I/O/analog I/O in 1-bit units.
The PMC1 and PMC4 registers can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears these registers to FFH.

Figure 4 - 6 Format of Port mode control register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PMC1	PMC17	PMC16	1	1	1	1	1	1	F0061H	FFH	R/W
PMC4	1	1	1	1	1	PMC42	PMC41	1	F0064H	FFH	R/W

PMCmn	Pmn pin digital I/O/analog I/O selection (mn = 16, 17, 41, or 42)
0	Digital I/O (alternate function other than analog input)
1	Analog I/O

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

4.4 Port Function Operations

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

4.4.1 Writing to I/O port

(1) Output mode

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

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

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

(2) Input mode

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

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

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

4.4.2 Reading from I/O port

(1) Output mode

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

(2) Input mode

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

4.4.3 Operations on I/O port

(1) Output mode

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

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

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

(2) Input mode

The pin level is read and an operation is performed on its contents. The result of the operation is written to the output latch, but since the output buffer is off, the pin status does not change. Therefore, byte data can be written to the ports used for both input and output.

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

4.4.4 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers

It is possible to connect an external device operating on a different potential (1.8 V, 2.5 V or 3 V) by switching I/O buffers with the port input mode register (PIMxx) and port output mode register (POMxx).
When receiving input from an external device with a different potential (1.8 V, 2.5 V or 3 V), set port input mode register 1 (PIM1) on a bit-by-bit basis to enable normal input (CMOS)/TTL input buffer switching.
When outputting data to an external device with a different potential (1.8 V, 2.5 V or 3 V), set port output mode register 1 (POM1) on a bit-by-bit basis to enable normal output (CMOS)/N-ch open drain (V_{DD} tolerance) switching.

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

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

In case of UART0:	P14
In case of UART1:	P11
In case of CSI00:	P14, P15
In case of CSI01:	P11, P12

<1> Using an external resistor, pull up the pin to be used to the power supply of the target device (on-chip pull-up resistor cannot be used).
<2> Set the corresponding bit of the PIM1 register to 1 to switch to the TTL input buffer. For V_{IH} and V_{IL}, refer to the DC characteristics when the TTL input buffer is selected.
<3> Enable the operation of the serial array unit and set the mode to the UART/Simplified SPI (CSI [Note]) mode.

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

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

In case of UART0:	P13
In case of UART1:	P10
In case of CSI00:	P13
In case of CSI01:	P10

<1> Using an external resistor, pull up the pin to be used to the power supply of the target device (on-chip pull-up resistor cannot be used).
<2> After reset release, the port mode is the input mode (Hi-Z).
<3> Set the output latch of the corresponding port to 1.
<4> Set the corresponding bit of the POM1 register to 1 to set the N-ch open drain output (V_{DD} tolerance) mode.
<5> Enable the operation of the serial array unit and set the mode to the UART/Simplified SPI (CSI) mode.
<6> Set the corresponding bit of the PM1 register to the output mode. At this time, the output data is high level, so the pin is in the Hi-Z state.

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

In case of simplified IIC00: P14, P15
In case of simplified IIC01: P11, P12

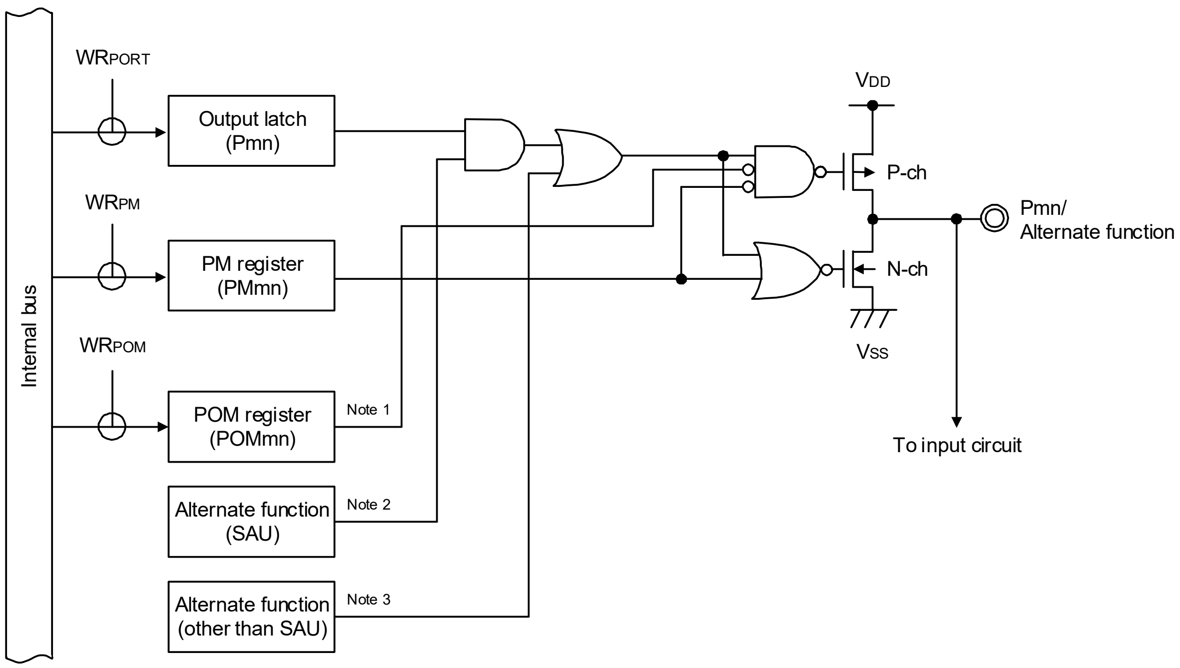
<1> Using an external resistor, pull up the pin to be used to the power supply of the target device (on-chip pull-up resistor cannot be used).
<2> After reset release, the port mode is the input mode (Hi-Z).
<3> Set the output latch of the corresponding port to 1.
<4> Set the corresponding bit of the POM1 register to 1 to set the N-ch open drain output (V_{DD} tolerance) mode.
<5> Set the corresponding bit of the PIM1 register to 1 to switch to the TTL input buffer. For V_{IH} and V_{IL}, refer to the DC characteristics when the TTL input buffer is selected.
<6> Enable the operation of the serial array unit and set the mode to the simplified I^2C mode.
<7> Set the corresponding bit of the PM1 register to the output mode (data I/O is possible in the output mode). At this time, the output data is high level, so the pin is in the Hi-Z state.

4.5 Register Settings When Using Alternate Function

4.5.1 Basic concept when using alternate function

In the beginning, for a pin also assigned to be used for analog function, use the port mode control register (PMCxx) to specify whether to use the pin for analog function or digital input/output.
Figure 4 - 7 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 (such as timers, RTC, and clock/buzzer output) is connected to the other input pin of the OR gate. When such kind of pins are used by the port function or an alternate function, the unused alternate function must not hinder the output of the function to be used. An idea of basic settings for this kind of case is shown in Table 4 - 3.

Figure 4 - 7 Basic Configuration of Output Circuit for Pins



Note 1. When there is no POM register, this signal should be considered to be low level (0).
Note 2. When there is no alternate function, this signal should be considered to be high level (1).
Note 3. When there is no alternate function, this signal should be considered to be low level (0).

Table 4 - 3 Concept of Basic Settings

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

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

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

When the output of an alternate function of the pin is not used, specify the following settings.

(1) SOp = 1, TxDq = 1 (settings when the serial output (SOp/TxDq) of SAU is not used)
When the serial output (SOp/TxDq) is not used, such as, a case in which only the serial input of SAU is used, set the bit in serial output enable register m (SOEm) which corresponds to the unused output to 0 (output disabled) and set the SOmn bit in serial output register m (SOm) to 1 (high). These are the same settings as the initial state.

(2) SCKp = 1, SDAr = 1, SCLr = 1 (settings when channel n in SAU is not used)
When SAU is not used, set bit n (SEmn) in serial channel enable status register m (SEm) to 0 (operation stopped state), set the bit in serial output enable register m (SOEm) which corresponds to the unused output to 0 (output disabled), and set the SOmn and CKOmn bits in serial output register m (SOm) to 1 (high). These are the same settings as the initial state.

(3) TOmn = 0 (settings when the output of channel n in TAU is not used)
When the TOmn output of TAU is not used, set the bit in timer output enable register 0 (TOE0) which corresponds to the unused output to 0 (output disabled) and set the bit in timer output register 0 (TO0) to 0 (low). These are the same settings as the initial state.

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

(5) TRJIO0 = 0/TRJO0 = 0 (setting when timer RJ output is not used)
When the pulse output function of timer RJ is not used with the TRJO0 pin, set bit 2 (TOENA) in the timer RJ I/O control register (TRJIOC0) to 0 (TRJO output disabled). This is the same setting as the initial state.
When the TRJIO0 pin of timer RJ is not used for the output function, set bits 2 to 0 (TMOD2 to TMOD0) in timer RJ mode register 0 (TRJMR0) to a value other than 001b (pulse output mode). The initial value is 000b (timer mode).

(6) TRGIOA = 0/TRGIOB = 0 (setting when timer RG output is not used)
When the output function of timer RG is not used, set the pins not used for timer RG output function to “pin output by compare match is disabled” using the timer RG I/O control register (TRGIOR). This is the same setting as the initial state.

4.5.3 Register setting examples for used ports and alternate functions

Register setting examples for used ports and alternate functions are shown in Tables 4 - 4 and 4 - 5. The registers used to control the port functions should be set as shown in Tables 4 - 4 and 4 - 5. See the following remark for legends used in Tables 4 - 4 and 4 - 5.

Remark

—:	Not supported
×:	Don't care
POMxx:	Port output mode register
PMCxx:	Port mode control register
PMxx:	Port mode register
Pxx:	Port output latch

Table 4 - 4 Setting Examples of Registers When Using P10 to P17, P40 to P42, and P137 Pin Functions (1/2)

Pin Name	Used Function		POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		32-pin	36-pin
	Function Name	I/O					SAU Output Function	Other than SAU		
P10	P10	Input	×	—	1	×	—	—	√	√
		Output	0	—	0	0/1	SO01 = 1, TxD1 = 1	TO01 = 0, TRGIOA = 0	√	√
		Nch-OD output	1	—	0	0/1			√	√
	SO01	Output	0/1	—	0	1	TxD1 = 1		√	√
	TxD1	Output	0/1	—	0	1	SO01 = 1		√	√
	TI01	Input	×	—	1	×	—	—	√	√
	TO01	Output	0	—	0	0	SO01 = 1, TxD1 = 1	TRGIOA = 0	√	√
	INTP1	Input	×	—	1	×	—	—	√	√
	TRGIOA	Input	×	—	1	×	—	—	√	√
		Output	0	—	0	0	SO01 = 1, TxD1 = 1	TO01 = 0	√	√
P11	P11	Input	×	—	1	×	—	—	√	√
		Output	0	—	0	0/1	SDA01 = 1	TO03 = 0, TRJIO0 = 0		
		Nch-OD output	1	—	0	0/1				
	SI01	Input	×	—	1	×	—	—	√	√
	RxD1	Input	×	—	1	×	—	—	√	√
	SDA01	I/O	1	—	0	1	—	TO03 = 0, TRJIO0 = 0	√	√
	TI03	Input	×	—	1	×	—	—	√	√
	TO03	Output	0	—	0	0	SDA01 = 1	TRJIO0 = 0	√	√
	INTP2	Input	×	—	1	×	—	—	√	√
	TRGCLKA	Input	×	—	1	×	—	—	√	√
	TRJIO0	Input	×	—	1	×	—	—	√	√
		Output	0	—	0	0	SDA01 = 1	TO03 = 0	√	√
P12	P12	Input	×	—	1	×	—	—	√	√
		Output	0	—	0	0/1	SCK01 = 1, SCL01 = 1	TO11 = 0, PCLBUZ0 = 0, TRJO0 = 0		
		Nch-OD output	1	—	0	0/1				
	SCK01	Input	×	—	1	×	—	—	√	√
		Output	0/1	—	0	1	—	TO11 = 0, PCLBUZ0 = 0, TRJO0 = 0	√	√
	SCL01	Output	0/1	—	0	1	—		√	√
	TI11	Input	×	—	1	×	—	—	√	√
	TO11	Output	0	—	0	0	SCK01 = 1, SCL01 = 1	PCLBUZ0 = 0, TRJO0 = 0	√	√
	INTP3	Input	×	—	1	×	—	—	√	√
	PCLBUZ0	Output	0	—	0	0	SCK01 = 1, SCL01 = 1	TO11 = 0, TRJO0 = 0	√	√
	TRGIOB	Input	×	—	1	×	—	—	√	√
		Output	0	—	0	0	SCK01 = 1, SCL01 = 1	TO11 = 0, PCLBUZ0 = 0	√	√
	TRJO0	Output	0	—	0	0			√	√
P13	P13	Input	×	—	1	×	—	—	√	√
		Output	0	—	0	0/1	SO00 = 1, TxD0 = 1	TO00 = 0, RTC1HZ = 0		
		Nch-OD output	1	—	0	0/1				
	SO00	Output	0/1	—	0	1	—		√	√
	TxD0	Output	0/1	—	0	1	—		√	√
	TI00	Input	×	—	1	×	—	—	√	√
	TO00	Output	0	—	0	0	SO00 = 1, TxD0 = 1	RTC1HZ = 0	√	√
	INTP4	Input	×	—	1	×	—	—	√	√
	RTC1HZ	Output	0	—	0	0	SO00 = 1, TxD0 = 1	TO00 = 0	√	√

Table 4 - 4 Setting Examples of Registers When Using P10 to P17, P40 to P42, and P137 Pin Functions (2/2)

Pin Name	Used Function		POMxx	PMCxx	PMxx	Pxx	Alternate Function Output		32-pin	36-pin
	Function Name	I/O					SAU Output Function	Other than SAU		
P14	P14	Input	×	—	1	×	—	—	√	√
		Output	0	—	0	0/1	SDA00 = 1	TO02 = 0		
		Nch-OD output	1	—	0	0/1				
	SI00	Input	×	—	1	×	—	—	√	√
	RxD0	Input	×	—	1	×	—	—	√	√
	SDA00	I/O	1	—	0	1	—	TO02 = 0	√	√
	TI02	Input	×	—	1	×	—	—	√	√
	TO02	Output	0	—	0	0	SDA00 = 1	—	√	√
	INTP5	Input	×	—	1	×	—	—	√	√
P15	P15	Input	×	—	1	×	—	—	√	√
		Output	0	—	0	0/1	SCK00 = 1, SCL00 = 1	TO10 = 0		
		Nch-OD output	1	—	0	0/1				
	SCK00	Input	×	—	1	×	—	—	√	√
		Output	0/1	—	0	1	—	TO10 = 0	√	√
	SCL00	Output	0/1	—	0	1	—		√	√
	TI10	Input	×	—	1	×	—	—	√	√
	TO10	Output	0	—	0	0	SCK00 = 1, SCL00 = 1	—	√	√
	INTP6	Input	×	—	1	×	—	—	√	√
	TRGCLKB	Input	×	—	1	×	—	—	√	√
P16	P16	Input	—	0	1	×	—	—	—	√
		Output	—	0	0	0/1	—	—		
	INTP7	Input	—	0	1	×	—	—	—	√
	ANI9	Analog input	—	1	1	×	—	—	—Note	√
	AMP2P	Analog input	—	1	1	×	—	—	—Note	√
	ANX5	Analog I/O	—	1	1	×	—	—	—Note	√
P17	P17	Input	—	0	1	×	—	—	—	√
		Output	—	0	0	0/1	—	—		
	ANI8	Analog input	—	1	1	×	—	—	—	√
	AMP2N	Analog input	—	1	1	×	—	—	—	√
	ANX4	Analog I/O	—	1	1	×	—	—	—	√
P40	P40	Input	×	—	1	×	—	—	√	√
		Output	0	—	0	0/1	—	—		
P41	P41	Input	—	0	1	×	—	—	—	√
		Output	—	0	0	0/1	—	—		√
	ANI6	Analog input	—	1	1	×	—	—	—Note	√
	AMP1P	Analog input	—	1	1	×	—	—	—Note	√
	ANX3	Analog I/O	—	1	1	×	—	—	—Note	√
P42	P42	Input	—	0	1	×	—	—	—	√
		Output	—	0	0	0/1	—	—		
	ANI5	Analog input	—	1	1	×	—	—	—Note	√
	AMP1N	Analog input	—	1	1	×	—	—	—Note	√
	ANX2	Analog I/O	—	1	1	×	—	—	—Note	√
P137	P137	Input	—	—	—	×	—	—	√	√
	$\overline{SSI00}$	Input	—	—	—	×	—	—	√	√
	INTP0	Input	—	—	—	×	—	—	√	√

Note In 32-pin products, this pin does not function as a port but functions as analog function port.

Table 4 - 5 Setting Examples of Registers When Using P121 and P122 Pin Functions

Pin Name	Used Function		CMC	Pxx	32-pin	36-pin
	Function Name	I/O	(EXCLK, OSCSEL)			
P121	P121	Input	00/10/11	×	√	√
	X1	—	01	—		
P122	P122	Input	00/10/11	×	√	√
	EXCLK	Input	11	—		
	X2	—	01	—		

4.6 Cautions When Using Port Function

4.6.1 Cautions on 1-bit manipulation instruction for port register n (Pn)

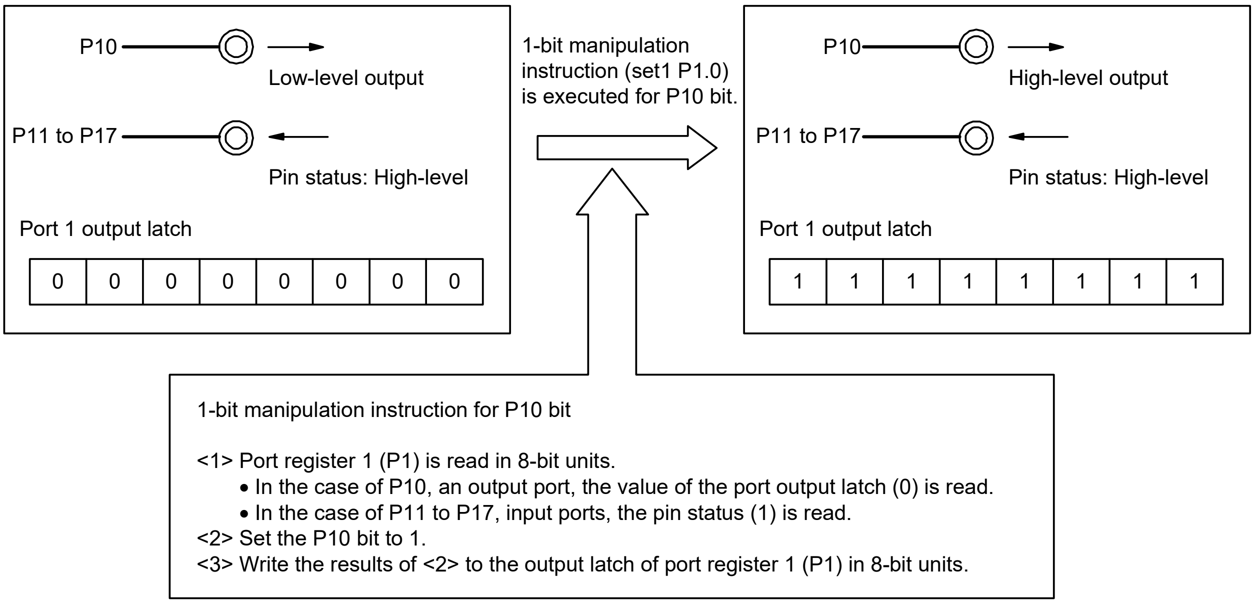
When a 1-bit manipulation instruction is executed on a port that provides both input and output functions, the output latch value of an input port that is not subject to manipulation may be written in addition to the targeted bit.
Therefore, it is recommended to rewrite the output latch when switching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

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

Description: 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/I1E.

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

Figure 4 - 8 Bit Manipulation Instruction (P10)



4.6.2 Cautions on specifying the pin settings

For an output pin to which multiple functions are assigned, the output of the unused alternate functions must be set to its initial state so as to prevent conflicting outputs. For details about the alternate function output, see 4.5 Register Settings When Using Alternate Function.
No specific setting is required for input pins because the output of their alternate functions is disabled (the buffer output is Hi-Z).

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

CHAPTER 5 CLOCK GENERATOR

5.1 Functions of Clock Generator

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

(1) Main system clock

<1> X1 oscillator

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

<2> High-speed on-chip oscillator (High-speed OCO)

The frequency at which to oscillate can be selected from among f_{HOCO} = 32, 24, 16, 12, 8, 6, 4, 3, 2, or 1 MHz (TYP.) by using the option byte (000C2H). After a reset release, the CPU always starts operating with this high-speed on-chip oscillator clock. Oscillation can be stopped by executing the STOP instruction or setting of the HIOSTOP bit (bit 0 of the CSC register).

The frequency specified by using an option byte can be changed by using the high-speed on-chip oscillator frequency select register (HOCODIV). For details about the frequency, see **Figure 5 - 11 Format of High-speed on-chip oscillator frequency select register (HOCODIV)**.

The frequencies that can be specified for the high-speed on-chip oscillator by using the option byte and the high-speed on-chip oscillator frequency select register (HOCODIV) are shown below.

Power Supply Voltage	Oscillation Frequency (MHz)									
	1	2	3	4	6	8	12	16	24	32
2.7 V ≤ V_{DD} ≤ 5.5 V	√	√	√	√	√	√	√	√	√	√
2.4 V ≤ V_{DD} ≤ 5.5 V	√	√	√	√	√	√	√	√	—	—

<3> PLL clock

A clock with a frequency of f_{PLL} = 24, 32, 48, or 64 MHz can be generated by oscillating the X1 oscillator at 4 or 8 MHz and multiplying the obtained clock by 3, 4, 6, or 8 in the PLL. If the CKSELR bit is set to 1, the clock generated by dividing f_{PLL} by 2, 4, or 8 (the division factor is determined by the setting of the RDIV0 and RDIV1 bits) is selected as the source of the main system clock (f_{IH}).

The PLL can be started and stopped by using the DSCON bit (bit 0 of the DSCCTL register).

Use the PLL clock if you want to select a 20 MHz or faster clock as the CPU/peripheral hardware clock (f_{CLK}) when the X1 oscillator clock is used as the 24-bit ΔΣ A/D converter clock. For details about the PLL setting, see Figure 5 - 12 Format of PLL control register (DSCCTL). For the relationship between the relationship between the PLL and the 24-bit ΔΣ A/D converter clock, see Figure 5 - 13 Relationship Between PLL and ΔΣ A/D Converter.

Remark A 4 or 8 MHz clock can be input to the PLL.

An external main system clock (f_{EX} = 1 to 20 MHz) can also be supplied from the EXCLK/X2/P122 pin. An external main system clock input can be disabled by executing the STOP instruction or setting of the MSTOP bit. As the main system clock, a high-speed system clock (X1 clock or external main system clock) or 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) Low-speed on-chip oscillator (Low-speed OCO)
This circuit oscillates a clock of f_{IL} = 15 kHz (TYP.).
The low-speed on-chip oscillator clock cannot be used as the CPU clock.
Only the following peripheral hardware runs on the low-speed on-chip oscillator clock.

- Watchdog timer
- Timer RJ
- Interval timer[Note]

Note When using the real-time clock, the low-speed on-chip oscillator clock (f_{IL}) cannot be selected as the count clock for the interval timer.

This clock operates when bit 4 (WDTON) of the option byte (000C0H), bit 4 (WUTMMCK0) of the subsystem clock supply mode control register (OSMC), or both are set to 1.
However, if WDTON is 1, WUTMMCK0 is 0, and bit 0 (WDSTBYON) of the option byte (000C0H) is 0, the low-speed on-chip oscillator stops oscillating when the HALT or STOP instruction is executed.

Caution The low-speed on-chip oscillator clock (f_{IL}) can only be selected as the real-time clock count clock when the fixed-cycle interrupt function is used.

Remark
f_X: X1 clock oscillation frequency
f_{IH}: Main system clock source frequency generated by dividing high-speed on-chip oscillator clock frequency (f_{HOCO}) or PLL clock frequency (f_{PLL}) by 2, 4, or 8
f_{EX}: External main system clock frequency
f_{PLL}: PLL clock oscillation frequency
f_{IL}: Low-speed on-chip oscillator frequency

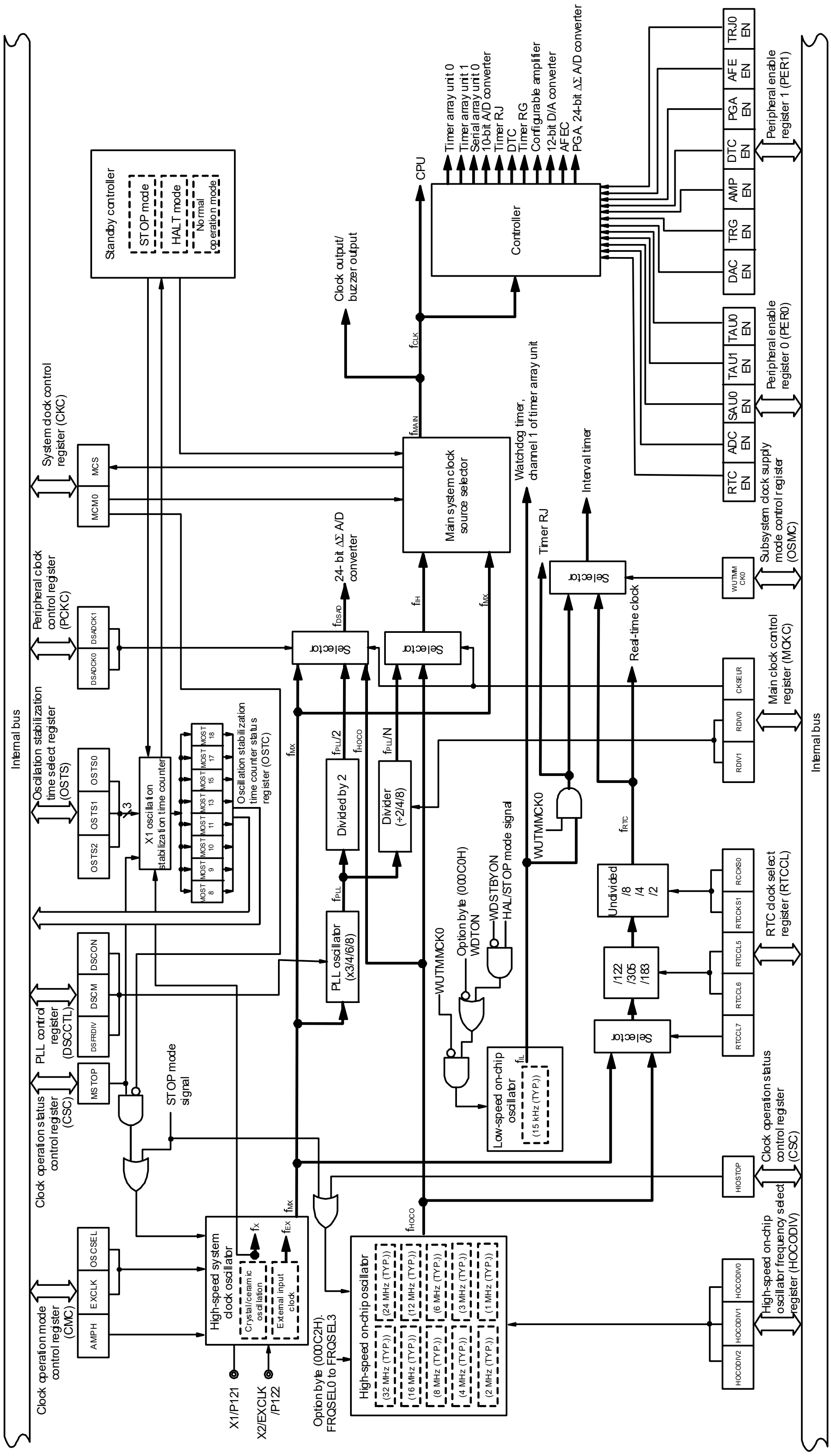
5.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 5 - 1 Configuration of Clock Generator

Item	Configuration
Control registers	Clock operation mode control register (CMC) System clock control register (CKC) Clock operation status control register (CSC) Oscillation stabilization time counter status register (OSTC) Oscillation stabilization time select register (OSTS) Peripheral enable registers 0, 1 (PER0, PER1) Subsystem clock supply mode control register (OSMC) High-speed on-chip oscillator frequency select register (HOCODIV) PLL control register (DSCCTL) Main clock control register (MCKC) Peripheral control register (PCKC)
Oscillators	X1 oscillator High-speed on-chip oscillator Low-speed on-chip oscillator PLL oscillator

Figure 5 - 1 Block Diagram of Clock Generator



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

Remark

f_X:	X1 clock oscillation frequency
f_{HOCO}:	High-speed on-chip oscillator clock frequency
f_{IH}:	Main system clock source frequency generated by dividing high-speed on-chip oscillator clock frequency (f_{HOCO}) or PLL clock frequency (f_{PLL}) by 2, 4, or 8
f_{PLL}:	PLL clock frequency
f_{RTC}:	RTC clock frequency
f_{DSAD}:	24-bit ΔΣ A/D converter clock frequency
f_{EX}:	External main system clock frequency
f_{MX}:	High-speed system clock frequency
f_{MAIN}:	Main system 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 following registers are used to control the clock generator.

- Clock operation mode control register (CMC)
- System clock control register (CKC)
- Clock operation status control register (CSC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)
- Peripheral enable registers 0, 1 (PER0, PER1)
- Subsystem clock supply mode control register (OSMC)
- High-speed on-chip oscillator frequency select register (HOCODIV)
- PLL control register (DSCCTL)
- Main clock control register (MCKC)
- Peripheral control register (PCKC)

5.3.1 Clock operation mode control register (CMC)

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

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

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

Reset signal generation clears this register to 00H.

Figure 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	0	0	0	0	0	AMPH

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

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

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

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

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

Caution 4. Specify the settings for the AMPH bit while f_{IH} is selected as f_{CLK} after a reset ends (before f_{CLK} is switched to f_{MX}).

Caution 5. Although the maximum system clock frequency is 32 MHz, the maximum frequency of the X1 oscillator is 20 MHz.

Remark fx: X1 clock oscillation 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]

Symbol	7	6	<5>	<4>	3	2	1	0
CKC	0	0	MCS	MCM0	0	0	0	0

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	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 Bit 5 is read-only.

Remark f_{HOCO}: High-speed on-chip oscillator clock frequency
f_{IH}: Main system clock source frequency generated by dividing high-speed on-chip oscillator clock frequency (f_{HOCO}) or PLL clock frequency (f_{PLL}) by 2, 4, or 8
f_{MX}: High-speed system clock frequency
f_{MAIN}: Main system clock frequency

Caution Be sure to set bits 0 to 3, 6, and 7 to 0.

5.3.3 Clock operation status control register (CSC)

This register is used to control the operations of the high-speed system clock and high-speed on-chip oscillator clock (except the low-speed on-chip oscillator clock).
The CSC register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to C0H.

Figure 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	1	0	0	0	0	0	HIOSTOP

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

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

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

Caution 2. Set the oscillation stabilization time select register (OSTS) before setting the MSTOP bit to 0 after releasing reset. Note that if the OSTS register is being used with its default settings, the OSTS register is not required to be set here.

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

Caution 4. Do not stop the clock selected for the CPU peripheral hardware clock (f_{CLK}) with the OSC register.

Caution 5. The setting of the flags of the register to stop clock oscillation (invalidate the external clock input) and the condition before clock oscillation is to be stopped are as Table 5 - 2.
When stopping the clock, confirm the condition before stopping clock.

Table 5 - 2 Stopping Clock Method

Clock	Condition Before Stopping Clock (Invalidating External Clock Input)	Setting of CSC Register Flags
X1 clock External main system clock	The CPU/peripheral hardware clock operates with a clock other than the high-speed system clock. (MCS = 0)	MSTOP = 1
High-speed on-chip oscillator clock	The CPU/peripheral hardware clock operates with a clock other than the high-speed on-chip oscillator clock. (MCS = 1)	HIOSTOP = 1

5.3.4 Oscillation stabilization time counter status register (OSTC)

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

- When the X1 clock starts oscillating while the high-speed on-chip oscillator clock is being used as the CPU clock.
- When the STOP mode is entered and then exited 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.
Reset signal generation and STOP instruction execution, and setting MSTOP (bit 7 of clock operation status control register (CSC)) to 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 exited

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	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18

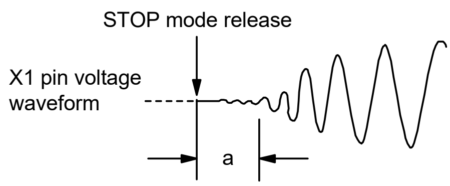
MOST 8	MOST 9	MOST 10	MOST 11	MOST 13	MOST 15	MOST 17	MOST 18	Oscillation stabilization time status		
									fx = 10 MHz	fx = 20 MHz
0	0	0	0	0	0	0	0	2^8/fx max.	25.6 μs max.	12.8 μs max.
1	0	0	0	0	0	0	0	2^8/fx min.	25.6 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	2^9/fx min.	51.2 μs min.	25.6 μs min.
1	1	1	0	0	0	0	0	2^{10}/fx min.	102 μs min.	51.2 μs min.
1	1	1	1	0	0	0	0	2^{11}/fx min.	204 μs min.	102 μs min.
1	1	1	1	1	0	0	0	2^{13}/fx min.	819 μs min.	409 μs min.
1	1	1	1	1	1	0	0	2^{15}/fx min.	3.27 ms min.	1.63 ms min.
1	1	1	1	1	1	1	0	2^{17}/fx min.	13.1 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	2^{18}/fx min.	26.2 ms min.	13.1 ms min.

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

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

- **When the X1 clock starts oscillating while the high-speed on-chip oscillator clock is being used as the CPU clock.**
- **When the STOP mode is entered and then exited 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 applied to the OSTC register after the STOP mode is exited.)

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



Remark fx: X1 clock oscillation frequency

5.3.5 Oscillation stabilization time select register (OSTS)

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

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

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

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

Reset signal generation sets the OSTS register to 07H.

Figure 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		
				fx = 10 MHz	fx = 20 MHz
0	0	0	2^8/fx	25.6 μs	12.8 μs
0	0	1	2^9/fx	51.2 μs	25.6 μs
0	1	0	2^{10}/fx	102 μs	51.2 μs
0	1	1	2^{11}/fx	204 μs	102 μs
1	0	0	2^{13}/fx	819 μs	409 μs
1	0	1	2^{15}/fx	3.27 ms	1.63 ms
1	1	0	2^{17}/fx	13.1 ms	6.55 ms
1	1	1	2^{18}/fx	26.2 ms	13.1 ms

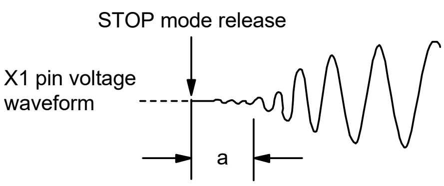
Caution 1. Change the setting of the OSTS register before setting the MSTOP bit of the clock operation status control register (CSC) to 0.

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

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

- **When the X1 clock starts oscillating while the high-speed on-chip oscillator clock is being used as the CPU clock.**
- **When the STOP mode is entered and then exited 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 applied to the OSTC register after the STOP mode is exited.)**

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



Remark fx: X1 clock oscillation frequency

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

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

- Real-time clock and interval timer
- A/D converter
- Serial array unit 0
- Timer array unit 1
- Timer array unit 0
- 12-bit D/A converter
- Timer RG
- Configurable amplifier
- DTC
- PGA
- 24-bit ΔΣ A/D converter
- Timer RJ

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)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	4	3	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	0	0	SAU0EN	TAU1EN	TAU0EN

RTCEN	Control of supplying input clock for real-time clock (RTC) and interval timer
0	Stops input clock supply. • SFRs used by the real-time clock (RTC) and interval timer cannot be written. • The real-time clock (RTC) and interval timer are in the reset status.
1	Enables input clock supply. • SFRs used by the real-time clock (RTC) and interval timer can be read and written.

ADCEN	Control of A/D converter input clock supply
0	Stops input clock supply. • SFRs used by the A/D converter cannot be written. • The A/D converter is in the reset status.
1	Enables input clock supply. • SFRs used by the A/D converter can be read and written.

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

TAU1EN	Control of timer array unit 1 input clock supply
0	Stops input clock supply. • SFRs used by timer array unit 1 cannot be written. • Timer array unit 1 is in the reset status.
1	Enables input clock supply. • SFRs used by timer array unit 1 can be read and written.

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

Caution Be sure to clear bits 3, 4, and 6 to 0

Figure 5 - 8 Format of Peripheral enable register 1 (PER1) (1/2)

Address: F007AH After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	<3>	<2>	<1>	<0>
PER1	DACEN	TRGEN	AMPEN	0	DTCEN	PGAEN	AFEEN	TRJ0EN

DACEN	Control of input clock supplied to 12-bit D/A converter
0	Stops input clock supply. • SFRs used by the 12-bit D/A converter cannot be written. • The 12-bit D/A converter is in the reset status.
1	Enables input clock supply. • SFRs used by the 12-bit D/A converter can be read and written.

TRGEN	Control of input clock supplied to timer RG
0	Stops input clock supply. • SFRs used by timer RG cannot be written. • Timer RG is in the reset status.
1	Enables input clock supply. • SFRs used by timer RG can be read and written.

AMPEN	Control of input clock supplied to configurable amplifier
0	Stops input clock supply. • SFRs used by the configurable amplifier cannot be written. • The configurable amplifier is in the reset status.
1	Enables input clock supply. • SFRs used by the configurable amplifier can be read and written.

DTCEN	Control of input clock supplied to DTC
0	Stops input clock supply. • DTC cannot operate.
1	Enables input clock supply. • DTC can operate.

PGAEN	Control of input clock supplied to PGA and 24-bit ΔΣ A/D converter
0	Stops input clock supply. • SFRs used by PGA and the 24-bit ΔΣ A/D converter cannot be written. • PGA and the 24-bit ΔΣ A/D converter are in the reset status.
1	Enables input clock supply. • SFRs used by PGA and the 24-bit ΔΣ A/D converter can be read and written.

AFEEN	Control of input clock supplied to AFE power supply/clock control block
0	Stops input clock supply. • SFRs used by the AFE power supply/clock control block cannot be written. • The AFE power supply/clock control block is in the reset status.
1	Enables input clock supply. • SFRs used by the AFE power supply/clock control block can be read and written.

Figure 5 - 9 Format of Peripheral enable register 1 (PER1) (2/2)

TRJ0EN	Control of input clock supplied to timer RJ0
0	Stops input clock supply. • SFRs used by timer RJ0 cannot be written. • Timer RJ0 is in the reset status.
1	Enables input clock supply. • SFRs used by timer RJ0 can be read and written.

Caution Be sure to clear bit 4 to 0.

5.3.7 Subsystem clock supply mode control register (OSMC)

The OSMC register can be used to select the operating clock for the real-time clock, interval timer, and timer RJ by using the WUTMMCK0 bit.

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

Reset signal generation clears this register to 00H.

Figure 5 - 10 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	0	0	0	WUTMMCK0	0	0	0	0

WUTMMCK0	Selection of operating clock for real-time clock, interval timer, and timer RJ
0	The operating clock (f_{RTC}) specified in the RTC clock select register (RTCCL) • The RTC operating clock is selected as the count clock for the real-time clock and the interval timer. • The low-speed on-chip oscillator cannot be selected as the count source for timer RJ.
1	• The low-speed on-chip oscillator clock is selected as the operation clock for the interval timer. • The low-speed on-chip oscillator can be selected as the count source for timer RJ.

Caution Be sure to set the WUTMMCK0 bit to 0 when using the real-time clock.

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

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

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

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

Figure 5 - 11 Format of High-speed on-chip oscillator frequency select register (HOCODIV)

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

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

HOCODIV2	HOCODIV1	HOCODIV0	Selection of high-speed on-chip oscillator clock frequency	
			24 MHz-based	32 MHz-based
			FRQSEL3 = 0	FRQSEL3 = 1
0	0	0	f_{IH} = 24 MHz	f_{IH} = 32 MHz
0	0	1	f_{IH} = 12 MHz	f_{IH} = 16 MHz
0	1	0	f_{IH} = 6 MHz	f_{IH} = 8 MHz
0	1	1	f_{IH} = 3 MHz	f_{IH} = 4 MHz
1	0	0	Setting prohibited	f_{IH} = 2 MHz
1	0	1	Setting prohibited	f_{IH} = 1 MHz
Other than above			Setting prohibited	

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

Operating Frequency Range	Operating Voltage Range
1 to 16 MHz	2.4 to 5.5 V
1 to 32 MHz	2.7 to 5.5 V

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**

Caution 4. When the frequency of the high-speed on-chip oscillator is changed with selecting PLL clock as system clock, the high-speed on-chip oscillator must be selected as system clock.

5.3.9 PLL control register (DSCCTL)

The DSCCTL register is used to control the operation of the PLL oscillator.
The DSCCTL register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 5 - 12 Format of PLL control register (DSCCTL)

Address: F02E5H After reset: 00H R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
DSCCTL	0	0	0	0	0	DSFRDIV	DSCM	DSCON

DSFRDIV	Selection of division factor for PLL reference clock[Note 1]
0	Undivided (f_{MX})
1	Divided by 2 ($f_{MX}/2$)

DSCM	Selection of multiplication factor for PLL[Note 2]
0	Multiplied by 12 (x6)
1	Multiplied by 16 (x8)

DSCON	Control of PLL oscillation and clock output
0	Stop
1	Starts oscillation and outputs the clock

Note 1. The high-speed system clock (f_{MX}) is used as the PLL reference clock.
Note 2. The actual multiplication factor is shown in parentheses because the PLL clock is divided by 2 at the final stage of the PLL oscillator.

Caution 1. When DSFRDIV and DSCM are changed, DSCON must be set to 0.
Caution 2. When PLL clock is selected as system clock, do not set DSCON to 0.
Caution 3. Be sure to clear bits 3 to 7 to “0”.

The following shows the combination of frequencies available when using PLL.

Figure 5 - 13 Relationship Between PLL and ΔΣ A/D Converter

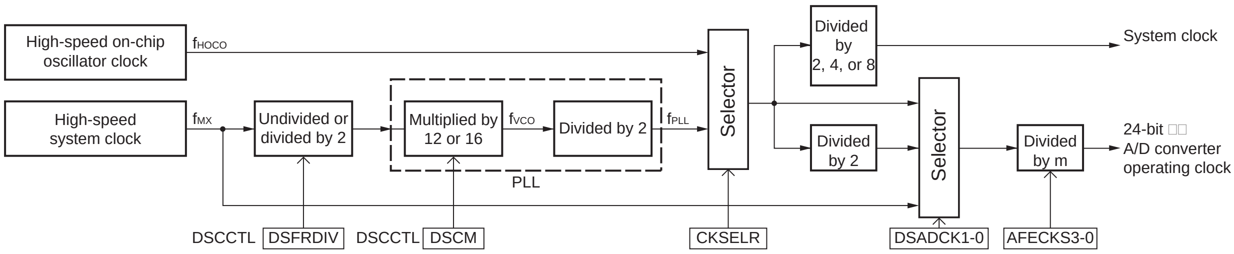


Table 5 - 3 PLL Clock Frequency Settings

High-speed System Clock (fMX)	Divided by (k): DSFRDIV	Multiplied by (n): DSCM	Frequency After Multiplied by n (fVCO)	PLL Clock (fPLL)
8 MHz	Undivided	12	96 MHz	48 MHz
	Undivided	16	128 MHz	64 MHz
	2	12	48 MHz	24 MHz
	2	16	64 MHz	32 MHz
4 MHz	Undivided	12	48 MHz	24 MHz
	Undivided	16	64 MHz	32 MHz
Other than above			Setting prohibited	

Caution When using the 24-bit ΔΣ A/D converter, divide fPLL so that its operating clock frequency becomes 4 MHz.

5.3.10 Main clock control register (MCKC)

The MCKC register is used to control the operation of the main clock.

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

Reset signal generation clears this register to 00H.

Figure 5 - 14 Format of Main clock control register (MCKC)

Address: F02E6H After reset: 00H R/W

Symbol	7	6	5	4	3	<2>	<1>	<0>
MCKC	0	0	0	0	0	RDIV1	RDIV0	CKSELR

RDIV1	RDIV0	Selection of PLL clock division ratio
0	0	Divided by 2
0	1	Divided by 4
1	0	Divided by 8
1	1	

CKSELR	Selection of high-speed on-chip oscillator clock/PLL clock
0	High-speed on-chip oscillator clock (f_{HOCO})
1	PLL clock (f_{PLL})

Caution 1. Both the PLL clock (f_{PLL}) and high-speed on-chip oscillator clock (f_{HOCO}) must be oscillating when they are switched.

Caution 2. Be sure to clear bits 3 to 7 to “0”.

Remark The clock selected by using the MCM0 bit when its value is 0 is selected as the main clock.

5.3.11 Peripheral clock control register (PCKC)

The PCKC register is used to select the operating clock for the 24-bit ΔΣ A/D converter (f_{DSAD}).
The PCKC register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 5 - 15 Format of Peripheral clock control register (PCKC)

Address: F02DEH After reset: 00H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
PCKC	0	0	0	0	0	0	DSADCK1	DSADCK0

DSADCK1	DSADCK0	Selection of 24-bit ΔΣ A/D converter operating clock (f_{DSAD})
0	0	High-speed on-chip oscillator clock (f_{HOCO}). Note 1
0	1	Clock obtained by dividing PLL clock (f_{PLL}) by 2. Note 2
1	0	High-speed system clock (f_{MX})
1	1	Setting prohibited

Note 1. Clear CKSELR of the MCKC register to 0.
Note 2. Set CKSELR of the MCKC register to 1.

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

5.4 System Clock Oscillator

5.4.1 X1 oscillator

The X1 oscillator oscillates with a crystal resonator or ceramic resonator (1 to 20 MHz) connected to the X1 and X2 pins.

An external clock can also be input. In this case, input the clock signal to the EXCLK pin.

To use the X1 oscillator, set bits 7 and 6 (EXCLK, OSCSEL) of the clock operation mode control register (CMC) as follows.

• Crystal or ceramic oscillation: EXCLK, OSCSEL = 0, 1

• External clock input: EXCLK, OSCSEL = 1, 1

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

When the pins are not used as input port pins, either, see **Table 2 - 3 Connection of Unused Pins**.

Figure 5 - 16 shows an example of the external circuit of the X1 oscillator.

Figure 5 - 16 Example of External Circuit of X1 Oscillator

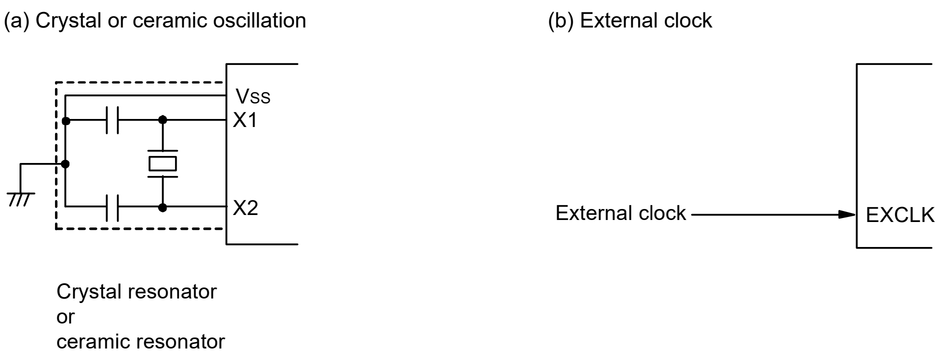
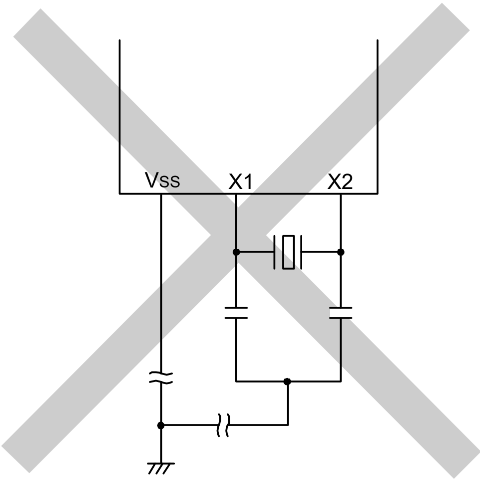


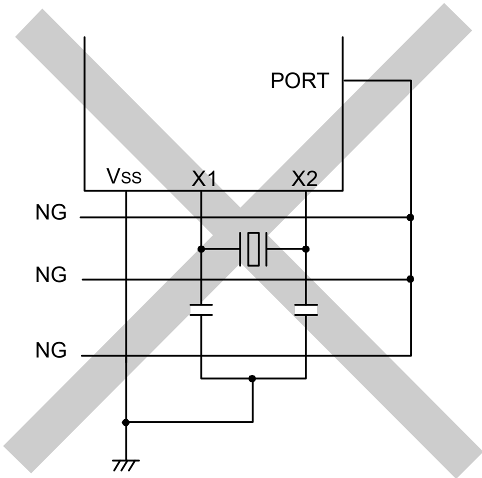
Figure 5 - 17 shows examples of incorrect resonator connection.

Figure 5 - 17 Examples of Incorrect Resonator Connection (1/2)

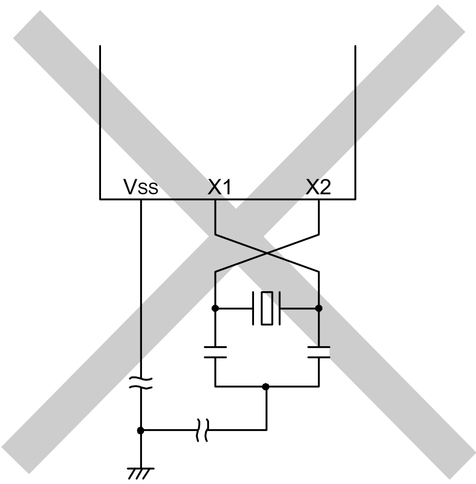
(a) Too long wiring



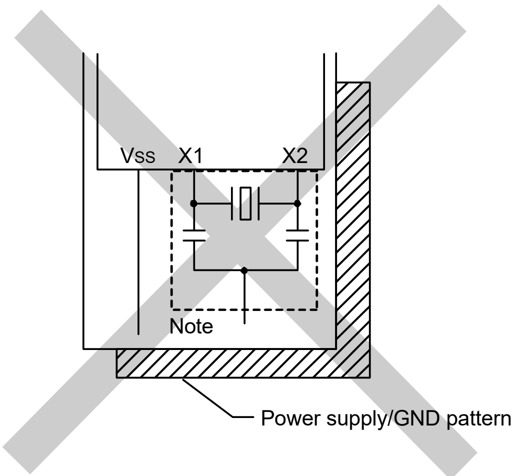
(b) Crossed signal line



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



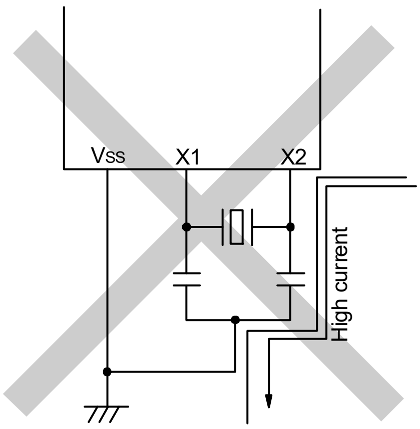
(d) A power supply/GND pattern exists under the X1 and X2 wires.



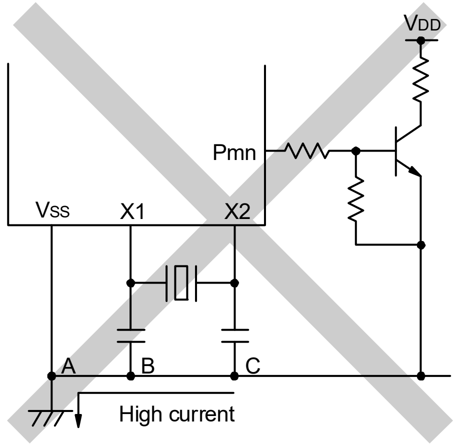
Note Do not place a power supply/GND pattern under the wiring section (section indicated by a broken line in the figure) of the X1 and X2 pins and the resonators in a multi-layer board or double-sided board.
Do not configure a layout that will cause capacitance elements and affect the oscillation characteristics.

Figure 5 - 18 Examples of Incorrect Resonator Connection (2/2)

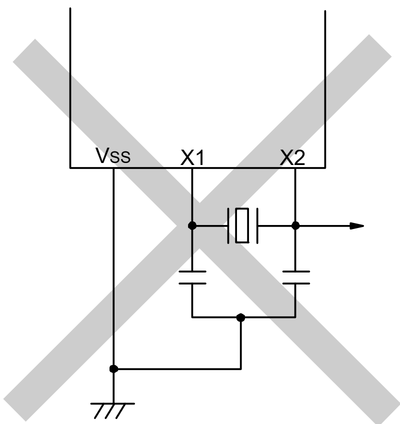
(e) Wiring near high alternating current



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



(g) Signals are fetched



5.4.2 High-speed on-chip oscillator

The high-speed on-chip oscillator is incorporated in the RL78/I1E. The frequency can be selected from among 32, 24, 16, 12, 8, 6, 4, 3, 2, or 1 MHz by using the option byte (000C2H). Oscillation can be controlled by bit 0 (HIOSTOP) of the clock operation status control register (CSC).

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

5.4.3 Low-speed on-chip oscillator

The low-speed on-chip oscillator is incorporated in the RL78/I1E.

The low-speed on-chip oscillator clock is used only as the interval timer and timer RJ clock. The low-speed on-chip oscillator clock cannot be used as the CPU clock.

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

Unless the watchdog timer is stopped and WUTMMCK0 is a value other than zero, oscillation of the low-speed on-chip oscillator continues. Note that only when the watchdog timer is operating and the WUTMMCK0 bit is 0, oscillation of the low-speed on-chip oscillator will stop while the WDSTBYON bit is 0 and operation is in the HALT, STOP, or SNOOZE mode. While the watchdog timer operates, the low-speed on-chip oscillator clock does not stop even if the program freezes.

5.4.4 PLL (Phase Locked Loop)

A PLL circuit is incorporated in the RL78/I1E.

The PLL circuit can be used to multiply the frequency of the high-speed system clock.

The operation can be controlled by using bit 0 (DSCON) of the PLL control register (DSCCTL).

Caution When switching the PLL clock to the high-speed on-chip oscillator clock or high-speed system clock, stop the functions for which the PLL output clock (f_{PLL}) is supplied, such as the 24-bit ΔΣ A/D converter

.

5.5 Clock Generator Operation

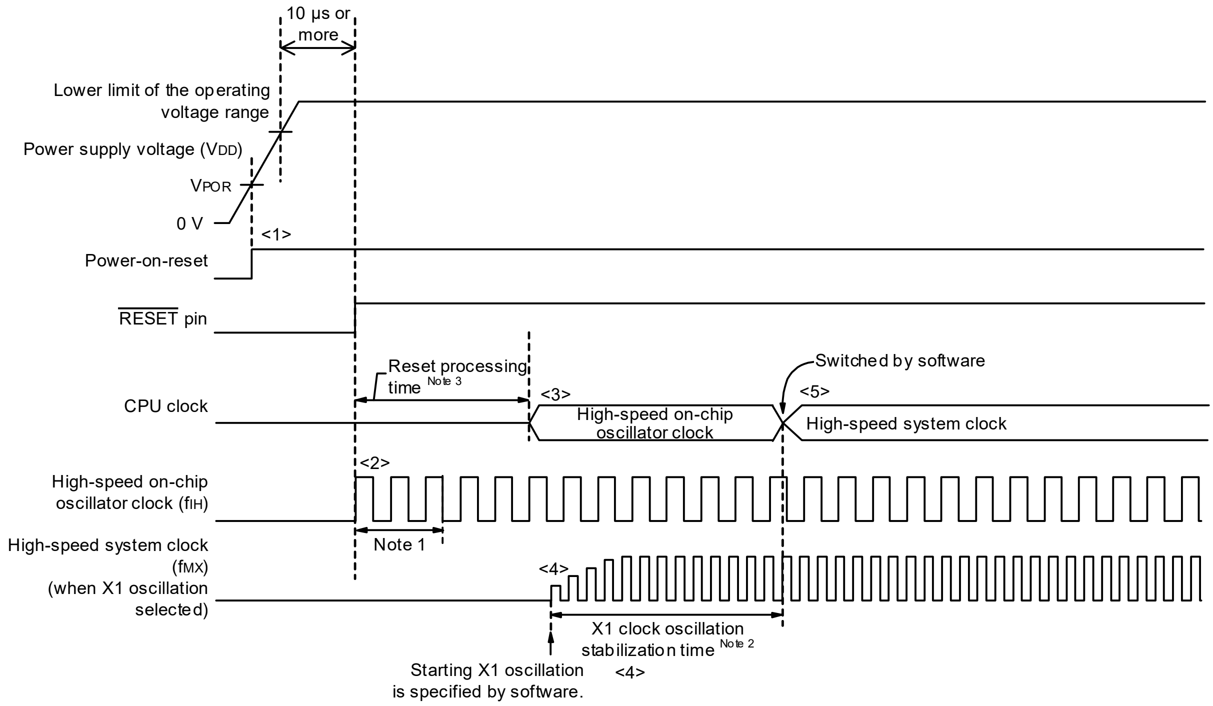
The clock generator generates the following clocks and controls the operation modes of the CPU, such as standby mode (see **Figure 5 - 1**).

- Main system clock f_{MAIN}
  - High-speed system clock f_{MX}
    X1 clock f_X
    External main system clock f_{EX}
  - High-speed on-chip oscillator clock f_{IH} (= f_{HOCO}/n; n = 2, 4, 8) Note
  - PLL clock f_{IH} (= f_{PLL}/n; n = 2, 4, 8) Note
- Low-speed on-chip oscillator clock f_{IL}
- CPU/peripheral hardware clock f_{CLK}

Note f_{IH} is the frequency of the main system clock source frequency generated by dividing the high-speed on-chip oscillator clock frequency or PLL clock frequency by 2, 4, or 8.

The CPU starts operation when the high-speed on-chip oscillator starts outputting after a reset release in the RL78/I1E.
When the power supply voltage is turned on, the clock generator operation is shown in Figure 5 - 19.

Figure 5 - 19 Clock Generator Operation When Power Supply Voltage Is Turned On



<1> When the power is turned on, an internal reset signal is generated by the power-on-reset (POR) circuit.
Note that the reset state is maintained after a reset by the voltage detector or an external reset until the voltage reaches the range of operating voltage described in **33.4** or **34.4** AC Characteristics (the above figure is an example when the external reset is in use).

<2> When the reset is released, the high-speed on-chip oscillator automatically starts oscillating.

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

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

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

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

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

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

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

5.6 Controlling Clock

5.6.1 Example of setting high-speed on-chip oscillator

After a reset release, the CPU/peripheral hardware clock (f_{CLK}) always starts operating with the high-speed on-chip oscillator clock. The frequency of the high-speed on-chip oscillator can be selected from 32, 24, 16, 12, 8, 6, 4, 3, 2, and 1 MHz by using FRQSEL0 to FRQSEL3 of the option byte (000C2H). In addition, Oscillation can be changed by the high-speed on-chip oscillator frequency select register (HOCODIV).

[Option byte setting]

Address: 000C2H

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

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator
1	0	0	0	32 MHz
0	0	0	0	24 MHz
1	0	0	1	16 MHz
0	0	0	1	12 MHz
1	0	1	0	8 MHz
0	0	1	0	6 MHz
1	0	1	1	4 MHz
0	0	1	1	3 MHz
1	1	0	0	2 MHz
1	1	0	1	1 MHz
Other than above				Setting prohibited

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

Address: F00A8H

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

HOCODIV2	HOCODIV1	HOCODIV0	Selection of high-speed on-chip oscillator clock frequency	
			24 MHz-based	32 MHz-based
			FRQSEL3 = 0	FRQSEL3 = 1
0	0	0	f_{IH} = 24 MHz	f_{IH} = 32 MHz
0	0	1	f_{IH} = 12 MHz	f_{IH} = 16 MHz
0	1	0	f_{IH} = 6 MHz	f_{IH} = 8 MHz
0	1	1	f_{IH} = 3 MHz	f_{IH} = 4 MHz
1	0	0	Setting prohibited	f_{IH} = 2 MHz
1	0	1	Setting prohibited	f_{IH} = 1 MHz
Other than above			Setting prohibited	

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 oscillation clock, set the oscillator and start oscillation by using the oscillation stabilization time select register (OSTS), clock operation mode control register (CMC), and clock operation status control register (CSC) and wait for oscillation to stabilize by using the oscillation stabilization time counter status register (OSTC). After the oscillation stabilizes, set the X1 oscillation clock to f_{CLK} by using the system clock control register (CKC).

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

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

	7	6	5	4	3	2	1	0
CMC	EXCLK 0	OSCSEL 1	0	0	0	0	0	AMPH 0/1

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

Example: Setting values when a wait of at least 102 μs is set based on a 10 MHz resonator.

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

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

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

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

Example: Wait until the bits reach the following values when a wait of at least 102 μs is set based on a 10 MHz resonator.

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

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

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

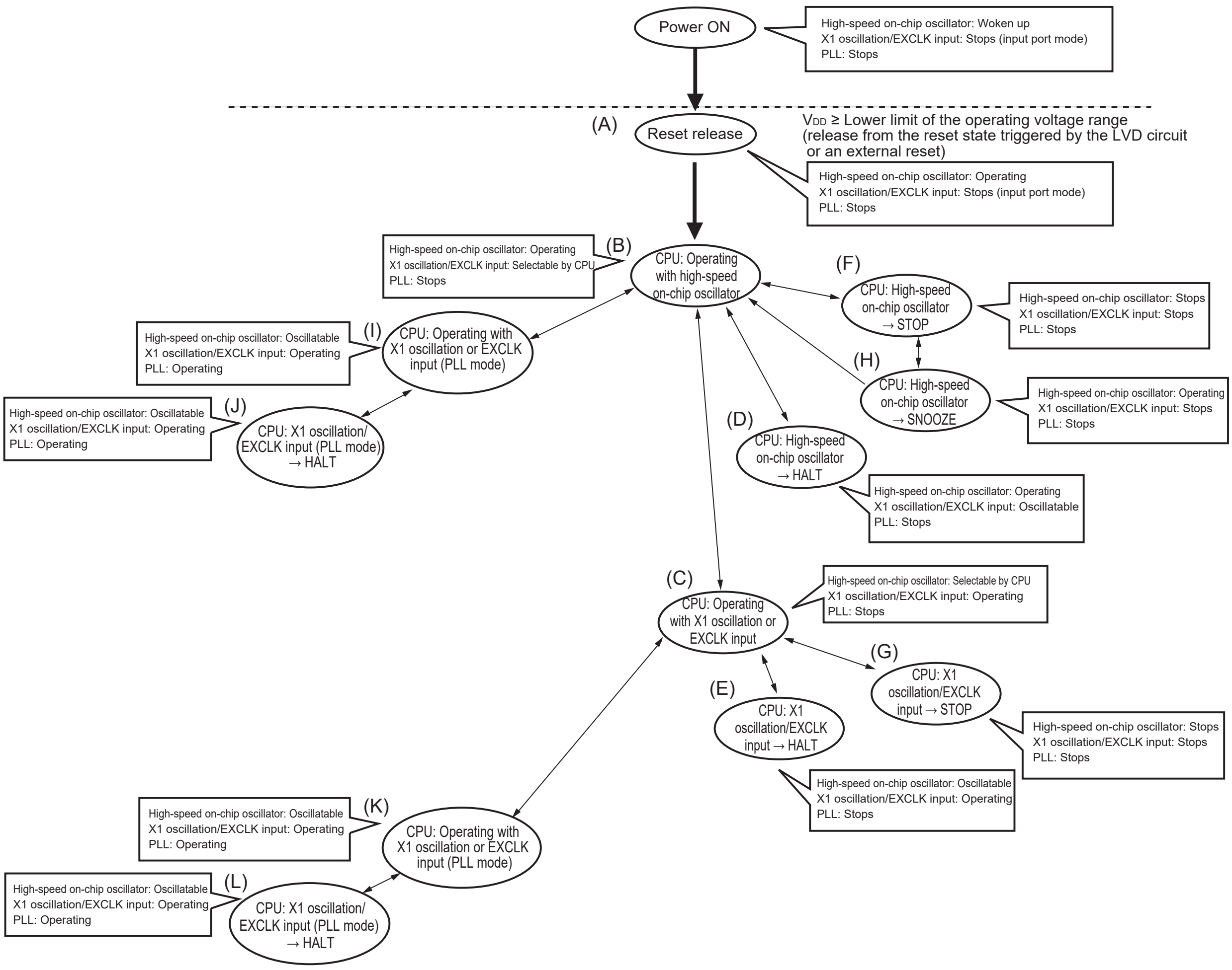
Caution Set the HOCODIV register within the operable voltage range before and after the frequency change.

Operating Frequency Range	Operating Voltage Range
1 to 16 MHz	2.4 to 5.5 V
1 to 32 MHz	2.7 to 5.5 V

5.6.3 CPU clock status transition diagram

Figure 5 - 20 shows the CPU clock status transition diagram of this product.

Figure 5 - 20 CPU Clock Status Transition Diagram



Tables 5 - 4 to 5 - 8 show transition of the CPU clock and examples of setting the SFR registers.

Table 5 - 4 CPU Clock Transition and SFR Register Setting Examples (1/5)

(1) CPU operating with high-speed on-chip oscillator clock (B) after reset release (A)

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

(2) CPU operating with high-speed system clock (C) after 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 ≤ fx ≤ 10 MHz)	0	1	0	Note 2	0	Must be checked	1
(A) → (B) → (C) (X1 clock: 10 MHz < fx ≤ 20 MHz)	0	1	1	Note 2	0	Must be checked	1
(A) → (B) → (C) (external main clock)	1	1	×	Note 2	0	Must not 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.

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 or CHAPTER 33 or CHAPTER 34 ELECTRICAL SPECIFICATIONS).

Remark 1. ×: Don't care

Remark 2. (A) to (L) in Tables 5 - 4 to 5 - 8 correspond to (A) to (L) in Figure 5 - 20.

Table 5 - 5 CPU Clock Transition and SFR Register Setting Examples (2/5)

(3) 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 ≤ fx ≤ 10 MHz)	0	1	0	Note 2	0	Must be checked	1
(B) → (C) (X1 clock: 10 MHz < fx ≤ 20 MHz)	0	1	1	Note 2	0	Must be checked	1
(B) → (C) (external main clock)	1	1	×	Note 2	0	Need not be checked	1

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

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

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

Note 2. Set the oscillation stabilization time as follows.

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

Caution **Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 33 or CHAPTER 34 ELECTRICAL SPECIFICATIONS).**

(4) 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	18 μs to 65 μs	0

CSC Register and Oscillation accuracy stabilization time: Unnecessary if the CPU is operating with the high-speed on-chip oscillator clock

Remark 1. ×: Don't care

Remark 2. (A) to (L) in Tables 5 - 4 to 5 - 8 correspond to (A) to (L) in Figure 5 - 20.

Remark 3. The oscillation accuracy stabilization time changes according to the temperature conditions and the STOP mode period.

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

(5) • CPU clock changing from high-speed on-chip oscillator clock (B) to high-speed system clock in PLL mode (I)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register Note 1			OSTS Register	CSC Register	OSTC Register	DSCCTL Register		MCKC Register		DSCCTL Register	Waiting for Oscillation Stabilization	MCKC Register
	EXCLK	OSCSEL	AMPH		MSTOP		DSFRDIV	DSCM	RDIV1	RDIV0	DSCON		CKSELR
(B) → (I) divided by 2	0/1	1	0	Note 2	0	Must be checked	0/1	0/1	0	0	1	40 μs	1
(B) → (I) divided by 4	0/1	1	0	Note 2	0	Must be checked	0/1	0/1	0	1	1		1
(B) → (I) divided by 8	0/1	1	0	Note 2	0	Must be checked	0/1	0/1	1	0	1		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.

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 The clock switching takes max.25 clocks after setting CKSELR = 1. Do not stop the high-speed on-chip oscillator until completion of the switching.

Remark (A) to (L) in Tables 5 - 4 to 5 - 8 correspond to (A) to (L) in Figure 5 - 20.

(6) • CPU clock changing from high-speed system clock in PLL mode (I) to high-speed on-chip oscillator clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	MCKC Register	Waiting for Clock Switching	DSCCTL Register
	CKSELR		DSCON
(I) → (B)	0	256 clocks	0

(7) • CPU clock changing from high-speed system clock (C) to high-speed system clock in PLL mode (K)

((Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CSC Register	DSCCTL Register		MCKC Register		DSCCTL Register	MCKC Register	Waiting for Oscillation Stabilization	CSC Register	CKC Register
	HIOSTOP	DSFRDIV	DSCM	RDIV1	RDIV0	DSCON	CKSELR		HIOSTOP	MCM0
(C) → (K) divided by 2	0 Note 1	0/1	0/1	0	0	1	1 Note 1	65 μs Note 2	1 Note 1	0
(C) → (K) divided by 4	0 Note 1	0/1	0/1	0	1	1	1 Note 1		1 Note 1	0
(C) → (K) divided by 8	0 Note 1	0/1	0/1	1	0	1	1 Note 1		1 Note 1	0

Note 1. When the clock is switched to PLL at CKSELR = 1, the setting is not needed.
The high-speed on-chip oscillator must be operated to set CKSELR = 1.

Note 2. When HIOSTOP = 0 is not set, it waits 40 μs for oscillation stabilization.

Remark (A) to (L) in Tables 5 - 4 to 5 - 8 correspond to (A) to (L) in Figure 5 - 20.

Table 5 - 7 CPU Clock Transition and SFR Register Setting Examples (4/5)

(8) CPU clock changing from high-speed system clock in PLL mode (K) to high-speed system clock (C)

Setting Flag of SFR Register / Status Transition	CKC Register MCM0	Waiting for Clock Switching	DSCCTL Register DSCON
(K) → (C) divided by 2 (RDIV1,0 = 00) High-speed system clock (f_{MX}) = 8 MHz	1	8 clocks	0
(K) → (C) divided by 2 (RDIV1,0 = 00) High-speed system clock (f_{MX}) = 4 MHz		8 clocks	
(K) → (C) divided by 4 (RDIV1,0 = 01) High-speed system clock (f_{MX}) = 8 MHz		4 clocks	
(K) → (C) divided by 4 (RDIV1,0 = 01) High-speed system clock (f_{MX}) = 4 MHz		4 clocks	
(K) → (C) divided by 8 (RDIV1,0 = 10) High-speed system clock (f_{MX}) = 8 MHz		2 clocks	
(K) → (C) divided by 8 (RDIV1,0 = 10) High-speed system clock (f_{MX}) = 4 MHz		2 clocks	

Remark (A) to (L) in Tables 5 - 4 to 5 - 8 correspond to (A) to (L) in Figure 5 - 20.

(9)
- HALT mode (D) set while CPU is operating with high-speed on-chip oscillator clock (B)
- HALT mode (E) set while CPU is operating with high-speed system clock (C)
- HALT mode (J) set while CPU is operating with high-speed system clock in PLL mode (I)
- HALT mode (L) set while CPU is operating with high-speed system clock in PLL mode (K)

Status Transition	Setting
(B) → (D) (C) → (E) (I) → (J) (K) → (L)	Executing HALT instruction

Remark (A) to (L) in Tables 5 - 4 to 5 - 8 correspond to (A) to (L) in Figure 5 - 20.

Table 5 - 8 CPU Clock Transition and SFR Register Setting Examples (5/5)

(10) • STOP mode (F) set while CPU is operating with high-speed on-chip oscillator clock (B)
• STOP mode (G) set while CPU is operating with high-speed system clock (C)

(Setting sequence) →

<table>
<tr><th colspan="2">Status Transition</th><th colspan="3">Setting</th></tr>
<tr><td colspan="2">(B) → (F)</td><td rowspan="3">Stopping peripheral functions that cannot operate in STOP mode</td><td>—</td><td rowspan="3">Executing STOP instruction</td></tr>
<tr><td rowspan="2">(C) → (G)</td><td>In X1 oscillation</td><td>Sets the OSTS register</td></tr>
<tr><td>External main system clock</td><td>—</td></tr>
</table>

(11) CPU changing from STOP mode (F) to SNOOZE mode (H)
For details about the setting for switching from the STOP mode to the SNOOZE mode, see **16.8 SNOOZE Mode Function**, **19.5.7 SNOOZE mode function**, and **19.7.3 SNOOZE mode function**.

(12) • STOP mode (F) set while CPU is operating with high-speed on-chip oscillator clock (B)
• STOP mode (F) set while CPU is operating with high-speed system clock in PLL mode (I)
• STOP mode (G) set while CPU is operating with high-speed system clock in PLL mode (K)

Switch the PLL clock operation to high-speed on-chip oscillator clock (see Table 5 - 6 (**6**)) or high-speed system clock operation (see Table 5 - 7 (**8**)), stop PLL (DSCON = 0), and then execute the STOP instruction.

Remark (A) to (L) in Tables 5 - 4 to 5 - 8 correspond to (A) to (L) in Figure 5 - 20.

5.6.4 Conditions before changing CPU clock and processing after changing CPU clock

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

Table 5 - 9 Changing CPU Clock(1/2)

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 • After elapse of oscillation stabilization time	After confirming that the CPU clock has changed from the high-speed on-chip oscillator clock to the X1 clock or external main system clock, operating current can be reduced by stopping the high-speed on-chip oscillator (HIOSTOP = 1).
	External main system clock	Enabling input of external clock from the EXCLK pin • OSCSEL = 1, EXCLK = 1, MSTOP = 0	
	PLL clock	Stabilization of X1 oscillation • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time Enabling input of external clock from the EXCLK pin • OSCSEL = 1, EXCLK = 1, MSTOP = 0 Oscillation of PLL • DSCON = 1	—
X1 clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • After elapse of oscillation stabilization time	After confirming that the CPU clock has changed from the X1 clock to the high-speed on-chip oscillator clock, X1 oscillation can be stopped (MSTOP = 1).
	External main system clock	Transition not possible	—
	PLL clock	Oscillation of PLL • DSCON = 1 Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • After elapse of oscillation stabilization time	—
External main system clock	High-speed on-chip oscillator clock	Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • After elapse of oscillation stabilization time	After confirming that the CPU clock has changed from the external main system clock to the high-speed on-chip oscillator clock, external main system clock input can be disabled (MSTOP = 1).
	X1 clock	Transition not possible	—
	PLL clock	Oscillation of PLL • DSCON = 1 Enabling oscillation of high-speed on-chip oscillator • HIOSTOP = 0 • After elapse of oscillation stabilization time	—

Table 5 - 10 Changing CPU Clock(2/2)

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
PLL clock	High-speed on-chip oscillator clock	Oscillation of high-speed on-chip oscillator • HIOSTOP = 0	After confirming that the CPU clock has changed from the PLL clock to the high-speed on-chip oscillator clock, X1 clock, or external main system clock, operating current can be reduced by stopping the PLL (DSCON = 1).
	X1 clock	Stabilization of X1 oscillation • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time	
	External main system clock	Enabling input of external clock from the EXCLK pin • OSCSEL = 1, EXCLK = 1, MSTOP = 0	

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

By setting bit 4 (MCM0) of the system clock control register (CKC), the main system clock can be switched (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 **Tables 5 - 11** and **5 - 12**).

Whether the CPU is operating on the main system clock or the high-speed on-chip oscillator clock can be checked by 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 clock is also switched.

Table 5 - 11 Maximum Time Required for Main System Clock Switchover

Clock A	Switching directions	Clock B	Remark
f_{IH}	←→	f_{MX}	See **Table 5 - 12**

Table 5 - 12 Maximum Number of Clock Cycles Required for $f_{IH} \leftrightarrow f_{MX}$

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

Remark 1. The number of clock cycles listed in Tables 5 - 12 is the number of CPU clock cycles before switchover.

Remark 2. Calculate the number of clock cycles in Tables 5 - 12 by rounding up the number after the decimal position.

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, f_{MX} = 10 MHz)

2 f_{MX}/f_{IH} = 2 (10/8) = 2.5 → 3 clock cycles

5.6.6 Conditions before clock oscillation is stopped

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

When stopping the clock oscillation, confirm the conditions before clock oscillation is stopped.

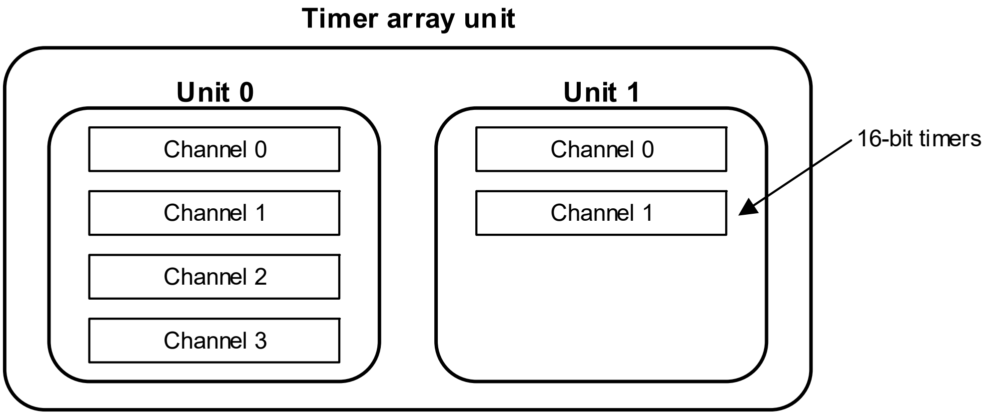
Table 5 - 13 Conditions Before the Clock Oscillation Is Stopped and Flag Settings

Clock	Conditions Before Clock Oscillation Is Stopped (External Clock Input Disabled)	SFR Flag Setting
High-speed on-chip oscillator clock	MCS = 1 (The CPU is operating on a clock other than the high-speed on-chip oscillator clock.)	HIOSTOP = 1
X1 clock	MCS = 0 (The CPU is operating on a clock other than the high-speed system clock.)	MSTOP = 1
External main system clock		

CHAPTER 6 TIMER ARRAY UNIT

The timer array unit has four and two 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 • Square wave output • External event counter • Input pulse interval measurement • Measurement of high-/low-level width of input signal • Delay counter	• One-shot pulse output • PWM output • Multiple PWM output

It is possible to use the 16-bit timer of channels 1 and 3 of unit 0 as two 8-bit timers (higher and lower). The following functions become available by using channels 1 and 3 of unit 0 as 8-bit timers:

• 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)

Channel 3 of unit 0 can be used for LIN-bus communication operating in combination with UART0 of the serial array unit.

6.1 Functions of Timer Array Unit

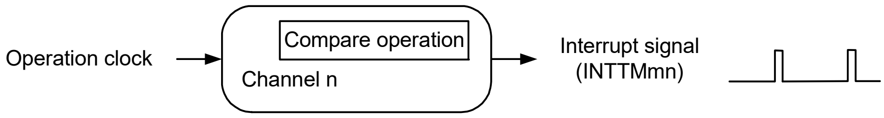
Timer array unit has the following functions.

6.1.1 Independent channel operation function

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

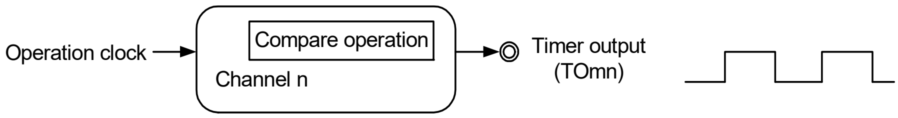
(1) Interval timer

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



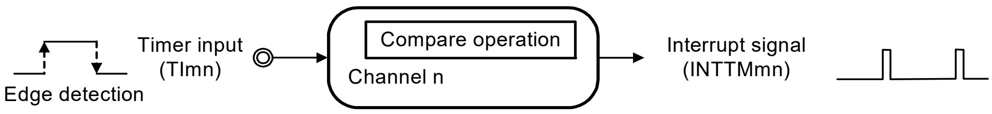
(2) Square wave output

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



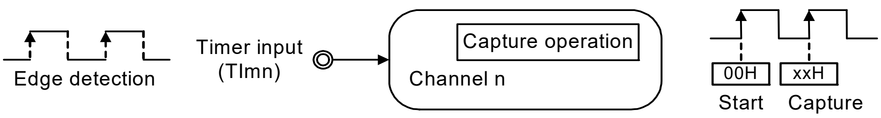
(3) External event counter

Each timer of a unit can be used as an event counter that generates an interrupt when the number of the valid edges of a signal input to the timer input pin (TImn) has reached a specific value.



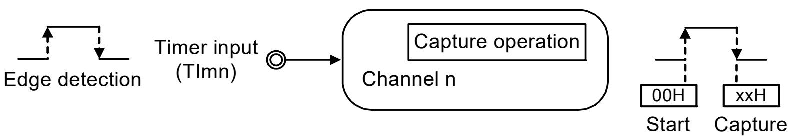
(4) Input pulse interval measurement

Counting starts at detection of 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 is measured.



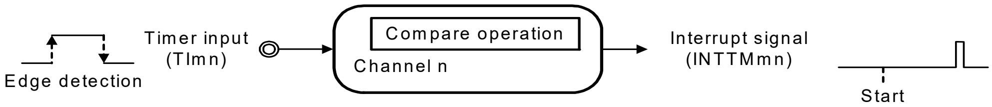
(5) Measurement of high-/low-level width of input signal

Counting starts at detection of a single edge of the signal input to the timer input pin (TImn), and the count value is captured at detection of the other edge. In this way, the high-level or low-level width of the input signal is measured.



(6) Delay counter

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



Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

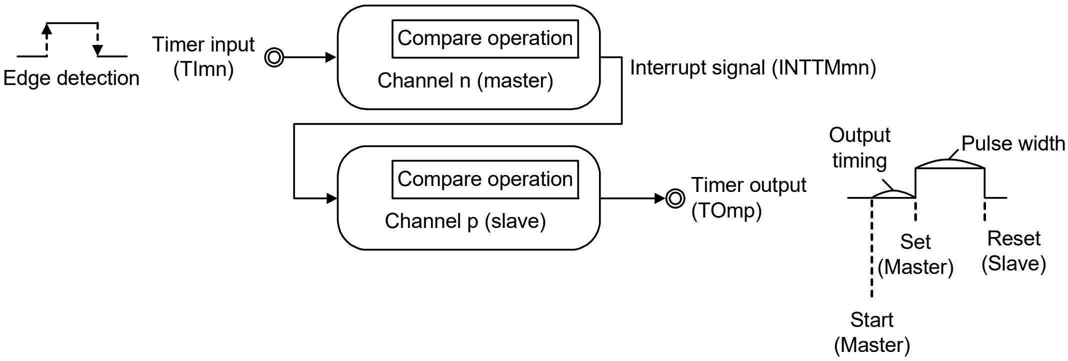
Remark 2. The presence or absence of timer I/O pins of channel 0 to 3 depends on the product. See **Table 6 - 2 Timer I/O Pins Provided in Each Product** for details.

6.1.2 Simultaneous channel operation function

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

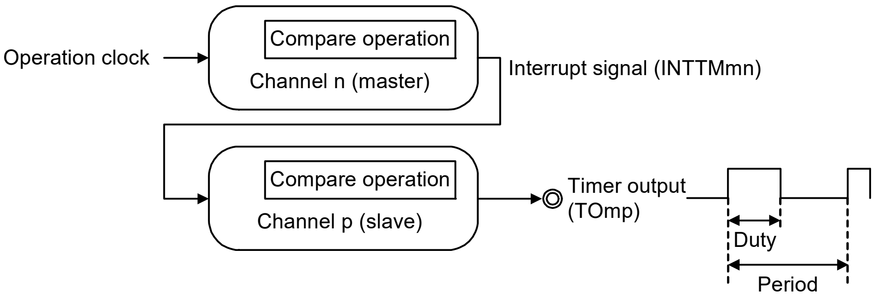
(1) One-shot pulse output

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



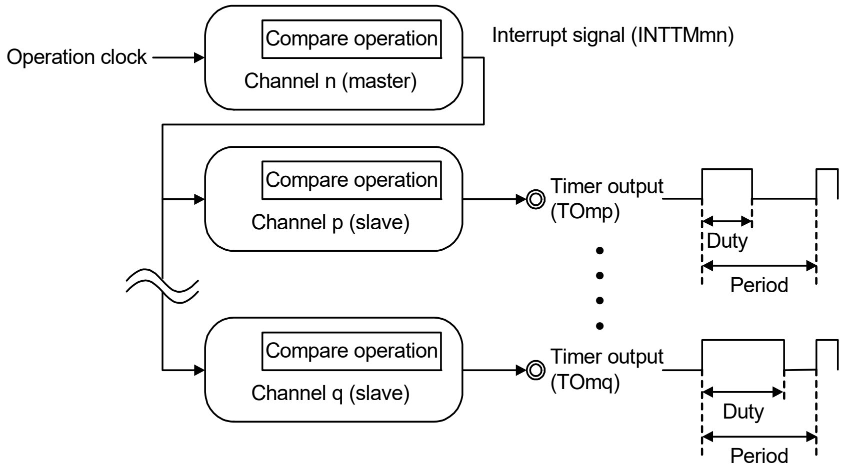
(2) PWM (Pulse Width Modulation) output

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



(3) Multiple PWM (Pulse Width Modulation) output

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



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

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3),
p, q: Slave channel number (n < p < q ≤ 3)

6.1.3 8-bit timer operation function (available for channels 1 and 3 of unit 0)

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 of unit 0.

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

6.1.4 LIN-bus supporting function (available for channel 3 of unit 0)

Timer array unit is used to check whether signals received in LIN-bus communication match the LIN-bus communication format.

(1) Detection of wakeup signal
The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD0) of UART0 and the count value of the timer is captured at the rising edge. In this way, a low-level width can be measured. If the low-level width is greater than a specific value, it is recognized as a wakeup signal.

(2) Detection of break field
The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD0) of UART0 after a wakeup signal is detected, and the count value of the timer is captured at the rising edge. In this way, a low-level width is measured. If the low-level width is greater than a specific value, it is recognized as a break field.

(3) Measurement of pulse width of sync field
After a break field is detected, the low-level width and high-level width of the signal input to the serial data input pin (RxD0) of UART0 are measured. From the bit interval of the sync field measured in this way, a baud rate is calculated.

Remark For details about setting up the operations used to implement the LIN-bus, see **6.3.13 Input switch control register (ISC)** and **6.8.4 Operation as input signal high-/low-level width measurement**.

6.2 Configuration of Timer Array Unit

Timer array unit includes the following hardware.

Table 6 - 1 Configuration of Timer Array Unit

Item	Configuration
Timer/counter	Timer counter register mn (TCRmn)
Register	Timer data register mn (TDRmn)
Timer input	TI00 to TI03, TI10, and TI11 pins, and RxD0 pin (for LIN-bus)
Timer output	TO00 to TO03, TO10, and TO11 pins, output controller
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Timer clock select register m (TPSm) • Timer channel enable status register m (TEm) • Timer channel start register m (TSm) • Timer channel stop register m (TTm) • Timer input select register 0 (TIS0) • Timer output enable register m (TOEm) • Timer output register m (TOm) • Timer output level register m (TOLm) • Timer output mode register m (TOMm)
	<Registers of each channel> • Timer mode register mn (TMRmn) • Timer status register mn (TSRmn) • Input switch control register (ISC) • Noise filter enable registers 1, 2 (NFEN1, NFEN2) • Port mode register 1 (PM1) • Port register 1 (P1)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

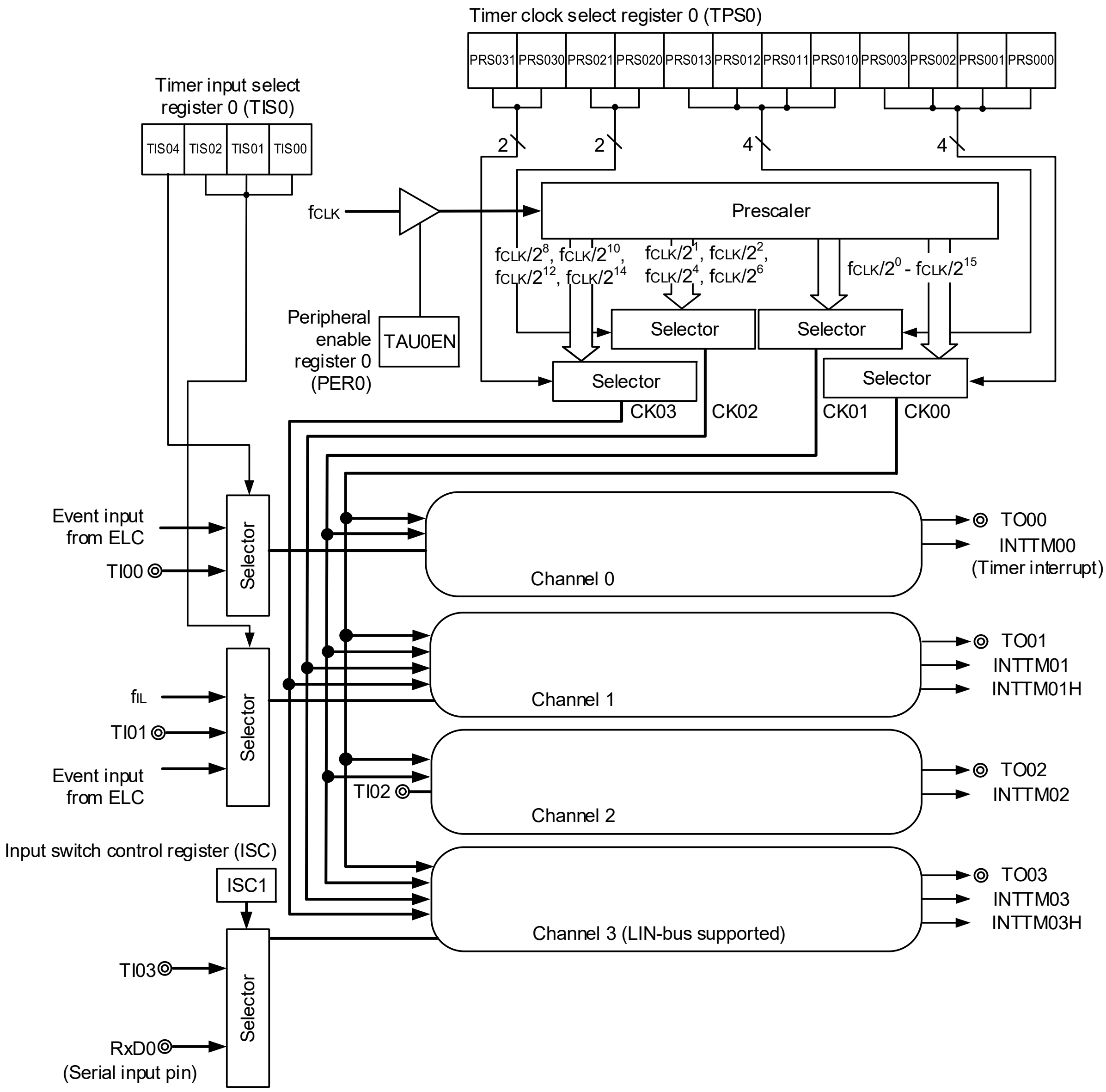
Table 6 - 2 Timer I/O Pins Provided in Each Product

Timer array unit channels		I/O pins of each product (32-pin and 36-pin products)
Unit 0	Channel 0	TI00/TO00
	Channel 1	TI01/TO01
	Channel 2	TI02/TO02
	Channel 3	TI03/TO03
Unit 1	Channel 0	TI10/TO10
	Channel 1	TI11/TO11

Remark The timer input and timer output functions are shared by the same pin, so only one of them can be used at a time.

Figures 6 - 1 to 6 - 8 show the block diagrams of the timer array unit.

Figure 6 - 1 Entire Configuration of Timer Array Unit 0



Remark fIL: Low-speed on-chip oscillator clock frequency

Figure 6 - 2 Internal Block Diagram of Channel 0 of Timer Array Unit 0

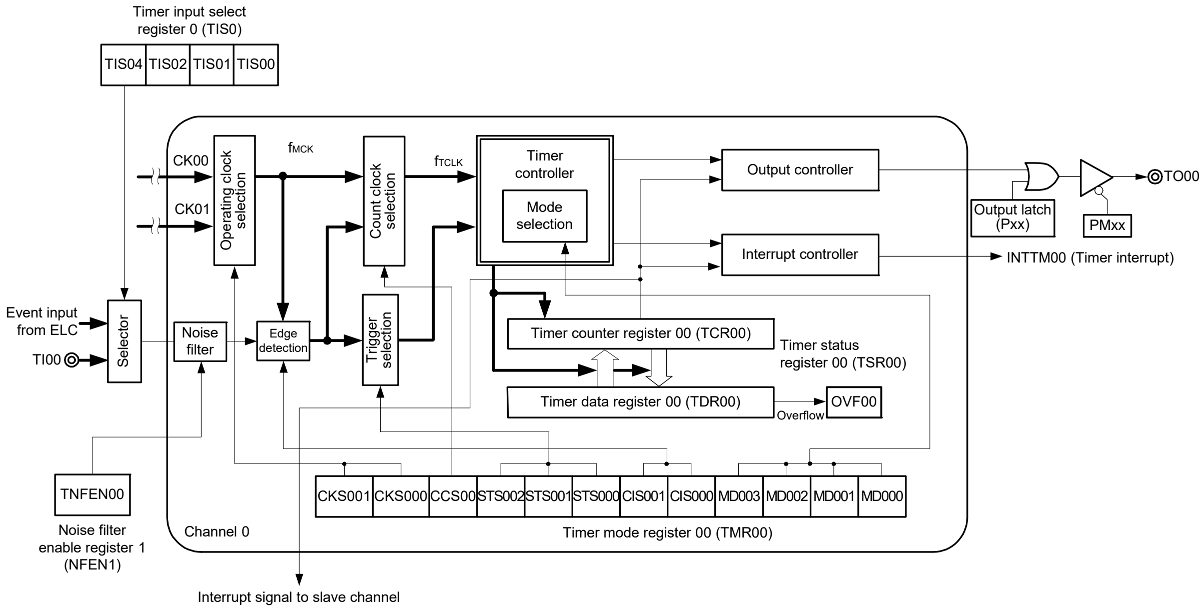


Figure 6 - 3 Internal Block Diagram of Channel 1 of Timer Array Unit 0

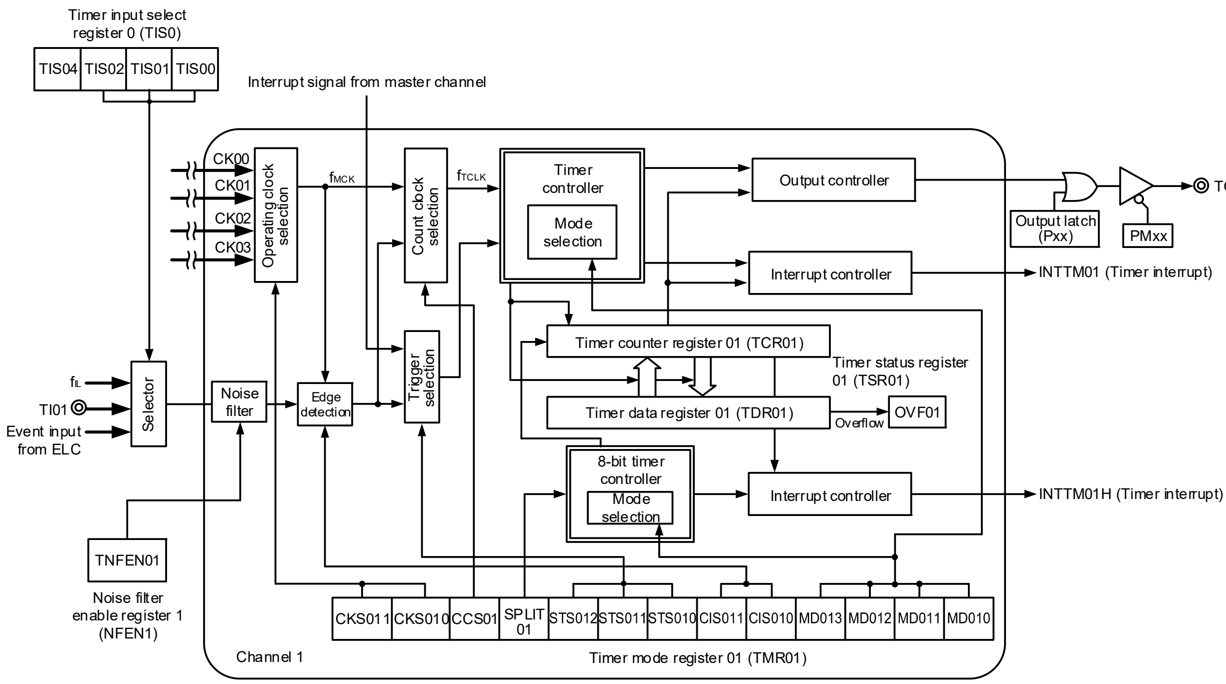


Figure 6 - 4 Internal Block Diagram of Channel 2 of Timer Array Unit 0

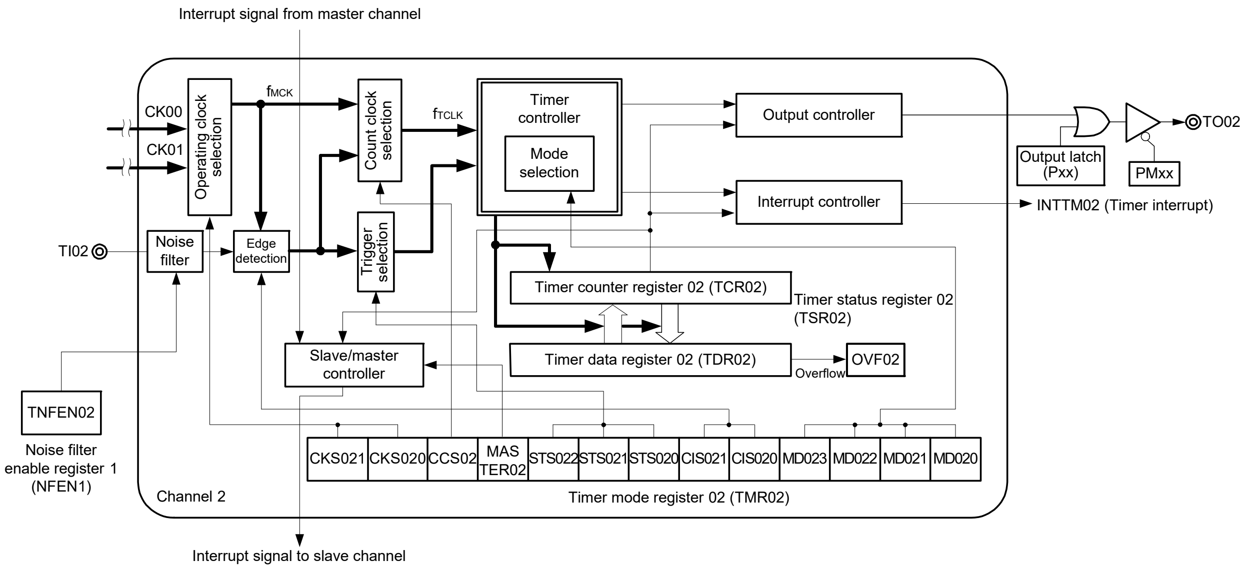


Figure 6 - 5 Internal Block Diagram of Channel 3 of Timer Array Unit 0

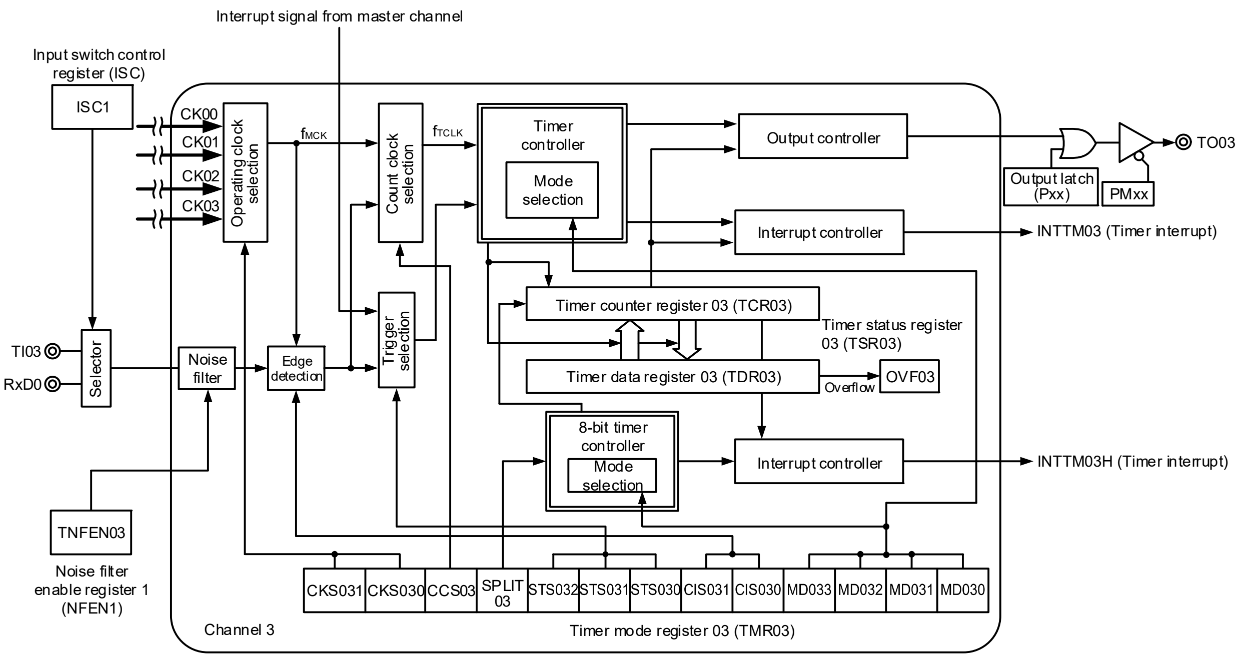


Figure 6 - 6 Entire Configuration of Timer Array Unit 1

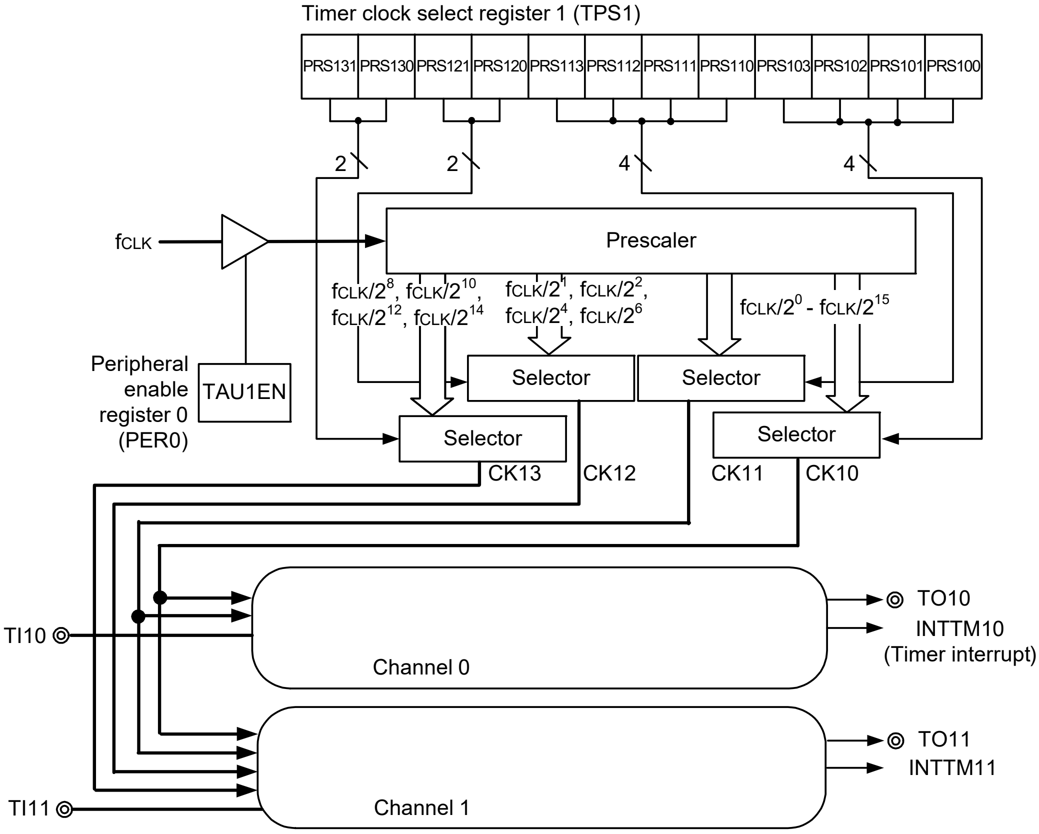


Figure 6 - 7 Internal Block Diagram of Channel 0 of Timer Array Unit 1

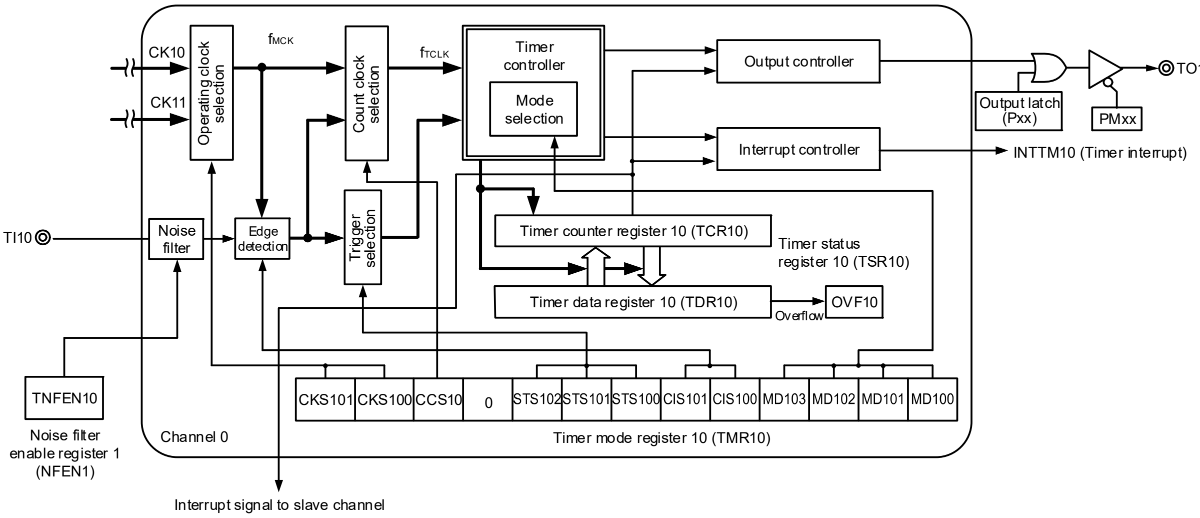
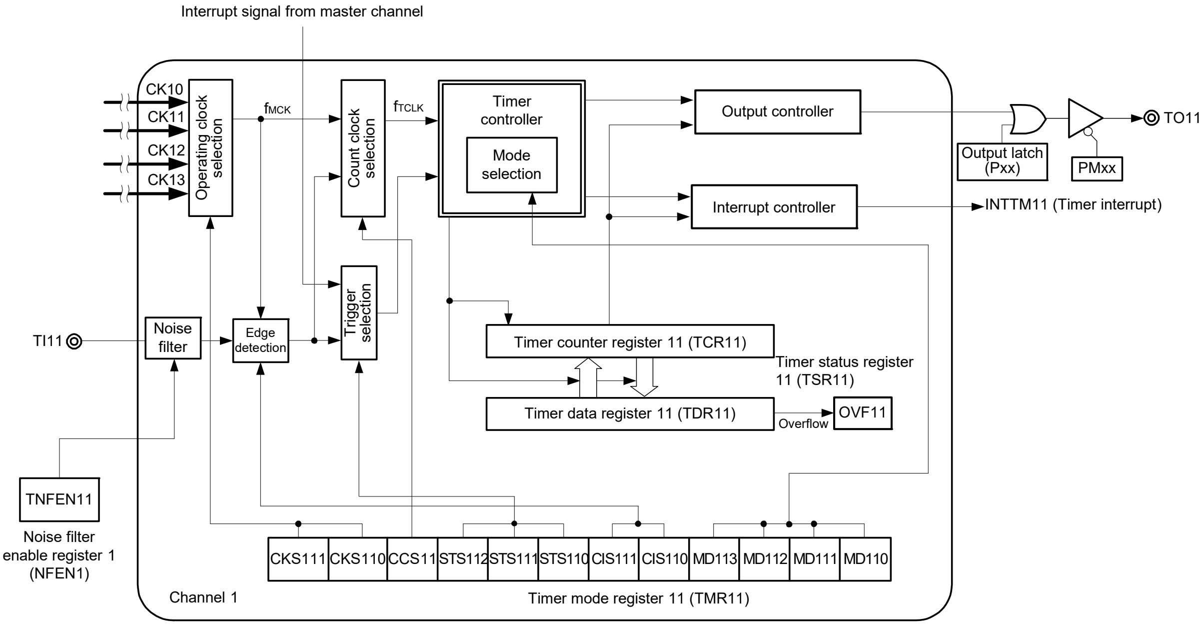


Figure 6 - 8 Internal Block Diagram of Channel 1 of Timer Array Unit 1



6.2.1 Timer counter register mn (TCRmn)

The TCRmn register is a 16-bit read-only register that is used to count the number of count clock cycles.
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 - 9 Format of Timer counter register mn (TCRmn)

Address: F0180H, F0181H (TCR00) to F0186H, F0187H (TCR03) After reset: FFFFH R
F01C0H, F01C1H (TCR10), F01C2H, F01C3H (TCR11)

	F0181H (TCR00)								F0180H (TCR00)							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCRmn																

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

The count value can be read by reading timer count register mn (TCRmn).
The count value is set to FFFFH in the following cases.

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

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

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

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

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

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

Operation Mode	Count Mode	Timer count register mn (TCRmn) Read Value Note			
		Value if the operation mode was changed after releasing reset	Value if the Operation was restarted after count operation paused (TTmn = 1)	Value if the operation mode was changed after count operation paused (TTmn = 1)	Value when waiting for a start trigger after one count
Interval timer mode	Count down	FFFFH	Value if stop	Undefined	—
Capture mode	Count up	0000H	Value if stop	Undefined	—
Event counter mode	Count down	FFFFH	Value if stop	Undefined	—
One-count mode	Count down	FFFFH	Value if stop	Undefined	FFFFH
Capture & one-count mode	Count up	0000H	Value if stop	Undefined	Capture value of TDRmn register + 1

Note This indicates the value read from the TCRmn register when channel n has stopped operating as a timer (TEmn = 0) and has been enabled to operate as a counter (TSmn = 1). The read value is held in the TCRmn register until the count operation starts.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

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 TDR01 and TDR03 registers, while in the 8-bit timer mode (when the SPLIT01 and SPLIT03 bits of timer mode registers 01 and 03 (TMR01, TMR03) are 1), it is possible to read and write the data in 8-bit units, with TDR01H and TDR03H used as the higher 8 bits, and TDR01L and TDR03L used as the lower 8 bits.

Reset signal generation clears this register to 0000H.

Figure 6 - 10 Format of Timer data register mn (TDRmn) (n = 0, 2)

Address: FFF18H, FFF19H (TDR00), FFF64H, FFF65H (TDR02)   After reset: 0000H   R/W
FFF70H, FFF71H (TDR10)

	FFF19H (TDR00)								FFF18H (TDR00)							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDRmn																

Figure 6 - 11 Format of Timer data register mn (TDRmn) (n = 1, 3)

Address: FFF1AH, FFF1BH (TDR01), FFF66H, FFF67H (TDR03)   After reset: 00H   R/W
FFF72H, FFF73H (TDR11)

	FFF1BH (TDR01H)								FFF1AH (TDR01L)							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDRmn																

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

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

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

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

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

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

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

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 input select register 0 (TIS0)
- Timer output enable register m (TOEm)
- Timer output register m (TOm)
- Timer output level register m (TOLm)
- Timer output mode register m (TOMm)
- Input switch control register (ISC)
- Noise filter enable registers 1, 2 (NFEN1, NFEN2)
- Port mode register 1 (PM1)
- Port register 1 (P1)

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

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

6.3.1 Peripheral enable register 0 (PER0)

This registers is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When timer array unit 0 is used, be sure to set bit 0 (TAU0EN) of this register to 1.

When timer array unit 1 is used, be sure to set bit 1 (TAU1EN) 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 - 12 Format of Peripheral enable register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	4	3	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	0	0	SAU0EN	TAU1EN	TAU0EN

TAU1EN	Control of timer array unit 1 input clock
0	Stops supply of input clock. • SFRs used by timer array unit 1 cannot be written. • Timer array unit 1 is in the reset status.
1	Supplies input clock. • SFRs used by timer array unit 1 can be read/written.

TAU0EN	Control of timer array 0 unit input clock
0	Stops supply of input clock. • SFRs used by timer array unit 0 cannot be written. • Timer array unit 0 is in the reset status.
1	Supplies input clock. • SFRs used by 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 the timer input select register 0 (TIS0), input switch control register (ISC), noise filter enable registers 1, 2 (NFEN1, NFEN2), port mode register 1 (PM1), and port register 1 (P1)).

- **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 bits 3, 4, and 6 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 3):
All channels for which CKm0 is selected as the operation clock (CKSmn1, CKSmn0 = 0, 0) are stopped (TEmn = 0).

If the PRSm10 to PRSm13 bits can be rewritten (n = 0 to 3):
All channels for which CKm2 is selected as the operation clock (CKSmn1, CKSmn0 = 0, 1) are stopped (TEmn = 0).

If the PRSm20 and PRSm21 bits can be rewritten (n = 1, 3):
All channels for which CKm1 is selected as the operation clock (CKSmn1, CKSmn0 = 1, 0) are stopped (TEmn = 0).

If the PRSm30 and PRSm31 bits can be rewritten (n = 1, 3):
All channels for which CKm3 is selected as the operation clock (CKSmn1, CKSmn0 = 1, 1) are stopped (TEmn = 0).

The TPSm register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 6 - 13 Format of Timer clock select register m (TPSm) (1/2)

Address: F01B6H, F01B7H (TPS0), F01F6H, F01F7H (TPS1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPSm	0	0	PRSm 31	PRSm 30	0	0	PRSm 21	PRSm 20	PRSm 13	PRSm 12	PRSm 11	PRSm 10	PRSm 03	PRSm 02	PRSm 01	PRSm 00

PRS mk3	PRS mk2	PRS mk1	PRS mk0	Selection of operation clock (CKmk) Note (k = 0, 1)					
					f_{CLK} = 2 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 20 MHz	f_{CLK} = 32 MHz
0	0	0	0	f_{CLK}	2 MHz	4 MHz	8 MHz	20 MHz	32 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2 MHz	4 MHz	10 MHz	16 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1 MHz	2 MHz	5 MHz	8 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	500 kHz	1 MHz	2.5 MHz	4 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	250 kHz	500 kHz	1.25 MHz	2 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	125 kHz	250 kHz	625 kHz	1 MHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	62.5 kHz	125 kHz	313 kHz	500 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	31.3 kHz	62.5 kHz	156 kHz	250 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	15.6 kHz	31.3 kHz	78.1 kHz	125 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	7.81 kHz	15.6 kHz	39.1 kHz	62.5 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	3.91 kHz	7.81 kHz	19.5 kHz	31.25 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	1.95 kHz	3.91 kHz	9.77 kHz	15.6 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	977 Hz	1.95 kHz	4.88 kHz	7.81 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	488 Hz	977 Hz	2.44 kHz	3.91 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	244 Hz	488 Hz	1.22 kHz	1.95 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61.0 Hz	122 Hz	244 Hz	610 Hz	977 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop timer array unit (TTm = 000FH).
The timer array unit must also be stopped if the operating clock (f_{MCK}) or the valid edge of the signal input from the TImn pin is selected.

Caution 1. Be sure to clear bits 15, 14, 11, and 10 to "0".

Caution 2. If f_{CLK} (undivided) is selected as the operation clock (CKmk) and TDRnm is set to 0000H (n = 0 or 1, m = 0 to 3), interrupt requests output from timer array units cannot be used.

Remark 1. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 2. Waveform of the clock to be selected in the TPSm register becomes high level for one f_{CLK} cycle from its rising edge (m = 1 to 15). For details, see **6.5.1 Count clock (f_{TCLK})**.

Figure 6 - 14 Format of Timer clock select register m (TPSm) (2/2)

Address: F01B6H, F01B7H (TPS0), F01F6H, F01F7H (TPS1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPSm	0	0	PRSm 31	PRSm 30	0	0	PRSm 21	PRSm 20	PRSm 13	PRSm 12	PRSm 11	PRSm 10	PRSm 03	PRSm 02	PRSm 01	PRSm 00

PRS m21	PRS m20		Selection of operation clock (CKm2) Note				
			f_{CLK} = 2 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 20 MHz	f_{CLK} = 32 MHz
0	0	$f_{CLK}/2$	1 MHz	2 MHz	4 MHz	10 MHz	16 MHz
0	1	$f_{CLK}/2^2$	500 kHz	1 MHz	2 MHz	5 MHz	8 MHz
1	0	$f_{CLK}/2^4$	125 kHz	250 kHz	500 kHz	1.25 MHz	2 MHz
1	1	$f_{CLK}/2^6$	31.3 kHz	62.5 kHz	125 kHz	313 kHz	500 kHz

PRS m31	PRS m30		Selection of operation clock (CKm3) Note				
			f_{CLK} = 2 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 20 MHz	f_{CLK} = 32 MHz
0	0	$f_{CLK}/2^8$	7.81 kHz	15.6 kHz	31.3 kHz	78.1 kHz	125 kHz
0	1	$f_{CLK}/2^{10}$	1.95 kHz	3.91 kHz	7.81 kHz	19.5 kHz	31.3 kHz
1	0	$f_{CLK}/2^{12}$	488 Hz	977 Hz	1.95 kHz	4.88 kHz	7.81 kHz
1	1	$f_{CLK}/2^{14}$	122 Hz	244 Hz	488 Hz	1.22 kHz	1.95 kHz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop timer array unit (TTm = 000FH).
The timer array unit must also be stopped if the operating clock (f_{MCK}) or the valid edge of the signal input from the TImn pin is selected.

Caution Be sure to clear bits 15, 14, 11, and 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 (f_{CLK} = 32 MHz)			
		10 μs	100 μs	1 ms	10 ms
CKm2	$f_{CLK}/2$	√	—	—	—
	$f_{CLK}/2^2$	√	—	—	—
	$f_{CLK}/2^4$	√	√	—	—
	$f_{CLK}/2^6$	√	√	—	—
CKm3	$f_{CLK}/2^8$	—	√	√	—
	$f_{CLK}/2^{10}$	—	√	√	—
	$f_{CLK}/2^{12}$	—	—	√	√
	$f_{CLK}/2^{14}$	—	—	√	√

Note The margin is within 5%.

Remark 1. f_{CLK}: CPU/peripheral hardware clock frequency

Remark 2. For details about a signal of $f_{CLK}/2^r$ selected by using the TPSm register, see **6.5.1 Count clock (f_{TCLK})**.

6.3.3 Timer mode register mn (TMRmn)

The TMRmn register sets an operation mode of channel n. This register is used to select the operation clock (f_{MCK}), select the count clock, select the master/slave, select the 16 or 8-bit timer, 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: MASTERmn bit (n = 2)
TMR01, TMR03: SPLIT0n bit (n = 1, 3)
TMR11: Fixed to 0
TMRm0: Fixed to 0

Figure 6 - 15 Format of Timer mode register mn (TMRmn) (1/4)

Address: F0190H, F0191H (TMR00) to F0196H, F0197H (TMR03), F01D0H, F01D1H (TMR10), F01D2H, F01D3H (TMR11) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2)	CKSm n1	CKSm n0	0	CCSm n	MAST ERmn	STSm n2	STSm n1	STSm n0	CISmn 1	CISmn 0	0	0	MDmn 3	MDmn 2	MDmn 1	MDmn 0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 1, 3)	CKS 0n1	CKS 0n0	0	CCS 0n	SPLIT 0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR11	CKS 111	CKS 110	0	CCS 11	0	STS 112	STS 111	STS 110	CIS 111	CIS 110	0	0	MD 113	MD 112	MD 111	MD 110

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0)	CKSm n1	CKSm n0	0	CCSm n	0 Note	STSm n2	STSm n1	STSm n0	CISmn 1	CISmn 0	0	0	MDmn 3	MDmn 2	MDmn 1	MDmn 0

CKS mn1	CKS mn0	Selection of operation clock (f_{MCK}) of channel n
0	0	Operation clock CKm0 set by timer clock select register m (TPSm)
0	1	Operation clock CKm2 set by timer clock select register m (TPSm)
1	0	Operation clock CKm1 set by timer clock select register m (TPSm)
1	1	Operation clock CKm3 set by timer clock select register m (TPSm)
Operation clock (f_{MCK}) is used by the edge detector. A count clock (f_{TCLK}) and a sampling clock are generated depending on the setting of the CCSmn bit. 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	When using unit 0: In channel 0, valid edge of input signal selected by TIS0 In channel 1, valid edge of input signal selected by TIS0 In channel 3, valid edge of input signal selected by ISC
Count clock (f_{TCLK}) is used for the counter, output controller, and interrupt controller.	

Note 1. Bit 11 is read-only and fixed to 0, so writing to this bit is ignored.

Caution 1. For the bits to which no function is assigned, be sure to set their values to 0.

Caution 2. The timer array unit must be stopped (TTm = 00FFH) if the clock selected for f_{CLK} is changed (by changing the value of the system clock control register (CKC)), even if the operating clock specified by using the CKSmn0 and CKSmn1 bits (f_{MCK}) or the valid edge of the signal input from the TImn pin is selected as the count clock (f_{TCLK}).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

Figure 6 - 16 Format of Timer mode register mn (TMRmn) (2/4)

Address: F0190H, F0191H (TMR00) to F0196H, F0197H (TMR03), F01D0H, F01D1H (TMR10), F01D2H, F01D3H (TMR11)   After reset: 0000H   R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2)	CKSmn1	CKSmn0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 1, 3)	CKS0n1	CKS0n0	0	CCS0n	SPLIT0n	STS0n2	STS0n1	STS0n0	CIS0n1	CIS0n0	0	0	MD0n3	MD0n2	MD0n1	MD0n0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR11	CKS111	CKS110	0	CCS11	0	STS112	STS111	STS110	CIS111	CIS110	0	0	MD113	MD112	MD111	MD110

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0)	CKSmn1	CKSmn0	0	CCSmn	0 Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

(Bit 11 of TMRmn (n = 2))

MASTERmn	Selection between using channel n independently or simultaneously with another channel (as a slave or master)
0	Operates in independent channel operation function or as slave channel in simultaneous channel operation function.
1	Operates as master channel in simultaneous channel operation function.
Only the channel 2 can be set as a master channel (MASTERmn = 1). Be sure to use channel 0 is fixed to 0 (regardless of the bit setting, channel 0 operates as master, because it is the highest channel). Clear the MASTERmn bit to 0 for a channel that is used with the independent channel operation function.	

(Bit 11 of TMR0n (n = 1, 3))

SPLIT0n	Selection of 8 or 16-bit timer operation for channels 1 and 3
0	Operates as 16-bit timer. (Operates in independent channel operation function or as slave channel in simultaneous channel operation function.)
1	Operates as 8-bit timer.

STSmn2	STSmn1	STSmn0	Setting of start trigger or capture trigger of channel n
0	0	0	Only software trigger start is valid (other trigger sources are unselected).
0	0	1	Valid edge of the TImn pin input is used as both the start trigger and capture trigger.
0	1	0	Both the edges of the TImn pin input are used as a start trigger and a capture trigger.
1	0	0	Interrupt signal of the master channel is used (when the channel is used as a slave channel with the simultaneous channel operation function).
Other than above			Setting prohibited

Note Bit 11 is read-only and fixed to 0, so writing to this bit is ignored.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

Figure 6 - 17 Format of Timer mode register mn (TMRmn) (3/4)

Address: F0190H, F0191H (TMR00) to F0196H, F0197H (TMR03), F01D0H, F01D1H (TMR10), F01D2H, F01D3H (TMR11) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2)	CKSmn1	CKSmn0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 1, 3)	CKS0n1	CKS0n0	0	CCS0n	SPLIT0n	STS0n2	STS0n1	STS0n0	CIS0n1	CIS0n0	0	0	MD0n3	MD0n2	MD0n1	MD0n0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR11	CKS111	CKS110	0	CCS11	0	STS112	STS111	STS110	CIS111	CIS110	0	0	MD113	MD112	MD111	MD110

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0)	CKSmn1	CKSmn0	0	CCSmn	0 Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

CISmn1	CISmn0	Selection of TImn pin input valid edge
0	0	Falling edge
0	1	Rising edge
1	0	Both edges (when low-level width is measured) Start trigger: Falling edge, Capture trigger: Rising edge
1	1	Both edges (when high-level width is measured) Start trigger: Rising edge, Capture trigger: Falling edge
If both the edges are specified when the value of the STSmn2 to STSmn0 bits is other than 010B, set the CISmn1 to CISmn0 bits to 10B.		

Note Bit 11 is read-only and fixed to 0, so writing to this bit is ignored.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

Figure 6 - 18 Format of Timer mode register mn (TMRmn) (4/4)

Address: F0190H, F0191H (TMR00) to F0196H, F0197H (TMR03), F01D0H, F01D1H (TMR10), F01D2H, F01D3H (TMR11) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 2)	CKSmn1	CKSmn0	0	CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n (n = 1, 3)	CKS0n1	CKS0n0	0	CCS0n	SPLIT0n	STS0n2	STS0n1	STS0n0	CIS0n1	CIS0n0	0	0	MD0n3	MD0n2	MD0n1	MD0n0

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR11	CKS111	CKS110	0	CCS11	0	STS112	STS111	STS110	CIS111	CIS110	0	0	MD113	MD112	MD111	MD110

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn (n = 0)	CKSmn1	CKSmn0	0	CCSmn	0 Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0	0	0	MDmn3	MDmn2	MDmn1	MDmn0

MDmn3	MDmn2	MDmn1	Operation mode of channel n	Corresponding function	Count operation of TCR
0	0	0	Interval timer mode	Interval timer / Square wave output / Divider function / PWM output (master)	Counting down
0	1	0	Capture mode	Input pulse interval measurement	Counting up
0	1	1	Event counter mode	External event counter	Counting down
1	0	0	One-count mode	Delay counter / One-shot pulse output / PWM output (slave)	Counting down
1	1	0	Capture & one-count mode	Measurement of high-/low-level width of input signal	Counting up
Other than above			Setting prohibited		
The operation of each mode varies depending on MDmn0 bit (see table below).					

Operation mode (Value set by the MDmn3 to MDmn1 bits (see table above))	MDmn0	Setting of starting counting and interrupt
• Interval timer mode (0, 0, 0) • Capture mode (0, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
	1	Timer interrupt is generated when counting is started (timer output also changes).
• Event counter mode (0, 1, 1)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• One-count mode Note 2 (1, 0, 0)	0	Start trigger is invalid during counting operation. At that time, interrupt is not generated.
	1	Start trigger is valid during counting operation Note 3. At that time, interrupt is not generated.
• Capture & one-count mode (1, 1, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either). Start trigger is invalid during counting operation. At that time, interrupt is not generated.

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

Note 1. Bit 11 is read-only and fixed to 0, so 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, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

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 set with an 8-bit memory manipulation instruction by using TSRmnL.

Reset signal generation clears this register to 0000H.

Figure 6 - 19 Format of Timer status register mn (TSRmn)

Address: F01A0H, F01A1H (TSR00) to F01A6H, F01A7H (TSR03) After reset: 0000H R
F01E0H, F01E1H (TSR10), F01E2H, F01E3H (TSR11)

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, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

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	— (Use prohibited)
	set	

Remark The OVF bit does not change immediately after the counter has overflowed, but changes upon the subsequent capture.

6.3.5 Timer channel enable status register m (TEm)

The TEm register is used to enable or stop the timer operation of each channel.

Each bit of the TEm register corresponds to each bit of the timer channel start register m (TSm) and the timer channel stop register m (TTm). When a bit of the TSm register is set to 1, the corresponding bit of this register is set to 1. When a bit of the TTm register is set to 1, the corresponding bit of this register is cleared to 0.

The TEm register can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of the TEm register can be set with a 1-bit or 8-bit memory manipulation instruction by using TEmL.

Reset signal generation clears this register to 0000H.

Figure 6 - 20 Format of Timer channel enable status register m (TEm)

Address: F01B0H, F01B1H (TE0), F01F0H, F01F1H (TE1) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEm	0	0	0	0	TEHm3	0	TEHm1	0	0	0	0	0	TEm3	TEm2	TEm1	TEm0

TEHm3	Indication of whether operation of the higher 8-bit timer is enabled or stopped when channel 3 is in the 8-bit timer mode
0	Operation is stopped.
1	Operation is enabled.

TEHm1	Indication of whether operation of the higher 8-bit timer is enabled or stopped when channel 1 is in the 8-bit timer mode
0	Operation is stopped.
1	Operation is enabled.

TEmn	Indication of operation enable/stop status of channel n
0	Operation is stopped.
1	Operation is enabled.
This bit displays whether operation of the lower 8-bit timer for TE01 and TE03 is enabled or stopped when channel 1 or 3 of unit 0 is in the 8-bit timer mode.	

Caution When the TEH13, TEH11, TE13, and TE12 bits in the TE1 register is read, the initial value is always read.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

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, TSH01, TSH03 bits are immediately cleared when operation is enabled (TEmn, TEH01, TEH03 = 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 by using TSmL.

Reset signal generation clears this register to 0000H.

Figure 6 - 21 Format of Timer channel start register m (TSm)

Address: F01B2H, F01B3H (TS0) to F01F2H, F01F3H (TS1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSm	0	0	0	0	TSHm3	0	TSHm1	0	0	0	0	0	TSm3	TSm2	TSm1	TSm0

TSHm3	Trigger to enable operation (start operation) of the higher 8-bit timer when channel 3 is in the 8-bit timer mode
0	No trigger operation
1	The TEHm3 bit is set to 1 and the count operation becomes enabled. The TCRm3 register count operation start in the interval timer mode in the count operation enabled state (see **Table 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 of unit 0 is in the 8-bit timer mode.

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

Caution 1. Be sure to clear bits 15 to 12, 10, and 8 to 4 to “0”

Caution 2. When switching from a function that does not use TImn pin input to one that does, the following wait period is required from when timer mode register mn (TMRmn) is set until the TSmn (TSHm1, TSHm3) bit is set to 1.
When the TImn pin noise filter is enabled (TNFENmn = 1): Four cycles of the operation clock (f_{MCK})
When the TImn pin noise filter is disabled (TNFENmn = 0): Two cycles of the operation clock (f_{MCK})

Caution 3. Be sure to set the TSH13, TSH11, TS13, and TS12 bits of the TS1 register to their initial value.

Remark 1. When the TSm register is read, 0 is always read.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

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, TTH01, TTH03 bits are immediately cleared when operation is stopped (TEmn, TEH01, TEH03 = 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 by using TTmL.
Reset signal generation clears this register to 0000H.

Figure 6 - 22 Format of Timer channel stop register m (TTm)

Address: F01B4H, F01B5H (TT0), F01F4H, F01F5H (TT1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTm	0	0	0	0	TTHm3	0	TTHm1	0	0	0	0	0	TTm3	TTm2	TTm1	TTm0

TTHm3	Trigger to stop operation of the higher 8-bit timer when channel 3 is in the 8-bit timer mode
0	No trigger operation
1	TEHm3 bit is cleared to 0 and the count operation is stopped.

TTHm1	Trigger to stop operation of the higher 8-bit timer when channel 1 is in the 8-bit timer mode
0	No trigger operation
1	TEHm1 bit is cleared to 0 and the count operation is stopped.

TTmn	Operation stop trigger of channel n
0	TEmn bit is cleared to 0 and the count operation is stopped.
1	Operation is stopped (stop trigger is generated). This bit is the trigger to stop operation of the lower 8-bit timer for TTm1 and TTm3 when channel 1 or 3 of unit 0 is in the 8-bit timer mode.

Caution 1. Be sure to clear bits 15 to 12, 10, 8 to 4 of the TTm register to "0".
Caution 2. Be sure to set the TTH13, TTH11, TT13, and TT12 bits of the TT1 register to their initial value.

Remark 1. When the TTm register is read, 0 is always read.
Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

6.3.8 Timer input select register 0 (TIS0)

The TIS0 register is used to select the timer input for channels 0 and 1 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 - 23 Format of Timer input select register 0 (TIS0)

Address: F0074H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	TIS04	0	TIS02	TIS01	TIS00

TIS04	Selection of timer input used with channel 0
0	Input signal of timer input pin (TI00)
1	Event input signal from ELC

TIS02	TIS01	TIS00	Selection of timer input used with channel 1
0	0	0	Input signal of timer input pin (TI01)
0	0	1	Event input signal from ELC
0	1	0	Input signal of timer input pin (TI01)
0	1	1	
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
Other than above			Setting prohibited

Caution 1. At least 1/f_{MCK} + 10 ns is necessary as the high-level and low-level widths of the timer input to be selected.

Caution 2. When selecting an event input signal from the ELC using timer input select register 0 (TIS0), select f_{CLK} by using timer clock select register 0 (TPS0).

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 by using TOEmL.

Reset signal generation clears this register to 0000H.

Figure 6 - 24 Format of Timer output enable register m (TOEm)

Address: F01BAH, F01BBH (TOE0), F01FAH, F01FBH (TOE1) After reset: 0000H R/W

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

TOE mn	Timer output enable/disable of channel n
0	Timer output is disabled. Timer operation is not applied to the TOmn bit and the output is fixed. Writing to the TOmn bit is enabled and the level set in the TOmn bit is output from the TOmn pin.
1	Timer output is enabled. Timer operation is applied to the TOmn bit and an output waveform is generated. Writing to the TOmn bit is ignored.

Caution 1. Be sure to clear bits 15 to 4 to “0”.

Caution 2. Be sure to set the TOE13 and TOE12 bits of the TOE1 register to their initial value.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

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 oh this register can be rewritten by software only when timer output is disabled (TOEmn = 0).
When timer output is enabled (TOEmn = 1), rewriting this register by software is ignored, and the value is changed only by the timer operation.
To use the TI00/TO00, TI01/TO01, TI02/TO02, TI03/TO03, TI10/TO10, and TI11/TO11 pins as a port function pin, set the corresponding TOmn bit to “0”.
The TOm register can be set by a 16-bit memory manipulation instruction.
The lower 8 bits of the TOm register can be set with an 8-bit memory manipulation instruction by using TOmL.
Reset signal generation clears this register to 0000H.

Figure 6 - 25 Format of Timer output register m (TOm)

Address: F01B8H, F01B9H (TO0), F01F8H, F01F9H (TO1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOm	0	0	0	0	0	0	0	0	0	0	0	0	TOm3	TOm2	TOm1	TOm0

TOmn	Timer output of channel n
0	Timer output value is “0”.
1	Timer output value is “1”.

Caution 1. Be sure to clear bits 15 to 4 to “0”.
Caution 2. Be sure to set the TO13, and TO12 bits of the TO1 register to their initial value.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

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 Slave channel output mode (TOMmn = 1). In the master channel output mode (TOMmn = 0), this register setting is invalid.
The TOLm register can be set by a 16-bit memory manipulation instruction.
The lower 8 bits of the TOLm register can be set with an 8-bit memory manipulation instruction by using TOLmL.
Reset signal generation clears this register to 0000H.

Figure 6 - 26 Format of Timer output level register m (TOLm)

Address: F01BCH, F01BDH (TOL0), F01FCH, F01FDH (TOL1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOLm	0	0	0	0	0	0	0	0	0	0	0	0	TOLm3	TOLm2	TOLm1	0

TOLmn	Control of timer output level of channel n
0	Positive logic output (active-high)
1	Negative logic output (active-low)

Caution 1. Be sure to clear bits 15 to 4, and 0 to “0”.
Caution 2. Be sure to set the TOL13 and TOL12 bits of the TOL1 register to their initial value.

Remark 1. If the value of this register is rewritten during timer operation, the timer output logic is inverted when the timer output signal changes next, instead of immediately after the register value is rewritten.
Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

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, or multiple PWM output), set the corresponding bit of the master channel to 0 and the corresponding bit of the slave channel to 1.

The setting of each channel n by this register is reflected at the timing when the timer output signal is set or reset while the timer output is enabled (TOEmn = 1).

The TOMm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the TOMm register can be set with an 8-bit memory manipulation instruction by using TOMmL.

Reset signal generation clears this register to 0000H.

Figure 6 - 27 Format of Timer output mode register m (TOMm)

Address: F01BEH, F01BFH (TOM0), F01FEH, F01FFH (TOM1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOMm	0	0	0	0	0	0	0	0	0	0	0	0	TOMm3	TOMm2	TOMm1	0

TOMmn	Control of timer output mode of channel n
0	Master channel output mode (to produce toggle output by timer interrupt request signal (INTTMmn))
1	Slave channel output mode (output is set by the timer interrupt request signal (INTTMmn) of the master channel, and reset by the timer interrupt request signal (INTTM0p) of the slave channel)

Caution 1. Be sure to clear bits 15 to 4, and 0 to “0”.

Caution 2. Be sure to set the TOM13 and TOM12 bits of the TOM1 register to their initial value.

Remark m: Unit number (m = 0, 1)

n: Channel number (n = 0 to 3)

(mn = 00, 02, or 10 for master channel)

p: Slave channel number (p = 1 to 3)

Unit 0: mp = 01 to 03 when n = 0

mp = 03 when n = 2

Unit 1: mp = 11 when n = 0

(For details about 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 ISC1 and ISC0 bits of the ISC register are used to implement LIN-bus communication operation by using channel 3 in association with the serial array unit. When the ISC1 bit is set to 1, the input signal of the serial data input pin (RxD0) is selected as a timer input signal.

For details about setting the SSIE00 bit, see **19.3.15 Input switch control register (ISC)**.

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 - 28 Format of Input switch control register (ISC)

Address: F0073H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	SSIE00	0	0	0	0	0	ISC1	ISC0

SSIE00	Setting $\overline{SSI00}$ pin input when CSI00 communication and slave mode are applied
0	$\overline{SSI00}$ pin input is invalid.
1	$\overline{SSI00}$ pin input is valid.

ISC1	Switching channel 3 input of timer array unit 0
0	Uses the input signal of the TI03 pin as a timer input (normal operation).
1	Input signal of the RXD0 pin is used as timer input (detects the wakeup signal and measures the low width of the break field and the pulse width of the sync field).

ISC0	Switching external interrupt (INTP0) input
0	Uses the input signal of the INTP0 pin as an external interrupt (normal operation).
1	Uses the input signal of the RXD0 pin as an external interrupt (wakeup signal detection).

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

Remark When the LIN-bus communication function is used, select the input signal of the RxD0 pin by setting ISC1 to 1.

6.3.14 Noise filter enable registers 1, 2 (NFEN1, NFEN2)

The NFEN1 and NFEN2 registers are used to set whether the noise filter can be used for the timer input signal to each channel.

Enable the noise filter by setting the corresponding bits to 1 on the pins in need of noise removal. When the noise filter is enabled, after synchronization with the operating clock (f_{MCK}) for the target channel, whether the signal keeps the same value for two clock cycles is detected.

When the noise filter is OFF, only synchronization is performed with the operation clock of target channel (f_{MCK}) Note.

The NFEN1 and NFEN2 registers can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Note For details, see **6.5.1 (2) When valid edge of input signal via the TImn pin is selected (CCSmn = 1)** and **6.5.2 Start timing of counter**, and **6.7 Timer Input (TImn) Control**.

Figure 6 - 29 Format of Noise filter enable register 1, 2 (NFEN1, NFEN2).

Address: F0071H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN1	0	0	0	0	TNFEN03	TNFEN02	TNFEN01	TNFEN00

Address: F0072H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN2	0	0	0	0	0	0	TNFEN11	TNFEN10

TNFEN03	Enable/disable using noise filter of TI03 pin or RxD0 pin input signal Note
0	Noise filter OFF
1	Noise filter ON

TNFEN02	Enable/disable using noise filter of TI02 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN01	Enable/disable using noise filter of TI01 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN00	Enable/disable using noise filter of TI00 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN11	Enable/disable using noise filter of TI11 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN10	Enable/disable using noise filter of TI10 pin input signal
0	Noise filter OFF
1	Noise filter ON

Note The applicable pin can be switched by setting the ISC1 bit of the ISC register.
ISC1 = 0: Whether or not to use the noise filter of the TI03 pin can be selected.
ISC1 = 1: Whether or not to use the noise filter of the RxD0 pin can be selected.

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 1 (PM1) and port register 1 (P1). For details, see **4.3.1 Port mode registers (PMxx)** and **4.3.2 Port registers (Pxx)**.

When using a port (such as P10/TO01 and P11/TO03) as a timer output pin, set the bit corresponding to the port in port mode register 1 (PM1) and port register 1 (P1) to 0.

Example: When using P10/TO01 for timer output
Set the PM10 bit of port mode register 1 to 0.
Set the P10 bit of port register 1 to 0.

When using a port (such as P10/TI01 and P11/TI03) as a timer input pin, set the bit corresponding to the port in port mode register 1 (PM1) to 1. At this time, port register 1 (P1) bit may be 0 or 1.

Example: When using P10/TI01 for timer input
Set the PM10 bit of port mode register 1 to 1.
Set the P10 bit of port register 1 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 or 2) can be set as a master channel.
(2) Any channel, except channel 0, can be set as a slave channel.
(3) The slave channel must be lower than the master channel.
Example: If channel 0 is set as a master channel, channel 1 or those that follow (channels 1, 2, 3) can be set as a slave channel.

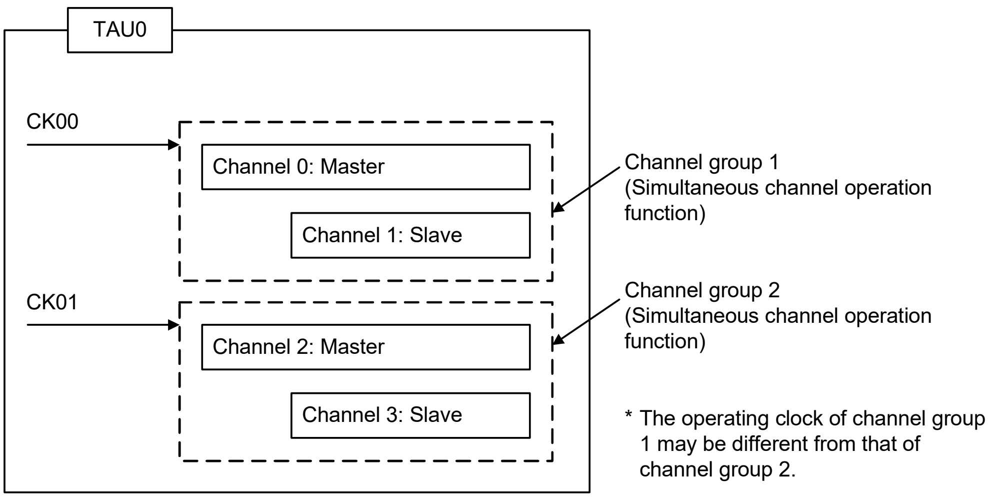
(4) Two or more slave channels can be set for one master channel.
(5) When two or more master channels are to be used, slave channels with a master channel between them may not be set.
Example: If channels 0 and 2 are set as master channels, channels 1 can be set as the slave channel of master channel 0. Channel 3 cannot be set as the slave channel of master channel 0.

(6) The operating clock for a slave channel in combination with a master channel must be the same as that of the master channel. The CKSmn0, CKSmn1 bits (bits 15 and 14 of timer mode register mn (TMRmn)) of the slave channel that operates in combination with the master channel must be the same value as that of the master channel.
(7) A master channel can transmit INTTMmn (interrupt), start software trigger, and count clock to the lower channels.
(8) A slave channel can use INTTMmn (interrupt), a start software trigger, or the count clock of the master channel as a source clock, but cannot transmit its own INTTMmn (interrupt), start software trigger, or count clock to channels with lower channel numbers.
(9) A master channel cannot use INTTMmn (interrupt), a start software trigger, or the count clock from the other higher master channel as a source clock.
(10) To simultaneously start channels that operate in combination, the channel start trigger bit (TSmn) of the channels in combination must be set at the same time.
(11) During the counting operation, a TSmn bit of a master channel or TSmn bits of all channels which are operating simultaneously can be set. It cannot be applied to TSmn bits of slave channels alone.
(12) To stop the channels in combination simultaneously, the channel stop trigger bit (TTmn) of the channels in combination must be set at the same time.
(13) CKm2/CKm3 cannot be selected while channels are operating simultaneously, because the operating clocks of master channels and slave channels have to be synchronized.
(14) Timer mode register m0 (TMRm0) has no master bit (it is fixed to 0). However, as channel 0 is the highest channel, it can be used as a master channel during simultaneous operation.

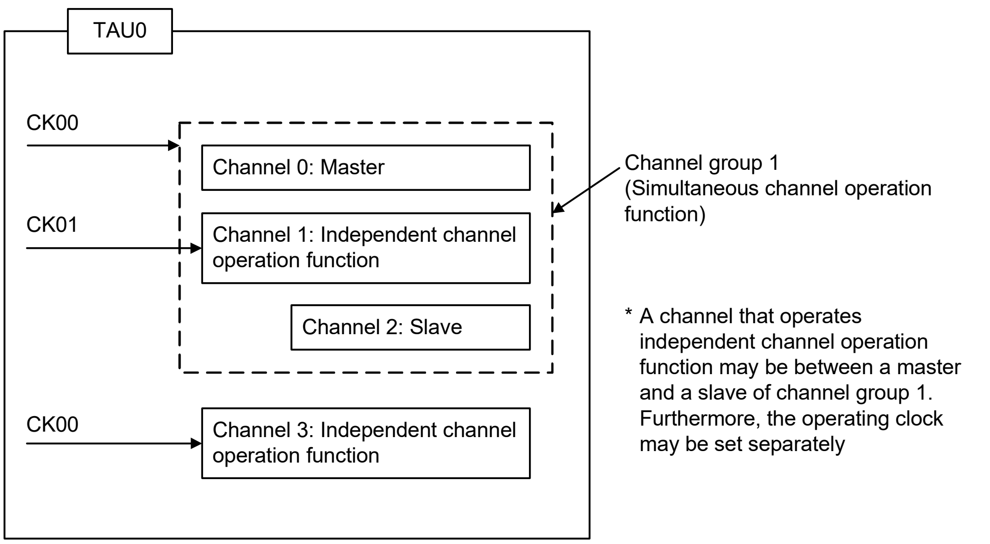
The rules of the simultaneous channel operation function are applied in a channel group (a master channel and slave channels forming one simultaneous channel operation function).
If two or more channel groups that do not operate in combination are specified, the basic rules of the simultaneous channel 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, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

Example 1



Example 2



6.4.2 Basic rules of 8-bit timer operation function (channels 1 and 3 only)

The 8-bit timer operation function makes it possible to use a 16-bit timer channel in a configuration consisting of two 8-bit timer channels.

This function can only be used for channels 1 and 3, and there are several rules for using it.

The basic rules for this function are as follows:

(1) The 8-bit timer operation function applies only to channels 1 and 3 of unit 0.

(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 four 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, TTm1, and TTm3 bits are manipulated to operate channels 1 and 3. The TEHm3 and TEHm1 bits are not changed.

(10) For the 8-bit timer function, the simultaneous operation functions (one-shot pulse, PWM, and multiple PWM) cannot be used.

Remark m: Unit number (m = 0), n: Channel number (n = 1, 3), mn = 01, 03

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 from the following by using the 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

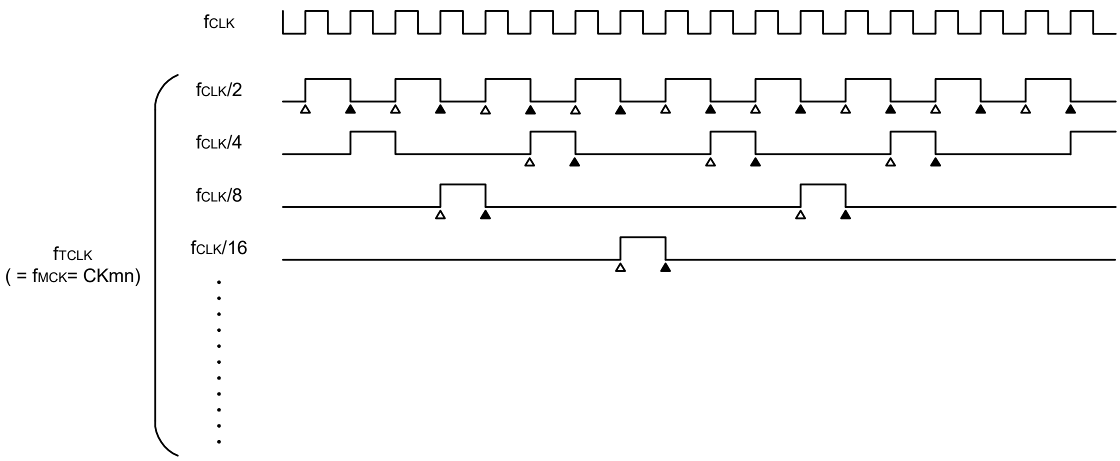
Because the timer array unit is designed to operate in synchronization with f_{CLK}, the timings of the count clock (f_{TCLK}) are shown below.

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

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

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

Figure 6 - 30 Timing of f_{CLK} and count clock (f_{TCLK}) (When CCSmn = 0)

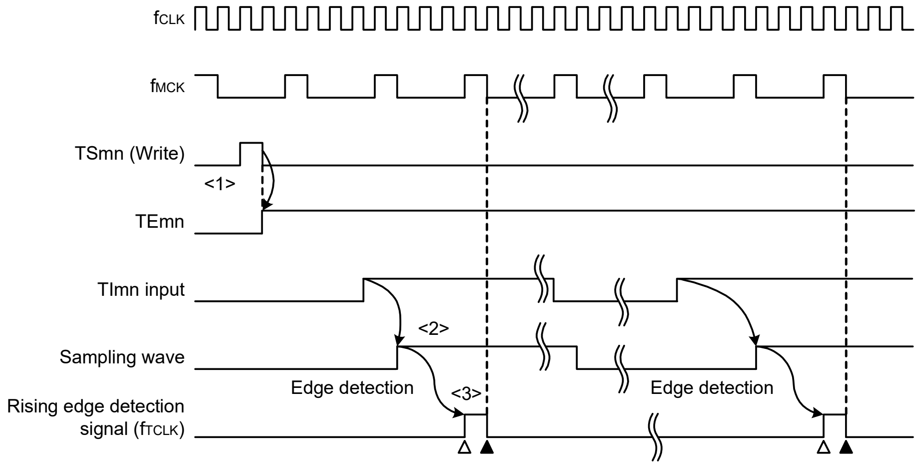


Remark 1. △ : Rising edge of the count clock
▲ : Synchronization or the counter is incremented or decremented

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}) is the signal that detects the valid edge of the input signal via the TImn pin and synchronizes next rising f_{MCK}. The count clock (f_{TCLK}) is delayed by 1 or 2 f_{MCK} cycles from the input signal via the TImn pin (when a noise filter is used, the delay becomes 3 or 4 clock cycles).
Counting by timer count register mn (TCRmn) is delayed by one f_{CLK} cycle from the rising edge of the count clock for synchronization with f_{CLK}. But, this is referred to as “counting at the valid edge of input signal via the TImn pin”, as a matter of convenience.

Figure 6 - 31 Timing of f_{CLK} and count clock (f_{TCLK}) (When CCSmn = 1, noise filter unused)



<1> Setting the 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 or the counter is incremented or decremented

Remark 2. f_{CLK}: CPU/peripheral hardware clock
f_{MCK}: Operation clock of channel n

Remark 3. The waveform of the signal input via the TImn pin of the input pulse interval measurement, the measurement of high/low width of input signal, the delay counter, and the one-shot pulse output are the same as that shown in Figure 6 - 31.

6.5.2 Start timing of counter

Timer count register mn (TCRmn) becomes enabled to operation by setting of TSmn bit of timer channel start register m (TSm).

Operations from count operation enabled state to timer count register mn (TCRmn) count start is shown in Table 6 - 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 until count clock generation. The first count clock loads 0000H to the TCRmn register and the subsequent count clock performs count up operation (see **6.5.3 (3) Operation of capture mode (input pulse interval measurement)**).
• One-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads the value of the TDRmn register to the TCRmn register and the subsequent count clock performs count down operation (see **6.5.3 (4) Operation of one-count mode**).
• Capture & one-count mode	The waiting-for-start-trigger state is entered by writing 1 to the TSmn bit while the timer is stopped (TEmn = 0). No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to the TCRmn register and the subsequent count clock performs count up operation (see **6.5.3 (5) Operation of capture & one-count mode (high-level width measurement)**).

6.5.3 Operation of counter

Here, the counter operation in each mode is explained.

(1) Operation of interval timer mode

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit. Timer count register mn (TCRmn) holds the initial value until count clock generation.

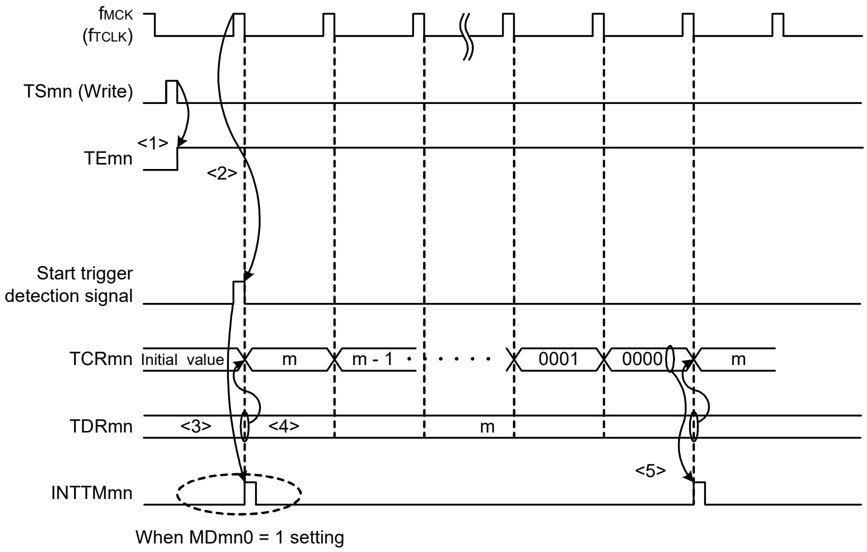
<2> A start trigger is generated at the first count clock after operation is enabled.

<3> When the MDmn0 bit is set to 1, INTTMmn is generated by the start trigger.

<4> By the first count clock after the operation enable, the value of timer data register mn (TDRmn) is loaded to the TCRmn register and counting starts in the interval timer mode.

<5> When the TCRmn register counts down and its count value is 0000H, INTTMmn is generated and the value of timer data register mn (TDRmn) is loaded to the TCRmn register and counting keeps on.

Figure 6 - 32 Operation Timing (In Interval Timer Mode)



Caution **In the first cycle operation of count clock after writing the TSmn bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.**

Remark fMCK, the start trigger detection signal, and INTTMmn become active between one clock in synchronization with fCLK.

(2) Operation of event counter mode

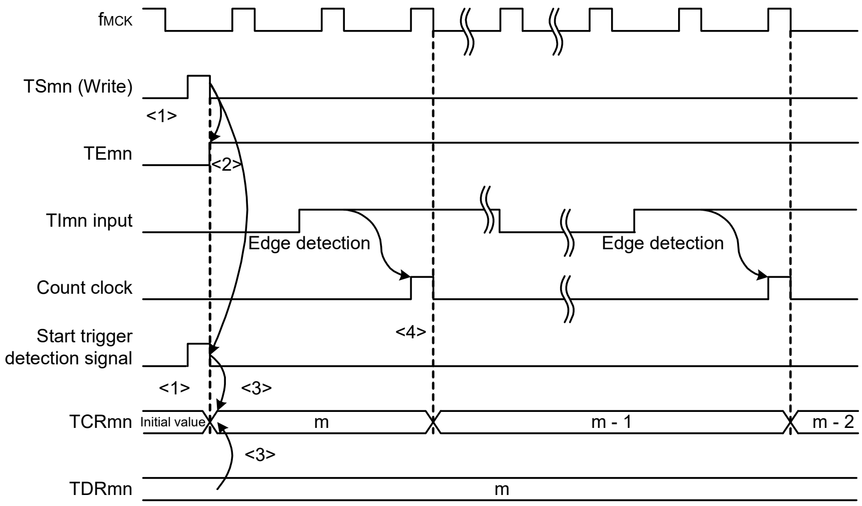
<1> Timer count register mn (TCRmn) holds its initial value while operation is stopped (TEmn = 0).

<2> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit.

<3> As soon as 1 has been written to the TSmn bit and 1 has been set to the TEmn bit, the value of timer data register mn (TDRmn) is loaded to the TCRmn register to start counting.

<4> After that, the TCRmn register value is counted down according to the count clock of the valid edge of the TImn input.

Figure 6 - 33 Operation Timing (In Event Counter Mode)



Remark Figure 6 - 33 shows the timing of when the noise filter is not used. By turning the noise filter on-state, the edge detection becomes 2 fMCK cycles (3 to 4 cycles in total) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (fMCK).

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

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

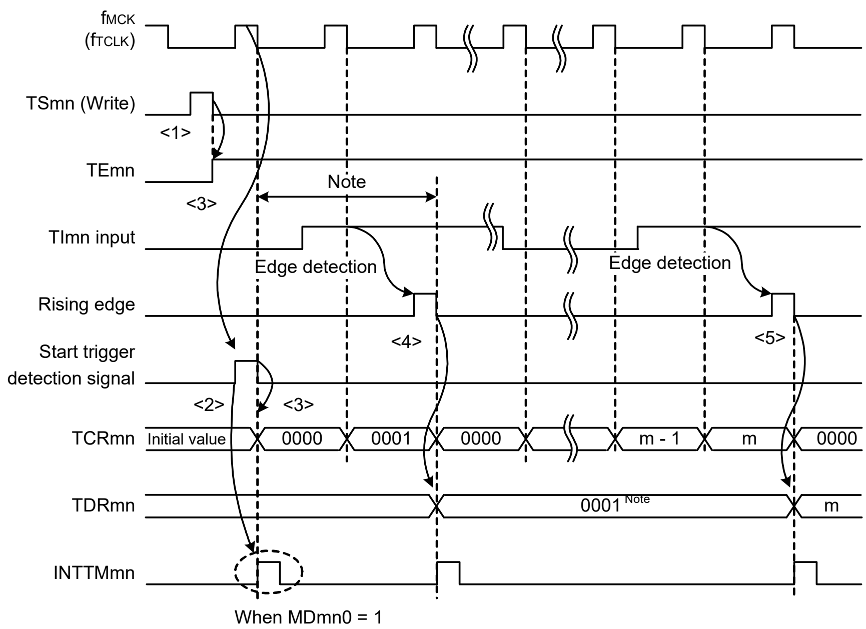
<2> Timer count register mn (TCRmn) holds the initial value until count clock generation.

<3> A start trigger is generated at the first count clock after operation is enabled. And the value of 0000H is loaded to the TCRmn register and counting starts in the capture mode. (When the MDmn0 bit is set to 1, INTTMmn is generated by the start trigger.)

<4> On detection of the valid edge of the TImn input, the value of the TCRmn register is captured to timer data register mn (TDRmn) and INTTMmn is generated. However, this capture value is no meaning. The TCRmn register keeps on counting from 0000H.

<5> On next detection of the valid edge of the TImn input, the value of the TCRmn register is captured to timer data register mn (TDRmn) and INTTMmn is generated.

Figure 6 - 34 Operation Timing (In Capture Mode: Input Pulse Interval Measurement)



Note If a clock has been input to TImn (the trigger exists) when capturing starts, counting starts when a trigger is detected, even if no edge is detected. Therefore, the first captured value (<4>) does not determine a pulse interval (in the above figure, 0001 just indicates two clock cycles but does not determine the pulse interval) and so the user can ignore it.

Caution In the first cycle operation of count clock after writing the TSmn bit, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.

Remark Figure 6 - 34 shows the timing of when the noise filter is not used. By turning the noise filter on, the edge detection becomes 2 fMCK cycles (3 to 4 cycles in total) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (fMCK).

(4) Operation of one-count mode

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

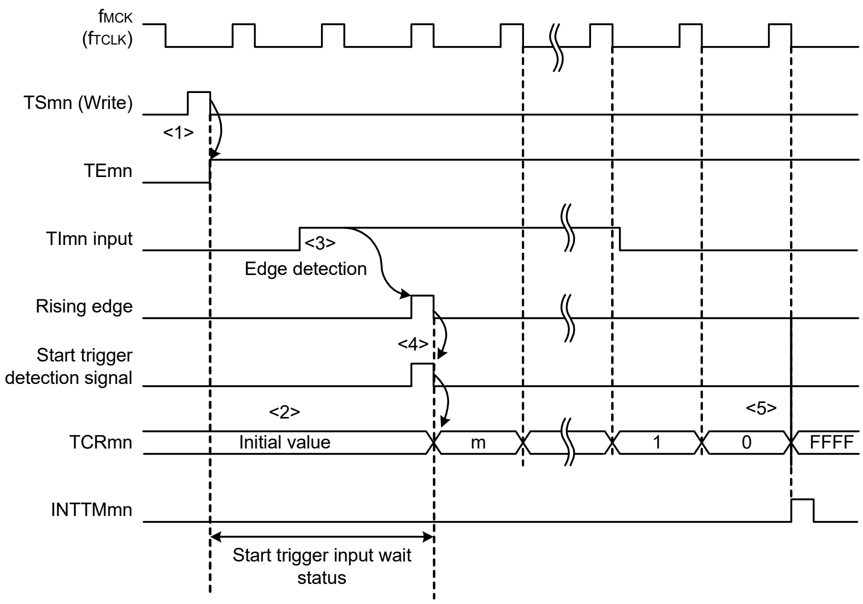
<2> Timer count register mn (TCRmn) holds the initial value until start trigger generation.

<3> Rising edge of the TImn input is detected.

<4> On start trigger detection, the value of timer data register mn (TDRmn) is loaded to the TCRmn register and count starts.

<5> When the TCRmn register counts down and its count value is 0000H, INTTMmn is generated and the value of the TCRmn register becomes FFFFH and counting stops.

Figure 6 - 35 Operation Timing (In One-count Mode)



Remark Figure 6 - 35 shows the timing of when the noise filter is not used. By turning the noise filter on, the edge detection becomes 2 fMCK cycles (3 to 4 cycles in total) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (fMCK).

(5) Operation of capture & one-count mode (high-level width measurement)

<1> Operation is enabled (TEmn = 1) by writing 1 to the TSmn bit of timer channel start register m (TSm).

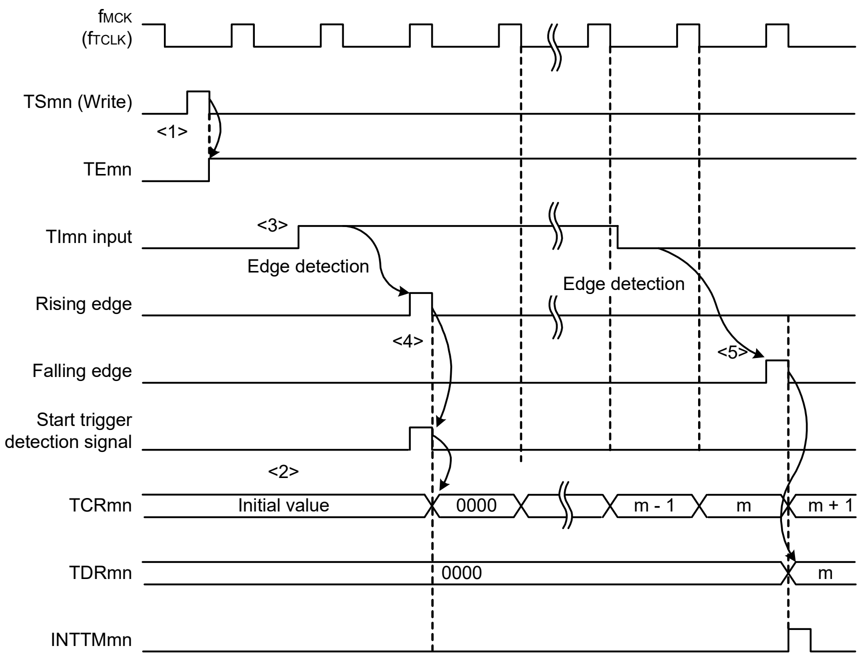
<2> Timer count register mn (TCRmn) holds the initial value until start trigger generation.

<3> Rising edge of the TImn input is detected.

<4> On start trigger detection, the value of 0000H is loaded to the TCRmn register and count starts.

<5> On detection of the falling edge of the TImn input, the value of the TCRmn register is captured to timer data register mn (TDRmn) and INTTMmn is generated.

Figure 6 - 36 Operation Timing (In Capture & One-count Mode: High-level Width Measurement)

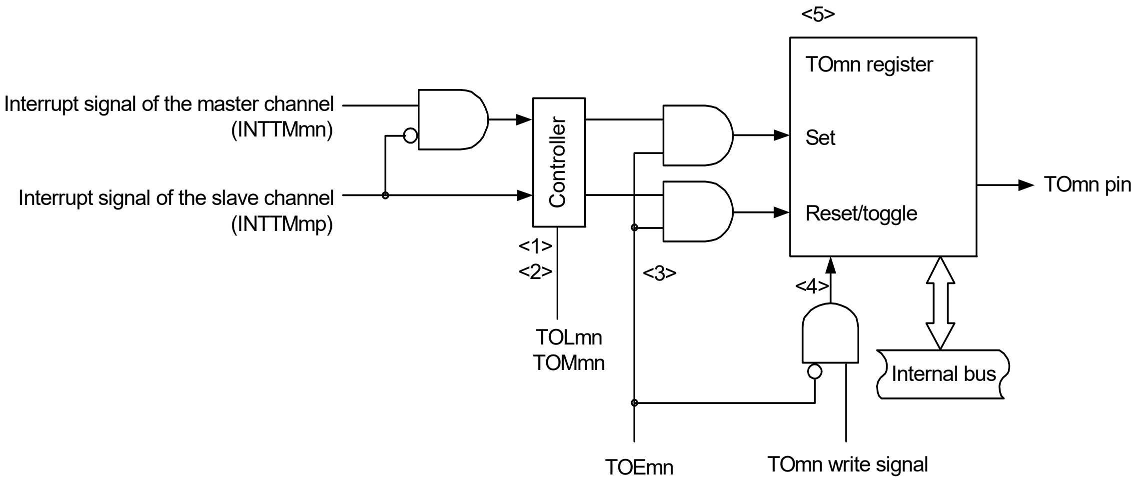


Remark Figure 6 - 36 shows the timing of when the noise filter is not used. By turning the noise filter on, the edge detection becomes 2 fMCK cycles (3 to 4 cycles in total) later than the normal cycle of TImn input. The error per one period occurs be the asynchronous between the period of the TImn input and that of the count clock (fMCK).

6.6 Channel Output (TOmn pin) Control

6.6.1 TOmn pin output circuit configuration

Figure 6 - 37 Output Circuit Configuration



The following describes the TOmn pin output circuit.

<1> When TOMmn = 0 (master channel output mode), the set value of timer output level register m (TOLm) is ignored and only INTTM0p (slave channel timer interrupt) is transmitted to timer output register m (TOm).

<2> When TOMmn = 1 (slave channel output mode), both INTTMmn (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TOm register.
At this time, the TOLm register becomes valid and the signals are controlled as follows:

When TOLmn = 0: Forward operation (INTTMmn → set, INTTM0p → reset)
When TOLmn = 1: Reverse operation (INTTMmn → reset, INTTM0p → set)

When INTTMmn and INTTM0p are simultaneously generated, (0% output of PWM), INTTM0p (reset signal) takes priority, and INTTMmn (set signal) is masked.

<3> While timer output is enabled (TOEmn = 1), INTTMmn (master channel timer interrupt) and INTTM0p (slave channel timer interrupt) are transmitted to the TOm register. Writing to the TOm register (TOmn write signal) becomes invalid.
When TOEmn = 1, the TOmn pin output never changes with signals other than interrupt signals.
To initialize the TOmn pin output level, it is necessary to set timer operation is stopped (TOEmn = 0) and to write a value to the TOm register.

<4> While timer output is disabled (TOEmn = 0), writing to the TOmn bit to the target channel (TOmn write signal) becomes valid. When timer output is disabled (TOEmn = 0), neither INTTMmn (master channel timer interrupt) nor INTTMmp (slave channel timer interrupt) is transmitted to the TOm register.

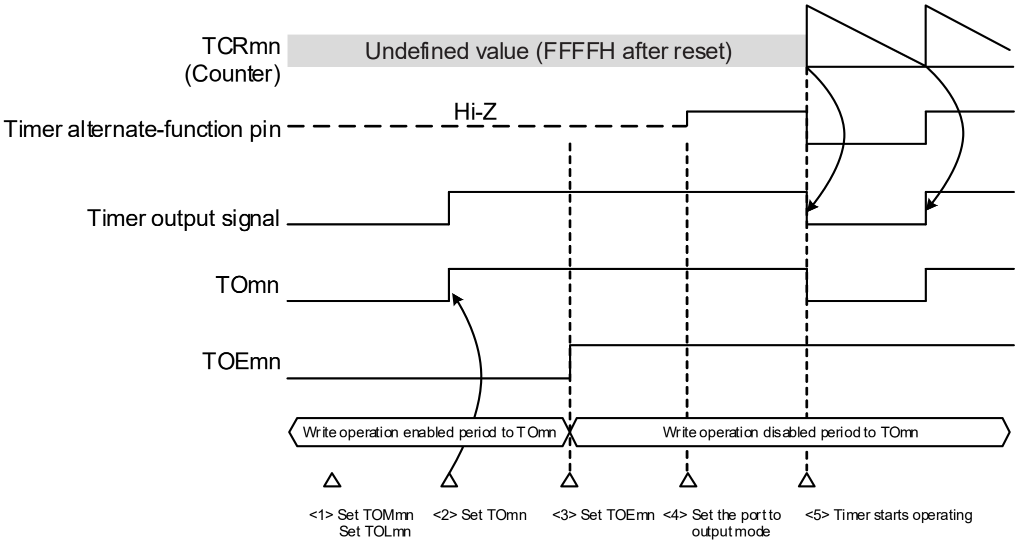
<5> The TOm register can always be read, and the TOmn pin output level can be checked.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00, 02, or 10 for master channel)
p: Slave channel number (p = 1 to 3)
Unit 0: mp = 01 to 03 when n = 0
mp = 03 when n = 2
Unit 1: mp = 11 when n = 0

6.6.2 TOmn pin output setting

The following figure shows the procedure and status transition of the TOmn output pin from initial setting to timer operation start.

Figure 6 - 38 Status Transition from Timer Output Setting to Operation Start



<1> The operation mode of timer output is set.

- TOMmn bit (0: Master channel output mode, 1: Slave channel output mode)
- TOLmn bit (0: 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 is enabled by writing 1 to the TOEmn bit (writing to the TOm register is disabled).

<4> 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**).

<5> The timer operation is enabled (TSmn = 1).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

6.6.3 Cautions on channel output operation

(1) Changing values set in the registers TOm, TOEm, TOLm, and TOMm during timer operation

Since the timer operations (operations of timer count register mn (TCRmn) and timer data register mn (TDRmn)) are independent of the TOmn output circuit and changing the values set in timer output register m (TOm), timer output enable register m (TOEm), and timer output level register m (TOLm) does not affect the timer operation, the values can be changed during timer operation. To output an expected waveform from the TOmn pin by timer operation, however, set the TOm, TOEm, TOLm, and TOMm registers to the values stated in the register setting example of each operation shown by 6.8 and 6.9.

When the values set to the TOEm and TOLm registers (but not the TOm register) are changed close to the occurrence of the timer interrupt (INTTMmn) of each channel, the waveform output to the TOmn pin might differ, depending on whether the values are changed immediately before or immediately after the timer interrupt (INTTMmn) occurs.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

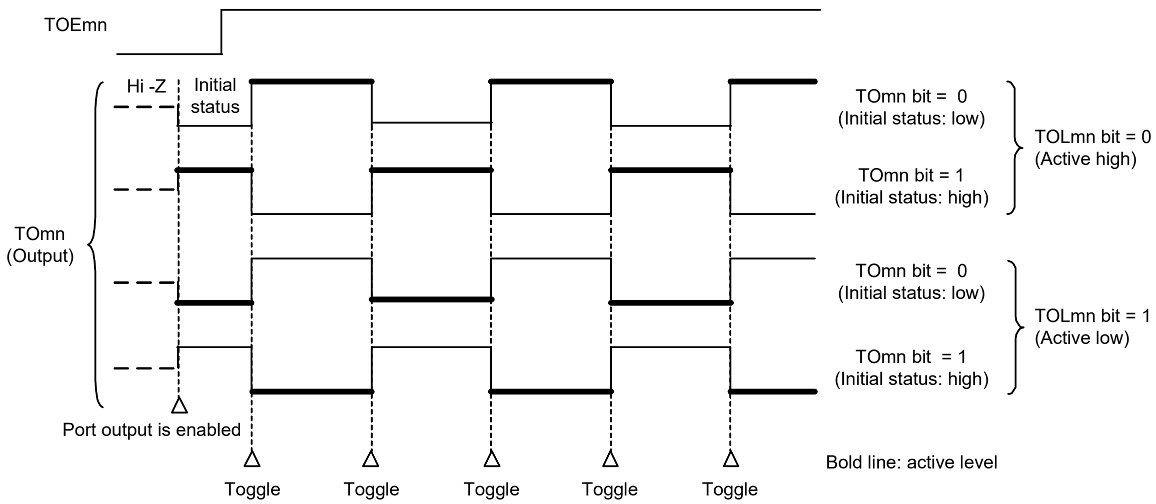
(2) Default level of TOmn pin and output level after timer operation start

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 - 39 TOmn Pin Output Status at Toggle Output (TOMmn = 0)

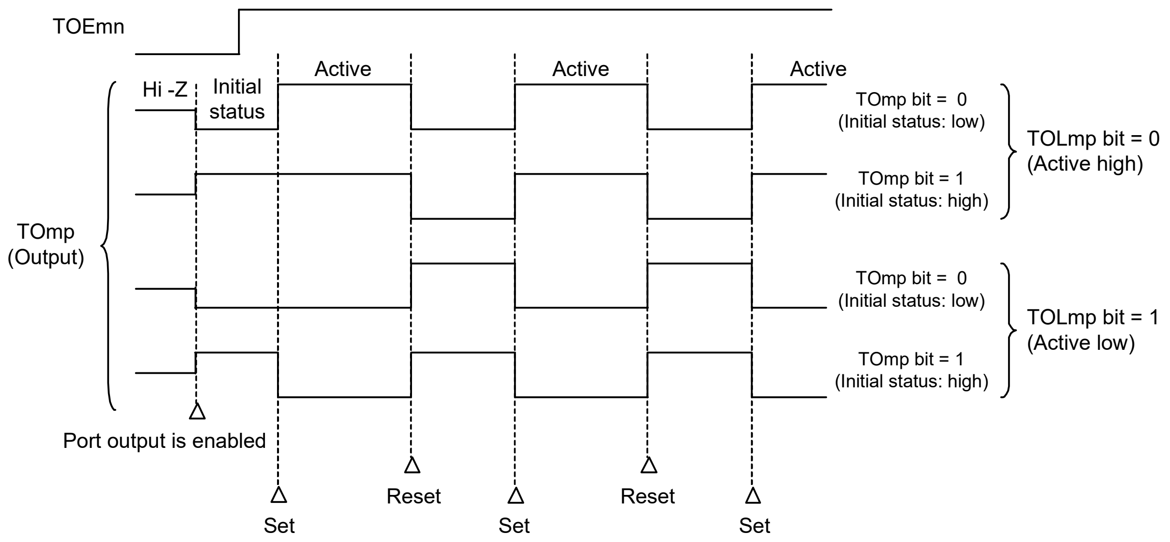


Remark 1. Toggle: Reverse TOmn pin output status

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

(b) When operation starts with slave channel output mode (TOMmn = 1) setting (PWM output))
When slave channel output mode (TOMmn = 1), the active level is determined by timer output level register m (TOLm) setting.

Figure 6 - 40 TOmn Pin Output Status at PWM Output (TOMmn = 1)



Remark 1. Set: The output signal of the TOmp pin changes from inactive level to active level.
Reset: The output signal of the TOmp pin changes from active level to inactive level.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (p = 1 to 3), p: Slave channel number (p = 1 to 3)
mn = 01 to 03, 10, 11, mp = 01 to 03 or 11

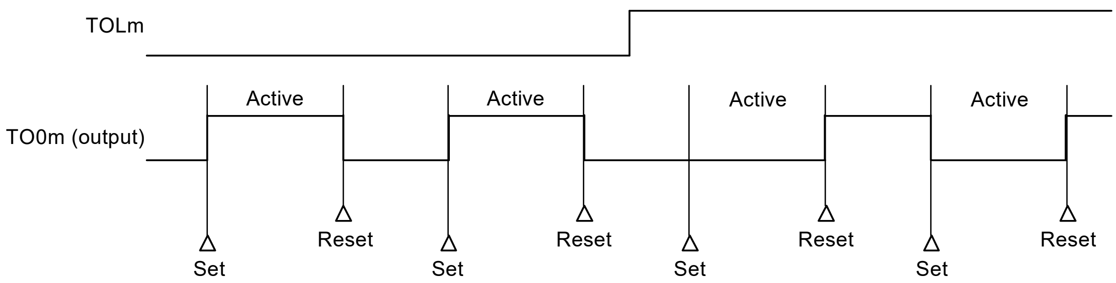
(3) Operation of TOmn pin in slave channel output mode (TOMmn = 1)

(a) When timer output level register m (TOLm) setting has been changed during timer operation

When the TOLm register setting has been changed during timer operation, the setting becomes valid at the generation timing of the TOmn pin change condition. Rewriting the TOLm register does not change the output level of the TOmn pin.

The operation when TOMmn is set to 1 and the value of the TOLm register is changed while the timer is operating (TEmn = 1) is shown below.

Figure 6 - 41 Operation when TOLm Register Has Been Changed during Timer Operation



Remark 1. Set: The output signal of the TOmn pin changes from inactive level to active level.
Reset: The output signal of the TOmn pin changes from active level to inactive level.

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

(b) Set/reset timing

To realize 0%/100% output at PWM output, the TOmn pin/TOmn bit set timing at master channel timer interrupt (INTTMmn) generation is delayed by 1 count clock by the slave channel.

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

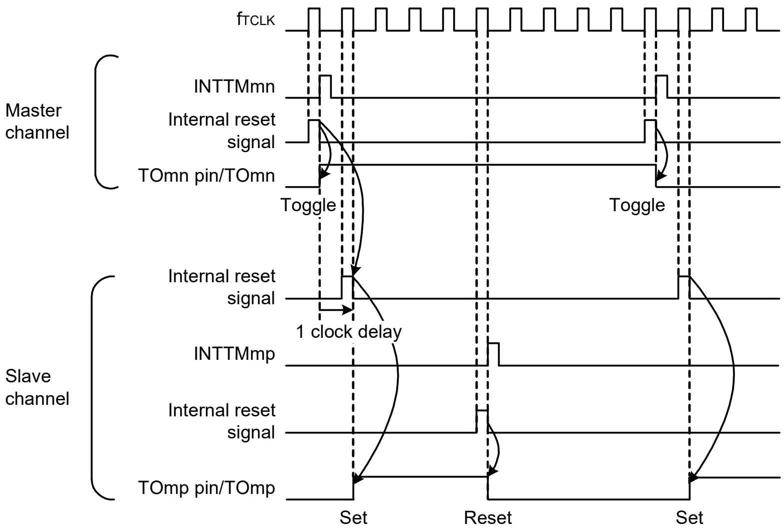
Figure 6 - 42 shows the set/reset operating statuses where the master/slave channels are set as follows.

Master channel: TOEmn = 1, TOMmn = 0, TOLmn = 0

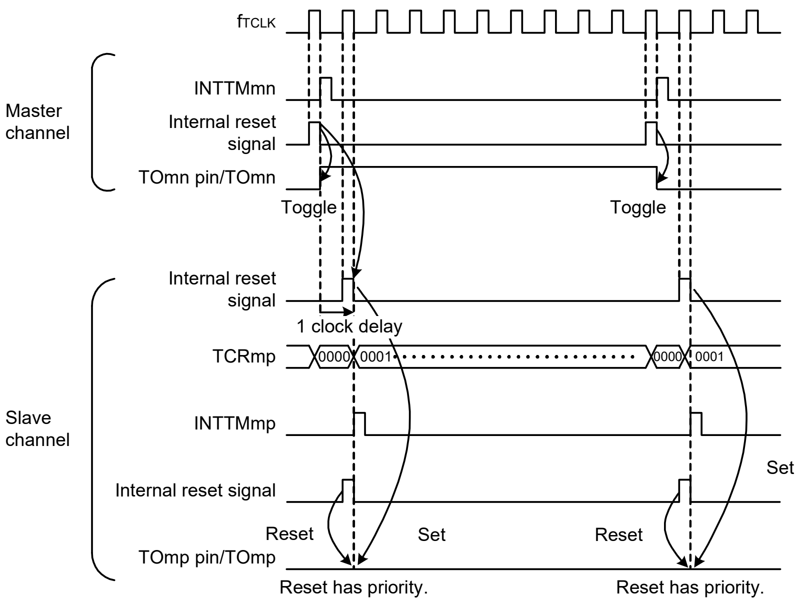
Slave channel: TOEmp = 1, TOMmp = 1, TOLmp = 0

Figure 6 - 42 Set/Reset Timing Operating Statuses

(1) Basic operation timing



(2) Operation timing when 0% duty



Remark 1. Internal reset signal: TOmn pin reset/toggle signal
Internal set signal: TOmn pin set signal

Remark 2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)
(mn = 00, 02, or 10 for master channel)
p: Slave channel number)
Unit 0: mp = 01 to 03 when n = 0
mp = 03 when n = 2
Unit 1: mp = 11 when n = 0

6.6.4 Collective manipulation of TOmn bit

In timer output register m (TOm), the setting bits for all the channels are located in one register in the same way as timer channel start register m (TSm). Therefore, the TOmn bit of all the channels can be manipulated collectively.

Only the desired bits can also be manipulated by enabling writing only to the TOmn bits (TOEmn = 0) that correspond to the relevant bits of the channel used to perform output (TOmn).

Figure 6 - 43 Example of TO0n Bit Collective Manipulation

Before writing

TO0	0	0	0	0	0	0	0	0	0	0	0	0	TO03 1	TO02 0	TO01 1	TO00 0
TOE0	0	0	0	0	0	0	0	0	0	0	0	0	TOE03 0	TOE02 0	TOE01 0	TOE00 1

Data to be written

0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

After writing

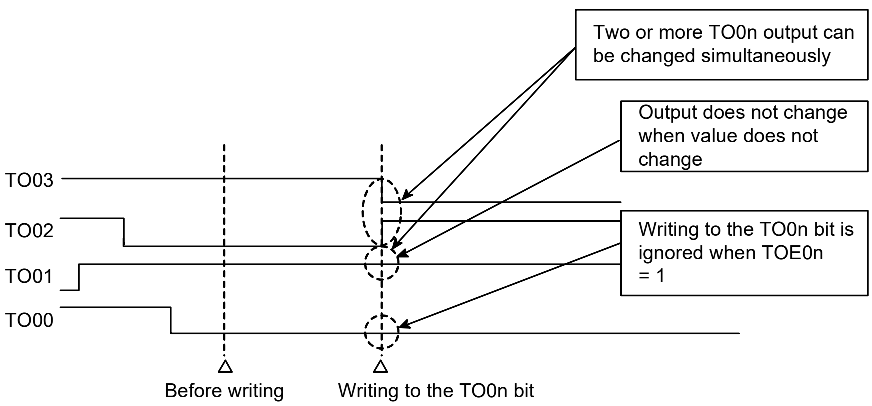
TO0	0	0	0	0	0	0	0	0	0	0	0	0	TO03 0	TO02 1	TO01 1	TO00 0

Writing is done only to the TOmn bit with TOEmn = 0, and writing to the TOmn bit with TOEmn = 1 is ignored.

TOmn (channel output) to which TOEmn = 1 is set is not affected by the write operation. Even if the write operation is done to the TOmn bit, it is ignored and the output change by timer operation is normally done.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 6 - 44 TO0n Pin Statuses by Collective Manipulation of TO0n Bit



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

6.6.5 Timer Interrupt and TOmn pin output at operation start

In the interval timer mode or capture mode, the MDmn0 bit in timer mode register mn (TMRmn) sets whether or not to generate a timer interrupt at count start.

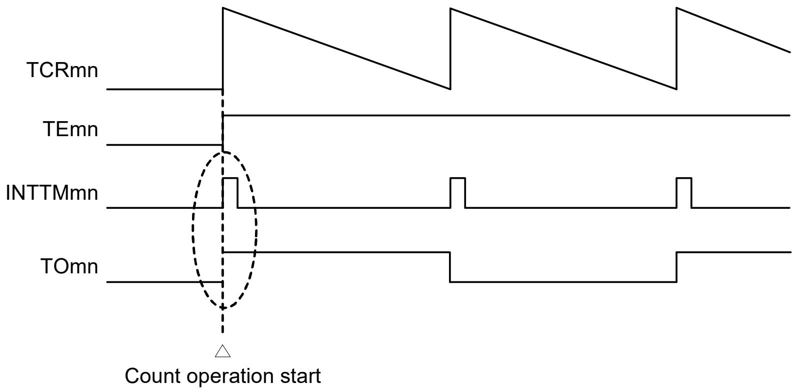
When MDmn0 is set to 1, the count operation start timing can be known by the timer interrupt (INTTMmn) generation.

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

Figure 6 - 45 shows operation examples when the interval timer mode (TOEmn = 1, TOMmn = 0) is set.

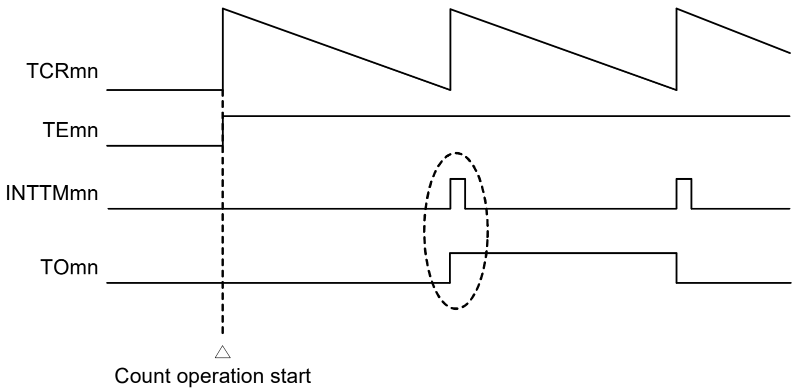
Figure 6 - 45 Operation examples of timer interrupt at count operation start and TOmn output

(a) When MDmn0 is set to 1



When MDmn0 is set to 1, a timer interrupt (INTTMmn) is output at count operation start, and TOmn performs a toggle operation.

(b) When MDmn0 is set to 0



When MDmn0 is set to 0, a timer interrupt (INTTMmn) is not output at count operation start, and TOmn does not change either. After counting one cycle, INTTMmn is output and TOmn performs a toggle operation.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

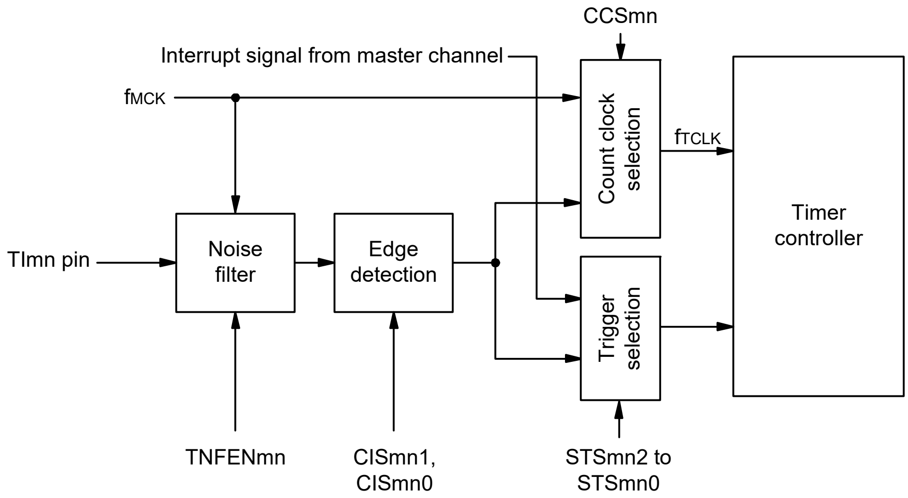
6.7 Timer Input (TImn) Control

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.

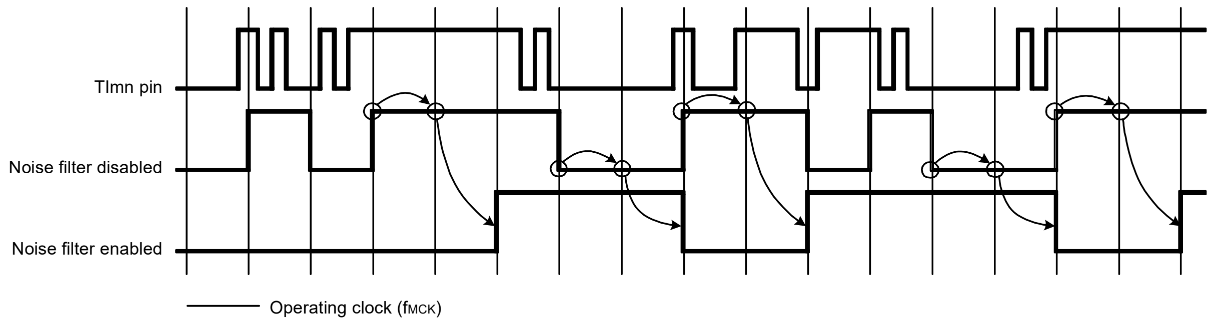
Figure 6 - 46 Input Circuit Configuration



6.7.2 Noise filter

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

Figure 6 - 47 Sampling Waveforms through TImn Input Pin with Noise Filter Enabled and Disabled



Caution The input waveforms to the TImn pin are shown to explain the operation when the noise filter is enabled or disabled. When actually inputting waveforms, input them according to the TImn input high-level and low-level widths listed in 33.4 or 34.4 AC Characteristics.

6.7.3 Cautions on channel input operation

When a timer input pin is not used, the operating clock is not supplied to the noise filter. Therefore, after the timer input pin is set to be used, the following wait time is necessary before a trigger is specified to enable the 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 timer mode register mn (TMRmn) are 0 and then one of them is set to 1, wait for at least two cycles of the operating clock (f_{MCK}), and then set the operation enable trigger bit in the timer channel start register (TSm).

(2) Noise filter is enabled

When bits 12 (CCSmn), 9 (STSmn1), and 8 (STSmn0) in timer mode register mn (TMRmn) are all 0 and then one of them is set to 1, wait for at least four cycles of the operating clock (f_{MCK}), and then set the operation enable trigger bit in the timer channel start register (TSm).

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.

Generation period of INTTMmn (timer interrupt) = Period of count clock × (Set value of TDRmn + 1)

(2) Operation as square wave output

TOmn performs a toggle operation as soon as INTTMmn has been generated, and outputs a square wave with a duty factor of 50%.

The period and frequency for outputting a square wave from TOmn can be calculated by the following expressions.

• Period of square wave output from TOmn = Period of count clock × (Set value of TDRmn + 1) × 2
• Frequency of square wave output from TOmn = Frequency of count clock/{(Set value of TDRmn + 1) × 2}

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

The TCRmn register loads the value of timer data register mn (TDRmn) at the first count clock after the channel start trigger bit (TSmn, TSHm1[Note], TSHm3[Note]) 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.

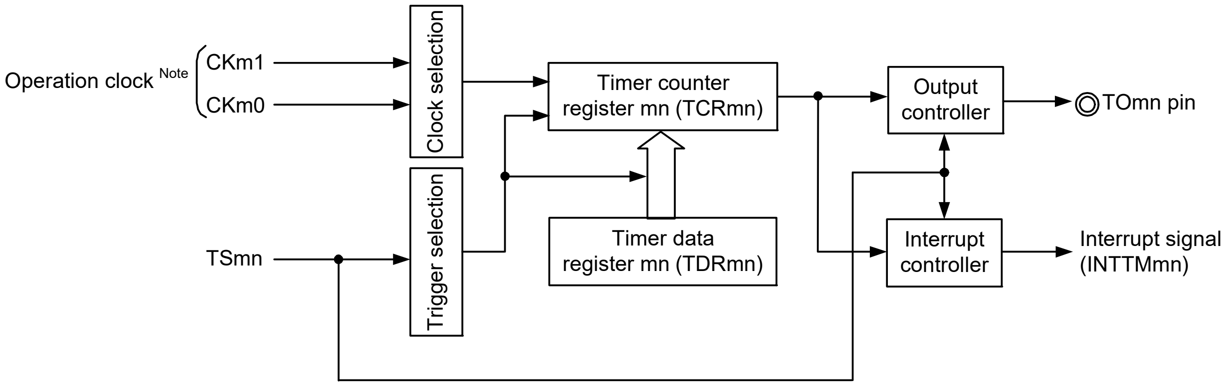
When TCRmn = 0000H, INTTMmn is output and TOmn is toggled at the next count clock. At the same time, the TCRmn register loads the value of the TDRmn register again. After that, the same operation is repeated.

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

Note Be sure to set the TSH11 and TSH13 bits to its initial value.

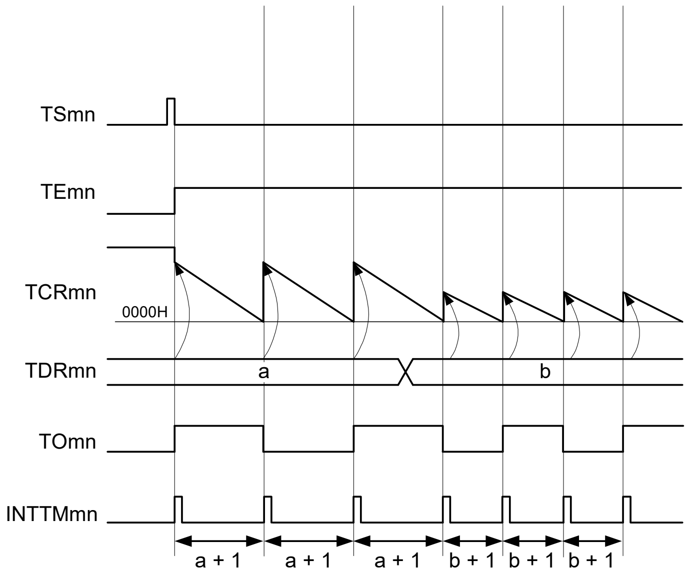
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

Figure 6 - 48 Block Diagram of Operation as Interval Timer/Square Wave Output



Note When channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Figure 6 - 49 Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MDmn0 = 1)



Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

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 - 50 Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (1/2)

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	M/S Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	1/0	0	0	0/1	0	0	0	0	0	0	0	0	0	0	1/0

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
0: Neither generates INTTMmn nor inverts timer output when counting is started.
1: Generates INTTMmn and inverts timer output when counting is started.

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

Start trigger selection
000B: Selects only software start.

Setting of MASTERmn bit (channel 2)
0: Independent channel operation function.
Setting of SPLIT0n bit (channels 1 and 3 of unit 0)
0: 16-bit timer mode
1: 8-bit timer mode

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected using channels 1 and 3 in the 8-bit timer mode).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected using channels 1 and 3 in the 8-bit timer mode).

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	1/0

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

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	1/0

0: Stops the TOmn output operation by counting operation.
1: Enables the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn
	0

0: Cleared to 0 when TOMmn = 0 (master channel output mode)

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn
	0

0: Sets master channel output mode.

Note

TMRm2:	MASTERmn bit
TMR01, TMR03:	SPLIT0n bit
TMR11:	Fixed to 0
TMRm0:	Fixed to 0

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

Figure 6 - 51 Operation Procedure of Interval Timer/Square Wave Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit m is stopped (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Input clock supply for timer array unit m is supplied (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets timer mode register mn (TMRmn) (determines operation mode of channel). Sets interval (period) value to timer data register mn (TDRmn).	Channel stops operating. (Clock is supplied and some power is consumed.)
	To use the TOmn output Clears the TOMmn bit of timer output mode register m (TOMm) to 0 (master channel output mode). Clears the TOLmn bit to 0. Sets the TOmn bit and determines default level of the TOmn output. →	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 (TSHm1Note 1, TSHm3Note 1) bit to 1. → The TSmn (TSHm1Note 1, TSHm3Note 1) bit automatically returns to 0 because it is a trigger bit.	TEmn (TEHm1, TEHm3) = 1, and count operation starts. Value of the TDRmn register is loaded to timer count register mn (TCRmn). INTTMmn is generated and TOmn performs toggle operation if the MDmn0 bit of the TMRmn register is 1.
During operation	Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TOm and TOEm registers can be changed. Set values of the TMRmn register, TOMmn, and TOLmn bits cannot be changed.	Counter (TCRmn) counts down. When count value reaches 0000H, the value of the TDRmn register is loaded to the TCRmn register again and the count operation is continued. By detecting TCRmn = 0000H, INTTMmn is generated and TOmn performs toggle operation. After that, the above operation is repeated.
Operation stop	The TTmn (TTHm1Note 2, TTHm3Note 2) bit is set to 1. → The TTmn (TTHm1Note 2, TTHm3Note 2) 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: arrow from Operation stop back to Operation start.)

(Remark is listed on the next page.)

Figure 6 - 52 Operation Procedure of Interval Timer/Square Wave Output Function (2/2)

	Software Operation	Hardware Status
TAU stop	To hold the TOmn pin output level Clears the TOmn bit to 0 after the value to be held is set to the port register. → When holding the TOmn pin output level is not necessary Setting not required.	The TOmn pin output level is held by port function.
	The TAUmEN bit of the PER0 register is cleared to 0. →	Input clock supply for timer array unit m is stopped 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.)

Note 1. Be sure to set the TSH11 and TSH13 bits to its initial value.

Note 2. Be sure to set the TTH11 and TTH13 bits to its initial value.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

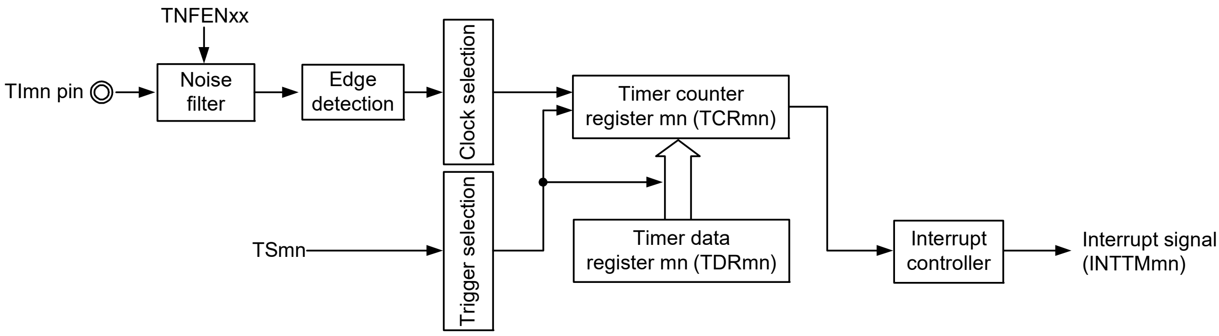
6.8.2 Operation as external event counter

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.

Specified number of counts = Set value of TDRmn + 1

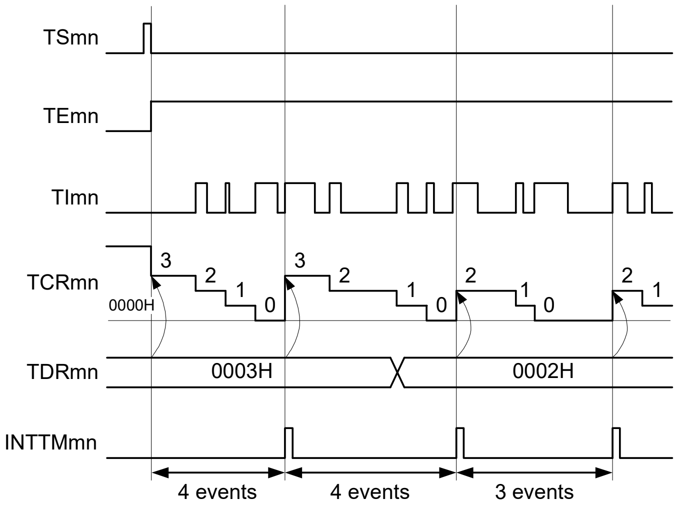
Timer count register mn (TCRmn) operates as a down counter in the event counter mode.
The TCRmn register loads the value of timer data register mn (TDRmn) by setting any channel start trigger bit TSmn of timer channel start register m (TSm) to 1.
The TCRmn register counts down each time the valid input 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 - 53 Block Diagram of Operation as External Event Counter



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

Figure 6 - 54 Example of Basic Timing of Operation as External Event Counter



Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

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 - 55 Example of Set Contents of Registers in External Event Counter Mode (1/2)

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	M/S Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	1/0	0	1	0/1	0	0	0	1/0	1/0	0	0	0	1	1	0

Operation mode of channel n
011B Event count mode

Setting of operation when counting is started
0: Neither generates INTTMmn nor inverts timer output when counting is started.

Selection of TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Start trigger selection
000B: Selects only software start.

Setting of MASTERmn bit (channel 2)
0: Independent channel operation function.
Setting of SPLIT0n bit (channels 1 and 3 of unit 0)
0: 16-bit timer mode
1: 8-bit timer mode

Count clock selection
1: Selects the TImn pin input valid edge.

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected using channels 1 and 3 in the 8-bit timer mode).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected using channels 1 and 3 in the 8-bit timer mode).

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	0

0: Outputs 0 from TOmn

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	0

0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn
	0

0: Cleared to 0 when TOMmn = 0 (master channel output mode).

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn
	0

0: Sets master channel output mode.

Note TMRm2: MASTER0n bit
TMR01, TMR03 SPLIT0n bit
TMR11: Fixed to 0
TMRm0: Fixed to 0

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

Figure 6 - 56 Operation Procedure When External Event Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit m is stopped (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Input clock supply for timer array unit m is supplied. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable registers 1, 2 (NFEN1, NFEN2) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). Sets number of counts to timer data register mn (TDRmn). Clears the TOEmn bit of timer output enable register m (TOEm) to 0.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and count operation starts. Value of the TDRmn register is loaded to timer count register mn (TCRmn) and detection of the TImn pin input edge is awaited.
During operation	Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TMRmn register, TOMmn, TOLmn, TOmn, and TOEmn bits cannot be changed.	Counter (TCRmn) counts down each time input edge of the TImn pin has been detected. When count value reaches 0000H, the value of the TDRmn register is loaded to the TCRmn register again, and the count operation is continued. By detecting TCRmn = 0000H, the INTTMmn output is generated. After that, the above operation is repeated.
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0. →	Input clock supply for timer array unit m is stopped All circuits are initialized and SFR of each channel is also initialized.

(Operation is resumed: from Operation stop back to Operation start.)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

6.8.3 Operation as input pulse interval measurement

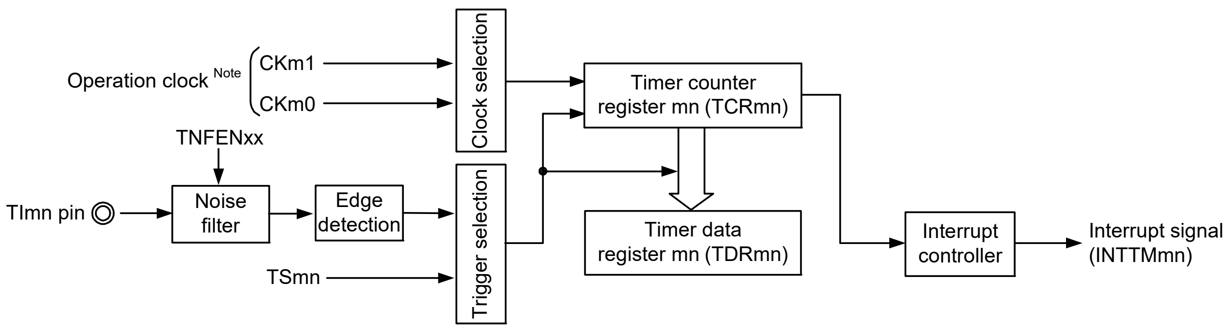
The count value can be captured at the TImn valid edge and the interval of the pulse input to TImn can be measured. In addition, the count value can be captured by using software operation (TSmn = 1) as a capture trigger while the TEmn bit is set to 1.
The pulse interval can be calculated by the following expression.

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

Caution The TImn pin input is sampled using the operating clock selected with the CKSmn bit of timer mode register mn (TMRmn), so an error of up to one operating clock cycle occurs.

Timer count register mn (TCRmn) operates as an up counter in the capture mode.
When the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, the TCRmn register counts up from 0000H in synchronization with the count clock.
When the TImn pin input valid edge is detected, the count value of the TCRmn register is transferred (captured) to timer data register mn (TDRmn) and, at the same time, the TCRmn register is cleared to 0000H, and the INTTMmn is output. If the counter overflows at this time, the OVF bit of timer status register mn (TSRmn) is set to 1. If the counter does not overflow, the OVF bit is cleared. After that, the above operation is repeated.
As soon as the count value has been captured to the TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.
If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSRmn register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.
Set the STSmn2 to STSmn0 bits of the TMRmn register to 001B to use the valid edges of TImn as a start trigger and a capture trigger.

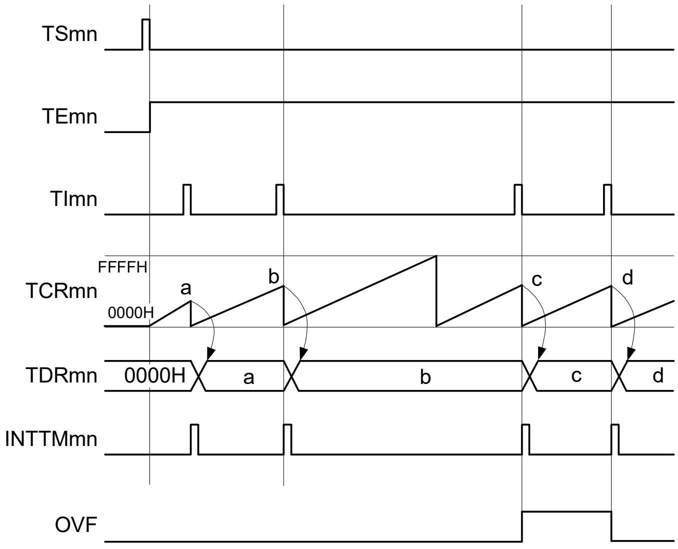
Figure 6 - 57 Block Diagram of Operation as Input Pulse Interval Measurement



Note When channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

Figure 6 - 58 Example of Basic Timing of Operation as Input Pulse Interval Measurement (MDmn0 = 0)



Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

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 - 59 Example of Set Contents of Registers to Measure Input Pulse Interval

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	M/S Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	0	0	0	0	1	1/0	1/0	0	0	0	1	0	1/0

Operation mode of channel n
010B: Capture mode

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

Selection of TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

Capture trigger selection
001B: Selects the TImn pin input valid edge.

Setting of MASTERmn bit (channel 2)
0: Independent channel operation function.
Setting of SPLIT0n bit (channels 1 and 3 of unit 0)
0: 16-bit timer mode

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	0

0: Stops TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn
	0

0: Cleared to 0 when master channel output mode (TOMmn = 0)

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn
	0

0: Sets master channel output mode.

Note
TMRm2: MASTERmn bit
TMR01, TMR03: SPLIT0n bit
TMR11: Fixed to 0
TMRm0: Fixed to 0

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

Figure 6 - 60 Operation Procedure When Input Pulse Interval Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit m is stopped (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Input clock supply for timer array unit m is supplied. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable registers 1, 2 (NFEN1, NFEN2) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel).	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and count operation starts. Timer count register mn (TCRmn) is cleared to 0000H. 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. →	Input clock supply for timer array unit m is stopped All circuits are initialized and SFR of each channel is also initialized.

(Operation is resumed: arrow from Operation stop back to Channel default setting/Operation start.)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

6.8.4 Operation as input signal high-/low-level width measurement

Caution When using a channel to implement the LIN-bus, set bit 1 (ISC1) of the input switch control register (ISC) to 1. In the following descriptions, read TImn as RxD0.

By starting counting at one edge of the TImn pin input and capturing the number of counts at another edge, the signal width (high-level width/low-level width) of TImn can be measured. The signal width of TImn can be calculated by the following expression.

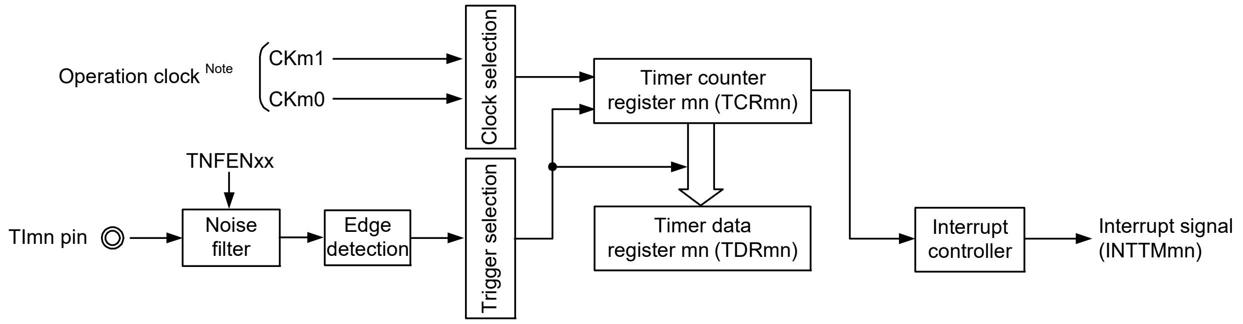
Signal width of TImn input = Period of count clock × ((10000H × TSRmn: OVF) + (Capture value of TDRmn + 1))

Caution The TImn pin input is sampled using the operating clock selected with the CKSmn bit of timer mode register mn (TMRmn), so an error equivalent to one operation clock occurs.

Timer count register mn (TCRmn) operates as an up counter in the capture & one-count mode.
When the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, the TEmn bit is set to 1 and the TImn pin start edge detection wait status is set.
When the TImn pin input start edge (rising edge of the TImn pin input when the high-level width is to be measured) is detected, the counter counts up from 0000H in synchronization with the count clock. When the valid capture edge (falling edge of the TImn pin input when the high-level width is to be measured) is detected later, the count value is transferred to timer data register mn (TDRmn) and, at the same time, INTTMmn is output. If the counter overflows at this time, the OVF bit of timer status register mn (TSRmn) is set to 1. If the counter does not overflow, the OVF bit is cleared. The TCRmn register stops at the value “value transferred to the TDRmn register + 1”, and the TImn pin start edge detection wait status is set. After that, the above operation is repeated.
As soon as the count value has been captured to the TDRmn register, the OVF bit of the TSRmn register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.
If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF bit of the TSRmn register is set to 1. However, a normal interval value cannot be measured for the OVF bit, if two or more overflows occur.
Whether the high-level width or low-level width of the TImn pin is to be measured can be selected by using the CISmn1 and CISmn0 bits of the TMRmn register.
Because this function is used to measure the signal width of the TImn pin input, the TSmn bit cannot be set to 1 while the TEmn bit is 1.

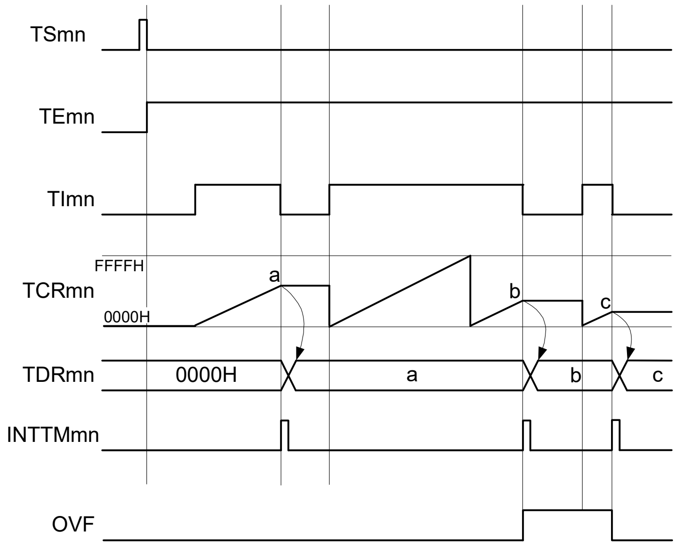
CISmn1, CISmn0 of TMRmn register = 10B: Low-level width is measured.
CISmn1, CISmn0 of TMRmn register = 11B: High-level width is measured.

Figure 6 - 61 Block Diagram of Operation as Input Signal High-/Low-Level Width Measurement



Note For channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Figure 6 - 62 Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement



Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

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 - 63 Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	M/S Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	0	0	0	1	0	1	1/0	0	0	1	1	0	0

Operation mode of channel n
110B: Capture & one-count

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

Selection of TImn pin input edge
10B: Both edges (to measure low-level width)
11B: Both edges (to measure high-level width)

Start trigger selection
010B: Selects the TImn pin input valid edge.

Setting of MASTERmn bit (channel 2)
0: Independent channel operation function.
Setting of SPLIT0n bit (channels 1 and 3 of unit 0)
0: 16-bit timer mode

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected channels 1 and 3).

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	0

0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn
	0

0: Cleared to 0 when master channel output mode (TOMmn = 0)

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn
	0

0: Sets master channel output mode.

Note

TMRm2:	MASTERmn bit
TMR01, TMR03:	SPLIT0n bit
TMR11:	Fixed to 0
TMRm0:	Fixed to 0

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

Figure 6 - 64 Operation Procedure When Input Signal High-/Low-Level Width Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit m is stopped (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Input clock supply for timer array unit m is supplied. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable registers 1, 2 (NFEN1, NFEN2) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). Clears the TOEmn bit to 0 and stops operation of TOmn.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and the TImn pin start edge detection wait status is set.
	Detects the TImn pin input count start valid edge. →	Clears timer count register mn (TCRmn) to 0000H and starts counting up.
During operation	Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TMRmn register, TOMmn, TOLmn, TOmn, and TOEmn bits cannot be changed.	When the TImn pin start edge is detected, the counter (TCRmn) counts up from 0000H. If a capture edge of the TImn pin is detected, the count value is transferred to timer data register mn (TDRmn) and INTTMmn is generated. If an overflow occurs at this time, the OVF bit of timer status register mn (TSRmn) is set; if an overflow does not occur, the OVF bit is cleared. The TCRmn register stops the count operation until the next TImn pin start edge is detected.
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. The TCRmn register holds count value and stops. The OVF bit of the TSRmn register is also held.
TAU stop	The TAUmEN bit of the PER0 register is cleared to 0. →	Input clock supply for timer array unit m is stopped All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed. (Operation stop → Operation start)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

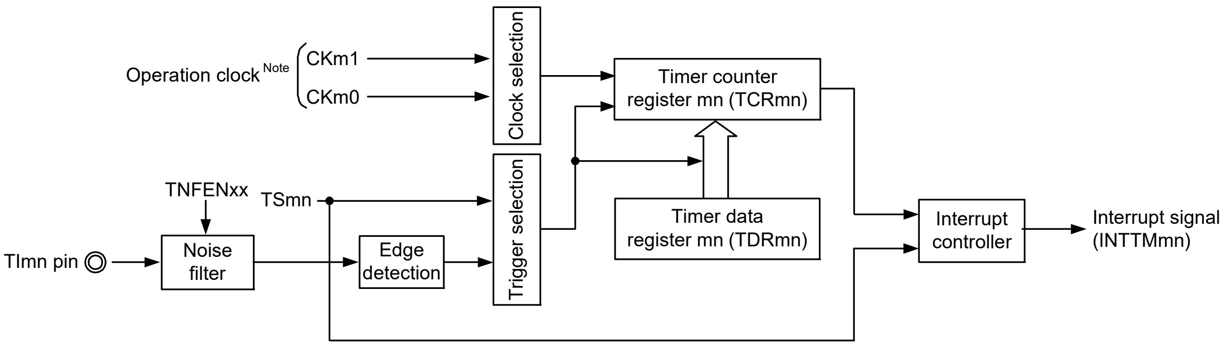
6.8.5 Operation as delay counter

Counting down can be started 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.
Counting down can also be started by setting TSmn to 1 by using software and generate INTTMmn (timer interrupt) at any interval while TEmn is 1.
The interrupt generation period can be calculated by the following expression.

Generation period of INTTMmn (timer interrupt) = Period of count clock × (Set value of TDRmn + 1)

Timer count register mn (TCRmn) operates as a down counter in the one-count mode.
When the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, the TEmn bit is 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.

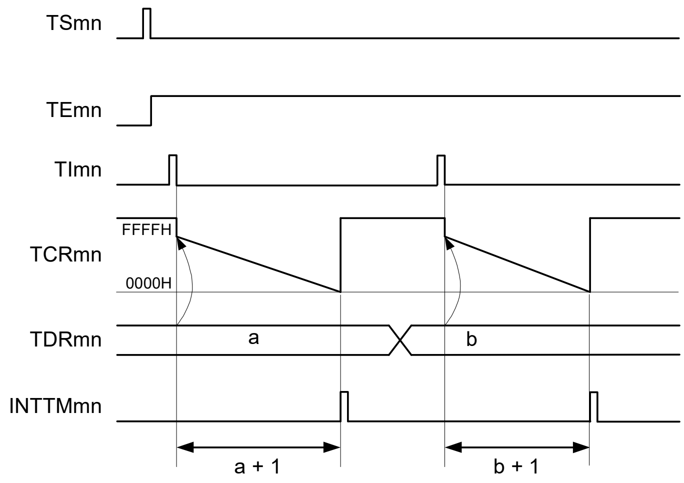
Figure 6 - 65 Block Diagram of Operation as Delay Counter



Note For using channels 1 and 3, the clock can be selected from CKm0, CKm1, CKm2 and CKm3.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

Figure 6 - 66 Example of Basic Timing of Operation as Delay Counter



Remark 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

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 - 67 Example of Set Contents of Registers to Delay Counter

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	M/S Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	1/0	0	0	0/1	0	0	1	1/0	1/0	0	0	1	0	0	1/0

Operation mode of channel n
100B: One-count mode

Start trigger during operation
0: Trigger input is invalid.
1: Trigger input is valid.

Selection of TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

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

Setting of MASTERmn bit (channel 2)
0: Independent channel operation function.
Setting of SPLIT0n bit (channels 1 and 3 of unit 0)
0: 16-bit timer mode
1: 8-bit timer mode

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.
01B: Selects CKm2 as operation clock of channels 1, 3 (This can only be selected using channels 1 and 3 in the 8-bit timer mode).
11B: Selects CKm3 as operation clock of channels 1, 3 (This can only be selected using channels 1 and 3 in the 8-bit timer mode).

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	0

0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn
	0

0: Cleared to 0 when master channel output mode (TOMmn = 0)

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn
	0

0: Sets master channel output mode.

Note TMRm2: MASTERmn bit
TMR01, TMR03: SPLIT0n bit
TMR11: Fixed to 0
TMRm0: Fixed to 0

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

Figure 6 - 68 Operation Procedure When Delay Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit m is stopped (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Input clock supply for timer array unit m is supplied. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 to CKm3.	
Channel default setting	Sets the corresponding bit of the noise filter enable registers 1, 2 (NFEN1, NFEN2) to 0 (off) or 1 (on). Sets timer mode register mn (TMRmn) (determines operation mode of channel). INTTMmn output delay is set to timer data register mn (TDRmn). Clears the TOEmn bit to 0 and stops operation of TOmn.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and the 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. →	Input clock supply for timer array unit m is stopped All circuits are initialized and SFR of each channel is also initialized.

(Operation is resumed: Operation stop → Operation start)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 10, 11

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.

Delay time = {Set value of TDRmn (master) + 2} × Count clock period Pulse width = {Set value of TDRmp (slave)} × Count clock period

The master channel operates in the one-count mode and counts the delays. Timer count register mn (TCRmn) of the master channel starts operating upon start trigger detection and loads the value of timer data register mn (TDRmn).

The TCRmn register counts down from the value of the TDRmn register it has loaded, in synchronization with the count clock. When TCRmn = 0000H, it outputs INTTMmn and stops counting until the next start trigger is detected.

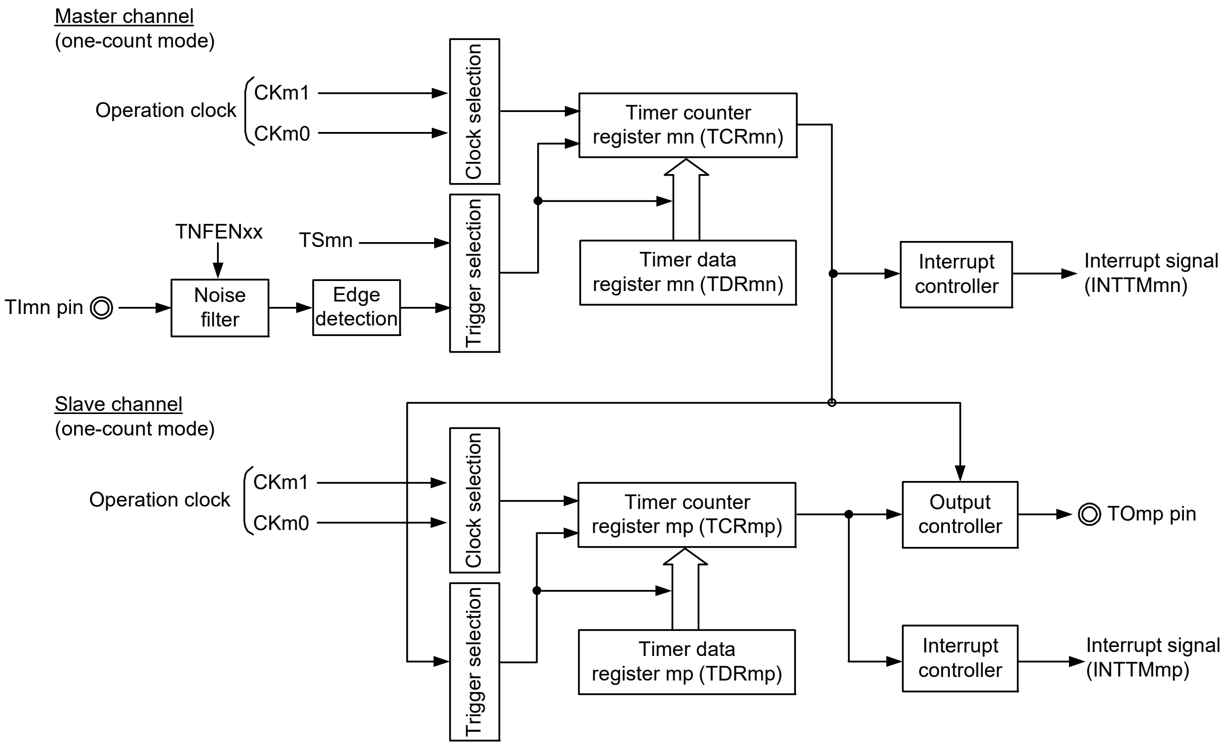
The slave channel operates in the one-count mode and counts the pulse width. The TCRmp register of the slave channel starts operation using INTTMmn of the master channel as a start trigger, and loads the value of the TDRmp register. The TCRmp register counts down from the value of The TDRmp register it has loaded, in synchronization with the count value. When count value = 0000H, it outputs INTTMmp and stops counting until the next start trigger (INTTMmn of the master channel) is detected. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.

Instead of using the TImn pin input, a one-shot pulse can also be output using the software operation (TSmn = 1) as a start trigger.

Caution Since the timing for loading of the TDRmn register of the master channel will be different from that for loading of the TDRmp register of the slave channel, writing to the TDRmn or TDRmp register while counting is in progress may lead to contention that causes an illegal waveform to be output. Only write new values to the TDRmn register after INTTMmn has been generated and to the TDRmp register after INTTMmp has been generated.

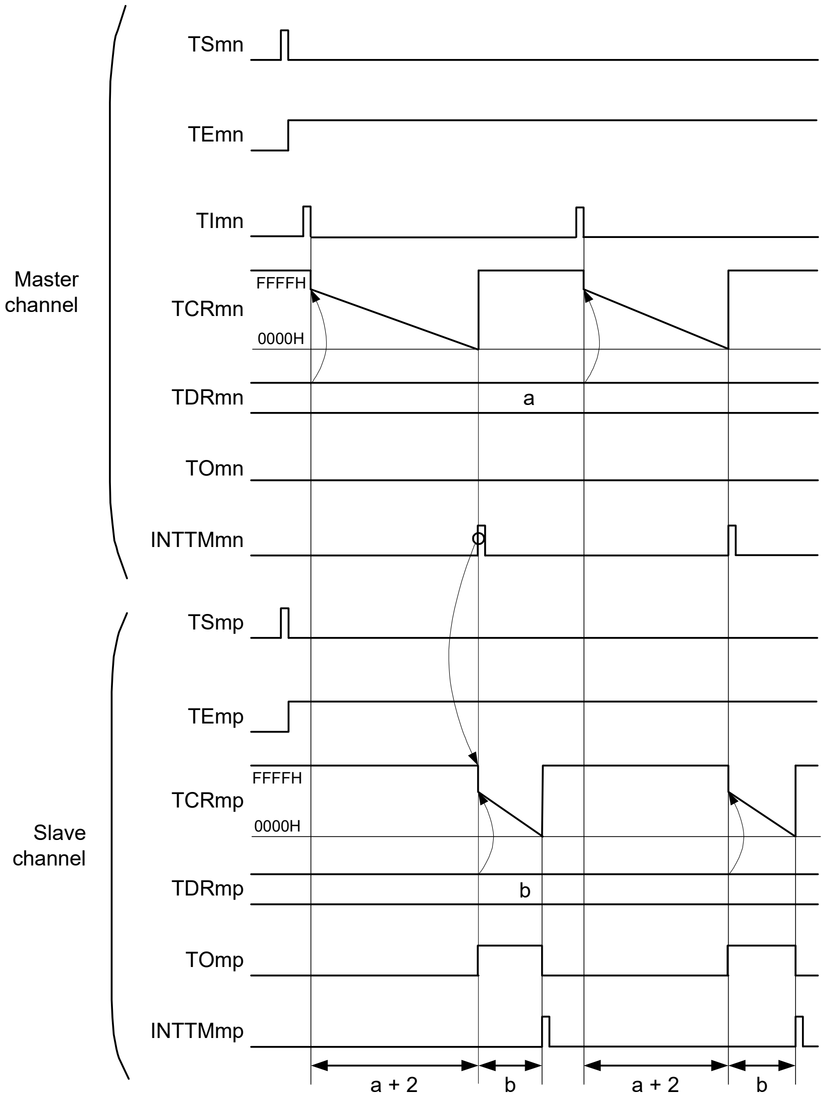
Remark m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2), mn = 00, 02, 10
p: Slave channel number (p = 1 to 3)
Unit 0: mp = 01 to 03 when n = 0
mp = 03 when n = 2
Unit 1: mp = 11 when n = 0

Figure 6 - 69 Block Diagram of Operation as One-Shot Pulse Output Function



Remark m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2), mn = 00, 02, 10
p: Slave channel number (p = 1 to 3)
Unit 0: mp = 01 to 03 when n = 0
mp = 03 when n = 2
Unit 1: mp = 11 when n = 0

Figure 6 - 70 Example of Basic Timing of Operation as One-Shot Pulse Output Function



Remark 1. m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2), mn = 00, 02, 10
p: Slave channel number (p = 1 to 3)
Unit 0: mp = 01 to 03 when n = 0
mp = 03 when n = 2
Unit 1: mp = 11 when n = 0

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 - 71 Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Master Channel)

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	MASTERmn Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	0	1	0	0	1	1/0	1/0	0	0	1	0	0	0

Operation mode of channel n
100B: One-count mode

Start trigger during operation
0: Trigger input is invalid.

Selection of TImn pin input edge
00B: Detects falling edge.
01B: Detects rising edge.
10B: Detects both edges.
11B: Setting prohibited

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

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

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channels n.
10B: Selects CKm1 as operation clock of channels n.

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	0

0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn
	0

0: Cleared to 0 when master channel output mode (TOMmn = 0)

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn
	0

0: Sets master channel output mode.

Note TMRm2: MASTERmn = 1
TMRm0: Fixed to 0

Remark m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2), mn = 00, 02, 10

Figure 6 - 72 Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Slave Channel)

(a) Timer mode register mp (TMRmp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmp	CKSmp1	CKSmp0		CCSmp	M/S Note	STSmp2	STSmp1	STSmp0	CISmp1	CISmp0			MDmp3	MDmp2	MDmp1	MDmp0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0

Operation mode of channel p
100B: One-count mode

Start trigger during operation
0: Trigger input is invalid.

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

Start trigger selection
100B: Selects INTTMmn of master channel.

Setting of MASTERmp bit (channel 2)
0: Slave channel
Setting of SPLIT0p bit (channels 1 and 3 of unit 0)
0: 16-bit timer mode

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel p.
10B: Selects CKm1 as operation clock of channel p.
* Make the same setting as master channel.

(b) Timer output register m (TOm)

	Bit p
TOm	TOmp
	1/0

0: Outputs 0 from TOmp.
1: Outputs 1 from TOmp.

(c) Timer output enable register m (TOEm)

	Bit p
TOEm	TOEmp
	1/0

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

(d) Timer output level register m (TOLm)

	Bit p
TOLm	TOLmp
	1/0

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

(e) Timer output mode register m (TOMm)

	Bit p
TOMm	TOMmp
	1

1: Sets the slave channel output mode.

Note TMRm2: MASTERmp bit
TMR01, TMR03: SPLIT0p bit
TMR11: Fixed to 0

Remark m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2), mn = 00, 02, 10
p: Slave channel number (p = 1 to 3), Unit 0: mp = 01 to 03 when n = 0, mp = 03 when n = 2
Unit 1: mp = 11 when n = 0

Figure 6 - 73 Operation Procedure of One-Shot Pulse Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit m is stopped (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable registers 0 (PER0) to 1. →	Input clock supply for timer array unit m is supplied. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets the corresponding bit of the noise filter enable registers 1, 2 (NFEN1, NFEN2) to 1. Sets timer mode register mn, mp (TMRmn, TMRmp) of two channels to be used (determines operation mode of channels). An output delay is set to timer data register mn (TDRmn) of the master channel, and a pulse width is set to the TDRmp register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOMmp bit of timer output mode register m (TOMm) is set to 1 (slave channel output mode). Sets the TOLmp bit. Sets the TOmp bit and determines default level of the TOmp output. → Sets the TOEmp bit to 1 and enables operation of TOmp. → Clears the port register and port mode register to 0. →	The TOmp pin goes into Hi-Z output state. The TOmp default setting level is output when the port mode register is in output mode and the port register is 0. TOmp does not change because channel stops operating. The TOmp pin outputs the TOmp set level.

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

Figure 6 - 74 Operation Procedure of One-Shot Pulse Output Function (2/2)

(Operation is resumed.)

	Software Operation	Hardware Status
Operation start	Sets the TOEmp bit (slave) to 1 (only when operation is resumed). The TSmn (master) and TSmp (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. → The TSmn and TSmp bits automatically return to 0 because they are trigger bits.	The TEmn and TEmp bits are set to 1 and the master channel enters the start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit of the master channel is set to 1) wait status. Counter stops operating.
	Count operation of the master channel is started by start trigger detection of the master channel. • Detects the TImn pin input valid edge. • Sets the TSmn bit of the master channel to 1 by software Note.	Master channel starts counting.
During operation	Set values of only the CISmn1 and CISmn0 bits of the TMRmn register can be changed. Set values of the TMRmp, TDRmn, TDRmp registers, TOMmn, TOMmp, TOLmn, and TOLmp bits cannot be changed. The TCRmn and TCRmp registers can always be read. The TSRmn and TSRmp registers are not used. Set values of the TOm and TOEm registers by slave channel can be changed.	Master channel loads the value of the TDRmn register to timer count register mn (TCRmn) by the start trigger detection (the valid edge of the TImn pin input is detected or the TSmn bit of the master channel is set to 1), and the counter starts counting down. When the count value reaches TCRmn = 0000H, the INTTMmn output is generated, and the counter stops until the next valid edge is input to the TImn pin. The slave channel, triggered by INTTMmn of the master channel, loads the value of the TDRmp register to the TCRmp register, and the counter starts counting down. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn (master) and TTmp (slave) bits are set to 1 at the same time. → The TTmn (master) and TTmp (slave) bits are set to 1 at the same time.	TEmn, TEmp = 0, and count operation stops. The TCRmn and TCRmp registers hold count value and stop. The TOmp output is not initialized but holds current status.
	The TOEmp bit of slave channel is cleared to 0 and value is set to the TOmp bit. →	The TOmp pin outputs the TOmp set level.
TAU stop	To hold the TOmp pin output level Clears the TOmp bit to 0 after the value to be held is set to the port register. → When holding the TOmp pin output level is not necessary Setting not required.	The TOmp pin output level is held by port function.
	The TAUmEN bit of the PER0 register is cleared to 0. →	Input clock supply for timer array unit m is stopped All circuits are initialized and SFR of each channel is also initialized. (The TOmp bit is cleared to 0 and the TOmp pin is set to port mode.)

Note Do not set the TSmn bit of the slave channel to 1.

Remark m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2), mn = 00, 02, 10
p: Slave channel number (p = 1 to 3) Unit 0: mp = 01 to 03 when n = 0, mp = 03 when n = 2
Unit 1: mp = 11 when n = 0

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.

Pulse period = {Set value of TDRmn (master) + 1} × Count clock period
Duty factor [%] = {Set value of TDRmp (slave)}/{Set value of TDRmn (master) + 1} × 100
0% output: Set value of TDRmp (slave) = 0000H
100% output: Set value of TDRmp (slave) ≥ {Set value of TDRmn (master) + 1}

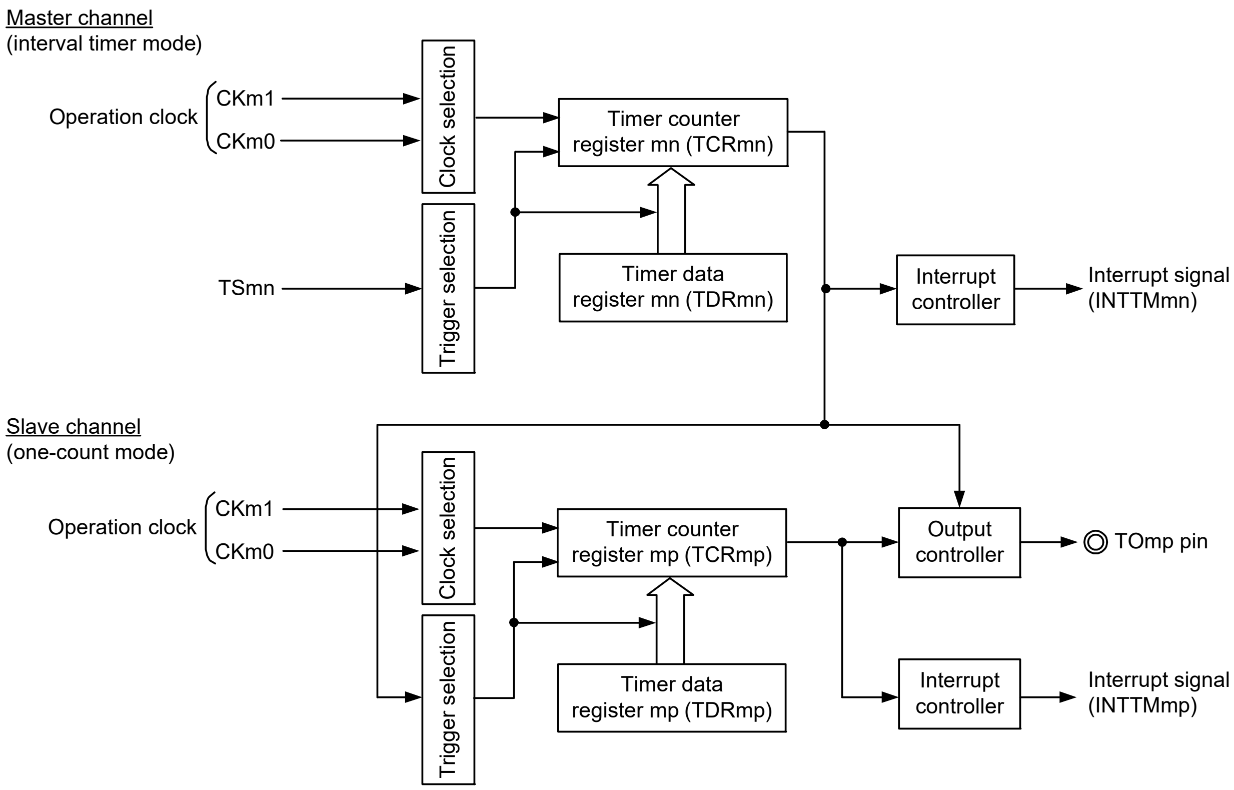
Remark The duty factor exceeds 100% if the set value of TDRmp (slave) > (set value of TDRmn (master) + 1), it summarizes to 100% output.

The master channel operates in the interval timer mode. If the channel start trigger bit (TSmn) of timer channel start register m (TSm) is set to 1, an interrupt (INTTMmn) is output, the value set to timer data register mn (TDRmn) is loaded to timer count register mn (TCRmn), and the counter counts down in synchronization with the count clock. When the counter reaches 0000H, INTTMmn is output, the value of the TDRmn register is loaded again to the TCRmn register, and the counter counts down. This operation is repeated until the channel stop trigger bit (TTmn) of timer channel stop register m (TTm) is set to 1.
If two channels are used to output a PWM waveform, the period until the master channel counts down to 0000H is the PWM output (TOmp) cycle.
The slave channel operates in one-count mode. By using INTTMmn from the master channel as a start trigger, the TCRmp register loads the value of the TDRmp register and the counter counts down to 0000H. When the counter reaches 0000H, it outputs INTTMmp and waits until the next start trigger (INTTMmn from the master channel) is generated.
If two channels are used to output a PWM waveform, the period until the slave channel counts down to 0000H is the PWM output (TOmp) duty.
PWM output (TOmp) goes to the active level one clock after the master channel generates INTTMmn and goes to the inactive level when the TCRmp register of the slave channel becomes 0000H.

Caution To rewrite both timer data register mn (TDRmn) of the master channel and the TDRmp register of the slave channel, a write access is necessary two times. The timing at which the values of the TDRmn and TDRmp registers are loaded to the TCRmn and TCRmp registers is upon occurrence of INTTMmn of the master channel. Thus, when rewriting is performed split before and after occurrence of INTTMmn of the master channel, the TOmp pin cannot output the expected waveform. To rewrite both the TDRmn register of the master and the TDRmp register of the slave, therefore, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel.

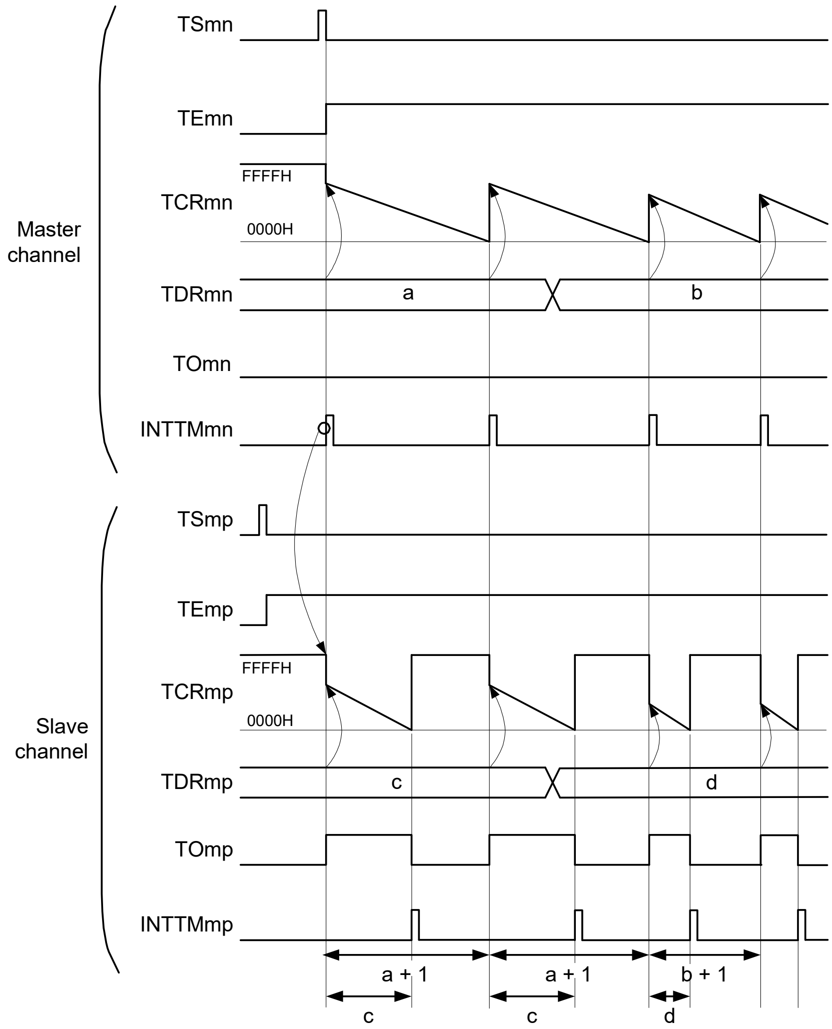
Remark m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2), mn = 00, 02, 10
p: Slave channel number (p = 1 to 3)
Unit 0: mp = 01 to 03 when n = 0
mp = 03 when n = 2
Unit 1: mp = 11 when n = 0

Figure 6 - 75 Block Diagram of Operation as PWM Function



Remark m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2), mn = 00, 02, 10
p: Slave channel number (p = 1 to 3)
Unit 0: mp = 01 to 03 when n = 0
mp = 03 when n = 2
Unit 1: mp = 11 when n = 0

Figure 6 - 76 Example of Basic Timing of Operation as PWM Function



Remark 1. m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2), mn = 00, 02, 10
p: Slave channel number (p = 1 to 3)
Unit 0: mp = 01 to 03 when n = 0
mp = 03 when n = 2
Unit 1: mp = 11 when n = 0

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 - 77 Example of Set Contents of Registers When PWM Function (Master Channel) Is Used

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	MASTERmn Note	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
1: Generates INTTMmn when counting is started.

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

Start trigger selection
000B: Selects only software start.

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

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	0

0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn
	0

0: Cleared to 0 when master channel output mode (TOMmn = 0)

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn
	0

0: Sets master channel output mode.

Note TMRm2: MASTERmn = 1
TMRm0: Fixed to 0

Remark m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2), mn = 00, 02, 10

Figure 6 - 78 Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used

(a) Timer mode register mp (TMRmp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmp	CKSmp1	CKSmp0		CCSmp	M/S Note	STSmp2	STSmp1	STSmp0	CISmp1	CISmp0			MDmp3	MDmp2	MDmp1	MDmp0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

Operation mode of channel p
100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

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

Start trigger selection
100B: Selects INTTMmn of master channel.

Setting of MASTERmp bit (channel 2)
0: Slave channel
Setting of SPLIT0p bit (channels 1 and 3 of unit 0)
0: 16-bit timer mode

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel p.
10B: Selects CKm1 as operation clock of channel p.
* Make the same setting as master channel.

(b) Timer output register m (TOm)

	Bit p
TOm	TOmp
	1/0

0: Outputs 0 from TOmp.
1: Outputs 1 from TOmp.

(c) Timer output enable register m (TOEm)

	Bit p
TOEm	TOEmp
	1/0

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

(d) Timer output level register m (TOLm)

	Bit p
TOLm	TOLmp
	1/0

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

(e) Timer output mode register m (TOMm)

	Bit p
TOMm	TOMmp
	1

1: Sets the slave channel output mode.

Note TMRm2: MASTERmp bit
TMR01, TMR03: SPLIT0p bit
TMR11: Fixed to 0

Remark m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2), mn = 00, 02, 10
p: Slave channel number (p = 1 to 3)
Unit 0: mp = 01 to 03 when n = 0
mp = 03 when n = 2
Unit 1: mp = 11 when n = 0

Figure 6 - 79 Operation Procedure When PWM Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit m is stopped (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Input clock supply for timer array unit m is supplied. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets timer mode registers mn, mp (TMRmn, TMRmp) of two channels to be used (determines operation mode of channels). An interval (period) value is set to timer data register mn (TDRmn) of the master channel, and a duty factor is set to the TDRmp register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOMmp bit of timer output mode register m (TOMm) is set to 1 (slave channel output mode). Sets the TOLmp bit. Sets the TOmp bit and determines default level of the TOmp output. →	The TOmp pin goes into Hi-Z output state. The TOmp default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets the TOEmp bit to 1 and enables operation of TOmp. →	TOmp does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOmp pin outputs the TOmp set level.

(Remark is listed on the next page.)

Figure 6 - 80 Operation Procedure When PWM Function Is Used (2/2)

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

Operation is resumed. (from Operation stop back to Operation start)

Remark m: Unit number (m = 0, 1), n: Master channel number (n = 0, 2), mn = 00, 02, 10
p: Slave channel number (p = 1 to 3) Unit 0: mp = 01 to 03 when n = 0, mn = 03 when n = 2
Unit 1: mp = 11 when n = 0

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.

Pulse period = {Set value of TDRmn (master) + 1} × Count clock period
Duty factor 1 [%] = {Set value of TDRmp (slave 1)}/{Set value of TDRmn (master) + 1} × 100
Duty factor 2 [%] = {Set value of TDRmq (slave 2)}/{Set value of TDRmn (master) + 1} × 100

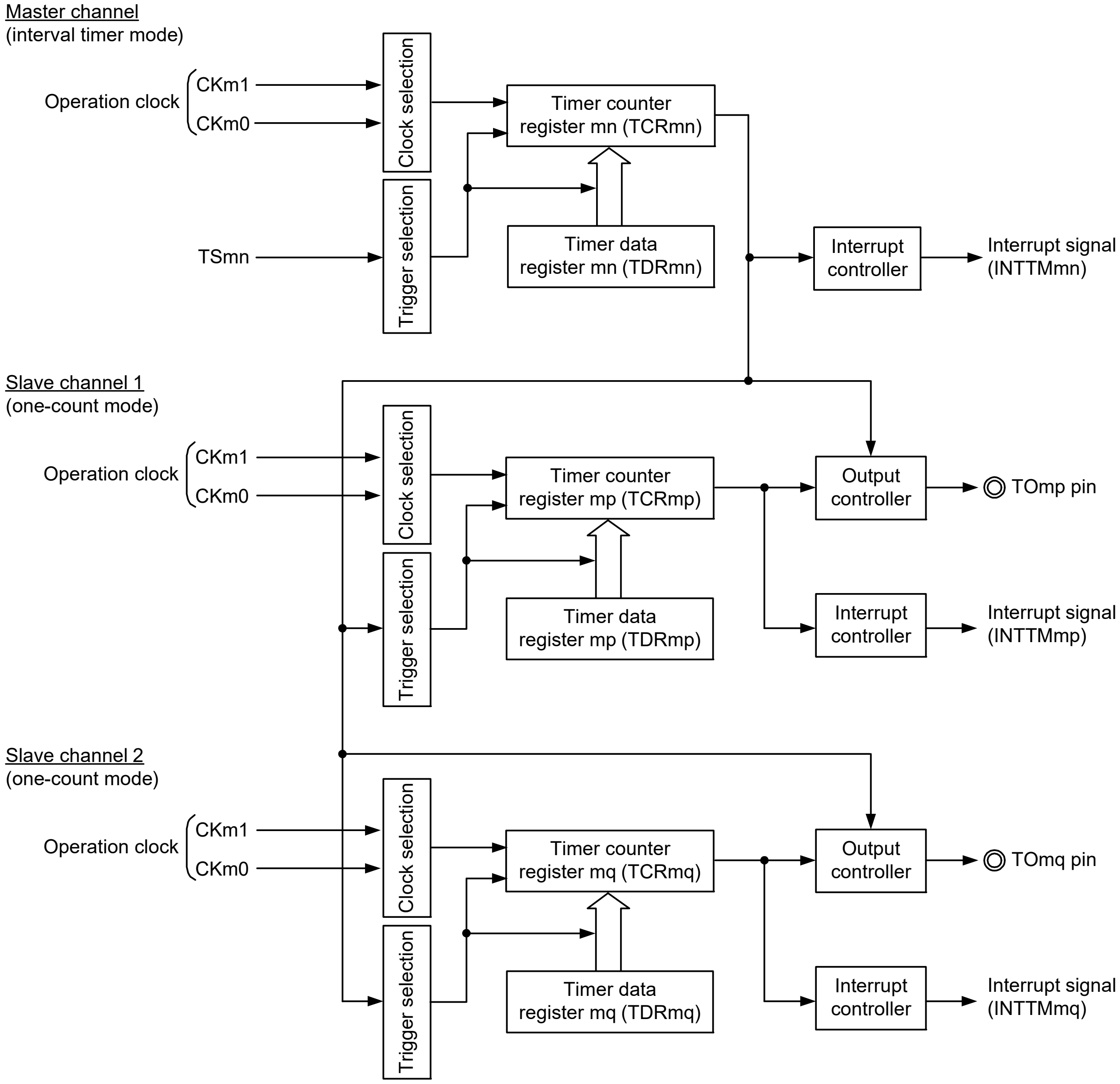
Remark Although the duty factor exceeds 100% if the set value of TDRmp (slave 1) > {set value of TDRmn (master) + 1} or if the {set value of TDRmq (slave 2)} > {set value of TDRmn (master) + 1}, it is summarized into 100% output.

Timer count register mn (TCRmn) of the master channel operates in the interval timer mode and counts the periods.
The TCRmp register of the slave channel 1 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmp pin. The TCRmp register loads the value of timer data register mp (TDRmp), using INTTMmn of the master channel as a start trigger, and starts counting down. When TCRmp = 0000H, TCRmp outputs INTTMmp and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of TOmp becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmp = 0000H.
In the same way as the TCRmp register of the slave channel 1, the TCRmq register of the slave channel 2 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TOmq pin. The TCRmq register loads the value of the TDRmq register, using INTTMmn of the master channel as a start trigger, and starts counting down. When TCRmq = 0000H, the TCRmq register outputs INTTMmq and stops counting until the next start trigger (INTTMmn of the master channel) has been input. The output level of TOmq becomes active one count clock after generation of INTTMmn from the master channel, and inactive when TCRmq = 0000H.
When channel 0 is used as the master channel as above, up to three types of PWM signals can be output at the same time.

Caution **To rewrite both timer data register mn (TDRmn) of the master channel and the TDRmp register of the slave channel 1, write access is necessary at least twice. Since the values of the TDRmn and TDRmp registers are loaded to the TCRmn and TCRmp registers after INTTMmn is generated from the master channel, if rewriting is performed separately before and after generation of INTTMmn from the master channel, the TOmp pin cannot output the expected waveform. To rewrite both the TDRmn register of the master and the TDRmp register of the slave, be sure to rewrite both the registers immediately after INTTMmn is generated from the master channel (This applies also to the TDRmq register of the slave channel 2).**

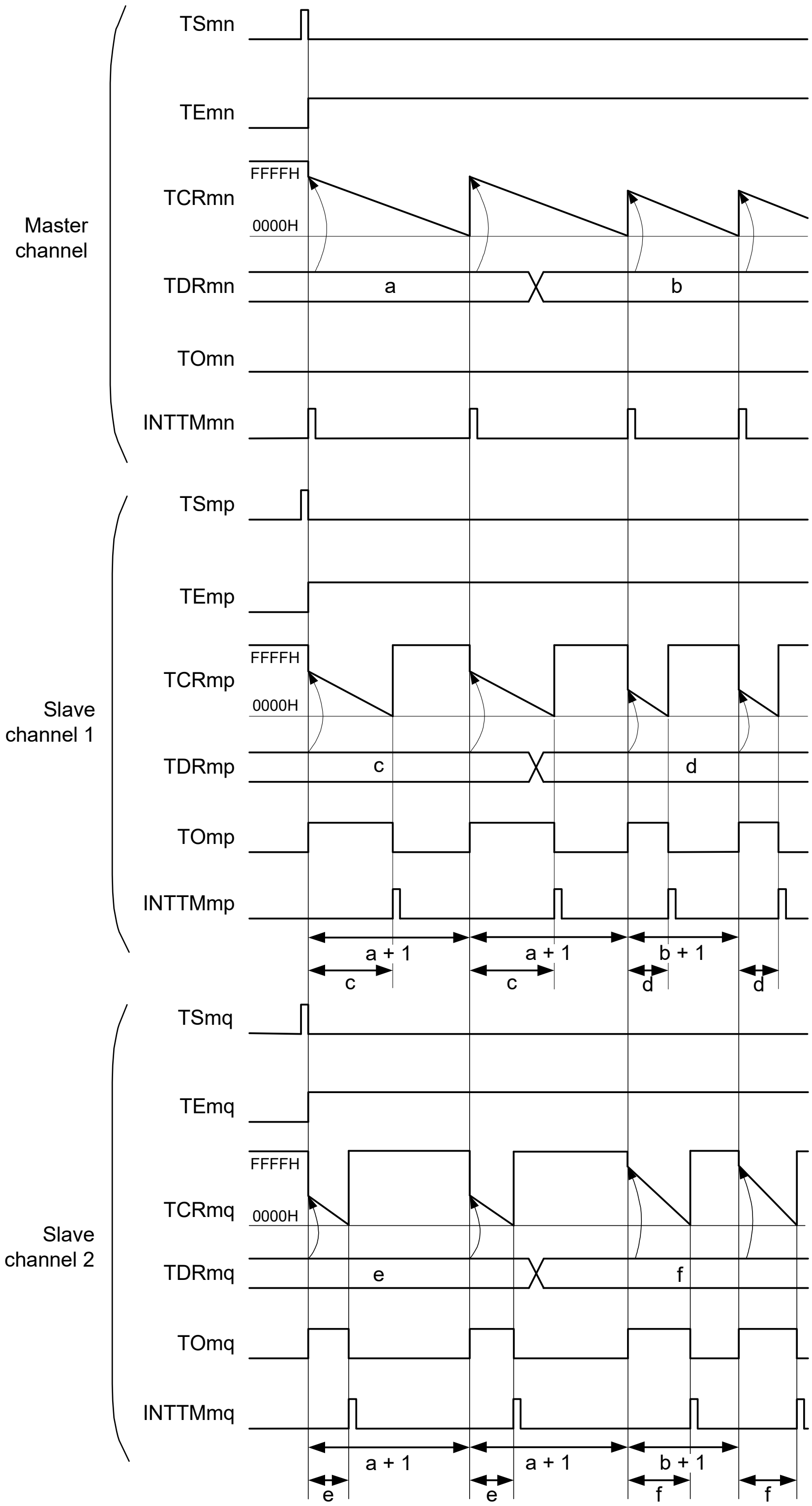
Remark m: Unit number (m = 0), n: Master channel number (n = 0)
p: Slave channel number 1, q: Slave channel number 2
$n < p < q \leq 3$ (Where p and q are integers greater than n)

Figure 6 - 81 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)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 3 (Where p and q are integers greater than n)

Figure 6 - 82 Example of Basic Timing of Operation as Multiple PWM Output Function (Output two types of PWMs)



(Remark is listed on the next page.)

Remark 1. m: Unit number (m = 0), n: Master channel number (n = 0)
p: Slave channel number 1, q: Slave channel number 2
n < p < q ≤ 3 (Where p and q are integers greater than n)

Remark 2.

TSmn, TSmp, TSmq:	Bit n, p, q of timer channel start register m (TSm)
TEmn, TEmp, TEmq:	Bit n, p, q of timer channel enable status register m (TEm)
TCRmn, TCRmp, TCRmq:	Timer count registers mn, mp, mq (TCRmn, TCRmp, TCRmq)
TDRmn, TDRmp, TDRmq:	Timer data registers mn, mp, mq (TDRmn, TDRmp, TDRmq)
TOmn, TOmp, TOmq:	TOmn, TOmp, and TOmq pins output signal

Figure 6 - 83 Example of Set Contents of Registers When Multiple PWM Output Function (Master Channel) Is Used

(a) Timer mode register mn (TMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmn	CKSmn1	CKSmn0		CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
	1/0	0	0	0	Note 1	0	0	0	0	0	0	0	0	0	0	1

Operation mode of channel n
000B: Interval timer

Setting of operation when counting is started
1: Generates INTTMmn when counting is started.

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

Start trigger selection
000B: Selects only software start.

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

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel n.
10B: Selects CKm1 as operation clock of channel n.

(b) Timer output register m (TOm)

	Bit n
TOm	TOmn
	0

0: Outputs 0 from TOmn.

(c) Timer output enable register m (TOEm)

	Bit n
TOEm	TOEmn
	0

0: Stops the TOmn output operation by counting operation.

(d) Timer output level register m (TOLm)

	Bit n
TOLm	TOLmn
	0

0: Cleared to 0 when master channel output mode (TOMmn = 0)

(e) Timer output mode register m (TOMm)

	Bit n
TOMm	TOMmn
	0

0: Sets master channel output mode.

Note TMRm2: MASTERmn = 1
TMRm0: Fixed to 0

Remark m: Unit number (m = 0), n: Master channel number (n = 0)

Figure 6 - 84 Example of Set Contents of Registers When Multiple PWM Output Function (Slave Channel) Is Used (output two types of PWMs)

(a) Timer mode register mp, mq (TMRmp, TMRmq)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMRmp	CKSmp1	CKSmp0		CCSmp	M/S Note	STSmp2	STSmp1	STSmp0	CISmp1	CISmp0			MDmp3	MDmp2	MDmp1	MDmp0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

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

Operation mode of channel p, q
100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

Selection of TImp and TImq pins input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTMmn of master channel.

Setting of MASTERmp and MASTERmq bits (channel 2)
0: Slave channel
Setting of SPLITmp and SPLITmq bits (channels 1, 3)
0: 16-bit timer mode.

Count clock selection
0: Selects operation clock (f_{MCK}).

Operation clock (f_{MCK}) selection
00B: Selects CKm0 as operation clock of channel p, q.
10B: Selects CKm1 as operation clock of channel p, q.
* Make the same setting as master channel.

(b) Timer output register m (TOm)

	Bit q	Bit p
TOm	TOmq	TOmp
	1/0	1/0

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

(c) Timer output enable register m (TOEm)

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

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

(d) Timer output level register m (TOLm)

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

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

(e) Timer output mode register m (TOMm)

	Bit q	Bit p
TOMm	TOMmq	TOMmp
	1	1

1: Sets the slave channel output mode.

Note TMRm2: MASTERmp, MASTERmq bit
TMRm1, TMRm3: SPLITmp, SPLITmq bit

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

Figure 6 - 85 Operation Procedure When Multiple PWM Output Function Is Used (output two types of PWMs) (1/2)

	Software Operation	Hardware Status
TAU default setting		Input clock supply for timer array unit m is stopped (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAUmEN bit of peripheral enable register 0 (PER0) to 1. →	Input clock supply for timer array unit m is supplied. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets timer clock select register m (TPSm). Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets timer mode registers mn, mp, mq (TMRmn, TMRmp, TMRmq) of each channel to be used (determines operation mode of channels). An interval (period) value is set to timer data register mn (TDRmn) of the master channel, and a duty factor is set to the TDRmp and TDRmq registers of the slave channels.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channels. The TOMmp and TOMmq bits of timer output mode register m (TOMm) are set to 1 (slave channel output mode). Clears the TOLmp and TOLmq bits to 0. Sets the TOmp and TOmq bits and determines default level of the TOmp and TOmq outputs. → Sets the TOEmp and TOEmq bits to 1 and enables operation of TOmp and TOmq. → Clears the port register and port mode register to 0. →	The TOmp and TOmq pins go into Hi-Z output state. The TOmp and TOmq default setting levels are output when the port mode register is in output mode and the port register is 0. TOmp and TOmq do not change because channels stop operating. The TOmp and TOmq pins output the TOmp and TOmq set levels.

(Remark is listed on the next page.)

Figure 6 - 86 Operation Procedure When Multiple PWM Output Function Is Used (output two types of PWMs) (2/2)

	Software Operation	Hardware Status
Operation start	(Sets the TOEmp and TOEmq (slave) bits to 1 only when resuming operation.) The TSmn bit (master), and TSmp and TSmq (slave) bits of timer channel start register m (TSm) are set to 1 at the same time. → The TSmn, TSmp, and TSmq bits automatically return to 0 because they are trigger bits.	TEmn = 1, TEmp, TEmq = 1 When the master channel starts counting, INTTMmn is generated. Triggered by this interrupt, the slave channel also starts counting.
During operation	Set values of the TMRmn, TMRmp, TMRmq registers, TOMmn, TOMmp, TOMmq, TOLmn, TOLmp, and TOLmq bits cannot be changed. Set values of the TDRmn, TDRmp, and TDRmq registers can be changed after INTTMmn of the master channel is generated. The TCRmn, TCRmp, and TCRmq registers can always be read. The TSRmn, TSRmp, and TSRmq registers are not used.	The counter of the master channel loads the TDRmn register value to timer count register mn (TCRmn) and counts down. When the count value reaches TCRmn = 0000H, INTTMmn output is generated. At the same time, the value of the TDRmn register is loaded to the TCRmn register, and the counter starts counting down again. At the slave channel 1, the values of the TDRmp register are transferred to the TCRmp register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output levels of TOmp become active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmp = 0000H, and the counting operation is stopped. At the slave channel 2, the values of the TDRmq register are transferred to TCRmq register, triggered by INTTMmn of the master channel, and the counter starts counting down. The output levels of TOmq become active one count clock after generation of the INTTMmn output from the master channel. It becomes inactive when TCRmq = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TTmn bit (master), TTmp, and TTmq (slave) bits are set to 1 at the same time. → The TTmn, TTmp, and TTmq bits automatically return to 0 because they are trigger bits.	TEmn, TEmp, TEmq = 0, and count operation stops. The TCRmn, TCRmp, and TCRmq registers hold count value and stop. The TOmp and TOmq output are not initialized but hold current status.
	The TOEmp and TOEmq bits of slave channels are cleared to 0 and value is set to the TOmp and TOmq bits. →	The TOmp and TOmq pins output the TOmp and TOmq set levels.
TAU stop	To hold the TOmp and TOmq pin output levels Clears the TOmp and TOmq bits to 0 after the value to be held is set to the port register. → When holding the TOmp and TOmq pin output levels are not necessary Setting not required	The TOmp and TOmq pin output levels are held by port function.
	The TAUmEN bit of the PER0 register is cleared to 0. →	Input clock supply for timer array unit m is stopped All circuits are initialized and SFR of each channel is also initialized. (The TOmp and TOmq bits are cleared to 0 and the TOmp and TOmq pins are set to port mode.)

(Operation is resumed: arrow from Operation stop back to Operation start.)

Remark m: Unit number (m = 0), n: Master channel number (n = 0)
p: Slave channel number, q: Slave channel number
n < p < q ≤ 3 (Where p and q are integer greater than n)

6.10 Cautions When Using Timer Array Unit

6.10.1 Cautions When Using Timer output

Depends on products, a pin is assigned a timer output and other alternate functions. In this case, outputs of the other alternate functions must be set in initial status.

For details, see **4.5 Register Settings When Using Alternate Function**.

CHAPTER 7 TIMER RJ

7.1 Functions of Timer RJ

Timer RJ is a 16-bit timer that can be used for pulse output, external pulse width or period measurement, and counting external events.

This 16-bit timer consists of a reload register and a down counter. The reload register and the down counter are allocated to the same address, and they can be accessed by accessing the TRJ0 register.

Table 7 - 1 lists the Timer RJ Specifications. Figure 7 - 1 shows the Timer RJ Block Diagram.

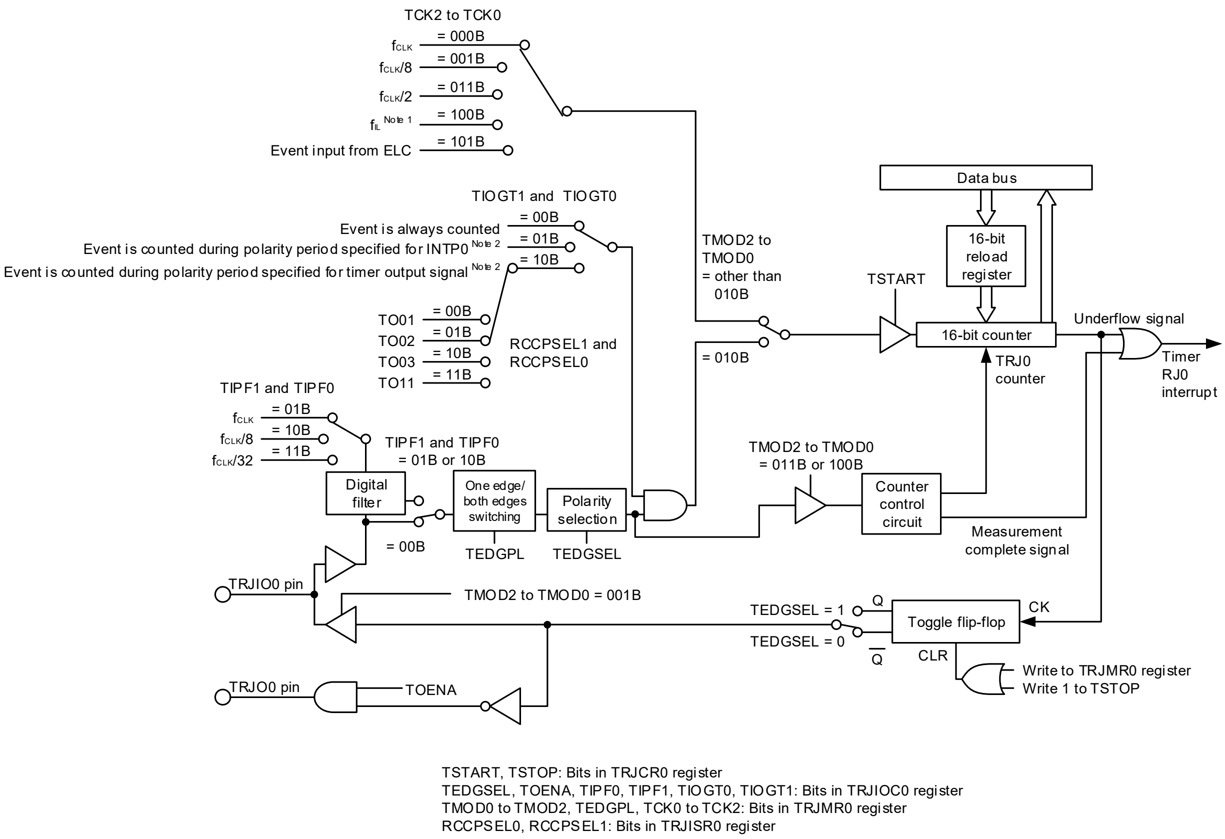
Table 7 - 1 Timer RJ Specifications

Item		Description
Operating modes	Timer mode	The count source is counted.
	Pulse output mode	The count source is counted and the output is inverted at each underflow of the timer.
	Event counter mode	An external event is counted. Operation is possible in STOP mode.
	Pulse width measurement mode	An external pulse width is measured.
	Pulse period measurement mode	An external pulse period is measured.
Count source (Operating clock)		f_{CLK}, $f_{CLK}/2$, $f_{CLK}/8$, f_{IL}, or event input from the event link controller (ELC) selectable
Interrupt		• When the counter underflows. • When the measurement of the active width of the external input (TRJIO0) is completed in pulse width measurement mode. • When the set edge of the external input (TRJIO0) is input in pulse period measurement mode.
Selectable functions		• Coordination with the event link controller (ELC). Event input from the ELC is selectable as a count source.

7.2 Configuration of Timer RJ

Figure 7 - 1 shows the Timer RJ Block Diagram and Table 7 - 2 lists the Timer RJ Pin Configuration.

Figure 7 - 1 Timer RJ Block Diagram



TSTART, TSTOP: Bits in TRJCR0 register
TEDGSEL, TOENA, TIPF0, TIPF1, TIOGT0, TIOGT1: Bits in TRJIOC0 register
TMOD0 to TMOD2, TEDGPL, TCK0 to TCK2: Bits in TRJMR0 register
RCCPSEL0, RCCPSEL1: Bits in TRJISR0 register

Note 1. When selecting fIL as the count source, set the WUTMMCK0 bit in the subsystem clock supply mode control register (OSMC) to 1. However, fIL cannot be selected as the count source for timer RJ when the operating clock (fRTC) specified in the RTC clock select register (RTCCL) is selected as the count source for the real-time clock or the interval timer.

Note 2. The polarity can be selected by the RCCPSEL2 bit in the TRJISR0 register.

Table 7 - 2 Timer RJ Pin Configuration

Pin Name	I/O	Function
INTP0	Input	Event counter mode control for timer RJ
TRJIO0	Input/output	External event input and pulse output for timer RJ
TRJO0	Output	Pulse output for timer RJ

7.3 Registers Controlling Timer RJ

Table 7 - 3 lists the Registers Controlling Timer RJ.

Table 7 - 3 Registers Controlling Timer RJ

Register Name	Symbol
Peripheral enable register 1	PER1
Subsystem clock supply mode control register	OSMC
Timer RJ counter register 0 [Note]	TRJ0
Timer RJ control register 0	TRJCR0
Timer RJ I/O control register 0	TRJIOC0
Timer RJ mode register 0	TRJMR0
Timer RJ event pin select register 0	TRJISR0
Port register 1	P1
Port mode register 1	PM1

Note When the TRJ0 register is accessed, the CPU does not proceed to the next instruction processing but enters the wait state for CPU processing. For this reason, if this wait state occurs, the number of instruction execution clocks is increased by the number of wait clocks. The number of wait clocks for access to the TRJ0 register is one clock for both writing and reading.

7.3.1 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise.
To use timer RJ, be sure to set bit 0 (TRJ0EN) to 1.

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	DACEN	TRGEN	AMPEN	0	DTCEN	PGAEN	AFEEN	TRJ0EN

TRJ0EN	Control of timer RJ0 input clock supply
0	Stops input clock supply. • SFRs used by timer RJ0 cannot be written. • Timer RJ0 is in the reset status.
1	Enables input clock supply. • SFRs used by timer RJ0 can be read and written.

Caution 1. When setting timer RJ, be sure to set the TRJ0EN bit to 1 first. If TRJ0EN = 0, writing to a control register of timer RJ is ignored, and all read values are default values (except for port mode register 1 (PM1) and port register 1 (P1)).
Caution 2. Be sure to set bit 4 to 0.

7.3.2 Subsystem clock supply mode control register (OSMC)

The WUTMMCK0 bit can be used to select the operation clock of real-time clock, interval timer, and timer RJ.
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	0	0	0	WUTMMCK0	0	0	0	0

WUTMMCK0	Selection of operating clock for real-time clock, interval timer, and timer RJ
0	RTC operating clock (f_{RTC}) specified in the RTC clock select register (RTCCL) • The RTC operating clock is selected as the count clock for the real-time clock and the interval timer. • The low-speed on-chip oscillator cannot be selected as the clock source for timer RJ.
1	• The low-speed on-chip oscillator clock is selected as the count clock for the interval timer. • The low-speed on-chip oscillator can be selected as the clock source for timer RJ.

Caution When using the real-time clock, be sure to set the WUTMMCK0 bit to 0.

7.3.3 Timer RJ counter register 0 (TRJ0)

TRJ0 is a 16-bit register. The write value is written to the reload register and the read value is read from the counter.

The states of the reload register and the counter are changed depending on the TSTART bit in the TRJCR0 register. For details, see **7.4.1 Reload register and counter rewrite operation**.

The TRJ0 register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to FFFFH.

Figure 7 - 4 Format of Timer RJ counter register 0 (TRJ0)

Address: F0500H After reset: FFFFH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRJ0																

—	Function	Setting Range
Bits 15 to 0	16-bit counter Notes 1, 2	0000H to FFFFH

Note 1. When 1 is written to the TSTOP bit in the TRJCR0 register, the 16-bit counter is forcibly stopped and set to FFFFH.

Note 2. When the setting of bits TCK2 to TCK0 in the TRJMR0 register is other than 001B (f_{CLK}/8) or 011B (f_{CLK}/2), if the TRJ0 register is set to 0000H, a request signal to the DTC and the ELC is generated only once immediately after the count starts. starts. However, the TRJO0 and TRJIO0 output is toggled.
When the TRJ0 register is set to 0000H in event counter mode, regardless of the value of bits TCK2 to TCK0, a request signal to the DTC and the ELC is generated only once immediately after the count starts.
In addition, the TRJO0 output is toggled even during a period other than the specified count period.
When the TRJ0 register is set to 0000H or a higher value, a request signal is generated each time TRJ underflows.

Caution When the TRJ0 register is accessed, the CPU does not proceed to the next instruction processing but enters the wait state for CPU processing. For this reason, if this wait state occurs, the number of instruction execution clocks is increased by the number of wait clocks. The number of wait clocks for access to the TRJ0 register is one clock for both writing and reading.

7.3.4 Timer RJ control register 0 (TRJCR0)

The TRJCR0 register starts or stops count operation and indicates the status of timer RJ.

The TRJCR0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7 - 5 Format of Timer RJ control register 0 (TRJCR0)

Address: F0240H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TRJCR0	0	0	TUNDF	TEDGF	0	TSTOP	TCSTF	TSTART

TUNDF	Timer RJ underflow flag
0	No underflow
1	Underflow
[Condition for setting to 0] • When 0 is written to this bit by a program. [Condition for setting to 1] • When the counter underflows.	

TEDGF	Active edge judgment flag
0	No active edge received
1	Active edge received
[Condition for setting to 0] • When 0 is written to this bit by a program. [Conditions for setting to 1] • When the measurement of the active width of the external input (TRJIO) is completed in pulse width measurement mode. • The set edge of the external input (TRJIO) is input in pulse period measurement mode.	

TSTOP	Timer RJ count forced stop Note 1
When 1 is written to this bit, the count is forcibly stopped. The read value is 0.	

TCSTF	Timer RJ count status flag Note 2
0	Count stops
1	Count in progress
[Conditions for setting to 0] • When 0 is written to the TSTART bit (the TCSTF bit is set to 0 in synchronization with the count source). • When 1 is written to the TSTOP bit. [Condition for setting to 1] • When 1 is written to the TSTART bit (the TCSTF bit is set to 1 in synchronization with the count source).	

TSTART	Timer RJ count start Note 2
0	Count stops
1	Count starts
Count operation is started by writing 1 to the TSTART bit and stopped by writing 0. When the TSTART bit is set to 1 (count starts), the TCSTF bit is set to 1 (count in progress) in synchronization with the count source. Also, after 0 is written to the TSTART bit, the TCSTF bit is set to 0 (count stops) in synchronization with the count source. For details, see **7.5.1 Count operation start and stop control**.	

Note 1. When 1 (count is forcibly stopped) is written to the TSTOP bit, bits TSTART and TCSTF are initialized at the same time. The pulse output level is also initialized.

Note 2. For notes on using bits TSTART and TCSTF, see **7.5.1 Count operation start and stop control**.

7.3.5 Timer RJ I/O control register 0 (TRJIOC0)

The TRJIOC0 register sets the input/output mode of timer RJ.

The TRJIOC0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7 - 6 Format of Timer RJ I/O control register 0 (TRJIOC0)

Address: F0241H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TRJIOC0	TIOGT1	TIOGT0	TIPF1	TIPF0	0	TOENA	0	TEDGSEL

TIOGT1	TIOGT0	TRJIO count control Notes 1, 2
0	0	Event is always counted
0	1	Event is counted during polarity period specified for INTP0
1	0	Event is counted during polarity period specified for timer output signal
Other than above		Setting prohibited

TIPF1	TIPF0	TRJIO input filter select
0	0	No filter
0	1	Filter sampled at f_{CLK}
1	0	Filter sampled at $f_{CLK}/8$
1	1	Filter sampled at $f_{CLK}/32$
These bits are used to specify the sampling frequency of the filter for the TRJIO input. If the input to the TRJIO0 pin is sampled and the value matches three successive times, that value is taken as the input value.		

TOENA	TRJO output enable
0	TRJO output disabled (port)
1	TRJO output enabled

TEDGSEL	I/O polarity switch
Function varies depending on the operating mode (see **Tables 7 - 4** and **7 - 5**).	

Note 1. When INTP0 or the timer output signal is used, the polarity to count an event can be selected by the RCCPSEL2 bit in the TRJISR0 register.

Note 2. Bits TIOGT0 and TIOGT1 are enabled only in event counter mode.

Table 7 - 4 TRJIO I/O Edge and Polarity Switching

Operating Mode	Function
Timer mode	Not used (I/O port)
Pulse output mode	0: Output is started at high (Initialization level: High) 1: Output is started at low (Initialization level: Low)
Event counter mode	0: Count at rising edge 1: Count at falling edge
Pulse width measurement mode	0: Low-level width is measured 1: High-level width is measured
Pulse period measurement mode	0: Measure from one rising edge to the next rising edge 1: Measure from one falling edge to the next falling edge

Table 7 - 5 TRJO Output Polarity Switching

Operating Mode	Function
All modes	0: Output is started at low (Initialization level: Low) 1: Output is started at high (Initialization level: High)

7.3.6 Timer RJ mode register 0 (TRJMR0)

The TRJMR0 register sets the operating mode of timer RJ.

The TRJMR0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7 - 7 Format of Timer RJ mode register 0 (TRJMR0)

Address: F0242H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TRJMR0	0	TCK2	TCK1	TCK0	TEDGPL	TMOD2	TMOD1	TMOD0

TCK2	TCK1	TCK0	Selection of timer RJ count source Notes 1, 2
0	0	0	f_{CLK}
0	0	1	$f_{CLK}/8$
0	1	1	$f_{CLK}/2$
1	0	0	f_{IL} Note 4
1	0	1	Event input from ELC
Other than above			Setting prohibited

TEDGPL	Selection of TRJIO edge polarity Note 5
0	One edge
1	Both edges

TMOD2	TMOD1	TMOD0	Selection of timer RJ operating mode Note 3
0	0	0	Timer mode
0	0	1	Pulse output mode
0	1	0	Event counter mode
0	1	1	Pulse width measurement mode
1	0	0	Pulse period measurement mode
Other than above			Setting prohibited

Note 1. When event counter mode is selected, the external input (TRJIO) is selected as the count source regardless of the setting of bits TCK0 to TCK2.

Note 2. Do not switch count sources during count operation. Count sources should be switched when both the TSTART and TCSTF bits in the TRJCR0 register are set to 0 (count stops).

Note 3. The operating mode can be changed only when the count is stopped while both the bits TSTART and TCSTF in the TRJCR0 register are set to 0 (count stops). Do not change the operating mode during count operation.

Note 4. When selecting f_{IL} as the count source, set the WUTMMCK0 bit in the subsystem clock supply mode control register (OSMC) to 1.

Note 5. The TEDGPL bit is enabled only in event counter mode.

Note 6. Write access to the TRJMR0 register initializes the output from pins TRJO0 and TRJIO0 of timer RJ. For details about the output level at initialization, refer to the description of **Figure 7 - 6 Format of Timer RJ I/O control register 0 (TRJIOC0)**.

7.3.7 Timer RJ event pin select register 0 (TRJISR0)

The TRJISR0 register selects the timer for controlling the event count period and sets the polarity in event counter mode.

The TRJISR0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7 - 8 Format of Timer RJ event pin select register 0 (TRJISR0)

Address: F0243H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TRJISR0	0	0	0	0	0	RCCPSEL2 Note	RCCPSEL1 Note	RCCPSEL0 Note

RCCPSEL2 Note	Selection of timer output signal and INTP0 polarity
0	An event is counted during the low-level period
1	An event is counted during the high-level period

RCCPSEL1 Note	RCCPSEL0 Note	Selection of timer output signal
0	0	TO01
0	1	TO02
1	0	TO03
1	1	TO11

Note Bits RCCPSEL0 to RCCPSEL2 are enabled only in event counter mode.

7.3.8 Port mode register 1 (PM1)

This register sets input/output mode of port 1 in 1-bit units.
When using a port (such as P11/TRJIO0 and P12/TRJO0) as a timer output pin, set the bit corresponding to the port in port mode register 1 (PM1) and port register 1 (P1) to 0.

Example: When using P11/TRJIO0 for timer output
Set the PM11 bit of port mode register 1 to 0.
Set the P11 bit of port register 1 to 0.

When using a port (such as P11/TRJIO0) as a timer input pin, set the bit corresponding to the port in port mode register 1 (PM1) to 1. At this time, port register 1 (P1) bit may be 0 or 1.

Example: When using P11/TRJIO0 for timer input
Set the PM11 bit of port mode register 1 to 1.
Set the P11 bit of port register 1 to 0 or 1.

The PM1 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to FFH.

Figure 7 - 9 Format of port mode register 1 (PM1)

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	Selection of P1n pin I/O mode (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

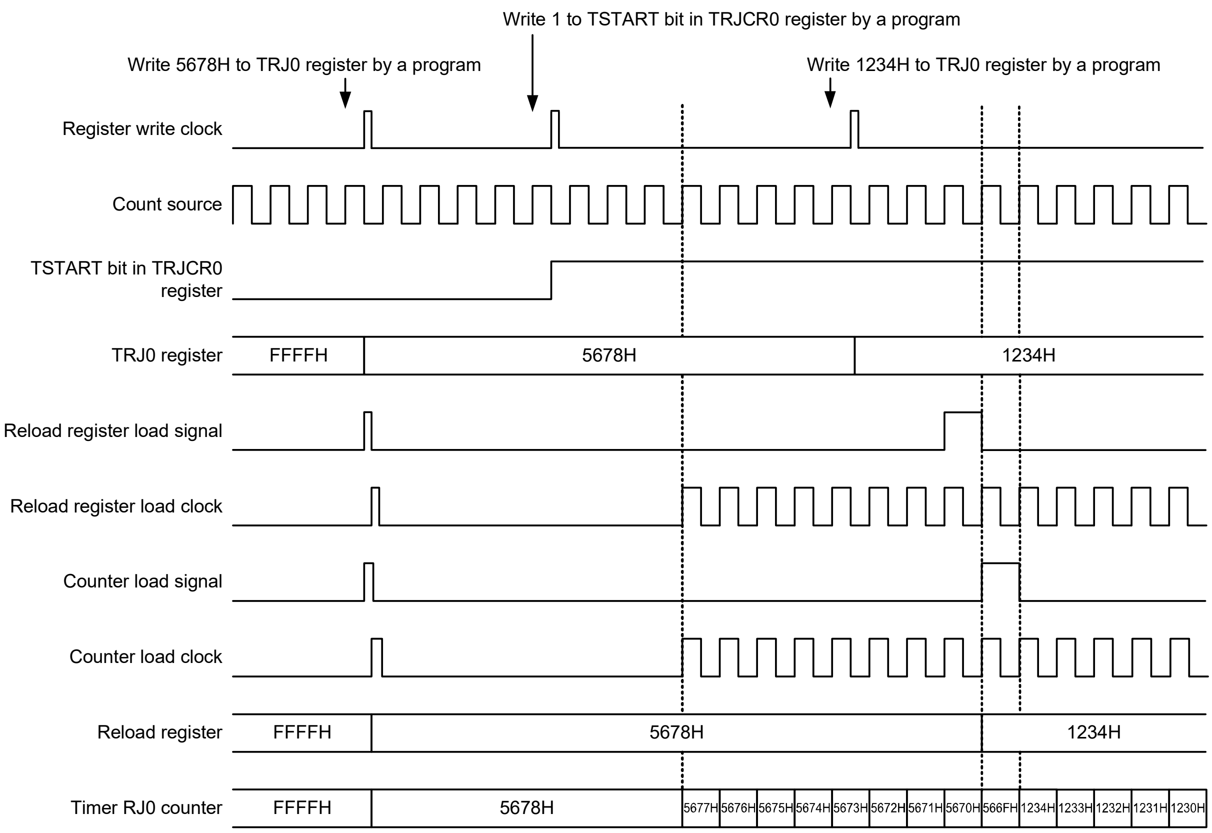
7.4 Timer RJ Operation

7.4.1 Reload register and counter rewrite operation

Regardless of the operating mode, the timing of the rewrite operation to the reload register and the counter differs depending on the value in the TSTART bit in the TRJCR0 register. When the TSTART bit is 0 (count stops), the count value is directly written to the reload register and the counter. When the TSTART bit is 1 (count starts), the value is written to the reload register in synchronization with the count source, and then to the counter in synchronization with the next count source.

Figure 7 - 10 shows the Timing of Rewrite Operation with TSTART Bit Value.

Figure 7 - 10 Timing of Rewrite Operation with TSTART Bit Value



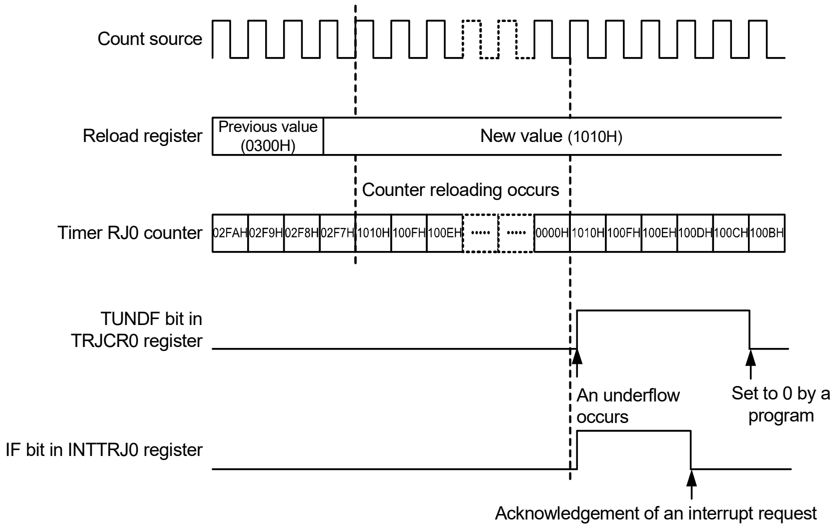
7.4.2 Timer mode

In this mode, the counter is decremented by the count source selected by bits TCK0 to TCK2 in the TRJMR0 register.

In timer mode, the count value is decremented by 1 each time the count source is input. When the count value reaches 0000H and the next count source is input, an underflow occurs and an interrupt request is generated.

Figure 7 - 11 shows the Operation Example in Timer Mode.

Figure 7 - 11 Operation Example in Timer Mode



7.4.3 Pulse output mode

In this mode, the counter is decremented by the count source selected by bits TCK0 to TCK2 in the TRJMR0 register, and the output level of pins TRJIO and TRJO pin is inverted each time an underflow occurs.

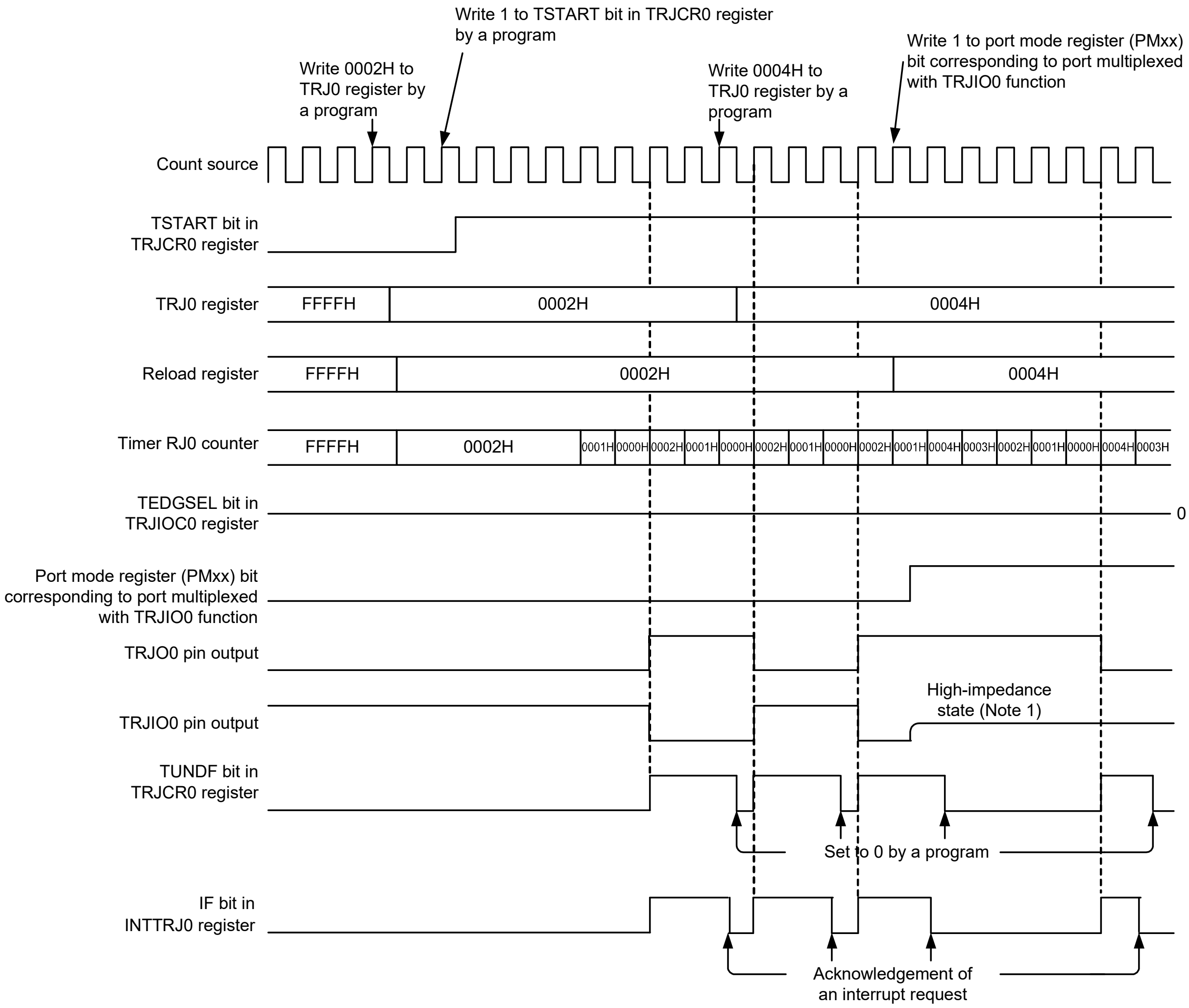
In pulse output mode, the count value is decremented by 1 each time the count source is input. When the count value reaches 0000H and the next count source is input, an underflow occurs and an interrupt request is generated.

In addition, a pulse can be output from pins TRJIO0 and TRJO0. The output level is inverted each time an underflow occurs. The pulse output from the TRJO0 pin can be stopped by the TOENA bit in the TRJIOC0 register.

Also, the output level can be selected by the TEDGSEL bit in the TRJIOC0 register.

Figure 7 - 12 shows the Operation Example in Pulse Output Mode.

Figure 7 - 12 Operation Example in Pulse Output Mode



Note 1: The TRJIO0 pin becomes high impedance by output enable control on the port selected as the TRJIO function.

7.4.4 Event counter mode

In this mode, the counter is decremented by an external event signal (count source) input to the TRJIO0 pin. Various periods for counting events can be set by bits TIOGT0 and TIOGT1 in the TRJIOC0 register and the TRJISR0 register. In addition, the filter function for the TRJIO0 input can be specified by bits TIPF0 and TIPF1 in the TRJIOC0 register.

Also, the output from the TRJO0 pin can be toggled even in event counter mode.

When event counter mode is used, see **7.5.5 Procedure for setting pins TRJO0 and TRJIO0**.

Figure 7 - 13 shows the Example of Operation in Event Counter Mode (1).

Figure 7 - 13 Example of Operation in Event Counter Mode (1)

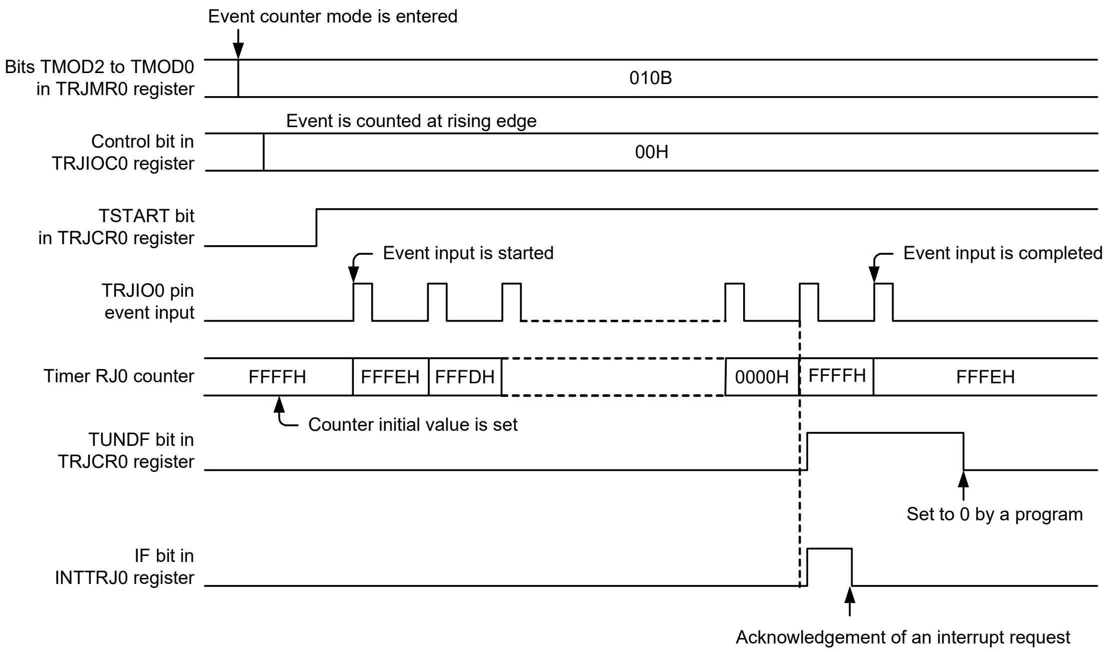
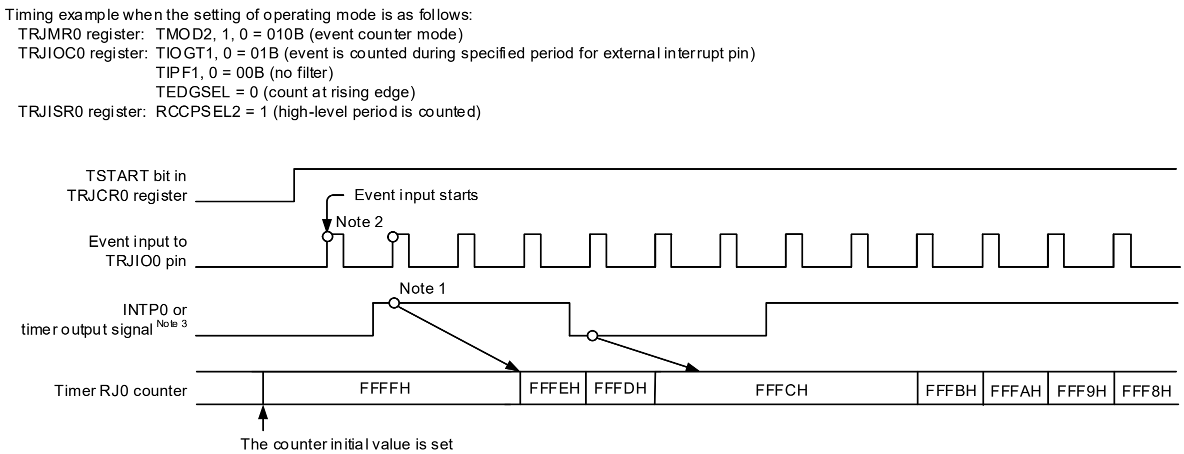


Figure 7 - 14 shows an operation example for counting during the specified period in event counter mode (bits TIOGT1 and TIOGT0 in the TRJIO0 register are set to 01B or 10B).

Figure 7 - 14 Example of Operation in Event Counter Mode (2)



The following notes apply only when bits TIOGT1 and TIOGT0 in the TRJIOC0 register are 01B or 10B for the setting of operating mode in event count mode.

Note 1. To control synchronization, there is a delay of two cycles of the count source until count operation is affected.

Note 2. Count operation may be performed for two cycles of the count source immediately after counting starts, depending on the previous state before counting stops.
To disable counting for two cycles immediately after counting starts, write 1 to the TSTOP bit in the TRJCR0 register to initialize the internal circuit, and then make operation settings before counting starts.

Note 3. For the timer output signal selected by the RCCPSEL1 and RCCPSEL0 bits in the TRJISR0 register, the pin assigned to the timer output function cannot be used as the output of any multiplexed function other than the timer.

7.4.5 Pulse width measurement mode

In this mode, the pulse width of an external signal input to the TRJIO0 pin is measured.

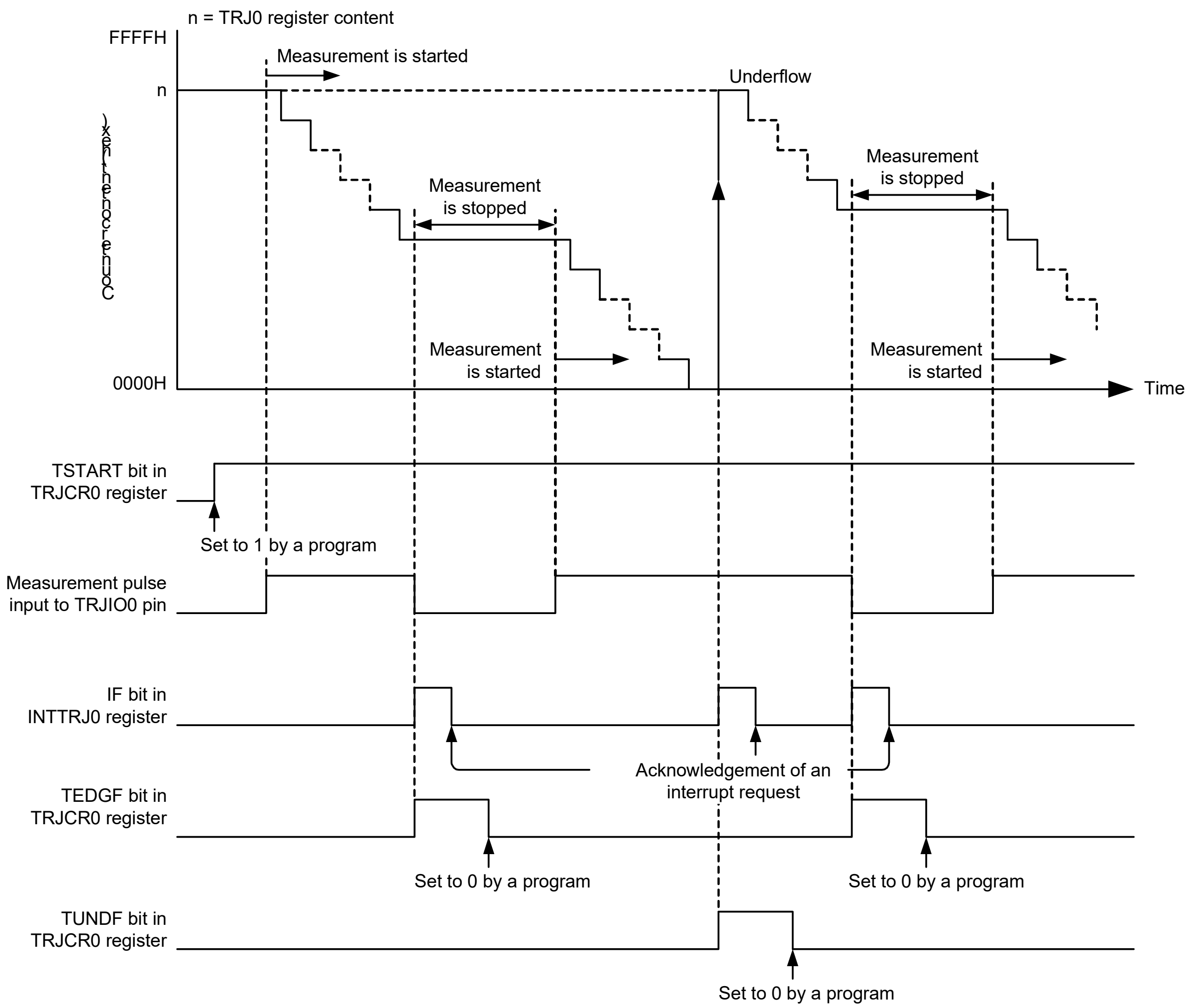
When the level specified by the TEDGSEL bit in the TRJIOC0 register is input to the TRJIO0 pin, the decrement is started with the selected count source. When the specified level on the TRJIO0 pin ends, the counter is stopped, the TEDGF bit in the TRJCR0 register is set to 1 (active edge received), and an interrupt request is generated. The measurement of pulse width data is performed by reading the count value while the counter is stopped. Also, when the counter underflows during measurement, the TUNDF bit in the TRJCR0 register is set to 1 (underflow) and an interrupt request is generated.

Figure 7 - 15 shows the Example of operation in Pulse Width Measurement Mode.

When accessing bits TEDGF and TUNDF in the TRJCR0 register, see **7.5.2 Access to flags (bits TEDGF and TUNDF in TRJCR0 register)**.

Figure 7 - 15 Example of operation in Pulse Width Measurement Mode

This example applies when the high-level width of the measurement pulse is measured (TEDGSEL bit in TRJIOC0 register = 1)



7.4.6 Pulse period measurement mode

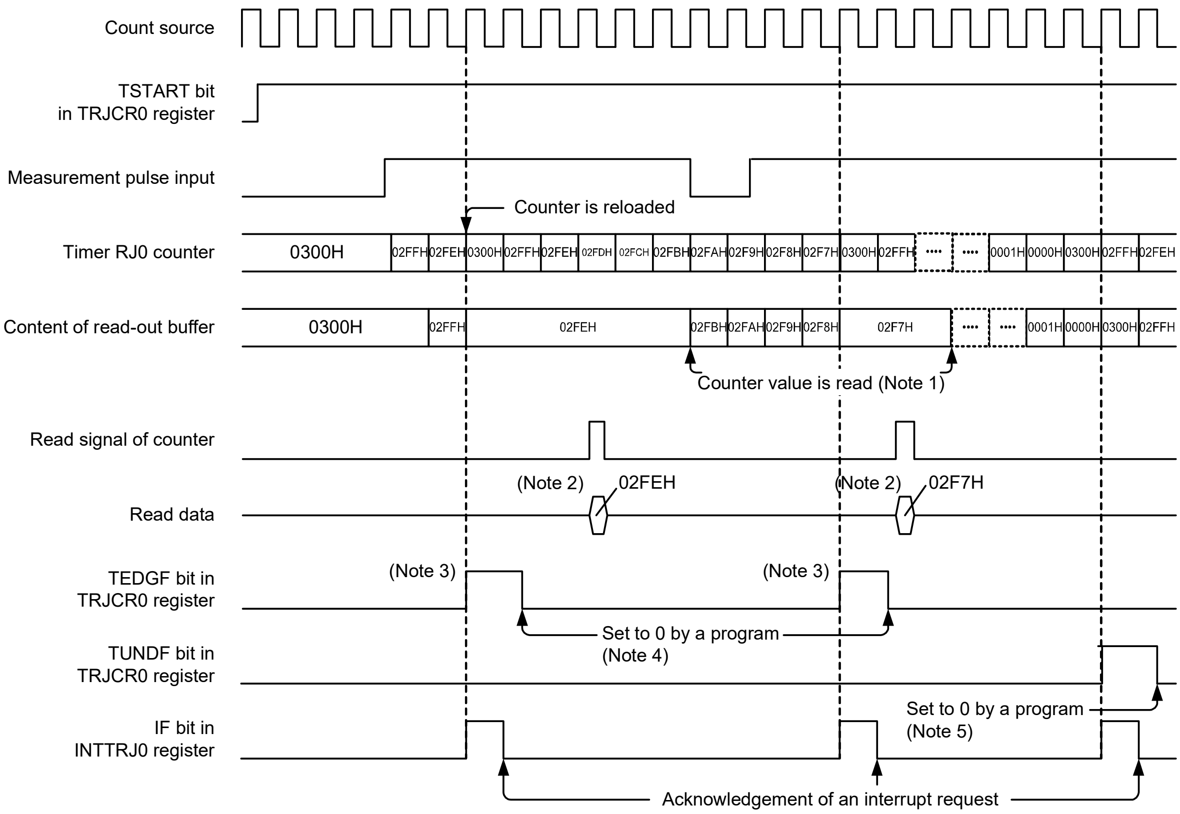
In this mode, the pulse period of an external signal input to the TRJIO0 pin is measured.

The counter is decremented by the count source selected by bits TCK0 to TCK2 in the TRJMR0 register. When a pulse with the period specified by the TEDGSEL bit in the TRJIOC0 register is input to the TRJIO0 pin, the count value is transferred to the read-out buffer at the rising edge of the count source. The value in the reload register is loaded to the counter at the next rising edge. Simultaneously, the TEDGF bit in the TRJCR0 register is set to 1 (active edge received) and an interrupt request is generated. The read-out buffer (TRJ0 register) is read at this time and the difference from the reload value is the period data of the input pulse. The period data is retained until the read-out buffer is read. When the counter underflows, the TUNDF bit in the TRJCR0 register is set to 1 (underflow) and an interrupt request is generated.

Figure 7 - 16 shows the Example of Operation in Pulse Period Measurement Mode.

Only input pulses with a period longer than twice the period of the count source. Also, the low-level and high-level widths must be both longer than the period of the count source. If a pulse period shorter than these conditions is input, the input may be ignored

Figure 7 - 16 Example of Operation in Pulse Period Measurement Mode



This example applies when the initial value of the TRJ0 register is set to 0300H, the TEDGSEL bit in the TRJIOC0 register is set to 0, and the period from one rising edge to the next edge of the measurement pulse is measured.

Note 1. The TRJ0 register must be read during the period from when the TEDGF bit is set to 1 (active edge received) until the next active edge is input. The content of the read-out buffer is retained until the TRJ0 register is read. If it is not read before the active edge is input, the measurement result of the previous period is retained.

Note 2. When the TRJ0 register is read in pulse period measurement mode, the content of the read-out buffer is read.

Note 3. When the active edge of the measurement pulse is input and then the set edge of an external pulse is input, the TEDGF bit in the TRJCR0 register is set to 1 (active edge received).

Note 4. To set this bit to 0 by a program, write 0 to the TEDGF bit in the TRJCR0 register by using an 8-bit memory manipulation instruction.

Note 5. To set this bit to 0 by a program, write 0 to the TUNDF bit in the TRJCR0 register by using an 8-bit memory manipulation instruction.

7.4.7 Coordination with Event Link Controller (ELC)

Through coordination with the ELC, event input from the ELC can be set to be the count source. Bits TCK0 to TCK2 in the TRJMR0 register count at the rising edge of event input from the ELC. However, ELC input does not function in event counter mode.

The ELC setting procedure is shown below:

• Procedure for starting operation

(1) Set the event output destination select register (ELSELRn) for the ELC.
(2) Set the operating mode for the event generation source.
(3) Set the mode for timer RJ.
(4) Start the count operation of timer RJ.
(5) Start the operation of the event generation source.

• Procedure for stopping operation

(1) Stop the operation of the event generation source.
(2) Stop the count operation of timer RJ.
(3) Set the event output destination select register (ELSELRn) for the ELC to 0.

7.4.8 Output settings for each mode

Tables 7 - 6 and 7 - 7 list the states of pins TRJO0 and TRJIO0 in each mode.

Table 7 - 6 TRJO0 Pin Setting

Operating Mode	TRJIOC0 Register		TRJO0 Pin Output
	TOENA Bit	TEDGSEL Bit	
All modes	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled

Table 7 - 7 TRJIO0 Pin Setting

Operating Mode	TRJIOC0 Register		TRJIO0 Pin I/O
	PMXX Bit [Note]	TEDGSEL Bit	
Timer mode	0 or 1	0 or 1	Input (Not used)
Pulse output mode	1	0 or 1	Output disabled (Hi-z output)
	0	1	Normal output
		0	Inverted output
Event counter mode	1	0 or 1	Input
Pulse width measurement mode			
Pulse period measurement mode			

Note The port mode register (PMxx) bit corresponding to port multiplexed with TRJIO0 function.

7.5 Cautions for Timer RJ

7.5.1 Count operation start and stop control

• When event count mode is set or the count source is set to other than the ELC

After 1 (count starts) is written to the TSTART bit in the TRJCR0 register while the count is stopped, the TCSTF bit in the TRJCR0 register remains 0 (count stops) for three cycles of the count source. Do not access the registers associated with timer RJ Note other than the TCSTF bit until this bit is set to 1 (count in progress).

After 0 (count stops) is written to the TSTART bit during a count operation, the TCSTF bit remains 1 for three cycles of the count source. When the TCSTF bit is set to 0, the count is stopped. Do not access the registers associated with timer RJ Note other than the TCSTF bit until this bit is set to 0.

Clear the interrupt register before changing the TATART bit from 0 to 1. Refer to **CHAPTER 22 INTERRUPT FUNCTIONS** for details.

Note Registers associated with timer RJ: TRJ0, TRJCR0, TRJIOC0, TRJMR0, and TRJISR0

• When event count mode is set or the count source is set to the ELC

After 1 (count starts) is written to the TSTART bit in the TRJCR0 register while the count is stopped, the TCSTF bit in the TRJCR0 register remains 0 (count stops) for two cycles of the CPU clock. Do not access the registers associated with timer RJ Note other than the TCSTF bit until this bit is set to 1 (count in progress).

After 0 (count stops) is written to the TSTART bit during a count operation, the TCSTF bit remains 1 for two cycles of the CPU clock. When the TCSTF bit is set to 0, the count is stopped. Do not access the registers associated with timer RJ Note other than the TCSTF bit until this bit is set to 0.

Clear the interrupt register before changing the TATART bit from 0 to 1. Refer to **CHAPTER 22 INTERRUPT FUNCTIONS** for details.

Note Registers associated with timer RJ: TRJ0, TRJCR0, TRJIOC0, TRJMR0, and TRJISR0

7.5.2 Access to flags (bits TEDGF and TUNDF in TRJCR0 register)

Bits TEDGF and TUNDF in the TRJCR0 register are set to 0 by writing 0 by a program, but writing 1 to these bits has no effect. If a read-modify-write instruction is used to set the TRJCR0 register, bits TEDGF and TUNDF may be erroneously set to 0 depending on the timing, even when the TEDGF bit is set to 1 (active edge received) and the TUNDF bit is set to 1 (underflow) during execution of the instruction. Use an 8-bit memory manipulation instruction to access to the TRJCR0 register.

7.5.3 Access to counter register

When bits TSTART and TCSTF in the TRJCR0 register are both 1 (count starts), allow at least three cycles of the count source clock between writes when writing to the TRJ0 register successively.

7.5.4 When changing mode

The registers associated with timer RJ operating mode (TRJIOC0, TRJMR0, and TRJISR0) can be changed only when the count is stopped with both the TSTART and TCSTF bits set to 0 (count stops). Do not change these registers during count operation.

When the registers associated with timer RJ operating mode are changed, the values of bits TSTART and TCSTF are undefined. Write 0 (no active edge received) to the TEDGF bit and 0 (no underflow) to the TUNDF bit before starting the count.

7.5.5 Procedure for setting pins TRJO0 and TRJIO0

After a reset, the I/O ports multiplexed with pins TRJO0 and TRJIO0 function as input ports.

To output from pins TRJO0 and TRJIO0, use the following setting procedure:
Changing procedure

(1) Set the mode.
(2) Set the initial value/output enabled.
(3) Set the port register bits corresponding to pins TRJO0 and TRJIO0 to 0.
(4) Set the port mode register bits corresponding to pins TRJO0 and TRJIO0 to output mode.
(Output is started from pins TRJO0 and TRJIO0)
(5) Start the count (TSTART in TRJCR0 register = 1).

To input from the TRJIO0 pin, use the following setting procedure:

(1) Set the mode.
(2) Set the initial value/edge selected.
(3) Set the port mode register bit corresponding to TRJIO0 pin to input mode.
(Input is started from the TRJIO0 pin)
(4) Start the count (TSTART in TRJMR0 register = 1).
(5) Wait until the TCSTF bit in the TRJCR0 register is set to 1 (count in progress).
(In event counter mode only)
(6) Input an external event from the TRJIO0 pin.
(7) The processing on completion of the first measurement is invalid (the measured value is valid for the second and subsequent times). (In pulse width measurement mode and pulse period measurement mode only)

7.5.6 When timer RJ is not used

When timer RJ is not used, set bits TMOD2 to TMOD0 in the TRJMR0 register to 000B (timer mode) and set the TOENA bit in the TRJIOC0 register to 0 (TRJO output disabled).

7.5.7 When timer RJ operating clock is stopped

Supplying or stopping the timer RJ clock can be controlled by the TRJ0EN bit in the PER1 register. Note that the following SFRs cannot be accessed while the timer RJ clock is stopped. Make sure the timer RJ clock is supplied before accessing any of these registers.
Registers TRJ0, TRJCR0, TRJMR0, TRJIOC0, and TRJISR0.

7.5.8 Procedure for setting STOP mode (event counter mode)

To perform event counter mode operation during STOP mode, first supply the timer RJ clock and then use the following procedure to enter STOP mode.

Setting procedure

(1) Set the operating mode.
(2) Start the count (TSTART = 1, TCSTF = 1).
(3) Stop supplying the timer RJ clock.

To stop event counter mode operation during STOP mode, use the following procedure to stop operation.

(1) Supply the timer RJ clock.
(2) Stop the count (TSTART = 0, TCSTF = 0)

7.5.9 Functional restriction in STOP mode (event counter mode only)

When event counter mode operation is performed during STOP mode, the digital filter function cannot be used.

7.5.10 When count is forcibly stopped by TSTOP bit

After the counter is forcibly stopped by the TSTOP bit in the TRJCR0 register, do not access the following SFRs for one cycle of the count source.

Registers TRJ0, TRJCR0, and TRJMR0

7.5.11 Digital filter

When the digital filter is used, do not start timer operation for five cycles of the digital filter clock after setting bits TIPF1 and TIPF0.

Also, do not start timer operation for five cycles of the digital filter clock when the TEDGSEL bit in the TRJIOC register is changed while the digital filter is used.

7.5.12 When selecting f_{IL} as count source

When selecting f_{IL} as the count source, set the WUTMMCK0 bit in the subsystem clock supply mode control register (OSMC) to 1. However, f_{IL} cannot be selected as the count source for timer RJ when the operating clock (f_{RTC}) specified in the RTC clock select register (RTCCL) is selected as the count source for the real-time clock or the interval timer.

CHAPTER 8 TIMER RG

8.1 Functions of Timer RG

Timer RG supports the following three modes:

- Timer mode:
  - Input capture function: Count at the rising edge, falling edge, or both rising/falling edges
  - Output compare function: Low output/high output/toggle output
- PWM mode: PWM output available with any duty cycle
- Phase counting mode: Automatic measurement available for the counts of the two-phase encoder

8.2 Configuration of Timer RG

Figure 8 - 1 shows the Timer RG Block Diagram and Table 8 - 1 lists the Timer RG Pin Configuration.

Figure 8 - 1 Timer RG Block Diagram

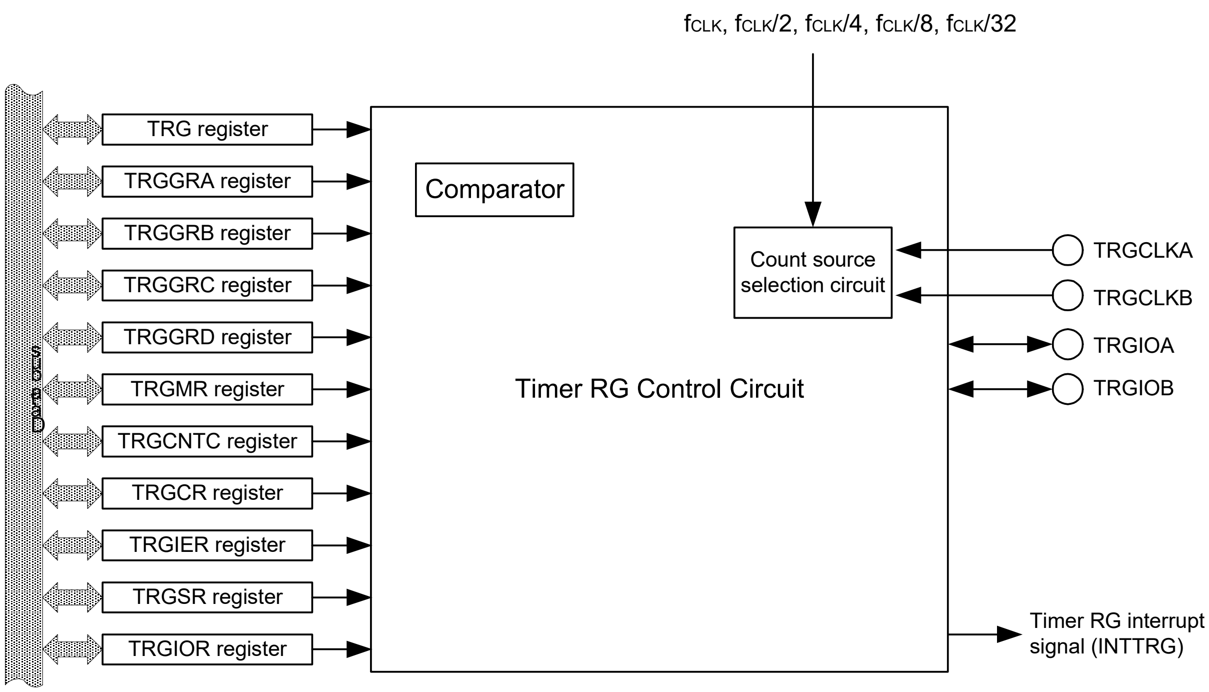


Table 8 - 1 Timer RG Pin Configuration

Pin Name	Alternate Port Name	I/O	Function
TRGCLKA	P11	Input	• In phase counting mode A-phase input • In other than phase counting mode External clock A input
TRGCLKB	P15	Input	• In phase counting mode B-phase input • In other than phase counting mode External clock B input
TRGIOA	P10	Input/Output	• In timer mode (output compare function) TRGGRA output-compare output • In timer mode (input capture function) TRGGRA input-capture input • In PWM mode PWM output
TRGIOB	P12	Input/Output	• In timer mode (output compare function) TRGGRB output-compare output • In timer mode (input capture function) TRGGRB input-capture input

8.3 Registers Controlling Timer RG

Table 8 - 2 lists the Registers Controlling Timer RG.

Table 8 - 2 Registers Controlling Timer RG

Register Name	Symbol
Peripheral enable register 1	PER1
Timer RG mode register	TRGMR
Timer RG count control register	TRGCNTC
Timer RG control register	TRGCR
Timer RG interrupt enable register	TRGIER
Timer RG status register	TRGSR
Timer RG I/O control register	TRGIOR
Timer RG counter	TRG
Timer RG general register A	TRGGRA
Timer RG general register B	TRGGRB
Timer RD general register C	TRGGRC
Timer RD general register D	TRGGRD
Port register 1	P1
Port mode register 1	PM1

8.3.1 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 using the timer RG, be sure to set bit 6 (TRGEN) 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 8 - 2 Format of Peripheral enable register 1 (PER1)

Address: F007AH After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	<3>	<2>	<1>	<0>
PER1	DACEN	TRGEN	AMPEN	0	DTCEN	PGAEN	AFEEN	TRJ0EN

TRGEN	Control of timer RG input clock supply
0	Stops input clock supply. • SFRs used by timer RG cannot be written. • Timer RG is in the reset status.
1	Enables input clock supply. • SFRs used by timer RG can be read and written.

Caution 1. When setting timer RG, be sure to set the TRGEN bit to 1 first. If TRGEN = 0, writing to a control register of timer RG is ignored, and all read values are default values (except for port mode register 1 (PM1) and port register 1 (P1).

Caution 2. Be sure to set bit 4 to 0.

8.3.2 Timer RG mode register (TRGMR)

Figure 8 - 3 Format of Timer RG mode register (TRGMR)

Address: F0250H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
TRGMR	TRGSTART	TRGELCICE	TRGDFCK1	TRGDFCK0	TRGDFB	TRGDFA	TRGMDF	TRGPWM

TRGSTART	TRG count start
0	Count stops, and PWM output signal (TRGIOA pin) is initialized (in PWM mode)
1	Count starts

TRGELCICE	Selection of ELC input capture request Notes 1, 2
0	External output signal B/digital filtering signal B is selected
1	Event input (input capture) from ELC is selected

TRGDFCK1	TRGDFCK0	Selection of digital filter function clock Note 1
0	0	$f_{CLK}/32$
0	1	$f_{CLK}/8$
1	0	f_{CLK}
1	1	Clock selected by bits TRGTCK0 to TRGTCK2 in TRGCR register

TRGDFB	Selection of digital filter function for TRGIOB pin
0	Digital filter function not used
1	Digital filter function used
When the digital filter is used, edge detection is performed after up to five cycles of the sampling clock.	

TRGDFA	Selection of digital filter function for TRGIOA pin
0	Digital filter function not used
1	Digital filter function used
When the digital filter is used, edge detection is performed after up to five cycles of the sampling clock.	

TRGMDF	Selection of phase counting mode
0	Increment
1	Phase counting mode
When the TRGMDF bit is set to 0, the counter counts the count source set by bits TRGTCK0 to TRGTCK2 in the TRGCR register. When the TRGMDF bit is set to 1, the counter counts the phase of input signals from the TRGCLKj pin (j = A or B) as listed in Table 8 - 15 Increment/Decrement Conditions for TRG Register	

TRGPWM	Selection of PWM mode
0	Timer mode
1	PWM mode

Note 1. Set this bit while the TRGSTART bit is 0 (count stops).

Note 2. To enable event input (input capture) from the ELC, set TRGIOB2 = 1 and TRGIOB1 and TRGIOB0 = 00B (rising edge) in the TRGIOR register.

8.3.3 Timer RG count control register (TRGCNTC)

The TRGCNTC register is used in phase counting mode. This register is used to set the count conditions for phase counting mode.

Figure 8 - 4 Format of Timer RG count control register (TRGCNTC)

Address: F0251H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TRGCNTC	CNTEN7	CNTEN6	CNTEN5	CNTEN4	CNTEN3	CNTEN2	CNTEN1	CNTEN0

CNTEN7	Counter enable 7
0	Disabled
1	Increment When TRGCLKA input is low level and at the rising edge of TRGCLKB input

CNTEN6	Counter enable 6
0	Disabled
1	Increment When TRGCLKB input is high level and at the rising edge of TRGCLKA input

CNTEN5	Counter enable 5
0	Disabled
1	Increment When TRGCLKA input is high level and at the falling edge of TRGCLKB input

CNTEN4	Counter enable 4
0	Disabled
1	Increment When TRGCLKB input is low level and at the falling edge of TRGCLKA input

CNTEN3	Counter enable 3
0	Disabled
1	Decrement When TRGCLKB input is high level and at the falling edge of TRGCLKA input

CNTEN2	Counter enable 2
0	Disabled
1	Decrement When TRGCLKA input is low level and at the falling edge of TRGCLKB input

CNTEN1	Counter enable 1
0	Disabled
1	Decrement When TRGCLKB input is low level and at the rising edge of TRGCLKA input

CNTEN0	Counter enable 0
0	Disabled
1	Decrement When TRGCLKA input is high level and at the rising edge of TRGCLKB input

8.3.4 Timer RG control register (TRGCR)

When writing to the TRGCR register, make sure the TRGSTART bit in the TRGMR register is 0 (count stops).

Figure 8 - 5 Format of Timer RG control register (TRGCR)

Address: F0252H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	<2>	<1>	<0>
TRGCR	0	TRGCCLR1	TRGCCLR0	TRGCKEG1	TRGCKEG0	TRGTCK2	TRGTCK1	TRGTCK0

TRGCCLR1	TRGCCLR0	Selection of TRG register clear source
0	0	Clear disabled
0	1	Clear by input capture or compare match with TRGGRA
1	0	Clear by input capture or compare match with TRGGRB
Other than above		Setting prohibited

TRGCKEG1	TRGCKEG0	Selection of external clock active edge Notes 1, 2
0	0	Count at the rising edge
0	1	Count at the falling edge
1	0	Count at both the rising/falling edges
Other than above		Setting prohibited

TRGTCK2	TRGTCK1	TRGTCK0	Selection of count source Note 1
0	0	0	f_{CLK}
0	0	1	$f_{CLK}/2$
0	1	0	$f_{CLK}/4$
0	1	1	$f_{CLK}/8$
1	0	0	$f_{CLK}/32$
1	0	1	TRGCLKA input
1	1	1	TRGCLKB input
Other than above			Setting prohibited

Note 1. In phase counting mode, the settings of bits TRGTCK0 to TRGTCK2 and bits TRGCKEG0 and TRGCKEG1 are disabled and the operation of phase counting mode has priority.

Note 2. Bits TRGCKEG0 and TRGCKEG1 are enabled when bits TRGTCK0 to TRGTCK2 are set to an external clock (TRGCLKA or TRGCLKB). When not set to an external clock, they are disabled.

8.3.5 Timer RG interrupt enable register (TRGIER)

Figure 8 - 6 Format of Timer RG interrupt enable register (TRGIER)

Address: F0253H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	<1>	<0>
TRGIER	0	0	0	0	TRGOVIE	TRGUDIE	TRGIMIEB	TRGIMIEA

TRGOVIE	Overflow interrupt enable
0	Interrupt by TRGOVF bit disabled
1	Interrupt by TRGOVF bit enabled

TRGUDIE	Underflow interrupt enable
0	Interrupt by TRGUDF bit disabled
1	Interrupt by TRGUDF bit enabled

TRGIMIEB	Input-capture/compare-match interrupt enable B
0	Interrupt by TRGIMFB bit disabled
1	Interrupt by TRGIMFB bit enabled

TRGIMIEA	Input-capture/compare-match interrupt enable A
0	Interrupt by TRGIMFA bit disabled
1	Interrupt by TRGIMFA bit enabled

Remark TRGIMFA, TRGIMFB, TRGUDF, TRGOVF: Bits in TRGSR register

8.3.6 Timer RG status register (TRGSR)

Figure 8 - 7 Format of Timer RG status register (TRGSR)

Address: F0254H After reset: 00H R/W

Symbol	7	6	5	<4>	<3>	<2>	<1>	<0>
TRGSR	0	0	0	TRGDIRF	TRGOVF	TRGUDF	TRGIMFB	TRGIMFA

TRGDIRF	Count direction flag
0	TRG register is decremented
1	TRG register is incremented

TRGOVF	Overflow flag [Note 1]
[Condition for setting to 0] Write 0 after reading [Note 2] [Condition for setting to 1] See **Table 8 - 3 Conditions for Setting Each Flag to 1**.	

TRGUDF	Underflow flag
[Condition for setting to 0] Write 0 after reading [Note 2] [Condition for setting to 1] See **Table 8 - 3 Conditions for Setting Each Flag to 1**.	

TRGIMFB	Input-capture/compare-match flag B
[Condition for setting to 0] Write 0 after reading [Notes 2, 3] [Condition for setting to 1] See **Table 8 - 3 Conditions for Setting Each Flag to 1**.	

TRGIMFA	Input-capture/compare-match flag A
[Condition for setting to 0] Write 0 after reading [Notes 2, 3] [Condition for setting to 1] See **Table 8 - 3 Conditions for Setting Each Flag to 1**.	

Note 1. When the counter value of timer RG changes from FFFFH to 0000H, the TRGOVF bit is set to 1. Also, if the counter value of timer RG changes from FFFFH to 0000H due to an input capture/compare match during operation according to the settings of bits TRGCCLR0 and TRGCCLR1 in the TRGCR register, the TRGOVF bit is set to 1.

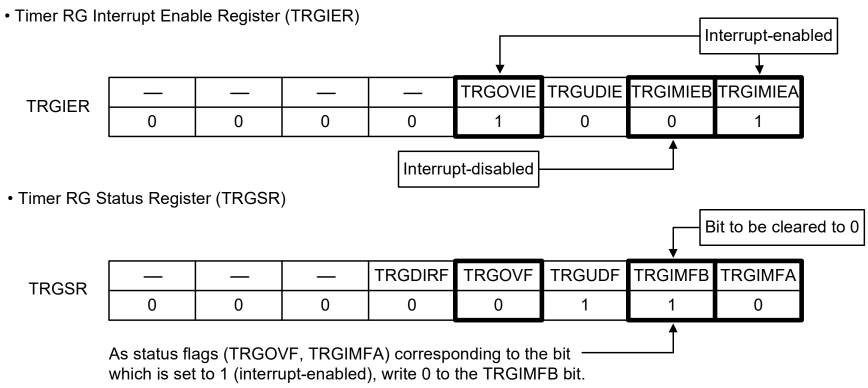
Note 2. The writing results are as follows:

- Writing 1 has no effect.
- If the read value is 0, the bit remains unchanged even if 0 is written to it.
  (Even if the bit is changed from 0 to 1.)
- If the read value is 1, writing 0 to the bit sets it to 0.
  When status flags of interrupt sources (applicable status flags) of the timer RG are set to 0 and their interrupts are disabled in the timer RG interrupt enable register (TRGIER), use either one of the following methods (a) to (c).

(a) Set 00H (all interrupts disabled) to timer RG interrupt enable register (TRGIER) and write 0 to applicable status flags.

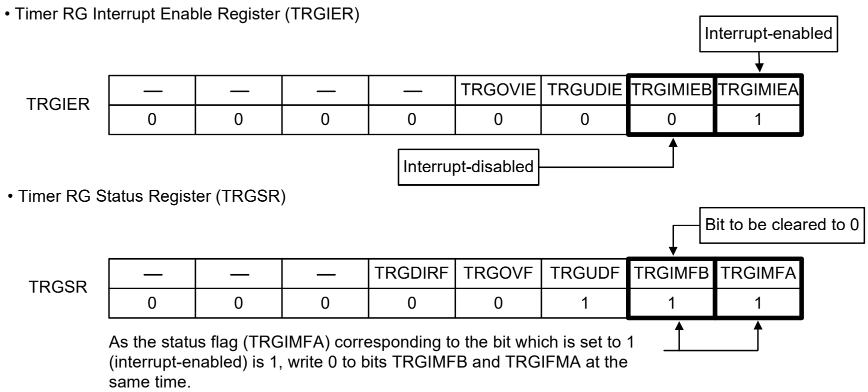
(b) When there are bits set to 1 (interrupt-enabled) in timer RG interrupt enable register (TRGIER) and status flags of interrupt sources related to their bits are 0, write 0 to applicable status flags.

Example: To clear the TRGIMFB bit to 0 when bits TRGIMIEA and TRGOVIE are set to 1 (interrupt-enabled) and the TRGIMIEB bit is set to 0 (interrupt-disabled)



(c) When there are bits set to 1 (interrupt-enabled) in timer RG interrupt enable register (TRGIER) and status flags of interrupt sources related to their bits are 1, write 0 to these status flags and applicable status flags at the same time.

Example: To clear the TRGIMFB bit to 0 when the TRGIMIEA bit is set to 1 (interrupt-enabled) and the TRGIMIEB bit is set to 0 (interrupt-disabled).



Note 3. When the DTC is used, bits TRGIMFA and TRGIMFB are set to 1 after DTC transfer is completed.

Table 8 - 3 Conditions for Setting Each Flag to 1

Bit Symbol	Timer Mode Note 1		PWM Mode
	Input Capture Function	Output Compare Function	
TRGOVF	When the TRG register overflows.		
TRGUDF	When the TRG register underflows (only in phase counting mode).		
TRGIMFB	Input edge of TRGIOB pin Note 2	When the values of registers TRG and TRGGRB match.	
TRGIMFA	Input edge of TRGIOA pin Note 2	When the values of registers TRG and TRGGRA match.	

Note 1. Phase counting mode is the counting method of the timer RG count register. The above timer modes and PWM mode can be used by making the corresponding settings.

Note 2. Edge selected by bits TRGIOj0 and TRGIOj1 (j = A or B) in the TRGIOR register.

8.3.7 Timer RG I/O control register (TRGIOR)

Figure 8 - 8 Format of Timer RG I/O control register (TRGIOR)

Address: F0255H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
TRGIOR	TRGBUFB	TRGIOB2	TRGIOB1	TRGIOB0	TRGBUFA	TRGIOA2	TRGIOA1	TRGIOA0

TRGBUFB	Selection of TRGGRD register function
0	Not used as buffer register for TRGGRB register
1	Used as buffer register for TRGGRB register

TRGIOB2	Selection of TRGGRB mode Notes 1, 2
0	Output compare function
1	Input capture function

TRGIOB1	TRGIOB0	TRGGRB control
0	0	Pin output by compare match is disabled
0	1	Low output
1	0	High output
1	1	Toggle output
In the output compare function, output of compare match between registers TRG and TRGGRB		

TRGIOB1	TRGIOB0	TRGGRB control
0	0	Rising edge of TRGIOB
0	1	Falling edge of TRGIOB
1	0	Both edges of TRGIOB
Other than above		Setting prohibited
In the input capture function, input capture of content of TRG register to TRGGRB register		

TRGBUFA	Selection of TRGGRC register function
0	Not used as buffer register for TRGGRA register
1	Used as buffer register for TRGGRA register

TRGIOA2	TRGGRA mode select Notes 1, 2
0	Output compare function
1	Input capture function

TRGIOA1	TRGIOA0	TRGGRA control
0	0	Pin output by compare match is disabled
0	1	Low output
1	0	High output
1	1	Toggle output
In the output compare function, output of compare match between registers TRG and TRGGRA		

TRGIOA1	TRGIOA0	TRGGRA control
0	0	Rising edge of TRGIOA
0	1	Falling edge of TRGIOA
1	0	Both edges of TRGIOA
Other than above		Setting prohibited
In the input capture function, input capture of content of TRG register to TRGGRA register		

Note 1. When the TRGIOj2 (j = A or B) bit is 1 (input capture function), the TRGGRj register functions as an input capture register.

Note 2. When the TRGIOj2 (j = A or B) bit is 0 (output compare function), the TRGGRj register functions as a compare match register. After a reset, the TRGIOj pin outputs as follows until bits TRGIOj0 and TRGIOj1 are set and the first compare match occurs.

TRGIOj1 and TRGIOj0 = 01B: High output
10B: Low output
11B: Low output

This TRGIOR register controls I/O pins in timer mode. It is disabled in PWM mode. It is disabled in PWM mode. Set the TRGIOR register while the count is stopped (TRGSTART in TRGMR register = 0).

8.3.8 Timer RG counter (TRG)

The TRG register is connected to the CPU via the internal 16-bit bus and should be always accessed in 16-bit units. This register operates incrementing and can also operate free-running, period counting, or external event counting. It can be cleared to 0000H by the compare match with the corresponding TRGGRA or TRGGRB register, or the input capture to registers TRGGRA and TRGGRB (count clear function).

When the TRG register overflows (FFFFH → 0000H), the TRGOVF flag in the TRGSR register is set to 1.

When the TRG register underflows (0000H → FFFFH), the TRGUDF flag in the TRGSR register is set to 1.

Figure 8 - 9 Format of Timer RG counter (TRG)

Address: F0256H After reset: 0000H RW

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

—	Function	Setting Range
Bits 15 to 0	In phase counting mode, count operation is increment/decrement. In other modes, count operation is increment.	0000H to FFFFH

8.3.9 Timer RG general registers A, B, C, and D (TRGGRA, TRGGRB, TRGGRC, TRGGRD)

Registers TRGGRA and TRGGRB are 16-bit readable/writable registers with both the output compare and input capture register functions. These functions can be switched by setting the TRGIOR register.
When registers TRGGRA and TRGGRB are used as output compare registers, the values of registers TRGGRA and TRGGRB and the value of the TRG register are always compared. When their values match (compare match), bits TRGIMFA and TRGIMFB in the TRGSR register are set to 1. Compare match output can be set by the TRGIOR register.
When registers TRGGRA and TRGGRB are used as input capture registers, the value of the TRG register is stored upon detecting externally input capture signals. At this time, the TRGIMFA/TRGIMFB bit is set to 1. The detection edge of input capture signals is selected by setting the TRGIOR register.
The TRGGRC register can also be used as the buffer register for the TRGGRA register and the TRGGRD register can be used as the buffer register for the TRGGRB register, respectively. These functions can be selected by setting bits TRGBUFA and TRGBUFB in the TRGIOR register.
For example, when the TRGGRA register is set as an output compare register and the TRGGRC register is set as the buffer register for the TRGGRA register, the value of the TRGGRC register is transferred to the TRGGRA register each time compare match A occurs.
When the TRGGRA register is set as an input capture register and the TRGGRC register is set as the buffer register for the TRGGRA register, the value of the TRG register is transferred to the TRGGRA register and the value of the TRGGRA register value is transferred to the TRGGRC register each time an input capture occurs.
Registers TRGGRA, TRGGRB, TRGGRC, and TRGGRD can be read or written in 16-bit units.

Figure 8 - 10 Format of Timer RG general registers A, B, C, and D (TRGGRA, TRGGRB, TRGGRC, TRGGRD)

Address: F0258H (TRGGRA), F025AH (TRGGRB), FFF60H (TRGGRC), FFF62H (TRGGRD) After Reset: FFFFH RW

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

—	Function
Bits 15 to 0	Function varies depending on the mode or the function. Table 8 - 4 lists the TRGGRA, TRGGRB, TRGGRC, and TRGGRD Register Functions.

Remark i = A, B, C, D

Table 8 - 4 TRGGRA, TRGGRB, TRGGRC, and TRGGRD Register Functions

Mode, Function	Register	Setting	Function
Input capture	TRGGRA	TRGIOR (TRGIOA2 = 1) TRGMR (TRGPWM = 0)	Input capture register (stores value of TRG register)
	TRGGRB	TRGIOR (TRGIOB2 = 1) TRGMR (TRGPWM = 0)	Input capture register (stores value of TRG register)
Output compare	TRGGRA	TRGIOR (TRGIOA2 = 0) TRGMR (TRGPWM = 0)	Output compare register (stores compare value with TRG register and outputs set value to TRGIOA at compare match)
	TRGGRB	TRGIOR (TRGIOB2 = 0) TRGMR (TRGPWM = 0)	Output compare register (stores compare value with TRG register and outputs set value to TRGIOB at compare match)
PWM	TRGGRA	TRGMR (TRGPWM = 1)	Output compare register (outputs high level to TRGIOA at compare match)
	TRGGRB		Output compare register (outputs low level to TRGIOA at compare match)
Common	TRGGRC	TRGIOR (TRGBUFA = 0)	Not used
	TRGGRD	TRGIOR (TRGBUFB = 0)	Not used
	TRGGRC	TRGIOR (TRGBUFA = 1)	Buffer register for TRGGRA (transfers from/to TRGGRA) • When TRGIOA2 = 1 Input capture signal: Receive previous input capture value from TRGGRA • When TRGIOA2 = 0 TRG and TRGGRA compare match: Send next expected compare value to TRGGRA
	TRGGRD	TRGIOR (TRGBUFB = 1)	Buffer register for TRGGRB (transfers from/to TRGGRB) • When TRGIOB2 = 1 Input capture signal: Receive previous input capture value from TRGGRB • When TRGIOB2 = 0 TRG and TRGGRB compare match: Send next expected compare value to TRGGRB

Caution **When the setting of bits TRGTCK2 to TRGTCK0 in the TRGCR register is 000B (f_{CLK}) and the compare value is set to 0000H, a request signal to the DTC and the ELC is generated only once immediately after the count starts. When the compare value is 0001H or higher, a request signal is generated each time a compare match occurs.**

8.3.10 Port mode register 1 (PM1)

This register sets input/output mode of port 1 in 1-bit units.
When using a port (such as P10/TRGIOA and P12/TRGIOB) as a timer output pin, set the bit corresponding to the port in port mode register 1 (PM1) and port register 1 (P1) to 0.

Example When using P10/TRGIOA for timer output
Set the PM10 bit of port mode register 1 to 0.
Set the P10 bit of port register 1 to 0.

When using a port (such as P10/TRGIOA and P12/TRGIOB) as a timer input pin, set the bit corresponding to the port in port mode register 1 (PM1) to 1. At this time, the corresponding bit in port register 1 (P1) may be 0 or 1.

Example When using P10/TRGIOA for timer input
Set the PM10 bit of port mode register 1 to 1.
Set the P10 bit of port register 1 to 0 or 1.

The PM1 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation sets this register to FFH.

Figure 8 - 11 Format of Port mode register 1 (PM1)

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

PM1n	Selection of P1n pin I/O mode (n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

8.4 Timer RG Operation

8.4.1 Items common to multiple modes and functions

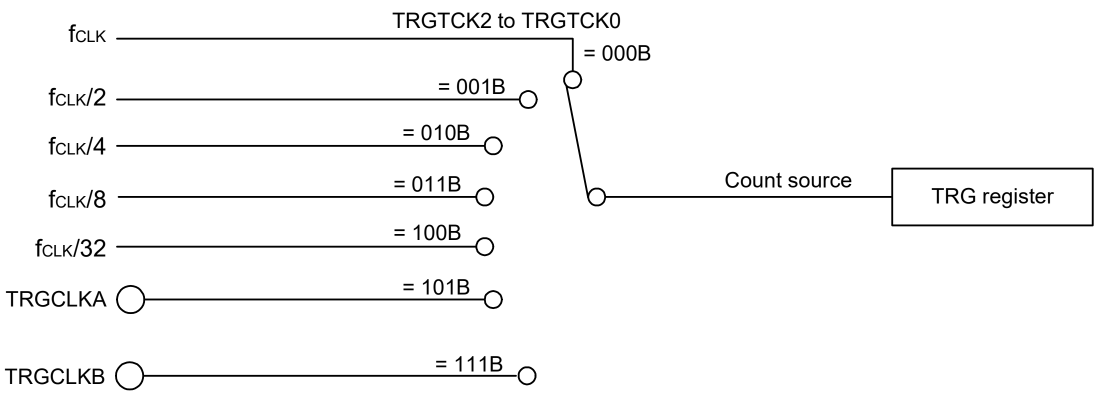
(1) Count sources

Table 8 - 5 lists the Count Source Selection and Figure 8 - 12 shows the Count Source Block Diagram.
When phase counting mode is selected, the settings of bits TRGTCK0 to TRGTCK2 and bits TRGCKEG0 and TRGCKEG1 in the TRGCR register are disabled.

Table 8 - 5 Count Source Selection

Count Source	Selection Method
f_{CLK}, $f_{CLK}/2$, $f_{CLK}/4$, $f_{CLK}/8$, $f_{CLK}/32$	The count source is selected by bits TRGTCK0 to TRGTCK2 in the TRGCR register.
External signal input to TRGCLKA or TRGCLKB pin	Bits TRGTCK2 to TRGTCK0 in the TRGCR register are set to 101B (TRGCLKA input) or 111B (TRGCLKB input). The active edge is selected by bits TRGCKEG0 and TRGCKEG1 in the TRGCR register. The corresponding bit of the port mode register is set to 1 (input mode).

Figure 8 - 12 Count Source Block Diagram



Remark: TRGTCK0 to TRGTCK2: Bits in TRGCR register

The pulse width of an external clock input to the TRGCLKj pin (j = A or B) should be set to three cycles or more of the timer RG operating clock (f_{CLK}).

(2) Buffer operation

The TRGBUFA or TRGBUFB bit in the TRGIOR register can be used to select the TRGGRC or TRGGRD register as the buffer register for the TRGGRA or TRGGRB register.

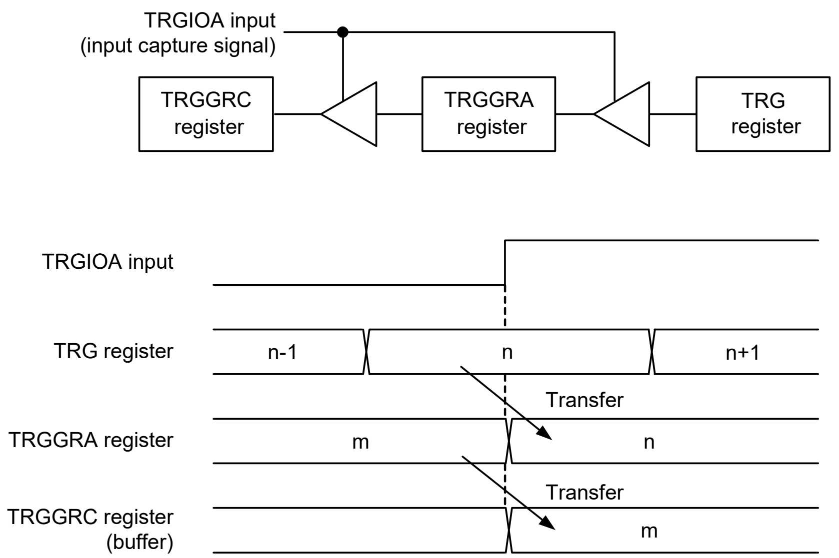
- Buffer register for TRGGRA register: TRGGRC register
- Buffer register for TRGGRB register: TRGGRD register

Buffer operation differs depending on the mode. Table 8 - 6 lists the Buffer Operation in Each Mode, Figure 8 - 13 shows the Buffer Operation for Input Capture Function and Figure 8 - 14 shows the Buffer Operation for Output Compare Function.

Table 8 - 6 Buffer Operation in Each Mode

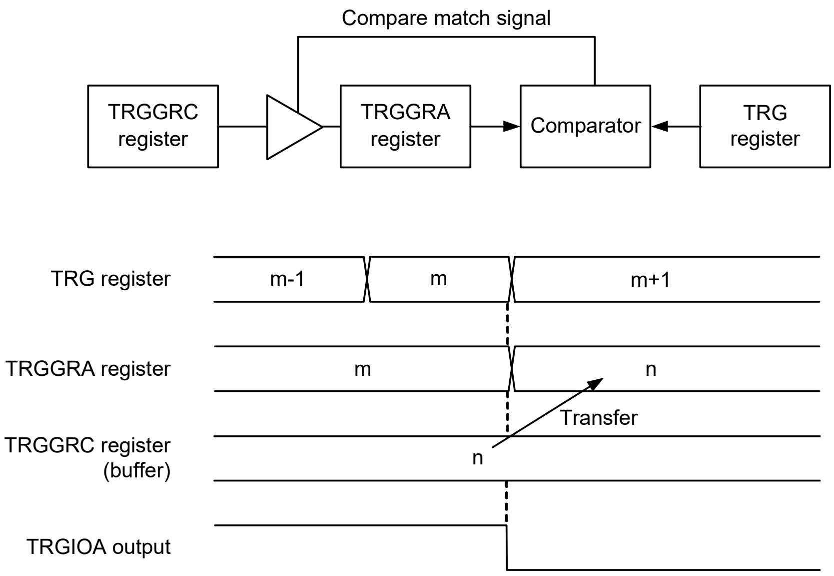
Function, Mode	Transfer Timing	Transfer Destination Register
Input capture function	Input capture signal input	The content of the TRGGRA (TRGGRB) register is transferred to the buffer register.
Output compare function	Compare match between the TRG register and the TRGGRA (TRGGRB) register	The content of the buffer register is transferred to the TRGGRA (TRGGRB) register.
PWM mode		

Figure 8 - 13 Buffer Operation for Input Capture Function



The above diagram applies under the following conditions:
• The TRGBUFA bit in the TRGIOR register is set to 1 (TRGGRC register is used as buffer register for TRGGRA register).
• Bits TRGIOA2 to TRGIOA0 in the TRGIOR register are set to 100B (input capture at the rising edge).

Figure 8 - 14 Buffer Operation for Output Compare Function



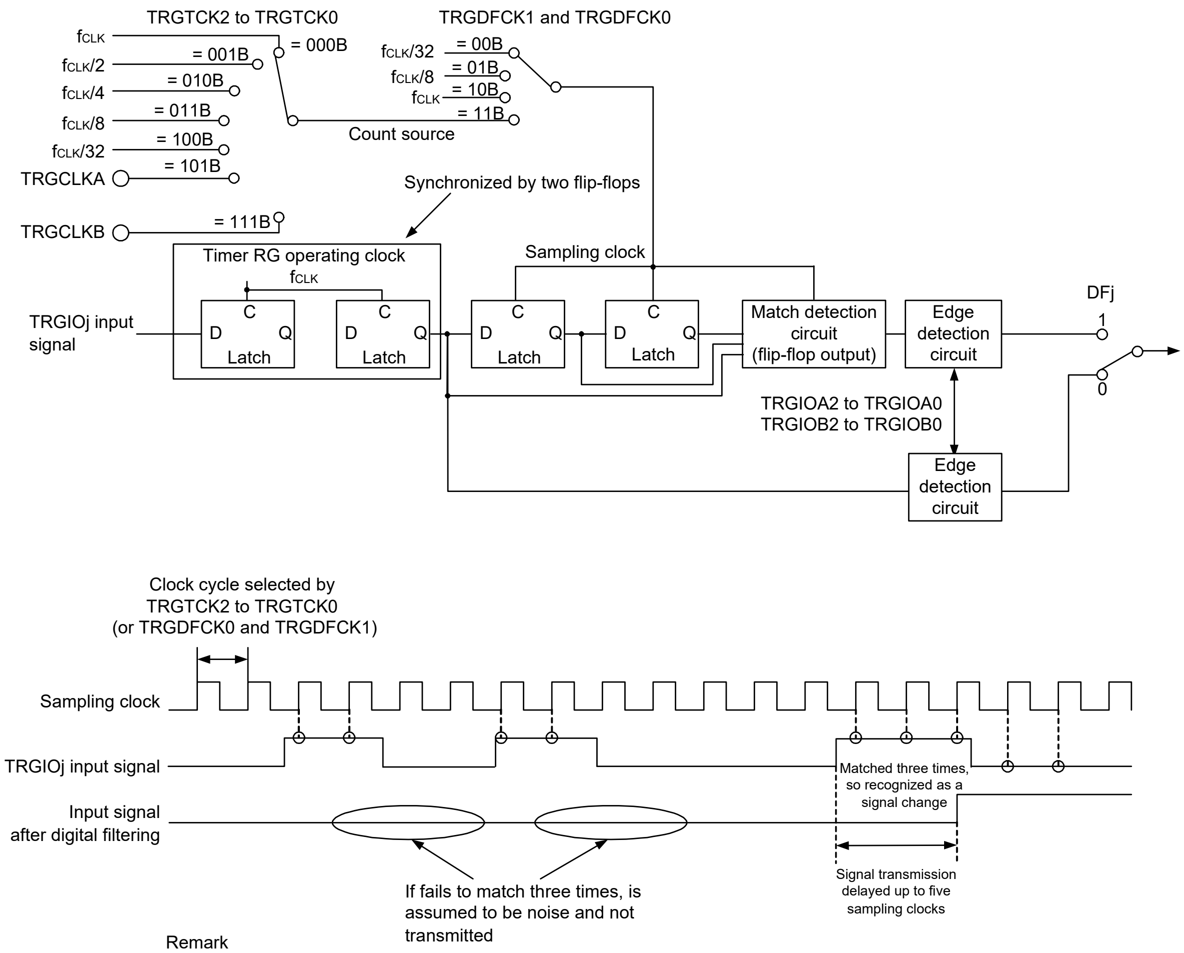
The above diagram applies under the following conditions:
• The TRGBUFA bit in the TRGIOR register is set to 1 (TRGGRC register is used as buffer register for TRGGRA register).
• Bits TRGIOA2 to TRGIOA0 in the TRGIOR register are set to 001B (low output by compare match).

(3) Digital filter

The TRGIOj input (j = A or B) is sampled, and when the sampled input level matches three times, its level is determined. Select the digital filter function and sampling clock by using the TRGMR register.

Figure 8 - 15 shows a Block Diagram of Digital Filter.

Figure 8 - 15 Block Diagram of Digital Filter



Remark
j = A or B

TRGTCK0 to TRGTCK2: Bits in TRGCR register
TRGDFCK0, TRGDFCK1, TRGDFj: Bits in TRGMR register
TRGIOA0 to TRGIOA2, TRGIOB0 to TRGIOB2: Bits in TRGIOR register

(4) Event input from event link controller (ELC)

Timer RG performs input capture operation B by event input from the ELC. The TRGIMFB bit in the TRGSR register is set to 1 at this time.

To use this function, select the input capture function of timer mode/phase counting mode, and set the TRGELCICE bit in the TRGMR register to 1. This function is disabled in other modes (the output compare function of timer mode/phase counting mode and PWM mode).

Setting procedure

<1> Set timer RG as the ELC event link destination.

<2> Set the TRGELCICE bit in the TRGMR register to 1.

(5) Event output to event link controller (ELC)

Table 8 - 7 lists the ELC Event Output according to TRGIMFA Bit. Table 8 - 8 lists the ELC Event Output according to TRGIMFB Bit.

Table 8 - 7 ELC Event Output according to TRGIMFA Bit

Mode, Function	ELC Source
Input capture function (TRGPWM = 0, TRGIOA2 = 1)	Detection of TRGIOA edge set by bits TRGIOA0 and TRGIOA1
Output compare function (TRGPWM = 0, TRGIOA2 = 0)	Compare match between registers TRG and TRGGRA
PWM mode (TRGPWM = 1)	Compare match between registers TRG and TRGGRA

Remark TRGPWM: Bit in TRGMR register
TRGIOA0, TRGIOA1, TRGIOA2: Bits in TRGIOR register

Table 8 - 8 ELC Event Output according to TRGIMFB Bit

Mode, Function	ELC Source
Input capture function (TRGPWM = 0, TRGIOB2 = 1)	Detection of TRGIOB edge set by bits TRGIOB0 and TRGIOB1
Output compare function (TRGPWM = 0, TRGIOB2 = 0)	Compare match between registers TRG and TRGGRB
PWM mode (TRGPWM = 1)	Compare match between registers TRG and TRGGRB

Remark TRGPWM: Bit in TRGMR register
TRGIOB0, TRGIOB1, TRGIOB2: Bits in TRGIOR register

8.4.2 Timer mode (input capture function)

The value of the TRG register can be transferred to registers TRGGRA and TRGGRB upon detecting the input edge of the input capture/output compare pins (TRGIOA and TRGIOB). The detection edge can be selected from the rising edge/falling edge/both edges.

The input capture function can be used for measuring pulse widths and periods.

Table 8 - 9 lists the Input Capture Function Specifications.

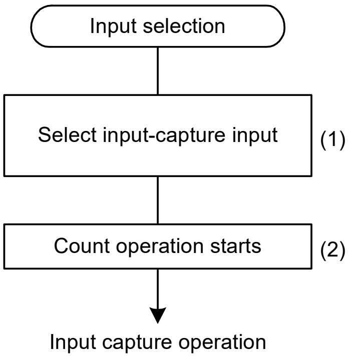
Table 8 - 9 Input Capture Function Specifications

Item	Specification
Count sources	f_{CLK}, $f_{CLK}/2$, $f_{CLK}/4$, $f_{CLK}/8$, $f_{CLK}/32$ External signal input to the TRGCLKA or TRGCLKB pin (active edge selectable by a program)
Count operation	Increment
Count period	When bits TRGCCLR1 to TRGCCLR0 in the TRGCR register are set to 00B (free-running operation) $1/fk \times 65,536$ fk: Frequency of count source
Count start condition	1 (count starts) is written to the TRGSTART bit in the TRGMR register.
Count stop condition	0 (count stops) is written to the TRGSTART bit in the TRGMR register.
Interrupt request generation timing	• Input capture (active edge of TRGIOA and TRGIOB pin input) • TRG register overflow
TRGIOA, TRGIOB pin function	I/O port or input-capture input (selectable for each pin)
TRGCLKA, TRGCLKB pin function	I/O port or external clock input
Read from timer	The count value can be read by reading the TRG register.
Write to timer	The TRG register can be written to.
Selectable functions	• Input-capture input pin selection Either one or both of pins TRGIOA and TRGIOB • Active edge selection for input-capture input Rising edge, falling edge, or both rising and falling edges • Timing for setting the TRG register to 0000H At overflow or input capture • Buffer operation (see **8.4.1 (2) Buffer operation**) • Digital filter (see **8.4.1 (3) Digital filter**) • Input capture operation by event input signal (input capture) from ELC

(1) Procedure example for setting input capture operation

Figure 8 - 16 shows a Procedure Example for Setting Input Capture Operation.

Figure 8 - 16 Procedure Example for Setting Input Capture Operation



(1) Use the TRGIOR register to set TRGGRj (j = A or B) as an input capture register and select the input edge of input capture signals from the following three: the rising edge/falling edge/both edges.

(2) Set the TRGSTART bit in the TRGMR register to 1 to start the count operation of the TRG register.

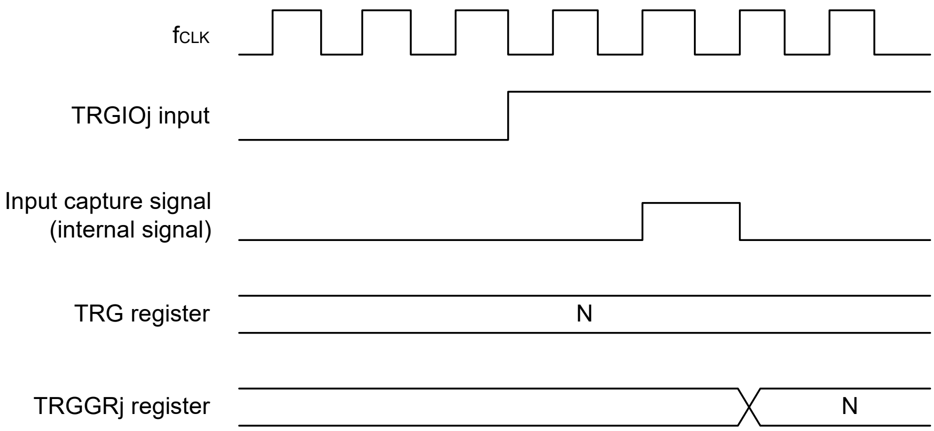
(2) Input capture signal timing

For input-capture input, the rising edge/falling edge/both edges can be selected by setting the TRGIOR register.

Figure 8 - 17 shows the Input-Capture Input Signal Timing.

The pulse width of input-capture input signals should be 1.5 f_{CLK} or more for a single edge and 2.5 f_{CLK} or more for both edges.

Figure 8 - 17 Input-Capture Input Signal Timing



j = A or B

(3) Operation example

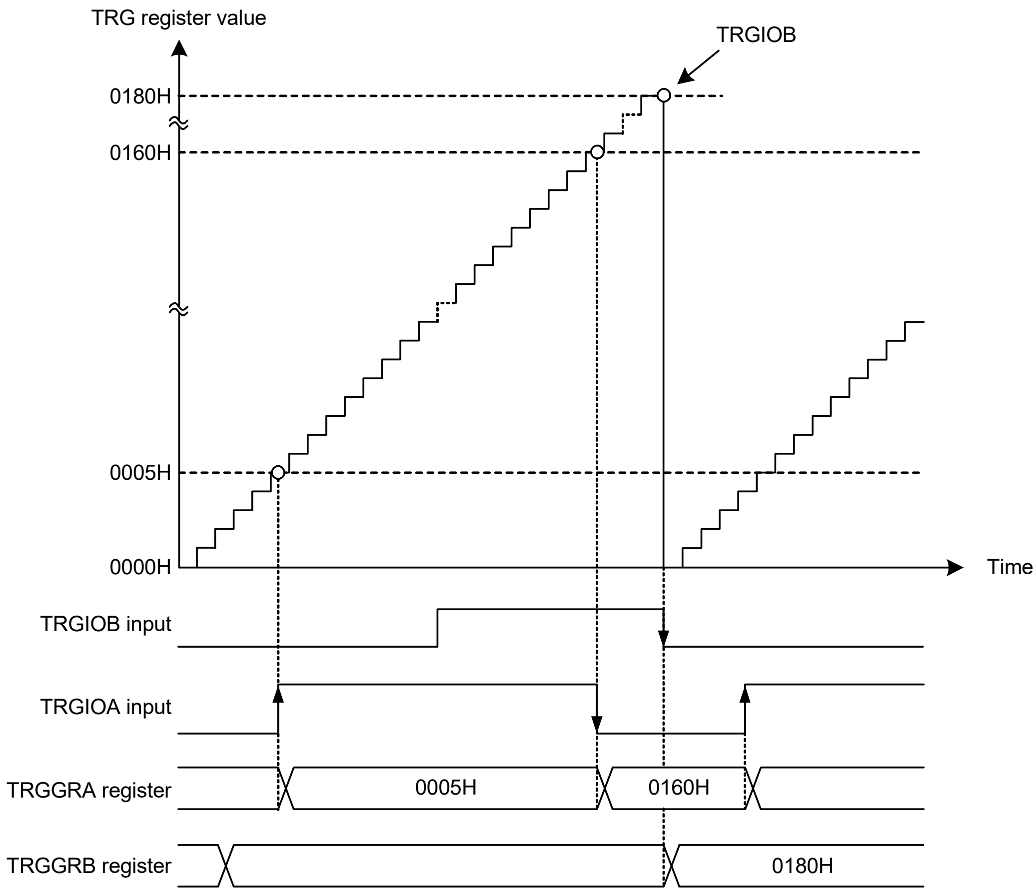
Figure 8 - 18 shows an Operation Example of Input Capture.

This example applies when both the rising/falling edges are selected as the input-capture input edge of the TRGIOA pin and the falling edge is selected as the input-capture input edge of the TRGIOB pin, and the TRG register is set to be cleared by the input capture to the TRGGRB register.

(a) Use the TRGIOR register to set registers TRGGRA and TRGGRB as input capture registers and select the input edge of input capture signals from the following three: the rising edge/falling edge/both edges.

(b) Set the TRGSTART bit in TRGMR to 1 and start the count operation of the TRG register.

Figure 8 - 18 Operation Example of Input Capture



By setting bits TRGCCLR0 and TRGCCLR1 in the TRGCR register, the count can be cleared by input capture A or B.

Figure 8 - 18 shows an operation example with bits TRGCCLR1 and TRGCCLR0 set to 10B. If the input capture operation has been set to clear the count during operation and is performed when the timer count value is FFFFH, depending on the timing between the count source and input capture operation interrupt flags bits TRGIMFA, TRGIMFB, and TRGOVF may be set to 1 simultaneously.

8.4.3 Timer mode (output compare function)

This mode (output compare function) detects when the contents of the TRG register and the TRGGRA or TRGGRB register match (compare match). When a match occurs, a signal is output from the TRGIOA or TRGIOB pin at a given level.

Table 8 - 10 lists the Output Compare Function Specifications.

Table 8 - 10 Output Compare Function Specifications

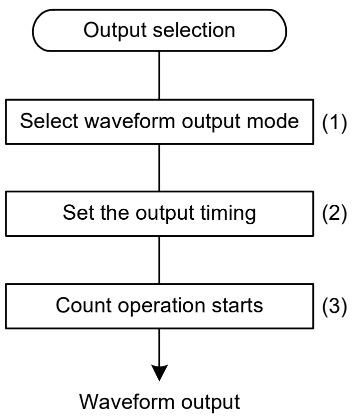
Item	Specification
Count sources	f_{CLK}, $f_{CLK}/2$, $f_{CLK}/4$, $f_{CLK}/8$, $f_{CLK}/32$ External signal input to the TRGCLKj pin (active edge selectable by a program)
Count operation	Increment
Count periods	• When bits TRGCCLR1 and TRGCCLR0 in the TRGCR register are set to 00B (free-running operation) $1/fk \times 65,536$ fk: Frequency of count source • When bits TRGCCLR1 and TRGCCLR0 in the TRGCR register are set to 01B or 10B (TRG is set to 0000H by compare match with TRGGRj) $1/fk \times (n + 1)$ n: Value set in the TRGGRj register
Waveform output timing	Compare match (contents of registers TRG and TRGGRj match)
Count start condition	1 (count starts) is written to the TRGSTART bit in the TRGMR register.
Count stop condition	0 (count stops) is written to the TRGSTART bit in the TRGMR register.
Interrupt request generation timing	• Compare match (contents of registers TRG and TRGGRj match) • TRG register overflow
TRGIOA, TRGIOB pin function	I/O port or output-compare output (selectable for each pin)
TRGCLKA, TRGCLKB pin function	I/O port or external clock input
Read from timer	The count value can be read by reading the TRG register.
Write to timer	The TRG register can be written to.
Selectable functions	• Output-compare output pin selection Either one or both of pins TRGIOA and TRGIOB • Output level selection at compare match Low output, high output, or inverted output level • Timing for setting the TRG register to 0000H Overflow or compare match with the TRGGRj register • Buffer operation (see **8.4.1 (2) Buffer operation**)

Remark j = A or B

(1) Procedure example for setting waveform output by compare match

Figure 8 - 19 shows a Procedure Example for Setting Waveform Output by Compare Match.

Figure 8 - 19 Procedure Example for Setting Waveform Output by Compare Match



(1) Use the TRGIOR register to select the compare match output from the following three: Low output/high output/toggle output. When waveform output mode is set, the ports function as compare match output pins (TRGIOA and TRGIOB). The output levels of these pins depend on the settings of bits TRGIOA0 and TRGIOA1 and bits TRGIOB0 and TRGIOB1 in the TRGIOR register until the first compare match occurs.

(2) Set the timing for generating a compare match into registers TRGGRA and TRGGRB.

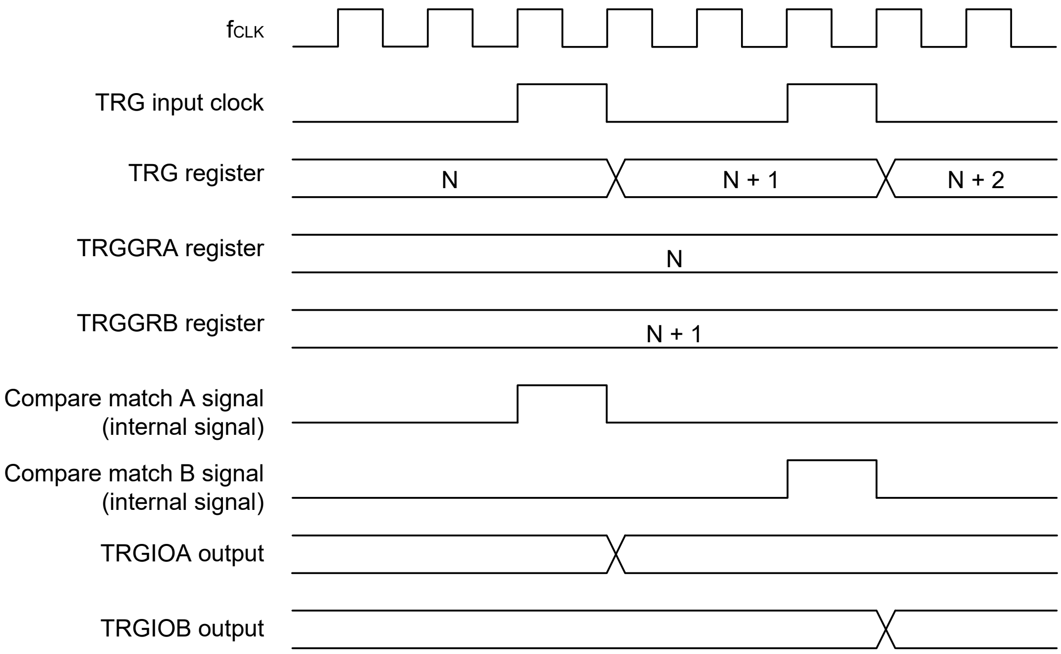
(3) Set the TRGSTART bit in the TRGMR register to 1 to start the count operation of the TRG register.

(2) Output-compare output timing

A compare match signal is generated at the last state when the TRG register and the TRGGRA or TRGGRB register match (at the timing for updating the count value that the TRG register matches). When the compare match signal is generated, the output value set by the TRGIOR register is output to the output-compare output pin (TRGIOA or TRGIOB). After the TRG register and the TRGGRA or TRGGRB register match, no compare match signal is generated until the TRG input clock is generated.

Figure 8 - 20 shows the Output-Compare Output Timing.

Figure 8 - 20 Output-Compare Output Timing



(3) Operation example

Figure 8 - 21 shows an Operation Example of Low Output and High Output.

This example applies when the TRG register is set for free-running operation, and low output is set at compare match A, and high output is set at compare match B. When the set level and the pin level match, the pin level does not change.

Figure 8 - 21 Operation Example of Low Output and High Output

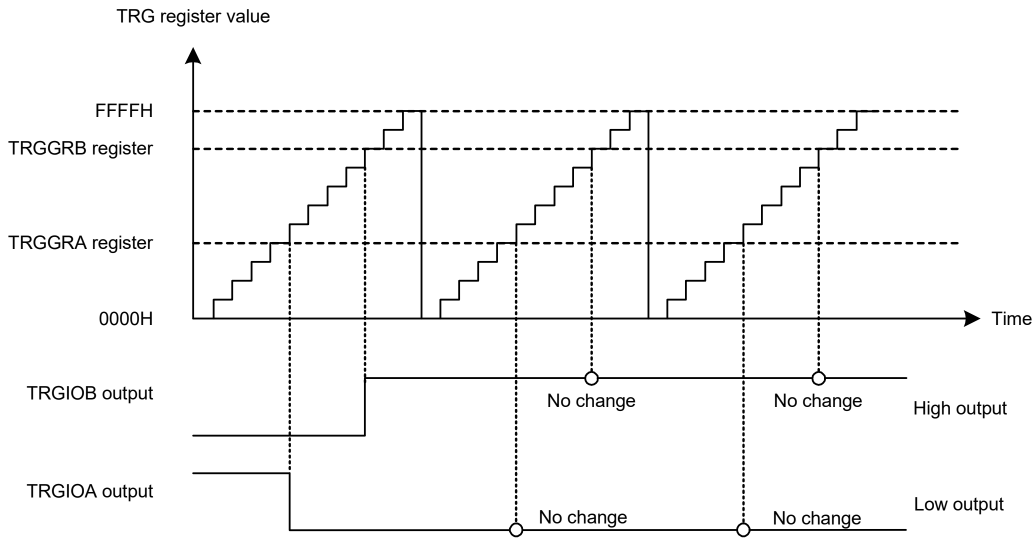


Figure 8 - 22 shows the Operation Example of Toggle Output. This example applies when the TRG register is set for period counting operation (counter clear at compare match B), and toggle output is set at both compare match A and B.

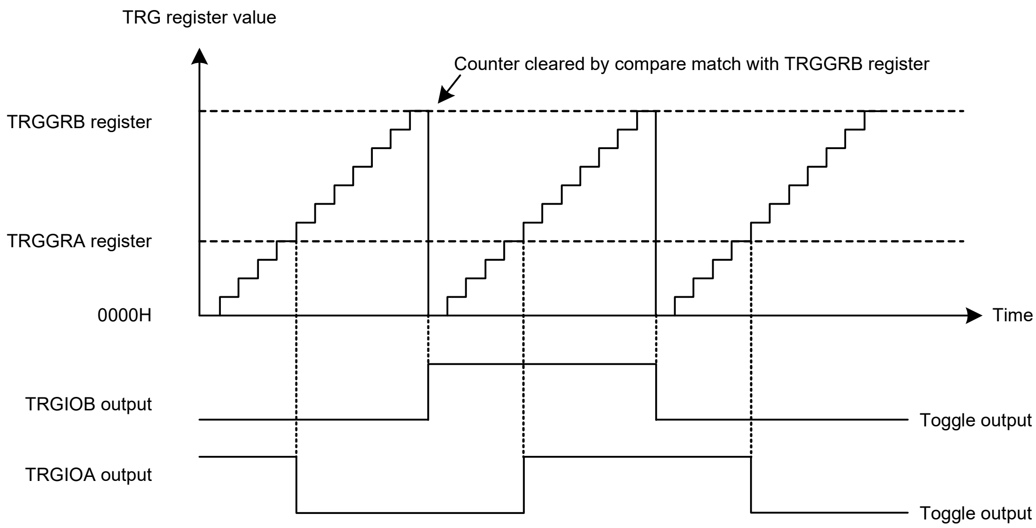
(a) Use the TRGIOR register to select the compare match output from the following three: Low output/high output/toggle output. When waveform output mode is set, the ports function as compare match output pins (TRGIOA and TRGIOB).

(b) Set the timing for generating a compare match into registers TRGGRA and TRGGRB.

(c) Set the TRGSTART bit in the TRGMR register to 1 to start the count operation of the TRG register.

The compare match output pins (TRGIOA and TRGIOB) are not initialized by setting the TRGSTART bit to 0 during operation. To return to initial values, write to the TRGIOR register to initialize the output. (The output is only initialized when bits TRGIOA0, TRGIOA1, TRGIOB0, and TRGIOB1 in the TRGIOR register are set to low output or high output.) By setting bits TRGCCLR0 and TRGCCLR1 in the TRGCR register, the timer RG counter value is reset by an input capture/compare match (match with the TRGGRA or TRGGRB register). If the expected compare value is FFFFH at this time, FFFFH changes to 0000H, same as the overflow operation, and the TRGOVF bit is set to 1.
This operation is the same for modes where the output compare function is used on the timer RG counter value and expected compare value.

Figure 8 - 22 Operation Example of Toggle Output



8.4.4 PWM mode

In PWM mode, registers TRGGRA and TRGGRB are used as a pair and a PWM waveform is output from the TRGIOA output pin. The output setting by the TRGIOR register is invalid for the pins set to PWM mode. Set the high output timing for a PWM waveform into the TRGGRA register and the low output timing for a PWM waveform into the TRGGRB register.

By setting the compare match with either the TRGGRA or TRGGRB register as the counter clear source for the TRG register, a PWM waveform with duty cycle 0% to 100% can be output from the TRGIOA pin.

Table 8 - 11 lists the PWM Mode Specifications and Table 8 - 12 lists the Combination of PWM Output Pins and Registers. When the setting values in registers TRGGRA and TRGGRB are the same, the output value does not change even if a compare match occurs.

Table 8 - 11 PWM Mode Specifications

Item	Specification
Count sources	f_{CLK}, $f_{CLK}/2$, $f_{CLK}/4$, $f_{CLK}/8$, $f_{CLK}/32$ External signal input to the TRGCLKj pin (active edge selectable by a program)
Count operation	Increment
PWM waveform	• The high output timing of a PWM waveform is set into the TRGGRA register. • The low output timing of a PWM waveform is set into the TRGGRB register.
Count start condition	1 (count starts) is written to the TRGSTART bit in the TRGMR register.
Count stop condition	0 (count stops) is written to the TRGSTART bit in the TRGMR register.
Interrupt request generation timing	• Compare match (contents of registers TRG and TRGGRj match) • TRG register overflow
TRGIOA pin function	PWM output
TRGIOB pin function	I/O port
TRGCLKA, TRGCLKB pin function	I/O port or external clock input
Read from timer	The count value can be read by reading the TRG register.
Write to timer	The TRG register can be written to.
Selectable functions	• Timing for setting the TRG register to 0000H Overflow or compare match with the TRGGRj register • Buffer operation (see **8.4.1 (2) Buffer operation**)

Remark j = A or B

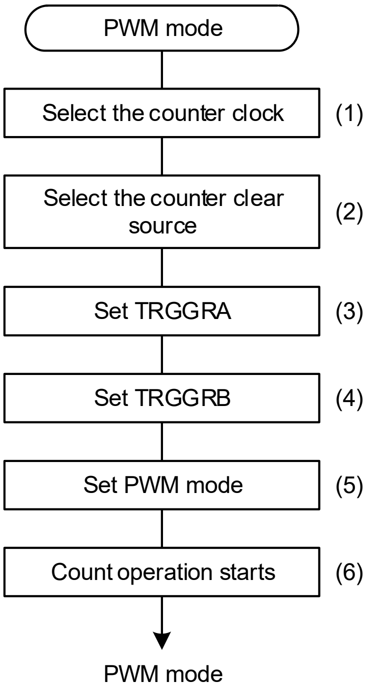
Table 8 - 12 Combination of PWM Output Pins and Registers

Output Pin	High Output	Low Output
TRGIOA	TRGGRA	TRGGRB
TRGIOB	I/O port function	

(1) Procedure example for setting PWM mode

Figure 8 - 23 shows a Procedure Example for Setting PWM Mode.

Figure 8 - 23 Procedure Example for Setting PWM Mode



(1) Use bits TRGTCK0 to TRGTCK2 in the TRGCR register to select the count source. When an external clock is selected, use bits TRGCKEG0 and TRGCKEG1 in the TRGCR register to select the edge of the clock.

(2) Use bits TRGCCLR0 and TRGCCLR1 in the TRGCR register to select the counter clear source.

(3) Set the high output timing for a PWM output waveform into the TRGGRA register.

(4) Set the low output timing for a PWM output waveform into the TRGGRB register.

(5) Use the TRGPWM bit in the TRGMR register to set PWM mode. When PWM mode is set, registers TRGGRA and TRGGRB are set as the output compare registers for setting the high output/low output timing for a PWM output waveform, regardless of the content of the TRGIOR register.
When the PM10 bit in the PM1 register is 0 and the P10 bit in the P1 register is 0, the TRGIOA pin automatically functions as a PWM output pin. However, the TRGIOB pin functions as an I/O port.

(6) Set the TRGSTART bit in the TRGMR register to 1 to start the count operation of the TRG register.

(2) Operation example

Figure 8 - 24 shows Example of Operation in PWM Mode (1).

When the PM10 bit in the PM1 register is 0 and the P10 bit in the P1 register is 0, the TRGIOA pin automatically functions as an output pin, and high output is set at the compare match with the TRGGRA register and low output is set at the compare match with the TRGGRB register. However, regardless of the setting of the TRGIOR register, the TRGIOB pin functions as an I/O port.

This example applies when the compare match with the TRGGRA or TRGGRB register is set as the counter clear source for the TRG register. The initial state of the TRGIOA pin depends only on the counter clear sources. This correspondence is shown in Table 8 - 13.

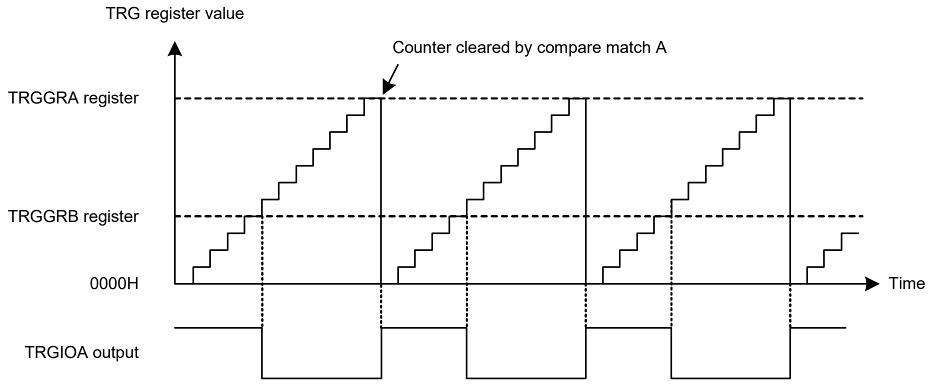
This initialization is performed when the TRGSTART bit in the TRGMR register is 0 (count stops).

Table 8 - 13 Correspondence between Initial State of TRGIOA Pin and Counter Clear Sources

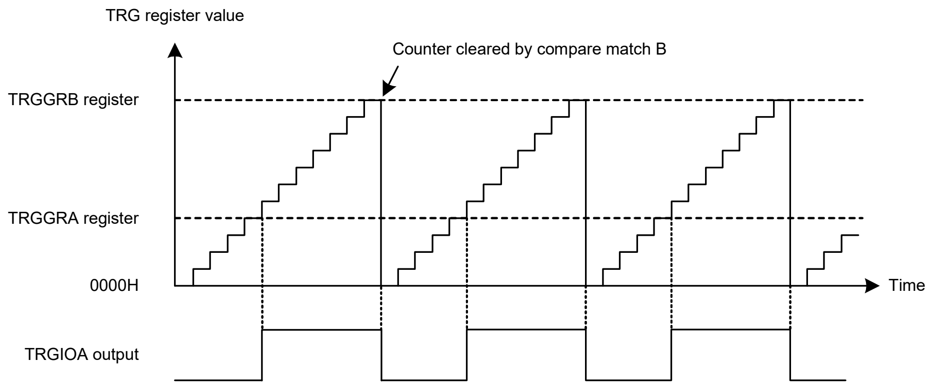
Counter Clear Source	Initial State of TRGIOA Pin
Compare match with TRGGRA register	High
Compare match with TRGGRB register	Low

When bits TRGCCLR1 and TRGCCLR0 in the TRGCR register are set to 00B (clear disabled), the initial state of the TRGIOA pin becomes high.

Figure 8 - 24 Example of Operation in PWM Mode (1)



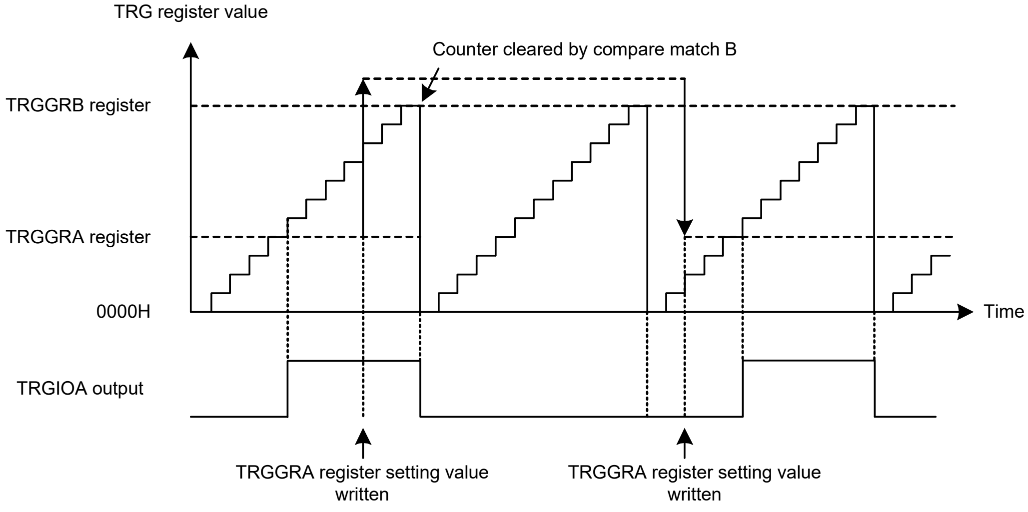
(a) Counter clear by the compare match with the TRGGRA register



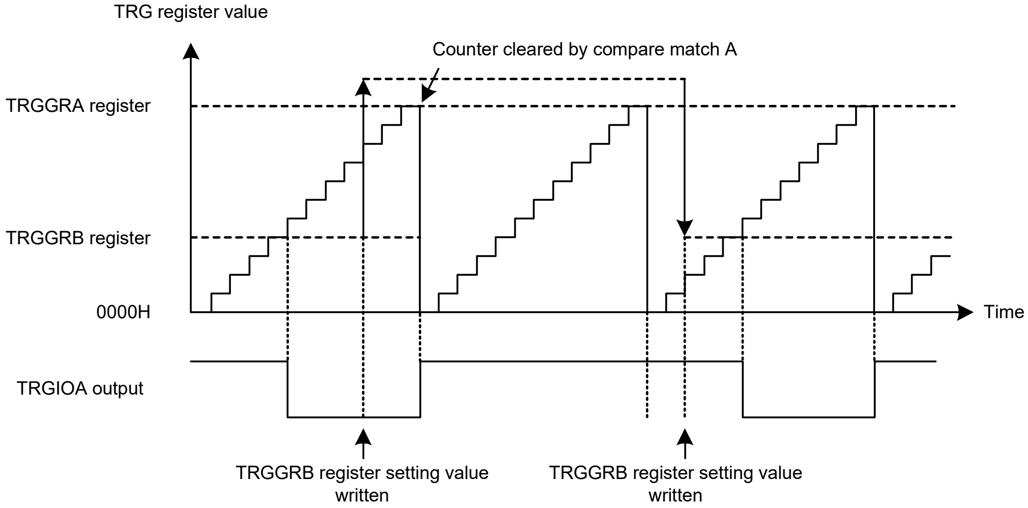
(b) Counter clear by the compare match with the TRGGRB register

Figure 8 - 25 shows an example for outputting a PWM waveform with duty cycle 0% and duty cycle 100%.
A PWM waveform is set to duty cycle 0% when the compare match with the TRGGRB register is set as the counter clear source with the following:
• Value set in TRGGRA register > Value set in TRGGRB register
A PWM waveform is set to duty cycle 100% when the compare match with TRGGRA register is set as the counter clear source with the following:
• Value set in TRGGRB register > Value set in TRGGRA register
Output value is unchanged even if a compare match is generated with the following:
• Value set in TRGGRA register = Value set in TRGGRB register

Figure 8 - 25 Example of Operation in PWM Mode (2)



(a) Duty cycle 0%



(b) Duty cycle 100%

8.4.5 Phase counting mode

In phase counting mode, a phase difference between external input signals from two pins TRGCLKA and TRGCLKB is detected and the TRG register is incremented/decremented.

When phase counting mode is set when bits PM11 and PM15 in the PM1 register are 1, regardless of the settings of bits TRGTCK0 to TRGTCK2 and bits TRGCKEG0 and TRGCKEG1 in the TRGCR register, pins TRGCLKA and TRGCLKB automatically function as external clock input pins and the TRG register is incremented/decremented by bits CNTEN0 to CNTEN7 in the TRGCNTC register. However, bits TRGCCLR0 and TRGCCLR1 in the TRGCR register and registers TRGIOR, TRGIER, TRGSR, TRGGRA, and TRGGRB are enabled. This allows the input capture/output compare functions, PWM output function, and interrupt sources to be used.

The TRG register operates counting at both the rising/falling edges of pins TRGCLKA and TRGCLKB by bits CNTEN0 to CNTEN7.

Table 8 - 14 lists the Phase Counting Mode Specifications and Table 8 - 15 lists the Increment/Decrement Conditions for TRG Register.

Table 8 - 14 Phase Counting Mode Specifications

Item	Specification
Count source	External signal input to the TRGCLKj pin
Count operations	Increment/decrement
Count start condition	1 (count starts) is written to the TRGSTART bit in the TRGMR register.
Count stop condition	0 (count stops) is written to the TRGSTART bit in the TRGMR register.
Interrupt request generation timing	• Input capture (active edge of TRGIOj input) • Compare match (contents of registers TRG and TRGGRj match) • TRG register overflow • TRG register underflow
TRGIOA pin function	I/O port, input-capture input, output-compare output, or PWM output
TRGIOB pin function	I/O port, input-capture input, or output-compare output
TRGCLKA, TRGCLKB pin function	External clock input
Read from timer	The count value can be read by reading the TRG register.
Write to timer	The TRG register can be written to.
Selectable functions	• Selection of counter increment/decrement conditions Selectable by bits CNTEN0 to CNTEN7 in the TRGCNTC register. • Input capture/output compare functions and PWM function can be used.

Remark j = A or B

Table 8 - 15 Increment/Decrement Conditions for TRG Register

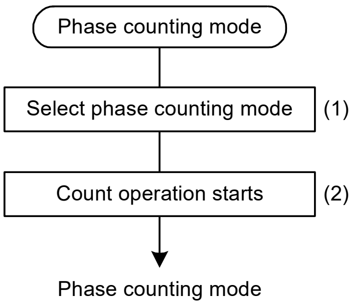
TRGCLKB pin	↑	High	↓	Low	High	↓	Low	↑
TRGCLKA pin	Low	↑	High	↓	↓	Low	↑	High
Bits CNTEN7 to CNTEN0 in TRGCNTC register	CNTEN7	CNTEN6	CNTEN5	CNTEN4	CNTEN3	CNTEN2	CNTEN1	CNTEN0
Count direction Note	+1	+1	+1	+1	-1	-1	-1	-1

Note The count direction when each bit in the TRGCNTC register is 1 (decrement or increment) is shown. When a bit is 0 (disabled), count is not performed.

(1) Procedure example for setting phase counting mode

Figure 8 - 26 shows a Procedure Example for Setting Phase Counting Mode.

Figure 8 - 26 Procedure Example for Setting Phase Counting Mode



(1) Set the TRGMDF bit in the TRGMR register to 1 to select phase counting mode.

(2) Set the TRGSTART bit in the TRGMR register to 1 to start count operation of the TRG register.

(2) Operation example

Figures 8 - 27 to 8 - 30 show Operation Examples in Phase Counting Mode.

In phase counting mode, the TRG register is incremented/decremented at both the rising(↑)/falling(↓) edges of pins TRGCLKA and TRGCLKB by bits CNTEN0 to CNTEN7 in the TRGCNTC register.

Figure 8 - 27 Operation Example 1 in Phase Counting Mode

• When the TRGCNTC register value is FFH

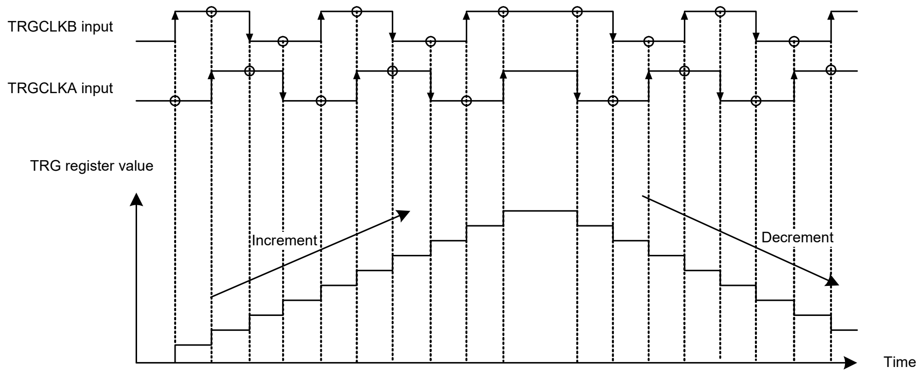


Figure 8 - 28 Operation Example 2 in Phase Counting Mode

• When the TRGCNTC register value is 24H

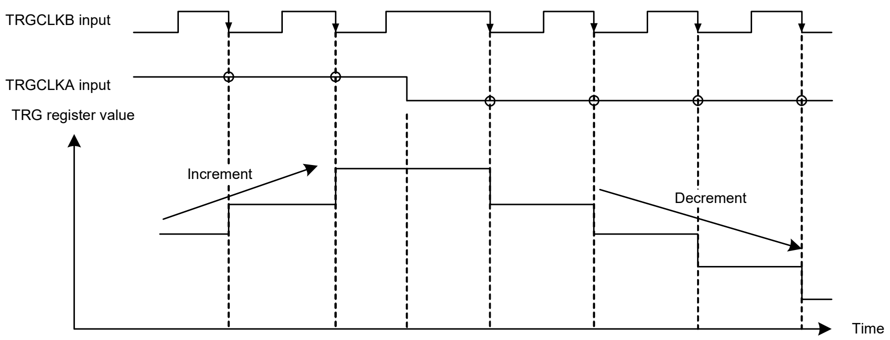


Figure 8 - 29 Operation Example 3 in Phase Counting Mode

• When the TRGCNTC register value is 28H

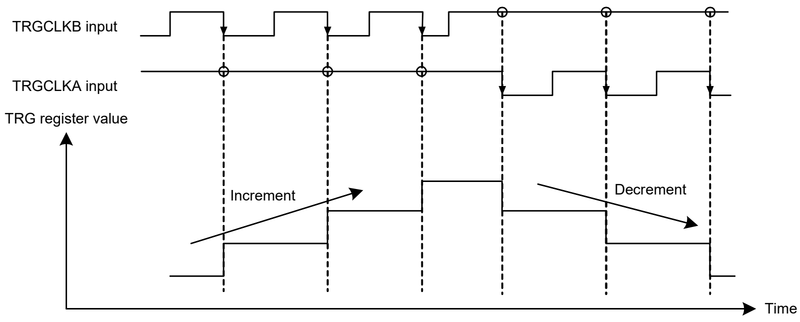
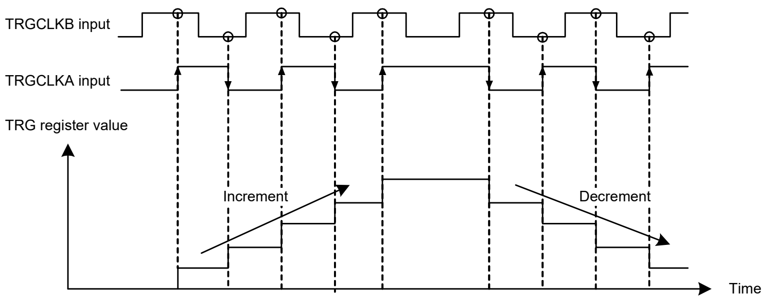


Figure 8 - 30 Operation Example 4 in Phase Counting Mode

• When the TRGCNTC register value is 5AH



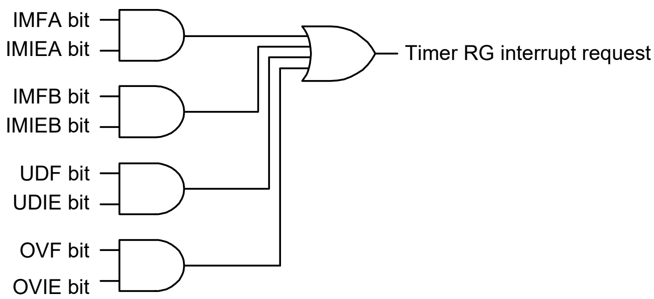
8.5 Timer RG Interrupt

Timer RG generates the timer RG interrupt request from four sources. Table 8 - 16 lists the Registers Associated with Timer RG Interrupt and Figure 8 - 31 shows the Timer RG Interrupt Block Diagram.

Table 8 - 16 Registers Associated with Timer RG Interrupt

	Timer RG Status Register	Timer RG Interrupt Enable Register	Interrupt Request Flag (Register)	Interrupt Mask Flag (Register)	Priority Specification Flag (Register)
Timer RG	TRGSR	TRGIER	TRGIF (IF2H)	TRGMK (MK2H)	TRGPR0 (PR02H) TRGPR1 (PR12H)

Figure 8 - 31 Timer RG Interrupt Block Diagram



IMFA, IMFB, UDF, OVF: Bits in TRGSR register
IMIEA, IMIEB, UDIE, OVIE: Bits in TRGSR register

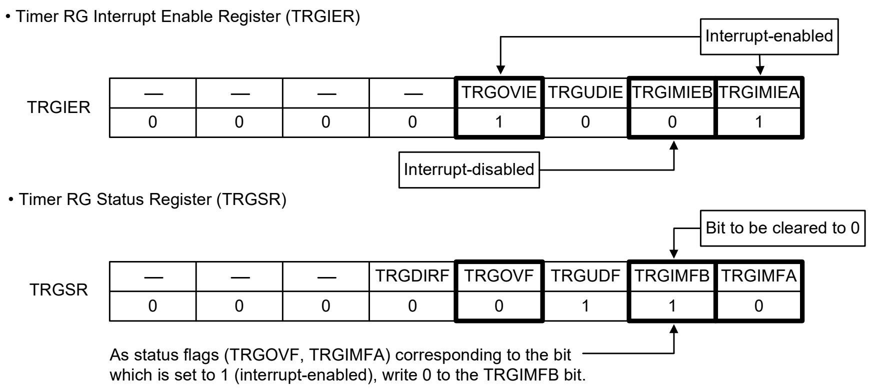
Since the interrupt source (timer RG interrupt) is generated by a combination of multiple interrupt request sources for timer RG, the following differences from other maskable interrupts except timer RD interrupt apply:

- When a bit in the TRGSR register is 1 and the corresponding bit in the TRGIER register is 1 (interrupt enabled), the TRGIF bit in the IF2H register is set to 1 (interrupt requested).
- If multiple bits in the TRGIER register are set to 1, use the TRGSR register to determine the source of the interrupt request.
- Since the bits in the TRGSR register are not automatically set to 0 even if the interrupt is acknowledged, set the corresponding bit to 0 in the interrupt routine.
- When status flags of interrupt sources (applicable status flags) of timer RG are set to 0 and their interrupts are disabled in the timer RG interrupt enable register (TRGIER), use either one of the following methods (a) to (c).

(a) Set 00H (all interrupts disabled) to the TRGIER register and write 0 to applicable status flags.

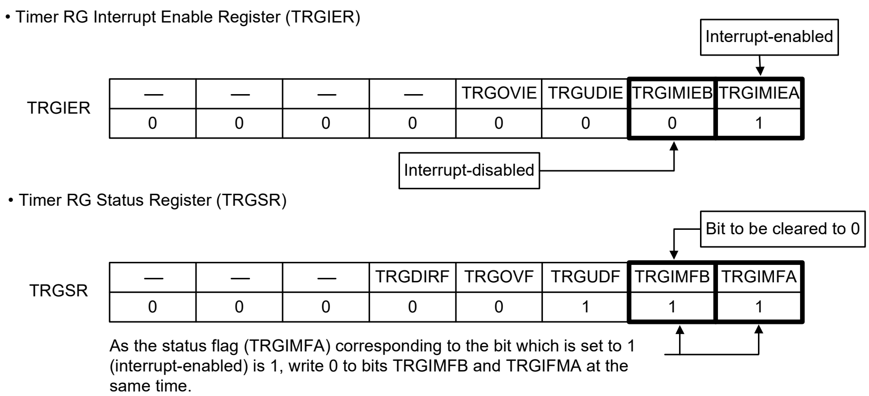
(b) When there are bits set to 1 (interrupt-enabled) in timer RG interrupt enable register (TRGIER) and status flags of interrupt sources related to their bits are 0, write 0 to applicable status flags.

Example: To clear the TRGIMFB bit to 0 when bits TRGIMIEA and TRGOVIE are set to 1 (interrupt-enabled) and the TRGIMIEB bit is set to 0 (interrupt-disabled).



(c) When there are bits set to 1 (interrupt-enabled) in the timer RG interrupt enable register (TRGIER) and status flags of interrupt sources related to their bits are 1, write 0 to these status flags and applicable status flags at the same time.

Example: To clear the TRGIMFB bit to 0 when the TRGIMIEA bit is set to 1 (interrupt-enabled) and the TRGIMIEB bit is set to 0 (interrupt-disabled).

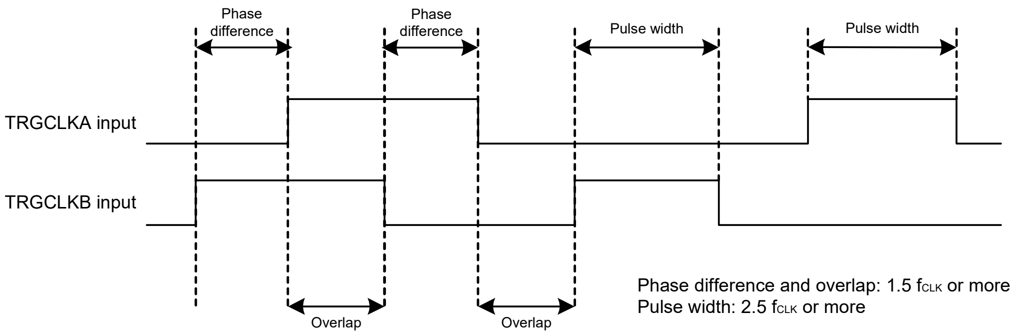


8.6 Cautions for Timer RG

8.6.1 Phase difference, overlap, and pulse width in phase counting mode

The phase difference and overlap between external input signals from pins TRGCLKA and TRGCLKB should be 1.5 f_{CLK} or more, respectively. The pulse width should be 2.5 f_{CLK} or more.
Figure 8 - 32 shows the Phase Difference, Overlap, and Pulse Width in Phase Counting Mode.

Figure 8 - 32 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode



8.6.2 Mode switching

- When switching modes during operation, set the TRGSTART bit in the TRGMR register to 0 (count stops) before switching.
- After switching modes, set the TRGIF bit to 0 before starting operation.
  Refer to **CHAPTER 22 INTERRUPT FUNCTIONS** for details.

8.6.3 Count source switching

- Stop the count before switching the count source [Note].

Changing procedure
(1) Set the TRGSTART bit in the TRGMR register to 0 (count stops).
(2) Change bits TRGTCK0 to TRGTCK2 in the TRGCR register.

Note The registers and bits that cannot be rewritten during count operation are as follows:
- All bits except TRGSTART in the TRGMR register
- The TRGCNTC register
- The TRGCR register
- The TRGIOR register

8.6.4 Procedure for setting pins TRGIOA and TRGIOB

To output from pins TRGIOA and TRGIOB, use the following setting procedure:

Changing procedure

(1) Set the mode and the initial value/output enabled (in order to make the initial value and enable settings using the same SFRs).

(2) Set the port register bits corresponding to pins TRGIOA and TRGIOB to 0.

(3) Set the port mode register bits corresponding to pins TRGIOA and TRGIOB to output mode (output is started from pins TRGIOA and TRGIOB).

(4) Start the count (TRGSTART in TRGMR register = 1).

To change the port mode register bits corresponding to pins TRGIOA and TRGIOB from output mode to input mode, use the following setting procedure:

(1) Set the port mode register bits corresponding to pins TRGIOA and TRGIOB to input mode (input is started from pins TRGIOA and TRGIOB).

(2) Set to the input capture function.

(3) Start the count (TRGSTART in TRGMR register = 1).

When switching pins TRGIOA and TRGIOB from output mode to input mode, input capture operation may be performed depending on the states of these pins. When the digital filter is not used, edge detection is performed after two or more cycles of the CPU clock have elapsed. When the digital filter is enabled, edge detection is performed after up to five cycles of the digital filter sampling clock.

8.6.5 External clock TRGCLKA, TRGCLKB

The pulse width of an external clock input to the TRGCLKj pin (j = A or B) should be set to three cycles or more of the timer RG operating clock (f_{CLK}).

8.6.6 SFR read/write access

When setting timer RG, set the TRGEN bit in the PER1 register to 1 first. If the TRGEN bit is 0, writes to the timer RG control registers are ignored and all the read values are the initial values (except for the port registers and the port mode registers).

(1) TRGMR register

Use the following setting procedure when switching the digital filter clock.

(a) With the TRGSTART bit set to 0 (count stops), set bits TRGDFA and TRGDFB (digital filter function select bits of pins TRGIOA and TRGIOB) in the TRGMR register, and bits TRGDFCK0 and TRGDFCK1 (clock select bits used by digital filter function) in the TRGMR register.

(b) Set the TRGSTART bit to 1.

However, when the digital filter is not set and TRGDFCK1 and TRGDFCK0 = 00B remain unchanged after a reset, the setting can be performed in a single step.

Besides external input pins (TRGIOA and TRGIOB), event input from the ELC can also be selected as an operating source for input capture. To use this function, set the TRGELCICE bit in the TRGMR register to 1, and set the input capture function (the rising edge as the active edge for input capture (TRGIOB2 to TRGIOB0 = 100B)).

This function is disabled in PWM mode and the timer mode output compare function (TRGPWM = 1 and TRGIOB2 = 0).

(2) TRG register

- Writing to the TRGMR register has priority over count reset operations generated by timer RG operating conditions.

8.6.7 Input capture operation when count is stopped

In input capture mode, an input capture interrupt request for the active edge of the TRGIOj input is also generated when the TRGSTART bit in the TRGMR register is 0 (count stops) if the edge selected by bits TRGIOj0 and TRGIOj1 in the TRGIOR register is input to the TRGIOj pin (j = A or B).

CHAPTER 9 REAL-TIME CLOCK

9.1 Functions of Real-time Clock

The real-time clock has the following features.

• Having counters of year, month, week, day, hour, minute, and second, and can count up to 99 years.
• Constant-period interrupt function (period: 0.5 seconds, 1 second, 1 minute, 1 hour, 1 day, 1 month)
• Alarm interrupt function (alarm: week, hour, minute)
• Pin output function of 1 Hz
• Watch error correction register

Caution The year, month, week, day, hours, minutes, and seconds can be counted by selecting the high-speed on-chip oscillator or high-speed system clock as the real-time clock operating clock. (Note that, because the RTC operating clock f_{RTC} is approximately equal to 32.787 kHz, the counted year, month, week, day, hours, minutes, and seconds includes a clock error.)
Select the source and frequency divisor for the operating clock by using the RTC clock select register (RTCCL).
Do not select fIL as the real-time clock operating clock.
When the real-time clock is operating, make sure that OSMC.WUTMMCK0 is 0.

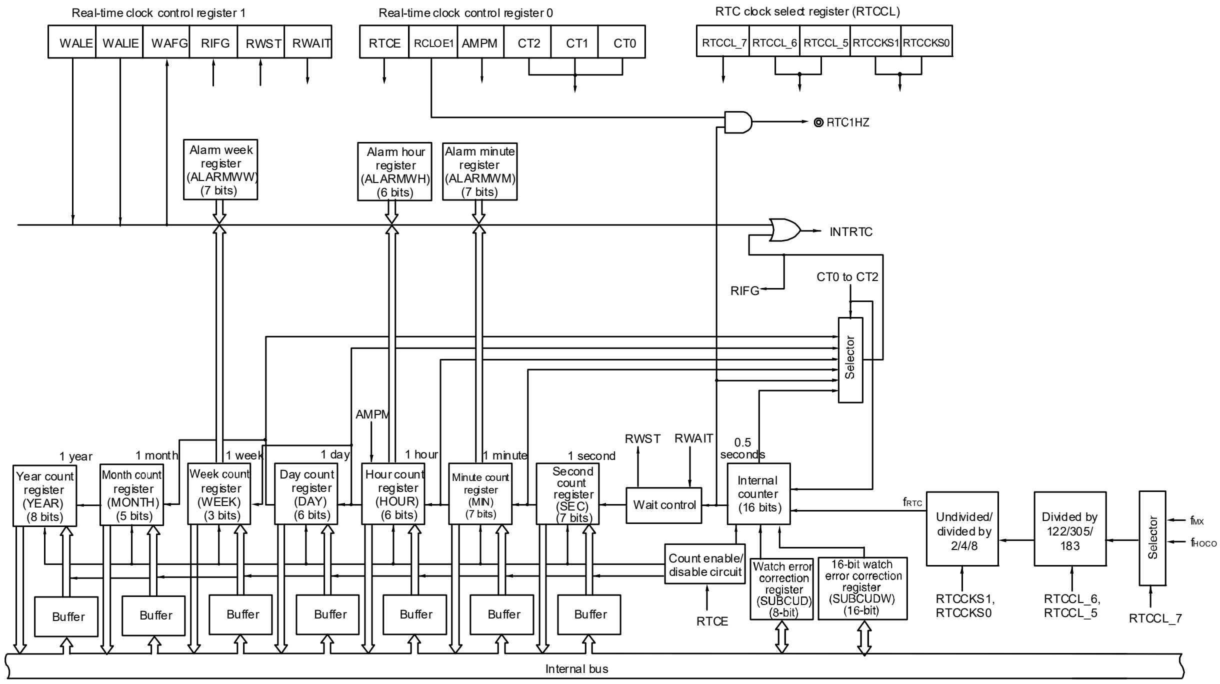
9.2 Configuration of Real-time Clock

The real-time clock includes the following hardware.

Table 9 - 1 Configuration of Real-time Clock

Item	Configuration
Counter	Internal counter (16-bit)
Control registers	Peripheral enable register 0 (PER0)
	Subsystem clock supply mode control register (OSMC)
	RTC clock select register (RTCCL)
	Real-time clock control register 0 (RTCC0)
	Real-time clock control register 1 (RTCC1)
	Second count register (SEC)
	Minute count register (MIN)
	Hour count register (HOUR)
	Day count register (DAY)
	Week count register (WEEK)
	Month count register (MONTH)
	Year count register (YEAR)
	Watch error correction register (SUBCUD)
	16-bit watch error correction register (SUBCUDW)
	Alarm minute register (ALARMWM)
	Alarm hour register (ALARMWH)
	Alarm week register (ALARMWW)

Figure 9 - 1 Block Diagram of Real-time Clock



Caution **The year, month, week, day, hours, minutes, and seconds can be counted by selecting the high-speed on-chip oscillator or high-speed system clock as the real-time clock operating clock. (Note that, because the RTC operating clock f_{RTC} is approximately equal to 32.787 kHz, the counted year, month, week, day, hours, minutes, and seconds includes a clock error.)**
Select the source and frequency divisor for the operating clock by using the RTC clock select register (RTCCL).
Do not select fIL as the real-time clock operating clock.
When the real-time clock is operating, make sure that OSMC.WUTMMCK0 is 0.

9.3 Registers Controlling Real-time Clock

The real-time clock is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Subsystem clock supply mode control register (OSMC)
- RTC clock select register (RTCCL)
- Real-time clock control register 0 (RTCC0)
- Real-time clock control register 1 (RTCC1)
- Second count register (SEC)
- Minute count register (MIN)
- Hour count register (HOUR)
- Day count register (DAY)
- Week count register (WEEK)
- Month count register (MONTH)
- Year count register (YEAR)
- Watch error correction register (SUBCUD)
- 16-bit watch error correction register (SUBCUDW)
- Alarm minute register (ALARMWM)
- Alarm hour register (ALARMWH)
- Alarm week register (ALARMWW)
- Port mode register 1 (PM1)
- Port register 1 (P1)

9.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the real-time clock is used, be sure to set bit 7 (RTCEN) 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 9 - 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	RTCEN	0	ADCEN	0	0	SAU0EN	TAU1EN	TAU0EN

RTCEN	Control of real-time clock (RTC) and interval timer input clock supply
0	Stops input clock supply. • SFRs used by the real-time clock (RTC) and interval timer cannot be written. • The real-time clock (RTC) and interval timer are in the reset status.
1	Input clock supply. • SFRs used by the real-time clock (RTC) and interval timer can be read/written.

Caution 1. When using the real-time clock, first set the RTCEN bit to 1 and then set the following registers, while oscillation of the count clock (f_{RTC}) is stable. If RTCEN = 0, writing to the control registers of the real-time clock is ignored, and, even if the registers are read, only the default values are read (except for the subsystem clock supply mode control register (OSMC), RTC clock select register (RTCCL), port mode register 1 (PM1), and port register 1 (P1)).

- **Real-time clock control register 0 (RTCC0)**
- **Real-time clock control register 1 (RTCC1)**
- **Second count register (SEC)**
- **Minute count register (MIN)**
- **Hour count register (HOUR)**
- **Day count register (DAY)**
- **Week count register (WEEK)**
- **Month count register (MONTH)**
- **Year count register (YEAR)**
- **Watch error correction register (SUBCUD)**
- **16-bit watch error correction register (SUBCUDW)**
- **Alarm minute register (ALARMWM)**
- **Alarm hour register (ALARMWH)**
- **Alarm week register (ALARMWW)**

Caution 2. Specify the RTC operating clock by using the RTCCL register before setting RTCEN to 1.

Caution 3. Be sure to clear bits 3, 4, and 6 to 0.

9.3.2 Subsystem clock supply mode control register (OSMC)

The WUTMMCK0 bit can be used to select the operating clock of the real-time clock, interval timer, and timer RJ.
Be sure to set the WUTMMCK0 bit to 0 when using the real-time clock.
The OSMC register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 9 - 3 Format of Subsystem clock supply mode control register (OSMC)

Address: F00F3H After reset: 00H R/W

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

WUTMMCK0	Selection of operating clock (f_{RTC}) for real-time clock, interval timer, and timer RJ
0	The RTC operating clock (f_{RTC}) specified in the RTC clock select register (RTCCL) • The RTC operating clock is selected as the count clock for the real-time clock and the interval timer. • The low-speed on-chip oscillator cannot be selected as the count source for timer RJ.
1	• The low-speed on-chip oscillator clock is selected as the count clock for the interval timer. • The low-speed on-chip oscillator can be selected as the count source for timer RJ.

Caution The year, month, week, day, hours, minutes, and seconds can be counted by selecting the high-speed on-chip oscillator or high-speed system clock as the real-time clock operating clock. (Note that, because the RTC operating clock f_{RTC} is approximately equal to 32.787 kHz, the counted year, month, week, day, hours, minutes, and seconds includes a clock error.)
Select the source and frequency divisor for the operating clock by using the RTC clock select register (RTCCL).
Do not select fIL as the real-time clock operating clock.
When the real-time clock is operating, make sure that OSMC.WUTMMCK0 is 0.

9.3.3 RTC clock select register (RTCCL)

This register is used to select the operating clock for the RTC and interval timer.
The RTCCL register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 9 - 4 Format of RTC clock select register (RTCCL)

Address: F02D8H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RTCCL	RTCCL7	RTCCL6	RTCCL5	0	0	0	RTCCKS1	RTCCKS0

RTCCL7	Selection of source of operating clock of RTC and interval timer
0	High-speed system clock (f_{MX})
1	High-speed on-chip oscillator clock (f_{HOCO})

RTCCL6	RTCCL5	Selection of frequency divisor for operating clock of RTC and interval timer 1
0	0	Divided by 122
0	1	Divided by 305
1	0	Divided by 183
1	1	Setting prohibited

RTCCKS1	RTCCKS0	Selection of frequency divisor for operating clock of RTC and interval timer 2
0	0	Undivided
0	1	Divided by 2
1	0	Divided by 4
1	1	Divided by 8

Remark The following shows the frequency division settings to generate a clock equivalent to 32.768 kHz. Select any of the following for operating the interval timer.

RTCCL7	RTCCL6	RTCCL5	RTCCKS1	RTCCKS0	Clock source	RTC operating clock (f_{RTC} = 32.787 kHz)
0	0	0	0	0	f_{MX} = 4 MHz	f_{MX}/122
0	0	0	0	1	f_{MX} = 8 MHz	f_{MX}/122/2
0	1	0	0	1	f_{MX} = 12 MHz	f_{MX}/183/2
0	0	0	1	0	f_{MX} = 16 MHz	f_{MX}/122/4
0	0	1	0	1	f_{MX} = 20 MHz	f_{MX}/305/2
1	0	0	0	0	f_{HOCO} = 4 MHz	f_{HOCO}/122
1	1	0	0	0	f_{HOCO} = 6 MHz	f_{HOCO}/183
1	0	0	0	1	f_{HOCO} = 8 MHz	f_{HOCO}/122/2
1	1	0	0	1	f_{HOCO} = 12 MHz	f_{HOCO}/183/2
1	0	0	1	0	f_{HOCO} = 16 MHz	f_{HOCO}/122/4
1	1	0	1	0	f_{HOCO} = 24 MHz	f_{HOCO}/183/4
1	0	0	1	1	f_{HOCO} = 32 MHz	f_{HOCO}/122/8
Other than above					Setting prohibited	

9.3.4 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 operation, control the RTC1HZ pin, and set a 12- or 24-hour system and the constant-period interrupt function.
The RTCC0 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 9 - 5 Format of Real-time clock control register 0 (RTCC0)

Address: FFF9DH After reset: 00H R/W

Symbol	<7>	6	<5>	4	3	2	1	0
RTCC0	RTCE	0	RCLOE1	0	AMPM	CT2	CT1	CT0

RTCE	Real-time clock operation control
0	Stops counter operation.
1	Starts counter operation.

RCLOE1	RTC1HZ pin output control
0	Disables output of the RTC1HZ pin (1 Hz).
1	Enables output of the RTC1HZ pin (1 Hz).

AMPM	Selection of 12-/24-hour system
0	12-hour system (a.m. and p.m. are displayed.)
1	24-hour system
• Rewrite the AMPM bit value after setting the RWAIT bit (bit 0 of real-time clock control register 1 (RTCC1)) to 1. If the AMPM bit value is changed, the values of the hour count register (HOUR) change according to the specified time system. • Table 9 - 2 shows the Displayed Time Digits.	

CT2	CT1	CT0	Constant-period interrupt (INTRTC) selection
0	0	0	Does not use fixed-cycle interrupt function.
0	0	1	Once per 0.5 s (synchronized with second count up)
0	1	0	Once per 1 s (same time as second count up)
0	1	1	Once per 1 m (second 00 of every minute)
1	0	0	Once per 1 hour (minute 00 and second 00 of every hour)
1	0	1	Once per 1 day (hour 00, minute 00, and second 00 of every day)
1	1	×	Once per 1 month (Day 1, hour 00 a.m., minute 00, and second 00 of every month)
When changing the values of the CT2 to CT0 bits while the counter operates (RTCE = 1), rewrite the values of the CT2 to CT0 bits after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, after rewriting the values of the CT2 to CT0 bits, enable interrupt servicing after clearing the RIFG and RTCIF flags.			

Caution 1. Do not change the value of the RTCLOE1 bit when RTCE = 1.
Caution 2. 1 Hz is not output even if RCCOE1 is set to 1 when RTCE = 0.

Remark ×: Don't care

9.3.5 Real-time clock control register 1 (RTCC1)

The RTCC1 register is an 8-bit register that is used to control the alarm interrupt function and the wait time of the counter.
The RTCC1 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 9 - 6 Format of Real-time clock control register 1 (RTCC1) (1/2)

Address: FFF9EH After reset: 00H R/W

Symbol	<7>	<6>	5	<4>	<3>	2	<1>	<0>
RTCC1	WALE	WALIE	0	WAFG	RIFG	0	RWST[Note]	RWAIT

WALE	Alarm operation control
0	Match operation is invalid.
1	Match operation is valid.
When setting a value to the WALE bit while the counter operates (RTCE = 1) and WALIE = 1, rewrite the WALE bit after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG and RTCIF flags after rewriting the WALE bit. When setting each alarm register (WALIE flag of real-time clock control register 1 (RTCC1), the alarm minute register (ALARMWM), the alarm hour register (ALARMWH), and the alarm week register (ALARMWW)), set match operation to be invalid (“0”) for the WALE bit.	

WALIE	Control of alarm interrupt (INTRTC) function operation
0	Does not generate interrupt on matching of alarm.
1	Generates interrupt on matching of alarm.

WAFG	Alarm detection status flag
0	Alarm mismatch
1	Detection of matching of alarm
This is a status flag that indicates detection of matching with the alarm. It is valid only when WALE = 1 and is set to “1” one cycle of f_{RTC} after matching of the alarm is detected. This flag is cleared when “0” is written to it. Writing “1” to it is invalid.	

Note The RWST bit is read-only.

Figure 9 - 7 Format of Real-time clock control register 1 (RTCC1) (2/2)

RIFG	Constant-period interrupt status flag
0	Fixed-cycle interrupt is not generated.
1	Fixed-cycle interrupt is generated.
This flag indicates the status of generation of the fixed-cycle interrupt. When the fixed-cycle interrupt is generated, it is set to “1”. This flag is cleared when “0” is written to it. Writing “1” to it is invalid.	

RWST[Note 1]	Wait status flag of real-time clock
0	Counter is operating.
1	Mode to read or write counter value
This status flag indicates whether the setting of the RWAIT bit is valid. Before reading or writing the counter value, confirm that the value of this flag is 1.	

RWAIT	Wait control of real-time clock
0	Sets counter operation.
1	Stops SEC to YEAR counters. Mode to read or write counter value
This bit controls the operation of the counter. Be sure to write “1” to it to read or write the counter value. As the internal counter (16-bit) is continuing to run, complete reading or writing within one second and turn back to 0. When reading or writing to the counter is required while generation of the alarm interrupt is enabled, first set the CT2 to CT0 bits to 010B (generating the constant-period interrupt once per 1 second). Then, complete the processing from setting the RWAIT bit to 1 to setting it to 0 before generation of the next constant-period interrupt. When RWAIT = 1, it takes up to one cycle of f_{RTC} until the counter value can be read or written (RWST = 1).[Notes 2, 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. The RWST bit is read-only.

Note 2. When the RWAIT bit is set to 1 within one cycle of f_{RTC} clock after setting the RTCE bit to 1, the RWST bit being set to 1 may take up to two cycles of the operating clock (f_{RTC}).

Note 3. When the RWAIT bit is set to 1 within one cycle of f_{RTC} clock after release from the standby mode (HALT mode, STOP mode, or SNOOZE mode), the RWST bit being set to 1 may take up to two cycles of the operating clock (f_{RTC}).

Caution **If writing is performed to the RTCC1 register with a 1-bit manipulation instruction, the RIFG flag and WAFG flag may be cleared. Therefore, to perform writing to the RTCC1 register, be sure to use an 8-bit manipulation instruction. To prevent the RIFG flag and WAFG flag from being cleared during writing, disable writing by setting 1 to the corresponding bit. If the RIFG flag and WAFG flag are not used and the value may be changed, the RTCC1 register may be written by using a 1-bit manipulation instruction.**

Remark 1. Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

Remark 2. The internal counter (16 bits) is cleared when the second count register (SEC) is written.

9.3.6 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 counts up when the internal counter (16-bit) overflows.
When data is written to this register, it is written to a buffer and then to the counter up to two cycles of fRTC 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 clears this register to 00H.

Figure 9 - 8 Format of Second count register (SEC)

Address: FFF92H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SEC	0	SEC40	SEC20	SEC10	SEC8	SEC4	SEC2	SEC1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in 9.4.3 Reading/writing real-time clock.

Remark The internal counter (16 bits) is cleared when the second count register (SEC) is written.

9.3.7 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 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 two cycles of fRTC 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 clears this register to 00H.

Figure 9 - 9 Format of Minute count register (MIN)

Address: FFF93H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
MIN	0	MIN40	MIN20	MIN10	MIN8	MIN4	MIN2	MIN1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in 9.4.3 Reading/writing real-time clock.

9.3.8 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 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 two cycles 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 clears this register to 12H.
However, the value of this register is 00H if the AMPM bit (bit 3 of the RTCC0 register) is set to 1 after reset.

Figure 9 - 10 Format of Hour count register (HOUR)

Address: FFF94H After reset: 12H R/W

Symbol	7	6	5	4	3	2	1	0
HOUR	0	0	HOUR20	HOUR10	HOUR8	HOUR4	HOUR2	HOUR1

Caution 1. Bit 5 (HOUR20) of the HOUR register indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

Caution 2. When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in 9.4.3 Reading/writing real-time clock.

Table 9 - 2 shows the relationship between the setting value of the AMPM bit, the hour count register (HOUR) value, and time.

Table 9 - 2 Displayed Time Digits

24-Hour Display (AMPM = 1)		12-Hour Display (AMPM = 1)	
Time	HOUR Register	Time	HOUR Register
0	00 H	12 a.m.	12 H
1	01 H	1 a.m.	01 H
2	02 H	2 a.m.	02 H
3	03 H	3 a.m.	03 H
4	04 H	4 a.m.	04 H
5	05 H	5 a.m.	05 H
6	06 H	6 a.m.	06 H
7	07 H	7 a.m.	07 H
8	08 H	8 a.m.	08 H
9	09 H	9 a.m.	09 H
10	10 H	10 a.m.	10 H
11	11 H	11 a.m.	11 H
12	12 H	12 p.m.	32 H
13	13 H	1 p.m.	21 H
14	14 H	2 p.m.	22 H
15	15 H	3 p.m.	23 H
16	16 H	4 p.m.	24 H
17	17 H	5 p.m.	25 H
18	18 H	6 p.m.	26 H
19	19 H	7 p.m.	27 H
20	20 H	8 p.m.	28 H
21	21 H	9 p.m.	29 H
22	22 H	10 p.m.	30 H
23	23 H	11 p.m.	31 H

The HOUR register value is set to 12-hour display when the AMPM bit is “0” and to 24-hour display when the AMPM bit is “1”.
In 12-hour display, the fifth bit of the HOUR register displays 0 for AM and 1 for PM.

9.3.9 Day count register (DAY)

The DAY register is an 8-bit register that takes a value of 1 to 31 (decimal) and indicates the count value of days.
It counts up when the hour counter overflows.
This counter counts as follows.

- 01 to 31 (January, March, May, July, August, October, December)
- 01 to 30 (April, June, September, November)
- 01 to 29 (February, leap year)
- 01 to 28 (February, normal year)

When data is written to this register, it is written to a buffer and then to the counter up to two cycles 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 clears this register to 01H.

Figure 9 - 11 Format of Day count register (DAY)

Address: FFF96H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
DAY	0	0	DAY20	DAY10	DAY8	DAY4	DAY2	DAY1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in 9.4.3 Reading/writing real-time clock.

9.3.10 Week count register (WEEK)

The WEEK register is an 8-bit register that takes a value of 0 to 6 (decimal) and indicates the count value of weekdays.
It counts up in synchronization with the day counter.
When data is written to this register, it is written to a buffer and then to the counter up to two cycles 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 clears this register to 00H.

Figure 9 - 12 Format of Week count register (WEEK)

Address: FFF95H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
WEEK	0	0	0	0	0	WEEK4	WEEK2	WEEK1

Caution 1. The value corresponding to the month count register (MONTH) or the day count register (DAY) is not stored in the week count register (WEEK) automatically. After reset release, set the week count register as follow.

Day	WEEK
Sunday	00 H
Monday	01 H
Tuesday	02 H
Wednesday	03 H
Thursday	04 H
Friday	05 H
Saturday	06 H

Caution 2. When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in 9.4.3 Reading/writing real-time clock.

9.3.11 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 counts up when the day counter overflows.
When data is written to this register, it is written to a buffer and then to the counter up to two cycles 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 clears this register to 01H.

Figure 9 - 13 Format of Month count register (MONTH)

Address: FFF97H After reset: 01H R/W

Symbol	7	6	5	4	3	2	1	0
MONTH	0	0	0	MONTH10	MONTH8	MONTH4	MONTH2	MONTH1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in 9.4.3 Reading/writing real-time clock.

9.3.12 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 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 two cycles 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 clears this register to 00H.

Figure 9 - 14 Format of Year count register (YEAR)

Address: FFF98H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
YEAR	YEAR80	YEAR40	YEAR20	YEAR10	YEAR8	YEAR4	YEAR2	YEAR1

Caution When it reads or writes from/to the register while the counter is in operation (RTCE = 1), follow the procedures described in 9.4.3 Reading/writing real-time clock.

9.3.13 Watch error correction register (SUBCUD)

This register is used to correct the watch with high accuracy when it is slow or fast by changing the value that overflows from the internal counter (16-bit) to the second count register (SEC) (reference value: 7FFFH).
The SUBCUD register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 9 - 15 Format of Watch error correction register (SUBCUD)

Address: FFF99H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SUBCUD	DEV	F12	F5	F4	F3	F2	F1	F0

DEV	Setting of watch error correction timing
0	Corrects watch error when the second digits are at 00, 20, or 40 (every 20 seconds).
1	Corrects watch error only when the second digits are at 00 (every 60 seconds).
Writing to the SUBCUD register at the following timing is prohibited. • When DEV = 0 is set: For a period of SEC = 00H, 20H, 40H • When DEV = 1 is set: For a period of SEC = 00H	

F12	Setting of watch error correction value
0	Increases by {(F5, F4, F3, F2, F1, F0) – 1} × 2.
1	Decreases by {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2.
When (F12, F5, F4, F3, F2, F1, F0) = (*, 0, 0, 0, 0, 0, *), the watch error is not corrected. Setting (F12, F5, F4, F3, F2, F1, F0) to (1, 0, 0, 0, 0, 0, *) is prohibited. Range of correction value: (when F12 = 0) 2, 4, 6, 8, ... , 120, 122, 124 (when F12 = 1) –2, –4, –6, –8, ... , –120, –122, –124	

Caution 1. * indicates 0 or 1.
Caution 2. / indicates the inverted values of the bits.

The range of value that can be corrected by using the watch error correction register (SUBCUD) is shown below.

Item	DEV = 0 (correction every 20 seconds Note)	DEV = 1 (correction every 60 seconds Note)
Correctable range	–189.2 ppm to 189.2 ppm	–63.1 ppm to 63.1 ppm
Maximum excludes quantization error	±1.53 ppm	±0.51 ppm
Minimum resolution	±3.05 ppm	±1.02 ppm

Note Because the selected clock f_{RTC} is approximately equal to 32.787 kHz, the counted value includes a clock error.

Caution If a correctable range is −63.1 ppm or lower and 63.1 ppm or higher, set 0 to DEV.

9.3.14 16-bit watch error correction register (SUBCUDW)

This register is used to correct the watch with high accuracy when it is slow or fast by changing the value that overflows from the internal counter (16-bit) to the second count register (SEC) (reference value: 7FFFH).

The SUBCUDW register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 9 - 16 Format of 16-bit watch error correction register (SUBCUDW)

Address: FFF54H After reset: 0000H R/W

Symbol	7	6	5	4	3	2	1	0
SUBCUDW	DEV	0	0	F12	F11	F10	F9	F8
Symbol	7	6	5	4	3	2	1	0
	F7	F6	F5	F4	F3	F2	F1	F0

DEV	Setting of watch error correction timing
0	Corrects watch error when the second digits are at 00, 20, or 40 (every 20 seconds).
1	Corrects watch error only when the second digits are at 00 (every 60 seconds).
Writing to the SUBCUD register at the following timing is prohibited. • When DEV = 0 is set: For a period of SEC = 00H, 20H, 40H • When DEV = 1 is set: For a period of SEC = 00H	

F12	Setting of watch error correction value
0	Increases by {(F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) – 1} × 2.
1	Decreases by {(/F11, /F10, /F9, /F8, /F7, /F6, /F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2.
When (F12, F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) = (*, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, *), the watch error is not corrected. Range of correction value: (when F12 = 0) 2, 4, 6, 8, ... , 8184, 8186, 8188 (when F12 = 1) –2, –4, –6, –8, ... , –8184, –8186, –8188	

Caution 1. * indicates 0 or 1.

Caution 2. / indicates the inverted values of the bits.

The range of value that can be corrected by using the 16-bit watch error correction register (SUBCUDW) is shown below.

Item	DEV = 0 (correction every 20 seconds Note)	DEV = 1 (correction every 60 seconds Note)
Correctable range	–12496.9 ppm to 12496.9 ppm	–4165.6 ppm to 4165.6 ppm
Maximum excludes quantization error	±1.53 ppm	±0.51 ppm
Minimum resolution	±3.05 ppm	±1.02 ppm

Note Because the selected clock f_{RTC} is approximately equal to 32.787 kHz, the counted value includes a clock error.

Remark If a correctable range is −4165.6 ppm or lower and 4165.6 ppm or higher, set DEV to 0.

9.3.15 Alarm minute register (ALARMWM)

This register is used to set minutes of alarm.
The ALARMWM register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Caution Set a decimal value of 00 to 59 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 9 - 17 Format of Alarm minute register (ALARMWM)

Address: FFF9AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWM	0	WM40	WM20	WM10	WM8	WM4	WM2	WM1

9.3.16 Alarm hour register (ALARMWH)

This register is used to set hours of alarm.
The ALARMWH register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 12H.
However, the value of this register is 00H if the AMPM bit is set to 1 after reset.

Caution Set a decimal value of 00 to 23, 01 to 12, or 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 9 - 18 Format of Alarm hour register (ALARMWH)

Address: FFF9BH After reset: 12H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWH	0	0	WH20	WH10	WH8	WH4	WH2	WH1

Caution Bit 5 (WH20) of the ALARMWH register indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

9.3.17 Alarm week register (ALARMWW)

This register is used to set date of alarm.
The ALARMWW register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 9 - 19 Format of Alarm week register (ALARMWW)

Address: FFF9CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ALARMWW	0	WW6	WW5	WW4	WW3	WW2	WW1	WW0

Here is an example of setting the alarm.

Time of Alarm	Day							12-Hour Display				24-Hour Display			
	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Hour 10	Hour 1	Minute 10	Minute 1	Hour 10	Hour 1	Minute 10	Minute 1
	WW0	WW1	WW2	WW3	WW4	WW5	WW6								
Every day, 0:00 a.m.	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
Every day, 1:30 a.m.	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
Every day, 11:59 a.m.	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
Monday through Friday, 0:00 p.m.	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
Sunday, 1:30 p.m.	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
Monday, Wednesday, Friday, 11:59 p.m.	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

9.3.18 Port mode register 1 (PM1)

The PM1 register can be set by a 1-bit or 8-bit manipulation instruction.
Reset signal generation sets this register to FFH.
When using port 1 as the RTC1HZ pin for output of 1 Hz, set the PM13 bit to 0.

Figure 9 - 20 Format of Port mode register 1 (PM1)

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10

9.3.19 Port register 1 (P1)

The P1 register can be set by a 1-bit or 8-bit manipulation instruction.
Reset signal generation sets this register to 00H.
When using port 1 as 1 Hz output to the RTC1Hz pin, set the P13 bit to 0.

Figure 9 - 21 Format of Port register 1 (P1)

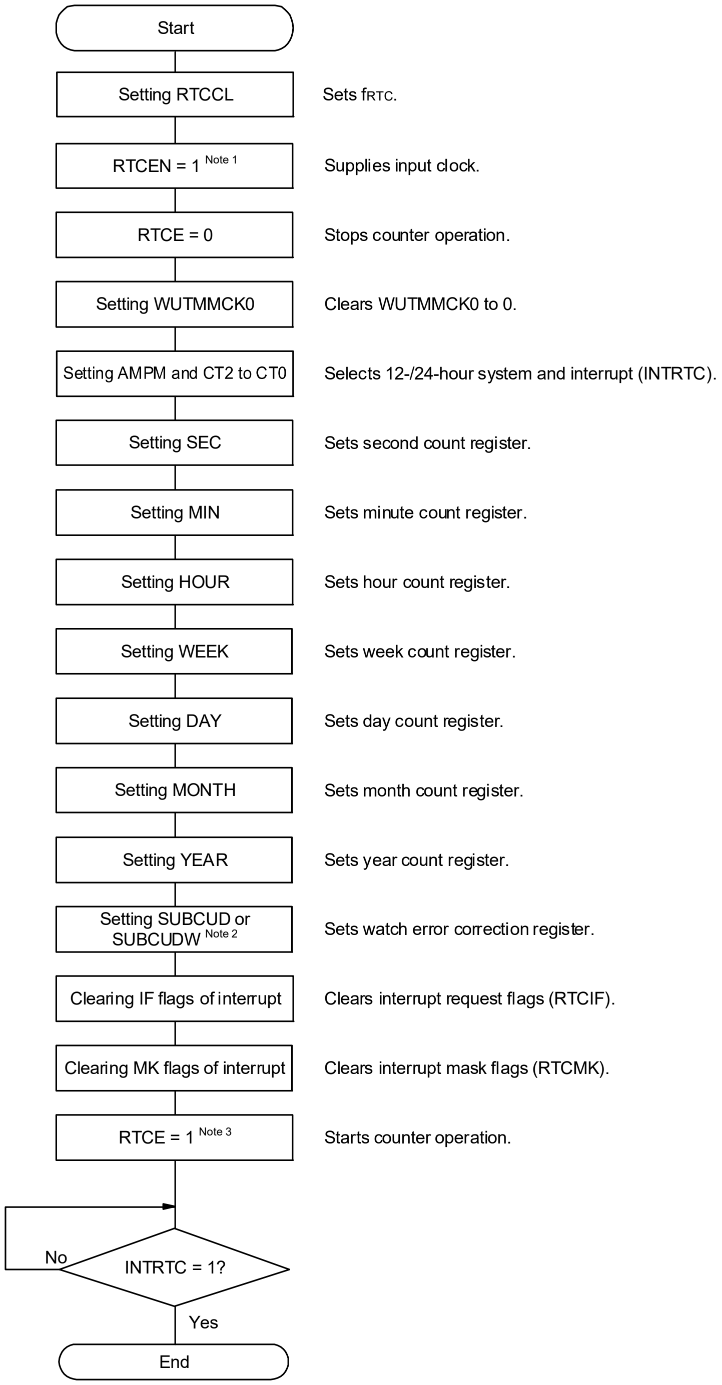
Address: FFF01H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
P1	P17	P16	P15	P14	P13	P12	P11	P10

9.4 Real-time Clock Operation

9.4.1 Starting operation of real-time clock

Figure 9 - 22 Procedure for Starting Operation of Real-time Clock



Note 1. First set the RTCEN bit to 1, while oscillation of the count clock (f_{RTC}) is stable.

Note 2. Set up the SUBCUD register only if the watch error must be corrected. Set up the SUBCUDW register if higher precision is required. For how to calculate the correction value, see **9.4.6 Example of watch error correction of real-time clock**.

Note 3. Confirm the procedure described in 9.4.2 Shifting to HALT/STOP mode after starting operation when shifting to HALT/STOP mode without waiting for INTRTC = 1 after RTCE = 1.

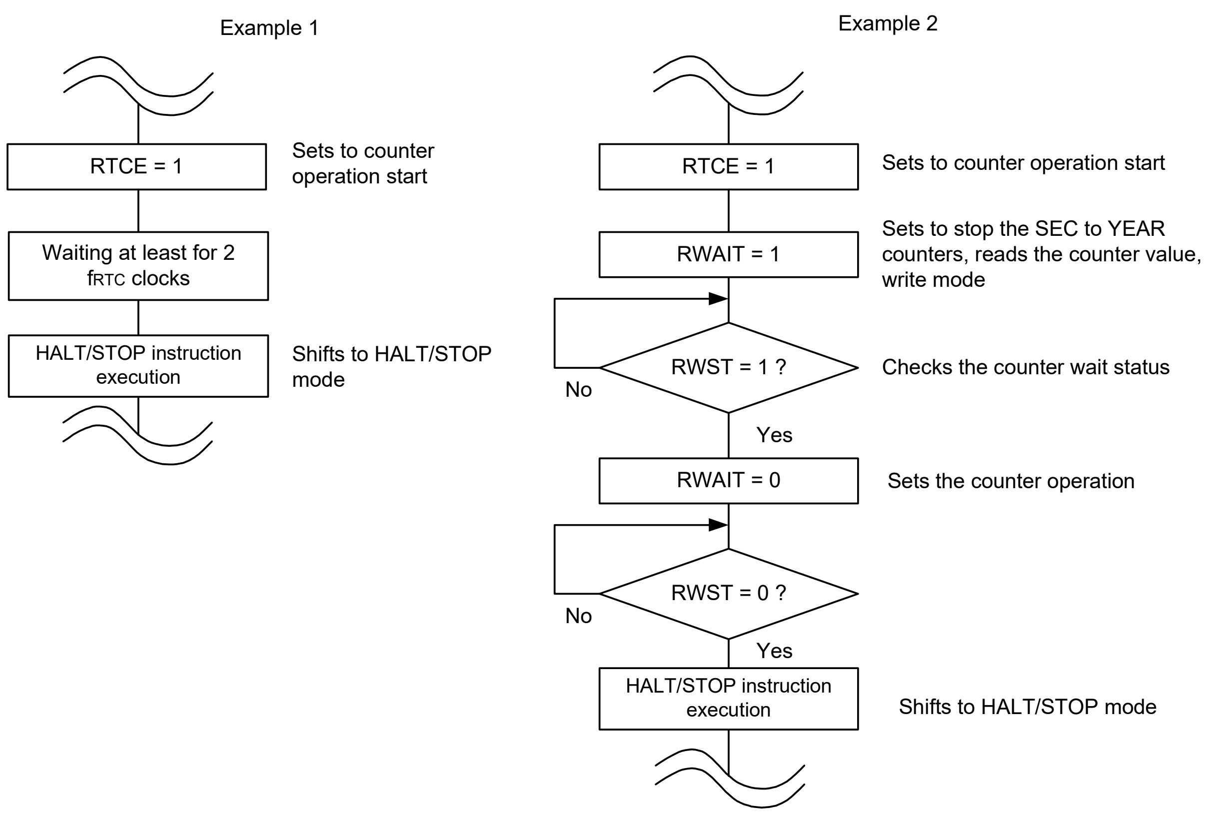
9.4.2 Shifting to HALT/STOP mode after starting operation

Perform one of the following processing when shifting to HALT/STOP mode immediately after setting the RTCE bit to 1.
However, after setting the RTCE bit to 1, this processing is not required when shifting to HALT/STOP mode after INTRTC interrupt has occurred.

- Shifting to HALT/STOP mode when at least two cycles of the count clock (fRTC) have elapsed after setting the RTCE bit to 1 (see **Figure 9 - 23**, **Example 1**).
- Checking by polling the RWST bit to become 1, after setting the RTCE bit to 1 and then setting the RWAIT bit to 1. Afterward, setting the RWAIT bit to 0 and shifting to HALT/STOP mode after checking again by polling that the RWST bit has become 0 (see **Figure 9 - 23**, **Example 2**).

Caution The RTC operating clock (fRTC) stops in STOP mode.

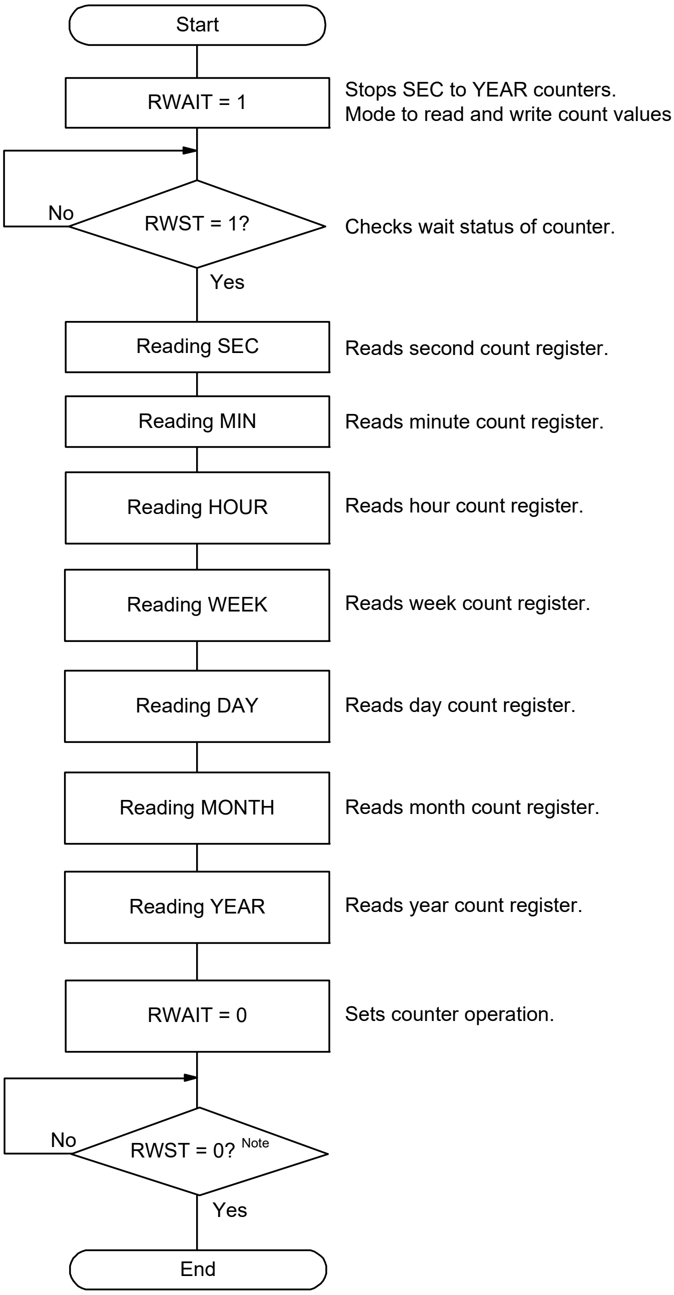
Figure 9 - 23 Procedure for Shifting to HALT/STOP Mode After Setting RTCE bit to 1



9.4.3 Reading/writing real-time clock

Read or write the counter after setting RWAIT to 1.
Clear RWAIT to 0 after completion of reading or writing the counter.

Figure 9 - 24 Procedure for Reading Real-time Clock

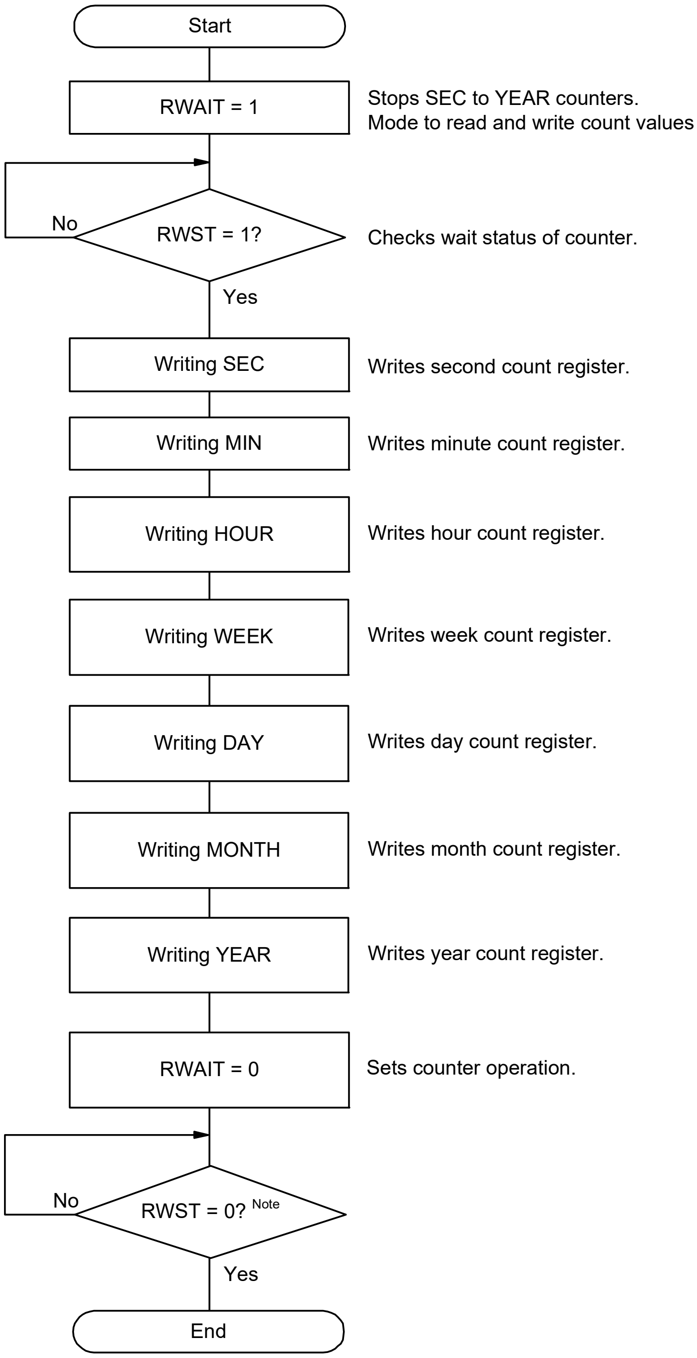


Note Make sure that RWST is 0 before shifting to HALT/STOP mode.

Caution **Complete setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.**
When reading to the counter is required while generation of the alarm interrupt is enabled, first set the CT2 to CT0 bits to 010B (generating the constant-period interrupt once per 1 second). Then, complete the processing from setting the RWAIT bit to 1 to setting it to 0 before generation of the next constant-period interrupt.

Remark The second count register (SEC), minute count register (MIN), hour count register (HOUR), week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be read in any sequence.
All the registers do not have to read and only some registers may be read.

Figure 9 - 25 Procedure for Writing Real-time Clock



Note Make sure that RWST is 0 before shifting to HALT/STOP mode.

Caution 1. Complete setting the RWAIT bit to 1 to clearing the RWAIT bit to 0 within 1 second.
When writing to the counter is required while generation of the alarm interrupt is enabled, first set the CT2 to CT0 bits to 010B (generating the constant-period interrupt once per 1 second). Then, complete the processing from setting the RWAIT bit to 1 to setting it to 0 before generation of the next constant-period interrupt.

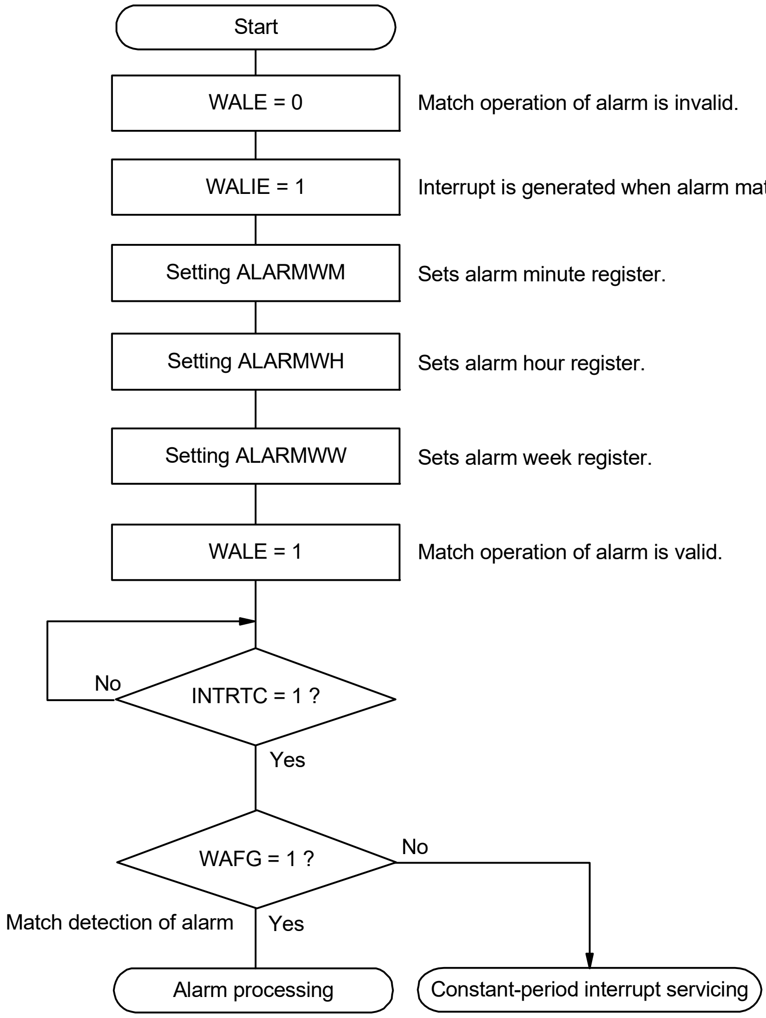
Caution 2. When changing a value of the SEC, MIN, HOUR, WEEK, DAY, MONTH, or YEAR register while the counter is operating (RTCE = 1), first disable servicing of interrupt INTRTC by using the interrupt mask flag register. Furthermore, clear the WAFG, RIFG and RTCIF flags after rewriting the value.

Remark The second count register (SEC), minute count register (MIN), hour count register (HOUR), week count register (WEEK), day count register (DAY), month count register (MONTH), and year count register (YEAR) may be written in any sequence.
All the registers do not have to be set and only some registers may be written.

9.4.4 Setting alarm of real-time clock

Set time of alarm after setting 0 to WALE (alarm operation invalid) first.

Figure 9 - 26 Alarm Setting Procedure

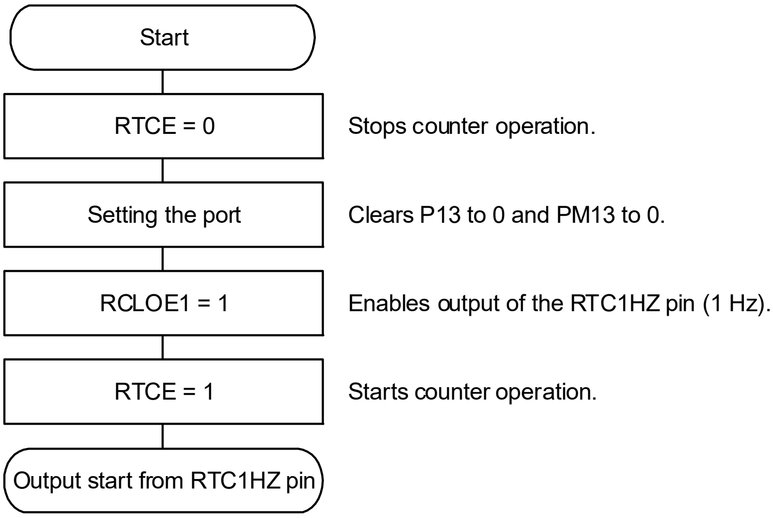


Remark 1. The alarm week register (ALARMWW), alarm hour register (ALARMWH), and alarm week register (ALARMWW) may be written in any sequence.

Remark 2. Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be determined by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) when INTRTC occurs.

9.4.5 1 Hz output of real-time clock

Figure 9 - 27 1 Hz Output Setting Procedure



Caution First set the RTCEN bit to 1, while oscillation of the count clock (the clock selected in the RTCCL register) is stable.

9.4.6 Example of watch error correction of real-time clock

The watch can be corrected with high accuracy when it is slow or fast, by setting a value to the 16-bit watch error correction register (SUBCUDW).

Caution The following is an example for correcting the f_{RTC} clock frequency.

Example of calculating the correction value

The correction value used when correcting the count value of the internal counter (16-bit) is calculated by using the following expression.
Set the DEV bit to 0 when the correction range is –4165.6 ppm or less, or 4165.6 ppm or more.

When DEV = 0:
Correction value Note = Number of correction counts in 1 minute ÷ 3 = (Oscillation frequency ÷ Target frequency − 1) × 32768 × 60 ÷ 3

When DEV = 1:
Correction value Note = Number of correction counts in 1 minute = (Oscillation frequency ÷ Target frequency − 1) × 32768 × 60

Note The correction value is the watch error correction value calculated by using bits 12 to 0 of the 16-bit watch error correction register (SUBCUDW).
When F12 = 0: Correction value = {(F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) − 1} × 2
When F12 = 1: Correction value = − {(/F11, /F10, /F9, /F8, /F7, /F6, /F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2

When (F12, F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) is (*, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, *), the watch error is not corrected. "*" is 0 or 1.
/F11 to /F0 are bit-inverted values (000000000011 when 111111111100).

Remark 1. The correction value is 2, 4, 6, 8, … 8184, 8186, 8188 or −2, −4, −6, −8, … −8184, −8186, −8188.
Remark 2. The oscillation frequency is the count clock (f_{RTC}).
It can be calculated from the output frequency of the RTC1HZ pin × 32768 when the 16-bit watch error correction register (SUBCUDW) is set to its initial value (0000H).
Remark 3. The target frequency is the frequency resulting after correction performed by using the 16-bit watch error correction register (SUBCUDW).

Correction example

Example of correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm)

Measuring the oscillation frequency

The oscillation frequency Note of each product is measured by outputting about 1 Hz from the RTC1HZ pin when the 16-bit watch error correction register (SUBCUDW) is set to its initial value (0000H).

Note See **9.4.5 1 Hz output of real-time clock** for the setting procedure of outputting about 1 Hz from the RTC1HZ pin.

Calculating the correction value

When the output frequency from the RTCCL pin is 0.9999817 Hz:

Oscillation frequency = 32768 × 0.9999817 ≈ 32767.4 Hz

Assume the target frequency to be 32768 Hz (32767.4 Hz + 18.3 ppm) and DEV to be 1.

The expression for calculating the correction value when DEV is 1 is applied.

Correction value = Number of correction counts in 1 minute
= (Oscillation frequency ÷ Target frequency – 1) × 32768 × 60
= (32767.4 ÷ 32768 – 1) × 32768 × 60
= –36

Calculating the values to be set to (F12 to F0)

When the correction value is –36

If the correction value is 0 or less (when quickening), assume F12 to be 1.

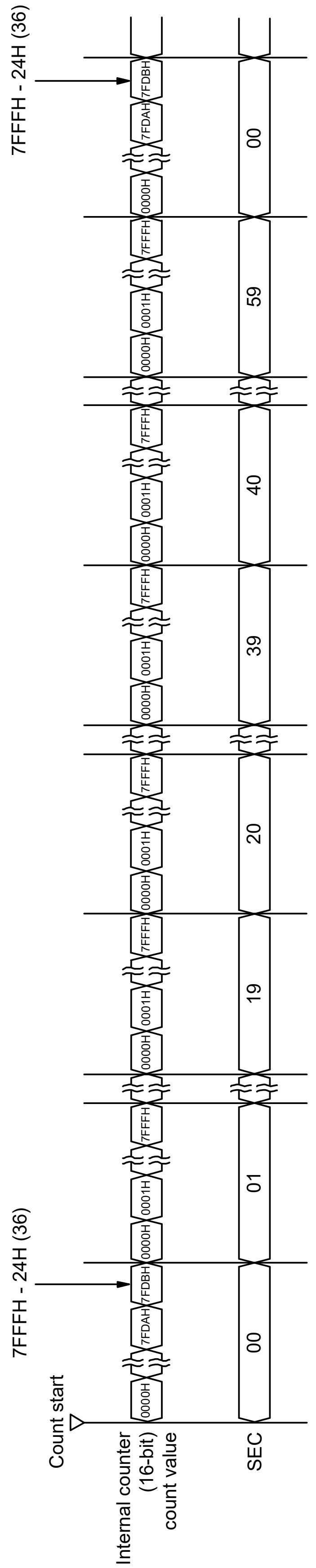
Calculate (F12, F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) from the correction value.

- {(/F11, /F10, /F9, /F8, /F7, /F6, /F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2= –36
(/F11, /F10, /F9, /F8, /F7, /F6, /F5, /F4, /F3, /F2, /F1, /F0)= 17
(/F11, /F10, /F9, /F8, /F7, /F6, /F5, /F4, /F3, /F2, /F1, /F0)= (0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1)
(/F11, /F10, /F9, /F8, /F7, /F6, F5, F4, F3, F2, F1, F0)= (1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0)

Consequently, when correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm), setting the correction register such that DEV is 1 and the correction value is –36 (bits 12 to 0 of the SUBCUD register: 1111111101110) results in 32768 Hz (0 ppm).

Figure 9 - 28 shows the Operation when (DEV, F12, F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) = (1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0).

Figure 9 - 28 Operation when (DEV, F12, F11, F10, F9, F8, F7, F6, F5, F4, F3, F2, F1, F0) = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0)



CHAPTER 10 INTERVAL TIMER

10.1 Functions of Interval Timer

An interrupt (INTIT) is generated at any previously specified time interval. It can be used for wakeup from STOP mode and triggering an A/D converter's SNOOZE mode.

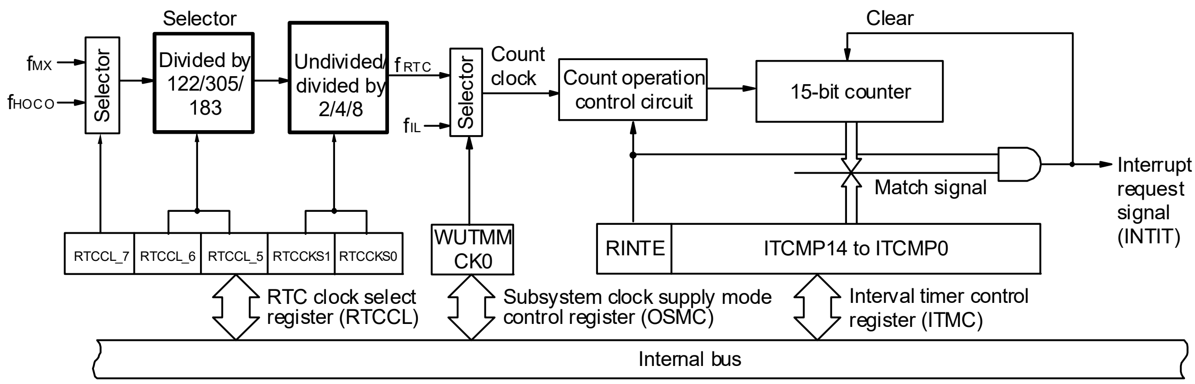
10.2 Configuration of Interval Timer

The interval timer includes the following hardware.

Table 10 - 1 Configuration of Interval Timer

Item	Configuration
Counter	15-bit counter
Control registers	Peripheral enable register 0 (PER0)
	Subsystem clock supply mode control register (OSMC)
	RTC clock select register (RTCCL)
	Interval timer control register (ITMC)

Figure 10 - 1 Block Diagram of Interval Timer



10.3 Registers Controlling Interval Timer

The interval timer is controlled by the following registers.

- Peripheral enable register 0 (PER0)
- Subsystem clock supply mode control register (OSMC)
- RTC clock select register (RTCCL)
- Interval timer control register (ITMC)

10.3.1 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 interval timer is used, be sure to set bit 7 (RTCEN) 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 10 - 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	RTCEN	0	ADCEN	0	0	SAU0EN	TAU1EN	TAU0EN

RTCEN	Control of real-time clock (RTC) and interval timer input clock supply
0	Stops input clock supply. • SFRs used by the real-time clock (RTC) and interval timer cannot be written. • The real-time clock (RTC) and interval timer are in the reset status.
1	Enables input clock supply. • SFRs used by the real-time clock (RTC) and interval timer can be read/written.

Caution 1. When using the interval timer, be sure to first set the RTCEN bit to 1 and then set the following register, while oscillation of the count clock is stable. If RTCEN = 0, writing to the control register controlling the interval timer is ignored, and, even if the register is read, only the default value is read (except the subsystem clock supply mode control register (OSMC) and RTC clock select register (RTCCL)).

• Interval timer control register (ITMC)

Caution 2. Be sure to clear bits 3, 4, and 6 to 0.

10.3.2 Subsystem clock supply mode control register (OSMC)

The OSMC register can be used to select the operating clock for the interval timer, real-time clock, and timer RJ by using the WUTMMCK0 bit.

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

Reset signal generation clears this register to 00H.

Figure 10 - 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	0	0	0	WUTMMCK0	0	0	0	0

WUTMMCK0	Selection of operation clock for real-time clock, interval timer, and timer RJ
0	The operating clock (f_{RTC}) specified in the RTC clock select register (RTCCL) • The RTC operating clock is selected as the count clock for the real-time clock and the interval timer. • The low-speed on-chip oscillator cannot be selected as the count source for timer RJ.
1	Low-speed on-chip oscillator clock (f_{IL}) • The low-speed on-chip oscillator clock is selected as the operating clock for the interval timer. • The low-speed on-chip oscillator can be selected as the count source for timer RJ.

Caution Be sure to set the WUTMMCK0 bit to 0 when using the real-time clock.

10.3.3 RTC clock select register (RTCCL)

This register is used to select the operating clock for the RTC and interval timer.
The RTCCL register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 10 - 4 Format of RTC clock select register (RTCCL)

Address: F02D8H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RTCCL	RTCCL7	RTCCL6	RTCCL5	0	0	0	RTCCKS1	RTCCKS0

RTCCL7	Selection of source of operating clock of RTC and interval timer
0	High-speed system clock (f_{MX})
1	High-speed on-chip oscillator clock (f_{HOCO})

RTCCL6	RTCCL5	Selection of frequency divisor for operating clock of RTC and interval timer 1
0	0	Divided by 122
0	1	Divided by 305
1	0	Divided by 183
1	1	Setting prohibited

RTCCKS1	RTCCKS0	Selection of frequency divisor for operating clock of RTC and interval timer 2
0	0	Undivided
0	1	Divided by 2
1	0	Divided by 4
1	1	Divided by 8

Remark The following shows the frequency division settings to generate a clock equivalent to 32.768 kHz. Select any of the following for operating the interval timer.

RTCCL7	RTCCL6	RTCCL5	RTCCKS1	RTCCKS0	Clock source	Interval timer operating clock (f_{RTC} = 32.787 kHz)
0	0	0	0	0	f_{MX} = 4 MHz	f_{MX}/122
0	0	0	0	1	f_{MX} = 8 MHz	f_{MX}/122/2
0	1	0	0	1	f_{MX} = 12 MHz	f_{MX}/183/2
0	0	0	1	0	f_{MX} = 16 MHz	f_{MX}/122/4
0	0	1	0	1	f_{MX} = 20 MHz	f_{MX}/305/2
1	0	0	0	0	f_{HOCO} = 4 MHz	f_{HOCO}/122
1	1	0	0	0	f_{HOCO} = 6 MHz	f_{HOCO}/183
1	0	0	0	1	f_{HOCO} = 8 MHz	f_{HOCO}/122/2
1	1	0	0	1	f_{HOCO} = 12 MHz	f_{HOCO}/183/2
1	0	0	1	0	f_{HOCO} = 16 MHz	f_{HOCO}/122/4
1	1	0	1	0	f_{HOCO} = 24 MHz	f_{HOCO}/183/4
1	0	0	1	1	f_{HOCO} = 32 MHz	f_{HOCO}/122/8
Other than above					Setting prohibited	

10.3.4 Interval timer control register (ITMC)

This register is used to set up the starting and stopping of the interval timer operation and to specify the timer compare value.
The ITMC register can be set by a 16-bit memory manipulation instruction.
Reset signal generation clears this register to 0FFFH.

Figure 10 - 5 Format of interval timer control register (ITMC)

Address: FFF90H After reset: 7FFFH R/W

Symbol	15	14 to 0
ITMC	RINTE	ITCMP14 to ITCMP0

RINTE	Interval timer operation control
0	Stop counter operation. (Clear the count.)
1	Start count operation.

ITCMP14 to ITCMP0	Specification of the interval timer compare value
0001H	Generate a fixed-cycle interrupt (count clock cycles x (ITCMP setting + 1)).
• • •	
7FFFH	
0000H	Setting prohibited

Example interrupt cycles when 0001H or 7FFFH is specified for ITCMP14 to ITCMP0
- When ITCMP14 to ITCMP0 = 0001H and count clock $f_{RTC} \cong$ 32.7689 kHz
  1/32.7689 [kHz] × (1 + 1) = 0.061 [ms] = 61 [μs]
- When ITCMP14 to ITCMP0 = 7FFFH and count clock $f_{RTC} \cong$ 32.7689 kHz
  1/32.7689 [kHz] × (32,767 + 1) = 999.424 [ms]

Caution 1. Before changing the RINTE bit from 1 to 0, use the interrupt mask flag register to disable the INTIT interrupt servicing. When the operation starts (from 0 to 1) again, clear the ITIF flag, and then enable the interrupt servicing.

Caution 2. The value read from the RINTE bit is applied one count clock cycle after setting the RINTE bit.

Caution 3. When setting the RINTE bit after returned from standby mode and entering standby mode again, confirm that the written value of the RINTE bit is reflected, or wait that more than one clock of the count clock has elapsed after returned from standby mode. Then enter standby mode.

Caution 4. Change the setting of the ITCMP14 to ITCMP0 bits only when RINTE = 0.
However, the settings of the ITCMP14 to ITCMP0 bits can be changed at the same time as changing RINTE from 0 to 1 or 1 to 0.

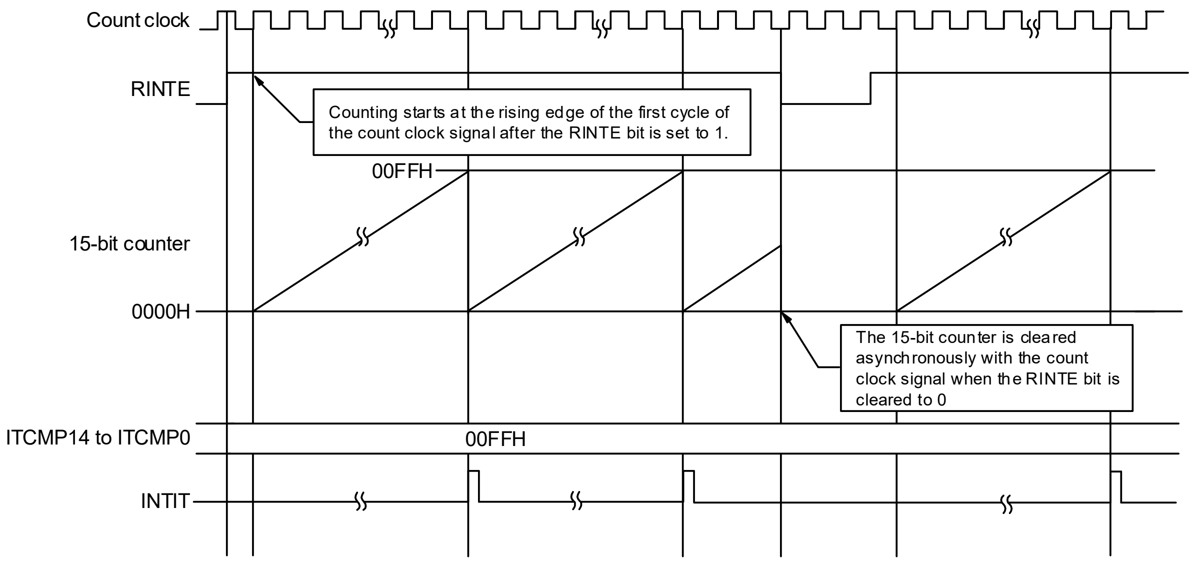
10.4 Interval Timer Operation

10.4.1 interval timer operation timing

The count value specified for the ITCMP14 to ITCMP0 bits is used as an interval to operate an interval timer that repeatedly generates interrupt requests (INTIT).
When the RINTE bit is set to 1, the 15-bit counter starts counting.
When the value of the 15-bit counter matches the value specified for the ITCMP14 to ITCMP0 bits, the 15-bit counter is cleared to 0, the 15-bit counter continues counting, and an interrupt request signal (INTIT) is generated at the same time.

The basic operation of the interval timer is as follows.

Figure 10 - 6 Interval Timer Operation Timing (ITCMP14 to ITCMP0 = 00FFH, count clock: f_{SUB} = 32.768 kHz)

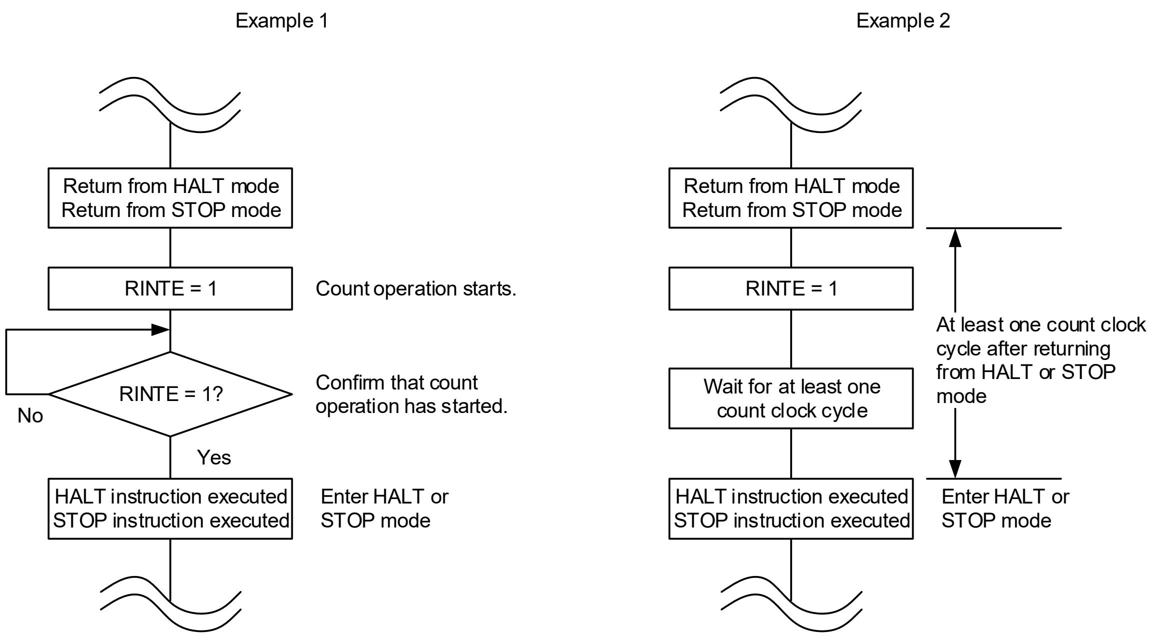


10.4.2 Start of count operation and re-enter to HALT/STOP mode after returned from HALT/STOP mode

When setting the RINTE bit after returned from HALT or STOP mode and entering HALT or STOP mode again, write 1 to the RINTE bit, and confirm the written value of the RINTE bit is reflected or wait for at least one cycle of the count clock.
Then, enter HALT or STOP mode.

- After setting RINTE to 1, confirm by polling that the RINTE bit has become 1, and then enter HALT or STOP mode (see **Example 1** in **Figure 10 - 7**).
- After setting RINTE to 1, wait for at least one count clock cycle and then enter HALT or STOP mode (see **Example 2** in **Figure 10 - 7**).

Figure 10 - 7 Procedure of entering to HALT or STOP mode after setting RINTE to 1



CHAPTER 11 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

11.1 Functions of Clock Output/Buzzer Output Controller

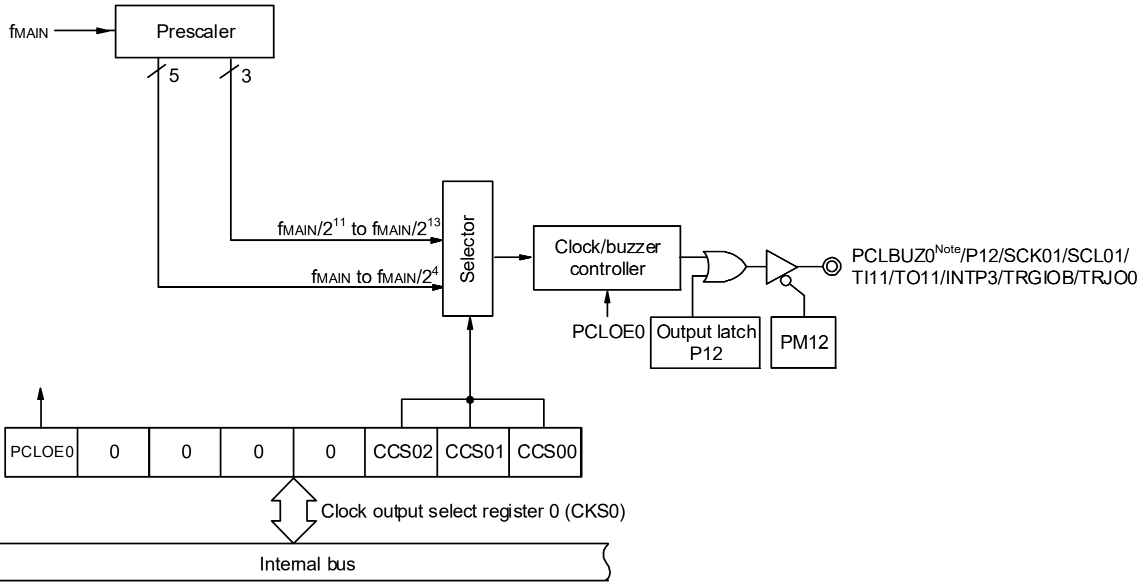
The clock output controller is intended for clock output for supply to peripheral ICs. Buzzer output is a function to output a square wave of buzzer frequency.

One pin can be used to output a clock or buzzer sound.

The PCLBUZ0 pin outputs a clock selected by clock output select register 0 (CKS0).

Figure 11 - 1 shows the Block Diagram of Clock Output/Buzzer Output Controller.

Figure 11 - 1 Block Diagram of Clock Output/Buzzer Output Controller



Note For the frequencies that can be output from the PCLBUZ0 pin, refer to **33.4** or **34.4 AC Characteristics**.

11.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

Table 11 - 1 Configuration of Clock Output/Buzzer Output Controller

Item	Configuration
Control registers	Clock output select register 0 (CKS0) Port mode register 1 (PM1) Port register 1 (P1)

11.3 Registers Controlling Clock Output/Buzzer Output Controller

11.3.1 Clock output select register 0 (CKS0)

This register specifies whether to enable or disable clock output or buzzer output from the PCLBUZ0 pin and specifies the output clock frequency.

The CKS0 register is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 11 - 2 Format of Clock output select register 0 (CKS0)

Address: FFFA5H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CKS0	PCLOE0	0	0	0	0	CCS02	CCS01	CCS00

PCLOE0	PCLBUZ0 pin output enable/disable
0	Output disable (default)
1	Output enable

CCS02	CCS01	CCS00	Selection of PCLBUZ0 pin output clock				
				f_{MAIN} = 5 MHz	f_{MAIN} = 10 MHz	f_{MAIN} = 20 MHz	f_{MAIN} = 32 MHz
0	0	0	f_{MAIN}	5 MHz	10 MHz Note	Setting prohibited Note	Setting prohibited Note
0	0	1	$f_{MAIN}/2$	2.5 MHz	5 MHz	10 MHz Note	16 MHz Note
0	1	0	$f_{MAIN}/2^2$	1.25 MHz	2.5 MHz	5 MHz	8 MHz
0	1	1	$f_{MAIN}/2^3$	625 kHz	1.25 MHz	2.5 MHz	4 MHz
1	0	0	$f_{MAIN}/2^4$	312.5 kHz	625 kHz	1.25 MHz	2 MHz
1	0	1	$f_{MAIN}/2^{11}$	2.44 kHz	4.88 kHz	9.77 kHz	15.63 kHz
1	1	0	$f_{MAIN}/2^{12}$	1.22 kHz	2.44 kHz	4.88 kHz	7.81 kHz
1	1	1	$f_{MAIN}/2^{13}$	610 Hz	1.22 kHz	2.44 kHz	3.91 kHz

Note Use the output clock within a range of 16 MHz. See **33.4** or **34.4 AC Characteristics** for details.

Caution 1. Change the output clock after disabling clock output (PCLOE0 = 0).
Caution 2. To shift to STOP mode, set PCLOE0 to 0 before executing the STOP instruction.

Remark f_{MAIN}: Main system clock frequency

11.3.2 Registers controlling port functions of pins to be used for clock or buzzer output

When using a port pin as a clock or buzzer output pin, set port mode register 1 (PM1) and port register 1 (P1) that control the port functions multiplexed on that pin. For details, see **4.3.1 Port mode registers (PMxx)** and **4.3.2 Port registers (Pxx)**.

When using a port pin as a clock or buzzer output function (P12/SCK01/SCL01/TI11/TO11/INTP3/PCLBUZ0/TRGIOB/TRJO0), set the P12 bit in port mode register 1 (PM1) and port register 1 (P1), which corresponds to P12, to 0.

Example: When using P12/SCK01/SCL01/TI11/TO11/INTP3/PCLBUZ0/TRGIOB/TRJO0 for clock or buzzer output
Set the PM12 bit of port mode register 1 to 0.
Set the P12 bit of port register 1 to 0.

11.4 Operations of Clock Output/Buzzer Output Controller

One pin can be used to output a clock or buzzer sound.
The PCLBUZ0 pin outputs a clock/buzzer selected by clock output select register 0 (CKS0).

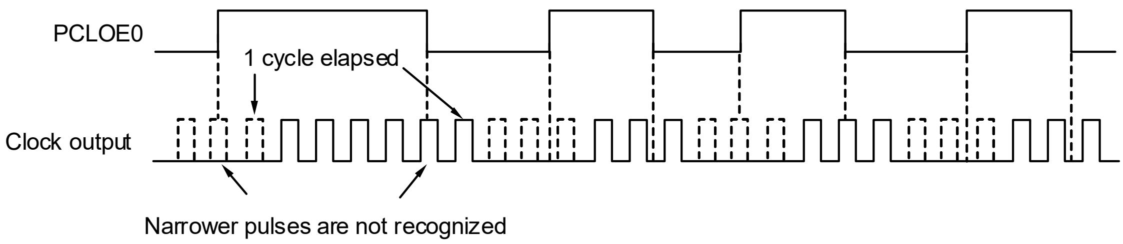
11.4.1 Operation as output pin

To enable output from the PCLBUZ0 pin, perform the following procedure.

<1> Write 0 to the bits in port mode register 1 (PM1) and port register 1 (P1) that correspond to the port pin to be used as the PCLBUZ0 pin.
<2> Select the output frequency by using bits 0 to 3 (CCS00 to CCS02) of clock output select register 0 (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 outputting the clock starts or stops outputting the clock one clock after enabling or disabling clock output (by using the PCLOE0 bit) is switched. At this time, pulses with a narrow width are not output. Figure 11 - 3 shows enabling or stopping output using the PCLOE0 bit and the timing of outputting the clock.

Figure 11 - 3 Timing of Outputting Clock from PCLBUZ0 Pin



11.5 Cautions on clock output/buzzer output controller

If STOP mode is entered within 1.5 clock cycles output from the PCLBUZ0 pin after the output is disabled (PCLOE0 = 0), the PCLBUZ0 output width becomes shorter.

CHAPTER 12 WATCHDOG TIMER

12.1 Functions of Watchdog Timer

The counting operation of the watchdog timer is set by the option byte (000C0H).
The watchdog timer operates on the low-speed on-chip oscillator clock (f_{IL}).
The watchdog timer is used to detect an inadvertent program loop. If a program loop is detected, an internal reset signal is generated.
Program loop is detected in the following cases.

- If the watchdog timer counter overflows
- If a 1-bit manipulation instruction is executed on the watchdog timer enable register (WDTE)
- If data other than "ACH" is written to the WDTE register
- If data is written to the WDTE register during a window close period

When a reset occurs due to the watchdog timer, bit 4 (WDTRF) of the reset control flag register (RESF) is set to 1.
For details of the RESF register, see **CHAPTER 24 RESET FUNCTION**.
When 75% + 1/2 f_{IL} of the overflow time is reached, an interval interrupt can be generated.

12.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 12 - 1 Configuration of Watchdog Timer

Item	Configuration
Counter	Internal counter (17 bits)
Control register	Watchdog timer enable register (WDTE)

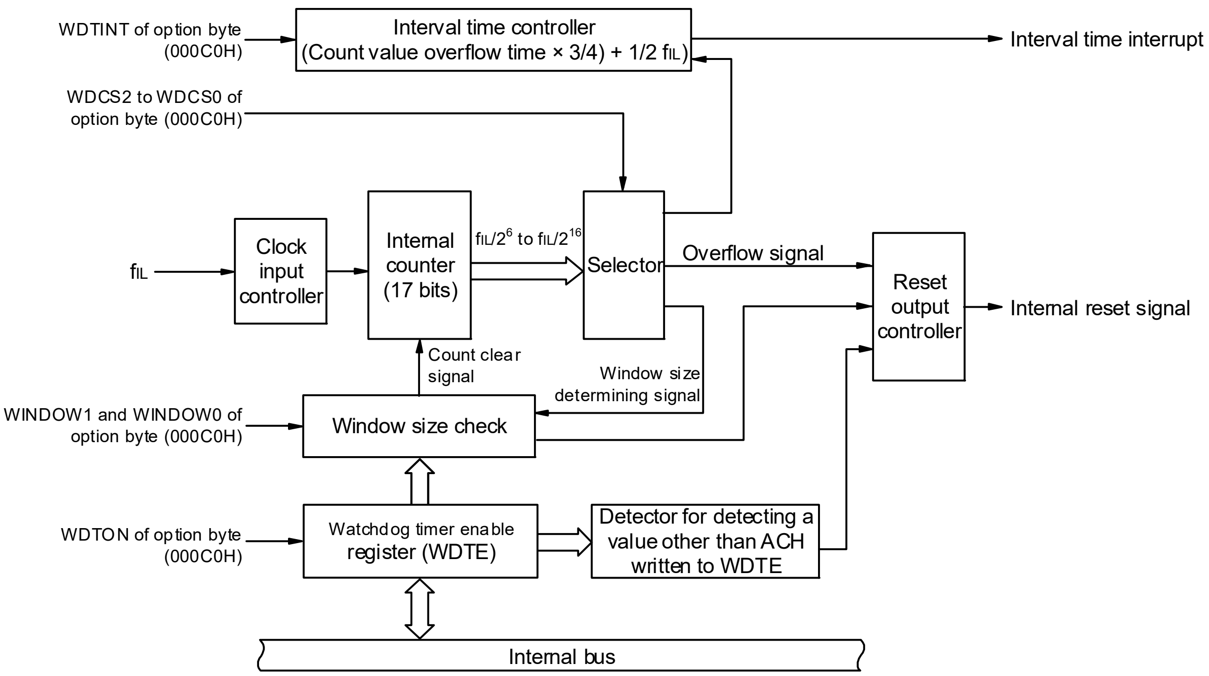
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by using the option byte.

Table 12 - 2 Setting of Option Bytes and Watchdog Timer

Setting of Watchdog Timer	Option Byte (000C0H)
Watchdog timer interval interrupt	Bit 7 (WDTINT)
Window open period	Bits 6 and 5 (WINDOW1, WINDOW0)
Controlling counter operation of watchdog timer	Bit 4 (WDTON)
Overflow time of watchdog timer	Bits 3 to 1 (WDCS2 to WDCS0)
Controlling counter operation of watchdog timer (in HALT/STOP mode)	Bit 0 (WDSTBYON)

Remark For the option byte, see **CHAPTER 29 OPTION BYTE**.

Figure 12 - 1 Block Diagram of Watchdog Timer



Remark fIL: Low-speed on-chip oscillator clock

12.3 Register Controlling Watchdog Timer

The watchdog timer is controlled by using the watchdog timer enable register (WDTE).

12.3.1 Watchdog timer enable register (WDTE)

Writing “ACH” to the WDTE register clears the watchdog timer counter and the watchdog timer starts counting again.

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

Reset signal generation sets this register to 9AH or 1AH Note.

Figure 12 - 2 Format of Watchdog timer enable register (WDTE)

Address: FFFABH After reset: 9AH/1AH Note R/W

Symbol	7	6	5	4	3	2	1	0
WDTE								

Note The WDTE register reset value differs depending on the value of the WDTON bit of the option byte (000C0H). To start the watchdog timer operating, set the WDTON bit to 1.

WDTON Bit Setting Value	WDTE Register Reset Value
0 (watchdog timer count operation disabled)	1AH
1 (watchdog timer count operation enabled)	9AH

Caution 1. If a value other than “ACH” is written to the WDTE register, an internal reset signal is generated.

Caution 2. If a 1-bit memory manipulation instruction is executed for the WDTE register, an internal reset signal is generated.

Caution 3. The value read from the WDTE register is 9AH or 1AH, which differs from the written value (ACH).

12.4 Operation of Watchdog Timer

12.4.1 Controlling operation of watchdog timer

1. When using the watchdog timer, specify the settings below by using by the option byte (000C0H).

- Enable counting operation of the watchdog timer by setting bit 4 (WDTON) of the option byte (000C0H) to 1 (the counter starts operating after a reset release) (for details, see **CHAPTER 29**).

WDTON	Watchdog Timer Counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

- Set an overflow time by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H) (for details, see **12.4.2** and **CHAPTER 29**).
- Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (000C0H) (for details, see **12.4.3** and **CHAPTER 29**).

2. After a reset release, the watchdog timer starts counting.
3. By writing "ACH" to the watchdog timer enable register (WDTE) after the watchdog timer starts counting and before the overflow time set by the option byte, the watchdog timer is cleared and starts counting again.
4. After that, write the WDTE register the second time or later after a reset release during the window open period. If the WDTE register is written during a window close period, an internal reset signal is generated.
5. If the overflow time expires without "ACH" being 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.When data is written to the watchdog timer enable register (WDTE) for the first time after reset release, the watchdog timer is cleared at any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

Caution 2. After "ACH" is written to the WDTE register, an error of up to 2 f_{IL} cycles may occur before the watchdog timer is cleared.

Caution 3. The watchdog timer can be cleared immediately before the count value overflows.

Caution 4. The operation of the watchdog timer in the HALT and STOP modes differs as follows depending on the value set to bit 0 (WDSTBYON) of the option byte (000C0H).

	WDSTBYON = 0	WDSTBYON = 1
In HALT mode	The watchdog timer stops operating.	The watchdog timer continues operating.
In STOP mode		
In SNOOZE mode		

If WDSTBYON = 0, the watchdog timer resumes counting after the HALT or STOP mode is exited. At this time, the counter is cleared to 0 and counting starts.
When operating with the X1 oscillation clock after exiting 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 too short, an overflow occurs during the oscillation stabilization time, causing a reset.
Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock and when the watchdog timer is to be cleared after the STOP mode is exited by an interval interrupt.

12.4.2 Setting overflow time of watchdog timer

Set the overflow time of the watchdog timer by using bits 3 to 1 (WDCS2 to WDCS0) of the option byte (000C0H).
If an overflow occurs, an internal reset signal is generated. The present count is cleared and the watchdog timer starts counting again by writing “ACH” to the watchdog timer enable register (WDTE) during the window open period before the overflow time.
The following overflow times can be set.

Table 12 - 3 Setting of Overflow Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer (f_{IL} = 17.25 kHz (MAX.))
0	0	0	$2^6/f_{IL}$ (3.71 ms)
0	0	1	$2^7/f_{IL}$ (7.42 ms)
0	1	0	$2^8/f_{IL}$ (14.84 ms)
0	1	1	$2^9/f_{IL}$ (29.68 ms)
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)
1	0	1	$2^{13}/f_{IL}$ (474.89 ms)
1	1	0	$2^{14}/f_{IL}$ (949.79 ms)
1	1	1	$2^{16}/f_{IL}$ (3799.18 ms)

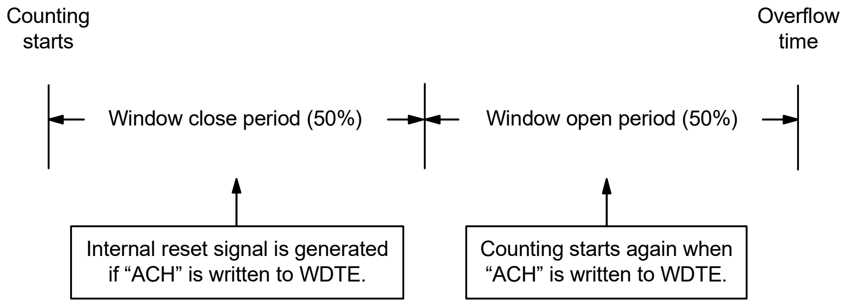
Remark f_{IL}: Low-speed on-chip oscillator clock frequency

12.4.3 Setting window open period of watchdog timer

Set the window open period of the watchdog timer by using bits 6 and 5 (WINDOW1, WINDOW0) of the option byte (000C0H). The outline of the window is as follows.

- If “ACH” is written to the watchdog timer enable register (WDTE) during the window open period, the watchdog timer is cleared and starts counting again.
- Even if “ACH” is written to the WDTE register during the window close period, an abnormality is detected and an internal reset signal is generated.

Example: If the window open period is 50%



Caution When data is written to the WDTE register for the first time after reset release, the watchdog timer is cleared at any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

The following window open period can be set.

Table 12 - 4 Setting Window Open Period of Watchdog Timer

WINDOW1	WINDOW0	Window Open Period of Watchdog Timer
0	0	Setting prohibited
0	1	50%
1	0	75%
1	1	100%

<R> **Note** When the window open period is set to 75%, clearing the counter of the watchdog timer (writing ACH to WDTE) must proceed outside the corresponding period from among those listed below, over which clearing of the counter is prohibited (for example, confirming that the interval timer interrupt request flag (WDTIIF) of the watchdog timer is set).

<R>

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time (f_{IL} = 17.25 kHz (MAX.))	Period over which clearing the counter is prohibited when the window open period is set to 75%
0	0	0	$2^6/f_{IL}$ (3.71 ms)	1.85 ms to 2.51 ms
0	0	1	$2^7/f_{IL}$ (7.42 ms)	3.71 ms to 5.02 ms
0	1	0	$2^8/f_{IL}$ (14.84 ms)	7.42 ms to 10.04 ms
0	1	1	$2^9/f_{IL}$ (29.68 ms)	14.84 ms to 20.08 ms
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)	56.36 ms to 80.32 ms
1	0	1	$2^{13}/f_{IL}$ (474.89 ms)	237.44 ms to 321.26 ms
1	1	0	$2^{14}/f_{IL}$ (949.79 ms)	474.89 ms to 642.51 ms
1	1	1	$2^{16}/f_{IL}$ (3799.18 ms)	1899.59 ms to 2570.04 ms

Caution When bit 0 (WDSTBYON) of the option byte (000C0H) is 0, the window open period is 100% regardless of the values of the WINDOW1 and WINDOW0 bits.

Remark If the overflow time is set to $2^9/f_{IL}$, the window close time and open time are as follows.

	Setting of Window Open Period		
	50%	75%	100%
Window close time	0 to 20.08 ms	0 to 10.04 ms	None
Window open time	20.08 to 29.68 ms	10.04 to 29.68 ms	0 to 29.68 ms

<When window open period is 50%>

- Overflow time:
  $2^9/f_{IL}$ (MAX.) = 2^9/17.25 kHz (MAX.) = 29.68 ms
- Window close time:
  0 to $2^9/f_{IL}$ (MIN.) × (1 - 0.5) = 0 to 2^9/12.75 kHz × 0.5 = 0 to 20.08 ms
- Window open time:
  $2^9/f_{IL}$ (MIN.) × (1 - 0.5) to $2^9/f_{IL}$ (MAX.) = 2^9/12.75 kHz × 0.5 to 2^9/17.25 kHz = 20.08 to 29.68 ms

12.4.4 Setting watchdog timer interval interrupt

Setting bit 7 (WDTINT) of an option byte (000C0H) can generate an interval interrupt (INTWDTI) when 75% + 1/2 f_{IL} of the overflow time is reached.

Table 12 - 5 Setting of Watchdog Timer Interval Interrupt

WDTINT	Use of Watchdog Timer Interval Interrupt
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% + 1/2 f_{IL} of overflow time is reached.

Caution When operating with the X1 oscillation clock after exiting the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed.
Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset.
Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock and when the watchdog timer is to be cleared after the STOP mode is exited by an interval interrupt.

Remark The watchdog timer continues counting even after INTWDTI is generated (until ACH is written to the watchdog timer enable register (WDTE)). If ACH is not written to the WDTE register before the overflow time, an internal reset signal is generated.

CHAPTER 13 ANALOG FRONT-END POWER SUPPLY CIRCUIT

13.1 Functions of Analog Front-End Power Supply Circuit

The analog front-end (AFE) power supply circuit consists of an AFE reference power supply circuit (ABGR), LDO for supplying power to internal circuits (REGA), and LDO for supplying power to a sensor (external device) (SBIAS).

Table 13 - 1 Combination of Functions and Circuits

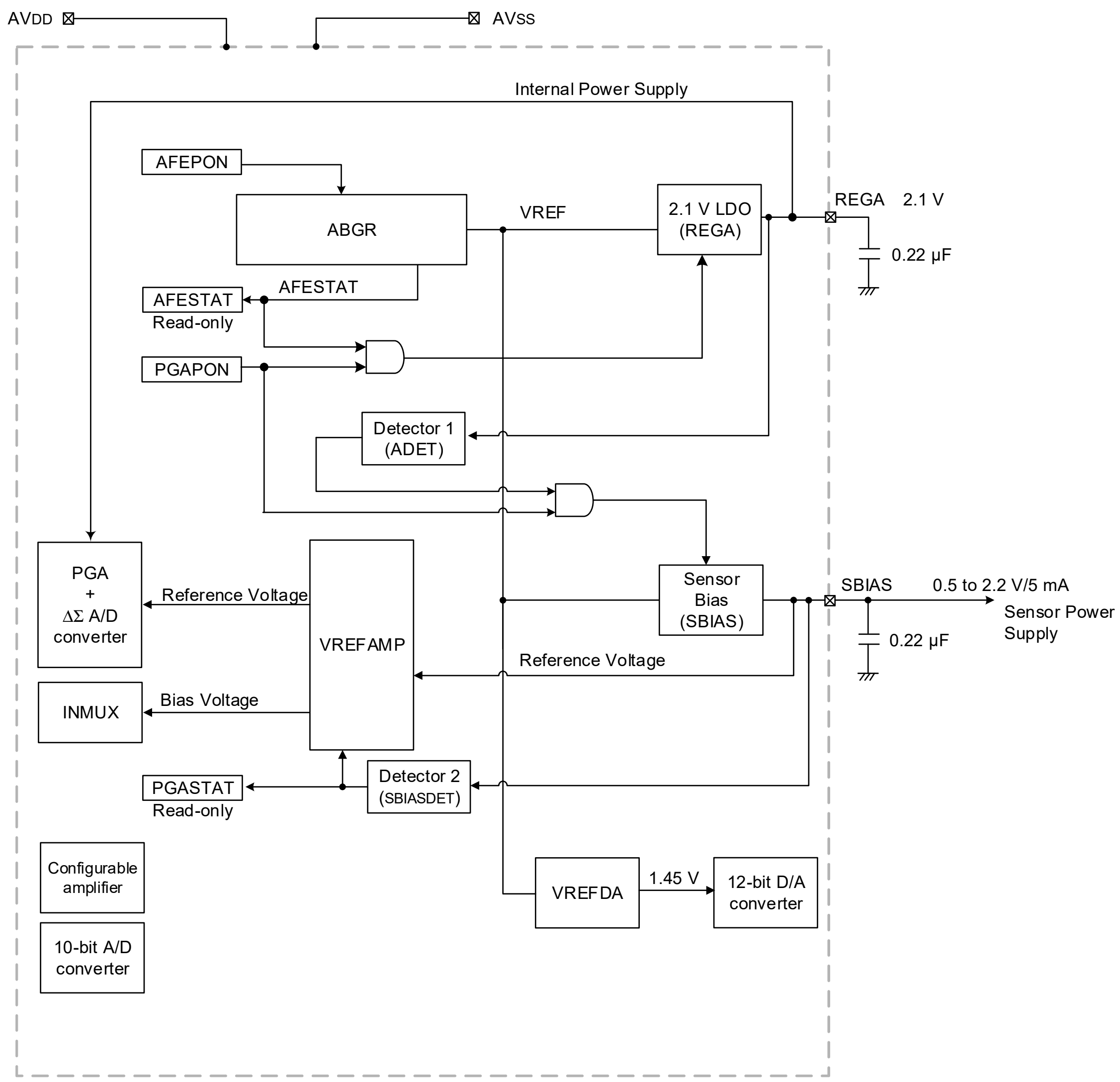
Combination of Functions				Combination of Analog Circuit				
PGA + ΔΣ A/D converter	Configurable amplifier n	12-bit D/A converter	10-bit A/D converter	ABGR	REGA/SBIAS/ VREFAMP/ PGA/ ΔΣ A/D converter	Configurable amplifier	12-bit D/A converter	10-bit A/D converter
			Run					√
	Run			√		√		
	Run		Run	√		√		√
	Run	Run		√		√	√	
	Run	Run	Run	√		√	√	√
Run				√	√			
Run			Run	√	√			√
Run	Run			√	√	√		
Run	Run		Run	√	√	√		√
Run	Run	Run		√	√	√	√	
Run	Run	Run	Run	√	√	√	√	√
AFE power control				AFEPON = 1	AFEPON = 1 & PGAPON = 1	AFEPON = 1 & AMPnPON = 1	AFEPON = 1 & DACPON = 1	N/A

The operating mode can be selected by setting a PON signal that controls the power to the corresponding function and the AFEPON signal that controls the reference power supply for the entire AFE by using a control register.
All the PON signals are initialized when the AFEEN bit of peripheral enable register 1 (PER1) is cleared to 0, which stops the power supplied to all AFE circuits.
You can check the power status of ABGR by reading the AFESTAT bit of the analog front-end power supply detection register (AFEPWD).
You can check the power status of REGA and SBIAS by reading the PGASTAT bit of the AFEPWD register.

13.2 Configuration of Analog Front-End Power Supply Circuit

Figure 13 - 1 shows the block diagram of the analog front-end power supply circuit.

Figure 13 - 1 Block Diagram of Analog Front-End Power Supply Circuit



13.3 Registers Controlling the Analog Front-End Power Supply Circuit

The following registers are used to control the analog front-end power supply circuit.

- Peripheral enable register 1 (PER1)
- Analog front-end power supply selection register (AFEPWS)
- Analog front-end power supply detection register (AFEPWD)
- Sensor reference voltage setting register (VSBIAS)

13.3.1 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.

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 13 - 2 Format of Peripheral Enable Register 1 (PER1)

Address: F007AH After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	<3>	<2>	<1>	<0>
PER1	DACEN	TRGEN	AMPEN	0	DTCEN	PGAEN	AFEEN	TRJ0EN

AFEEN	Control of input clock supplied to AFE power supply/clock control block
0	Stops input clock supply. • SFRs used by the AFE power supply/clock control block cannot be written. • The AFE power supply/clock control block is in the reset status.
1	Enables input clock supply. • SFRs used by the AFE power supply/clock control block can be read and written.

Caution Be sure to clear bit 4 to “0”.

13.3.2 Analog front-end power supply selection register (AFEPWS)

The operating mode can be selected by setting a PON signal that controls the power to the corresponding function and the AFEPON signal that controls the reference power supply for the entire AFE by using the AFEPWS register.
The AFEPWS register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 13 - 3 Format of Analog Front-End Power Supply Selection Register (AFEPWS)

Address: F0440H After reset:00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	<0>
AFEPWS	DACPON	AMP2PON	AMP1PON	AMP0PON	0	PGAPON	0	AFEPON

DACPON	Control of power supplied to 12-bit D/A converter block
0	Power-off (default)
1	Power-on

AMP2PON	Control of power supplied to configurable amplifier 2 (AMP2) block
0	Power-off (default)
1	Power-on

AMP1PON	Control of power supplied to configurable amplifier 1 (AMP1) block
0	Power-off (default)
1	Power-on

AMP0PON	Control of power supplied to configurable amplifier 0 (AMP0) block
0	Power-off (default)
1	Power-on

PGAPON	Control of power supplied to programmable gain instrumentation amplifier (PGA) block
0	Power-off (default)
1	Power-on

AFEPON	Control of power supplied to AFE reference power supply (ABGR) block
0	Power-off (default)
1	Power-on

Caution Be sure to clear bits 1 and 3 to “0”.

13.3.3 Analog front-end power supply detection register (AFEPWD)

You can check the power status of the AFE reference power supply (ABGR) by reading the AFESTAT bit of the analog front-end power supply detection register (AFEPWD).
You can check the power status of REGA and SBIAS by reading the PGASTAT bit of the AFEPWD register.
The AFEPWD register can be read by a 1-bit or an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 13 - 4 Format of Analog Front-End Power Supply Detection Register (AFEPWD)

Address: F0441H After reset: 00H R

Symbol	7	6	5	4	3	<2>	1	<0>
AFEPWD	0	0	0	0	0	PGASTAT	0	AFESTAT

PGASTAT	Status of power supplied to programmable gain instrumentation amplifier (PGA) block
0	Off or stabilizing
1	Stabilized

AFESTAT	Status of power supplied to AFE reference power supply (ABGR) block
0	Off or stabilizing
1	Stabilized

13.3.4 Sensor reference voltage setting register (VSBIAS)

This register is used to specify the SBIAS output voltage.

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

Reset signal generation sets this register to 10H.

Figure 13 - 5 Format of Sensor Reference Voltage Setting Register (VSBIAS)

Address: F0443H After reset: 10H R/W

Symbol	7	6	5	4	3	2	1	0
VSBIAS	0	0	0	VSBIAS4	VSBIAS3	VSBIAS2	VSBIAS1	VSBIAS0

VSBIAS4	VSBIAS3	VSBIAS2	VSBIAS1	VSBIAS0	SBIAS output voltage (V)
0	1	0	0	1	0.5
0	1	0	1	0	0.6
0	1	0	1	1	0.7
0	1	1	0	0	0.8
0	1	1	0	1	0.9
0	1	1	1	0	1.0
0	1	1	1	1	1.1
1	0	0	0	0	1.2 (default)
1	0	0	0	1	1.3
1	0	0	1	0	1.4
1	0	0	1	1	1.5
1	0	1	0	0	1.6
1	0	1	0	1	1.7
1	0	1	1	0	1.8
1	0	1	1	1	1.9
1	1	0	0	0	2.0
1	1	0	0	1	2.1
1	1	0	1	0	2.2
Other than above					Setting prohibited

13.4 AFE Internal Reference Voltage Generator

13.4.1 Overview of AFE internal reference voltage generator

The AFE internal reference voltage generator consists of an AFE reference power supply circuit (ABGR) and an analog circuit reference voltage generator (VREFAMP).

The VREF reference voltage output from ABGR is supplied to the REGA, SBIAS, and VREFDA circuits.

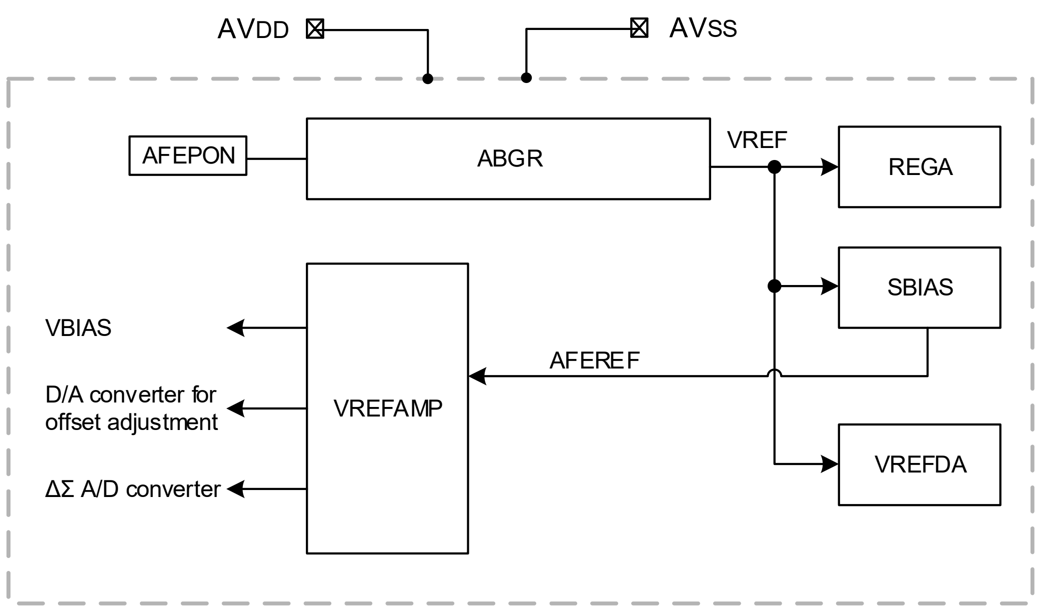
The VREF reference voltage output from ABGR passes through the SBIAS circuit and is output from the VREFAMP circuit, and is then used as a reference voltage in the D/A converter for offset voltage adjustment, and as an internal bias voltage (VBIAS) to be connected to input multiplexers.

ABGR can achieve high precision in the output voltage because it is less dependent on the temperature.

13.4.2 Configuration of AFE internal reference voltage generator

Figure 13 - 6 shows the block diagram of the AFE internal reference voltage generator.

Figure 13 - 6 Block Diagram of AFE Internal Reference Voltage Generator



13.4.3 Operation of AFE internal reference voltage generator

The AFEPON bit of the AFEPWS register controls whether to turn ABGR on and off. It is recommended to power off ABGR (by clearing the AFEPON bit to 0) after the 24-bit ΔΣ A/D converter, configurable amplifier, and 12-bit D/A converter turn off.

13.5 Sensor Power Supply (SBIAS)

13.5.1 Overview of sensor power supply (SBIAS)

SBIAS supplies power to the sensor connected to the RL78/I1E. The VREF reference voltage output from ABGR is input. SBIAS outputs a voltage between 0.5 and 2.2 V, which can be specified in units of 0.1 V. SBIAS outputs up to 5 mA of current. An external capacitor of 0.22 μF (recommended) must be connected to the SBIAS pin.

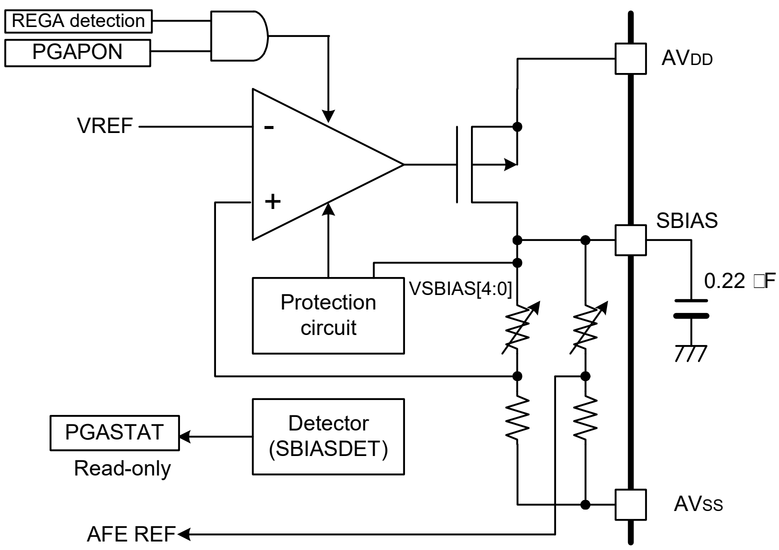
SBIAS has a protection circuit against an overcurrent (a current exceeding the rated upper limit). When an overcurrent occurs, the protection circuit works to protect the internal circuits. SBIAS also has a circuit (SBIASDET) that monitors and detects the voltage output by SBIAS.

The VREF reference voltage output from ABGR passes through the SBIAS circuit and is then used as a reference voltage in the D/A converter for offset voltage adjustment and ΔΣ A/D converter, and as an internal bias voltage (VBIAS) to be connected to input multiplexers.

13.5.2 Configuration of sensor power supply (SBIAS)

Figure 13 - 7 shows the block diagram of the sensor power supply (SBIAS).

Figure 13 - 7 Block Diagram of Sensor Power Supply (SBIAS)



13.5.3 Operation of sensor power supply (SBIAS)

In addition to supplying power to the sensor connected to the RL78/I1E, SBIAS is involved in generating reference voltages used in the ΔΣ A/D converter and the D/A converter for offset voltage adjustment, and an internal bias voltage (VBIAS) to be connected to input multiplexers.

SBIAS has SBIASDET, a circuit that monitors and detects the voltage output by SBIAS, and is used to start analog circuits such as VREFAMP, programmable gain instrumentation amplifier (PGA), and ΔΣ A/D converter. When SBIASDET detects the SBIAS output voltage, starting the analog circuits is enabled. When SBIASDET detects that the SBIAS output voltage has not risen normally, analog circuits stop operating.

When "0" is written to the AFEPON bit of the analog front-end power supply selection register (AFEPWS), SBIASDET detects the SBIAS output voltage and analog circuits such as VREFAMP, PGA, ΔΣ A/D converter, and SBIAS stop operating. When VREFAMP stops operating, a reference voltage in the ΔΣ A/D converter and D/A converter for offset voltage adjustment and an internal bias voltage (VBIAS) to be connected to input multiplexers are not generated.

13.6 Internal Power Supply Circuit for PGA and ΔΣ A/D Converter (REGA)

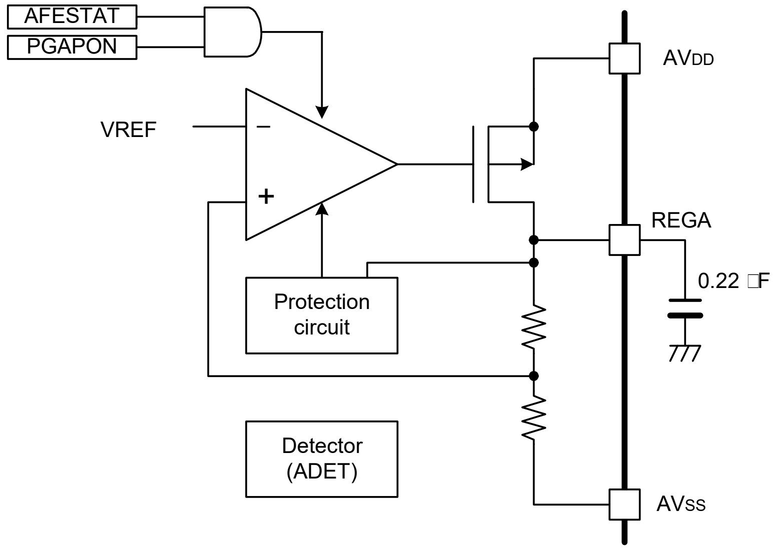
13.6.1 Overview of internal power supply circuit (REGA)

REGA generates a voltage based on the voltage output from the ABGR, and supplies power to the PGA and ΔΣ A/D converter. REGA outputs a voltage of 2.1 V (Typ.). An external capacitor of 0.22 μF (recommended) must be connected to the REGA output pin.
REGA has a protection circuit against an overcurrent and a low voltage detector (ADET).

13.6.2 Configuration of internal power supply circuit (REGA)

Figure 13 - 8 shows the block diagram of the internal power supply circuit (REGA).

Figure 13 - 8 Block Diagram of Internal Power Supply Circuit (REGA)



13.7 Reference Voltage Generator for 12-bit D/A Converter (VREFDA)

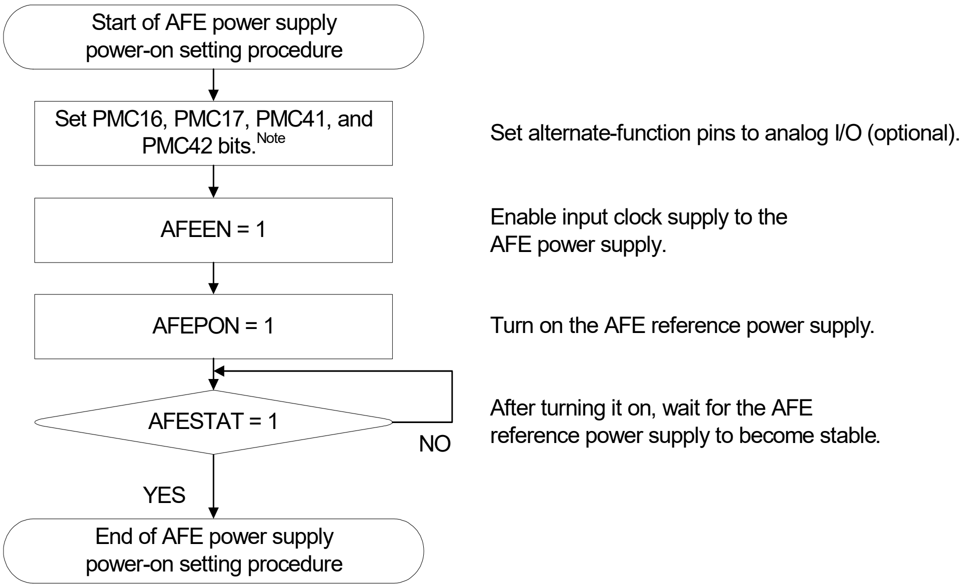
13.7.1 Overview of reference voltage generator for 12-bit D/A converter (VREFDA)

VREFDA generates an internal reference voltage and supplies 1.45 V to the 12-bit D/A converter.
VREFDA generates a voltage based on the voltage output from ABGR.

13.8 Procedure for Controlling Analog Front-End Power Supply Circuit

Figure 13 - 9 and Figure 13 - 10 show the flowcharts for powering on/off the analog front-end power supply. Figure 13 - 11 shows the timing diagram for the power-on sequence.

Figure 13 - 9 Flowchart for Powering on the Analog Front-End (AFE) Power Supply



Note When a reset signal is generated, PMC16, PMC17, PMC41, and PMC42 are set to 1.

Figure 13 - 10 Flowchart for Powering off the Analog Front-End (AFE) Power Supply

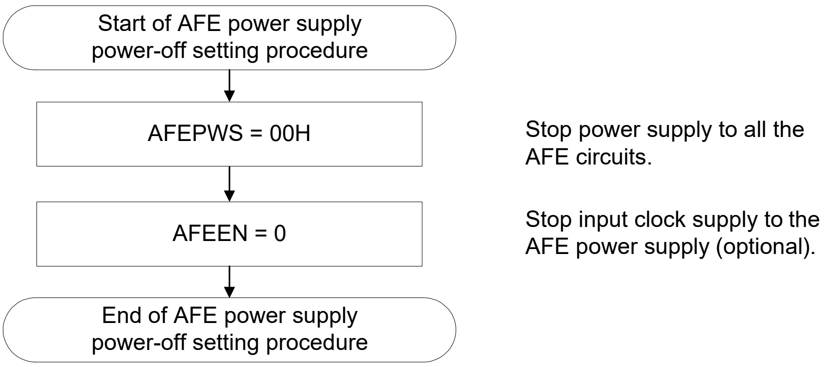
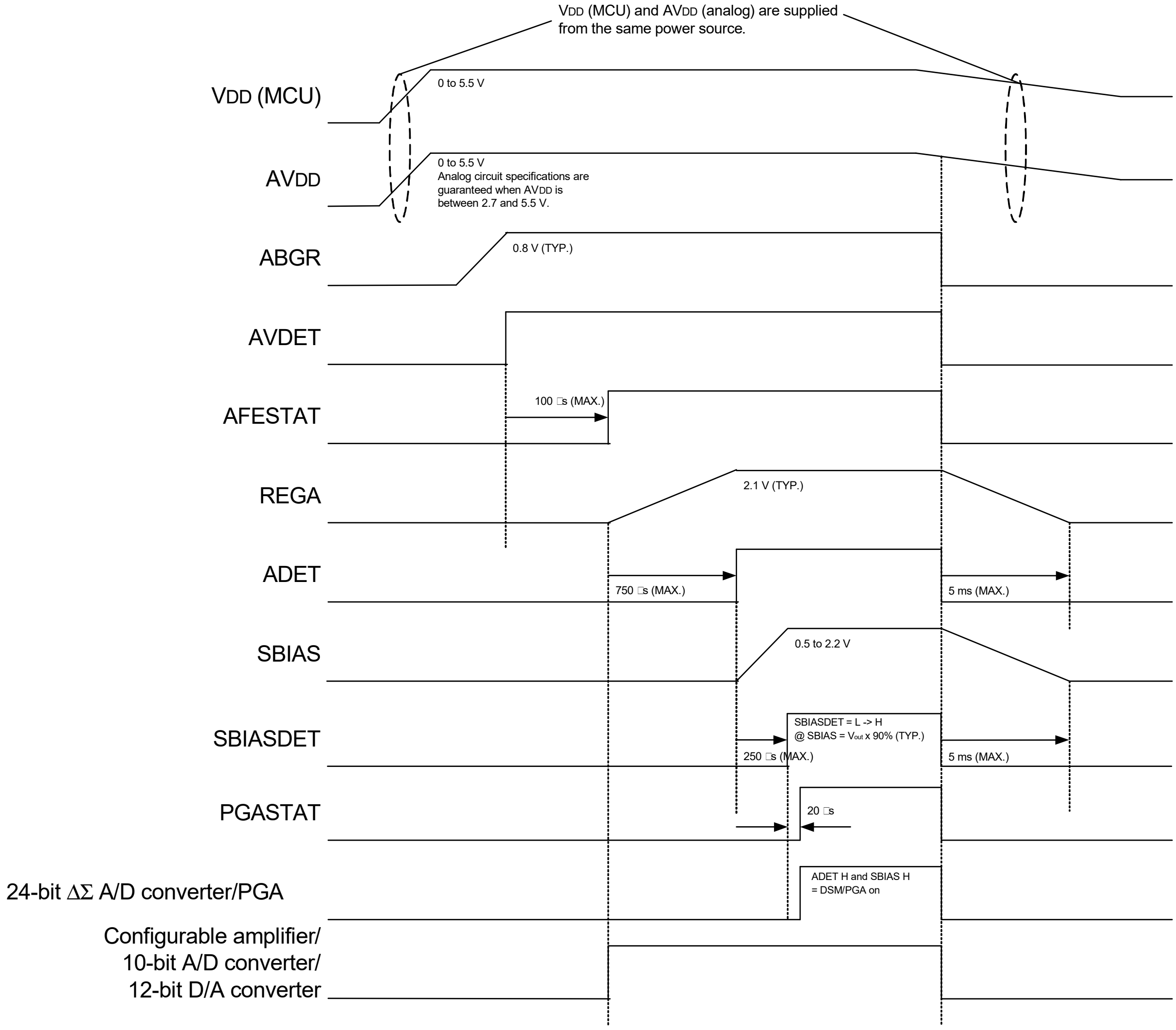


Figure 13 - 11 Timing for Power Supply Startup Sequence



CHAPTER 14 24-BIT ΔΣ A/D CONVERTER WITH PROGRAMMABLE GAIN INSTRUMENTATION AMPLIFIER

14.1 Functions of 24-bit ΔΣ A/D Converter with Programmable Gain Instrumentation Amplifier

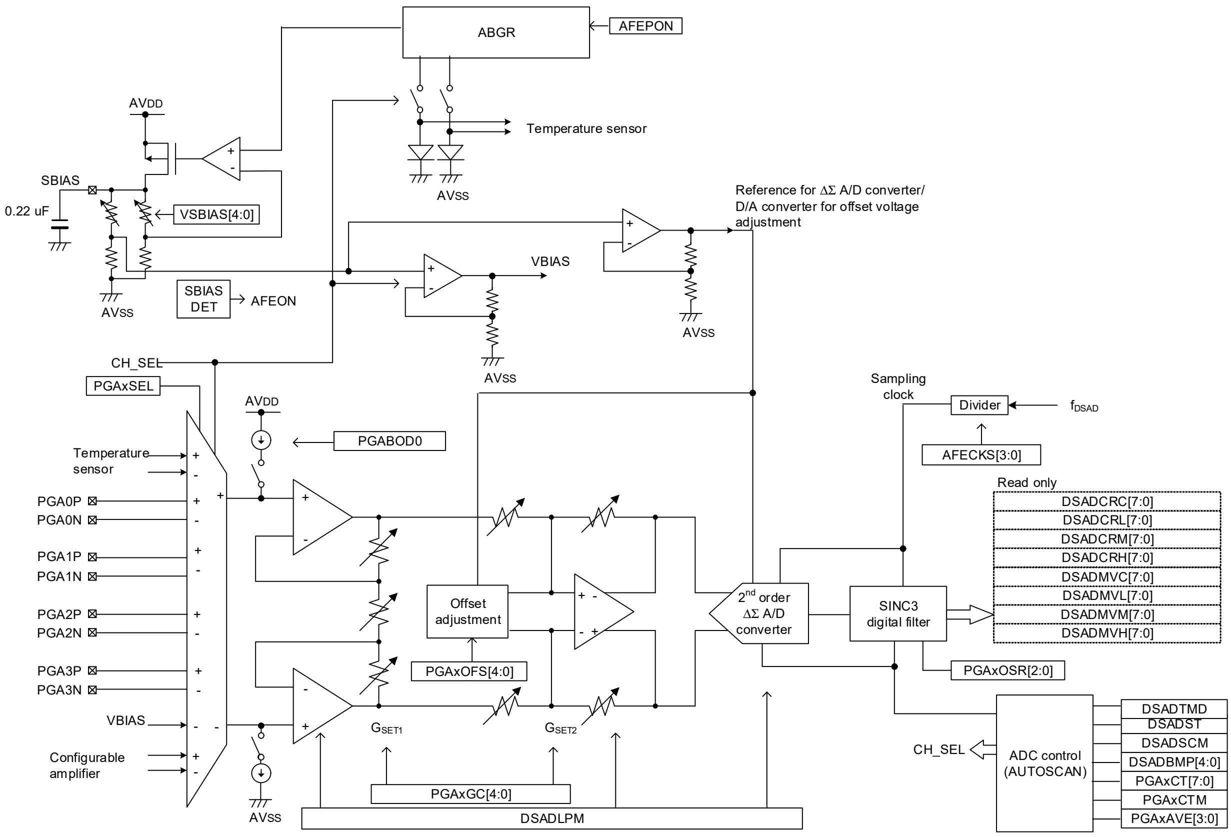
The RL78/I1E incorporates a 24-bit ΔΣ A/D converter with programmable gain instrumentation amplifier. The signal from an input multiplexer (there are 5 channels in total) is input to the 24-bit ΔΣ A/D converter via the programmable gain instrumentation amplifier (PGA). The A/D conversion result is filtered by the SINC3 digital filter, and is then stored in an output register.

A/D conversion is performed based on the clock generated in the high-speed on-chip oscillator (HOCO) (sampling frequency = 1 MHz (TYP.)). The high-speed system clock, and the PLL clock generated based on the high-speed system clock can also be used. A/D conversion is performed based on a built-in sequencer called AUTOSCAN. The data rate (frequency with which each A/D conversion result is output) can be specified for each channel.

14.2 Configuration of 24-bit ΔΣ A/D Converter with Programmable Gain Instrumentation Amplifier

Figure 14 - 1 shows the block diagram of the 24-bit ΔΣ A/D converter with programmable gain instrumentation amplifier.

Figure 14 - 1 Block Diagram of 24-bit ΔΣ A/D Converter with Programmable Gain Instrumentation Amplifier



14.3 Input Multiplexer

14.3.1 Overview of input multiplexer

The input multiplexer has five analog input channels, four of which (input multiplexers 0 to 3) can be used to input an external signal, and the remaining one of which (input multiplexer 4) is connected to the internal temperature sensor. For input multiplexers 0 to 3, differential input mode or single-ended input mode can be selected for each channel. If single-ended input mode is selected, an internal bias voltage (VBIAS) is connected to the negative input pin.

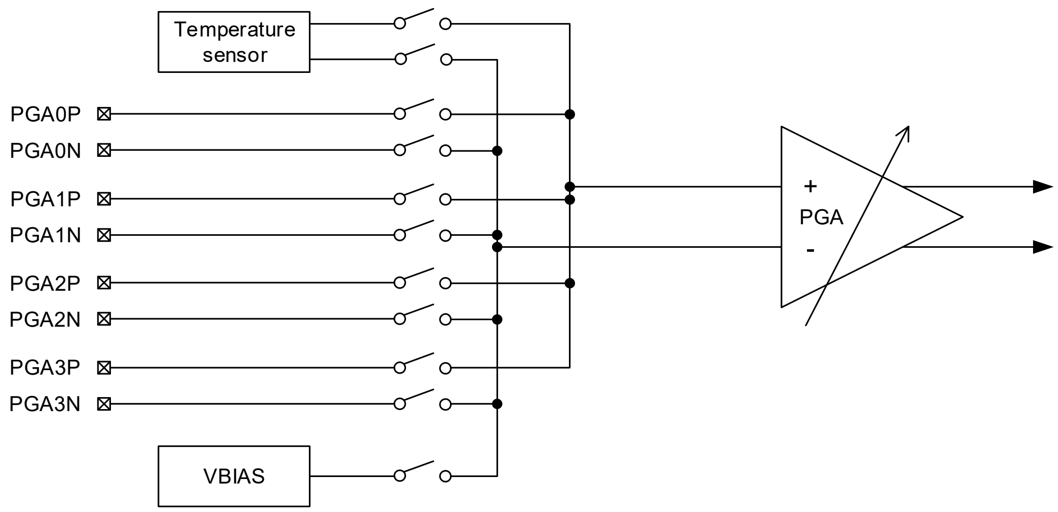
The number of analog input channels of input multiplexers differs depending on the product.

	32-pin products	36-pin products
Number of input multiplexer channels	4 channels (input multiplexers 0 to 2 and 4)	5 channels (input multiplexers 0 to 4)

14.3.2 Configuration of input multiplexer

Figure 14 - 2 shows the block diagram of an input multiplexer.

Figure 14 - 2 Block Diagram of Input Multiplexer



14.3.3 Registers controlling input multiplexers

The following register is used to control input multiplexers.

(1) Input multiplexer x (x = 0 to 3) setting register 1 (PGAxCTL1)

This register is used to specify differential input mode or single-ended input mode for each input multiplexer.

The PGAxCTL1 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 10H.

Figure 14 - 3 Format of Input Multiplexer x (x = 0 to 3) Setting Register 1 (PGAxCTL1)

Address: F045BH (PGA0CTL1), F045FH (PGA1CTL1), After reset: 10H R/W
F0463H (PGA2CTL1), F0467H (PGA3CTL1)

Symbol	7	6	5	4	3	2	1	0
PGAxCTL1	PGAxSEL	PGA3TSEL Note 1	0	PGAxOFS4 Note 2	PGAxOFS3 Note 2	PGAxOFS2 Note 2	PGAxOFS1 Note 2	PGAxOFS0 Note 2

PGAxSEL	Input multiplexer x (x = 0 to 3)
0	Differential input
1	Single-ended input

Note 1. Bit 6 of the PGA0CTL1 to PGA2CTL1 registers is fixed to 0. For details about the PGA3TSEL bit of the PGA3CTL1 register, refer to 14.4.6 (3) Configurable amplifier 0 output selection register (AMP0S0) and input multiplexer 3 setting register 1 (PGA3CTL1).

Note 2. For details about the PGAxOFS0 to PGAxOFS4 bits, refer to 14.4.6 (2) Input multiplexer x (x = 0 to 4) setting register 1 (PGAxCTL1).

14.4 Programmable Gain Instrumentation Amplifier (PGA)

14.4.1 Overview of programmable gain instrumentation amplifier (PGA)

The programmable gain instrumentation amplifier (PGA) features low offset voltage, low 1/f noise, and high impedance. The PGA operates in differential input mode, single-ended input mode, or internal temperature sensor input mode, according to the setting of the input multiplexer used.

In differential input mode and single-ended input mode, a gain from x1 to x64 (G_{TOTAL}) can be specified by combining the gain in the preamplifier (G_{SET1}) and the gain in the post amplifier (G_{SET2}) in the instrumentation amplifier. The gain cannot be changed in internal temperature sensor input mode. G_{SET1} and G_{SET2} are internally fixed to be $G_{TOTAL} = 2$.

A D/A converter for adjusting the offset voltage is connected to the post amplifier. In differential input mode and single-ended input mode, the offset voltage can be adjusted (from -164 mV to +164 mV, in 31 steps (5 bits)) by using this D/A converter. In internal temperature sensor input mode, the offset voltage cannot be adjusted. The D/A converter output is internally fixed to 0 mV.

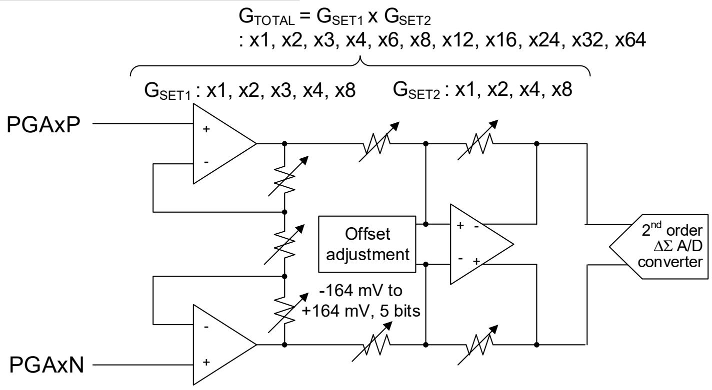
To detect disconnection between a sensor and a PGA input, a current source load can be internally connected to the PGA input.

14.4.2 Configuration of programmable gain instrumentation amplifier (PGA)

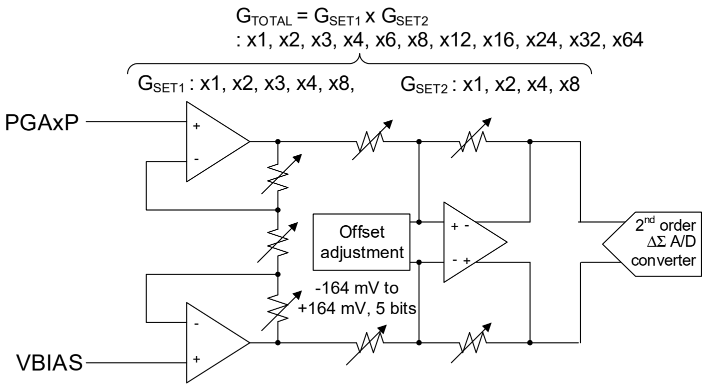
Figure 14 - 4 shows the block diagram of the programmable gain instrumentation amplifier (PGA).

Figure 14 - 4 Block Diagram of Programmable Gain Instrumentation Amplifier (PGA)

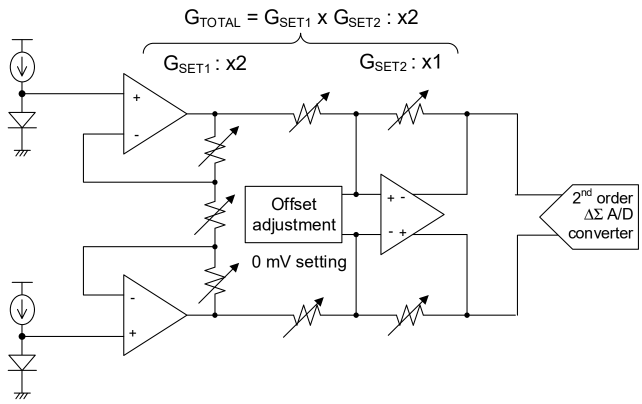
In differential input mode



In single-ended input mode



In internal temperature sensor input mode



14.4.3 Input voltage range

This section describes the range of voltage input to the programmable gain instrumentation amplifier (PGA). Figure 14 - 5 and Figure 14 - 7 show the input voltage range in differential input mode, single-ended input mode, and internal temperature sensor input mode.

14.4.4 Input voltage range in differential input mode

V_{SIG} indicates the input-referred amplitude of the differential voltage input signal, V_{COM} indicates the input-referred common mode input voltage, and d_{OFR} indicates the input-referred D/A converter output voltage for adjusting the offset voltage. The voltage input to an amplifier should be 0.2 to 1.8 V. Therefore, the signal that passes through the preamplifier in the instrumentation amplifier and is then input to the post amplifier must satisfy the conditions in Formula 1.
The signal that passes through the preamplifier in the instrumentation amplifier and is then output from the post amplifier must satisfy the conditions in Formula 2.

Formula 1

$$0.2V + \frac{|V_{SIG}| \times G_{SET1}}{2} \le V_{COM} \le 1.8V - \frac{|V_{SIG}| \times G_{SET1}}{2}$$

Formula 2

$$-0.8V \le (V_{SIG} + d_{OFR}) \times G_{TOTAL} \le 0.8V$$

When d_{OFR} = 0 mV, the input signal is equivalent to the full-scale differential input voltage. V_{COM} can be expressed by using Formula 3, where $V_{SIG} = V_{ID}$ (full-scale differential input voltage).

Formula 3

$$0.2V + \frac{|V_{ID}| \times G_{SET1}}{2} \le V_{COM} \le 1.8V - \frac{|V_{ID}| \times G_{SET1}}{2}$$

Figure 14 - 5 Input Voltage Range in Differential Input Mode and Internal Temperature Sensor Input Mode

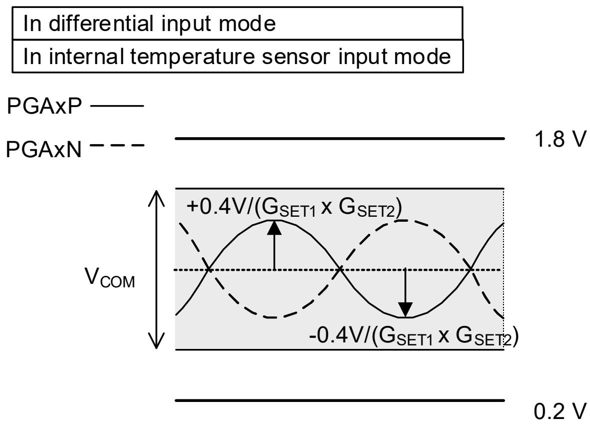
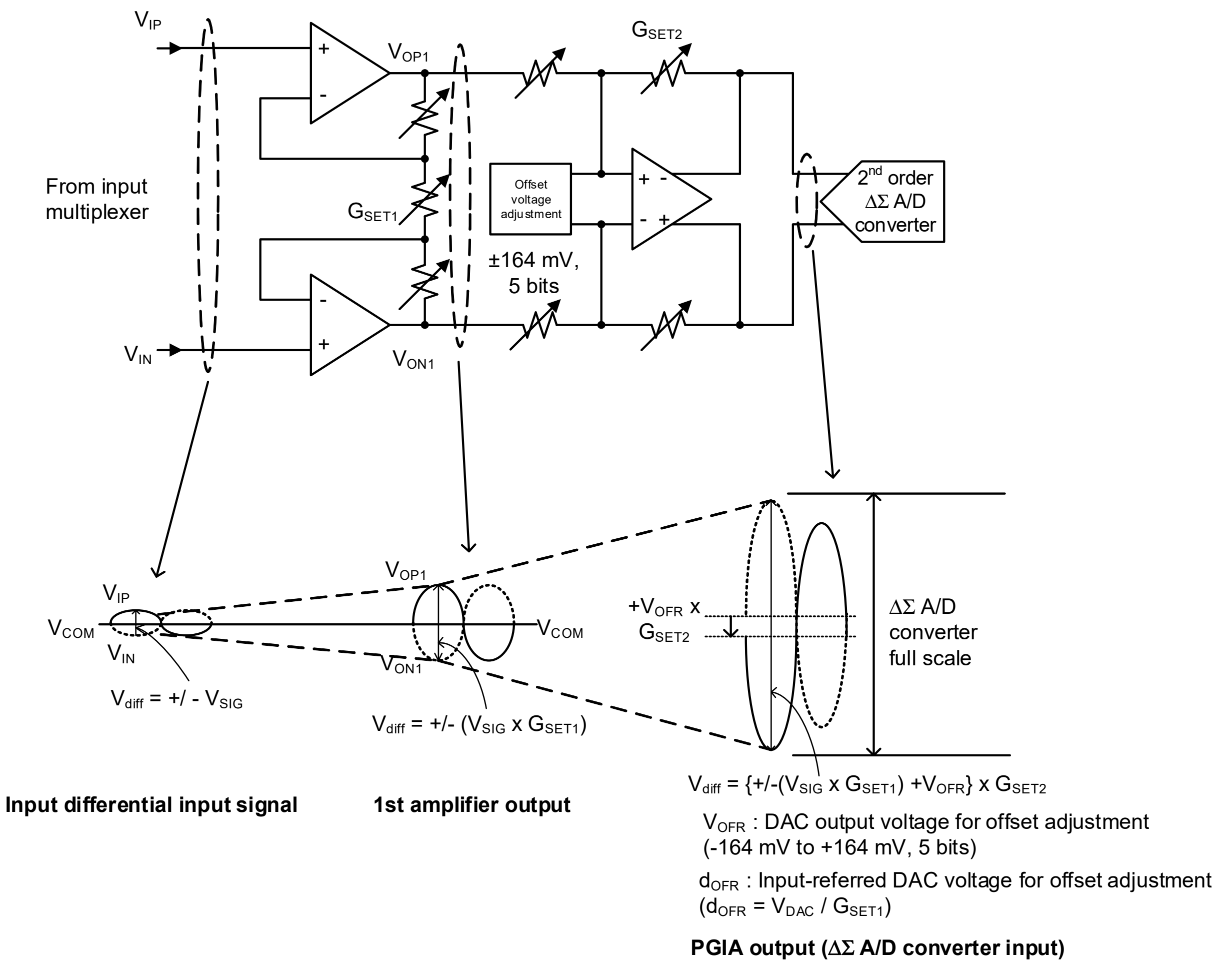


Figure 14 - 6 shows the transition of the differential input voltage level in each phase in the programmable gain instrumentation amplifier (PGA).

Figure 14 - 6 Transition of the Differential Input Voltage Level in the Programmable Gain Instrumentation Amplifier (PGA)

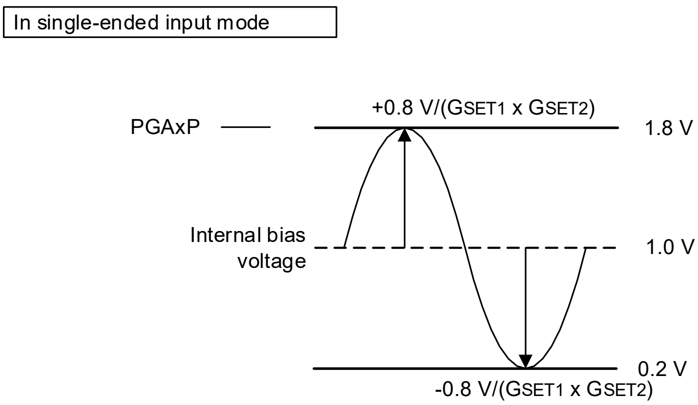


14.4.5 Input voltage range in single-ended input mode

In single-ended input mode, the signal from input multiplexer x (x = 0 to 3) is connected to the non-inverting input in the programmable gain instrumentation amplifier (PGA). The internal bias voltage (VBIAS = 1.0 V (TYP.)) is connected to the inverting input in the programmable gain instrumentation amplifier (PGA) as a reference voltage. A differential signal in the range of 0.2 to 1.8 V is output based on the reference voltage.
The input voltage range (V_I) must satisfy the conditions in the following formulas:

Formula 1: $0.2 \text{ V} \leq V_I \leq 1.8 \text{ V}$
Formula 2: $-0.8 \text{ V} \leq (V_I - 1.0 \text{ V} + d_{ORF}) \times G_{TOTAL} \leq +0.8 \text{ V}$

Figure 14 - 7 Input Voltage Range in Single-Ended Input Mode



14.4.6 Registers controlling the programmable gain instrumentation amplifier (PGA)

The following registers are used to control the programmable gain instrumentation amplifier (PGA).

- Input multiplexer x (x = 0 to 4) setting register 0 (PGAxCTL0)
- Input multiplexer x (x = 0 to 4) setting register 1 (PGAxCTL1)
- Configurable amplifier 0 output selection register (AMP0S0)
- Disconnection detection setting register (PGABOD)

(1) Input multiplexer x (x = 0 to 4) setting register 0 (PGAxCTL0)

This register is used to specify the gain of the programmable gain instrumentation amplifier for input multiplexer x (x = 0 to 4).

The PGAxCTL0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 40H.

Figure 14 - 8 Format of Input Multiplexer x (x = 0 to 4) Setting Register 0 (PGAxCTL0)

Address: F045AH (PGA0CTL0), F045EH (PGA1CTL0), F0462H (PGA2CTL0), F0466H (PGA3CTL0), F046AH (PGA4CTL0) After reset: 40H R/W

Symbol	7	6	5	4	3	2	1	0
PGAxCTL0	PGAxOSR2 Note 1	PGAxOSR1 Note 1	PGAxOSR0 Note 1	PGAxGC4 Note 2	PGAxGC3 Note 2	PGAxGC2 Note 2	PGAxGC1 Note 2	PGAxGC0 Note 2

PGAxGC4	PGAxGC3	PGAxGC2	PGAxGC1	PGAxGC0	Gain setting		
					GSET1	GSET2	GTOTAL
0	0	0	0	0	1	1	1
0	0	1	0	0	2	1	2
0	1	0	0	0	3	1	3
0	1	1	0	0	4	1	4
1	0	0	0	0	8	1	8
0	0	0	0	1	1	2	2
0	0	1	0	1	2	2	4
0	1	0	0	1	3	2	6
0	1	1	0	1	4	2	8
1	0	0	0	1	8	2	16
0	0	0	1	0	1	4	4
0	0	1	1	0	2	4	8
0	1	0	1	0	3	4	12
0	1	1	1	0	4	4	16
1	0	0	1	0	8	4	32
0	0	0	1	1	1	8	8
0	0	1	1	1	2	8	16
0	1	0	1	1	3	8	24
0	1	1	1	1	4	8	32
1	0	0	1	1	8	8	64
Other than above					Setting prohibited		

(Notes are listed on the next page.)

Note 1. For details about the PGAxOSR2 to PGAxOSR0 bits, refer to 14.5.4 (7) Input multiplexer x (x = 0 to 4) setting register 0 (PGAxCTL0).

Note 2. Bits 0 to 4 of the PGA4CTL0 register is fixed to 0.

(2) Input multiplexer x (x = 0 to 4) setting register 1 (PGAxCTL1)

This register is used to adjust the offset voltage for each input multiplexer channel.

The offset voltage dOFR (input-referred D/A converter output voltage for adjusting the offset voltage) is calculated from the following formula.

dOFR (mV) = (-175 + 350/32 x m)/GSET1

(m = 1 to 31: value set in the PGAxCTL1 register)

The PGAxCTL1 register can be set by an 8-bit memory manipulation instruction.

Setting the PGA4CTL1 register is ignored. When this register is read, 00H is always returned.

Reset signal generation sets this register to 10H.

Figure 14 - 9 Format of Input Multiplexer x (x = 0 to 4) Setting Register 1 (PGAxCTL1)

Address: F045BH (PGA0CTL1), F045FH (PGA1CTL1), F0463H (PGA2CTL1), F0467H (PGA3CTL1) F046BH (PGA4CTL1)Note 1 After reset: 10H R/W

Symbol	7	6	5	4	3	2	1	0
PGAxCTL1	PGAxSEL Note 2	PGA3TSEL Note 3	0	PGAxOFS4	PGAxOFS3	PGAxOFS2	PGAxOFS1	PGAxOFS0

PGAxOFS4	PGAxOFS3	PGAxOFS2	PGAxOFS1	PGAxOFS0	dOFR
0	0	0	0	0	Setting prohibited
0	0	0	0	1	−164.06/GSET1
0	0	0	1	0	−153.13/GSET1
...	...	...	...	...	...
1	0	0	0	0	0
...	...	...	...	...	...
1	1	1	0	1	+142.19/GSET1
1	1	1	1	0	+153.13/GSET1
1	1	1	1	1	+164.06/GSET1

Note 1. Setting the PGA4CTL1 register is ignored. When this register is read, 00H is always returned.

Note 2. For details about the PGAxSEL bits, refer to 14.3.3 (1) Input multiplexer x (x = 0 to 3) setting register 1 (PGAxCTL1).

Note 3. Bit 6 of the PGA0CTL1 to PGA2CTL1 registers is fixed to 0. For details about the PGA3TSEL bit of the PGA3CTL1 register, refer to (3) Configurable amplifier 0 output selection register (AMP0S0) and input multiplexer 3 setting register 1 (PGA3CTL1).

(3) Configurable amplifier 0 output selection register (AMP0S0) and input multiplexer 3 setting register 1 (PGA3CTL1)

These registers are used to measure the offset of the configurable amplifier and perform self-diagnosis for the offset of the PGA.

The AMP0S0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 14 - 10 Format of Configurable Amplifier 0 Output Selection Register (AMP0S0)

Address: F0470H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
AMP0S0	AMPMONI1	AMPMONI0	0	0	0	0	AMP0OX1	AMP0OX0

Address: F0467H After reset: 10H R/W

Symbol	7	6	5	4	3	2	1	0
PGA3CTL1	PGA3SEL Note 1	PGA3TSEL	0	PGA3OFS4 Note 2	PGA3OFS3 Note 2	PGA3OFS2 Note 2	PGA3OFS1 Note 2	PGA3OFS0 Note 2

PGA3TSEL	AMPMONI1	AMPMONI0	Mode
0	×	×	Input mode of input multiplexer 3
1	0	0	PGA offset self-diagnosis mode
1	0	1	AMP0 offset measurement mode
1	1	0	AMP1 offset measurement mode
1	1	1	AMP2 offset measurement mode

Note 1. For details about the PGAxSEL bits, refer to 14.3.3 (1) Input multiplexer x (x = 0 to 3) setting register 1 (PGAxCTL1).

Note 2. For details about the PGA3OFS0 to PGA3OFS4 bits, refer to (2) Input multiplexer x (x = 0 to 4) setting register 1 (PGAxCTL1).

Remark ×: Don't care

(4) Disconnection detection setting register (PGABOD)

This register is used to specify whether to enable detection of a disconnection of signal lines connected to PGAxP or PGAxN (x = 0 to 3).

When the PGABOD register is set for detection of the disconnection state, 1 μA (typ.) from the current supply DAC is connected for input to the PGA. When a signal line is disconnected or the power supply capacity falls below 1 μA (typ.), the result of A/D conversion is clipped.

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

Reset signal generation clears this register to 00H.

Figure 14 - 11 Format of Disconnection Detection Setting Register (PGABOD)

Address: F046EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PGABOD	0	0	0	0	0	0	0	PGABOD0

PGABOD0	Control of disconnection detection
0	Normal operation
1	State of disconnection detection

14.5 24-bit ΔΣ A/D Converter

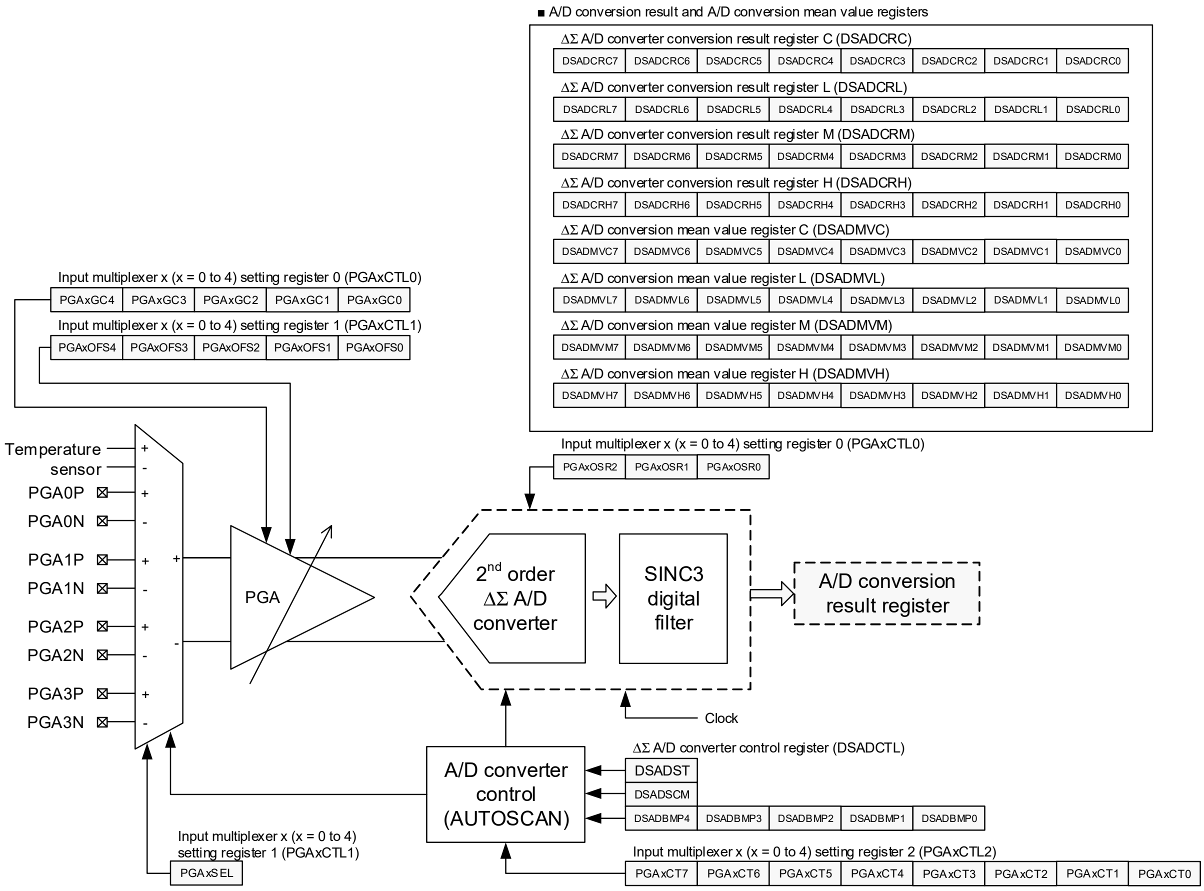
14.5.1 Overview of 24-bit ΔΣ A/D converter

The RL78/I1E incorporates a 24-bit ΔΣ A/D converter. The signal from an input multiplexer (there are 5 channels in total) is input to the 24-bit ΔΣ A/D converter via the programmable gain instrumentation amplifier (PGA). The A/D conversion result is filtered by the SINC3 digital filter, and is then stored in an output register. A/D conversion is performed based on the clock generated in the high-speed on-chip oscillator (HOCO) (sampling frequency = 1 MHz (TYP.)). The high-speed system clock and the PLL clock generated based on the high-speed system clock can also be used. A/D conversion is performed based on a built-in sequencer called AUTOSCAN. The data rate (frequency with which each A/D conversion result is output) can be specified for each channel.

14.5.2 Configuration of 24-bit ΔΣ A/D converter

Figure 14 - 12 shows the block diagram of the 24-bit ΔΣ A/D converter.

Figure 14 - 12 Block Diagram of 24-bit ΔΣ A/D Converter



14.5.3 Voltage input to the 24-bit ΔΣ A/D converter and A/D conversion result

This section describes the relationship between the voltage input to the 24-bit ΔΣ A/D converter and A/D conversion result. The figure and table below show the A/D conversion result when the full-scale range of voltage can be input to the A/D converter.

Figure 14 - 13 Voltage Input to the 24-bit ΔΣ A/D Converter and A/D Conversion Result

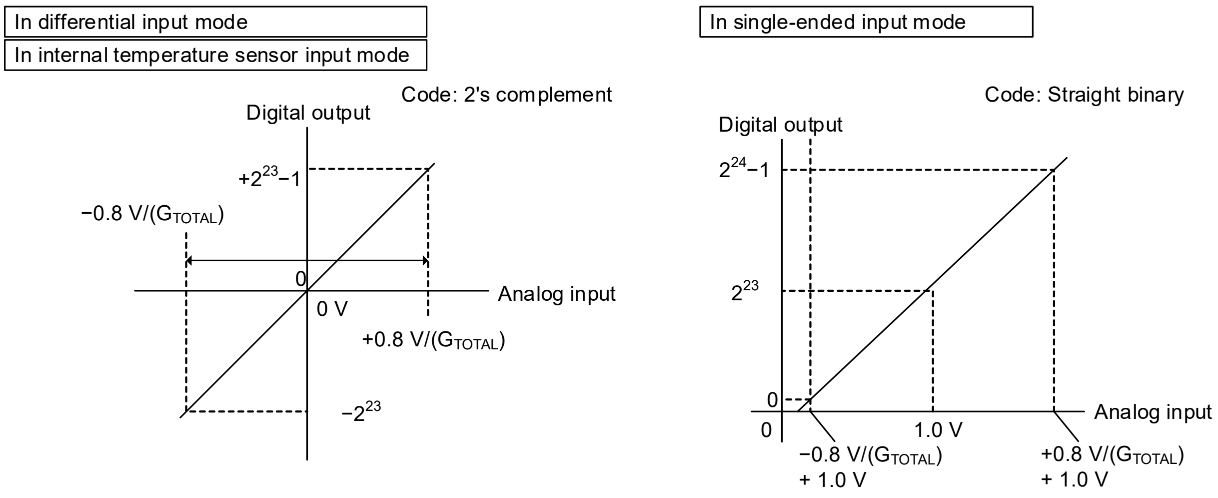


Table 14 - 1 Voltage Input to the 24-bit ΔΣ A/D Converter and A/D Conversion Result

Differential input mode		Single-ended input mode	
Voltage input to ΔΣ A/D converter	A/D conversion result (2's complement)	Voltage input to ΔΣ A/D converter	A/D conversion result (straight binary)
+0.8 V/(G_{TOTAL})	2^{23}-1	+0.8 V/(G_{TOTAL}) + 1.0 V	2^{24} − 1
0 V	0	1.0 V	2^{23}
−0.8 V/(G_{TOTAL})	-2^{23}	−0.8 V/(G_{TOTAL}) + 1.0 V	0

The results in Table 14 - 1 are calculated from the following formulae.

- In differential input mode or internal temperature sensor input mode (G_{TOTAL} = 2)
  Voltage input to ΔΣ A/D converter = (1.6 V/G_{TOTAL}) x (ADCDATA1/2^{24})
  ADCDATA1: 2's complement of 24-bit result of A/D conversion (the higher-order 8 bits in DSADCRH, the middle-order 8 bits in DSADCRM, and the lower-order 8 bits in DSADCRL)
- In single-ended input mode
  Voltage input to ΔΣ A/D converter = (1.6 V/G_{TOTAL}) x (ADCDATA2/2^{24}) + 0.2 V
  ADCDATA2: Straight binary value of 24-bit result of A/D conversion (the higher-order 8 bits in DSADCRH, the middle-order 8 bits in DSADCRM, and the lower-order 8 bits in DSADCRL)

14.5.4 Registers controlling the 24-bit ΔΣ A/D converter

The following registers are used to control the 24-bit ΔΣ A/D converter.

- Peripheral enable register (PER1)
- Analog front-end power supply selection register (AFEPWS)
- Analog front-end power supply detection register (AFEPWD)
- Analog front-end clock selection register (AFECKS)
- ΔΣ A/D converter mode register (DSADMR)
- ΔΣ A/D converter control register (DSADCTL)
- Input multiplexer x (x = 0 to 4) setting register 0 (PGAxCTL0)
- Input multiplexer x (x = 0 to 4) setting register 1 (PGAxCTL1)
- Input multiplexer x (x = 0 to 4) setting register 2 (PGAxCTL2)
- Input multiplexer x (x = 0 to 4) setting register 3 (PGAxCTL3)
- ΔΣ A/D converter conversion result register C (DSADCRC)
- ΔΣ A/D converter conversion result register L (DSADCRL)
- ΔΣ A/D converter conversion result register M (DSADCRM)
- ΔΣ A/D converter conversion result register H (DSADCRH)
- ΔΣ A/D converter mean value register C (DSADMVC)
- ΔΣ A/D converter mean value register L (DSADMVL)
- ΔΣ A/D converter mean value register M (DSADMVM)
- ΔΣ A/D converter mean value register H (DSADMVH)
- ΔΣ A/D converter conversion result register 0 (DSADCR0)
- ΔΣ A/D converter conversion result register 1 (DSADCR1)
- ΔΣ A/D converter mean value register 0 (DSADMV0)
- ΔΣ A/D converter mean value register 1 (DSADMV1)
- Disconnection detection setting register (PGABOD)

(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.

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

Reset signal generation clears this register to 00H.

Figure 14 - 14 Format of Peripheral Enable Register 1 (PER1)

Address: F007AH After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	<3>	<2>	<1>	<0>
PER1	DACEN	TRGEN	AMPEN	0	DTCEN	PGAEN	AFEEN	TRJ0EN

PGAEN	Control of input clock supplied to PGA and 24-bit ΔΣ A/D converter
0	Stops input clock supply. • SFRs used by PGA and the 24-bit ΔΣ A/D converter cannot be written. • PGA and the 24-bit ΔΣ A/D converter are in the reset status.
1	Enables input clock supply. • SFRs used by PGA and the 24-bit ΔΣ A/D converter can be read and written.

Caution Be sure to clear bit 4 to “0”.

(2) Analog front-end power supply selection register (AFEPWS)

The AFEPWS register is used to control the power supplied to the programmable gain instrumentation amplifier (PGA) and AFE reference voltage (ABGR) blocks.

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

Reset signal generation clears this register to 00H.

Figure 14 - 15 Format of Analog Front-End Power Supply Selection Register (AFEPWS)

Address: F0440H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	<0>
AFEPWS	DACPON	AMP2PON	AMP1PON	AMP0PON	0	PGAPON	0	AFEPON

PGAPON	Control of power supplied to programmable gain instrumentation amplifier (PGA) block
0	Power-off (default)
1	Power-on

AFEPON	Control of power supplied to AFE reference voltage (ABGR) block
0	Power-off (default)
1	Power-on

Caution Be sure to clear bits 1 and 3 to “0”.
For the setting of bits 4 to 7, refer to 13.3.2 Analog front-end power supply selection register (AFEPWS) .

(3) Analog front-end power supply detection register (AFEPWD)

The AFEPWD register is a status register that shows the status of the power supplied to the programmable gain instrumentation amplifier (PGA) block and AFE reference voltage block (ABGR).

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

Reset signal generation clears this register to 00H.

Figure 14 - 16 Format of Analog Front-End Power Supply Selection Register (AFEPWS)

Address: F0441H After reset: 00H R

Symbol	7	6	5	4	3	<2>	1	<0>
AFEPWD	0	0	0	0	0	PGASTAT	0	AFESTAT

PGASTAT	Status of power supplied to programmable gain instrumentation amplifier (PGA) block
0	Off or stabilizing
1	Stabilized

AFESTAT	Status of power supplied to AFE reference voltage (ABGR) block
0	Off or stabilizing
1	Stabilized

(4) Analog front-end clock selection register (AFECKS)

This register is used to generate the AFE operating clock (f_{DSADCK}) that is only used by the ΔΣ A/D converter based on the 24-bit ΔΣ A/D converter clock (f_{DSAD}) selected in the peripheral clock control register (PCKC). The setting that makes the frequency of the AFE operating clock (f_{DSADCK}) to be 4 MHz must be specified by using the AFECKS3 to AFECKS0 bits, according to the base clock frequency. When the ΔΣ A/D converter is used in low power mode, the specified frequency of the AFE operating clock (f_{DSADCK}) is divided by 8 by using an internal frequency divider.

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

Reset signal generation clears this register to 00H.

Figure 14 - 17 Format of Analog Front-End Clock Selection Register (AFECKS)

Address: F0442H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
AFECKS	0	0	0	0	AFECKS3	AFECKS2	AFECKS1	AFECKS0

AFECKS3	AFECKS2	AFECKS1	AFECKS0	Selection of AFE operating clock (f_{DSADCK})
0	×	×	×	Stop the clock output (default)
1	0	0	0	f_{DSAD} (undivided)
1	0	0	1	$f_{DSAD}/2$ (divided by 2)
1	0	1	0	$f_{DSAD}/3$ (divided by 3)
1	0	1	1	$f_{DSAD}/4$ (divided by 4)
1	1	0	0	$f_{DSAD}/5$ (divided by 5)
1	1	0	1	$f_{DSAD}/6$ (divided by 6)
1	1	1	×	$f_{DSAD}/8$ (divided by 8)

Remark ×: Don't care

(5) ΔΣ A/D converter mode register (DSADMR)

This register is used to select the trigger signal for the ΔΣ A/D converter to start operating and its operating mode.

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

Reset signal generation clears this register to 00H.

Figure 14 - 18 Format of ΔΣ A/D Converter Mode Register (DSADMR)

Address: F0458H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
DSADMR	DSADTMD	DSADLPM	0	0	0	0	0	0

DSADTMD	Selection of A/D conversion trigger signal
0	Software trigger (Conversion starts when the corresponding SFR is written; default)
1	Hardware trigger (Conversion starts when an event signal selected for the ELC is received)

DSADLPM	Selection of A/D conversion mode
0	Normal operating mode, frequency of AFE operating clock (f_{DSADCK}) is 4 MHz (default)
1	Low power mode, frequency of AFE operating clock (f_{DSADCK}) / 8 is 500 kHz (1/8 the frequency in normal operating mode)

Caution The setting to specify the frequency of AFE operating clock (fDSADCK) to be 4 MHz must be specified by using the AFECKS3 to AFECKS0 bits in advance. For details, refer to (4) Analog front-end clock selection register (AFECKS).

(6) ΔΣ A/D converter control register (DSADCTL)

This register is used to start and stop ΔΣ A/D converter operation. This register is also used to enable or disable A/D conversion of input signals, for each input multiplexer channel. For details, refer to 14.5.5 Control of ΔΣ A/D converter (AUTOSCAN).

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

Reset signal generation clears this register to 00H.

Figure 14 - 19 Format of ΔΣ A/D Converter Control Register (DSADCTL)

Address: F0459H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
DSADCTL	DSADST	0	DSADSCM	DSADBMP4	DSADBMP3	DSADBMP2	DSADBMP1	DSADBMP0

DSADST	Control of A/D conversion (based on AUTOSCAN)
0	Stop A/D conversion.
1	Start A/D conversion.

DSADSCM	Selection of autoscan mode
0	Successive scan mode.
1	Single scan mode.

DSADBMP4	Signal from input multiplexer 4 (temperature sensor)
0	Enable A/D conversion.
1	Stop A/D conversion.

DSADBMPn	Signal from input multiplexer n (n = 0 to 3)
0	Enable A/D conversion.
1	Stop A/D conversion.

(7) Input multiplexer x (x = 0 to 4) setting register 0 (PGAxCTL0)

This register is used to specify the data rate (frequency with which each A/D conversion result is output) for input multiplexer x (x = 0 to 4). For details, refer to 14.5.5 Control of ΔΣ A/D converter (AUTOSCAN).

The PGAxCTL0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 40H.

Figure 14 - 20 Format of Input Multiplexer x (x = 0 to 4) Setting Register 0 (PGAxCTL0)

Address: F045AH (PGA0CTL0), F045EH (PGA1CTL0), After reset: 40H R/W
F0462H (PGA2CTL0), F0466H (PGA3CTL0),
F046AH (PGA4CTL0)

Symbol	7	6	5	4	3	2	1	0
PGAxCTL0	PGAxOSR2	PGAxOSR1	PGAxOSR0	PGAxGC4 Note	PGAxGC3 Note	PGAxGC2 Note	PGAxGC1 Note	PGAxGC0 Note

PGAxOSR2	PGAxOSR1	PGAxOSR0	OSR (oversampling ratio)
0	0	0	64
0	0	1	128
0	1	0	256
0	1	1	512
1	0	0	1024
1	0	1	2048
Other than above			Setting prohibited

Note For details about the PGAxGC4 to PGAxGC0 bits of the PGAxCTL0 register, refer to 14.4.6 (1) Input multiplexer x (x = 0 to 4) setting register 0 (PGAxCTL0) (x = 0 to 4). The PGA4GC4 to PGA4GC0 bits are not implemented in the PGA4CTL0 register.

(8) Input multiplexer x (x = 0 to 4) setting register 2 (PGAxCTL2)

This register is used to specify the number of A/D conversions per AUTOSCAN cycle for input multiplexer x (x = 0 to 4). The number of A/D conversions *N* can be expressed by using the formula below. For details, refer to 14.5.5 Control of ΔΣ A/D converter (AUTOSCAN).

The PGAxCTL2 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation sets this register to 01H.

$N = 32 \times (2^n - 1) + m \times 2^n$ (m and n correspond to the values set to the PGAxCTL2 register)

Figure 14 - 21 Format of Input Multiplexer x (x = 0 to 4) Setting Register 2 (PGAxCTL2)

Address: F045CH (PGA0CTL2), F0460H (PGA1CTL2), After reset: 01H R/W
F0464H (PGA2CTL2), F0468H (PGA3CTL2)
F046CH (PGA4CTL2)

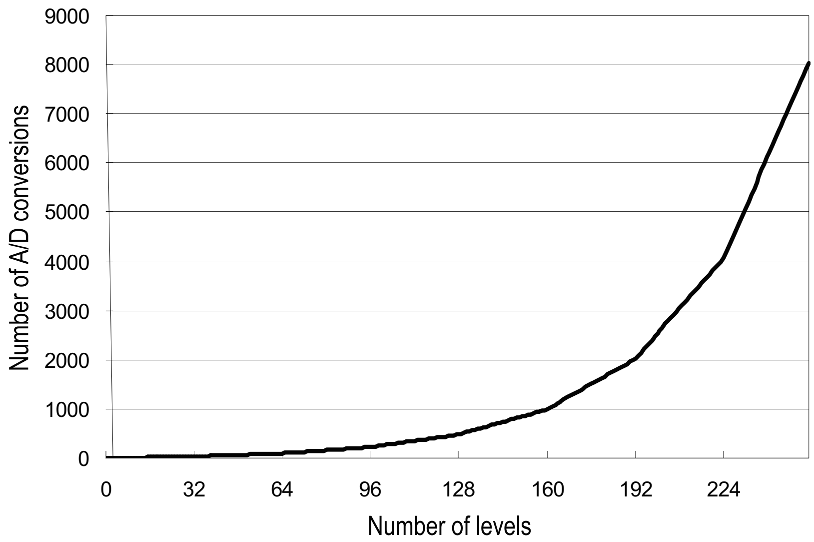
Symbol	7	6	5	4	3	2	1	0
PGAxCTL2	PGAxCT7	PGAxCT6	PGAxCT5	PGAxCT4	PGAxCT3	PGAxCT2	PGAxCT1	PGAxCT0

PGAxCT4	PGAxCT3	PGAxCT2	PGAxCT1	PGAxCT0	m
0	0	0	0	0	0
0	0	0	0	1	1
0	0	0	1	0	2
...	...	...	...	...	...
1	0	0	0	0	16
...	...	...	...	...	...
1	1	1	0	1	29
1	1	1	1	0	30
1	1	1	1	1	31

PGAxCT7	PGAxCT6	PGAxCT5	n
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Up to 256 levels can be selected by combining m and n. The following shows the correlation of the number of levels (register value) and the number of A/D conversions.

Figure 14 - 22 Correlation of the Number of Levels (Register Value) and the Number of A/D Conversions



(9) Input multiplexer x (x = 0 to 4) setting register 3 (PGAxCTL3)

This register is used to specify the mode for specifying the number of A/D conversions per AUTOSCAN cycle for input multiplexer x (x = 0 to 4) and how A/D conversion results are averaged.

The PGAxCTL3 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 14 - 23 Format of Input Multiplexer x (x = 0 to 4) Setting Register 3 (PGAxCTL3)

Address: F045DH (PGA0CTL3), F0461H (PGA1CTL3), After reset: 00H R/W
F0465H (PGA2CTL3), F0469H (PGA3CTL3)
F046DH (PGA4CTL3)

Symbol	7	6	5	4	3	2	1	0
PGAxCTL3	PGAxCTM	0	0	0	PGAxAVE3	PGAxAVE2	PGAxAVE1	PGAxAVE0

PGAxCTM	Selection of the mode for specifying the number of A/D conversions
0	Specify 1 to 8,032 times by using the value set in the PGAxCTL2 register (default)
1	Specify 1 to 255 times linearly by using the value set in the PGAxCTL2 register

PGAxAVE3	PGAxAVE2	Selection of averaging processing
0	0	Do not average the A/D conversion results (default)
0	1	
1	0	Average the A/D conversion results and generates INTDSAD each time an A/D conversion occurs.
1	1	Average the A/D conversion results and generates INTDSAD each time the mean value (N consecutive results of A/D conversion) is output.

PGAxAVE1	PGAxAVE0	Selection of N (the number of data units to be averaged)
0	0	8
0	1	16
1	0	32
1	1	64

(10) ΔΣ A/D converter conversion result register C (DSADCRC)

This is a read-only register that is used to check the number of the channel corresponding to the A/D conversion result. You can check the state of the result of A/D conversion and the number of the input multiplexer channel corresponding to the conversion result. For details, refer to 14.5.5 Control of ΔΣ A/D converter (AUTOSCAN).

The DSADCRC register can be read by using an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 14 - 24 Format of ΔΣ A/D Converter Conversion Result Register C (DSADCRC)

Address: F0450H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
DSADCRC	DSADCRC7	DSADCRC6	DSADCRC5	DSADCRC4	0	0	0	0

DSADCRC7	DSADCRC6	DSADCRC5	Number of the channel corresponding to the A/D conversion result
0	0	0	Invalid
0	0	1	Input multiplexer 0 (PGA0P/PGA0N)
0	1	0	Input multiplexer 1 (PGA1P/PGA1N)
0	1	1	Input multiplexer 2 (PGA2P/PGA2N)
1	0	0	Input multiplexer 3 (PGA3P/PGA3N)
1	0	1	Input multiplexer 4 (temperature sensor)
1	1	0	Invalid
1	1	1	Invalid

DSADCRC4	Flag that indicates the state of the result of A/D conversion
0	Normal state (within range)
1	Clipping has occurred.[Note]

Note The result of A/D conversion is clipped in the range listed in Table 14 - 1.

(11) ΔΣ A/D converter conversion result register L (DSADCRL)

This is a read-only register that is used to check the A/D conversion result. This register stores the lower 8 bits of the 24-bit A/D conversion result. For details, refer to 14.5.5 Control of ΔΣ A/D converter (AUTOSCAN).

The DSADCRL register can be read by using an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 14 - 25 Format of ΔΣ A/D Converter Conversion Result Register L (DSADCRL)

Address: F0451H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
DSADCRL	DSADCRL7	DSADCRL6	DSADCRL5	DSADCRL4	DSADCRL3	DSADCRL2	DSADCRL1	DSADCRL0

(12) ΔΣ A/D converter conversion result register M (DSADCRM)

This is a read-only register that is used to check the A/D conversion result. This register stores the middle 8 bits of the 24-bit A/D conversion result. For details, refer to 14.5.5 Control of ΔΣ A/D converter (AUTOSCAN).

The DSADCRM register can be read by using an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 14 - 26 Format of ΔΣ A/D Converter Conversion Result Register M (DSADCRM)

Address: F0452H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
DSADCRM	DSADCRM7	DSADCRM6	DSADCRM5	DSADCRM4	DSADCRM3	DSADCRM2	DSADCRM1	DSADCRM0

(13) ΔΣ A/D converter conversion result register H (DSADCRH)

This is a read-only register that is used to check the A/D conversion result. This register stores the higher 8 bits of the 24-bit A/D conversion result. For details, refer to 14.5.5 Control of ΔΣ A/D converter (AUTOSCAN).

The DSADCRH register can be read by using an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 14 - 27 Format of ΔΣ A/D Converter Conversion Result Register H (DSADCRH)

Address: F0453H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
DSADCRH	DSADCRH7	DSADCRH6	DSADCRH5	DSADCRH4	DSADCRH3	DSADCRH2	DSADCRH1	DSADCRH0

(14) ΔΣ A/D converter mean value register C (DSADMVC)

This is a read-only register that is used to check the number of the channel corresponding to the mean value. You can check the state of the mean value and the number of the input multiplexer channel corresponding to the mean value. For details, refer to 14.5.5 Control of ΔΣ A/D converter (AUTOSCAN).

The DSADMVC register can be read by using an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 14 - 28 Format of ΔΣ A/D Converter Mean Value Register C (DSADMVC)

Address: F0454H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
DSADMVC	DSADMVC7	DSADMVC6	DSADMVC5	DSADMVC4	0	0	0	0

DSADMVC7	DSADMVC6	DSADMVC5	Number of the channel corresponding to the mean value
0	0	0	Invalid
0	0	1	Input multiplexer 0 (PGA0P/PGA0N)
0	1	0	Input multiplexer 1 (PGA1P/PGA1N)
0	1	1	Input multiplexer 2 (PGA2P/PGA2N)
1	0	0	Input multiplexer 3 (PGA3P/PGA3N)
1	0	1	Input multiplexer 4 (temperature sensor)
1	1	0	Invalid
1	1	1	Invalid

DSADMVC4	Flag that indicates the state of the mean value
0	Normal state (within range)
1	Clipping has occurred.[Note]

Note DSADMVC4 being 1 indicates that at least one result of A/D conversion used for averaging was clipped in the range listed in Table 14 - 1. The mean value is not limited to having the maximum or minimum value.

Caution Even when the corresponding PGAxAVE3 bit (x = 0 to 4) is set to 0, which is the setting for not averaging the results of A/D conversion, the value in the mean value register may change in response to the start of A/D conversion. However, the channel number indicated by bits DSADMVC7 to DSADMVC5 will always be 0 (invalid).

(15) ΔΣ A/D converter mean value register L (DSADMVL)

This is a read-only register that is used to check the mean value. This register stores the lower 8 bits of the 24-bit mean value. For details, refer to 14.5.5 Control of ΔΣ A/D converter (AUTOSCAN).

The DSADMVL register can be read by using an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 14 - 29 Format of ΔΣ A/D Converter Mean Value Register L (DSADMVL)

Address: F0455H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
DSADMVL	DSADMVL7	DSADMVL6	DSADMVL5	DSADMVL4	DSADMVL3	DSADMVL2	DSADMVL1	DSADMVL0

(16) ΔΣ A/D converter mean value register M (DSADMVM)

This is a read-only register that is used to check the mean value. This register stores the middle 8 bits of the 24-bit mean value. For details, refer to 14.5.5 Control of ΔΣ A/D converter (AUTOSCAN).

The DSADMVM register can be read by using an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 14 - 30 Format of ΔΣ A/D Converter Mean Value Register M (DSADMVM)

Address: F0456H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
DSADMVM	DSADMVM7	DSADMVM6	DSADMVM5	DSADMVM4	DSADMVM3	DSADMVM2	DSADMVM1	DSADMVM0

(17) ΔΣ A/D converter mean value register H (DSADMVH)

This is a read-only register that is used to check the mean value. This register stores the higher 8 bits of the 24-bit mean value. For details, refer to 14.5.5 Control of ΔΣ A/D converter (AUTOSCAN).

The DSADMVH register can be read by using an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 14 - 31 Format of ΔΣ A/D Converter Mean Value Register H (DSADMVH)

Address: F0457H After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
DSADMVH	DSADMVH7	DSADMVH6	DSADMVH5	DSADMVH4	DSADMVH3	DSADMVH2	DSADMVH1	DSADMVH0

(18) ΔΣ A/D converter conversion result register 0 (DSADCR0)

This is a read-only register that is used to check the A/D conversion result. The DSADCRC and DSADCRL registers can be read in a batch by using a 16-bit memory manipulation instruction. For details, refer to 14.5.5 Control of ΔΣ A/D converter (AUTOSCAN).

The DSADCRC register can be read by using a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 14 - 32 Format of ΔΣ A/D Converter Conversion Result Register 0 (DSADCR0)

Address: F0450H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8
DSADCR0	DSADCRL							
	7	6	5	4	3	2	1	0
	DSADCRC							

(19) ΔΣ A/D converter conversion result register 1 (DSADCR1)

This is a read-only register that is used to check the A/D conversion result. The DSADCRM and DSADCRH registers can be read in a batch by using a 16-bit memory manipulation instruction. For details, refer to 14.5.5 Control of ΔΣ A/D converter (AUTOSCAN).

The DSADCR1 register can be read by using a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 14 - 33 Format of ΔΣ A/D Converter Conversion Result Register 1 (DSADCR1)

Address: F0452H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8
DSADCR1	DSADCRH							
	7	6	5	4	3	2	1	0
	DSADCRM							

(20) ΔΣ A/D converter mean value register 0 (DSADMV0)

This is a read-only register that is used to check the mean value. The DSADMVC and DSADMVL registers can be read in a batch by using a 16-bit memory manipulation instruction. For details, refer to 14.5.5 Control of ΔΣ A/D converter (AUTOSCAN).

The DSADMV0 register can be read by using a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 14 - 34 Format of ΔΣ A/D Converter Mean Value Register 0 (DSADMV0)

Address: F0454H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8
DSADMV0	DSADMVL							
	7	6	5	4	3	2	1	0
	DSADMVC							

(21) ΔΣ A/D converter mean value register 1 (DSADMV1)

This is a read-only register that is used to check the mean value. The DSADMVM and DSADMVH registers can be read in a batch by using a 16-bit memory manipulation instruction. For details, refer to 14.5.5 Control of ΔΣ A/D converter (AUTOSCAN).

The DSADMV1 register can be read by using a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 14 - 35 Format of ΔΣ A/D Converter Mean Value Register 1 (DSADMV1)

Address: F0456H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8
DSADMV1	DSADMVH							
	7	6	5	4	3	2	1	0
	DSADMVM							

14.5.5 Control of ΔΣ A/D converter (AUTOSCAN)

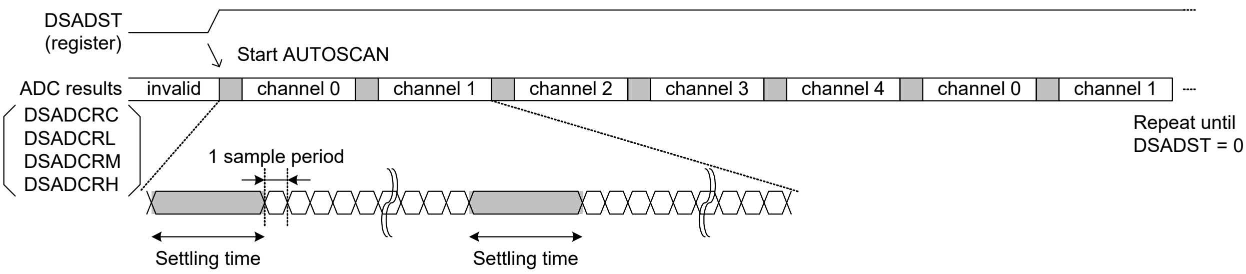
All A/D conversions is controlled based on a built-in sequencer called AUTOSCAN. When "1" is written to the DSADST bit of the DSADCTL register to enable AUTOSCAN, the signal input from each channel is A/D-converted in round-robin fashion. A/D conversion of the signal input from a specific channel can be skipped by setting the DSADBMPn bit (n = 0 to 4) of the DSADCTL register.

Use the PGAxCTy bit (x = 0 to 4, y = 0 to 7) of the PGAxCTL2 register to specify the number of times A/D conversion is to be performed in an active channel until execution shifts to the next channel. If PGAxCTy is set to 00H, it sets one-shot operation, which stops A/D conversion each time a conversion ends. Other A/D conversion parameters such as the PGA gain and oversampling ratio can also be configured for each channel.

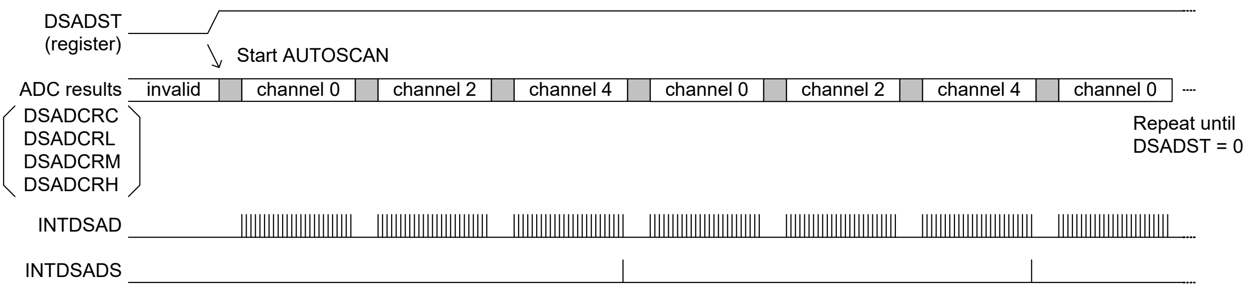
The A/D conversion result is stored in the DSADCRC, DSADCRH, DSADCRM, and DSADCRL registers.

An interrupt request (INTDSAD) is generated each time A/D conversion is completed. When averaging of the results of A/D conversion is enabled by the setting of the PGAxCTL3 register, an interrupt request (INTDSAD) can be generated each time A/D conversion is completed or each time the mean value is updated. The interrupt request (INTDSADS) is generated on completion of each cycle of AUTOSCAN from channels 0 to 4.

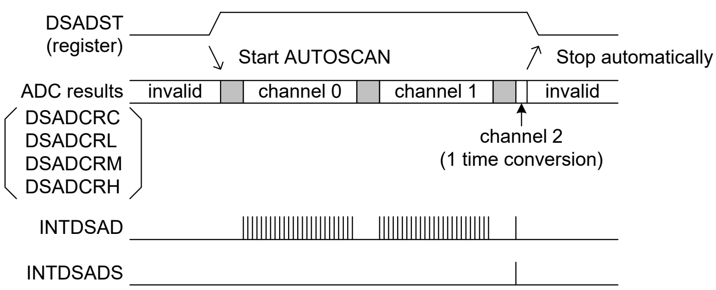
Figure 14 - 36 AUTOSCAN sequence



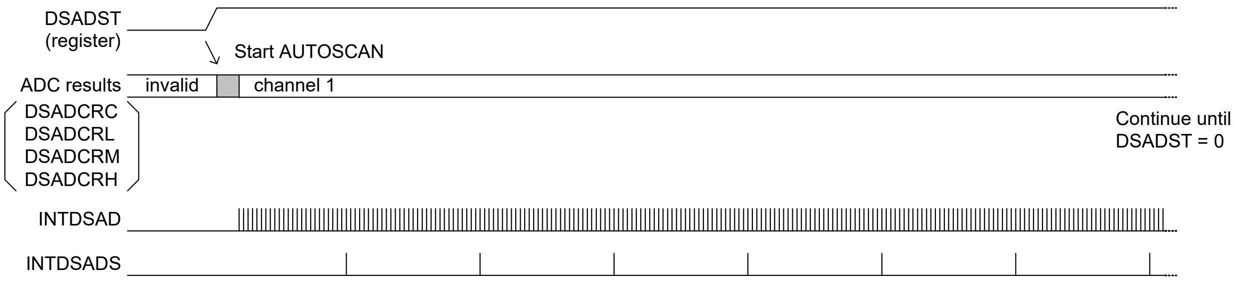
Example 1: Skipping an A/D conversion channel (DSADBMP4 to DSADBMP0 = 01010B, PGAxCTy (x = 0, 2, 4) > 0, and DASDSCM = 0)



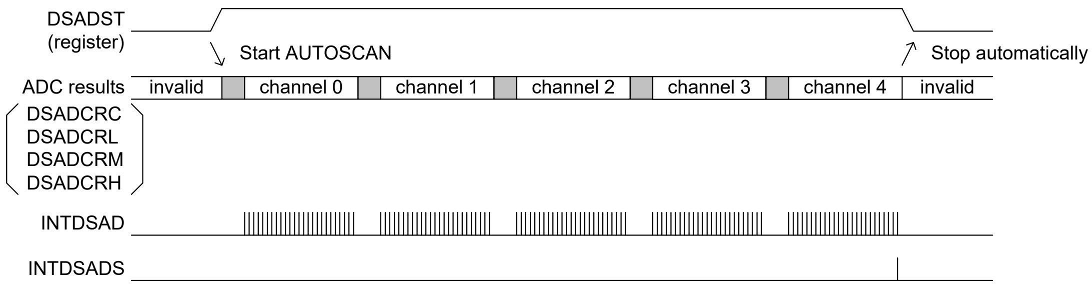
Example 2: One-shot operation (DSADBMP4 to DSADBMP0 = 11000B, PGAxCTy (x = 0 or 1) > 0, PGA2CTy = 0, and DSADSCM = 0)



Example 3: Sequential conversion on a single channel (DSADBMP4 to DSADBMP0 = 11101B, PGA1CTy > 0, and DSADSCM = 0)



Example 4: Single-scan operation (DSADBMP4 to DSADBMP0 = 00000B and DSADSCM = 1)



Remark Even for sequential conversion, an interrupt request (INTDSADS) is generated each time the number of rounds of A/D conversion set in the PGAxCTL2 register is completed.

14.5.6 Overview of digital filter

A SINC3 digital filter is used to downsample A/D conversion results. The digital filter transfer function is expressed by using the following equation. M in the equation of the transfer function represents the factor of decimation by the digital filter, which is itself determined by the OSR (oversampling ratio) set in the PGAxOSRn bit of the PGAxCTL0 register (x = 0 to 4, n = 0 to 2).

$$H(z) = \left(\frac{1}{M} \cdot \frac{1 - z^{-M}}{1 - z^{-1}} \right)^3$$

14.5.7 Configuration of digital filter

Figure 14 - 37 shows the block diagram of the digital filter. Three integrators and three differentiators are cascaded. Considering the A/D converter stabilization time, clock synchronization at the input stage in the digital filter, and a delay caused due to 3 stages of the differentiator, three times the sampling period (= 3 x 1/fout) + 128 μs is required as the settling time.

Remark The settling time is automatically generated by the built-in sequencer AUTOSCAN.

Figure 14 - 37 Block Diagram of Digital Filter

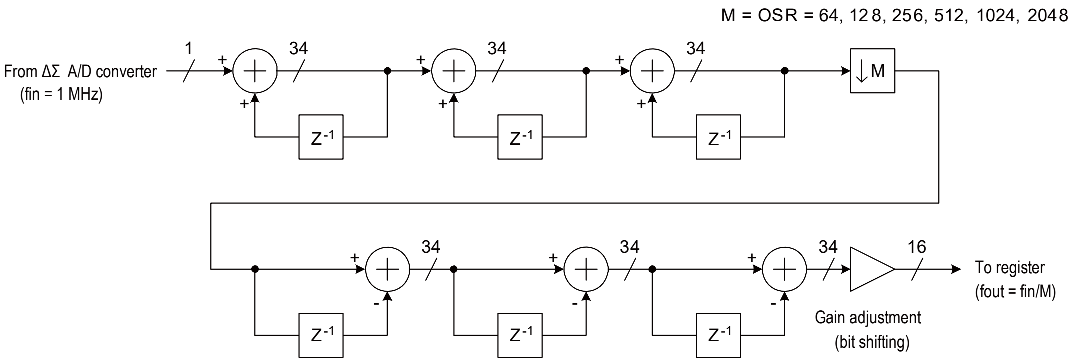
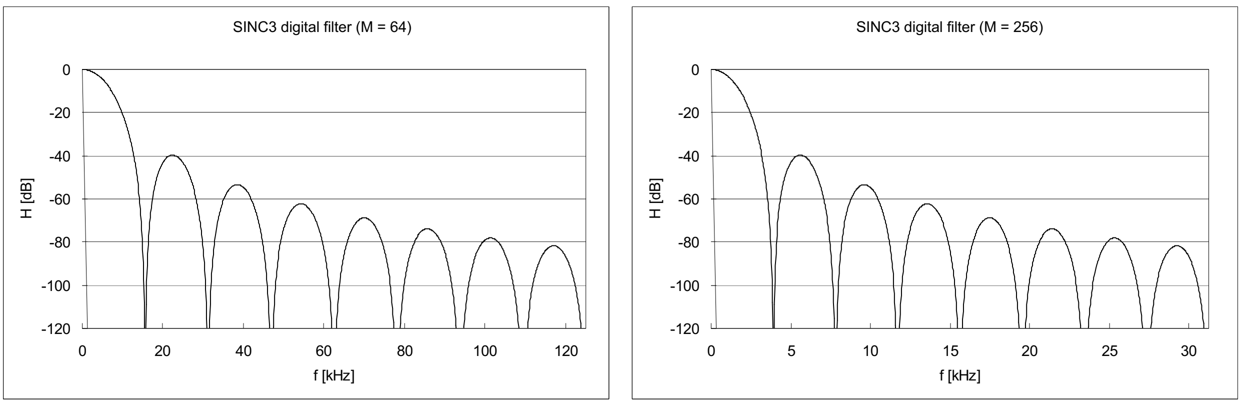


Figure 14 - 38 shows the frequency response of the digital filter.

Figure 14 - 38 SINC3 Filter Frequency Response



14.6 Procedure for Controlling 24-bit ΔΣ A/D Converter with Programmable Gain Instrumentation Amplifier

Figures 14 - 39 to 14 - 42 show the flowchart for starting the 24-bit ΔΣ A/D converter with programmable gain instrumentation amplifier, A/D conversion, stopping the A/D converter, and measuring the temperature sensor.

Figure 14 - 39 Flowchart for Starting the 24-bit ΔΣ A/D Converter with Programmable Gain Instrumentation Amplifier

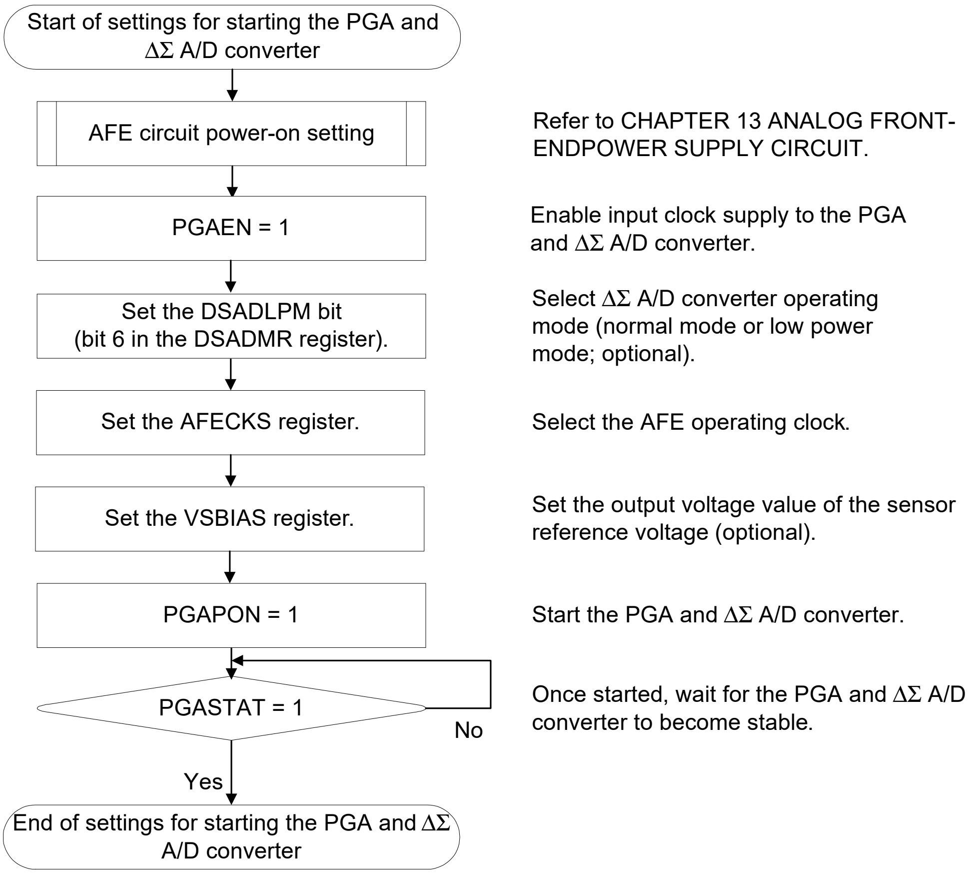


Figure 14 - 40 Flowchart for A/D Conversion by the 24-bit ΔΣ A/D Converter with Programmable Gain Instrumentation Amplifier

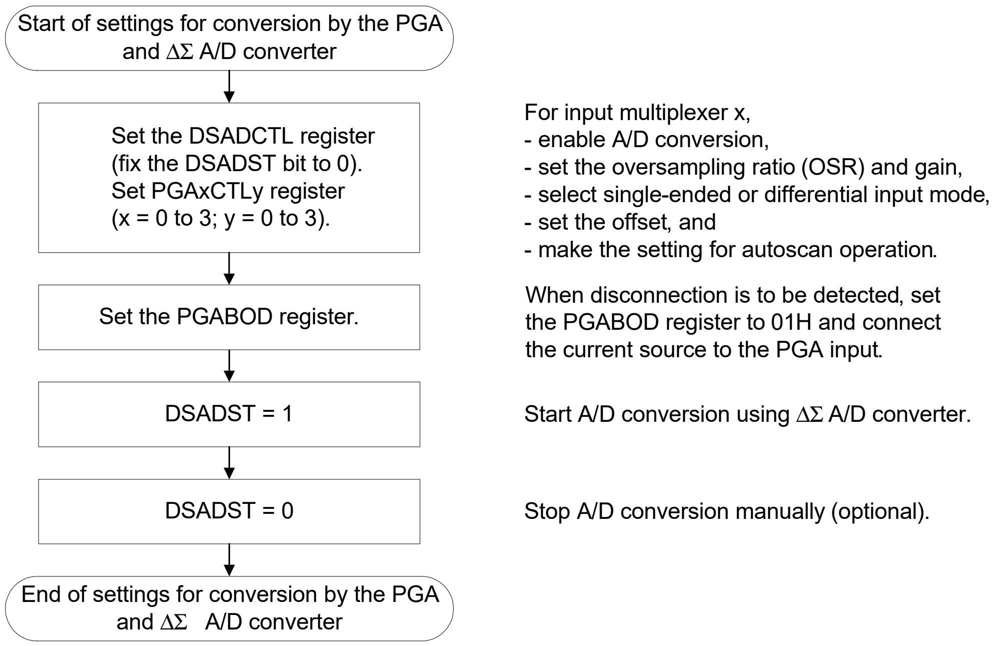


Figure 14 - 41 Flowchart for Stopping the 24-bit ΔΣ A/D Converter with Programmable Gain Instrumentation Amplifier

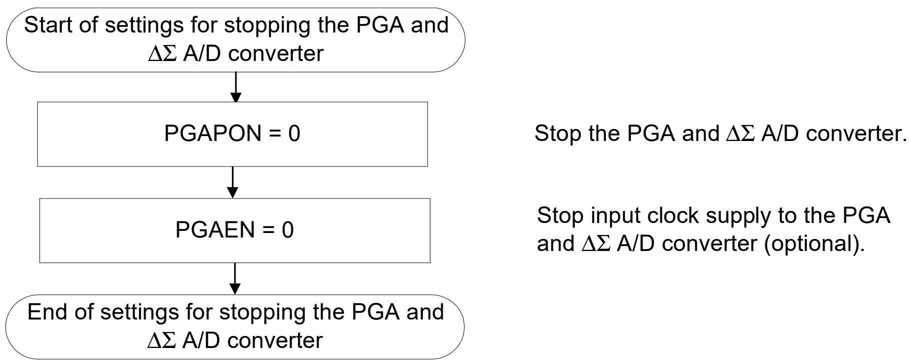
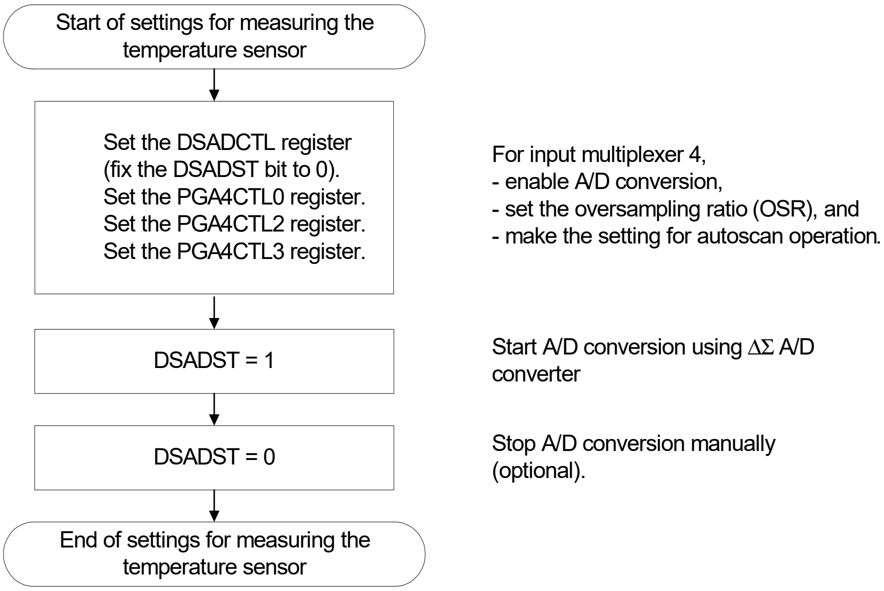


Figure 14 - 42 Flowchart of Settings for Measuring the Temperature Sensor by the 24-bit ΔΣ A/D Converter with Programmable Gain Instrumentation Amplifier



14.7 Cautions for the 24-bit ΔΣ A/D Converter with Programmable Gain Instrumentation Amplifier

(1) When using the temperature sensor (input multiplexer 4), gain G_{SET1} is fixed to 2, G_{SET2} is fixed to 1, and input mode is fixed to differential input mode.

(2) It is recommended to change the low power mode setting (by using the DSADLPM bit) and division ratio setting (by using the AFECKS bit) before turning on the PGA power (PGAPON = 0).

CHAPTER 15 TEMPERATURE SENSOR

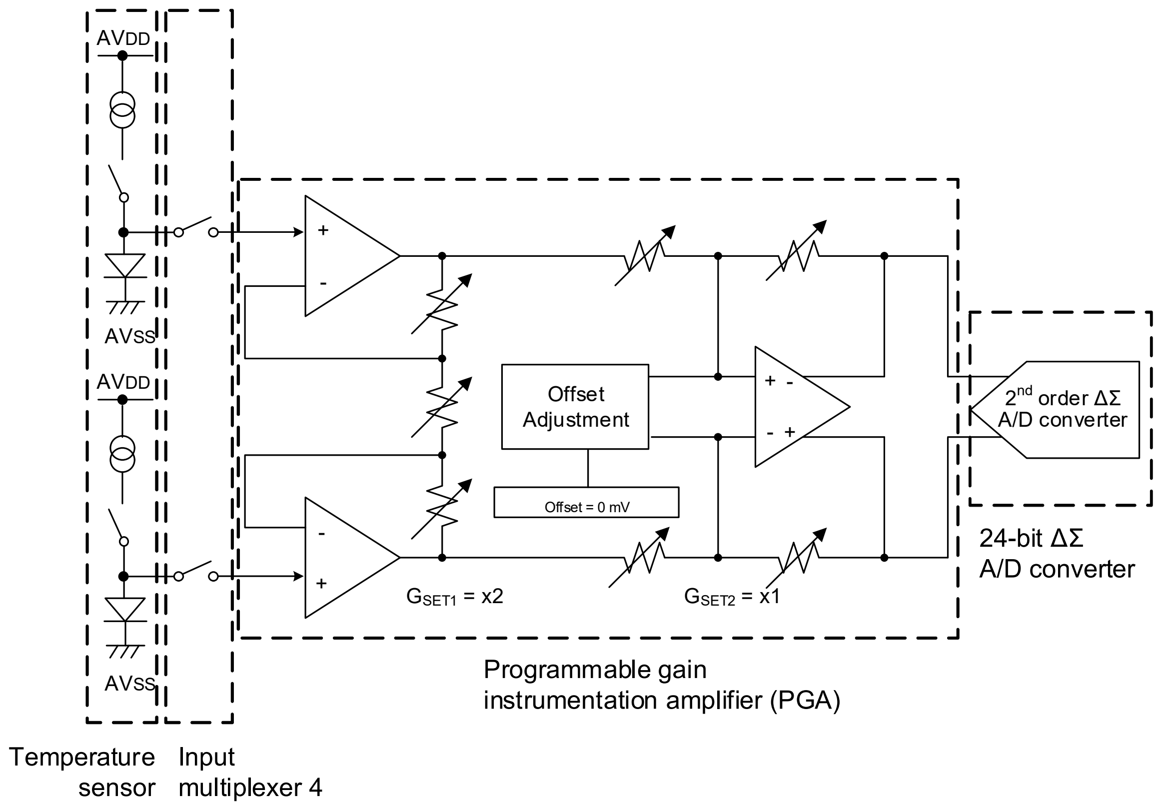
15.1 Overview of Temperature Sensor

The RL78/I1E has one on-chip temperature sensor channel. The output from the on-chip temperature sensor passes through input multiplexer 4 and the programmable gain instrumentation amplifier (PGA), and is then input to the ΔΣ A/D converter. Gain G_{TOTAL} for the temperature sensor is fixed to 2 and this setting cannot be changed. For the temperature dependency, refer to **CHAPTER 33 ELECTRICAL SPECIFICATIONS (G: T_A = -40 to +105°C)** or **CHAPTER 34 ELECTRICAL SPECIFICATIONS (M: T_A = -40 to +125°C)**.

15.2 Configuration of Temperature Sensor

Figure 15 - 1 shows the block diagram of the temperature sensor.

Figure 15 - 1 Block Diagram of Temperature Sensor



15.3 Registers Controlling the Temperature Sensor

The following registers are used to control the temperature sensor.

- ΔΣ A/D converter control register (DSADCTL)
- Input multiplexer 4 setting register 0 (PGA4CTL0)
- Input multiplexer 4 setting register 2 (PGA4CTL2)
- Input multiplexer 4 setting register 3 (PGA4CTL3)
- ΔΣ A/D converter conversion result register C (DSADCRC)
- ΔΣ A/D converter conversion result register L (DSADCRL)
- ΔΣ A/D converter conversion result register M (DSADCRM)
- ΔΣ A/D converter conversion result register H (DSADCRH)
- ΔΣ A/D converter mean value register C (DSADMVC)
- ΔΣ A/D converter mean value register L (DSADMVL)
- ΔΣ A/D converter mean value register M (DSADMVM)
- ΔΣ A/D converter mean value register H (DSADMVH)

For details about registers and operations, refer to CHAPTER 14 24-BIT ΔΣ A/D CONVERTER WITH PROGRAMMABLE GAIN INSTRUMENTATION AMPLIFIER.

CHAPTER 16 A/D CONVERTER

The number of analog input channels of the A/D converter differs, depending on the product.

	32-pin	36-pin
Number of analog input channels	8 channels (ANI1 to ANI7, ANI9)	10 channels (ANI0 to ANI9)

16.1 Function of A/D Converter

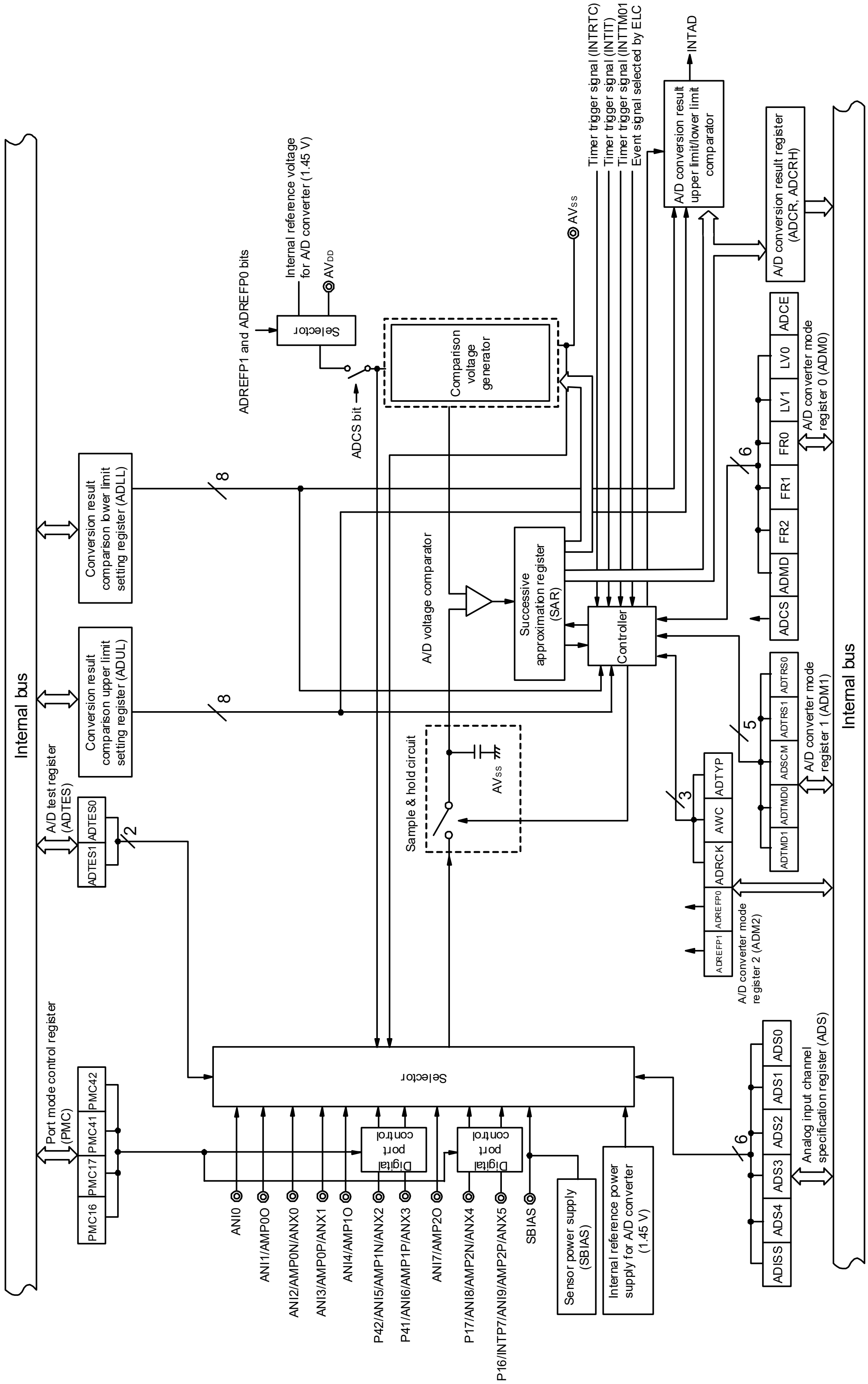
The A/D converter is a converter that converts analog input signals into digital values, and is configured to control analog inputs, including up to ten channels of A/D converter analog inputs (ANI0 to ANI9). 10-bit or 8-bit resolution can be selected by the ADTYP bit of the A/D converter mode register 2 (ADM2).
The A/D converter has the following function.

- 10-bit or 8-bit resolution A/D conversion
  10-bit or 8-bit resolution A/D conversion is carried out repeatedly for one analog input channel selected from ANI0 to ANI9. Each time an A/D conversion operation ends, an interrupt request (INTAD) is generated (when in the select mode).

Various A/D conversion modes can be specified by using the mode combinations below.

Trigger mode	Software trigger	Conversion is started by software.
	Hardware trigger no-wait mode	Conversion is started by detecting a hardware trigger.
	Hardware trigger wait mode	The power is turned on by detecting a hardware trigger while the system is off and in the conversion standby state, and conversion is then started automatically after the stabilization wait time passes. When using the SNOOZE mode function, specify the hardware trigger wait mode.
Channel selection mode	Select mode	A/D conversion is performed on the analog input of one selected channel.
	Scan mode	A/D conversion is performed on the analog input of four channels in order. Four consecutive channels can be selected from ANI0 to ANI7 (36-pin products) or ANI1 to ANI7 (32-pin products) as analog input channels.
Conversion operation mode	One-shot conversion mode	A/D conversion is performed on the selected channel once.
	Sequential conversion mode	A/D conversion is sequentially performed on the selected channels until it is stopped by software.
Operation voltage mode	Standard 1 or standard 2 mode	Select this mode for conversion in the operation voltage range of 2.7 V ≤ AV_{DD} ≤ 5.5 V.
Sampling time selection	Sampling clock cycles: 7 f_{AD}	The sampling time in standard 1 mode is seven cycles of the conversion clock (f_{AD}). Select this mode when the output impedance of the analog input source is high and the sampling time should be long.
	Sampling clock cycles: 5 f_{AD}	The sampling time in standard 2 mode is five cycles of the conversion clock (f_{AD}). Select this mode when enough sampling time is ensured (for example, when the output impedance of the analog input source is low).

Figure 16 - 1 Block Diagram of A/D Converter



Remark Analog input pins shown in Figure 16 - 1 are of 36-pin products.

16.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

(1) ANI0 to ANI9 pins

These are the analog input pins of the ten channels of the A/D converter. They input analog signals to be converted into digital signals. Each of ANI5, ANI6, ANI8, and ANI9 can be used as an I/O port pin when it is not selected as an analog input pin.

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.

A/D voltage comparator

This A/D voltage comparator compares the voltage generated from the voltage tap of the comparison voltage generator with the analog input voltage. If the analog input voltage is found to be greater than the reference voltage (1/2 AV_{REF}) as a result of the comparison, the most significant bit (MSB) of the successive approximation register (SAR) is set. If the analog input voltage is less than the reference voltage (1/2 AV_{REF}), the MSB 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 AV_{REF})

Bit 9 = 1: (3/4 AV_{REF})

The voltage tap of the comparison voltage generator and the analog input voltage are compared and bit 8 of the SAR register is manipulated according to the result of the comparison.

Analog input voltage ≥ Voltage tap of comparison voltage generator: Bit 8 = 1

Analog input voltage ≤ Voltage tap of comparison voltage generator: Bit 8 = 0

Comparison is continued like this to bit 0 of the SAR register.

When performing A/D conversion at a resolution of 8 bits, the comparison continues until bit 2 of the SAR register.

Remark AV_{REF}: The + side reference voltage of the A/D converter. This can be selected from the internal reference voltage for A/D converter (1.45 V) and AV_{DD}.

(2) Comparison voltage generator

The comparison voltage generator generates the comparison voltage input from an analog input pin.

(3) Successive approximation register (SAR)

The SAR register is a register that sets voltage tap data whose values from the comparison voltage generator match the voltage values of the analog input pins, 1 bit at a time starting from the most significant bit (MSB).

If data is set in the SAR register all the way to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register (conversion results) are held in the A/D conversion result register (ADCR). When all the specified A/D conversion operations have ended, an A/D conversion end interrupt request signal (INTAD) is generated.

(4) 10-bit A/D conversion result register (ADCR)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCR register holds the A/D conversion result in its higher 10 bits (the lower 6 bits are fixed to 0).

(5) 8-bit A/D conversion result register (ADCRH)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCRH register stores the higher 8 bits of the A/D conversion result.

(6) Controller

This circuit controls the conversion time of an input analog signal that is to be converted into a digital signal, as well as starting and stopping of the conversion operation. When A/D conversion has been completed, this controller generates INTAD through the A/D conversion result upper limit/lower limit comparator.

16.3 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 1 (ADM1)
- A/D converter mode register 2 (ADM2)
- 10-bit A/D conversion result register (ADCR)
- 8-bit A/D conversion result register (ADCRH)
- Analog input channel specification register (ADS)
- Conversion result comparison upper limit setting register (ADUL)
- Conversion result comparison lower limit setting register (ADLL)
- A/D test register (ADTES)
- Port mode control registers 1 and 4 (PMC1, PMC4)
- Port mode registers 1 and 4 (PM1, PM4)

16.3.1 Peripheral enable register 0 (PER0)

This register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the A/D converter is used, be sure to set bit 5 (ADCEN) of this register to 1.

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

Reset signal generation clears this register to 00H.

Figure 16 - 2 Format of Peripheral enable register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	4	3	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	0	0	SAU0EN	TAU1EN	TAU0EN

ADCEN	Control of A/D converter input clock supply
0	Stops input clock supply. • SFRs used by the A/D converter cannot be written. • The A/D converter is in the reset status.
1	Enables input clock supply. • SFRs used by the A/D converter can be read/written.

Caution 1. When setting the A/D converter, be sure to set the following registers first while the ADCEN bit is set to 1.

If ADCEN = 0, the values of the A/D converter control registers are cleared to their initial values and writing to them is ignored (except for port mode registers 1 and 4 (PM1, PM4), port mode control registers 1 and 4 (PMC1, PMC4), and A/D port configuration register (ADPC)).

- **A/D converter mode register 0 (ADM0)**
- **A/D converter mode register 1 (ADM1)**
- **A/D converter mode register 2 (ADM2)**
- **10-bit A/D conversion result register (ADCR)**
- **8-bit A/D conversion result register (ADCRH)**
- **Analog input channel specification register (ADS)**
- **Conversion result comparison upper limit setting register (ADUL)**
- **Conversion result comparison lower limit setting register (ADLL)**
- **A/D test register (ADTES)**

Caution 2. Be sure to clear bits 6, 4, and 3 to 0.

16.3.2 A/D converter mode register 0 (ADM0)

This register sets the conversion time for analog input to be A/D converted, and starts/stops conversion.
The ADM0 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 16 - 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	ADMD	FR2 Note 1	FR1 Note 1	FR0 Note 1	LV1 Note 1	LV0 Note 1	ADCE

ADCS	A/D conversion operation control	
0	Stops conversion operation [When read] Conversion stopped/standby status	
1	Enables conversion operation [When read] While in the software trigger mode: While in the hardware trigger wait mode:	 Conversion operation status A/D power supply stabilization wait status + conversion operation status

ADMD	Specification of the A/D conversion channel selection mode
0	Select mode
1	Scan mode

ADCE	A/D voltage comparator operation control Note 2
0	Stops A/D voltage comparator operation
1	Enables A/D voltage comparator operation

Note 1. For details of the FR2 to FR0, LV1, LV0 bits, and A/D conversion, see **Tables 16 - 3** and **Table 16 - 4 A/D Conversion Time Selection**.

Note 2. While in the software trigger mode or hardware trigger no-wait mode, the operation of the A/D voltage comparator is controlled by the ADCS and ADCE bits, and it takes 1 μs from the start of operation for the operation to stabilize. Therefore, when the ADCS bit is set to 1 after 1 μs or more has elapsed from the time ADCE bit is set to 1, the conversion result at that time has priority over the first conversion result. Otherwise, ignore data of the first conversion.

Caution 1. Change the ADMD, FR2 to FR0, LV1, and LV0 bits while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 2. Do not set the ADCS bit to 1 and the ADCE bit to 0 at the same time.

Caution 3. Do not change the ADCS and ADCE bits from 0 to 1 at the same time by using an 8-bit manipulation instruction. Be sure to set these bits in the order described in 16.7 A/D Converter Setup Flowchart.

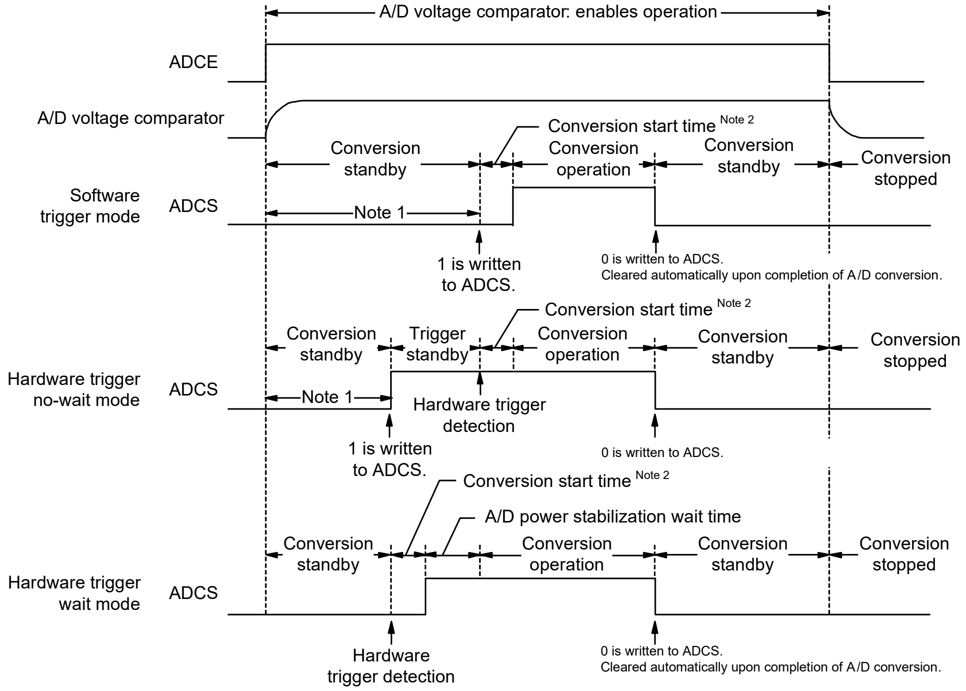
Table 16 - 1 Settings of ADCS and ADCE Bits

ADCS	ADCE	A/D Conversion Operation
0	0	Conversion stopped state
0	1	Conversion standby state
1	0	Setting prohibited
1	1	Conversion-in-progress state

Table 16 - 2 Setting and Clearing Conditions for ADCS Bit

A/D Conversion Mode			Set Conditions	Clear Conditions
Software trigger	Select mode	Sequential conversion mode	When 1 is written to ADCS	When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when A/D conversion ends.
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when conversion ends on the specified four channels.
Hardware trigger no-wait mode	Select mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		When 0 is written to ADCS
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		When 0 is written to ADCS
Hardware trigger wait mode	Select mode	Sequential conversion mode	When a hardware trigger is input	When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when A/D conversion ends.
	Scan mode	Sequential conversion mode		When 0 is written to ADCS
		One-shot conversion mode		• When 0 is written to ADCS • The bit is automatically cleared to 0 when conversion ends on the specified four channels.

Figure 16 - 4 Timing Chart When A/D Voltage Comparator Is Used



Note 1. While in the software trigger mode or hardware trigger no-wait mode, the time from the rising of the ADCE bit to the falling of the ADCS bit must be 1 μs or longer to stabilize the internal circuit.

Note 2. The following time is the maximum amount of time necessary to start conversion.

ADM0			Conversion Clock (f_{AD})	Conversion Start Time (Number of f_{CLK} Clocks)	
FR2	FR1	FR0		Software trigger mode/ Hardware trigger no wait mode	Hardware trigger wait mode
0	0	0	$f_{CLK}/64$	63	1
0	0	1	$f_{CLK}/32$	31	
0	1	0	$f_{CLK}/16$	15	
0	1	1	$f_{CLK}/8$	7	
1	0	0	$f_{CLK}/6$	5	
1	0	1	$f_{CLK}/5$	4	
1	1	0	$f_{CLK}/4$	3	
1	1	1	$f_{CLK}/2$	1	

In hardware trigger mode, for the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 2, and 3 in scan mode, the stabilization wait time for A/D power supply does not occur after a hardware trigger is detected.

Caution 1. If using the hardware trigger wait mode, setting the ADCS bit to 1 is prohibited (but the bit is automatically switched to 1 when the hardware trigger signal is detected). However, it is possible to clear the ADCS bit to 0 to specify the A/D conversion standby status.

Caution 2. While in the one-shot conversion mode of the hardware trigger no-wait mode, the ADCS flag is not automatically cleared to 0 when A/D conversion ends. Instead, 1 is retained.

Caution 3. Only rewrite the value of the ADCE bit when ADCS = 0 (while in the conversion stopped/conversion standby status).

Caution 4. To complete A/D conversion, specify at least the following time as the hardware trigger interval:
Hardware trigger no wait mode: 2 f_{CLK} clock + Conversion start time + A/D conversion time
Hardware trigger wait mode: 2 f_{CLK} clock + Conversion start time + A/D power supply stabilization wait time + A/D conversion time

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 16 - 3 A/D Conversion Time Selection (1/2)

(1) When there is no A/D power supply stabilization wait time Normal mode 1, 2 (software trigger mode/hardware trigger no-wait mode)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of Conversion Clock Note	Conversion Time	Conversion Time at 10-Bit Resolution				
									2.7 V ≤ AV_{DD} ≤ 5.5 V				
FR2	FR1	FR0	LV1	LV0					f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	0	0	Normal 1	f_{CLK}/64	19 f_{AD} (number of sampling clock: 7 f_{AD})	1216/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	76 μs	38 μs
0	0	1				f_{CLK}/32		608/f_{CLK}			76 μs	38 μs	19 μs
0	1	0				f_{CLK}/16		304/f_{CLK}		76 μs	38 μs	19 μs	9.5 μs
0	1	1				f_{CLK}/8		152/f_{CLK}		38 μs	19 μs	9.5 μs	4.75 μs
1	0	0				f_{CLK}/6		114/f_{CLK}		28.5 μs	14.25 μs	7.125 μs	3.5625 μs
1	0	1				f_{CLK}/5		95/f_{CLK}	95 μs	23.75 μs	11.875 μs	5.938 μs	2.9688 μs
1	1	0				f_{CLK}/4		76/f_{CLK}	76 μs	19 μs	9.5 μs	4.75 μs	2.375 μs
1	1	1				f_{CLK}/2		38/f_{CLK}	38 μs	9.5 μs	4.75 μs	2.375 μs	Setting prohibited
0	0	0	0	1	Normal 2	f_{CLK}/64	17 f_{AD} (number of sampling clock: 5 f_{AD})	1088/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	68 μs	34 μs
0	0	1				f_{CLK}/32		544/f_{CLK}			68 μs	34 μs	17 μs
0	1	0				f_{CLK}/16		272/f_{CLK}		68 μs	34 μs	17 μs	8.5 μs
0	1	1				f_{CLK}/8		136/f_{CLK}		34 μs	17 μs	8.5 μs	4.25 μs
1	0	0				f_{CLK}/6		102/f_{CLK}		25.5 μs	12.75 μs	6.375 μs	3.1875 μs
1	0	1				f_{CLK}/5		85/f_{CLK}	85 μs	21.25 μs	10.625 μs	5.3125 μs	2.6563 μs
1	1	0				f_{CLK}/4		68/f_{CLK}	68 μs	17 μs	8.5 μs	4.25 μs	2.125 μs
1	1	1				f_{CLK}/2		34/f_{CLK}	34 μs	8.5 μs	4.25 μs	2.125 μs	Setting prohibited

Note These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (f_{AD}).

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 33.6.4 or 34.6.4 A/D converter characteristics.

Caution 2. Rewrite the FR2 to FR0, LV1 and LV0 bits to other than the same data while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Table 16 - 4 A/D Conversion Time Selection (2/2)

(2) When there is A/D power supply stabilization wait time Normal mode 1, 2 (hardware trigger wait mode Note 1)

A/D Converter Mode Register 0 (ADM0)					Mode	Conversion Clock (f_{AD})	Number of A/D Power Supply Stabilization Wait Clock	Number of Conversion Clock Note 2	A/D Power Supply Stabilization Wait Time + Conversion Time	A/D Power Supply Stabilization Wait Time + Conversion Time at 10-Bit Resolution 2.7 V ≤ AV_{DD} ≤ 5.5 V				
FR2	FR1	FR0	LV1	LV0						f_{CLK} = 1 MHz	f_{CLK} = 4 MHz	f_{CLK} = 8 MHz	f_{CLK} = 16 MHz	f_{CLK} = 32 MHz
0	0	0	0	0	Normal 1	f_{CLK}/64	8 f_{AD}	19 f_{AD} (number of sampling clock: 7 f_{AD})	1728/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	108 µs	54 µs
0	0	1				f_{CLK}/32			864/f_{CLK}			108 µs	54 µs	27 µs
0	1	0				f_{CLK}/16			432/f_{CLK}		108 µs	54 µs	27 µs	13.5 µs
0	1	1				f_{CLK}/8			216/f_{CLK}		54 µs	27 µs	13.5 µs	6.75 µs
1	0	0				f_{CLK}/6			162/f_{CLK}		40.5 µs	20.25 µs	10.125 µs	5.0625 µs
1	0	1				f_{CLK}/5			135/f_{CLK}	135 µs	33.75 µs	16.875 µs	8.4375 µs	4.21875 µs
1	1	0				f_{CLK}/4			108/f_{CLK}	108 µs	27 µs	13.5 µs	6.75 µs	3.375 µs
1	1	1				f_{CLK}/2			54/f_{CLK}	54 µs	13.5 µs	6.75 µs	3.375 µs	Setting prohibited
0	0	0	0	1	Normal 2	f_{CLK}/64	8 f_{AD}	17 f_{AD} (number of sampling clock: 5 f_{AD})	1600/f_{CLK}	Setting prohibited	Setting prohibited	Setting prohibited	100 µs	50 µs
0	0	1				f_{CLK}/32			800/f_{CLK}			100 µs	50 µs	25 µs
0	1	0				f_{CLK}/16			400/f_{CLK}		100 µs	50 µs	25 µs	12.5 µs
0	1	1				f_{CLK}/8			200/f_{CLK}		50 µs	25 µs	12.5 µs	6.25 µs
1	0	0				f_{CLK}/6			150/f_{CLK}		37.5 µs	18.75 µs	9.375 µs	4.6875 µs
1	0	1				f_{CLK}/5			125/f_{CLK}	125 µs	31.25 µs	15.625 µs	7.8125 µs	3.90625 µs
1	1	0				f_{CLK}/4			100/f_{CLK}	100 µs	25 µs	12.5 µs	6.25 µs	3.125 µs
1	1	1				f_{CLK}/2			50/f_{CLK}	50 µs	12.5 µs	6.25 µs	3.125 µs	Setting prohibited

Note 1. For the second and subsequent conversion in sequential conversion mode and for conversion of the channel specified by scan 1, 2, and 3 in scan mode, the conversion start time and stabilization wait time for A/D power supply do not occur after a hardware trigger is detected (see **Table 16 - 3**).

Note 2. These are the numbers of clock cycles when conversion is with 10-bit resolution. When eight-bit resolution is selected, the values are shorter by two cycles of the conversion clock (f_{AD}).

Caution 1. The A/D conversion time must also be within the relevant range of conversion times (t_{CONV}) described in 33.6.4 or 34.6.4 A/D converter characteristics.
Note that the conversion time (t_{CONV}) does not include the A/D power supply stabilization wait time.

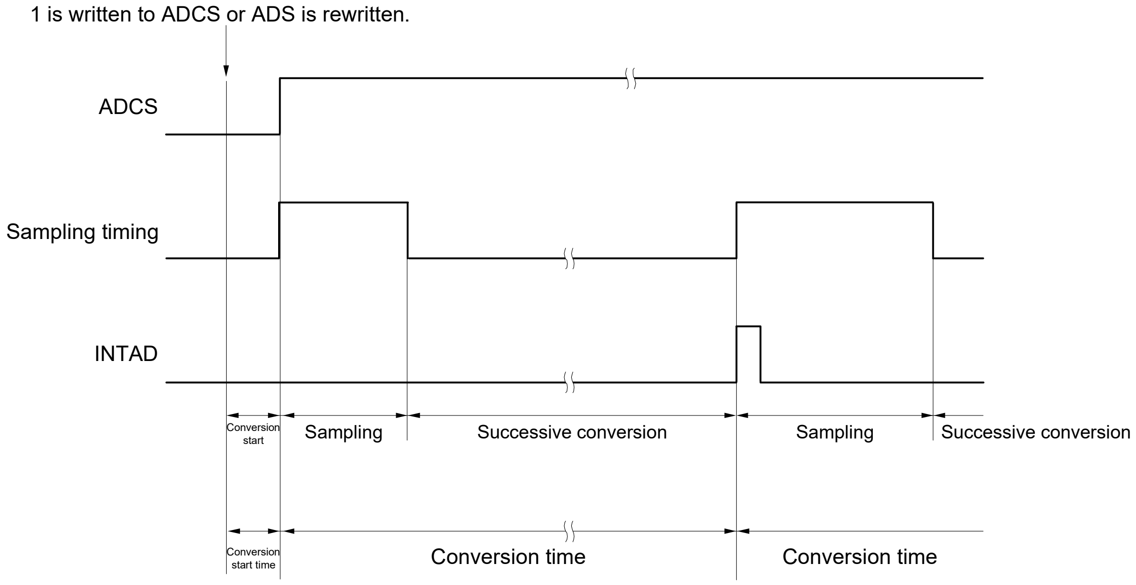
Caution 2. Rewrite the FR2 to FR0, LV1 and LV0 bits to other than the same data while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The above conversion time does not include conversion state time. Conversion state time add in the first conversion. Select conversion time, taking clock frequency errors into consideration.

Caution 4. When hardware trigger wait mode, specify the conversion time, including the A/D power supply stabilization wait time from the hardware trigger detection.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

Figure 16 - 5 A/D Converter Sampling and A/D Conversion Timing (Example for Software Trigger Mode)



16.3.3 A/D converter mode register 1 (ADM1)

This register is used to specify the A/D conversion trigger, conversion mode, and hardware trigger signal.
The ADM1 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 16 - 6 Format of A/D converter mode register 1 (ADM1)

Address: FFF32H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADM1	ADTMD1	ADTMD0	ADSCM	0	0	0	ADTRS1	ADTRS0

ADTMD1	ADTMD0	Selection of the A/D conversion trigger mode
0	0	Software trigger mode
0	1	
1	0	Hardware trigger no-wait mode
1	1	Hardware trigger wait mode

ADSCM	Specification of the A/D conversion mode
0	Sequential conversion mode
1	One-shot conversion mode

ADTRS1	ADTRS0	Selection of the hardware trigger signal
0	0	End of timer channel 1 count or capture interrupt signal (INTTM01)
0	1	Event signal selected by ELC
1	0	Real-time clock interrupt signal (INTRTC)
1	1	Interval timer interrupt signal (INTIT)

Caution 1. Rewrite the value of the ADM1 register while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 2. To complete A/D conversion, specify at least the following time as the hardware trigger interval:
Hardware trigger no wait mode: 2 f_{CLK} clock + conversion start time + A/D conversion time
Hardware trigger wait mode: 2 f_{CLK} clock + conversion start time + A/D power supply stabilization wait time + A/D conversion time

Caution 3. In modes other than SNOOZE mode, input of the next INTRTC or INTIT will not be recognized as a valid hardware trigger for up to four f_{CLK} cycles after the first INTRTC or INTIT is input.

Remark f_{CLK}: CPU/peripheral hardware clock frequency

16.3.4 A/D converter mode register 2 (ADM2)

This register is used to select the + side or - side reference voltage of the A/D converter, check the upper limit and lower limit A/D conversion result values, select the resolution, and specify whether to use the SNOOZE mode.

The ADM2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 16 - 7 Format of A/D converter mode register 2 (ADM2) (1/2)

Address: F0010H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	1	<0>
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

ADREFP1	ADREFP0	Selection of the positive reference voltage source of the A/D converter
0	0	Supplied from AV_{DD}
0	1	Setting prohibited
1	0	Supplied from the internal reference voltage for A/D converter (1.45 V)
1	1	Setting prohibited

- When ADREFP1 or ADREFP0 bit is rewritten, this must be configured in accordance with the following procedures.

(1) Set ADCE = 0
(2) Change the values of ADREFP1 and ADREFP0
(3) Reference voltage stabilization wait time (A)
(4) Set ADCE = 1
(5) Reference voltage stabilization wait time (B)
When ADREFP1 and ADREFP0 are set to 1 and 0, the setting is changed to A = 5 µs, B = 1 µs.
When ADREFP1 and ADREFP0 are set to 0 and 0, A needs no wait and B = 1 µs.

- When ADREFP1 and ADREFP0 are set to 1 and 0, respectively, A/D conversion cannot be performed on the internal reference voltage for A/D converter.
  Be sure to perform A/D conversion while ADISS = 0.

ADREFM	Selection of the negative reference voltage of the A/D converter
0	Supplied from AV_{SS}
1	Setting prohibited

Caution 1. Rewrite the value of the ADM2 register while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 2. When entering STOP mode, do not set ADREFP1 to 1. When selecting the internal reference voltage for A/D converter (ADREFP1, ADREFP0 = 1, 0), the current value of A/D converter reference voltage current (I_{ADREF}) shown in 33.3.2 or 34.3.2 Supply current characteristics is added.

Figure 16 - 8 Format of A/D converter mode register 2 (ADM2) (2/2)

Address: F0010H After reset: 00H R/W

Symbol	7	6	5	4	<3>	<2>	1	<0>
ADM2	ADREFP1	ADREFP0	ADREFM	0	ADRCK	AWC	0	ADTYP

ADRCK	Checking the upper limit and lower limit conversion result values
0	The interrupt signal (INTAD) is output when the ADLL register ≤ the ADCR register ≤ the ADUL register (AREA1).
1	The interrupt signal (INTAD) is output when the ADCR register < the ADLL register (AREA2) or the ADUL register < the ADCR register (AREA3).
Figure 16 - 9 shows the generation range of the interrupt signal (INTAD) for AREA1 to AREA3.	

AWC	Specification of the SNOOZE mode
0	Do not use the SNOOZE mode function.
1	Use the SNOOZE mode function.

When there is a hardware trigger signal in the STOP mode, the STOP mode is exited, and A/D conversion is performed without operating the CPU (the SNOOZE mode).

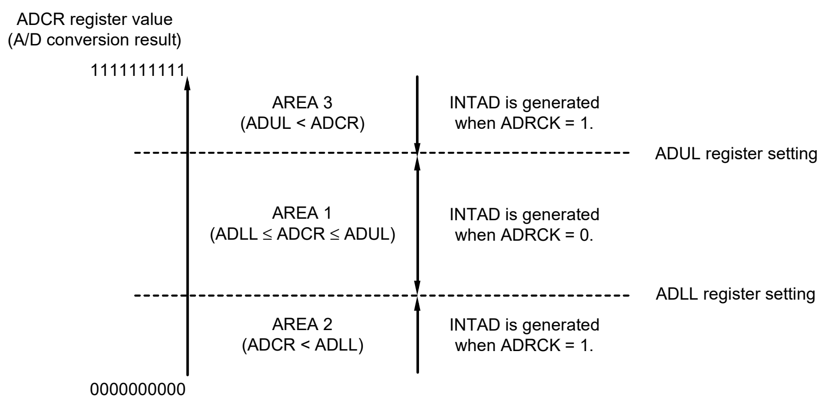
- The SNOOZE mode function can only be specified when the high-speed on-chip oscillator clock is selected for the CPU/peripheral hardware clock (f_{CLK}). If any other clock is selected, specifying this mode is prohibited.
- Using the SNOOZE mode function in the software trigger mode or hardware trigger no-wait mode is prohibited.
- Using the SNOOZE mode function in the sequential conversion mode is prohibited.
- When using the SNOOZE mode function, specify a hardware trigger interval of at least “shift time to SNOOZE mode Note + conversion start time + A/D power supply stabilization wait time + A/D conversion time + 2 f_{CLK} clock”
- Even when using SNOOZE mode, be sure to set the AWC bit to 0 in normal operation and change it to 1 just before shifting to STOP mode.
  Also, be sure to change the AWC bit to 0 after returning from STOP mode to normal operation.
  If the AWC bit is left set to 1, A/D conversion will not start normally in spite of the subsequent SNOOZE or normal operation mode.

ADTYP	Selection of the A/D conversion resolution
0	10-bit resolution
1	8-bit resolution

Note Refer to “Transition time from STOP mode to SNOOZE mode” in **23.3.3 SNOOZE mode**.

Caution Only rewrite the value of the ADM2 register while conversion operation is stopped (which is indicated by the ADCS and ADCE bits of A/D converter mode register 0 (ADM0) being 0).

Figure 16 - 9 ADRCK Bit Interrupt Signal Generation Range



Remark If INTAD does not occur, the A/D conversion result is not stored in the ADCR or ADCRH register.

16.3.5 10-bit A/D conversion result register (ADCR)

This register is a 16-bit register that stores the A/D conversion result in the select mode. The lower 6 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register (SAR). The higher 8 bits of the conversion result are stored in FFF1FH and the lower 2 bits are stored in the higher 2 bits of FFF1EH [Note].

The ADCR register can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Note If the A/D conversion result is outside the range specified by using the A/D conversion comparison function (the value specified by the ADRCK bit of the ADM2 register and ADUL/ADLL registers; see **Figure 16 - 9**), the result is not stored.

Figure 16 - 10 Format of 10-bit A/D conversion result register (ADCR)

Address: FFF1FH, FFF1EH After reset: 0000H R

Symbol	FFF1FH								FFF1EH							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCR											0	0	0	0	0	0

Caution 1. When 8-bit resolution A/D conversion is selected (when the ADTYP bit of A/D converter mode register 2 (ADM2) is 1) and the ADCR register is read, 0 is read from the lower two bits (bits 7 and 6 of the ADCR register).

Caution 2. When the ADCR register is accessed in 16-bit units, the higher 10 bits of the conversion result are read in order starting at bit 15.

16.3.6 8-bit A/D conversion result register (ADCRH)

This register is an 8-bit register that stores the A/D conversion result. The higher 8 bits of 10-bit resolution are stored.

The ADCRH register can be read by an 8-bit memory manipulation instruction [Note].

Reset signal generation clears this register to 00H.

Note If the A/D conversion result is outside the range specified by using the A/D conversion comparison function (the value specified by the ADRCK bit of the ADM2 register and ADUL/ADLL registers; see **Figure 16 - 9**), the result is not stored.

Figure 16 - 11 Format of 8-bit A/D conversion result register (ADCRH)

Address: FFF1FH After reset: 00H R

Symbol	7	6	5	4	3	2	1	0
ADCRH								

Caution When writing to A/D converter mode register 0 (ADM0) or Analog input channel specification register (ADS), the contents of the ADCRH register may become undefined. Read the conversion result following conversion completion before writing to the ADM0 or ADS registers. Using timing other than the above may cause an incorrect conversion result to be read.

16.3.7 Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted.
The ADS register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 16 - 12 Format of Analog input channel specification register (ADS) (1/2)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

• Select mode (ADMD = 0)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	1	0	0	0	ANI0	ANI0 pin
0	0	1	0	0	1	ANI1/AMP0O	ANI1/AMP0O pin
0	0	1	0	1	0	ANI2/ANX0	ANI2/AMP0N/ANX0 pin
0	0	1	0	1	1	ANI3/ANX1	ANI3/AMP0P/ANX1 pin
0	0	1	1	0	0	ANI4/AMP1O	ANI4/AMP1O pin
0	0	1	1	0	1	ANI5/ANX2	P42/ANI5/AMP1N/ANIX2 pin
0	0	1	1	1	0	ANI6/ANX3	P41/ANI6/AMP1P/ANX3 pin
0	0	1	1	1	1	ANI7/AMP2O	ANI7/AMP2O pin
0	1	0	0	0	0	ANI8/ANX4	P17/ANI8/AMP2N/ANX4 pin
0	1	0	0	0	1	ANI9/ANX5	P16/ANI9/AMP2P/ANX5 pin
0	1	0	0	1	0	—	SBIAS
1	0	0	0	0	1	—	Internal reference voltage for A/D converter (1.45 V)
Other than above						Setting prohibited	

(Cautions are listed on the next page.)

Figure 16 - 13 Format of Analog input channel specification register (ADS) (2/2)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

• Scan mode (ADMD = 1)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel			
						Scan 0	Scan 1	Scan 2	Scan 3
0	0	1	0	0	0	ANI0	ANI1/AMP0O	ANI2/ANX0	ANI3/ANX1
0	0	1	0	0	1	ANI1/AMP0O	ANI2/ANX0	ANI3/ANX1	ANI4/AMP1O
0	0	1	0	1	0	ANI2/ANX0	ANI3/ANX1	ANI4/AMP1O	ANI5/ANX2
0	0	1	0	1	1	ANI3/ANX1	ANI4/AMP1O	ANI5/ANX2	ANI6/ANX3
0	0	1	1	0	0	ANI4/AMP1O	ANI5/ANX2	ANI6/ANX3	ANI7/AMP2O
Other than above						Setting prohibited			

Caution 1. Be sure to clear bits 5 and 6 to 0.

Caution 2. Select by using port mode register 1 or 4 (PM1 or PM4) the input mode for a channel set as an analog input pin by using port mode control register 1 or 4 (PMC1 or PMC4).

Caution 3. Do not set the pin that is set to be a digital I/O pin by using the PMC1 or PMC4 register, by using the ADS register.

Caution 4. Rewrite the value of the ADISS bit while conversion is not running (ADCS = 0, ADCE = 0).

Caution 5. If the ADISS bit is set to 1, the internal reference voltage for A/D converter (1.45 V) cannot be used for the positive side reference voltage. The result of conversion immediately after the ADISS bit is set to 1 cannot be used. For the setting procedure, see 16.7.4 Setup when internal reference voltage for A/D converter is selected (example for software trigger mode and one-shot conversion mode).

Caution 6. Do not set the ADISS bit to 1 when shifting to STOP mode. When the ADISS bit is set to 1, the A/D converter reference voltage current (I_{ADREF}) indicated in 33.3.2 or 34.3.2 Supply current characteristics will be added.

Caution 7. For 32-pin products, do not set ADISS and ADS4 to ADS0 to 001000B.

16.3.8 Conversion result comparison upper limit setting register (ADUL)

This register is used to specify the setting for checking the upper limit of the A/D conversion results.

The A/D conversion results and ADUL register value are compared, and interrupt signal (INTAD) generation is controlled in the range specified for the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in **Figure 16 - 9**).

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

Reset signal generation sets this register to FFH.

Caution 1.When 10-bit resolution A/D conversion is selected, the higher eight bits of the 10-bit A/D conversion result register (ADCR) are compared with the ADUL and ADLL registers.

Caution 2. Only write new values to the ADUL and ADLL registers while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The setting of the ADUL and ADLL registers must be greater than that of the ADLL register.

Figure 16 - 14 Format of Conversion result comparison upper limit setting register (ADUL)

Address: F0011H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
ADUL	ADUL7	ADUL6	ADUL5	ADUL4	ADUL3	ADUL2	ADUL1	ADUL0

16.3.9 Conversion result comparison lower limit setting register (ADLL)

This register is used to specify the setting for checking the lower limit of the A/D conversion results.

The A/D conversion results and ADLL register value are compared, and interrupt signal (INTAD) generation is controlled in the range specified for the ADRCK bit of A/D converter mode register 2 (ADM2) (shown in **Figure 16 - 9**).

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

Reset signal generation clears this register to 00H.

Figure 16 - 15 Format of Conversion result comparison lower limit setting register (ADLL)

Address: F0012H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADLL	ADLL7	ADLL6	ADLL5	ADLL4	ADLL3	ADLL2	ADLL1	ADLL0

Caution 1. When 10-bit resolution A/D conversion is selected, the higher eight bits of the 10-bit A/D conversion result register (ADCR) are compared with the ADUL and ADLL registers.

Caution 2. Only write new values to the ADUL and ADLL registers while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 3. The setting of the ADUL and ADLL registers must be greater than that of the ADLL register.

16.3.10 A/D test register (ADTES)

This register is used to select the item subject to A/D conversion from the positive side reference voltage or negative side reference voltage for the converter, an analog input channel (ANIxx), or the internal reference voltage for A/D converter (1.45 V).
When using this register to test the converter, set as follows.
• For zero-scale measurement, select the negative side reference voltage as the item subject to A/D conversion.
• For full-scale measurement, select the positive side reference voltage as the item subject to A/D conversion.

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

Figure 16 - 16 Format of A/D test register (ADTES)

Address: F0013H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	0	0	ADTES1	ADTES0

ADTES1	ADTES0	Item subject to A/D conversion
0	0	One specified by using the analog input channel specification register (ADS)
0	1	Setting prohibited
1	0	Negative side reference voltage (AVss)
1	1	Positive side reference voltage. (Select by using the ADREFP1 and ADREFP0 bits of A/D converter mode register (ADM2).)

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

16.3.11 Registers controlling port function of analog input pins

Set up the registers for controlling the functions of the ports shared with the analog input pins of the A/D converter (port mode registers (PMxx) and port mode control registers (PMCxx)). For details, see **4.3.1 Port mode registers (PMxx)** and **4.3.6 Port mode control registers (PMCxx)**.

When using the ANI5, ANI6, ANI8, or ANI9 pin as an analog input pin of the A/D converter, set the bit corresponding to the pin in the port mode register (PMxx) and port mode control register (PMCxx) to 1.

16.4 A/D Converter Conversion Operations

The A/D converter conversion operations are described below.

<1> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<2> When sampling has been done for a certain time, the sample & hold circuit is placed in the hold state and the sampled voltage is held until the A/D conversion operation has ended.

<3> Bit 9 of the successive approximation register (SAR) is set. The series resistor string voltage tap is set to (1/2) AV_{REF} 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) AV_{REF}, the MSB bit of the SAR register remains set to 1. If the analog input is smaller than (1/2) AV_{REF}, the MSB bit is reset to 0.

<5> Next, bit 8 of the SAR register is automatically set to 1, and the operation proceeds to the next comparison. The series resistor string voltage tap is selected according to the preset value of bit 9, as described below.

- Bit 9 = 1: (3/4) AV_{REF}
- Bit 9 = 0: (1/4) AV_{REF}

The voltage tap and sampled voltage are compared and bit 8 of the SAR register is manipulated as follows.

- Sampled voltage ≥ Voltage tap: Bit 8 = 1
- Sampled voltage < Voltage tap: Bit 8 = 0

<6> Comparison is continued in this way up to bit 0 of the SAR register.

<7> Upon completion of the comparison of 10 bits, an effective digital result value remains in the SAR register, and the result value is transferred to the A/D conversion result register (ADCR, ADCRH) and then latched [Note 1]. At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.

<8> Repeat steps <1> to <7>, until the ADCS bit is cleared to 0 [Note 2]. To stop the A/D converter, clear the ADCS bit to 0.

Note 1. If the A/D conversion result is outside the A/D conversion result range specified by the ADRCK bit and the ADUL and ADLL registers (see **Figure 16 - 9**), the A/D conversion result interrupt request signal is not generated and no A/D conversion results are stored in the ADCR and ADCRH registers.

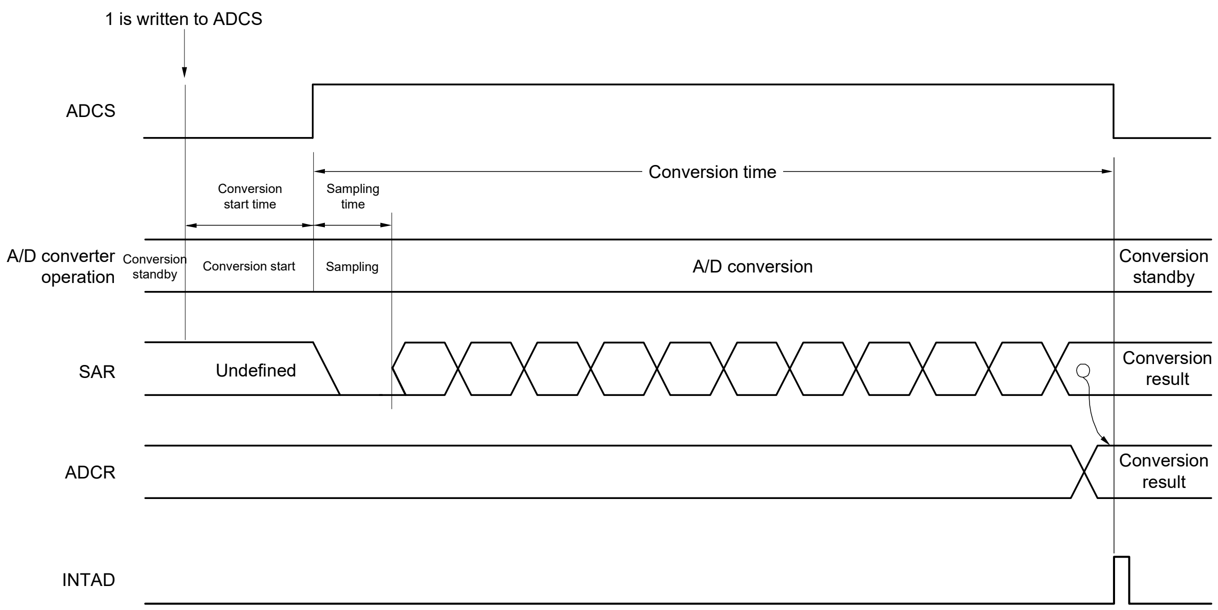
Note 2. While in the sequential conversion mode, the ADCS flag is not automatically cleared to 0. This flag is not automatically cleared to 0 while in the one-shot conversion mode of the hardware trigger no-wait mode, either. Instead, 1 is retained.

Remark 1. Two types of the A/D conversion result registers are available.

- ADCR register (16 bits): Store 10-bit A/D conversion value
- ADCRH register (8 bits): Store 8-bit A/D conversion value

Remark 2. AV_{REF}: The positive side reference voltage of the A/D converter. This can be selected from the internal reference voltage for A/D converter (1.45 V) and AV_{DD}.

Figure 16 - 17 Conversion Operation of A/D Converter (Software Trigger Mode)



In one-shot conversion mode, the ADCS bit is automatically cleared to 0 after completion of A/D conversion.

In sequential conversion mode, A/D conversion operations proceed continuously until the software clears bit 7 (ADCS) of the A/D converter mode register 0 (ADM0) to 0.

Writing to the analog input channel specification register (ADS) during A/D conversion interrupts the current conversion after which A/D conversion of the analog input specified by the ADS register proceeds. Data from the A/D conversion that was in progress are discarded.

Reset signal generation clears the A/D conversion result register (ADCR, ADCRH) to 0000H or 00H.

16.5 Input Voltage and Conversion Results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI9) and the theoretical A/D conversion result (stored in the 10-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$\mathrm{SAR} = \mathrm{INT}\left(\frac{V_{AIN}}{AV_{REF}} \times 1024 + 0.5\right)$$

$$\mathrm{ADCR} = \mathrm{SAR} \times 64$$

or

$$\left(\frac{\mathrm{ADCR}}{64} - 0.5\right) \times \frac{AV_{REF}}{1024} \leq V_{AIN} < \left(\frac{\mathrm{ADCR}}{64} + 0.5\right) \times \frac{AV_{REF}}{1024}$$

where, INT( ): Function which returns integer part of value in parentheses

V_{AIN}: Analog input voltage

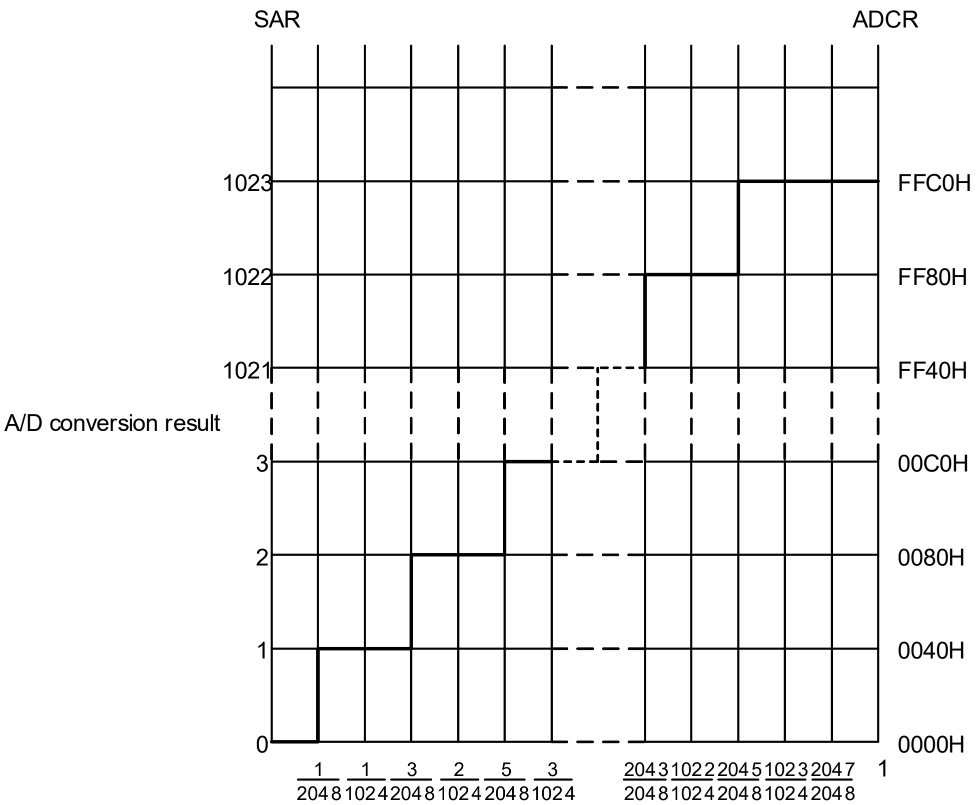
AV_{REF}: AV_{REF} pin voltage

ADCR: A/D conversion result register (ADCR) value

SAR: Successive approximation register

Figure 16 - 18 shows the Relationship Between Analog Input Voltage and A/D Conversion Result.

Figure 16 - 18 Relationship Between Analog Input Voltage and A/D Conversion Result



Remark AV_{REF}: The + side reference voltage of the A/D converter. This can be selected from the internal reference voltage for A/D converter (1.45 V) and AV_{DD}.

16.6 A/D Converter Operation Modes

The operation of each A/D converter mode is described below. In addition, the procedure for specifying each mode is described in **16.7 A/D Converter Setup Flowchart**.

16.6.1 Software trigger mode (select mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform the A/D conversion of the analog input specified by the analog input channel specification register (ADS).

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts.

<4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

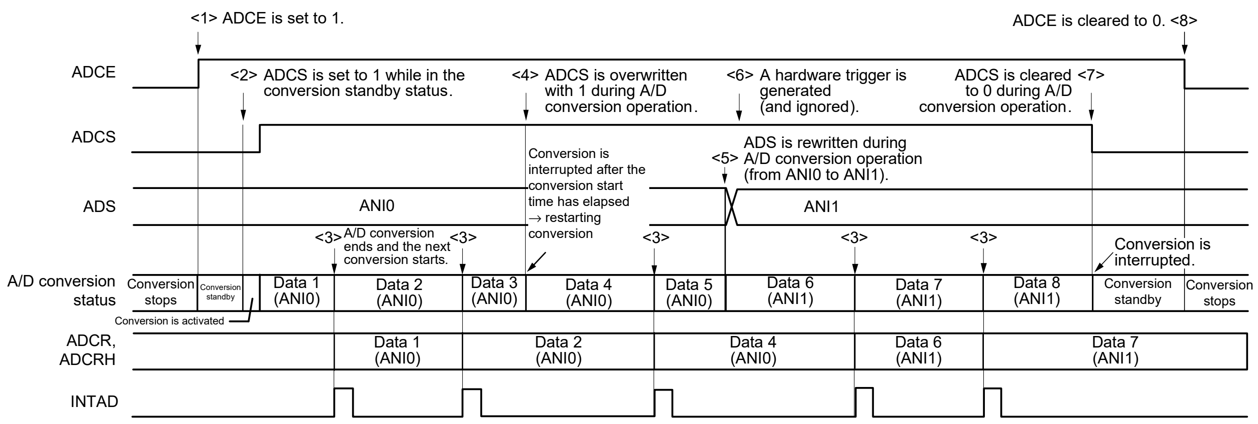
<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<6> Even if a hardware trigger is input during conversion operation, A/D conversion does not start.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 16 - 19 Example of Software Trigger Mode (Select Mode, Sequential Conversion Mode) Operation Timing



16.6.2 Software trigger mode (select mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform the A/D conversion of the analog input specified by the analog input channel specification register (ADS).

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the system enters the A/D conversion standby status.

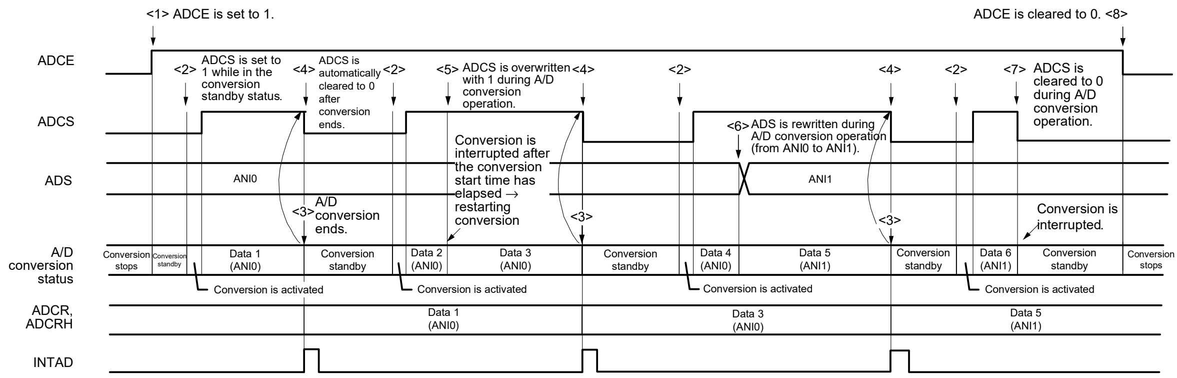
<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby status.

Figure 16 - 20 Example of Software Trigger Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



16.6.3 Software trigger mode (scan mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts (until all four channels are finished).

<4> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

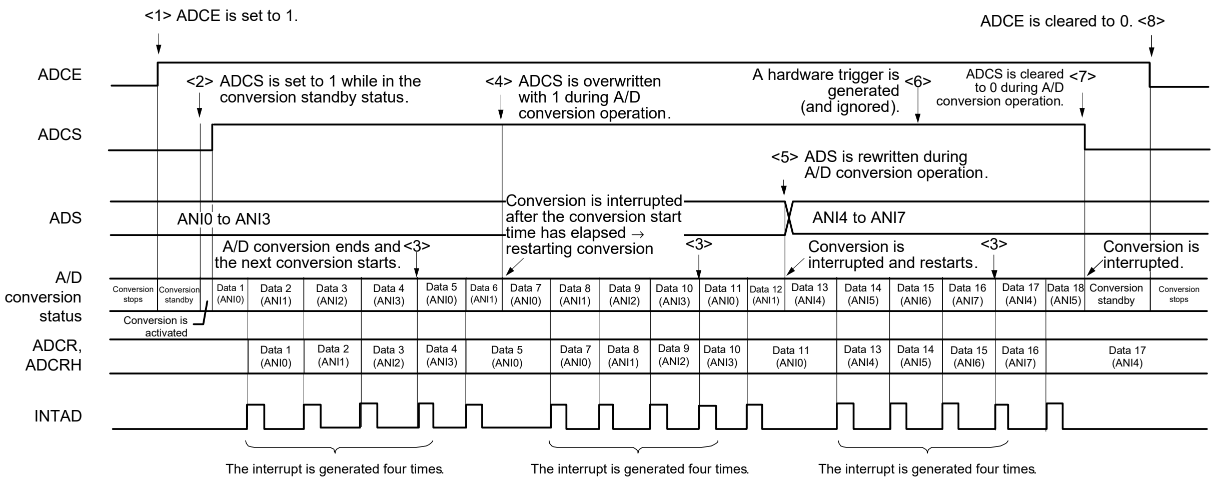
<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<6> Even if a hardware trigger is input during conversion operation, A/D conversion does not start.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 16 - 21 Example of Software Trigger Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



16.6.4 Software trigger mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to perform A/D conversion on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion of the four channels ends, the ADCS bit is automatically cleared to 0, and the system enters the A/D conversion standby status.

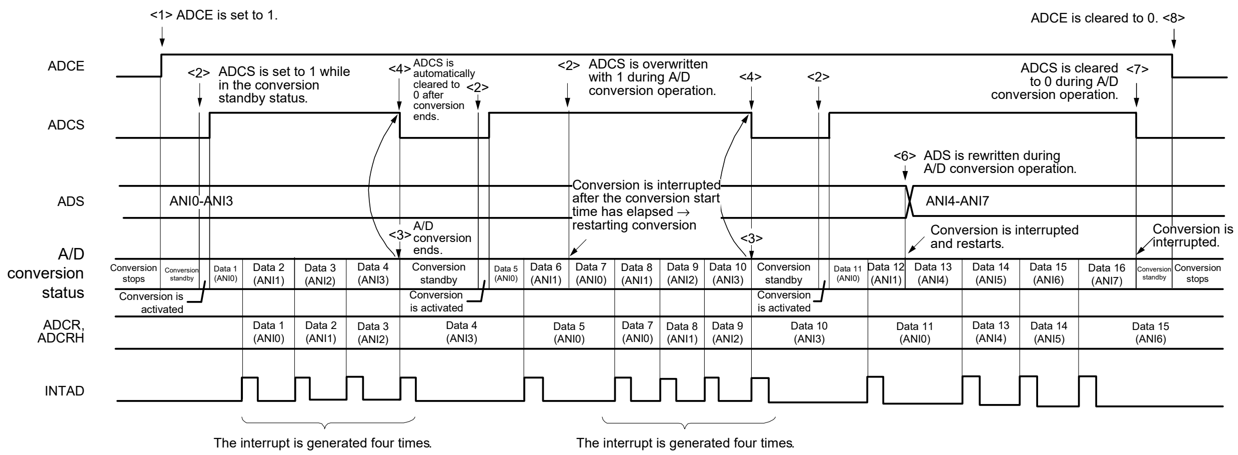
<5> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status.

<8> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start. In addition, A/D conversion does not start even if a hardware trigger is input while in the A/D conversion standby status.

Figure 16 - 22 Example of Software Trigger Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing



16.6.5 Hardware trigger no-wait mode (select mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).

<4> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts.

<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

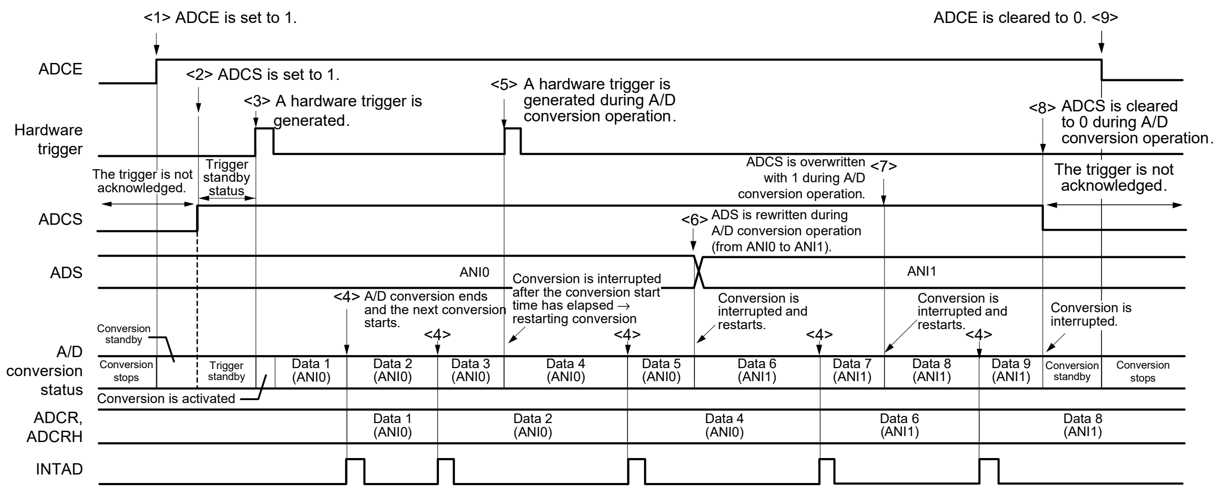
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<9> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 16 - 23 Example of Hardware Trigger No-Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



16.6.6 Hardware trigger no-wait mode (select mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS).

<4> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<5> After A/D conversion ends, the ADCS bit remains set to 1, and the system enters the A/D conversion standby status.

<6> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

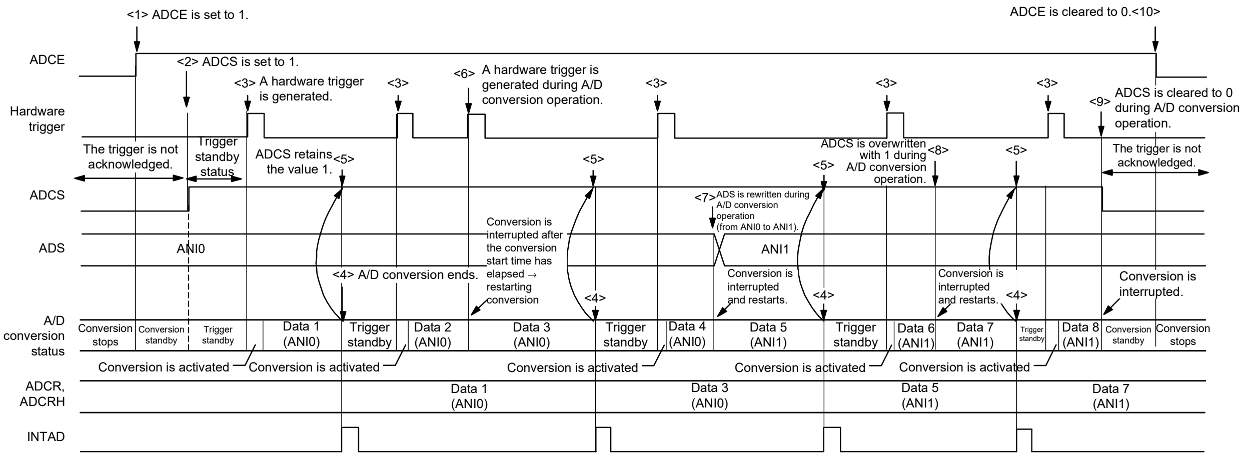
<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<9> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<10>When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 16 - 24 Example of Hardware Trigger No-Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



16.6.7 Hardware trigger no-wait mode (scan mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<4> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts.

<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

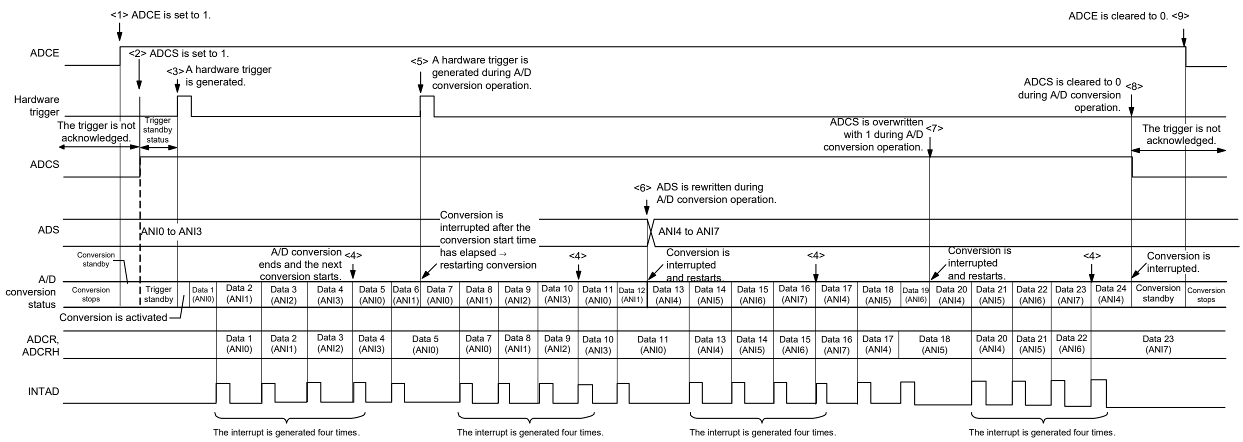
<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<9> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCE = 0, specifying 1 for ADCS is ignored and A/D conversion does not start.

Figure 16 - 25 Example of Hardware Trigger No-Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



16.6.8 Hardware trigger no-wait mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> After the software counts up to the stabilization wait time (1 μs), the ADCS bit of the ADM0 register is set to 1 to place the system in the hardware trigger standby status (and conversion does not start at this stage). Note that, while in this status, A/D conversion does not start even if ADCS is set to 1.

<3> If a hardware trigger is input while ADCS = 1, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<4> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<5> After A/D conversion of the four channels ends, the ADCS bit remains set to 1, and the system enters the A/D conversion standby status.

<6> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

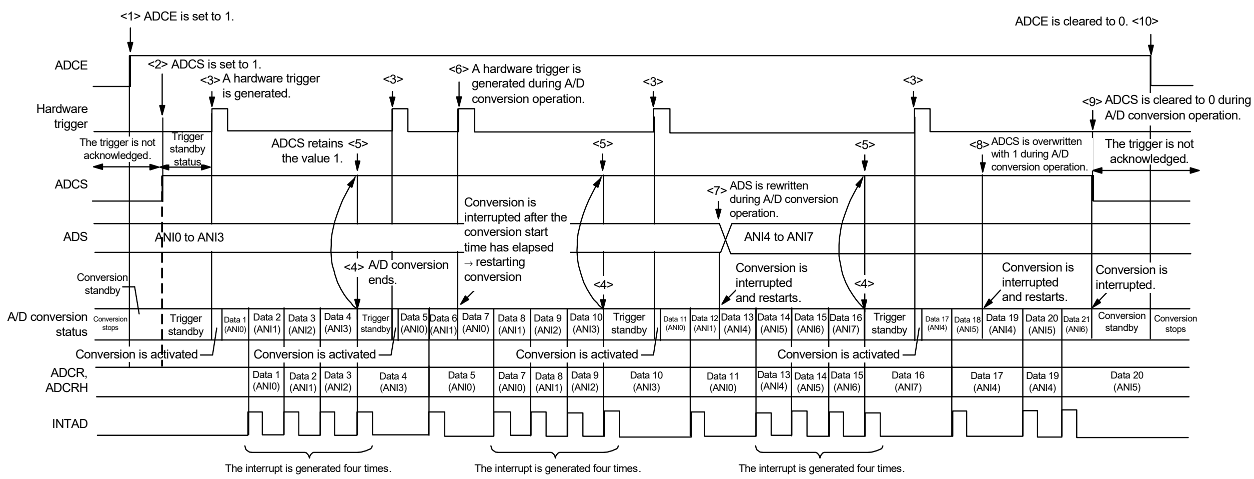
<7> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<8> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<9> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, and the system enters the A/D conversion standby status. However, the A/D converter does not stop in this status.

<10> When ADCE is cleared to 0 while in the A/D conversion standby status, the A/D converter enters the stop status. When ADCS = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 16 - 26 Example of Hardware Trigger No-Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing



16.6.9 Hardware trigger wait mode (select mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the hardware trigger standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input.

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion ends, the next A/D conversion immediately starts. (At this time, no hardware trigger is necessary.)

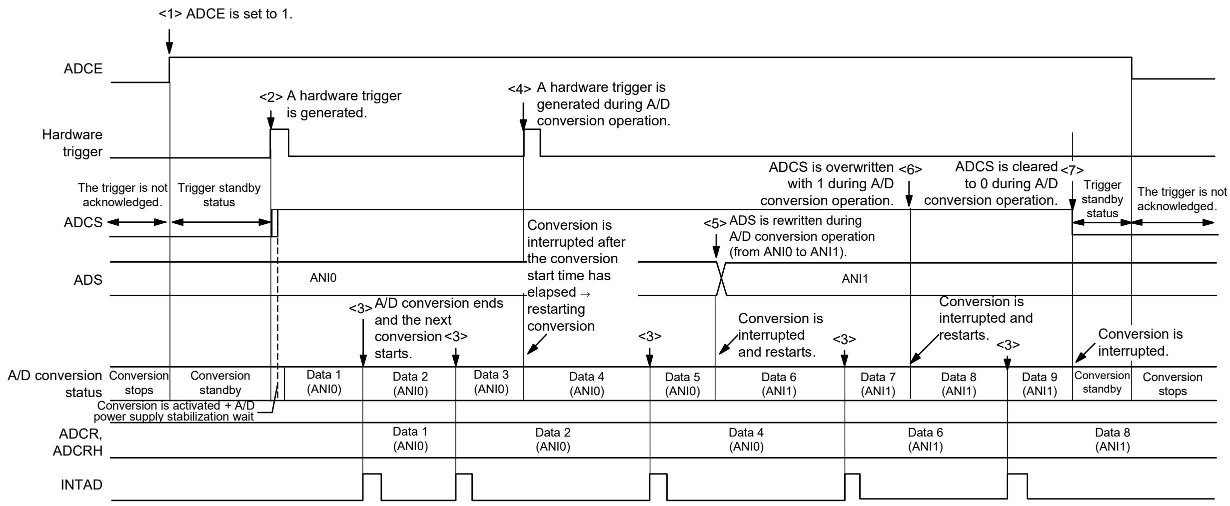
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<6> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 16 - 27 Example of Hardware Trigger Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing



16.6.10 Hardware trigger wait mode (select mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the hardware trigger standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the analog input specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input.

<3> When A/D conversion ends, the conversion result is stored in the A/D conversion result register (ADCR, ADCRH), and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the stop status.

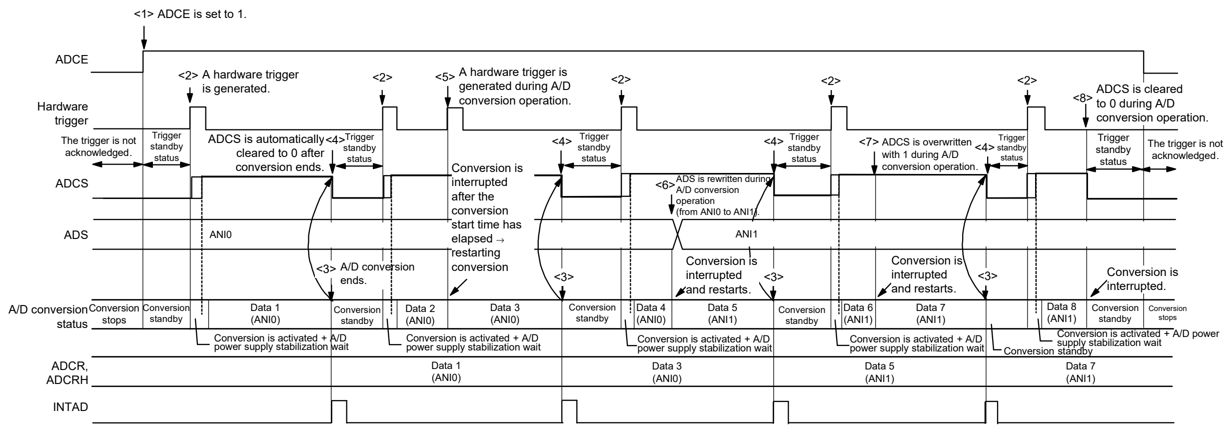
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the analog input respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is initialized.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 16 - 28 Example of Hardware Trigger Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing



16.6.11 Hardware trigger wait mode (scan mode, sequential conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input. A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated. After A/D conversion of the four channels ends, the A/D conversion of the channel following the specified channel automatically starts.

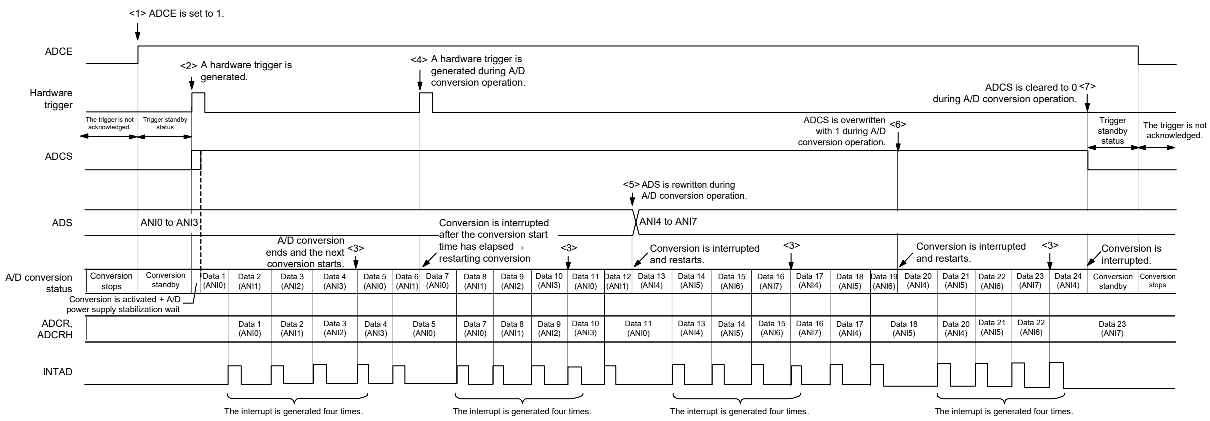
<4> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<5> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<6> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<7> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 16 - 29 Example of Hardware Trigger Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing



16.6.12 Hardware trigger wait mode (scan mode, one-shot conversion mode)

<1> In the stop status, the ADCE bit of A/D converter mode register 0 (ADM0) is set to 1, and the system enters the A/D conversion standby status.

<2> If a hardware trigger is input while in the hardware trigger standby status, A/D conversion is performed on the four analog input channels specified by scan 0 to scan 3, which are specified by the analog input channel specification register (ADS). The ADCS bit of the ADM0 register is automatically set to 1 according to the hardware trigger input. A/D conversion is performed on the analog input channels in order, starting with that specified by scan 0.

<3> A/D conversion is sequentially performed on the four analog input channels, the conversion results are stored in the A/D conversion result register (ADCR, ADCRH) each time conversion ends, and the A/D conversion end interrupt request signal (INTAD) is generated.

<4> After A/D conversion ends, the ADCS bit is automatically cleared to 0, and the A/D converter enters the stop status.

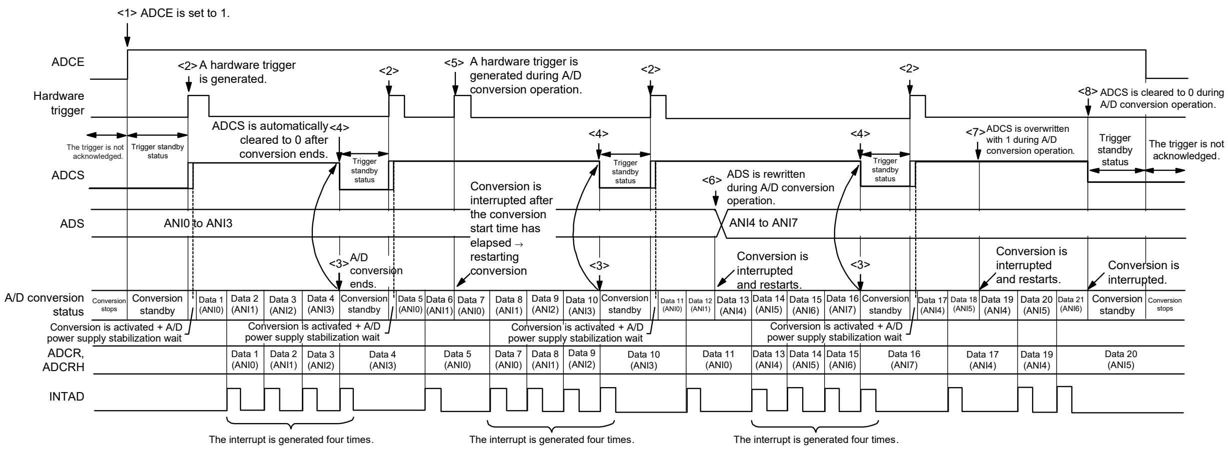
<5> If a hardware trigger is input during conversion operation, the current A/D conversion is interrupted, and conversion restarts at the first channel. The partially converted data is discarded.

<6> When the value of the ADS register is rewritten or overwritten during conversion operation, the current A/D conversion is interrupted, and A/D conversion is performed on the first channel respecified by the ADS register. The partially converted data is discarded.

<7> When ADCS is overwritten with 1 during conversion operation, the current A/D conversion is interrupted, and conversion restarts. The partially converted data is discarded.

<8> When ADCS is cleared to 0 during conversion operation, the current A/D conversion is interrupted, the system enters the hardware trigger standby status, and the A/D converter enters the stop status. When ADCE = 0, inputting a hardware trigger is ignored and A/D conversion does not start.

Figure 16 - 30 Example of Hardware Trigger Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing

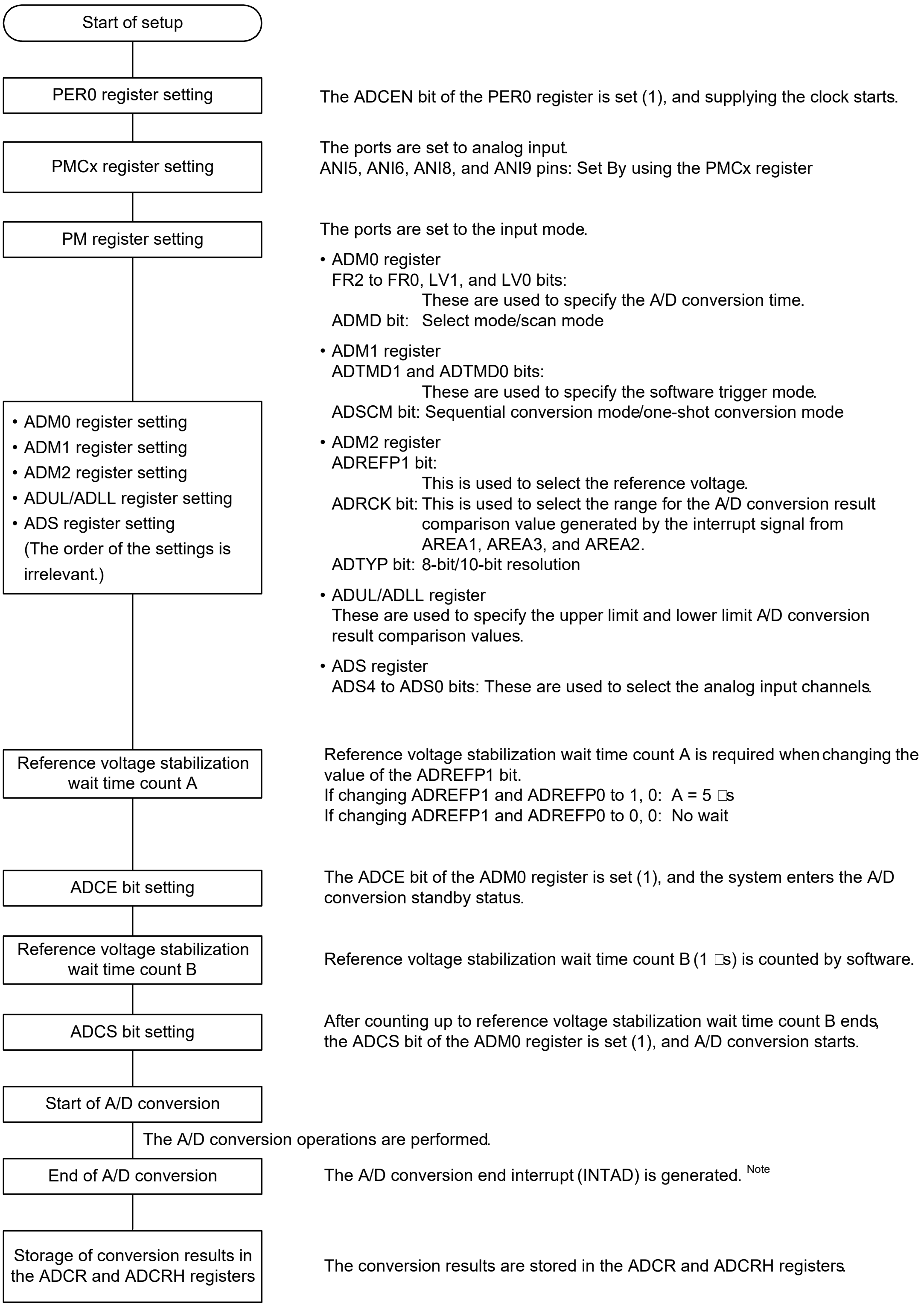


16.7 A/D Converter Setup Flowchart

The A/D converter setup flowchart in each operation mode is described below.

16.7.1 Setting up software trigger mode

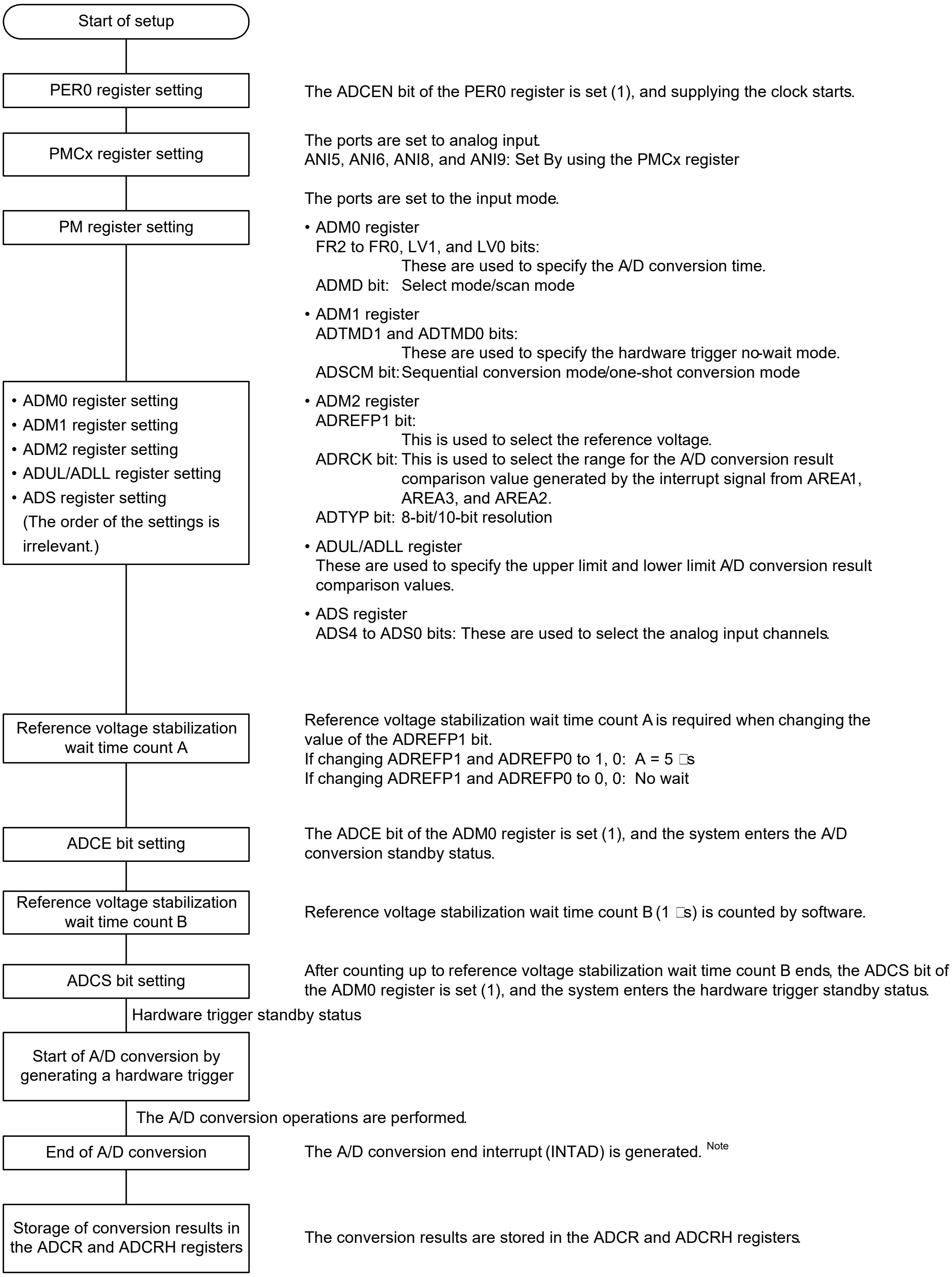
Figure 16 - 31 Setting up Software Trigger Mode



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH register.

16.7.2 Setting up hardware trigger no-wait mode

Figure 16 - 32 Setting up Hardware Trigger No-Wait Mode



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH register.

16.7.3 Setting up hardware trigger wait mode

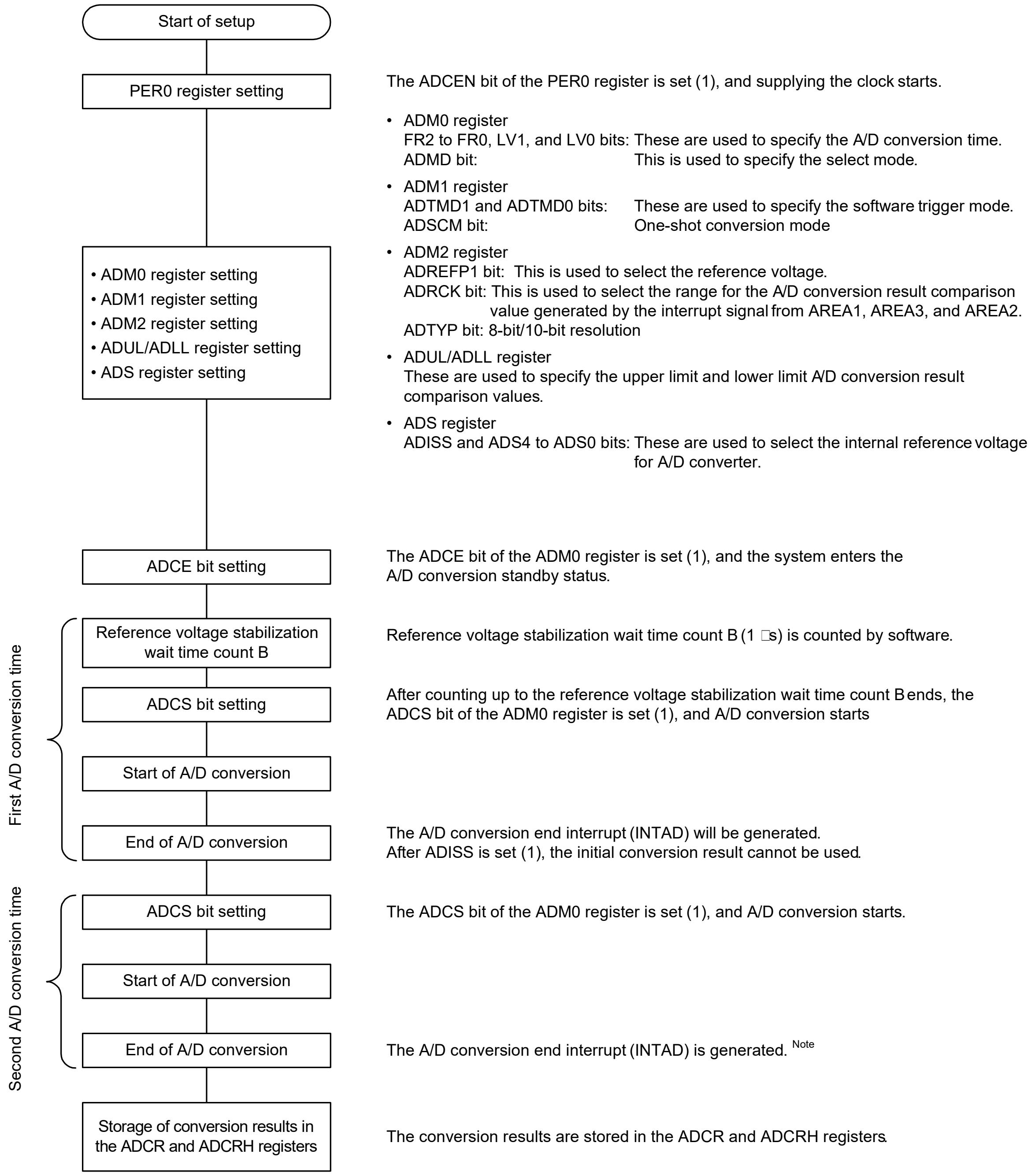
Figure 16 - 33 Setting up Hardware Trigger Wait Mode

Step	Description
Start of setup	
PER0 register setting	The ADCEN bit of the PER0 register is set (1), and supplying the clock starts.
PMCx register setting	The ports are set to analog input. ANI5, ANI6, ANI8, and ANI9 pins: Set By using the PMCx register
PM register setting	The ports are set to the input mode.
• ADM0 register setting • ADM1 register setting • ADM2 register setting • ADUL/ADLL register setting • ADS register setting (The order of the settings is irrelevant.)	• ADM0 register FR2 to FR0, LV1, and LV0 bits: These are used to specify the A/D conversion time. ADMD bit: Select mode/scan mode • ADM1 register ADTMD1 and ADTMD0 bits: These are used to specify the hardware trigger wait mode. ADSCM bit: Sequential conversion mode/one-shot conversion mode ADTRS1 and ADTRS0 bits: These are used to select the hardware trigger signal. • ADM2 register ADREFP1 bit: This is used to select the reference voltage. ADRCK bit: This is used to select the range for the A/D conversion result comparison value generated by the interrupt signal from AREA1, AREA3, and AREA2. ADTYP bit: 8-bit/10-bit resolution • ADUL/ADLL register These are used to specify the upper limit and lower limit A/D conversion result comparison values. • ADS register ADS4 to ADS0 bits: These are used to select the analog input channels.
Reference voltage stabilization wait time count A	Reference voltage stabilization wait time count A is required when changing the value of the ADREFP1 bit. If changing ADREFP1 and ADREFP0 to 1, 0: A = 5 µs If changing ADREFP1 and ADREFP0 to 0, 0: No wait
ADCE bit setting	The ADCE bit of the ADM0 register is set (1), and the system enters the A/D conversion standby status.
Hardware trigger generation	
Stabilization wait time for A/D power supply	The system automatically counts up to the stabilization wait time for A/D power supply.
Start of A/D conversion	After counting up to the stabilization wait time for A/D power supply ends, A/D conversion starts.
(The A/D conversion operations are performed.)	
End of A/D conversion	The A/D conversion end interrupt (INTAD) is generated. Note
Storage of conversion results in the ADCR and ADCRH registers	The conversion results are stored in the ADCR and ADCRH registers.

Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH register.

16.7.4 Setup when internal reference voltage for A/D converter is selected (example for software trigger mode and one-shot conversion mode)

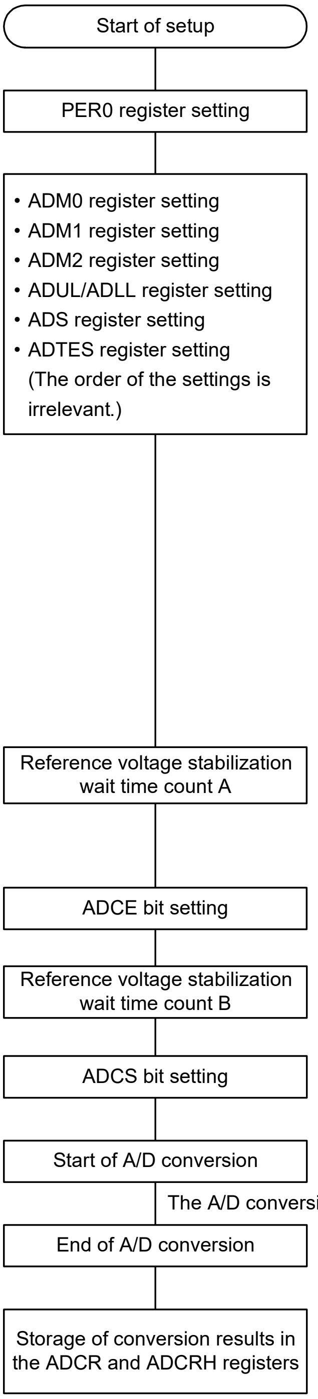
Figure 16 - 34 Setup when internal reference voltage for A/D converter is selected



Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH register.

16.7.5 Setting up test mode

Figure 16 - 35 Setting up Test Mode



PER0 register setting: The ADCEN bit of the PER0 register is set (1), and supplying the clock starts.

- ADM0 register
  FR2 to FR0, LV1, and LV0 bits: These are used to specify the A/D conversion time.
  ADMD bit: This is used to specify the select mode
- ADM1 register
  ADTMD1 and ADTMD0 bits: These are used to specify the software trigger mode.
  ADSCM bit: This is used to specify the one-shot conversion mode.
- ADM2 register
  ADREFP1 bit: This is used to select the reference voltage.
  ADRCK bit: This is used to set the range for the A/D conversion result comparison value generated by the interrupt signal to AREA2.
  ADTYP bit: This is used to specify 10-bit resolution.
- ADUL/ADLL register
  These set ADUL to FFH and ADLL to 00H (initial values).
- ADS register
  ADS4 to ADS0 bits: These are used to set to ANI0.
- ADTES register
  ADTES1, ADTES0 bits: AV_{SS}/AV_{DD}/internal reference voltage for A/D converter

Reference voltage stabilization wait time count A: Reference voltage stabilization wait time count A is required when changing the value of the ADREFP1 bit.
If changing ADREFP1 and ADREFP0 to 1, 0: A = 5 μs
If changing ADREFP1 and ADREFP0 to 0, 0: No wait

ADCE bit setting: The ADCE bit of the ADM0 register is set (1), and the system enters the A/D conversion standby status.

Reference voltage stabilization wait time count B: Reference voltage stabilization wait time count B (1 μs) is counted by software.

ADCS bit setting: After counting up to reference voltage stabilization wait time count B ends, the ADCS bit of the ADM0 register is set (1), and A/D conversion starts.

Start of A/D conversion: The A/D conversion operations are performed.

End of A/D conversion: The A/D conversion end interrupt (INTAD) is generated. Note

Storage of conversion results: The conversion results are stored in the ADCR and ADCRH registers.

Note Depending on the settings of the ADRCK bit and ADUL/ADLL register, there is a possibility of no interrupt signal being generated. In this case, the results are not stored in the ADCR, ADCRH register.

Caution For the procedure for testing the A/D converter, see 27.3.8 A/D test function.

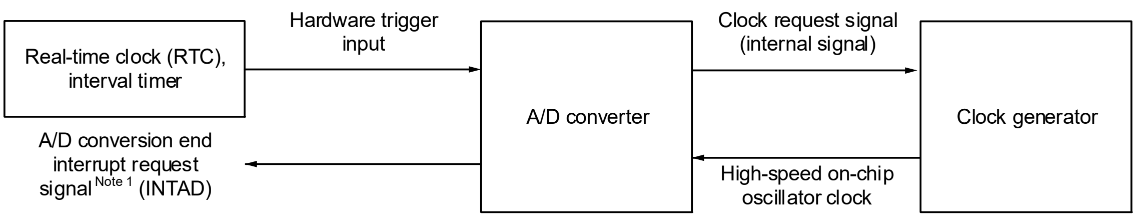
16.8 SNOOZE Mode Function

In the SNOOZE mode, A/D conversion is triggered by inputting a hardware trigger in the STOP mode. Normally, A/D conversion is stopped while in the STOP mode, but, by using the SNOOZE mode, A/D conversion can be performed without operating the CPU by inputting a hardware trigger. This is effective for reducing the operation current.
If the A/D conversion result range is specified using the ADUL and ADLL registers, A/D conversion results can be judged at a certain interval of time in SNOOZE mode. Using this function enables power supply voltage monitoring and input key judgment based on A/D inputs.

In the SNOOZE mode, only the following two conversion modes can be used:
• Hardware trigger wait mode (select mode, one-shot conversion mode)
• Hardware trigger wait mode (scan mode, one-shot conversion mode)

Caution That the SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected for f_{CLK}.

Figure 16 - 36 Block Diagram When Using SNOOZE Mode Function



When using the SNOOZE mode function, specify the initial hardware trigger and wait mode settings for each register before shifting to the STOP mode. (For details about these settings, see **16.7.3 Setting up hardware trigger wait mode** Note 2.) At this time, bit 2 (AWC) of A/D converter mode register 2 (ADM2) is set to 1. After the initial settings are specified, bit 0 (ADCE) of A/D converter mode register 0 (ADM0) is set to 1.
If a hardware trigger is input after switching to the STOP mode, the high-speed on-chip oscillator clock is supplied to the A/D converter. After supplying this clock, the system automatically counts up to the A/D power supply stabilization wait time, and then A/D conversion starts.
The SNOOZE mode operation after A/D conversion ends differs depending on whether an interrupt signal is generated Note 1.

Note 1. Depending on the setting of the A/D conversion result comparison function (ADRCK bit, ADUL/ADLL register), there is a possibility of no interrupt signal being generated.
Note 2. Be sure to set the ADM1 register to E1H, E2H or E3H.

Remark The hardware trigger is event selected by ELC, INTRTC or INTIT.
Specify the hardware trigger by using the A/D Converter Mode Register 1 (ADM1).

(1) If an interrupt is generated after A/D conversion ends

If the A/D conversion result value is inside the range of values specified by the A/D conversion result comparison function (which is set up by using the ADRCK bit and ADUL/ADLL register), the A/D conversion end interrupt request signal (INTAD) is generated.

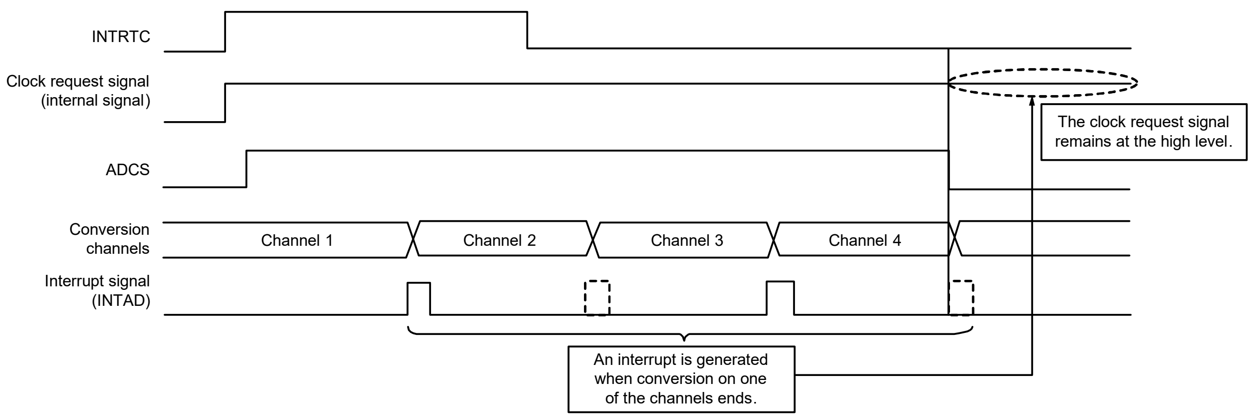
- While in the select mode

When A/D conversion ends and an A/D conversion end interrupt request signal (INTAD) is generated, the A/D converter returns to normal operation mode from SNOOZE mode. At this time, be sure to clear bit 2 (AWC = 0: SNOOZE mode release) of the A/D converter mode register 2 (ADM2). If the AWC bit is left set to 1, A/D conversion will not start normally in the subsequent SNOOZE or normal operation mode.

- While in the scan mode

If even one A/D conversion end interrupt request signal (INTAD) is generated during A/D conversion of the four channels, the clock request signal remains at the high level, and the A/D converter switches from the SNOOZE mode to the normal operation mode. At this time, be sure to clear bit 2 (AWC = 0: SNOOZE mode release) of A/D converter mode register 2 (ADM2) to 0. If the AWC bit is left set to 1, A/D conversion will not start normally in the subsequent SNOOZE or normal operation mode.

Figure 16 - 37 Operation Example When Interrupt Is Generated After A/D Conversion Ends (While in Scan Mode)



(2) If no interrupt is generated after A/D conversion ends

If the A/D conversion result value is outside the range of values specified by the A/D conversion result comparison function (which is set up by using the ADRCK bit and ADUL/ADLL register), the A/D conversion end interrupt request signal (INTAD) is not generated.

- While in the select mode

  If the A/D conversion end interrupt request signal (INTAD) is not generated after A/D conversion ends, the clock request signal (an internal signal) is automatically set to the low level, and supplying the high-speed on-chip oscillator clock stops. If a hardware trigger is input later, A/D conversion work is again performed in the SNOOZE mode.

- While in the scan mode

  If the A/D conversion end interrupt request signal (INTAD) is not generated even once during A/D conversion of the four channels, the clock request signal (an internal signal) is automatically set to the low level after A/D conversion of the four channels ends, and supplying the high-speed on-chip oscillator clock stops. If a hardware trigger is input later, A/D conversion work is again performed in the SNOOZE mode.

Figure 16 - 38 Operation Example When No Interrupt Is Generated After A/D Conversion Ends (While in Scan Mode)

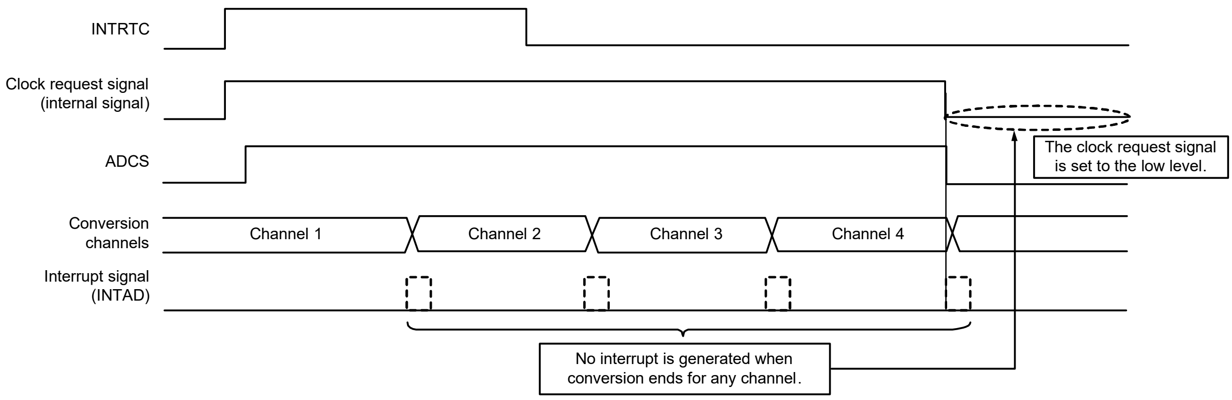
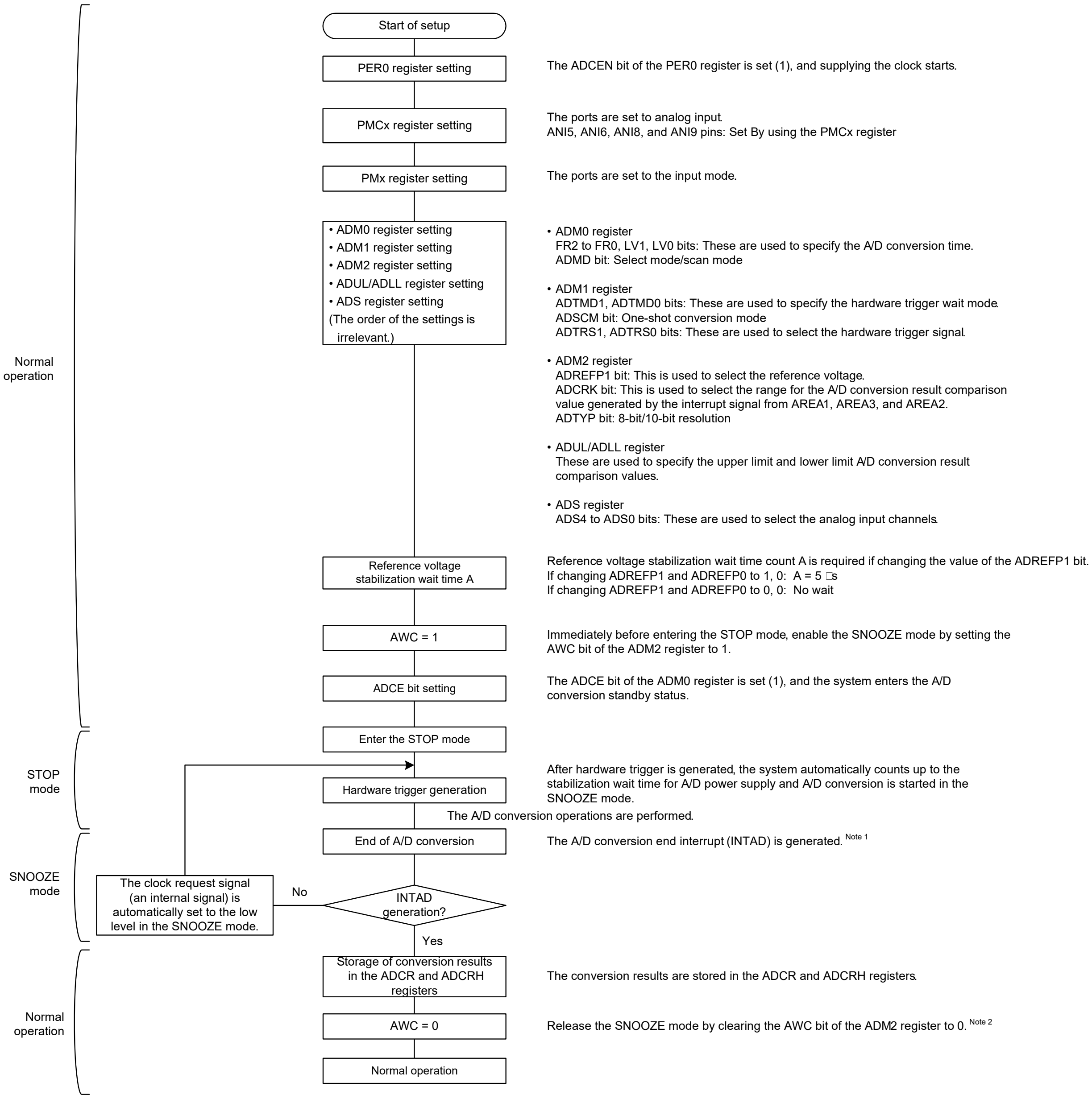


Figure 16 - 39 Flowchart for Setting up SNOOZE Mode



Note 1. If the A/D conversion end interrupt request signal (INTAD) is not generated by setting ADRCK bit and ADUL/ADLL register, the result is not stored in the ADCR and ADCRH registers.
The system enters the STOP mode again. If a hardware trigger is input later, A/D conversion operation is again performed in the SNOOZE mode.

Note 2. If the AWC bit is left set to 1, A/D conversion will not start normally in spite of the subsequent SNOOZE or normal operation mode. Be sure to clear the AWC bit to 0.

16.9 How to Read A/D Converter Characteristics Table

Here, special terms unique to the A/D converter are explained.

(1) Resolution

This is the minimum analog input voltage that can be identified. That is, the percentage of the analog input voltage per bit of digital output is called 1LSB (Least Significant Bit). The percentage of 1LSB with respect to the full scale is expressed by %FSR (Full Scale Range).

1LSB is as follows when the resolution is 10 bits.

$1\ LSB = 1/2^{10} = 1/1024$

$= 0.098\%FSR$

Accuracy has no relation to resolution, but is determined by overall error.

(2) Overall error

This shows the maximum error value between the actual measured value and the theoretical value.

Zero-scale error, full-scale error, integral linearity error, and differential linearity errors that are combinations of these express the overall error.

Note that the quantization error is not included in the overall error in the characteristics table.

(3) Quantization error

When analog values are converted to digital values, a ±1/2LSB error naturally occurs. In an A/D converter, an analog input voltage in a range of ±1/2LSB is converted to the same digital code, so a quantization error cannot be avoided.

Note that the quantization error is not included in the overall error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

Figure 16 - 40 Overall Error

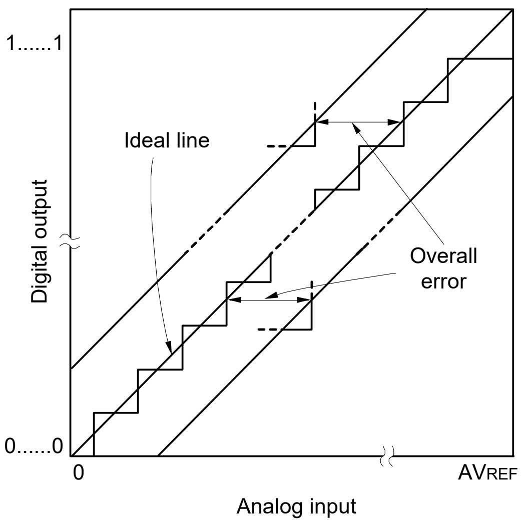
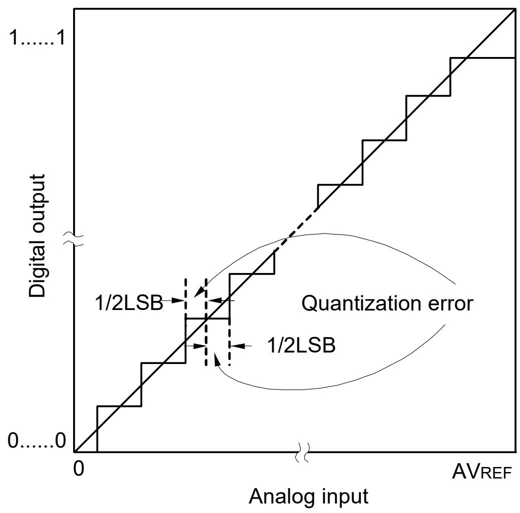


Figure 16 - 41 Quantization Error



(4) Zero-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (1/2LSB) when the digital output changes from 0......000 to 0......001.

If the actual measurement value is greater than the theoretical value, it shows the difference between the actual measurement value of the analog input voltage and the theoretical value (3/2LSB) when the digital output changes from 0……001 to 0……010.

(5) Full-scale error

This shows the difference between the actual measurement value of the analog input voltage and the theoretical value (Full-scale – 3/2LSB) when the digital output changes from 1......110 to 1......111.

(6) Integral linearity error

This shows the degree to which the conversion characteristics deviate from the ideal linear relationship. It expresses the maximum value of the difference between the actual measurement value and the ideal straight line when the zero-scale error and full-scale error are 0.

(7) Differential linearity error

While the ideal width of code output is 1LSB, this indicates the difference between the actual measurement value and the ideal value.

Figure 16 - 42 Zero-Scale Error

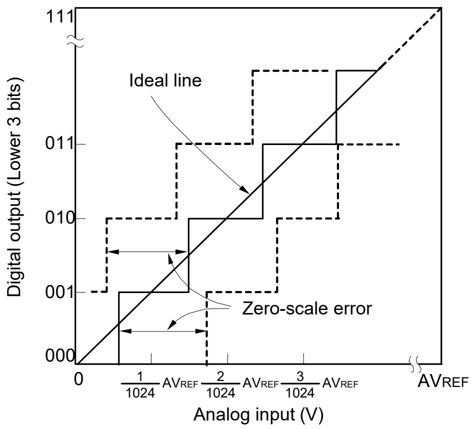


Figure 16 - 43 Full-Scale Error

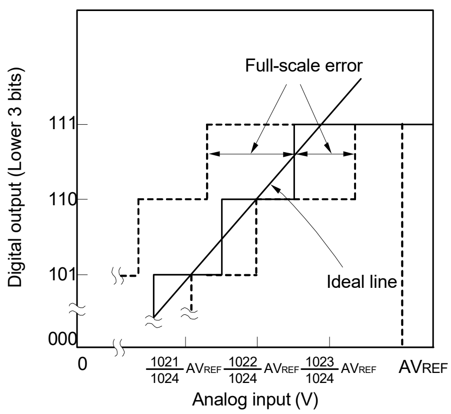


Figure 16 - 44 Integral Linearity Error

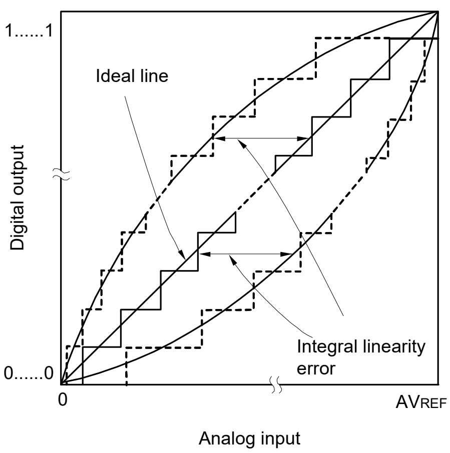
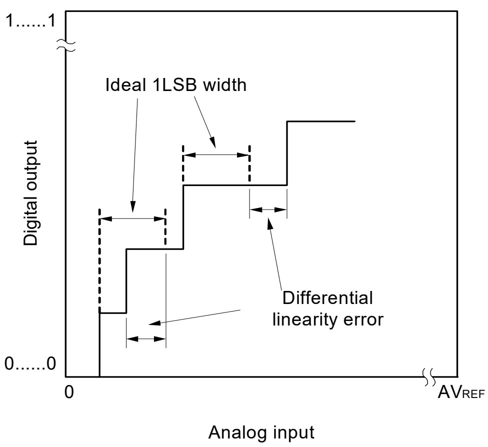


Figure 16 - 45 Differential Linearity Error

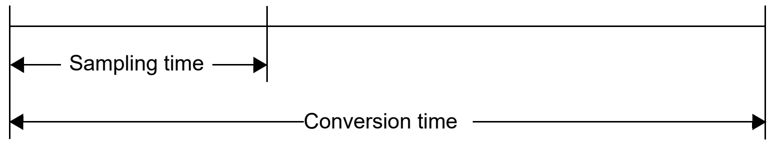


(8) Conversion time

This expresses the time from the start of sampling to when the digital output is obtained.
The sampling time is included in the conversion time in the characteristics table.

(9) Sampling time

This is the time the analog switch is turned on for the analog voltage to be sampled by the sample & hold circuit.



16.10 Cautions for A/D Converter

(1) Operating current in STOP mode

Shift to STOP mode after stopping the A/D converter (by setting bit 7 (ADCS) of A/D converter mode register 0 (ADM0) to 0). The operating current can be reduced by setting bit 0 (ADCE) of the ADM0 register to 0 at the same time.

To restart from the standby status, clear bit 0 (ADIF) of interrupt request flag register 1H (IF1H) to 0 and start operation.

(2) Input range of ANI0 to ANI9 pins

Observe the rated range of the ANI0 to ANI9 pins input voltage. If a voltage exceeding AV_{DD} or below AV_{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.

When the internal reference voltage for A/D converter (1.45 V) is selected reference voltage for the positive side of the A/D converter, do not input voltage exceeding the internal reference voltage to a pin selected by the ADS register. However, it is no problem that a pin not selected by the ADS register is input voltage exceeding the internal reference voltage.

(3) Conflicting operations

<1> Conflict between the A/D conversion result register (ADCR, ADCRH) write and the ADCR or ADCRH register read by instruction upon the end of conversion

The ADCR or ADCRH register read has priority. After the read operation, the new conversion result is written to the ADCR or ADCRH registers.

<2> Conflict between writing to the ADCR or ADCRH register and writing to A/D converter mode register 0 (ADM0) or analog input channel specification register (ADS) at the end of A/D conversion

Writing to the ADM0 and ADS registers has priority. Writing to the ADCR or ADCRH register is not performed, nor is the conversion end interrupt signal (INTAD) generated.

(4) Noise countermeasures

To maintain the 10-bit resolution, attention must be paid to noise input to the AV_{DD} and ANI0 to ANI9 pins.

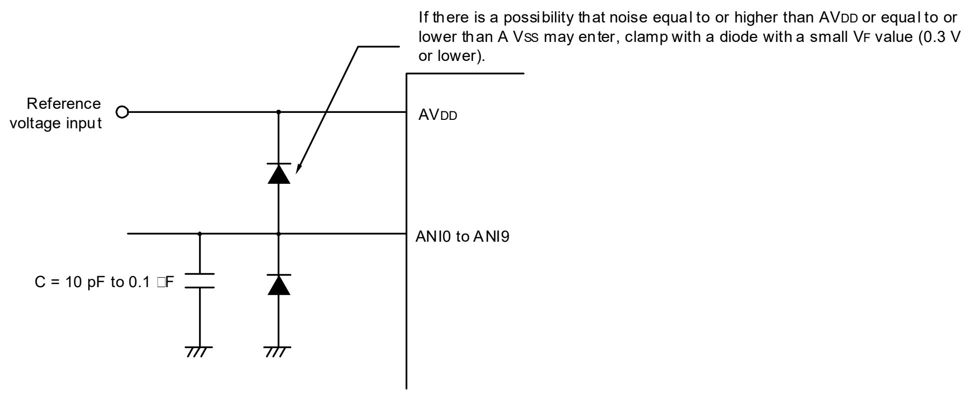
<1> Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.

<2> The higher the output impedance of the analog input source, the greater the influence. To reduce the noise, connecting external capacitor as shown in Figure 16 - 46 is recommended.

<3> Do not switch these pins with other pins during conversion.

<4> The accuracy is improved if the HALT mode is set immediately after the start of conversion.

Figure 16 - 46 Analog Input Pin Connection



(5) Analog input (ANIn) pins

<1> The analog input pins (ANI5, ANI6, ANI8, and ANI9) are also used as input port pins (P42, P41, P17, and P16).
When A/D conversion is performed with any of the ANI5, ANI6, ANI8, and ANI9 pins selected, do not change to the value output to P42, P41, P17, and P16 while conversion is in progress; otherwise the conversion resolution may be degraded.

<2> If a pin adjacent to a pin that is being A/D converted is used as a digital I/O port pin, the A/D conversion result might differ from the expected value due to a coupling noise. Be sure to prevent such a pulse from being input or output.

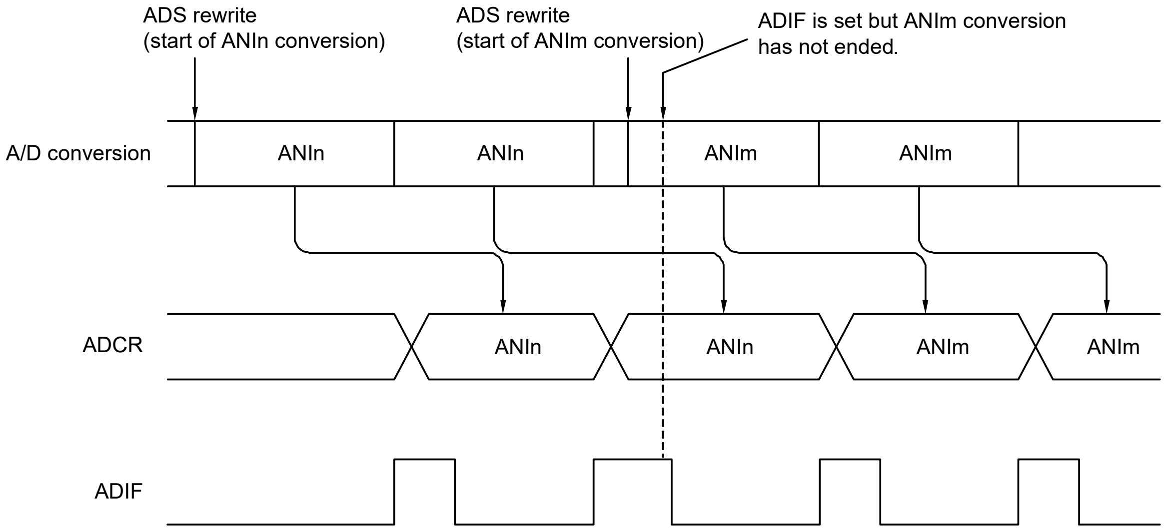
(6) Input impedance of analog input (ANIn) pins

This A/D converter charges a sampling capacitor for sampling during sampling time.
Therefore, only a leakage current flows when sampling is not in progress, and a current that charges the capacitor flows during sampling. Consequently, the input impedance fluctuates depending on whether sampling is in progress, and on the other states.
To make sure that sampling is effective, however, we recommend using the converter with analog input sources that have output impedances no greater than 1 kΩ. If a source has a higher output impedance, lengthen the sampling time or connect a larger capacitor (with a value of about 0.1 μF) to the pin from among ANI0 to ANI9 which the source is connected (see **Figure 16 - 46 Analog Input Pin Connection**). The sampling capacitor may be being charged while the setting of the ADCS bit is 0 and immediately after sampling is restarted and so is not defined at these times. Accordingly, the state of conversion is undefined after charging starts in the next round of conversion after the value of the ADCS bit has been 1 or when conversion is repeated. Thus, to secure full charging regardless of the size of fluctuations in the analog signal, ensure that the output impedances of the sources of analog inputs are low or secure sufficient time for the completion of conversion.

(7) Interrupt request flag (ADIF)

The interrupt request flag (ADIF) is not cleared even if the analog input channel specification register (ADS) is changed.
Therefore, if an analog input pin is changed during A/D conversion, the A/D conversion result and ADIF flag for the pre-change analog input may be set just before the ADS register rewrite. Caution is therefore required since, at this time, when ADIF flag is read immediately after the ADS register rewrite, ADIF flag is set despite the fact A/D conversion for the post-change analog input has not ended.
When A/D conversion is stopped and then resumed, clear ADIF flag before the A/D conversion operation is resumed.

Figure 16 - 47 Timing of A/D Conversion End Interrupt Request Generation



(8) Conversion results just after A/D conversion start

While in the software trigger mode or hardware trigger no-wait mode, the first A/D conversion value immediately after A/D conversion starts may not fall within the rating range if the ADCS bit is set to 1 within 1 μs after the ADCE bit was set to 1. Take measures such as polling the A/D conversion end interrupt request (INTAD) and removing the first conversion result.

(9) A/D conversion result register (ADCR, ADCRH) read operation

When a write operation is performed to A/D converter mode register 0 (ADM0), analog input channel specification register (ADS), and port mode control register (PMCxx), the contents of the ADCR and ADCRH registers may become undefined. Read the conversion result following conversion completion before writing to the ADM0, ADS, or PMC register. Using a timing other than the above may cause an incorrect conversion result to be read.

(10) Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 16 - 48 Internal Equivalent Circuit of ANIn Pin

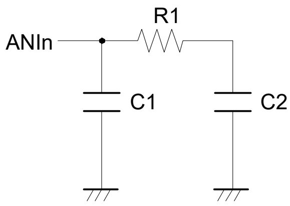


Table 16 - 5 Resistance and Capacitance Values of Equivalent Circuit (Reference Values)

VDD	ANIn Pins	R1 [kΩ]	C1 [pF]	C2 [pF]
3.6 V ≤ AVDD ≤ 5.5 V	ANI0 to ANI9	18	8	7.0
2.7 V ≤ AVDD < 3.6 V	ANI0 to ANI9	53	8	7.0

Remark The resistance and capacitance values shown in Table 16 - 5 are not guaranteed values.

(11) Starting the A/D converter

Start the A/D converter after the VDD voltage stabilizes.

CHAPTER 17 CONFIGURABLE AMPLIFIER

The number of input and output pins of the configurable amplifier differs depending on the product.

Unit	I/O pins		32-pin	36-pin
	When using the configurable amplifier as a general amplifier	When using the configurable amplifier as a configurable amplifier		
Operational amplifier 0	AMP0O (output of operational amplifier 0)	AMP0O (output of operational amplifier 0)	√	√
	AMP0N (negative input to operational amplifier 0)	ANX0/ANX1 (general-purpose analog I/O)	√	√
	AMP0P (positive input to operational amplifier 0)	ANX0/ANX1 (general-purpose analog I/O)	√	√
Operational amplifier 1	AMP1O (output of operational amplifier 1)	AMP1O (output of operational amplifier 1)	√	√
	AMP1N (negative input to operational amplifier 1)	ANX0/ANX1/ANX2/ANX3 (general-purpose analog I/O)	√	√
	AMP1P (positive input to operational amplifier 1)	ANX0/ANX1/ANX2/ANX3 (general-purpose analog I/O)	√	√
Operational amplifier 2	AMP2O (output of operational amplifier 2)	AMP2O (output of operational amplifier 2)	√	√
	AMP2N (negative input to operational amplifier 2)	ANX0/ANX1/ANX2/ANX3/ANX4Note/ANX5 (general-purpose analog I/O)	—	√
	AMP2P (positive input to operational amplifier 2)	ANX0/ANX1/ANX2/ANX3/ANX4Note/ANX5 (general-purpose analog I/O)	√	√

Note Available in 36-pin products only

17.1 Features of configurable amplifier

A configurable amplifier consists of three differential operational amplifier units, each of which has 2 inputs and 1 output, and can be used as a three-channel general operational amplifier. The configurable amplifier incorporates up to six general analog I/O ports (ANX0 to ANX5) and configurable switches. The configurable amplifier can be used as various types of operational amplifiers by controlling the general analog I/O ports and configurable switches.

The configurable amplifier has the following features.

17.1.1 When using the configurable amplifier as a general amplifier

By default, the configurable amplifier is used as a general operational amplifier with 2 inputs and 1 output.

- By default, the AMP0P to AMP2P pins are used as the positive input to the operational amplifier.
- By default, the AMP0N to AMP2N pins are used as the negative input to the operational amplifier.
- The AMP0O to AMP2O pins can be used as operational amplifier output pins that output signals without passing through a switch.
- The I/O signals of all operational amplifier units can be used as 10-bit A/D converter input signals.
- The signal output from the 12-bit D/A converter can be used as the positive input signal for each operational amplifier.
- A general operational amplifier can configure a voltage follower by feeding back its own output signal as its own negative input signal.
- A general operational amplifier can operate in normal speed and high-speed mode. Normal mode enables power saving, whereas high-speed mode enables a better response speed.

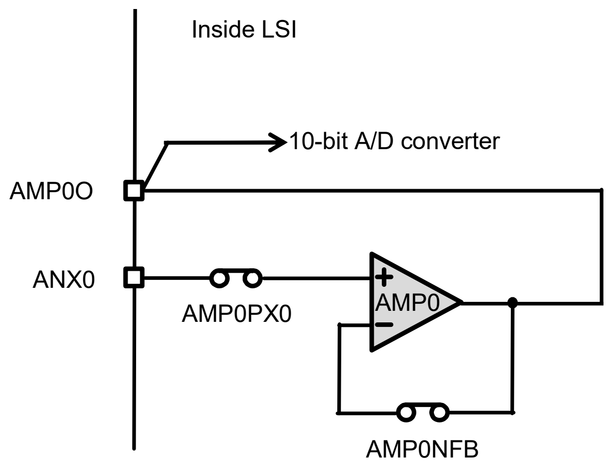
17.1.2 Using the configurable amplifier as a configurable amplifier

A configurable amplifier can be used as various types of operational amplifiers by controlling the configurable switches in combination with external resistors and capacitors. Typical examples are provided below.

(1) Voltage follower

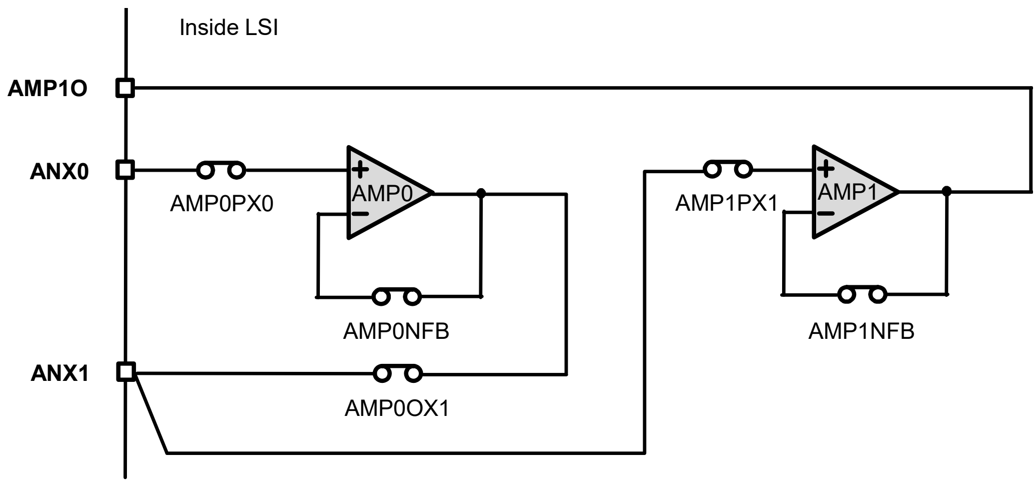
A general operational amplifier can configure a voltage follower by feeding back its own output signal as its own negative input signal. To configure a feedback circuit, set bit 7 (AMPnNFB) of the configurable amplifier n negative input selection register (AMPnS1) to 1.

Figure 17 - 1 Example of Using a Configurable Amplifier as a Voltage Follower



Use general-purpose analog port n (ANXn) to input the preamplifier output signal to the post amplifier. To connect the signal output from the voltage follower of operational amplifier 0 to the positive input of operational amplifier 1, for example, set bit 1 (AMP0OX1) of the configurable amplifier 0 output selection register (AMP0S0) to 1. The operational amplifier 0 output is then connected to the general-purpose analog I/O port ANX1. Then, connect ANX1 to the positive input signal of post amplifier 1 by setting bit 1 (AMP1PX1) of the configurable amplifier 1 positive input selection register (AMP1S2) to 1.

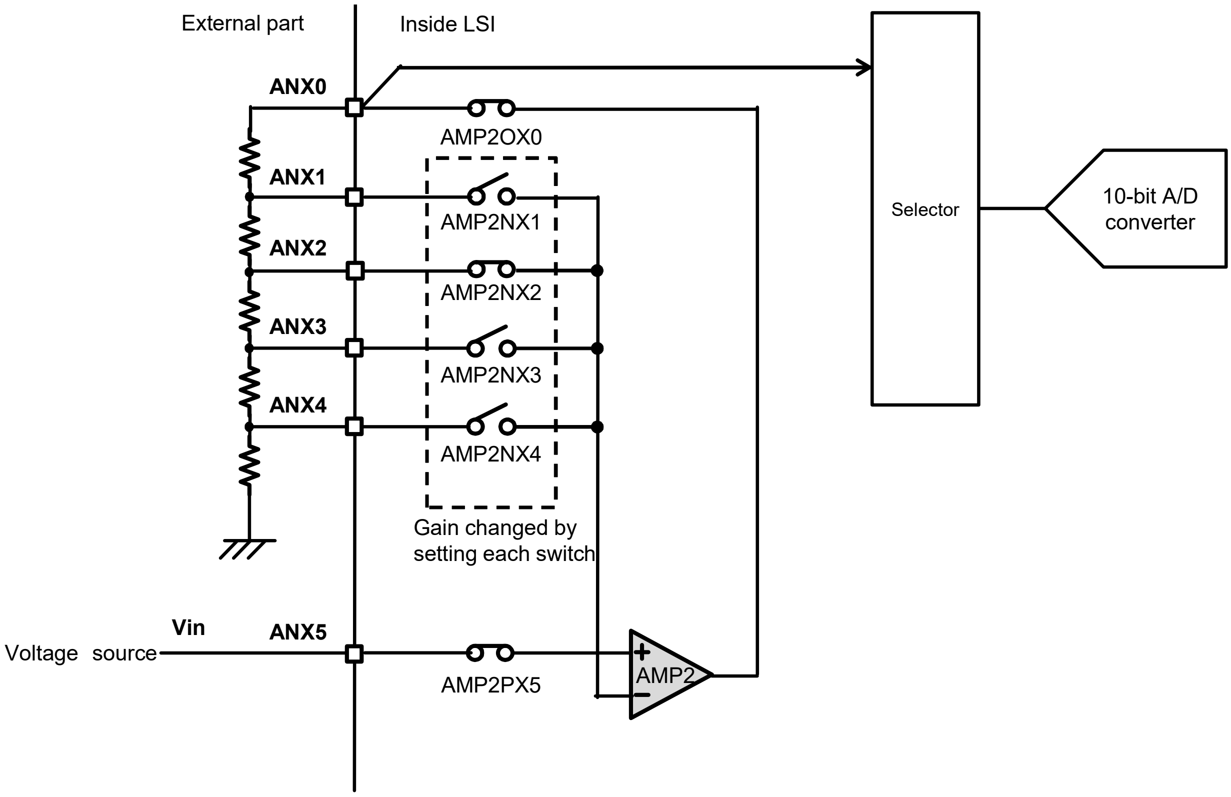
Figure 17 - 2 Example of Using a Configurable Amplifier as a Cascaded Voltage Follower



(2) Programmable non-inverting amplifier

A programmable non-inverting amplifier can be configured by using a combination of configurable switches and external resistors connected to general-purpose analog ports. An example of a non-inverting amplifier is shown below.

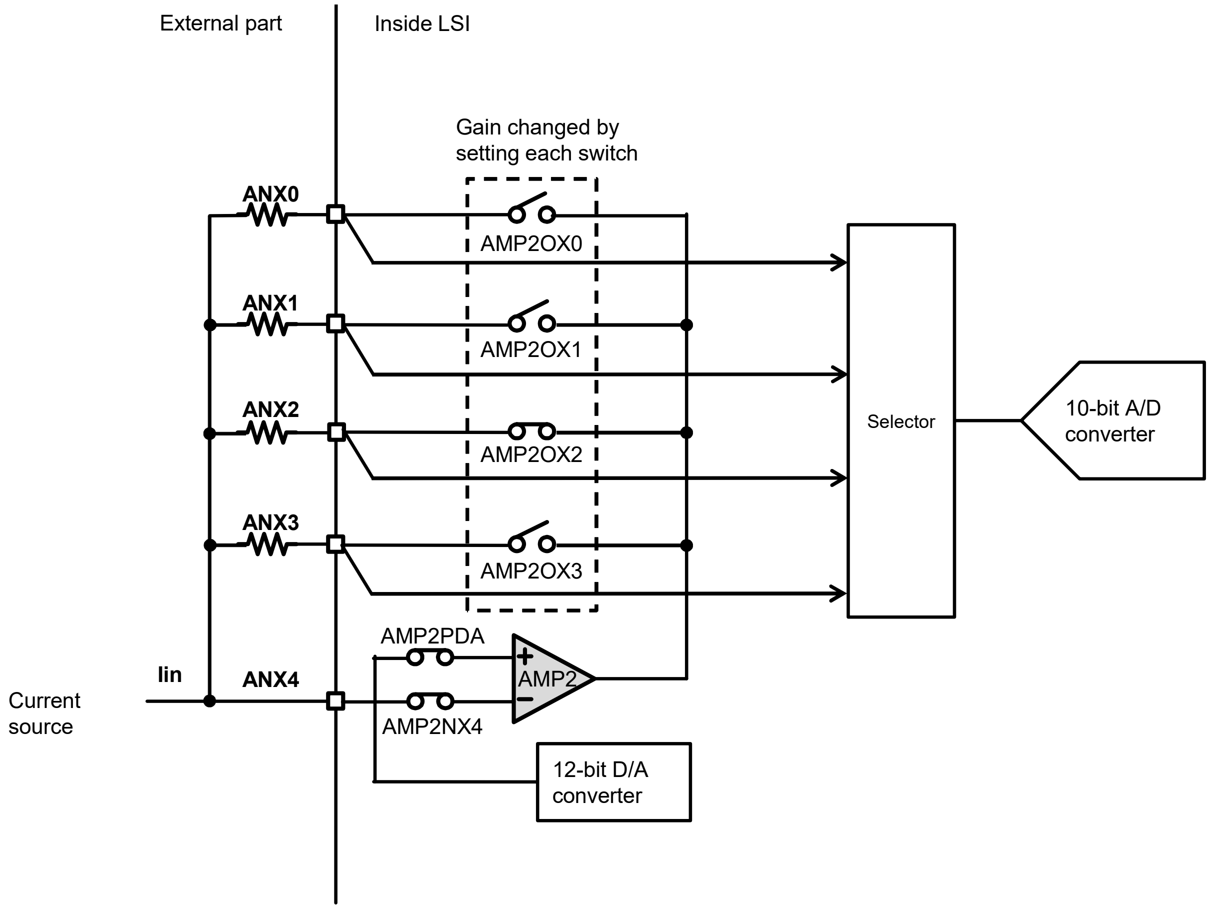
Figure 17 - 3 Example of Using a Configurable Amplifier as a Programmable Non-inverting Amplifier



(3) Programmable transimpedance amplifier

An example of a transimpedance amplifier in which the gain can be switched by using software is shown below.

Figure 17 - 4 Example of Using a Configurable Amplifier as a Programmable Transimpedance Amplifier

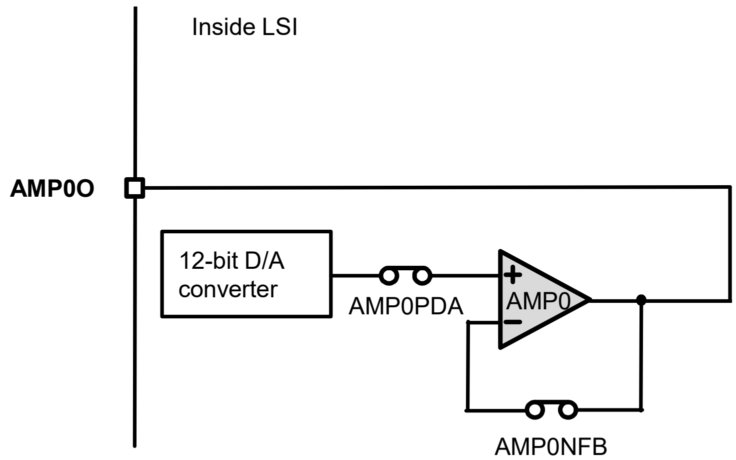


17.1.3 Using the configurable amplifier as a 12-bit D/A converter output amplifier

You can use the configurable amplifier to output the 12-bit D/A converter output to outside the LSI. To input the output from the 12-bit D/A converter to the positive input pin of operational amplifier n, set bit 7 (AMPnPDA) of the configurable amplifier n positive input selection register (AMPnS2) to 1. To output the 12-bit D/A converter output from the AMPnO pin, configure a voltage follower by setting bit 7 (AMPnNFB) of the configurable amplifier n negative input selection register (AMPnS1) to 1.

Remark n: Unit number (n = 0 to 2)

Figure 17 - 5 Using the Configurable Amplifier as a 12-bit D/A Converter Output Amplifier



17.1.4 Offset calibration

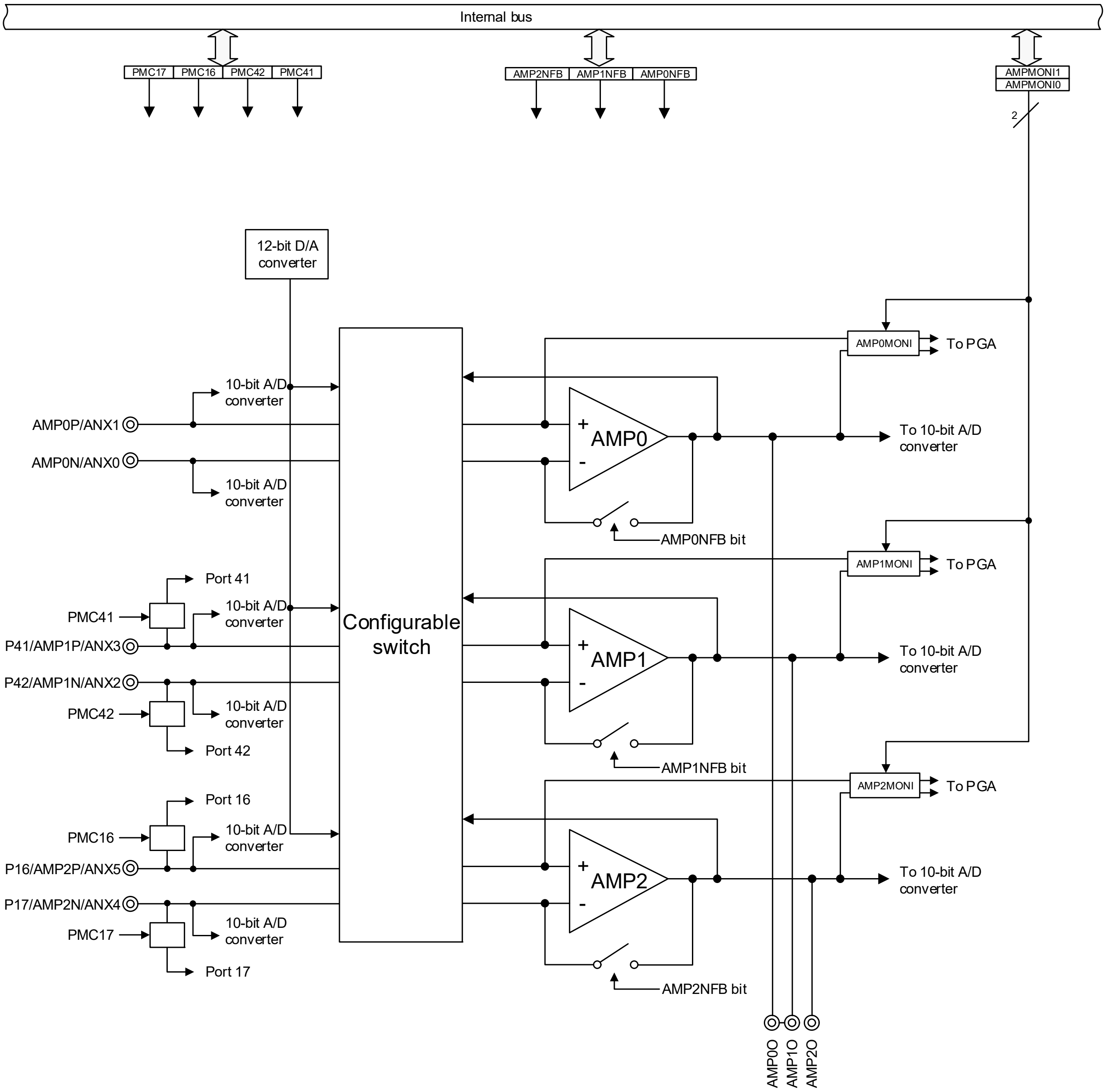
The offset voltage can be measured by using the positive input signal (AMPnP) and output signal (AMPnO) of each operational amplifier as the input signals to the PGAs for the 24-bit ΔΣ A/D converter. The offset voltage can be calibrated according to environmental variations by measuring the offset voltage value by using the 24-bit ΔΣ A/D converter and trimming the differential input offset of the amplifier by using the configurable amplifier n trimming register (AMPnCAL).

Remark n: Unit number (n = 0 to 2)

17.2 Configuration of Configurable Amplifier

Figure 17 - 6 shows the block diagram of the configurable amplifier.

Figure 17 - 6 Block Diagram of Configurable Amplifier



Caution 32-pin products do not support the digital functions of general-purpose port pins P16, P17, P41, and P42.

Figure 17 - 7 to Figure 17 - 9 show the block diagrams of general amplifiers 0 to 2.

Figure 17 - 7 Block Diagram of Operational Amplifier 0

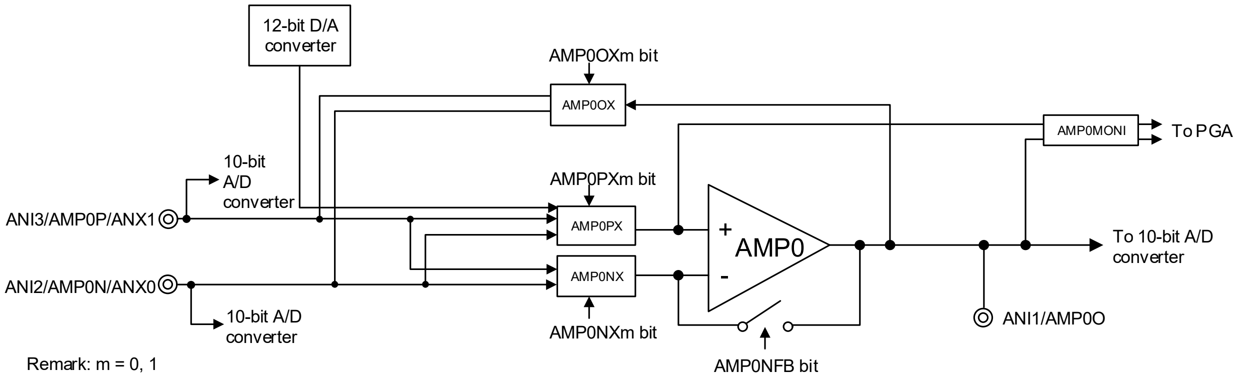


Figure 17 - 8 Block Diagram of Operational Amplifier 1

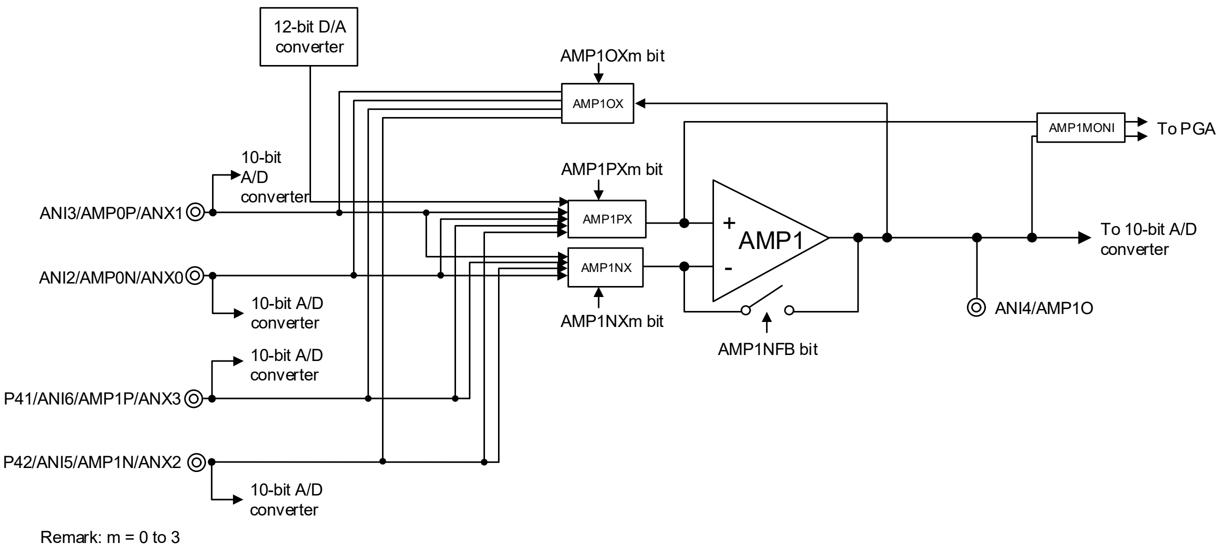
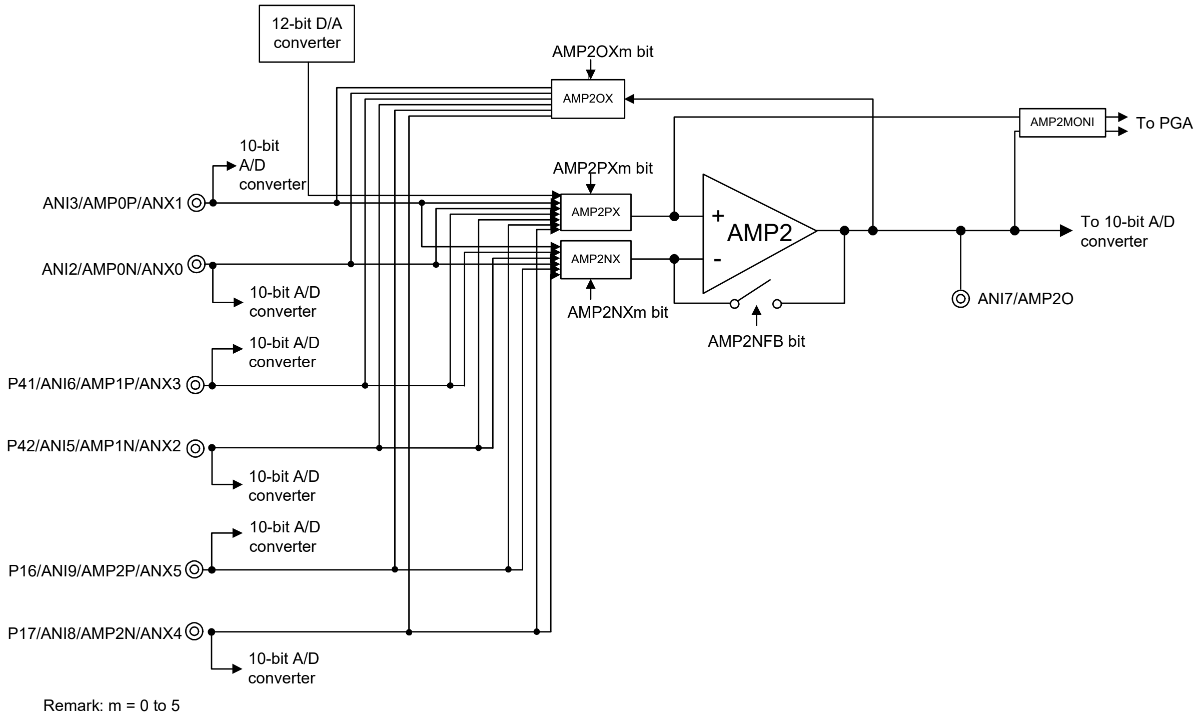


Figure 17 - 9 Block Diagram of Operational Amplifier 2



17.3 Registers Controlling the Configurable Amplifier

Table 17 - 1 shows the registers that control the configurable amplifier.

Table 17 - 1 Registers Controlling the Configurable Amplifier

Item	Settings specified by:
Control registers	Port mode control registers (PMC1, PMC4)
	Peripheral enable register 1 (PER1)
	Analog front-end power supply selection register (AFEPWS)
	Configurable amplifier 0 mode register (AMP0MR)
	Configurable amplifier 1 mode register (AMP1MR)
	Configurable amplifier 2 mode register (AMP2MR)
	Configurable amplifier 0 output selection register (AMP0S0)
	Configurable amplifier 1 output selection register (AMP1S0)
	Configurable amplifier 2 output selection register (AMP2S0)
	Configurable amplifier 0 negative input selection register (AMP0S1)
	Configurable amplifier 1 negative input selection register (AMP1S1)
	Configurable amplifier 2 negative input selection register (AMP2S1)
	Configurable amplifier 0 positive input selection register (AMP0S2)
	Configurable amplifier 1 positive input selection register (AMP1S2)
	Configurable amplifier 2 positive input selection register (AMP2S2)
	Configurable amplifier 0 trimming register (AMP0CAL)
	Configurable amplifier 1 trimming register (AMP1CAL)
	Configurable amplifier 2 trimming register (AMP2CAL)
	Configurable amplifier 0 trimming code register (AMP0TRM)
	Configurable amplifier 1 trimming code register (AMP1TRM)
	Configurable amplifier 2 trimming code register (AMP2TRM)

17.3.1 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 using the configurable amplifier, be sure to set bit 5 (AMPEN) to 1.

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

Reset signal generation clears this register to 00H.

Figure 17 - 10 Format of Peripheral Enable Register 1 (PER1)

Address: F007AH After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	<3>	<2>	<1>	<0>
PER1	DACEN	TRGEN	AMPEN	0	DTCEN	PGAEN	AFEEN	TRJ0EN

AMPEN	Control of input clock supplied to the configurable amplifier
0	Stops input clock supply. • SFRs used by the configurable amplifier cannot be written. • The configurable amplifier is in the reset status.
1	Enables input clock supply. • SFRs used by the configurable amplifier can be read and written.

Caution 1. Be sure to set AMPEN to 1 before setting up the configurable amplifier. When AMPEN is 0, writing to the configurable amplifier control registers is ignored and the value read from these registers is the initial value.

Caution 2. Be sure to clear bit 4 to “0”.

17.3.2 Analog front-end power supply selection register (AFEPWS)

The AFEPWS register is used to control the power supplied to the programmable gain instrumentation amplifier (PGA) and AFE reference voltage (ABGR) blocks.
The AFEPWS register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 17 - 11 Format of Analog Front-End Power Supply Selection Register (AFEPWS)

Address: F0440H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	<0>
AFEPWS	DACPON	AMP2PON	AMP1PON	AMP0PON	0	PGAPON	0	AFEPON

AMP2PON	Control of power supplied to configurable amplifier 2 (AMP2) block
0	Power-off (default)
1	Power-on

AMP1PON	Control of power supplied to configurable amplifier 1 (AMP1) block
0	Power-off (default)
1	Power-on

AMP0PON	Control of power supplied to configurable amplifier 0 (AMP0) block
0	Power-off (default)
1	Power-on

Caution Be sure to clear bits 1 and 3 to “0”.
For the setting of bits 0, 2, and 7 refer to 13.3.2 Analog front-end power supply selection register (AFEPWS).

17.3.3 Configurable amplifier n mode register (AMPnMR)

This register is used to specify the operating mode of the operational amplifiers.
Rewrite the AMPnMR register while operational amplifier n is stopped (AMPnPON = 0).
The AMPnMR register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 17 - 12 Format of Configurable Amplifier n Mode Register (AMPnMR)

Address: F0473H (AMP0MR), F0477H (AMP1MR), F047BH (AMP2MR) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
AMPnMR	0	0	0	0	0	0	0	AMPnHSM

AMPnHSM	Selection of operating mode of operational amplifier n (n = 0 to 2)
0	Normal mode
1	High-speed mode

Caution Rewrite the AMPnMR register while operational amplifier n is stopped (AMPnPON = 0).

Remark n: Unit number (n = 0 to 2)

17.3.4 Configurable amplifier 0 output selection register (AMP0S0)

This register is used to specify whether to connect the operational amplifier 0 output pin to a general-purpose analog I/O port and select the operational amplifier whose input offset is to be trimmed.
The general-purpose analog I/O port can be selected from ANX0 or ANX1.
The AMP0S0 register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 17 - 13 Format of Configurable Amplifier 0 Output Selection Register (AMP0S0)

Address: F0470H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
AMP0S0	AMPMONI1	AMPMONI0	0	0	0	0	AMP0OX1	AMP0OX0

AMPMONI1	AMPMONI0	Selection of offset trimming monitor
0	0	No connection
0	1	Input the positive input signal and output signal of operational amplifier 0 to the PGA.
1	0	Input the positive input signal and output signal of operational amplifier 1 to the PGA.
1	1	Input the positive input signal and output signal of operational amplifier 2 to the PGA.

AMP0OXm	Connection of operational amplifier 0 output pin to general-purpose analog I/O port ANXm (m = 0, 1)
0	None
1	Connect

17.3.5 Configurable amplifier 1 output selection register (AMP1S0)

This register is used to specify whether to connect the operational amplifier 1 output pin to a general-purpose analog I/O port.

The general-purpose analog I/O port can be selected from ANX0 to ANX3.

The AMP1S0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 17 - 14 Format of Configurable Amplifier 1 Output Selection Register (AMP1S0)

Address: F0474H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
AMP1S0	0	0	0	0	AMP1OX3	AMP1OX2	AMP1OX1	AMP1OX0

AMP1OXm	Connection of operational amplifier 1 output pin to general-purpose analog I/O port ANXm (m = 0 to 3)
0	None
1	Connect

17.3.6 Configurable amplifier 2 output selection register (AMP2S0)

This register is used to specify whether to connect the operational amplifier 2 output pin to a general-purpose analog I/O port.

The general-purpose analog I/O port can be selected from ANX0 to ANX5[Note].

The AMP2S0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 17 - 15 Format of Configurable Amplifier 2 Output Selection Register (AMP2S0)

Address: F0478H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
AMP2S0	0	0	AMP2OX5	AMP2OX4[Note]	AMP2OX3	AMP2OX2	AMP2OX1	AMP2OX0

AMP2OXm	Connection of operational amplifier 2 output pin to general-purpose analog I/O port ANXm (m = 0 to 5)
0	None
1	Connect

Note General-purpose analog I/O port ANX4 is not available in 32-pin products, so be sure to write 0 to the AMP2OX4 bit.

17.3.7 Configurable amplifier 0 negative input selection register (AMP0S1)

This register is used to specify whether to connect the operational amplifier 0 negative input pin to a general-purpose analog I/O port and set up a feedback circuit for configuring a voltage follower.

The general-purpose analog I/O port can be selected from ANX0 or ANX1.

The AMP0S1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 01H.

Figure 17 - 16 Format of Configurable Amplifier 0 Negative Input Selection Register (AMP0S1)

Address: F0471H After reset: 01H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
AMP0S1	AMP0NFB	0	0	0	0	0	AMP0NX1	AMP0NX0

AMP0NFB	Connection of operational amplifier 0 output pin to negative input pin and configuration of feedback circuit
0	None
1	Connect

AMP0NXm	Connection of operational amplifier 0 negative input pin to general-purpose analog I/O port ANXm (m = 0, 1)
0	None
1	Connect

Caution **The AMP0NFB and AMP0NXm bits must not be set to 1 at the same time. Write 80H to the AMP0S1 register when configuring a voltage follower. (Set only the AMP0NFB bit to 1.)**

17.3.8 Configurable amplifier 1 negative input selection register (AMP1S1)

This register is used to specify whether to connect the operational amplifier 1 negative input pin to a general-purpose analog I/O port and set up a feedback circuit for configuring a voltage follower.

The general-purpose analog I/O port can be selected from ANX0 to ANX3.

The AMP1S1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 04H.

Figure 17 - 17 Format of Configurable Amplifier 1 Negative Input Selection Register (AMP1S1)

Address: F0475H After reset: 04H R/W

Symbol	7	6	5	4	<3>	<2>	<1>	<0>
AMP1S1	AMP1NFB	0	0	0	AMP1NX3	AMP1NX2	AMP1NX1	AMP1NX0

AMP1NFB	Connection of operational amplifier 1 output pin to negative input pin and configuration of feedback circuit
0	None
1	Connect

AMP1NXm	Connection of operational amplifier 1 negative input pin to general-purpose analog I/O port ANXm (m = 0 to 3)
0	None
1	Connect

Caution **The AMP1NFB and AMP1NXm bits must not be set to 1 at the same time. Write 80H to the AMP1S1 register when configuring a voltage follower. (Set only the AMP1NFB bit to 1.)**

17.3.9 Configurable amplifier 2 negative input selection register (AMP2S1)

This register is used to specify whether to connect the operational amplifier 2 negative input pin to a general-purpose analog I/O port and set up a feedback circuit for configuring a voltage follower.

The general-purpose analog I/O port can be selected from ANX0 to ANX5[Note].

The AMP2S1 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 10H.

Figure 17 - 18 Format of Configurable Amplifier 2 Negative Input Selection Register (AMP2S1)

Address: F0479H After reset: 10H R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
AMP2S1	AMP2NFB	0	AMP2NX5	AMP2NX4[Note]	AMP2NX3	AMP2NX2	AMP2NX1	AMP2NX0

AMP2NFB	Connection of operational amplifier 2 output pin to negative input pin and configuration of feedback circuit
0	None
1	Connect

AMP2NXm	Connection of operational amplifier 2 negative input pin to general-purpose analog I/O port ANXm (m = 0 to 5)
0	None
1	Connect

Note General-purpose analog I/O port ANX4 is not available in 32-pin products, so be sure to write 0 to the AMP2NX4 bit.

Caution The AMP2NFB and AMP2NXm bits must not be set to 1 at the same time. Write 80H to the AMP2S1 register when configuring a voltage follower. (Set only the AMP2NFB bit to 1.)

17.3.10 Configurable amplifier 0 positive input selection register (AMP0S2)

This register is used to specify whether to connect the operational amplifier 0 positive input pin to a general-purpose analog I/O port and whether to input the 12-bit D/A converter output signal to the operational amplifier positive input pin.

The general-purpose analog I/O port can be selected from ANX0 or ANX1.

The AMP0S2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 02H.

Figure 17 - 19 Format of Configurable Amplifier 0 Positive input Selection Register (AMP0S2)

Address: F0472H After reset: 02H R/W

Symbol	7	6	5	4	3	2	<1>	<0>
AMP0S2	AMP0PDA	0	0	0	0	0	AMP0PX1	AMP0PX0

AMP0PDA	Connection of the 12-bit D/A converter output signal to the operational amplifier 0 positive input pin
0	None
1	Connect

AMP0PXm	Connection of operational amplifier 0 positive input pin to general-purpose analog I/O port ANXm (m = 0, 1)
0	None
1	Connect

Caution When a general-purpose analog I/O port-pin function is connected, the D/A converter is not connected even if AMP0PDA = 1.

17.3.11 Configurable amplifier 1 positive input selection register (AMP1S2)

This register is used to specify whether to connect the operational amplifier 1 positive input pin to a general-purpose analog I/O port and whether to input the 12-bit D/A converter output signal to the operational amplifier positive input pin.

The general-purpose analog I/O port can be selected from ANX0 to ANX3.

The AMP1S2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 08H.

Figure 17 - 20 Format of Configurable Amplifier 1 Positive Input Selection Register (AMP1S2)

Address: F0476H After reset: 08H R/W

Symbol	7	6	5	4	<3>	<2>	<1>	<0>
AMP1S2	AMP1PDA	0	0	0	AMP1PX3	AMP1PX2	AMP1PX1	AMP1PX0

AMP1PDA	Connection of 12-bit D/A converter output signal to the operational amplifier 1 positive input pin
0	None
1	Connect

AMP1PXm	Connection of operational amplifier 1 positive input pin to general-purpose analog I/O port ANXm (m = 0 to 3)
0	None
1	Connect

Caution When a general-purpose analog I/O port-pin function is connected, the D/A converter is not connected even if AMP1PDA = 1.

17.3.12 Configurable amplifier 2 positive input selection register (AMP2S2)

This register is used to specify whether to connect the operational amplifier 2 positive input pin to a general-purpose analog I/O port and whether to input the 12-bit D/A converter output signal to the operational amplifier positive input pin.

The general-purpose analog I/O port can be selected from ANX0 to ANX5[Note].

The AMP2S2 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to 20H.

Figure 17 - 21 Format of Configurable Amplifier 2 Positive Input Selection Register (AMP2S2)

Address: F047AH After reset: 20H R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
AMP2S2	AMP2PDA	0	AMP2PX5	AMP2PX4[Note]	AMP2PX3	AMP2PX2	AMP2PX1	AMP2PX0

AMP2PDA	Connection of 12-bit D/A converter output signal to the operational amplifier 2 positive input pin
0	None
1	Connect

AMP2PXm	Connection of operational amplifier 2 positive input pin to general-purpose analog I/O port ANXm (m = 0 to 5)
0	None
1	Connect

Note General-purpose analog I/O port ANX4 is not available in 32-pin products, so be sure to write 0 to the AMP2PX4 bit.

Caution When a general-purpose analog I/O port-pin function is connected, the D/A converter is not connected even if AMP2PDA = 1.

17.3.13 Configurable amplifier n trimming register (AMPnCAL)

This register is used to specify the offset trimming code (correction value) for each operational amplifier[Note].
For details about trimming, refer to 17.4.5 Offset trimming.
The AMPnCAL register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 17 - 22 Format of Configurable Amplifier n Trimming Register (AMPnCAL)

Address: F047CH (AMP0CAL), F047DH (AMP1CAL), F047EH (AMP2CAL) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
AMPnCAL	0	AMPnCAL6	AMPnCAL5	AMPnCAL4	AMPnCAL3	AMPnCAL2	AMPnCAL1	AMPnCAL0

Note Before trimming the offset voltage by using the PGA, specify the operational amplifier to be used as a voltage follower by using the AMPnSm register and connect the target operational amplifier to the PGA by using the AMP0S0 register.

Caution Be sure to clear bit 7 of the AMPnCAL register to 0.

Remark n: Unit number (n = 0 to 2)

17.3.14 Configurable amplifier n trimming code register (AMPnTRM)

This register stores the factory default offset trimming code for each operational amplifier. After a reset ends, copy the AMPnTRM value to the AMPnCAL register before using the configurable amplifier.
The AMPnTRM register can be read by using an 8-bit memory manipulation instruction.

Figure 17 - 23 Format of Configurable Amplifier n Trimming Code Register (AMPnTRM)

Address: F00ACH (AMP0TRM), F00ADH (AMP1TRM), F00AEH (AMP2TRM) After reset: Initial trimming data R

Symbol	7	6	5	4	3	2	1	0
AMPnTRM	0	AMPnTRM6	AMPnTRM5	AMPnTRM4	AMPnTRM3	AMPnTRM2	AMPnTRM1	AMPnTRM0

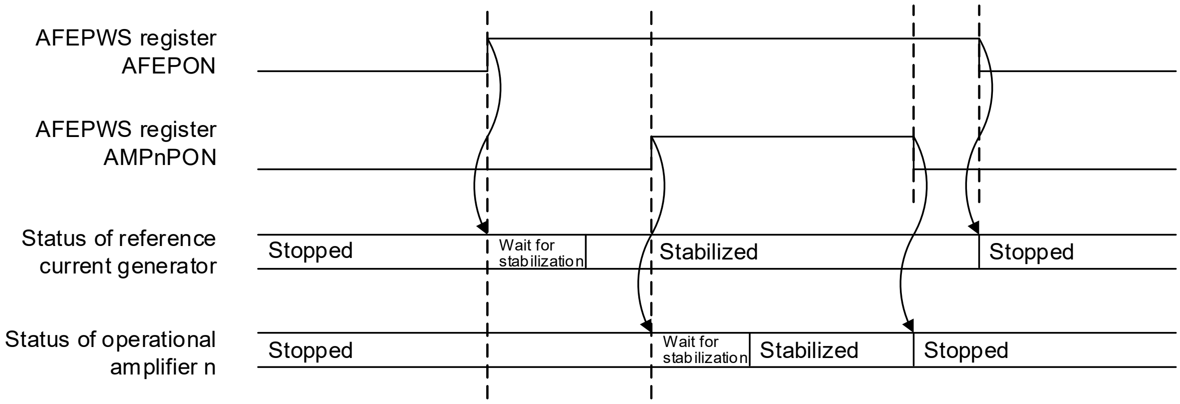
Remark n: Unit number (n = 0 to 2)

17.4 Operation

17.4.1 Configurable amplifier control operation

Figure 17 - 24 shows the configurable amplifier control operation.

Figure 17 - 24 Configurable Amplifier Control Operation



Remark n: Unit number (n = 0 to 2)

17.4.2 Procedure for controlling the configurable amplifiers

The flowchart for starting and stopping the configurable amplifier is shown below. The flowchart of an example for setting up each register is shown below.

Figure 17 - 25 Procedure for Starting the Configurable Amplifiers

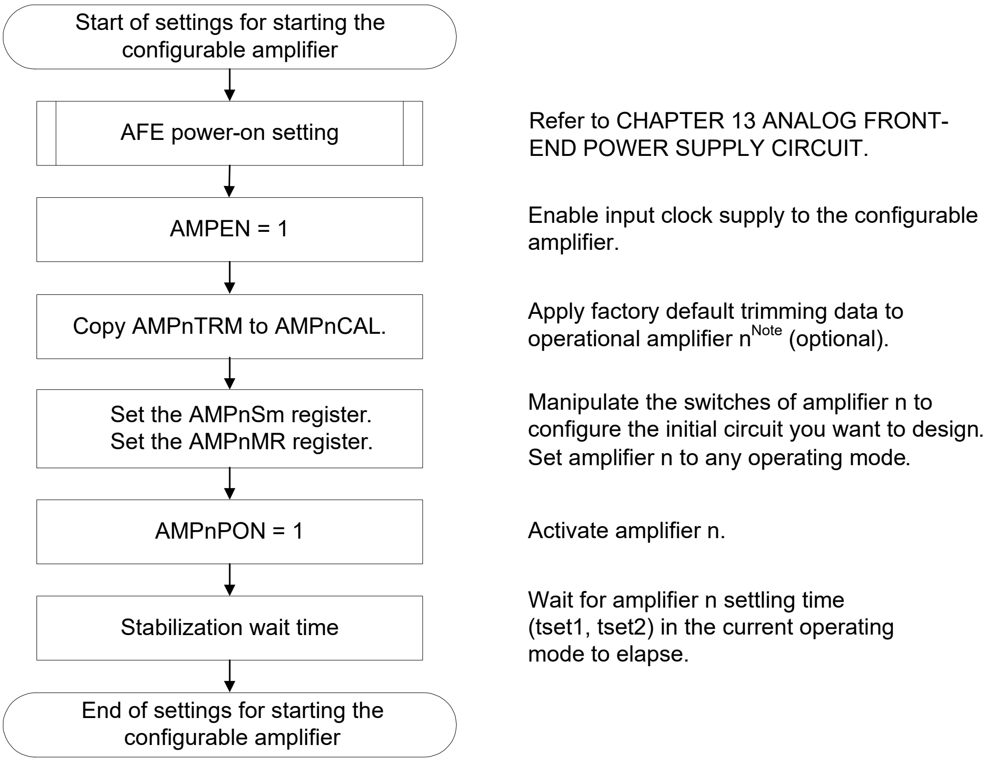
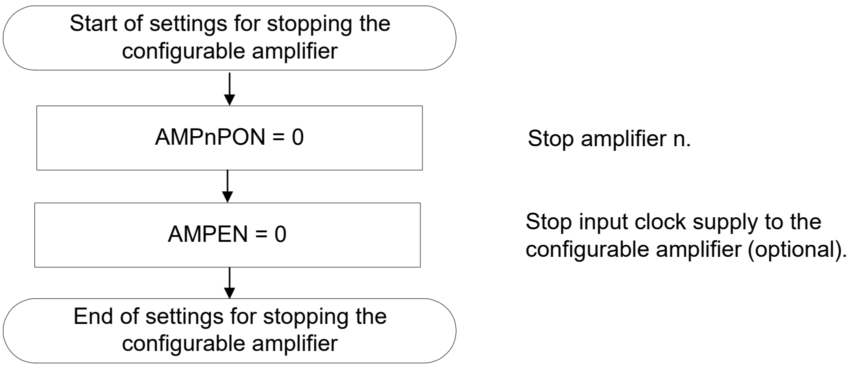


Figure 17 - 26 Procedure for Stopping the Configurable Amplifiers



Note At shipment, data is trimmed with a voltage follower configured and no load is applied.

Remark 1. n: Unit number (n = 0 to 2)

m: m = 0 → Configurable amplifier x output selection register

m = 1 → Configurable amplifier x negative input selection register

m = 2 → Configurable amplifier x positive input selection register

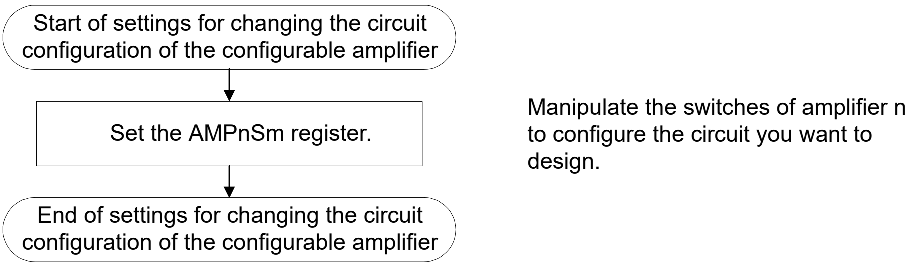
Remark 2. For details about the stabilization wait time, refer to **CHAPTER 33 ELECTRICAL SPECIFICATIONS (G: TA = -40 to +105°C)** and **CHAPTER 34 ELECTRICAL SPECIFICATIONS (M: TA = -40 to +125°C)**.

17.4.3 Changing the configurable amplifier configuration by using switches

The initial configuration of the configurable amplifier can be changed by using the switches without stopping the operational amplifier[Note]. The flowchart of an example of the settings is shown below.

Note Although the switch settings can be changed while operational amplifiers are operating (AMPnPON = 1), thoroughly consider the effect so that changing the switch settings during operation does not impact the user-prepared peripheral components.

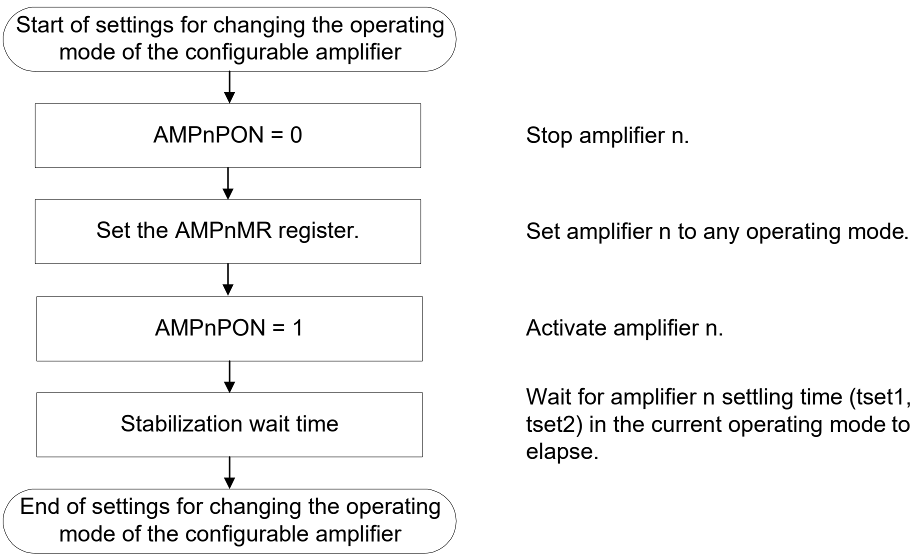
Figure 17 - 27 Procedure for Changing the Circuit Configuration of the Configurable Amplifier



17.4.4 Changing the operating mode of the configurable amplifier

When changing the operating mode of the configurable amplifier, turn off power to the amplifier first.
The flowchart of an example of the settings is shown below.

Figure 17 - 28 Procedure for Changing the Operating Mode of the Configurable Amplifier



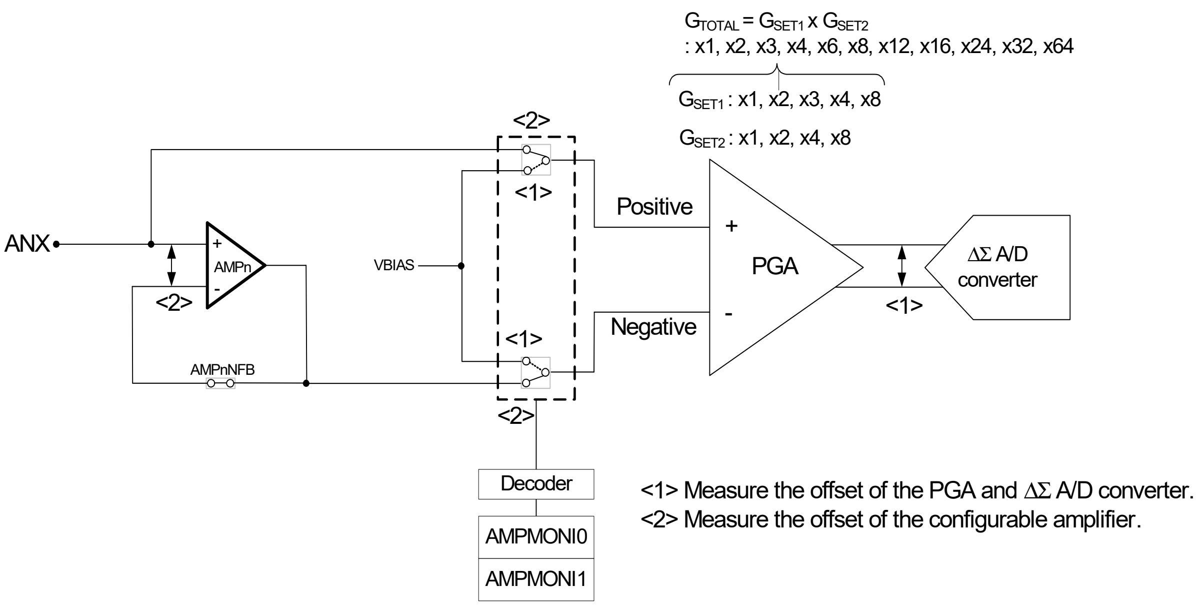
17.4.5 Offset trimming

The offset of the signal input to each operational amplifier in the configurable amplifier can be trimmed.
In the RL78/I1E, the offset of the configurable amplifier is trimmed at shipment[Note 1]. The trimmed data is stored in the AMPnTRM register. The offset of each operational amplifier can be trimmed by copying this data to the AMPnCAL register[Note 2].
You can trim the offset according to the environment in which the product is used by modifying the trimming code set to the AMPnCAL register.
As an example of how to trim the offset, Figure 17 - 29 shows how the offset of the operational amplifier input signal is trimmed by using the PGA and 24-bit ΔΣ A/D converter[Note 3].
Connect the operational amplifier input signal to the PGA input pin by setting the AMPMONI1 and AMPMONI0 bits of the AMP0S0 register.

Note 1. At shipment, data is trimmed with a voltage follower configured and no load is applied.
Note 2. Reset signal input clears the AMPnCAL register to 00H.
Note 3. To trim the offset by using the PGA and 24-bit ΔΣ A/D converter, specify the operational amplifier to be used as a voltage follower, and write 0 to bits 0 to 5 of the AMPnS0 register.

Remark n: Unit number (n = 0 to 2)

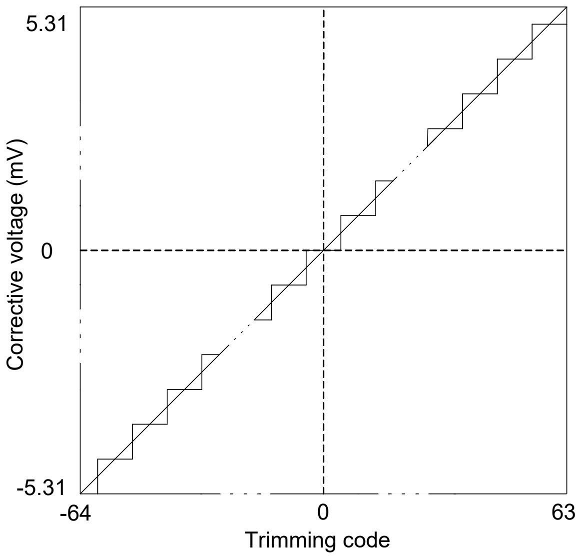
Figure 17 - 29 Offset Trimming Circuit Configuration



Each operational amplifier has 128 offset trimming codes.

Voltage corrected per code: $\frac{10.62[mV]}{128} = 82.97[\mu V]$ (TYP.)

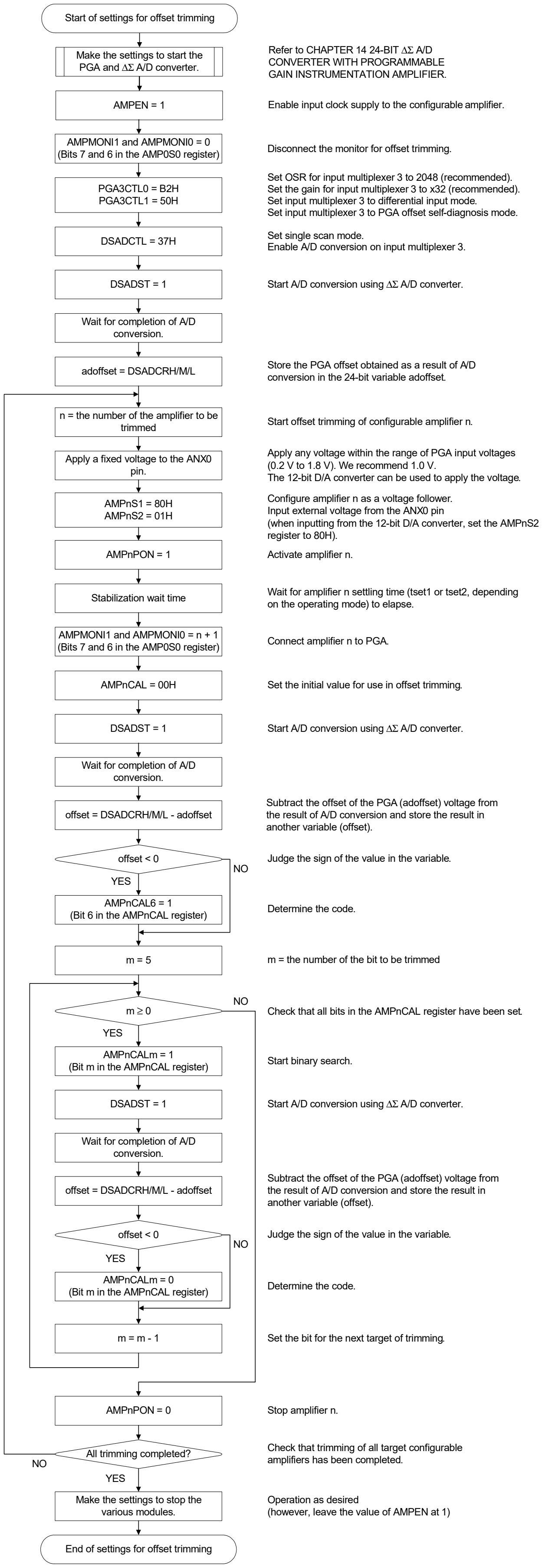
Figure 17 - 30 Relationship Between the Trimming Code and Corrected Voltage



17.4.6 Procedure for trimming the offset

The flowchart for connecting the configurable amplifier to the 24-bit ΔΣ A/D converter with programmable gain instrumentation amplifier and trimming the offset is shown below.

Figure 17 - 31 Procedure for Setting up the Offset Trimming



17.4.7 Analog/digital pins

In 36-pin products, general-purpose analog I/O ports ANX2, ANX3, ANX4, and ANX5 can be used as digital I/O port pins P42, P41, P17, and P16 by controlling port mode control registers 1 and 4 (PMC1 and PMC4).

Caution Specify the settings of port mode control registers while the operational amplifier n is stopped (AMPnPON = 0).

17.5 Cautions for the Configurable Amplifier

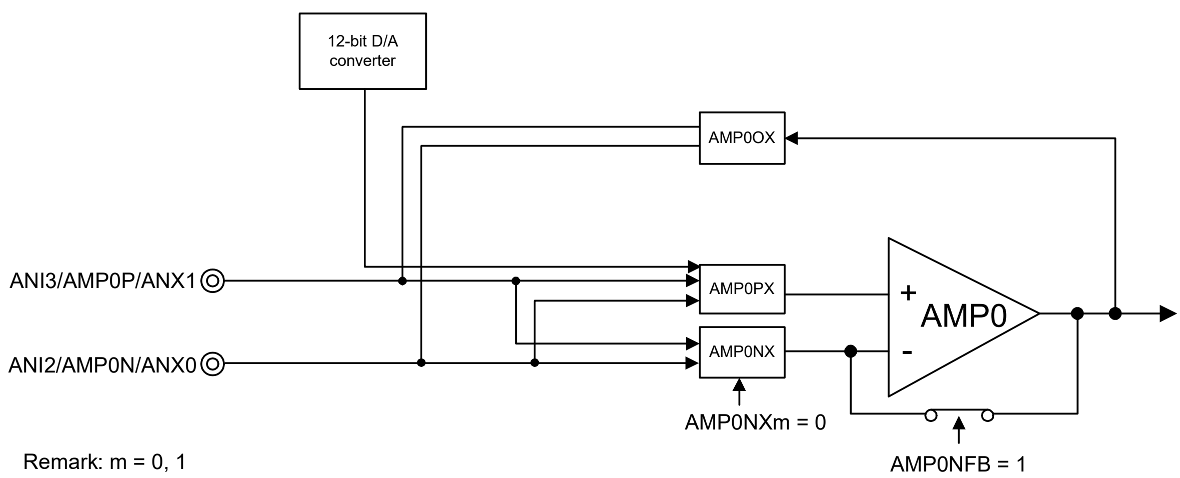
(1) When connecting a bypass capacitor to AV_{DD}/AV_{SS}, which are the power supply pins of the configurable amplifier, place the capacitor as close to the chip as possible (that is, make the wiring as short as possible) and minimize the transfer of noise between the capacitor and the device, board, or peripheral components.

(2) The factory default trimming codes copied from the AMPnTRM register to the AMPnCAL register are reset to 00H when the reset signal is input. Therefore, after the reset has ended, copy the trimming codes to the AMPnTRM register again.

(3) To prevent the output of each operational amplifier in the configurable amplifier from shorting to the same general-purpose analog I/O port (ANX), the output switches are connected exclusively.
The priority order is AMP0 > AMP1 > AMP2.

(4) The AMPnNFB and AMPnNXm bits in the same unit must not be set to 1 at the same time. Write 80H to the AMPnS1 register when configuring a voltage follower. (Set only the AMPnNFB bit to 1.)

Figure 17 - 32 Voltage Follower Circuit Configuration



(5) To prevent the operational amplifier inputs being left open, the power supply of the operational amplifiers is forcibly turned off when all input switches are off.

(6) To connect the 12-bit D/A converter to the input pins of the operational amplifiers, set the DACPON bit of the AFEPWS register to 1.

(7) When using the pins that also function as digital I/O port pins as general-purpose analog I/O ports (ANX), do not switch the function to a digital I/O port.

CHAPTER 18 12-BIT D/A CONVERTER

18.1 Function of 12-Bit D/A Converter

The RL78/I1E incorporates one 12-bit D/A converter channel.

Table 18 - 1 shows the specifications of the 12-bit D/A converter. Figure 18 - 1 shows a block diagram of the 12-bit D/A converter.

Table 18 - 1 Specifications of 12-bit D/A Converter

Item	Description
Resolution	12 bits
Number of output channels	1
Power saving feature	Can stop modules.
Event link feature (input)	D/A output values can be changed by receiving an event signal

Figure 18 - 1 Block Diagram of 12-bit D/A Converter

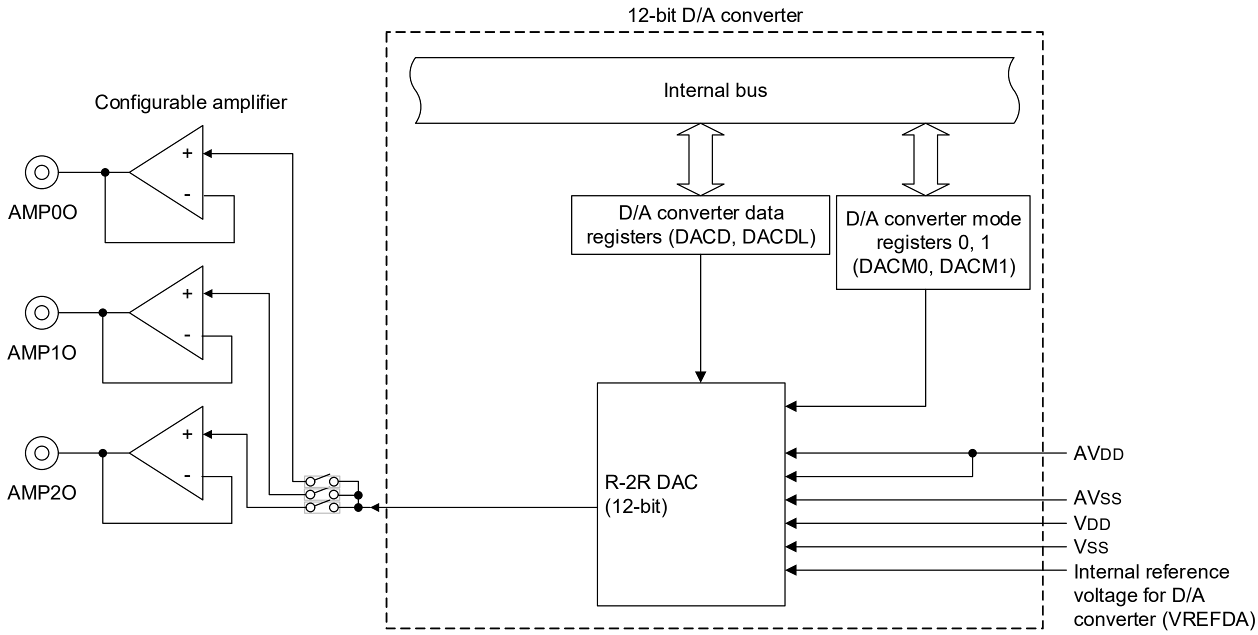


Table 18 - 2 shows the I/O pins used for the 12-bit D/A converter.

Table 18 - 2 I/O Pins

Pin Name	I/O	Description
V_{DD}	Input	Digital power supply pin
V_{SS}	Input	Ground pin
AV_{DD}	Input	Analog power supply pin
AV_{SS}	Input	Analog ground pin
AMPnO	Output	Analog output from configurable amplifier (n = 0, 1, 2)
VREFDA	Internally connected	Internal reference voltage for D/A converter

18.2 Registers Controlling the 12-Bit D/A Converter

Table 18 - 3 lists the registers controlling the 12-bit D/A converter.

Table 18 - 3 Registers Controlling 12-bit D/A Converter

Item	Settings specified by:
Control registers	Peripheral enable register 1 (PER1)
	Analog front-end power supply selection register (AFEPWS)
	D/A converter mode register 0 (DACM0)
	D/A converter mode register 1 (DACM1)
	D/A converter data register (DACD)
	D/A converter data register L (DACDL)

18.2.1 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 using the D/A converter, be sure to set bit 7 (DACEN) to 1.
The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 18 - 2 Format of Peripheral Enable Register 1 (PER1)

Address: F007AH After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	<3>	<2>	<1>	<0>
PER1	DACEN	TRGEN	AMPEN	0	DTCEN	PGAEN	AFEEN	TRJ0EN

DACEN	Control of input clock supplied to 12-bit D/A converter
0	Stops input clock supply. • SFRs used by the 12-bit D/A converter cannot be written. • The 12-bit D/A converter is in the reset status.
1	Enables input clock supply. • SFRs used by the 12-bit D/A converter can be read and written.

Caution 1. Be sure to set DACEN to 1 before setting up the D/A converter. When DACEN is 0, writing to the D/A converter control registers is ignored and the value read from these registers is the initial value.
Caution 2. Be sure to clear bit 4 to "0".

18.2.2 Analog front-end power supply selection register (AFEPWS)

The AFEPWS register is used to control the power supplied to the programmable gain instrumentation amplifier (PGA) and AFE reference voltage (ABGR) blocks.
The AFEPWS register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 18 - 3 Format of Analog Front-End Power Supply Selection Register (AFEPWS)

Address: F0440H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	1	<0>
AFEPWS	DACPON	AMP2PON	AMP1PON	AMP0PON	0	PGAPON	0	AFEPON

DACPON	Control of power supplied to 12-bit D/A converter block
0	Power-off (default)
1	Power-on

Caution Be sure to clear bits 1 and 3 to "0".
For the setting of bits 0, 2, 4 to 6, refer to 13.3.2 Analog front-end power supply selection register (AFEPWS) .

18.2.3 D/A converter mode register 0 (DACM0)

This register is used to enable or disable operation of the D/A converter.

The DACM0 register can be set by an 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 18 - 4 Format of D/A Converter Mode Register 0 (DACM0)

Address: F0480H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
DACM0	DACTMD	0	0	0	0	0	0	DACRES

DACTMD	Selection of trigger mode
0	Software trigger mode
1	Hardware trigger mode

DACRES	Selection of D/A converter resolution
0	12 bits
1	8 bits

18.2.4 D/A converter mode register 1 (DACM1)

This register is used to enable or disable operation of the D/A converter.
The DACM1 register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 18 - 5 Format of D/A Converter Mode Register 1 (DACM1)

Address: F0481H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
DACM1	0	0	0	0	0	0	0	DACVRF

DACVRF	Selection of D/A converter reference voltage
0	Select AV_{DD}.
1	Select internal reference voltage for D/A converter (VREFDA): 1.45 V.

18.2.5 D/A converter data registers (DACD, DACDL)

DACD is a 16-bit register that stores the data to be D/A-converted. DACDL is an 8-bit register that stores the data to be D/A-converted.
DACD is used for D/A conversion in 12-bit mode (DACRES = 0).
DACDL is used for D/A conversion in 8-bit mode (DACRES = 1).
The DACD register can be set by a 16-bit memory manipulation instruction.
The DACDL register can be set by an 8-bit memory manipulation instruction.
Reset signal input clears these registers to 0000H/00H.

Figure 18 - 6 Format of D/A Converter Data Registers (DACD, DACDL)

Address: F0482H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
DACD	0	0	0	0	DACDR11	DACDR10	DACDR9	DACDR8
	7	6	5	4	3	2	1	0
	DACDR7	DACDR6	DACDR5	DACDR4	DACDR3	DACDR2	DACDR1	DACDR0

Address: F0482H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
DACDL	DACDR7	DACDR6	DACDR5	DACDR4	DACDR3	DACDR2	DACDR1	DACDR0

18.3 Operation

18.3.1 Normal operation in software trigger mode

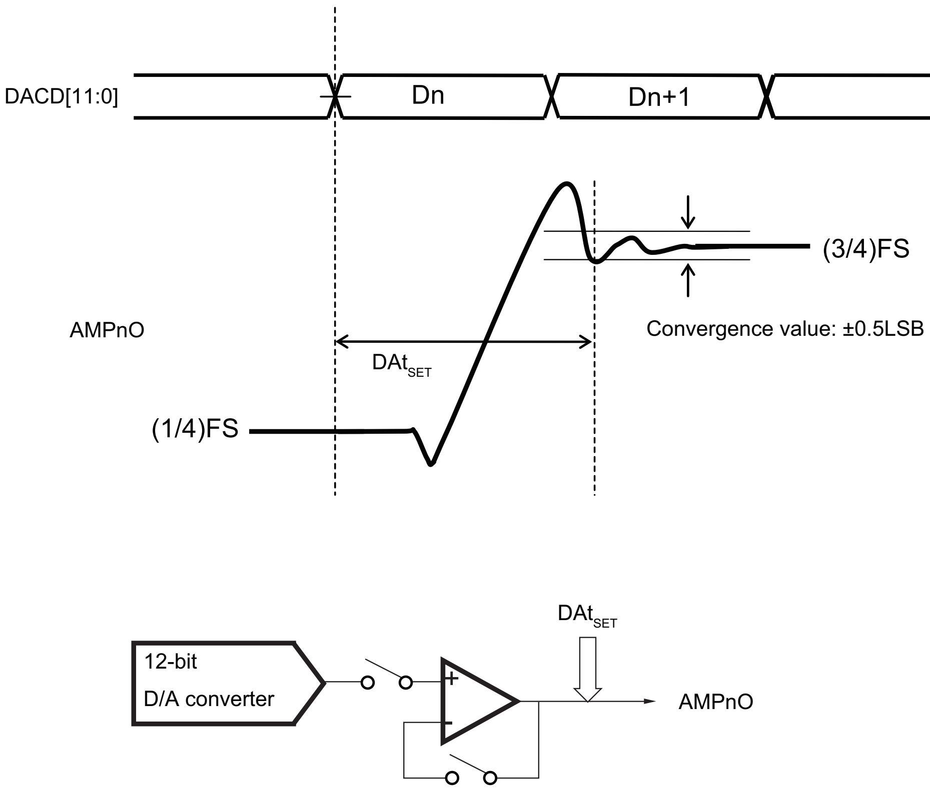
Writing to the DACD or DACDL register triggers D/A conversion.
The procedure for setting up the operation is shown below.

(1) Set the data to be D/A-converted in the DACD or DACDL register.
(2) D/A conversion starts. After the settling time (DAtset) has elapsed, the conversion result is output from the configurable amplifier output pin AMPnO (n = 0 to 2). This conversion result continues to be output until the value of the DACD or DACDL register is changed. For details about the output value, refer to **CHAPTER 33 ELECTRICAL SPECIFICATIONS (G: T_A = -40 to +105°C)** or **CHAPTER 34 ELECTRICAL SPECIFICATIONS (M: T_A = -40 to +125°C)**.
(3) D/A conversion starts immediately after the values of the DACD or DACDL register is changed. The conversion result is output after the settling time (DAtset) has elapsed.

Figure 18 - 7 shows the D/A conversion operation timing. Figure 18 - 8 shows the relationship between the digital inputs and analog outputs.

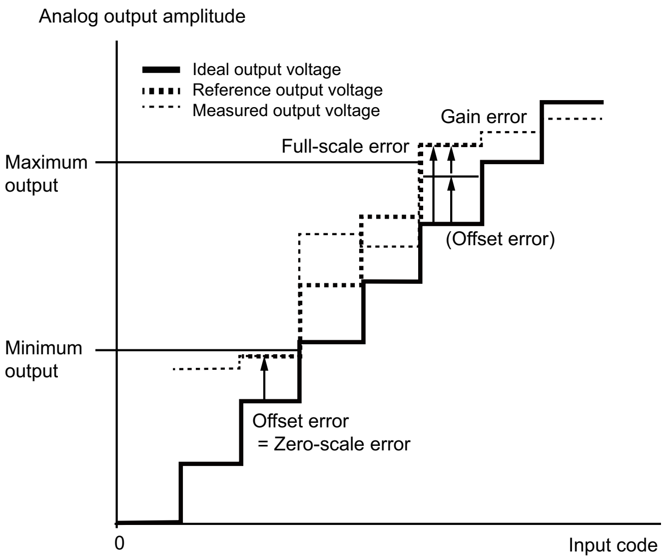
Caution The settling time refers to the time required for the voltage to stabilize when the voltage range is changed from 1/4 full scale to 3/4 full scale. The settling time in the 12-bit D/A converter is measured at the configurable amplifier output pin if the D/A converter is connected to a configurable amplifier positive input pin. The time until which the voltage stabilizes in the range of the reference voltage ± 0.5 LSB is measured, where the 3/4 full-scale voltage at timelike infinity is used as the reference voltage for stabilization. The settling time in the D/A converter includes the settling time in the configurable amplifier.

Figure 18 - 7 Example of 12-Bit D/A Converter Operation



Remark The above figure shows the case in which the D/A converter performs 12-bit conversion in software trigger mode (D/A converter mode register 0 (DACM0) = 00H).

Figure 18 - 8 Relationship Between the Digital Inputs and Analog Outputs



Caution 1. An offset error in the 12-bit D/A converter indicates the difference between the ideal output voltage and the reference output voltage when the reference output voltage is the minimum value (zero-scale).

Caution 2. A gain error in the 12-bit D/A converter indicates the difference between the ideal output voltage and the reference output voltage when the reference output voltage is the maximum value (full-scale), from which the offset error is subtracted.

18.3.2 Action to take when receiving an event signal in hardware trigger mode

Receiving an event signal from the event link controller (ELC) triggers D/A conversion.
The procedure for setting up the operation is shown below.

(1) Write 1 to the DACTMD bit to set hardware trigger mode.
(2) Set the data to be D/A-converted in the DACD or DACDL register. Unlike software trigger mode, writing to the DACD or DACDL register does not trigger D/A conversion.
(3) Set the event signal (07H) to be linked to the ELSELRn register of the event link controller (ELC).
(4) When the event link signal is asserted, D/A conversion starts and the conversion result is output from the configurable amplifier output pin AMPnO (n = 0 to 2) after the settling time (DAtset) has elapsed.
(5) To stop event linking to the 12-bit D/A converter, set the corresponding ELSELRn register to 00H.
(6) To perform D/A conversion successively in conjunction with events, set new data to be D/A converted to the DACD or DACDL register before the next event link signal is asserted.

18.3.3 Procedure for controlling 12-bit D/A converter

The figures below show the flowcharts for starting or stopping the 12-bit D/A converter.

Figure 18 - 9 Procedure for Starting the 12-bit D/A Converter

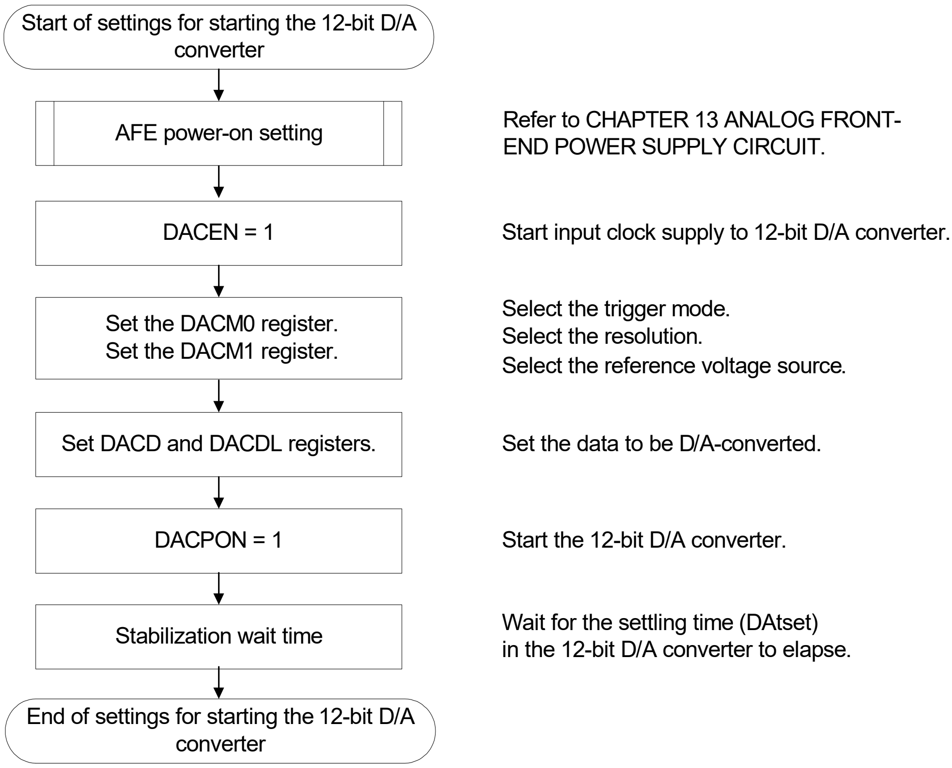
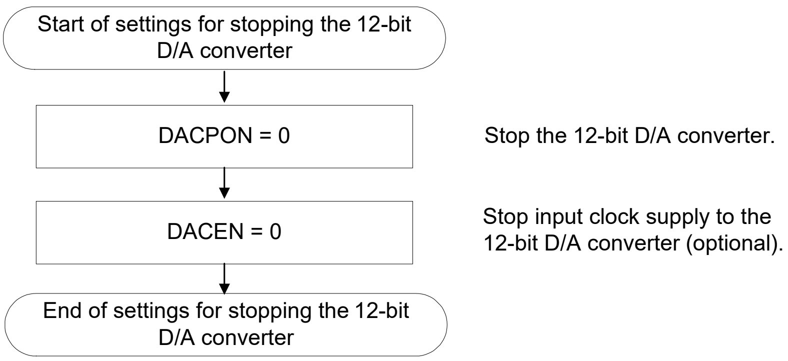


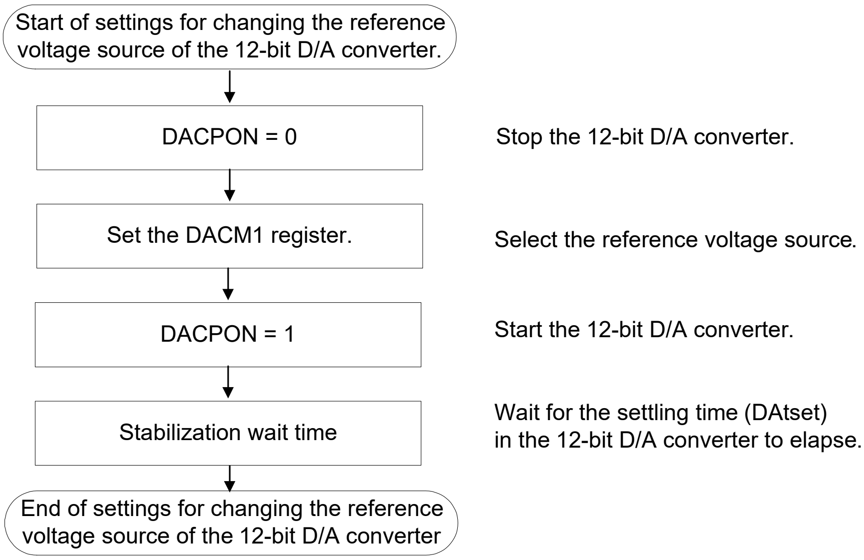
Figure 18 - 10 Procedure for Stopping the 12-bit D/A Converters



18.3.4 Changing the reference voltage source of the 12-bit D/A converter

When changing the reference voltage source for the 12-bit D/A converter, turn off power to the D/A converter. The flowchart of an example of the settings is shown below.

Figure 18 - 11 Procedure for Changing the Reference Voltage Source of the 12-bit D/A Converter



18.4 Cautions for 12-bit D/A Converter

18.4.1 12-bit D/A converter operation while the CPU is in the standby state

The 12-bit D/A converter can operate in any CPU standby mode. In the STOP mode and SNOOZE mode, the settings before entering the STOP mode are retained.

18.4.2 12-bit D/A converter operation while the AFE is in the standby state

The 12-bit D/A converter and configurable amplifier output Hi-Z when the power to the AFE is off (AFEPON = 0).

CHAPTER 19 SERIAL ARRAY UNIT

Serial array unit has four serial channels. All channels can achieve UART, and only channel 0 can achieve Simplified SPI (CSI [Note]) and simplified I^2C.
Function assignment of each channel supported by the RL78/I1E is as shown below.

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

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00 (supporting slave select input function)	UART0 (supporting LIN-bus) (supporting SNOOZE)	IIC00
	1	CSI01		IIC01
	2	—	UART1	—
	3	—		—

When “UART0” is used for channels 0 and 1, CSI00 and CSI01 cannot be used.

19.1 Functions of Serial Array Unit

Each serial interface supported by the RL78/I1E has the following features.

19.1.1 Simplified SPI (CSI00, CSI01)

Data is transmitted or received in synchronization with the serial clock (SCK) output from the master channel. simplified SPI 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 **19.5 Operation of Simplified SPI (CSI00, CSI01) Communication**.

[Data transmission/reception]
- Data length of 7 or 8 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable

[Clock control]
- Master/slave selection
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel
- Maximum transfer rate [Note]
  - During master communication: Max. $f_{CLK}/4$
  - During slave communication: Max. $f_{MCK}/12$

[Interrupt function]
- Transfer end interrupt/buffer empty interrupt

[Error detection flag]
- Overrun error

CSI00 supports the SNOOZE mode. When SCK input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible.

CSI00 also supports the slave select input function.

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics. For details, see **CHAPTER 33** or **CHAPTER 34 ELECTRICAL SPECIFICATIONS**.

19.1.2 UART (UART0, UART1)

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel). The LIN-bus can also be used by using a timer array unit with an external interrupt (INTP0). For details about the settings, see **19.7 Operation of UART (UART0, UART1) Communication**.

[Data transmission/reception]
- Data length of 7, 8, or 9 bits [Note]
- Select the MSB/LSB first
- Level setting of transmit/receive data and select of reverse
- Parity bit appending and parity check functions
- Stop bit appending

[Interrupt function]
- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]
- Framing error, parity error, or overrun error

UART0 supports the SNOOZE mode. When RxD input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible.

UART0 (channels 0 and 1 of unit 0) supports the LIN bus.

[LIN-bus functions]

• Wakeup signal detection • Break field (BF) detection • Sync field measurement, baud rate calculation	Using the external interrupt (INTP0) and timer array unit

Note The 9-bit data length can only be selected when using UART0.

19.1.3 Simplified I^{2}C (IIC00, IIC01)

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This simplified I^{2}C is designed for single communication with a device such as EEPROM, flash memory, or A/D converter, and therefore, it functions only as a master.
Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.
For details about the settings, see **19.9 Operation of Simplified I^{2}C (IIC00, IIC01) Communication**.

[Data transmission/reception]
• Master transmission, master reception (only master function with a single master)
• ACK output function Note and ACK detection function
• Data length of 8 bits (When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
• Manual generation of start condition and stop condition
[Interrupt function]
• Transfer end interrupt
[Error detection flag]
• ACK error or overrun error

* [Functions not supported by simplified I^{2}C]
• Slave transmission, slave reception
• Arbitration loss detection function
• Wait detection functions

Note When receiving the last data, ACK will not be output if 0 is written to the SOEmn bit (serial output enable register m (SOEm)) and serial communication data output is stopped. See the processing flow in **19.9.3 (2)** for details.

19.2 Configuration of Serial Array Unit

The serial array unit includes the following hardware.

Table 19 - 1 Configuration of Serial Array Unit

Item	Configuration
Shift register	8 bits or 9 bits [Note 1]
Buffer register	Lower 8 bits or 9 bits of serial data register mn (SDRmn) [Notes 1, 2]
Serial clock I/O	SCK00 and SCK01 pins (for Simplified SPI), SCL00 and SCL01 pins (for simplified I^2C)
Serial data input	SI00 and SI01 pins (for Simplified SPI), RxD0 pin (for UART supporting LIN-bus), RxD1 pin (for UART)
Serial data output	SO00 and SO01 pins (for Simplified SPI), TxD0 pin (for UART supporting LIN-bus), TxD1 pin (for UART)
Serial data I/O	SDA00 and SDA01 pins (for simplified I^2C)
Slave select input	$\overline{SSI00}$ pin (for slave select input function)
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Serial clock select register m (SPSm) • Serial channel enable status register m (SEm) • Serial channel start register m (SSm) • Serial channel stop register m (STm) • Serial output enable register m (SOEm) • Serial output register m (SOm) • Serial output level register m (SOLm) • Serial standby control register m (SSCm) • Input switch control register (ISC) • Noise filter enable register 0 (NFEN0)
	<Registers of each channel> • Serial data register mn (SDRmn) • Serial mode register mn (SMRmn) • Serial communication operation setting register mn (SCRmn) • Serial status register mn (SSRmn) • Serial flag clear trigger register mn (SIRmn)
	• Port input mode register 1 (PIM1) • Port output mode register 1 (POM1) • Port mode register 1 (PM1) • Port register 1 (P1)

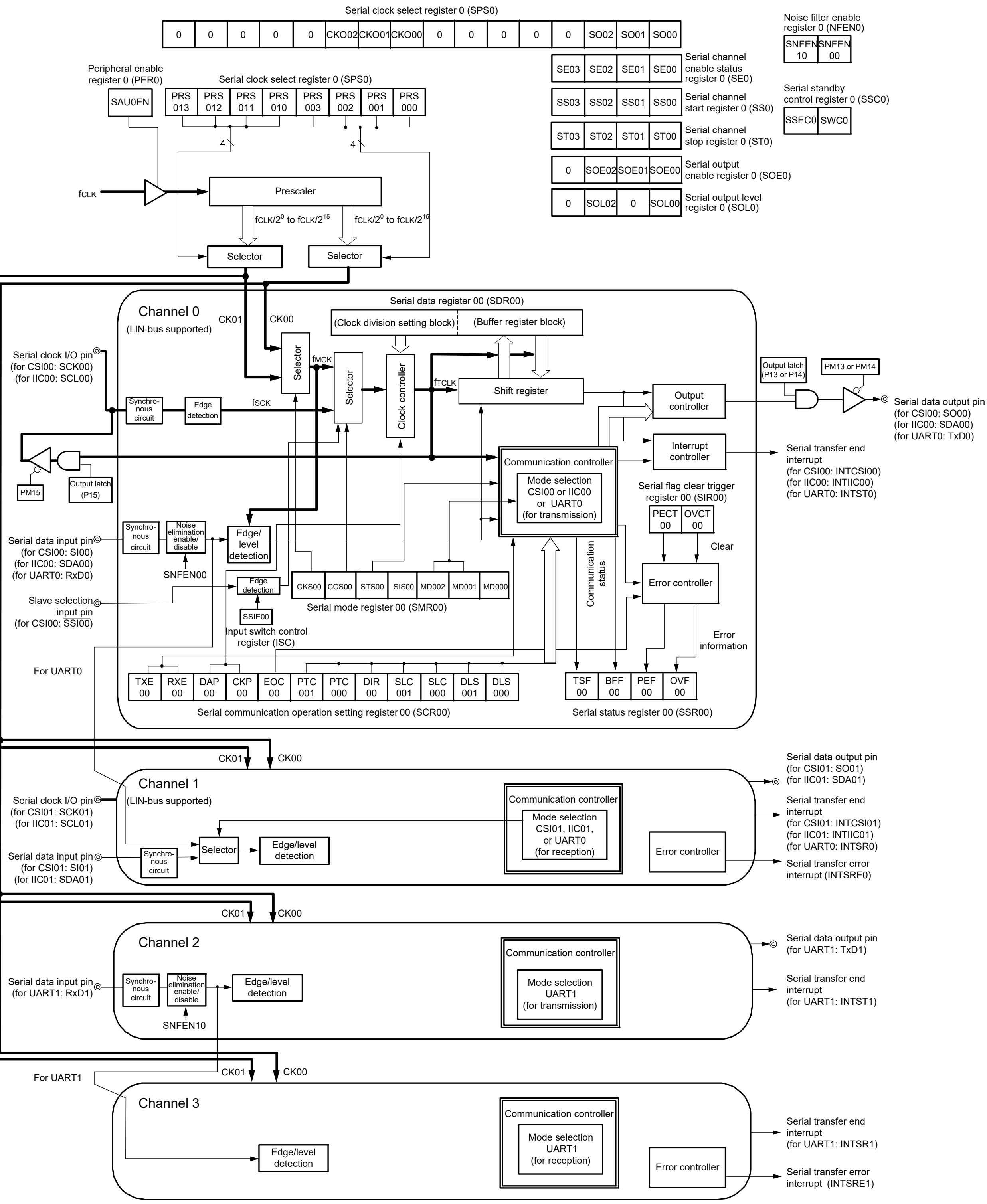
Note 1. The number of bits used as the shift register and buffer register differs depending on the unit and channel.
- Channels 0, 1: mn = 00, 01: lower 9 bits
- Other than above: lower 8 bits

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), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01),
q: UART number (q = 0, 1), r: IIC number (r = 00, 01)

Figure 19 - 1 shows the Block Diagram of Serial Array Unit 0.

Figure 19 - 1 Block Diagram of Serial Array Unit 0



19.2.1 Shift register

This is a 9-bit register that converts parallel data into serial data or vice versa.

In case of the UART communication of nine bits of data, nine bits (bits 0 to 8) are used [Note].

During reception, it converts data input to the serial pin into parallel data.

When data is transmitted, the value set to this register is output as serial data from the serial output pin.

The shift register cannot be directly manipulated by program.

To read or write the shift register, use the lower 8/9 bits of serial data register mn (SDRmn).

	8	7	6	5	4	3	2	1	0
Shift register									

Note The 9-bit data length can only be selected when using UART0.

19.2.2 Lower 8/9 bits of the serial data register mn (SDRmn)

The SDRmn register is the transmit/receive data register (16 bits) of channel n. Bits 8 to 0 (lower 9 bits) [Note 1] or bits 7 to 0 (lower 8 bits) function as a transmit/receive buffer register, and bits 15 to 9 are used as a register that sets the division ratio of the operation clock (f_{MCK}).

When data is received, parallel data converted by the shift register is stored in the lower 8/9 bits. When data is to be transmitted, set transmit data to be transferred to the shift register to the lower 8/9 bits.

The data stored in the lower 8/9 bits of this register is as follows, depending on the setting of bits 0 and 1 (DLSmn0, DLSmn1) of serial communication operation setting register mn (SCRmn), regardless of the output sequence of the data.

- 7-bit data length (stored in bits 0 to 6 of SDRmn register)
- 8-bit data length (stored in bits 0 to 7 of SDRmn register)
- 9-bit data length (stored in bits 0 to 8 of SDRmn register) [Note 1]

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

The lower 8/9 bits of the SDRmn register can be read or written [Note 2] as the following 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)

Reset signal generation clears the SDRmn register to 0000H.

Note 1. The 9-bit data length can only be selected when using UART0.

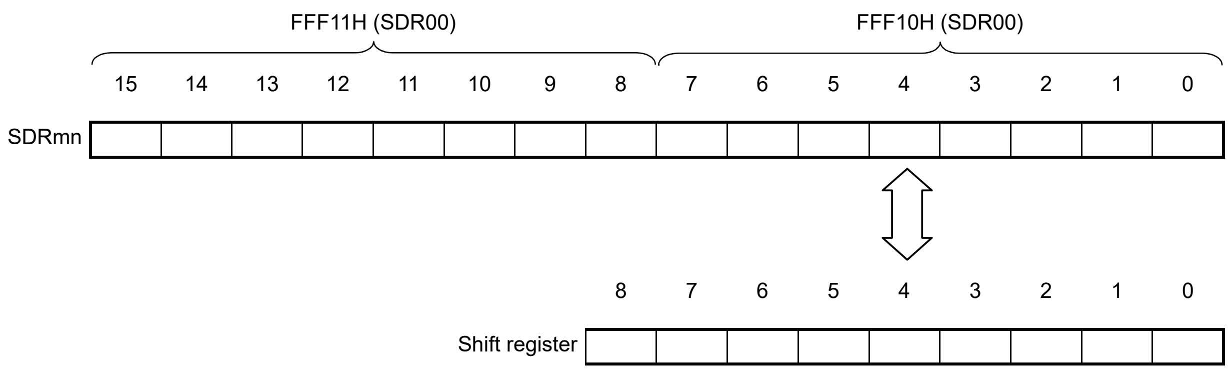
Note 2. When operation is stopped (SEmn = 0), do not rewrite SDRmn[7:0] by an 8-bit memory manipulation instruction (SDRmn[15:9] are all cleared to 0).

Remark 1. After data is received, “0” is stored in bits 0 to 8 in bit portions that exceed the data length.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01), q: UART number (q = 0, 1), r: IIC number (r = 00, 01)

Figure 19 - 2 Format of Serial data register mn (SDRmn) (mn = 00, 01)

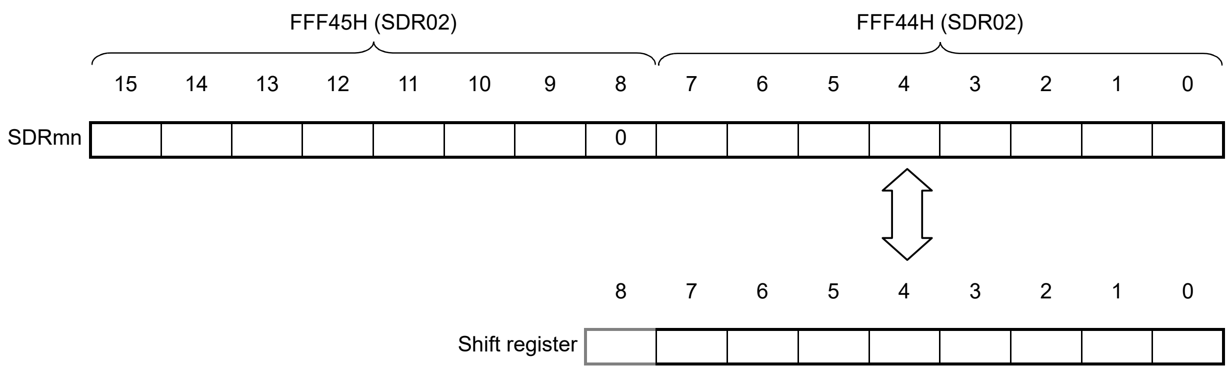
Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01) After reset: 0000H R/W



Remark For the function of the higher 7 bits of the SDRmn register, see **19.3 Registers Controlling Serial Array Unit**.

Figure 19 - 3 Format of Serial data register mn (SDRmn) (mn = 02, 03)

Address: FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03) After reset: 0000H R/W



Caution Be sure to clear bit 8 to "0".

Remark For the function of the higher 7 bits of the SDRmn register, see **19.3 Registers Controlling Serial Array Unit**.

19.3 Registers Controlling Serial Array Unit

Serial array unit is controlled by the following registers.

- 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)
- Serial standby control register m (SSCm)
- Input switch control register (ISC)
- Noise filter enable register 0 (NFEN0)
- Port input mode register 1 (PIM1)
- Port output mode register 1 (POM1)
- Port mode register 1 (PM1)
- Port register 1 (P1)

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

19.3.1 Peripheral enable register 0 (PER0)

PER0 is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.
When serial array unit 0 is used, be sure to set bit 2 (SAU0EN) of this register to 1.
The PER0 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears the PER0 register to 00H.

Figure 19 - 4 Format of Peripheral enable register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	6	<5>	4	3	<2>	<1>	<0>
PER0	RTCEN	0	ADCEN	0	0	SAU0EN	TAU1EN	TAU0EN

SAUmEN	Control of input clock supplied to serial array unit m
0	Stops supply of input clock. • SFRs used by serial array unit m cannot be written. • Serial array unit m is in the reset status.
1	Enables input clock supply. • SFRs used by serial array unit m can be read/written.

Caution 1. When setting serial array unit m, be sure to first set the following registers with the SAUmEN bit set to 1. If SAUmEN = 0, writing to a control register of serial array unit m is ignored, and, even if the register is read, only the default value is read (except for the input switch control register (ISC), noise filter enable register 0 (NFEN0), port input mode register 1 (PIM1), port output mode register 1 (POM1), port mode register 1 (PM1), and port register 1 (P1).

- **Serial clock select register m (SPSm)**
- **Serial mode register mn (SMRmn)**
- **Serial communication operation setting register mn (SCRmn)**
- **Serial data register mn (SDRmn)**
- **Serial flag clear trigger register mn (SIRmn)**
- **Serial status register mn (SSRmn)**
- **Serial channel start register m (SSm)**
- **Serial channel stop register m (STm)**
- **Serial channel enable status register m (SEm)**
- **Serial output enable register m (SOEm)**
- **Serial output level register m (SOLm)**
- **Serial output register m (SOm)**
- **Serial standby control register m (SSCm)**

Caution 2. Be sure to clear bits 3, 4, and 6 to “0”.

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

19.3.2 Serial clock select register m (SPSm)

The SPSm register is a 16-bit register that is used to select two types of operation clocks (CKm0, CKm1) that are commonly supplied to each channel. CKm1 is selected by bits 7 to 4 of the SPSm register, and CKm0 is selected by bits 3 to 0.
Rewriting the SPSm register is prohibited when the register is in operation (when SEmn = 1).
The SPSm register can be set by a 16-bit memory manipulation instruction.
The lower 8 bits of the SPSm register can be set with an 8-bit memory manipulation instruction by using SPSmL.
Reset signal generation clears the SPSm register to 0000H.

Figure 19 - 5 Format of Serial clock select register m (SPSm)

Address: F0126H, F0127H (SPS0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPSm	0	0	0	0	0	0	0	0	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRS mk3	PRS mk2	PRS mk1	PRS mk0		Section of operation clock (CKmk) Note				
					fCLK = 2 MHz	fCLK = 5 MHz	fCLK = 10 MHz	fCLK = 20 MHz	fCLK = 32 MHz
0	0	0	0	fCLK	2 MHz	5 MHz	10 MHz	20 MHz	32 MHz
0	0	0	1	fCLK/2	1 MHz	2.5 MHz	5 MHz	10 MHz	16 MHz
0	0	1	0	fCLK/2^2	500 kHz	1.25 MHz	2.5 MHz	5 MHz	8 MHz
0	0	1	1	fCLK/2^3	250 kHz	625 kHz	1.25 MHz	2.5 MHz	4 MHz
0	1	0	0	fCLK/2^4	125 kHz	313 kHz	625 kHz	1.25 MHz	2 MHz
0	1	0	1	fCLK/2^5	62.5 kHz	156 kHz	313 kHz	625 kHz	1 MHz
0	1	1	0	fCLK/2^6	31.3 kHz	78.1 kHz	156 kHz	313 kHz	500 kHz
0	1	1	1	fCLK/2^7	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz	250 kHz
1	0	0	0	fCLK/2^8	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz	125 kHz
1	0	0	1	fCLK/2^9	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz	62.5 kHz
1	0	1	0	fCLK/2^{10}	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz	31.3 kHz
1	0	1	1	fCLK/2^{11}	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz	15.6 kHz
1	1	0	0	fCLK/2^{12}	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz	7.8 kHz
1	1	0	1	fCLK/2^{13}	244 Hz	610 Hz	1.22 kHz	2.44 kHz	3.9 kHz
1	1	1	0	fCLK/2^{14}	122 Hz	305 Hz	610 Hz	1.22 kHz	1.95 kHz
1	1	1	1	fCLK/2^{15}	61 Hz	153 Hz	305 Hz	610 Hz	977 Hz

Note When changing the clock selected for fCLK (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. fCLK: CPU/peripheral hardware clock frequency
Remark 2. m: Unit number (m = 0)
Remark 3. k = 0, 1

19.3.3 Serial mode register mn (SMRmn)

The SMRmn register is a register that sets an operation mode of channel n. It is also used to select an operation clock (f_{MCK}), specify whether the serial clock (f_{SCK}) may be input or not, set a start trigger, an operation mode (Simplified SPI (CSI), UART, or simplified I[2]C), and an interrupt source. This register is also used to invert the level of the receive data only in the UART mode.

Rewriting the SMRmn register is prohibited when the register is in operation (when SEmn = 1). However, the MDmn0 bit can be rewritten during operation.

The SMRmn register can be set by a 16-bit memory manipulation instruction.

Reset signal generation sets the SMRmn register to 0020H.

Figure 19 - 6 Format of Serial mode register mn (SMRmn) (1/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03) After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn	CCSmn	0	0	0	0	0	STSmn Note	0	SISmn0 Note	1	0	0	MDmn2	MDmn1	MDmn0

CKSmn	Selection of operation clock (f_{MCK}) of channel n
0	Operation clock CKm0 set by the SPSm register
1	Operation clock CKm1 set by the SPSm register
Operation clock (f_{MCK}) is used by the edge detector. In addition, depending on the setting of the CCSmn bit and the higher 7 bits of the SDRmn register, a transfer clock (f_{TCLK}) is generated.	

CCSmn	Selection of transfer clock (f_{TCLK}) of channel n
0	Divided operation clock f_{MCK} specified by the CKSmn bit
1	Clock input f_{SCK} from the SCKp pin (slave transfer in Simplified SPI (CSI) mode)
Transfer clock f_{TCLK} is used for the shift register, communication controller, output controller, interrupt controller, and error controller. When CCSmn = 0, the division ratio of operation clock (f_{MCK}) is set by the higher 7 bits of the SDRmn register.	

STSmn Note	Selection of start trigger source
0	Only software trigger is valid (selected for Simplified SPI (CSI), UART transmission, and simplified I[2]C).
1	Valid edge of the RxDq pin (selected for UART reception)
Transfer is started when the above source is satisfied after 1 is set to the SSm register.	

Note The SMR01 and SMR03 registers only.

Caution Be sure to clear bits 13 to 9, 7, 4, and 3 (or bits 13 to 6, 4, and 3 in the SMR00 and SMR02 registers) to “0”. Be sure to set bit 5 to “1”.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01), q: UART number (q = 0, 1), r: IIC number (r = 00, 01)

Figure 19 - 7 Format of Serial mode register mn (SMRmn) (2/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03), After reset: 0020H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKS mn	CCS mn	0	0	0	0	0	STS mn Note	0	SIS mn0 Note	1	0	0	MD mn2	MD mn1	MD mn0

SIS mn0 Note	Controls inversion of level of receive data of channel n in UART mode
0	Falling edge is detected as the start bit. The input communication data is captured as is.
1	Rising edge is detected as the start bit. The input communication data is inverted and captured.

MD mn2	MD mn1	Setting of operation mode of channel n
0	0	Simplified SPI (CSI) mode
0	1	UART mode
1	0	Simplified I^2C mode
1	1	Setting prohibited

MD mn0	Selection of interrupt source of channel n
0	Transfer end interrupt
1	Buffer empty interrupt (Occurs when data is transferred from the SDRmn register to the shift register.)
For successive transmission, the next transmit data is written by setting the MDmn0 bit to 1 when SDRmn data has run out.	

Note The SMR01 and SMR03 registers only.

Caution Be sure to clear bits 13 to 9, 7, 4, and 3 (or bits 13 to 6, 4, and 3 in the SMR00 and SMR02 registers) to “0”. Be sure to set bit 5 to “1”.

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01),
q: UART number (q = 0, 1), r: IIC number (r = 00, 01)

19.3.4 Serial communication operation setting register mn (SCRmn)

The SCRmn register is a communication operation setting register of channel n. It is used to set a data transmission/reception mode, phase of data and clock, whether an error signal is to be masked or not, parity bit, start bit, stop bit, and data length.

Rewriting the SCRmn register is prohibited when the register is in operation (when SEmn = 1).

The SCRmn register can be set by a 16-bit memory manipulation instruction.

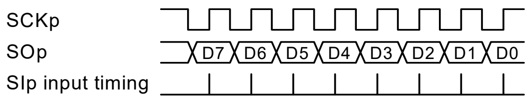
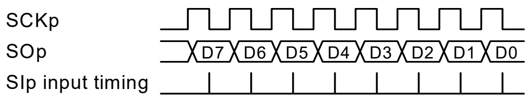
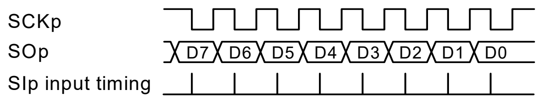
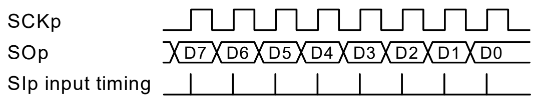
Reset signal generation sets the SCRmn register to 0087H.

Figure 19 - 8 Format of Serial communication operation setting register mn (SCRmn) (1/2)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLCm n1 Note 1	SLC mn0	0	1	DLSm n1 Note 2	DLS mn0

TXE mn	RXE mn	Setting of operation mode of channel n
0	0	Disable communication.
0	1	Reception only
1	0	Transmission only
1	1	Transmission/reception

DAP mn	CKP mn	Selection of data and clock phase in Simplified SPI (CSI) mode	Type
0	0		1
0	1		2
1	0		3
1	1		4
Be sure to set DAPmn, CKPmn = 0, 0 in the UART mode and simplified I²C mode.			

EOC mn	Mask control of error interrupt signal (INTSREx (x = 0 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 I²C mode, and during UART transmission Note 3.	

Note 1. The SCR00 and SCR02 registers only.
Note 2. The SCR00 and SCR01 registers only. Others are fixed to 1.
Note 3. When using CSImn without clearing EOCmn to 0, error interrupt INTSREn may be generated.

Caution **Be sure to clear bits 3, 6, and 11 to "0". (Also clear bit 5 of the SCR01 and SCR03 registers to 0). Be sure to set bit 2 to "1".**

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01)

Figure 19 - 9 Format of Serial communication operation setting register mn (SCRmn) (2/2)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLCm n1 Note 1	SLC mn0	0	1	DLSm n1 Note 2	DLS mn0

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 3.	No parity judgment
1	0	Outputs even parity.	Judged as even parity.
1	1	Outputs odd parity.	Judges as odd parity.
Be sure to set PTCmn1, PTCmn0 = 0, 0 in the Simplified SPI (CSI) mode and simplified I^{2}C mode.			

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^{2}C mode.	

SLCmn1 Note 1	SLCmn0	Setting of stop bit in UART mode
0	0	No stop bit
0	1	Stop bit length = 1 bit
1	0	Stop bit length = 2 bits (mn = 00, 02, 10, 12 only)
1	1	Setting prohibited
When the transfer end interrupt is selected, the interrupt is generated when all stop bits have been completely transferred. Specify 1 bit (SLCmn1, SLCmn0 = 0, 1) during UART reception and in the simplified I^{2}C mode. Specify no stop bit (SLCmn1, SLCmn0 = 0, 0) in the Simplified SPI (CSI) mode. Specify 1 bit (SLCmn1, SLCmn0 = 0, 1) or 2 bits (SLCmn1, SLCmn0 = 1, 0) during UART transmission.		

DLSmn1 Note 2	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^{2}C mode.		

Note 1. The SCR00 and SCR02 registers only.
Note 2. The SCR00 and SCR01 registers only. Others are fixed to 1.
Note 3. 0 is always added regardless of the data contents.

Caution **Be sure to clear bits 3, 6, and 11 to "0". (Also clear bit 5 of the SCR01 and SCR03 registers to 0). Be sure to set bit 2 to "1".**

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 01)

19.3.5 Serial data register mn (SDRmn)

The SDRmn register is the transmit/receive data register (16 bits) of channel n.

Bits 8 to 0 (lower 9 bits) of SDR00 and SDR01 or bits 7 to 0 (lower 8 bits) of SDR02 and SDR03 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 operating clock by the higher 7 bits of the SDRmn register is used as the transfer clock.

If the CCSmn bit of serial mode register mn (SMRmn) is set to 1, set bits 15 to 9 (upper 7 bits) of SDR00 and SDR01 to 0000000B. The input clock f_{SCK} (slave transfer in Simplified SPI (CSI) mode) from the SCKp pin is used as the transfer clock.

The lower 8/9 bits of the SDRmn register function as a transmit/receive buffer register. During reception, the parallel data converted by the shift register is stored in the lower 8/9 bits, and during transmission, the data to be transmitted to the shift register is set to the lower 8/9 bits.

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

However, the higher 7 bits can be written or read only when the operation is stopped (SEmn = 0). During operation (SEmn = 1), a value is written only to the lower 8/9 bits of the SDRmn register. When the SDRmn register is read during operation, the higher 7 bits are always read as 0.

Reset signal generation clears the SDRmn register to 0000H.

Figure 19 - 10 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), FFF46H, FFF47H (SDR03) 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								

SDRmn[15:9]							Transfer clock set by dividing the operating clock
0	0	0	0	0	0	0	$f_{MCK}/2$
0	0	0	0	0	0	1	$f_{MCK}/4$
0	0	0	0	0	1	0	$f_{MCK}/6$
0	0	0	0	0	1	1	$f_{MCK}/8$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1	1	1	1	1	1	0	$f_{MCK}/254$
1	1	1	1	1	1	1	$f_{MCK}/256$

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

Caution 1. Be sure to clear bit 8 of the SDR02 and SDR03 registers to “0”.

Caution 2. Setting SDRmn[15:9] = (0000000B, 0000001B) is prohibited when UART is used.

Caution 3. Setting SDRmn[15:9] = 0000000B is prohibited when simplified I²C is used. Set SDRmn[15:9] to 0000001B or greater.

Caution 4. When operation is stopped (SEmn = 0), do not rewrite SDRmn [7:0] by an 8-bit memory manipulation instruction (SDRmn [15:9] are all cleared to 0).

Remark 1. For the function of the lower 8/9 bits of the SDRmn register, see **19.2 Configuration of Serial Array Unit**.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 3)

19.3.6 Serial flag clear trigger register mn (SIRmn)

The SIRmn register is a trigger register that is used to clear each error flag of channel n.
When each bit (FECTmn, PECTmn, OVCTmn) of this register is set to 1, the corresponding bit (FEFmn, PEFmn, OVFmn) of serial status register mn is cleared to 0. Because the SIRmn register is a trigger register, it is cleared immediately when the corresponding bit of the SSRmn register is cleared.
The SIRmn register can be set by a 16-bit memory manipulation instruction.
The lower 8 bits of the SIRmn register can be set with an 8-bit memory manipulation instruction by using SIRmnL.
Reset signal generation clears the SIRmn register to 0000H.

Figure 19 - 11 Format of Serial flag clear trigger register mn (SIRmn)

Address: F0108H, F0109H (SIR00) to F010EH, F010FH (SIR03) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIRmn	0	0	0	0	0	0	0	0	0	0	0	0	0	FECTmn Note	PECTmn	OVCTmn

FECTmn Note	Clear trigger of framing error of channel n
0	Not cleared
1	Clears the FEFmn bit of the SSRmn register to 0.

PECTmn	Clear trigger of parity error flag of channel n
0	Not cleared
1	Clears the PEFmn bit of the SSRmn register to 0.

OVCTmn	Clear trigger of overrun error flag of channel n
0	Not cleared
1	Clears the OVFmn bit of the SSRmn register to 0.

Note The SIR01 and SIR03 registers only.

Caution Be sure to clear bits 15 to 3 (or bits 15 to 2 of the SIR00 and SIR02 registers) to “0”.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)
Remark 2. When the SIRmn register is read, 0000H is always read.

19.3.7 Serial status register mn (SSRmn)

The SSRmn register is a register that indicates the communication status and error occurrence status of channel n. The errors indicated by this register are a framing error, parity error, and overrun error.
The SSRmn register can be read by a 16-bit memory manipulation instruction.
The lower 8 bits of the SSRmn register can be set with an 8-bit memory manipulation instruction by using SSRmnL.
Reset signal generation clears the SSRmn register to 0000H.

Figure 19 - 12 Format of Serial status register mn (SSRmn) (1/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03), After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn Note	PEF mn	OVF mn

TSF mn	Communication status indication flag of channel n
0	Communication is stopped or suspended.
1	Communication is in progress.
<Clear conditions> • The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is suspended). • Communication ends. <Set condition> • Communication starts.	

BFF mn	Buffer register status indication flag of channel n
0	Valid data is not stored in the SDRmn register.
1	Valid data is stored in the SDRmn register.
<Clear conditions> • Transferring transmit data from the SDRmn register to the shift register ends during transmission. • Reading receive data from the SDRmn register ends during reception. • The STmn bit of the STm register is set to 1 (communication is stopped) or the SSmn bit of the SSm register is set to 1 (communication is enabled). <Set conditions> • Transmit data is written to the SDRmn register while the TXEmn bit of the SCRmn register is set to 1 (transmission or transmission and reception mode in each communication mode). • Receive data is stored in the SDRmn register while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode). • A reception error occurs.	

Note The SSR01 and SSR03 registers only.

Caution **When the Simplified SPI (CSI) is performing reception operations in the SNOOZE mode (SWCm = 1), the BFFmn flag will not change.**

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

Figure 19 - 13 Format of Serial status register mn (SSRmn) (2/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSF mn	BFF mn	0	0	FEF mn Note	PEF mn	OVF mn

FEF mn Note	Framing error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception).

<Clear condition>
- 1 is written to the FECTmn bit of the SIRmn register.

<Set condition>
- A stop bit is not detected when UART reception ends.

PEF mn	Parity error detection flag of channel n
0	No error occurs.
1	An error occurs (during UART reception) or ACK is not detected (during I^{2}C transmission).

<Clear condition>
- 1 is written to the PECTmn bit of the SIRmn register.

<Set condition>
- The parity of the transmit data and the parity bit do not match when UART reception ends (parity error).
- No ACK signal is returned from the slave channel at the ACK reception timing during I^{2}C transmission (ACK is not detected).

OVF mn	Overrun error detection flag of channel n
0	No error occurs.
1	An error occurs

<Clear condition>
- 1 is written to the OVCTmn bit of the SIRmn register.

<Set condition>
- Even though receive data is stored in the SDRmn register, that data is not read and transmit data or the next receive data is written while the RXEmn bit of the SCRmn register is set to 1 (reception or transmission and reception mode in each communication mode).
- Transmit data is not ready for slave transmission or transmission and reception in Simplified SPI (CSI) mode.

Note The SSR01 and SSR03 registers only.

Caution 1. If data is written to the SDRmn register when BFFmn = 1, the transmit/receive data stored in the register is discarded and an overrun error (OVEmn = 1) is detected.

Caution 2. When the Simplified SPI (CSI) is performing reception operations in the SNOOZE mode (SWCm = 1), the OVFmn flag will not change.

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

19.3.8 Serial channel start register m (SSm)

The SSm register is a trigger register that is used to enable starting communication/count by each channel.
When 1 is written to a bit of this register (SSmn), the corresponding bit (SEmn) of serial channel enable status register m (SEm) is set to 1 (Operation is enabled). Because the SSmn bit is a trigger bit, it is cleared immediately when SEmn is set to 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 by using SSmL.
Reset signal generation clears the SSm register to 0000H.

Figure 19 - 14 Format of Serial channel start register m (SSm)

Address: F0122H, F0123H (SS0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SS0	0	0	0	0	0	0	0	0	0	0	0	0	SS03	SS02	SS01	SS00

SSmn	Operation start trigger of channel n
0	No trigger operation
1	Sets the SEmn bit to 1 and enters the communication wait status Note.

Note If set the SSmn = 1 to during a communication operation, will wait status to stop the communication.
At this time, holding status value of control register and shift register, SCKmn and SOmn pins, and FEFmn, PEFmn, OVFmn flags.

Caution 1. Be sure to clear bits 15 to 4 of the SS0 register to “0”.
Caution 2. For the UART reception, set the RXEmn bit of SCRmn register to 1, and then set SSmn to 1 after 4 or more fMCK clocks have elapsed.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)
Remark 2. When the SSm register is read, 0000H is always read.

19.3.9 Serial channel stop register m (STm)

The STm register is a trigger register that is used to enable stopping communication/count by each channel.
When 1 is written to a bit of this register (STmn), the corresponding bit (SEmn) of serial channel enable status register m (SEm) is cleared to 0 (operation is stopped). Because the STmn bit is a trigger bit, it is cleared immediately when SEmn is cleared to 0.
The STm register can set written by a 16-bit memory manipulation instruction.
The lower 8 bits of the STm register can be set with a 1-bit or 8-bit memory manipulation instruction by using STmL.
Reset signal generation clears the STm register to 0000H.

Figure 19 - 15 Format of Serial channel stop register m (STm)

Address: F0124H, F0125H (ST0)   After reset: 0000H   R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST0	0	0	0	0	0	0	0	0	0	0	0	0	ST03	ST02	ST01	ST00

STmn	Operation stop trigger of channel n
0	No trigger operation
1	Clears the SEmn bit to 0 and stops the communication operation Note.

Note Holding status value of the control register and shift register, the SCKmn and SOmn pins, and FEFmn, PEFmn, OVFmn flags.

Caution Be sure to clear bits 15 to 4 of the ST0 register to "0".

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0 to 3)
Remark 2. When the STm register is read, 0000H is always read.

19.3.10 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.
Channel n that is enabled to operate cannot rewrite by software the value of the CKOmn bit (serial clock output of channel n) of serial output register m (SOm), and a value reflected by a communication operation is output from the serial clock pin.
Channel n that stops operation can set the value of the CKOmn bit of the SOm register by software and output its value from the serial clock pin. In this way, any waveform, such as that of a start condition/stop condition, can be created by software.
The SEm register can be read by a 16-bit memory manipulation instruction.
The lower 8 bits of the SEm register can be set with a 1-bit or 8-bit memory manipulation instruction by using SEmL.
Reset signal generation clears the SEm register to 0000H.

Figure 19 - 16 Format of Serial channel enable status register m (SEm)

Address: F0120H, F0121H (SE0) After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE0	0	0	0	0	0	0	0	0	0	0	0	0	SE03	SE02	SE01	SE00

SEmn	Indication of operation enable/stop status of channel n
0	Operation stops
1	Operation is enabled.

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

19.3.11 Serial output enable register m (SOEm)

The SOEm register is a register that is used to enable or stop output of the serial communication operation of each channel.

Channel n that enables serial output cannot rewrite by software the value of the SOmn bit of serial output register m (SOm), and a value reflected by a communication operation is output from the serial data output pin.

For channel n, whose serial output is stopped, the SOmn bit value of the SOm register can be set by software, and that value can be output from the serial data output pin. In this way, any waveform of the start condition and stop condition can be created by software.

The SOEm register can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of the SOEm register can be set with a 1-bit or 8-bit memory manipulation instruction by using SOEmL.

Reset signal generation clears the SOEm register to 0000H.

Figure 19 - 17 Format of Serial output enable register m (SOEm)

Address: F012AH, F012BH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE02	SOE01	SOE00

SOEmn	Serial output enable/stop of channel n
0	Stops output by serial communication operation.
1	Enables output by serial communication operation.

Caution Be sure to clear bits 15 to 3 of the SOE0 register to “0”.

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

19.3.12 Serial output register m (SOm)

The SOm register is a buffer register for serial output of each channel.
The value of the SOmn bit of this register is output from the serial data output pin of channel n.
The value of the CKOmn bit of this register is output from the serial clock output pin of channel n.
The SOmn bit of this register can be rewritten by software only when serial output is disabled (SOEmn = 0). When serial output is enabled (SOEmn = 1), rewriting by software is ignored, and the value of the register can be changed only by a serial communication operation.
The CKOmn bit of this register can be rewritten by software only when the channel operation is stopped (SEmn = 0). While channel operation is enabled (SEmn = 1), rewriting by software is ignored, and the value of the CKOmn bit can be changed only by a serial communication operation.
To use a pin for the serial interface as a port function pin other than a serial interface function pin, set the corresponding the CKOmn and SOmn bits to 1.
The SOm register can be set by a 16-bit memory manipulation instruction.
Reset signal generation clears the SOm register to 0F0FH.

Figure 19 - 18 Format of Serial output register m (SOm)

Address: F0128H, F0129H After reset: 0F0FH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO0	0	0	0	0	1	1	CKO01	CKO00	0	0	0	0	1	SO02	SO01	SO00

CKOmn	Serial clock output of channel n
0	Serial clock output value is “0”.
1	Serial clock output value is “1”.

SOmn	Serial data output of channel n
0	Serial data output value is “0”.
1	Serial data output value is “1”.

Caution **Be sure to clear bits 15 to 12 and 7 to 4 of the SO0 register to “0”.**
Be sure to set bits 11, 10, and 3 of the SO0 register to “1”.

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

19.3.13 Serial output level register m (SOLm)

The SOLm register is a register that is used to set inversion of the data output level of each channel.
This register can be set only in the UART mode. Be sure to set 0 for corresponding bit in the Simplified SPI (CSI) mode and simplifies I^2C mode.
Inverting channel n by using this register is reflected on pin output only when serial output is enabled (SOEmn = 1). When serial output is disabled (SOEmn = 0), the value of the SOmn bit is output as is.
Rewriting the SOLm register is prohibited when the register is in operation (when SEmn = 1).
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 by using SOLmL.
Reset signal generation clears the SOLm register to 0000H.

Figure 19 - 19 Format of Serial output level register m (SOLm)

Address: F0134H, F0135H (SOL0) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOL0	0	0	0	0	0	0	0	0	0	0	0	0	0	SOL02	0	SOL00

SOLmn	Selects inversion of the level of the transmit data of channel n in UART mode
0	Communication data is output as is.
1	Communication data is inverted and output.

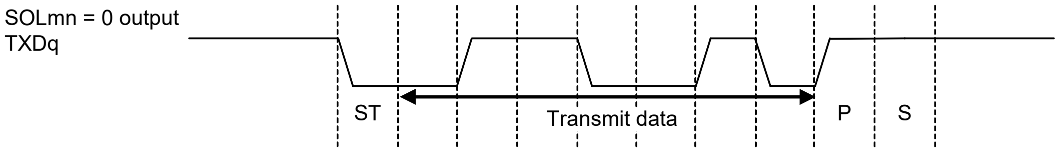
Caution Be sure to clear bits 15 to 3, and 1 of the SOL0 register to “0”.

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

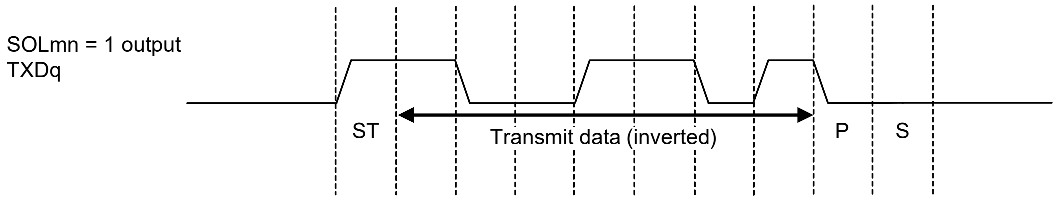
Figure 19 - 20 shows examples in which the level of transmit data is reversed during UART transmission.

Figure 19 - 20 Examples of Reverse Transmit Data

(a) Non-reverse Output (SOLmn = 0)



(b) Reverse Output (SOLmn = 1)



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

19.3.14 Serial standby control register m (SSCm)

The SSC0 register is used to control the startup of reception (the SNOOZE mode) while in the STOP mode when receiving CSI00 or UART0 serial data.
The SSCm register can be set by a 16-bit memory manipulation instruction.
The lower 8 bits of the SSCm register can be set with an 8-bit memory manipulation instruction by using SSCmL.
Reset signal generation clears the SSCm register to 0000H.

Caution The maximum transfer rate in the SNOOZE mode is as follows.
• When using CSI00: Up to 1 Mbps
• When using UART0: 4800 bps only

Figure 19 - 21 Format of Serial standby control register m (SSCm)

Address: F0138H (SSC0) After reset: 0000H R/W

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

SSECm	Selection of whether to enable or disable the generation of communication error interrupts in the SNOOZE mode
0	Enable the generation of error interrupts (INTSRE0).
1	Disable the generation of error interrupts (INTSRE0).

• The SSECm bit can be set to 1 or 0 only when both the SWCm and EOCmn bits are set to 1 during UART reception in the SNOOZE mode. In other cases, clear the SSECm bit to 0.
• Setting SSECm, SWCm = 1, 0 is prohibited.

SWCm	Setting of the SNOOZE mode
0	Do not use the SNOOZE mode function.
1	Use the SNOOZE mode function.

• When there is a hardware trigger signal in the STOP mode, the STOP mode is exited, and A/D conversion is performed without operating the CPU (the SNOOZE mode).
• The SNOOZE mode function can only be specified when the high-speed on-chip oscillator clock is selected for the CPU/peripheral hardware clock (f_{CLK}). If any other clock is selected, specifying this mode is prohibited.
• Even when using SNOOZE mode, be sure to set the SWCm bit to 0 in normal operation mode and change it to 1 just before shifting to STOP mode.
Also, be sure to change the SWCm bit to 0 after returning from STOP mode to normal operation mode.

Caution Setting SSECm, SWCm = 1, 0 is prohibited.

Table 19 - 2 Interrupt in UART Reception Operation in SNOOZE Mode

EOCmn Bit	SSECm Bit	Reception Ended Successfully	Reception Ended in an Error
0	0	INTSRx is generated.	INTSRx is generated.
0	1	INTSRx is generated.	INTSRx is generated.
1	0	INTSRx is generated.	INTSREx is generated.
1	1	INTSRx is generated.	No interrupt is generated.

19.3.15 Input switch control register (ISC)

The ISC1 and ISC0 bits of the ISC register are used to enable a LIN-bus communication operation by UART0 in coordination with an external interrupt and the timer array unit.
When bit 0 is set to 1, the input signal of the serial data input (RxD0) pin is selected as an external interrupt (INTP0) that can be used to detect a wakeup signal.
When bit 1 is set to 1, the input signal of the serial data input (RxD0) pin is selected as a timer input, so that wake up signal can be detected, the low width of the break field, and the pulse width of the sync field can be measured by the timer.
The SSIE0 bit controls the $\overline{\text{SSI00}}$ pin input of channel 0 during CSI00 communication and in slave mode.
While a high level is being input to the $\overline{\text{SSI00}}$ pin, no transmission/reception operation is performed even if a serial clock is input. While a low level is being input to the $\overline{\text{SSI00}}$ pin, a transmission/reception operation is performed according to each mode setting if a serial clock is input.
The ISC register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears the ISC register to 00H.

Figure 19 - 22 Format of Input switch control register (ISC)

Address: F0073H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	SSIE00	0	0	0	0	0	ISC1	ISC0

SSIE00	Setting of channel 0 $\overline{\text{SSI00}}$ input during Simplified SPI (CSI) communication in slave mode
0	Disables $\overline{\text{SSI00}}$ pin input.
1	Enables $\overline{\text{SSI00}}$ pin input.

ISC1	Switching channel 3 input of timer array unit 0
0	Uses the input signal of the TI03 pin as a timer input (normal operation).
1	Input signal of the RxD0 pin is used as timer input (detects the wakeup signal and measures the low width of the break field and the pulse width of the sync field).

ISC0	Switching external interrupt (INTP0) input
0	Uses the input signal of the INTP0 pin as an external interrupt (normal operation).
1	Uses the input signal of the RxD0 pin as an external interrupt (wakeup signal detection).

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

19.3.16 Noise filter enable register 0 (NFEN0)

The NFEN0 register is used to specify whether to use the noise filter 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 I²C 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, 2-clock match detection is performed after synchronization with the operation clock (f_{MCK}) of the channel. When the noise filter is disabled, only synchronization with the operation clock of channel (f_{MCK}) is performed.

The NFEN0 register can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears the NFEN0 register to 00H.

Figure 19 - 23 Format of Noise filter enable register 0 (NFEN0)

Address: F0070H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN0	0	0	0	0	0	SNFEN10	0	SNFEN00

SNFEN10	Use of noise filter of RxD1 pin
0	Noise filter OFF
1	Noise filter ON
Set the SNFEN10 bit to 1 to use the RxD1 pin. Clear the SNFEN10 bit to 0 to use the other than RxD1 pin.	

SNFEN00	Use of noise filter of RxD0 pin
0	Noise filter OFF
1	Noise filter ON
Set the SNFEN00 bit to 1 to use the RxD0 pin. Clear the SNFEN00 bit to 0 to use the other than RxD0 pin.	

Caution Be sure to clear bits 7 to 3 and 1 to “0”.

19.3.17 Registers controlling port functions of serial input/output pins

Using the serial array unit requires setting of the registers that control the port functions multiplexed on the target channel (port mode register (PMxx), port register (Pxx), port input mode register (PIMxx), port output mode register (POMxx), port mode control register (PMCxx)).

For details, see **4.3.1 Port mode registers (PMxx)**, **4.3.2 Port registers (Pxx)**, **4.3.4 Port input mode registers (PIMxx)**, and **4.3.5 Port output mode registers (POMxx)**.

Specifically, using a port pin with a multiplexed serial data or serial clock output function (e.g. P10/SO01/TxD1/TI01/TO01/INTP1/TRGIOA) for serial data or serial clock output, requires setting the corresponding bits in the port mode register (PMxx) to 0, and the corresponding bit in the port register (Pxx) to 1. When using the port pin in N-ch open-drain output (V_{DD} tolerance) mode, set the corresponding bit in the port output mode register (POMxx) to 1. When connecting an external device operating on a different potential (1.8 V, 2.5 V or 3 V), see **4.4.4 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers**.

Example When P10/SO01/TxD1/TI01/TO01/INTP1/TRGIOA is to be used for serial data output
Set the PM10 bit of port mode register 1 to 0.
Set the P10 bit of port register 1 to 1.

Specifically, using a port pin with a multiplexed serial data or serial clock input function (e.g. P11/SI01/RxD1/SDA01/TI03/TO03/INTP2/TRGCLKA/TRJIO0) for serial data or serial clock input, requires setting the corresponding bit in the port mode register (PMxx) to 1. In this case, the corresponding bit in the port register (Pxx) can be set to 0 or 1.

When the TTL input buffer is selected, set the corresponding bit in the port input mode register (PIMxx) to 1. When connecting an external device operating on a different potential (1.8 V, 2.5 V or 3 V), see **4.4.4 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers**.

Example When P11/SI01/RxD1/SDA01/TI03/TO03/INTP2/TRGCLKA/TRJIO0 is to be used for serial data input
Set the PM11 bit of port mode register 1 to 1.
Set the P11 bit of port register 1 to 0 or 1.

19.4 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.

19.4.1 Stopping the operation by units

The stopping of the operation by units is set by using peripheral enable register 0 (PER0).
The PER0 register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.
To stop the operation of serial array unit 0, set bit 2 (SAU0EN) to 0.

Figure 19 - 24 Peripheral Enable Register 0 (PER0) Setting When Stopping the Operation by Units

(a) Peripheral enable register 0 (PER0)... Set only the bit of SAUm to be stopped to 0.

	7	6	5	4	3	2	1	0
PER0	RTCEN ×	0	ADCEN ×	0	0	SAU0EN 0/1	TAU1EN ×	TAU0EN ×

Control of SAUm input clock (SAU0EN)
0: Stops supply of input clock
1: Supplies input clock

Caution 1. If SAU0EN = 0, writing to a control register of serial array unit 0 is ignored and, even if the register is read, only the default value is read
Note that this does not apply to the following registers.
- **Input switch control register (ISC)**
- **Noise filter enable register 0 (NFEN0)**
- **Port input mode register 1 (PIM1)**
- **Port output mode register 1 (POM1)**
- **Port mode register 1 (PM1)**
- **Port register 1 (P1)**

Caution 2. Be sure to clear bits 3, 4, and 6 to “0”.

Remark ×: Bits not used with serial array units (depending on the settings of other peripheral functions)
0/1: Set to 0 or 1 depending on the usage of the user

19.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

Figure 19 - 25 Each Register Setting When Stopping the Operation by Channels

(a) Serial channel stop register m (STm)... This is a trigger register that is used to enable stopping communication/count by each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STm	0	0	0	0	0	0	0	0	0	0	0	0	STm3 0/1	STm2 0/1	STm1 0/1	STm0 0/1

1: Clears the SEmn bit to 0 and stops the communication operation

* Because the STmn bit is a trigger bit, it is cleared immediately when SEmn is cleared to 0.

(b) Serial Channel Enable Status Register m (SEm)... This register indicates whether data transmission/ reception operation of each channel is enabled or stopped.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEm	0	0	0	0	0	0	0	0	0	0	0	0	SEm3 0/1	SEm2 0/1	SEm1 0/1	SEm0 0/1

0: Operation stops

* The SEm register is a read-only status register, whose operation is stopped by using the STm register.
With a channel whose operation is stopped, the value of the CKOmn bit of the SOm register can be set by software.

(c) Serial output enable register m (SOEm)... This is a register that is used to enable or stop output of the serial communication operation of each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 0/1	SOEm1 0/1	SOEm0 0/1

0: Stops output by serial communication operation

* For channel n, whose serial output is stopped, the SOmn bit value of the SOm register can be set by software.

(d) Serial output register m (SOm)... This is a buffer register for serial output of each channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	CKOm2 0/1	CKOm1 0/1	CKOm0 0/1	0	0	0	0	0	SOm2 0/1	SOm1 0/1	SOm0 0/1

1: Serial clock output value is “1”

1: Serial data output value is “1”

* When using pins corresponding to each channel as port function pins, set the corresponding CKOmn, SOmn bits to “1”.

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

Remark 2. ■: Setting disabled (set to the initial value)
0/1: Set to 0 or 1 depending on the usage of the user

19.5 Operation of Simplified SPI (CSI00, CSI01) Communication

This is a clocked communication function that uses three lines: serial clock (SCK) and serial data (SI and SO) lines.

[Data transmission/reception]
• Data length of 7 or 8 bits
• Phase control of transmit/receive data
• MSB/LSB first selectable
[Clock control]
• Master/slave selection
• Phase control of I/O clock
• Setting of transfer period by prescaler and internal counter of each channel
• Maximum transfer rate Note
  During master communication: Max. $f_{CLK}/4$
  During slave communication: Max. $f_{MCK}/12$
[Interrupt function]
• Transfer end interrupt/buffer empty interrupt
[Error detection flag]
• Overrun error

CSI00 supports the SNOOZE mode. When SCK00 input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible.

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics. For details, see **CHAPTER 33** or **CHAPTER 34 ELECTRICAL SPECIFICATIONS**.

The channels supporting Simplified SPI (CSI00, CSI01) are channels 0 and 1 of SAU0.

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I²C
0	0	CSI00 (supporting slave select input function)	UART0 (supporting LIN-bus)	IIC00
	1	CSI01		IIC01
	2	—	UART1	—
	3	—		—

Simplified SPI (CSI00, CSI01) performs the following seven types of communication operations.
• Master transmission (See **19.5.1**.)
• Master reception (See **19.5.2**.)
• Master transmission/reception (See **19.5.3**.)
• Slave transmission (See **19.5.4**.)
• Slave reception (See **19.5.5**.)
• Slave transmission/reception (See **19.5.6**.)
• SNOOZE mode function (See **19.5.7**.)

19.5.1 Master transmission

Master transmission is that the RL78 microcontroller outputs a transfer clock and transmits data to another device.

Simplified SPI	CSI00	CSI01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCK00, SO00	SCK01, SO01
Interrupt	INTCSI00	INTCSI01
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	None	
Transfer data length	7 or 8 bits	
Transfer rate Note	Max. $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK}: System clock frequency	
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data output starts from the start of the operation of the serial clock. • DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse	
Data direction	MSB or LSB first	

Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 33** or **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

(1) Register setting

Figure 19 - 26 Example of Contents of Registers for Master Transmission of Simplified SPI (CSI00, CSI01)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **19.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15–9	8	7–0
SDRmn	Baud rate setting (Operation clock (f_{MCK}) division setting)	0	Transmit data (Transmit data setting)

Bits 7–0: SIOp

(d) Serial output register m (SOm)... Set only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	CKOm2 ×	CKOm1 0/1	CKOm0 0/1	0	0	0	0	0	SOm2 ×	SOm1 0/1	SOm0 0/1

Communication starts when these bits are 1 if the clock phase is non-reversed (the CKPmn bit of the SCRmn = 0). If the clock phase is reversed (CKPmn = 1), communication starts when these bits are 0.

(e) Serial output enable register m (SOEm)... Set only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOm2 ×	SOEm1 0/1	SOEm0 0/1

(f) Serial channel start register m (SSm)... Set only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 0/1	SSm0 0/1

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Remark 2. (bold box): Setting is fixed in the Simplified SPI (CSI) master transmission mode,
(grey box): 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 19 - 27 Initial Setting Procedure for Master Transmission

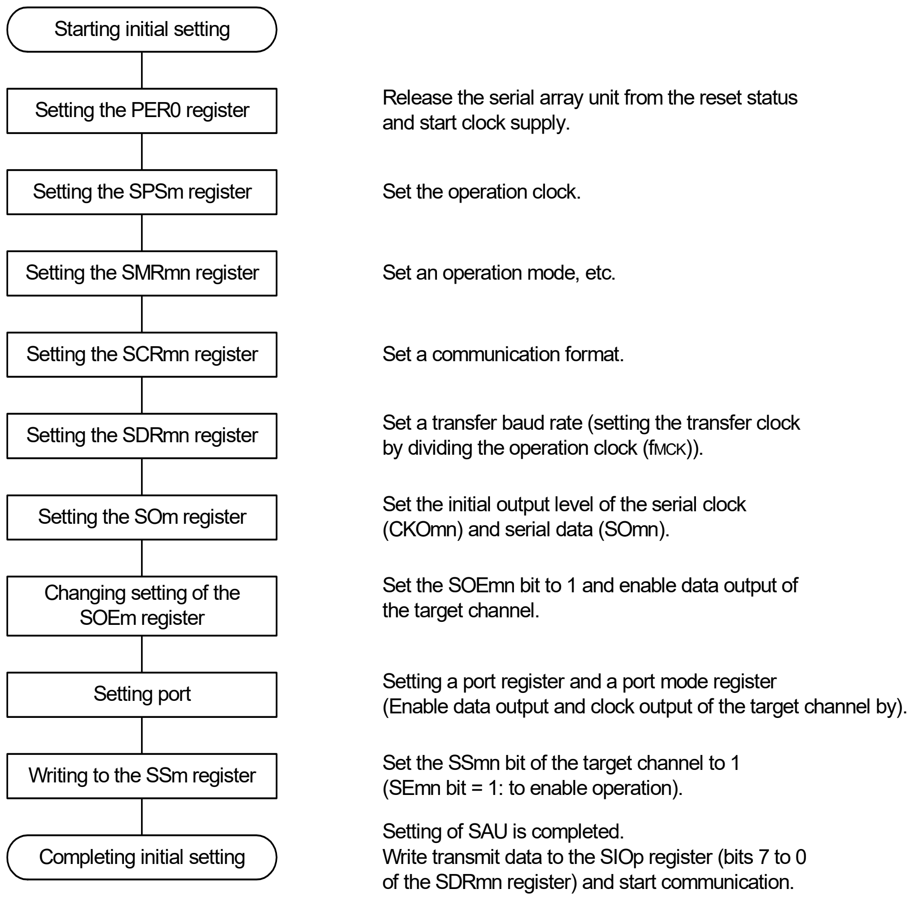


Figure 19 - 28 Procedure for Stopping Master Transmission

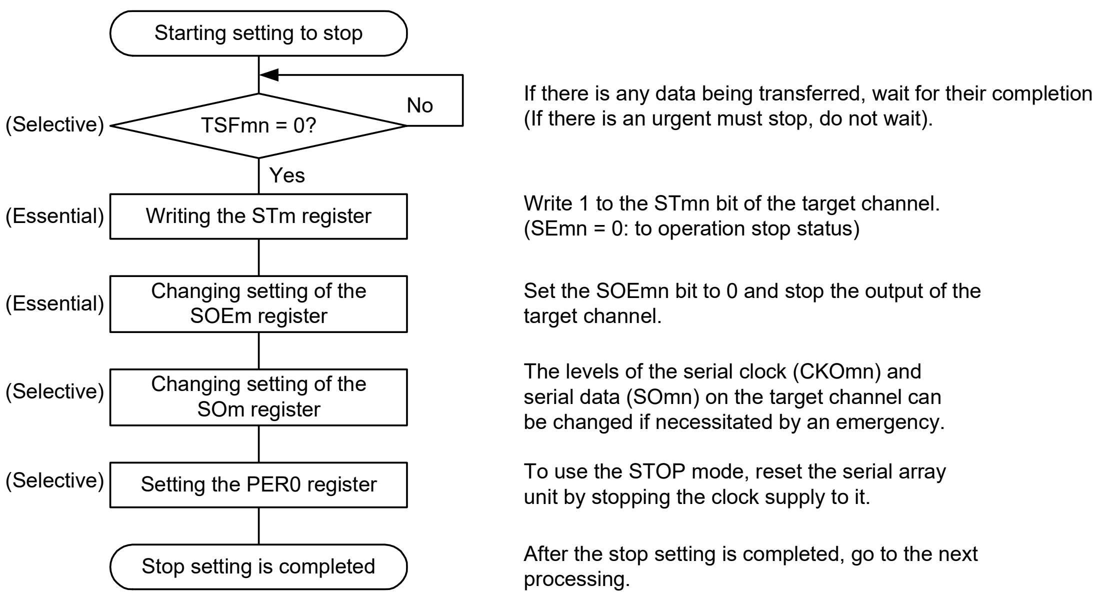
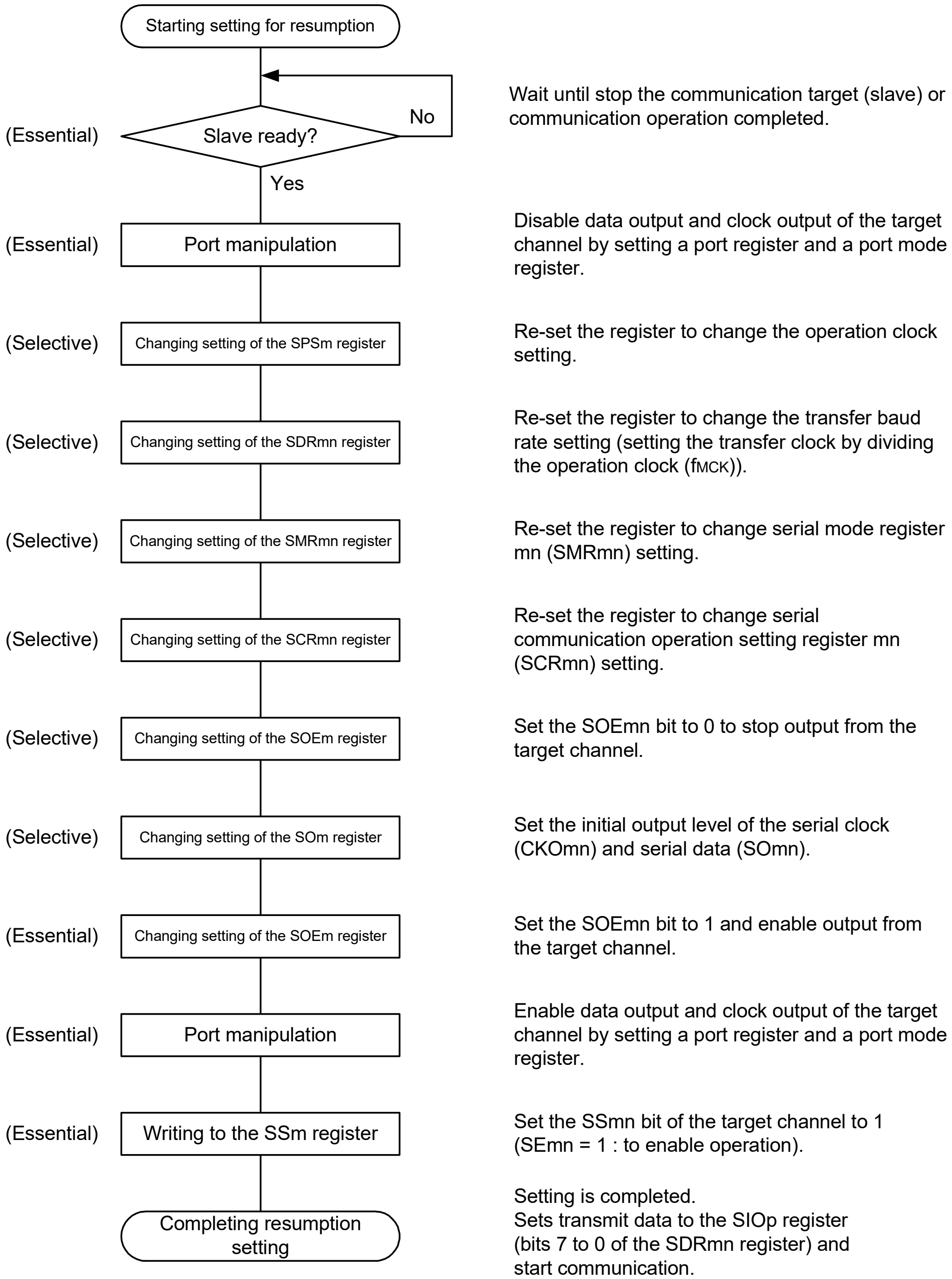


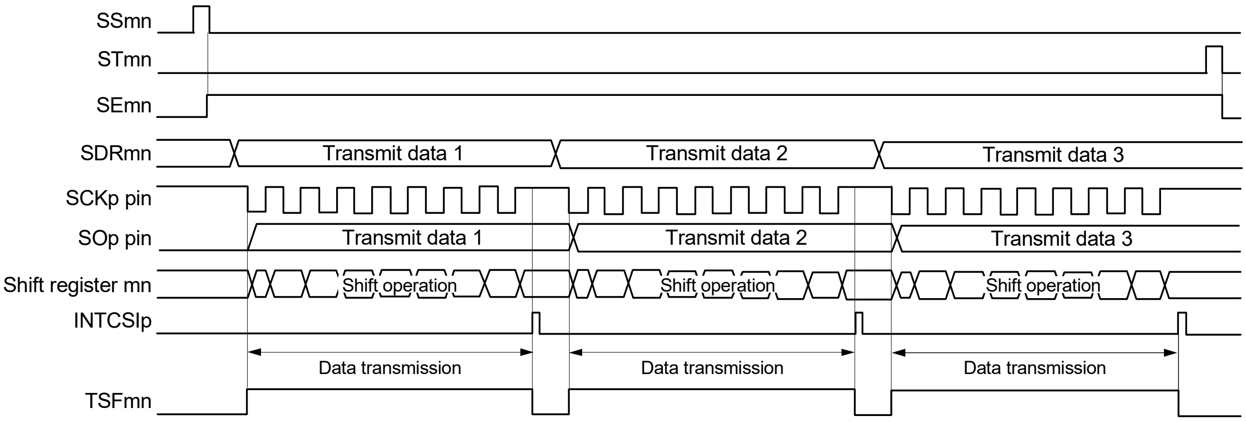
Figure 19 - 29 Procedure for Resuming Master Transmission



Remark If PER0 is rewritten to stop clock supply while the master transmission is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

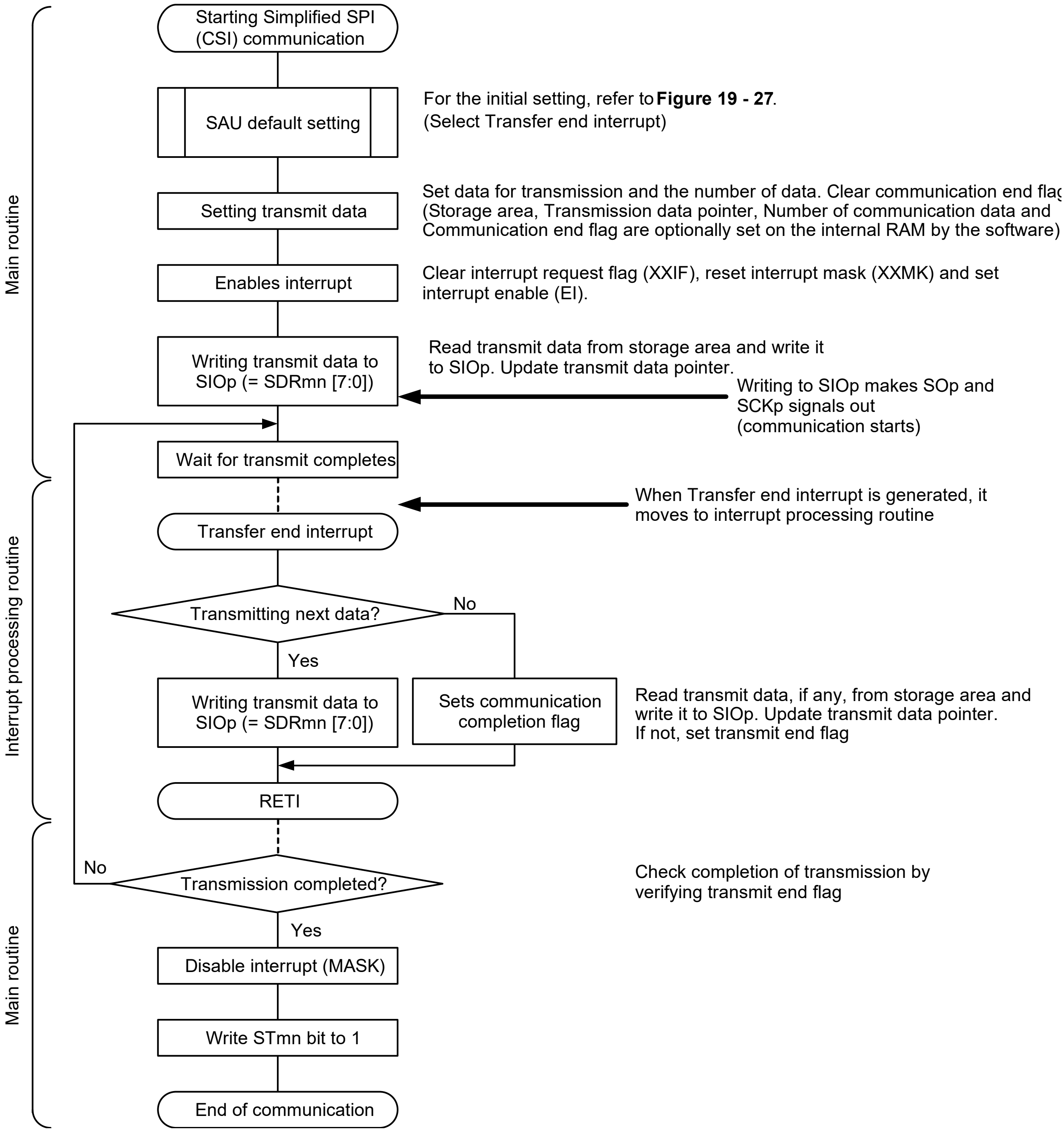
(3) Processing details (in single-transmission mode)

Figure 19 - 30 Timing Chart of Master Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



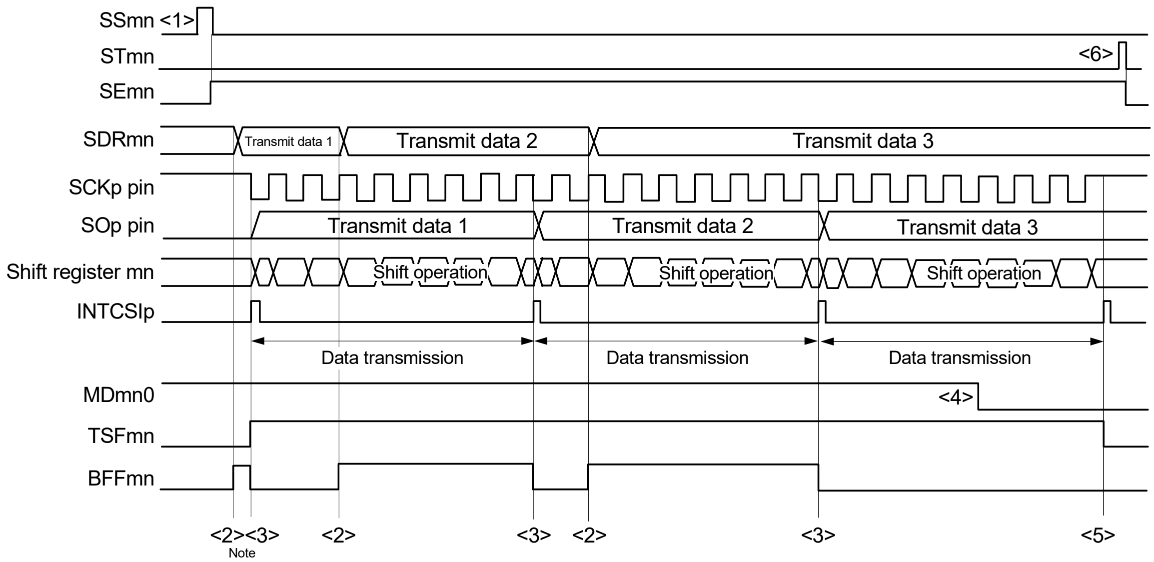
Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 19 - 31 Flowchart of Master Transmission (in Single-Transmission Mode)



(4) Processing details (in continuous transmission mode)

Figure 19 - 32 Timing Chart of Master Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)

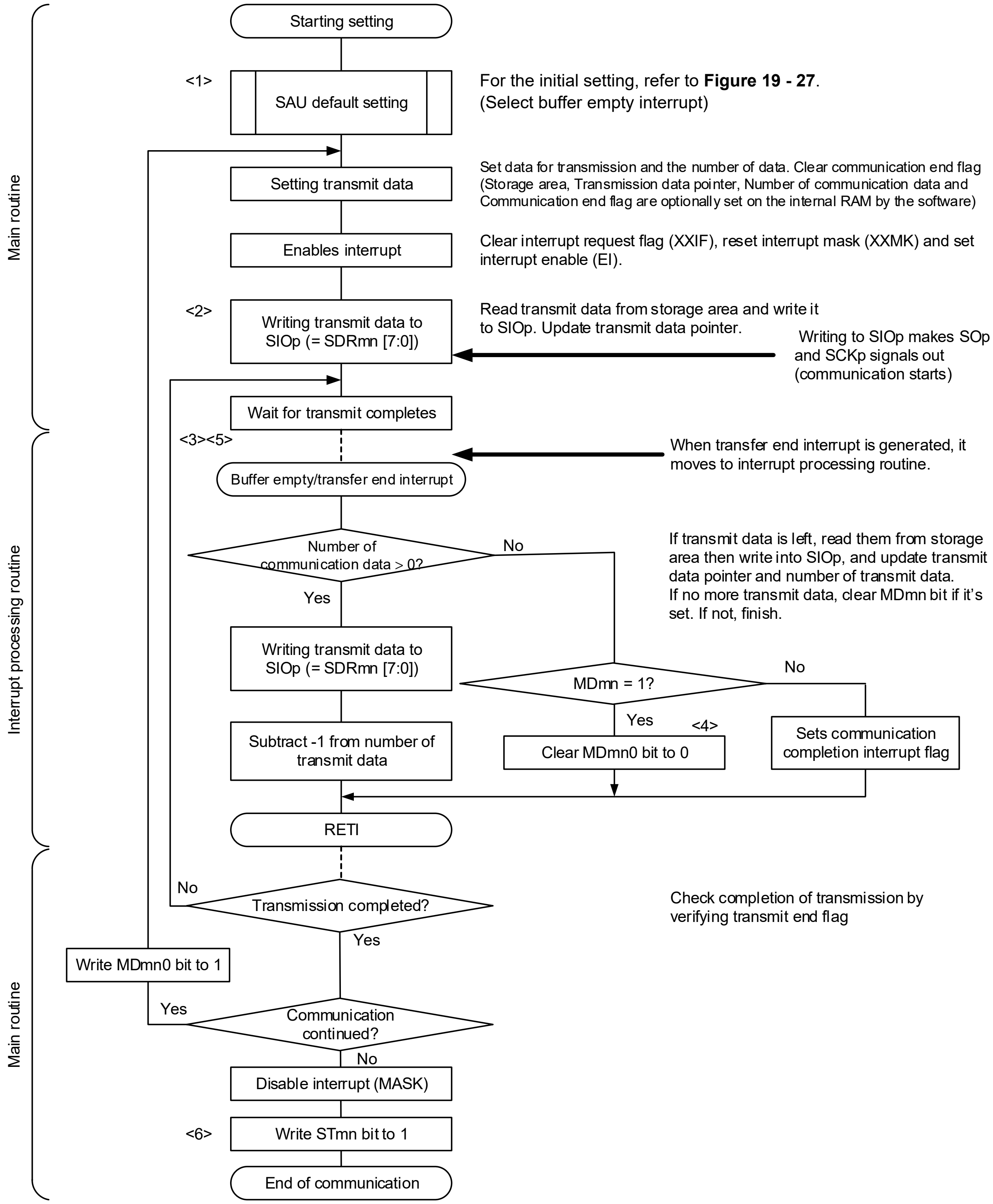


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution **The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit starts, so that it will be rewritten before the transfer end interrupt of the last transmit data.**

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 19 - 33 Flowchart of Master Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in Figure 19 - 32 Timing Chart of Master Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0).

19.5.2 Master reception

Master reception is that the RL78 microcontroller outputs a transfer clock and receives data from other device.

Simplified SPI	CSI00	CSI01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCK00, SI00	SCK01, SI01
Interrupt	INTCSI00	INTCSI01
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	Overrun error detection flag (OVFmn) only	
Transfer data length	7 or 8 bits	
Transfer rate Note	Max. $f_{CLK}/4$ [Hz] Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [Hz] f_{CLK}: System clock frequency	
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data input starts from the start of the operation of the serial clock. • DAPmn = 1: Data input starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse	
Data direction	MSB or LSB first	

Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 33** or **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

(1) Register setting

Figure 19 - 34 Example of Contents of Registers for Master Reception of Simplified SPI (CSI00, CSI01)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (fMCK) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **19.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting (Operation clock (fMCK) division setting)							0	Receive data (Write FFH as dummy data.)							

Bits 7 to 0: SIOp

(d) Serial output register m (SOm)... Set only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	CKOm2 ×	CKOm1 0/1	CKOm0 0/1	0	0	0	0	0	SOm2 ×	SOm1 ×	SOm0 ×

Communication starts when these bits are 1 if the clock phase is non-reversed (the CKPmn bit of the SCRmn = 0). If the clock phase is reversed (CKPmn = 1), communication starts when these bits are 0.

(e) Serial output enable register m (SOEm)... 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	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	SOEm1 ×	SOEm0 ×

(f) Serial channel start register m (SSm)... Set only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 0/1	SSm0 0/1

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Remark 2. ■ (bold-outlined box): Setting is fixed in the Simplified SPI (CSI) master reception mode,
(gray box): 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 19 - 35 Initial Setting Procedure for Master Reception

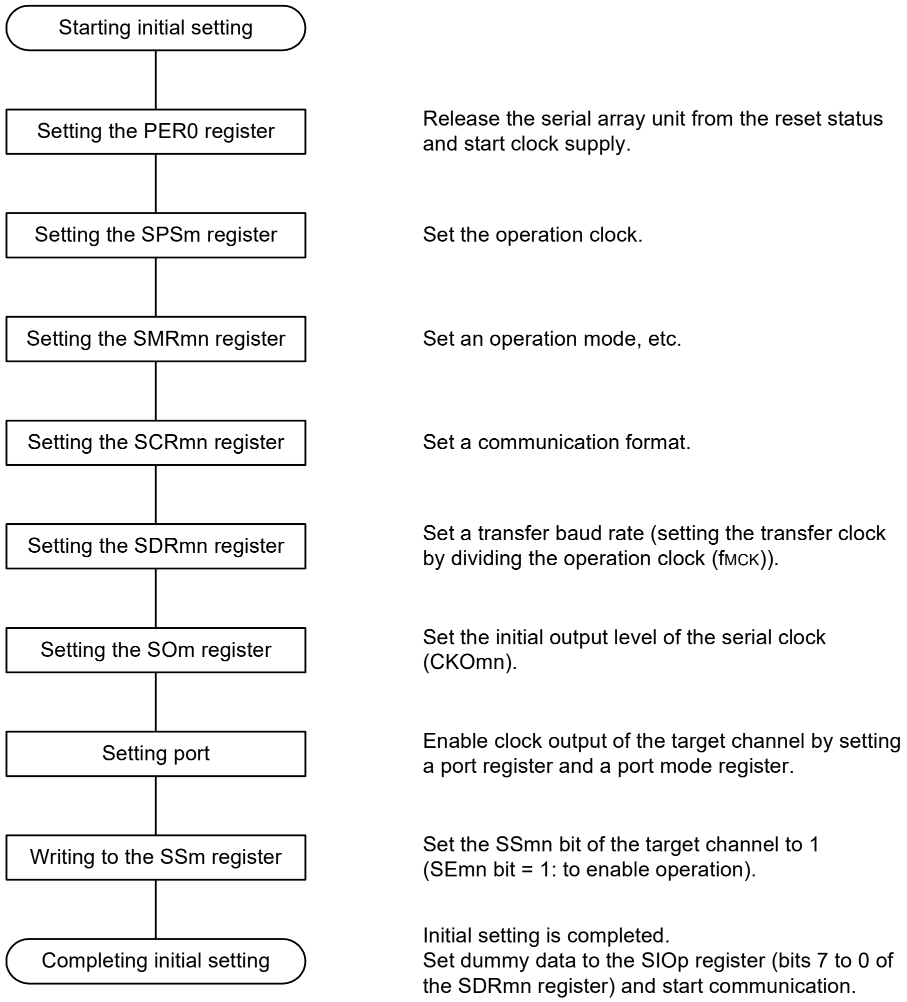


Figure 19 - 36 Procedure for Stopping Master Reception

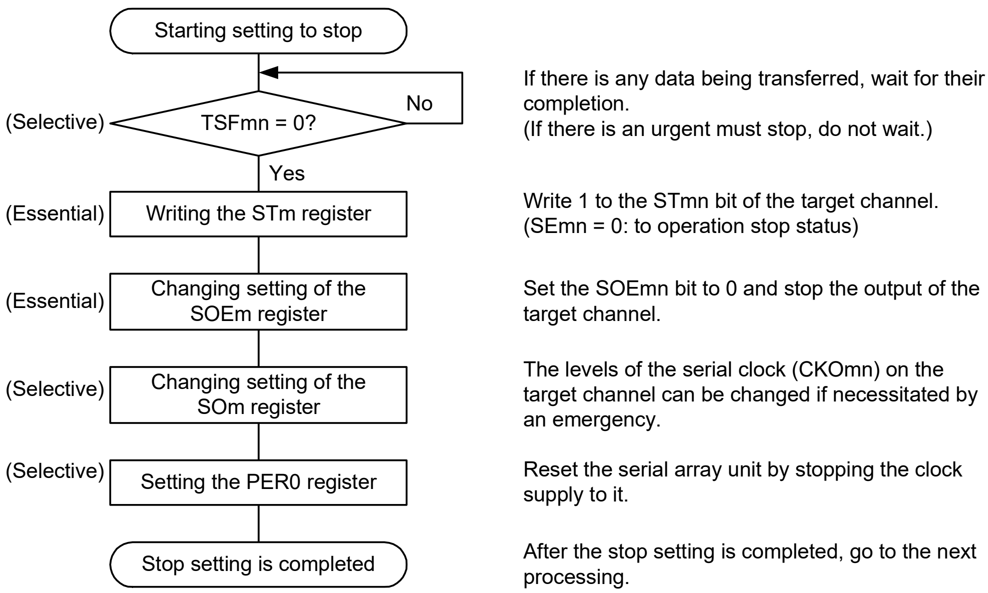
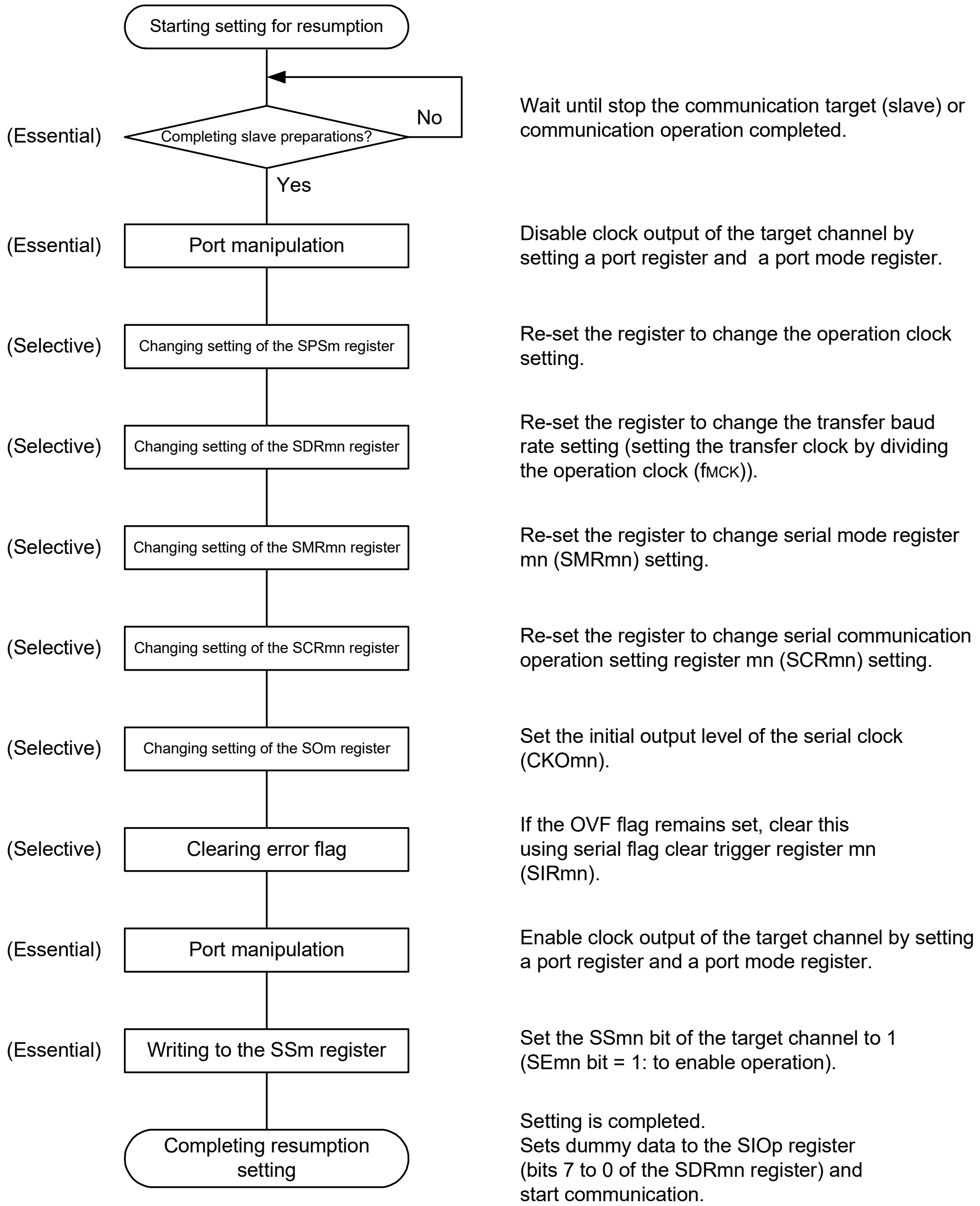


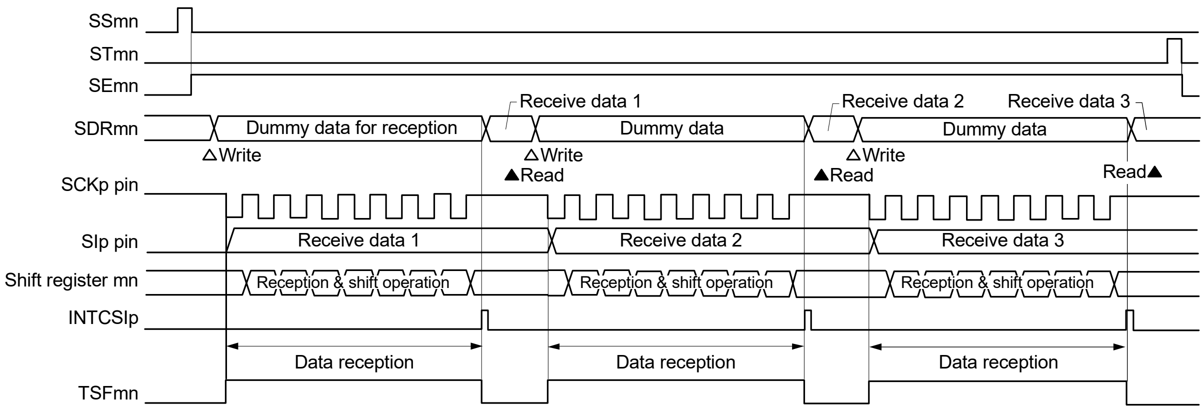
Figure 19 - 37 Procedure for Resuming Master Reception



Remark If PER0 is rewritten to stop clock supply while the master transmission is stopped, wait until the transmission target (slave) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

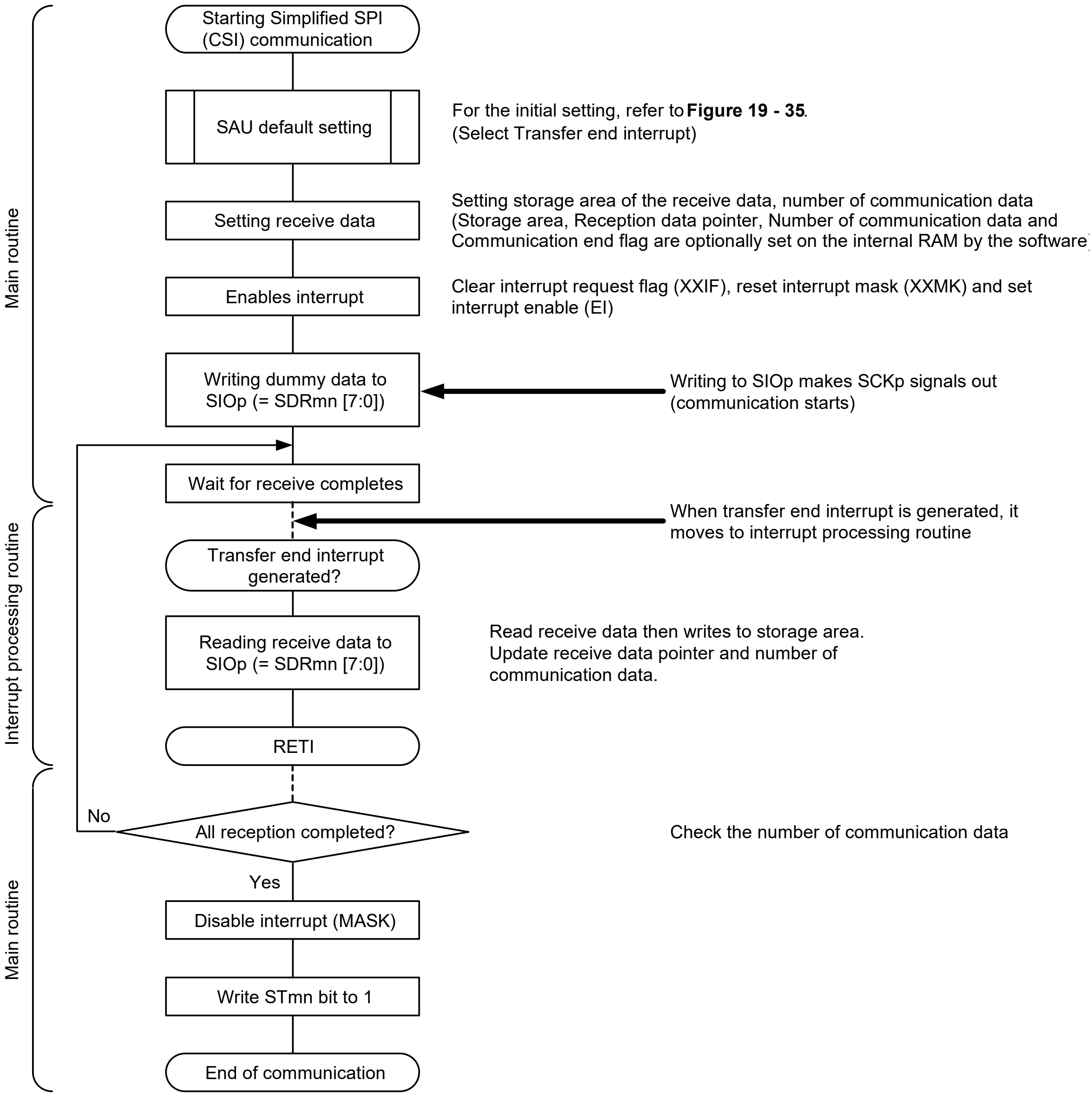
(3) Processing details (in single-reception mode)

Figure 19 - 38 Timing Chart of Master Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



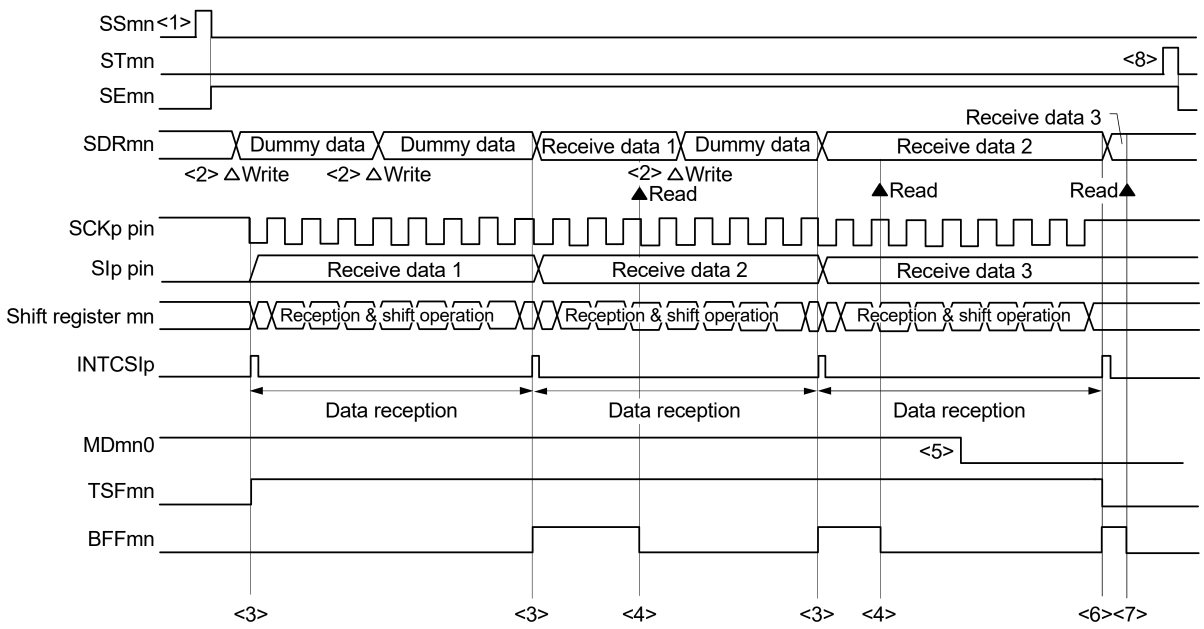
Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 19 - 39 Flowchart of Master Reception (in Single-Reception Mode)



(4) Processing details (in continuous reception mode)

Figure 19 - 40 Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)

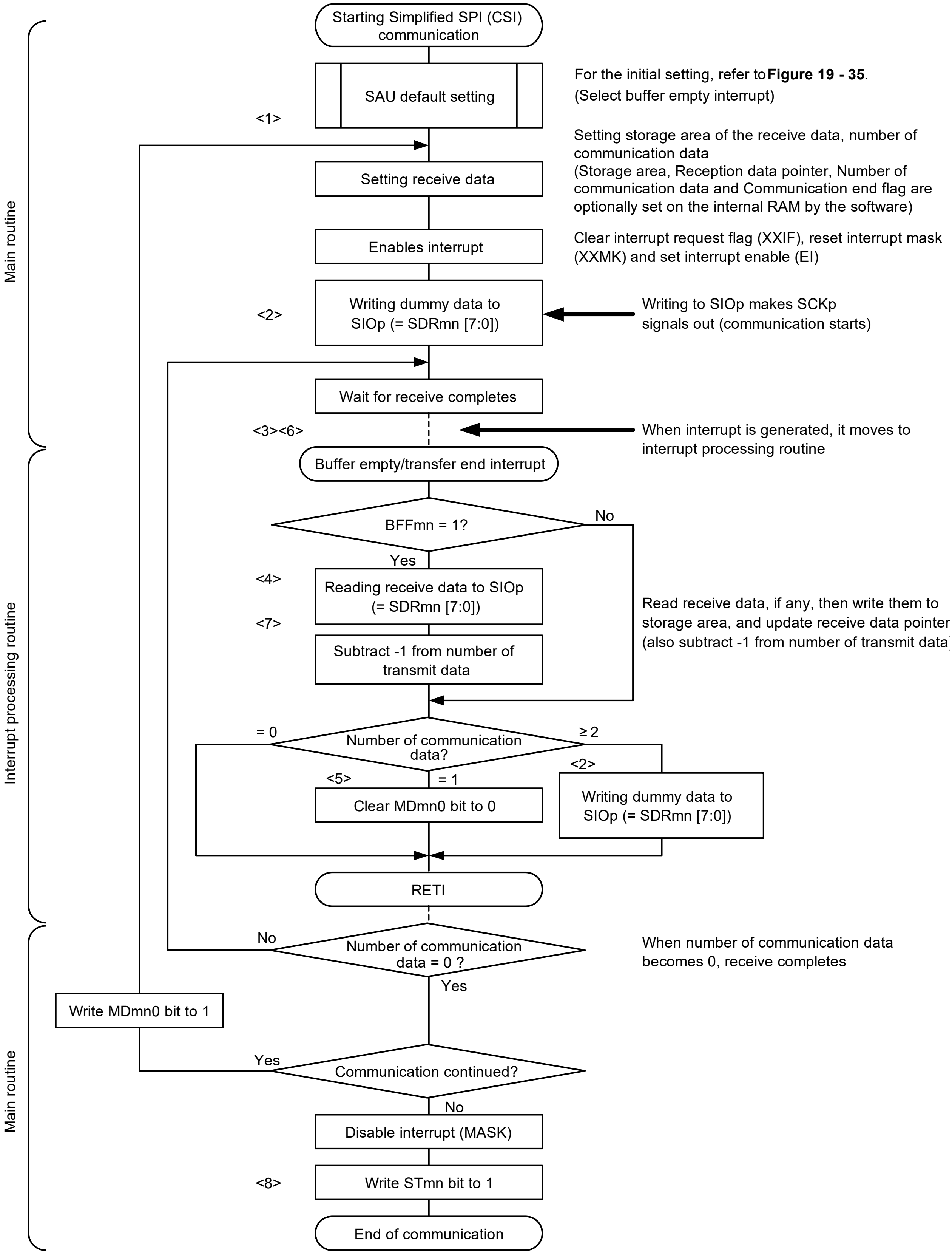


Caution **The MDmn0 bit can be rewritten even during operation.**
However, rewrite it before reception of the last bit starts, so that it has been rewritten before the transfer end interrupt of the last receive data.

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in Figure 19 - 41 Flowchart of Master Reception (in Continuous Reception Mode).

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 19 - 41 Flowchart of Master Reception (in Continuous Reception Mode)



Remark <1> to <8> in the figure correspond to <1> to <8> in Figure 19 - 40 Timing Chart of Master Reception (in Continuous Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

19.5.3 Master transmission/reception

Master transmission/reception is that the RL78 microcontroller outputs a transfer clock and transmits/receives data to/from other device.

Simplified SPI	CSI00	CSI01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01
Interrupt	INTCSI00	INTCSI01
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	Overrun error detection flag (OVFmn) only	
Transfer data length	7 or 8 bits	
Transfer rate Note	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 before the start of the serial clock operation.	
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse	
Data direction	MSB or LSB first	

Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 33** or **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

(1) Register setting

Figure 19 - 42 Example of Contents of Registers for Master Transmission/Reception of Simplified SPI (CSI00, CSI01)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (fMCK) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **19.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting (Operation clock (fMCK) division setting)							0	Transmit data setting/receive data register (SIOp)							

(d) Serial output register m (SOm)... Set only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	CKOm2 ×	CKOm1 0/1	CKOm0 0/1	0	0	0	0	0	SOm2 ×	SOm1 0/1	SOm0 0/1

Communication starts when these bits are 1 if the clock phase is non-reversed (the CKPmn bit of the SCRmn = 0). If the clock phase is reversed (CKPmn = 1), communication starts when these bits are 0.

(e) Serial output enable register m (SOEm)... Set only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	SOEm1 0/1	SOEm0 0/1

(f) Serial channel start register m (SSm)... Set only the bits of the target channel to 1.

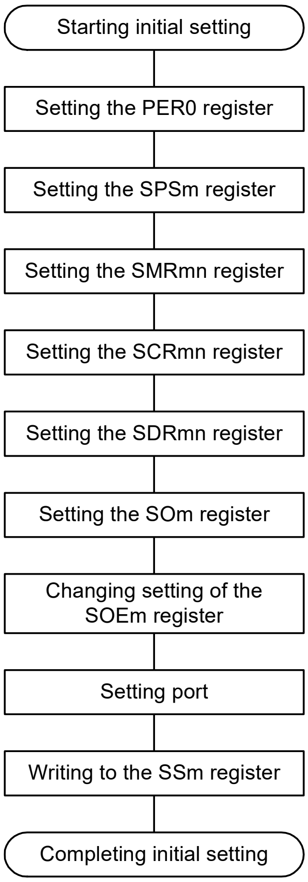
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 0/1	SSm0 0/1

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Remark 2. ■: Setting is fixed in the Simplified SPI (CSI) master transmission/reception mode,
(shaded): 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 19 - 43 Initial Setting Procedure for Master Transmission/Reception



Release the serial array unit from the reset status and start clock supply.

Set the operation clock.

Set an operation mode, etc.

Set a communication format.

Set a transfer baud rate (setting the transfer clock by dividing the operation clock (f_{MCK})).

Set the initial output level of the serial clock (CKOmn) and serial data (SOmn).

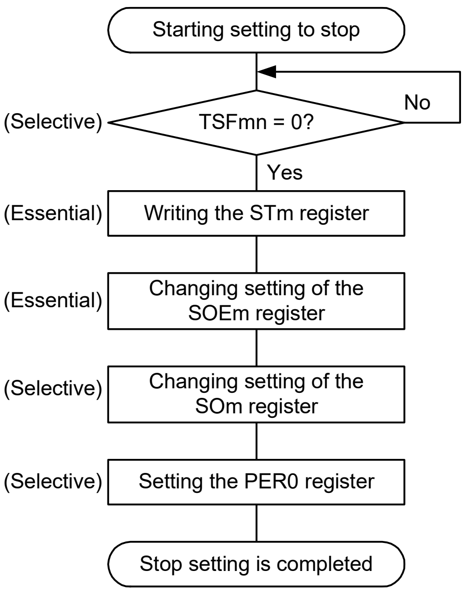
Set the SOEmn bit to 1 and enable data output of the target channel.

Enable data output and clock output of the target channel by setting a port register and a port mode register.

Set the SSmn bit of the target channel to 1 (SEmn bit = 1: to enable operation).

Initial setting is completed.
Set transmit data to the SIOp register (bits 7 to 0 of the SDRmn register) and start.

Figure 19 - 44 Procedure for Stopping Master Transmission/Reception



If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait.)

Write 1 to the STmn bit of the target channel.
(SEmn = 0: to operation stop status)

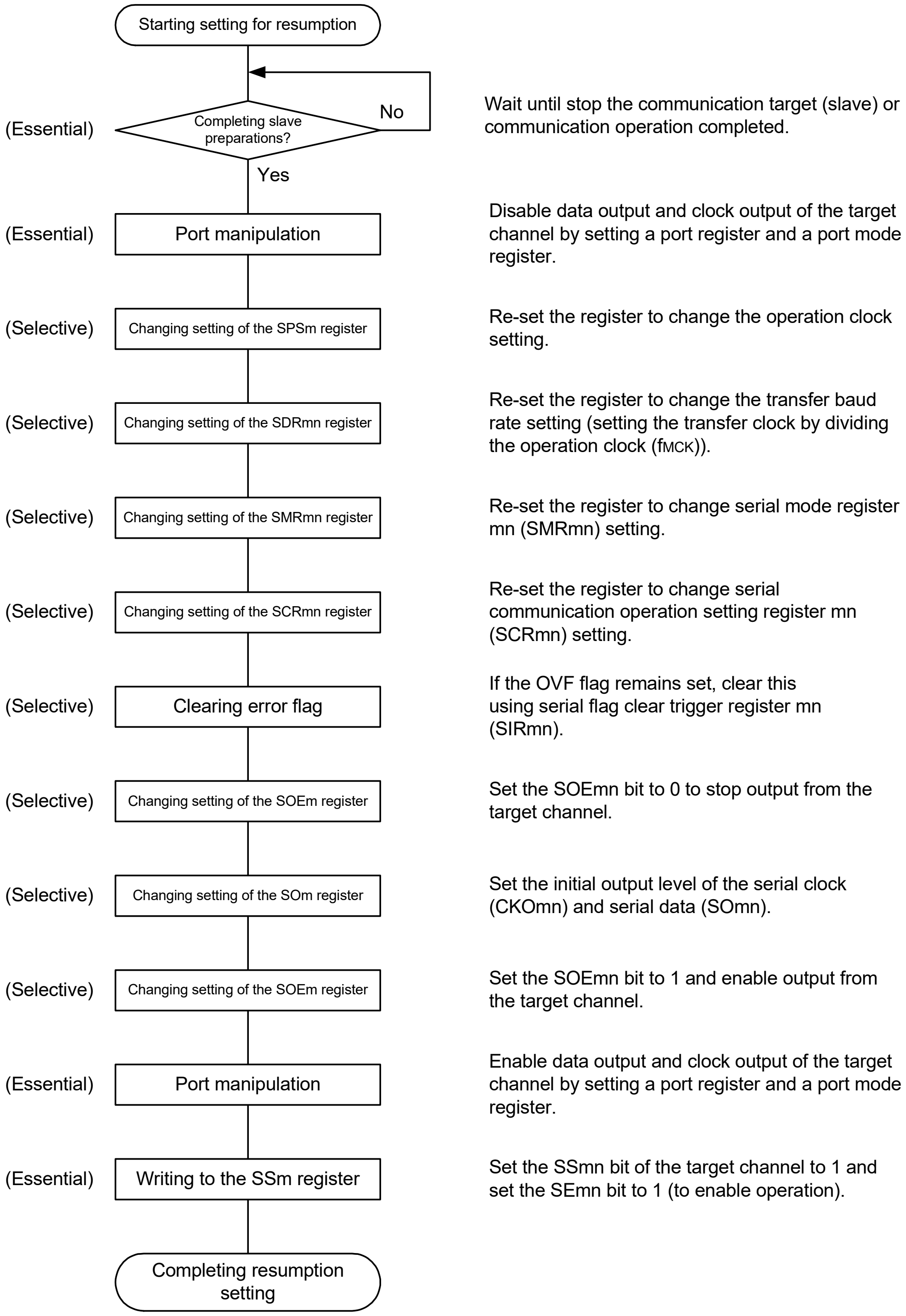
Set the SOEmn bit to 0 and stop the output of the target channel.

The levels of the serial clock (CKOmn) and serial data (SOmn) on the target channel can be changed if necessitated by an emergency.

Reset the serial array unit by stopping the clock supply to it.

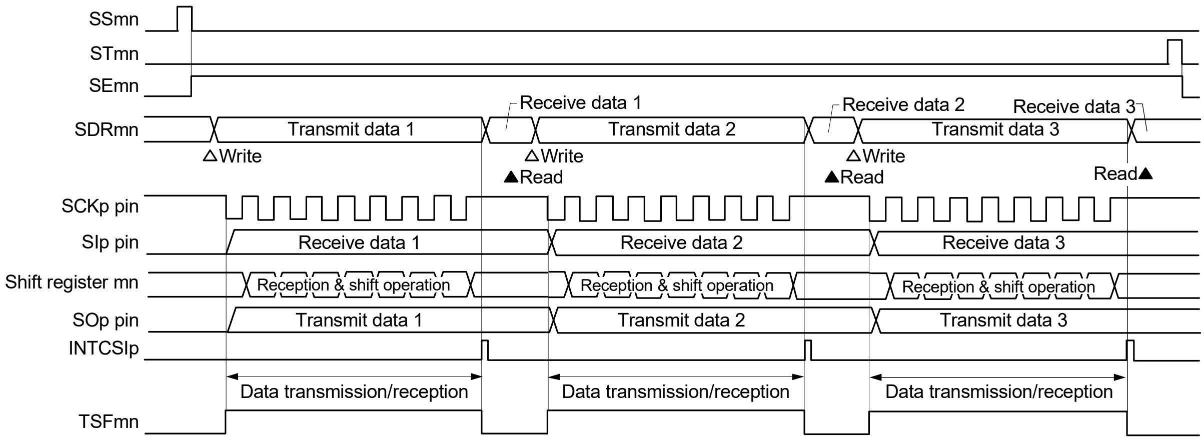
After the stop setting is completed, go to the next processing.

Figure 19 - 45 Procedure for Resuming Master Transmission/Reception



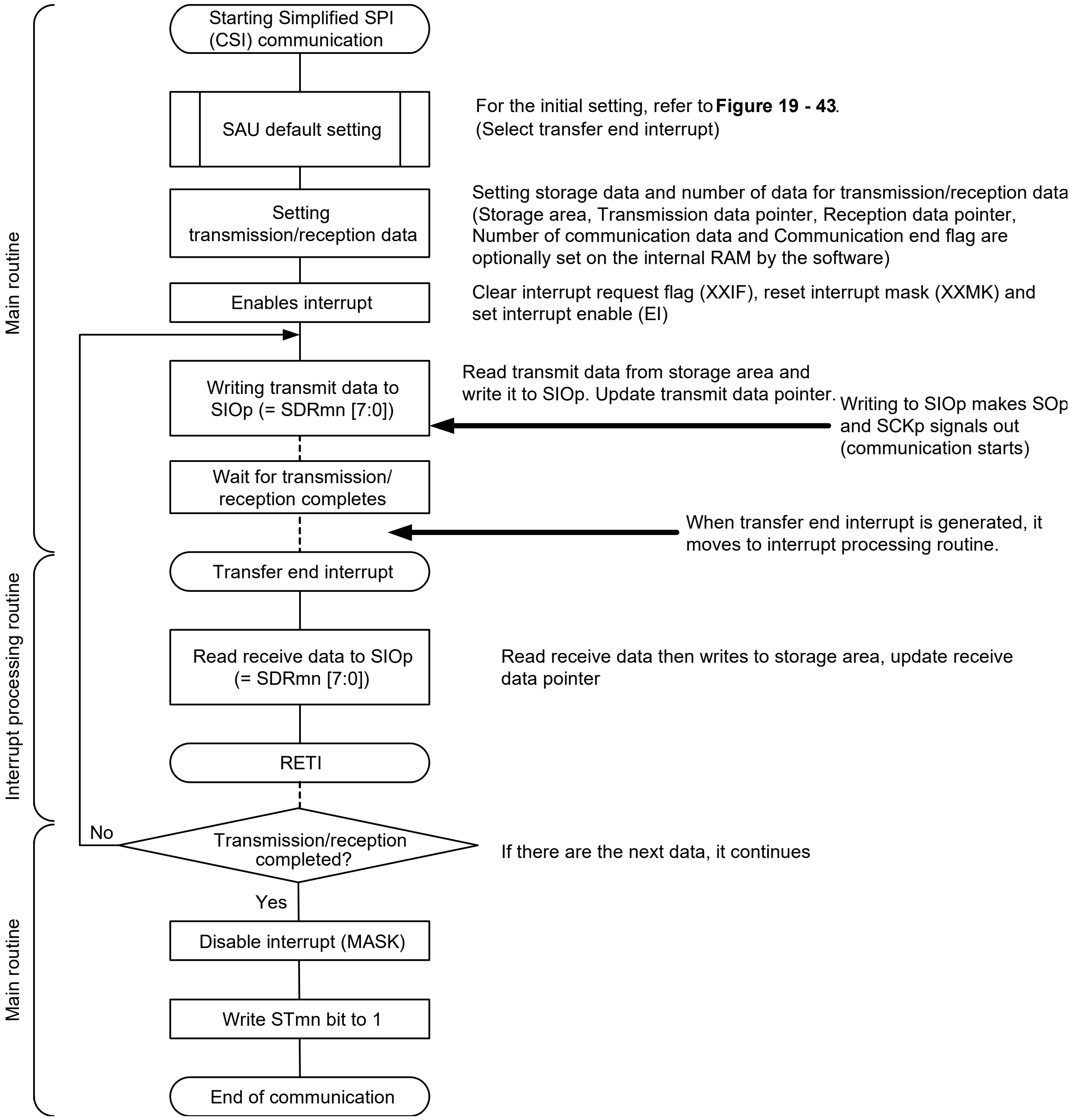
(3) Processing details (in single-transmission/reception mode)

Figure 19 - 46 Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



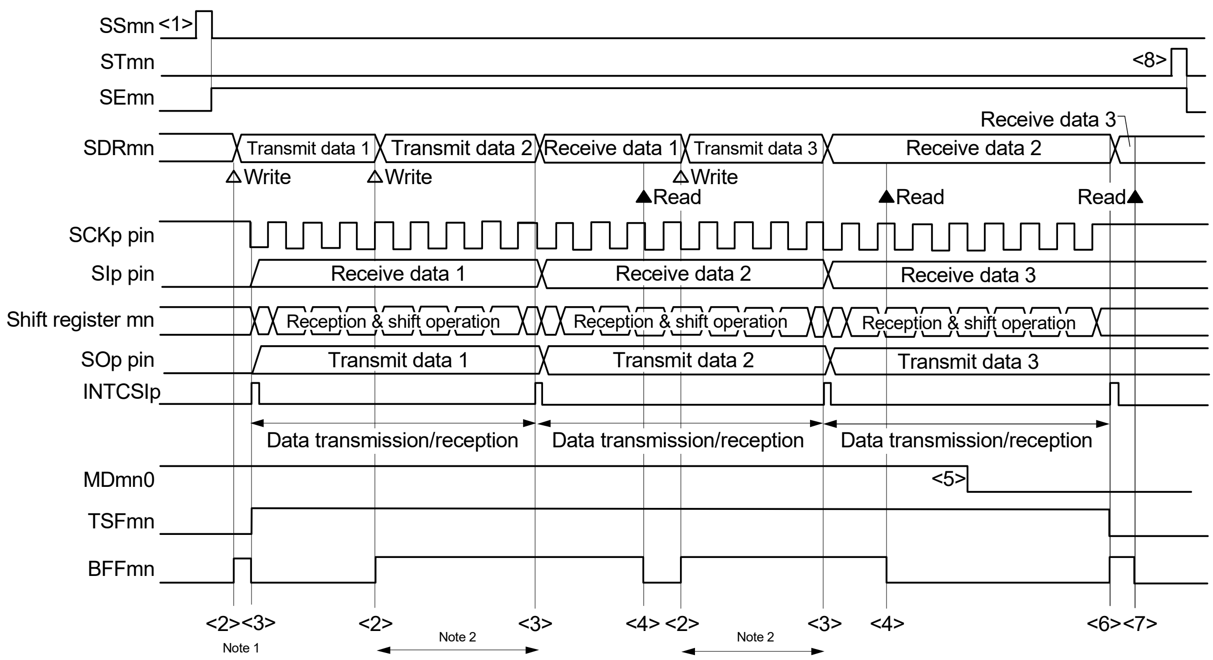
Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 19 - 47 Flowchart of Master Transmission/Reception (in Single- Transmission/Reception Mode)



(4) Processing details (in continuous transmission/reception mode)

Figure 19 - 48 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Note 1. If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

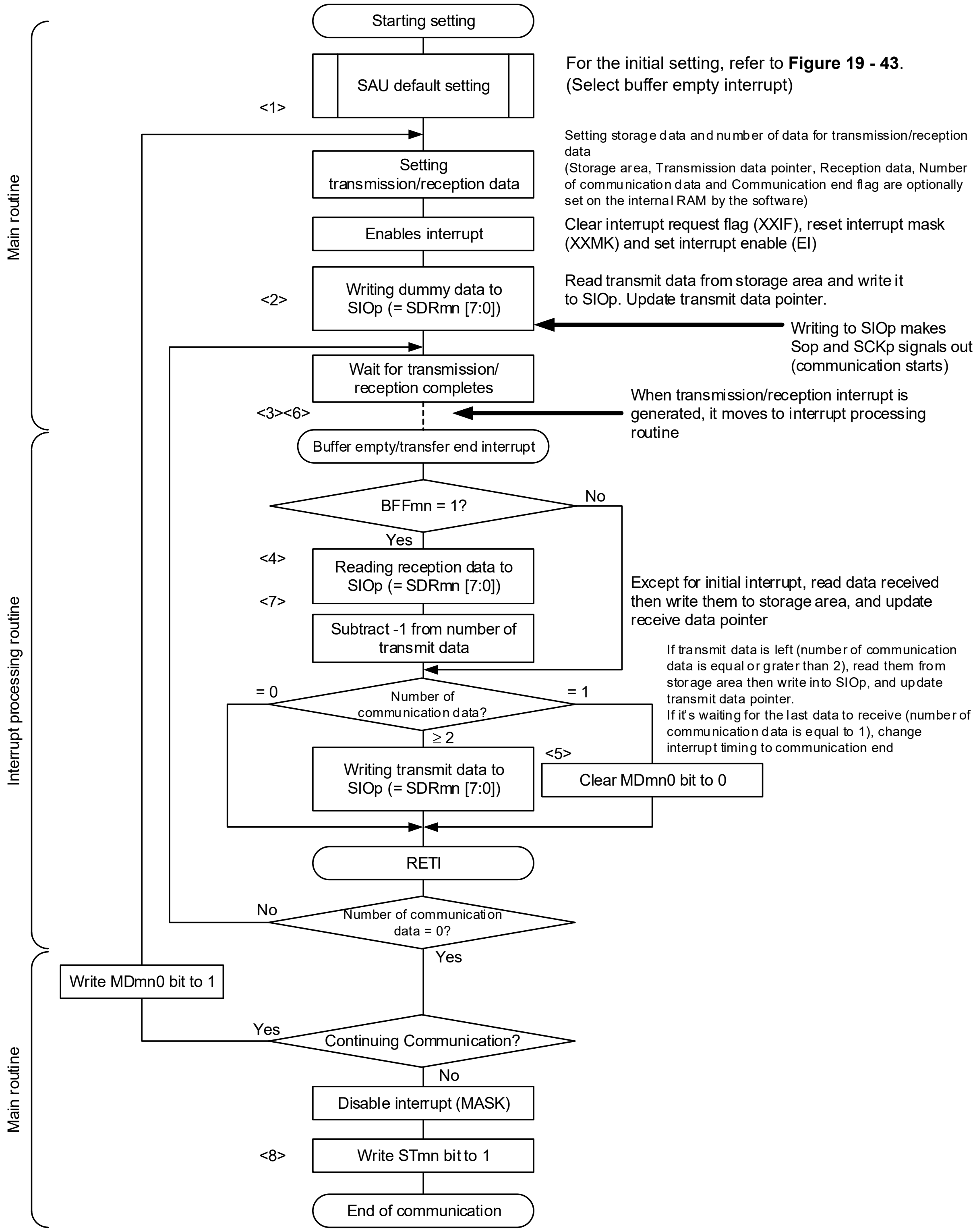
Note 2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

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 starts, 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 19 - 49 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode).

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 19 - 49 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



Remark <1> to <8> in the figure correspond to <1> to <8> in Figure 19 - 48 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

19.5.4 Slave transmission

Slave transmission is that the RL78 microcontroller transmits data to another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00	CSI01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCK00, SO00	SCK01, SO01
Interrupt	INTCSI00	INTCSI01
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	Overrun error detection flag (OVFmn) only	
Transfer data length	7 or 8 bits	
Transfer rate	Max. $f_{MCK}/12$ [Hz] Notes 1, 2.	
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data output starts from the start of the operation of the serial clock. • DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse	
Data direction	MSB or LSB first	

Note 1. Because the external serial clock input to the SCK00 and SCK01 pins is sampled internally and used, the fastest transfer rate is $f_{MCK}/12$ [Hz].

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 33** or **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark 1. f_{MCK}: Operation clock frequency of target channel

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

(1) Register setting

Figure 19 - 50 Example of Contents of Registers for Slave Transmission of Simplified SPI (CSI00, CSI01)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **19.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15–9	8	7–0
SDRmn	0000000 Baud rate setting	0	Transmit data setting (SIOp)

(d) Serial output register m (SOm)... Set only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	CKOm2 ×	CKOm1 ×	CKOm0 ×	0	0	0	0	0	SOm2 ×	SOm1 0/1	SOm0 0/1

(e) Serial output enable register m (SOEm)... Set only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	SOEm1 0/1	SOEm0 0/1

(f) Serial channel start register m (SSm)... Set only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 0/1	SSm0 0/1

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Remark 2. ■: Setting is fixed in the Simplified SPI (CSI) slave transmission mode,
■ (gray): 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 19 - 51 Initial Setting Procedure for Slave Transmission

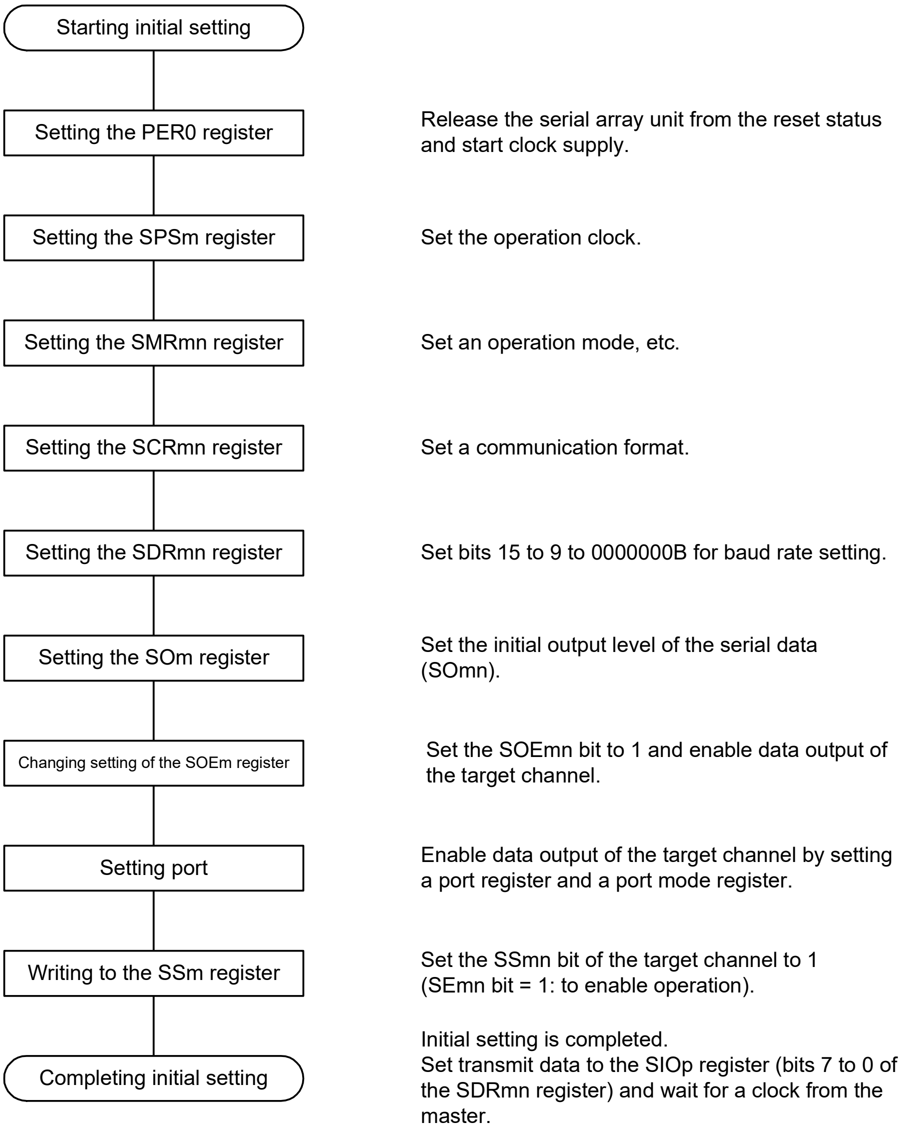


Figure 19 - 52 Procedure for Stopping Slave Transmission

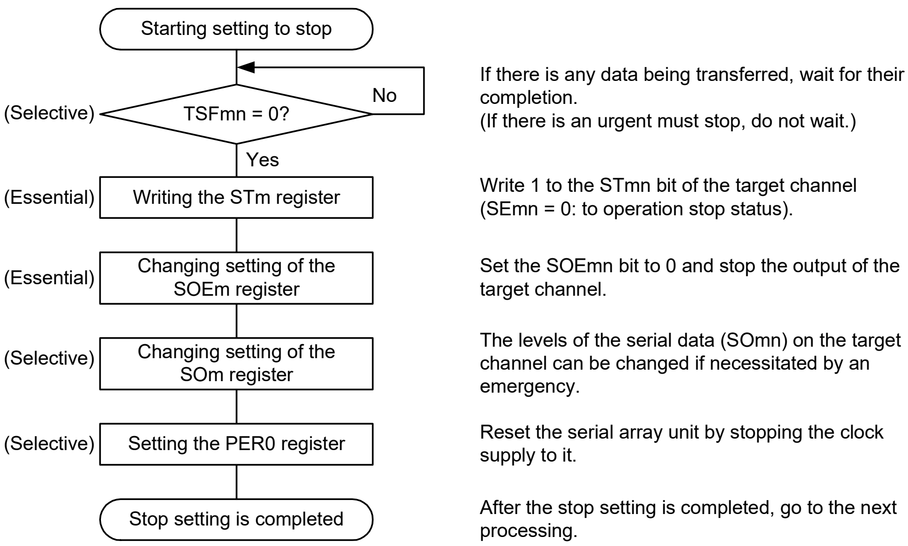
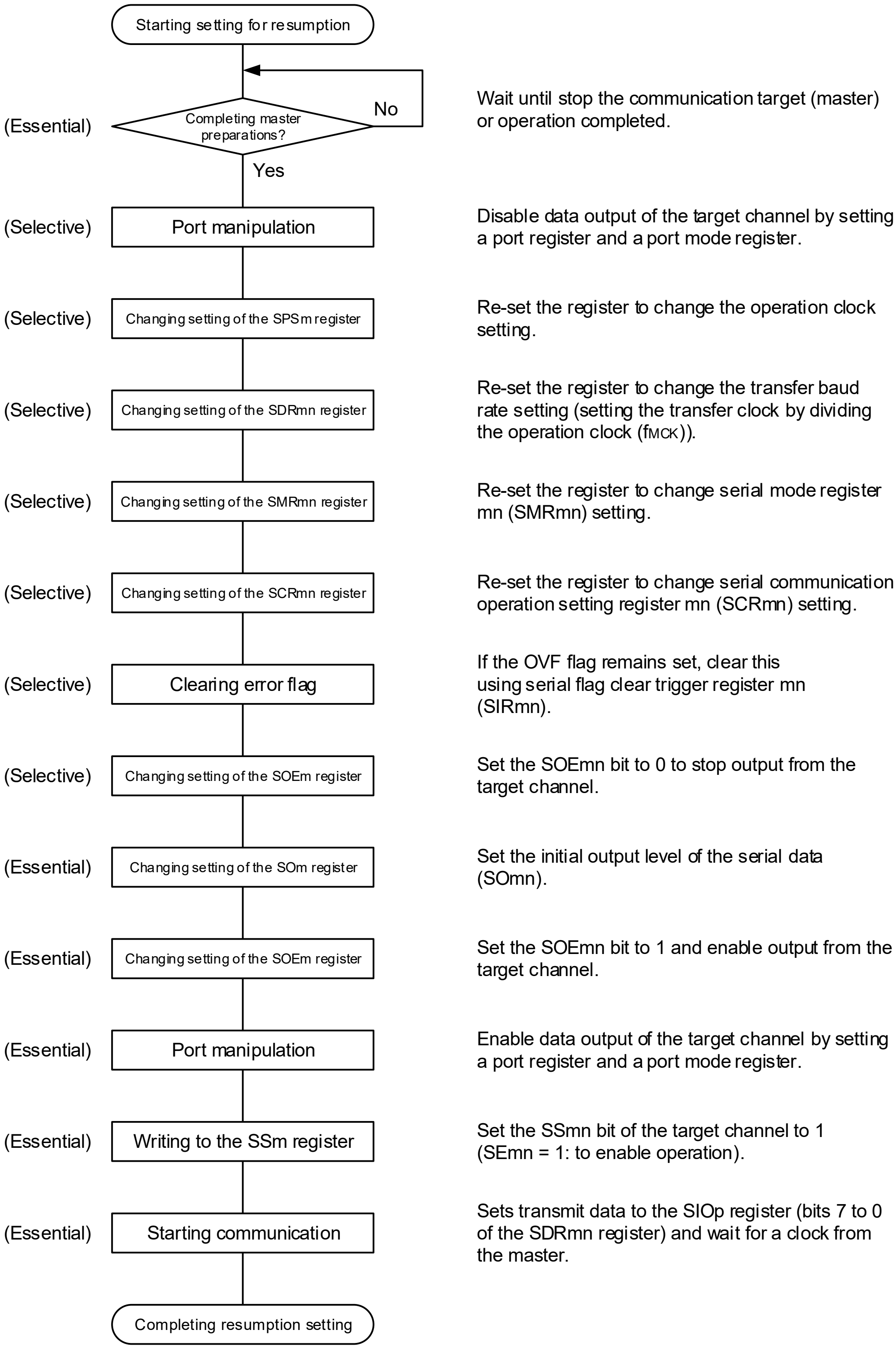


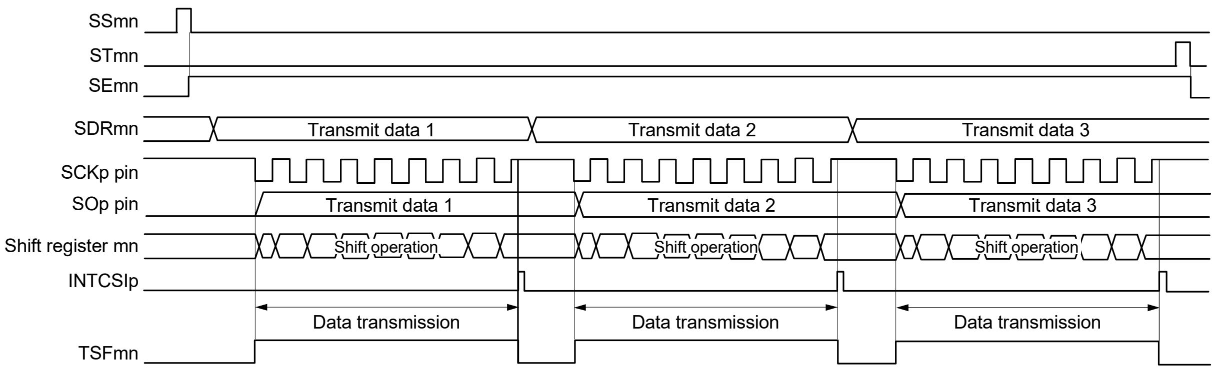
Figure 19 - 53 Procedure for Resuming Slave Transmission



Remark If PER0 is rewritten to stop clock supply while the master transmission is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

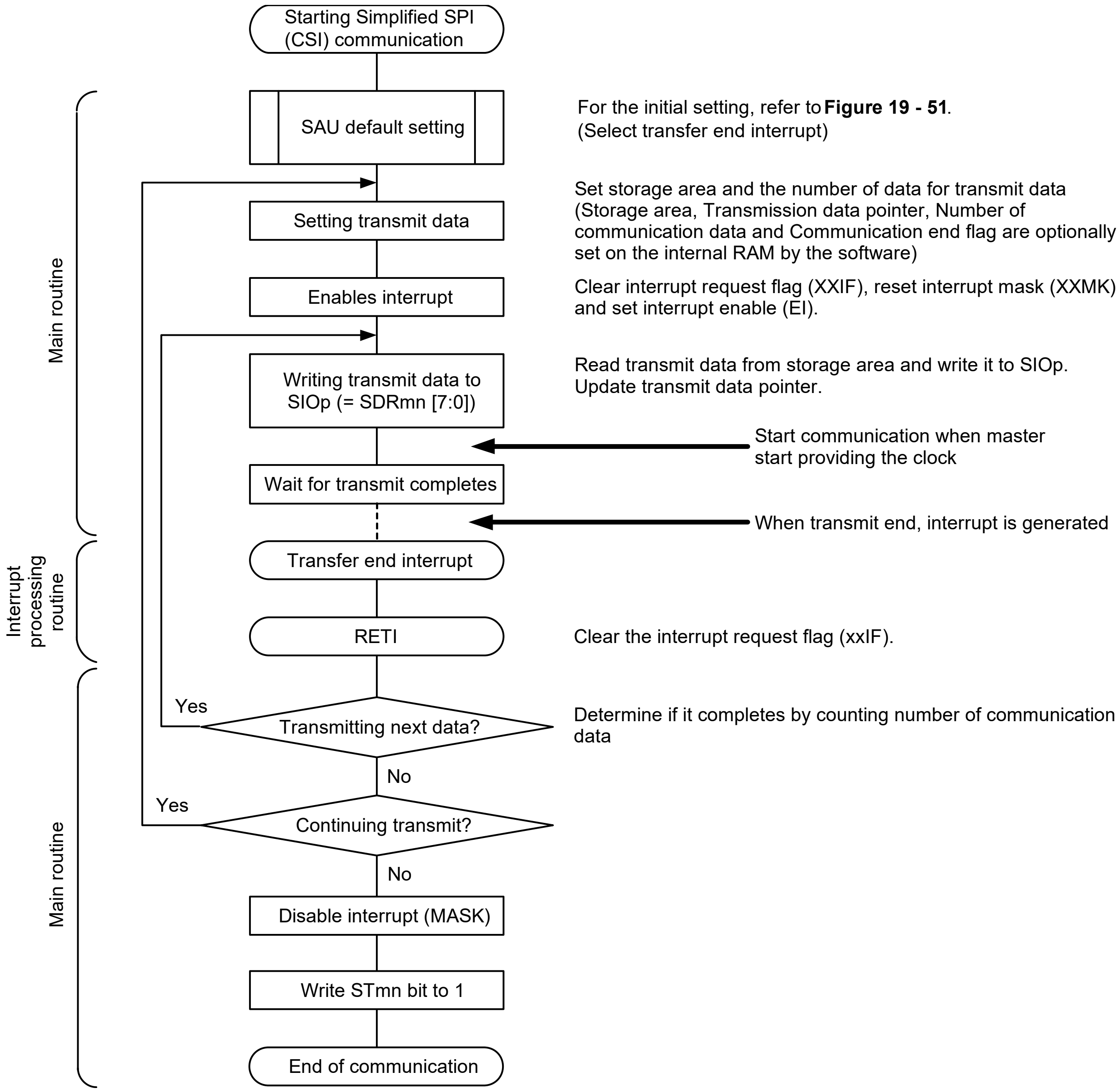
(3) Processing details (in single-transmission mode)

Figure 19 - 54 Timing Chart of Slave Transmission (in Single-Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)



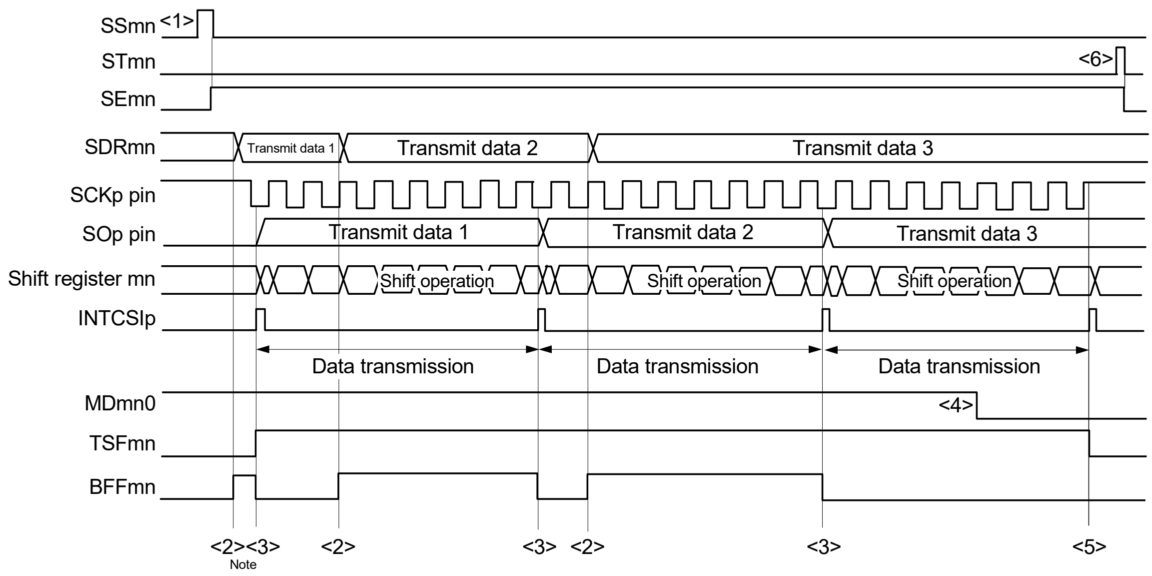
Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 19 - 55 Flowchart of Slave Transmission (in Single-Transmission Mode)



(4) Processing details (in continuous transmission mode)

Figure 19 - 56 Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)

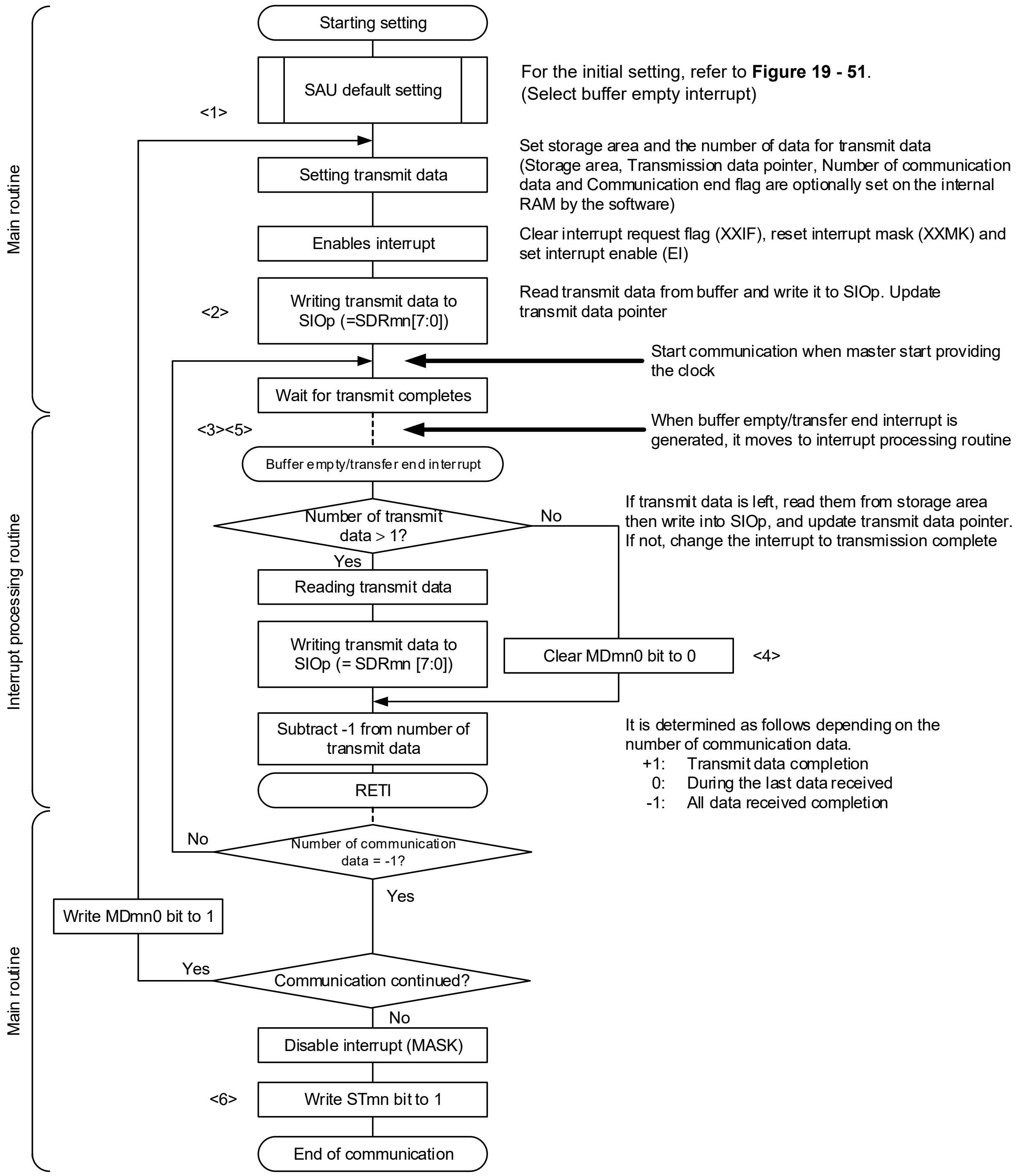


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit starts.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 19 - 57 Flowchart of Slave Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in Figure 19 - 56 Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0).

19.5.5 Slave reception

Slave reception is that the RL78 microcontroller receives data from another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00	CSI01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCK00, SI00	SCK01, SI01
Interrupt	INTCSI00	INTCSI01
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	Overrun error detection flag (OVFmn) only	
Transfer data length	7 or 8 bits	
Transfer rate	Max. $f_{MCK}/12$ [Hz] Notes 1, 2	
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data input starts from the start of the operation of the serial clock. • DAPmn = 1: Data input starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse	
Data direction	MSB or LSB first	

Note 1. Because the external serial clock input to the SCK00 and SCK01 pins is sampled internally and used, the fastest transfer rate is $f_{MCK}/12$ [Hz].

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 33** or **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark 1. f_{MCK}: Operation clock frequency of target channel

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

(1) Register setting

Figure 19 - 58 Example of Contents of Registers for Slave Reception of Simplified SPI (CSI00, CSI01)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0

Operation clock (fMCK) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **19.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	Receive data (SIOp)							

(d) Serial output register m (SOm)... 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	0	0	0	0	0	CKOm2 ×	CKOm1 ×	CKOm0 ×	0	0	0	0	0	SOm2 ×	SOm1 ×	SOm0 ×

(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	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	SOEm1 ×	SOEm0 ×

(f) Serial channel start register m (SSm)... Set only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 0/1	SSm0 0/1

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Remark 2. ■: Setting is fixed in the Simplified SPI (CSI) slave reception mode,
▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 19 - 59 Initial Setting Procedure for Slave Reception

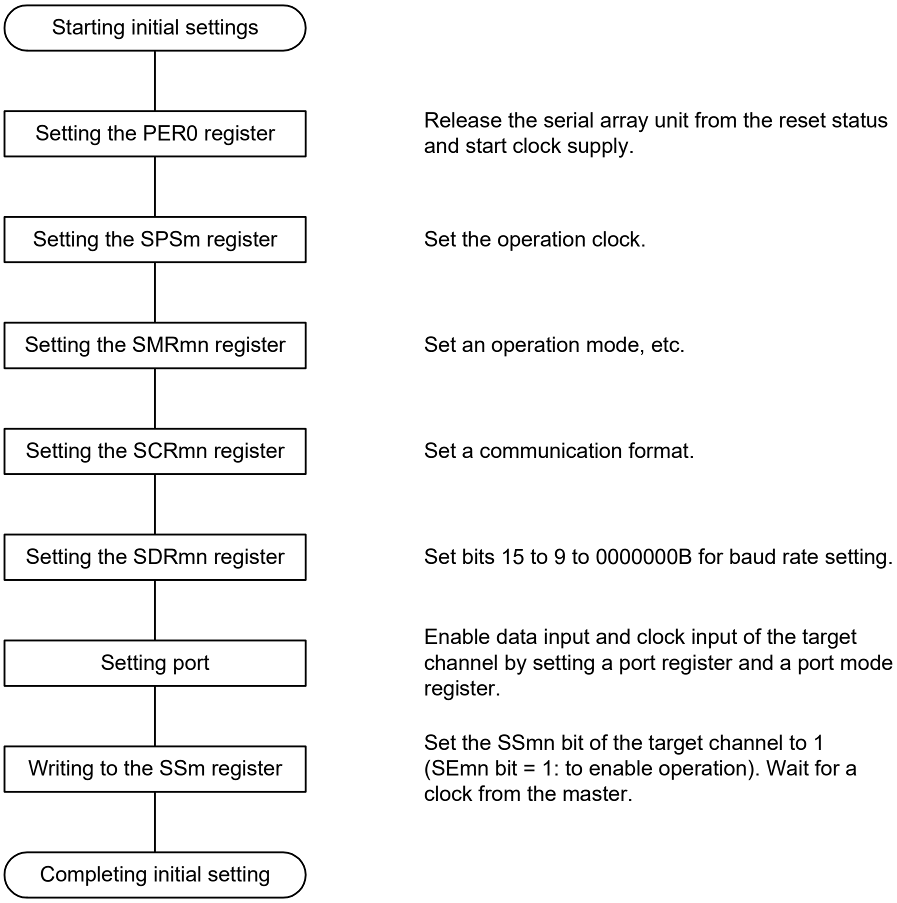


Figure 19 - 60 Procedure for Stopping Slave Reception

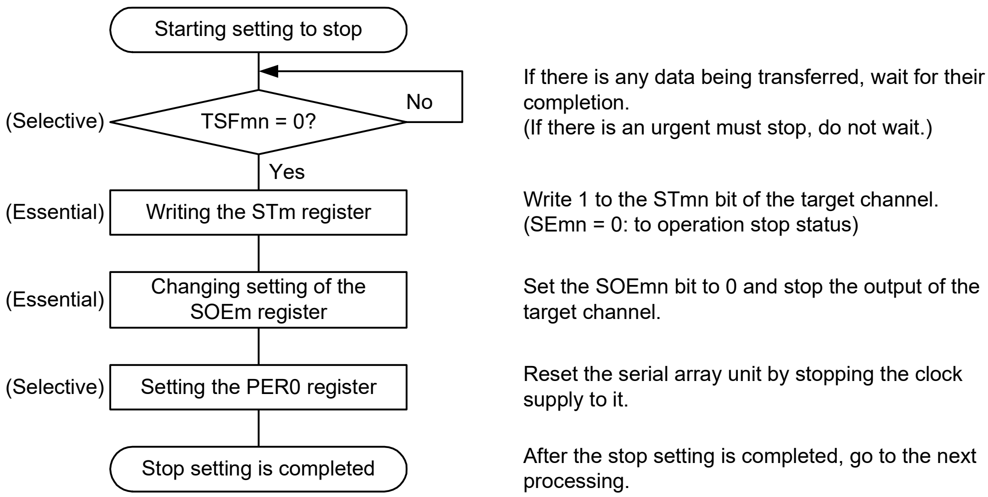
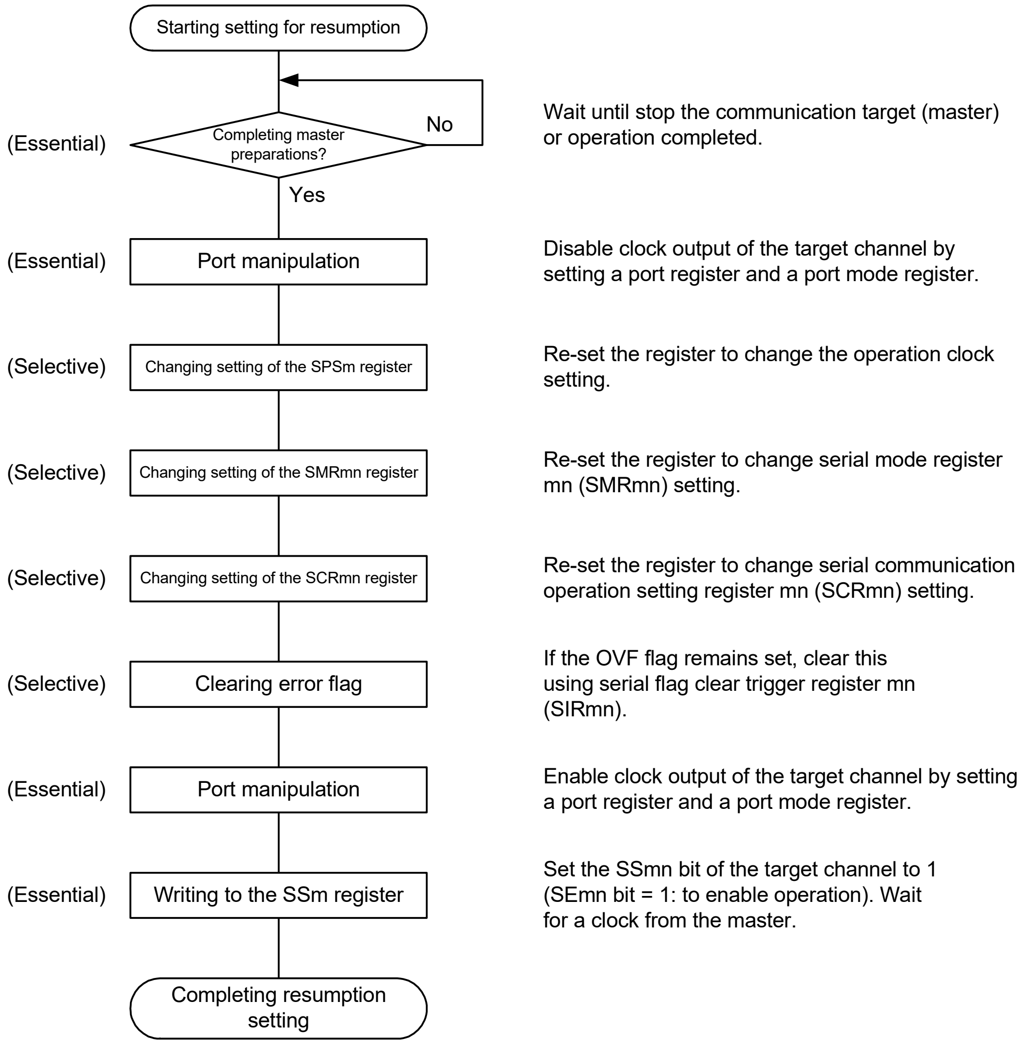


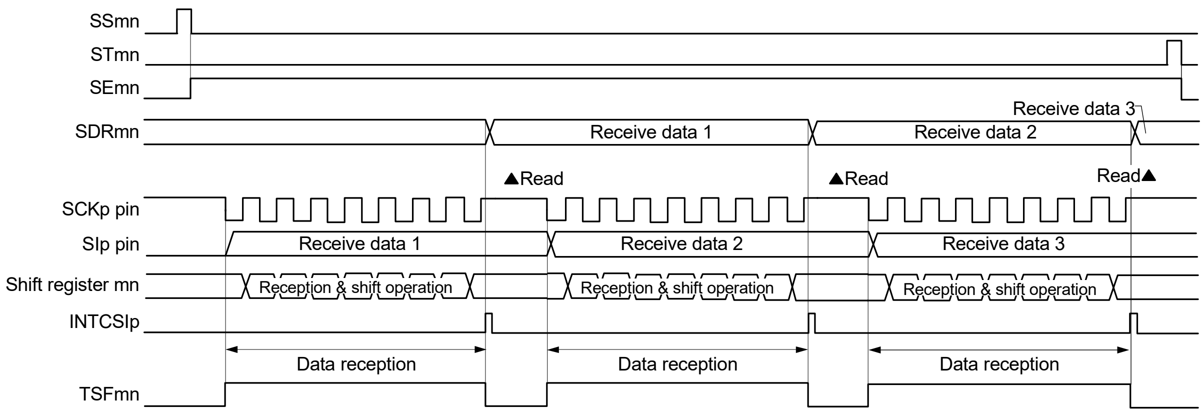
Figure 19 - 61 Procedure for Resuming Slave Reception



Remark If PER0 is rewritten to stop clock supply while the master transmission is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

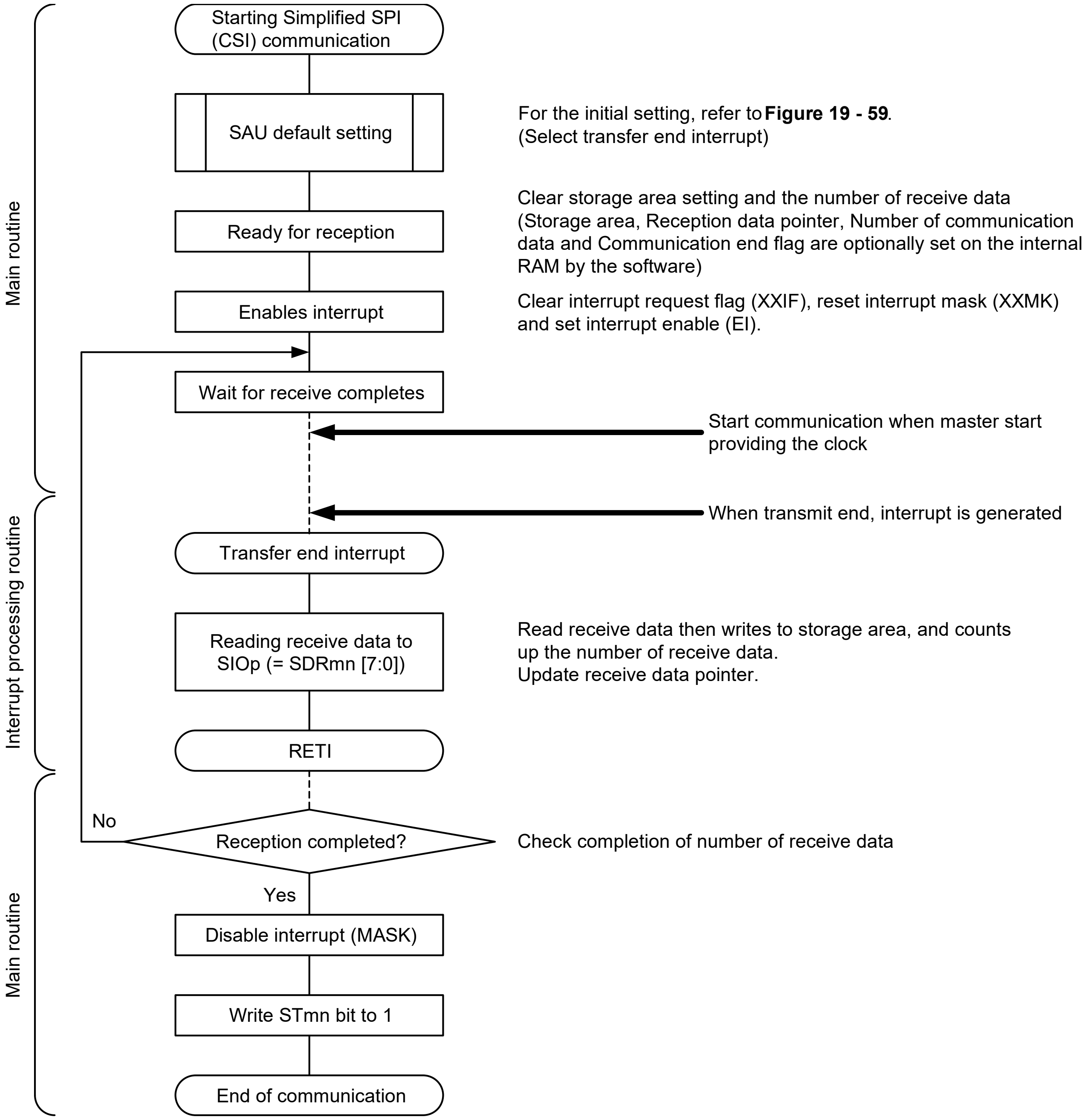
(3) Processing details (in single-reception mode)

Figure 19 - 62 Timing Chart of Slave Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 19 - 63 Flowchart of Slave Reception (in Single-Reception Mode)



19.5.6 Slave transmission/reception

Slave transmission/reception is that the RL78 microcontroller transmits/receives data to/from another device in the state of a transfer clock being input from another device.

Simplified SPI	CSI00	CSI01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCK00, SI00, SO00	SCK01, SI01, SO01
Interrupt	INTCSI00	INTCSI01
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	Overrun error detection flag (OVFmn) only	
Transfer data length	7 or 8 bits	
Transfer rate	Max. $f_{MCK}/12$ [Hz] Notes 1, 2.	
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data I/O starts from the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation.	
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse	
Data direction	MSB or LSB first	

Note 1. Because the external serial clock input to the SCK00 and SCK01 pins is sampled internally and used, the fastest transfer rate is $f_{MCK}/12$ [Hz].

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 33** or **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark 1. f_{MCK}: Operation clock frequency of target channel

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

(1) Register setting

Figure 19 - 64 Example of Contents of Registers for Slave Transmission/Reception of Simplified SPI (CSI00, CSI01)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **19.3 Registers Controlling Serial Array Unit**.)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15–9	8	7–0
SDRmn	0000000 Baud rate setting	0	Transmit data setting/receive data register (SIOp)

(d) Serial output register m (SOm)... Set only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	CKOm2 ×	CKOm1 ×	CKOm0 ×	0	0	0	0	0	SOm2 ×	SOm1 0/1	SOm0 0/1

(e) Serial output enable register m (SOEm)... Set only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	SOEm1 0/1	SOEm0 0/1

(f) Serial channel start register m (SSm)... Set only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 0/1	SSm0 0/1

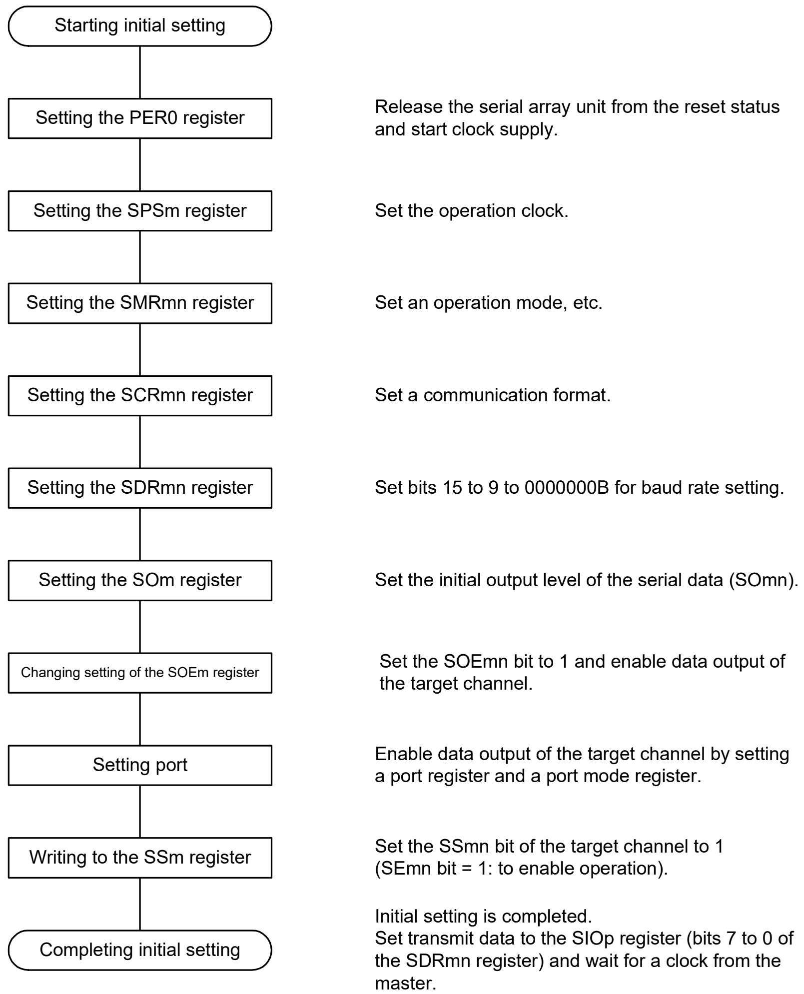
Caution Be sure to set transmit data to the SIOp register before clock supply from the master starts.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Remark 2. ■: Setting is fixed in the Simplified SPI (CSI) master transmission/reception mode
▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 19 - 65 Initial Setting Procedure for Slave Transmission/Reception



Caution Be sure to set transmit data to the SIOp register before clock supply from the master starts.

Figure 19 - 66 Procedure for Stopping Slave Transmission/Reception

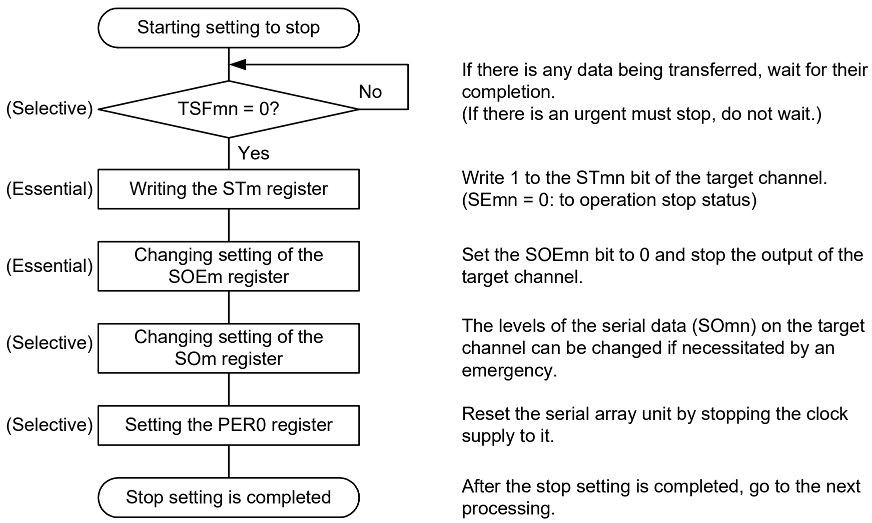
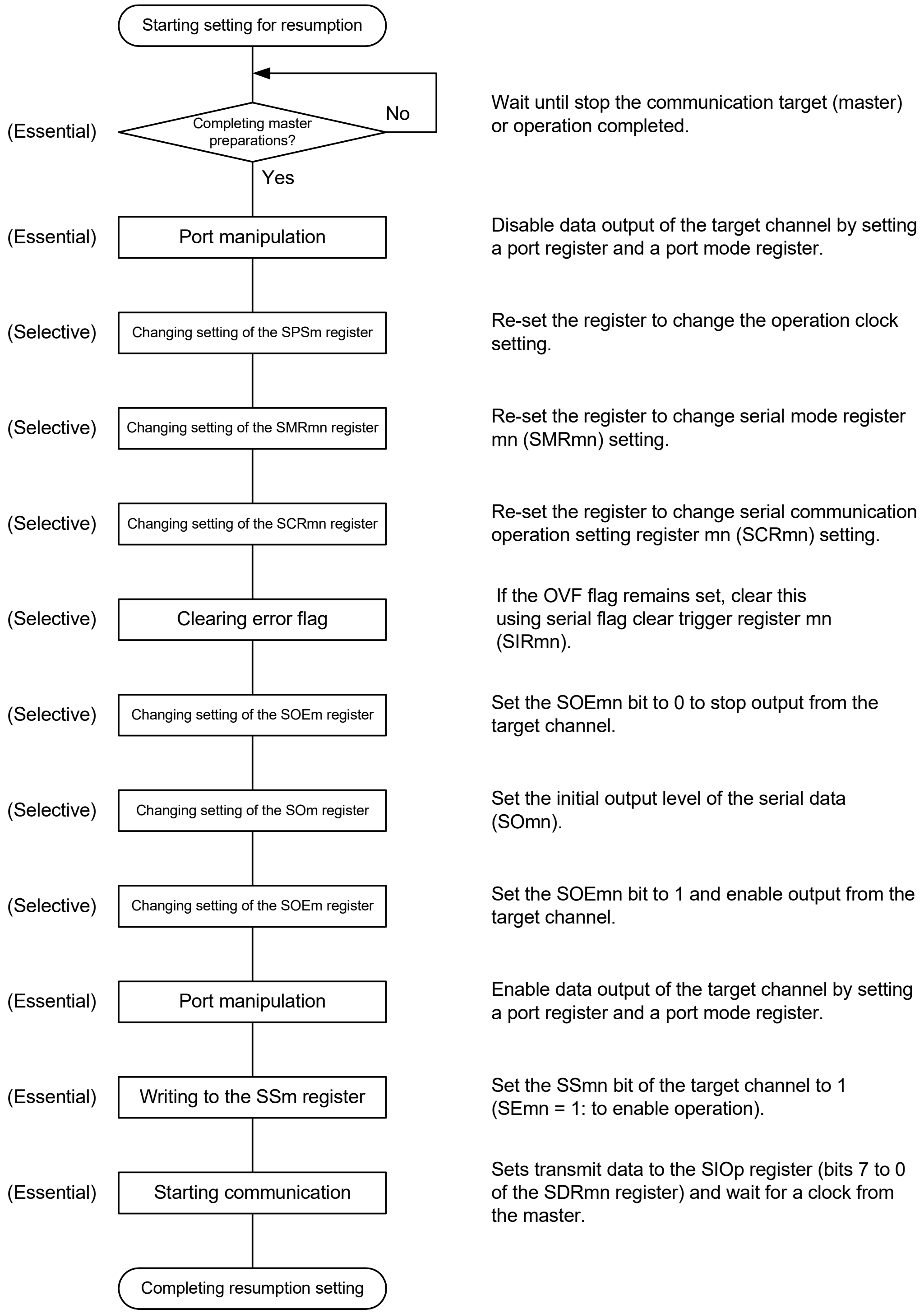


Figure 19 - 67 Procedure for Resuming Slave Transmission/Reception

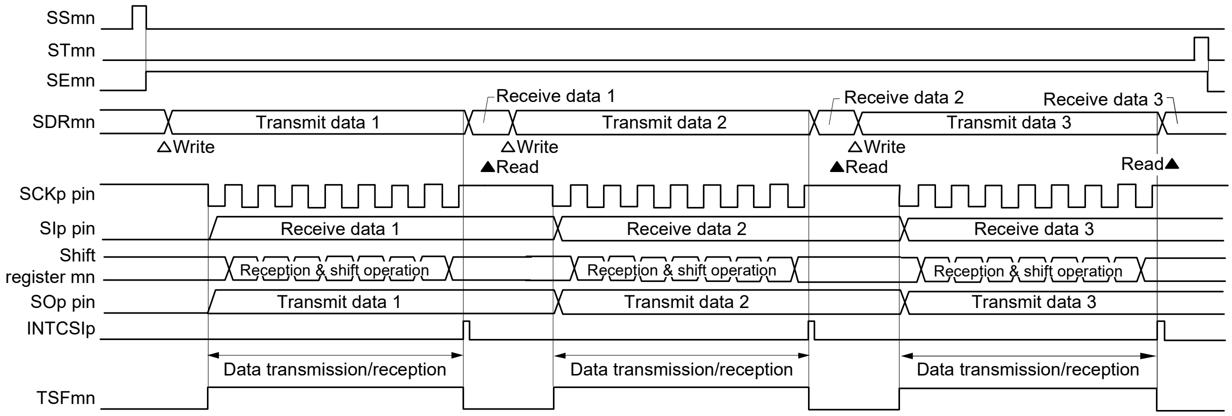


Caution 1. Be sure to set transmit data to the SIOp register before clock supply from the master starts.

Caution 2. If PER0 is rewritten to stop clock supply while the master transmission is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

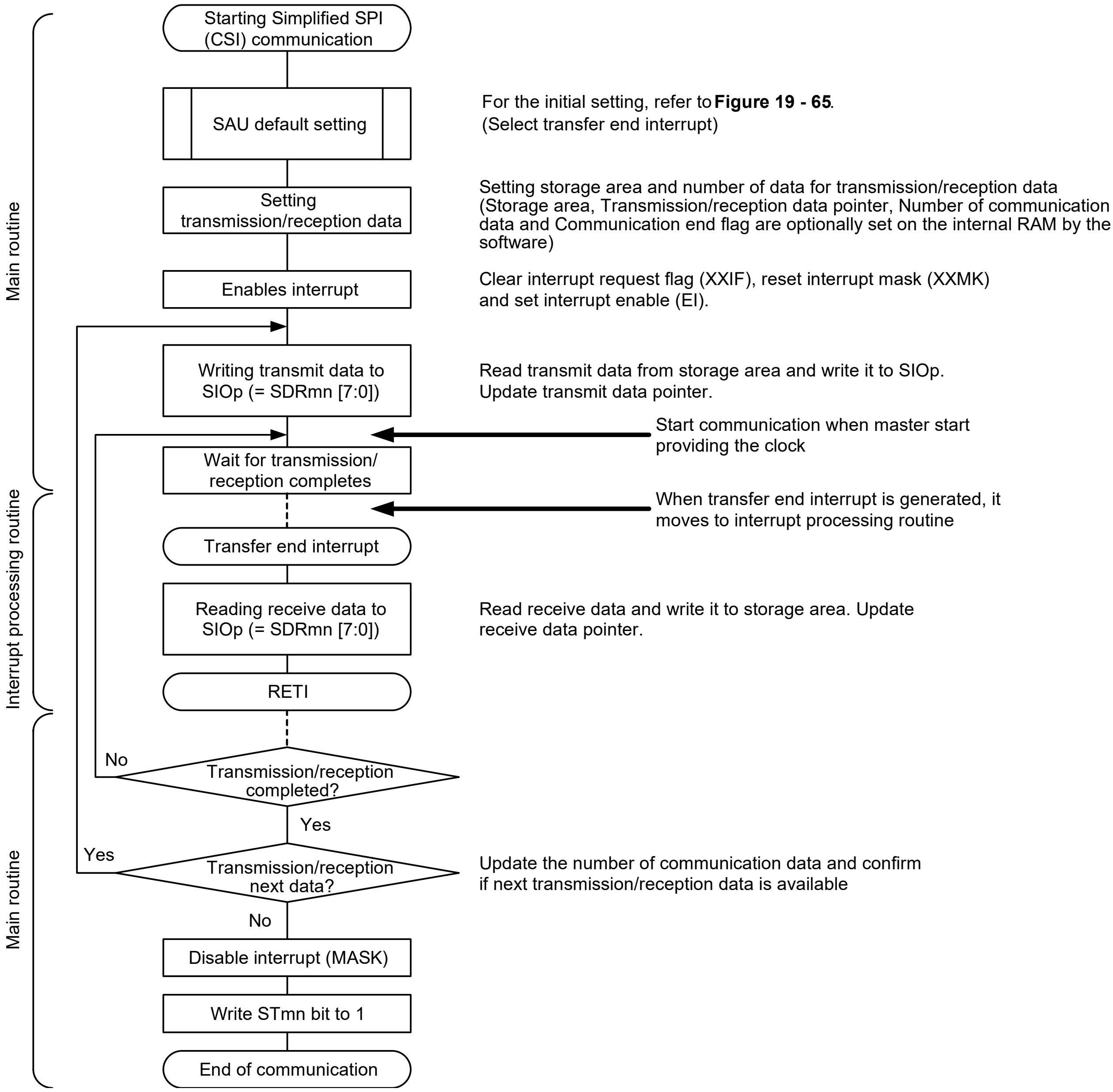
(3) Processing details (in single-transmission/reception mode)

Figure 19 - 68 Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

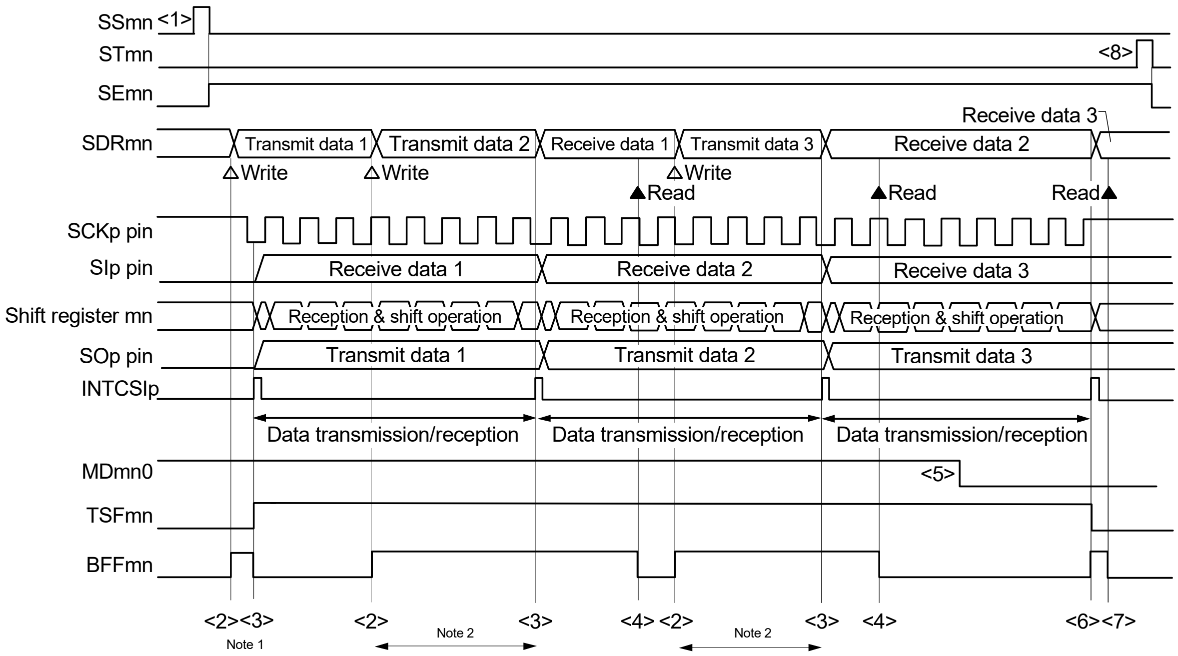
Figure 19 - 69 Flowchart of Slave Transmission/Reception (in Single- Transmission/Reception Mode)



Caution **Be sure to set transmit data to the SIOp register before clock supply from the master starts.**

(4) Processing details (in continuous transmission/reception mode)

Figure 19 - 70 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Note 1. If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

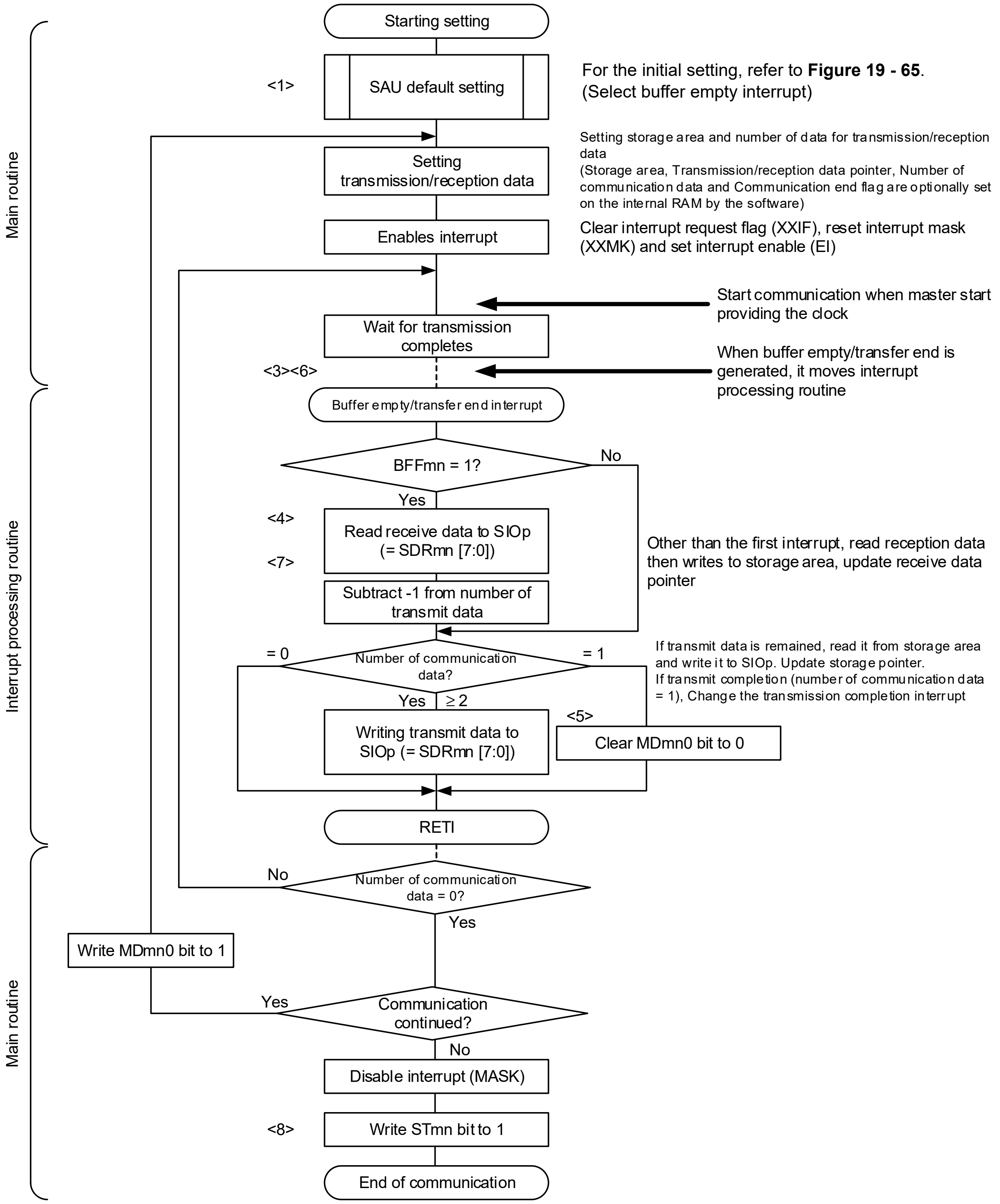
Note 2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution **The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit starts, 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 19 - 71 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode).

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1), p: CSI number (p = 00, 01), mn = 00, 01

Figure 19 - 71 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)



Caution **Be sure to set transmit data to the SIOp register before clock supply from the master starts.**

Remark <1> to <8> in the figure correspond to <1> to <8> in Figure 19 - 70 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

19.5.7 SNOOZE mode function

SNOOZE mode makes simplified SPI (CSI) operate reception by SCKp pin input detection while the STOP mode. Normally simplified SPI (CSI) stops communication in the STOP mode. But, using the SNOOZE mode makes reception simplified SPI (CSI) operate unless the CPU operation by detecting SCKp pin input. The SNOOZE mode can only be used by CSI00.

When using the simplified SPI (CSI) in SNOOZE mode, make the following setting before switching to the STOP mode (see **Figure 19 - 73** and **Figure 19 - 75 Flowchart of SNOOZE Mode Operation**).

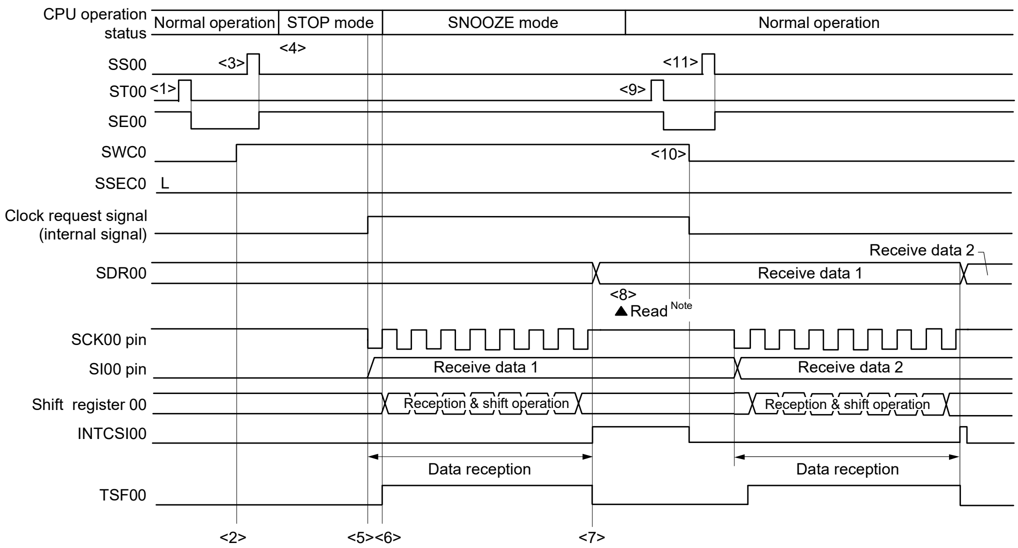
- When using the SNOOZE mode function, set the SWCm bit of serial standby control register m (SSCm) to 1 just before switching to the STOP mode. After the initial setting has been completed, set the SSm0 bit of serial channel start register m (SSm) to 1.
- The CPU shifts to the SNOOZE mode on detecting the valid edge of the SCKp signal following a transition to the STOP mode. A CSIp starts reception on detecting input of the serial clock on the SCKp pin.

Caution 1. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected for fCLK.

Caution 2. The maximum transfer rate when using CSIp in the SNOOZE mode is 1 Mbps.

(1) SNOOZE mode operation (once startup)

Figure 19 - 72 Timing Chart of SNOOZE Mode Operation (once startup) (Type 1: DAPmn = 0, CKPmn = 0)



Note Only read received data while SWCm = 1 and before the next valid edge of the SCKp pin input is detected.

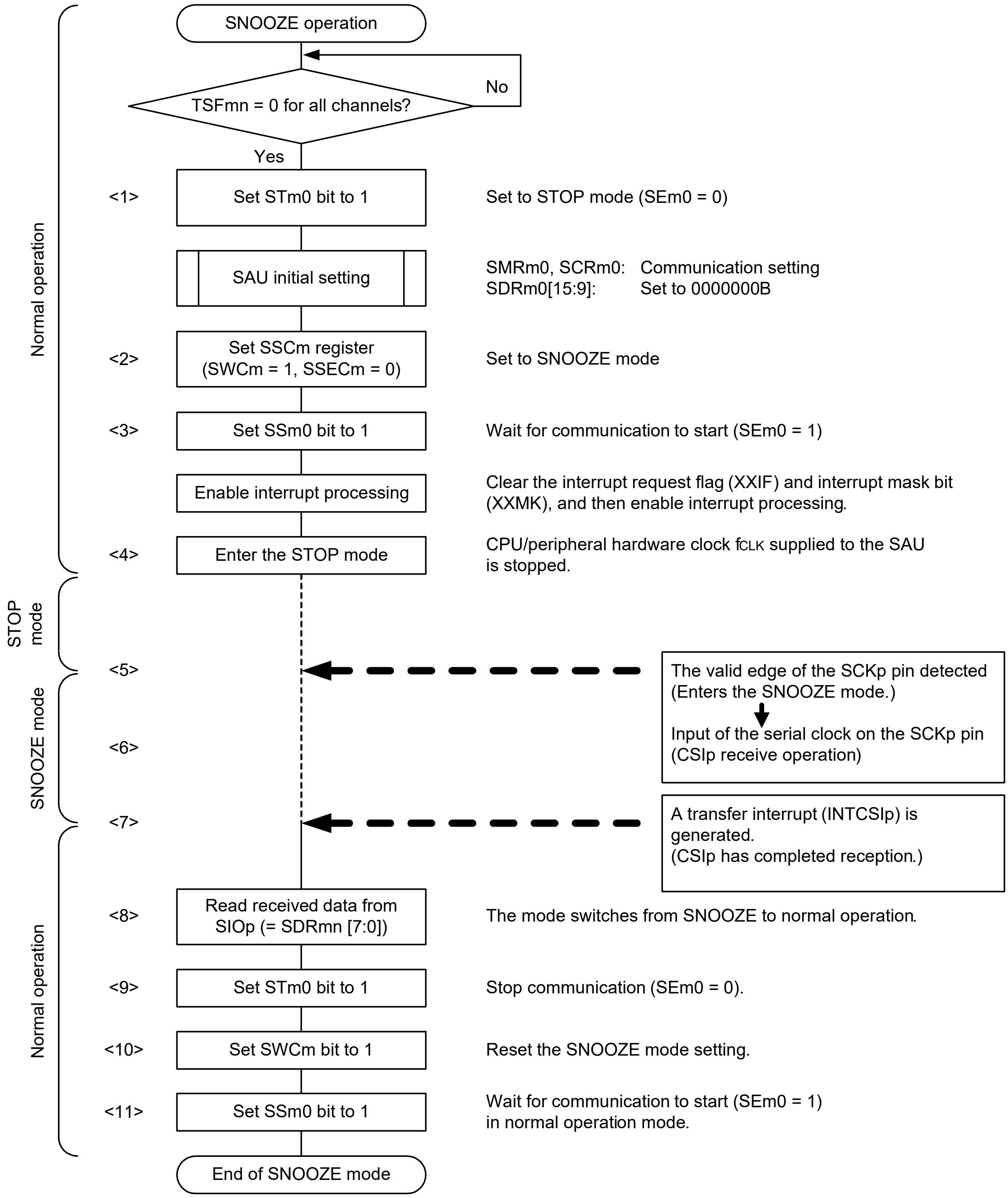
Caution 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (clear the SEm0 bit to stop the operation).
After the reception finishes, also clear the SWCm bit to 0 (to exit SNOOZE mode).

Caution 2. When SWCm = 1, the BFFm1 and OVFm1 flags will not change.

Remark 1. <1> to <11> in the figure correspond to <1> to <11> in Figure 19 - 73 Flowchart of SNOOZE Mode Operation (once startup).

Remark 2. m = 0; n = 0; p = 00

Figure 19 - 73 Flowchart of SNOOZE Mode Operation (once startup)

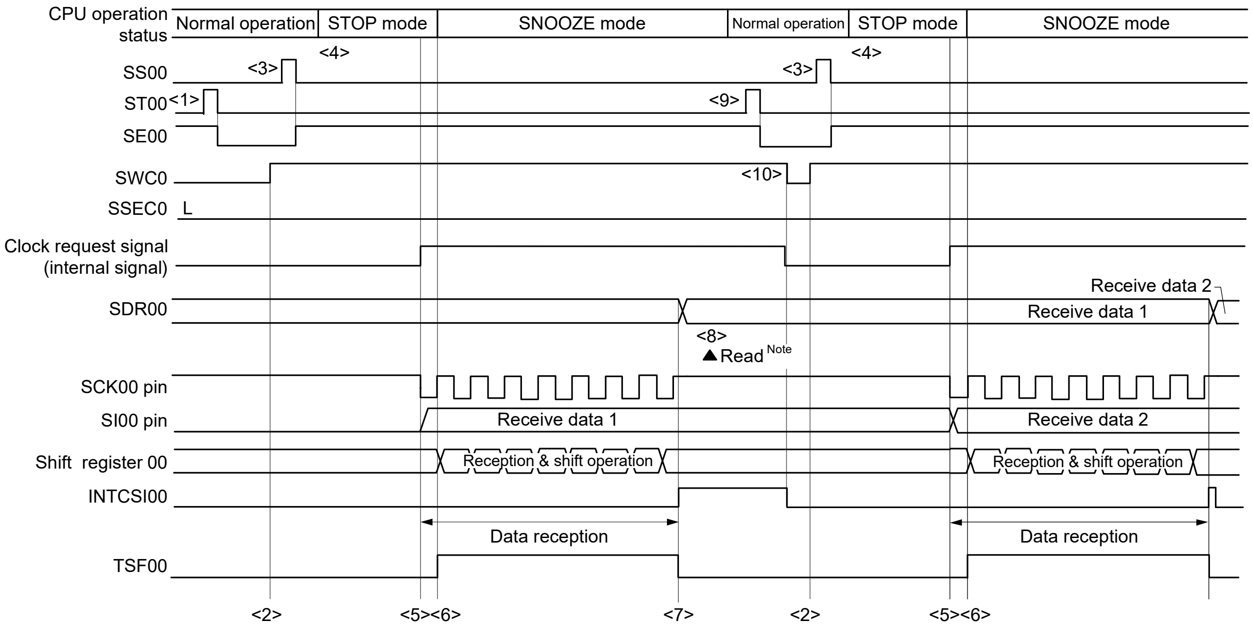


Remark 1. <1> to <11> in the figure correspond to <1> to <11> in Figure 19 - 72 Timing Chart of SNOOZE Mode Operation (once startup) (Type 1: DAPmn = 0, CKPmn = 0).

Remark 2. m = 0; p = 00

(2) SNOOZE mode operation (continuous startup)

Figure 19 - 74 Timing Chart of SNOOZE Mode Operation (continuous startup) (Type 1: DAPmn = 0, CKPmn = 0)



Note Only read received data while SWCm = 1 and before the next valid edge of the SCKp pin input is detected.

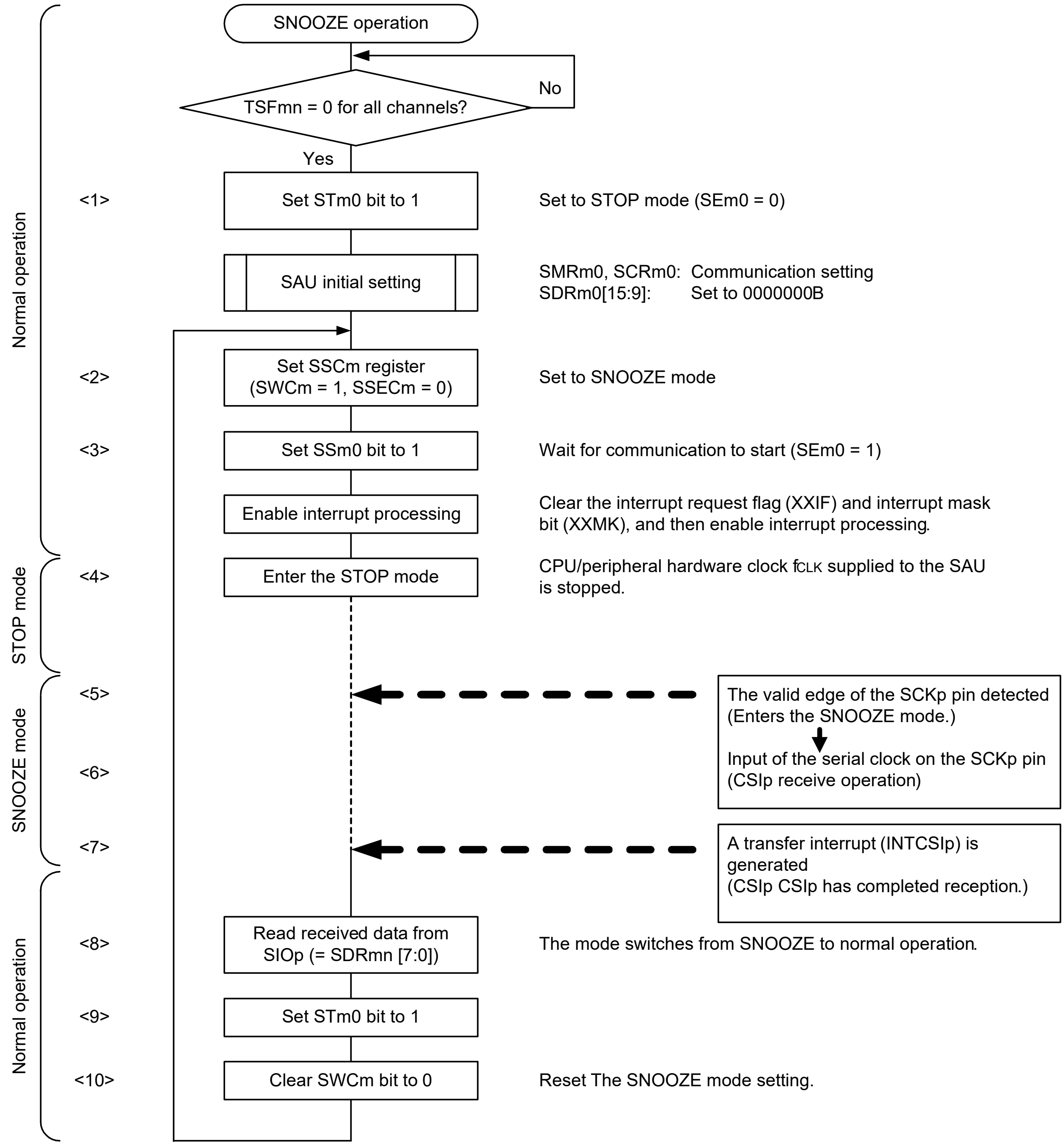
Caution 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm0 bit to 1 (clear the SEm0 bit to stop the operation).
After the reception finishes, also clear the SWCm bit to 0 (to exit SNOOZE mode).

Caution 2. When SWCm = 1, the BFFm1 and OVFm1 flags will not change.

Remark 1. <1> to <10> in the figure correspond to <1> to <10> in Figure 19 - 75 Flowchart of SNOOZE Mode Operation (continuous startup).

Remark 2. m = 0; n = 0

Figure 19 - 75 Flowchart of SNOOZE Mode Operation (continuous startup)



Remark 1. <1> to <10> in the figure correspond to <1> to <10> in Figure 19 - 74 Timing Chart of SNOOZE Mode Operation (continuous startup) (Type 1: DAPmn = 0, CKPmn = 0).

Remark 2. m = 0; p = 00

19.5.8 Calculating transfer clock frequency

The transfer clock frequency for Simplified SPI (CSI00, CSI01) communication can be calculated by the following expressions.

(1) Master

Transfer clock frequency = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [Hz]

(2) Slave

Transfer clock frequency = {Frequency of serial clock (SCK) supplied by master} Note [Hz]

Note The permissible maximum transfer clock frequency is $f_{MCK}/6$.

Remark The value of SDRmn[15:9] is the value of bits 15 to 9 of serial data register mn (SDRmn) (0000000B to 1111111B) and therefore is 0 to 127.

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 19 - 3 Selection of Operation Clock for Simplified SPI

SMRmn Register	SPSm Register								Operation Clock (f_{MCK}) Note	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 32 MHz
0	×	×	×	×	0	0	0	0	f_{CLK}	32 MHz
	×	×	×	×	0	0	0	1	$f_{CLK}/2$	16 MHz
	×	×	×	×	0	0	1	0	$f_{CLK}/2^2$	8 MHz
	×	×	×	×	0	0	1	1	$f_{CLK}/2^3$	4 MHz
	×	×	×	×	0	1	0	0	$f_{CLK}/2^4$	2 MHz
	×	×	×	×	0	1	0	1	$f_{CLK}/2^5$	1 MHz
	×	×	×	×	0	1	1	0	$f_{CLK}/2^6$	500 kHz
	×	×	×	×	0	1	1	1	$f_{CLK}/2^7$	250 kHz
	×	×	×	×	1	0	0	0	$f_{CLK}/2^8$	125 kHz
	×	×	×	×	1	0	0	1	$f_{CLK}/2^9$	62.5 kHz
	×	×	×	×	1	0	1	0	$f_{CLK}/2^{10}$	31.25 kHz
	×	×	×	×	1	0	1	1	$f_{CLK}/2^{11}$	15.63 kHz
	×	×	×	×	1	1	0	0	$f_{CLK}/2^{12}$	7.81 kHz
	×	×	×	×	1	1	0	1	$f_{CLK}/2^{13}$	3.91 kHz
	×	×	×	×	1	1	1	0	$f_{CLK}/2^{14}$	1.95 kHz
	×	×	×	×	1	1	1	1	$f_{CLK}/2^{15}$	977 Hz
1	0	0	0	0	×	×	×	×	f_{CLK}	32 MHz
	0	0	0	1	×	×	×	×	$f_{CLK}/2$	16 MHz
	0	0	1	0	×	×	×	×	$f_{CLK}/2^2$	8 MHz
	0	0	1	1	×	×	×	×	$f_{CLK}/2^3$	4 MHz
	0	1	0	0	×	×	×	×	$f_{CLK}/2^4$	2 MHz
	0	1	0	1	×	×	×	×	$f_{CLK}/2^5$	1 MHz
	0	1	1	0	×	×	×	×	$f_{CLK}/2^6$	500 kHz
	0	1	1	1	×	×	×	×	$f_{CLK}/2^7$	250 kHz
	1	0	0	0	×	×	×	×	$f_{CLK}/2^8$	125 kHz
	1	0	0	1	×	×	×	×	$f_{CLK}/2^9$	62.5 kHz
	1	0	1	0	×	×	×	×	$f_{CLK}/2^{10}$	31.25 kHz
	1	0	1	1	×	×	×	×	$f_{CLK}/2^{11}$	15.63 kHz
	1	1	0	0	×	×	×	×	$f_{CLK}/2^{12}$	7.81 kHz
	1	1	0	1	×	×	×	×	$f_{CLK}/2^{13}$	3.91 kHz
	1	1	1	0	×	×	×	×	$f_{CLK}/2^{14}$	1.95 kHz
	1	1	1	1	×	×	×	×	$f_{CLK}/2^{15}$	977 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Remark 1. ×: Don't care

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

19.5.9 Procedure for processing errors that occurred during Simplified SPI (CSI00, CSI01) communication

The procedure for processing errors that occurred during Simplified SPI (CSI00, CSI01) communication is described in Figure 19 - 76.

Figure 19 - 76 Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

19.6 Clocked Serial Communication with Slave Select Input Function

Channel 0 of SAU0 correspond to the clocked serial communication with slave select input function.

[Data transmission/reception]
- Data length of 7 or 8 bits
- Phase control of transmit/receive data
- MSB/LSB first selectable
- Level setting of transmit/receive data

[Clock control]
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel
- Maximum transfer rate Note

  During slave communication: Max. f_{MCK}/12

[Interrupt function]
- Transfer end interrupt/buffer empty interrupt

[Error detection flag]
- Overrun error

Note Use the clocks within a range satisfying the SCK cycle time (t_{KCY}) characteristics. For details, see **CHAPTER 33** or **CHAPTER 34 ELECTRICAL SPECIFICATIONS**.

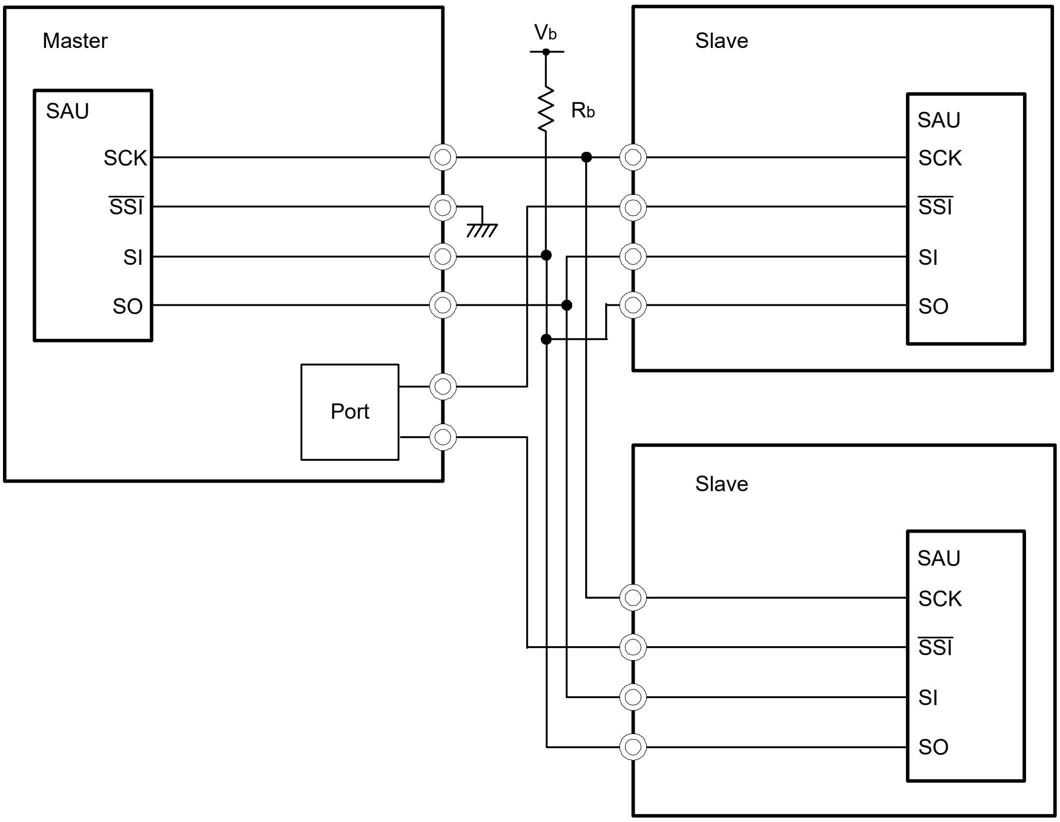
Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^{2}C
0	0	**CSI00 (supporting slave select input function)**	UART0 (supporting LIN-bus)	IIC00
	1	CSI01		IIC01
	2	—	UART1	—
	3	—		—

Slave select input function performs the following three types of communication operations.
- Slave transmission (See **19.6.1**.)
- Slave reception (See **19.6.2**.)
- Slave transmission/reception (See **19.6.3**.)

Multiple slaves can be connected to a master and communication can be performed by using the slave select input function. The master outputs a slave select signal to the slave (one) that is the other party of communication, and each slave judges whether it has been selected as the other party of communication and controls the SO pin output. When a slave is selected, transmit data can be communicated from the SO pin to the master. When a slave is not selected, the SO pin is set to high-level output. Therefore, in an environment where multiple slaves are connected, it is necessary set the SO pin to N-ch open-drain and pull up the node. Furthermore, when a slave is not selected, no transmission/reception operation is performed even if a serial clock is input from the master.

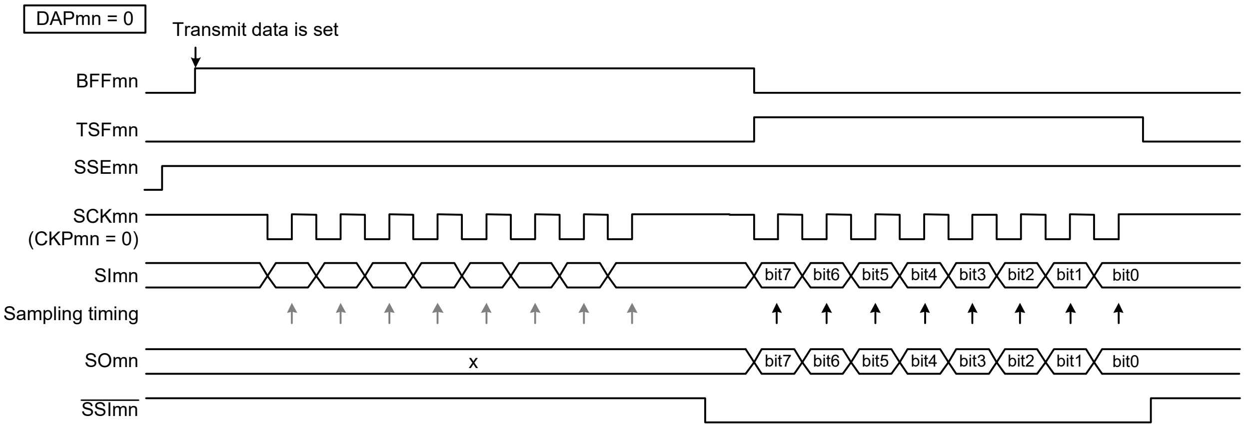
Caution Output the slave select signal by port manipulation.

Figure 19 - 77 Example of Slave Select Input Function Configuration

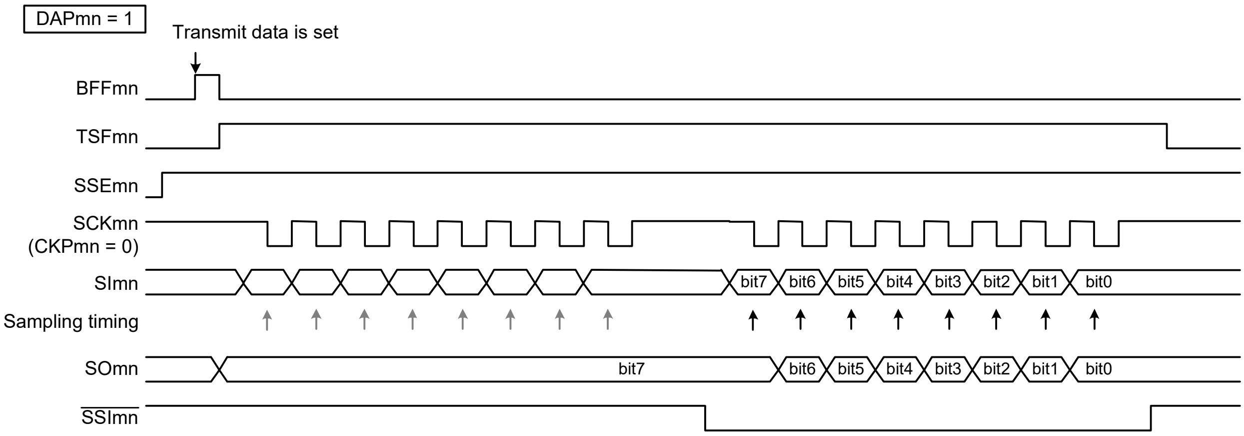


Caution Make sure VDD ≥ Vb.
Select the N-ch open-drain output (VDD tolerance) mode for the SO00 pin.

Figure 19 - 78 Slave Select Input Function Timing Diagram



While $\overline{\text{SSImn}}$ is at high level, transmission is not performed even if the falling edge of SCKmn (serial clock) arrives, and neither is receive data sampled in synchronization with the rising edge.
When $\overline{\text{SSImn}}$ goes to low level, data is output (shifted) in synchronization with the falling edge of the serial clock and a reception operation is performed in synchronization with the rising edge.



If DAPmn = 1, when transmit data is set while $\overline{\text{SSImn}}$ is at high level, the first data (bit 7) is output to the data output. However, no shift operation is performed even if the rising edge of SCKmn (serial clock) arrives, and neither is receive data sampled in synchronization with the falling edge. When $\overline{\text{SSImn}}$ goes to low level, data is output (shifted) in synchronization with the next rising edge and a reception operation is performed in synchronization with the falling edge.

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

19.6.1 Slave transmission

Slave transmission is that the RL78 microcontroller transmits data to another device in the state of a transfer clock being input from another device.

Slave select Input function	CSI00
Target channel	Channel 0 of SAU0
Pins used	SCK00, SO00, $\overline{SSI00}$
Interrupt	INTCSI00
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	Overrun error detection flag (OVFmn) only
Transfer data length	7 or 8 bits
Transfer rate	Max. $f_{MCK}/12$ [Hz] Notes 1, 2
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data output starts from the start of the operation of the serial clock. • DAPmn = 1: Data output starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse
Data direction	MSB or LSB first
Slave select Input function	Slave select input function operation selectable

Note 1. Because the external serial clock input to the SCK00 pin is sampled internally and used, the fastest transfer rate is $f_{MCK}/12$ [Hz].

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 33** or **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark 1. f_{MCK}: Operation clock frequency of target channel

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

(1) Register setting

Figure 19 - 79 Example of Contents of Registers for Slave Transmission of Slave Select Input Function (CSI00) (1/2)

(a) Serial mode register mn (SMRmn)

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

SCRmn	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	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 **19.3 Registers Controlling Serial Array Unit**)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

SDRmn	15–9	8	7–0
	0000000 Baud rate setting	0	Transmit data setting (SIOp)

(d) Serial output register m (SOm)... Set only the bits of the target channel.

SOm	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						CKOm2	CKOm1	CKOm0						SOm2	SOm1	SOm0
	0	0	0	0	0	×	×	×	0	0	0	0	0	×	×	0/1

(e) Serial output enable register m (SOEm)... Set only the bits of the target channel to 1.

SOEm	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
														SOEm2	SOEm1	SOEm0
	0	0	0	0	0	0	0	0	0	0	0	0	0	×	×	0/1

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Remark 2. □: Setting is fixed in the Simplified SPI (CSI) slave transmission mode,
■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 19 - 80 Example of Contents of Registers for Slave Transmission of Slave Select Input Function (CSI00) (2/2)

(f) Serial channel start register m (SSm)... Set only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm													SSm3	SSm2	SSm1	SSm0
	0	0	0	0	0	0	0	0	0	0	0	0	×	×	×	0/1

(g) Input switch control register (ISC)... $\overline{\text{SSI00}}$ input setting in CSI00 slave channel (channel 0 of unit 0).

	7	6	5	4	3	2	1	0
ISC	SSIE00						ISC1	ISC0
	0/1	0	0	0	0	0	0/1	0/1

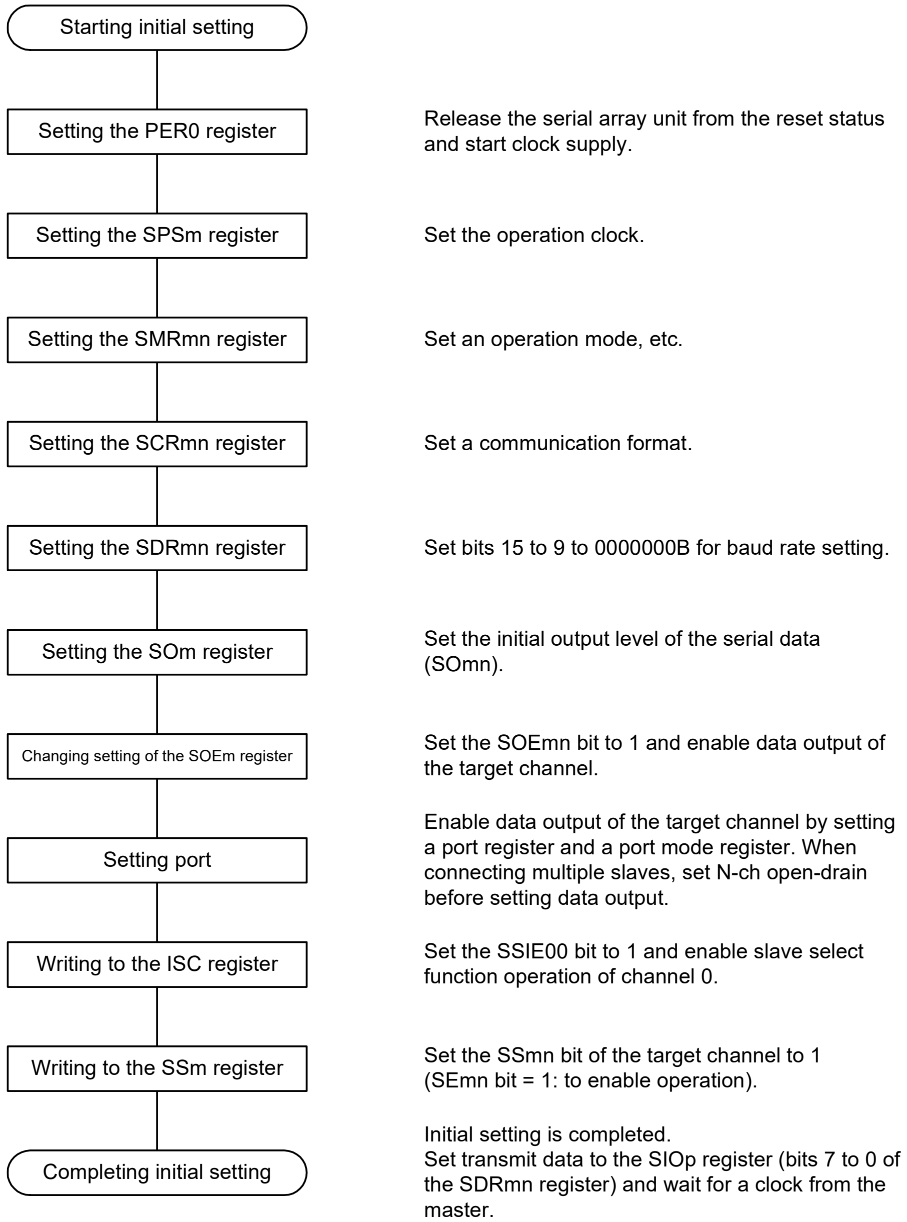
SSIE00:
0: Disables the input value of the $\overline{\text{SSI00}}$ pin
1: Enables the input value of the $\overline{\text{SSI00}}$ pin

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Remark 2. ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

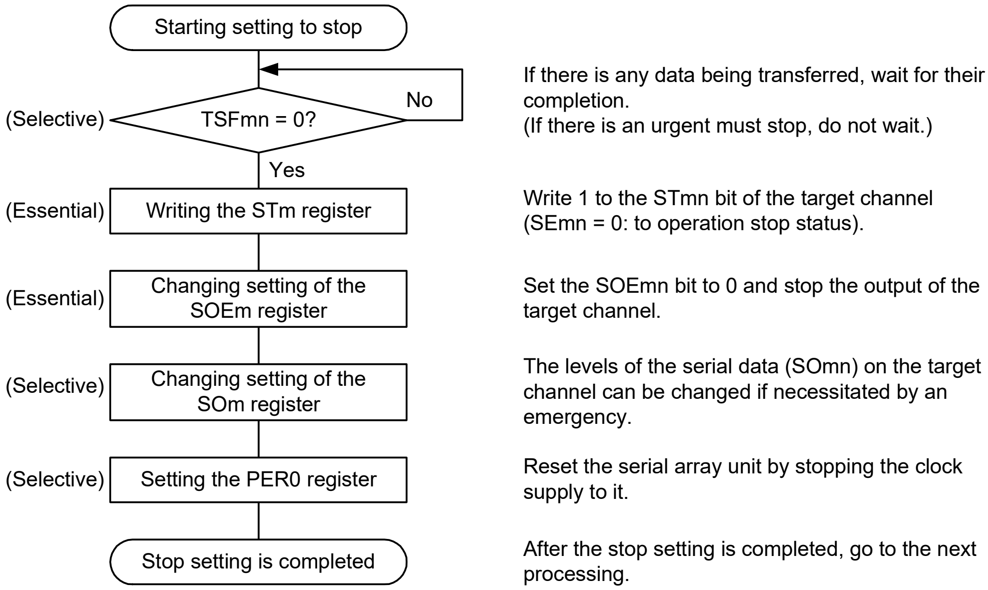
(2) Operation procedure

Figure 19 - 81 Initial Setting Procedure for Slave Transmission



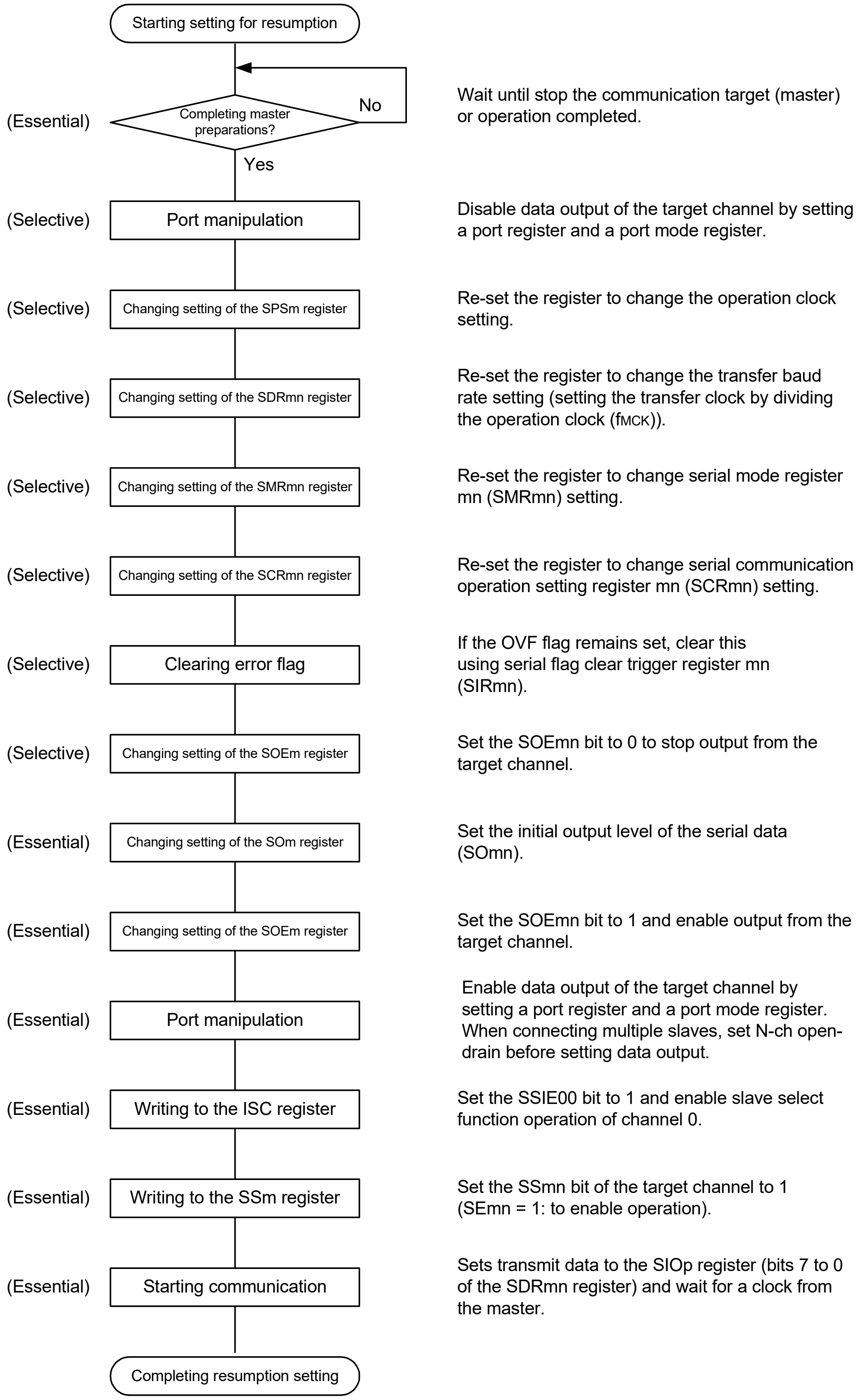
Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 19 - 82 Procedure for Stopping Slave Transmission



Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 19 - 83 Procedure for Resuming Slave Transmission

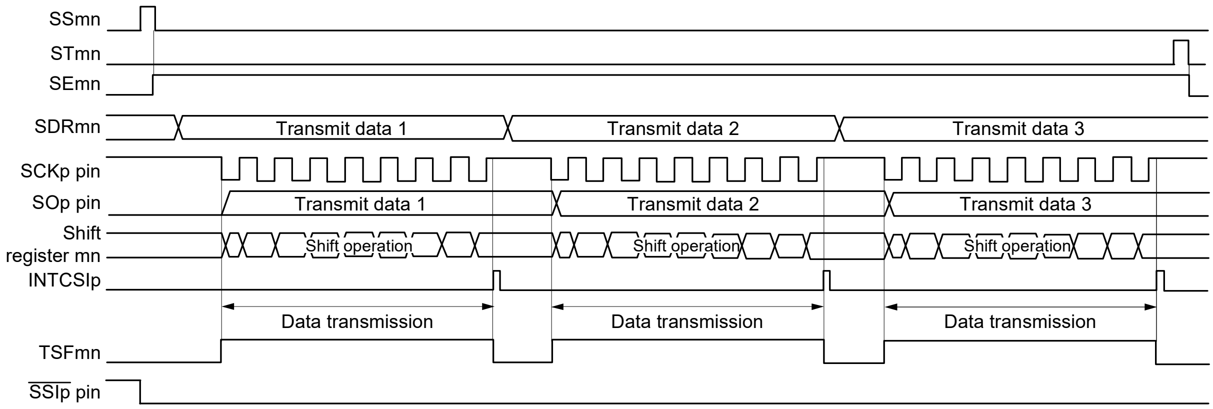


Remark 1. If PER0 is rewritten to stop clock supply while the master transmission is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

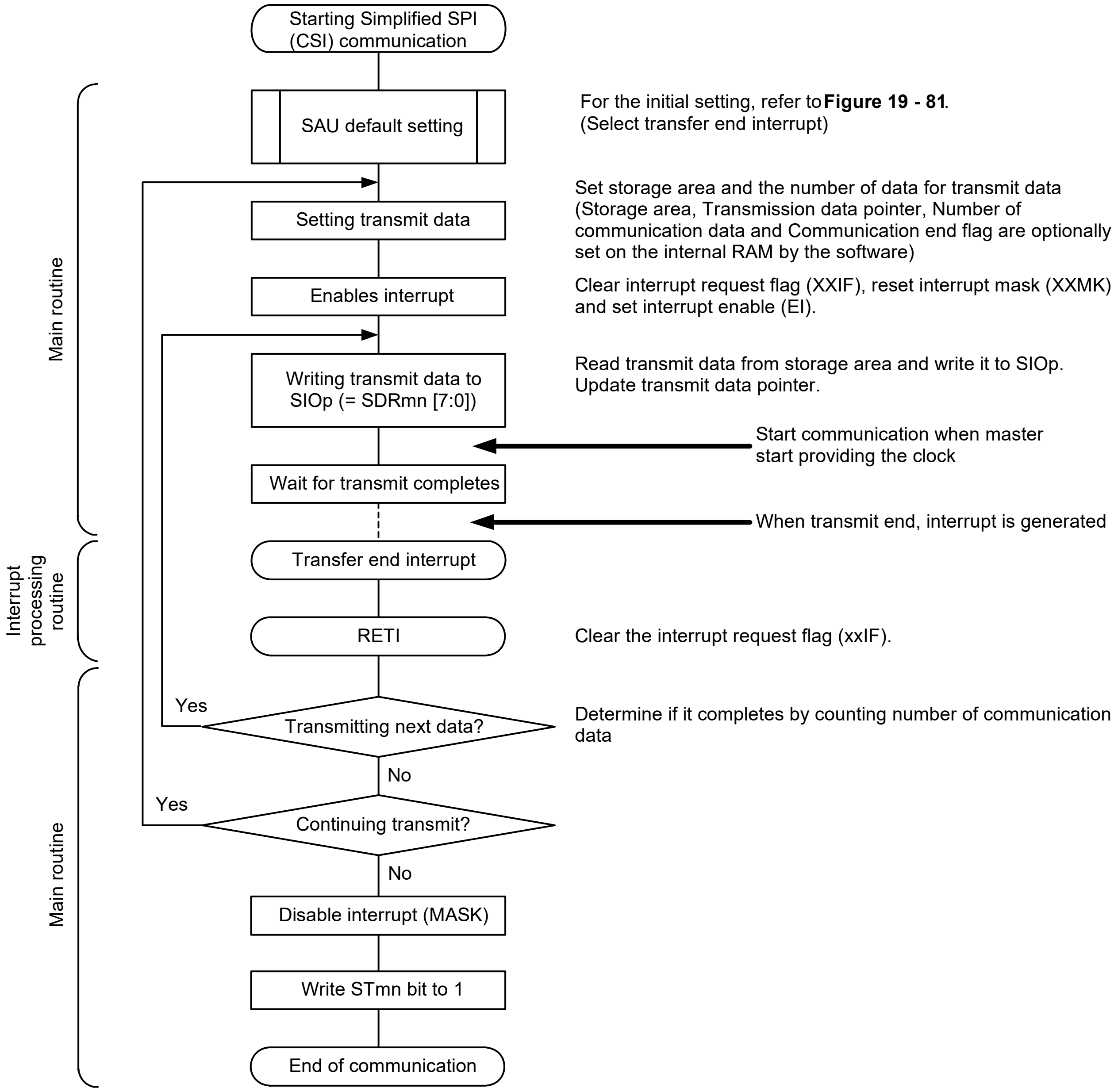
(3) Processing details (in single-transmission mode)

Figure 19 - 84 Timing Chart of Slave Transmission (in Single-Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

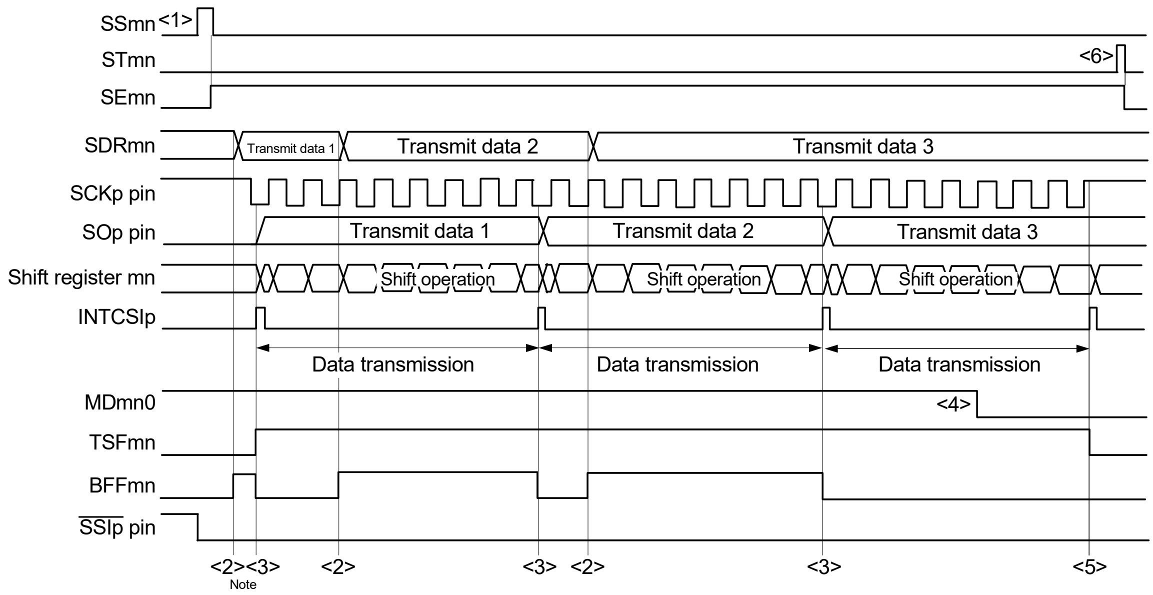
Figure 19 - 85 Flowchart of Slave Transmission (in Single-Transmission Mode)



Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

(4) Processing details (in continuous transmission mode)

Figure 19 - 86 Timing Chart of Slave Transmission (in Continuous Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)

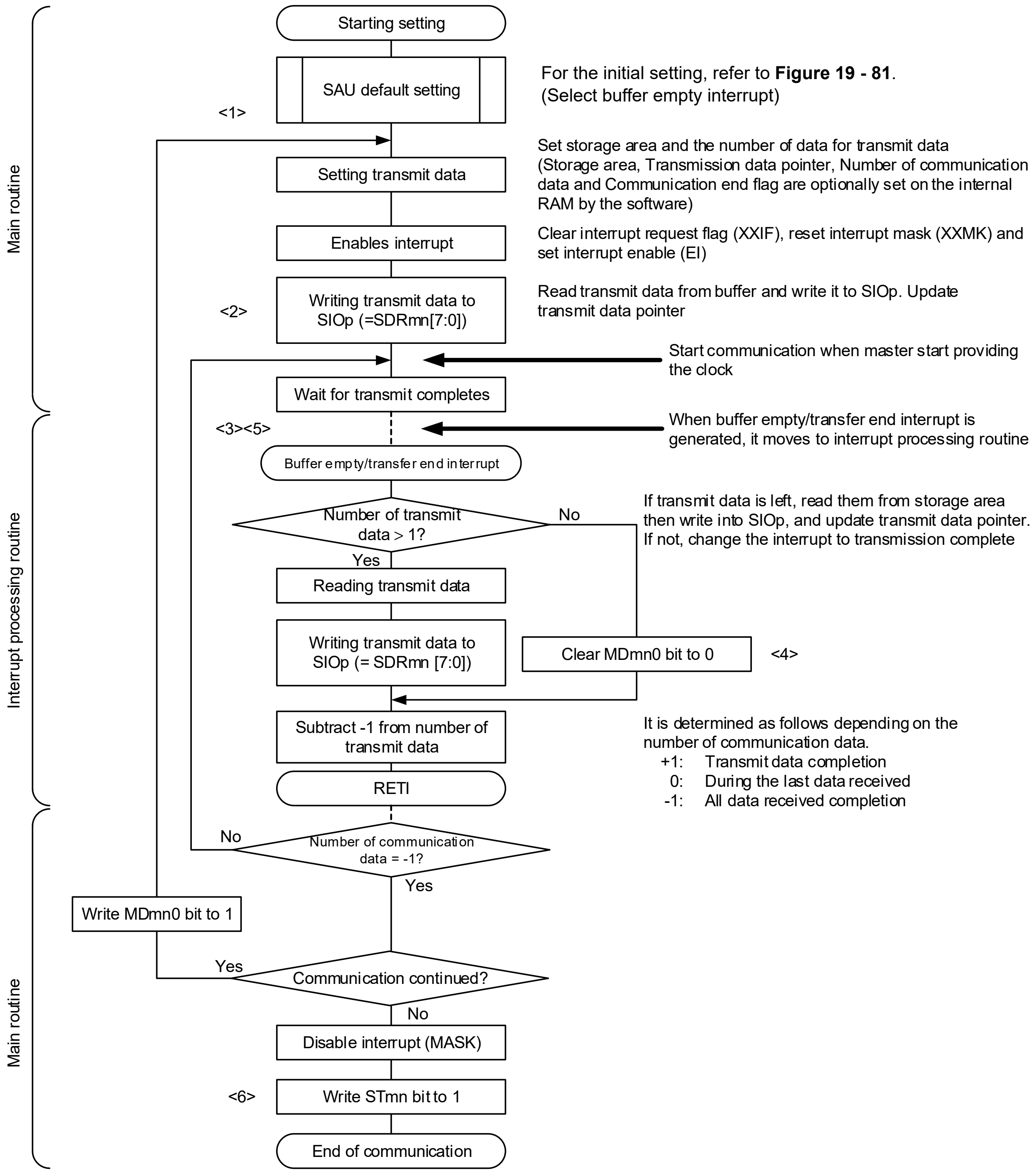


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit starts.

Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 19 - 87 Flowchart of Slave Transmission (in Continuous Transmission Mode)



Remark 1. <1> to <6> in the figure correspond to <1> to <6> in Figure 19 - 86 Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0).

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

19.6.2 Slave reception

Slave reception is that the RL78 microcontroller receives data from another device in the state of a transfer clock being input from another device.

Slave select input function	CSI00
Target channel	Channel 0 of SAU0
Pins used	SCK00, SI00, $\overline{\text{SSI00}}$
Interrupt	INTCSI00
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)
Error detection flag	Overrun error detection flag (OVFmn) only
Transfer data length	7 or 8 bits
Transfer rate	Max. f_{MCK}/12 [Hz] Notes 1, 2
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data input starts from the start of the operation of the serial clock. • DAPmn = 1: Data input starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse
Data direction	MSB or LSB first
Slave select input function	Slave select input function operation selectable

Note 1. Because the external serial clock input to the SCK00 pin is sampled internally and used, the fastest transfer rate is f_{MCK}/12 [Hz].

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 33** or **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark 1. f_{MCK}: Operation clock frequency of target channel

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

(1) Register setting

Figure 19 - 88 Example of Contents of Registers for Slave Reception of Slave Select Input Function (CSI00) (1/2)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **19.3 Registers Controlling Serial Array Unit**)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 Baud rate setting							0	Receive data (SIOp)							

(d) Serial output register m (SOm)... 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	0	0	0	0	0	CKOm2 ×	CKOm1 ×	CKOm0 ×	0	0	0	0	0	SOm2 ×	SOm1 ×	SOm0 ×

(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	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	SOEm1 ×	SOEm0 ×

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Remark 2. (bold box): Setting is fixed in the Simplified SPI (CSI) slave reception mode,
(shaded box): Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 19 - 89 Example of Contents of Registers for Slave Reception of Slave Select Input Function (CSI00) (2/2)

(f) Serial channel start register m (SSm)... Set only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 ×	SSm0 0/1

(g) Input switch control register (ISC)... $\overline{\text{SSI00}}$ input setting in CSI00 slave channel (channel 0 of unit 0).

	7	6	5	4	3	2	1	0
ISC	SSIE00 0/1	0	0	0	0	0	ISC1 0/1	ISC0 0/1

SSIE00:
0: Disables the input value of the $\overline{\text{SSI00}}$ pin
1: Enables the input value of the $\overline{\text{SSI00}}$ pin

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Remark 2. ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

Figure 19 - 90 Initial Setting Procedure for Slave Reception

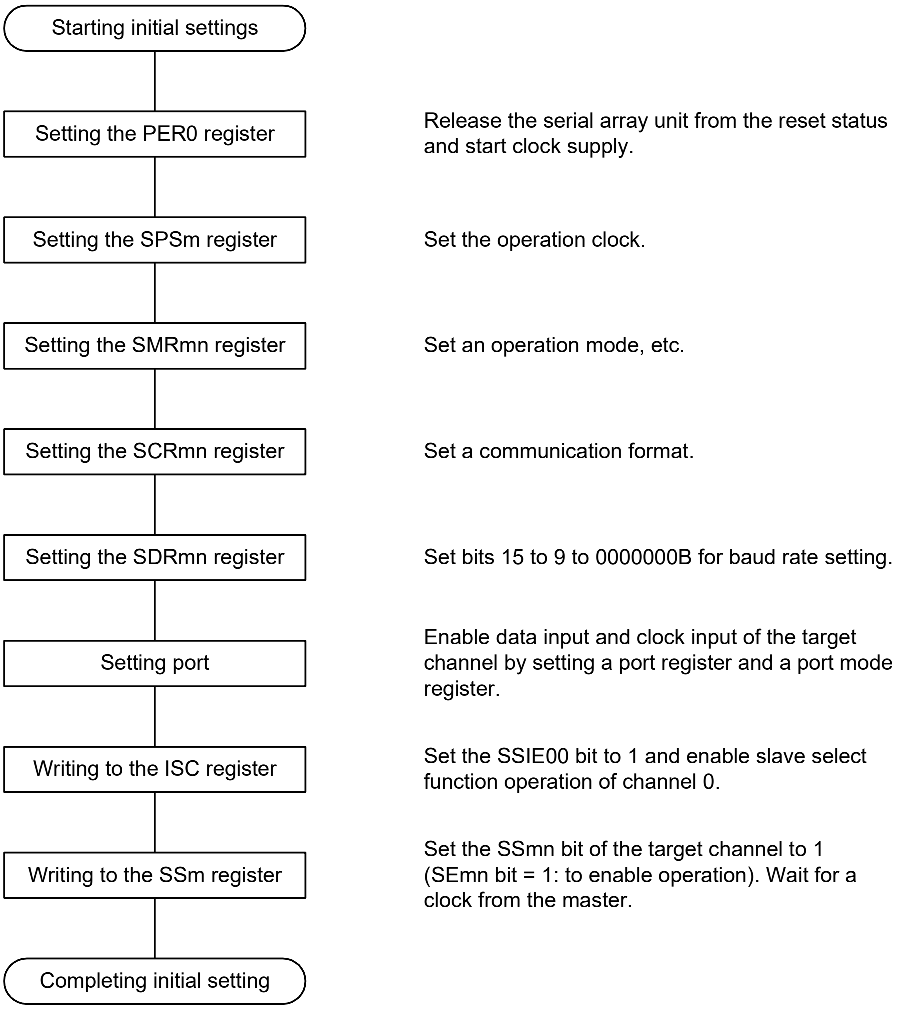
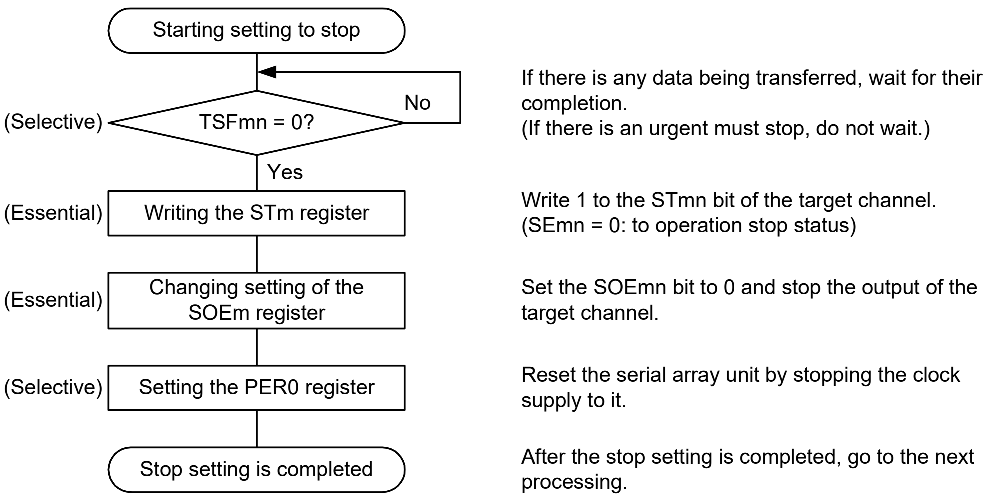
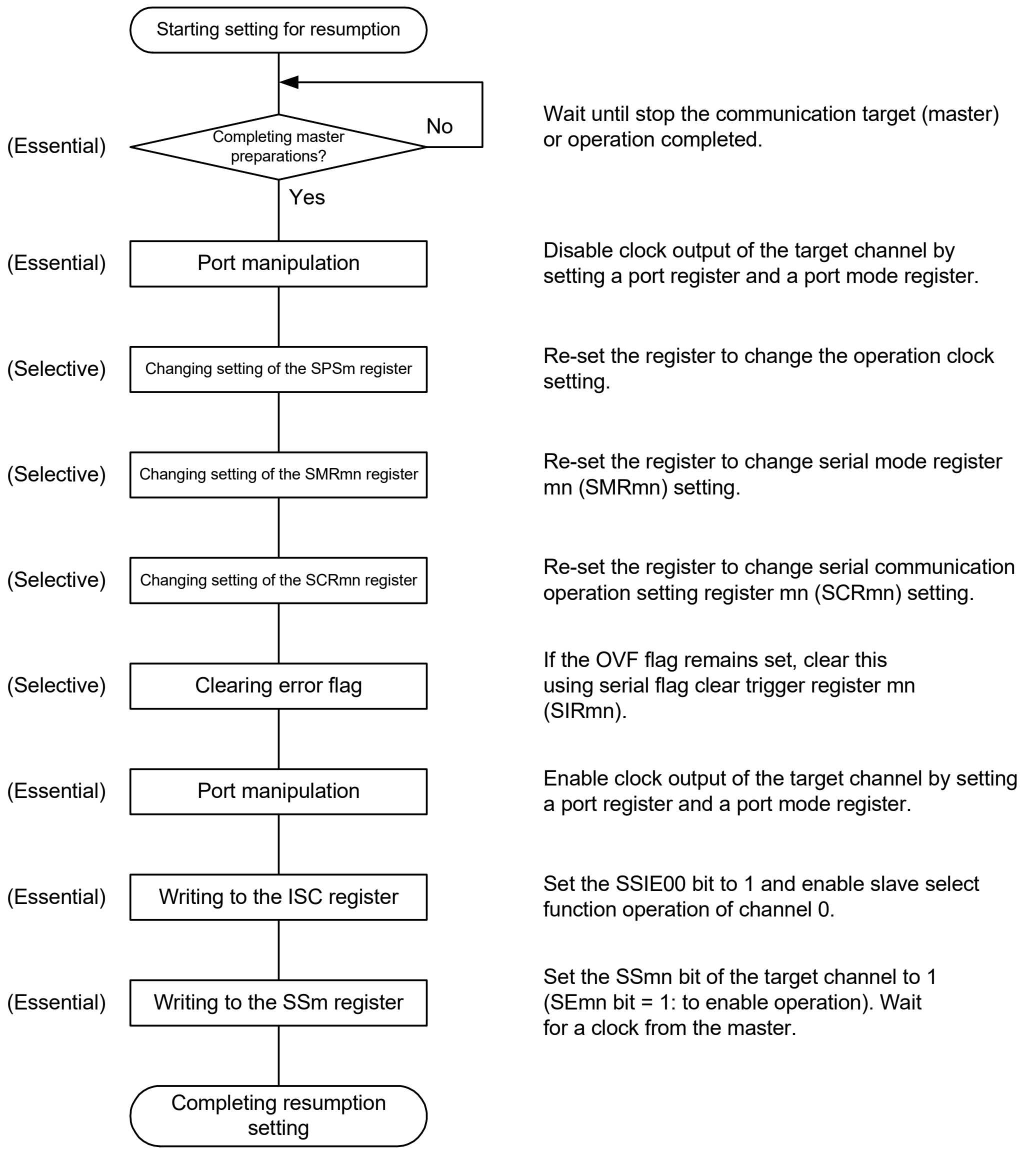


Figure 19 - 91 Procedure for Stopping Slave Reception



Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

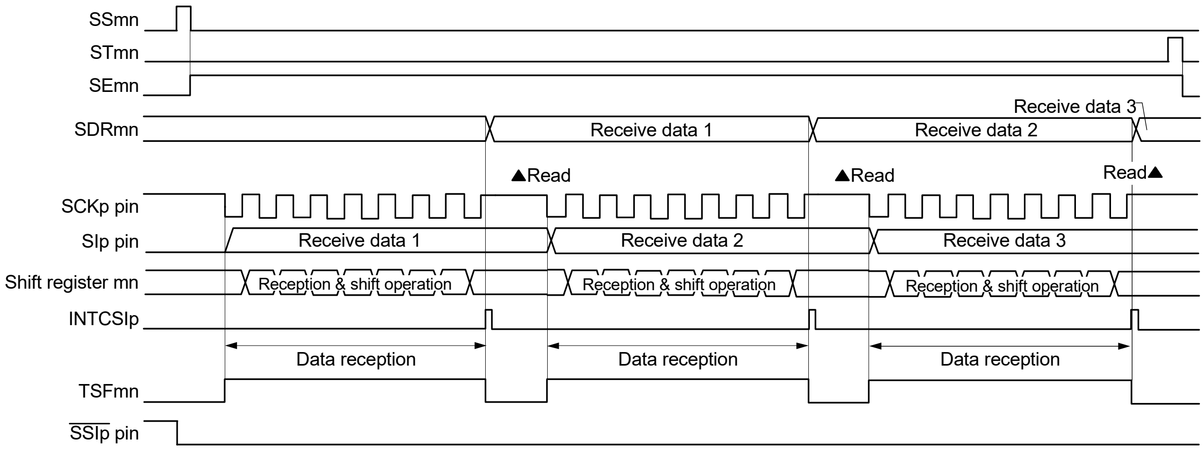
Figure 19 - 92 Procedure for Resuming Slave Reception



Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

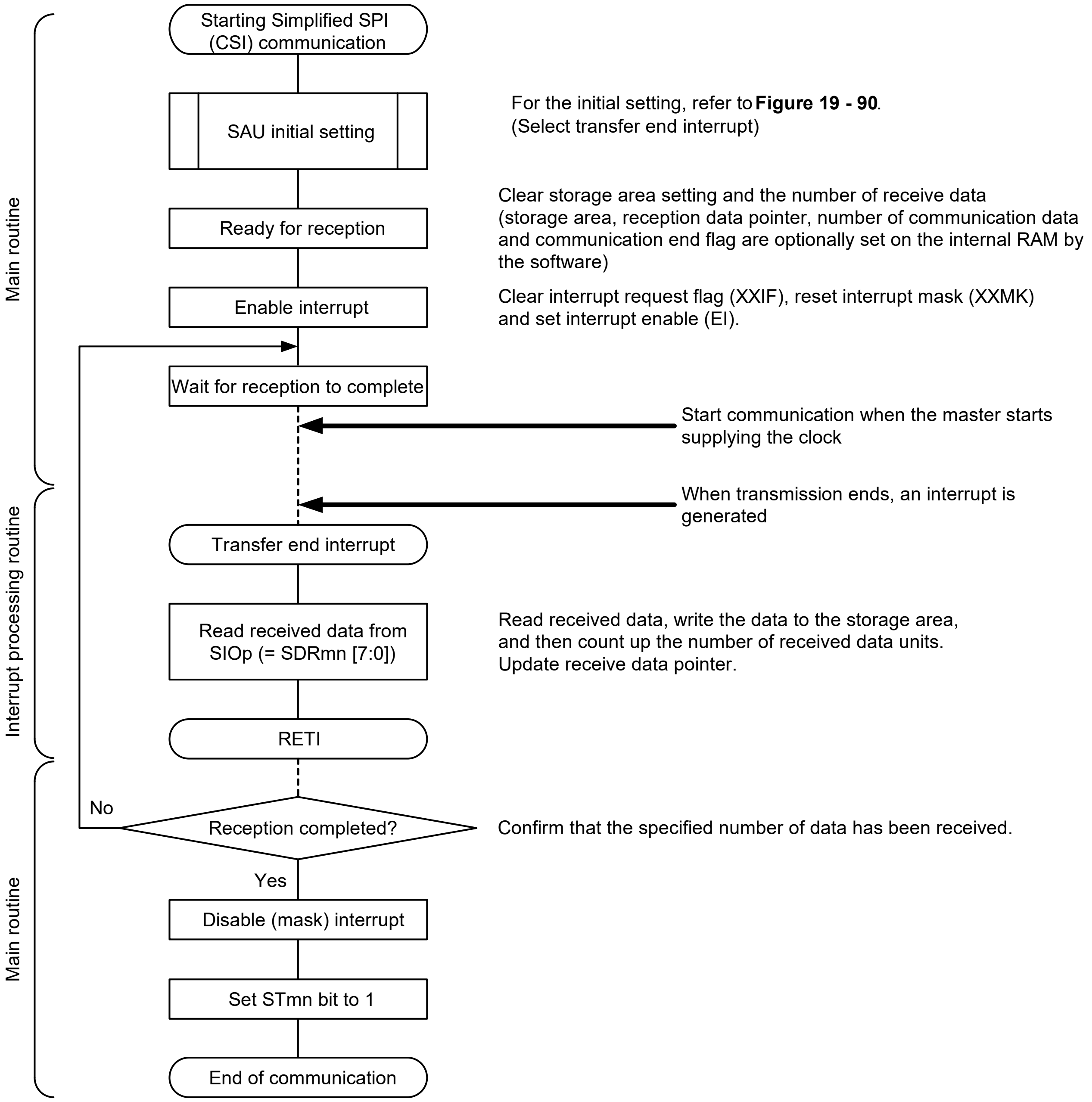
(3) Processing details (in single-reception mode)

Figure 19 - 93 Timing Chart of Slave Reception (in Single-Reception Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 19 - 94 Flowchart of Slave Reception (in Single-Reception Mode)



19.6.3 Slave transmission/reception

Slave transmission/reception is that the RL78 microcontroller transmits/receives data to/from another device in the state of a transfer clock being input from another device.

Slave select input function	CSI00
Target channel	Channel 0 of SAU0
Pins used	SCK00, SI00, SO00, $\overline{SSI00}$
Interrupt	INTCSI00
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.
Error detection flag	Overrun error detection flag (OVFmn) only
Transfer data length	7 or 8 bits
Transfer rate	Max. f_{MCK}/12 [Hz] Notes 1, 2
Data phase	Selectable by the DAPmn bit of the SCRmn register • DAPmn = 0: Data I/O starts from the start of the operation of the serial clock. • DAPmn = 1: Data I/O starts half a clock before the start of the serial clock operation.
Clock phase	Selectable by the CKPmn bit of the SCRmn register • CKPmn = 0: Non-reverse • CKPmn = 1: Reverse
Data direction	MSB or LSB first
Slave select input function	Slave select input function operation selectable

Note 1. Because the external serial clock input to the SCK00 pin is sampled internally and used, the fastest transfer rate is f_{MCK}/12 [Hz].

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 33** or **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark 1. f_{MCK}: Operation clock frequency of target channel

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

(1) Register setting

Figure 19 - 95 Example of Contents of Registers for Slave Transmission/Reception of Slave Select Input Function (CSI00) (1/2)

(a) Serial mode register mn (SMRmn)

SMRmn	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	CKSmn 0/1	CCSmn 1	0	0	0	0	0	STSmn 0	0	SISmn0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation clock (fMCK) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

SCRmn	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	TXEmn 1	RXEmn 1	DAPmn 0/1	CKPmn 0/1	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0/1	0	SLCmn1 0	SLCmn0 0	0	1	DLSmn1 1	DLSmn0 0/1

Selection of the data and clock phase (For details about the setting, see **19.3 Registers Controlling Serial Array Unit**)

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length
0: 7-bit data length
1: 8-bit data length

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

SDRmn	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0000000 Baud rate setting							0	Transmit data setting/receive data register							

SIOp (bits 7 to 0)

(d) Serial output register m (SOm)... Set only the bits of the target channel.

SOm	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	CKOm2 ×	CKOm1 ×	CKOm0 ×	0	0	0	0	0	SOm2 ×	SOm1 ×	SOm0 0/1

(e) Serial output enable register m (SOEm)... Set only the bits of the target channel to 1.

SOEm	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	SOEm1 ×	SOEm0 0/1

Caution **Be sure to set transmit data to the SIOp register before clock supply from the master starts.**

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Remark 2. ■: Setting is fixed in the Simplified SPI (CSI) slave transmission/reception mode

▒: Setting disabled (set to the initial value)

×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)

0/1: Set to 0 or 1 depending on the usage of the user

Figure 19 - 96 Example of Contents of Registers for Slave Transmission/Reception of Slave Select Input Function (CSI00) (2/2)

(f) Serial channel start register m (SSm)... Set only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 ×	SSm0 0/1

(g) Input switch control register (ISC)... $\overline{SSI00}$ input setting in CSI00 slave channel (channel 0 of unit 0).

	7	6	5	4	3	2	1	0
ISC	SSIE00 0/1	0	0	0	0	0	ISC1 0/1	ISC0 0/1

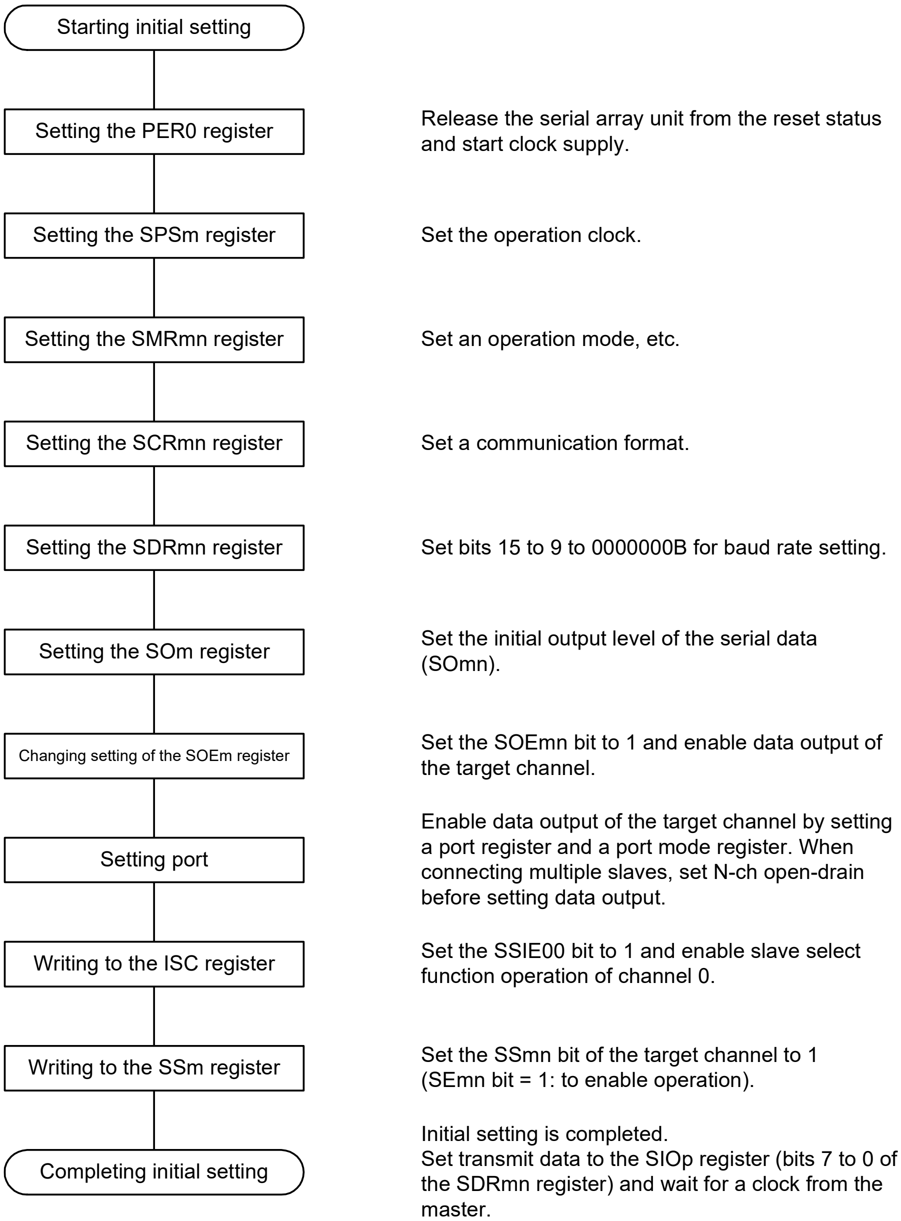
SSIE00:
0: Disables the input value of the $\overline{SSI00}$ pin
1: Enables the input value of the $\overline{SSI00}$ pin

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Remark 2. ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

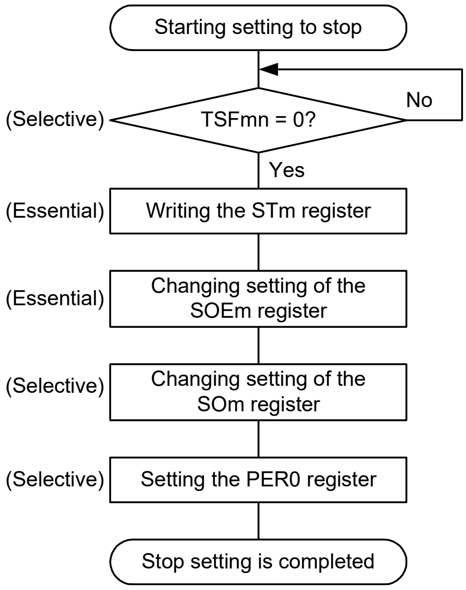
Figure 19 - 97 Initial Setting Procedure for Slave Transmission/Reception



Caution **Be sure to set transmit data to the SIOp register before clock supply from the master starts.**

Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 19 - 98 Procedure for Stopping Slave Transmission/Reception



If there is any data being transferred, wait for their completion.
(If there is an urgent must stop, do not wait.)

Write 1 to the STmn bit of the target channel.
(SEmn = 0: to operation stop status)

Set the SOEmn bit to 0 and stop the output of the target channel.

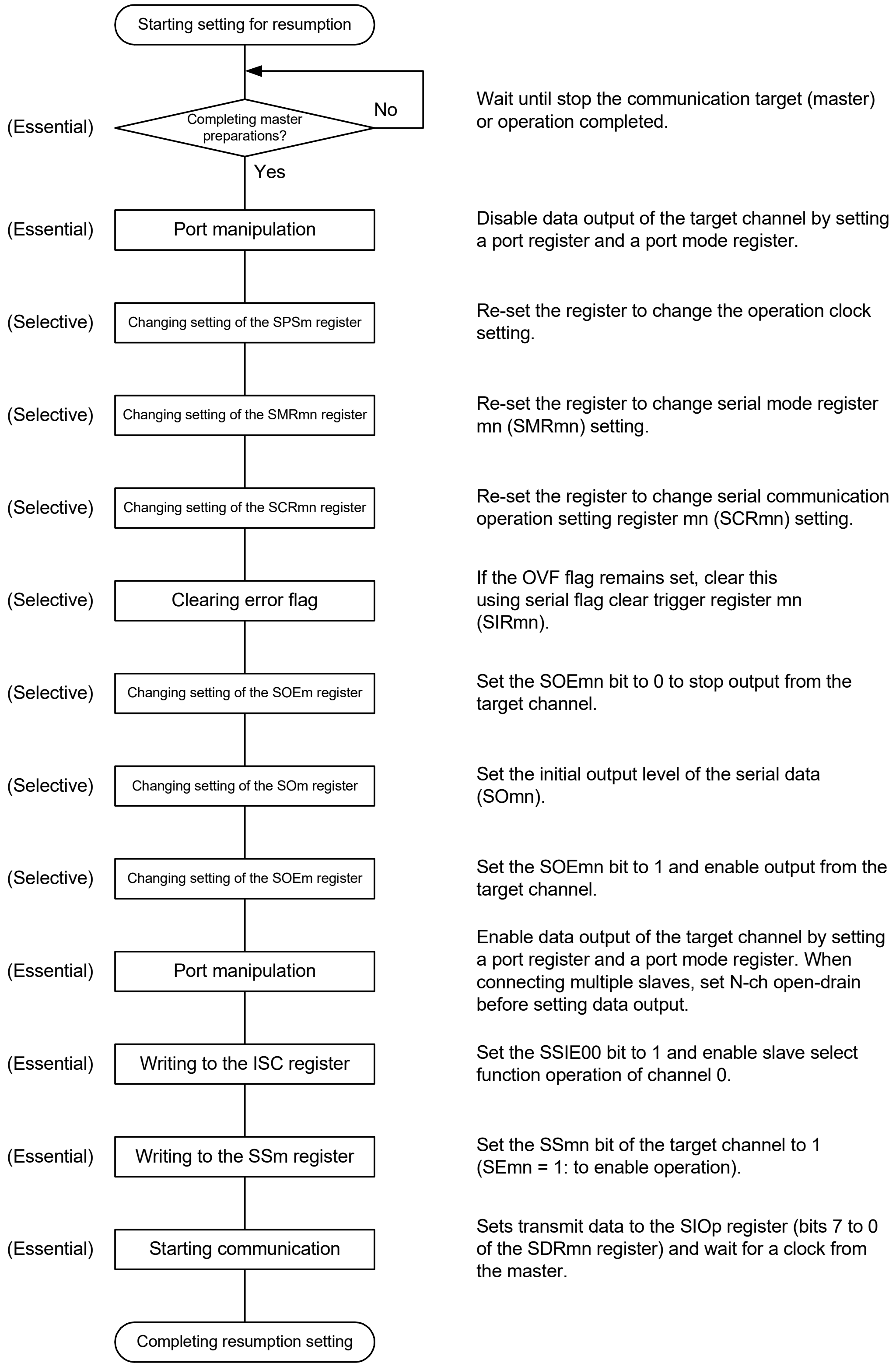
The levels of the serial data (SOmn) on the target channel can be changed if necessitated by an emergency.

Reset the serial array unit by stopping the clock supply to it.

After the stop setting is completed, go to the next processing.

Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 19 - 99 Procedure for Resuming Slave Transmission/Reception

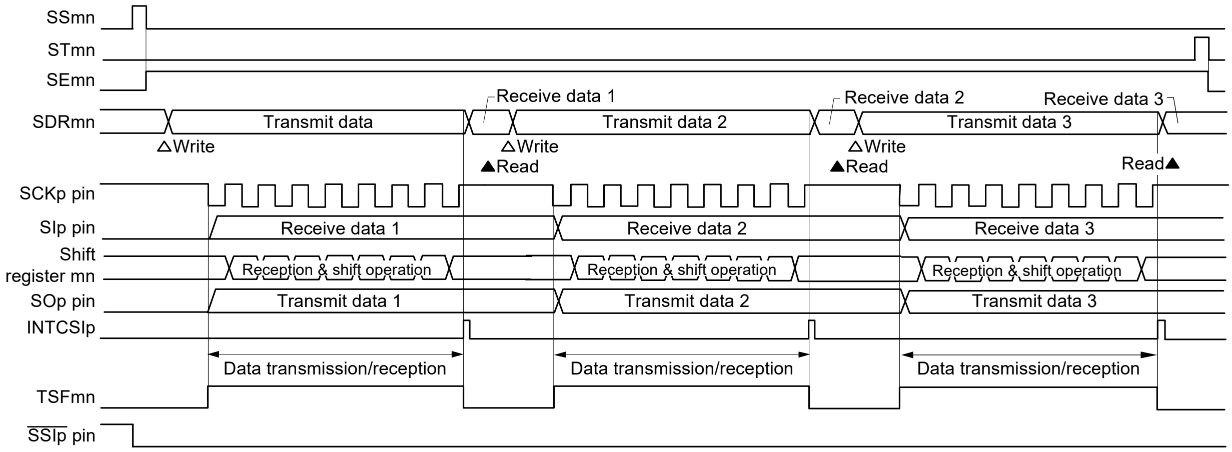


Caution 1. Be sure to set transmit data to the SIOp register before clock supply from the master starts.

Caution 2. If PER0 is rewritten to stop clock supply while the master transmission is stopped, wait until the transmission target (master) stops or transmission finishes, and then perform initialization instead of restarting the transmission.

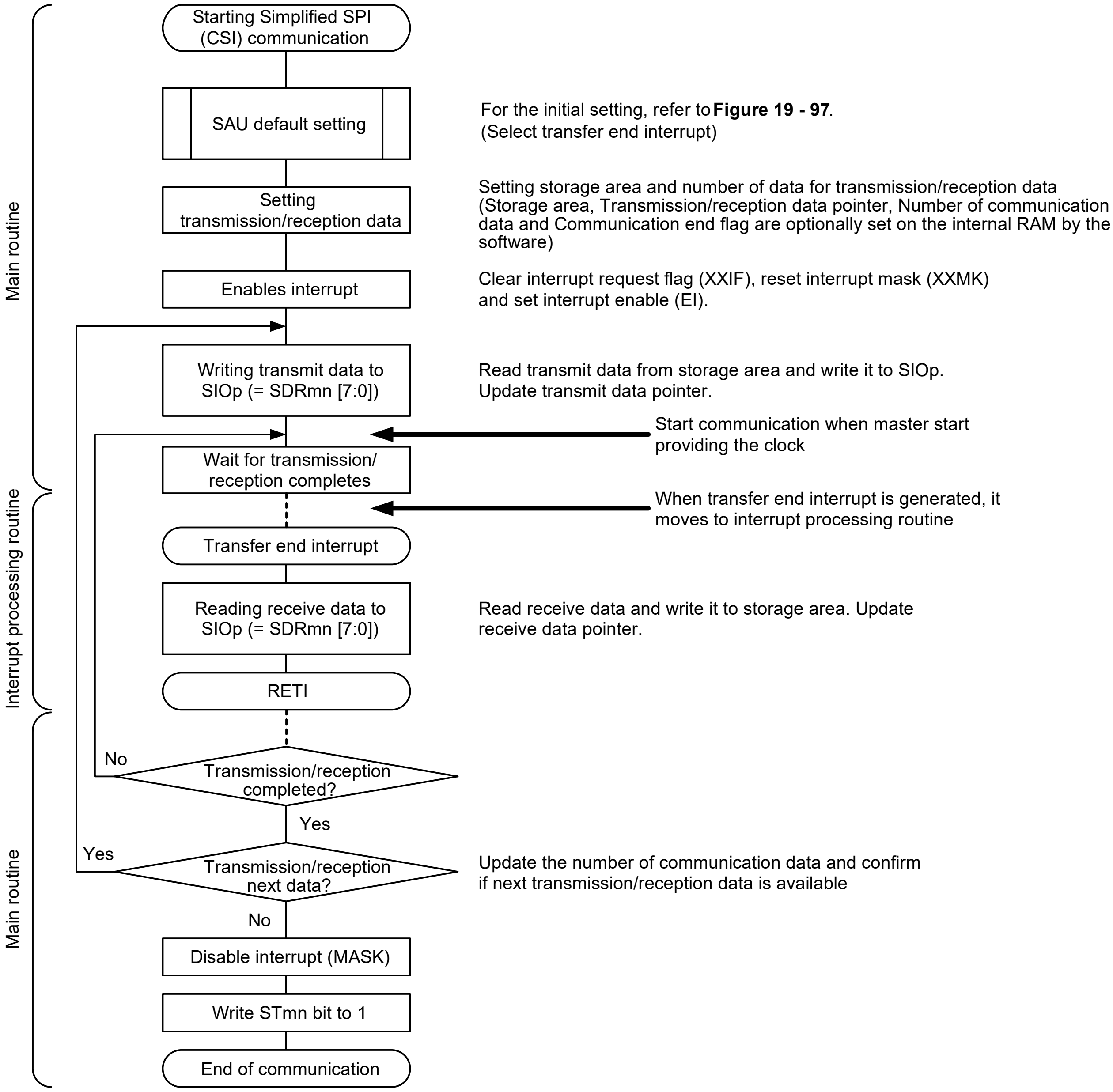
(3) Processing details (in single-transmission/reception mode)

Figure 19 - 100 Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 19 - 101 Flowchart of Slave Transmission/Reception (in Single- Transmission/Reception Mode)

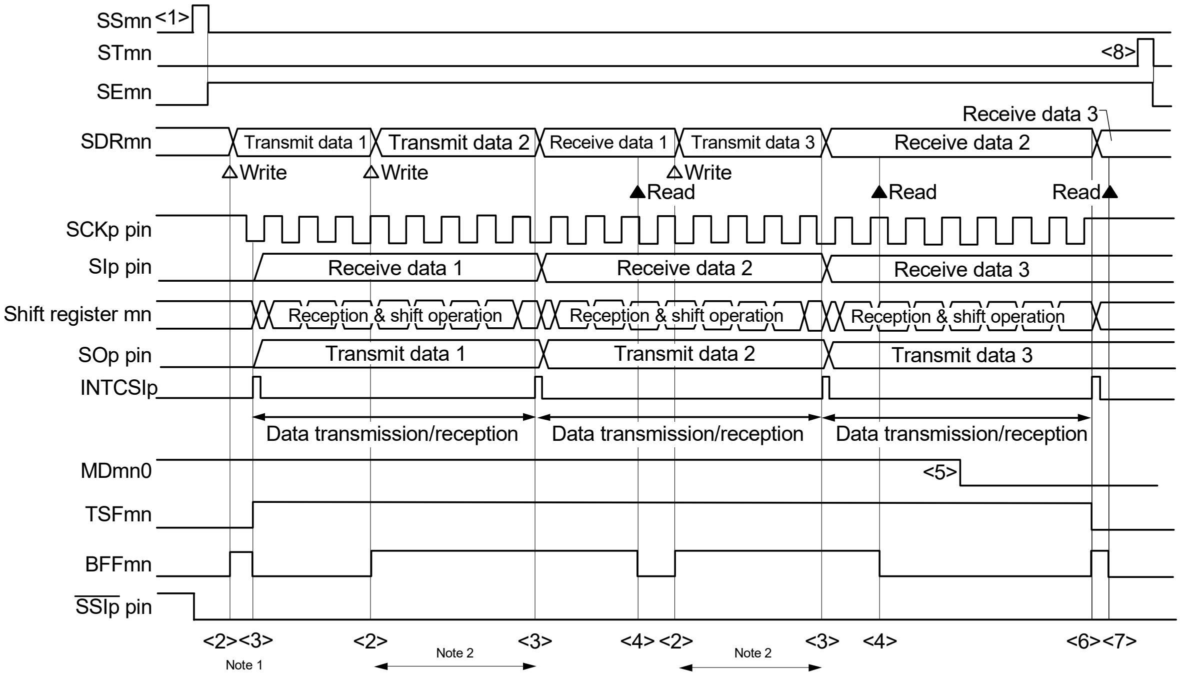


Caution Be sure to set transmit data to the SIOp register before clock supply from the master starts.

Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

(4) Processing details (in continuous transmission/reception mode)

Figure 19 - 102 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.

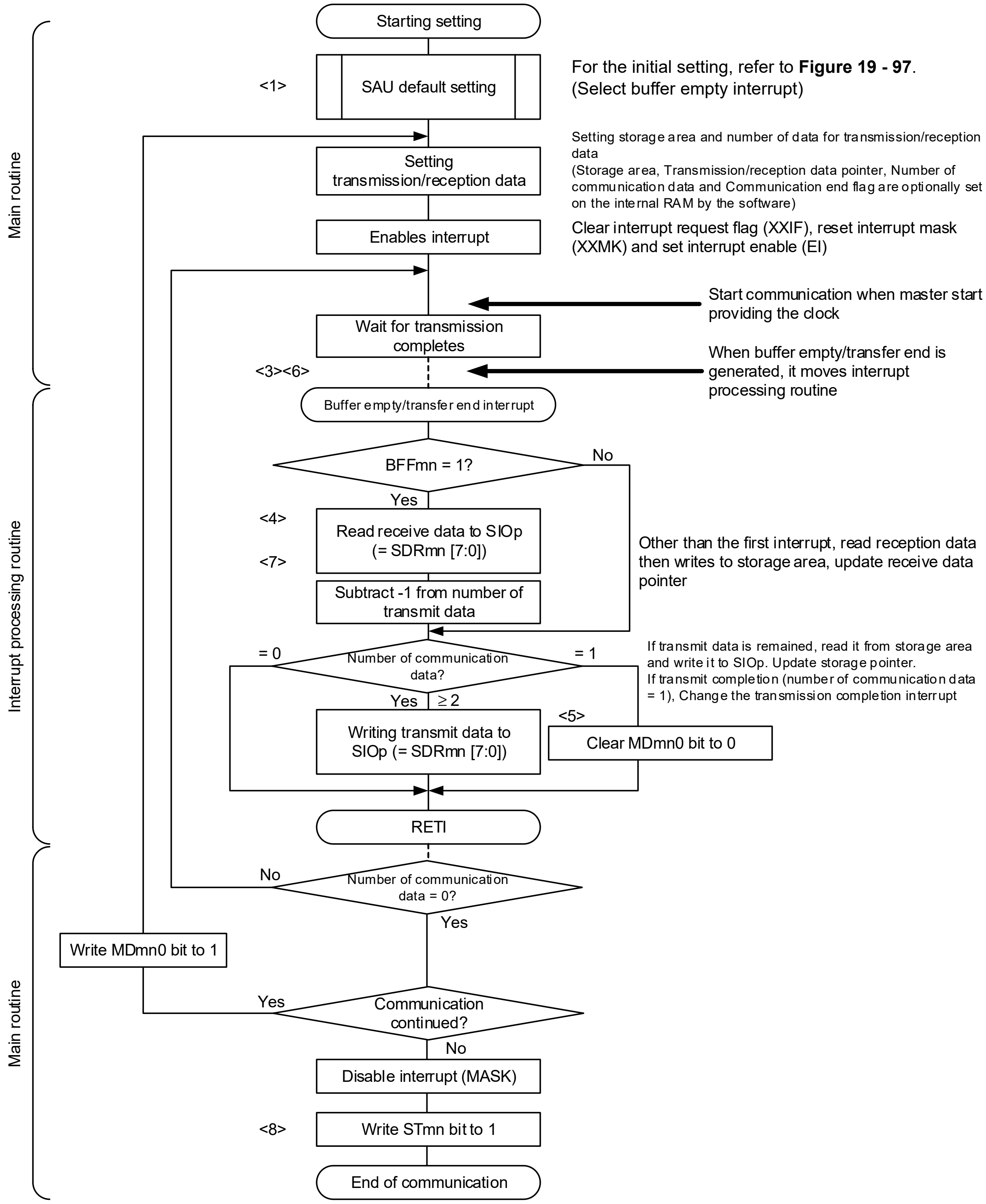
Note 2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

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 starts, 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 19 - 103 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode).

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Figure 19 - 103 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)



Caution **Be sure to set transmit data to the SIOp register before clock supply from the master starts.**

Remark 1. <1> to <8> in the figure correspond to <1> to <8> in Figure 19 - 102 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0).

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

19.6.4 Calculating transfer clock frequency

The transfer clock frequency for slave select input function (CSI00) communication can be calculated by the following expressions.

(1) Slave

Transfer clock frequency = {Frequency of serial clock (SCK) supplied by master} Note [Hz]

Note The permissible maximum transfer clock frequency is $f_{MCK}/12$.

Remark m: Unit number (m = 0), n: Channel number (n = 0), p: CSI number (p = 00)

Table 19 - 4 Selection of Operation Clock for Slave Select Input Function

SMRmn Register	SPSm Register								Operation Clock (f_{MCK}) Note	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 32 MHz
0	×	×	×	×	0	0	0	0	f_{CLK}	32 MHz
	×	×	×	×	0	0	0	1	$f_{CLK}/2$	16 MHz
	×	×	×	×	0	0	1	0	$f_{CLK}/2^2$	8 MHz
	×	×	×	×	0	0	1	1	$f_{CLK}/2^3$	4 MHz
	×	×	×	×	0	1	0	0	$f_{CLK}/2^4$	2 MHz
	×	×	×	×	0	1	0	1	$f_{CLK}/2^5$	1 MHz
	×	×	×	×	0	1	1	0	$f_{CLK}/2^6$	500 kHz
	×	×	×	×	0	1	1	1	$f_{CLK}/2^7$	250 kHz
	×	×	×	×	1	0	0	0	$f_{CLK}/2^8$	125 kHz
	×	×	×	×	1	0	0	1	$f_{CLK}/2^9$	62.5 kHz
	×	×	×	×	1	0	1	0	$f_{CLK}/2^{10}$	31.25 kHz
	×	×	×	×	1	0	1	1	$f_{CLK}/2^{11}$	15.63 kHz
	×	×	×	×	1	1	0	0	$f_{CLK}/2^{12}$	7.81 kHz
	×	×	×	×	1	1	0	1	$f_{CLK}/2^{13}$	3.91 kHz
	×	×	×	×	1	1	1	0	$f_{CLK}/2^{14}$	1.95 kHz
	×	×	×	×	1	1	1	1	$f_{CLK}/2^{15}$	977 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Remark 1. ×: Don't care

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

19.6.5 Procedure for processing errors that occurred during slave select input function communication

The procedure for processing errors that occurred during slave select input function communication is described in Figure 19 - 104.

Figure 19 - 104 Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

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

19.7 Operation of UART (UART0, UART1) Communication

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. By using these two communication lines, each data frame, which consist of a start bit, data, parity bit, and stop bit, is transferred asynchronously (using the internal baud rate) between the microcontroller and the other communication party. Full-duplex UART communication can be performed by using a channel dedicated to transmission (even-numbered channel) and a channel dedicated to reception (odd-numbered channel). The LIN-bus can also be used by using UART0, timer array unit 0 (channel 3), and an external interrupt (INTP0).

[Data transmission/reception]

- Data length of 7, 8, or 9 bits [Note]
- Select the MSB/LSB first
- Level setting of transmit/receive data (selecting whether to reverse the level)
- Parity bit appending and parity check functions
- Stop bit appending, stop bit check function

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

UART0 supports the SNOOZE mode. When RxD input is detected while in the STOP mode, the SNOOZE mode makes data reception that does not require the CPU possible.

UART0 (channels 0 and 1 of unit 0) supports the LIN bus.

[LIN-bus functions]

- Wakeup signal detection
- Break field (BF) detection
- Sync field measurement, baud rate calculation

} Using the external interrupt (INTP0) and timer array unit 0 (channel 3)

Note The 9-bit data length can only be selected when using UART0.

UART0 uses channels 0 and 1 of SAU0.
UART1 uses channels 2 and 3 of SAU0.

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^2C
0	0	CSI00 (supporting slave select input function)	UART0 (supporting LIN-bus)	IIC00
	1	CSI01		IIC01
	2	—	UART1	—
	3	—		—

Select any function for each channel. Only use as 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 CSI01.

Caution When using a serial array unit for UART, both the transmitter side (even-numbered channel) and the receiver side (odd-numbered channel) can only be used for UART.

UART performs the following four types of communication operations.
- UART transmission (See **19.7.1**.)
- UART reception (See **19.7.2**.)
- LIN transmission (UART0 only) (See **19.8.1**.)
- LIN reception (UART0 only) (See **19.8.2**.)

19.7.1 UART transmission

UART transmission is an operation to transmit data from the RL78 microcontroller to another device asynchronously (start-stop synchronization).
Of two channels used for UART, the even channel is used for UART transmission.

UART	UART0	UART1
Target channel	Channel 0 of SAU0	Channel 2 of SAU0
Pins used	TxD0	TxD1
Interrupt	INTST0	INTST1
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	None	
Transfer data length	7, 8, or 9 bits Note 1	
Transfer rate	Max. $f_{MCK}/12$ [bps] (SDRmn [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps] Note 2	
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)	
Parity bit	The following selectable • No parity bit • Appending 0 parity • Appending even parity • Appending odd parity	
Stop bit	The following selectable • Appending 1 bit • Appending 2 bits	
Data direction	MSB or LSB first	

Note 1. The 9-bit data length can only be selected when using UART0.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 33** or **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 2), mn = 00, 02

(1) Register setting

Figure 19 - 105 Example of Contents of Registers for UART Transmission of UART (UART0, UART1) (1/2)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	0	0	0	1	0	0	MDmn2 0	MDmn1 1	MDmn0 0/1

Operation clock (fMCK) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

Interrupt source of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(b) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0	CKPmn 0	0	EOCmn 0	PTCmn1 0/1	PTCmn0 0/1	DIRmn 0/1	0	SLCmn1 0/1	SLCmn0 0/1	0	1	DLSmn1 0/1	DLSmn0 0/1

Setting of parity bit
00B: No parity
01B: Appending 0 parity
10B: Appending Even parity
11B: Appending Odd parity

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of stop bit
01B: Appending 1 bit
10B: Appending 2 bits

(c) Serial data register mn (SDRmn) (lower 8 bits: TXDq)

	15–9	8	7–0
SDRmn	Baud rate setting	0 Note	Transmit data setting (TXDq)

(d) Serial output level register m (SOLm)... Set only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOLm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOLm2 0/1	0	SOLm0 0/1

0: Non-reverse (normal) transmission
1: Reverse transmission

Note When performing 9-bit communication, bits 0 to 8 of the SDRm0 register are used to specify the transmission data.
The 9-bit data length can only be selected when using UART0.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 2), q: UART number (q = 0, 1), mn = 00, 02

Remark 2. ■: Setting is fixed in the UART transmission mode,
▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 19 - 106 Example of Contents of Registers for UART Transmission of UART (UART0, UART1) (2/2)

(e) Serial output register m (SOm)... Set only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	CKOm2 ×	CKOm1 ×	CKOm0 ×	0	0	0	0	0	SOm2 0/1 Note	SOm1 ×	SOm0 0/1 Note

0: Serial data output value is “0”
1: Serial data output value is “1”

(f) Serial output enable register m (SOEm)... Set only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 0/1	SOEm1 ×	SOEm0 0/1

(g) Serial channel start register m (SSm)... Set only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

Note Before transmission starts, be sure to set this bit to 1 when the SOLmn bit of the target channel is 0, and clear this bit to 0 when the SOLmn bit of the target channel is 1. The value varies depending on the communication data during communication operation.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 2), q: UART number (q = 0, 1), mn = 00, 02

Remark 2. ▒: Setting 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 19 - 107 Initial Setting Procedure for UART Transmission

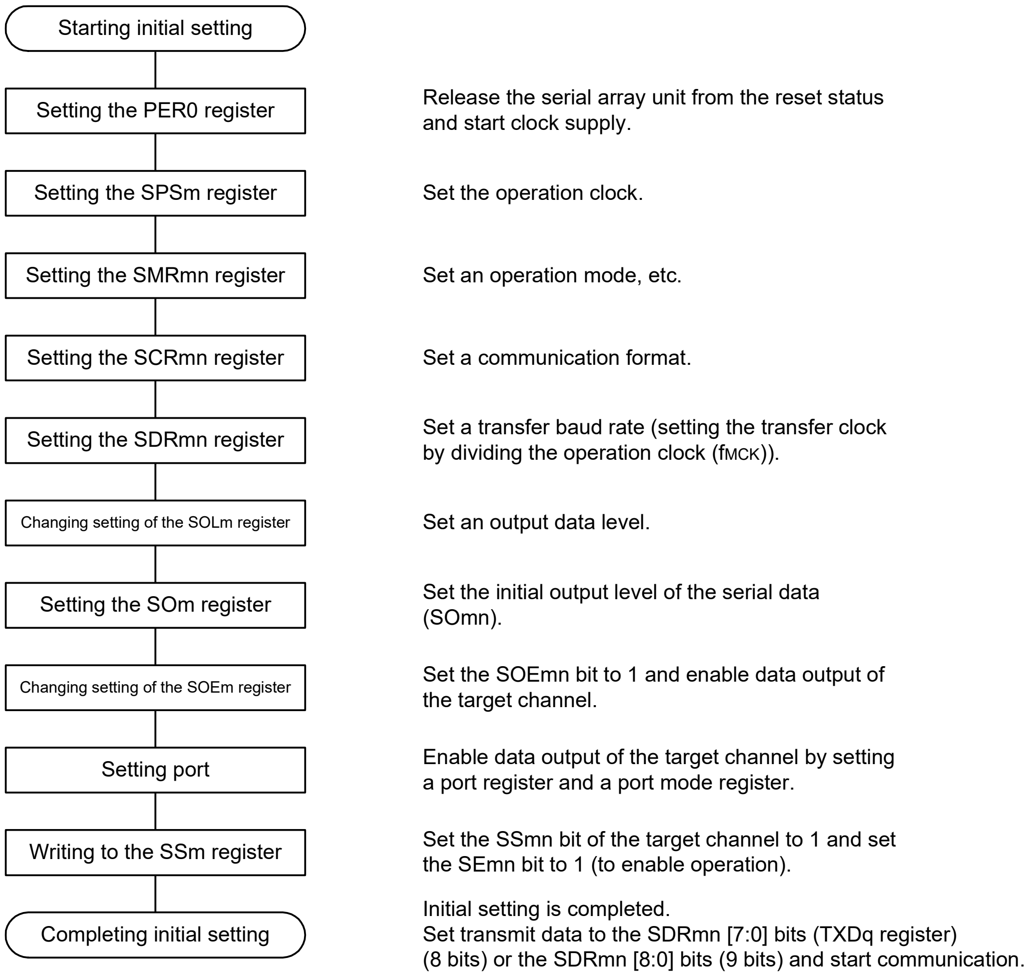


Figure 19 - 108 Procedure for Stopping UART Transmission

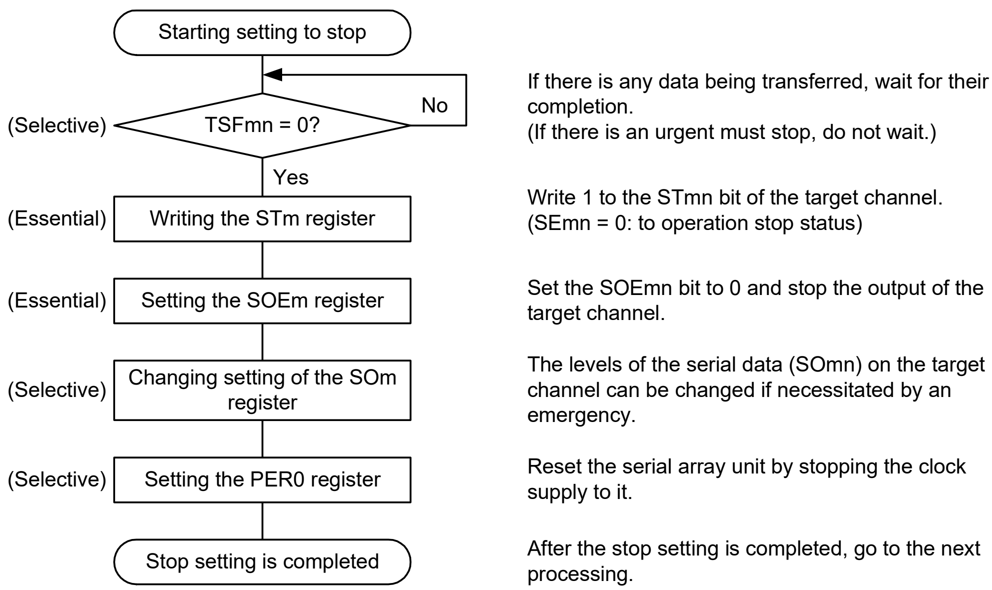
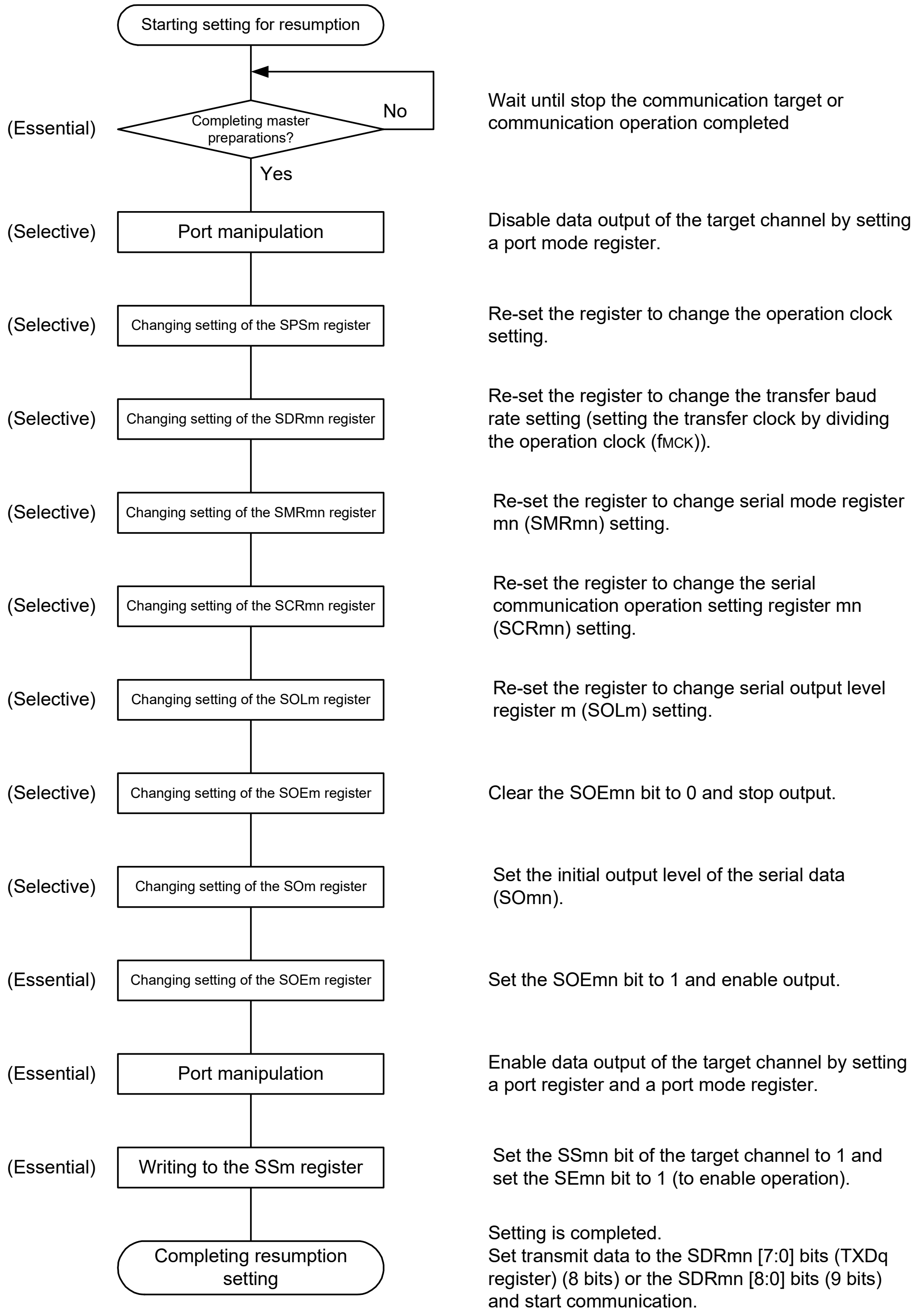


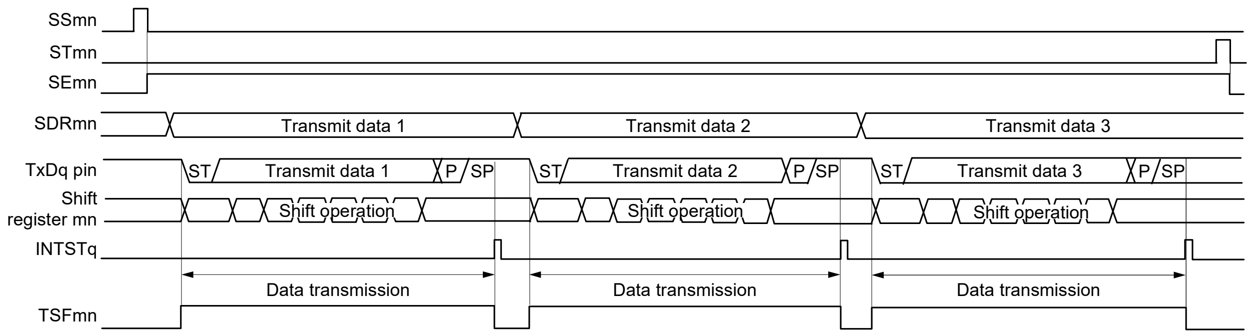
Figure 19 - 109 Procedure for Resuming UART Transmission



Remark If PER0 is rewritten to stop clock supply while the master transmission is stopped, wait until the transmission target stops or transmission finishes, and then perform initialization instead of restarting the transmission.

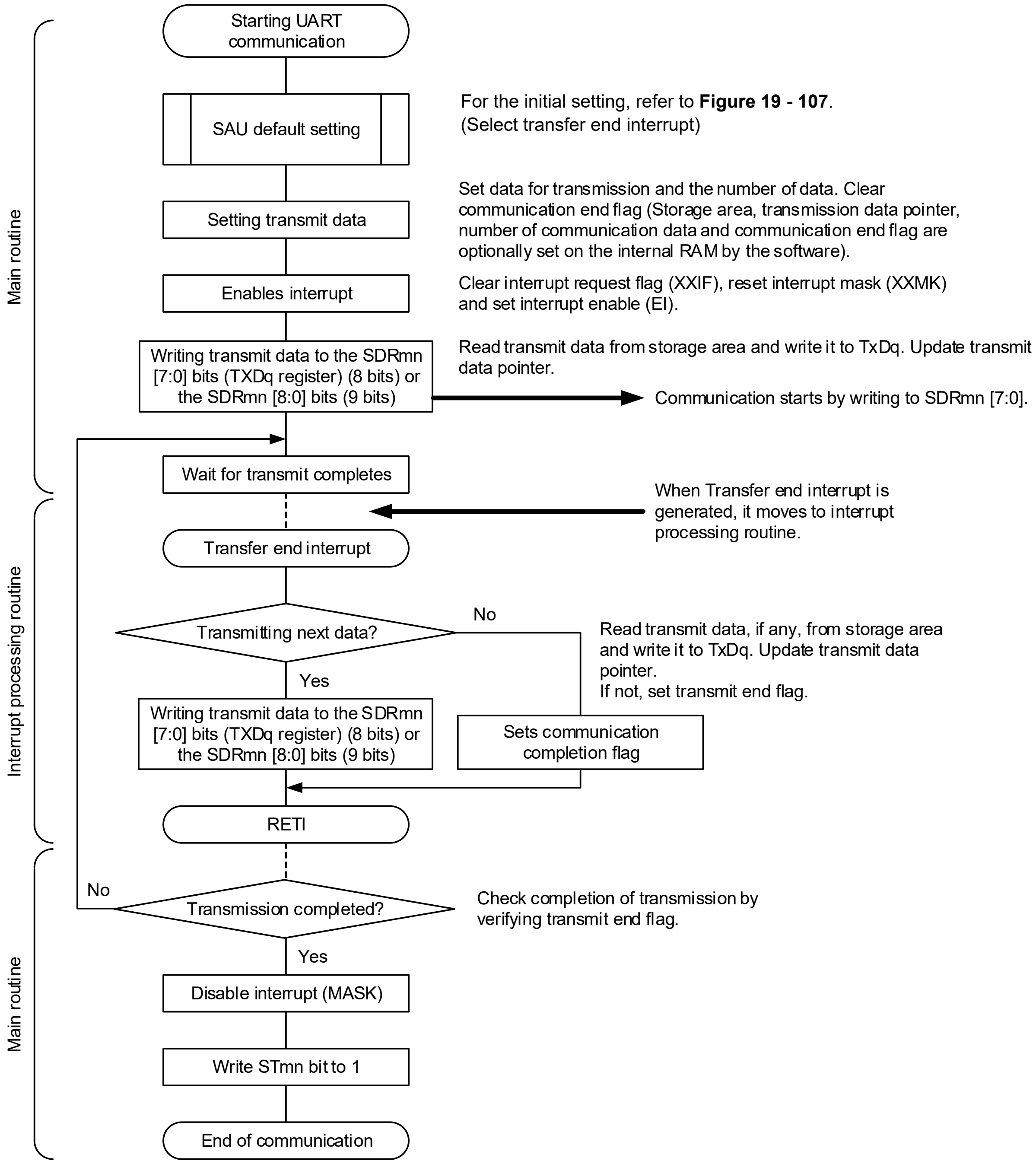
(3) Processing details (in single-transmission mode)

Figure 19 - 110 Timing Chart of UART Transmission (in Single-Transmission Mode)



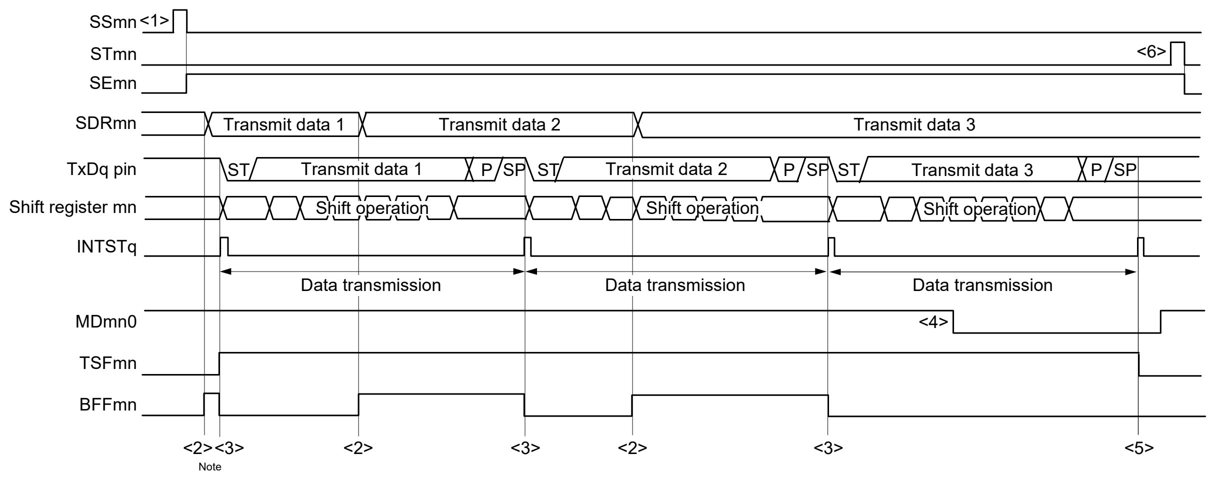
Remark m: Unit number (m = 0), n: Channel number (n = 0, 2), q: UART number (q = 0, 1), mn = 00, 02

Figure 19 - 111 Flowchart of UART Transmission (in Single-Transmission Mode)



(4) Processing details (in continuous transmission mode)

Figure 19 - 112 Timing Chart of UART Transmission (in Continuous Transmission Mode)

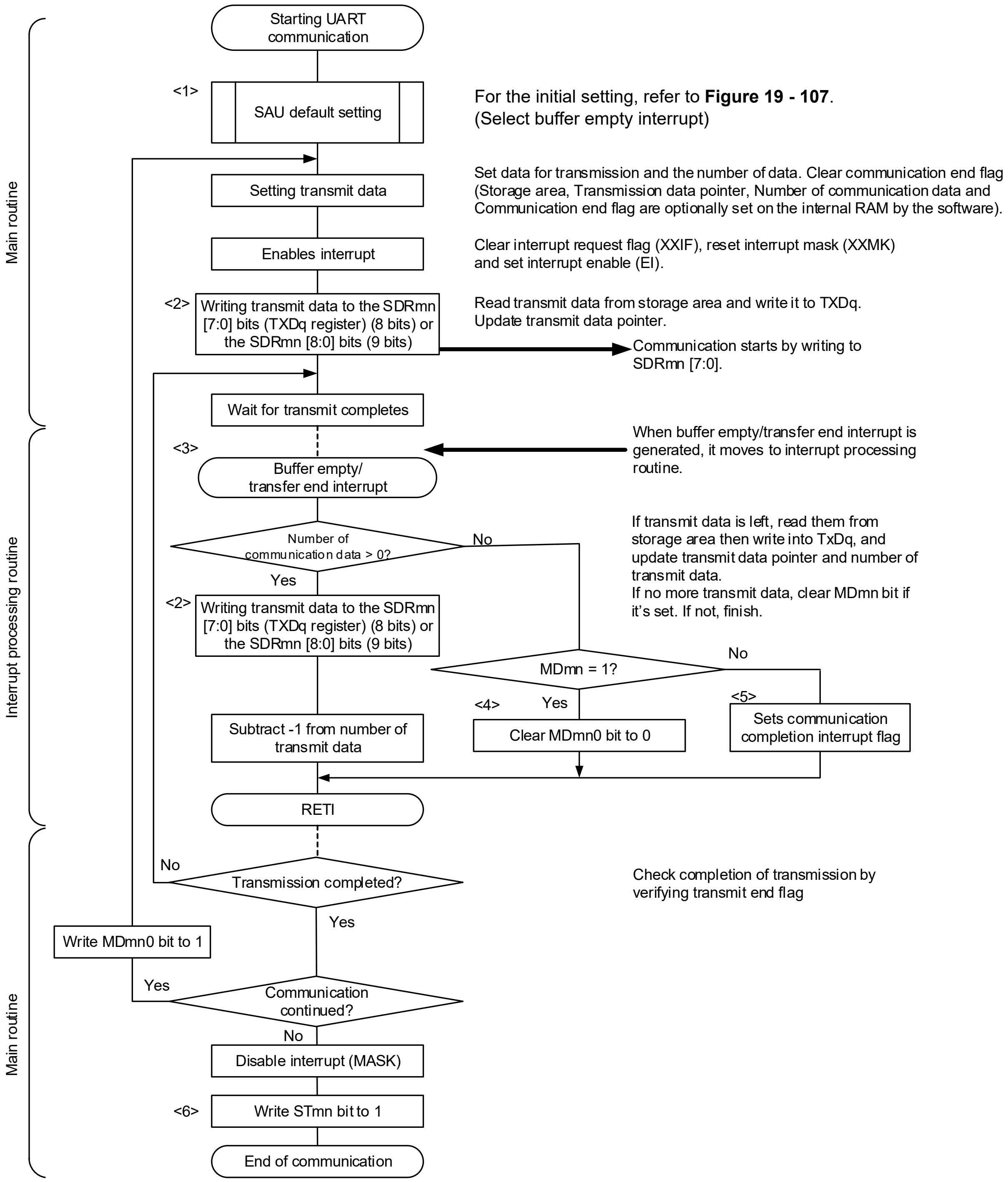


Note If transmit data is written to the SDRmn register while the BFFmn bit of serial status register mn (SSRmn) is 1 (valid data is stored in serial data register mn (SDRmn)), the transmit data is overwritten.

Caution The MDmn0 bit of serial mode register mn (SMRmn) can be rewritten even during operation. However, rewrite it before transfer of the last bit starts, so that it will be rewritten before the transfer end interrupt of the last transmit data.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 2), q: UART number (q = 0, 1), mn = 00, 02

Figure 19 - 113 Flowchart of UART Transmission (in Continuous Transmission Mode)



Remark <1> to <6> in the figure correspond to <1> to <6> in Figure 19 - 112 Timing Chart of UART Transmission (in Continuous Transmission Mode).

19.7.2 UART reception

UART reception is an operation wherein the RL78 microcontroller asynchronously receives data from another device (start-stop synchronization).

For UART reception, the odd-number channel of the two channels used for UART is used. The SMR register of both the odd- and even-numbered channels must be set.

UART	UART0	UART1
Target channel	Channel 1 of SAU0	Channel 3 of SAU0
Pins used	RxD0	RxD1
Interrupt	INTSR0	INTSR1
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error interrupt	INTSRE0	INTSRE1
Error detection flag	• Framing error detection flag (FEFmn) • Parity error detection flag (PEFmn) • Overrun error detection flag (OVFmn)	
Transfer data length	7, 8 or 9 bits Note 1	
Transfer rate Note 2	Max. f_{MCK}/12 [bps] (SDRmn [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]	
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)	
Parity bit	The following selectable • No parity bit (no parity check) • Appending 0 parity (no parity check) • Appending even parity • Appending odd parity	
Stop bit	Appending 1 bit	
Data direction	MSB or LSB first	

Note 1. The 9-bit data length can only be selected when using UART0.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 33** or **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark 1. f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

Remark 2. m: Unit number (m = 0), n: Channel number (n = 1, 3), mn = 01, 03

(1) Register setting

Figure 19 - 114 Example of Contents of Registers for UART Reception of UART (UART0, UART1) (1/2)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 1	0	SISmn0 0/1	1	0	0	MDmn2 0	MDmn1 1	MDmn0 0

Operation clock (f_{MCK}) of channel n
0: Prescaler output clock CKm0 set by the SPSm register
1: Prescaler output clock CKm1 set by the SPSm register

0: Forward (normal) reception
1: Reverse reception

Operation mode of channel n
0: Transfer end interrupt

(b) Serial mode register mr (SMRmr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmr	CKSmr 0/1	CCSmr 0	0	0	0	0	0	0	0	0	1	0	0	MDmr2 0	MDmr1 1	MDmr0 0/1

Same setting value as CKSmn bit

Operation mode of channel r
0: Transfer end interrupt
1: Buffer empty interrupt

(c) Serial communication operation setting register mn (SCRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0	CKPmn 0	0	EOCmn 1	PTCmn1 0/1	PTCmn0 0/1	DIRmn 0/1	0	SLCmn1 0	SLCmn0 1	0	1	DLSmn1 0/1	DLSmn0 0/1

Setting of parity bit
00B: No parity
01B: No parity judgment
10B: Appending Even parity
11B: Appending Odd parity

Selection of data transfer sequence
0: Inputs/outputs data with MSB first
1: Inputs/outputs data with LSB first

Setting of data length

(d) Serial data register mn (SDRmn) (lower 8 bits: RXDq)

	15–9	8	7–0
SDRmn	Baud rate setting	0 Note	Receive data register (RXDq)

Note When performing 9-bit communication, bits 0 to 8 of the SDRm1 register are used to specify the transmission data.
The 9-bit data length can only be selected when using UART0.

Caution For the UART reception, be sure to set the SMRmr register of channel r that is to be paired with channel n.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 1, 3), mn = 01, 03
r: Channel number (r = n – 1), q: UART number (q = 0, 1)

Remark 2. ■: Setting is fixed in the UART reception mode,
■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 19 - 115 Example of Contents of Registers for UART Reception of UART (UART0, UART1) (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	0	0	0	0	0	CKOm2 ×	CKOm1 ×	CKOm0 ×	0	0	0	0	0	SOm2 ×	SOm1 ×	SOm0 ×

(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	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	SOEm1 ×	SOEm0 ×

(g) Serial channel start register m (SSm)... Set only the bits of the target channel is 1.

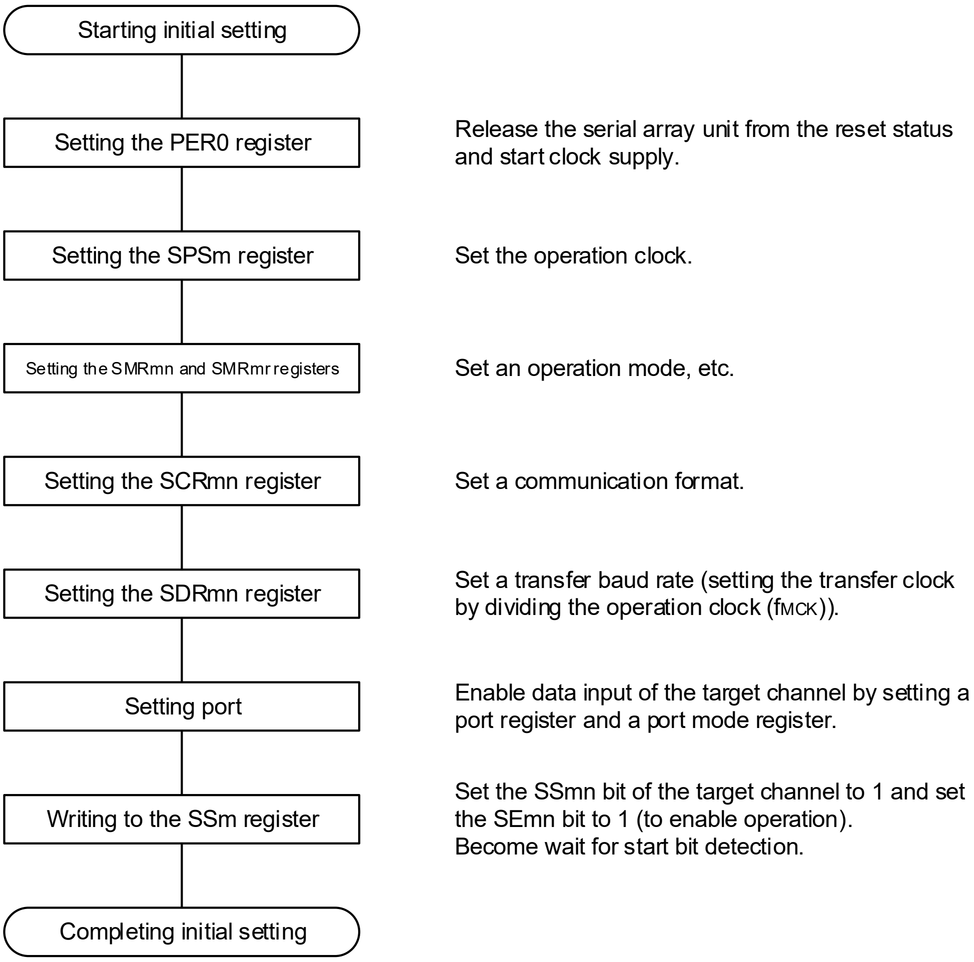
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 ×	SSm1 0/1	SSm0 ×

Remark 1. m: Unit number (m = 0)

Remark 2. ■: Setting 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 19 - 116 Initial Setting Procedure for UART Reception



Caution Set the RXEmn bit of SCRmn register to 1, and then set SSmn to 1 after 4 or more f_{MCK} clocks have elapsed.

Figure 19 - 117 Procedure for Stopping UART Reception

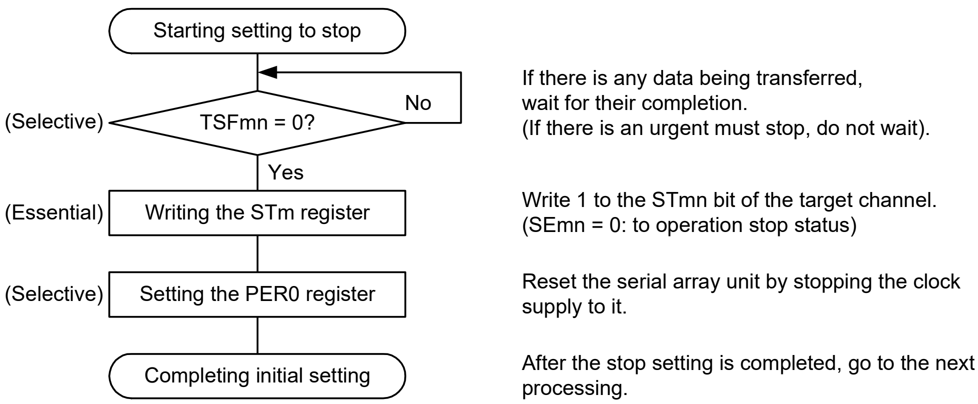
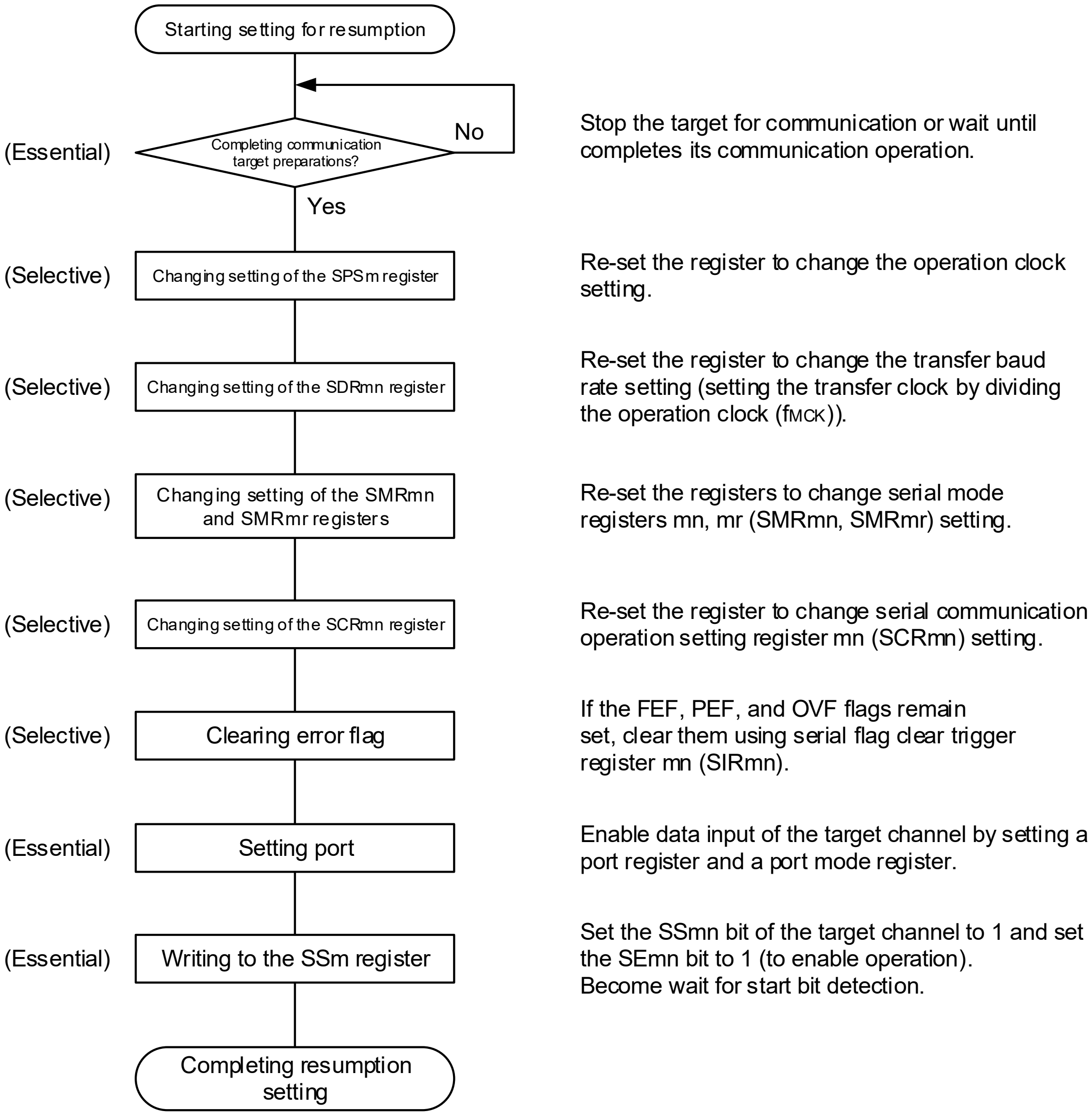


Figure 19 - 118 Procedure for Resuming UART Reception

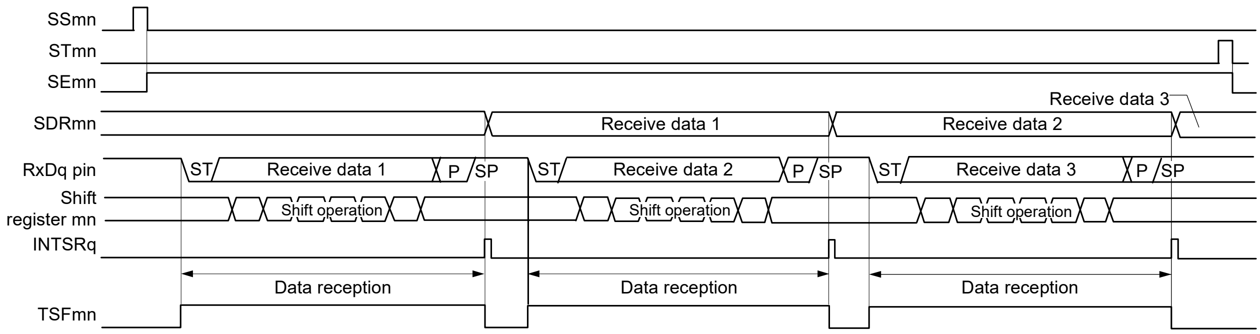


Caution **After is set RXEmn bit to 1 of SCRmn register, set the SSmn = 1 from an interval of at least four clocks of f_{MCK}.**

Remark If PER0 is rewritten to stop clock supply while the communication target is stopped, wait until the communication target stops or communication finishes, and then perform initialization instead of restarting the communication.

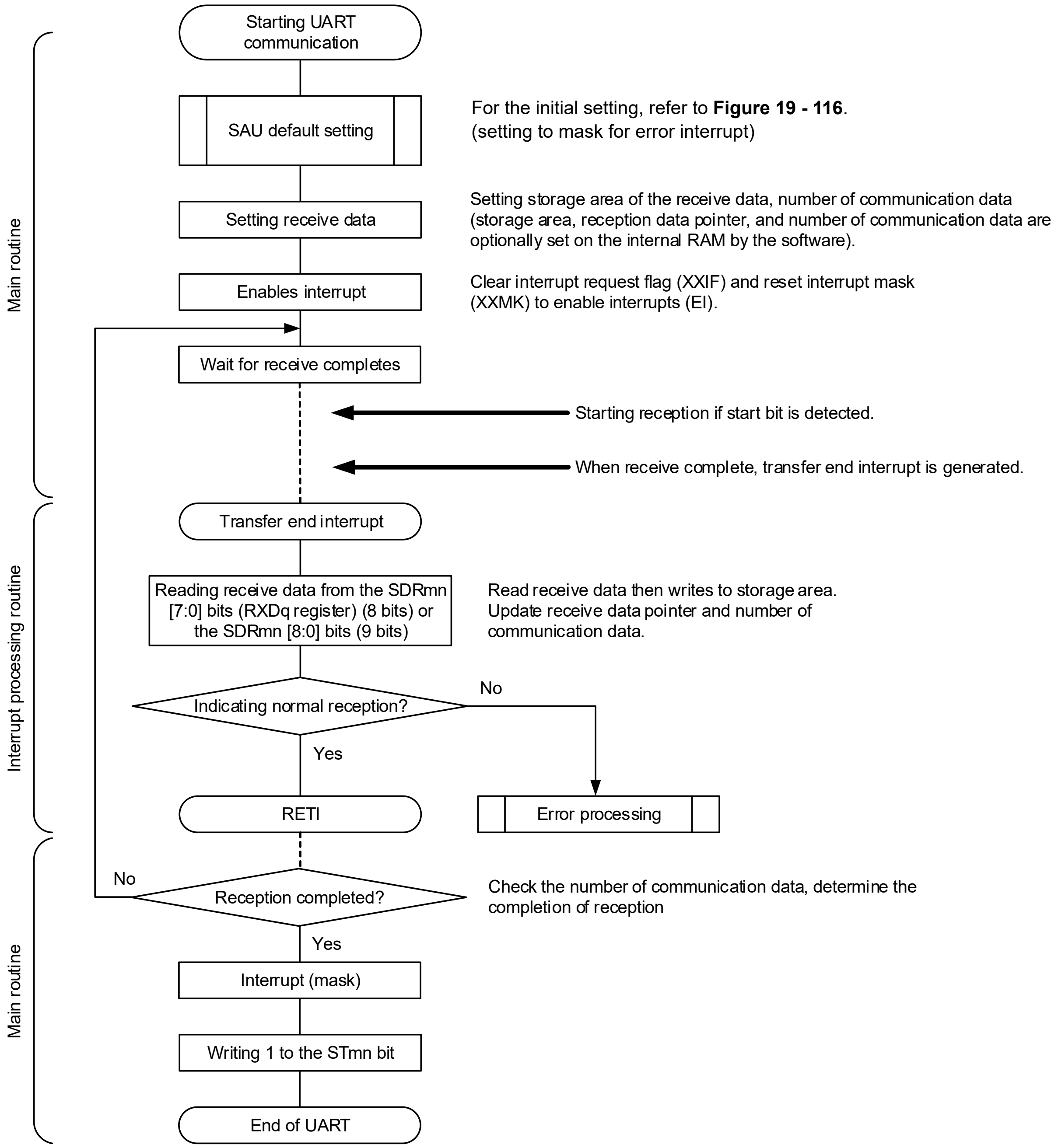
(3) Processing details

Figure 19 - 119 Timing Chart of UART Reception



Remark m: Unit number (m = 0), n: Channel number (n = 1, 3), mn = 01, 03
r: Channel number (r = n – 1), q: UART number (q = 0, 1)

Figure 19 - 120 Flowchart of UART Reception



19.7.3 SNOOZE mode function

The SNOOZE mode makes the UART perform reception operations upon RxDq pin input detection while in the STOP mode. Normally the UART stops communication in the STOP mode. However, using the SNOOZE mode enables the UART to perform reception operations without CPU operation.

When using UARTq in the SNOOZE mode, make the following settings before entering the STOP mode (See **Figures 19 - 123** and **19 - 125 Flowchart of SNOOZE Mode Operation**).

- In the SNOOZE mode, the baud rate setting for UART reception needs to be changed to a value different from that in normal operation. Set the SPSm register and bits 15 to 9 of the SDRmn register with reference to Table 19 - 5.
- Set the EOCmn and SSECmn bits. This is for enabling or stopping generation of an error interrupt (INTSRE0) when a communication error occurs.
- When using the SNOOZE mode function, set the SWCm bit of serial standby control register m (SSCm) to 1 just before switching to the STOP mode. After the initial setting has completed, set the SSm1 bit of serial channel start register m (SSm) to 1.
- A UARTq starts reception in SNOOZE mode on detecting input of the start bit on the RxDq pin following a transition of the CPU to the STOP mode.

Caution 1. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock (f_{IH}) is selected for f_{CLK}.

Caution 2. The transfer rate in the SNOOZE mode is only 4800 bps.

Caution 3. When SWCm = 1, UARTq can be used only when the reception operation is started in the STOP mode. When used simultaneously with another SNOOZE mode function or interrupt, if the reception operation is started in a state other than the STOP mode, such as those given below, data may not be received correctly and a framing error or parity error may be generated.

- **When after the SWCm bit has been set to 1, the reception operation is started before the STOP mode is entered**
- **When the reception operation is started while another function is in the SNOOZE mode**
- **When after returning from the STOP mode to normal operation due to an interrupt or other cause, the reception operation is started before the SWCm bit is returned to 0**

Caution 4. If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFmn, FEFmn, or OVFmn flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFmn, FEFmn, or OVFmn flag before setting the SWC0 bit to 1 and read the value in bits 7 to 0 (RxDq register) of the SDRm1 register.

Caution 5. The CPU shifts from the STOP mode to the SNOOZE mode on detecting the valid edge of the RxDq signal. Note, however, that transfer through the UART channel may not start and the CPU may remain in the SNOOZE mode if an input pulse on the RxDq pin is too short to be detected as a start bit.

In such cases, data may not be received correctly, and this may lead to a framing error or parity error in the next UART transfer.

Table 19 - 5 Baud Rate Setting for UART Reception in SNOOZE Mode

High-speed On-chip Oscillator (f_{IH})	Baud Rate for UART Reception in SNOOZE Mode			
	Baud Rate of 4800 bps			
	Operation Clock (f_{MCK})	SDRmn [15:9]	Maximum Permissible Value	Minimum Permissible Value
32 MHz ± 1.0% Note	$f_{CLK}/2^5$	105	2.27%	-1.53%
24 MHz ± 1.0% Note	$f_{CLK}/2^5$	79	1.60%	-2.18%
16 MHz ± 1.0% Note	$f_{CLK}/2^4$	105	2.27%	-1.53%
12 MHz ± 1.0% Note	$f_{CLK}/2^4$	79	1.60%	-2.19%
8 MHz ± 1.0% Note	$f_{CLK}/2^3$	105	2.27%	-1.53%
6 MHz ± 1.0% Note	$f_{CLK}/2^3$	79	1.60%	-2.19%
4 MHz ± 1.0% Note	$f_{CLK}/2^2$	105	2.27%	-1.53%
3 MHz ± 1.0% Note	$f_{CLK}/2^2$	79	1.60%	-2.19%
2 MHz ± 1.0% Note	$f_{CLK}/2$	105	2.27%	-1.54%
1 MHz ± 1.0% Note	f_{CLK}	105	2.27%	-1.57%

Note When the accuracy of the clock frequency of the high-speed on-chip oscillator is ±1.5% or ±2.0%, the permissible range becomes smaller as shown below.

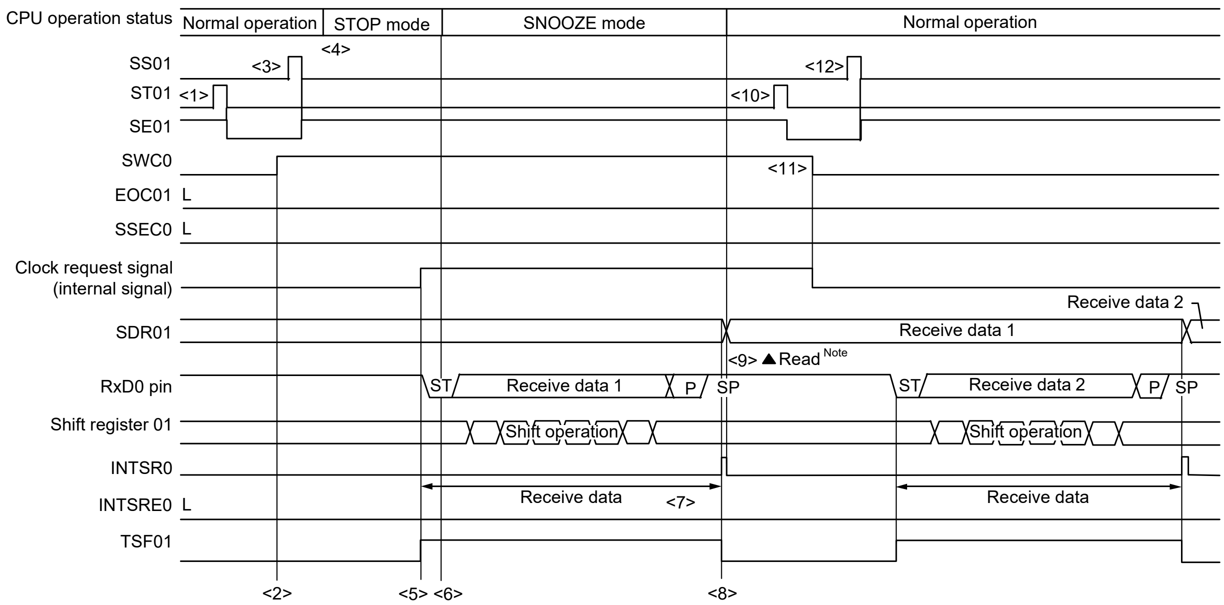
- In the case of f_{IH} ± 1.5%, perform (Maximum permissible value - 0.5%) and (Minimum permissible value + 0.5%) to the values in the above table.
- In the case of f_{IH} ± 2.0%, perform (Maximum permissible value - 1.0%) and (Minimum permissible value + 1.0%) to the values in the above table.

Remark The maximum permissible value and minimum permissible value are permissible values for the baud rate in UART reception. The baud rate on the transmitting side should be set to fall inside this range.

(1) SNOOZE mode operation (EOCm1 = 0, SSECm = 0/1)

Because of the setting of EOCm1 = 0, even though a communication error occurs, an error interrupt (INTSREq) is not generated, regardless of the setting of the SSECm bit. A transfer end interrupt (INTSRq) will be generated.

Figure 19 - 121 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)



Note Read the received data when SWCm is 1.

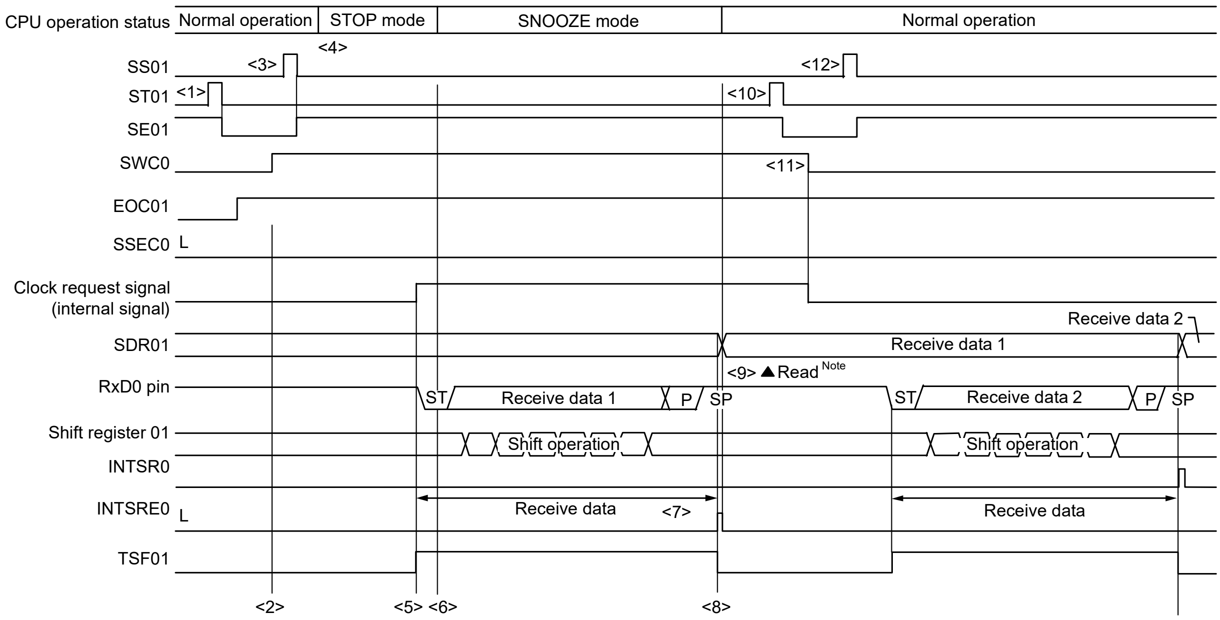
Caution Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit to stop the operation).
After the reception finishes, also clear the SWCm bit to 0 (to exit SNOOZE mode).

Remark 1. <1> to <12> in the figure correspond to <1> to <12> in Figure 19 - 123 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0).

Remark 2. m = 0; q = 0

(2) SNOOZE mode operation (EOCm1 = 1, SSECm = 0: Error interrupt (INTSREq) generation is enabled)
Because EOCm1 = 1 and SSECm = 0, an error interrupt (INTSREq) is generated when a communication error occurs.

Figure 19 - 122 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)



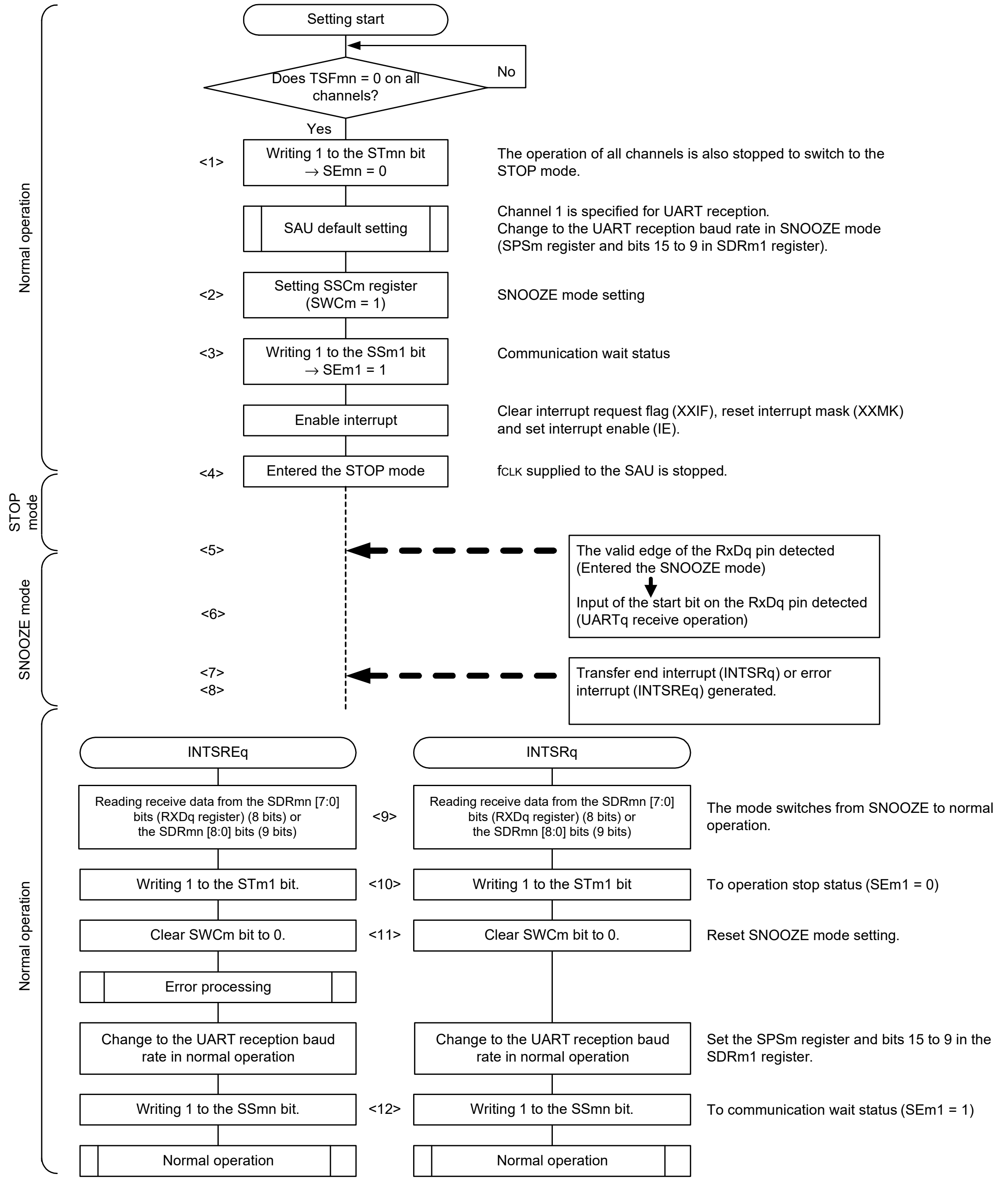
Note Read the received data when SWCm = 1.

Caution **Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit to stop the operation).**
After the reception finishes, also clear the SWCm bit to 0 (to exit SNOOZE mode).

Remark 1. <1> to <12> in the figure correspond to <1> to <12> in Figure 19 - 123 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0).

Remark 2. m = 0; q = 0

Figure 19 - 123 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)

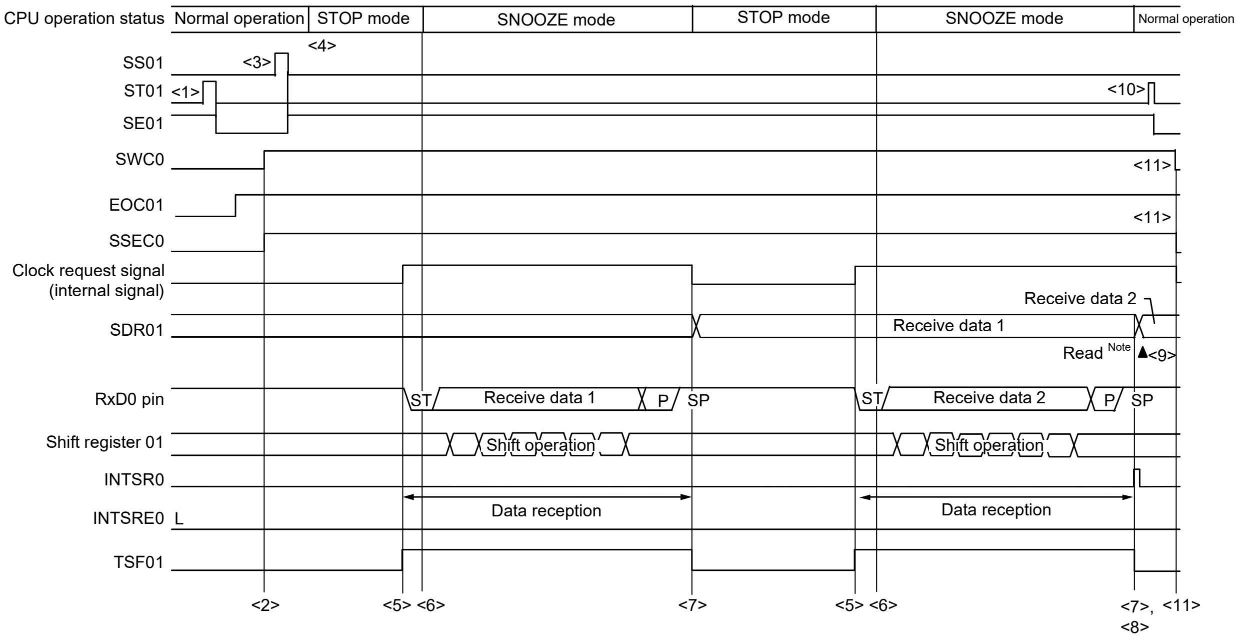


Remark 1. <1> to <12> in the figure correspond to <1> to <12> in Figure 19 - 121 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1) and Figure 19 - 122 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0).

Remark 2. m = 0; q = 0

(3) SNOOZE mode operation (EOCm1 = 1, SSECm = 1: Error interrupt (INTSREq) generation is stopped)
Because EOCm1 = 1 and SSECm = 1, an error interrupt (INTSREq) is not generated when a communication error occurs.

Figure 19 - 124 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)



Note Only read received data while SWCm = 1.

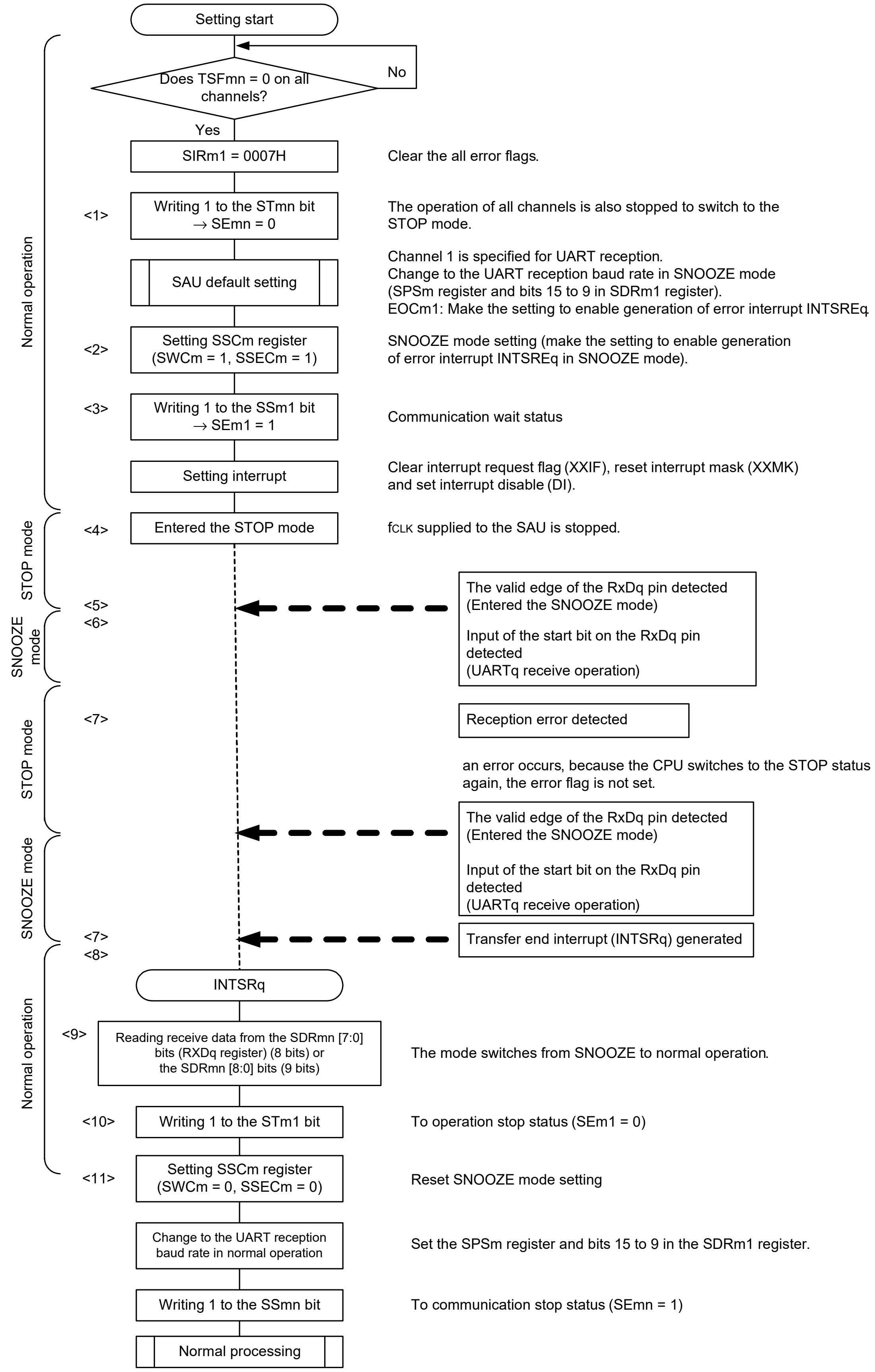
Caution 1. Before switching to the SNOOZE mode or after reception operation in the SNOOZE mode finishes, set the STm1 bit to 1 (clear the SEm1 bit to stop the operation).
After the reception finishes, also clear the SWCm bit to 0 (to exit SNOOZE mode).

Caution 2. If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFm1, FEFm1, or OVFm1 flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFm1, FEFm1, or OVFm1 flag before setting the SWCm bit to 1 and read the value in SDRm1[7:0] (RxDq register) (8 bits) or SDRm1[8:0] (9 bits).

Remark 1. <1> to <11> in the figure correspond to <1> to <11> in Figure 19 - 125 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1).

Remark 2. m = 0; q = 0

Figure 19 - 125 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)



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

Caution **If a parity error, framing error, or overrun error occurs while the SSECm bit is set to 1, the PEFm1, FEFm1, or OVFm1 flag is not set and an error interrupt (INTSREq) is not generated. Therefore, when the setting of SSECm = 1 is made, clear the PEFm1, FEFm1, or OVFm1 flag before setting the SWCm bit to 1 and read the value in SDRm1[7:0] (RxDq register) (8 bits) or SDRm1[8:0] (9 bits).**

Remark 1. <1> to <11> in the figure correspond to <1> to <11> in Figure 19 - 124 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1).

Remark 2. m = 0; q = 0

19.7.4 Calculating baud rate

(1) Baud rate calculation expression

The baud rate for UART (UART0, UART1) communication can be calculated by the following expressions.

(Baud rate) = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [bps]

Caution Setting serial data register mn (SDRmn) SDRmn[15:9] = (0000000B, 0000001B) is prohibited.

Remark 1. When UART is used, the value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000010B to 1111111B) and therefore is 2 to 127.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 3), mn = 00 to 03

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 19 - 6 Selection of Operation Clock for UART

SMRmn Register	SPSm Register								Operation Clock (f_{MCK}) Note	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 32 MHz
0	×	×	×	×	0	0	0	0	f_{CLK}	32 MHz
	×	×	×	×	0	0	0	1	$f_{CLK}/2$	16 MHz
	×	×	×	×	0	0	1	0	$f_{CLK}/2^2$	8 MHz
	×	×	×	×	0	0	1	1	$f_{CLK}/2^3$	4 MHz
	×	×	×	×	0	1	0	0	$f_{CLK}/2^4$	2 MHz
	×	×	×	×	0	1	0	1	$f_{CLK}/2^5$	1 MHz
	×	×	×	×	0	1	1	0	$f_{CLK}/2^6$	500 kHz
	×	×	×	×	0	1	1	1	$f_{CLK}/2^7$	250 kHz
	×	×	×	×	1	0	0	0	$f_{CLK}/2^8$	125 kHz
	×	×	×	×	1	0	0	1	$f_{CLK}/2^9$	62.5 kHz
	×	×	×	×	1	0	1	0	$f_{CLK}/2^{10}$	31.25 kHz
	×	×	×	×	1	0	1	1	$f_{CLK}/2^{11}$	15.63 kHz
	×	×	×	×	1	1	0	0	$f_{CLK}/2^{12}$	7.81 kHz
	×	×	×	×	1	1	0	1	$f_{CLK}/2^{13}$	3.91 kHz
	×	×	×	×	1	1	1	0	$f_{CLK}/2^{14}$	1.95 kHz
	×	×	×	×	1	1	1	1	$f_{CLK}/2^{15}$	977 Hz
1	0	0	0	0	×	×	×	×	f_{CLK}	32 MHz
	0	0	0	1	×	×	×	×	$f_{CLK}/2$	16 MHz
	0	0	1	0	×	×	×	×	$f_{CLK}/2^2$	8 MHz
	0	0	1	1	×	×	×	×	$f_{CLK}/2^3$	4 MHz
	0	1	0	0	×	×	×	×	$f_{CLK}/2^4$	2 MHz
	0	1	0	1	×	×	×	×	$f_{CLK}/2^5$	1 MHz
	0	1	1	0	×	×	×	×	$f_{CLK}/2^6$	500 kHz
	0	1	1	1	×	×	×	×	$f_{CLK}/2^7$	250 kHz
	1	0	0	0	×	×	×	×	$f_{CLK}/2^8$	125 kHz
	1	0	0	1	×	×	×	×	$f_{CLK}/2^9$	62.5 kHz
	1	0	1	0	×	×	×	×	$f_{CLK}/2^{10}$	31.25 kHz
	1	0	1	1	×	×	×	×	$f_{CLK}/2^{11}$	15.63 kHz
	1	1	0	0	×	×	×	×	$f_{CLK}/2^{12}$	7.81 kHz
	1	1	0	1	×	×	×	×	$f_{CLK}/2^{13}$	3.91 kHz
	1	1	1	0	×	×	×	×	$f_{CLK}/2^{14}$	1.95 kHz
	1	1	1	1	×	×	×	×	$f_{CLK}/2^{15}$	977 Hz

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Remark 1. ×: Don’t care

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0 to 3), mn = 00 to 03

(2) Baud rate error during transmission

The baud rate error of UART (UART0, UART1) communication during transmission can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

Baud rate error = (Calculated baud rate value) ÷ (Target baud rate) ×100 –100 [%]

Here is an example of setting a UART baud rate at f_{CLK} = 32 MHz.

UART Baud Rate (Target Baud Rate)	f_{CLK} = 32 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Baud Rate	Error from Target Baud Rate
300 bps	$f_{CLK}/2^9$	103	300.48 bps	+0.16%
600 bps	$f_{CLK}/2^8$	103	600.96 bps	+0.16%
1200 bps	$f_{CLK}/2^7$	103	1201.92 bps	+0.16%
2400 bps	$f_{CLK}/2^6$	103	2403.85 bps	+0.16%
4800 bps	$f_{CLK}/2^5$	103	4807.69 bps	+0.16%
9600 bps	$f_{CLK}/2^4$	103	9615.38 bps	+0.16%
19200 bps	$f_{CLK}/2^3$	103	19230.8 bps	+0.16%
31250 bps	$f_{CLK}/2^3$	63	31250.0 bps	±0.0%
38400 bps	$f_{CLK}/2^2$	103	38461.5 bps	+0.16%
76800 bps	$f_{CLK}/2$	103	76923.1 bps	+0.16%
153600 bps	f_{CLK}	103	153846 bps	+0.16%
312500 bps	f_{CLK}	50	312500 bps	±0.39%

Remark m: Unit number (m = 0), n: Channel number (n = 0, 2), mn = 00, 02

(3) Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UART0, UART1) communication can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

$$\text{Maximum receivable baud rate} = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times \text{Brate}$$

$$\text{Minimum receivable baud rate} = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times \text{Brate}$$

Brate: Calculated baud rate value at the reception side (See **19.7.4 (1) Baud rate calculation expression**.)

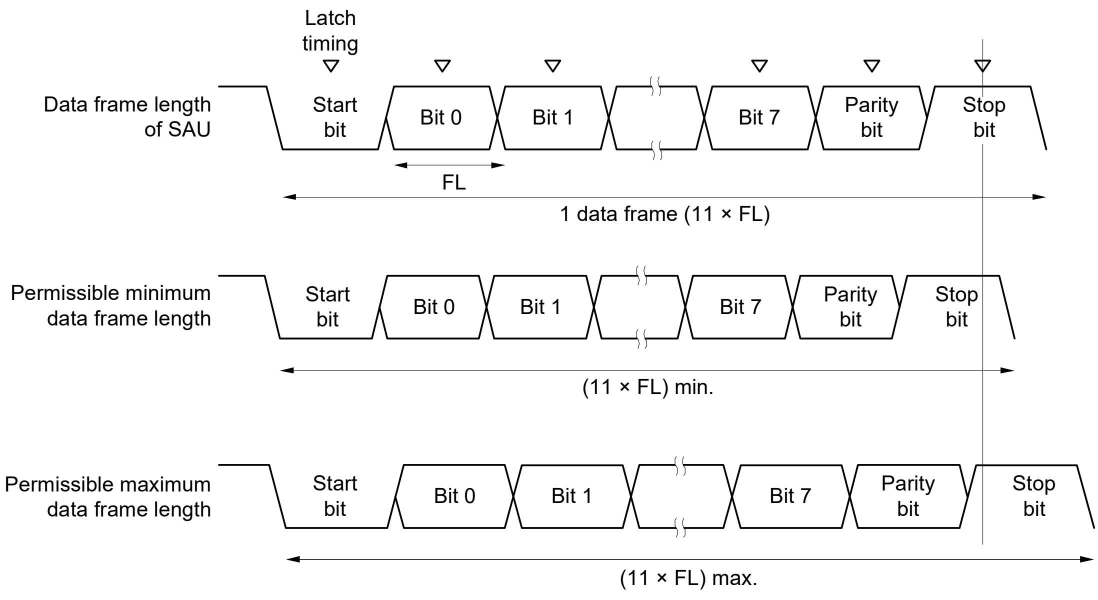
k: SDRmn[15:9] + 1

Nfr: 1 data frame length [bits]

= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

Remark m: Unit number (m = 0), n: Channel number (n = 1, 3), mn = 01, 03

Figure 19 - 126 Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in Figure 19 - 126, 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.

19.7.5 Procedure for processing errors that occurred during UART (UART0, UART1) communication

The procedure for processing errors that occurred during UART (UART0, UART1) communication is described in Figures 19 - 127 and 19 - 128.

Figure 19 - 127 Processing Procedure in Case of Parity Error or Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn) →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.

Figure 19 - 128 Processing Procedure in Case of Framing Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets the STmn bit of serial channel stop register m (STm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operating.	
Synchronization with other party of communication		Synchronization with the other party of communication is re-established and communication is resumed because it is considered that a framing error has occurred because the start bit has been shifted.
Sets the SSmn bit of serial channel start register m (SSm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate.	

Remark m: Unit number (m = 0), n: Channel number (n = 0 to 3), mn = 00 to 03

19.8 LIN Communication Operation

19.8.1 LIN transmission

Of UART transmission, UART0 support LIN communication.
For LIN transmission, channel 0 of unit 0 is used.

UART	UART0	UART1
LIN communication	Supported	Not supported
Target channel	Channel 0 of SAU0	—
Pins used	TxD0	—
Interrupt	INTST0	—
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.	
Error detection flag	None	
Transfer data length	8 bits	
Transfer rate Note	Max. $f_{MCK}/12$ [bps] (SDR00 [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]	
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)	
Parity bit	No parity bit	
Stop bit	Appending 1 bit	
Data direction	MSB first	

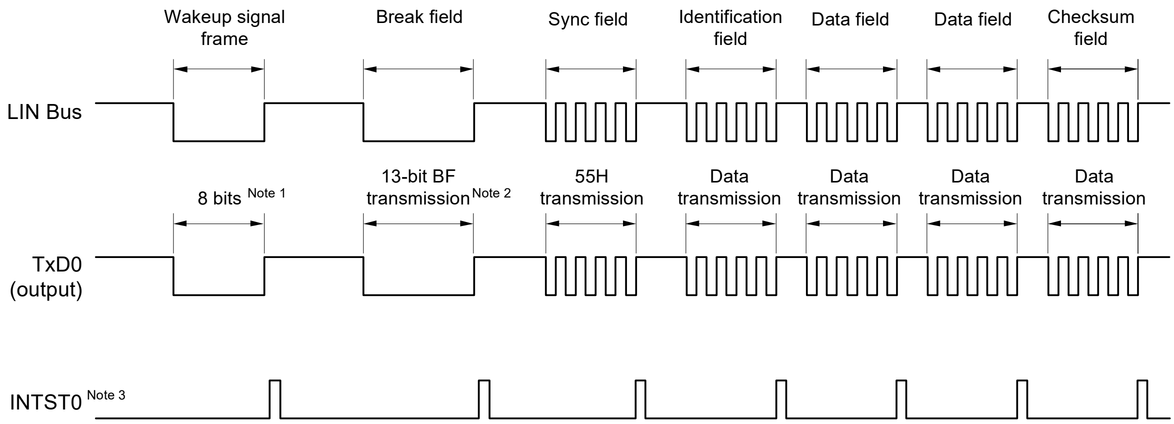
Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 33** or **CHAPTER 34 ELECTRICAL SPECIFICATIONS**). In addition, LIN communication is usually 2.4/9.6/19.2 kbps is often used.

Remark f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol designed to reduce the cost of an automobile network. Communication of LIN is single-master communication and up to 15 slaves can be connected to one master. The slaves are used to control switches, actuators, and sensors, which are connected to the master via LIN.
Usually, the master is connected to a network such as CAN (Controller Area Network). A LIN bus is a single-wire bus to which nodes are connected via transceiver conforming to ISO9141.
According to the protocol of LIN, the master transmits a frame by attaching baud rate information to it. A slave receives this frame and corrects a baud rate error from the master. If the baud rate error of a slave is within ±15%, communication can be established.

Figure 19 - 129 outlines a transmission operation of LIN.

Figure 19 - 129 Transmission Operation of LIN



Note 1. Set the baud rate in accordance with the wakeup signal regulations and transmit data of 80H.

Note 2. A break field is defined to have a width of 13 bits and output a low level. Where the baud rate for main transfer is N [bps], therefore, the baud rate of the break field is calculated as follows.

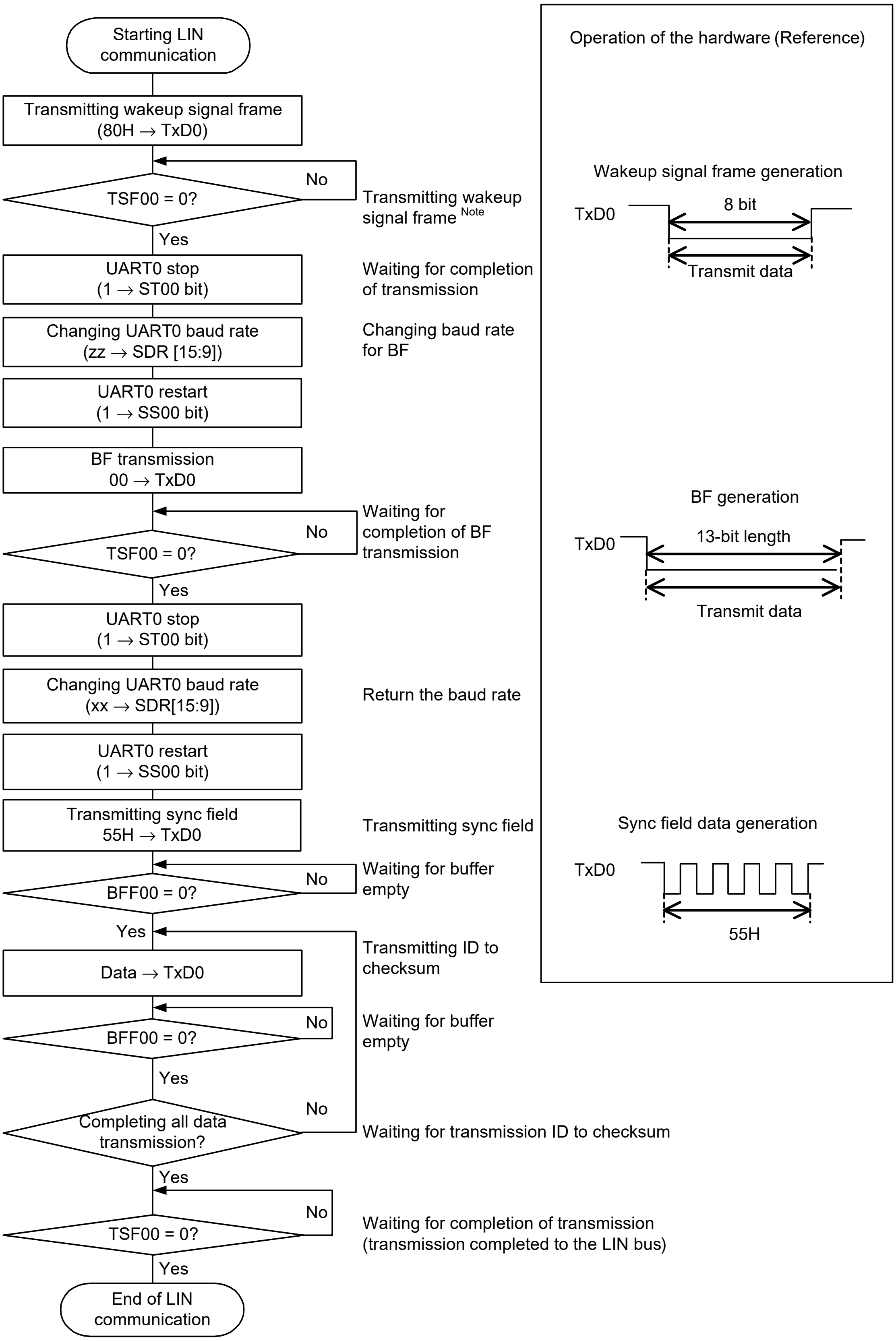
(Baud rate of break field) = 9/13 × N

By transmitting data of 00H at this baud rate, a break field is generated.

Note 3. INTST0 is output upon completion of transmission. INTST0 is also output at BF transmission.

Remark The interval between fields is controlled by software.

Figure 19 - 130 Flowchart for LIN Transmission



Note When LIN-bus start from sleep status only.

Remark Default setting of the UART is complete, and the flow from the transmission enable status.

19.8.2 LIN reception

Of UART reception, UART0 support LIN communication.

For LIN reception, channel 1 of unit 0 is used.

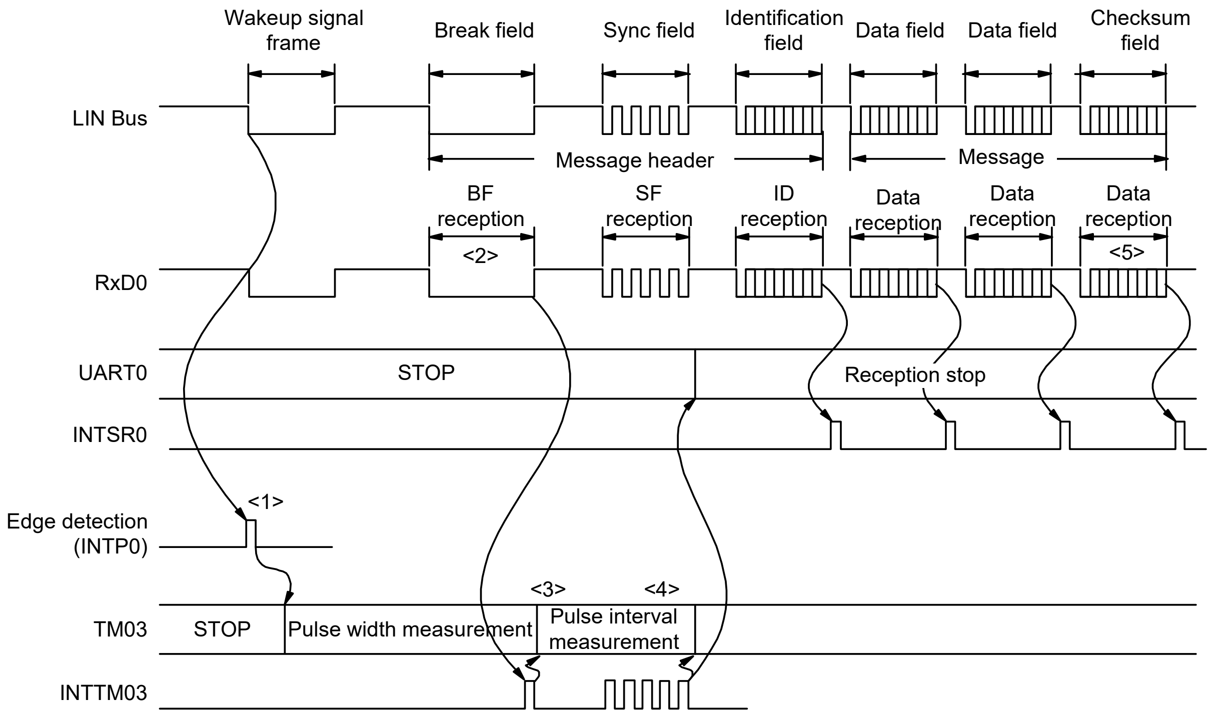
UART	UART0	UART1
LIN communication	Supported	Not supported
Target channel	Channel 1 of SAU0	—
Pins used	RxD0	—
Interrupt	INTSR0	—
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error interrupt	INTSRE0	—
Error detection flag	• Framing error detection flag (FEF01) • Overrun error detection flag (OVF01)	
Transfer data length	8 bits	
Transfer rate Note	Max. $f_{MCK}/12$ [bps] (SDR01 [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{15} \times 128)$ [bps]	
Data phase	Non-reverse output (default: high level) Reverse output (default: low level)	
Parity bit	No parity bit (The parity bit is not checked.)	
Stop bit	Appending 1 bit	
Data direction	LSB first	

Note Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 33** or **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock frequency of target channel
f_{CLK}: System clock frequency

Figure 19 - 131 outlines a reception operation of LIN.

Figure 19 - 131 Reception Operation of LIN



Here is the flow of signal processing.

<1> The wakeup signal is detected by detecting an interrupt edge (INTP0) on a pin. When the wakeup signal is detected, change TM03 to pulse width measurement upon detection of the wakeup signal to measure the low level width of the BF signal. Then wait for BF signal reception.

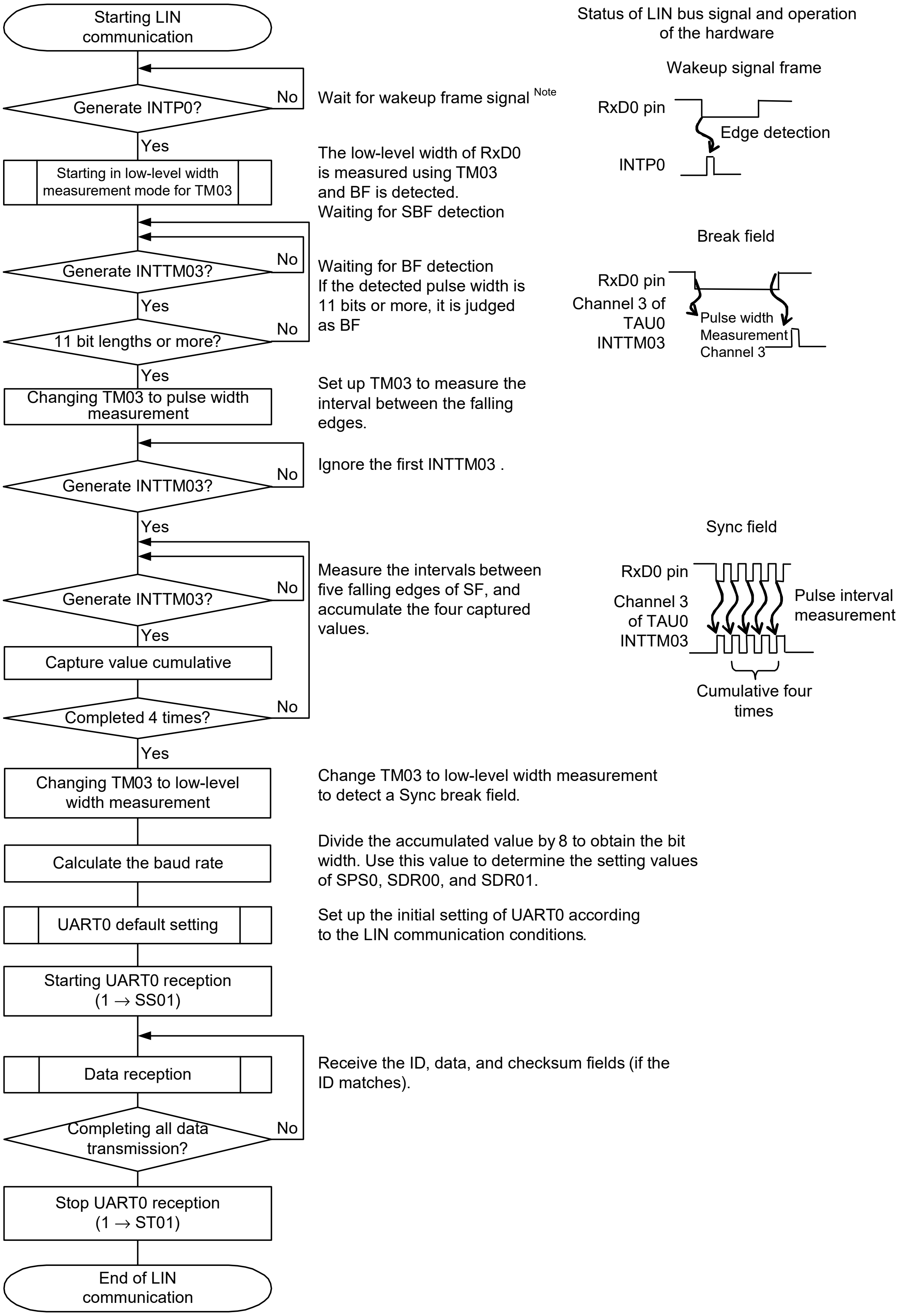
<2> TM03 starts measuring the low-level width upon detection of the falling edge of the BF signal, and then captures the data upon detection of the rising edge of the BF signal. The captured data is used to judge whether it is the BF signal.

<3> When the BF signal has been received normally, change TM03 to pulse interval measurement and measure the interval between the falling edges of the RxD0 signal in the Sync field four times (see **6.8.3 Operation as input pulse interval measurement**).

<4> Calculate a baud rate error from the bit interval of sync field (SF). Stop UART0 once and adjust (re-set) the baud rate.

<5> The checksum field should be distinguished by software. In addition, processing to initialize UART0 after the checksum field is received and to wait for reception of BF should also be performed by software.

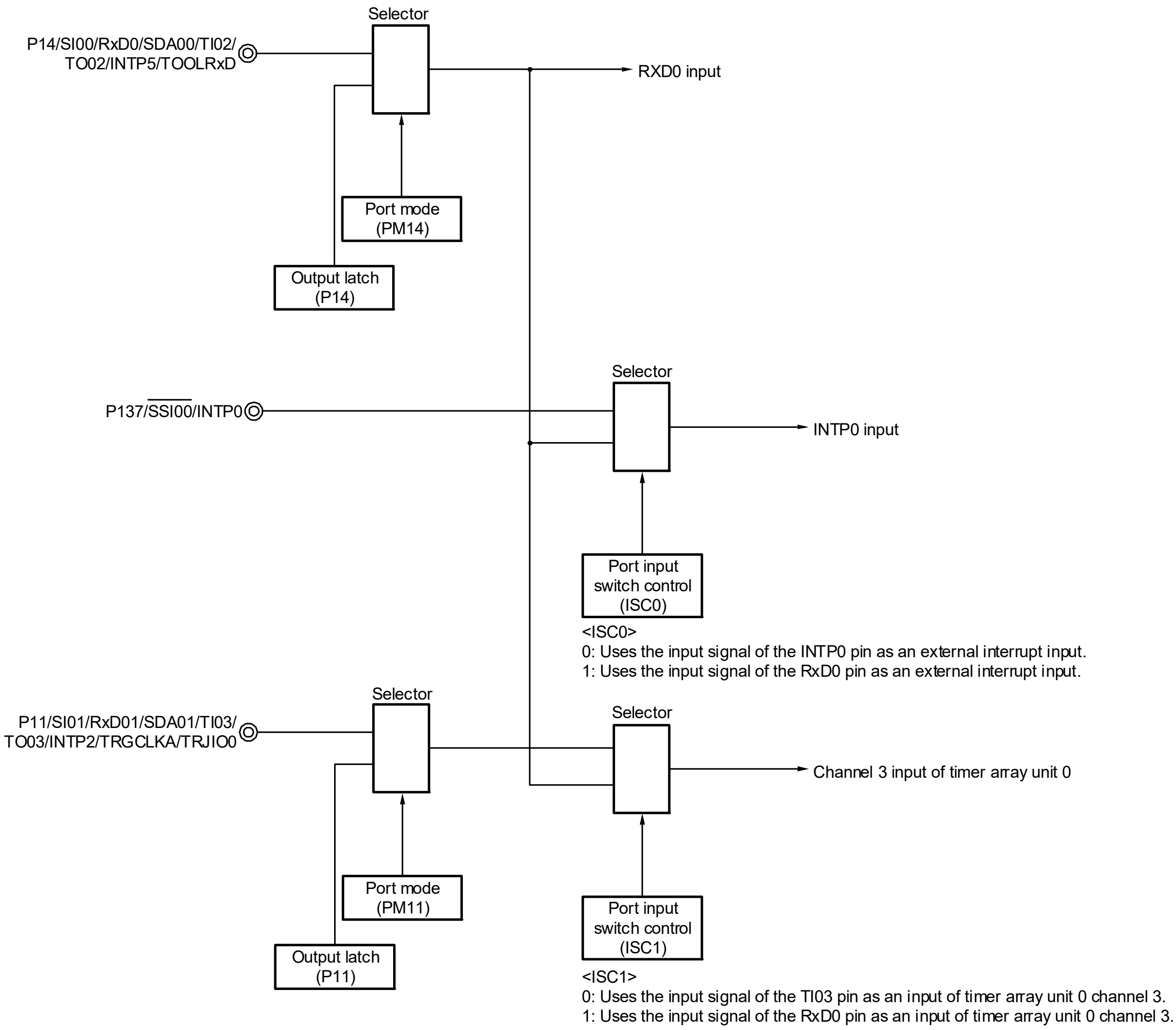
Figure 19 - 132 Flowchart for LIN Reception



Note Required in the sleep status only.

Figure 19 - 133 shows the configuration of a port that manipulates reception of LIN.
The wakeup signal transmitted from the master of LIN is received by detecting an edge of an external interrupt (INTP0). The length of the sync field transmitted from the master can be measured by using the external event capture operation of timer array unit 0 to calculate a baud-rate error.
By controlling switch of port input (ISC0/ISC1), the input source of port input (RxD0) for reception can be input to the external interrupt pin (INTP0) and timer array unit

Figure 19 - 133 Port Configuration for Manipulating Reception of LIN



Remark ISC0, ISC1: Bits 0 and 1 of the input switch control register (ISC) (See **Figure 19 - 22**.)

The peripheral functions used for the LIN communication operation are as follows.

<Peripheral functions used>

- External interrupt (INTP0); Wakeup signal detection
  Usage: To detect an edge of the wakeup signal and the start of communication
- Channel 3 of timer array unit; Baud rate error detection, break field (BF) detection.
  Usage: To detect the length of the sync field (SF) and divide it by the number of bits in order to detect an error
  (The interval of the edge input to RxD0 is measured in the capture mode.)
  Measured the low-level width, determine whether break field (BF).
- Channels 0 and 1 (UART0) of serial array unit 0 (SAU0)

19.9 Operation of Simplified I^{2}C (IIC00, IIC01) Communication

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This communication function is designed to execute single communication with devices such as EEPROM, flash memory, and A/D converter, and therefore, can be used only by the master.
Make sure by using software, as well as operating the control registers, that the AC specifications of the start and stop conditions are observed.

[Data transmission/reception]
• Master transmission, master reception (only master function with a single master)
• ACK output function Note and ACK detection function
• Data length of 8 bits
(When an address is transmitted, the address is specified by the higher 7 bits, and the least significant bit is used for R/W control.)
• Generation of start condition and stop condition for software

[Interrupt function]
• Transfer end interrupt

[Error detection flag]
• Overrun error
• ACK error

* [Functions not supported by simplified I^{2}C]
• Slave transmission, slave reception
• Multi-master function (arbitration loss detection function)
• Wait detection function

Note When receiving the last data, ACK will not be output if 0 is written to the SOEmn (SOEm register) bit and serial communication data output is stopped. See the processing flow in **19.9.3 (2)** for details.

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

The channel supporting simplified I^{2}C (IIC00, IIC01) is channels 0 and 1 of SAU0.

Unit	Channel	Used as Simplified SPI (CSI)	Used as UART	Used as Simplified I^{2}C
0	0	CSI00 (supporting slave select input)	UART0 (supporting LIN-bus)	IIC00
	1	CSI01		IIC01
	2	—	UART1	—
	3	—		—

Simplified I^{2}C (IIC00, IIC01) performs the following four types of communication operations.
• Address field transmission (See **19.9.1**.)
• Data transmission (See **19.9.2**.)
• Data reception (See **19.9.3**.)
• Stop condition generation (See **19.9.4**.)

19.9.1 Address field transmission

Address field transmission is a transmission operation that first executes in I2C communication to identify the target for transfer (slave). After a start condition is generated, an address (7 bits) and a transfer direction (1 bit) are transmitted in one frame.

Simplified I2C	IIC00	IIC01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCL00, SDA00[Note 1]	SCL01, SDA01[Note 1]
Interrupt	INTIIC00	INTIIC01
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	ACK error detection flag (PEFmn)	
Transfer data length	8 bits (transmitted with specifying the higher 7 bits as address and the least significant bit as R/W control)	
Transfer rate[Note 2]	Max. $f_{MCK}/4$ [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I2C. • Max. 1 MHz (fast mode plus) • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)	
Data level	Non-reversed output (default: high level)	
Parity bit	No parity bit	
Stop bit	Appending 1 bit (for ACK reception timing)	
Data direction	MSB first	

Note 1. To perform communication via simplified I2C, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) with the port output mode register (POMxx). For details, see **4.3 Registers Controlling Port Function** and **4.5 Register Settings When Using Alternate Function**.
When IIC00 or IIC01 is communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) also for the clock input/output pins (SCL00, SCL01).
For details, see **4.4.4 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers**.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 33** or **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

(1) Register setting

Figure 19 - 134 Example of Contents of Registers for Address Field Transmission of Simplified I²C (IIC00, IIC01)

(a) Serial mode register mn (SMRmn)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0 Note 1	0	SISmn0 0 Note 1	1	0	0	MDmn2 1	MDmn1 0	MDmn0 0

Operation clock (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 1	RXEmn 0	DAPmn 0	CKPmn 0	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0	0	SLCmn1 0 Note 2	SLCmn0 1	0	1	DLSmn1 1	DLSmn0 1

Setting of parity bit
00B: No parity

Setting of stop bit
01B: Appending 1 bit (ACK)

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)

	15–9	8	7–0
SDRmn	Baud rate setting	0	Transmit data setting (address + R/W)

Bits 7–0: SIOr

(d) Serial output register m (SOm)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	CKOm2 ×	CKOm1 0/1	CKOm0 0/1	0	0	0	0	0	SOm2 ×	SOm1 0/1	SOm0 0/1

Start condition is generated by manipulating the SOmn bit.

(e) Serial output enable register m (SOEm)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	SOEm1 0/1	SOEm0 0/1

SOEmn = 0 until the start condition is generated, and SOEmn = 1 after generation.

(f) Serial channel start register m (SSm)... Set only the bits of the target channel is 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 0/1	SSm0 0/1

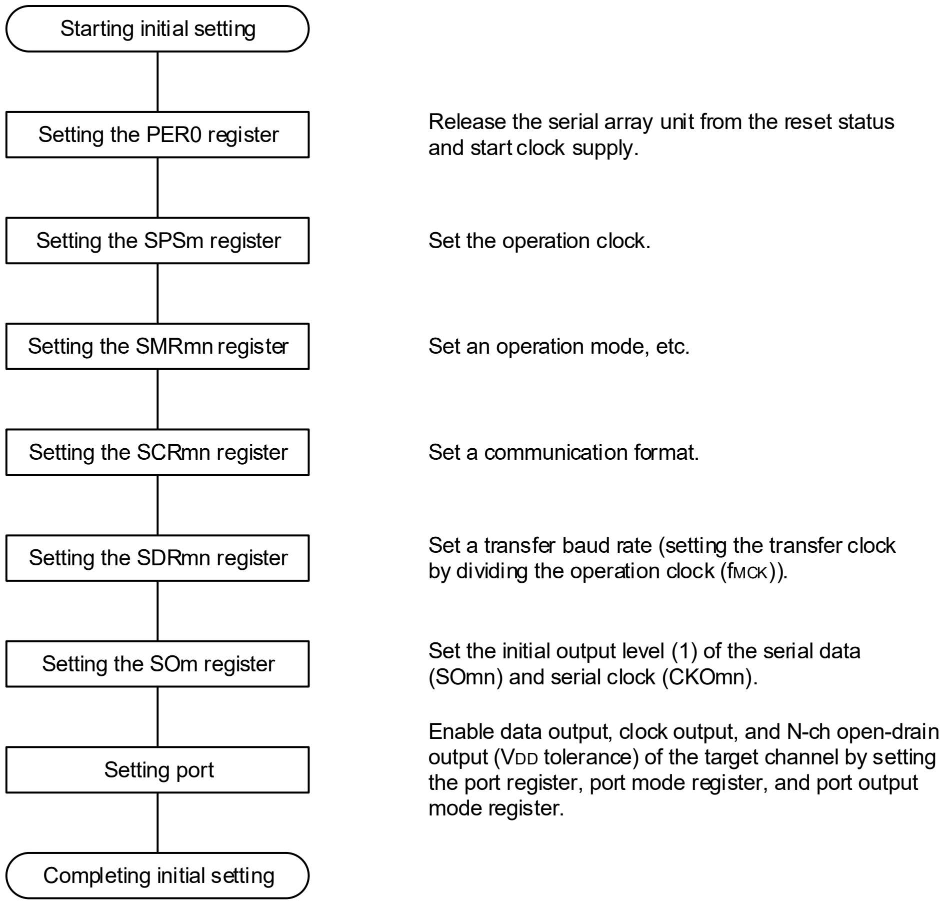
Note 1. Only provided for the SMR00 register.
Note 2. Only provided for the SCR00 register.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01), mn = 00, 01

Remark 2. □: Setting is fixed in the IIC mode,
■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

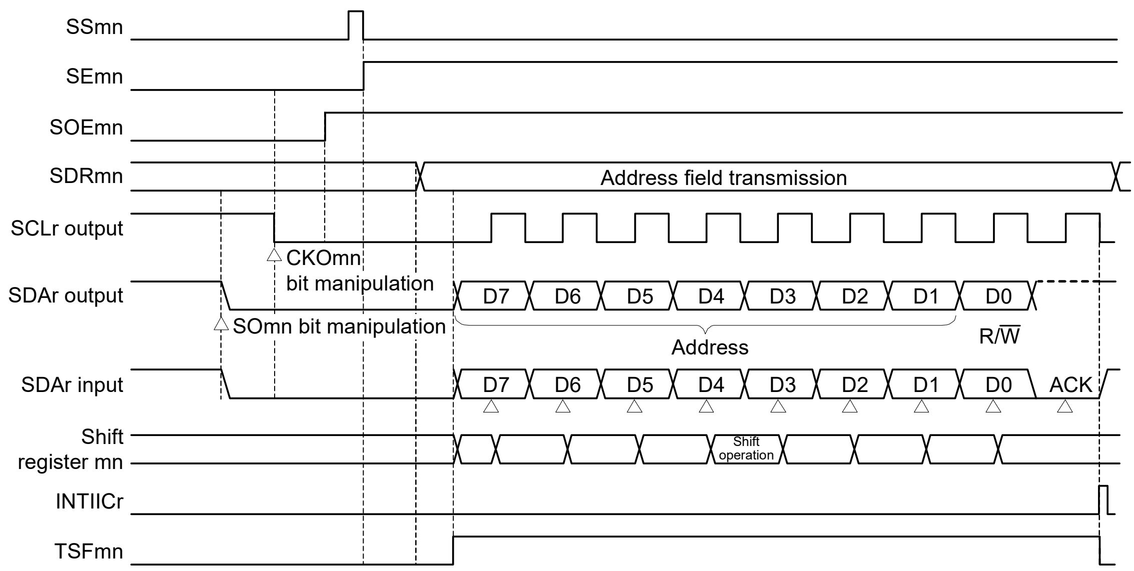
Figure 19 - 135 Initial Setting Procedure for Address Field Transmission



Remark At the end of the initial setting, the simplified I^{2}C (IIC00, IIC01) must be set so that output is disabled and operations are stopped.

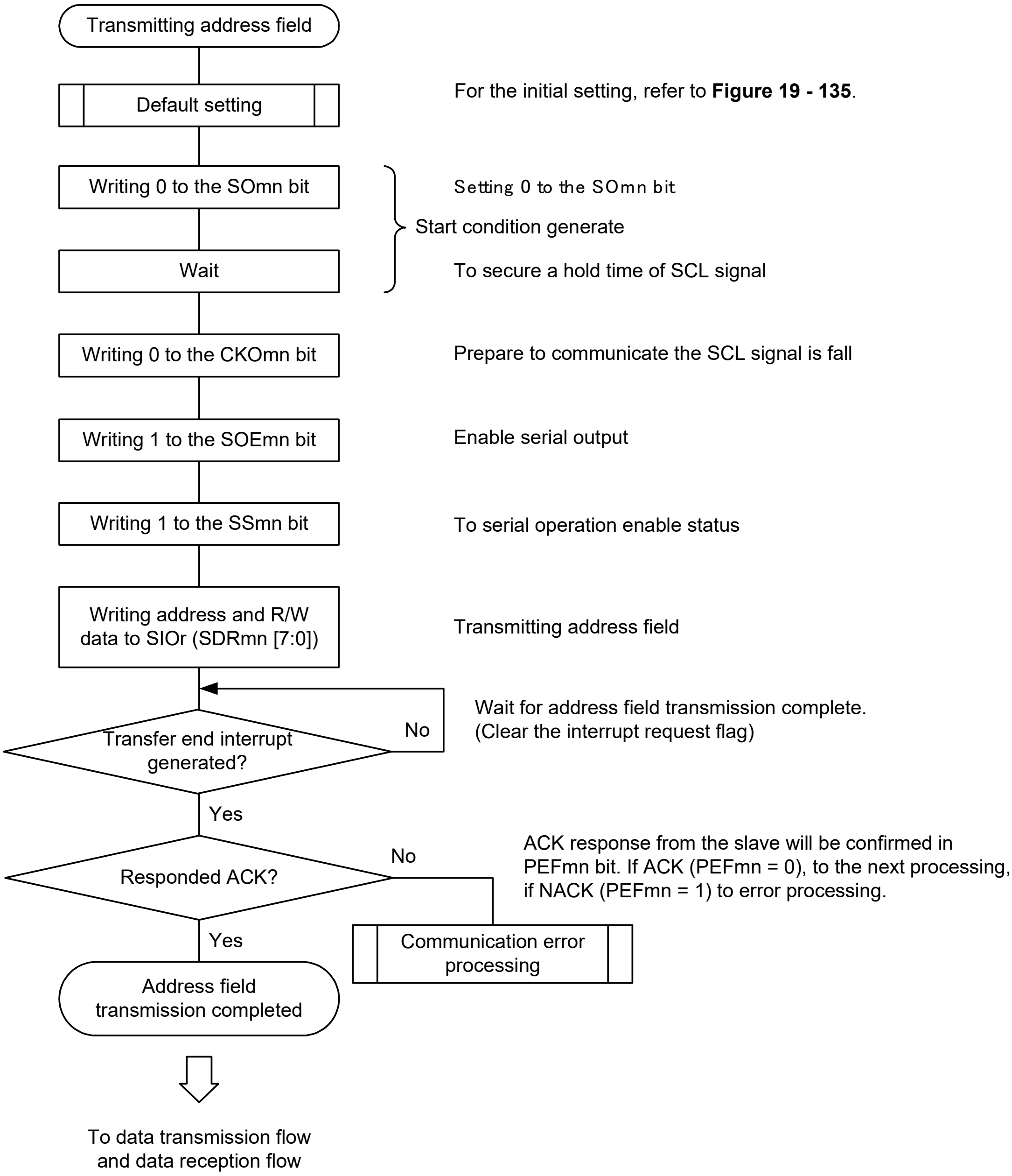
(3) Processing details

Figure 19 - 136 Timing Chart of Address Field Transmission



Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01), mn = 00, 01

Figure 19 - 137 Flowchart of Address Field Transmission



19.9.2 Data transmission

Data transmission is an operation to transmit data to the target for transfer (slave) after transmission of an address field. After all data are transmitted to the slave, a stop condition is generated and the bus is released.

Simplified I^2C	IIC00	IIC01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCL00, SDA00[Note 1]	SCL01, SDA01[Note 1]
Interrupt	INTIIC00	INTIIC01
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	ACK error flag (PEFmn)	
Transfer data length	8 bits	
Transfer rate[Note 2]	Max. $f_{MCK}/4$ [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I^2C. • Max. 1 MHz (fast mode plus) • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)	
Data level	Non-reverse output (default: high level)	
Parity bit	No parity bit	
Stop bit	Appending 1 bit (for ACK reception timing)	
Data direction	MSB first	

Note 1. To perform communication via simplified I^2C, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) with the port output mode register (POMxx). For details, see **4.3 Registers Controlling Port Function** and **4.5 Register Settings When Using Alternate Function**.
When IIC00 or IIC01 is communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) also for the clock input/output pins (SCL00, SCL01).
For details, see **4.4.4 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers**.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 33** or **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

(1) Register setting

Figure 19 - 138 Example of Contents of Registers for Data Transmission of Simplified I2C (IIC00, IIC01)

(a) Serial mode register mn (SMRmn)... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0 Note 1	0	SISmn0 0 Note 1	1	0	0	MDmn2 1	MDmn1 0	MDmn0 0

(b) Serial communication operation setting register mn (SCRmn)... Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 1	RXEmn 0	DAPmn 0	CKPmn 0	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0	0	SLCmn1 0 Note 2	SLCmn0 1	0	1	DLSmn1 1	DLSmn0 1

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)... Only the lower 8 bits (SIOr) are valid during data transmission/reception..

	15–9	8	7–0
SDRmn	Baud rate setting Note 3	0	Transmit data setting (SIOr)

(d) Serial output register m (SOm)... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	CKOm2 ×	CKOm1 0/1 Note 4	CKOm0 0/1 Note 4	0	0	0	0	0	SOm2 ×	SOm1 0/1 Note 4	SOm0 0/1 Note 4

(e) Serial output enable register m (SOEm)... Do not manipulate this register during data transmission/reception

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	SOEm1 1	SOEm0 1

(f) Serial channel start register m (SSm)... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 0/1	SSm0 0/1

Note 1. Only provided for the SMR01 register.
Note 2. Only provided for the SCR00 register.
Note 3. Because the setting is completed by address field transmission, setting is not required.
Note 4. The value varies depending on the communication data during communication operation.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01), mn = 00, 01

Remark 2. ■ (bold box): Setting is fixed in the IIC mode,
■ (gray): 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 details

Figure 19 - 139 Timing Chart of Data Transmission

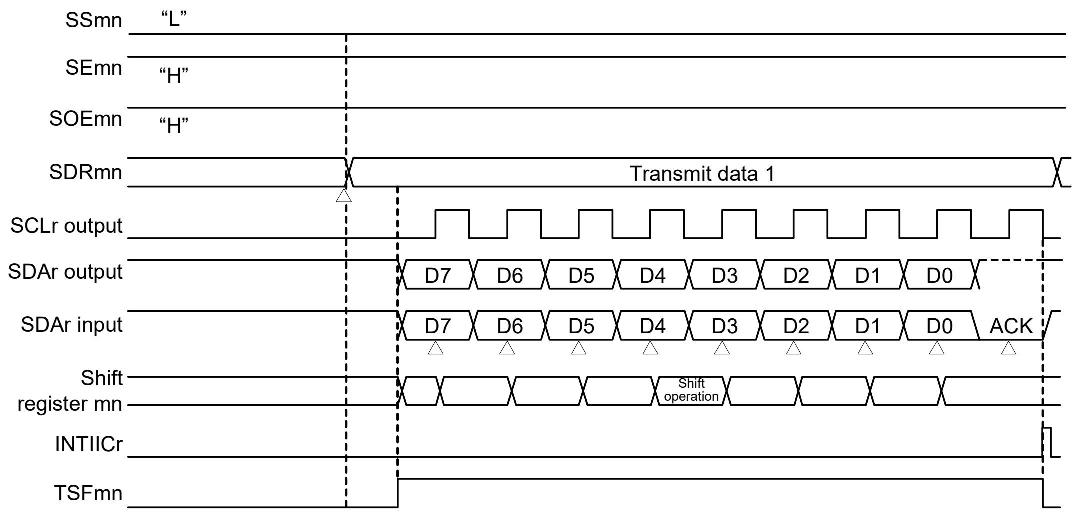
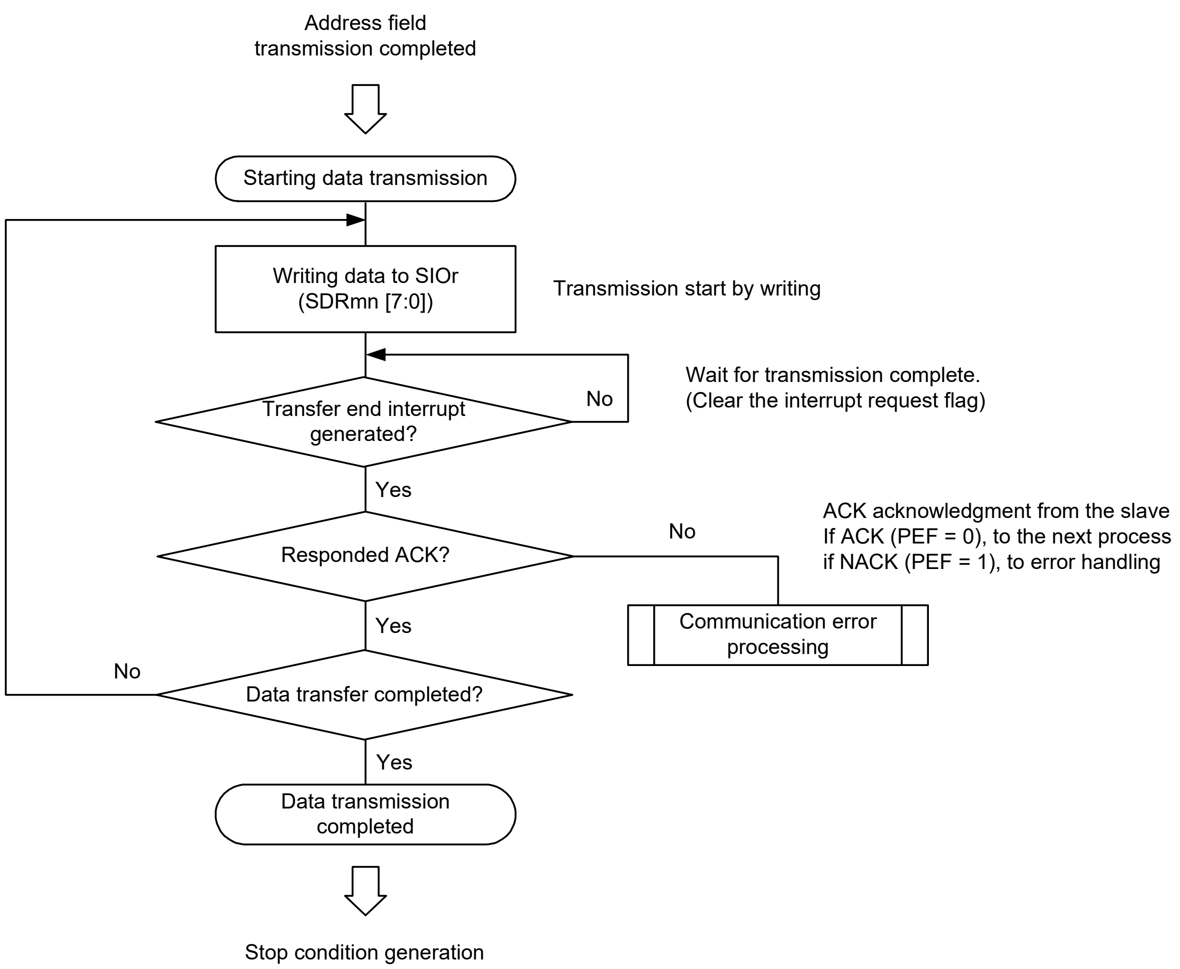


Figure 19 - 140 Flowchart of Data Transmission



19.9.3 Data reception

Data reception is an operation to receive data to the target for transfer (slave) after transmission of an address field. After all data are received to the slave, a stop condition is generated and the bus is released.

Simplified I^2C	IIC00	IIC01
Target channel	Channel 0 of SAU0	Channel 1 of SAU0
Pins used	SCL00, SDA00[Note 1]	SCL01, SDA01[Note 1]
Interrupt	INTIIC00	INTIIC01
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	Overrun error detection flag (OVFmn) only	
Transfer data length	8 bits	
Transfer rate[Note 2]	Max. f_{MCK}/4 [Hz] (SDRmn[15:9] = 1 or more) f_{MCK}: Operation clock frequency of target channel However, the following condition must be satisfied in each mode of I^2C. • Max. 1 MHz (fast mode plus) • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)	
Data level	Non-reverse output (default: high level)	
Parity bit	No parity bit	
Stop bit	Appending 1 bit (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**.
When IIC00 or IIC01 is communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POMxx = 1) also for the clock input/output pins (SCL00, SCL01).
For details, see **4.4.4 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers**.

Note 2. Use this operation within a range that satisfies the conditions above and the peripheral functions characteristics in the electrical specifications (see **CHAPTER 33** or **CHAPTER 34 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

(1) Register setting

Figure 19 - 141 Example of Contents of Registers for Data Reception of Simplified I²C (IIC00, IIC01)

(a) Serial mode register mn (SMRmn)... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMRmn	CKSmn 0/1	CCSmn 0	0	0	0	0	0	STSmn 0 Note 1	0	SISmn0 0 Note 1	1	0	0	MDmn2 1	MDmn1 0	MDmn0 0

(b) Serial communication operation setting register mn (SCRmn)... Do not manipulate the bits of this register, except the TXEmn and RXEmn bits, during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0	CKPmn 0	0	EOCmn 0	PTCmn1 0	PTCmn0 0	DIRmn 0	0	SLCmn1 0 Note 2	SLCmn0 1	0	1	DLSmn1 1	DLSmn0 1

(c) Serial data register mn (SDRmn) (lower 8 bits: SIOr)

	15–9	8	7–0
SDRmn	Baud rate setting Note 3	0	Dummy transmit data setting (FFH)
			SIOr

(d) Serial output register m (SOm)... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	0	CKOm2 ×	CKOm1 0/1 Note 4	CKOm0 0/1 Note 4	0	0	0	0	0	SOm2 ×	SOm1 0/1 Note 4	SOm0 0/1 Note 4

(e) Serial output enable register m (SOEm)... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 ×	SOEm1 0/1	SOEm0 0/1

(f) Serial channel start register m (SSm)... Do not manipulate this register during data transmission/reception.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 ×	SSm1 0/1	SSm0 0/1

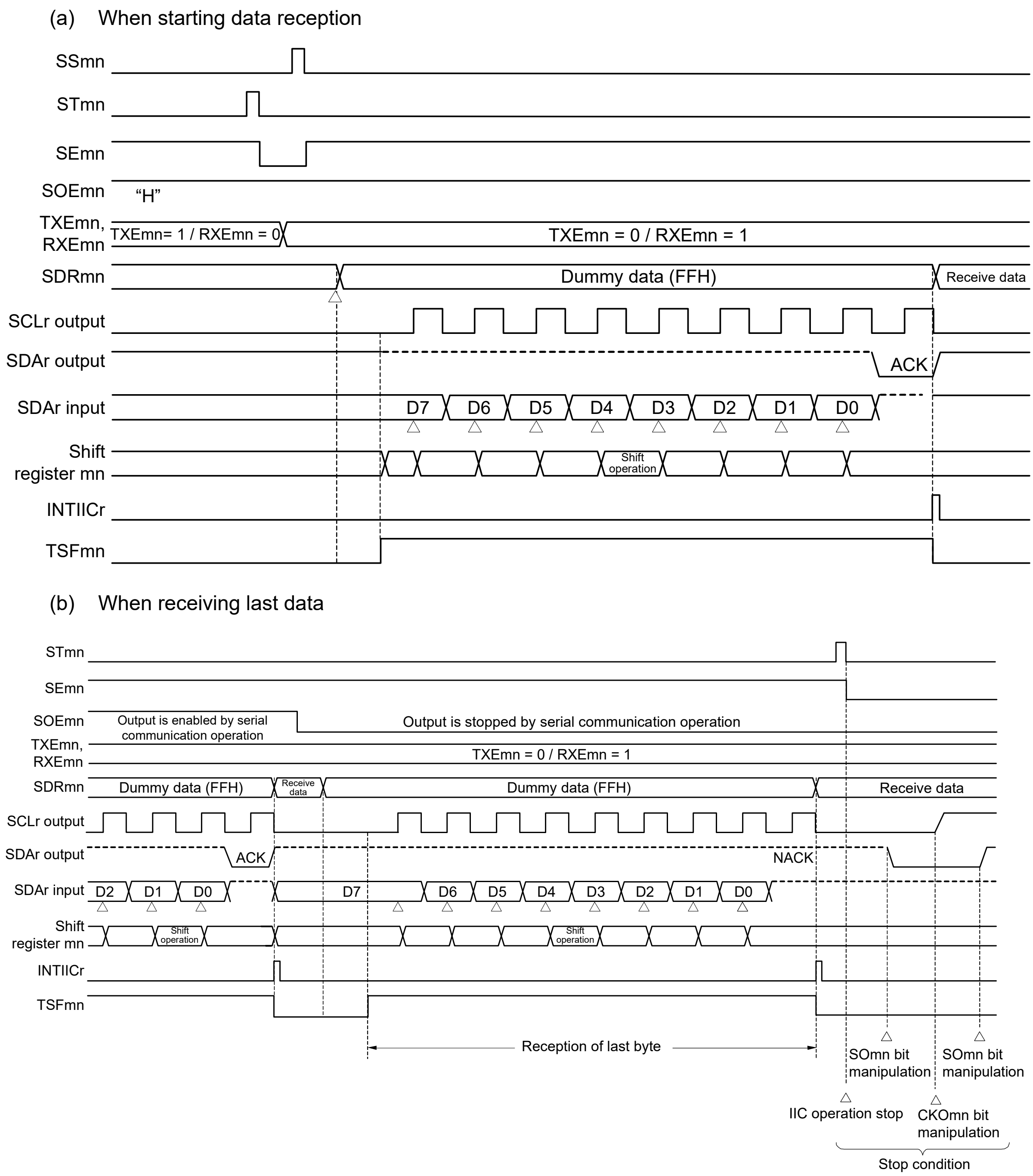
Note 1. Only provided for the SMR01 register.
Note 2. Only provided for the SCR00 register.
Note 3. The baud rate setting is not required because the baud rate has already been set when the address field was transmitted.
Note 4. The value varies depending on the communication data during communication operation.

Remark 1. m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01), mn = 00, 01

Remark 2. □: Setting is fixed in the IIC mode,
■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

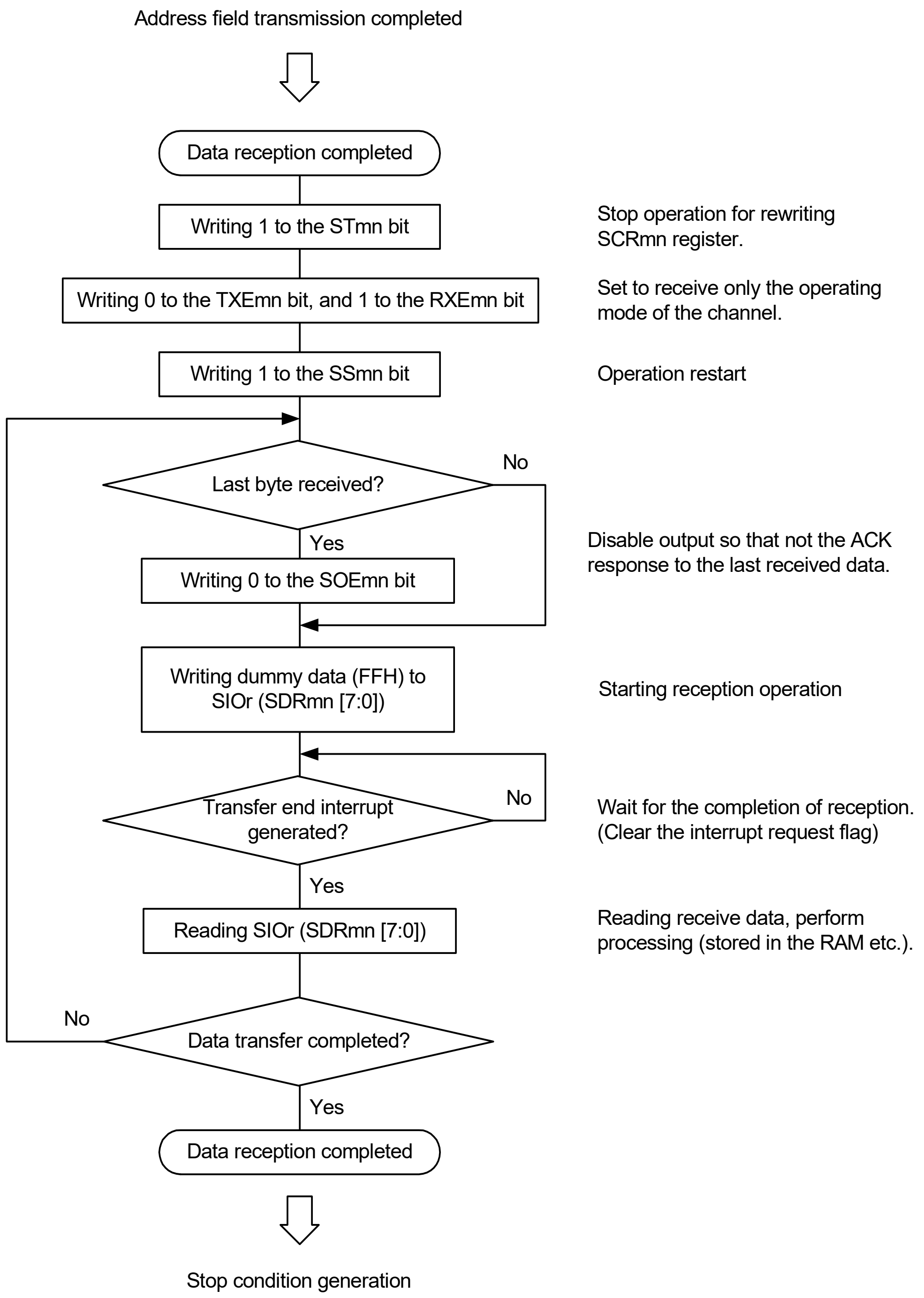
(2) Processing details

Figure 19 - 142 Timing Chart of Data Reception



Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01), mn = 00, 01

Figure 19 - 143 Flowchart of Data Reception



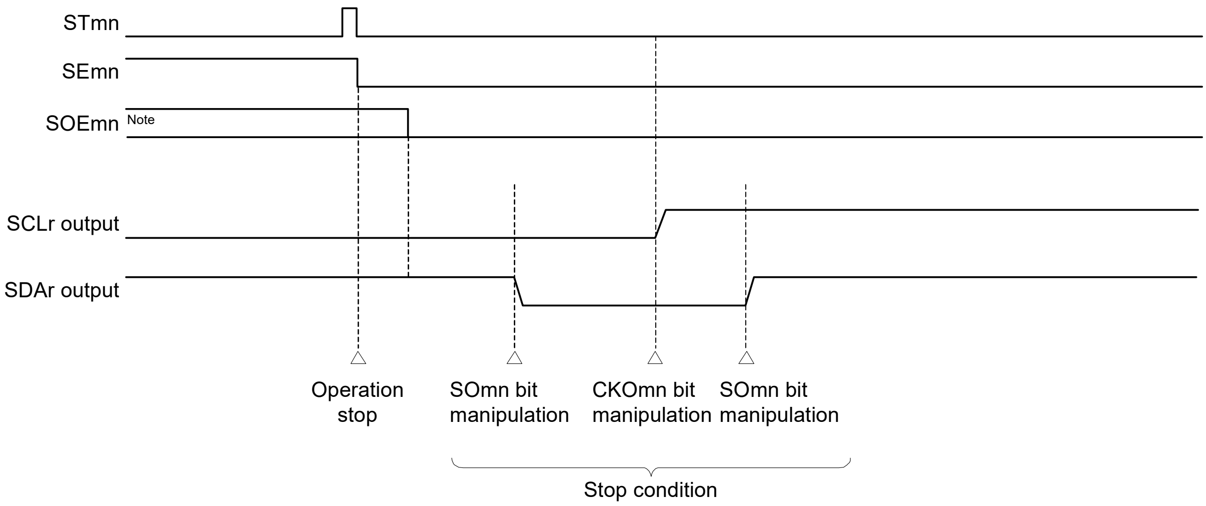
Caution **ACK is not output when the last data is received (NACK). Communication is then completed by setting “1” to the STmn bit of serial channel stop register m (STm) to stop operation and generating a stop condition.**

19.9.4 Stop condition generation

After all data are transmitted to or received from the target slave, a stop condition is generated and the bus is released.

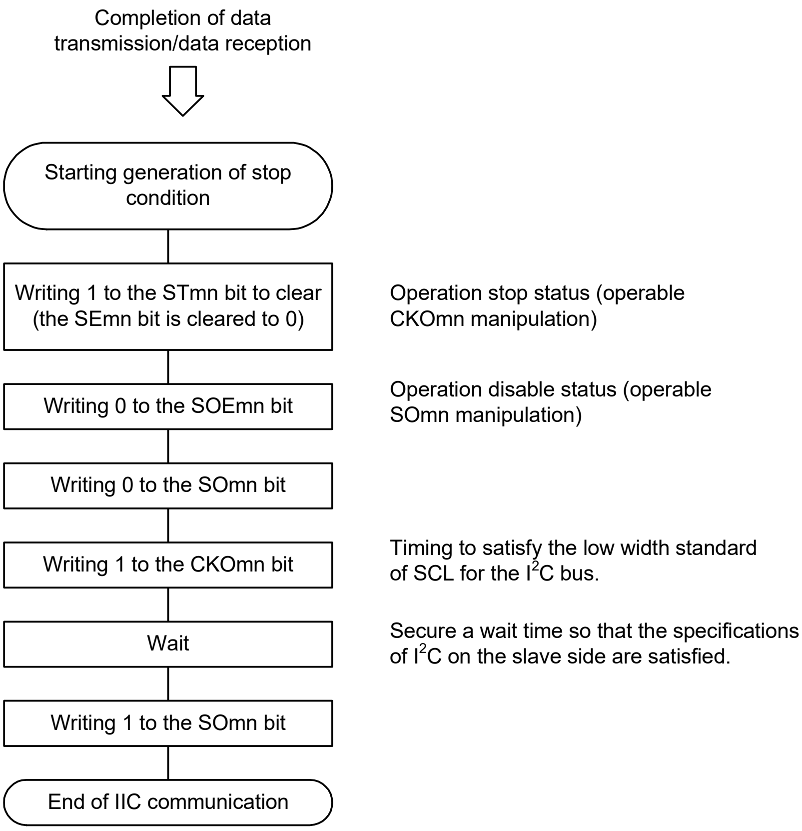
(1) Processing details

Figure 19 - 144 Timing Chart of Stop Condition Generation



Note During a reception, the SOEmn bit of serial output enable register m (SOEm) is cleared to 0 before receiving the last data.

Figure 19 - 145 Flowchart of Stop Condition Generation



19.9.5 Calculating transfer rate

The transfer rate for simplified I²C (IIC00, IIC01) communication can be calculated by the following expressions.

Transfer rate = {Operation clock (f_{MCK}) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2

Caution SDRmn[15:9] must not be set to 0000000B. Be sure to set a value of 0000001B or greater for SDRmn[15:9]. The duty ratio of the SCL signal output by the simplified I²C is 50%. The I²C bus specifications define that the low-level width of the SCL signal is longer than the high-level width. If 400 kbps (fast mode) or 1 Mbps (fast mode plus) is specified, therefore, the low-level width of the SCL output signal becomes shorter than the value specified in the I²C bus specifications. Make sure that the SDRmn[15:9] value satisfies the I²C bus specifications.

Remark 1. The value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000001B to 1111111B) and therefore is 1 to 127.

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

The operation clock (f_{MCK}) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 19 - 7 Selection of Operation Clock for Simplified I2C

SMRmn Register	SPSm Register								Operation Clock (f_{MCK}) Note	
CKSmn	PRSm13	PRSm12	PRSm11	PRSm10	PRSm03	PRSm02	PRSm01	PRSm00		f_{CLK} = 32 MHz
0	×	×	×	×	0	0	0	0	f_{CLK}	32 MHz
	×	×	×	×	0	0	0	1	$f_{CLK}/2$	16 MHz
	×	×	×	×	0	0	1	0	$f_{CLK}/2^2$	8 MHz
	×	×	×	×	0	0	1	1	$f_{CLK}/2^3$	4 MHz
	×	×	×	×	0	1	0	0	$f_{CLK}/2^4$	2 MHz
	×	×	×	×	0	1	0	1	$f_{CLK}/2^5$	1 MHz
	×	×	×	×	0	1	1	0	$f_{CLK}/2^6$	500 kHz
	×	×	×	×	0	1	1	1	$f_{CLK}/2^7$	250 kHz
	×	×	×	×	1	0	0	0	$f_{CLK}/2^8$	125 kHz
	×	×	×	×	1	0	0	1	$f_{CLK}/2^9$	62.5 kHz
	×	×	×	×	1	0	1	0	$f_{CLK}/2^{10}$	31.25 kHz
	×	×	×	×	1	0	1	1	$f_{CLK}/2^{11}$	15.63 kHz
1	0	0	0	0	×	×	×	×	f_{CLK}	32 MHz
	0	0	0	1	×	×	×	×	$f_{CLK}/2$	16 MHz
	0	0	1	0	×	×	×	×	$f_{CLK}/2^2$	8 MHz
	0	0	1	1	×	×	×	×	$f_{CLK}/2^3$	4 MHz
	0	1	0	0	×	×	×	×	$f_{CLK}/2^4$	2 MHz
	0	1	0	1	×	×	×	×	$f_{CLK}/2^5$	1 MHz
	0	1	1	0	×	×	×	×	$f_{CLK}/2^6$	500 kHz
	0	1	1	1	×	×	×	×	$f_{CLK}/2^7$	250 kHz
	1	0	0	0	×	×	×	×	$f_{CLK}/2^8$	125 kHz
	1	0	0	1	×	×	×	×	$f_{CLK}/2^9$	62.5 kHz
	1	0	1	0	×	×	×	×	$f_{CLK}/2^{10}$	31.25 kHz
	1	0	1	1	×	×	×	×	$f_{CLK}/2^{11}$	15.63 kHz
Other than above									Setting prohibited	

Note When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (serial channel stop register m (STm) = 000FH) the operation of the serial array unit (SAU).

Remark 1. ×: Don’t care

Remark 2. m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01

Here is an example of setting an I²C transfer rate where $f_{MCK} = f_{CLK} = 32$ MHz.

I²C Transfer Mode (Desired Transfer Rate)	f_{CLK} = 32 MHz			
	Operation Clock (f_{MCK})	SDRmn[15:9]	Calculated Transfer Rate	Error from Desired Transfer Rate
100 kHz	f_{CLK}/2	79	100 kHz	0.0%
400 kHz	f_{CLK}	41	380 kHz	5.0% Note
1 MHz	f_{CLK}	18	0.84 MHz	16.0% Note

Note The error cannot be set to about 0% because the duty ratio of the SCL signal is 50%.

19.9.6 Procedure for processing errors that occurred during simplified I²C (IIC00, IIC01) communication

The procedure for processing errors that occurred during simplified I²C (IIC00, IIC01) communication is described in **Figures 19 - 146** and **19 - 147**.

Figure 19 - 146 Processing Procedure in Case of Overrun Error

Software Manipulation	Hardware Status	Remark
Reads serial data register mn (SDRmn). →	The BFFmn bit of the SSRmn register is set to 0 and channel n is enabled to receive data.	This is to prevent an overrun error if the next reception is completed during error processing.
Reads serial status register mn (SSRmn).		The error type is identified and the read value is used to clear the error flag.
Writes 1 to serial flag clear trigger register mn (SIRmn). →	The error flag is cleared.	The error only during reading can be cleared, by writing the value read from the SSRmn register to the SIRmn register without modification.

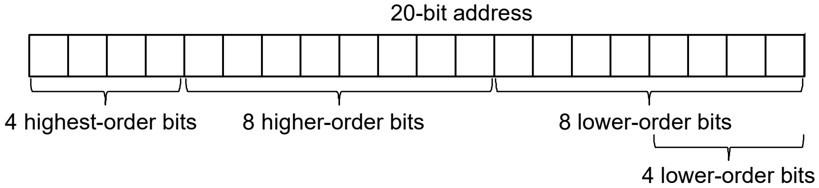
Figure 19 - 147 Processing Procedure in Case of ACK error in Simplified I²C Mode

Software Manipulation	Hardware Status	Remark
Reads serial status register mn (SSRmn).		Error type is identified and the read value is used to clear error flag.
Writes serial flag clear trigger register mn (SIRmn). →	Error flag is cleared.	Error can be cleared only during reading, by writing the value read from the SSRmn register to the SIRmn register without modification.
Sets the STmn bit of serial channel stop register m (STm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 0 and channel n stops operation.	Slave is not ready for reception because ACK is not returned. Therefore, a stop condition is created, the bus is released, and communication is started again from the start condition. Or, a restart condition is generated and transmission can be redone from address transmission.
Creates stop condition.		
Creates start condition.		
Sets the SSmn bit of serial channel start register m (SSm) to 1. →	The SEmn bit of serial channel enable status register m (SEm) is set to 1 and channel n is enabled to operate.	

Remark m: Unit number (m = 0), n: Channel number (n = 0, 1), r: IIC number (r = 00, 01), mn = 00, 01

CHAPTER 20 DATA TRANSFER CONTROLLER (DTC)

The term “8 higher-order bits of the address” in this chapter indicates bits 15 to 8 of 20-bit address as shown below.



Unless otherwise specified, the 4 highest-order address bits all become 1 (values are of the form FxxxxH).

20.1 Functions of DTC

The data transfer controller (DTC) is a function that transfers data between memories without using the CPU. The DTC is activated by a peripheral function interrupt to perform data transfers. The DTC and CPU use the same bus, and the DTC takes priority over the CPU in using the bus.

Table 20 - 1 lists the DTC Specifications.

Table 20 - 1 DTC Specifications

Item		Specification
Activation sources		22 sources
Allocatable control data		24 sets
Address space which can be transferred	Address space	64 Kbytes (F0000H to FFFFFH), excluding general-purpose registers
	Sources	Special function register (SFR), RAM area (excluding general-purpose registers), mirror area [Note], data flash memory area [Note], extended special function register (2nd SFR)
	Destinations	Special function register (SFR), RAM area (excluding general-purpose registers), extended special function register (2nd SFR)
Maximum number of transfers	Normal mode	256 times
	Repeat mode	255 times
Maximum size of block to be transferred	Normal mode (8-bit transfer)	256 bytes
	Normal mode (16-bit transfer)	512 bytes
	Repeat mode	255 bytes
Unit of transfers		8 bits/16 bits
Transfer mode	Normal mode	Transfers end on completion of the transfer causing the DTCCTj register value to change from 1 to 0.
	Repeat mode	On completion of the transfer causing the DTCCTj register value to change from 1 to 0, the repeat area address is initialized and the DTRLDj register value is reloaded to the DTCCTj register to continue transfers.
Address control	Normal mode	Fixed or incremented
	Repeat mode	Addresses of the area not selected as the repeat area are fixed or incremented.
Priority of activation sources		Refer to **Table 20 - 5 DTC Activation Sources and Vector Addresses**.
Interrupt request	Normal mode	When the data transfer causing the DTCCTj register value to change from 1 to 0 is performed, the activation source interrupt request is generated for the CPU, and interrupt handling is performed on completion of the data transfer.
	Repeat mode	When the data transfer causing the DTCCTj register value to change from 1 to 0 is performed while the RPTINT bit in the DTCCRj register is 1 (interrupt generation enabled), the activation source interrupt request is generated for the CPU, and interrupt handling is performed on completion of the transfer.
Transfer start		When bits DTCENi0 to DTCENi7 in the DTCENi registers are 1 (activation enabled), data transfer is started each time the corresponding DTC activation sources are generated.
Transfer stop	Normal mode	• When bits DTCENi0 to DTCENi7 are set to 0 (activation disabled). • When the data transfer causing the DTCCTj register value to change from 1 to 0 is completed.
	Repeat mode	• When bits DTCENi0 to DTCENi7 are set to 0 (activation disabled). • When the data transfer causing the DTCCTj register value to change from 1 to 0 is completed while the RPTINT bit is 1 (interrupt generation enabled).

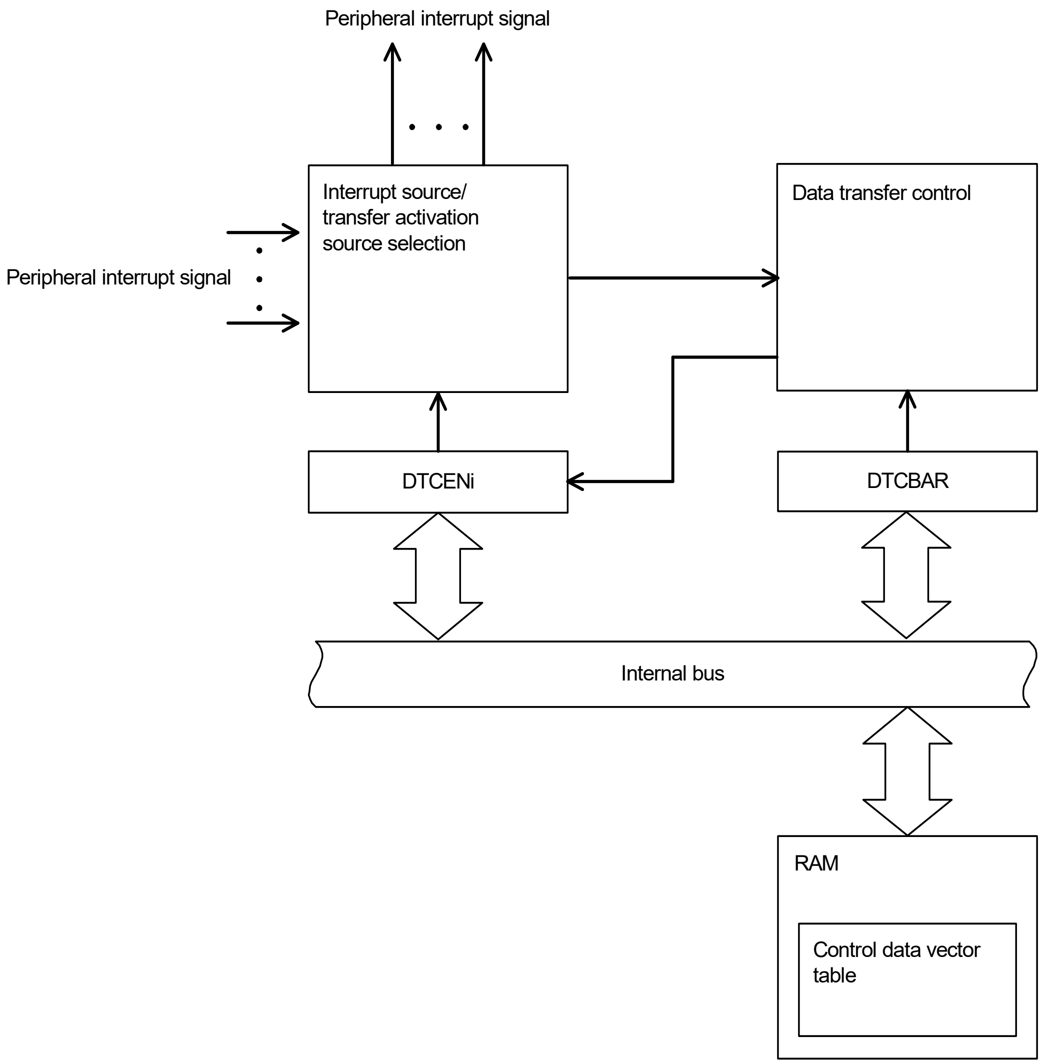
Note In the HALT mode or SNOOZE mode, these areas cannot be set as the sources for DTC transfer since the flash memory is stopped.

Remark i = 0 to 2, j = 0 to 23

20.2 Configuration of DTC

Figure 20 - 1 shows the DTC Block Diagram.

Figure 20 - 1 DTC Block Diagram



20.3 Registers Controlling DTC

Table 20 - 2 lists the Registers Controlling DTC.

Table 20 - 2 Registers Controlling DTC

Register Name	Symbol
Peripheral enable register 1	PER1
DTC activation enable register 0	DTCEN0
DTC activation enable register 1	DTCEN1
DTC activation enable register 2	DTCEN2
DTC base address register	DTCBAR

Table 20 - 3 lists DTC Control Data.

DTC control data is allocated in the DTC control data area in RAM.

The DTCBAR register is used to set the 256-byte area, including the DTC control data area and the DTC vector table area where the start address for control data is stored.

Table 20 - 3 DTC Control Data

Register Name	Symbol
DTC Control Register j	DTCCRj
DTC Block Size Register j	DTBLSj
DTC Transfer Count Register j	DTCCTj
DTC Transfer Count Reload Register j	DTRLDj
DTC Source Address Register j	DTSARj
DTC Destination Address Register j	DTDARj

Remark j = 0 to 23

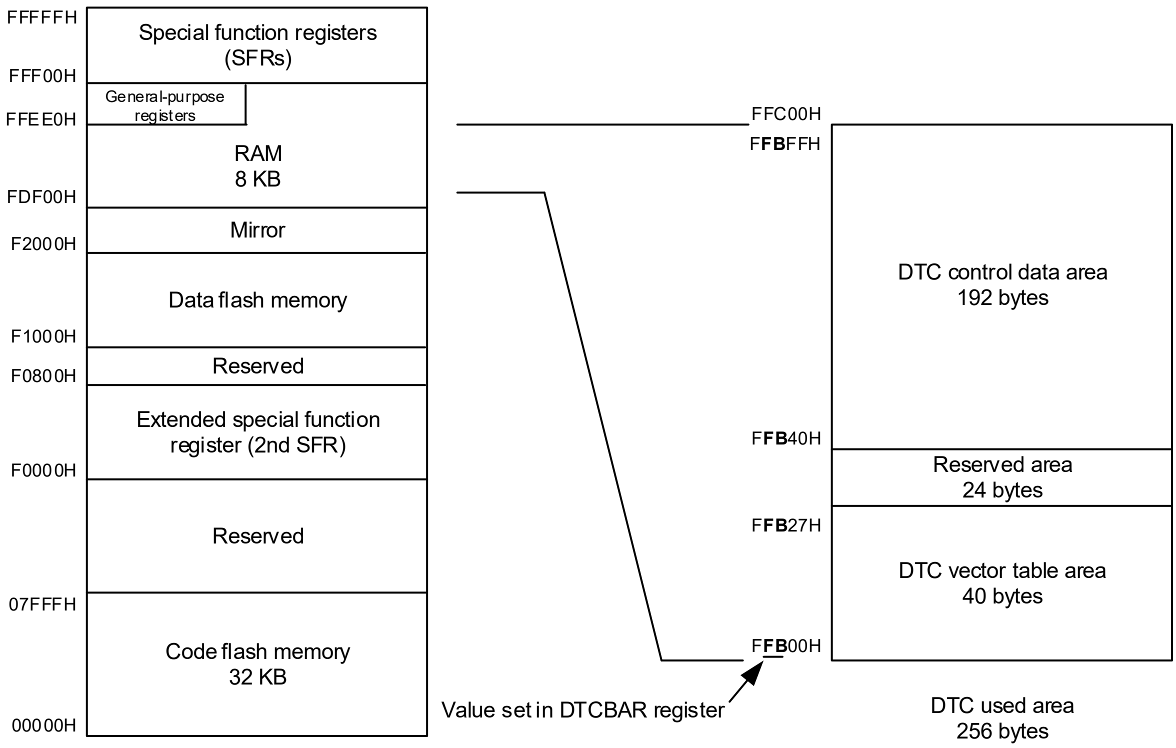
20.3.1 Allocation of DTC control data area and DTC vector table area

The DTCBAR register is used to set the 256-byte area where DTC control data and the vector table within the RAM area.

Figure 20 - 2 shows a Memory Map Example when DTCBAR Register is Set to FBH.

In the 192-byte DTC control data area, the space not used by the DTC can be used as RAM.

Figure 20 - 2 Memory Map Example when DTCBAR Register is Set to FBH



Caution 1. It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space as the DTC control data area or DTC vector table area.

Caution 2. Make sure the stack area, the DTC control data area, and the DTC vector table area do not overlap.

Caution 3. The internal RAM area in FDF00H to FE309H cannot be used as the DTC control data area or DTC vector table area when using the self-programming and data flash functions.

Caution 4. The internal RAM area in FE300H to FE6FFH cannot be used as the DTC control data area or DTC vector table area when using the on-chip debugging trace function.

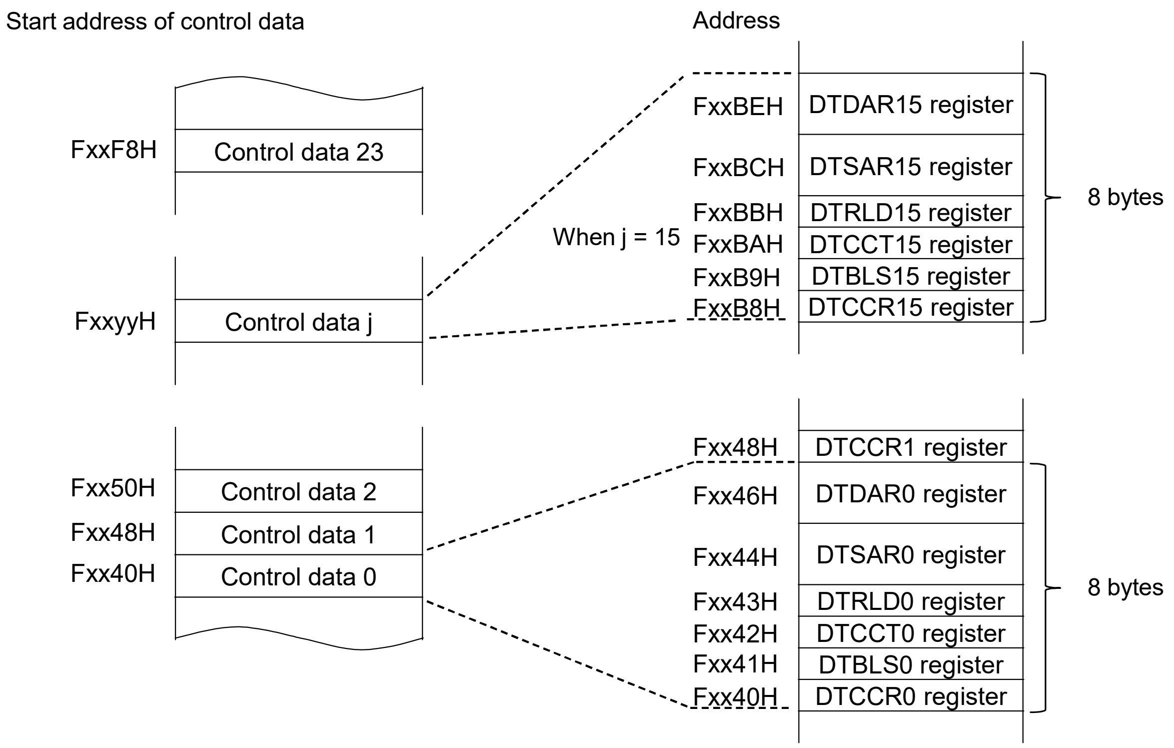
20.3.2 Control data allocation

Control data is allocated beginning with each start address in the order: Registers DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, and DTDARj (j = 0 to 23).
The higher 8 bits for start addresses 0 to 23 are set by the DTCBAR register, and the lower 8 bits are separately set according to the vector table assigned to each activation source.
Figure 20 - 3 shows Control Data Allocation.

Note 1. Change the data in registers DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, and DTDARj when the corresponding bit among bits DTCENi0 to DTCENi7 (i = 0 to 2) in the DTCENi register is set to 0 (activation disabled).

Note 2. Do not access DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, or DTDARj using a DTC transfer.

Figure 20 - 3 Control Data Allocation



Remark xx: Value set in DTCBAR register

Table 20 - 4 Start Address of Control Data

j	Address	j	Address
11	Fxx98H	23	FxxF8H
10	Fxx90H	22	FxxF0H
9	Fxx88H	21	FxxE8H
8	Fxx80H	20	FxxE0H
7	Fxx78H	19	FxxD8H
6	Fxx70H	18	FxxD0H
5	Fxx68H	17	FxxC8H
4	Fxx60H	16	FxxC0H
3	Fxx58H	15	FxxB8H
2	Fxx50H	14	FxxB0H
1	Fxx48H	13	FxxA8H
0	Fxx40H	12	FxxA0H

Remark xx: Value set in DTCBAR register

20.3.3 Vector table

When the DTC is activated, one control data is selected according to the data read from the vector table which has been assigned to each activation source, and the selected control data is read from the DTC control data area.

Table 20 - 5 lists the DTC Activation Sources and Vector Addresses. A one byte of the vector table is assigned to each activation source, and data from 40H to F8H is stored in each area to select one of the 24 control data sets. The higher 8 bits for the vector address are set by the DTCBAR register, and 00H to 16H are allocated to the lower 8 bits corresponding to the activation source.

Note Change the start address of the DTC control data area to be set in the vector table when the corresponding bit among bits DTCENi0 to DTCENi7 (i = 0 to 2) in the DTCENi register is set to 0 (activation disabled).

Figure 20 - 4 Start Address of Control Data and Vector Table

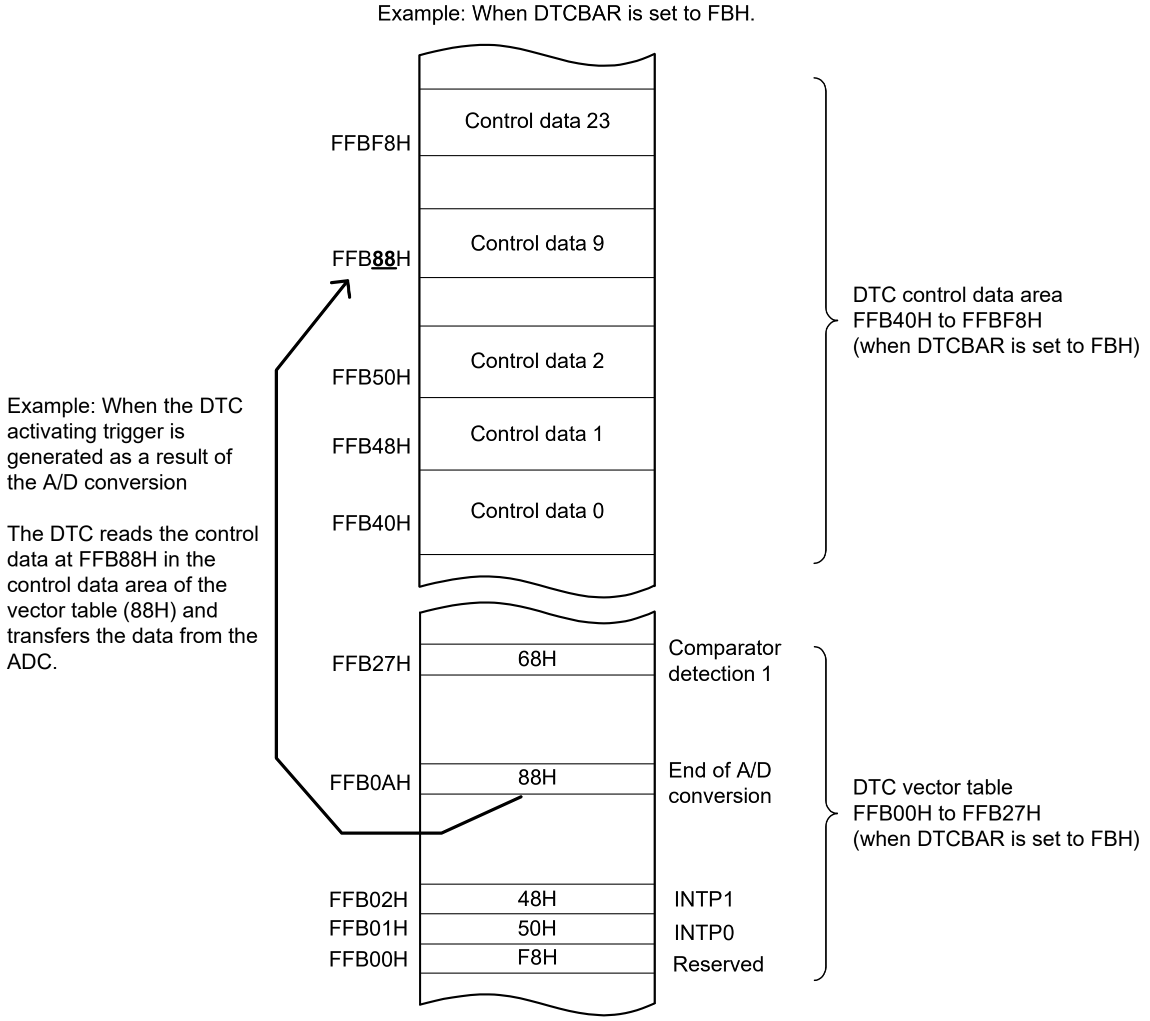


Table 20 - 5 DTC Activation Sources and Vector Addresses

DTC Activation Sources (Interrupt Request Source)	Source No.	Vector Address	Priority
Reserved	0	Address set in DTCBAR register +00H	Highest
INTP0	1	Address set in DTCBAR register +01H	↑
INTP1	2	Address set in DTCBAR register +02H	
INTP2	3	Address set in DTCBAR register +03H	
INTP3	4	Address set in DTCBAR register +04H	
INTP4	5	Address set in DTCBAR register +05H	
INTP5	6	Address set in DTCBAR register +06H	
INTP6	7	Address set in DTCBAR register +07H	
INTP7	8	Address set in DTCBAR register +08H	
A/D conversion end	9	Address set in DTCBAR register +09H	
UART0 reception transfer end/CSI01 transfer end or buffer empty/IIC01 transfer end	10	Address set in DTCBAR register +0AH	
UART0 transmission transfer end/CSI00 transfer end or buffer empty/IIC00 transfer end	11	Address set in DTCBAR register +0BH	
UART1 reception transfer end	12	Address set in DTCBAR register +0CH	
UART1 transmission transfer end	13	Address set in DTCBAR register +0DH	
End of channel 0 of timer array unit 0 count or capture	14	Address set in DTCBAR register +0EH	
End of channel 1 of timer array unit 0 count or capture	15	Address set in DTCBAR register +0FH	
End of channel 2 of timer array unit 0 count or capture	16	Address set in DTCBAR register +10H	
End of channel 3 of timer array unit 0 count or capture	17	Address set in DTCBAR register +11H	
End of channel 0 of timer array unit 1 count or capture	18	Address set in DTCBAR register +12H	
End of channel 1 of timer array unit 1 count or capture	19	Address set in DTCBAR register +13H	
Timer RG compare match A	20	Address set in DTCBAR register +14H	
Timer RG compare match B	21	Address set in DTCBAR register +15H	↓
Timer RJ0 underflow	22	Address set in DTCBAR register +16H	Lowest

20.3.4 Peripheral enable register 1 (PER1)

The PER1 register is used to enable or disable supplying the clock to the peripheral hardware. Clock supply to the hardware that is not used is also stopped so as to decrease the power consumption and noise.
When using the DTC, be sure to set bit 3 (DTCEN) to 1.

The PER1 register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 20 - 5 Format of Peripheral enable register 1 (PER1)

Address: F007AH After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	<3>	<2>	<1>	<0>
PER1	DACEN	TRGEN	AMPEN	0	DTCEN	PGAEN	AFEEN	TRJ0EN

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

Caution Be sure to set bit 4 to “0”.

20.3.5 DTC control register j (DTCCRj) (j = 0 to 23)

The DTCCRj register is used to control the DTC operating mode.

Figure 20 - 6 Format of DTC control register j (DTCCRj)

Address: Refer to **20.3.2 Control data allocation**. After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
DTCCRj	0	SZ	RPTINT	CHNE	DAMOD	SAMOD	RPTSEL	MODE

SZ	Transfer Data size selection
0	8 bits
1	16 bits

RPTINT	Enabling/disabling repeat mode interrupts
0	Interrupt generation disabled
1	Interrupt generation enabled
The setting of the RPTINT bit is invalid when the MODE bit is 0 (normal mode).	

CHNE	Enabling/disabling chain transfers
0	Chain transfers disabled
1	Chain transfers enabled
Set the CHNE bit in the DTCCR23 register to 0 (chain transfers disabled).	

DAMOD	Transfer destination address control
0	Fixed
1	Incremented
The setting of the DAMOD bit is invalid when the MODE bit is 1 (repeat mode) and the RPTSEL bit is 0 (transfer destination is the repeat area).	

SAMOD	Transfer source address control
0	Fixed
1	Incremented
The setting of the SAMOD bit is invalid when the MODE bit is 1 (repeat mode) and the RPTSEL bit is 1 (transfer source is the repeat area).	

RPTSEL	Repeat area selection
0	Transfer destination is the repeat area
1	Transfer source is the repeat area
The setting of the RPTSEL bit is invalid when the MODE bit is 0 (normal mode).	

MODE	Transfer mode selection
0	Normal mode
1	Repeat mode

Caution Do not access the DTCCRj register using a DTC transfer.

20.3.6 DTC block size register j (DTBLSj) (j = 0 to 23)

This register is used to set the block size of the data to be transferred by one activation.

Figure 20 - 7 Format of DTC block size register j (DTBLSj)

Address: Refer to **20.3.2 Control data allocation**. After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
DTBLSj	DTBLSj7	DTBLSj6	DTBLSj5	DTBLSj4	DTBLSj3	DTBLSj2	DTBLSj1	DTBLSj0

DTBLSj	Transfer Block Size	
	8-Bit Transfer	16-Bit Transfer
00H	256 bytes	512 bytes
01H	1 byte	2 bytes
02H	2 bytes	4 bytes
03H	3 bytes	6 bytes
• • •	• • •	• • •
FDH	253 bytes	506 bytes
FEH	254 bytes	508 bytes
FFH	255 bytes	510 bytes

Caution Do not access the DTBLSj register using a DTC transfer.

20.3.7 DTC transfer count register j (DTCCTj) (j = 0 to 23)

This register is used to set the number of DTC data transfers. The value is decremented by 1 each time DTC transfer is activated once.

Figure 20 - 8 Format of DTC transfer count register j (DTCCTj)

Address: Refer to **20.3.2 Control data allocation**. After reset: Undefined R/W

Symbol	7	6	5	4	3	2	1	0
DTCCTj	DTCCTj7	DTCCTj6	DTCCTj5	DTCCTj4	DTCCTj3	DTCCTj2	DTCCTj1	DTCCTj0

DTCCTj	Number of Transfers
00H	256 times
01H	Once
02H	2 times
03H	3 times
• • •	• • •
FDH	253 times
FEH	254 times
FFH	255 times

Caution Do not access the DTCCTj register using a DTC transfer.

20.3.8 DTC transfer count reload register j (DTRLDj) (j = 0 to 23)

This register is used to set the initial value of the transfer count register in repeat mode. Since the value of this register is reloaded to the DTCCT register in repeat mode, set the same value as the initial value of the DTCCT register.

Figure 20 - 9 Format of DTC transfer count reload register j (DTRLDj)

Address: Refer to **20.3.2 Control data allocation**. After reset: Undefined R/W

	7	6	5	4	3	2	1	0
DTRLDj	DTRLDj7	DTRLDj6	DTRLDj5	DTRLDj4	DTRLDj3	DTRLDj2	DTRLDj1	DTRLDj0

Caution Do not access the DTRLDj register using a DTC transfer.

20.3.9 DTC source address register j (DTSARj) (j = 0 to 23)

This register is used to specify the transfer source address for data transfer.
When the SZ bit in the DTCCRj register is set to 1 (16-bit transfer), the lowest bit is ignored and the address is handled as an even address.

Figure 20 - 10 Format of DTC source address register j (DTSARj)

Address: Refer to **20.3.2 Control data allocation**. After reset: Undefined R/W

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTSARj	DTS ARj15	DTS ARj14	DTS ARj13	DTS ARj12	DTS ARj11	DTS ARj10	DTS ARj9	DTS ARj8	DTS ARj7	DTS ARj6	DTS ARj5	DTS ARj4	DTS ARj3	DTS ARj2	DTS ARj1	DTS ARj0

Caution 1. Do not set the general-purpose register (FFEE0H to FFEFFH) space to the transfer source address.
Caution 2. Do not access the DTSARj register using a DTC transfer.

20.3.10 DTC destination address register j (DTDARj) (j = 0 to 23)

This register is used to specify the transfer destination address for data transfer.
When the SZ bit in the DTCCRj register is set to 1 (16-bit transfer), the lowest bit is ignored and the address is handled as an even address.

Figure 20 - 11 Format of DTC destination address register j (DTDARj)

Address: Refer to **20.3.2 Control data allocation**. After reset: Undefined R/W

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTDARj	DTD ARj15	DTD ARj14	DTD ARj13	DTD ARj12	DTD ARj11	DTD ARj10	DTD ARj9	DTD ARj8	DTD ARj7	DTD ARj6	DTD ARj5	DTD ARj4	DTD ARj3	DTD ARj2	DTD ARj1	DTD ARj0

Caution 1. Do not set the general-purpose register (FFEE0H to FFEFFH) space to the transfer source address.
Caution 2. Do not access the DTDARj register using a DTC transfer.

20.3.11 DTC activation enable register i (DTCENi) (i = 0 to 2)

This is an 8-bit register which enables or disables DTC activation by interrupt sources. Table 20 - 6 lists the Correspondences between Interrupt Sources and Bits DTCENi0 to DTCENi7.
The DTCENi register can be set by an 8-bit memory manipulation instruction and a 1-bit memory manipulation instruction.

Caution 1. Modify bits DTCENi0 to DTCENi7 if an activation source corresponding to the bit has not been generated.
Caution 2. Do not access the DTCENi register using a DTC transfer.
Caution 3. The assigned functions differ depending on the product. For the bits to which no function is assigned, be sure to set their values to 0.

Figure 20 - 12 Format of DTC activation enable register i (DTCENi) (i = 0 to 2)

Address: F02E8H (DTCEN0), F02E9H (DTCEN1), F02EAH (DTCEN2) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
DTCENi	DTCENi7	DTCENi6	DTCENi5	DTCENi4	DTCENi3	DTCENi2	DTCENi1	DTCENi0

DTCENi7	DTC activation enable i7
0	Activation disabled
1	Activation enabled
The DTCENi7 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi6	DTC activation enable i6
0	Activation disabled
1	Activation enabled
The DTCENi6 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi5	DTC activation enable i5
0	Activation disabled
1	Activation enabled
The DTCENi5 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi4	DTC activation enable i4
0	Activation disabled
1	Activation enabled
The DTCENi4 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi3	DTC activation enable i3
0	Activation disabled
1	Activation enabled
The DTCENi3 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi2	DTC activation enable i2
0	Activation disabled
1	Activation enabled
The DTCENi2 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi1	DTC activation enable i1
0	Activation disabled
1	Activation enabled
The DTCENi1 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

DTCENi0	DTC activation enable i0
0	Activation disabled
1	Activation enabled
The DTCENi0 bit is set to 0 (activation disabled) by a condition for generating a transfer end interrupt.	

Table 20 - 6 Correspondences between Interrupt Sources and Bits DTCENi0 to DTCENi7

Register	DTCENi7 Bit	DTCENi6 Bit	DTCENi5 Bit	DTCENi4 Bit	DTCENi3 Bit	DTCENi2 Bit	DTCENi1 Bit	DTCENi0 Bit
DTCEN0	Reserved	INTP0	INTP1	INTP2	INTP3	INTP4	INTP5	INTP6
DTCEN1	INTP7	A/D conversion end	UART0 reception transfer end/CSI01 transfer end or buffer empty/IIC01 transfer end	UART0 transmission transfer end/CSI00 transfer end or buffer empty/IIC00 transfer end	UART1 reception transfer end	UART1 transmission transfer end	End of channel 0 of timer array unit 0 count or capture	End of channel 1 of timer array unit 0 count or capture
DTCEN2	End of channel 2 of timer array unit 0 count or capture	End of channel 3 of timer array unit 0 count or capture	End of channel 0 of timer array unit 1 count or capture	End of channel 1 of timer array unit 1 count or capture	Timer RG compare match A	Timer RG compare match B	Timer RJ0 underflow	Reserved

Caution For the bits to which no function is assigned, be sure to set their values to “0”.

Remark i = 0 to 2

20.3.12 DTC base address register (DTCBAR)

This is an 8-bit register used to set the following addresses: the vector address where the start address of the DTC control data area is stored and the address of the DTC control data area. The value of the DTCBAR register is handled as the higher 8 bits to generate a 16-bit address.

Caution 1. Change the DTCBAR register value with all DTC activation sources set to activation disabled.
Caution 2. Do not rewrite the DTCBAR register more than once.
Caution 3. Do not access the DTCBAR register using a DTC transfer.
Caution 4. For the allocation of the DTC control data area and the DTC vector table area, refer to the notes on 20.3.1 Allocation of DTC control data area and DTC vector table area.

Figure 20 - 13 Format of DTC base address register (DTCBAR)

Address: F02E0H After reset: FDH R/W

Symbol	7	6	5	4	3	2	1	0
DTCBAR	DTCBAR7	DTCBAR6	DTCBAR5	DTCBAR4	DTCBAR3	DTCBAR2	DTCBAR1	DTCBAR0

20.4 DTC Operation

When the DTC is activated, control data is read from the DTC control data area to perform data transfers and control data after data transfer is written back to the DTC control data area. Twenty-four sets of control data can be stored in the DTC control data area, which allows 24 types of data transfers to be performed.
There are two transfer modes (normal mode and repeat mode) and two transfer sizes (8-bit transfer and 16-bit transfer). When the CHNE bit in the DTCCRj (j = 0 to 23) register is set to 1 (chain transfers enabled), multiple control data is read and data transfers are continuously performed by one activation source (chain transfers).
A transfer source address is specified by the 16-bit register DTSARj, and a transfer destination address is specified by the 16-bit register DTDARj.
The values in registers DTSARj and DTDARj are separately incremented or fixed according to the control data after the data transfer.

20.4.1 Activation sources

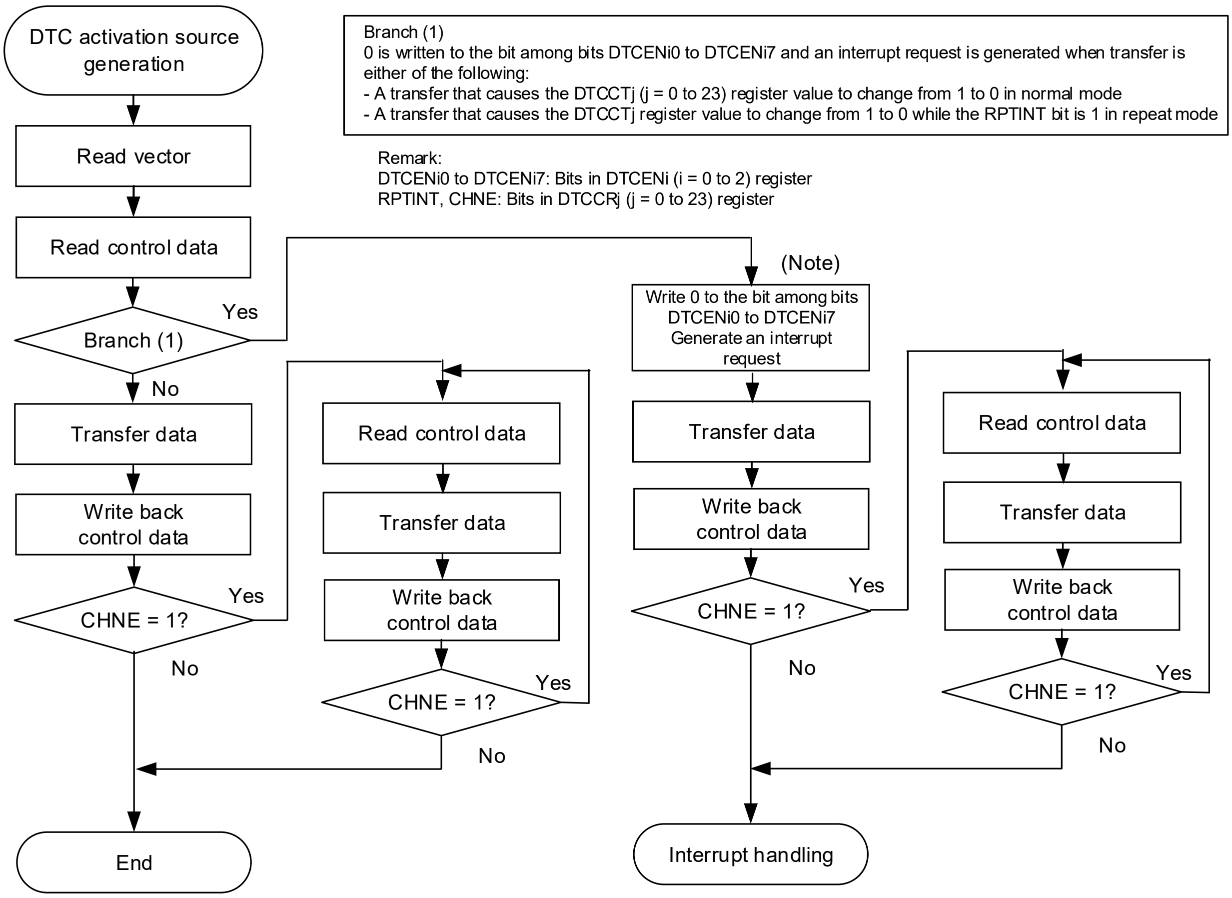
The DTC is activated by an interrupt signal from the peripheral functions. The interrupt signals to activate the DTC are selected with the DTCENi (i = 0 to 2) register.

The DTC sets the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi register to 0 (activation disabled) during operation when the setting of data transfer (the first transfer in chain transfers) is either of the following:

- A transfer that causes the DTCCTj (j = 0 to 23) register value to change to 0 in normal mode
- A transfer that causes the DTCCTj register value to change to 0 while the RPTINT bit in the DTCCRj register is 1 (interrupt generation enabled) in repeat mode

Figure 20 - 14 shows the DTC Internal Operation Flowchart.

Figure 20 - 14 DTC Internal Operation Flowchart



Note 0 is not written to the bit among bits DTCENi0 to DTCENi7 for data transfers activated by the setting to enable chain transfers (the CHNE bit is 1). Also, no interrupt request is generated.

20.4.2 Normal mode

One to 256 bytes of data are transferred by one activation during 8-bit transfer and 2 to 512 bytes during 16-bit transfer. The number of transfers can be 1 to 256 times. When the data transfer causing the DTCCTj (j = 0 to 23) register value to change to 0 is performed, the DTC generates an interrupt request corresponding to the activation source to the interrupt controller during DTC operation, and sets the corresponding bit among bits DTCENi0 to DTCENi7 (i = 0 to 2) in the DTCENi register to 0 (activation disabled).

Table 20 - 7 shows Register Functions in Normal Mode. Figure 20 - 15 shows Data Transfers in Normal Mode.

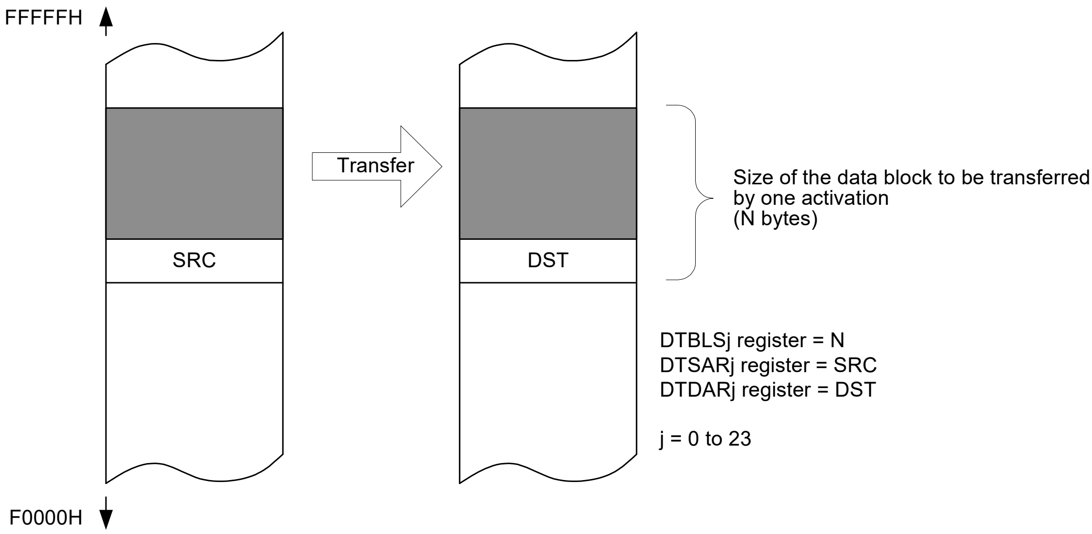
Table 20 - 7 Register Functions in Normal Mode

Register Name	Symbol	Function
DTC block size register j	DTBLSj	Size of the data block to be transferred by one activation
DTC transfer count register j	DTCCTj	Number of data transfers
DTC transfer count reload register j	DTRLDj	Not used Note
DTC source address register j	DTSARj	Data transfer source address
DTC destination address register j	DTDARj	Data transfer destination address

Note Initialize this register to 00H when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.

Remark j = 0 to 23

Figure 20 - 15 Data Transfers in Normal Mode



DTCCR Register Setting				Source Address Control	Destination Address Control	Source Address after Transfer	Destination Address after Transfer
DAMOD	SAMOD	RPTSEL	MODE				
0	0	X	0	Fixed	Fixed	SRC	DST
0	1	X	0	Incremented	Fixed	SRC + N	DST
1	0	X	0	Fixed	Incremented	SRC	DST + N
1	1	X	0	Incremented	Incremented	SRC + N	DST + N

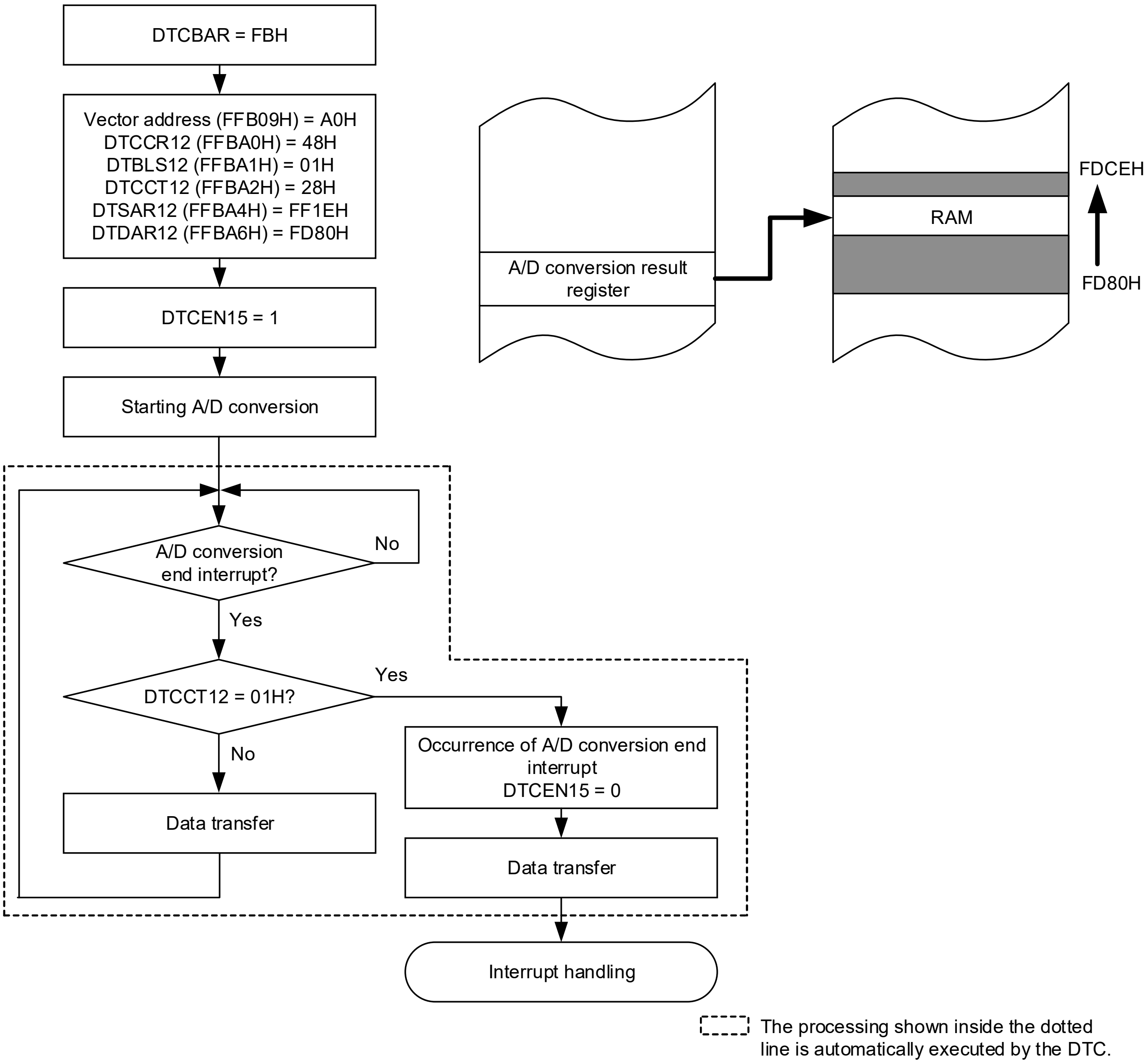
X: 0 or 1

(1) Example of using normal mode - 1: Consecutively capturing A/D conversion results

The DTC is activated by an A/D conversion end interrupt and the value of the A/D conversion result register is transferred to RAM.

- The vector address is FFB09H and control data is allocated at FFBA0H to FFBA7H
- Transfers 2-byte data of the A/D conversion result register (FFF1EH, FFF1FH) to 80 bytes of FFD80H to FFDCFH of RAM 40 times.

Figure 20 - 16 Example of using normal mode - 1: Consecutively capturing A/D conversion results



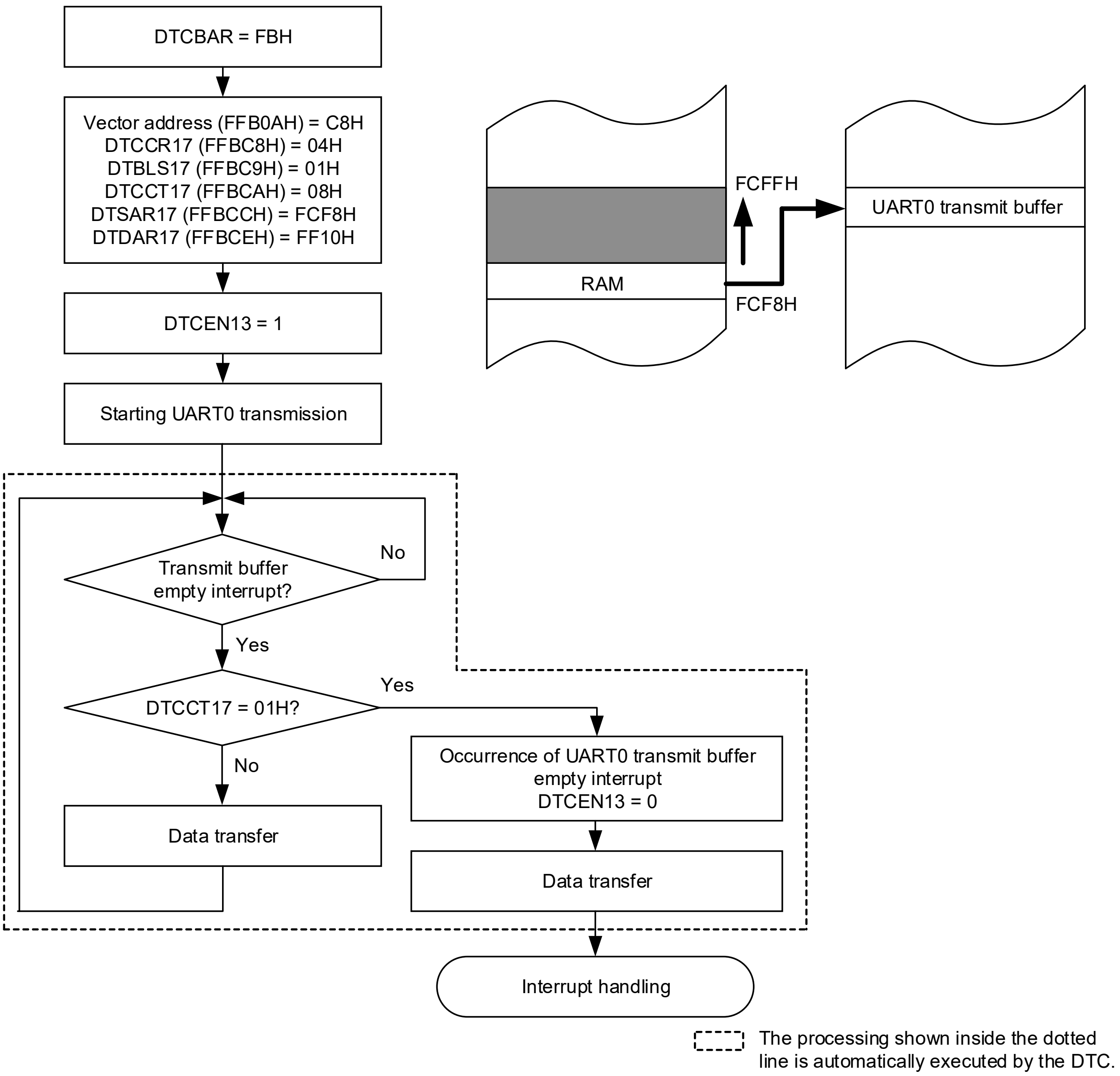
The value of the DTRLD12 register is not used because of normal mode, but initialize the register to 00H when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.

(2) Example of using normal mode - 2: UART0 consecutive transmission

The DTC is activated by a UART0 transmit buffer empty interrupt and the value of RAM is transferred to the UART0 transmit buffer.

- The vector address is FFB0AH and control data is allocated at FFBC8H to FFBCFH
- Transfers 8 bytes of FFCF8H to FFCFFH of RAM to the UART0 transmit buffer (FFF10H)

Figure 20 - 17 Example of using normal mode - 2: UART0 consecutive transmission



The value of the DTRLD17 register is not used because of normal mode, but initialize the register to 00H when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.

Start the first UART0 transmission by software. The second and subsequent transmissions are automatically sent when the DTC is activated by a transmit buffer empty interrupt.

20.4.3 Repeat mode

One to 255 bytes of data are transferred by one activation. Either of the transfer source or destination should be specified as the repeat area. The number of transfers can be 1 to 255 times. On completion of the specified number of transfers, the DTCCTj (i = 0 to 23) register and the address specified for the repeat area are initialized to continue transfers. When the data transfer causing the DTCCTj register value to change to 0 is performed while the RPTINT bit in the DTCCRj register is 1 (interrupt generation enabled), the DTC generates an interrupt request corresponding to the activation source to the interrupt controller during DTC operation, and sets the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi register (i = 0 to 2) to 0 (activation disabled). When the RPTINT bit in the DTCCRj register is 0 (interrupt generation disabled), no interrupt request is generated even if the data transfer causing the DTCCTj register value to change to 0 is performed. Also, bits DTCENi0 to DTCENi7 are not set to 0.

Table 20 - 8 lists Register Functions in Repeat Mode. Figure 20 - 18 shows Data Transfers in Repeat Mode.

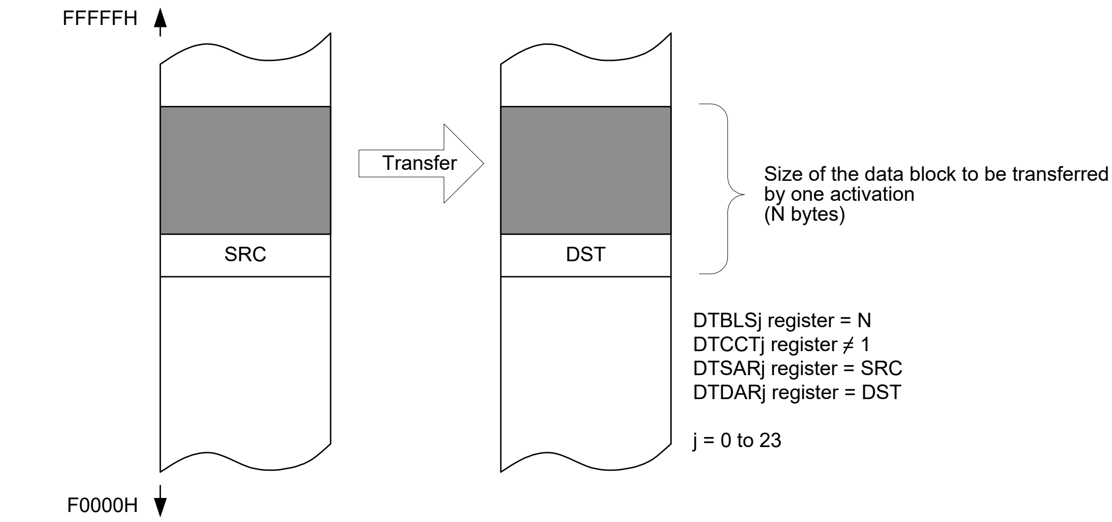
Table 20 - 8 Register Functions in Repeat Mode

Register Name	Symbol	Function
DTC block size register j	DTBLSj	Size of the data block to be transferred by one activation
DTC transfer count register j	DTCCTj	Number of data transfers
DTC transfer count reload register j	DTRLDj	This register value is reloaded to the DTCCT register (the number of transfers is initialized).
DTC source address register j	DTSARj	Data transfer source address
DTC destination address register j	DTDARj	Data transfer destination address

Remark j = 0 to 23

Figure 20 - 18 Data Transfers in Repeat Mode

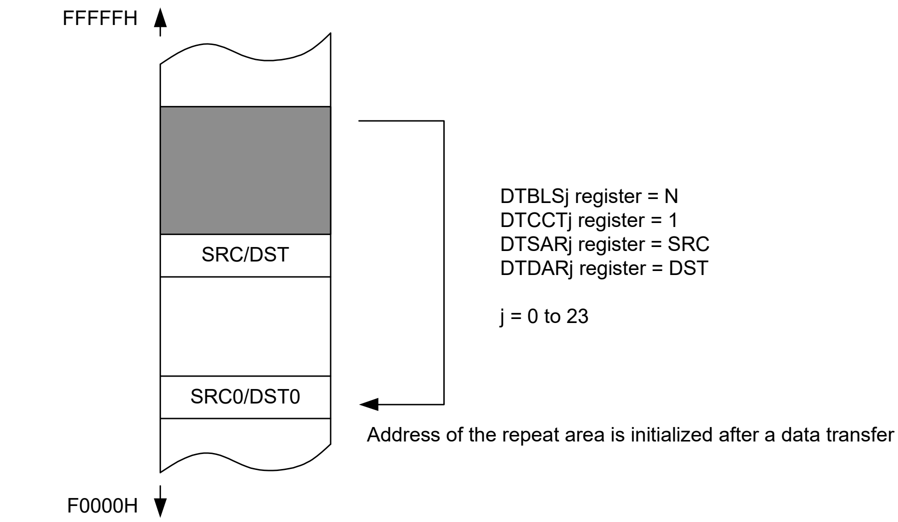
DTCCTj register ≠ 1



DTCCR Register Setting				Source Address Control	Destination Address Control	Source Address after Transfer	Destination Address after Transfer
DAMOD	SAMOD	RPTSEL	MODE				
0	X	1	1	Repeat area	Fixed	SRC + N	DST
1	X	1	1	Repeat area	Incremented	SRC + N	DST + N
X	0	0	1	Fixed	Repeat area	SRC	DST + N
X	1	0	1	Incremented	Repeat area	SRC + N	DST + N

X: 0 or 1

DTCCTj register = 1



DTCCR Register Setting				Source Address Control	Destination Address Control	Source Address after Transfer	Destination Address after Transfer
DAMOD	SAMOD	RPTSEL	MODE				
0	X	1	1	Repeat area	Fixed	SRC0	DST
1	X	1	1	Repeat area	Incremented	SRC0	DST + N
X	0	0	1	Fixed	Repeat area	SRC	DST0
X	1	0	1	Incremented	Repeat area	SRC + N	DST0

SRC0: Initial source address value
DST0: Initial destination address value
X: 0 or 1

Caution 1. When repeat mode is used, the lower 8 bits of the initial value for the repeat area address must be 00H.

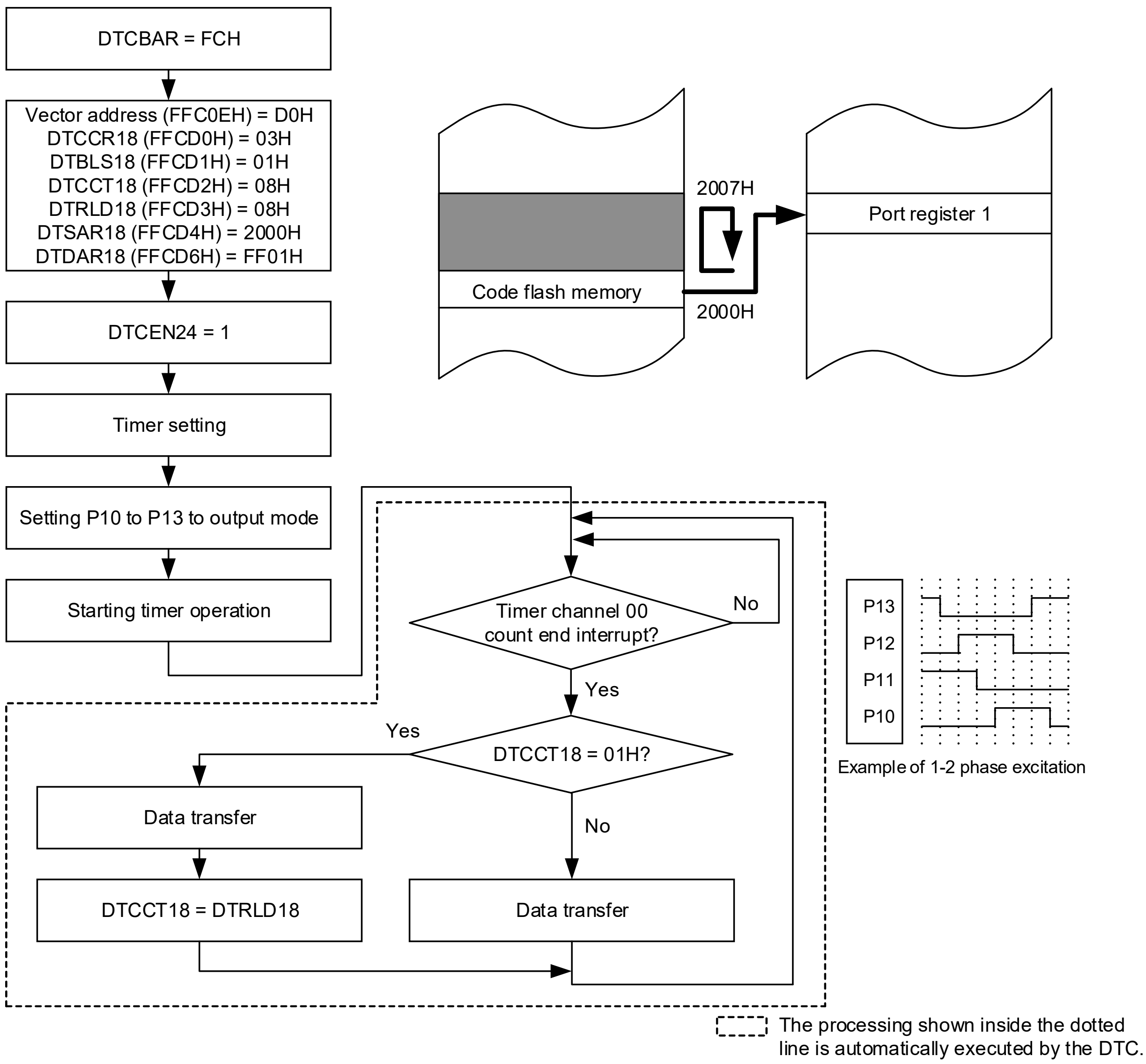
Caution 2. When repeat mode is used, the data size of the repeat area must be set to 255 bytes or less.

(1) Example of using repeat mode - 1: Outputting a stepping motor control pulse using ports

The DTC is activated using the interval timer function of channel 0 of timer array unit 0, and the pattern of the motor control pulse stored in the code flash memory is transferred to the general-purpose port.

- The vector address is FFC0EH and control data is allocated at FFCD0H to FFCD7H
- Transfers 8-byte data of 02000H to 02007H of the code flash memory from the mirror area (F2000H to F2007H) to port register 1 (FFF01H)
- A repeat mode interrupt is disabled

Figure 20 - 19 Example of using repeat mode - 1: Outputting a stepping motor control pulse using ports



To stop the output, stop the timer first and then clear DTCEN11.

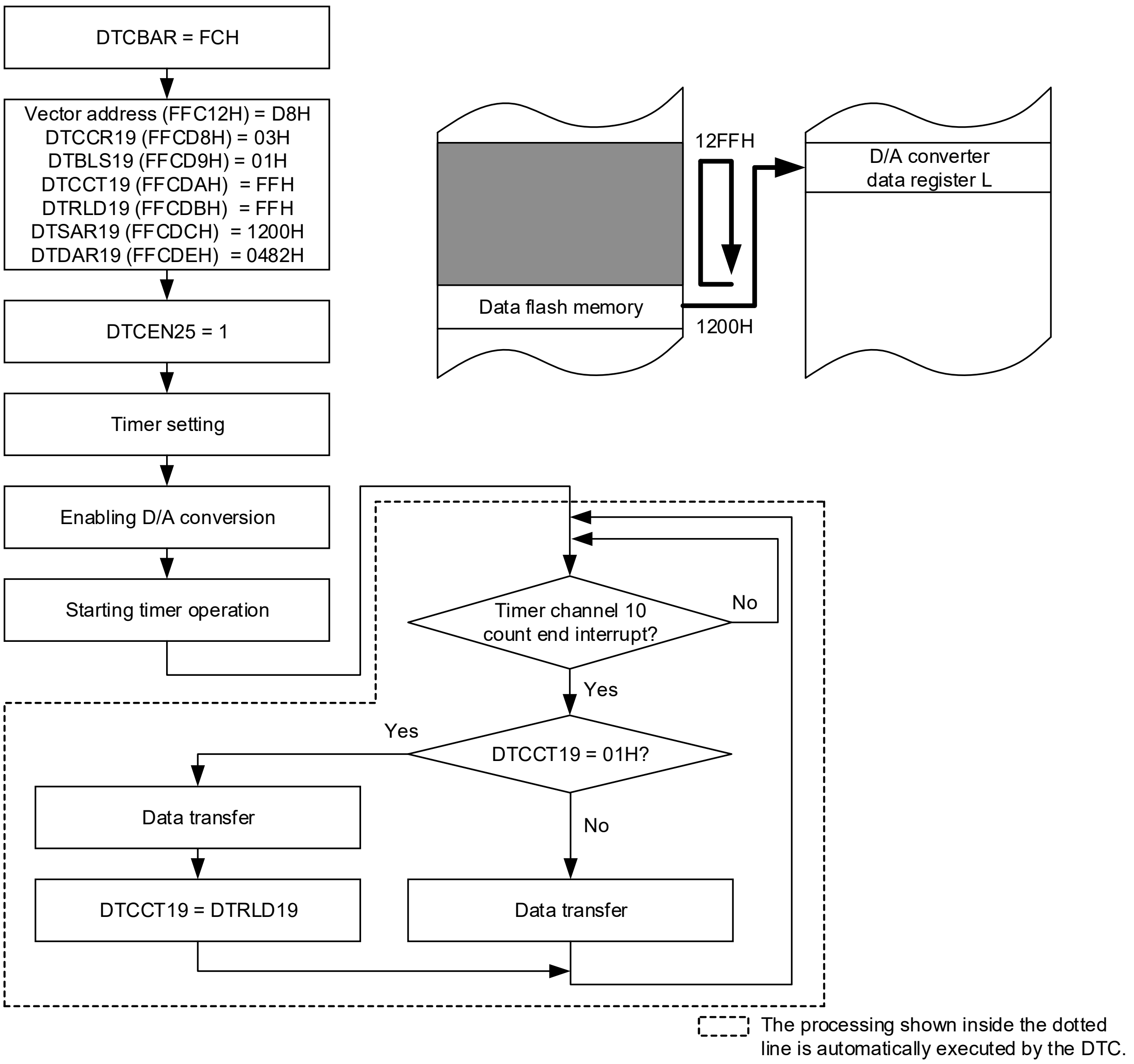
(2) Example of using repeat mode - 2: Outputting a sine wave using the 12-bit D/A converter (8-bit mode)

The DTC is activated using the interval timer function of channel 0 of timer array unit 1, and the table of the sine wave stored in the data flash memory is transferred to 8-bit D/A converter data register L (DACDL) (F0482H).

The timer interval time is set to the D/A output setup time.

- The vector address is FFC12H and control data is allocated at FFCD8H to FFCDFH
- Transfers 255-byte data of F1200H to F12FEH of the data flash memory to D/A converter data register L (DACDL) (F0482H)
- A repeat mode interrupt is disabled

Figure 20 - 20 Example of using repeat mode - 2: Outputting a sine wave using the 12-bit D/A converter (8-bit mode)



To stop the output, stop the timer first and then clear DTCEN25.

20.4.4 Chain transfers

When the CHNE bit in the DTCCRj (j = 0 to 22) register is 1 (chain transfers enabled), multiple data transfers can be continuously performed by one activation source.

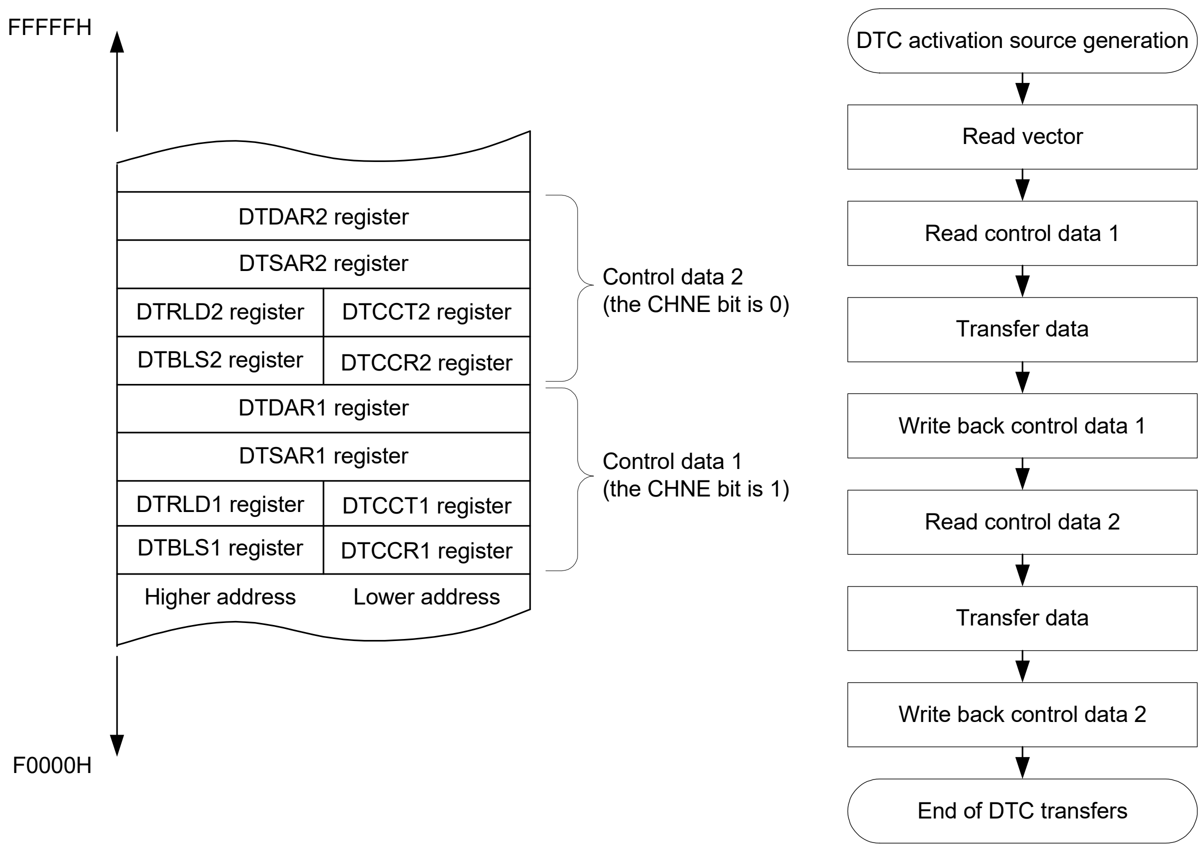
When the DTC is activated, one control data is selected according to the data read from the vector address corresponding to the activation source, and the selected control data is read from the DTC control data area.

When the CHNE bit for the control data is 1 (chain transfers enabled), the next control data immediately following the current control data is read and transferred after the current transfer is completed. This operation is repeated until the data transfer with the control data for which the CHNE bit is 0 (chain transfers disabled) is completed.

When chain transfers are performed using multiple control data, the number of transfers set for the first control data is enabled, and the number of transfers set for the second and subsequent control data to be processed will be invalid.

Figure 20 - 21 shows Data Transfers during Chain Transfers.

Figure 20 - 21 Data Transfers during Chain Transfers



Note 1. Set the CHNE bit in the DTCCR23 register to 0 (chain transfers disabled).

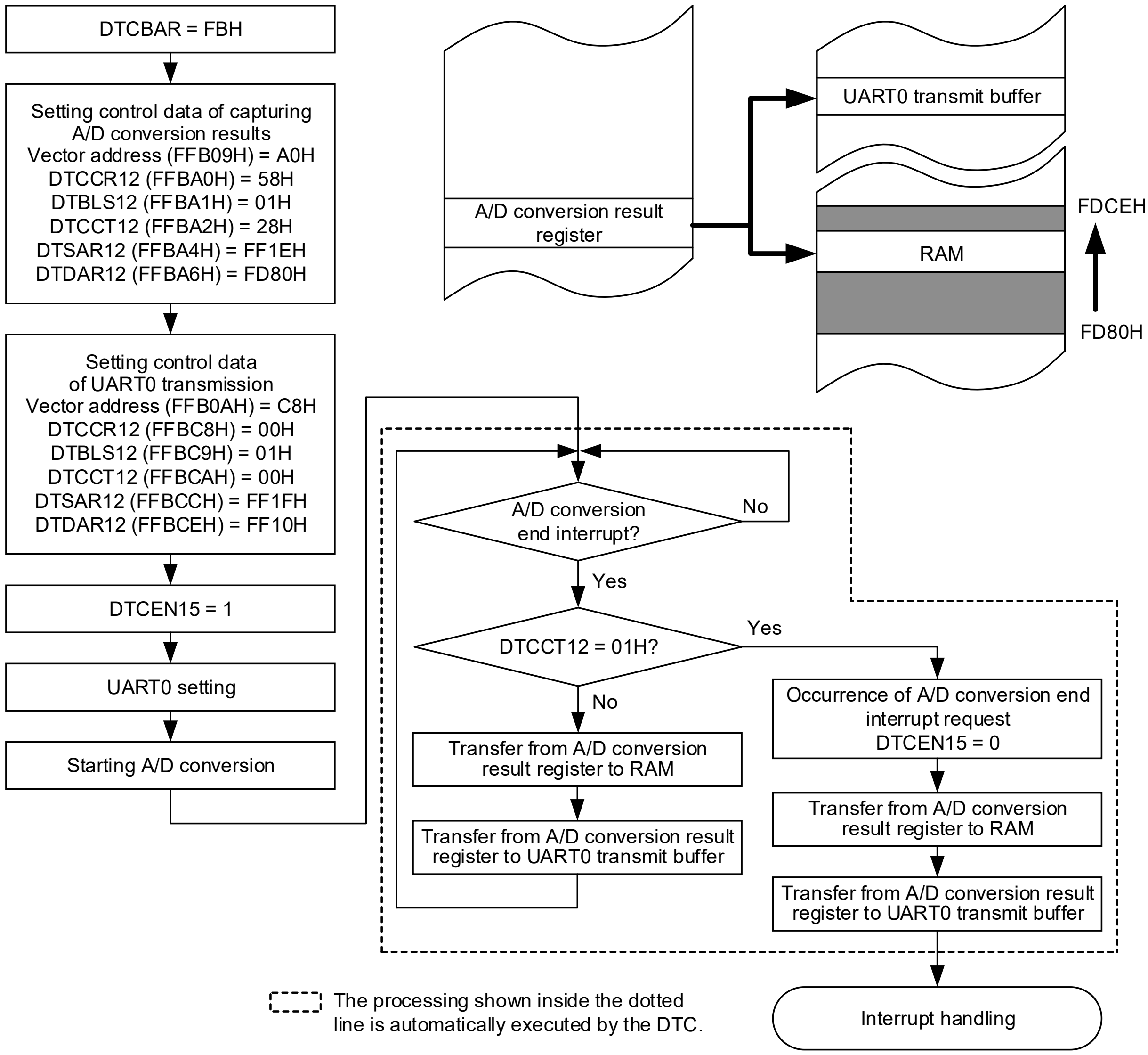
Note 2. During chain transfers, bits DTCENi0 to DTCENi7 (i = 0 to 2) in the DTCENi register are not set to 0 (activation disabled) for the second and subsequent transfers. Also, no interrupt request is generated.

(1) Example of using chain transfers: Consecutively capturing A/D conversion results and UART0 transmission

The DTC is activated by an A/D conversion end interrupt and A/D conversion results are transferred to RAM, and then transmitted using the UART0.

- The vector address is FFB09H
- Control data of capturing A/D conversion results is allocated at FFBA0H to FFBA7H
- Control data of UART0 transmission is allocated at FFBA8H at FFBAFH
- Transfers 2-byte data of the A/D conversion result register (FFF1FH, FFF1EH) to FFD80H to FFDCFH of RAM, and transfers the upper 1 byte (FFF1FH) of the A/D conversion result register to the UART transmit buffer (FFF10H)

Figure 20 - 22 Example of using chain transfers: Consecutively capturing A/D conversion results and UART0 transmission



20.5 Cautions for DTC

20.5.1 Setting DTC control data and vector table

- Do not access the DTC extended special function register (2nd SFR), the DTC control data area, the DTC vector table area, or the general-register (FFEE0H to FFEFFH) space using a DTC transfer.
- Modify the DTC base address register (DTCBAR) while all DTC activation sources are set to activation disabled.
- Do not rewrite the DTC base address register (DTCBAR) twice or more.
- Modify the data of the DTCCRj, DTBLSj, DTCCTj, DTRLDj, DTSARj, or DTDARj register when the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi (i = 0 to 2) register is 0 (activation disabled).
- Modify the start address of the DTC control data area to be set in the vector table when the corresponding bit among bits DTCENi0 to DTCENi7 in the DTCENi (i = 0 to 2) register is 0 (activation disabled).
- Do not allocate RAM addresses which are used as a DTC transfer destination/transfer source to the area FFE20H to FFEDFH when performing self-programming and rewriting the data flash memory.

20.5.2 Allocation of DTC control data area and DTC vector table area

The areas where the DTC control data and vector table can be allocated differ, depending on the usage conditions.

- It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space as the DTC control data area or DTC vector table area.
- Make sure the stack area, the DTC control data area, and the DTC vector table area do not overlap.
- The internal RAM area in FFE20H to FFEDFH cannot be used as the DTC control data area or DTC vector table area when using the self-programming and data flash functions.
- Initialize the DTRLD register to 00H even in normal mode when parity error resets are enabled (RPERDIS = 0) using the RAM parity error detection function.

20.5.3 DTC pending instruction

Even if a DTC transfer request is generated, DTC transfer is held pending immediately after the following instructions. Also, the DTC is not activated between PREFIX instruction code and the instruction immediately after that code.

- Call/return instruction
- Unconditional branch instruction
- Conditional branch instruction
- Read access instruction for code flash memory
- Bit manipulation instructions for IFxx, MKxx, PRxx, and PSW, and an 8-bit manipulation instruction that has the ES register as operand
- Instruction for accessing the data flash memory
- Instruction of Multiply, Divide, Multiply & Accumulate (excluding MULU)

Caution 1. When a DTC transfer request is acknowledged, all interrupt requests are held pending until DTC transfer is completed.

Caution 2. While the DTC is held pending by the DTC pending instruction, all interrupt requests are held pending.

20.5.4 Operation when accessing data flash memory space

When accessing the data flash space after an instruction execution from the start of DTC data transfer, a wait of three clock cycles will be inserted to the next instruction.

Instruction 1
DTC data transfer
Instruction ← The wait of three clock cycles occurs.
MOV A, ! Data Flash space

20.5.5 Number of DTC execution clock cycles

Table 20 - 9 lists the Operations Following DTC Activation and Required Number of Cycles for each operation.

Table 20 - 9 Operations Following DTC Activation and Required Number of Cycles

Vector Read	Control Data		Data Read	Data Write
	Read	Write-back		
1	4	Note 1	Note 2	Note 2

Note 1. For the number of clock cycles required for control data write-back, refer to **Table 20 - 10 Number of Clock Cycles Required for Control Data Write-Back Operation**.

Note 2. For the number of clock cycles required for data read/write, refer to **Table 20 - 11 Number of Clock Cycles Required for One Data Read/Write Operation**.

Table 20 - 10 Number of Clock Cycles Required for Control Data Write-Back Operation

DTCCR Register Setting				Address Setting		Control Register to be Written Back				Number of Clock Cycles
DAMOD	SAMOD	RPTSEL	MODE	Source	Destination	DTCCTj Register	DTRLDj Register	DTSARj Register	DTDARj Register	
0	0	X	0	Fixed	Fixed	Written back	Written back	Not written back	Not written back	1
0	1	X	0	Incremented	Fixed	Written back	Written back	Written back	Not written back	2
1	0	X	0	Fixed	Incremented	Written back	Written back	Not written back	Written back	2
1	1	X	0	Incremented	Incremented	Written back	Written back	Written back	Written back	3
0	X	1	1	Repeat area	Fixed	Written back	Written back	Written back	Not written back	2
1	X	1	1		Incremented	Written back	Written back	Written back	Written back	3
X	0	0	1	Fixed	Repeat area	Written back	Written back	Not written back	Written back	2
X	1	0	1	Incremented		Written back	Written back	Written back	Written back	3

Remark j = 0 to 23; X: 0 or 1

Table 20 - 11 Number of Clock Cycles Required for One Data Read/Write Operation

Operation	RAM	Code Flash Memory	Data Flash Memory	Special function register (SFR)	Extended special function register (2nd SFR)	
					No Wait State	Wait States
Data read	1	2	4	1	1	1 + number of wait states Note
Data write	1	—	—	1	1	1 + number of wait states Note

Note The number of wait states differs depending on the specifications of the register allocated to the extended special function register (2nd SFR) to be accessed.

20.5.6 DTC response time

Table 20 - 12 lists the DTC Response Time. The DTC response time is the time from when the DTC activation source is detected until DTC transfer starts. It does not include the number of DTC execution clocks.

Table 20 - 12 DTC Response Time

	Minimum Time	Maximum Time
Response Time	3 clock cycles	19 clock cycles

Note that the response from the DTC may be further delayed under the following cases. The number of delayed clock cycles differs depending on the conditions.

- When executing an instruction from the internal RAM
  Maximum response time: 20 clock cycles
- When executing a DTC pending instruction (refer to **20.5.3 DTC pending instruction**)
- Maximum response time: Maximum response time for each condition + execution clock cycles for the instruction to be held pending under the condition.
- When accessing the TRJ0 register that a wait occurs
  Maximum response time: Maximum response time for each condition + 1 clock cycle

Remark 1 clock cycle: 1/f_{CLK} (f_{CLK}: CPU/peripheral hardware clock)

20.5.7 DTC activation sources

- After inputting a DTC activation source, do not input the same activation source again until DTC transfer is completed.
- While a DTC activation source is generated, do not manipulate the DTC activation enable bit corresponding to the source.
- If DTC activation sources conflict, their priority levels are determined in order to select the source for activation when the CPU acknowledges the DTC transfer. For details on the priority levels of activation sources, refer to **20.3.3 Vector table**.

20.5.8 Operation in standby mode status

Status	DTC Operation
HALT mode	Operable (Operation is disabled while in the low power consumption RTC mode)
STOP mode	DTC activation sources can be accepted Note 2
SNOOZE mode	Operable Notes 1, 3, 4, 5

Note 1. The SNOOZE mode can only be specified when the high-speed on-chip oscillator clock is selected as f_{CLK}.

Note 2. In the STOP mode, detecting a DTC activation source enables transition to SNOOZE mode and DTC transfer. After completion of transfer, the system returns to the STOP mode. However, since the code flash memory and the data flash memory are stopped during the SNOOZE mode, the flash memory cannot be set as the transfer source.

Note 3. When a transfer end interrupt is set as a DTC activation source from the CSIp SNOOZE mode function, release the SNOOZE mode using the transfer end interrupt to start CPU processing after completion of DTC transfer, or use a chain transfer to set CSIp reception again (writing 1 to the STm0 bit, writing 0 to the SWCm bit, setting of the SSCm register, and writing 1 to the SSm0 bit).

Note 4. When a transfer end interrupt is set as a DTC activation source from the UARTq SNOOZE mode function, release the SNOOZE mode using the transfer end interrupt to start CPU processing after completion of DTC transfer, or use a chain transfer to set UARTq reception again (writing 1 to the STm1 bit, writing 0 to the SWCm bit, setting of the SSCm register, and writing 1 to the SSm1 bit).

Note 5. When an A/D conversion end interrupt is set as a DTC activation source from the A/D converter SNOOZE mode function, release the SNOOZE mode using the A/D conversion end interrupt to start CPU processing after completion of DTC transfer, or use a chain transfer to set the A/D converter SNOOZE mode function again (writing 0 to the AWC bit and then writing 1 to the AWC bit).

Remark p = 00; q = 0; m = 0

CHAPTER 21 EVENT LINK CONTROLLER (ELC)

21.1 Functions of ELC

The event link controller (ELC) mutually connects (links) events output from each peripheral function. By linking events, it becomes possible to coordinate operation between peripheral functions directly without going through the CPU.

The ELC has the following functions.

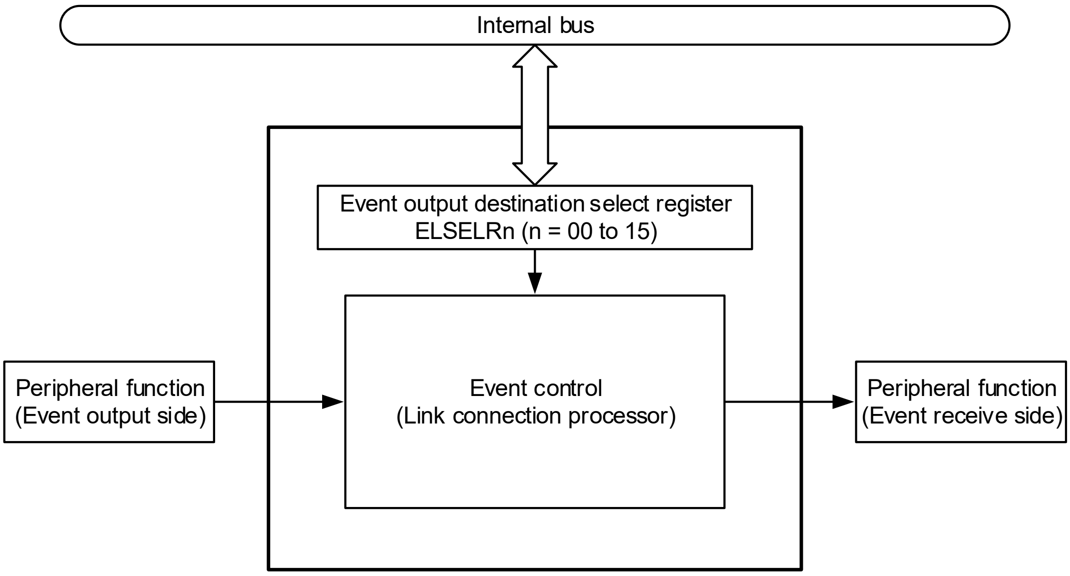
• Capable of directly linking event signals from 16 types of peripheral functions to specified peripheral functions

• Event signals can be used as activation sources for operating any one of seven types of peripheral functions

21.2 Configuration of ELC

Figure 21 - 1 shows the ELC Block Diagram.

Figure 21 - 1 ELC Block Diagram



21.3 Registers Controlling ELC

Table 21 - 1 lists the Registers Controlling ELC.

Table 21 - 1 Registers Controlling ELC

Register name	Symbol
Event output destination select register 00	ELSELR00
Event output destination select register 01	ELSELR01
Event output destination select register 02	ELSELR02
Event output destination select register 03	ELSELR03
Event output destination select register 04	ELSELR04
Event output destination select register 05	ELSELR05
Event output destination select register 06	ELSELR06
Event output destination select register 07	ELSELR07
Event output destination select register 08	ELSELR08
Event output destination select register 09	ELSELR09
Event output destination select register 10	ELSELR10
Event output destination select register 11	ELSELR11
Event output destination select register 12	ELSELR12
Event output destination select register 13	ELSELR13
Event output destination select register 14	ELSELR14
Event output destination select register 15	ELSELR15

21.3.1 Event output destination select register n (ELSELRn) (n = 00 to 15)

An ELSELRn register links each event signal to an operation of an event-receiving peripheral function (link destination peripheral function) after reception.
Do not set multiple event inputs to the same event output destination (event receive side). The operation of the event-receiving peripheral function will become undefined, and event signals may not be received correctly. In addition, do not set the event link generation source and the event link output destination to the same function.
Set an ELSELRn register during a period when no event output peripheral functions are generating event signals.
Table 21 - 2 lists the Correspondence Between ELSELRn (n = 00 to 15) Registers and Peripheral Functions, and Table 21 - 3 lists the Correspondence Between Values Set to ELSELRn (n = 00 to 15) Registers and Operation of Link Destination Peripheral Functions at Reception.

Figure 21 - 2 Format of Event output destination select register n (ELSELRn)

Address: F0300H (ELSELR00) to F030FH (ELSELR15) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ELSELRn	0	0	0	0	0	ELSELn2	ELSELn1	ELSELn0

ELSELn2	ELSELn1	ELSELn0	Event Link Selection
0	0	0	Event link disabled
0	0	1	Select operation of peripheral function 1 to link Note
0	1	0	Select operation of peripheral function 2 to link Note
0	1	1	Select operation of peripheral function 3 to link Note
1	0	0	Select operation of peripheral function 4 to link Note
1	0	1	Select operation of peripheral function 5 to link Note
1	1	0	Select operation of peripheral function 6 to link Note
1	1	1	Select operation of peripheral function 7 to link Note
Other than above			Setting prohibited

Note See **Table 21 - 3 Correspondence Between Values Set to ELSELRn (n = 00 to 15) Registers and Operation of Link Destination Peripheral Functions at Reception**.

Table 21 - 2 Correspondence Between ELSELRn (n = 00 to 15) Registers and Peripheral Functions

Register Name	Event Generator (Output Source of Event Input n)	Event
ELSELR00	External interrupt edge detection 0	INTP0
ELSELR01	External interrupt edge detection 1	INTP1
ELSELR02	External interrupt edge detection 2	INTP2
ELSELR03	External interrupt edge detection 3	INTP3
ELSELR04	External interrupt edge detection 4	INTP4
ELSELR05	External interrupt edge detection 5	INTP5
ELSELR06	RTC fixed-cycle signal/Alarm match detection	INTRTC
ELSELR07	Timer RJ0 underflow	INTTRJ0
ELSELR08	Timer RG input capture A/compare match A	INTTRG
ELSELR09	Timer RG input capture B/compare match B	INTTRG
ELSELR10	TAU channel 00 count end/capture end	INTTM00
ELSELR11	TAU channel 01 count end/capture end	INTTM01
ELSELR12	TAU channel 02 count end/capture end	INTTM02
ELSELR13	TAU channel 03 count end/capture end	INTTM03
ELSELR14	TAU channel 10 count end/capture end	INTTM10
ELSELR15	TAU channel 11 count end/capture end	INTTM11

Table 21 - 3 Correspondence Between Values Set to ELSELRn (n = 00 to 15) Registers and Operation of Link Destination Peripheral Functions at Reception

Bits ELSELn2 to ELSELn0 in ELSELRn Register	Link Destination Number	Link Destination Peripheral Function	Operation When Receiving Event
001B	1	A/D converter	A/D conversion starts
010B	2	Timer input of timer array unit 0 channel 0 Note 1	Delay counter, input pulse interval measurement, external event counter
011B	3	Timer input of timer array unit 0 channel 1 Note 2	Delay counter, input pulse interval measurement, external event counter
100B	4	Timer RJ0	Count source
101B	5	Timer RG	TRGIOB input capture
110B	6	24-bit ΔΣ A/D converter	A/D conversion starts
111B	7	12-bit D/A converter Note 3	D/A-converted output value changes

Note 1. To select the timer input of timer array unit 0 channel 0 as the link destination peripheral function, set the operating clock for channel 0 to fCLK using timer clock select register 0 (TPS0), set the noise filter of the TI00 pin to OFF (TNFEN00 = 0) using noise filter enable register 1 (NFEN1), and then set the timer output used for channel 0 to an event input signal from the ELC using timer input select register 0 (TIS0).

Note 2. To select the timer input of timer array unit 0 channel 1 as the link destination peripheral function, set the operating clock for channel 1 to fCLK using timer clock select register 0 (TPS0), set the noise filter of the TI01 pin to OFF (TNFEN01 = 0) using noise filter enable register 1 (NFEN1), and then set the timer output used for channel 1 to an event input signal from the ELC using timer input select register 0 (TIS0).

Note 3. When entering the STOP mode while hardware trigger mode for D/A conversion is enabled, disable linking of ELC events before entering STOP mode.

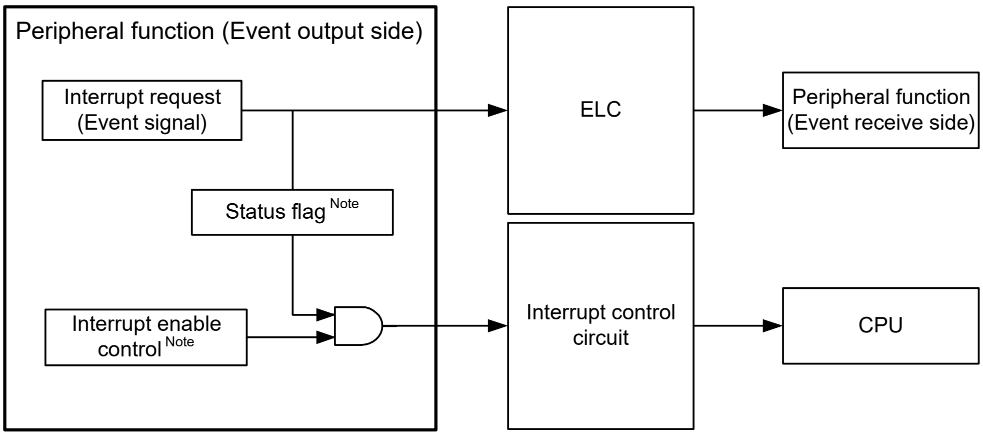
21.4 ELC Operation

The path for using an event signal generated by a peripheral function as an interrupt request to the interrupt control circuit is independent from the path for using it as an ELC event. Therefore, each event signal can be used as an event signal for operation of an event-receiving peripheral function, regardless of interrupt control.

Figure 21 - 3 shows the Relationship Between Interrupt Handling and ELC. The figure show an example of an interrupt request status flag and a peripheral function possessing the enable bits that control enabling/disabling of such interrupts.

A peripheral function which receives an event from the ELC will perform the operation corresponding to the event-receiving peripheral function after reception of an event (See **Table 21 - 3 Correspondence Between Values Set to ELSELRn (n = 00 to 15) Registers and Operation of Link Destination Peripheral Functions at Reception**).

Figure 21 - 3 Relationship Between Interrupt Handling and ELC



Note Not available depending on the peripheral function.

Table 21 - 4 lists the Response of Peripheral Functions That Receive Events.

Table 21 - 4 Response of Peripheral Functions That Receive Events

Event Receiver No.	Event Link Destination Function	Operation after Event Reception	Response
1	A/D converter	A/D conversion	An event from the ELC is directly used as a hardware trigger of A/D conversion.
2	Timer array unit 0 Timer input of channel 0	Delay counter Input pulse width measurement External event counter	The edge is detected 3 or 4 cycles of f_{CLK} after an ELC event is generated.
3	Timer array unit 0 Timer input of channel 1	Delay counter Input pulse width measurement External event counter	The edge is detected 3 or 4 cycles of f_{CLK} after an ELC event is generated.
4	Timer RJ	Count source	An event from the ELC is directly used as the count source of timer RJ.
5	Timer RG	TRGIOB input capture	A count start trigger is generated 2 or 3 cycles of f_{CLK} after an ELC event is generated.
6	24-bit ΔΣ A/D converter	A/D conversion	An ELC event is used as a hardware trigger for A/D conversion 2 or 3 cycles of f_{DSADCK} after the ELC event has been generated.
7	12-bit D/A converter	D/A-converted output value changes	D/A conversion starts 2 or 3 cycles of f_{CLK} after an ELC event is generated.

CHAPTER 22 INTERRUPT FUNCTIONS

The interrupt function switches the program execution to other processing. When the branch processing is finished, the program returns to the interrupted processing.
The number of interrupt sources differs, depending on the product.

		32-pin	36-pin
Maskable interrupts	External	7	8
	Internal	21	

22.1 Interrupt Function Types

The following two types of interrupt functions are used.

(1) Maskable interrupts
These interrupts undergo mask control. Maskable interrupts can be divided into four priority groups by setting the priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H).
Multiple interrupt servicing can be applied to low-priority interrupts when high-priority interrupts are generated. If two or more interrupt requests, each having the same priority, are simultaneously generated, then they are processed according to the default priority of vectored interrupt servicing. For the default priority, see **Table 22 - 1**, **Table 22 - 2**, and **Figure 22 - 1**.
A standby release signal is generated and STOP, HALT, and SNOOZE modes are released.
External interrupt requests and internal interrupt requests are provided as maskable interrupts.

(2) Software interrupt
This is a vectored interrupt generated by executing the BRK instruction. It is acknowledged even when interrupts are disabled. The software interrupt does not undergo interrupt priority control.

22.2 Interrupt Sources and Configuration

Interrupt sources include maskable interrupts and software interrupts. In addition, they also have up to seven reset sources (see **Table 22 - 1**, **Table 22 - 2**, and **Figure 22 - 1**). The vector codes that store the program start address when branching due to the generation of a reset or various interrupt requests are two bytes each, so interrupts jump to a 64 K address of 00000H to 0FFFFH.

Table 22 - 1 Interrupt Source List (1/2)

Interrupt Type	Default Priority [Note 1]	Interrupt Source: Name	Interrupt Source: Trigger	Internal/External	Vector Table Address	Basic Configuration Type [Note 2]	36-pin	32-pin
Maskable	0	INTWDTI	Watchdog timer interval [Note 3] (75% of overflow time + 1/2 f_{IL})	Internal	0004H	(A)	√	√
	1	INTLVI	Voltage detection [Note 4]		0006H		√	√
	2	INTP0	Pin input edge detection	External	0008H	(B)	√	√
	3	INTP1			000AH		√	√
	4	INTP2			000CH		√	√
	5	INTP3			000EH		√	√
	6	INTP4			0010H		√	√
	7	INTP5			0012H		√	√
	8	INTST0/ INTCSI00/ INTIIC00	UART0 transmission transfer end or buffer empty interrupt/CSI00 transfer end or buffer empty interrupt/IIC00 transfer end	Internal	001EH	(A)	√	√
	9	INTSR0/ INTCSI01/ INTIIC01	UART0 reception transfer end/CSI01 transfer end or buffer empty interrupt/IIC01 transfer end		0020H		√	√
	10	INTSRE0	UART0 reception communication error occurrence		0022H		√	√
		INTTM01H	End of timer channel 01 count or capture (at higher 8-bit timer operation)				√	√
	11	INTST1	UART1 transmission transfer end or buffer empty interrupt		0024H		√	√
	12	INTSR1	UART1 reception transfer end		0026H		√	√
	13	INTSRE1	UART1 reception communication error occurrence		0028H		√	√
		INTTM03H	End of timer channel 03 count or capture (at higher 8-bit timer operation)				√	√
	14	INTTM00	End of timer channel 00 count or capture		002CH		√	√
	15	INTTM01	End of timer channel 01 count or capture		002EH		√	√
	16	INTTM02	End of timer channel 02 count or capture		0030H		√	√
	17	INTTM03	End of timer channel 03 count or capture		0032H		√	√
	18	INTAD	End of 10-bit A/D conversion		0034H		√	√
	19	INTRTC	Fixed-cycle signal of real-time clock/alarm match detection		0036H		√	√
	20	INTIT	Interval signal detection		0038H		√	√
	21	INTTRJ0	Timer RJ interrupt		0040H		√	√

Note 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 29 indicates the lowest priority.

Note 2. Basic configuration types (A) to (C) correspond to (A) to (C) in Figures 22 - 1.

Note 3. When bit 7 (WDTINT) of the option byte (000C0H) is set to 1.

Note 4. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is cleared to 0.

Table 22 - 2 Interrupt Source List (2/2)

Interrupt Type	Default Priority Note 1	Interrupt Source: Name	Interrupt Source: Trigger	Internal/External	Vector Table Address	Basic Configuration Type Note 2	36-pin	32-pin
Maskable	22	INTTM10	End of timer channel 10 count or capture	Internal	0042H	(A)	√	√
	23	INTTM11	End of timer channel 11 count or capture		0044H		√	√
	24	INTP6	Pin input edge detection	External	004AH	(B)	√	√
	25	INTP7			004CH		√	√
	26	INTDSAD	End of 24-bit ΔΣ A/D conversion	Internal	004EH	(A)	√	√
	27	INTDSADS	End of 24-bit ΔΣ A/D scan		0050H		√	√
	28	INTTRG	Timer RG input capture, compare match, overflow, underflow interrupt		005AH		√	√
	29	INTFL	Reserved Note 3		0062H		√	√
Software	—	BRK	Execution of BRK instruction	—	007EH	(C)	√	√
Reset	—	RESET	$\overline{\text{RESET}}$ pin input	—	0000H	—	√	√
		POR	Power-on-reset				√	√
		LVD	Voltage detection Note 4				√	√
		WDT	Overflow of watchdog timer				√	√
		TRAP	Execution of illegal instruction Note 5				√	√
		IAW	Illegal-memory access				√	√
		RPE	RAM parity error				√	√

Note 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 29 indicates the lowest priority.

Note 2. Basic configuration types (A) to (C) correspond to (A) to (C) in Figures 22 - 1.

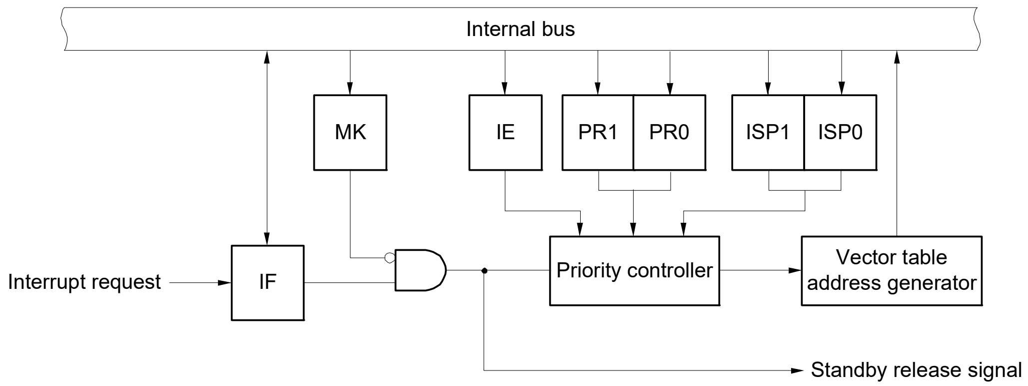
Note 3. Be used at the flash self-programming library or the data flash library.

Note 4. When bit 7 (LVIMD) of the voltage detection level register (LVIS) is set to 1.

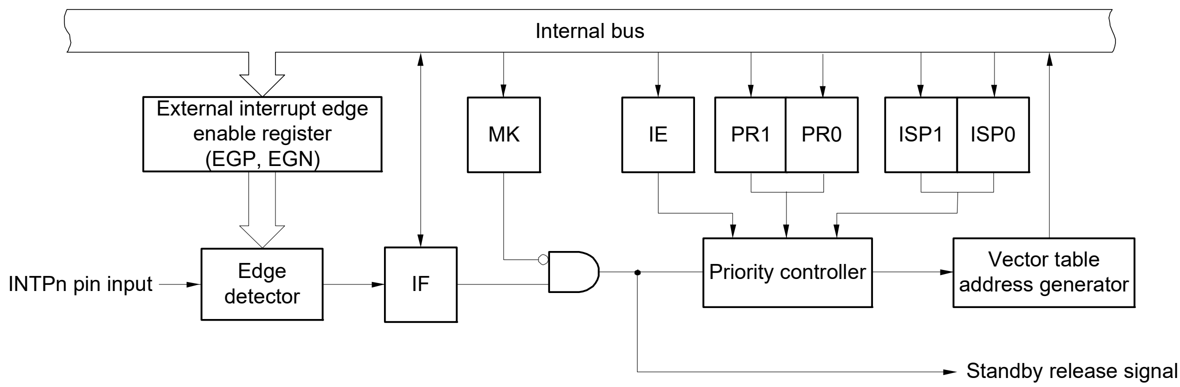
Note 5. This occurs when the instruction code FFH is executed. Reset caused by an illegal instruction execution does not occur during emulation using the in-circuit emulator or on-chip debug emulator.

Figure 22 - 1 Basic Configuration of Interrupt Function

(A) Internal maskable interrupt



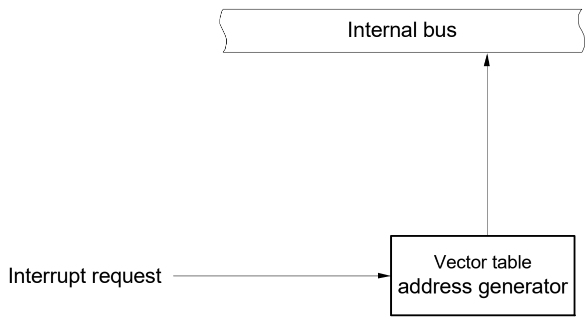
(B) External maskable interrupt (INTPn)



IF:	Interrupt request flag
IE:	Interrupt enable flag
ISP0:	In-service priority flag 0
ISP1:	In-service priority flag 1
MK:	Interrupt mask flag
PR0:	Priority specification flag 0
PR1:	Priority specification flag 1

Remark 32-pin: n = 0 to 6
36-pin: n = 0 to 7

(C) Software interrupt



22.3 Registers Controlling Interrupt Functions

The following six types of registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)
- Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)
- Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H)
- External interrupt rising edge enable register (EGP0)
- External interrupt falling edge enable register (EGN0)
- Program status word (PSW)

Tables 22 - 3 to 22 - 5 show a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 22 - 3 Flags Corresponding to Interrupt Request Sources (1/2)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		36-pin	32-pin
		Register		Register		Register		
INTWDTI	WDTIIF	IF0L	WDTIMK	MK0L	WDTIPR0, WDTIPR1	PR00L, PR10L	√	√
INTLVI	LVIIF		LVIMK		LVIPR0, LVIPR1		√	√
INTP0	PIF0		PMK0		PPR00, PPR10		√	√
INTP1	PIF1		PMK1		PPR01, PPR11		√	√
INTP2	PIF2		PMK2		PPR02, PPR12		√	√
INTP3	PIF3		PMK3		PPR03, PPR13		√	√
INTP4	PIF4		PMK4		PPR04, PPR14		√	√
INTP5	PIF5		PMK5		PPR05, PPR15		√	√

Table 22 - 4 Flags Corresponding to Interrupt Request Sources (2/2)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag		36-pin	32-pin
		Register		Register		Register		
INTST0 Note 1	STIF0 Note 1	IF0H	STMK0 Note 1	MK0H	STPR00, STPR10 Note 1	PR00H, PR10H	√	√
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 Note 2	SRIF0 Note 2		SRMK0 Note 2		SRPR00, SRPR10 Note 2		√	√
INTCSI01 Note 2	CSIIF01 Note 2		CSIMK01 Note 2		CSIPR001, CSIPR101 Note 2		√	√
INTIIC01 Note 2	IICIF01 Note 2		IICMK01 Note 2		IICPR001, IICPR101 Note 2		√	√
INTSRE0 Note 3	SREIF0 Note 3		SREMK0 Note 3		SREPR00, SREPR10 Note 3		√	√
INTTM01H Note 3	TMIF01H Note 3		TMMK01H Note 3		TMPR001H, TMPR101H Note 3		√	√
INTST1	STIF1	IF1L	STMK1	MK1L	STPR01, STPR11	PR01L, PR11L	√	√
INTSR1	SRIF1		SRMK1		SRPR01, SRPR11		√	√
INTSRE1 Note 4	SREIF1 Note 4		SREMK1 Note 4		SREPR01, SREPR11 Note 4		√	√
INTTM03H Note 4	TMIF03H Note 4		TMMK03H Note 4		TMPR003H, TMPR103H Note 4		√	√
INTTM00	TMIF00		TMMK00		TMPR000, TMPR100		√	√
INTTM01	TMIF01		TMMK01		TMPR001, TMPR101		√	√
INTTM02	TMIF02		TMMK02		TMPR002, TMPR102		√	√
INTTM03	TMIF03		TMMK03		TMPR003, TMPR103		√	√
INTAD	ADIF	IF1H	ADMK	MK1H	ADPR0, ADPR1	PR01H, PR11H	√	√
INTRTC	RTCIF		RTCMK		RTCPR0, RTCPR1		√	√
INTIT	ITIF		ITMK		ITPR0, ITPR1		√	√
INTTRJ0	TRJIF0		TRJMK0		TRJPR00, TRJPR10		√	√
INTTM10	TMIF10		TMMK10		TMPR010, TMPR110		√	√
INTTM11	TMIF11	IF2L	TMMK11	MK2L	TMPR011, TMPR111	PR02L, PR12L	√	√
INTP6	PIF6		PMK6		PPR06, PPR16		√	√
INTP7	PIF7		PMK7		PPR07, PPR17		—	√
INTDSAD	DSADIF		DSADMK		DSADPR0, DSADPR1		√	√
INTDSADS	DSADSIF		DSADSMK		DSADSPR0, DSADSPR1		√	√
INTTRG	TRGIF	IF2H	TRGMK	MK2H	TRGPR0, TRGPR1	PR02H, PR12H	√	√
INTFL	FLIF		FLMK		FLPR0, FLPR1		√	√

Note 1. If one of the interrupt sources INTST0, INTCSI00, and INTIIC00 is generated, bit 5 of the IF0H register is set to 1. Bit 5 of the MK0H, PR00H, and PR10H registers supports these three interrupt sources.

Note 2. If one of the interrupt sources INTSR0, INTCSI01, and INTIIC01 is generated, bit 6 of the IF0H register is set to 1. Bit 6 of the MK0H, PR00H, and PR10H registers supports these three interrupt sources.

Note 3. Do not use a UART0 reception error interrupt and an interrupt of channel 1 of TAU0 (at higher 8-bit timer operation) at the same time because they share flags for the interrupt request sources. When the UART0 reception error interrupt is not used (EOC01 = 0), UART0 and channel 1 of TAU0 (at higher 8-bit timer operation) can be used at the same time. If either of the interrupt sources INTSRE0 or INTTM01H is generated, bit 7 of the IF0H register is set to 1. Bit 7 of the MK0H, PR00H, and PR10H registers support these two interrupt sources.

Note 4. Do not use a UART1 reception error interrupt and an interrupt of channel 3 of TAU0 (at higher 8-bit timer operation) at the same time because they share flags for the interrupt request sources. When the UART1 reception error interrupt is not used (EOC03 = 0), UART1 and channel 3 of TAU0 (at higher 8-bit timer operation) can be used at the same time. If either of the interrupt sources INTSRE1 or INTTM03H is generated, bit 2 of the IF1H register is set to 1. Bit 2 of the MK1L, PR01L, and PR11L registers support these two interrupt sources.

22.3.1 Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)

The interrupt request flags are set to 1 when the corresponding interrupt request is generated or an instruction is executed. They are cleared to 0 when an instruction is executed upon acknowledgment of an interrupt request or upon reset signal generation.

When an interrupt is acknowledged, the interrupt request flag is automatically cleared and then the interrupt routine is entered.

The IF0L, IF0H, IF1L, IF1H, IF2L, and IF2H registers can be set by a 1-bit or 8-bit memory manipulation instruction. When the IF0L and IF0H registers, the IF1L and IF1H registers, and the IF2L and IF2H registers are combined to form 16-bit registers IF0, IF1, and IF2, 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 clock cycles.

Figure 22 - 2 Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) (1/2)

Address: FFFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIF	WDTIIF

Address: FFFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
IF0H	SREIF0 TMIF01H	SRIF0 CSIIF01 IICIF01	STIF0 CSIIF00 IICIF00	0	0	0	0	0

Address: FFFE2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
IF1L	TMIF03	TMIF02	TMIF01	TMIF00	0	SREIF1 TMIF03H	SRIF1	STIF1

Address: FFFE3H After reset: 00H R/W

Symbol	<7>	<6>	5	4	3	<2>	<1>	<0>
IF1H	TMIF10	TRJIF0	0	0	0	ITIF	RTCIF	ADIF

Address: FFFD0H After reset: 00H R/W

Symbol	7	<6>	<5>	<4>	<3>	2	1	<0>
IF2L	0	DSADSIF	DSADIF	PIF7	PIF6	0	0	TMIF11

Figure 22 - 3 Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H) (2/2)

Address: FFFD1H After reset: 00H R/W

Symbol	<7>	6	5	4	<3>	2	1	0
IF2H	FLIF	0	0	0	TRGIF	0	0	0

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Caution 1. The available registers and bits differ depending on the product. For details about the registers and bits available for each product, see Tables 22 - 3 to 22 - 5. Be sure to set bits that are not available to the initial value.

Caution 2. When manipulating a flag of the interrupt request flag register, use a 1-bit memory manipulation instruction (CLR1). When describing in C language, use a bit manipulation instruction such as "IF0L.0 = 0;" or "_asm ("clr1 IF0L, 0");" because the compiled assembler must be a 1-bit memory manipulation instruction (CLR1).

If a program is described in C language using an 8-bit memory manipulation instruction such as "IF0L &= 0xfe;" and compiled, it becomes the assembler of three instructions.

```
mov a, IF0L
and a, #0FEH
mov IF0L, a
```

In this case, even if the request flag of the another bit of the same interrupt request flag register (IF0L) is set to 1 at the timing between "mov a, IF0L" and "mov IF0L, a", the flag is cleared to 0 at "mov IF0L, a". Therefore, care must be exercised when using an 8-bit memory manipulation instruction in C language.

22.3.2 Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt.

The MK0L, MK0H, MK1L, MK1H, MK2L, and MK2H registers can be set by a 1-bit or 8-bit memory manipulation instruction. When the MK0L and MK0H registers, the MK1L and MK1H registers, and the MK2L and MK2H registers are combined to form 16-bit registers MK0, MK1, and MK2, 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 clock cycles.

Figure 22 - 4 Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H) (1/2)

Address: FFFE4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK	WDTIMK

Address: FFFE5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
MK0H	SREMK0 TMMK01H	SRMK0 CSIMK01 IICMK01	STMK0 CSIMK00 IICMK00	1	1	1	1	1

Address: FFFE6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
MK1L	TMMK03	TMMK02	TMMK01	TMMK00	1	SREMK1 TMMK03H	SRMK1	STMK1

Address: FFFE7H After reset: FFH R/W

Symbol	<7>	<6>	5	4	3	<2>	<1>	<0>
MK1H	TMMK10	TRJMK0	1	1	1	ITMK	RTCMK	ADMK

Address: FFFD4H After reset: FFH R/W

Symbol	7	<6>	<5>	<4>	<3>	2	1	<0>
MK2L	1	DSADSMK	DSADMK	PMK7	PMK6	1	1	TMMK11

Figure 22 - 5 Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H) (2/2)

Address: FFFD5H After reset: FFH R/W

Symbol	<7>	6	5	4	<3>	2	1	0
MK2H	FLMK	1	1	1	TRGMK	1	1	1

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 Tables 22 - 3 to 22 - 5. Be sure to set bits that are not available to the initial value.**

22.3.3 Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H)

The priority specification flag registers are used to set the corresponding maskable interrupt priority level.
A priority level is set by using the PR0xy and PR1xy registers in combination (xy = 0L, 0H, 1L, 1H, 2L, or 2H).
The PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and the PR12H registers can be set by a 1-bit or 8-bit memory manipulation instruction. If the PR00L and PR00H registers, the PR01L and PR01H registers, the PR02L and PR02H registers, the PR10L and PR10H registers, the PR11L and PR11H registers, and the PR12L and PR12H registers are combined to form 16-bit registers PR00, PR01, PR02, PR10, PR11, and PR12, 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 clock cycles.

Figure 22 - 6 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (1/2)

Address: FFFE8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00L	PPR05	PPR04	PPR03	PPR02	PPR01	PPR00	LVIPR0	WDTIPR0

Address: FFFECH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10L	PPR15	PPR14	PPR13	PPR12	PPR11	PPR10	LVIPR1	WDTIPR1

Address: FFFE9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
PR00H	SREPR00 TMPR001H	SRPR00 CSIPR001 IICPR001	STPR00 CSIPR000 IICPR000	1	1	1	1	1

Address: FFFEDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
PR10H	SREPR10 TMPR101H	SRPR10 CSIPR101 IICPR101	STPR10 CSIPR100 IICPR100	1	1	1	1	1

Address: FFFEAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
PR01L	TMPR003	TMPR002	TMPR001	TMPR000	1	SREPR01 TMPR003H	SRPR01	STPR01

Figure 22 - 7 Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (2/2)

Address: FFFEEH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
PR11L	TMPR103	TMPR102	TMPR101	TMPR100	1	SREPR11 TMPR103H	SRPR11	STPR11

Address: FFFEBH After reset: FFH R/W

Symbol	<7>	<6>	5	4	3	<2>	<1>	<0>
PR01H	TMPR010	TRJPR00	1	1	1	ITPR0	RTCPR0	ADPR0

Address: FFFEFH After reset: FFH R/W

Symbol	<7>	<6>	5	4	3	<2>	<1>	<0>
PR11H	TMPR110	TRJPR10	1	1	1	ITPR1	RTCPR1	ADPR1

Address: FFFD8H After reset: FFH R/W

Symbol	7	<6>	<5>	<4>	<3>	2	1	<0>
PR02L	1	DSADSPR0	DSADPR0	PPR07	PPR06	1	1	TMPR011

Address: FFFDCH After reset: FFH R/W

Symbol	7	<6>	<5>	<4>	<3>	2	1	<0>
PR12L	1	DSADSPR1	DSADPR1	PPR17	PPR16	1	1	TMPR111

Address: FFFD9H After reset: FFH R/W

Symbol	<7>	6	5	4	<3>	2	1	0
PR02H	FLPR0	1	1	1	TRGPR0	1	1	1

Address: FFFDDH After reset: FFH R/W

Symbol	<7>	6	5	4	<3>	2	1	0
PR12H	FLPR1	1	1	1	TRGPR1	1	1	1

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 Tables 22 - 3 to 22 - 5. Be sure to set bits that are not available to the initial value.**

22.3.4 External interrupt rising edge enable register (EGP0), external interrupt falling edge enable register (EGN0)

These registers specify the valid edge for INTP0 to INTP7.
The EGP0 and EGN0 registers can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears these registers to 00H.

Figure 22 - 8 Format of External Interrupt Rising Edge Enable Register (EGP0) and External Interrupt Falling Edge Enable Register (EGN0)

Address: FFF38H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	EGP7	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FFF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 7)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 22 - 5 shows the Ports Corresponding to EGPn and EGNn Bits.

Table 22 - 5 Ports Corresponding to EGPn and EGNn Bits

Detection Enable Bit		Interrupt Request Signal	32-pin	36-pin
EGP0	EGN0	INTP0	√	√
EGP1	EGN1	INTP1	√	√
EGP2	EGN2	INTP2	√	√
EGP3	EGN3	INTP3	√	√
EGP4	EGN4	INTP4	√	√
EGP5	EGN5	INTP5	√	√
EGP6	EGN6	INTP6	√	√
EGP7	EGN7	INTP7	—	√

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 details about the edge detection ports, see **2.1 Port Functions**.
Remark 2. n = 0 to 7

22.3.5 Program status word (PSW)

The program status word is a register used to hold the instruction execution result and the current status for an interrupt request. The IE flag that sets maskable interrupt enable/disable and the ISP0 and ISP1 flags that controls multiple interrupt servicing are mapped to the PSW.

Besides 8-bit read/write, this register can carry out operations using bit manipulation instructions and dedicated instructions (EI and DI). When a vectored interrupt request is acknowledged, if the BRK instruction is executed, the contents of the PSW are automatically saved into a stack and the IE flag is reset to 0. Upon acknowledgment of a maskable interrupt request, if the value of the priority specification flag register of the acknowledged interrupt is not 00, its value minus 1 is transferred to the ISP0 and ISP1 flags. The PSW contents are also saved into the stack with the PUSH PSW instruction. They are restored from the stack with the RETI, RETB, and POP PSW instructions.

Reset signal generation sets PSW to 06H.

Figure 22 - 9 Configuration of Program Status Word

	7	6	5	4	3	2	1	0	After reset
PSW	IE	Z	RBS1	AC	RBS0	ISP1	ISP0	CY	06H

Z, RBS1, AC, RBS0, ISP1, ISP0, CY → Used when normal instruction is executed

ISP1	ISP0	Priority of interrupt currently being serviced
0	0	Enables interrupt of level 0 (while interrupt of level 1 or 0 is being serviced).
0	1	Enables interrupt of level 0 and 1 (while interrupt of level 2 is being serviced).
1	0	Enables interrupt of level 0 to 2 (while interrupt of level 3 is being serviced).
1	1	Enables all interrupts (waits for acknowledgment of an interrupt).

IE	Interrupt request acknowledgment enable/disable
0	Disabled
1	Enabled

22.4 Interrupt Servicing Operations

22.4.1 Maskable interrupt request acknowledgment

A maskable interrupt request becomes acknowledgeable when the interrupt request flag is set to 1 and the mask (MK) flag corresponding to that interrupt request is cleared to 0. A vectored interrupt request is acknowledged if interrupts are in the interrupt enabled state (when the IE flag is set to 1). However, a low-priority interrupt request is not acknowledged during servicing of a higher priority interrupt request.

The times from generation of a maskable interrupt request until vectored interrupt servicing is performed are listed in Table 22 - 6 below.

For the interrupt request acknowledgment timing, see **Figures 22 - 11** and **22 - 12**.

Table 22 - 6 Time from Generation of Maskable Interrupt Until Servicing

	Minimum Time	Maximum Time Note
Servicing time	9 clock cycles	16 clock cycles

Note Maximum time does not apply when an instruction from the internal RAM area is executed.

Remark 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

If two or more maskable interrupt requests are generated simultaneously, the request with a higher priority level specified in the priority specification flag is acknowledged first. If two or more interrupts requests have the same priority level, the request with the highest default priority is acknowledged first.

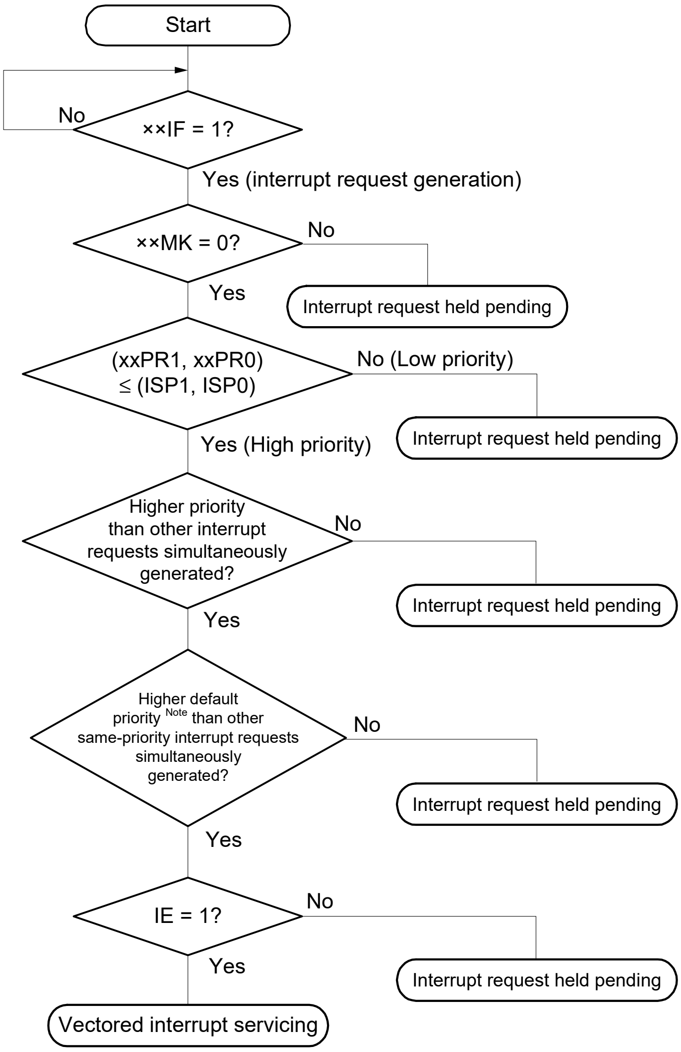
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 22 - 10 shows the Interrupt Request Acknowledgment Processing Algorithm.

If a maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag is reset (0), and the contents of the priority specification flag corresponding to the acknowledged interrupt are transferred to the ISP1 and ISP0 flags. The vector table data determined for each interrupt request is the loaded into the PC and branched.

Restoring from an interrupt is possible by using the RETI instruction.

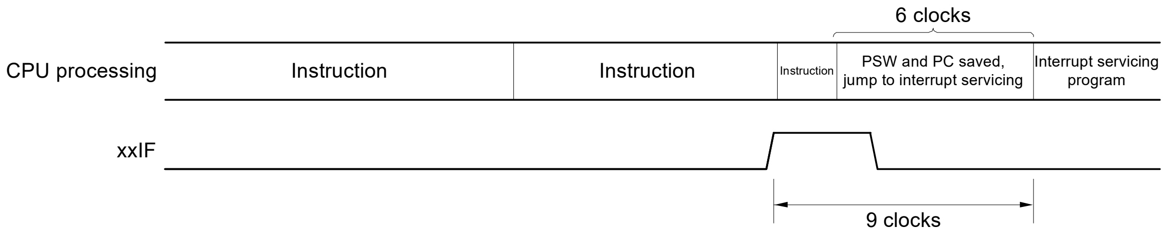
Figure 22 - 10 Interrupt Request Acknowledgment Processing Algorithm



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 22 - 9**)

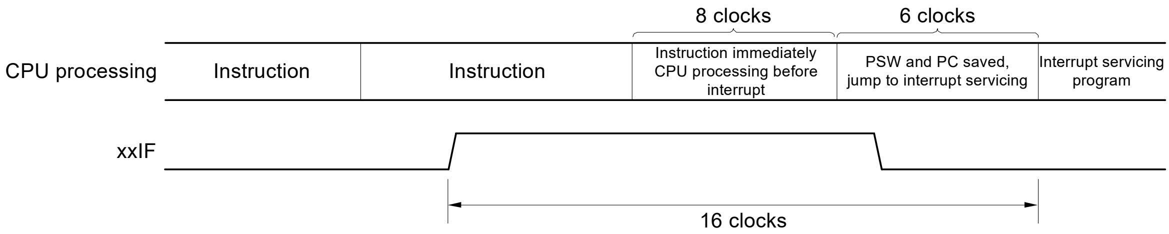
Note For the default priority, refer to **Table 22 - 1**, **Table 22 - 2**, and **Figure 22 - 1**.

Figure 22 - 11 Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

Figure 22 - 12 Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

22.4.2 Software interrupt request acknowledgment

A software interrupt request is acknowledged by BRK instruction execution. Software interrupts cannot be disabled.

If a software interrupt request is acknowledged, the contents are saved into the stacks in the order of the program status word (PSW), then program counter (PC), the IE flag is reset (0), and the contents of the vector table (0007EH, 0007FH) are loaded into the PC and branched.

Restoring from a software interrupt is possible by using the RETB instruction.

Caution The RETI instruction cannot be used for restoring from the software interrupt.

22.4.3 Multiple interrupt servicing

Multiple interrupt servicing occurs when another interrupt request is acknowledged during execution of an interrupt.

Multiple interrupt servicing does not occur unless the interrupt request acknowledgment enabled state is selected (IE = 1). When an interrupt request is acknowledged, interrupt request acknowledgment becomes disabled (IE = 0). Therefore, to enable multiple interrupt servicing, it is necessary to set (1) the IE flag with the EI instruction during interrupt servicing to enable interrupt acknowledgment.

Moreover, even if interrupts are enabled, multiple interrupt servicing may not be enabled, this being subject to interrupt priority control. Two types of priority control are available: default priority control and programmable priority control. Programmable priority control is used for multiple interrupt servicing.

In the interrupt enabled state, if an interrupt request with a priority equal to or higher than that of the interrupt currently being serviced is generated, it is acknowledged for multiple interrupt servicing. If an interrupt with a priority equal to or lower than that of the interrupt currently being serviced is generated during interrupt servicing, it is not acknowledged for multiple interrupt servicing. However, when setting the IE flag to 1 during the interruption at level 0, other level 0 interruptions can be allowed.

Interrupt requests that are not enabled because interrupts are in the interrupt disabled state or because they have a lower priority are held pending. When servicing of the current interrupt ends, the pending interrupt request is acknowledged following execution of at least one main processing instruction execution.

Table 22 - 7 shows Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing and Figures 22 - 13 and 22 - 14 show multiple interrupt servicing examples.

Table 22 - 7 Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing

Multiple Interrupt Request / Interrupt Being Serviced		Maskable Interrupt Request								Software Interrupt Request
		Priority Level 0 (PR = 00)		Priority Level 1 (PR = 01)		Priority Level 2 (PR = 10)		Priority Level 3 (PR = 11)		
		IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	
Maskable interrupt	ISP1 = 0 ISP0 = 0	√	×	×	×	×	×	×	×	√
	ISP1 = 0 ISP0 = 1	√	×	√	×	×	×	×	×	√
	ISP1 = 1 ISP0 = 0	√	×	√	×	√	×	×	×	√
	ISP1 = 1 ISP0 = 1	√	×	√	×	√	×	√	×	√
Software interrupt		√	×	√	×	√	×	√	×	√

Remark 1. √: Multiple interrupt servicing enabled

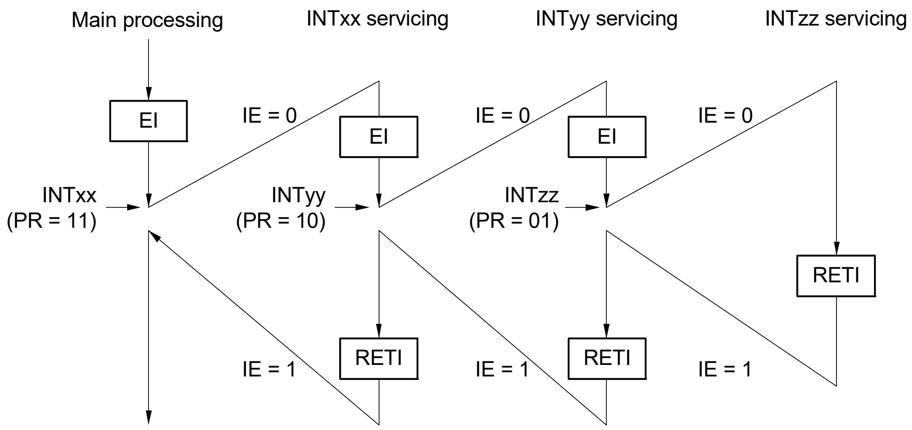
Remark 2. ×: Multiple interrupt servicing disabled

Remark 3. ISP0, ISP1, and IE are flags contained in the PSW.
ISP1 = 0, ISP0 = 0: An interrupt of level 1 or level 0 is being serviced.
ISP1 = 0, ISP0 = 1: An interrupt of level 2 is being serviced.
ISP1 = 1, ISP0 = 0: An interrupt of level 3 is being serviced.
ISP1 = 1, ISP0 = 1: Wait for An interrupt acknowledgment (all interrupts enabled).
IE = 0: Interrupt request acknowledgment is disabled.
IE = 1: Interrupt request acknowledgment is enabled.

Remark 4. PR is a flag contained in the PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H registers.
PR = 00: Specify level 0 with ××PR1× = 0, ××PR0× = 0 (higher priority level)
PR = 01: Specify level 1 with ××PR1× = 0, ××PR0× = 1
PR = 10: Specify level 2 with ××PR1× = 1, ××PR0× = 0
PR = 11: Specify level 3 with ××PR1× = 1, ××PR0× = 1 (lower priority level)

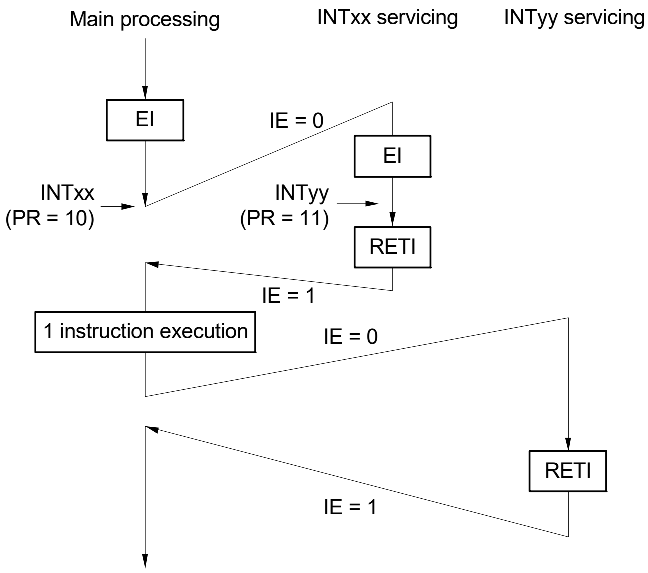
Figure 22 - 13 Examples of Multiple Interrupt Servicing (1/2)

Example 1. Multiple interrupt servicing occurs twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and multiple interrupt servicing takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledgment.

Example 2. Multiple interrupt servicing does not occur due to priority control

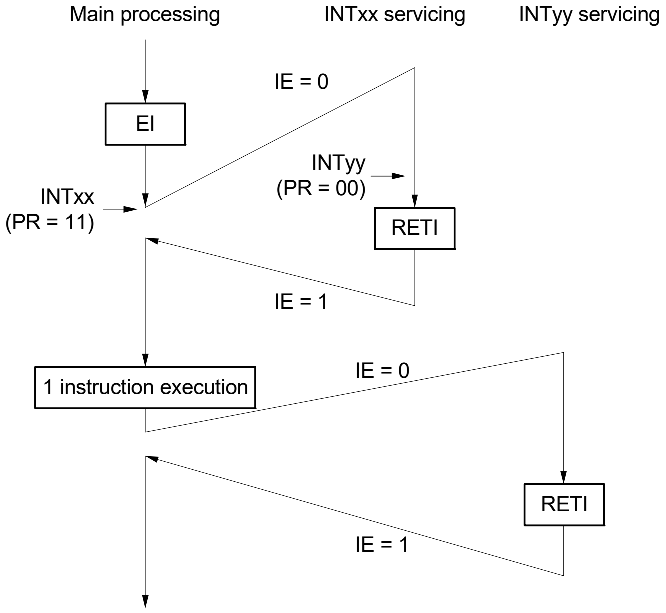


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with xxPR1x = 0, xxPR0x = 0 (higher priority level)
PR = 01: Specify level 1 with xxPR1x = 0, xxPR0x = 1
PR = 10: Specify level 2 with xxPR1x = 1, xxPR0x = 0
PR = 11: Specify level 3 with xxPR1x = 1, xxPR0x = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

Figure 22 - 14 Examples of Multiple Interrupt Servicing (2/2)

Example 3. Multiple interrupt servicing does not occur because interrupts are not enabled



Interrupts are not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with xxPR1x = 0, xxPR0x = 0 (higher priority level)
PR = 01: Specify level 1 with xxPR1x = 0, xxPR0x = 1
PR = 10: Specify level 2 with xxPR1x = 1, xxPR0x = 0
PR = 11: Specify level 3 with xxPR1x = 1, xxPR0x = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

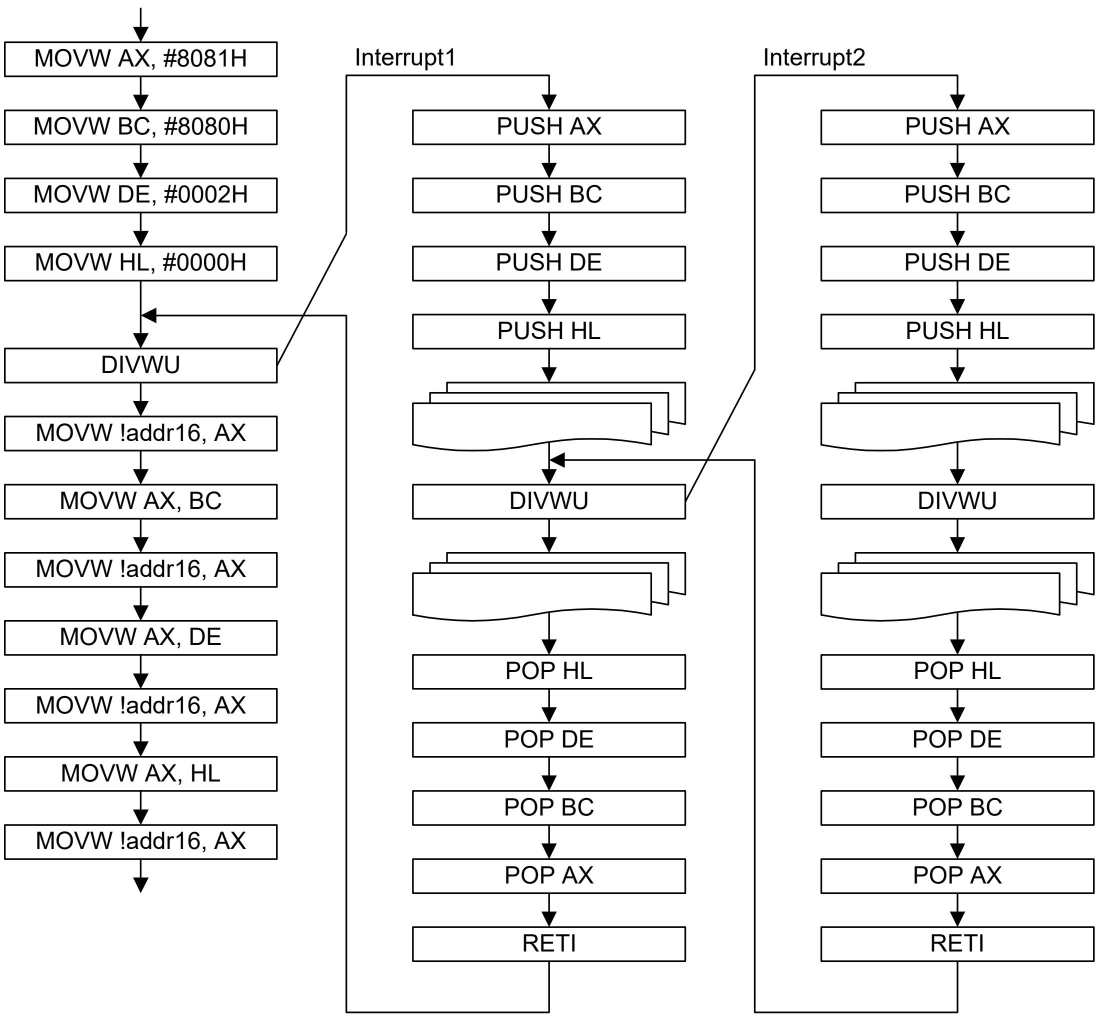
22.4.4 Interrupt servicing during division instruction

The RL78/I1E handles interrupts during the DIVHU/DIVWU instruction in order to enhance the interrupt response when a division instruction is executed.

• When an interrupt is generated while the DIVHU/DIVWU instruction is executed, the instruction is suspended
• After the instruction is suspended, the PC indicates the next instruction after DIVHU/DIVWU
• An interrupt is generated by the next instruction
• PC-3 is stacked to execute the DIVHU/DIVWU instruction again

Normal interrupt	Interrupts while Executing DIVHU/DIVWU Instruction
(SP-1) ← PSW	(SP-1) ← PSW
(SP-2) ← $(PC)_S$	(SP-2) ← $(PC-3)_S$
(SP-3) ← $(PC)_H$	(SP-3) ← $(PC-3)_H$
(SP-4) ← $(PC)_L$	(SP-4) ← $(PC-3)_L$
PC_S ← 0000	PC_S ← 0000
PC_H ← (Vector)	PC_H ← (Vector)
PC_L ← (Vector)	PC_L ← (Vector)
SP ← SP-4	SP ← SP-4
IE ← 0	IE ← 0

The AX, BC, DE, and HL registers are used for DIVHU/DIVWU. Use these registers by stacking them for interrupt servicing.



Caution **Disable interrupts when executing the DIVHU or DIVWU instruction in an interrupt servicing routine.**
Alternatively, unless they are executed in the RAM area, note that execution of a DIVHU or DIVWU instruction is possible even with interrupts enabled as long as a NOP instruction is added immediately after the DIVHU or DIVWU instruction in the assembly language source code. The following compilers automatically add a NOP instruction immediately after any DIVHU or DIVWU instruction output during the build process.
- V. 1.71 and later versions of the CA78K0R (Renesas Electronics compiler), for both C and assembly language source code
- Service pack 1.40.6 and later versions of the EWRL78 (IAR compiler), for C language source code
- GNURL78 (KPIT compiler), for C language source code

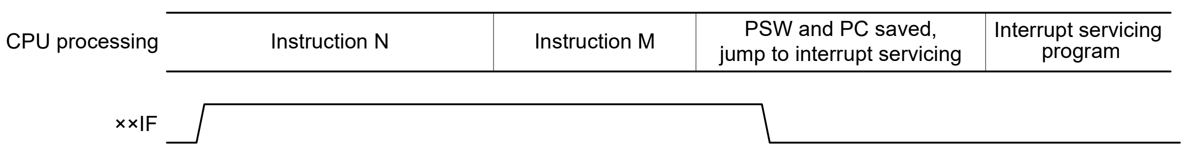
22.4.5 Interrupt request hold

There are instructions where, even if an interrupt request is issued while the instructions are being executed, interrupt request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, $addr20
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- SKH
- SKNH
- MULHU
- MULH
- MACHU
- MACH
- Write instructions for the IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H registers

Figure 22 - 15 shows the timing at which interrupt requests are held pending.

Figure 22 - 15 Interrupt Request Hold



Remark 1. Instruction N: Interrupt request hold instruction
Remark 2. Instruction M: Instruction other than interrupt request hold instruction

CHAPTER 23 STANDBY FUNCTION

23.1 Standby Function

The standby function reduces the operating current of the system, and the following three modes are available.

(1) HALT mode

HALT instruction execution sets the HALT mode. In the HALT mode, the CPU operation clock is stopped. If the high-speed system clock oscillator or high-speed on-chip oscillator is operating before the HALT mode is set, oscillation of each clock continues. In this mode, the operating current is not decreased as much as in the STOP mode, but the HALT mode is effective for restarting operation immediately upon interrupt request generation and carrying out intermittent operations frequently.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the high-speed system clock 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 exited when the X1 clock is selected, select the HALT mode if it is necessary to start processing immediately upon interrupt request generation.

(3) SNOOZE mode

In the case of CSIp or UARTq data reception, an A/D conversion request by the timer trigger signal (the interrupt request signal (INTIT) or ELC event input), and DTC start source, the STOP mode is exited, the CSIp or UARTq data is received without operating the CPU, A/D conversion is performed, and DTC start source. This can only be specified when the high-speed on-chip oscillator is selected for the CPU/peripheral hardware clock (f_{CLK}).

In either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latches and output buffer statuses are also held.

Caution 1. When shifting to the STOP mode, be sure to stop the peripheral hardware operation operating with main system clock before executing STOP instruction (except SNOOZE mode setting unit).

Caution 2. When using CSIp, UARTq, or the A/D converter in the SNOOZE mode, set up serial standby control register m (SSCm) and A/D converter mode register 2 (ADM2) before switching to the STOP mode. For details, see 19.3 Registers Controlling Serial Array Unit and 16.3 Registers Controlling A/D Converter.

Caution 3. 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 4. Whether to continue oscillating or stop the low-speed on-chip oscillator in the HALT or STOP mode can be selected by using the option byte. For details, see CHAPTER 29 OPTION BYTE.

Remark p = 00; q = 0; m = 0

23.2 Registers controlling standby function

The registers which control the standby function are described below.

• Subsystem clock supply mode control register (OSMC)
• Oscillation stabilization time counter status register (OSTC)
• Oscillation stabilization time select register (OSTS)

Remark For details of registers described above, see **CHAPTER 5 CLOCK GENERATOR**. For registers which control the SNOOZE mode, **CHAPTER 16 A/D CONVERTER** and **CHAPTER 19 SERIAL ARRAY UNIT**.

23.3 Standby Function Operation

23.3.1 HALT mode

(1) HALT mode

The HALT mode is set by executing the HALT instruction. HALT mode can be set regardless of whether the CPU clock before the setting was the high-speed system clock or high-speed on-chip oscillator clock.

The operating statuses in the HALT mode are shown below.

Caution Because the interrupt request signal is used to clear the HALT mode, if the interrupt mask flag is 0 (the interrupt processing is enabled) and the interrupt request flag is 1 (the interrupt request signal is generated), the HALT mode is not entered even if the HALT instruction is executed in such a situation.

Table 23 - 1 Operating Statuses in HALT Mode (1/2)

Item \ HALT Mode Setting			When HALT Instruction Is Executed While CPU Is Operating on Main System Clock			
			When CPU is operating on high-speed on-chip oscillator clock (f_{IH})	When CPU is operating on X1 clock (f_X)	When CPU is operating on external main system clock (f_{EX})	When CPU is operating on PLL clock (f_{PLL})
System clock			Clock supply to the CPU is stopped			
	Main system clock	f_{IH}	Operation continues (cannot be stopped)	Operation disabled		
		f_X	Operation disabled	Operation continues (cannot be stopped)	Cannot operate	Clock supply to the PLL cannot be stopped
		f_{EX}		Cannot operate	Operation continues (cannot be stopped)	
		f_{PLL}	Operation disabled			Operation continues (cannot be stopped)
	Low-speed on-chip oscillator clock	f_{IL}	Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply 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 (Operable while in the DTC is executed)			
Port (latch)			Status before HALT mode was set is retained			
Timer array unit			Operable			
Real-time clock (RTC)						
Interval timer						
Watchdog timer			See **CHAPTER 12 WATCHDOG TIMER**.			
Timer RJ			Operable			
Timer RG						
Clock output/buzzer output						
Serial array unit (SAU)						
Data transfer controller (DTC)						
Event link controller (ELC)			Operable function blocks can be linked			
Analog power supply			Status before HALT mode was set is retained			
Instrumentation amplifier			Operable			
24-bit ΔΣ A/D converter						
10-bit A/D converter						
12-bit D/A converter						
Configurable operational amplifier						
Power-on-reset function						
Voltage detection function						
External interrupt						
CRC operation function		High-speed CRC				
		General-purpose CRC	In the calculation of the RAM area, operable when DTC is executed only			
Illegal-memory access detection function			Operable when DTC is executed only			
RAM parity error detection function						
RAM guard function						
SFR guard function						

Remark Operation stopped: Operation is automatically stopped before switching to 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 f_{EX}: External main system clock
f_{PLL}: PLL clock

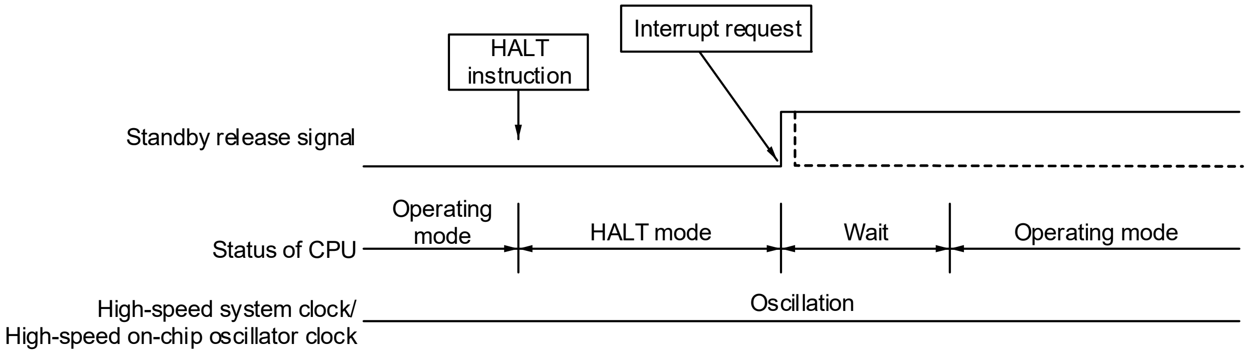
(2) HALT mode release

The HALT mode can be released by the following two sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the HALT mode is exited. If interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 23 - 1 HALT Mode Release by Interrupt Request Generation



Note 1. For details about the standby release signal, see **Figure 22 - 1 Basic Configuration of Interrupt Function**.

Note 2. Wait time for HALT mode release

- When vectored interrupt servicing is carried out
  Main system clock: 15 to 16 clock cycles
- When vectored interrupt servicing is not carried out
  Main system clock: 9 to 10 clock cycles

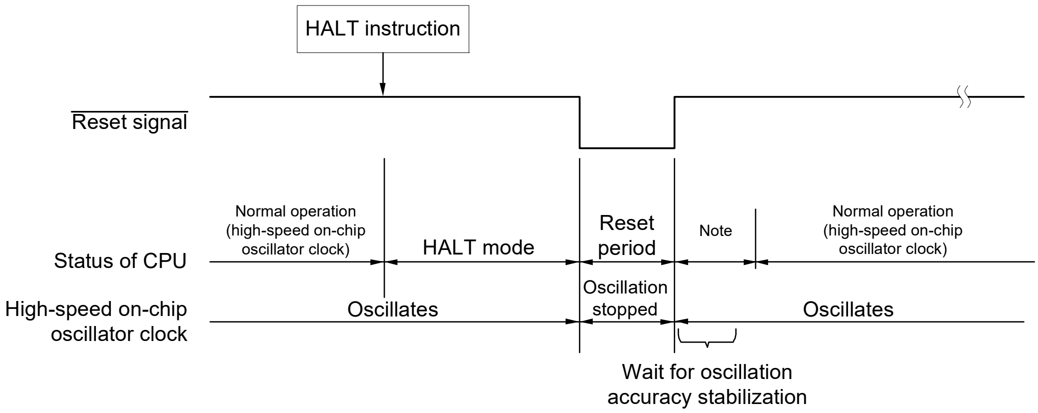
Remark The broken lines indicate the case when the interrupt request that has canceled the standby mode is acknowledged.

(b) Release by reset signal generation

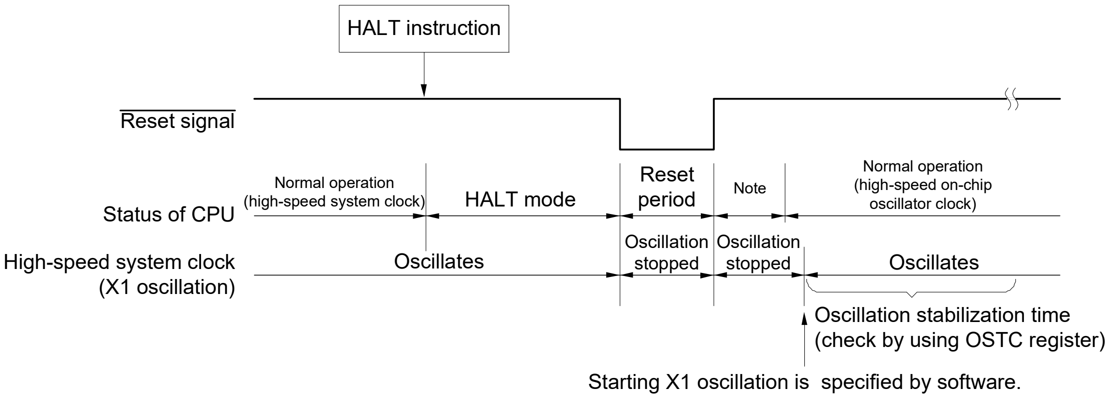
When the reset signal is generated, HALT mode is exited, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 23 - 2 HALT Mode Release by Reset

(1) When high-speed on-chip oscillator clock is used as CPU clock



(2) When high-speed system clock is used as CPU clock



Note For the reset processing time, see **CHAPTER 24 RESET FUNCTION**.
For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 25 POWER-ON-RESET CIRCUIT**.

23.3.2 STOP mode

(1) STOP mode setting and operating statuses

The STOP mode is set by executing the STOP instruction, and it can be set only when the CPU clock before the setting was the high-speed on-chip oscillator clock, X1 clock, or external main system clock.

Caution 1. 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.

Caution 2. When shifting to the STOP mode, stop PLL operation by using the DSCON bit of the DSCCTL register, and then execute the STOP instruction.

Remark p = 00; q = 0; m = 0

The operating statuses in the STOP mode are shown below.

Table 23 - 2 Operating Statuses in STOP Mode

Item (STOP Mode Setting)			When STOP Instruction Is Executed While CPU Is Operating on Main System Clock		
			When CPU is operating on high-speed on-chip oscillator clock (f_{IH})	When CPU is operating on X1 clock (f_X)	When CPU is operating on external main system clock (f_{EX})
System clock			Clock supply to the CPU is stopped		
	Main system clock	f_{IH}	Stopped		
		f_X			
		f_{EX}			
		f_{PLL}	Operation disabled		
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply mode control register (OSMC) • WUTMMCK0 = 1: Oscillates • 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					
Port (latch)			Status before STOP mode was set is retained		
Timer array unit			Operation disabled		
Real-time clock (RTC)			Operation disabled		
Interval timer			Operable when the low-speed on-chip oscillator is selected as the count source		
Watchdog timer			See **CHAPTER 12 WATCHDOG TIMER**.		
Timer RJ			• Operable in event count mode when TRJIO input with no filer is selected • Operable when the low-speed on-chip oscillator is selected as the count source • Operation is disabled under any conditions other than the above		
Timer RG			Operation disabled		
Clock output/buzzer output					
Serial array unit (SAU)			Wakeup operation is enabled only for CSIp and UARTq (switching to SNOOZE mode) Operation is disabled for anything other than CSIp and UARTq		
Data transfer controller (DTC)			DTC activation source receiving operation enabled (switching to SNOOZE mode)		
Event link controller (ELC)			Operable function blocks can be linked		
Analog power supply			Operable		
Instrumentation amplifier			Operation disabled		
24-bit ΔΣ A/D converter					
10-bit A/D converter			Wakeup operation is enabled (switching to SNOOZE mode)		
12-bit D/A converter			Operable (status before STOP mode was set is retained)		
Configurable operational amplifier					
Power-on-reset function			Operable		
Voltage detection function					
External interrupt					
CRC operation function	High-speed CRC		Operation stopped		
	General-purpose CRC				
Illegal-memory access detection function					
RAM parity error detection function					
RAM guard function					
SFR guard function					

Remark 1. Operation stopped: Operation is automatically stopped before switching to the STOP mode.
Operation disabled: Operation is stopped before switching to the STOP mode.
f_{IH}: High-speed on-chip oscillator clock f_{IL}: Low-speed on-chip oscillator clock
f_X: X1 clock f_{EX}: External main system clock
f_{PLL}: PLL clock

Remark 2. p = 00; q = 0

(2) STOP mode release

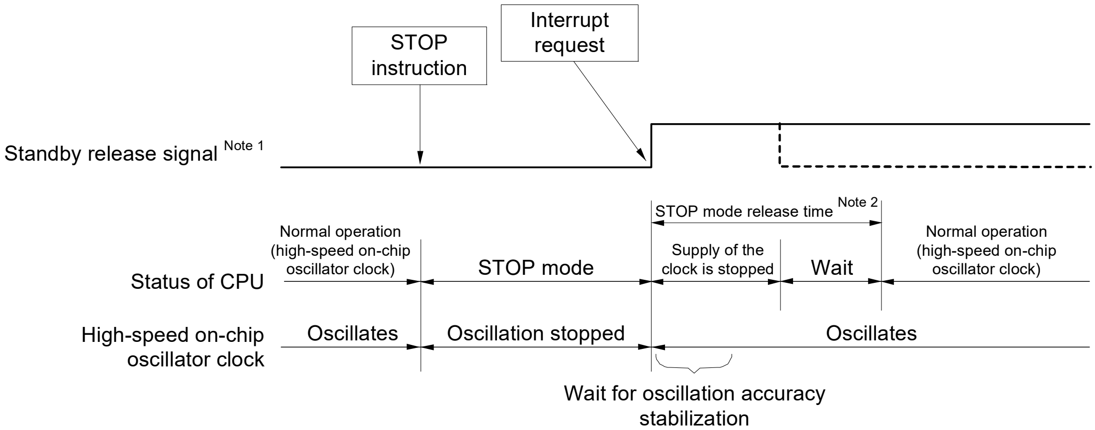
The STOP mode can be released by the following two sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the STOP mode is exited. After the oscillation stabilization time has elapsed, if interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 23 - 3 STOP Mode Release by Interrupt Request Generation (1/2)

(1) When high-speed on-chip oscillator clock is used as CPU clock



Note 1. For details of the standby release signal, see **Figure 22 - 1 Basic Configuration of Interrupt Function**.

Note 2. STOP mode release time

Supply of the clock is stopped:

- 18 µs to 65 µs

Wait:

- When vectored interrupt servicing is carried out: 7 clock cycles
- When vectored interrupt servicing is not carried out: 1 clock cycle

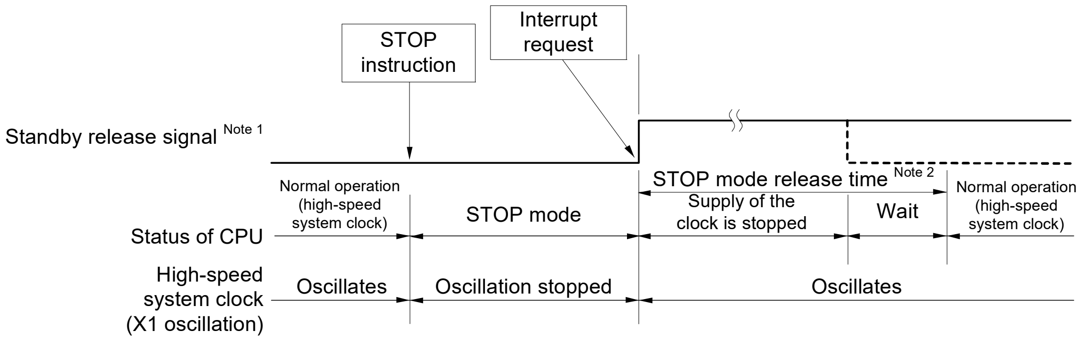
Caution **To shorten the oscillation stabilization time after the STOP mode is exited while the CPU is operating with the high-speed system clock (X1 oscillation), switch the CPU clock to the high-speed on-chip oscillator clock temporarily before executing the STOP instruction.**

Remark 1. The clock supply stop time varies depending on the temperature conditions and STOP mode period.

Remark 2. The broken lines indicate the case when the interrupt request that has canceled the standby mode is acknowledged.

Figure 23 - 4 STOP Mode Release by Interrupt Request Generation (2/2)

(2) When high-speed system clock (X1 oscillation) is used as CPU clock



Note 1. For details of the standby release signal, see **Figure 22 - 1 Basic Configuration of Interrupt Function**.

Note 2. STOP mode release time

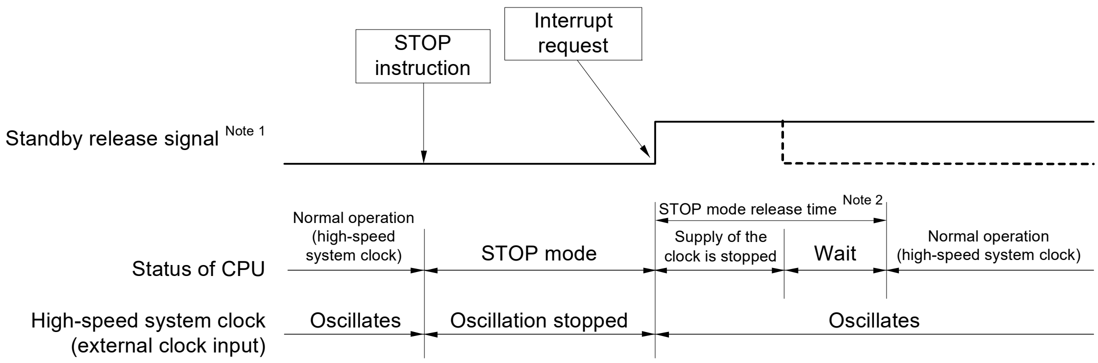
Supply of the clock is stopped:

- 18 µs to “whichever is longer 65 µs or the oscillation stabilization time (set by OSTS)”

Wait:

- When vectored interrupt servicing is carried out: 10 to 11 clock cycles
- When vectored interrupt servicing is not carried out: 4 to 5 clock cycles

(3) When high-speed system clock (external clock input) is used as CPU clock



Note 1. For details of the standby release signal, see **Figure 22 - 1 Basic Configuration of Interrupt Function**.

Note 2. STOP mode release time

Supply of the clock is stopped:

- 18 µs to 65 µs

Wait:

- When vectored interrupt servicing is carried out: 7 clock cycles
- When vectored interrupt servicing is not carried out: 1 clock cycle

Caution To shorten the the oscillation stabilization time after the STOP mode is exited while the CPU is operating 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.

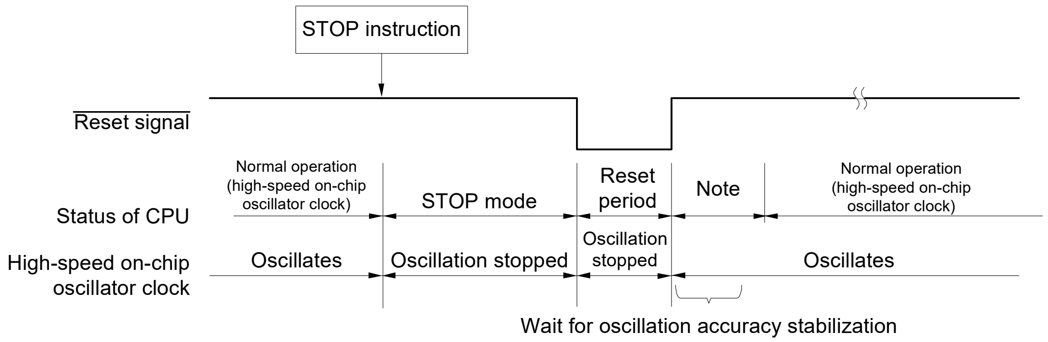
Remark 2. The broken lines indicate the case when the interrupt request that has canceled the standby mode is acknowledged.

(b) Release by reset signal generation

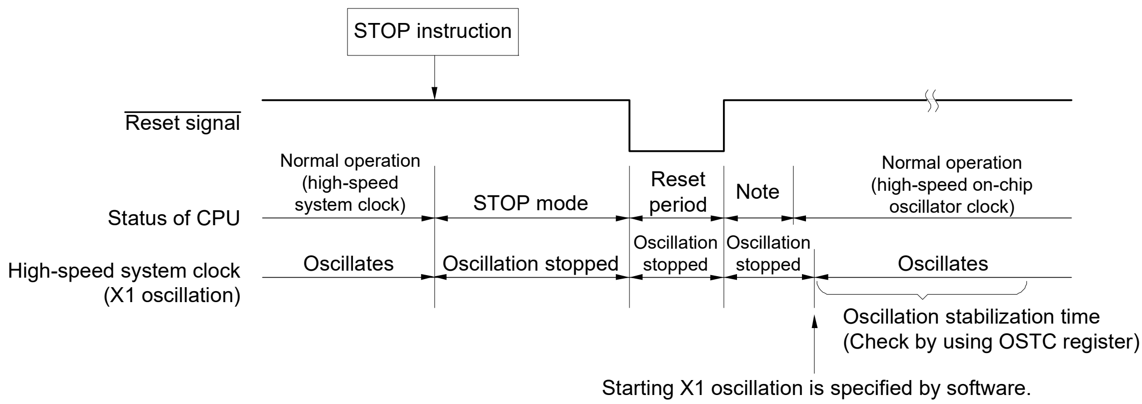
When the reset signal is generated, STOP mode is exited, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 23 - 5 STOP Mode Release by Reset

(1) When high-speed on-chip oscillator clock is used as CPU clock



(2) When high-speed system clock is used as CPU clock



Note For the reset processing time, see **CHAPTER 24 RESET FUNCTION**.
For the reset processing time of the power-on-reset circuit (POR) and voltage detector (LVD), see **CHAPTER 25 POWER-ON-RESET CIRCUIT**.

23.3.3 SNOOZE mode

(1) SNOOZE mode setting and operating statuses

The SNOOZE mode can only be specified for CSIp, the A/D converter, or DTC. This mode can only be specified if the CPU clock is the high-speed on-chip oscillator clock.

When using CSIp or UARTq in the SNOOZE mode, set up serial standby control register m (SSCm) before switching to the STOP mode. For details, see **19.3 Registers Controlling Serial Array Unit**.

When using the A/D converter in the SNOOZE mode, set up A/D converter mode register 2 (ADM2) before switching to the STOP mode. For details, see **16.3 Registers Controlling A/D Converter**.

When DTC transfer is used in SNOOZE mode, before switching to the STOP mode, allow DTC activation by interrupt to be used. During STOP mode, detecting DTC activation by interrupt enables DTC transit to SNOOZE mode, automatically. For details, see **20.3 Registers Controlling DTC**.

Remark p = 00; q = 0; m = 0

In SNOOZE mode transition, wait status to be only following time.

Transition time from STOP mode to SNOOZE mode

18 μs to 65 μs

Remark Transition time from STOP mode to SNOOZE mode varies depending on the temperature conditions and the STOP mode period.

Transition time from SNOOZE mode to normal operation:

• When vectored interrupt servicing is carried out:

H“4.99 μs to 9.44 μs” + 7 clock cycles

• When vectored interrupt servicing is not carried out:

“4.99 μs to 9.44 μs” + 1 clock cycle

The operating statuses in the SNOOZE mode are shown next.

Table 23 - 3 Operating Statuses in SNOOZE Mode

<table>
<tr><td colspan="3">STOP Mode Setting
Item</td><td>During STOP mode, receiving data signal from CSIp and UARTq, inputting timer trigger signal to A/D converter, and generating DTC activation by interrupt
When CPU is operating on high-speed on-chip oscillator clock (fIH)</td></tr>
<tr><td colspan="3">System clock</td><td>Clock supply to the CPU is stopped</td></tr>
<tr><td rowspan="5"></td><td rowspan="4">Main system clock</td><td>fIH</td><td>Operation started</td></tr>
<tr><td>fX</td><td rowspan="3">Stopped</td></tr>
<tr><td>fEX</td></tr>
<tr><td>fPLL</td></tr>
<tr><td colspan="2">fIL</td><td>Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H), and WUTMMCK0 bit of subsystem clock supply mode control register (OSMC)
• WUTMMCK0 = 1: Oscillates
• WUTMMCK0 = 0 and WDTON = 0: Stops
• WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 1: Oscillates
• WUTMMCK0 = 0, WDTON = 1, and WDSTBYON = 0: Stops</td></tr>
<tr><td colspan="3">CPU</td><td rowspan="3">Operation stopped</td></tr>
<tr><td colspan="3">Code flash memory</td></tr>
<tr><td colspan="3">Data flash memory</td></tr>
<tr><td colspan="3">RAM</td><td>Operation stopped (Operable while in the DTC is executed)</td></tr>
<tr><td colspan="3">Port (latch)</td><td>Use of the status while in the STOP mode continues</td></tr>
<tr><td colspan="3">Timer array unit</td><td>Operation disabled</td></tr>
<tr><td colspan="3">Real-time clock (RTC)</td><td rowspan="2">Operable</td></tr>
<tr><td colspan="3">Interval timer</td></tr>
<tr><td colspan="3">Watchdog timer</td><td>See CHAPTER 12 WATCHDOG TIMER.</td></tr>
<tr><td colspan="3">Timer RJ</td><td>• Operable in event count mode when TRJIO input with no filer is selected
• Operable when the low-speed on-chip oscillator is selected as the count source
• Operation is disabled under any conditions other than the above</td></tr>
<tr><td colspan="3">Timer RG</td><td rowspan="2">Operation disabled</td></tr>
<tr><td colspan="3">Clock output/buzzer output</td></tr>
<tr><td colspan="3">Serial array unit (SAU)</td><td>Only CSIp and UARTq operable.
Operation disabled for channels other than CSIp and UARTq.</td></tr>
<tr><td colspan="3">Data transfer controller (DTC)</td><td>Operable</td></tr>
<tr><td colspan="3">Event link controller (ELC)</td><td>Operable function blocks can be linked</td></tr>
<tr><td colspan="3">Analog power supply</td><td>Operable</td></tr>
<tr><td colspan="3">Instrumentation amplifier</td><td rowspan="2">Operation disabled</td></tr>
<tr><td colspan="3">24-bit ΔΣ A/D converter</td></tr>
<tr><td colspan="3">10-bit A/D converter</td><td>Operable</td></tr>
<tr><td colspan="3">12-bit D/A converter</td><td rowspan="2">Operable (status before SNOOZE mode was set is retained)</td></tr>
<tr><td colspan="3">Configurable operational amplifier</td></tr>
<tr><td colspan="3">Power-on-reset function</td><td rowspan="3">Operable</td></tr>
<tr><td colspan="3">Voltage detection function</td></tr>
<tr><td colspan="3">External interrupt</td></tr>
<tr><td colspan="2" rowspan="2">CRC operation function</td><td>High-speed CRC</td><td rowspan="6">Operation stopped</td></tr>
<tr><td>General-purpose CRC</td></tr>
<tr><td colspan="3">Illegal-memory access detection function</td></tr>
<tr><td colspan="3">RAM parity error detection function</td></tr>
<tr><td colspan="3">RAM guard function</td></tr>
<tr><td colspan="3">SFR guard function</td></tr>
</table>

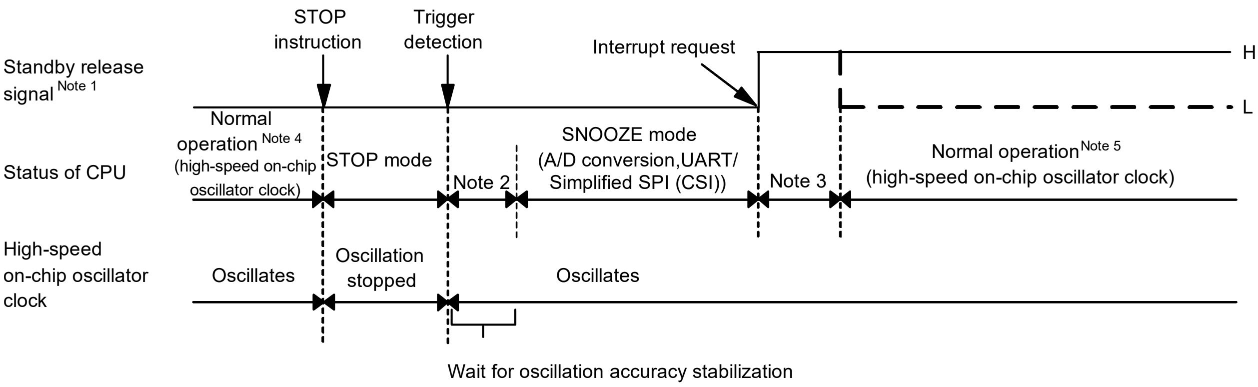
(Remarks are listed on the next page)

Remark 1. Operation stopped: Operation is automatically stopped before switching to the STOP mode.
Operation disabled: Operation is stopped before switching to the STOP mode.
f_{IH}: High-speed on-chip oscillator clock
f_{IL}: Low-speed on-chip oscillator clock
f_X: X1 clock
f_{EX}: External main system clock
f_{PLL}: PLL clock

Remark 2. p = 00; q = 0

(2) Timing diagram when the interrupt request signal is generated in the SNOOZE mode

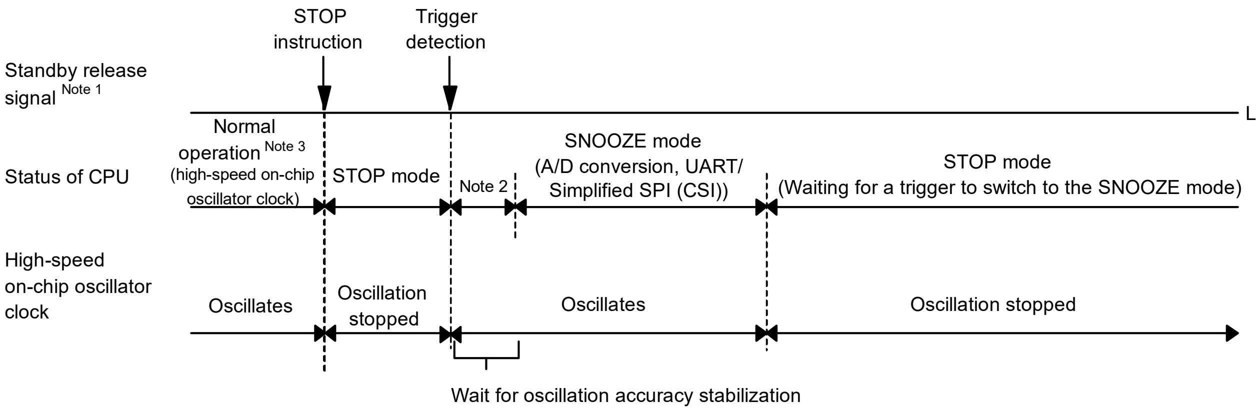
Figure 23 - 6 When the Interrupt Request Signal is Generated in the SNOOZE Mode



Note 1. For details of the standby release signal, see **Figure 22 - 1**.
Note 2. Transition time from STOP mode to SNOOZE mode
Note 3. Transition time from SNOOZE mode to normal operation
Note 4. Enable the SNOOZE mode (AWC = 1 or SWC = 1) immediately before switching to the STOP mode.
Note 5. Be sure to release the SNOOZE mode (AWC = 0 or SWC = 0) immediately after return to the normal operation.

(3) Timing diagram when the interrupt request signal is not generated in the SNOOZE mode

Figure 23 - 7 When the Interrupt Request Signal is not Generated in the SNOOZE Mode



Note 1. For details of the standby release signal, see **Figure 22 - 1**.
Note 2. Transition time from STOP mode to SNOOZE mode
Note 3. Enable the SNOOZE mode (AWC = 1 or SWC = 1) immediately before switching to the STOP mode.

Remark For details about the SNOOZE mode function, see **CHAPTER 16 A/D CONVERTER** and **CHAPTER 19 SERIAL ARRAY UNIT**.

CHAPTER 24 RESET FUNCTION

The following seven operations are available to generate a reset signal.

(1) External reset input via $\overline{\text{RESET}}$ pin
(2) Internal reset by watchdog timer program loop detection
(3) Internal reset by comparison of supply voltage and detection voltage of power-on-reset (POR) circuit
(4) Internal reset by comparison of supply voltage of the voltage detector (LVD) and detection voltage
(5) Internal reset by execution of illegal instruction Note
(6) Internal reset by RAM parity error
(7) Internal reset by illegal-memory access

External and internal resets start program execution from the address at 0000H and 0001H when the reset signal is generated.
A reset is effected when a low level is input to the $\overline{\text{RESET}}$ pin, the watchdog timer overflows, or by POR and LVD circuit voltage detection, execution of illegal instruction Note, RAM parity error or illegal-memory access, and each item of hardware is set to the status shown in Table 24 - 1.

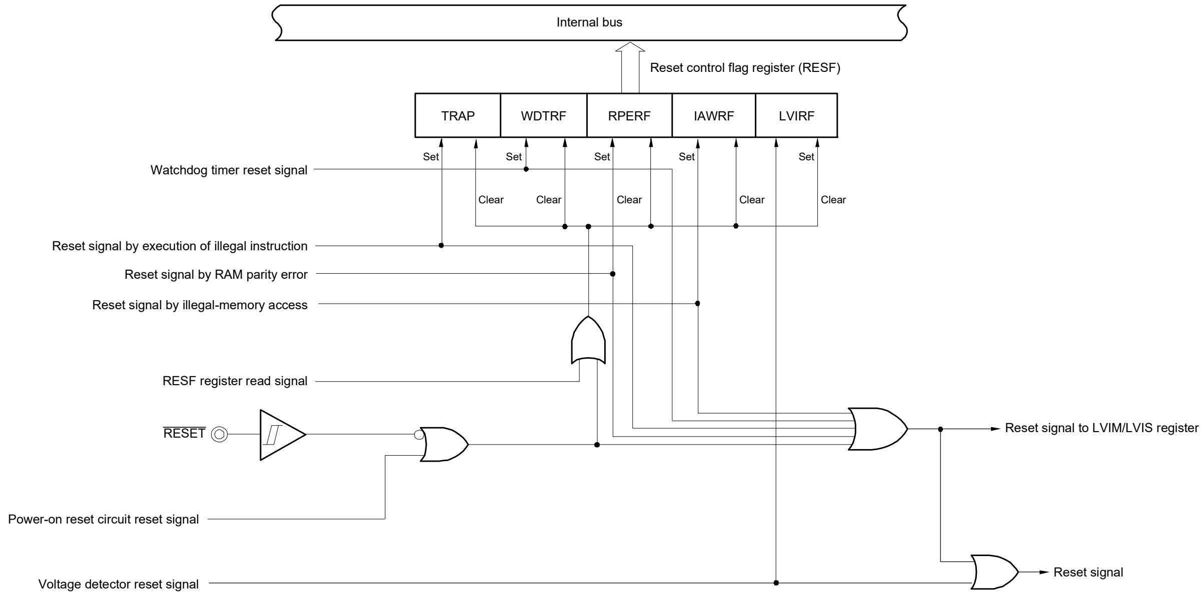
Note The illegal instruction is generated when instruction code FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Caution 1. For an external reset, input a low level for 10 μs or more to the $\overline{\text{RESET}}$ pin.
To perform an external reset upon power application, input a low level to the $\overline{\text{RESET}}$ pin, turn power on, continue to input a low level to the pin for 10 μs or more within the operating voltage range shown in 33.4 or 34.4 AC Characteristics, and then input a high level to the pin.

Caution 2. During reset input, the X1 clock, high-speed on-chip oscillator clock, and low-speed on-chip oscillator clock stop oscillating. External main system clock input becomes invalid.

Caution 3. The port pins become the following state because each SFR and 2nd SFR are initialized after reset.
- **P40: High-impedance during the external reset period or reset period by the POR. High level during other types of reset or after receiving a reset signal (connected to an on-chip pull-up resistor).**
- **Ports other than P40: High-impedance during the reset period or after receiving a reset signal.**

Figure 24 - 1 Block Diagram of Reset Function



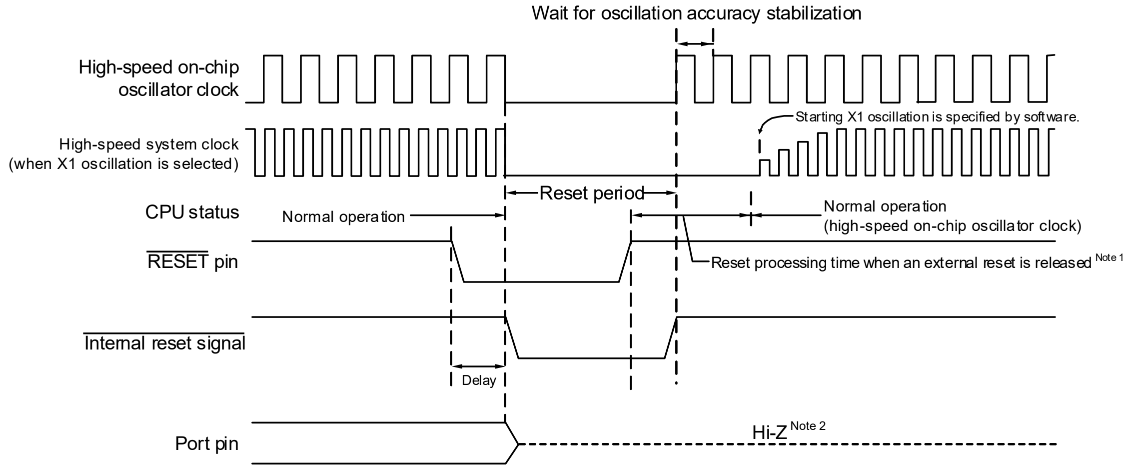
Caution An LVD circuit internal reset does not reset the LVD circuit.

Remark 1. LVIM: Voltage detection register
Remark 2. LVIS: Voltage detection level register

24.1 Timing of Reset Operation

This LSI is reset by input of the low level on the RESET pin and released from the reset state by input of the high level on the RESET pin. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts.

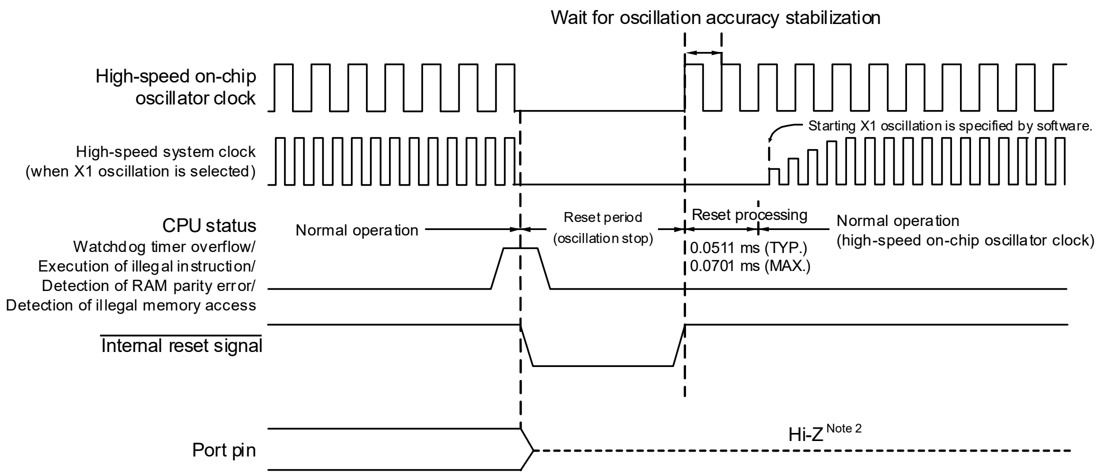
Figure 24 - 2 Timing of Reset by RESET Input



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

Release from the reset state is automatic in the case of a reset due to a watchdog timer overflow, execution of an illegal instruction, detection of a RAM parity error, or detection of illegal memory access. After reset processing, program execution starts with the high-speed on-chip oscillator clock as the operating clock.

Figure 24 - 3 Timing of Reset Due to Watchdog Timer Overflow, Execution of Illegal Instruction, Detection of RAM Parity Error, or Detection of Illegal Memory Access



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

Note 1. Reset times (times for release from the external reset state)

After the first release of the POR: 0.672 ms (TYP.), 0.832 ms (MAX.) when the LVD is in use.
0.399 ms (TYP.), 0.519 ms (MAX.) when the LVD is off.

After the second release of the POR: 0.531 ms (TYP.), 0.675 ms (MAX.) when the LVD is in use.
0.259 ms (TYP.), 0.362 ms (MAX.) when the LVD is off.

After power is supplied, a voltage stabilization waiting time of about 0.99 ms (TYP.) and up to 2.30 ms (MAX.) is required before reset processing starts after release of the external reset.

Note 2. The state of P40 is as follows.

- High-impedance during the external reset period or reset period by the POR.
- High level during other types of reset or after receiving a reset signal (connected to the on-chip pull-up resistance).

Reset by POR and LVD circuit supply voltage detection is automatically released when $V_{DD} \geq V_{POR}$ or $V_{DD} \geq V_{LVD}$ after the reset. After reset processing, execution of the program with the high-speed on-chip oscillator clock as the operating clock starts. For details, see **CHAPTER 25 POWER-ON-RESET CIRCUIT** or **CHAPTER 26 VOLTAGE DETECTOR**.

Remark V_{POR}: POR power supply rise detection voltage
V_{LVD}: LVD detection voltage

Table 24 - 1 Operation Statuses During Reset Period

Item			During Reset Period
System clock			Clock supply to the CPU is stopped.
	Main system clock	f_{IH}	Operation stopped
		f_X	Operation stopped (the X1 and X2 pins are input port mode)
		f_{EX}	Clock input invalid (the pin is input port mode)
		f_{PLL}	Operation stopped
	f_{IL}		Operation stopped
CPU			Operation stopped
Code flash memory			
Data flash memory			
RAM			
Port (latch)			Pins other than P40 outputs high impedance. P40 outputs high impedance during the external reset period or reset period by the POR. P40 outputs high level during other types of reset (when connected to an on-chip pull-up resistor).
Timer array unit			Operation stopped
Timer RJ			
Timer RG			
Real-time clock (RTC)			
Interval timer			
Watchdog timer			
Clock output/buzzer output			
Serial array unit (SAU)			
Data transfer controller (DTC)			
Event link controller (ELC)			
Analog power supply			
Instrumentation amplifier			
24-bit ΔΣ A/D converter			
10-bit A/D converter			
12-bit D/A converter			
Configurable operational amplifier			
Power-on-reset function			Detection operation possible
Voltage detection function			Operation is possible in the case of an LVD reset and stopped in the case of other types of reset.
External interrupt			Operation stopped
CRC operation function	High-speed CRC		
	General-purpose CRC		
Illegal-memory access detection function			
RAM parity error detection function			
RAM guard function			
SFR guard function			

Remark f_{IH}: High-speed on-chip oscillator clock
f_X: X1 oscillation clock
f_{EX}: External main system clock
f_{PLL}: PLL clock
f_{IL}: Low-speed on-chip oscillator clock

Table 24 - 2 Hardware Statuses After Reset Acknowledgment

Hardware		After Reset Acknowledgment Note
Program counter (PC)		The contents of the reset vector table (0000H, 0001H) are set.
Stack pointer (SP)		Undefined
Program status word (PSW)		06H
RAM	Data memory	Undefined
	General-purpose registers	Undefined

Note During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

Remark For the state of the special function register (SFR) after receiving a reset signal, see **3.1.4 Special function register (SFR) area** and **3.1.5 Extended special function register (2nd SFR: 2nd Special Function Register) area**.

24.2 Register for Confirming Reset Source

24.2.1 Reset control flag register (RESF)

Many internal reset generation sources exist in the RL78 microcontroller. The reset control flag register (RESF) is used to store which source has generated the reset request.

The RESF register can be read by an 8-bit memory manipulation instruction.

$\overline{\text{RESET}}$ input, reset by power-on-reset (POR) circuit, and reading the RESF register clear TRAP, WDTRF, RPERF, IAWRF, and LVIRF flags.

Figure 24 - 4 Format of Reset control flag register (RESF)

Address: FFFA8H After reset: Undefined Note 1 R

Symbol	7	6	5	4	3	2	1	0
RESF	TRAP	0	0	WDTRF	0	RPERF	IAWRF	LVIRF

TRAP	Internal reset request by execution of illegal instruction Note 2
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

WDTRF	Internal reset request by watchdog timer (WDT)
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

RPERF	Internal reset request t by RAM parity
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

IAWRF	Internal reset request t by illegal-memory access
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

LVIRF	Internal reset request by voltage detector (LVD)
0	Internal reset request is not generated, or the RESF register is cleared.
1	Internal reset request is generated.

Note 1. The value after reset varies depending on the reset source. See **Table 24 - 3**.

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

Caution 1. Do not read data by a 1-bit memory manipulation instruction.

Caution 2. When enabling RAM parity error resets (RPERDIS = 0), be sure to initialize the used RAM area at data access or the used RAM area + 10 bytes at execution of instruction from the RAM area.
Reset generation enables RAM parity error resets (RPERDIS = 0). For details, see 27.3.3 RAM parity error detection function.

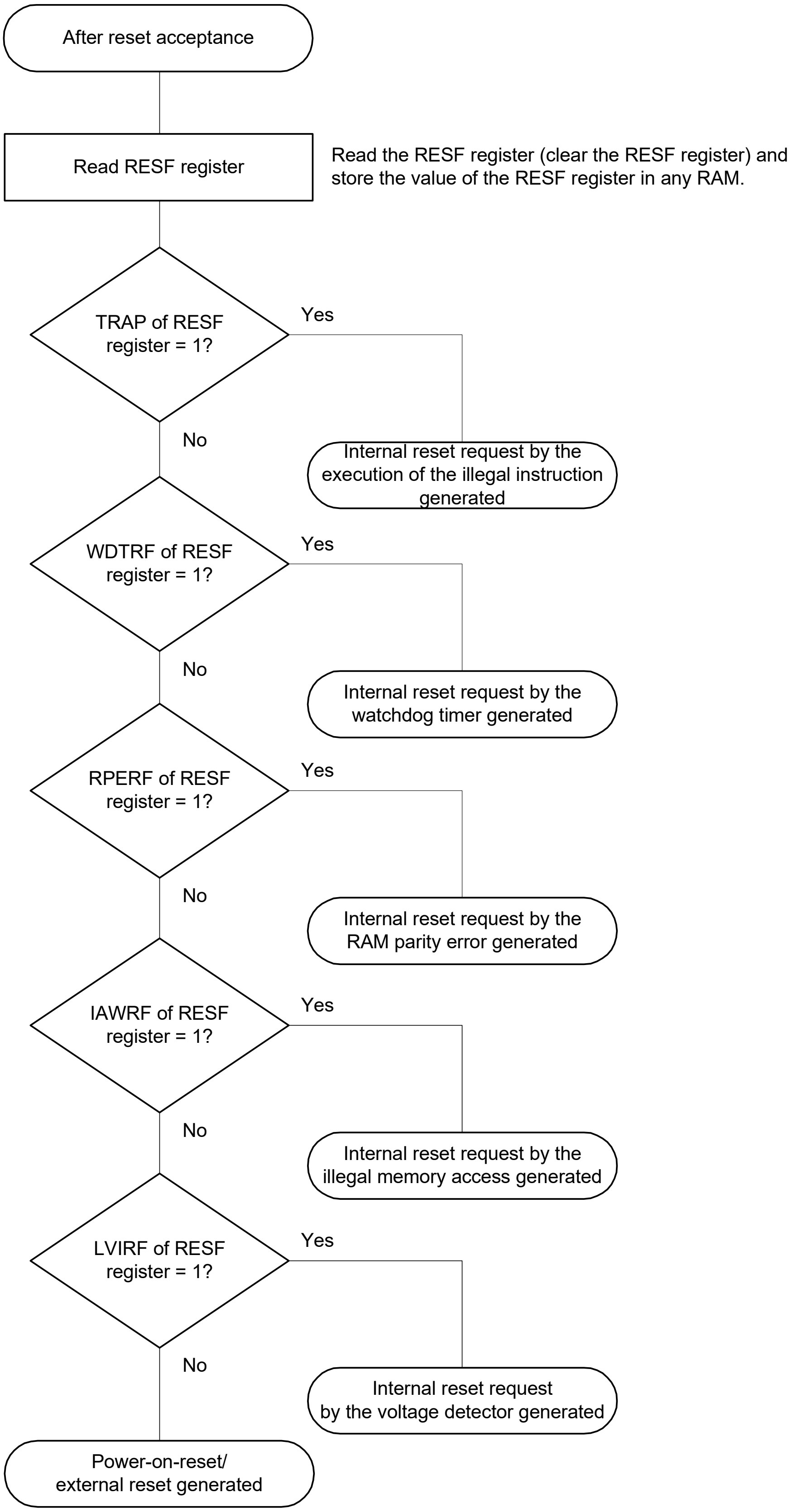
The status of the RESF register when a reset request is generated is shown in Table 24 - 3.

Table 24 - 3 RESF Register Status When Reset Request Is Generated

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

The RESF register is automatically cleared when it is read by an 8-bit memory manipulation instruction. Figure 24 - 5 shows the procedure for checking a reset source.

Figure 24 - 5 Example of Procedure for Checking Reset Source



CHAPTER 25 POWER-ON-RESET CIRCUIT

25.1 Functions of Power-on-Reset Circuit

The power-on-reset circuit (POR) has the following functions.

- Generates internal reset signal at power on.
  The reset signal is released when the supply voltage (V_{DD}) exceeds the detection voltage (V_{POR} = 1.56 V (TYP.)). Note that the reset state must be retained until the operating voltage becomes in the range defined in **33.4** or **34.4 AC Characteristics**.
  This is done by utilizing the voltage detector or controlling the externally input reset signal.

- Compares supply voltage (V_{DD}) and detection voltage (V_{PDR} = 1.55 V (TYP.)), generates internal reset signal when $V_{DD} < V_{PDR}$. Note that, after power is supplied, this LSI should be placed in the STOP mode, or in the reset state by utilizing the voltage detector or externally input reset signal, before the operation voltage falls below the range defined in **33.4** or **34.4 AC Characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Caution If an internal reset signal is generated in the power-on-reset circuit, the reset control flag register (RESF) is cleared.

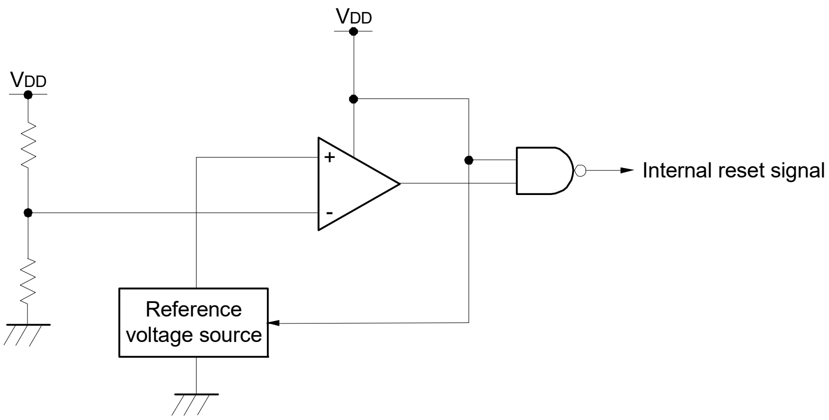
Remark 1. The RL78 microcontroller incorporates multiple hardware functions that generate an internal reset signal. A flag that indicates the reset source is located in the reset control flag register (RESF) for when an internal reset signal is generated by the watchdog timer (WDT), voltage-detector (LVD), illegal instruction execution, RAM parity error, or illegal-memory access. The RESF register is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by the watchdog timer (WDT), voltage-detector (LVD), illegal instruction execution, RAM parity error, or illegal-memory access.
For details of the RESF register, see **CHAPTER 24 RESET FUNCTION**.

Remark 2. V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage
For details, see **33.6.7** or **34.6.7 POR characteristics**.

25.2 Configuration of Power-on-Reset Circuit

The block diagram of the power-on-reset circuit is shown in Figure 25 - 1.

Figure 25 - 1 Block Diagram of Power-on-Reset Circuit

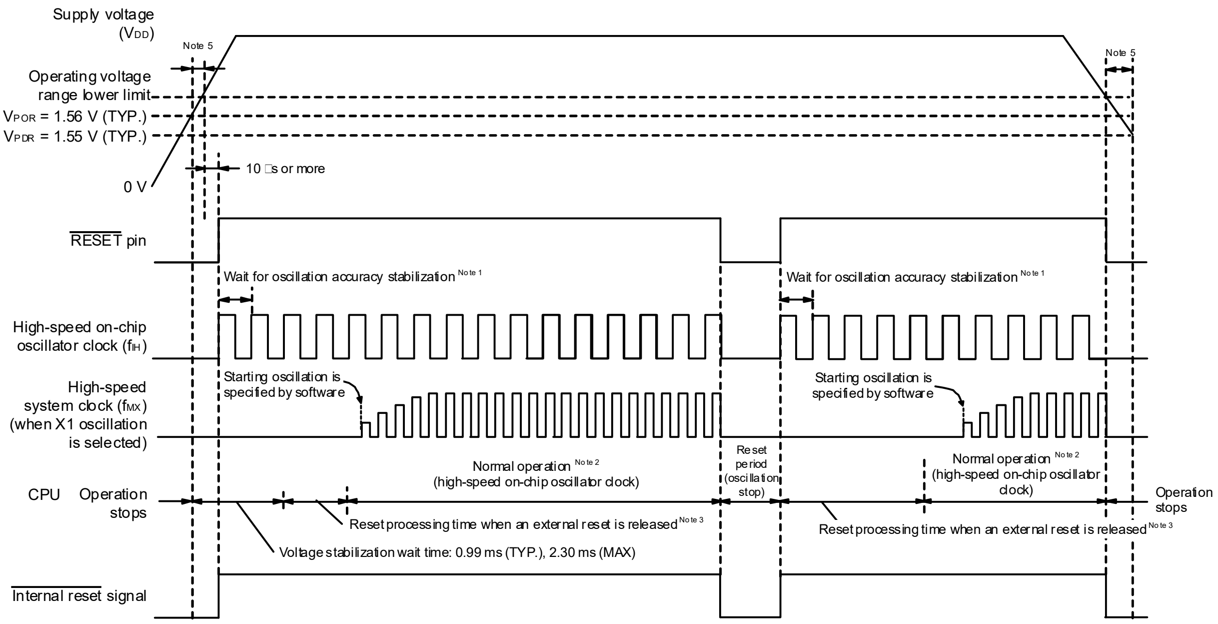


25.3 Operation of Power-on-Reset Circuit

The timing of generation of the internal reset signal by the power-on-reset circuit and voltage detector is shown below.

Figure 25 - 2 Timing of Generation of Internal Reset Signal by Power-on-Reset Circuit and Voltage Detector (1/3)

(1) When using an external reset by the $\overline{\text{RESET}}$ pin



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

Note 2. The high-speed on-chip oscillator clock and high-speed system clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time.

Note 3. The time until normal operation starts includes the following reset processing time when the external reset is released (release from the first external reset following release from the POR state) after the $\overline{\text{RESET}}$ signal is driven high (1) as well as the voltage stabilization wait time after V_{POR} (1.56 V (TYP.) is reached.
Reset processing time when the external reset is released is shown below.
Release from the first external reset following release from the POR state:
0.672 ms (TYP.), 0.832 ms (MAX.) (when the LVD is in use)
0.399 ms (TYP.), 0.519 ms (MAX.) (when the LVD is off)

Note 4. Reset times in cases of release from an external reset other than the above are listed below.
Release from the reset state for external resets other than the above case:
0.531 ms (TYP.), 0.675 ms (MAX.) (when the LVD is in use)
0.259 ms (TYP.), 0.362 ms (MAX.) (when the LVD is off)

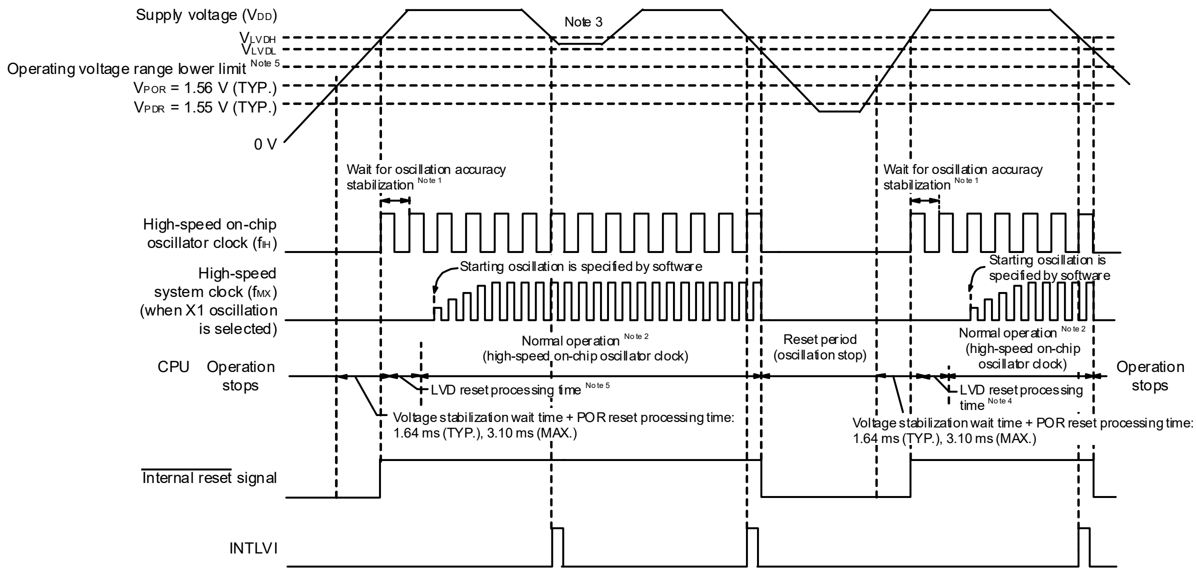
Note 5. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **33.4** or **34.4 AC Characteristics**. This is done by controlling the externally input reset signal.
After power supply is turned off, this LSI should be placed in the STOP mode, or in the reset state by utilizing the voltage detector or externally input reset signal, before the voltage falls below the operating range. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Caution **For power-on reset, be sure to use the externally input reset signal on the $\overline{\text{RESET}}$ pin when the LVD is off. For details, see CHAPTER 26 VOLTAGE DETECTOR.**

Figure 25 - 3 Timing of Generation of Internal Reset Signal by Power-on-Reset Circuit and Voltage Detector (2/3)

(2) LVD is interrupt & reset mode (option byte 000C1: LVIMDS1, LVIMDS0 = 1, 0)



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

Note 2. The high-speed on-chip oscillator clock or a high-speed system clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time.

Note 3. After the interrupt request signal (INTLVI) is generated, the LVILV and LVIMD bits of the voltage detection level register (LVIS) are automatically set to 1. After INTLVI is generated, appropriate settings should be made according to **Figure 26 - 10 Setting Procedure for Operating Voltage Check and Reset** and **Figure 26 - 11 Setting Procedure for Initial Setting of Interrupt and Reset Mode**, taking into consideration that the supply voltage might return to the high voltage detection level (V_{LVDH}) or higher without falling below the low voltage detection level (V_{LVDL}).

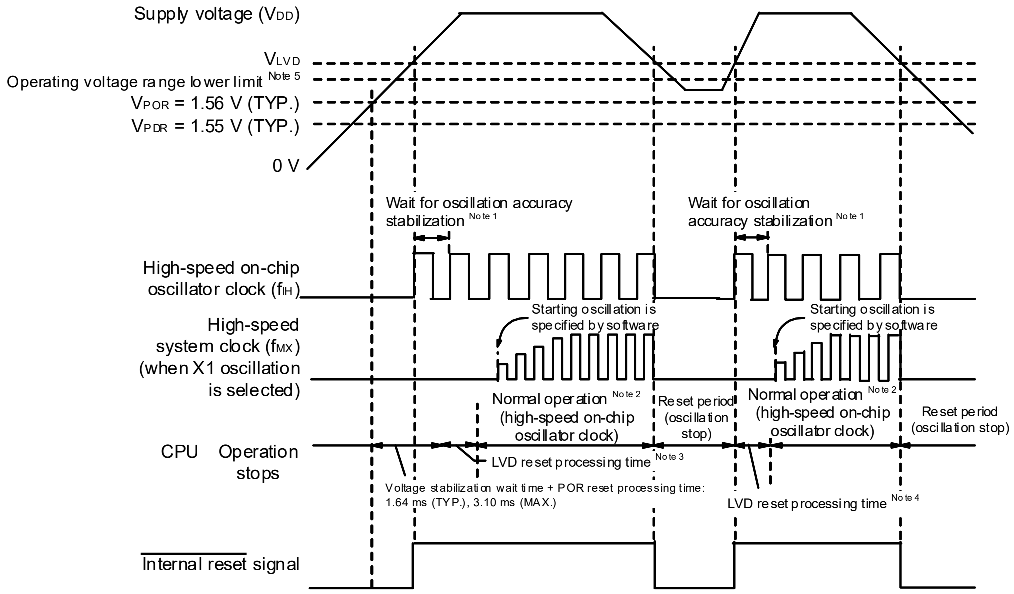
Note 4. The time until normal operation starts includes the following LVD reset processing time after the LVD detection level (V_{LVDH}) is reached as well as the voltage stabilization wait + POR reset processing time after the V_{POR} (1.56 V, TYP.) is reached.
LVD reset processing time: 0 ms to 0.0701 ms (MAX.)

Note 5. The operation guaranteed voltage range is 2.4 V ≤ V_{DD} ≤ 5.5 V. Be sure to start normal operations after the voltage reaches 2.4 V. To reset the system when the voltage falls below 2.7 V while the voltage level is falling, use the reset function of the voltage detector or input a low level to the $\overline{\text{RESET}}$ pin.

Remark V_{LVDH}, V_{LVDL}: LVD detection voltage
V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Figure 25 - 4 Timing of Generation of Internal Reset Signal by Power-on-Reset Circuit and Voltage Detector (3/3)

(3) LVD reset mode (option byte 000C1H: LVIMDS1, LVIMDS0 = 1, 1)



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

Note 2. The high-speed on-chip oscillator clock or a high-speed system clock can be selected as the CPU clock. To use the X1 clock, use the oscillation stabilization time counter status register (OSTC) to confirm the lapse of the oscillation stabilization time.

Note 3. The time until normal operation starts includes the following LVD reset processing time after the LVD detection level (V_{LVD}) is reached as well as the voltage stabilization wait + POR reset processing time after the V_{POR} (1.56 V (TYP.) is reached.
LVD reset processing time: 0 ms to 0.0701 ms (MAX.)

Note 4. When the power supply voltage is below the lower limit for operation and the power supply voltage is then restored after an internal reset is generated only by the voltage detector (LVD), the following LVD reset processing time is required after the LVD detection level (V_{LVD}) is reached.
LVD reset processing time: 0.0511 ms (TYP.), 0.0701 ms (MAX.)

Note 5. The operation guaranteed voltage range is 2.4 V ≤ V_{DD} ≤ 5.5 V. Be sure to start normal operations after the voltage becomes 2.4 V or higher. To set the system to the reset status when the voltage has fallen below 2.7 V while the power is turning off, use the reset function of the voltage detector or input a low level to the $\overline{\text{RESET}}$ pin.

Remark 1. V_{LVDH}, V_{LVDL}: LVD detection voltage
V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Remark 2. When the LVD interrupt mode is selected (option byte 000C1H: LVIMD1 = 0, LVIMD0 = 1), the time until normal operation starts after power is turned on is the same as the time specified in Note 3 of Figure 25 - 4 (3).

CHAPTER 26 VOLTAGE DETECTOR

26.1 Functions of Voltage Detector

The operation mode and detection voltages (V_{LVDH}, V_{LVDL}, V_{LVD}) for the voltage detector is set by using the option byte (000C1H). The voltage detector (LVD) has the following functions.

- The LVD circuit compares the supply voltage (V_{DD}) with the detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}), and generates an internal reset or internal interrupt signal.
- The detection level for the power supply detection voltage (V_{LVDH}, V_{LVDL}) can be selected by using the option byte as one of seven levels (For details, see **CHAPTER 29 OPTION BYTE**).
- Operable in STOP mode.
- After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in **33.4** or **34.4** AC Characteristics. This is done by utilizing the voltage detector or controlling the externally input reset signal. After the power supply is turned off, this LSI should be placed in the STOP mode, or placed in the reset state by utilizing the voltage detector or controlling the externally input reset signal before the voltage falls below the operating range. The range of operating voltage varies with the setting of the user option byte (000C2H or 010C2H).

(a) Interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0)
The two detection voltages (V_{LVDH}, V_{LVDL}) are selected by the option byte 000C1H. The high-voltage detection level (V_{LVDH}) is used for releasing resets and generating interrupts. The low-voltage detection level (V_{LVDL}) is used for generating resets.

(b) Reset mode (option byte LVIMDS1, LVIMDS0 = 1, 1)
The detection voltage (V_{LVD}) selected by the option byte 000C1H is used for triggering and ending resets.

(c) Interrupt mode (option byte LVIMDS1, LVIMDS0 = 0, 1)
The detection voltage (V_{LVD}) selected by the option byte 000C1H is used for generating interrupts/reset release.

The reset and internal interrupt signals are generated in each mode as follows.

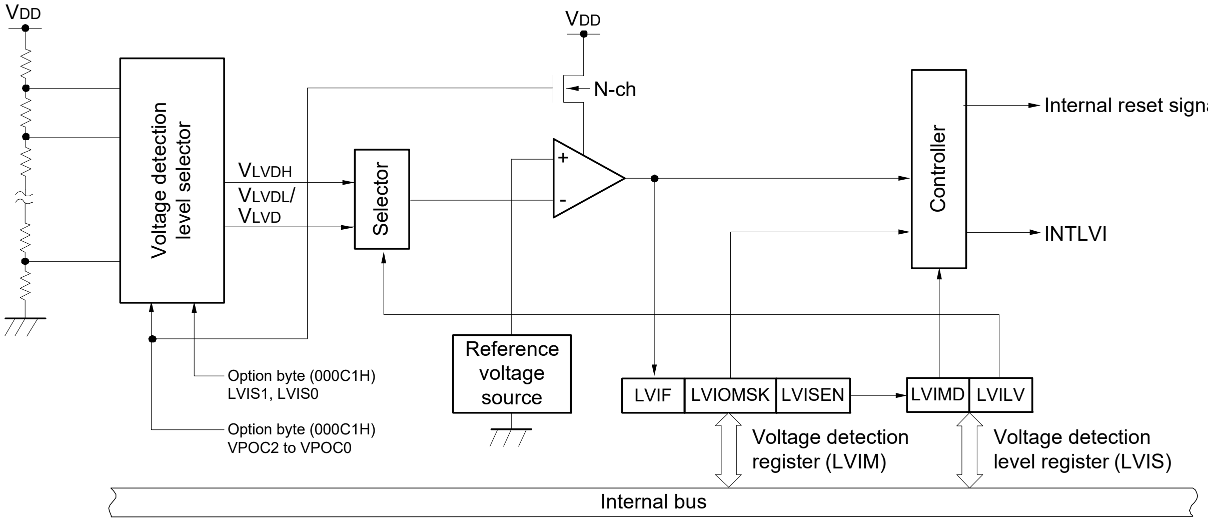
Interrupt & reset mode (LVIMDS1, LVIMDS0 = 1, 0)	Reset mode (LVIMDS1, LVIMDS0 = 1, 1)	Interrupt mode (LVIMDS1, LVIMDS0 = 0, 1)
Generates an interrupt request signal by detecting $V_{DD} < V_{LVDH}$ when the operating voltage falls, and an internal reset by detecting $V_{DD} < V_{LVDL}$. Releases an internal reset by detecting $V_{DD} \geq V_{LVDH}$.	Releases an internal reset by detecting $V_{DD} \geq V_{LVD}$. Generates an internal reset by detecting $V_{DD} < V_{LVD}$.	Retains the state of an internal reset by the LVD immediately after a reset until $V_{DD} \geq V_{LVD}$. Releases the LVD internal reset by detecting $V_{DD} \geq V_{LVD}$. Generates an interrupt request signal (INTLVI) by detecting $V_{DD} < V_{LVD}$ or $V_{DD} \geq V_{LVD}$ after the LVD internal reset is released.

While the voltage detector is operating, whether the supply voltage is more than or less than the detection level can be checked by reading the voltage detection flag (LVIF: bit 0 of the voltage detection register (LVIM)).
Bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of the RESF register, see **CHAPTER 24 RESET FUNCTION**.

26.2 Configuration of Voltage Detector

The block diagram of the voltage detector is shown in Figure 26 - 1.

Figure 26 - 1 Block Diagram of Voltage Detector



26.3 Registers Controlling Voltage Detector

The voltage detector is controlled by using the following registers.

- Voltage detection register (LVIM)
- Voltage detection level register (LVIS)

26.3.1 Voltage detection register (LVIM)

This register is used to specify whether to enable or disable rewriting the voltage detection level register (LVIS), as well as to check the LVD output mask status.

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

Reset signal generation clears this register to 00H.

Figure 26 - 2 Format of Voltage detection register (LVIM)

Address: FFFA9H After reset: 00H Note 1 R/W Note 2

Symbol	<7>	6	5	4	3	2	<1>	<0>
LVIM	LVISEN Note 3	0	0	0	0	0	LVIOMSK	LVIF

LVISEN Note 3	Specification of whether to enable or disable rewriting the voltage detection level register (LVIS)
0	Disabling of rewriting the LVIS register (LVIOMSK = 0 (Mask of LVD output is invalid)
1	Enabling of rewriting the LVIS register Note 3 (LVIOMSK = 1 (Mask of LVD output is valid)

LVIOMSK	Mask status flag of LVD output
0	Mask of LVD output is invalid
1	Mask of LVD output is valid Note 4

LVIF	Voltage detection flag
0	Supply voltage (V_{DD}) ≥ detection voltage (V_{LVD}), or when LVD is off
1	Supply voltage (V_{DD}) < detection voltage (V_{LVD})

Note 1. The reset value changes depending on the reset source.
If the LVIM register is reset by LVD, it is not reset but holds the current value. In other reset, LVISEN is cleared to 0.

Note 2. Bits 0 and 1 are read-only.

Note 3. LVISEN and LVIOMSK can only be set in the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0).
Do not change the initial value in other modes.

Note 4. LVIOMSK bit is only automatically set to “1” when the interrupt & reset mode is selected (option byte LVIMDS1, LVIMDS0 = 1, 0) and reset or interrupt by LVD is masked.
- Period during LVISEN = 1
- Waiting period from the time when LVD interrupt is generated until LVD detection voltage becomes stable
- Waiting period from the time when the value of LVILV bit changes until LVD detection voltage becomes stable

26.3.2 Voltage detection level register (LVIS)

This register selects the voltage detection level.

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

Reset signal generation input sets this register to 00H/01H/81H Note 1.

Figure 26 - 3 Format of Voltage detection level register (LVIS)

Address: FFFAAH After reset:00H/01H/81H Note 1 R/W

Symbol	<7>	6	5	4	3	2	1	<0>
LVIS	LVIMD Note 2	0	0	0	0	0	0	LVILV Note 2

LVIMD Note 2	Operation mode of voltage detection
0	Interrupt mode
1	Reset mode

LVILV Note 2	LVD detection level
0	High-voltage detection level (V_{LVDH})
1	Low-voltage detection level (V_{LVDL} or V_{LVD})

Note 1. The reset value changes depending on the reset source and the setting of the option byte.
This register is not cleared (00H) by LVD reset.
The generation of reset signal other than an LVD reset sets as follows.
- When option byte LVIMDS1, LVIMDS0 = 1, 0: 00H
- When option byte LVIMDS1, LVIMDS0 = 1, 1: 81H
- When option byte LVIMDS1, LVIMDS0 = 0, 1: 01H

Note 2. Writing “0” can only be allowed in the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0). Do not set LVIMD and LVILV in other cases. The value is switched automatically when reset or interrupt is generated in the interrupt & reset mode.

Caution 1. Rewrite the value of the LVIS register according to Figures 26 - 10 and 26 - 11.

Caution 2. Specify the LVD operation mode and detection voltage (V_{LVDH}, V_{LVDL}, V_{LVD}) of each mode by using the option byte 000C1H. Figure 26 - 4 shows the format of the user option byte (000C1H/010C1H). For details about the option byte, see CHAPTER 29 OPTION BYTE.

Figure 26 - 4 Format of User Option Byte (000C1H/010C1H) (1/2)

Address: 000C1H/010C1H[Note]

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD setting (interrupt & reset mode)

Detection voltage			Option byte setting value						
V_{LVDH}		V_{LVDL}	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge	Falling edge						LVIMDS1	LVIMDS0
3.02 V	2.96 V	2.55 V	0	0	0	0	1	1	0
3.22 V	3.15 V					0	0		
4.42 V	5.32 V	2.75 V		0	1	0	0		
4.62 V	4.52 V	2.75 V		1	0	0	0		
3.32 V	3.15 V	2.75 V		1	1	0	1		
4.74 V	4.64 V					0	0		
—			Settings other than the above are prohibited						

• LVD setting (reset mode)

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
2.61 V	2.55 V	0	0	0	1	1	1	1
2.81 V	2.75 V		1	1	1	1		
3.02 V	2.96 V		0	0	0	1		
3.22 V	3.15 V		1	1	0	1		
4.42 V	4.32 V		0	1	0	0		
4.62 V	4.52 V		1	0	0	0		
4.74 V	4.64 V		1	1	0	0		
—		Settings other than the above are prohibited						

• LVD setting (interrupt mode)

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
2.61 V	2.55 V	0	0	0	1	1	0	1
2.81 V	2.75 V		1	1	1	1		
3.02 V	2.96 V		0	0	0	1		
3.22 V	3.15 V		1	1	0	1		
4.42 V	4.32 V		0	1	0	0		
4.62 V	4.52 V		1	0	0	0		
4.74 V	4.64 V		1	1	0	0		
—		Settings other than the above are prohibited						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Remark The detection voltage is a TYP. value. For details, see **33.6.8** or **34.6.8** LVD characteristics.

(Cautions are listed on the next page.)

Figure 26 - 5 Format of User Option Byte (000C1H/010C1H) (2/2)

Address: 000C1H/010C1H[Note]

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD off (external reset input via RESET pin is used)

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
—	—	1	×	×	×	×	×	1
—		Settings other than the above are prohibited						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Caution 1. Set bit 4 to 1.

Caution 2. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 33.4 or 34.4 AC Characteristics. This is done by utilizing the voltage detector or controlling the externally input reset signal. After the power supply is turned off, this LSI should be placed in the STOP mode, or placed in the reset state by utilizing the voltage detector or controlling the externally input reset signal, before the voltage falls below the operating range. The range of operating voltage varies with the setting of the user option byte (000C2H or 010C2H).

Remark 1. ×: Don't care

Remark 2. For details on the LVD circuit, see **CHAPTER 26 VOLTAGE DETECTOR**.

Remark 3. The detection voltage is a TYP. value. For details, see **33.6.8** or **34.6.8** LVD characteristics.

26.4 Operation of Voltage Detector

26.4.1 When used as reset mode

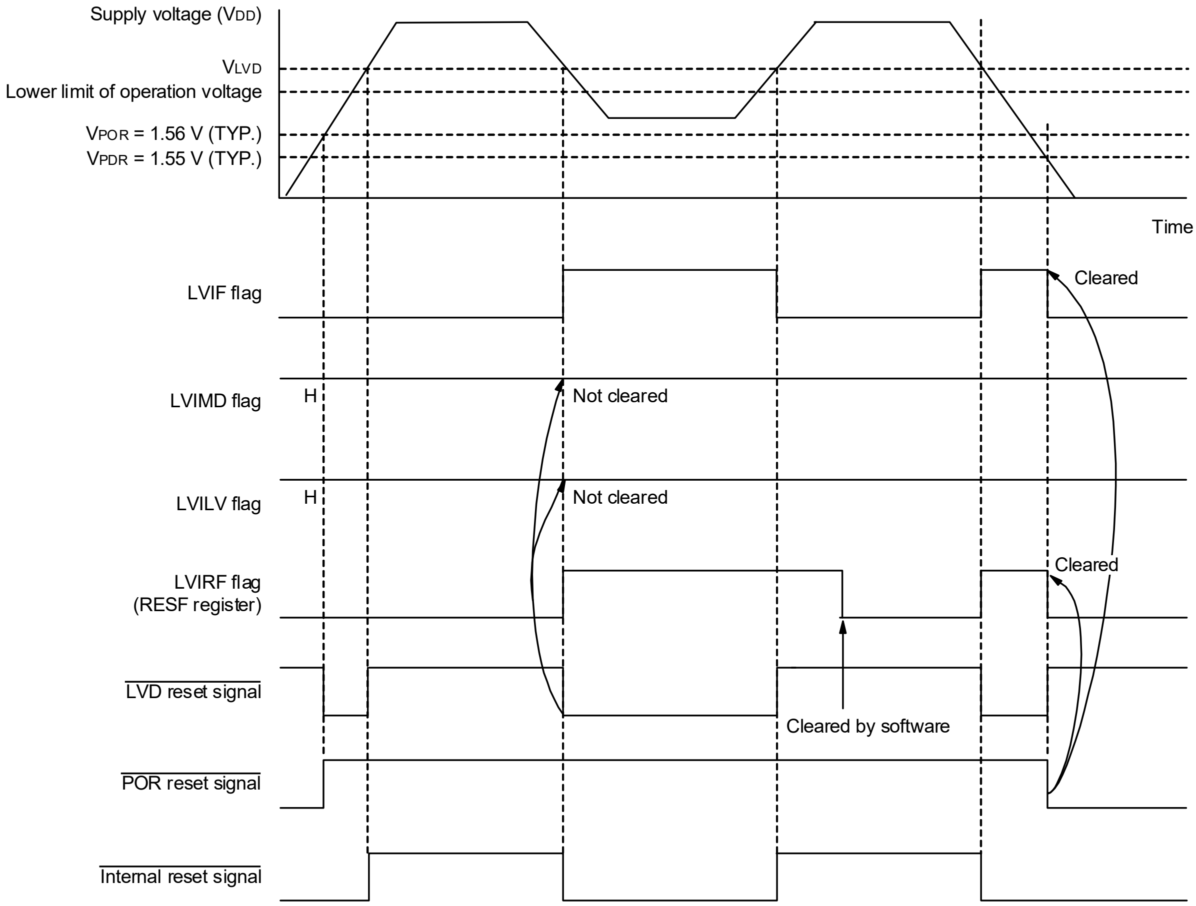
Specify the operation mode (the reset mode (LVIMDS1, LVIMDS0 = 1, 1)) and the detection voltage (V_{LVD}) by using the option byte 000C1H.

The operation is started in the following initial setting state when the reset mode is set.
- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- The initial value of the voltage detection level select register (LVIS) is set to 81H.
  Bit 7 (LVIMD) is 1 (reset mode).
  Bit 0 (LVILV) is 1 (low-voltage detection level: V_{LVD}).

• Operation in LVD reset mode
  In the reset mode (option byte LVIMDS1, LVIMDS0 = 1, 1), the state of an internal reset by LVD is retained until the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}) after power is supplied. The internal reset is released when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}).
  At the fall of the operating voltage, an internal reset by LVD is generated when the supply voltage (V_{DD}) falls below the voltage detection level (V_{LVD}).

Figure 26 - 6 shows the timing of the internal reset signal generated in the LVD reset mode.

Figure 26 - 6 Timing of Voltage Detector Internal Reset Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 1)



Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

26.4.2 When used as interrupt mode

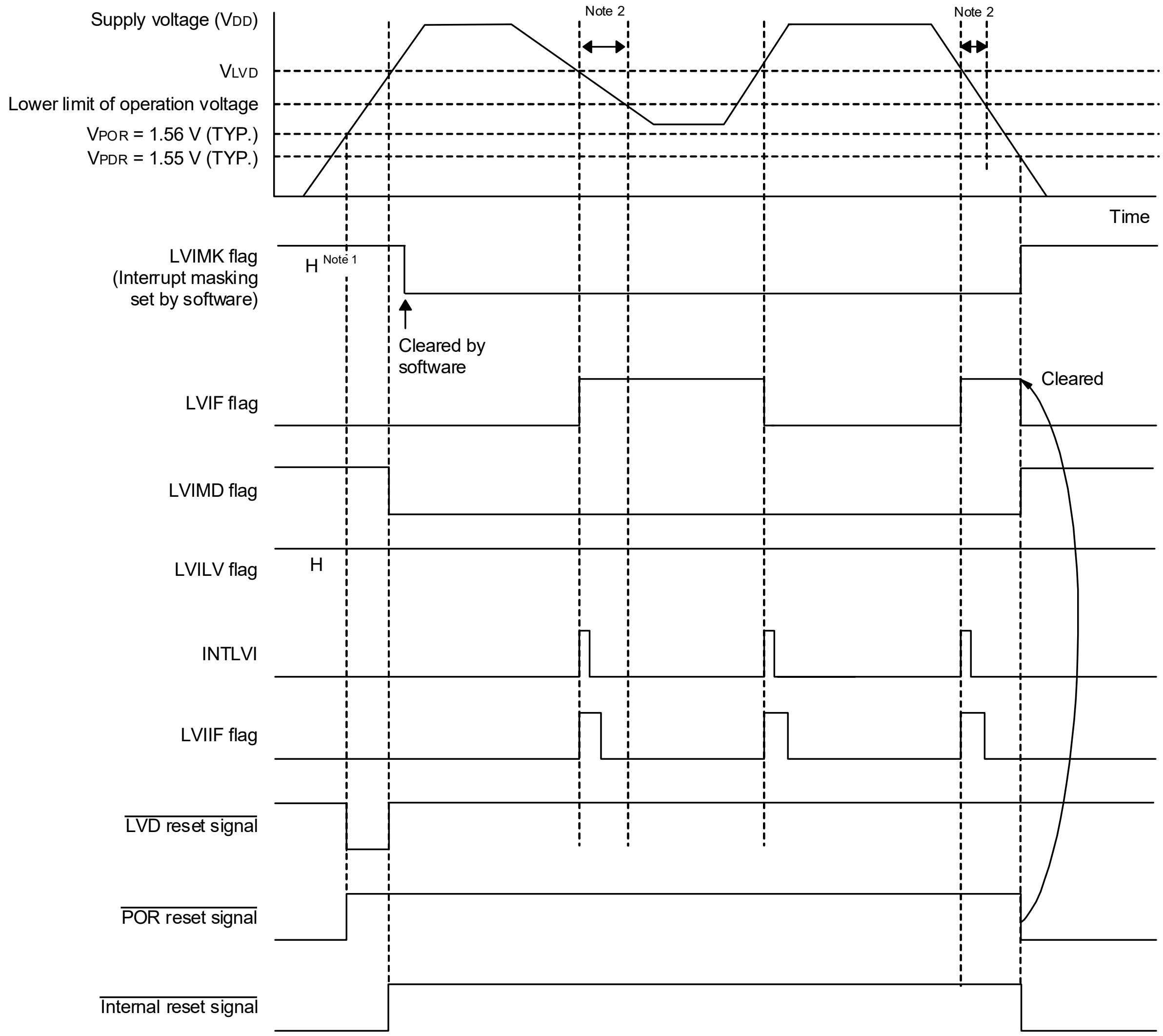
Specify the operation mode (the interrupt mode (LVIMDS1, LVIMDS0 = 0, 1)) and the detection voltage (V_{LVD}) by using the option byte 000C1H.

The operation is started in the following initial setting state when the interrupt mode is set.
- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- The initial value of the voltage detection level select register (LVIS) is set to 01H.
  Bit 7 (LVIMD) is 0 (interrupt mode).
  Bit 0 (LVILV) is 1 (low-voltage detection level: V_{LVD}).

• Operation in LVD interrupt mode
  In interrupt mode (LVIMDS1 and LVIMDS0 = 0 and 1 in the option byte), the state of an internal reset by the LVD is retained immediately after a reset until the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}). The LVD internal reset is released when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}).
  After the LVD internal reset is released, an interrupt request signal (INTLVI) by the LVD is generated when the supply voltage (V_{DD}) exceeds the voltage detection level (V_{LVD}). When the voltage falls, this LSI should be placed in the STOP mode, or placed in the reset state by controlling the externally input reset signal, before the voltage falls below the operating voltage range defined in **33.4** or **34.4** AC Characteristics. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Figure 26 - 7 shows the timing of the interrupt request signal generated in the LVD interrupt mode.

Figure 26 - 7 Timing of Voltage Detector Internal Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 0, 1)



Note 1. The LVIMK flag is set to “1” by reset signal generation.

Note 2. When the voltage falls, this LSI should be placed in the STOP mode, or placed in the reset state by controlling the externally input reset signal, before the voltage falls below the operating voltage range defined in **33.4** or **34.4 AC characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

26.4.3 When used as interrupt and reset mode

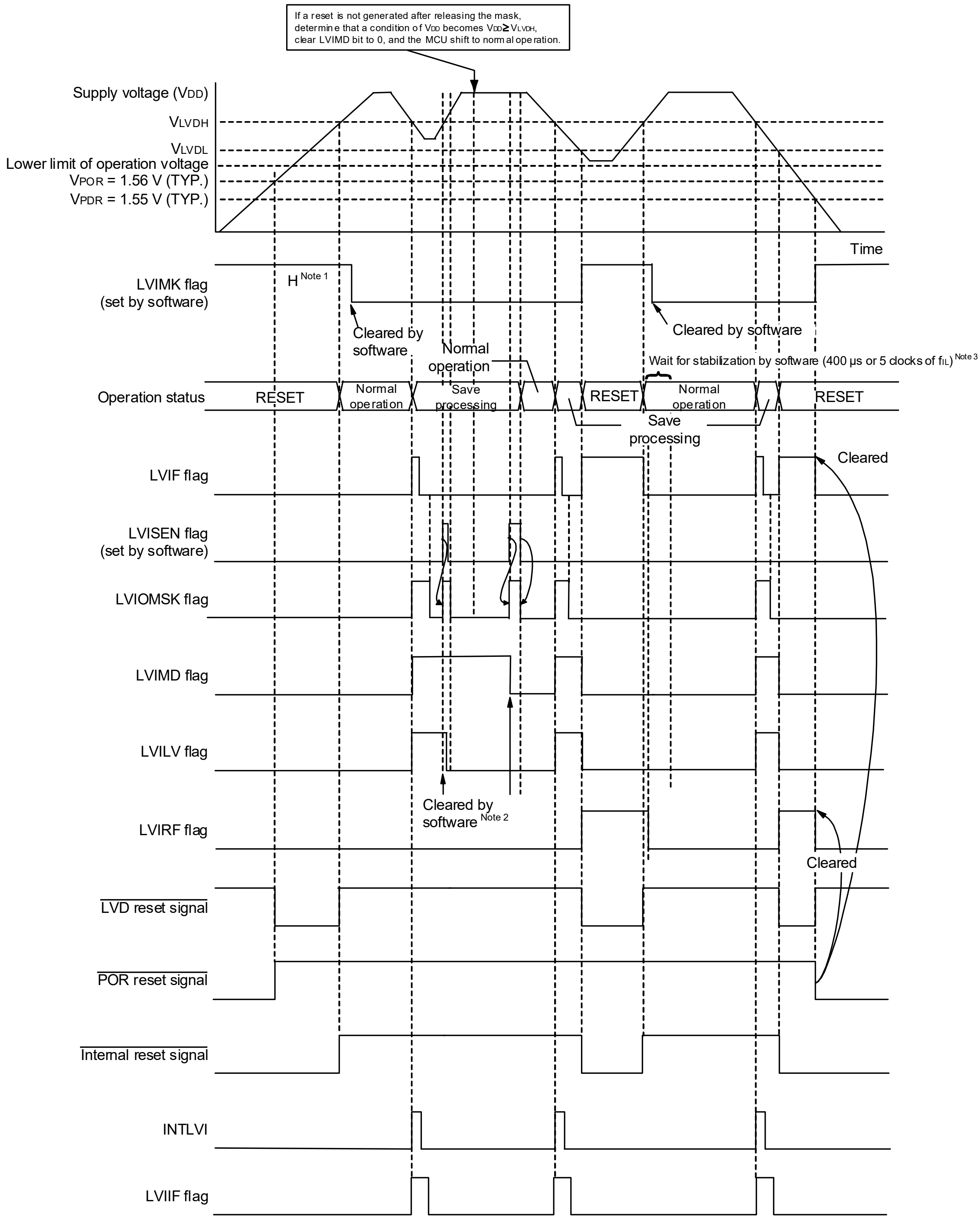
Specify the operation mode (the interrupt & reset (LVIMDS1, LVIMDS0 = 1, 0)) and the detection voltage (V_{LVDH}, V_{LVDL}) by using the option byte 000C1H.

The operation is started in the following initial setting state when the interrupt & reset mode is set.
- Bit 7 (LVISEN) of the voltage detection register (LVIM) is set to 0 (disable rewriting of voltage detection level register (LVIS))
- The initial value of the voltage detection level select register (LVIS) is set to 00H.
  Bit 7 (LVIMD) is 0 (interrupt mode).
  Bit 0 (LVILV) is 0 (high-voltage detection level: V_{LVDH}).

• Operation in LVD interrupt & reset mode
In the interrupt & reset mode (option byte LVIMDS1, LVIMDS0 = 1, 0), the state of an internal reset by LVD is retained until the supply voltage (V_{DD}) exceeds the high-voltage detection level (V_{LVDH}) after power is supplied. The internal reset is released when the supply voltage (V_{DD}) exceeds the high-voltage detection level (V_{LVDH}). An interrupt request signal by LVD (INTLVI) is generated and arbitrary save processing is performed when the supply voltage (V_{DD}) falls below the high-voltage detection level (V_{LVDH}). After that, an internal reset by LVD is generated when the supply voltage (V_{DD}) falls below the low-voltage detection level (V_{LVDL}). After INTLVI is generated, an interrupt request signal is not generated even if the supply voltage becomes equal to or higher than the high-voltage detection voltage (V_{LVDH}) without falling below the low-voltage detection voltage (V_{LVDL}).
To use the LVD reset & interrupt mode, perform the processing according to **Figure 26 - 10 Setting Procedure for Operating Voltage Check and Reset** and **Figure 26 - 11 Setting Procedure for Initial Setting of Interrupt and Reset Mode**.

Figures 26 - 8 and 26 - 9 show the timing of the internal reset signal and interrupt signal generated in the LVD interrupt & reset mode.

Figure 26 - 8 Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (1/2)



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

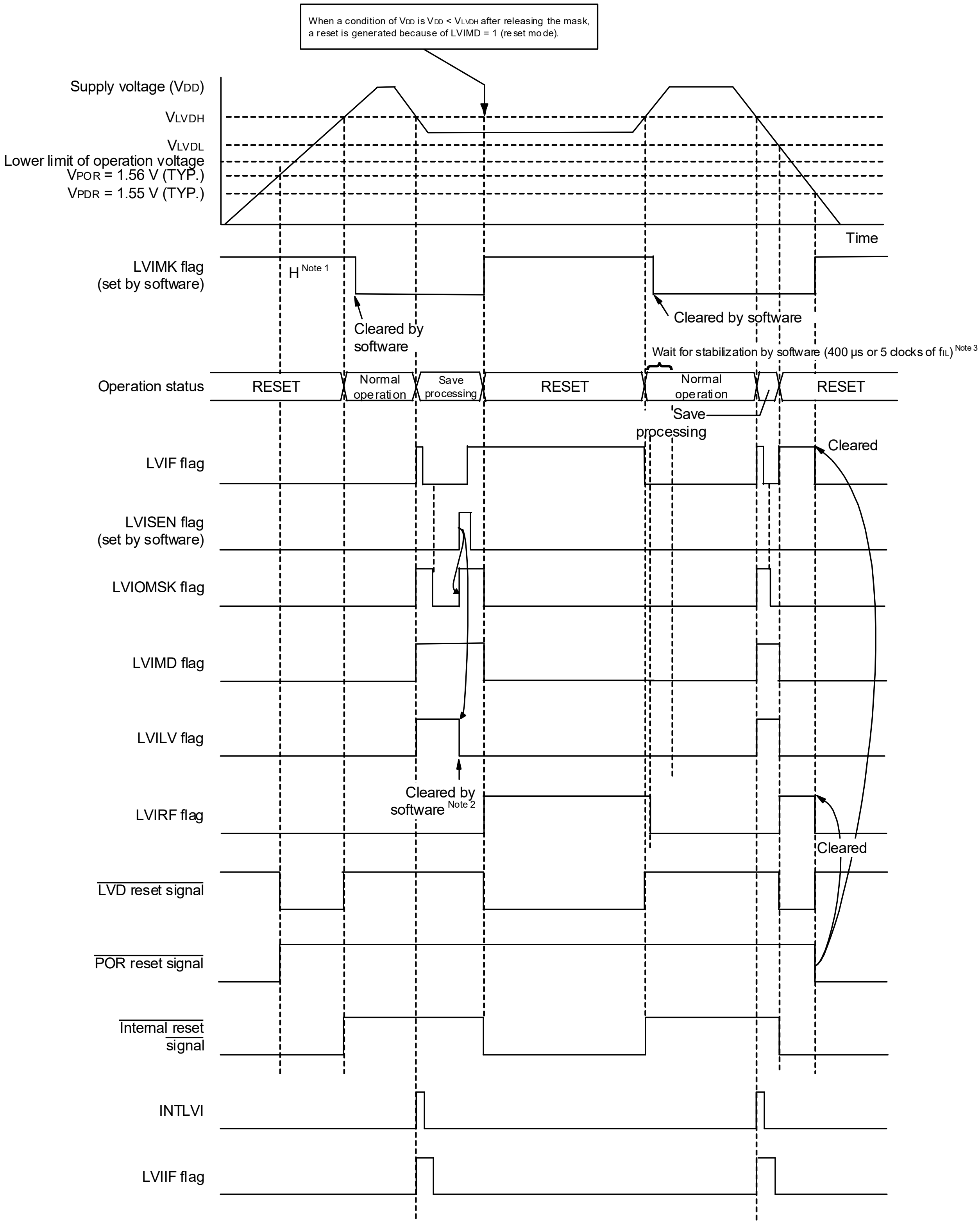
Note 1. The LVIMK flag is set to 1 by reset signal generation.

Note 2. After an interrupt is generated, perform the processing according to Figure 26 - 10 Setting Procedure for Operating Voltage Check and Reset in interrupt and reset mode.

Note 3. After a reset is released, perform the processing according to Figure 26 - 11 Setting Procedure for Initial Setting of Interrupt and Reset Mode in interrupt and reset mode.

Remark V_{POR}: POR power supply rise detection voltage
V_{POR}: POR power supply fall detection voltage

Figure 26 - 9 Timing of Voltage Detector Reset Signal and Interrupt Signal Generation (Option Byte LVIMDS1, LVIMDS0 = 1, 0) (2/2)



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

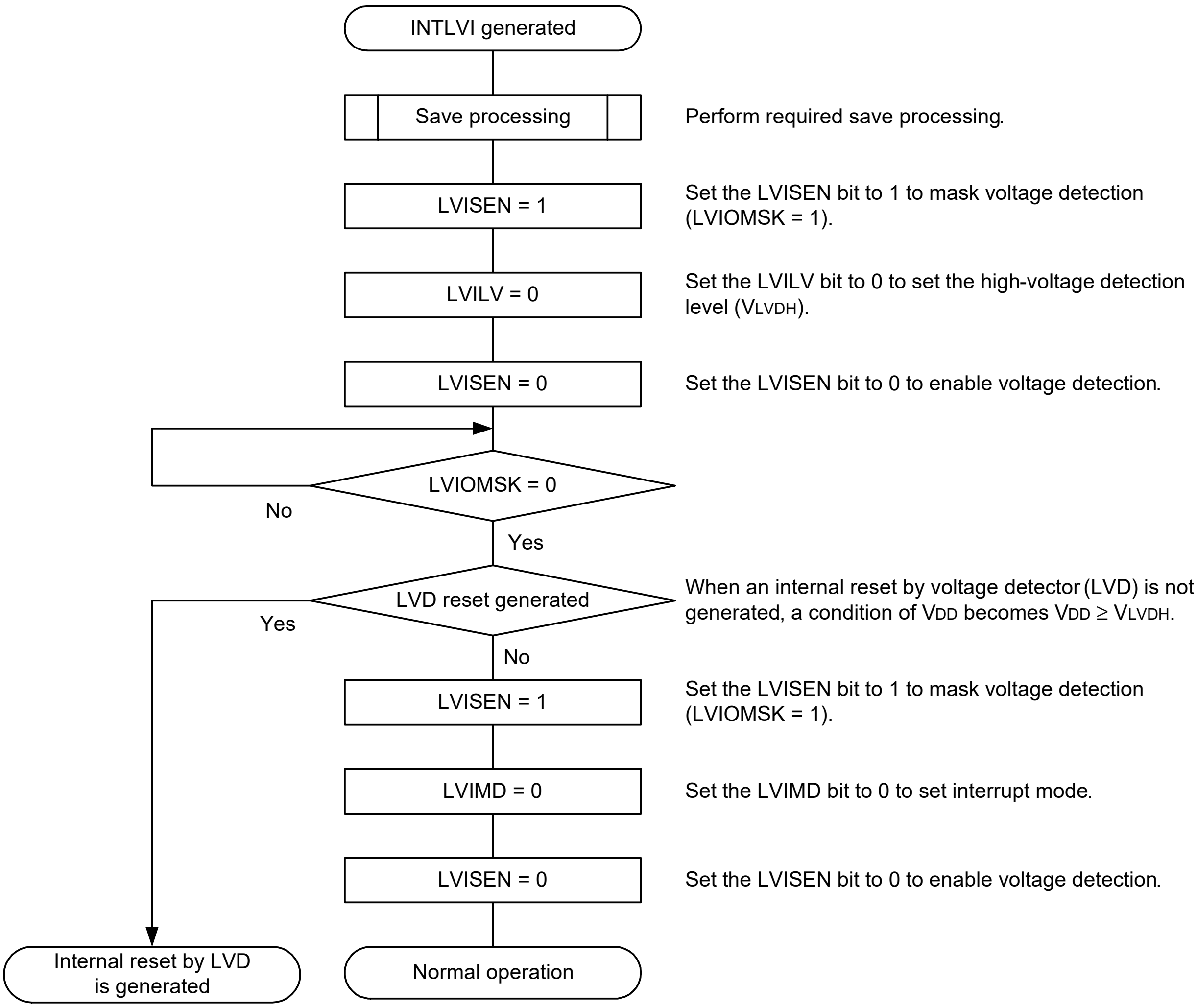
Note 1. The LVIMK flag is set to “1” by reset signal generation.

Note 2. After an interrupt is generated, perform the processing according to Figure 26 - 10 Setting Procedure for Operating Voltage Check and Reset in interrupt and reset mode.

Note 3. After a reset is released, perform the processing according to Figure 26 - 11 Setting Procedure for Initial Setting of Interrupt and Reset Mode in interrupt and reset mode.

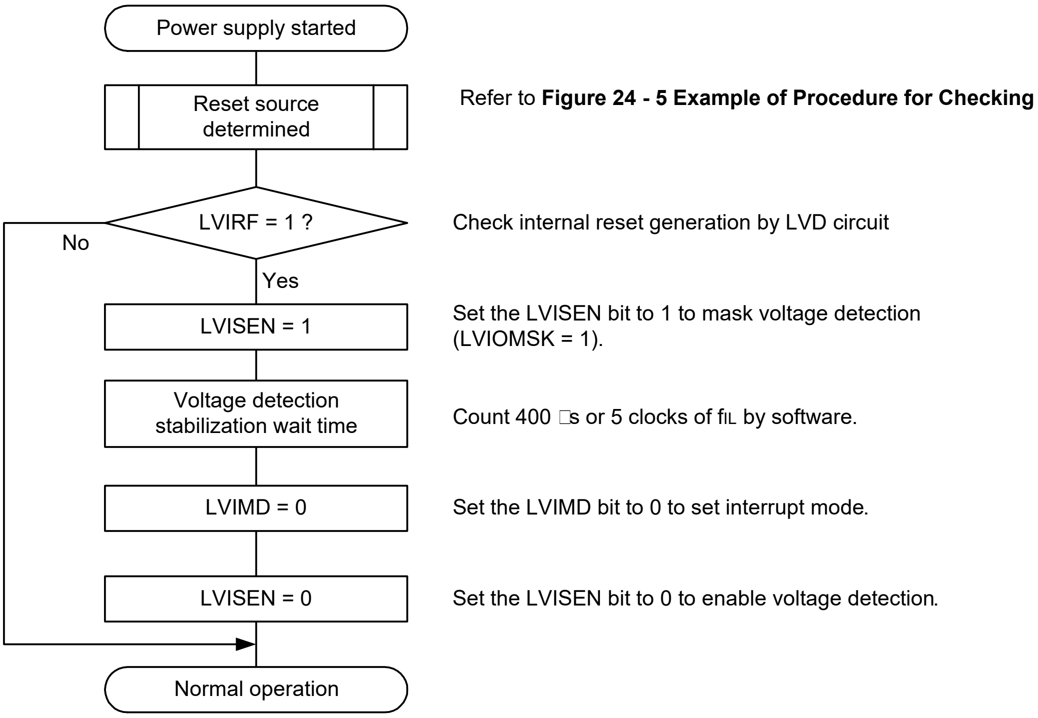
Remark V_{POR}: POR power supply rise detection voltage
V_{PDR}: POR power supply fall detection voltage

Figure 26 - 10 Setting Procedure for Operating Voltage Check and Reset



When setting an interrupt and reset mode (LVIMDS1, LVIMDS0 = 1, 0), voltage detection stabilization wait time for 400 μs or 5 clocks of fIL is necessary after LVD reset is released (LVIRF = 1). After waiting until voltage detection stabilizes, (0) clear the LVIMD bit for initialization. While voltage detection stabilization wait time is being counted and when the LVIMD bit is rewritten, set LVISEN to 1 to mask a reset or interrupt generation by LVD.
Figure 26 - 11 shows the procedure for Setting Procedure for Initial Setting of Interrupt and Reset Mode.

Figure 26 - 11 Setting Procedure for Initial Setting of Interrupt and Reset Mode



Remark fIL: Low-speed on-chip oscillator clock frequency

26.5 Cautions for Voltage Detector

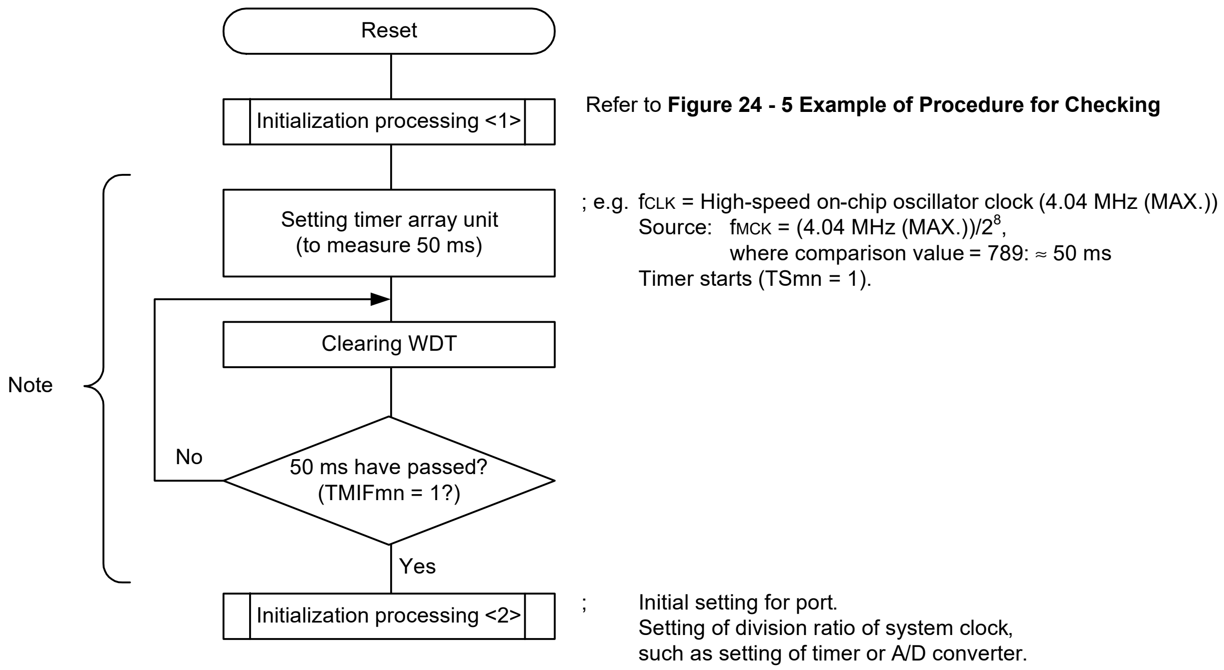
(1) Voltage fluctuation when power is supplied

In a system where the supply voltage (VDD) fluctuates for a certain period in the vicinity of the LVD detection voltage, the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller can be arbitrarily set by taking the following action.

<Action>

After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports.

Figure 26 - 12 Example of Software Processing If Supply Voltage Fluctuation is 50 ms or Less in Vicinity of LVD Detection Voltage



Note If reset is generated again during this period, initialization processing <2> is not started.

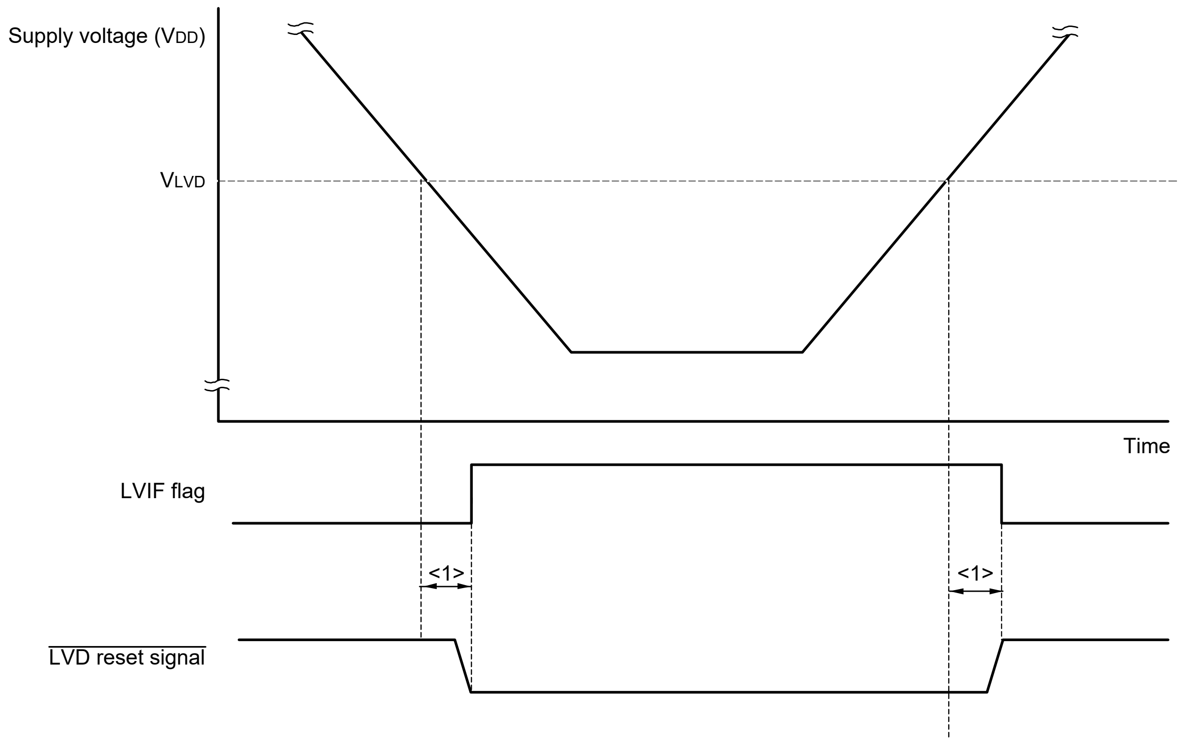
Remark m = 0, 1
n = 0 to 3

(2) Delay from the time LVD reset source is generated until the time LVD reset has been generated or released

There is some delay from the time supply voltage (VDD) < LVD detection voltage (VLVD) until the time LVD reset has been generated.

In the same way, there is also some delay from the time LVD detection voltage (VLVD) ≤ supply voltage (VDD) until the time LVD reset has been released (see **Figure 26 - 13**).

Figure 26 - 13 Delay from the time LVD reset source is generated until the time LVD reset has been generated or released



<1>: Detection delay (300 μs (MAX.))

(3) Power on when LVD is off

Use the external rest input via the RESET pin when the LVD is off.

For an external reset, input a low level for 10 μs or more to the RESET pin. To perform an external reset upon power application, input a low level to the RESET pin, turn power on, continue to input a low level to the pin for 10 μs or more within the operating voltage range shown in **33.4** or **34.4 AC Characteristics**, and then input a high level to the pin.

(4) Operating voltage fall when LVD is off or LVD interrupt mode is selected

When the operating voltage falls with the LVD is off or with the LVD interrupt mode is selected, this LSI should be placed in the STOP mode, or placed in the reset state by controlling the externally input reset signal, before the voltage falls below the operating voltage range defined in **33.4** or **34.4 AC Characteristics**. When restarting the operation, make sure that the operation voltage has returned within the range of operation.

CHAPTER 27 SAFETY FUNCTIONS

27.1 Overview of Safety Functions

<R> The following safety functions are provided in the RL78/I1E to comply with the IEC60730 safety standards. These functions enable the microcontroller to self-diagnose abnormalities and stop operating if an abnormality is detected.

(1) Flash memory CRC operation function (high-speed CRC, general-purpose CRC)
This detects data errors in the flash memory by performing CRC operations.
Two CRC functions are provided in the RL78/I1E that can be used according to the application or purpose of use.

- High-speed CRC: The CPU can be stopped and a high-speed check executed on its entire code flash memory area during the initialization routine.
- General CRC: This can be used for checking various data in addition to the code flash memory area while the CPU is running.

(2) RAM parity error detection function
This detects parity errors when the RAM is read as data.

(3) RAM guard function
This prevents RAM data from being rewritten when the CPU freezes.

(4) SFR guard function
This prevents SFRs from being rewritten when the CPU freezes.

(5) Invalid memory access detection function
This detects illegal accesses to invalid memory areas (such as areas where no memory is allocated and areas to which access is restricted).

(6) Frequency detection function
This uses the timer array unit to perform a self-check of the CPU/peripheral hardware clock frequency.

(7) A/D test function
This is used to perform a self-check of A/D converter by performing A/D conversion on the positive internal reference voltage, negative reference voltage, analog input channel (ANI), and internal reference voltage output.

(8) Digital output signal level detection function for I/O pins
When the I/O pins are output mode, the output level of the pin can be read.

Remark Refer to the IEC60730/60335 self-test library application notes (R01AN1062, R01AN1296) for the RL78 MCU Series, for more information on usage examples of the safety functions required to comply with the
<R> IEC60730 safety standards.

27.2 Registers Used by Safety Functions

The safety functions use the following registers:

Register	Each Function of Safety Function
• Flash memory CRC control register (CRC0CTL) • Flash memory CRC operation result register (PGCRCL)	Flash memory CRC operation function (high-speed CRC)
• CRC input register (CRCIN) • CRC data register (CRCD)	CRC operation function (general-purpose CRC)
• RAM parity error control register (RPECTL)	RAM parity error detection function
• Invalid memory access detection control register (IAWCTL)	RAM guard function
	SFR guard function
	Invalid memory access detection function
• Timer input select register 0 (TIS0)	Frequency detection function
• A/D test register (ADTES)	A/D test function
• Port mode select register (PMS)	Digital output signal level detection function for I/O pins

The content of each register is described in 27.3 Operation of Safety Functions.

27.3 Operation of Safety Functions

27.3.1 Flash memory CRC operation function (high-speed CRC)

The IEC60730 standard mandates the checking of data in the flash memory, and recommends using CRC to do it. The high-speed CRC provided in the RL78/I1E can be used to check the entire code flash memory area during the initialization routine. The high-speed CRC can be executed only when the program is allocated on the RAM and in the HALT mode of the main system clock.

The high-speed CRC performs an operation by reading 32-bit data per clock from the flash memory while stopping the CPU. This function therefore can finish a check in a shorter time (for example, 256 μs @ 32 MHz with 32 KB flash memory).

The CRC generator polynomial used complies with “$X^{16} + X^{12} + X^5 + 1$” of CRC-16-CCITT.

The high-speed CRC operates in MSB first order from bit 31 to bit 0.

Caution The CRC operation result might differ during on-chip debugging because the monitor program is allocated.

Remark The operation result is different between the high-speed CRC and the general CRC, because the general CRC operates in LSB first order.

27.3.1.1 Flash memory CRC control register (CRC0CTL)

This register is used to control the operation of the high-speed CRC ALU, as well as to specify the operation range.
The CRC0CTL register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 27 - 1 Format of Flash memory CRC control register (CRC0CTL)

Address: F02F0H After reset:00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CRC0CTL	CRC0EN	0	0	0	0	0	0	FEA0

CRC0EN	Control of high-speed CRC ALU operation
0	Stop the operation.
1	Start the operation according to HALT instruction execution.

FAE0	Range of high-speed CRC operation
0	0000H to 3FFBH (16 KB − 4 bytes)
1	0000H to 7FFBH (32 KB − 4 bytes)

Caution Be sure to clear bits 1 to 6 to “0”.

Remark Input the expected CRC operation result value to be used for comparison in the lowest 4 bytes of the flash memory. Note that the operation range will thereby be reduced by 4 bytes.

27.3.1.2 Flash memory CRC operation result register (PGCRCL)

This register is used to store the high-speed CRC operation results.
The PGCRCL register can be set by a 16-bit memory manipulation instruction.
Reset signal generation clears this register to 0000H.

Figure 27 - 2 Format of Flash memory CRC operation result register (PGCRCL)

Address: F02F2H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8
PGCRCL	PGCRC15	PGCRC14	PGCRC13	PGCRC12	PGCRC11	PGCRC10	PGCRC9	PGCRC8

	7	6	5	4	3	2	1	0
	PGCRC7	PGCRC6	PGCRC5	PGCRC4	PGCRC3	PGCRC2	PGCRC1	PGCRC0

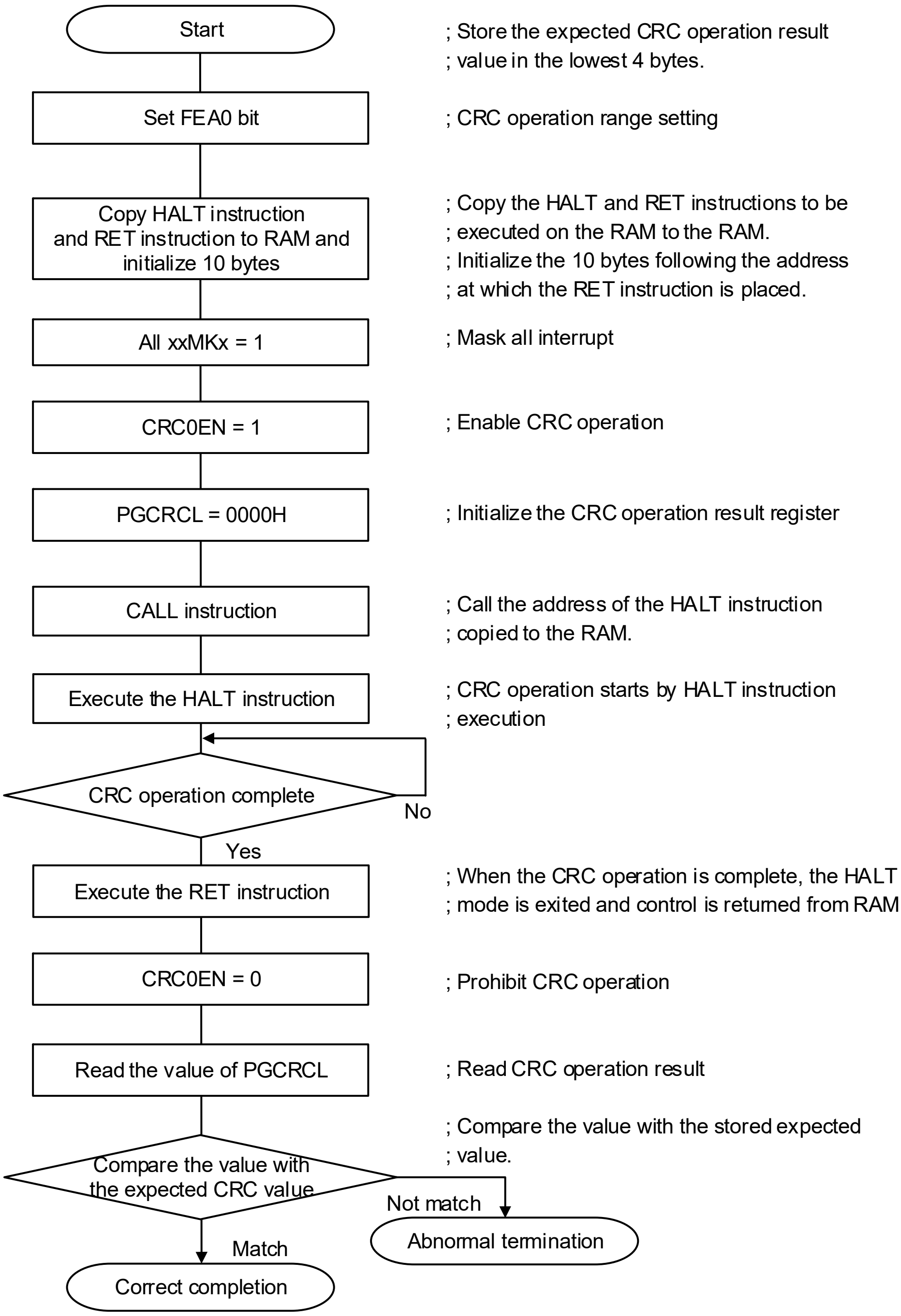
PGCRC15 to 0	High-speed CRC operation results
0000H to FFFFH	Store the high-speed CRC operation results.

Caution The PGCRCL register can only be written if CRC0EN (bit 7 of the CRC0CTL register) = 1.

Figure 27 - 3 shows the Flowchart of Flash Memory CRC Operation Function (High-speed CRC).

<Operation flow>

Figure 27 - 3 Flowchart of Flash Memory CRC Operation Function (High-speed CRC)



Caution 1. The CRC operation is executed only on the code flash.

Caution 2. Store the expected CRC operation value in the area below the operation range in the code flash.

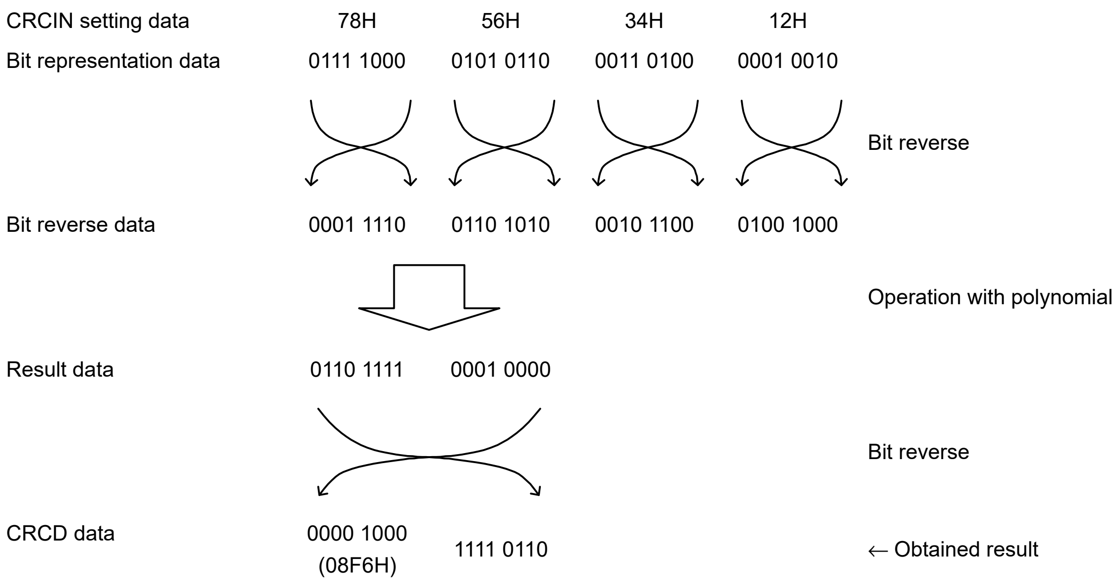
Caution 3. The CRC operation is enabled by executing the HALT instruction in the RAM area.
Be sure to execute the HALT instruction in RAM area.

The expected CRC operation value can be calculated by using the integrated development environment CubeSuite+ development environment. Refer to the CubeSuite+ integrated development environment user's manual for details.

27.3.2 CRC operation function (general-purpose CRC)

<R> In the RL78/I1E, a general CRC operation can be executed as a peripheral function while the CPU is operating. The general CRC can be used for checking various data in addition to the code flash memory area. The data to be checked can be specified by using software (a user-created program). In HALT mode, the CRC operation function can be used only during DTC transfer.

The 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.

27.3.2.1 CRC input register (CRCIN)

CRCIN register is an 8-bit register that is used to set the CRC operation data of general-purpose CRC.
The possible setting range is 00H to FFH.
The CRCIN register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 27 - 4 Format of CRC input register (CRCIN)

Address:FFFACH After reset:00H R/W

Symbol	7	6	5	4	3	2	1	0
CRCIN								

Bits 7 to 0	Function
00H to FFH	Data input

27.3.2.2 CRC data register (CRCD)

This register is used to store the general-purpose CRC operation result.

The possible setting range is 0000H to FFFFH.

After 1 clock of CPU/peripheral hardware clock (f_{CLK}) has elapsed from the time CRCIN register is written, the CRC operation result is stored to the CRCD register.

The CRCD register can be set by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 27 - 5 Format of CRC data register (CRCD)

Address: F02FAH After reset: 0000H R/W

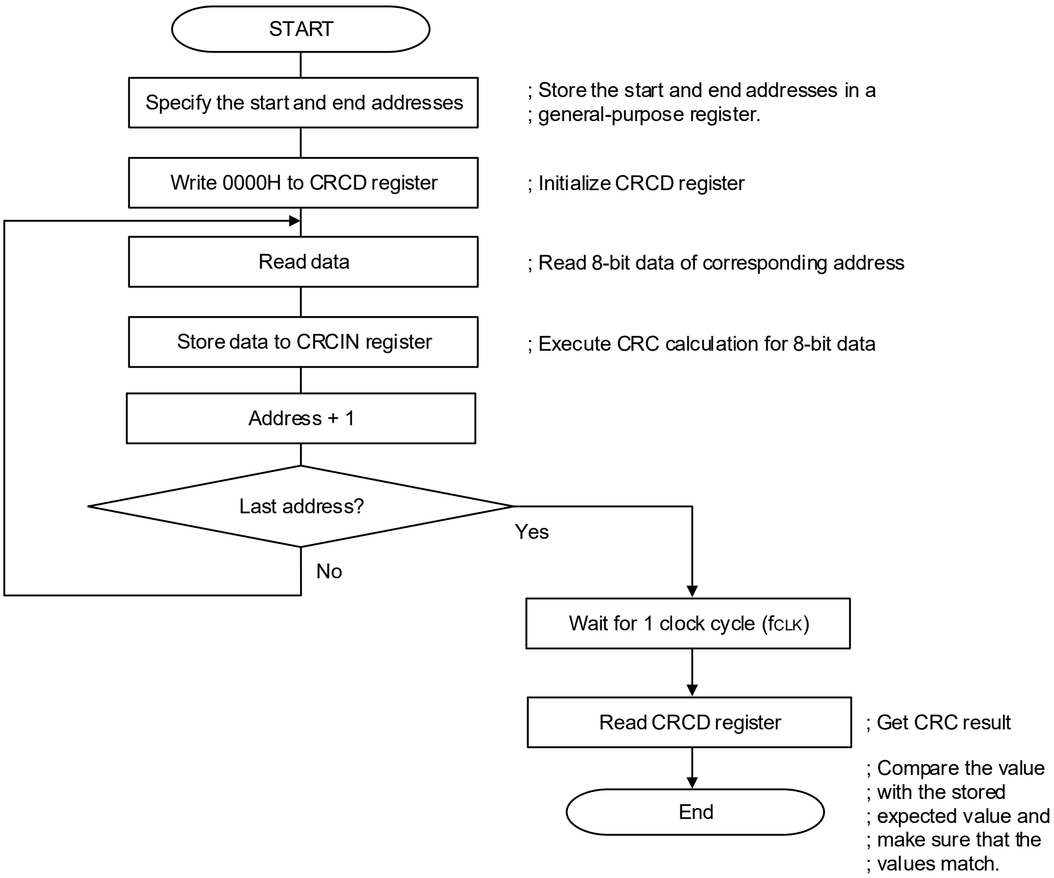
Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCD																

Caution 1. Read the value written to the CRCD register before writing to the CRCIN register.

Caution 2. If writing and storing the operation result to the CRCD register conflict, the writing is ignored.

<Operation flow>

Figure 27 - 6 CRC Operation Function (General-Purpose CRC)



27.3.3 RAM parity error detection function

The IEC60730 standard mandates the checking of RAM data. A single-bit parity bit is therefore added to all 8-bit data in the RL78/I1E's RAM. By using this RAM parity error detection function, the parity bit is appended when data is written, and the parity is checked when the data is read. This function can also be used to trigger a reset when a parity error occurs.

27.3.3.1 RAM parity error control register (RPECTL)

This register is used to control parity error generation check bit and reset generation due to parity errors.
The RPECTL register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 27 - 7 Format of RAM parity error control register (RPECTL)

Address: F00F5H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
RPECTL	RPERDIS	0	0	0	0	0	0	RPEF

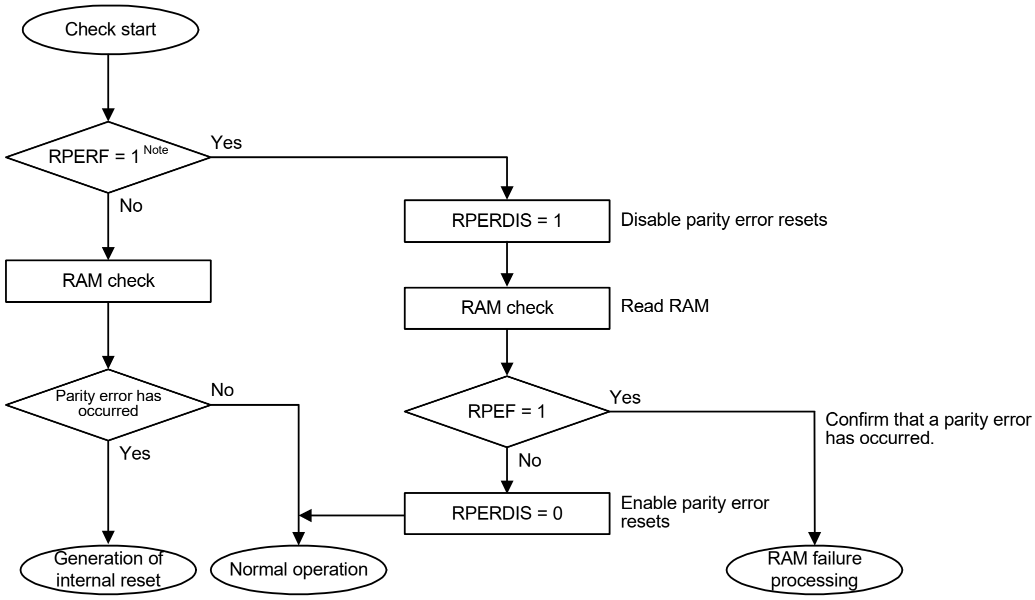
RPERDIS	Parity error reset mask flag
0	Enable parity error resets.
1	Disable parity error resets.

RPEF	Parity error status flag
0	No parity error has occurred.
1	A parity error has occurred.

Caution The parity bit is appended when data is written, and the parity is checked when the data is read. Therefore, while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize RAM areas where data access is to proceed before reading data.
The RL78's CPU executes look-ahead due to the pipeline operation, the CPU might read an uninitialized RAM area that is allocated beyond the RAM used, which causes a RAM parity error. Therefore, while RAM parity error resets are enabled (RPERDIS = 0), be sure to initialize the RAM area + 10 bytes when instructions are fetched from RAM areas.

Remark 1. The parity error reset is enabled by default (RPERDIS = 0).
Remark 2. Even if the parity error reset is disabled (RPERDIS = 1), the RPEF flag will be set (1) if a parity error occurs. If the parity error reset is enabled (RPERDIS = 0) while RPEF = 1, a parity error reset occurs when RPERDIS is cleared (0).
Remark 3. The RPECTL flag in the RESF register is set (1) by RAM parity errors and cleared (0) by writing 0 to it or by any reset source. When RPEF = 1, the value is retained even if RAM for which no parity error has occurred is read.
Remark 4. General-purpose registers are not included in the range of RAM parity error detection.

Figure 27 - 8 RAM Parity Error Check Flow



Note See **CHAPTER 24 RESET FUNCTION** for details on how to confirm internal resets due to RAM parity errors.

27.3.4 RAM guard function

<R> This RAM guard function is used to protect data in the specified memory space.

If the RAM guard function is specified, writing to the specified RAM space is disabled, but reading from the space can be carried out as usual.

27.3.4.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.

GRAM1 and GRAM0 bits are used in RAM guard function.

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

Reset signal generation clears this register to 00H.

Figure 27 - 9 Format of Invalid memory access detection control register (IAWCTL)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GRAM1	GRAM0	RAM guard space Note
0	0	Disabled. RAM can be written to.
0	1	The 128 bytes starting at the start RAM address
1	0	The 256 bytes starting at the start RAM address
1	1	The 512 bytes starting at the start RAM address

Note The RAM start address differs depending on the size of the RAM provided with the product.

27.3.5 SFR guard function

<R> This SFR guard function is used to protect data in the control registers used by the port function, interrupt function, clock control function, voltage detection function, and RAM parity error detection function.

If the SFR guard function is specified, writing to the specified SFRs is disabled, but reading from the SFRs can be carried out as usual.

27.3.5.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.

GPORT, GINT and GCSC bits are used in SFR guard function.

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

Reset signal generation clears this register to 00H.

Figure 27 - 10 Format of Invalid memory access detection control register (IAWCTL)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

GPORT	Control registers of port function guard
0	Disabled. Control registers of port function can be read or written to.
1	Enabled. Writing to control registers of port function is disabled. Reading is enabled. [Guarded SFR] PMxx, PUxx, PIMxx, POMxx, PMCxx Note

GINT	Registers of interrupt function guard
0	Disabled. Registers of interrupt function can be read or written to.
1	Enabled. Writing to registers of interrupt function is disabled. Reading is enabled. [Guarded SFR] IFxx, MKxx, PRxx, EGPx, EGNx

GCSC	Control registers of clock control function, voltage detector, and RAM parity error detection function guard
0	Disabled. Control registers of clock control function, voltage detector and RAM parity error detection function can be read or written to.
1	Enabled. Writing to control registers of clock control function, voltage detector and RAM parity error detection function is disabled. Reading is enabled. [Guarded SFR] CMC, CSC, OSTS, CKC, PERx, OSMC, LVIM, LVIS, RPECTL, DSCCTL RTCCL, PCKC, MCKC

Note Pxx (Port register) is not guarded.

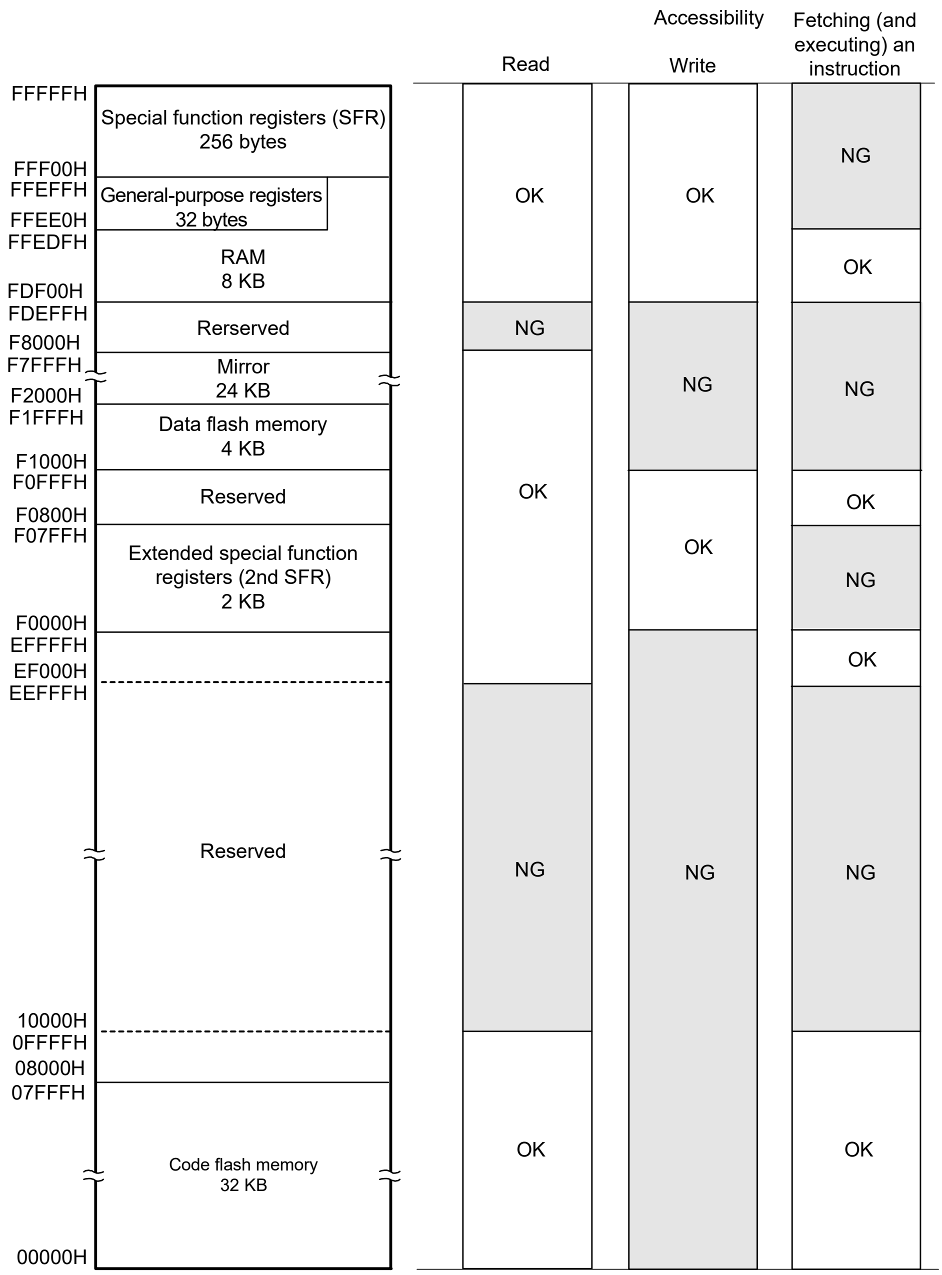
27.3.6 Invalid memory access detection function

The IEC60730 standard mandates checking that the CPU and interrupts are operating correctly.

The illegal memory access detection function triggers a reset if a memory space specified as access-prohibited is accessed.

The illegal memory access detection function applies to the areas indicated by NG in Figure 27 - 11.

Figure 27 - 11 Invalid access detection area



27.3.6.1 Invalid memory access detection control register (IAWCTL)

This register is used to control the detection of invalid memory access and RAM/SFR guard function.
IAWEN bit is used in invalid memory access detection function.
The IAWCTL register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 27 - 12 Format of Invalid memory access detection control register (IAWCTL)

Address: F0078H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IAWCTL	IAWEN Note	0	GRAM1	GRAM0	0	GPORT	GINT	GCSC

IAWEN Note	Control of invalid memory access detection
0	Disable the detection of invalid memory access.
1	Enable the detection of invalid memory access.

Note Only writing 1 to the IAWEN bit is enabled, not writing 0 to it after setting it to 1.

Remark By specifying WDTON = 1 for the option byte (watchdog timer operation enable), the invalid memory access detection function is enabled even if IAWEN = 0.

27.3.7 Frequency detection function

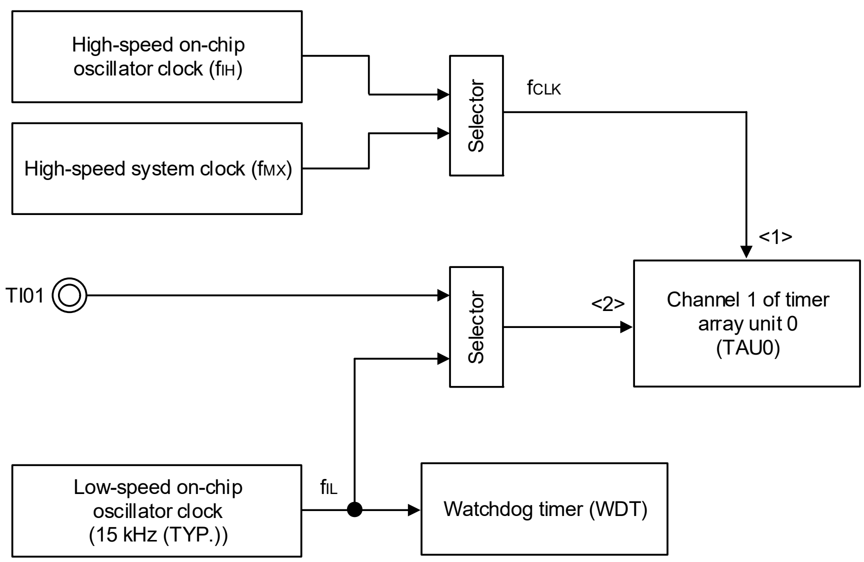
The IEC60730 standard mandates checking that the oscillation frequency is correct.
By using the CPU/peripheral hardware clock frequency (f_{CLK}) and measuring the pulse width of the input signal to channel 1 of timer array unit 0 (TAU0), whether the proportional relationship between the two clock frequencies is correct can be determined.
Note that, however, if one or both clock operations are stopped, the proportional relationship between the clocks cannot be determined.

<Clocks to be compared>

<1> CPU/peripheral hardware clock frequency (f_{CLK}):
- High-speed on-chip oscillator clock (f_{IH})
- High-speed system clock (f_{MX})

<2> Input to channel 1 of timer array unit 0
- Timer input to channel 1 (TI01)
- Low-speed on-chip oscillator clock (f_{IL}: 15 kHz (typ.))

Figure 27 - 13 Configuration of Frequency Detection Function



If pulse interval measurement results in an abnormal value, it can be concluded that the clock frequency is abnormal.
For how to execute pulse interval measurement, see **6.8.3 Operation as input pulse interval measurement**.

27.3.7.1 Timer input select register 0 (TIS0)

The TIS0 register is used to select the timer input of channels 0 and 1 of timer array unit 0 (TAU0).
The TIS0 register can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 27 - 14 Format of Timer input select register 0 (TIS0)

Address: F0074H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	0	0	0	TIS04	0	TIS02	TIS01	TIS00

TIS04	Selection of timer input used with channel 0
0	Input signal of timer input pin (TI00)
1	Event input signal from ELC

TIS02	TIS01	TIS00	Selection of timer input used with channel 1
0	0	0	Input signal of timer input pin (TI01)
0	0	1	Event input signal from ELC
0	1	0	Input signal of timer input pin (TI01)
0	1	1	
1	0	0	Low-speed on-chip oscillator clock (f_{IL})
Other than above			Setting prohibited

27.3.8 A/D test function

The IEC60730 standard mandates testing the A/D converter. The A/D test function is used to check whether the A/D converter is operating normally by executing A/D conversions of the positive reference voltage and negative reference voltage of the A/D converter, analog input channel (ANI), and internal reference voltage. For details on the checking method, refer to the safety function (A/D test) application note (R01AN0955).

The analog multiplexer can be checked using the following procedure.

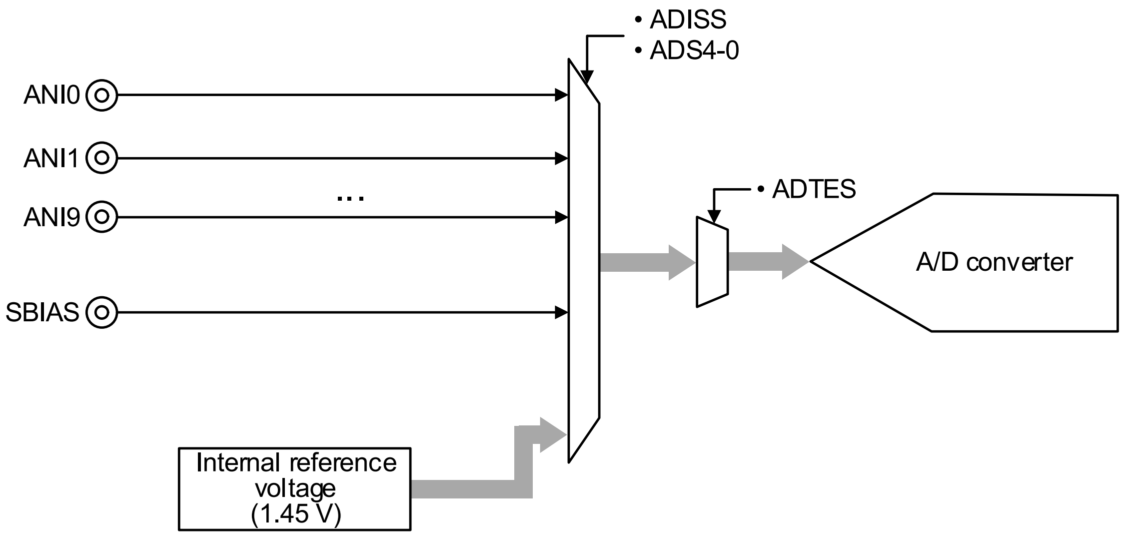
(1) Select the ANIx pin as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 0, 0).
(2) Perform A/D conversion for the ANIx pin (conversion result 1-1).
(3) Select the negative reference voltage of the A/D converter as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 1, 0).
(4) Perform A/D conversion of the negative reference voltage of the A/D converter (conversion result 2-1).
(5) Select the ANIx pin as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 0, 0).
(6) Perform A/D conversion for the ANIx pin (conversion result 1-2).
(7) Select the positive reference voltage of the A/D converter as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 1, 1).
(8) Perform A/D conversion of the positive reference voltage of the A/D converter (conversion result 2-2).
(9) Select the ANIx pin as the target for A/D conversion by setting the ADTES register (ADTES1, ADTES0 = 0, 0).
(10) Perform A/D conversion for the ANIx pin (conversion result 1-3).
(11) Make sure that “conversion result 1-1” = “conversion result 1-2” = “conversion result 1-3”.
(12) Make sure that the A/D conversion results of “conversion result 2-1” are all 0 and those of “conversion result 2-2” are all 1.

Using the procedure above can confirm that the analog multiplexer is selected and all wiring is connected.

Remark 1. If the analog input voltage is variable during conversion in steps (1) to (10) above, use another method to check the analog multiplexer.

Remark 2. The conversion results might contain an error. Consider an appropriate level of error when comparing the conversion results.

Figure 27 - 15 Configuration of A/D Test Function



27.3.8.1 A/D test register (ADTES)

This register is used to select the analog input channel (ANIxx), MEMS sensor bias output voltage, or internal reference voltage (1.45 V) as the target of A/D conversion.

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

Reset signal generation clears this register to 00H.

Figure 27 - 16 Format of A/D test register (ADTES)

Address: F0013H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADTES	0	0	0	0	0	0	ADTES1	ADTES0

ADTES1	ADTES0	A/D conversion target
0	0	Specified using the analog input channel specification register (ADS).
0	1	Setting prohibited
1	0	Negative reference voltage (AVss)
1	1	Positive reference voltage (selected by the ADREFP1 and ADREFP0 bits in A/D converter mode register 2 (ADM2))

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

27.3.8.2 Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted.

Set A/D test register (ADTES) to 00H when measuring the ANIxx, MEMS sensor bias output, or internal reference voltage (1.45 V).

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

Reset signal generation clears this register to 00H.

Figure 27 - 17 Format of Analog input channel specification register (ADS)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	ADISS	0	0	ADS4	ADS3	ADS2	ADS1	ADS0

○ Select mode (ADMD = 0)

ADISS	ADS4	ADS3	ADS2	ADS1	ADS0	Analog input channel	Input source
0	0	1	0	0	0	ANI0	ANI0 pin Note
0	0	1	0	0	1	ANI1/AMP0O	ANI1/AMP0O pin
0	0	1	0	1	0	ANI2/ANX0	ANI2/AMP0N/ANX0 pin
0	0	1	0	1	1	ANI3/ANX1	ANI3/AMP0P/ANX1 pin
0	0	1	1	0	0	ANI4/AMP1O	ANI4/AMP1O pin
0	0	1	1	0	1	ANI5/ANX2	P42/ANI5/AMP1N/ANX2 pin
0	0	1	1	1	0	ANI6/ANX3	P41/ANI6/AMP1P/ANX3 pin
0	0	1	1	1	1	ANI7/AMP2O	ANI7/AMP2O pin
0	1	0	0	0	0	ANI8/ANX4	P17/ANI8/AMP2N/ANX4 pin Note
0	1	0	0	0	1	ANI9/ANX5	P16/ANI9/AMP2P/ANX5 pin
0	1	0	0	1	0	—	SBIAS pin
1	0	0	0	0	1	—	Internal reference voltage output (1.45 V)
Other than above						Setting prohibited	

Note Do not specify this setting for 32-pin products.

Caution 1. Be sure to clear bits 5 and 6 to 0.

Caution 2. For ports that set to analog input using the PMC register, select input mode by using port mode register 1 or 4 (PM1, PM4).

Caution 3. Do not use the ADS register to set ports that to be set as digital I/O by using port mode control register 1 or 4 (PMC1, PMC4).

Caution 4. Only rewrite the value of the ADISS bit while conversion is stopped (ADCS = 0, ADCE = 0).

Caution 5. If ADISS is set to 1, the internal reference voltage output (1.45 V) cannot be used for the positive reference voltage. Also, the first conversion result cannot be used after ADISS is set to 1. For details on the setup flow, see 16.7.4 Setup when internal reference voltage for A/D converter is selected (example for software trigger mode and one-shot conversion mode).

Caution 6. Do not set ADISS to 1 when entering STOP mode. With ADISS = 1, the current value of the A/D converter reference voltage current (I_{ADREF}) listed in 33.3.2 or 34.3.2 Supply current characteristics is added.

27.3.9 Digital output signal level detection function for I/O pins

In the IEC60730, it is required to check that the I/O function correctly operates.
By using the digital output signal level detection function for I/O pins, the digital output level of the pin can be read when the port is set to output mode.

27.3.9.1 Port mode select register (PMS)

This register is used to select the output level from output latch level or pin output level when the pin is output mode in which PMm bit of port mode register (PMm) is 0.
This register can be set by a 1-bit or 8-bit memory manipulation instruction.
Reset signal generation clears these registers to 00H.

Figure 27 - 18 Format of Port mode select register (PMS)

Address: F007BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
PMS	0	0	0	0	0	0	0	PMS0

PMS0	Method for selecting output level to be read when pin is output mode
0	Pmn register value is read.
1	Digital output level of the pin is read.

Caution 1. While the PMS0 bit in the PMS register is set to 1, do not change the value of the port register (Pxx) using a bit manipulation instruction. To change the value of the port register (Pxx), use an 8-bit data manipulation instruction.

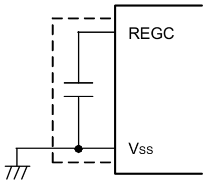
Caution 2. When the digital output level of a pin that is held in the high-impedance state by the timer RD pulse output forced cutoff function, the read value is 0.

Remark m = 1 or 4
n = 0 to 7

CHAPTER 28 REGULATOR

28.1 Regulator Overview

The RL78/I1E contains a circuit for operating the device with a constant voltage. At this time, in order to stabilize the regulator output voltage, connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF). Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.



Caution Keep the wiring length as short as possible for the broken-line part in the above figure.

The regulator output voltage, see **Table 28 - 1**.

Table 28 - 1 Regulator Output Voltage Conditions

Mode	Output Voltage	Condition
HS (high-speed main) mode	1.86 V	In STOP mode
	2.1 V	Other than above (include during OCD mode) Note

Note When the CPU shifts to the low-speed on-chip oscillator clock operation or STOP mode during the on-chip debugging, the regulator output voltage is kept at 2.1 V (not decline to 1.86 V).

CHAPTER 29 OPTION BYTE

29.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory of the RL78/I1E form an option byte area.
Option bytes consist of user option byte (000C0H to 000C2H) and on-chip debug option byte (000C3H).
Upon power application or resetting and starting, an option byte is automatically referenced and a specified function is set. When using the product, be sure to set the following functions by using the option bytes.
For the bits to which no function is allocated, do not change their initial values.
To use the boot swap operation during self-programming, 000C0H to 000C3H are replaced by 010C0H to 010C3H. Therefore, set the same values as 000C0H to 000C3H to 010C0H to 010C3H.

Remark The option bytes should always be set regardless of whether each function is used.

29.1.1 User option byte (000C0H to 000C2H/010C0H to 010C2H)

(1) 000C0H/010C0H

○ Setting of watchdog timer operation
- Enabling or disabling of counter operation
- Enabling or disabling of counter operation in the HALT or STOP mode

○ Setting of interval time of watchdog timer

○ Setting of window open period of watchdog timer

○ Setting of interval interrupt of watchdog timer
- Interval interrupt is used or not used

Caution Set the same value as 000C0H to 010C0H when the boot swap operation is used because 000C0H is replaced by 010C0H.

(2) 000C1H/010C1H

○ Setting of LVD operation mode
- Interrupt & reset mode
- Reset mode
- Interrupt mode
- LVD off (external reset input from the $\overline{\text{RESET}}$ pin is used)

○ Setting of LVD detection level (V_{LVDH}, V_{LVDL}, V_{LVD})

Caution 1. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 33.4 or 34.4 AC Characteristics. This is done by utilizing the voltage detector or controlling the externally input reset signal. After the power supply is turned off, this LSI should be placed in the STOP mode, or placed in the reset state by utilizing the voltage detector or controlling the externally input reset signal, before the voltage falls below the operating range. The range of operating voltage varies with the setting of the user option byte (000C2H or 010C2H).

Caution 2. Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

(3) 000C2H/010C2H

○ Setting of flash operation mode
  • HS (high-speed main) mode

○ Setting of the frequency of the high-speed on-chip oscillator
  • Select from 1 MHz to 32 MHz.

Caution Set the same value as 000C2H to 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.

29.1.2 On-chip debug option byte (000C3H/ 010C3H)

○ Control of on-chip debug operation
  • On-chip debug operation is disabled or enabled.

○ Handling of data of flash memory in case of failure in on-chip debug security ID authentication
  • Data of flash memory is erased or not erased in case of failure in on-chip debug security ID authentication.

Caution Set the same value as 000C3H to 010C3H when the boot swap operation is used because 000C3H is replaced by 010C3H.

29.2 Format of User Option Byte

The format of user option byte is shown below.

Figure 29 - 1 Format of User Option Byte (000C0H/010C0H)

Address: 000C0H/010C0H Note 1

7	6	5	4	3	2	1	0
WDTINT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTINT	Use of interval interrupt of watchdog timer
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% + 1/2 f_{IL} of the overflow time is reached.

WINDOW1	WINDOW0	Watchdog timer window open period Note 2
0	0	Setting prohibited
0	1	50%
1	0	75%
1	1	100%

WDTON	Operation control of watchdog timer counter
0	Counter operation disabled (counting stopped after reset)
1	Counter operation enabled (counting started after reset)

WDCS2	WDCS1	WDCS0	Watchdog timer overflow time (f_{IL} = 17.25 kHz (MAX.))
0	0	0	$2^6/f_{IL}$ (3.71 ms)
0	0	1	$2^7/f_{IL}$ (7.42 ms)
0	1	0	$2^8/f_{IL}$ (14.84 ms)
0	1	1	$2^9/f_{IL}$ (29.68 ms)
1	0	0	$2^{11}/f_{IL}$ (118.72 ms)
1	0	1	$2^{13}/f_{IL}$ (474.90 ms)
1	1	0	$2^{14}/f_{IL}$ (949.80 ms)
1	1	1	$2^{16}/f_{IL}$ (3799.19 ms)

WDSTBYON	Operation control of watchdog timer counter (HALT/STOP mode)
0	Counter operation stopped in HALT/STOP mode Note 2
1	Counter operation enabled in HALT/STOP mode

Note 1. Set the same value as 000C0H to 010C0H when the boot swap operation is used because 000C0H is replaced by 010C0H.

Note 2. The window open period is 100% when WDSTBYON = 0, regardless the value of the WINDOW1 and WINDOW0 bits.

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

Figure 29 - 2 Format of User Option Byte (000C1H/010C1H) (1/4)

Address: 000C1H/010C1H Note

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD setting (interrupt & reset mode)

Detection voltage			Option byte setting value						
V_{LVDH}		V_{LVDL}	VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge	Falling edge						LVIMDS1	LVIMDS0
3.02 V	2.96 V	2.55 V	0	0	0	0	1	1	0
3.22 V	3.15 V					0	0		
4.42 V	5.32 V	2.75 V		0	1	0	0		
4.62 V	4.52 V			1	0	0	0		
3.32 V	3.15 V			1	1	0	1		
4.74 V	4.64 V					0	0		
—			Settings other than the above are prohibited						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Caution Be sure to set bit 4 to 1.

Remark 1. For details on the LVD circuit, see **CHAPTER 26 VOLTAGE DETECTOR**.

Remark 2. The detection voltage is a typical value. For details, see **33.6.8** or **34.6.8** LVD characteristics.

Figure 29 - 3 Format of User Option Byte (000C1H/010C1H) (2/4)

Address: 000C1H/010C1H Note

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

- LVD setting (reset mode)

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
2.61 V	2.55 V	0	0	0	1	1	1	1
2.81 V	2.75 V		1	1	1	1		
3.02 V	2.96 V		0	0	0	1		
3.22 V	3.15 V		1	1	0	1		
4.42 V	4.32 V		0	1	0	0		
4.62 V	4.52 V		1	0	0	0		
4.74 V	4.64 V		1	1	0	0		
—		Settings other than the above are prohibited						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Caution Be sure to set bit 4 to 1.

Remark 1. For details on the LVD circuit, see **CHAPTER 26 VOLTAGE DETECTOR**.

Remark 2. The detection voltage is a typical value. For details, see **33.6.8** or **34.6.8** LVD characteristics.

Figure 29 - 4 Format of User Option Byte (000C1H/010C1H) (3/4)

Address: 000C1H/010C1H Note

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD setting (interrupt mode)

Detection voltage		Option byte setting value						
V_{LVD}		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
2.61 V	2.55 V	0	0	0	1	1	0	1
2.81 V	2.75 V		1	1	1	1		
3.02 V	2.96 V		0	0	0	1		
3.22 V	3.15 V		1	1	0	1		
4.42 V	4.32 V		0	1	0	0		
4.62 V	4.52 V		1	0	0	0		
4.74 V	4.64 V		1	1	0	0		
—		Settings other than the above are prohibited						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Caution Be sure to set bit 4 to 1.

Remark 1. For details on the LVD circuit, see **CHAPTER 26 VOLTAGE DETECTOR**.

Remark 2. The detection voltage is a typical value. For details, see **33.6.8** or **34.6.8** LVD characteristics.

Figure 29 - 5 Format of User Option Byte (000C1H/010C1H) (4/4)

Address: 000C1H/010C1H Note

7	6	5	4	3	2	1	0
VPOC2	VPOC1	VPOC0	1	LVIS1	LVIS0	LVIMDS1	LVIMDS0

• LVD off setting (external reset input from the $\overline{\text{RESET}}$ pin is used)

Detection voltage		Option byte setting value						
VLVD		VPOC2	VPOC1	VPOC0	LVIS1	LVIS0	Mode setting	
Rising edge	Falling edge						LVIMDS1	LVIMDS0
—	—	1	×	×	×	×	×	1
—		Settings other than the above are prohibited						

Note Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

Caution 1. Be sure to set bit 4 to 1.

Caution 2. After power is supplied, the reset state must be retained until the operating voltage becomes in the range defined in 33.4 or 34.4 AC Characteristics. This is done by utilizing the voltage detector or controlling the externally input reset signal. After the power supply is turned off, this LSI should be placed in the STOP mode, or placed in the reset state by utilizing the voltage detector or controlling the externally input reset signal, before the voltage falls below the operating range. The range of operating voltage varies with the setting of the user option byte (000C2H or 010C2H).

Remark 1. ×: Don’t care

Remark 2. For details on the LVD circuit, see **CHAPTER 26 VOLTAGE DETECTOR**.

Remark 3. The detection voltage is a typical value. For details, see **33.6.8** or **34.6.8** LVD characteristics.

Figure 29 - 6 Format of Option Byte (000C2H/010C2H)

Address: 000C2H/010C2H Note

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

CMODE1	CMODE0	Setting of flash operation mode		
			Operating Frequency Range (f_{MAIN})	Operating Voltage Range (V_{DD})
1	1	HS (high-speed main) mode	1 to 16 MHz	2.4 to 5.5 V
			1 to 32 MHz	2.7 to 5.5 V
Other than above		Setting prohibited		

FRQSEL3	FRQSEL2	FRQSEL1	FRQSEL0	Frequency of the high-speed on-chip oscillator (f_{HOCO})
1	0	0	0	32 MHz
0	0	0	0	24 MHz
1	0	0	1	16 MHz
0	0	0	1	12 MHz
1	0	1	0	8 MHz
0	0	1	0	6 MHz
1	0	1	1	4 MHz
0	0	1	1	3 MHz
1	1	0	0	2 MHz
1	1	0	1	1 MHz
Other than above				Setting prohibited

Note Set the same value as 000C2H to 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.

Caution 1. Be sure to set bit 5 to 1. Be sure to clear bit 4 to 0.

Caution 2. The operating frequency range and operating voltage range depend on each operating mode of the flash memory. See 33.4 or 34.4 AC Characteristics for details.

29.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

Figure 29 - 7 Format of On-chip Debug Option Byte (000C3H/010C3H)

Address: 000C3H/010C3H Note

7	6	5	4	3	2	1	0
OCDENSET	0	0	0	0	1	0	OCDERSD

OCDENSET	OCDERSD	Control of on-chip debug operation
0	0	Disables on-chip debug operation.
0	1	Setting prohibited
1	0	Enables on-chip debugging. Erases data of flash memory in case of failures in authenticating on-chip debug security ID.
1	1	Enables on-chip debugging. Does not erases data of flash memory in case of failures in authenticating on-chip debug security ID.

Note Set the same value as 000C3H to 010C3H when the boot swap operation is used because 000C3H is replaced by 010C3H.

Caution Bits 7 and 0 (OCDENSET and OCDERSD) can only be specified a value.
Be sure to set 000010B to bits 6 to 1.

Remark The value on bits 3 to 1 will be written over when the on-chip debug function is in use and thus it will become unstable after the setting.
However, be sure to set the default values (0, 1, and 0) to bits 3 to 1 at setting.

29.4 Setting of Option Byte

The user option byte and on-chip debug option byte can be set using the assembler linker option, in addition to describing in the source. When doing so, the contents set by using the link option take precedence, even if descriptions exist in the source, as mentioned below.

A software description example of the option byte setting is shown below.

```
OPT       CSEG      OPT_BYTE
          DB        36H      ; Does not use interval interrupt of watchdog timer,
                             ; Enables watchdog timer operation,
                             ; Window open period of watchdog timer is 50%,
                             ; Overflow time of watchdog timer is 2^9/fIL,
                             ; Stops watchdog timer operation during HALT/STOP mode
          DB        16H      ; Select 2.55 V for VLVDL
                             ; Select rising edge 3.02 V, falling edge 2.96 V for VLVDH
                             ; Select the interrupt & reset mode as the LVD operation mode
          DB        EDH      ; Select the HS (high-speed, main) mode as the flash operation mode
                               and 1 MHz as the frequency of the high-speed on-chip oscillator clock
          DB        85H      ; Enables on-chip debug operation, does not erase flash memory
                               data when security ID authorization fails
```

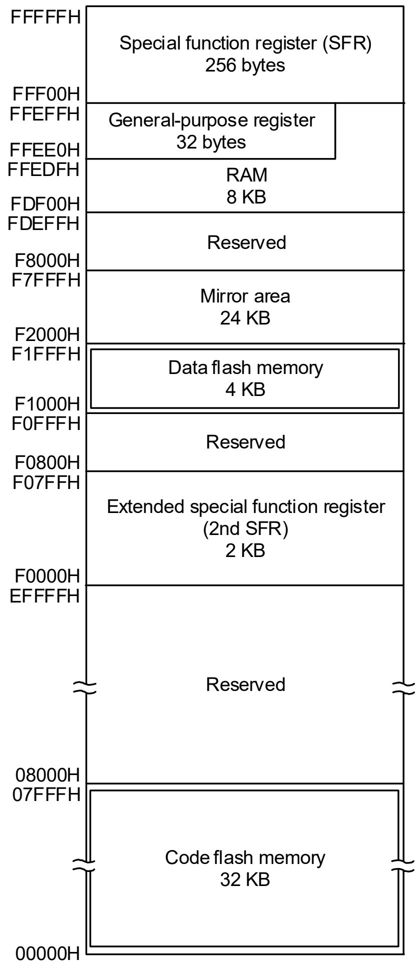
When the boot swap function is used during self-programming, 000C0H to 000C3H is switched to 010C0H to 010C3H. Describe to 010C0H to 010C3H, therefore, the same values as 000C0H to 000C3H as follows.

```
OPT2    CSEG      AT      010C0H
        DB                36H      ; Does not use interval interrupt of watchdog timer,
                                   ; Enables watchdog timer operation,
                                   ; Window open period of watchdog timer is 50%,
                                   ; Overflow time of watchdog timer is 2^9/fIL,
                                   ; Stops watchdog timer operation during HALT/STOP mode
        DB                16H      ; Select 2.55 V for VLVDL
                                   ; Select rising edge 3.02 V, falling edge 2.96 V for VLVDH
                                   ; Select the interrupt & reset mode as the LVD operation mode
        DB                EDH      ; Select the HS (high-speed, main) mode as the flash operation mode
                                     and 1 MHz as the frequency of the high-speed on-chip oscillator clock
        DB                85H      ; Enables on-chip debug operation, does not erase flash memory
                                     data when security ID authorization fails
```

Caution To specify the option byte by using assembly language, use OPT_BYTE as the relocation attribute name of the CSEG pseudo instruction. To specify the option byte to 010C0H to 010C3H in order to use the boot swap function, use the relocation attribute AT to specify an absolute address.

CHAPTER 30 FLASH MEMORY

The RL78 microcontroller incorporates the flash memory to which a program can be written, erased, and overwritten while mounted on the board. The flash memory includes the “code flash memory”, in which programs can be executed, and the “data flash memory”, an area for storing data.



The following methods for programming the flash memory are available.
The code flash memory can be rewritten to through serial programming using a flash memory programmer or an external device (UART communication), or through self-programming.

- Serial Programming Using Flash Memory Programmer (see **30.1**)
  Data can be written to the flash memory on-board or off-board by using a dedicated flash memory programmer.
- Serial Programming Using External Device (That Incorporates UART) (see **30.2**)
  Data can be written to the flash memory on-board through UART communication with an external device (microcontroller or ASIC).
- Self-Programming (see **30.6**)
  The user application can execute self-programming of the code flash memory by using the flash self-programming library.

The data flash memory can be rewritten to by using the data flash library during user program execution (background operation). For access and writing to the data flash memory, see **30.8 Data Flash**.

30.1 Serial Programming Using Flash Memory Programmer

The following dedicated flash memory programmer can be used to write data to the internal flash memory of the RL78 microcontroller.

- PG-FP5, FL-PR5
- E1 on-chip debugging emulator

Data can be written to the flash memory on-board or off-board, by using a dedicated flash memory programmer.

(1) On-board programming
The contents of the flash memory can be rewritten after the RL78 microcontroller has been mounted on the target system. The connectors that connect the dedicated flash memory programmer must be mounted on the target system.

(2) Off-board programming
Data can be written to the flash memory with a dedicated program adapter (FA series) before the RL78 microcontroller is mounted on the target system.

Remark FL-PR5 and FA series are products of Naito Densei Machida Mfg. Co., Ltd.

Table 30 - 1 Wiring Between RL78/I1E and Dedicated Flash Memory Programmer

Pin Configuration of Dedicated Flash Memory Programmer: Signal Name – PG-FP5, FL-PR5	Signal Name – E1 on-chip debugging emulator	I/O	Pin Function	Pin Name	Pin No. 32-pin HVQFN (5 × 5)	Pin No. 36-pin TFBGA (4 × 4)
—	TOOL0	I/O	Transmission/reception signal	TOOL0/P40	1	B5
SI/RxD	—	I/O	Transmission/reception signal			
—	$\overline{\text{RESET}}$	Output	Reset signal	$\overline{\text{RESET}}$	2	A4
/RESET	—	Output				
V_{DD}		I/O	V_{DD} voltage generation/ power monitoring	V_{DD}	8	A1
GND		—	Ground	V_{SS}	7	B1
				REGC Note	6	B2
FLMD1	EMV_{DD}	—	Driving power for TOOL0 pin	V_{DD}	8	A1

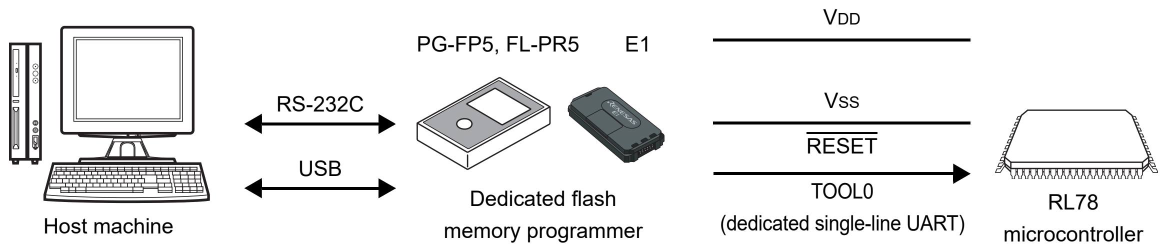
Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

Remark Pins that are not indicated in the above table can be left open when using the flash memory programmer for flash programming.

30.1.1 Programming Environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 30 - 1 Environment for Writing Program to Flash Memory



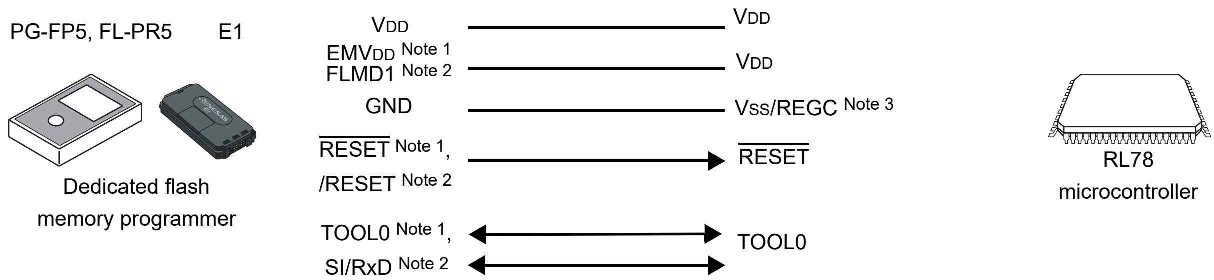
A host machine that controls the dedicated flash memory programmer is necessary.
To interface between the dedicated flash memory programmer and the RL78 microcontroller, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART.

30.1.2 Communication Mode

Communication between the dedicated flash memory programmer and the RL78 microcontroller is established by serial communication using the TOOL0 pin via a dedicated single-line UART of the RL78 microcontroller.

Transfer rate: 1 M, 500 k, 250 k, 115.2 kbps

Figure 30 - 2 Communication with Dedicated Flash Memory Programmer



Note 1. When using E1 on-chip debugging emulator.
Note 2. When using PG-FP5 or FL-PR5.
Note 3. Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

The dedicated flash memory programmer generates the following signals for the RL78 microcontroller. See the manual of PG-FP5, FL-PR5, or E1 on-chip debugging emulator for details.

Table 30 - 2 Pin Connection

Dedicated Flash Memory Programmer				RL78 microcontroller
Signal Name		I/O	Pin Function	Pin Name Note 2
PG-FP5, FL-PR5	E1 on-chip debugging emulator			
V_{DD}		I/O	V_{DD} voltage generation/power monitoring	V_{DD}
GND		—	Ground	V_{SS}, REGC Note 1
FLMD1	EMV_{DD}	—	Driving power for TOOL0 pin	V_{DD}
/RESET	—	Output	Reset signal	$\overline{RESET}$
—	$\overline{RESET}$	Output		
—	TOOL0	I/O	Transmission/reception signal	TOOL0
SI/RxD	—	I/O	Transmission/reception signal	

Note 1. Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

Note 2. Pins to be connected differ with the product. For details, see **Table 30 - 1**.

30.2 Serial Programming Using External Device (That Incorporates UART)

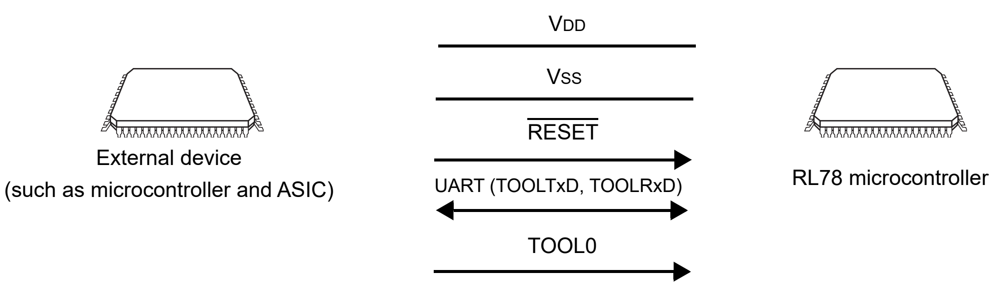
On-board data writing to the internal flash memory is possible by using the RL78 microcontroller and an external device (a microcontroller or ASIC) connected to a UART.

On the development of flash memory programmer by user, refer to the RL78 Microcontrollers (RL78 Protocol A) Programmer Edition Application (R01AN0815).

30.2.1 Programming Environment

The environment required for writing a program to the flash memory of the RL78 microcontroller is illustrated below.

Figure 30 - 3 Environment for Writing Program to Flash Memory



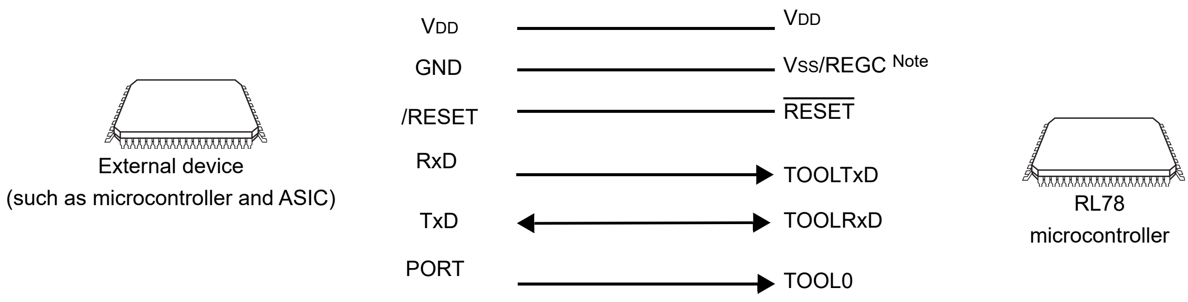
Processing to write data to or delete data from the RL78 microcontroller by using an external device is performed on-board. Off-board writing is not possible.

30.2.2 Communication Mode

Communication between the external device and the RL78 microcontroller is established by serial communication using the TOOLTxD and TOOLRxD pins via the dedicated UART of the RL78 microcontroller.

Transfer rate: 1 M, 500 k, 250 k, 115.2 kbps

Figure 30 - 4 Communication with External Device



Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

The external device generates the following signals for the RL78 microcontroller.

Table 30 - 3 Pin Connection

External Device			RL78 microcontroller
Signal Name	I/O	Pin Function	Pin Name
V_{DD}	I/O	V_{DD} voltage generation/power monitoring	V_{DD}
GND	—	Ground	V_{SS}, REGC Note
RESETOUT	Output	Reset signal output	$\overline{\text{RESET}}$
RxD	Input	Receive signal	TOOLTxD
TxD	Output	Transmit signal	TOOLRxD
PORT	Output	Mode signal	TOOL0

Note Connect REGC pin to ground via a capacitor (0.47 to 1 μF).

30.3 Connection of Pins on Board

To write the flash memory on-board by using the flash memory programmer, connectors that connect the dedicated flash memory programmer must be provided on the target system. First provide a function that selects the normal operation mode or flash memory programming mode on the board.
When the flash memory programming mode is set, all the pins not used for programming the flash memory are in the same status as immediately after reset. Therefore, if the external device does not recognize the state immediately after reset, the pins must be handled as described below.

Remark For the flash programming mode, see **30.4.2 Flash memory programming mode**.

30.3.1 P40/TOOL0 pin

In the flash memory programming mode, connect this pin to the dedicated flash memory programmer via an external 1 kΩ pull-up resistor.
When this pin is used as the port pin, use that by the following method.

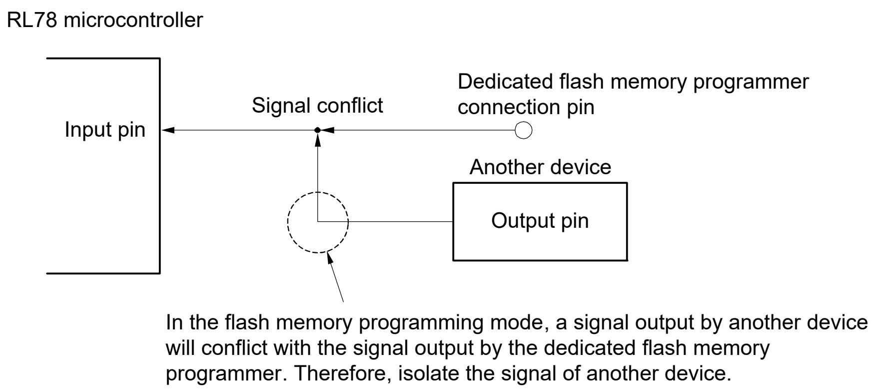
When used as an input pin: Input of low-level is prohibited for t_{HD} period after external reset release. However, when this pin is used via pull-down resistors, use the 500 kΩ or more resistors.
When used as an output pin: When this pin is used via pull-down resistors, use the 500 kΩ or more resistors.

Remark 1. t_{HD}: How long to keep the TOOL0 pin at the low level from when the external and internal resets end for setting of the flash memory programming mode (see **33.10** or **34.10 Timing for Switching Flash Memory Programming Modes**).
Remark 2. The SAU and IICA pins are not used for communication between the RL78 microcontroller and dedicated flash memory programmer, because single-line UART (TOOL0 pin) is used.

30.3.2 $\overline{\text{RESET}}$ pin

Signal conflict will occur if the reset signal of the dedicated flash memory programmer and external device are connected to the $\overline{\text{RESET}}$ pin that is connected to the reset signal generator on the board. To prevent this conflict, isolate the connection with the reset signal generator.
The flash memory will not be correctly programmed if the reset signal is input from the user system while the flash memory programming mode is set. Do not input any signal other than the reset signal of the dedicated flash memory programmer and external device.

Figure 30 - 5 Signal Conflict ($\overline{\text{RESET}}$ Pin)



30.3.3 Port pins

Example When the flash memory programming mode is set, all the pins not used for flash memory programming enter the same status as that immediately after reset. If external devices connected to the ports do not recognize the port status immediately after reset, the port pin must be connected to either V_{DD} or V_{SS} via a resistor.

30.3.4 REGC pin

Connect the REGC pin to GND via a capacitor having excellent characteristics (0.47 to 1 μF) in the same manner as during normal operation. Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.

30.3.5 X1 and X2 pins

Connect X1 and X2 in the same status as in the normal operation mode.

Remark In the flash memory programming mode, the high-speed on-chip oscillator clock (f_{IH}) is used.

30.3.6 Power supply

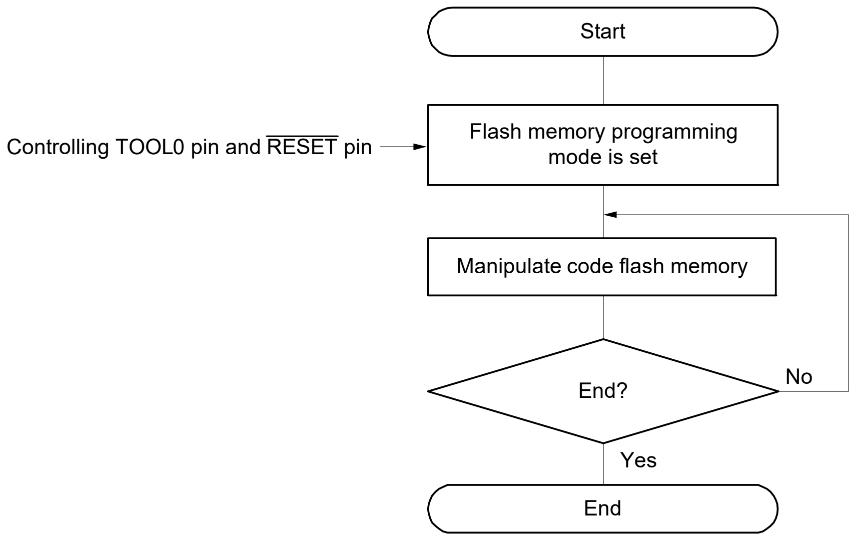
To use the supply voltage output of the flash memory programmer, connect the V_{DD} pin to V_{DD} of the flash memory programmer, and the V_{SS} pin to GND of the flash memory programmer.
To use the on-board supply voltage, connect in compliance with the normal operation mode.
However, when writing to the flash memory by using the flash memory programmer and using the on-board supply voltage, be sure to connect the V_{DD} and V_{SS} pins to V_{DD} and GND of the flash memory programmer to use the power monitor function with the flash memory programmer.

30.4 Programming Method

30.4.1 Serial programming procedure

The following figure illustrates a flow for rewriting the code flash memory through serial programming.

Figure 30 - 6 Code Flash Memory Manipulation Procedure



30.4.2 Flash memory programming mode

To rewrite the contents of the code flash memory through serial programming, specify the flash memory programming mode. To enter the mode, set as follows.

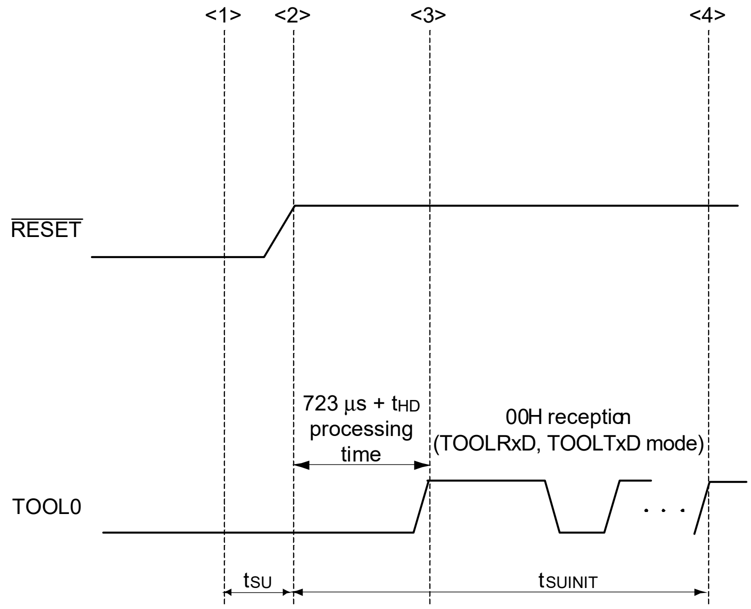
<When serial programming by using the dedicated flash memory programmer>
Connect the RL78 microcontroller to a dedicated flash memory programmer. Communication from the dedicated flash memory programmer is performed to automatically switch to the flash memory programming mode.

<When serial programming by using an external device>
Set the TOOL0 pin to the low level, and then cancel the reset (see **Table 30 - 4**). After that, enter flash memory programming mode according to the procedures <1> to <4> shown in **Figure 30 - 7**. For details, refer to the **RL78 microcontrollers (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

Table 30 - 4 Relationship Between TOOL0 Pin and Operation Mode After Reset Release

TOOL0	Operation Mode
V_{DD}	Normal operation mode
0 V	Flash memory programming mode

Figure 30 - 7 Setting of Flash Memory Programming Mode



<1> The low level is input to the TOOL0 pin.
<2> The external reset ends (POR and LVD reset must end before the external reset ends).
<3> The TOOL0 pin is set to the high level.
<4> Setting of the flash memory programming mode by UART reception and complete the baud rate setting.

Remark t_{SUINIT}: The segment shows that it is necessary to finish specifying the initial communication settings within 100 ms from when the external resets end.
t_{SU}: How long from when the TOOL0 pin is placed at the low level until a pin reset ends.
t_{HD}: How long to keep the TOOL0 pin at the low level from when the external resets end (the flash firmware processing time is excluded).

For details, see **33.10** or **34.10 Timing for Switching Flash Memory Programming Modes**.

There are two flash memory programming modes: wide voltage mode and full speed mode, but the RL78/I1E only supports full speed mode. The supply voltage value applied to the microcontroller during write operations and the setting information of the user option byte for setting of the flash memory programming mode determine which mode is selected.

When a dedicated flash memory programmer is used for serial programming, setting the voltage on GUI selects the mode automatically.

Table 30 - 5 Programming Modes and Voltages at Which Data Can Be Written, Erased, or Verified

Power Supply Voltage (V_{DD})	User Option Byte Setting for Switching to Flash Memory Programming Mode		Flash Programming Mode
	Flash Operation Mode	Operating Frequency (f_{CLK})	
2.7 V ≤ V_{DD} ≤ 5.5 V	Blank state		Full speed mode
	HS (high-speed main) mode	1 MHz to 32 MHz	Full speed mode
2.4 V ≤ V_{DD} < 2.7 V	Blank state		Full speed mode
	HS (high-speed main) mode	1 MHz to 16 MHz	Full speed mode

Remark For details about communication commands, see **30.4.4 Communication commands**.

30.4.3 Selecting communication mode

Communication mode of the RL78 microcontroller as follows.

Table 30 - 6 Communication Modes

Communication Mode	Standard Setting Note 1				Pins Used
	Port	Speed Note 2	Frequency	Multiply Rate	
1-line mode (when flash memory programmer is used, or when external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	—	—	TOOL0
Dedicated UART (when external device is used)	UART	115200 bps, 250000 bps, 500000 bps, 1 Mbps	—	—	TOOLTxD, TOOLRxD

Note 1. Selection items for Standard settings on GUI of the flash memory programmer.

Note 2. Because factors other than the baud rate error, such as the signal waveform slew, also affect UART communication, thoroughly evaluate the slew as well as the baud rate error.

30.4.4 Communication commands

The RL78 microcontroller executes serial programming through the commands listed in **Table 30 - 7**.

The signals sent from the dedicated flash memory programmer or external device to the RL78 microcontroller are called commands, and programming functions corresponding to the commands are executed. For details, refer to the **RL78 microcontroller (RL78 Protocol A) Programmer Edition Application Note (R01AN0815)**.

Table 30 - 7 Flash Memory Control Commands

Classification	Command Name	Function
Verify	Verify	Compares the contents of a specified area of the flash memory with data transmitted from the programmer.
Erase	Block Erase	Erases a specified area in the flash memory.
Blank check	Block Blank Check	Checks if a specified block in the flash memory has been correctly erased
Write	Programming	Writes data to a specified area in the flash memory [Note].
Getting information	Silicon Signature	Gets the RL78 microcontroller information (such as the part number, flash memory configuration, and programming firmware version).
	Checksum	Gets the checksum data for a specified area.
Security	Security Set	Sets security information.
	Security Get	Gets security information.
	Security Release	Release setting of prohibition of writing.
Others	Reset	Used to detect synchronization status of communication.
	Baud Rate Set	Sets baud rate when UART communication mode is selected.

Note Confirm that no data has been written to the write area. Because data cannot be erased after block erase is prohibited, do not write data if the data has not been erased.

Product information (such as product name and firmware version) can be obtained by executing the “Silicon Signature” command.

Tables 30 - 8 and 30 - 9 show signature data list and example of signature data list.

Table 30 - 8 Signature Data List

Field name	Description	Number of transmit data
Device code	The serial number assigned to the device	3 bytes
Device name	Device name (ASCII code)	10 bytes
Code flash memory area last address	Last address of code flash memory area (Sent from lower address. Example. 00000H to 07FFFH (32 KB) → FFH, 7FH, 00H)	3 bytes
Data flash memory area last address	Last address of data flash memory area (Sent from lower address. Example. F1000H to F1FFFH (8 KB) → FFH, 1FH, 0FH)	3 bytes
Firmware version	Version information of firmware for programming (Sent from upper address. Example. From Ver. 1.23 → 01H, 02H, 03H)	3 bytes

Table 30 - 9 Signature Data List

Field name	Description	Number of transmit data	Data (hexadecimal)
Device code	RL78 protocol A	3 bytes	10 00 06
Device name	R5F11CBC	10 bytes	52 = “R” 35 = “5” 46 = “F” 31 = “1” 31 = “1” 43 = “C” 42 = “B” 43 = “C” 20 = “ ” 20 = “ ”
Code flash memory area last address	Code flash memory area 00000H to 07FFFH (32 KB)	3 bytes	FF 7F 00
Data flash memory area last address	Data flash memory area F1000H to F1FFFH (4 KB)	3 bytes	FF 1F 0F
Firmware version	Ver.1.23	3 bytes	01 02 03

30.5 Processing Time for Each Command When PG-FP5 Is in Use (Reference Value)

The following shows the processing time for each command (reference value) when PG-FP5 is used as a dedicated flash memory programmer.

Table 30 - 10 Processing Time for Each Command When PG-FP5 Is in Use (Reference Value)

PG-FP5 Command	Port: TOOL0 (UART)
	Speed: 1M bps
	32 KB
Erasing	1 s
Writing	1.5 s
Verification	1.5 s
Writing after erasing	2 s

Remark The command processing times (reference values) shown in the table are typical values under the following conditions.

Port: TOOL0 (single-line UART)

Speed: 1,000,000 bps

Mode: Full speed mode (flash operation mode: HS (high speed main) mode)

30.6 Self-Programming

The RL78 microcontroller supports a self-programming function that can be used to rewrite the flash memory via a user program. Because this function allows a user application to rewrite the flash memory by using the RL78 microcontroller self-programming library, it can be used to upgrade the program in the field.

Caution 1. To prohibit an interrupt during self-programming, in the same way as in the normal operation mode, execute the flash self-programming library in the state where the IE flag is cleared (0) by the DI instruction. To enable an interrupt, clear (0) the interrupt mask flag to accept in the state where the IE flag is set (1) by the EI instruction, and then execute the flash self-programming library.

Caution 2. The high-speed on-chip oscillator should be kept operating during self-programming. If it is kept stopped, it should be operated (HIOSTOP = 0). Execute the flash self-programming library after 30 μs have elapsed.

Remark 1. For details of the self-programming function, refer to the **RL78 microcontroller Flash Self-Programming Library Type01 User's Manual (R01US0050)**.

Remark 2. For details of the time required to execute self-programming, see the notes on use that accompany the flash self-programming library tool.

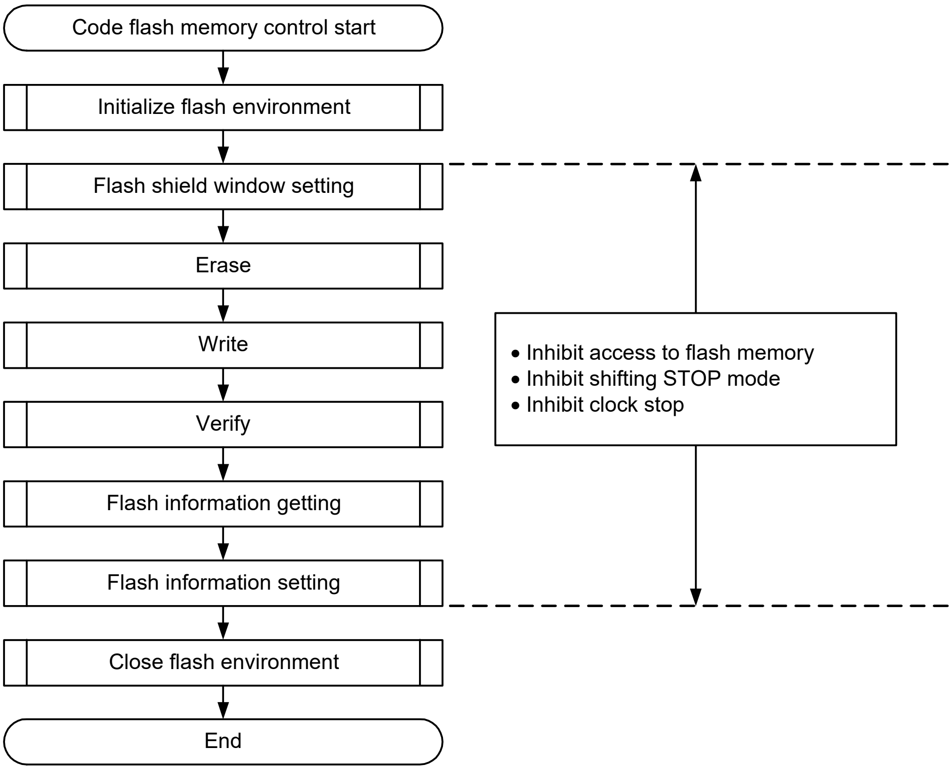
Self-programming can run in full speed mode during flash memory programming.
Specify the mode that corresponds to the flash operation mode specified in bits CMODE1 and CMODE0 in option byte 000C2H.
Specify the full speed mode when the HS (high-speed main) mode is specified.
If the argument fsl_flash_voltage_u08 is 00H when the FSL_Init function of the flash self-programming library provided by Renesas Electronics is executed, full speed mode is specified. If the argument is other than 00H, the wide voltage mode is specified.

Remark Using both the wide voltage mode and full speed mode imposes no restrictions on writing, erasing, or verification.

30.6.1 Self-programming procedure

The following figure illustrates a flow for rewriting the code flash memory by using a flash self-programming library.

Figure 30 - 8 Flow of Self-Programming (Rewriting Flash Memory)



30.6.2 Boot swap function

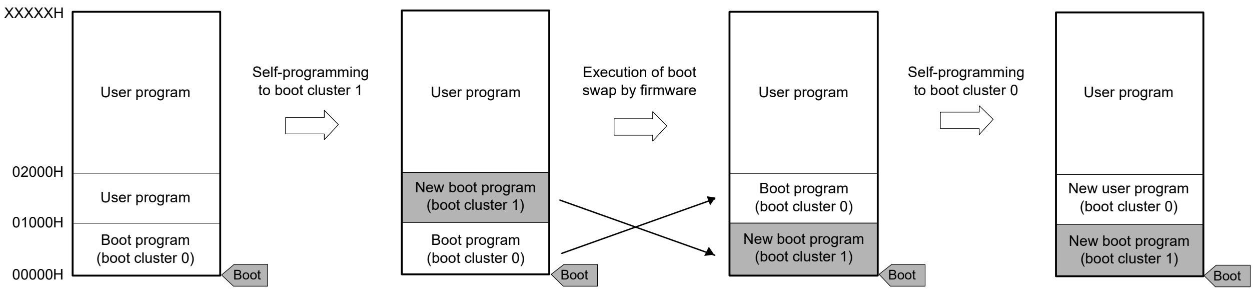
If rewriting the boot area failed by temporary power failure or other reasons, restarting a program by resetting or overwriting is disabled due to data destruction in the boot area.

The boot swap function is used to avoid this problem.

Before erasing boot cluster 0 Note, which is a boot program area, by self-programming, write a new boot program to boot cluster 1 in advance. When the program has been correctly written to boot cluster 1, swap this boot cluster 1 and boot cluster 0 by using the set information function of the firmware of the RL78 microcontroller, so that boot cluster 1 is used as a boot area. After that, erase or write the original boot program area, boot cluster 0. As a result, even if a power failure occurs while the area is being rewritten, the program is executed correctly because it is booted from boot cluster 1 to be swapped when the program is reset and started next.

Note A boot cluster is a 4 KB area and boot clusters 0 and 1 are swapped by the boot swap function.

Figure 30 - 9 Boot Swap Function

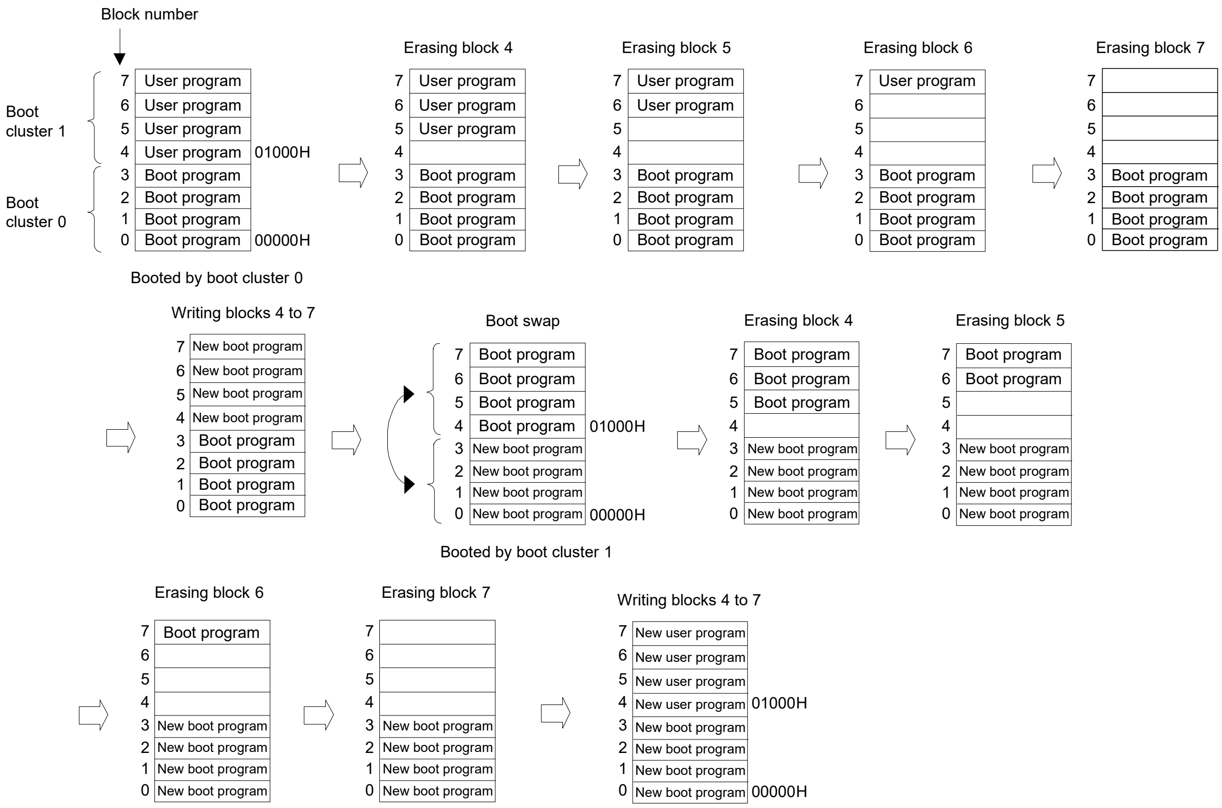


In an example of above figure, it is as follows.

Boot cluster 0: Boot area before boot swap

Boot cluster 1: Boot area after boot swap

Figure 30 - 10 Example of Executing Boot Swapping



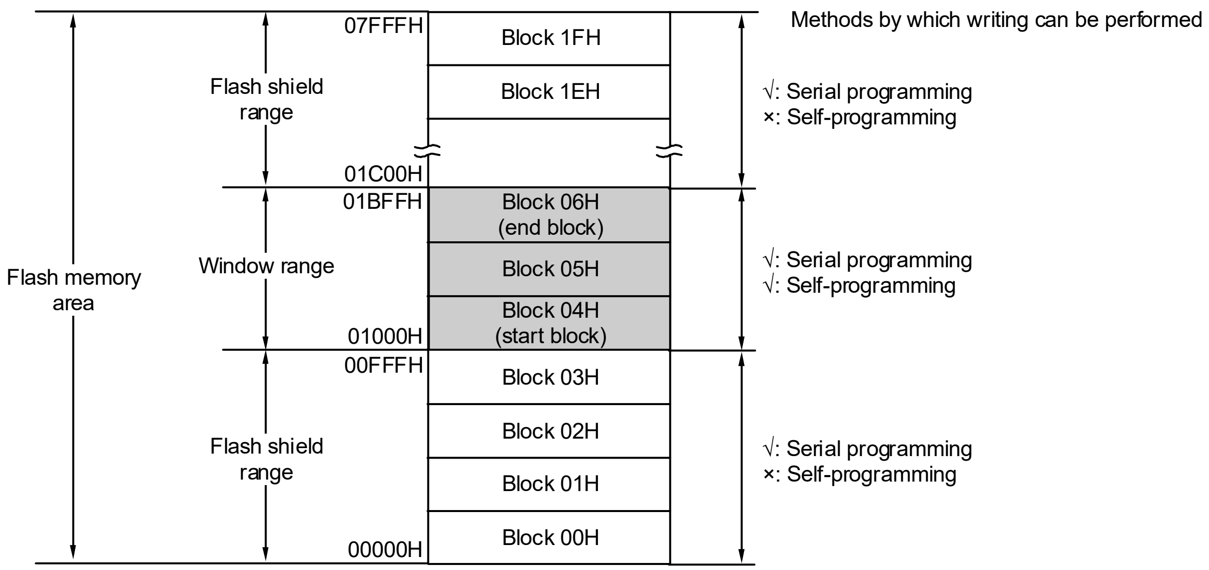
30.6.3 Flash shield window function

The flash shield window function is provided as one of the security functions for self-programming. It disables writing to and erasing areas outside the range specified as a window only during self-programming.

The window range can be set by specifying the start and end blocks. The window range can be set or changed during both serial programming and self-programming.

Writing to and erasing areas outside the window range are disabled during self-programming. During serial programming, however, areas outside the range specified as a window can be written and erased.

Figure 30 - 11 Flash Shield Window Setting Example
(Target Device: R5F11CBC, Start Block: 04H, End Block: 06H)



Caution 1. If the rewrite-prohibited area of the boot cluster 0 overlaps with the flash shield window range, prohibition to rewrite the boot cluster 0 takes priority.

Caution 2. The flash shield window can only be used for the code flash memory (and is not supported for the data flash memory).

Table 30 - 11 Relationship between Flash Shield Window Function Setting/Change Methods and Commands

Programming conditions	Window Range Setting/ Change Methods	Execution Commands	
		Block erase	Write
Self-programming	Specify the starting and ending blocks by the flash self-programming library.	Block erasing is enabled only within the window range.	Writing is enabled only within the range of window range.
Serial programming	Specify the starting and ending blocks on GUI of dedicated flash memory programmer, etc.	Block erasing is enabled also outside the window range.	Writing is enabled also outside the window range.

Remark See **30.7 Security Settings** to prohibit writing/erasing during serial programming.

30.7 Security Settings

The RL78 microcontroller supports a security function that prohibits rewriting the user program written to the internal flash memory, so that the program cannot be changed by an unauthorized person.
The operations shown below can be performed using the Security Set command.

- Disabling block erase
  Execution of the block erase command for a specific block in the flash memory is prohibited during serial programming. However, blocks can be erased by means of self-programming.

- Disabling write
  Execution of the write command for entire blocks in the flash memory is prohibited during serial programming. However, blocks can be written by means of self-programming.
  After the setting of prohibition of writing is specified, releasing the setting by the Security Release command is enabled by a reset.

- Disabling rewriting boot cluster 0
  Execution of the block erase command and write command on boot cluster 0 (00000H to 00FFFH) in the flash memory is prohibited by this setting.

The block erase, write commands, and rewriting boot cluster 0 are enabled by the default setting when the flash memory is shipped. Security can be set by serial programming and self-programming. Each security setting can be used in combination.
Table 30 - 12 shows the relationship between the erase and write commands when the RL78 microcontroller security function is enabled.
After the security settings are specified, releasing the security settings by the Security Release command is enabled by a reset.

Caution The security function of the flash programmer does not support self-programming.

Remark To prohibit writing and erasing during self-programming, use the flash shield window function (see **30.6.3** for detail).

Table 30 - 12 Relationship Between Enabling Security Function and Command

(1) During serial programming

Valid Security	Executed Command	
	Block Erase	Write
Prohibition of block erase	Blocks cannot be erased.	Can be performed. Note
Prohibition of writing	Blocks can be erased.	Cannot be performed.
Prohibition of rewriting boot cluster 0	Boot cluster 0 cannot be erased.	Boot cluster 0 cannot be written.

Note Confirm that no data has been written to the write area. Because data cannot be erased after block erase is prohibited, do not write data if the data has not been erased.

(2) During self-programming

Valid Security	Executed Command	
	Block Erase	Write
Prohibition of block erase	Blocks can be erased.	Can be performed.
Prohibition of writing		
Prohibition of rewriting boot cluster 0	Boot cluster 0 cannot be erased.	Boot cluster 0 cannot be written.

Remark To prohibit writing and erasing during self-programming, use the flash shield window function (see **30.6.3** for detail).

Table 30 - 13 Setting Security in Each Programming Mode

(1) During serial programming

Security	Security Setting	How to Disable Security Setting
Prohibition of block erase	Set via GUI of dedicated flash memory programmer, etc.	Cannot be disabled after set.
Prohibition of writing		Set via GUI of dedicated flash memory programmer, etc.
Prohibition of rewriting boot cluster 0		Cannot be disabled after set.

Caution Releasing the setting of prohibition of writing is enabled only when the security is not set as the block erase prohibition and the boot cluster 0 rewrite prohibition with code flash memory area and data flash memory area being blanks.

(2) During self-programming

Security	Security Setting	How to Disable Security Setting
Prohibition of block erase	Set by using flash self-programming library.	Cannot be disabled after set.
Prohibition of writing		Cannot be disabled during self-programming (set via GUI of dedicated flash memory programmer, etc. during serial programming).
Prohibition of rewriting boot cluster 0		Cannot be disabled after set.

30.8 Data Flash

30.8.1 Data flash overview

An overview of the data flash memory is provided below.

- The user program can rewrite the data flash memory by using the flash data library. For details, refer to RL78 Family Flash Data Library User's Manual.
- The data flash memory can also be rewritten to through serial programming using the dedicated flash memory programmer or an external device.
- The data flash can be erased in 1-block (1 KB) units.
- The data flash can be accessed only in 8-bit units.
- The data flash can be directly read by CPU instructions.
- Instructions can be executed from the code flash memory while rewriting the data flash memory (that is, background operation (BGO) is supported).
- Because the data flash memory is an area exclusively used for data, it cannot be used to execute instructions.
- Accessing the data flash memory is not possible while rewriting the code flash memory (during self-programming).
- Manipulating the DFLCTL register is not possible while rewriting the data flash memory.
- Transition to the STOP mode is not possible while rewriting the data flash memory.

Caution 1. The data flash memory is stopped after a reset is canceled. The data flash control register (DFLCTL) must be set up in order to use the data flash memory.

Caution 2. The high-speed on-chip oscillator should be kept operating during data flash rewrite. If it is kept stopped, it should be operated (HIOSTOP = 0). Execute the flash self-programming library after 30 μs have elapsed.

Remark Refer to flash programming mode, see **30.6 Self-Programming**.

30.8.2 Register controlling data flash memory

30.8.2.1 Data flash control register (DFLCTL)

This register is used to enable or disable accessing to the data flash.
The DFLCTL register is set by a 1-bit or 8-bit memory manipulation instruction.
Reset input sets this register to 00H.

Figure 30 - 12 Format of Data flash control register (DFLCTL)

Address: F0090H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	<0>
DFLCTL	0	0	0	0	0	0	0	DFLEN

DFLEN	Data flash access control
0	Disables data flash access
1	Enables data flash access

Caution Manipulating the DFLCTL register is not possible while rewriting the data flash memory.

30.8.3 Procedure for accessing data flash memory

The data flash memory is initially stopped after a reset ends and cannot be accessed (read or programmed). To access the memory, perform the following procedure:

<1> Write 1 to bit 0 (DFLEN) of the data flash control register (DFLCTL).
<2> Wait for the setup to finish for software timer, etc.
The time setup takes differs for each main clock mode.
<Setup time for each main clock mode>
• HS (High-speed main): 5 μs
<3> After the wait, the data flash memory can be accessed.

Caution 1. Accessing the data flash memory is not possible during the setup time.
Caution 2. Transition to the STOP mode is not possible during the setup time. To enter the STOP mode during the setup time, clear DFLEN to 0 and then execute the STOP instruction.
Caution 3. The high-speed on-chip oscillator should be kept operating during data flash rewrite. If it is kept stopped, it should be operated (HIOSTOP = 0). The flash self-programming library should be executed after 30 μs have elapsed.

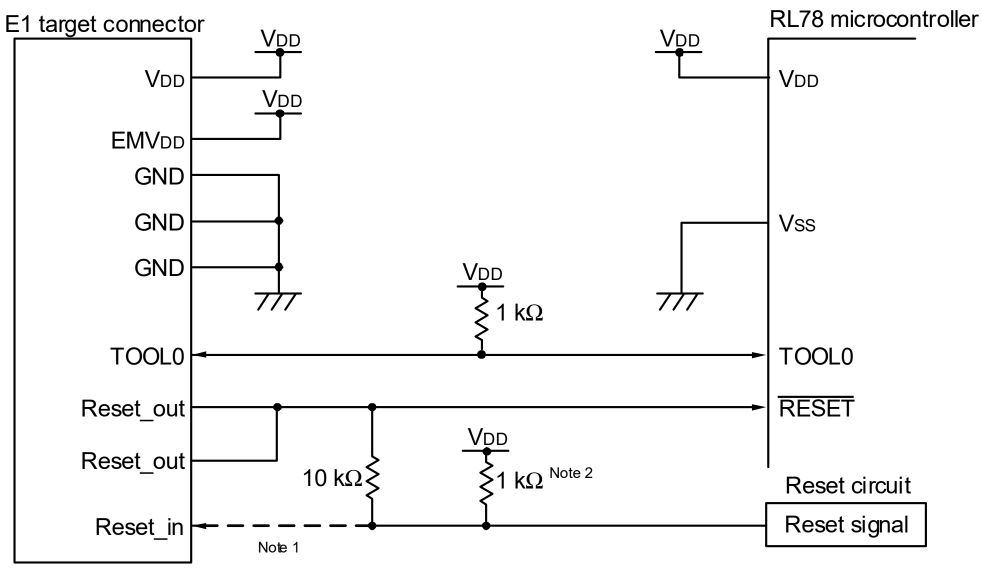
CHAPTER 31 ON-CHIP DEBUG FUNCTION

31.1 Connecting E1 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 E1 on-chip debugging emulator. Serial communication is performed by using a single-line UART that uses the TOOL0 pin.

Caution The RL78 microcontroller has an on-chip debug function, which is provided for development and evaluation. Do not use the on-chip debug function in products designated for mass production, because the guaranteed number of rewritable times of the flash memory may be exceeded when this function is used, and product reliability therefore cannot be guaranteed. Renesas Electronics is not liable for problems occurring when the on-chip debug function is used.

Figure 31 - 1 Connection Example of E1 On-chip Debugging Emulator



Note 1. Connecting the dotted line is not necessary during serial programming.

Note 2. If the reset circuit on the target system does not have a buffer and generates a reset signal only with resistors and capacitors, this pull-up resistor is not necessary.

Caution This circuit diagram is assumed that the reset signal outputs from an N-ch O.D. buffer (output resistor: 100 Ω or less)

31.2 On-Chip Debug Security ID

The RL78 microcontroller has an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 29 OPTION BYTE**) and an on-chip debug security ID setting area at 000C4H to 000CDH, to prevent third parties from reading memory content.

When the boot swap function is used, also set a value that is the same as that of 010C3H and 010C4H to 010CDH in advance, because 000C3H, 000C4H to 000CDH and 010C3H, and 010C4H to 010CDH are switched.

Table 31 - 1 On-Chip Debug Security ID

Address	On-Chip Debug Security ID
000C4H to 000CDH	Any ID code of 10 bytes
010C4H to 010CDH	

31.3 Securing of User Resources

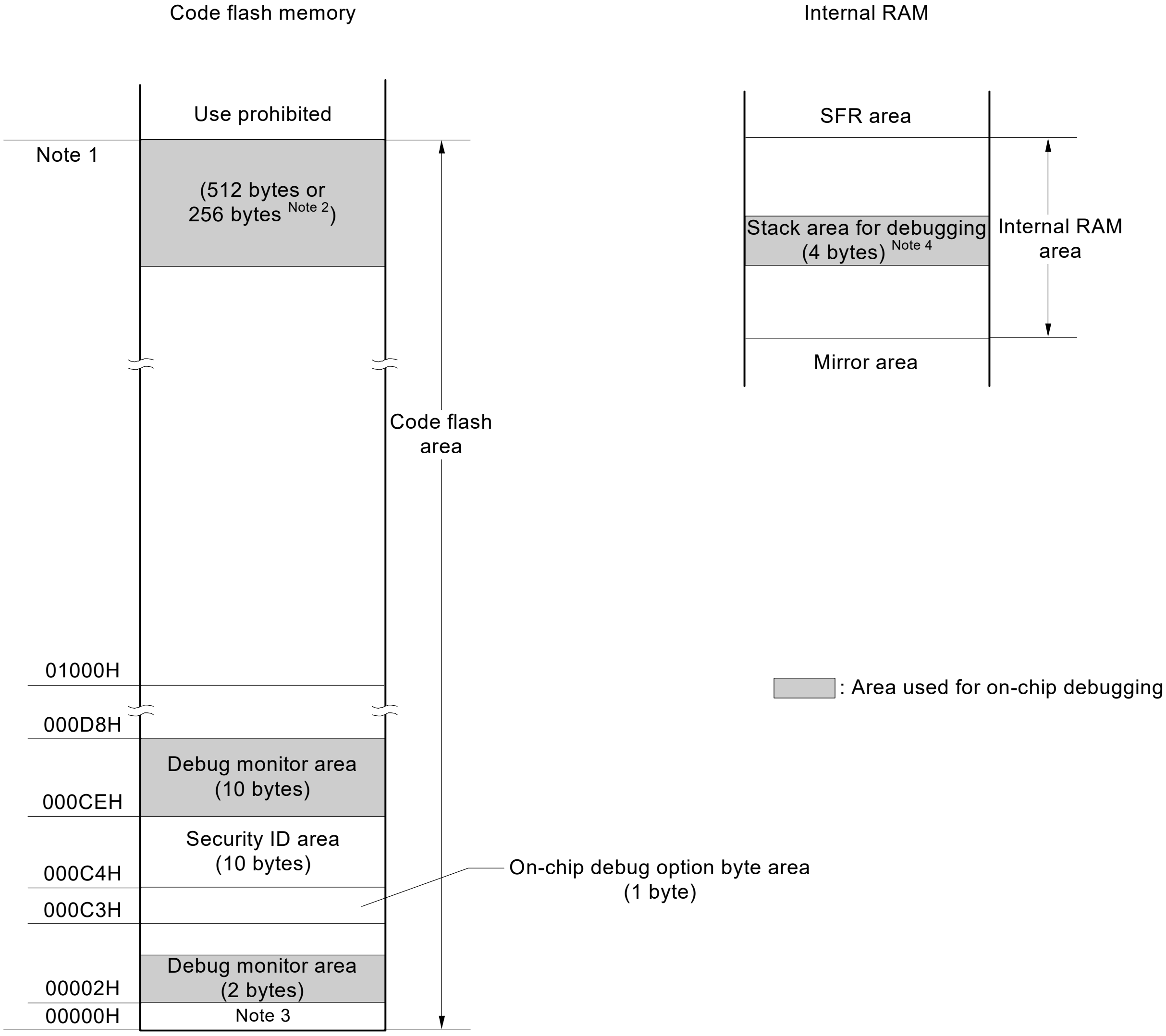
To perform communication between the RL78 microcontroller and E1 on-chip debugging emulator, as well as each debug function, the securing of memory space must be done beforehand.

If Renesas Electronics assembler or compiler is used, the items can be set by using link options.

(1) Securement of memory space

The shaded portions in Figure 31 - 2 are the areas reserved for placing the debug monitor program, so user programs or data cannot be allocated in these spaces. When using the on-chip debug function, these spaces must be secured so as not to be used by the user program. Moreover, this area must not be rewritten by the user program.

Figure 31 - 2 Memory Spaces Where Debug Monitor Programs Are Allocated



Note 1. Address differs depending on products as follows.

Products (code flash memory capacity)	Address of **Note 1.**
R5F11CxC (x = B, C)	7FFFFH

Note 2. When real-time RAM monitor (RRM) function and dynamic memory modification (DMM) function are not used, it is 256 bytes.

Note 3. In debugging, reset vector is rewritten to address allocated to a monitor program.

Note 4. Since this area is allocated immediately before the stack area, the address of this area varies depending on the stack increase and decrease. That is, 4 extra bytes are consumed for the stack area used.
When using self-programming, 12 extra bytes are consumed for the stack area used.

CHAPTER 32 BCD CORRECTION CIRCUIT

32.1 BCD Correction Circuit Function

The result of addition/subtraction of the BCD (binary-coded decimal) code and BCD code can be obtained as BCD code with this circuit.
The decimal correction operation result is obtained by performing addition/subtraction having the A register as the operand and then adding/ subtracting the BCD correction result register (BCDADJ).

32.2 Registers Used by BCD Correction Circuit

The BCD correction circuit uses the following registers.

• BCD correction result register (BCDADJ)

32.2.1 BCD correction result register (BCDADJ)

The BCDADJ register stores correction values for obtaining the add/subtract result as BCD code through add/subtract instructions using the A register as the operand.
The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags.
The BCDADJ register is read by an 8-bit memory manipulation instruction.
Reset input sets this register to undefined.

Figure 32 - 1 Format of BCD correction result register (BCDADJ)

Address: F00FEH After reset: Undefined R

Symbol	7	6	5	4	3	2	1	0
BCDADJ								

32.3 BCD Correction Circuit Operation

The basic operation of the BCD correction circuit is as follows.

(1) Addition: Calculating the result of adding a BCD code value and another BCD code value by using a BCD code value

<1> The BCD code value to which addition is performed is stored in the A register.

<2> By adding the value of the A register and the second operand (value of one more BCD code to be added) as are in binary, the binary operation result is stored in the A register and the correction value is stored in the BCD correction result register (BCDADJ).

<3> Decimal correction is performed by adding in binary the value of the A register (addition result in binary) and the BCDADJ register (correction value), and the correction result is stored in the A register and CY flag.

Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.

An example is shown below.

Examples 1: 99 + 89 = 188

Instruction			A Register	CY Flag	AC Flag	BCDADJ Register
MOV	A, #99H	; <1>	99H	—	—	—
ADD	A, #89H	; <2>	22H	1	1	66H
ADD	A, !BCDADJ	; <3>	88H	1	0	—

Examples 2: 85 + 15 = 100

Instruction			A Register	CY Flag	AC Flag	BCDADJ Register
MOV	A, #85H	; <1>	85H	—	—	—
ADD	A, #15H	; <2>	9AH	0	0	66H
ADD	A, !BCDADJ	; <3>	00H	1	1	—

Examples 3: 80 + 80 = 160

Instruction			A Register	CY Flag	AC Flag	BCDADJ Register
MOV	A, #80H	; <1>	80H	—	—	—
ADD	A, #80H	; <2>	00H	1	0	60H
ADD	A, !BCDADJ	; <3>	60H	1	0	—

(2) Subtraction: Calculating the result of subtracting a BCD code value from another BCD code value by using a BCD code value

<1> The BCD code value from which subtraction is performed is stored in the A register.

<2> By subtracting the value of the second operand (value of BCD code to be subtracted) from the A register as is in binary, the calculation result in binary is stored in the A register, and the correction value is stored in the BCD correction result register (BCDADJ).

<3> Decimal correction is performed by subtracting the value of the BCDADJ register (correction value) from the A register (subtraction result in binary) in binary, and the correction result is stored in the A register and CY flag.

Caution The value read from the BCDADJ register varies depending on the value of the A register when it is read and those of the CY and AC flags. Therefore, execute the instruction <3> after the instruction <2> instead of executing any other instructions. To perform BCD correction in the interrupt enabled state, saving and restoring the A register is required within the interrupt function. PSW (CY flag and AC flag) is restored by the RETI instruction.

An example is shown below.

Example: 91 - 52 = 39

Instruction			A Register	CY Flag	AC Flag	BCDADJ Register
MOV	A, #91H	; <1>	91H	—	—	—
SUB	A, #52H	; <2>	3FH	0	1	06H
SUB	A, !BCDADJ	; <3>	39H	0	0	—

CHAPTER 33 ELECTRICAL SPECIFICATIONS (G: T_A = -40 to +105°C)

This chapter describes the electrical specifications for the products “G: Industrial applications (T_A = -40 to +105°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 Functions to 2.2.1 Alternate functions other than AFE.

Caution 3. Please contact Renesas Electronics sales office for derating of operation under T_A = +85 to +105°C. Derating is the systematic reduction of load for the sake of improved reliability.

Remark The electrical characteristics of the products G: Industrial applications (T_A = -40 to +105°C) are different from those of the products “M: Industrial applications”. For details, refer to **33.1** to **33.10**.

33.1 Absolute Maximum Ratings

Absolute Maximum Ratings **(1/2)**

Parameter	Symbol	Conditions		Ratings	Unit
Supply voltage	V_{DD}			-0.5 to +6.5	V
	AV_{DD}	AV_{DD} = V_{DD}		-0.5 to +6.5	V
	AV_{SS}	AV_{SS} = V_{SS}		-0.5 to +0.3	V
REGC pin input voltage	V_{IREGC}	REGC		-0.3 to +2.8 and -0.3 to V_{DD} + 0.3 Note 1	V
REGA pin input voltage	V_{IREGA}	REGA		-0.3 to +2.8 and -0.3 to AV_{DD} + 0.3 Note 2	V
Input voltage	V_{I1}	P10 to P15, P40, P121, P122, P137, EXCLK, $\overline{RESET}$		-0.3 to V_{DD} +0.3 Note 3	V
Alternate-function pin input voltage	V_{I2}	P16, P17, P41, P42 (36-pin products only)	Digital input voltage	-0.3 to V_{DD} + 0.3 Note 3	V
			Analog input voltage	-0.3 to AV_{DD} + 0.3 Note 3	V
Analog input voltage	V_{IA}	PGA0P to PGA3P, PGA0N to PGA3N, ANI0 to ANI9, ANX0 to ANX5		-0.3 to AV_{DD} + 0.3 Note 3	V
Output voltage	V_{O1}	P10 to P15, P40		-0.3 to V_{DD} + 0.3 Note 3	V
Alternate-function pin output voltage	V_{O2}	P16, P17, P41, P42 (36-pin products only)	Digital output voltage	-0.3 to V_{DD} + 0.3 Note 3	V
			Analog output voltage	-0.3 to AV_{DD} + 0.3 Note 3	V
Analog output voltage	V_{OA}	SBIAS, AMP0O to AMP2O, ANX0 to ANX5		-0.3 to AV_{DD} + 0.3 Note 3	V

Note 1. Connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF). This value regulates the absolute maximum rating of the REGC pin. Do not use this pin with voltage applied to it.

Note 2. Connect the REGA pin to AV_{SS} via a capacitor (0.22 μF). This value regulates the absolute maximum rating of the REGA pin. Do not use this pin with voltage applied to it.

Note 3. Must be 6.5 V or lower.

Caution **Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.**

Remark 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

Remark 2. V_{SS} is used as the reference voltage.

Absolute Maximum Ratings **(2/2)**

Parameter	Symbol	Conditions		Ratings	Unit
Output current, high	I_{OH1}	Per pin	P10 to P17, P40 to P42	-40	mA
		Total of all pins	P10 to P17, P41, P42 Note	-100	mA
Analog output current, high	I_{OHA}	Per pin	AMP0O to AMP2O	-12	mA
			ANX0 to ANX5	-0.12	mA
		Total of all pins	AMP0O to AMP2O, ANX0 to ANX5	-18	mA
Output current, low	I_{OL1}	Per pin	P10 to P17, P40 to P42	40	mA
		Total of all pins	P10 to P17, P41, P42 Note	100	mA
Analog output current, low	I_{OLA}	Per pin	AMP0O to AMP2O	12	mA
			ANX0 to ANX5	0.12	mA
		Total of all pins	AMP0O to AMP2O, ANX0 to ANX5	18	mA
Operating ambient temperature	T_A	In normal operation mode		-40 to +105	°C
		In flash memory programming mode			
Storage temperature	T_{stg}			-65 to +150	°C

Note This indicates the total current value when P16, P17, P41, and P42 are used as digital input pins.

Caution **Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.**

Remark 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

Remark 2. V_{SS} is used as the reference voltage.

33.2 Oscillator Characteristics

33.2.1 X1 characteristics

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Resonator	Conditions	MIN.	TYP.	MAX.	Unit
X1 clock oscillation frequency (f_X) Note	Ceramic resonator/ crystal resonator	2.7 V ≤ V_{DD} ≤ 5.5 V	1.0		20.0	MHz
		2.4 V ≤ V_{DD} < 2.7 V	1.0		16.0	

Note Indicates only permissible oscillator frequency ranges. Refer to **AC Characteristics** for instruction execution time. Request evaluation by the manufacturer of the oscillator circuit mounted on a board to check the oscillator characteristics.

Caution Since the CPU is started by the high-speed on-chip oscillator clock after a reset release, check the X1 clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) by the user. Determine the oscillation stabilization time of the OSTC register and the oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.

Remark When using the X1 oscillator, refer to **5.4 System Clock Oscillator**.

33.2.2 On-chip oscillator characteristics

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator clock frequency Notes 1, 2	f_{IH}	2.7 V ≤ V_{DD} ≤ 5.5 V		1		32	MHz
		2.4 V ≤ V_{DD} < 2.7 V		1		16	MHz
High-speed on-chip oscillator clock frequency accuracy		-40 to +105°C	2.4 V ≤ V_{DD} ≤ 5.5 V	-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. High-speed on-chip oscillator frequency is selected with bits 0 to 3 of the option byte (000C2H) and bits 0 to 2 of the HOCODIV register.

Note 2. This only indicates the oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

33.2.3 PLL characteristics

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
PLL output frequency Notes 1, 2, 3	f_{PLL}	f_{MX} = 8 MHz	DSFRDIV = 0	DSCM = 0		48		MHz
				DSCM = 1		64		MHz
			DSFRDIV = 1	DSCM = 0		24		MHz
				DSCM = 1		32		MHz
		f_{MX} = 4 MHz	DSFRDIV = 0	DSCM = 0		24		MHz
				DSCM = 1		32		MHz
Lockup wait time		Time from when PLL output is enabled to when the phase is locked			40			μs
Interval wait time		Time from when the PLL stops operating to when the setting to start PLL operation is specified			4			μs
Setup wait time		Time required from when the PLL input clock stabilizes and the PLL setting is determined to when the PLL is activated			1			μs

Note 1. When using a PLL, input a clock of 4 MHz or 8 MHz to the PLL.

Note 2. Be sure to specify one of these settings when using a PLL.

Note 3. When using the PLL output as the CPU clock, f_{IH} is divided by 2, 4, or 8 according to the setting of the RDIV1 and RDIV0 bits.

33.3 DC Characteristics

33.3.1 Pin characteristics

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V) **(1/3)**

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high Note 1	I_{OH1}	Per pin for P10 to P17 and P40 to P42 Note 2	-40°C < T_A ≤ +85°C			-10.0 Note 3	mA
			85°C < T_A ≤ 105°C			-3.0 Note 3	mA
		Total of P10 to P17, P41, and P42 Note 2 (When duty ≤ 70% Note 4)	4.0 V ≤ V_{DD} ≤ 5.5 V, -40°C < T_A ≤ +85°C			-80.0	mA
			4.0 V ≤ V_{DD} ≤ 5.5 V, 85°C < T_A ≤ 105°C			-30.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			-19.0	mA
			2.4 V ≤ V_{DD} < 2.7 V			-10.0	mA
Output current, low Note 1	I_{OL1}	Per pin for P10 to P17 and P40 to P42 Note 2	-40°C < T_A ≤ +85°C			20.0 Note 3	mA
			85°C < T_A ≤ 105°C			8.5 Note 3	mA
		Total of P10 to P17, P41, and P42 Note 2 (When duty ≤ 70% Note 4)	4.0 V ≤ V_{DD} ≤ 5.5 V, -40°C < T_A ≤ +85°C			80.0	mA
			4.0 V ≤ V_{DD} ≤ 5.5 V, 85°C < T_A ≤ 105°C			40.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			35.0	mA
			2.4 V ≤ V_{DD} < 2.7 V			20.0	mA

Note 1. Value of current at which the device operation is guaranteed even if the current flows from the V_{DD} pin to an output pin.

Note 2. This indicates the total current value when P16, P17, P41, and P42 are used as digital I/O ports. When using these pins as analog function (AFE) pins, refer to **33.1 Absolute Maximum Ratings**.

Note 3. Do not exceed the total current value.

Note 4. Specification under conditions where the duty factor ≤ 70%.

The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).

- Total output current of pins = (I_{OH} × 0.7)/(n × 0.01)

  Example: n = 80% when I_{OH} = -10.0 mA

  Total output current of pins = (-10.0 × 0.7)/(80 × 0.01) ≈ -8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor.
A current higher than the absolute maximum rating must not flow into one pin.

Caution P10 to P15 do not output high level in N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V) **(2/3)**

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P10 to P17 and P40 to P42	Normal input buffer	0.8 V_{DD}		V_{DD}	V
	V_{IH2}	P11, P12, P14, P15	TTL input buffer, 4.0 V ≤ V_{DD} ≤ 5.5 V	2.2		V_{DD}	V
			TTL input buffer, 3.3 V ≤ V_{DD} < 4.0 V	2.0		V_{DD}	V
			TTL input buffer, 2.4 V ≤ V_{DD} < 3.3 V	1.5		V_{DD}	V
	V_{IH3}	P121, P122, P137, EXCLK, $\overline{RESET}$		0.8 V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	P10 to P17 and P40 to P42	Normal input buffer	0		0.2 V_{DD}	V
	V_{IL2}	P11, P12, P14, P15	TTL input buffer, 4.0 V ≤ V_{DD} ≤ 5.5 V	0		0.8	V
			TTL input buffer, 3.3 V ≤ V_{DD} < 4.0 V	0		0.5	V
			TTL input buffer, 2.4 V ≤ V_{DD} < 3.3 V	0		0.32	V
	V_{IL3}	P121, P122, P137, EXCLK, $\overline{RESET}$		0		0.2 V_{DD}	V
Output voltage, high	V_{OH1}	P10 to P17 and P40 to P42	4.0 V ≤ V_{DD} ≤ 5.5 V, T_A = -40 to +85°C, I_{OH1} = -10.0 mA	V_{DD} - 1.5			V
			4.0 V ≤ V_{DD} ≤ 5.5 V, 85°C < T_A ≤ 105°C, I_{OH1} = -3.0 mA	V_{DD} - 0.7			V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = -2.0 mA	V_{DD} - 0.6			V
			2.4 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = -1.5 mA	V_{DD} - 0.5			V
Output voltage, low	V_{OL1}	P10 to P17 and P40 to P42	4.0 V ≤ V_{DD} ≤ 5.5 V, T_A = -40 to +85°C, I_{OL1} = 20.0 mA			1.3	V
			4.0 V ≤ V_{DD} ≤ 5.5 V, 85°C < T_A ≤ 105°C, I_{OL1} = 8.5 mA			0.7	V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 3.0 mA			0.6	V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 1.5 mA			0.4	V
			2.4 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 0.6 mA			0.4	V

Caution **The maximum V_{IH} value on P10 to P15 is V_{DD}, even in the N-ch open-drain mode.**

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V) **(3/3)**

Item	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P10 to P17, and P40 to P42	V_I = V_{DD}				1	μA
	I_{LIH2}	P137, $\overline{RESET}$	V_I = V_{DD}				1	μA
	I_{LIH3}	P121, P122 (X1, X2, EXCLK)	V_I = V_{DD}	In input port mode or when using external clock input			1	μA
				When a resonator is connected			10	μA
Input leakage current, low	I_{LIL1}	P10 to P17, and P40 to P42	V_I = V_{SS}				-1	μA
	I_{LIL2}	P137, $\overline{RESET}$	V_I = V_{SS}				-1	μA
	I_{LIL3}	P121, P122 (X1, X2, EXCLK)	V_I = V_{SS}	In input port mode or when using external clock input			-1	μA
				When a resonator is connected			-10	μA
On-chip pull-up resistance	R_U	P10 to P15, P40	V_I = V_{SS}, in input port mode		10	20	100	kΩ

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

33.3.2 Supply current characteristics

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD1}	Operating mode Note 2	f_{HOCO} = 32 MHz, f_{MAIN} = 32 MHz Note 3	Basic operation	V_{DD} = 5.0 V		2.1		mA
					V_{DD} = 3.0 V		2.1		
			f_{HOCO} = 32 MHz, f_{MAIN} = 32 MHz Note 3	Normal operation	V_{DD} = 5.0 V		4.8	8.7	mA
					V_{DD} = 3.0 V		4.8	8.7	
			f_{HOCO} = 24 MHz, f_{MAIN} = 24 MHz Note 3	Normal operation	V_{DD} = 5.0 V		3.8	6.7	
					V_{DD} = 3.0 V		3.8	6.7	
			f_{HOCO} = 16 MHz, f_{MAIN} = 16 MHz Note 3	Normal operation	V_{DD} = 5.0 V		2.8	4.9	
					V_{DD} = 3.0 V		2.8	4.9	
			f_{MX} = 20 MHz, f_{MAIN} = 20 MHz Note 4, V_{DD} = 5.0 V	Normal operation	Square wave input		3.3	5.7	mA
					Resonator connection		3.5	5.8	
			f_{MX} = 20 MHz, f_{MAIN} = 20 MHz Note 4, V_{DD} = 3.0 V	Normal operation	Square wave input		3.3	5.7	
					Resonator connection		3.5	5.8	
			f_{MX} = 10 MHz, f_{MAIN} = 10 MHz Note 4, V_{DD} = 5.0 V	Normal operation	Square wave input		2.0	3.4	
					Resonator connection		2.1	3.5	
			f_{MX} = 10 MHz, f_{MAIN} = 10 MHz Note 4, V_{DD} = 3.0 V	Normal operation	Square wave input		2.0	3.4	
					Resonator connection		2.1	3.5	
			f_{MX} = 8 MHz, f_{MAIN} = 32 MHz Note 5, V_{DD} = 5.0 V	Normal operation	Square wave input		5.2	9.2	mA
					Resonator connection		5.3	9.3	
			f_{MX} = 8 MHz, f_{MAIN} = 32 MHz Note 5, V_{DD} = 3.0 V	Normal operation	Square wave input		5.2	9.2	
					Resonator connection		5.3	9.3	
			f_{MX} = 8 MHz, f_{MAIN} = 24 MHz Note 5, V_{DD} = 5.0 V	Normal operation	Square wave input		5.1	9.1	
					Resonator connection		5.2	9.2	
			f_{MX} = 8 MHz, f_{MAIN} = 24 MHz Note 5, V_{DD} = 3.0 V	Normal operation	Square wave input		5.1	9.1	
					Resonator connection		5.2	9.2	

Note 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The values below the MAX. column include the peripheral operation current. However, not including the current flowing into the RTC, interval timer, watchdog timer, LVD circuit, AFE, I/O ports, and on-chip pull-up/pull-down resistors and the current flowing during data flash rewrite.

Note 2. The relationship between the operation voltage range and the CPU operating frequency is as below.
2.7 V ≤ V_{DD} ≤ 5.5 V @ 1 MHz to 32 MHz
2.4 V ≤ V_{DD} ≤ 5.5 V @ 1 MHz to 16 MHz

Note 3. When the high-speed system clock is stopped

Note 4. When the high-speed on-chip oscillator and the PLL are stopped

Note 5. When the high-speed on-chip oscillator is stopped and the PLL is operating

Remark 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

Remark 2. f_{HOCO}: High-speed on-chip oscillator clock frequency

Remark 3. f_{MAIN}: Main system clock frequency

Remark 4. The temperature condition for the TYP. value is T_A = 25°C

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V) **(2/2)**

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD2} Note 2	HALT mode Note 3	f_{HOCO} = 32 MHz, f_{MAIN} = 32 MHz Note 4	V_{DD} = 5.0 V		0.54	3.67	mA
				V_{DD} = 3.0 V		0.54	3.67	
			f_{HOCO} = 24 MHz, f_{MAIN} = 24 MHz Note 4	V_{DD} = 5.0 V		0.44	2.85	
				V_{DD} = 3.0 V		0.44	2.85	
			f_{HOCO} = 16 MHz, f_{MAIN} = 16 MHz Note 4	V_{DD} = 5.0 V		0.40	2.08	
				V_{DD} = 3.0 V		0.40	2.08	
			f_{MX} = 20 MHz, f_{MAIN} = 20 MHz Note 5, V_{DD} = 5.0 V	Square wave input		0.28	2.45	mA
				Resonator connection		0.49	2.57	
			f_{MX} = 20 MHz, f_{MAIN} = 20 MHz Note 5, V_{DD} = 3.0 V	Square wave input		0.28	2.45	
				Resonator connection		0.49	2.57	
			f_{MX} = 10 MHz, f_{MAIN} = 10 MHz Note 5, V_{DD} = 5.0 V	Square wave input		0.19	1.28	
				Resonator connection		0.30	1.36	
			f_{MX} = 10 MHz, f_{MAIN} = 10 MHz Note 5, V_{DD} = 3.0 V	Square wave input		0.19	1.28	
				Resonator connection		0.30	1.36	
			f_{MX} = 8 MHz, f_{MAIN} = 32 MHz Note 6, V_{DD} = 5.0 V	Square wave input		0.91	4.17	mA
				Resonator connection		1.01	4.27	
			f_{MX} = 8 MHz, f_{MAIN} = 32 MHz Note 6, V_{DD} = 3.0 V	Square wave input		0.91	4.17	
				Resonator connection		1.01	4.27	
			f_{MX} = 8 MHz, f_{MAIN} = 24 MHz Note 6, V_{DD} = 5.0 V	Square wave input		0.76	3.27	
				Resonator connection		0.86	3.37	
			f_{MX} = 8 MHz, f_{MAIN} = 24 MHz Note 6, V_{DD} = 3.0 V	Square wave input		0.76	3.27	
				Resonator connection		0.86	3.37	
	I_{DD3} Note 7	STOP mode	T_A = -40°C			0.38	1.14	μA
			T_A = +25°C			0.50	1.14	
			T_A = +50°C			0.66	4.52	
			T_A = +70°C			1.04	7.98	
			T_A = +85°C			2.92	16.0	
			T_A = +105°C			11.0	100.0	

Note 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The values below the MAX. column include the peripheral operation current. However, not including the current flowing into the RTC, interval timer, watchdog timer, LVD circuit, AFE, I/O ports, and on-chip pull-up/pull-down resistors and the current flowing during writing to the data flash.

Note 2. During HALT instruction execution from flash memory

Note 3. The relationship between the operation voltage range and the CPU operating frequency is as below.
2.7 V ≤ V_{DD} ≤ 5.5 V @ 1 MHz to 32 MHz
2.4 V ≤ V_{DD} ≤ 5.5 V @ 1 MHz to 16 MHz

Note 4. When the high-speed system clock is stopped

Note 5. When the high-speed on-chip oscillator and the PLL are stopped

Note 6. When high-speed on-chip oscillator is stopped and the PLL is operating

Note 7. The MAX. value includes the leakage current in STOP mode.

Remark 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

Remark 2. f_{HOCO}: High-speed on-chip oscillator clock frequency

Remark 3. f_{MAIN}: Main system clock frequency

Remark 4. The temperature condition for the TYP. value is T_A = 25°C, except the operation in STOP mode.

• Peripheral functions

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Low-speed on-chip oscillator operating current	I_{FIL} Note 1				0.20		μA
RTC operating current	I_{RTC} Notes 1, 2, 3	f_{MX} = 4 MHz, RTCCL = 00H (f_{MX}/122)			22		μA
Interval timer operating current	I_{IT} Notes 1, 2, 4	f_{MX} = 4 MHz, RTCCL = 00H (f_{MX}/122)			22		μA
Watchdog timer operating current	I_{WDT} Notes 1, 5, 6	f_{IL} = 15 kHz			0.22		μA
LVD operating current	I_{LVD} Notes 1, 7				0.08		μA
Self-programming operating current	I_{FSP} Notes 1, 8				2.50	12.20	mA
BGO operating current	I_{BGO} Notes 1, 9				2.50	12.20	mA
SNOOZE operating current	I_{SNOZ} Note 1	A/D converter operation Notes 11,	The mode is performed Note 10		0.50	1.10	mA
			During A/D conversion, AV_{DD} = V_{DD} = 3.0 V		1.20	2.04	
		Simplified SPI (CSI)/UART operation			0.70	1.54	
		DTC operation			3.10		

Note 1. Current flowing to V_{DD}

Note 2. When the high-speed on-chip oscillator is stopped

Note 3. Current flowing only to the real-time clock (RTC). The supply current of the RL78 microcontrollers is the sum of the values of either I_{DD1} or I_{DD2}, and I_{RTC}, when the real-time clock is operating in operation mode or HALT mode.

Note 4. Current flowing only to the interval timer. The supply current of the RL78 microcontrollers is the sum of the values of either I_{DD1} or I_{DD2}, and I_{IT}, when the interval timer is operating in operation mode or HALT mode. When the low-speed on-chip oscillator is selected, also add I_{FIL}.

Note 5. When the high-speed on-chip oscillator and high-speed system clock are stopped.

Note 6. Current flowing only to the watchdog timer (including the operating current of the low-speed on-chip oscillator). The supply current of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{WDT} when the watchdog timer is operating.

Note 7. Current flowing only to the LVD circuit. The supply current of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{LVD} when the LVD circuit is operating.

Note 8. Current flowing during self-programming

Note 9. Current flowing during writing to the data flash

Note 10. For time required to shift to the SNOOZE mode, see **23.3.3 SNOOZE mode**.

Note 11. The current flowing into the AV_{DD} is included.

Remark 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

Remark 2. f_{IL}: Low-speed on-chip oscillator clock frequency

Remark 3. The temperature condition for the TYP. value is T_A = 25°C

• AFE functions

(T_A = -40 to +105°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
24-bit ΔΣ A/D converter operating current	I_{DSAD}	Normal mode Notes 1, 2 Circuits that operate: ABGR, REGA, SBIAS, VREFAMP, PGA, 24-bit ΔΣ A/D converter, and digital filter Differential input mode OSR = 256 SBIAS I_{OUT} = 0 mA		0.94	1.46	mA
		Low power mode Notes 1, 2 Circuits that operate: ABGR, REGA, SBIAS, VREFAMP, PGA, 24-bit ΔΣ A/D converter, and digital filter Differential input mode OSR = 256 SBIAS I_{OUT} = 0 mA		0.60	0.91	mA
10-bit A/D converter operating current	I_{ADC}	During conversion at the highest speed Notes 1, 2 AV_{DD} = 5.0 V		1.30	1.70	mA
Configurable amplifier operating current	I_{AMP}	Normal mode Notes 1, 2 Circuits that operate: ABGR and configurable amplifier I_L = 0 mA Per channel		0.13	0.24	mA
		High-speed mode Notes 1, 2 Circuits that operate: ABGR and configurable amplifier I_L = 0 mA Per channel		0.30	0.45	mA
12-bit D/A converter operating current	I_{DAC}	When AV_{DD} is selected as the reference voltage Notes 1, 2 Circuits that operate: ABGR and internal reference voltage (VREFDA)		0.61	0.97	mA

Note 1. Current flowing to AV_{DD}

Note 2. Current flowing only to the circuits that operate shown in the Conditions column.

33.4 AC Characteristics

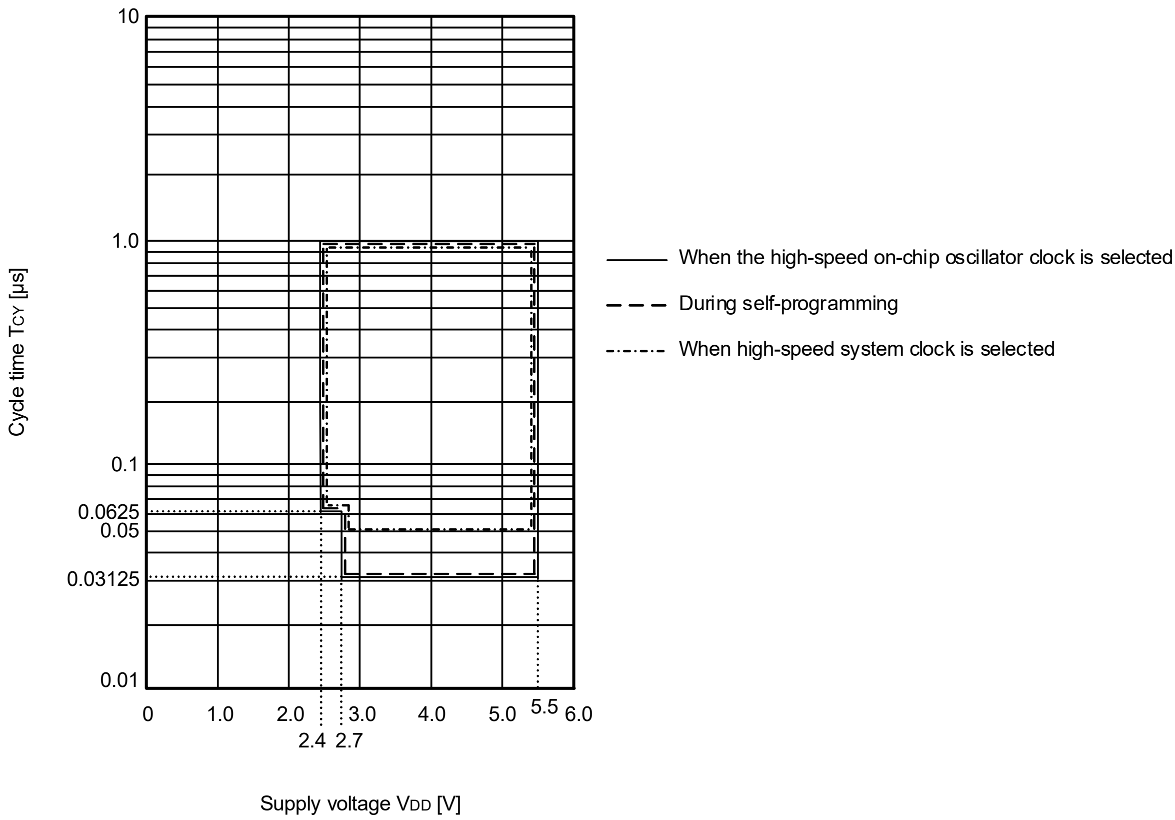
(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation	2.7 V ≤ V_{DD} ≤ 5.5 V	0.03125		1	µs
			2.4 V ≤ V_{DD} < 2.7 V	0.0625		1	µs
		In the self-programming mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.03125		1	µs
			2.4 V ≤ V_{DD} < 2.7 V	0.0625		1	µs
External system clock frequency	f_{EX}	2.7 V ≤ V_{DD} ≤ 5.5 V		1.0		20.0	MHz
		2.4 V ≤ V_{DD} < 2.7 V		1.0		16.0	MHz
External system clock input high-level width, low-level width	t_{EXH}, t_{EXL}	2.7 V ≤ V_{DD} ≤ 5.5 V		24			ns
		2.4 V ≤ V_{DD} < 2.7 V		30			ns
TI00 to TI03, TI10, TI11 input high-level width, low-level width	t_{TIH}, t_{TIL}			$1/f_{MCK}$ + 10			ns
Timer RJ input cycle	f_C	TRJIO0	2.7 V ≤ V_{DD} ≤ 5.5 V	100			ns
			2.4 V ≤ V_{DD} < 2.7 V	300			ns
Timer RJ input high-level width, low-level width	t_{TJIH}, t_{TJIL}	TRJIO0	2.7 V ≤ V_{DD} ≤ 5.5 V	40			ns
			2.4 V ≤ V_{DD} < 2.7 V	120			ns
Timer RG input high-level width, low-level width	t_{TGIH}, t_{TGIL}	TRGIOA, TRGIOB		$2.5/f_{CLK}$			ns
TO00 to TO03, TO10, TO11, TRJIO0, TRJO0, TRGIOA, TRGIOB output frequency	f_{TO}		4.0 V ≤ V_{DD} ≤ 5.5 V			16	MHz
			2.7 V ≤ V_{DD} ≤ 4.0 V			8	MHz
			2.4 V ≤ V_{DD} < 2.7 V			4	MHz
PCLBUZ0 output frequency	f_{PCL}		4.0 V ≤ V_{DD} ≤ 5.5 V			16	MHz
			2.7 V ≤ V_{DD} ≤ 4.0 V			8	MHz
			2.4 V ≤ V_{DD} < 2.7 V			4	MHz
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}	INTP1 to INTP7		1			µs
$\overline{RESET}$ low-level width	t_{RSL}			10			µs

Remark f_{MCK}: Timer array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of timer mode register mn (TMRmn). m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3))

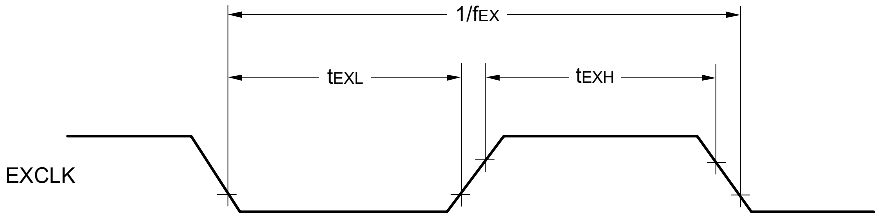
Minimum Instruction Execution Time During Main System Clock Operation

T_{CY} vs V_{DD}

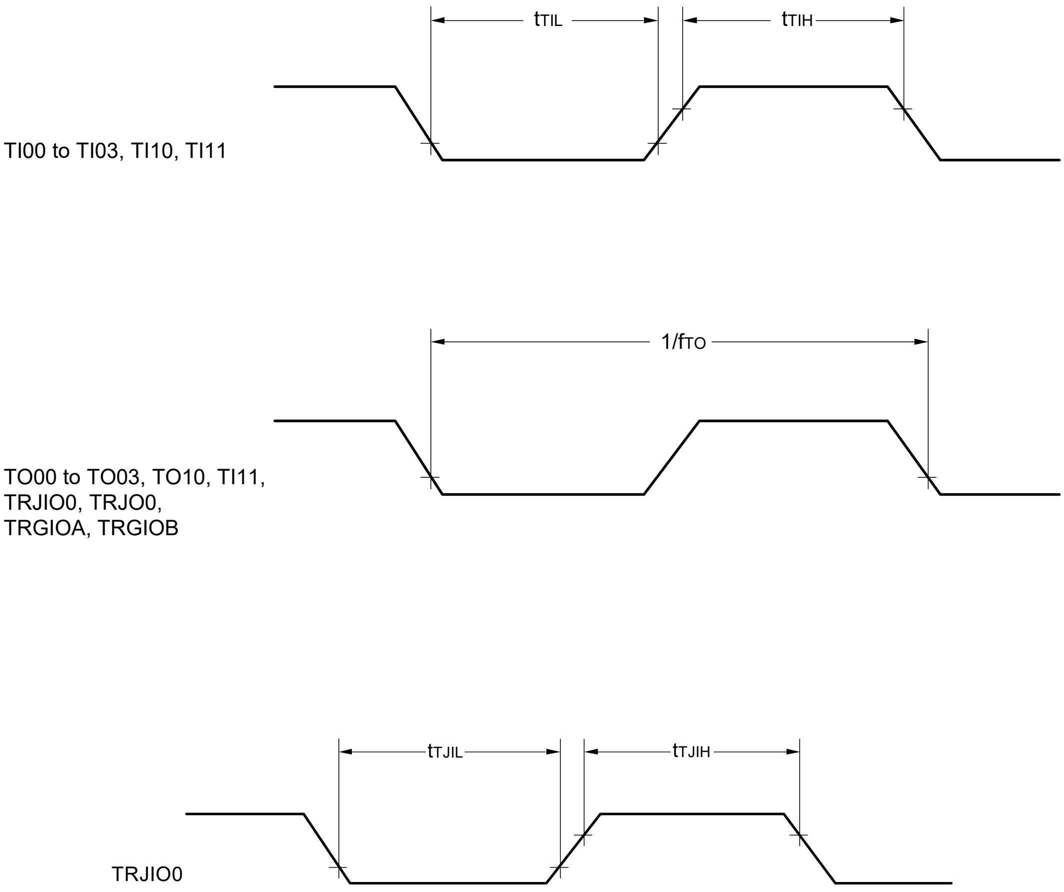


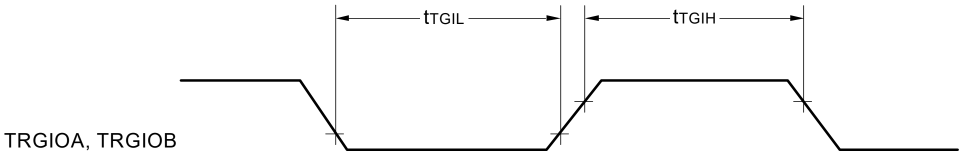
AC Timing Test Points

External System Clock Timing

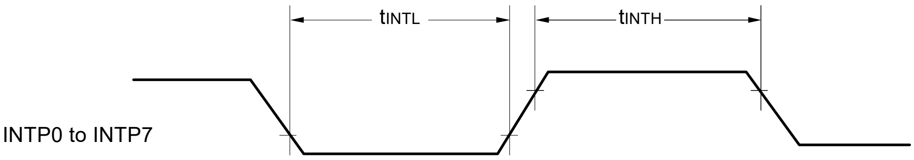


TI/TO Timing

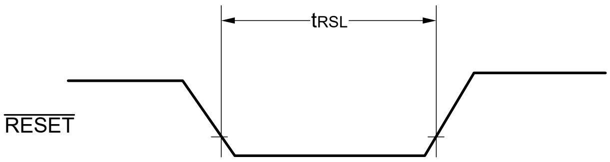




Interrupt Request Input Timing



$\overline{\text{RESET}}$ Input Timing



33.5 Peripheral Functions Characteristics

AC Timing Test Points

33.5.1 Serial array unit

(1) During communication at same potential (UART mode)

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

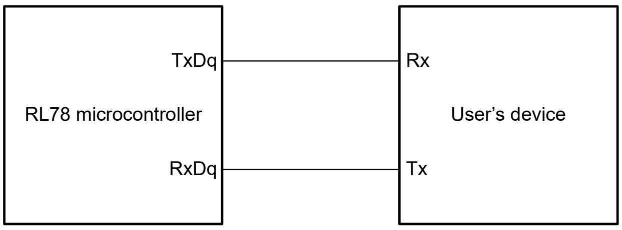
Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
Transfer rate Note 1				f_{MCK}/12	bps
		Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 2		2.6	Mbps

Note 1. Transfer rate in the SNOOZE mode is 4800 bps only.

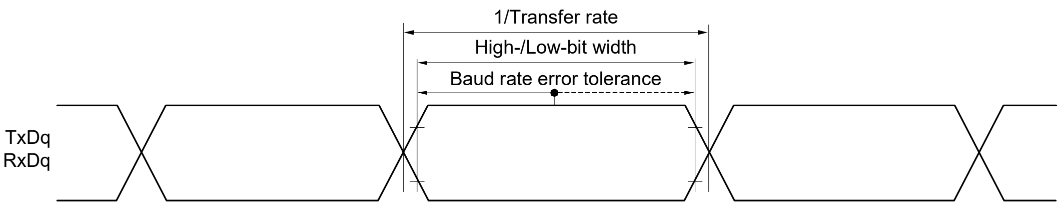
Note 2. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:
32 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)
16 MHz (2.4 V ≤ V_{DD} ≤ 5.5 V)

Caution **Select the normal input buffer for the RxDq pin and the normal output mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg).**

UART mode connection diagram (during communication at same potential)



UART mode bit width (during communication at same potential) (reference)



Remark 1. q: UART number (q = 0, 1), g: PIM or POM number (g = 1)

Remark 2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00 to 03))

(2) During communication at same potential (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) mode		Unit
				MIN.	MAX.	
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ 4/f_{CLK}	2.7 V ≤ V_{DD} ≤ 5.5 V	250		ns
			2.4 V ≤ V_{DD} ≤ 5.5 V	500		ns
SCKp high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2 - 24		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2 - 36		ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2 - 76		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V		66		ns
		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 output Note 2	t_{KSO1}	C = 30 pF Note 3			50	ns

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. C is the load capacitance of the SCKp and SOp output lines.

Caution **Select the normal input buffer for the SIp pin and the normal output mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg).**

Remark 1. p: CSI number (p = 00, 01), m: Unit number (m = 0), n: Channel number (n = 0, 1), g: PIM number (g = 1)

Remark 2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 01))

(3) During communication at same potential (Simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) mode MIN.	HS (high-speed main) mode MAX.	Unit
SCKp cycle time Note 1	t_{KCY2}	4.0 V ≤ V_{DD} ≤ 5.5 V	20 MHz < f_{MCK}	16/f_{MCK}		ns
			f_{MCK} ≤ 20 MHz	12/f_{MCK}		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V	16 MHz < f_{MCK}	16/f_{MCK}		ns
			f_{MCK} ≤ 16 MHz	12/f_{MCK}		ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		12/f_{MCK} and 1000		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ V_{DD} ≤ 5.5 V		t_{KCY2}/2 - 14		ns
		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 2	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 2	t_{KSI2}			1/f_{MCK} + 62		ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO2}	C = 30 pF Note 4	2.7 V ≤ V_{DD} ≤ 5.5 V		2/f_{MCK} + 66	ns
			2.4 V ≤ V_{DD} ≤ 5.5 V		2/f_{MCK} + 113	ns
$\overline{\text{SSI00}}$ setup time	t_{SSIK}	DAPmn = 0	2.7 V ≤ V_{DD} ≤ 5.5 V	240		ns
			2.4 V ≤ V_{DD} ≤ 5.5 V	400		ns
		DAPmn = 1	2.7 V ≤ V_{DD} ≤ 5.5 V	1/f_{MCK} + 240		ns
			2.4 V ≤ V_{DD} ≤ 5.5 V	1/f_{MCK} + 400		ns
$\overline{\text{SSI00}}$ hold time	t_{KSSI}	DAPmn = 0	2.7 V ≤ V_{DD} ≤ 5.5 V	1/f_{MCK} + 240		ns
			2.4 V ≤ V_{DD} ≤ 5.5 V	1/f_{MCK} + 400		ns
		DAPmn = 1	2.7 V ≤ V_{DD} ≤ 5.5 V	240		ns
			2.4 V ≤ V_{DD} ≤ 5.5 V	400		ns

Note 1. The maximum transfer rate in the SNOOZE mode is 1 Mbps.

Note 2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

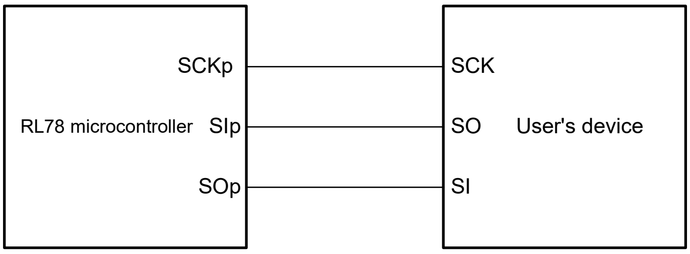
Note 4. C is the load capacitance of the SOp output lines.

Caution **Select the normal input buffer for the SIp and SCKp pins and the normal output mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg).**

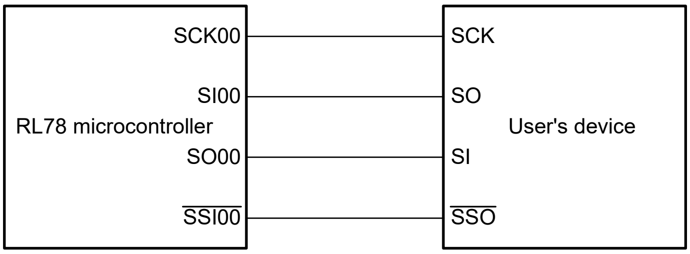
Remark 1. p: CSI number (p = 00, 01), m: Unit number (m = 0), n: Channel number (n = 0, 1), g: PIM number (g = 1)

Remark 2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number,
n: Channel number (mn = 00, 01))

Simplified SPI (CSI) mode connection diagram (during communication at same potential)

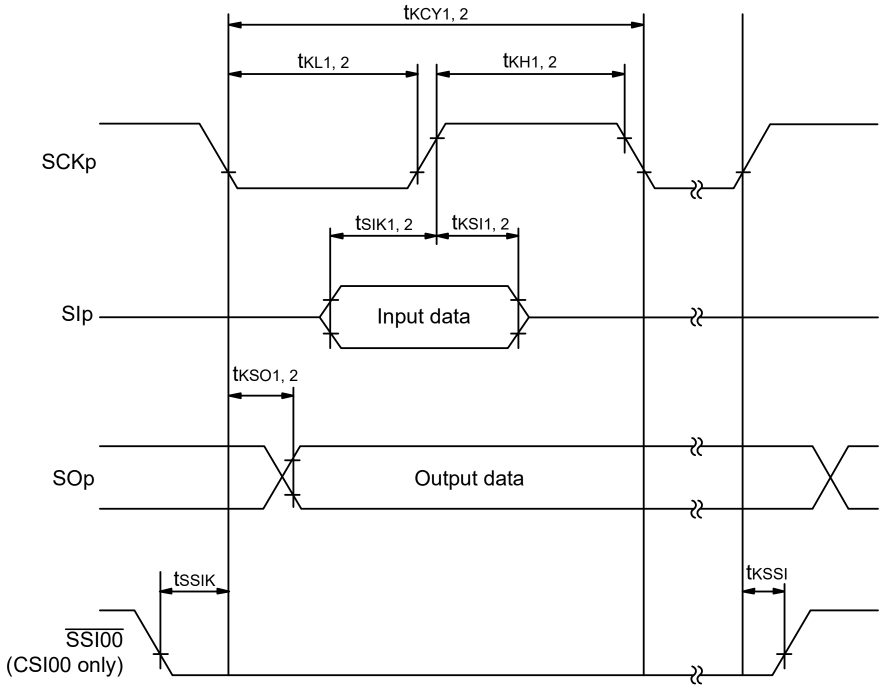


Simplified SPI (CSI) mode connection diagram (during communication at same potential)
(Slave Transmission of slave select input function (CSI00))

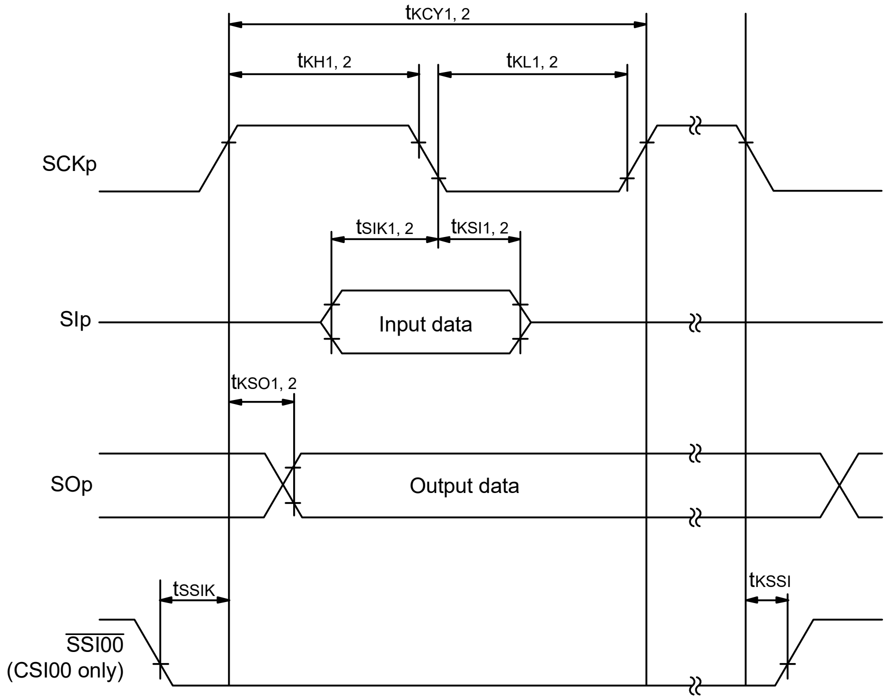


Remark 1. p: CSI number (p = 00, 01)
Remark 2. m: Unit number, n: Channel number (mn = 00, 01)

Simplified SPI (CSI) mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remark 1. p: CSI number (p = 00, 01)
Remark 2. m: Unit number, n: Channel number (mn = 00, 01)

(4) During communication at same potential (simplified I²C mode)

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) mode		Unit
			MIN.	MAX.	
SCLr clock frequency	f_{SCL}	2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 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 = 50 pF, R_b = 2.7 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 = 50 pF, R_b = 2.7 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 = 50 pF, R_b = 2.7 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 = 50 pF, R_b = 2.7 kΩ	0	770	ns
		2.4 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	0	1420	ns

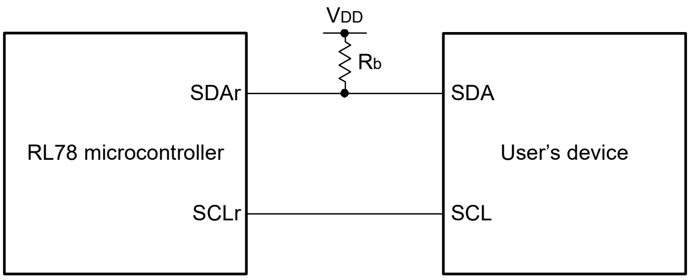
Note 1. The value must also be equal to or less than f_{MCK}/4.

Note 2. Set the f_{MCK} value to keep the hold time of SCLr = "L" and SCLr = "H".

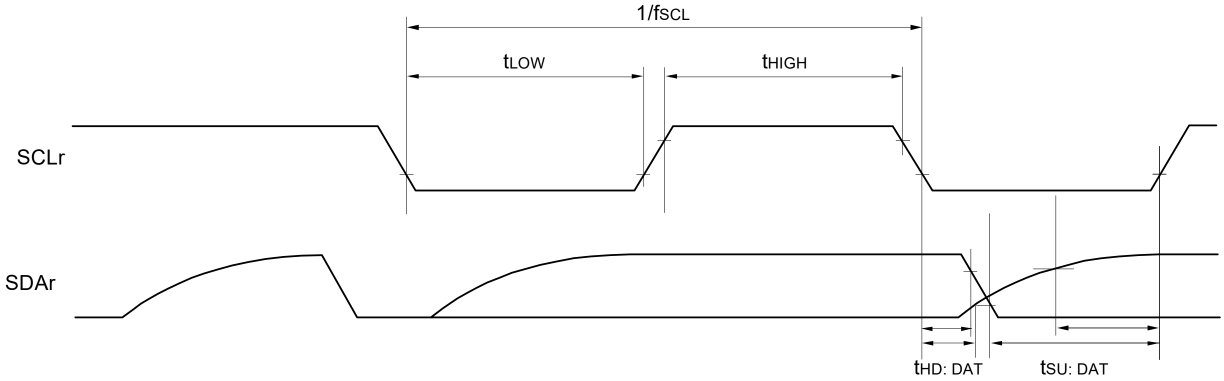
Caution Select the normal input buffer and the N-ch open drain output (V_{DD} tolerance) mode for the SDAr pin and the normal output mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register h (POMh).

(Remarks are listed on the next page.)

Simplified I2C mode connection diagram (during communication at same potential)



Simplified I2C mode serial transfer timing (during communication at same potential)



Remark 1. R_b [Ω]: Communication line (SDAr) pull-up resistance, C_b [F]: Communication line (SDAr, SCLr) load capacitance

Remark 2. r: IIC number (r = 00, 01), g: PIM number (g = 1), h: POM number (h = 1)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number (m = 0),
n: Channel number (n = 0, 1), mn = 00, 01)

(5) Communication at different potential (1.8 V, 2.5 V, 3 V) (UART mode)

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions		HS (high-speed main) mode MIN.	HS (high-speed main) mode MAX.	Unit
Transfer rate		Reception	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		$f_{MCK}/12$ Note 1	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 2		2.6	Mbps
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		$f_{MCK}/12$ Note 1	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 2		2.6	Mbps
			2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V		$f_{MCK}/12$ Note 1	bps
			Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 2		2.6	Mbps

Note 1. Transfer rate in the SNOOZE mode is 4800 bps only.

Note 2. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:
32 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)
16 MHz (2.4 V ≤ V_{DD} ≤ 5.5 V)

Caution **Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.**

Remark 1. V_b [V]: Communication line voltage

Remark 2. q: UART number (q = 0, 1), g: PIM or POM number (g = 1)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 01)

(5) Communication at different potential (1.8 V, 2.5 V, 3 V) (UART mode)

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V) (2/2)

Parameter	Symbol	Conditions			HS (high-speed main) mode MIN.	HS (high-speed main) mode MAX.	Unit
Transfer rate		Transmission	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V			Note 1	bps
				Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 1.4 kΩ, V_b = 2.7 V		2.6 Note 2	Mbps
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V			Note 3	bps
				Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 2.7 kΩ, V_b = 2.3 V		1.2 Note 4	Mbps
			2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V			Note 5	bps
				Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 5.5 kΩ, V_b = 1.6 V		0.43 Note 6	Mbps

Note 1. The smaller maximum transfer rate derived by using $f_{MCK}/12$ or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 4.0 V ≤ V_{DD} ≤ 5.5 V and 2.7 V ≤ V_b ≤ 4.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.2}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

Note 2. This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to **Note 1** above to calculate the maximum transfer rate under conditions of the customer.

Note 3. The smaller maximum transfer rate derived by using $f_{MCK}/12$ or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 2.7 V ≤ V_{DD} < 4.0 V and 2.3 V ≤ V_b ≤ 2.7 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1 - \frac{2.0}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

Note 4. This value as an example is calculated when the conditions described in the “Conditions” column are met. Refer to **Note 3** above to calculate the maximum transfer rate under conditions of the customer.

Note 5. The smaller maximum transfer rate derived by using $f_{MCK}/12$ or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 2.4 V ≤ V_{DD} < 3.3 V and 1.6 V ≤ V_b ≤ 2.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln\left(1 - \frac{1.5}{V_b}\right)\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln\left(1 - \frac{1.5}{V_b}\right)\}}{\left(\frac{1}{\text{Transfer rate}}\right) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

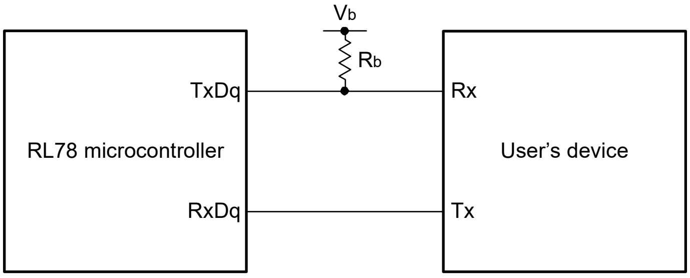
* This value is the theoretical value of the relative difference between the transmission and reception sides.

Note 6. This value as an example is calculated when the conditions described in the “Conditions” column are met.
Refer to **Note 5** above to calculate the maximum transfer rate under conditions of the customer.

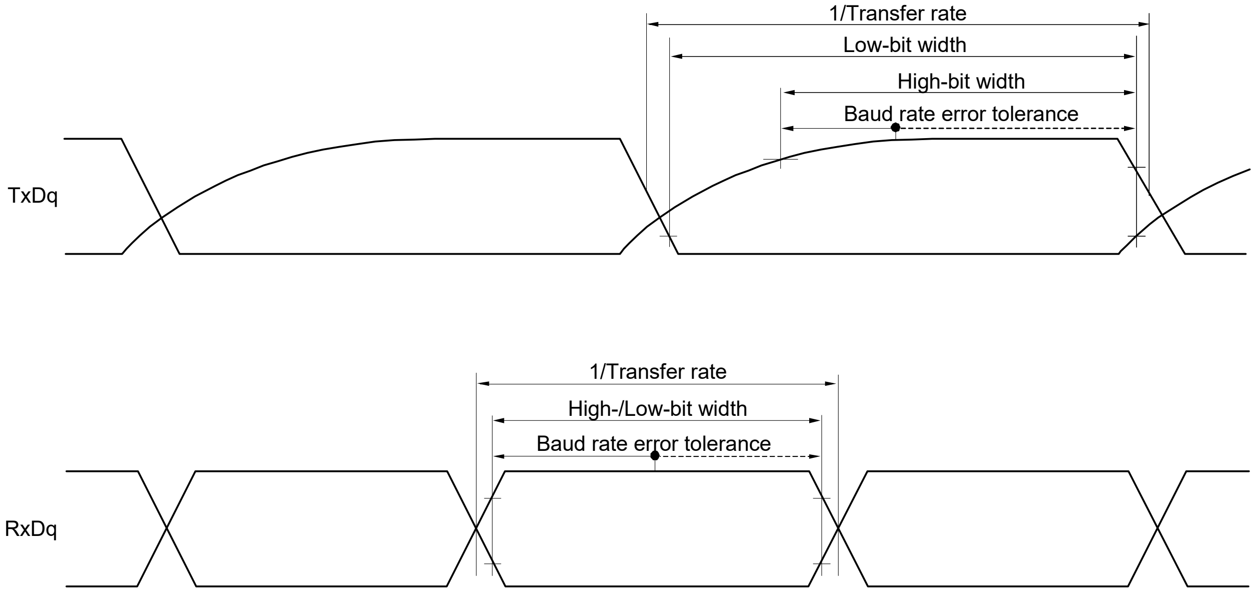
Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

(Remarks are listed on the next page.)

UART mode connection diagram (during communication at different potential)



UART mode bit width (during communication at different potential) (reference)



Remark 1. R_b [Ω]: Communication line (TxDq) pull-up resistance,
C_b [F]: Communication line (TxDq) load capacitance, V_b [V]: Communication line voltage

Remark 2. q: UART number (q = 0, 1), g: PIM or POM number (g = 1)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00, 01))

(6) Communication at different potential (1.8 V, 2.5 V, 3 V) (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V) **(1/3)**

Parameter	Symbol	Conditions		HS (high-speed main) mode		Unit
				MIN.	MAX.	
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ 4/f_{CLK}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	600		ns
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	1000		ns
			2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	2300		ns
SCKp high-level width	t_{KH1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KCY1}/2 - 150		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 - 340		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		t_{KCY1}/2 - 916		ns
SCKp low-level width	t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KCY1}/2 - 24		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 - 36		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		t_{KCY1}/2 - 100		ns

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

(Remarks are listed two pages after the next page.)

(6) Communication at different potential (1.8 V, 2.5 V, 3 V) (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V) **(2/3)**

Parameter	Symbol	Conditions	HS (high-speed main) mode		Unit
			MIN.	MAX.	
SIp setup time (to SCKp↑) Note	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	162		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	354		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	958		ns
SIp hold time (from SCKp↑) Note	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	38		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	38		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	38		ns
Delay time from SCKp↓ to SOp output Note	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		200	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		390	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		966	ns

Note When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

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

(6) Communication at different potential (1.8 V, 2.5 V, 3 V) (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V) (3/3)

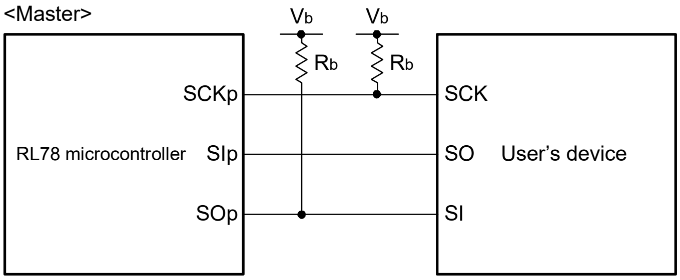
Parameter	Symbol	Conditions	HS (high-speed main) mode		Unit
			MIN.	MAX.	
SIp setup time (to SCKp↓) Note	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	88		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	88		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	220		ns
SIp hold time (from SCKp↓) Note	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	38		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	38		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	38		ns
Delay time from SCKp↑ to SOp output Note	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		50	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		50	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		50	ns

Note When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

(Remarks are listed on the next page.)

Simplified SPI (CSI) mode connection diagram (during communication at different potential

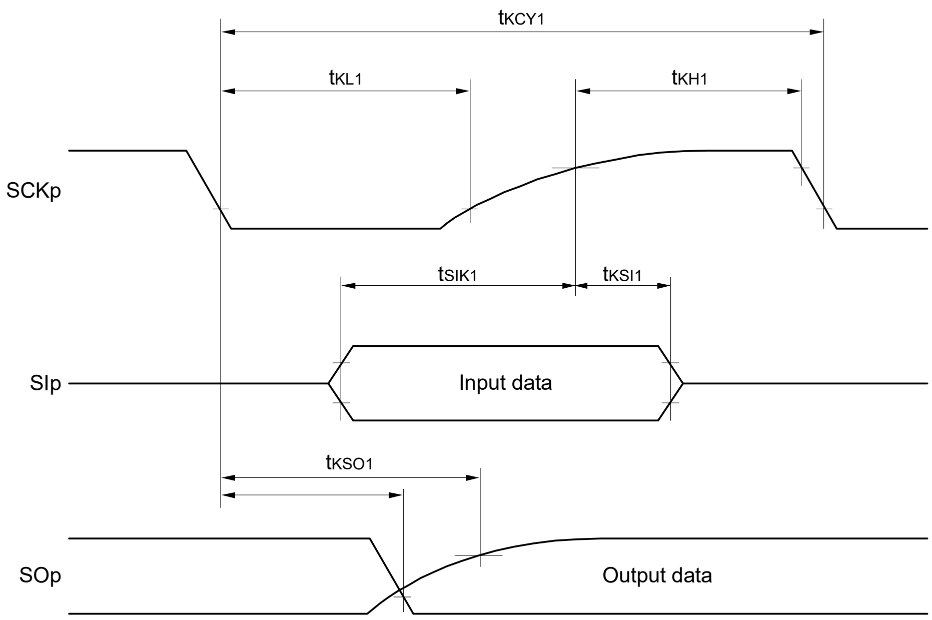


Remark 1. R_b [Ω]: Communication line (SCKp, SOp) pull-up resistance, C_b [F]: Communication line (SCKp, SOp) load capacitance, V_b [V]: Communication line voltage

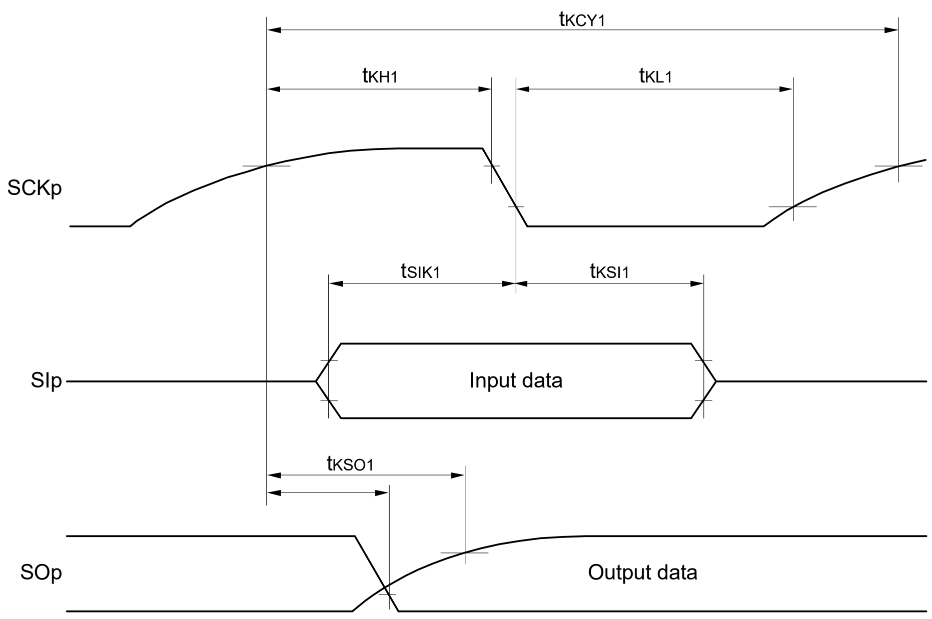
Remark 2. p: CSI number (p = 00, 01), m: Unit number (m = 0), n: Channel number (n = 0, 1), g: PIM or POM number (g = 1)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number,
n: Channel number (mn = 00, 01))

Simplified SPI (CSI) mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (master mode) (during communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remark p: CSI number (p = 00, 01), m: Unit number (m = 0), n: Channel number (n = 0, 1), g: PIM or POM number (g = 1)

(7) Communication at different potential (1.8 V, 2.5 V, 3 V) (Simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) mode MIN.	HS (high-speed main) mode MAX.	Unit
SCKp cycle time Note 1	t_{KCY2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V	24 MHz < f_{MCK}	28/f_{MCK}		ns
			20 MHz < f_{MCK} ≤ 24 MHz	24/f_{MCK}		ns
			8 MHz < f_{MCK} ≤ 20 MHz	20/f_{MCK}		ns
			4 MHz < f_{MCK} ≤ 8 MHz	16/f_{MCK}		ns
			f_{MCK} ≤ 4 MHz	12/f_{MCK}		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V	24 MHz < f_{MCK}	40/f_{MCK}		ns
			20 MHz < f_{MCK} ≤ 24 MHz	32/f_{MCK}		ns
			16 MHz < f_{MCK} ≤ 20 MHz	28/f_{MCK}		ns
			8 MHz < f_{MCK} ≤ 16 MHz	24/f_{MCK}		ns
			4 MHz < f_{MCK} ≤ 8 MHz	16/f_{MCK}		ns
			f_{MCK} ≤ 4 MHz	12/f_{MCK}		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V	24 MHz < f_{MCK}	96/f_{MCK}		ns
			20 MHz < f_{MCK} ≤ 24 MHz	72/f_{MCK}		ns
			16 MHz < f_{MCK} ≤ 20 MHz	64/f_{MCK}		ns
			8 MHz < f_{MCK} ≤ 16 MHz	52/f_{MCK}		ns
			4 MHz < f_{MCK} ≤ 8 MHz	32/f_{MCK}		ns
			f_{MCK} ≤ 4 MHz	20/f_{MCK}		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		t_{KCY2}/2 - 24		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		t_{KCY2}/2 - 36		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V		t_{KCY2}/2 - 100		ns
SIp setup time (to SCKp↑) Note 2	t_{SIK2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		1/f_{MCK} + 40		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		1/f_{MCK} + 40		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V		1/f_{MCK} + 60		ns
SIp hold time (from SCKp↑) Note 2	t_{KSI2}			1/f_{MCK} + 62		ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ			2/f_{MCK} + 240	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			2/f_{MCK} + 428	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ			2/f_{MCK} + 1146	ns

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

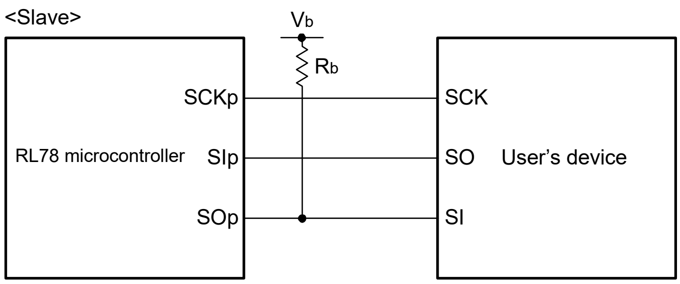
Note 1. Transfer rate in the SNOOZE mode: MAX. 1 Mbps

Note 2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Caution Select the TTL input buffer for the SIp and SCKp pins, and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Simplified SPI (CSI) mode connection diagram (during communication at different potential)



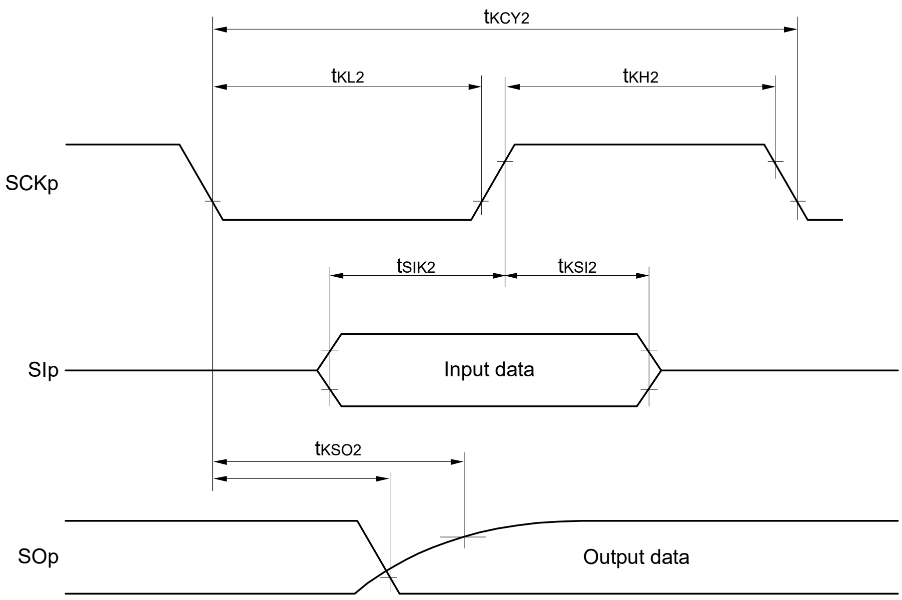
Remark 1. R_b [Ω]: Communication line (SOp) pull-up resistance, C_b [F]: Communication line (SOp) load capacitance, V_b [V]: Communication line voltage

Remark 2. p: CSI number (p = 00, 01), m: Unit number (m = 0), n: Channel number (n = 0 , 1), g: PIM or POM number (g = 1)

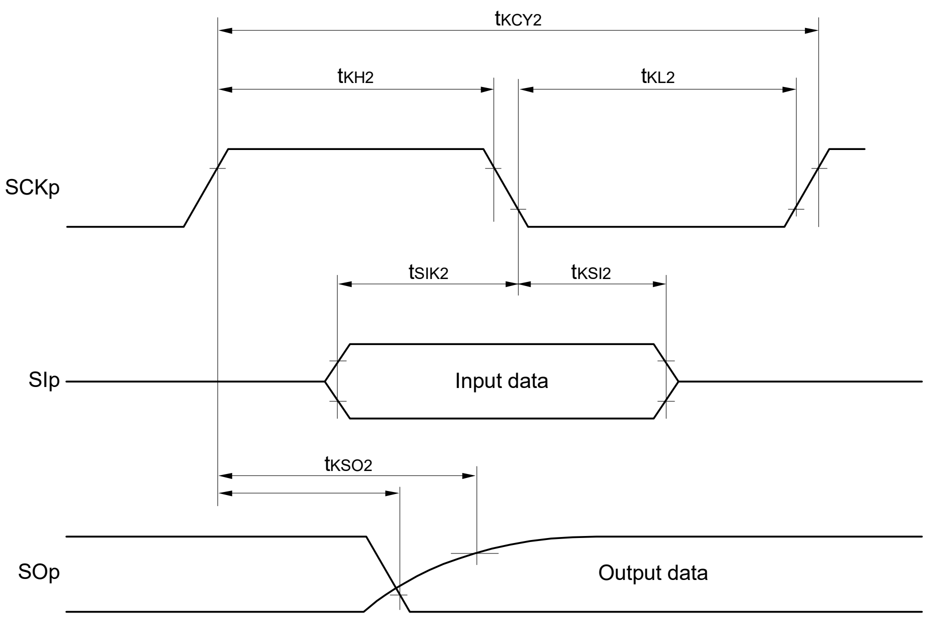
Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00, 01))

Remark 4. Communication at different potential cannot be performed during clocked serial communication with the slave select function.

Simplified SPI (CSI) mode serial transfer timing (slave mode) (during communication at different potential) (When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (slave mode) (during communication at different potential) (When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remark 1. p: CSI number (p = 00, 01), m: Unit number (m = 0), n: Channel number (n = 0, 1), g: PIM or POM number (g = 1)

Remark 2. Communication at different potential cannot be performed during clocked serial communication with the slave select function.

(8) Communication at different potential (1.8 V, 2.5 V, 3 V) (simplified I²C mode)

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions	HS (high-speed main) mode		Unit
			MIN.	MAX.	
SCLr clock frequency	f_{SCL}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ		400 Note 1	kHz
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ		400 Note 1	kHz
		4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ		100 Note 1	kHz
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ		100 Note 1	kHz
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 100 pF, R_b = 5.5 kΩ		100 Note 1	kHz
Hold time when SCLr = "L"	t_{LOW}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	1200		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	1200		ns
		4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	4600		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	4600		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 100 pF, R_b = 5.5 kΩ	4600		ns
Hold time when SCLr = "H"	t_{HIGH}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	620		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	500		ns
		4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	2700		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	2400		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 100 pF, R_b = 5.5 kΩ	1830		ns

(8) Communication at different potential (1.8 V, 2.5 V, 3 V) (simplified I²C mode)

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V) **(2/2)**

Parameter	Symbol	Conditions	HS (high-speed main) mode		Unit
			MIN.	MAX.	
Data setup time (reception)	$t_{SU:DAT}$	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	$1/f_{MCK}$ + 340 Note 2		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	$1/f_{MCK}$ + 340 Note 2		ns
		4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	$1/f_{MCK}$ + 760 Note 2		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	$1/f_{MCK}$ + 760 Note 2		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 100 pF, R_b = 5.5 kΩ	$1/f_{MCK}$ + 570 Note 2		ns
Data hold time (transmission)	$t_{HD:DAT}$	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	0	770	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	0	770	ns
		4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	0	1420	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	0	1420	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 100 pF, R_b = 5.5 kΩ	0	1215	ns

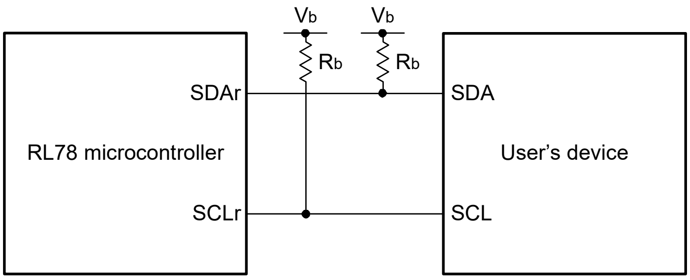
Note 1. The value must also be equal to or less than $f_{MCK}/4$.

Note 2. Set the f_{MCK} value to keep the hold time of SCLr = "L" and SCLr = "H".

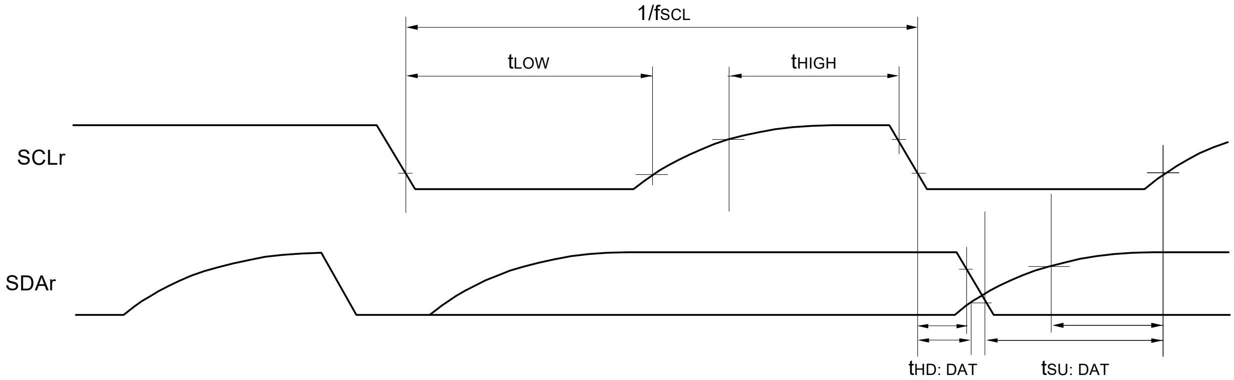
Caution **Select the TTL input buffer and the N-ch open drain output (V_{DD} tolerance) mode for the SDAr pin and the N-ch open drain output (V_{DD} tolerance) mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.**

(Remarks are listed on the next page.)

Simplified I²C mode connection diagram (during communication at different potential)



Simplified I²C mode serial transfer timing (during communication at different potential)



Remark 1. R_b [Ω]: Communication line (SDAr, SCLr) pull-up resistance, C_b [F]: Communication line (SDAr, SCLr) load capacitance, V_b [V]: Communication line voltage

Remark 2. r: IIC number (r = 00, 01), g: PIM, POM number (g = 1)

Remark 3. f_{MCK}: Serial array unit operation clock frequency (Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number (m = 0), n: Channel number (n = 0), mn = 00, 01)

33.6 Analog Characteristics

33.6.1 Programmable gain instrumentation amplifier and 24-bit ΔΣ A/D converter

(1) Analog input in differential input mode

(T_A = -40 to +105°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V, normal mode: fs1 = 1 MHz, FDATA1 = 3.90625 ksps, low-power mode: fs2 = 0.125 MHz, FDATA2 = 488.28125 sps, SBIAS = 1.2 V, d_{OFR} = 0 mV, V_{COM} = 1.0 V, external clock input used)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Full-scale differential input voltage range	V_{ID}	V_{ID} = (PGAxP - PGAxN) (x = 0 to 3)		± 800 /G_{TOTAL}		mV
Input voltage range	V_I	Each of PGAxP and PGAxN pins (x = 0 to 3)	0.2		1.8	V
Common mode input voltage	V_{COM}	d_{OFR} = 0 mV	0.2+(|V_{ID}|x G_{SET1})/2		1.8-(|V_{ID}|x G_{SET1})/2	V
Input bias current	I_{IN}	V_I = 1.0 V			±50	nA
Input offset current	I_{INOFR}	V_I = 1.0 V			±20	nA

(2) Analog input in single-ended input mode

(T_A = -40 to +105°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V, normal mode: fs1 = 1 MHz, FDATA1 = 3.90625 ksps, low-power mode: fs2 = 0.125 MHz, FDATA2 = 488.28125 sps, SBIAS = 1.2 V, d_{OFR} = 0 mV, V_{COM} = 1.0 V, external clock input used)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage range	V_I	Each of PGAxP and PGAxN pins (x = 0 to 3) G_{SET1} = 1, G_{SET2} = 1	0.2		1.8	V
Input bias current	I_{IN}	V_I = 1.0 V			±50	nA

(3) Programmable gain instrumentation amplifier and 24-bit ΔΣ A/D converter

(T_A = -40 to +105°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V, normal mode: fs1 = 1 MHz, FDATA1 = 3.90625 ksps, low-power mode: fs2 = 0.125 MHz, FDATA2 = 488.28125 sps, SBIAS = 1.2 V, d_{OFR} = 0 mV, V_{COM} = 1.0 V, external clock input used, in differential input mode) **(1/2)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	RES				24	bit
Sampling frequency	f_{S1}	Normal mode		1		MHz
	f_{S2}	Low power mode		0.125		MHz
Output data rate	f_{DATA1}	Normal mode	0.48828		15.625	ksps
	f_{DATA2}	Low power mode	61.03615		1953.125	sps
Gain setting range	G_{TOTAL}	G_{TOTAL} = G_{SET1} × G_{SET2}	1		64	V/V
1st gain setting range	G_{SET1}	In differential input mode only		1, 2, 3, 4, 8		V/V
2nd gain setting range	G_{SET2}	In differential input mode only		1, 2, 4, 8		V/V
Offset adjustment bit range	d_{OFFB}			5		bit
Offset adjustment range	d_{OFR}	Referred to input	-164/G_{SET1}		+164/G_{SET1}	mV
Offset adjustment steps	d_{OFS}	Referred to input		11/G_{SET1}		mV

(T_A = -40 to +105°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V, normal mode: fs1 = 1 MHz, FDATA1 = 3.90625 ksps, low-power mode: fs2 = 0.125 MHz, FDATA2 = 488.28125 sps, SBIAS = 1.2 V, d_{OFR} = 0 mV, V_{COM} = 1.0 V, external clock input used, in differential input mode) (2/2)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Gain error	E_G	T_A = 25°C G_{SET1} = 1, G_{SET2} = 1 Excluding SBIAS error		±0.2	±2.7	%
		T_A = 25°C G_{SET1} = 8, G_{SET2} = 4 Excluding SBIAS error		±0.1		%
Gain drift Note	dE_G	G_{SET1} = 1, G_{SET2} = 1 Excluding SBIAS drift		(5.6)	(22.0)	ppm/°C
		G_{SET1} = 8, G_{SET2} = 4 Excluding SBIAS drift		(9.1)		ppm/°C
Offset error	E_{OS}	T_A = 25°C G_{SET1} = 1, G_{SET2} = 1 Referred to input		±0.32	±2.90	mV
		T_A = 25°C G_{SET1} = 8, G_{SET2} = 4 Referred to input		±0.03		mV
Offset drift Note	dE_{OS}	G_{SET1} = 1, G_{SET2} = 1 Referred to input		(±0.02)	(±6.00)	μV/°C
		G_{SET1} = 8, G_{SET2} = 4 Referred to input		(±0.02)		μV/°C
SND ratio	SNDR	G_{SET1} = 1, G_{SET2} = 1, f_{IN} = 50 Hz Normal mode, pin = -1 dBFS	(82)	(85)		dB
		G_{SET1} = 8, G_{SET2} = 4, f_{IN} = 50 Hz Normal mode, pin = -1 dBFS	(73)	(80)		dB
Noise	Vn	G_{SET1} = 1, G_{SET2} = 1, OSR = 2048		(13)		μVRms
		G_{SET1} = 8, G_{SET2} = 4, OSR = 2048		(0.6)		μVRms
Integral non-linearity error	INL	G_{SET1} = 1, G_{SET2} = 1, OSR = 2048		(±10)		ppmFS
Common mode rejection ratio	CMRR	V_{COM} = 1.0±0.8 V, fin = 50 Hz G_{SET1} = 1, G_{SET2} = 1 Differential input mode	(72)	(90)		dB
Power supply rejection ratio	PSRR	AV_{DD} = 2.7 to 5.5 V G_{SET1} = 1, G_{SET2} = 1 Differential input mode		(85)		dB
ΔΣ A/D converter input clock frequency	f_{ADC}		3.8	4	4.2	MHz

Note Calculate the gain drift and offset drift by using the following expression (for 105°C products):
For gain drift: (MAX(E_G($T_{(-40)}$ to $T_{(105)}$)) - MIN(E_G($T_{(-40)}$ to $T_{(105)}$))) / (105°C -(-40°C))
For offset drift: (MAX(E_{OS}($T_{(-40)}$ to $T_{(105)}$)) - MIN(E_{OS}($T_{(-40)}$ to $T_{(105)}$))) / (105°C -(-40°C))
MAX(E_G($T_{(-40)}$ to $T_{(105)}$)): The maximum value of gain error when the temperature range is -40°C to 105°C
MIN(E_G($T_{(-40)}$ to $T_{(105)}$)): The minimum value of gain error when the temperature range is -40°C to 105°C
MAX(E_{OS}($T_{(-40)}$ to $T_{(105)}$)): The maximum value of offset error when the temperature range is -40°C to 105°C
MIN(E_{OS}($T_{(-40)}$ to $T_{(105)}$)):The minimum value of offset error when the temperature range is -40°C to 105°C

Remark In the ratings column, values in parentheses are the target design values and therefore are not tested for shipment.

33.6.2 Sensor power supply (SBIAS)

(T_A = -40 to +105°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V, C_{OUT} = 0.22 μF, V_{OUT} = 1.0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Output voltage range	V_{OUT}		0.5		2.2	V
Output voltage setting steps	V_{STEP}			0.1		V
Output voltage precision	V_A	I_{OUT} = 1 mA	(-3)		(+3)	%
Maximum output current	I_{OUT}		5			mA
Short circuit current	I_{SHORT}	V_{OUT} = 0 V		40	65	mA
Load regulation	L_R	1 mA ≤ I_{OUT} ≤ 5 mA			(15)	mV
Power supply rejection ratio	PSRR	AV_{DD} = 5.0 V + 0.1 $V_{pp\ ripple}$ f = 100 Hz, I_{OUT} = 2.5 mA	(45)	(50)		dB

Remark In the ratings column, values in parentheses are the target design values and therefore are not tested for shipment.

33.6.3 Temperature sensor

(T_A = -40 to +105°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Temperature coefficient for sensor	TC_{SNS}			(756)		μV/°C
Sensor output voltage	V_{TEMP}	T_A = 25°C		226.4		mV

Remark In the ratings column, values in parentheses are the target design values and therefore are not tested for shipment.

33.6.4 A/D converter characteristics

(1) When positive reference voltage (+) = AV_{DD} (ADREFP1 = 0, ADREFP0 = 0), negative reference voltage (-) = AV_{SS} (ADREFM = 0), pins subject to A/D conversion: ANI0 to ANI9 and SBIAS

(T_A = -40 to +105°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V, positive reference voltage (+) = AV_{DD}, negative reference voltage (-) = AV_{SS})

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error Note 1	AINL	10-bit resolution ANI0 to ANI9, SBIAS	4.0 V ≤ AV_{DD} ≤ 5.5 V		1.2	±6.5	LSB
			2.7 V ≤ AV_{DD} ≤ 5.5 V		1.2	±7.0	LSB
Conversion time	t_{CONV}	10-bit resolution	4.0 V ≤ AV_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ AV_{DD} ≤ 5.5 V	3.1875		39	μs
Zero-scale error Notes 1, 2	E_{ZS}	10-bit resolution ANI0 to ANI9, SBIAS	4.0 V ≤ AV_{DD} ≤ 5.5 V			±0.50	%FSR
			2.7 V ≤ AV_{DD} ≤ 5.5 V			±0.60	%FSR
Full-scale error Notes 1, 2	E_{FS}	10-bit resolution ANI0 to ANI9, SBIAS	4.0 V ≤ AV_{DD} ≤ 5.5 V			±0.50	%FSR
			2.7 V ≤ AV_{DD} ≤ 5.5 V			±0.60	%FSR
Integral linearity error Note 1	ILE	10-bit resolution ANI0 to ANI9, SBIAS	4.0 V ≤ AV_{DD} ≤ 5.5 V			±3.5	LSB
			2.7 V ≤ AV_{DD} ≤ 5.5 V			±4.0	LSB
Differential linearity error Note 1	DLE	10-bit resolution	2.7 V ≤ AV_{DD} ≤ 5.5 V			±2.0	LSB
Analog input voltage	V_{AIN}	ANI0 to ANI9		AV_{SS}		AV_{DD}	V

Note 1. Excludes quantization error (±1/2 LSB).
Note 2. This value is indicated as a ratio (%FSR) to the full-scale value.

Caution The number of pins depends on the product. For details, see a list of pin functions.

(2) When positive reference voltage (+) = Internal reference voltage (ADREFP1 = 1, ADREFP0 = 0), negative reference voltage (-) = AV_{SS} (ADREFM = 0), pins subject to A/D conversion: ANI0 to ANI9 and SBIAS

(T_A = -40 to +105°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V, positive reference voltage (+) = V_{BGR}, negative reference voltage (-) = AV_{SS})

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES				8		bit
Conversion time	t_{CONV}	8-bit resolution	2.7 V ≤ AV_{DD} ≤ 5.5 V	17		39	μs
Zero-scale error Notes 1, 2	E_{ZS}	8-bit resolution	2.7 V ≤ AV_{DD} ≤ 5.5 V			±0.60	%FSR
Integral linearity error Note 1	ILE	8-bit resolution	2.7 V ≤ AV_{DD} ≤ 5.5 V			±2.0	LSB
Differential linearity error Note 1	DLE	8-bit resolution	2.7 V ≤ AV_{DD} ≤ 5.5 V			±1.0	LSB
Internal reference voltage (+)	V_{BGR}	2.7 V ≤ AV_{DD} ≤ 5.5 V			V_{BGR} Note 3		V
Analog input voltage	V_{AIN}	ANI0 to ANI9		0		V_{BGR}	V

Note 1. Excludes quantization error (±1/2 LSB).
Note 2. This value is indicated as a ratio (%FSR) to the full-scale value.
Note 3. See the Internal reference voltage characteristics.

33.6.5 12-bit D/A converter

(1) When positive reference voltage (+) = AV_{DD} (DACVRF = 0)

(T_A = -40 to +105°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V, positive reference voltage (+) = AV_{DD})

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	DARES				(12)	bit
Output voltage range	DAOUT	12-bit resolution	0.35		AV_{DD}-0.47	V
Integral non-linearity error	DAILE	12-bit resolution			±4.0	LSB
Differential non-linearity error	DADLE	12-bit resolution			±1.0	LSB
Offset error	DAErr	12-bit resolution			±30	mV
Gain error	DAEG	12-bit resolution			±20	mV
Settling time	DAtset	12-bit resolution, C_L = 50 pF, R_L = 10 kΩ			(60)	μs

Remark 1. In the ratings column, values in parentheses are the target design values and therefore are not tested for shipment.
Remark 2. The 12-bit D/A converter characteristics are the values obtained with the configurable amplifier connected.

(2) When positive reference voltage (+) = internal reference voltage (DACVRF = 1)

(T_A = -40 to +105°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V, positive reference voltage (+) = VREFDA)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	DARES				(8)	bit
Internal reference voltage	VREFDA	8-bit resolution	1.34	1.45	1.54	V
Output voltage range	DAOUT	8-bit resolution	0.35		VREFDA	V
Integral non-linearity error	DAILE	8-bit resolution			±1.0	LSB
Differential non-linearity error	DADLE	8-bit resolution			±1.0	LSB
Offset error	DAErr	8-bit resolution			±30	mV
Gain error	DAEG	8-bit resolution			±20	mV
Settling time	DAtset	8-bit resolution, C_L = 50 pF, R_L = 10 kΩ			(60)	μs

Remark 1. In the ratings column, values in parentheses are the target design values and therefore are not tested for shipment.
Remark 2. The 12-bit D/A converter characteristics are the values obtained with the configurable amplifier connected.
Remark 3. Offset error and gain error do not include error in the internal reference voltage.

33.6.6 Configurable amplifier

(T_A = -40 to +105°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V, V_{COM} = 1/2 AV_{DD}, internally connected voltage follower)
AMP0 configuration SW setting: Positive (+) pin = ANX1, negative (-) pin = ANX0
AMP1 configuration SW setting: Positive (+) pin = ANX3, negative (-) pin = ANX2
AMP2 configuration SW setting: Positive (+) pin = ANX5, negative (-) pin = ANX4

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage	V_{IN}		AV_{SS}		AV_{DD}	V
Output voltage	V_{OL}	I_L= -1 mA, AV_{DD} = 2.7 to 5.5 V		AV_{SS} +0.02	AV_{SS} +0.07	V
	V_{OH}	I_L= 1 mA, AV_{DD} = 2.7 to 5.5 V	AV_{DD} -0.15	AV_{DD} -0.02		V
Maximum output current	I_{OUT}	4.5 V ≤ AV_{DD} ≤ 5.5 V	±10			mA
		2.7 V ≤ AV_{DD} ≤ 5.5 V	±5			mA
Input-referred offset voltage	V_{OFF}	T_A = 25°C without trimming I_L = 0 mA, V_{COM} = 1.0 V		±1	±4	mV
		T_A = 25°C with trimming I_L = 0 mA, V_{COM} = 1.0 V			±0.35	mV
Temperature coefficient for inputreferred offset voltage	V_{OTC}	I_L = 0 mA		(±2)	(±8)	μV/°C
Slew rate	SR1	Normal mode C_L = 50 pF, R_L = 10 kΩ		(0.1)		V/μs
	SR2	High-speed mode C_L = 50 pF, R_L = 10 kΩ		(0.8)		V/μs
Gain bandwidth	GBW1	Normal mode C_L = 50 pF, R_L = 10 kΩ		(350)		kHz
	GBW2	High-speed mode C_L = 50 pF, R_L = 10 kΩ		(1.8)		MHz
Phase margin	θM1	Normal mode C_L = 50 pF, R_L = 10 kΩ		(70)		deg
	θM2	High-speed mode C_L = 50 pF, R_L = 10 kΩ		(60)		deg
Settling time	tset1	Normal mode C_L = 50 pF, R_L = 10 kΩ		(20)		μs
	tset2	High-speed mode C_L = 50 pF, R_L = 10 kΩ		(10)		μs
Peak-to-peak voltage noise	Enb	0.1 to 10 Hz Normal mode C_L = 50 pF, R_L = 10 kΩ		(2.0)		μVrms
Input-referred noise	En	f = 1 kHz, Normal mode C_L = 50 pF, R_L = 10 kΩ		(70)		nV/√Hz
Common mode rejection ratio	CMRR	f = 1 kHz, C_L = 50 pF, R_L = 10 kΩ		(70)		dB
Power supply rejection ratio	PSRR	2.7 V ≤ AV_{DD} ≤ 5.5 V f = 1 kHz, C_L = 50 pF, R_L = 10 kΩ		(62)		dB

(Remarks are listed on the next page.)

Remark 1. In the ratings column, values in parentheses are the target design values and therefore are not tested for shipment.
Remark 2. The TYP. conditions are the conditions when T_A = 25°C and AV_{DD} = 5.0 V.
Remark 3. Unless otherwise specified, offset trimming has proceeded.
Remark 4. Unless otherwise specified, values are for operation in normal mode.

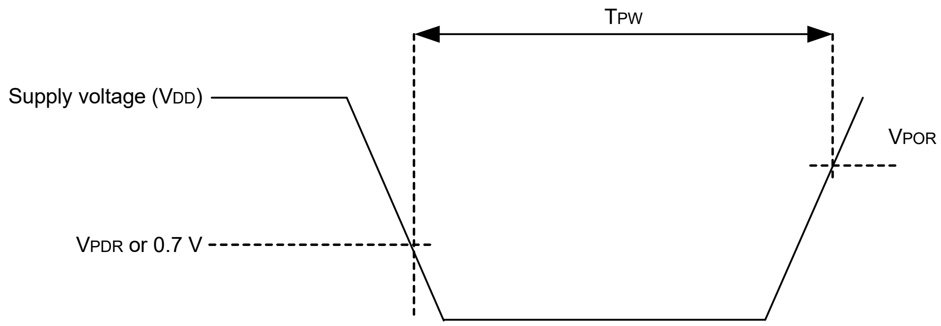
33.6.7 POR characteristics

(T_A = -40 to +105°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Power on/down reset threshold	V_{POR}	Voltage threshold on V_{DD} rising	1.48	1.56	1.62	V
	V_{PDR}	Voltage threshold on V_{DD} falling Note 1	1.47	1.55	1.61	V
Minimum pulse width Note 2	T_{PW}		300			μs

Note 1. However, when the operating voltage falls while the LVD is off, enter STOP mode, or enable the reset status using the external reset pin before the voltage falls below the operating voltage range shown in 33.4 AC Characteristics.

Note 2. Minimum time required for a POR reset when V_{DD} exceeds below V_{PDR}. This is also the minimum time required for a POR reset from when V_{DD} exceeds below 0.7 V to when V_{DD} exceeds V_{POR} while STOP mode is entered or the main system clock is stopped through setting bit 0 (HIOSTOP) and bit 7 (MSTOP) in the clock operation status control register (CSC).



33.6.8 LVD characteristics

(1) LVD detection voltage in reset mode and interrupt mode

(T_A = -40 to +105°C, V_{PDR} ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Voltage detection threshold	Supply voltage level	V_{LVD0}	Rising edge	4.62	4.74	4.84	V
			Falling edge	4.52	4.64	4.74	V
		V_{LVD1}	Rising edge	4.50	4.62	4.72	V
			Falling edge	4.40	4.52	4.62	V
		V_{LVD2}	Rising edge	4.30	4.42	4.51	V
			Falling edge	4.21	4.32	4.41	V
		V_{LVD3}	Rising edge	3.13	3.22	3.29	V
			Falling edge	3.07	3.15	3.22	V
		V_{LVD4}	Rising edge	2.95	3.02	3.09	V
			Falling edge	2.89	2.96	3.02	V
		V_{LVD5}	Rising edge	2.74	2.81	2.87	V
			Falling edge	2.68	2.75	2.81	V
		V_{LVD6}	Rising edge	2.55	2.61	2.67	V
			Falling edge	2.49	2.55	2.61	V
Minimum pulse width		t_{LW}		300			μs
Detection delay time						300	μs

(2) LVD detection voltage in interrupt & reset mode

(T_A = -40 to +105°C, V_{PDR} ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Voltage detection threshold	V_{LVDD6}	V_{POC2}, V_{POC1}, V_{POC0} = 0, 0, 0, falling reset voltage			2.49	2.55	2.61	V
	V_{LVDD4}		LVIS1, LVIS0 = 0, 1	Rising release reset voltage	2.95	3.02	3.09	V
				Falling interrupt voltage	2.89	2.96	3.02	V
	V_{LVDD3}		LVIS1, LVIS0 = 0, 0	Rising release reset voltage	3.13	3.22	3.29	V
				Falling interrupt voltage	3.07	3.15	3.22	V
	V_{LVDD5}	V_{POC2}, V_{POC1}, V_{POC0} = 0, 0, 1, falling reset voltage			2.68	2.75	2.81	V
	V_{LVDD2}		LVIS1, LVIS0 = 0, 0	Rising release reset voltage	4.30	4.42	4.51	V
				Falling interrupt voltage	4.21	4.32	4.41	V
	V_{LVDD5}	V_{POC2}, V_{POC1}, V_{POC0} = 0, 1, 0, falling reset voltage			2.68	2.75	2.81	V
	V_{LVDD1}		LVIS1, LVIS0 = 0, 0	Rising release reset voltage	4.50	4.62	4.72	V
				Falling interrupt voltage	4.40	4.52	4.62	V
	V_{LVDD5}	V_{POC2}, V_{POC1}, V_{POC0} = 0, 1, 1, falling reset voltage			2.68	2.75	2.81	V
	V_{LVDD3}		LVIS1, LVIS0 = 0, 1	Rising release reset voltage	3.13	3.22	3.29	V
				Falling interrupt voltage	3.07	3.15	3.22	V
	V_{LVDD0}		LVIS1, LVIS0 = 0, 0	Rising release reset voltage	4.62	4.74	4.84	V
				Falling interrupt voltage	4.52	4.64	4.74	V

33.6.9 Power supply voltage rising slope characteristics

(T_A = -40 to +105°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Power supply voltage rising slope	S_{VDD}				50	V/ms

Caution Make sure to keep the internal reset state by the LVD circuit or an external reset until V_{DD} reaches the operating voltage range shown in 33.4 AC Characteristics.

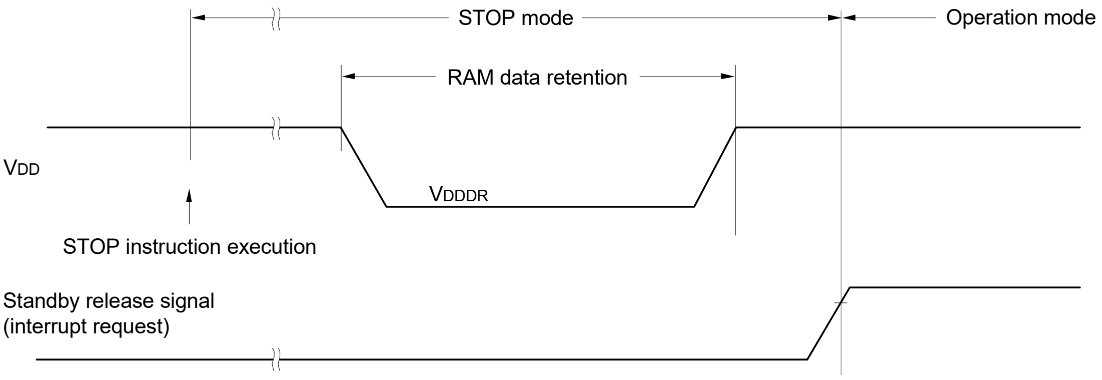
33.7 RAM Data Retention Characteristics

(T_A = -40 to +105°C, V_{SS} = 0 V))

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		1.47 Notes 1, 2		5.5	V

Note 1. The value depends on the POR detection voltage. When the voltage drops, the RAM data is retained before a POR reset is effected, but RAM data is not retained when a POR reset is effected.

Note 2. Enter STOP mode before the supply voltage falls below the recommended operating voltage.



33.8 Flash Memory Programming Characteristics

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
System clock frequency	f_{CLK}	2.4 V ≤ V_{DD} ≤ 5.5 V	1		32	MHz
Number of code flash rewrites Notes 1, 2, 3	C_{erwr}	Retained for 20 years T_A = 85°C Note 4	1,000			Times
Number of data flash rewrites Notes 1, 2, 3		Retained for 1 year T_A = 25°C Note 4		1,000,000		
		Retained for 5 years T_A = 85°C Note 4	100,000			
		Retained for 20 years T_A = 85°C Note 4	10,000			

Note 1. 1 erase + 1 write after the erase is regarded as 1 rewrite. The retaining years are until next rewrite after the rewrite.

Note 2. When using flash memory programmer and Renesas Electronics self-programming library

Note 3. These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas Electronics Corporation.

Note 4. This temperature is the average value at which data are retained.

33.9 Dedicated Flash Memory Programmer Communication (UART)

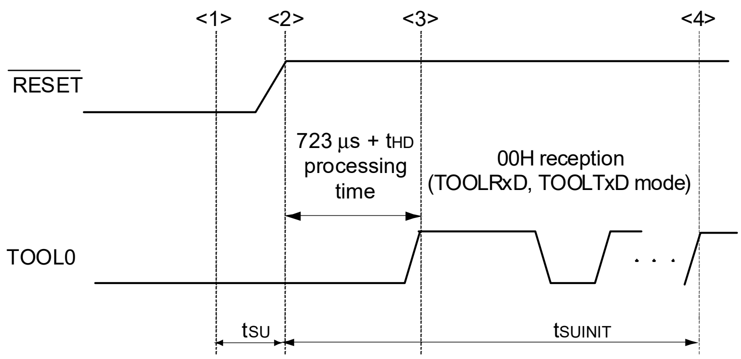
(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		During serial programming	115,200		1,000,000	bps

33.10 Timing for Switching Flash Memory Programming Modes

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
How long from when an external reset ends until the initial communication settings are specified	t_{SUINIT}	POR and LVD reset must end before the external reset ends.			100	ms
How long from when the TOOL0 pin is placed at the low level until an external reset ends	t_{SU}	POR and LVD reset must end before the external reset ends.	10			μs
How long the TOOL0 pin must be kept at the low level after an external reset ends (excluding the processing time of the firmware to control the flash memory)	t_{HD}	POR and LVD reset must end before the external reset ends.	1			ms



<1> The low level is input to the TOOL0 pin.
<2> The external reset ends (POR and LVD reset must end before the external reset ends).
<3> The TOOL0 pin is set to the high level.
<4> Setting of the flash memory programming mode by UART reception and complete the baud rate setting.

Remark t_{SUINIT}: The segment shows that it is necessary to finish specifying the initial communication settings within 100 ms from when the external resets end.
t_{SU}: How long from when the TOOL0 pin is placed at the low level until a pin reset ends
t_{HD}: How long to keep the TOOL0 pin at the low level from when the external resets end (excluding the processing time of the firmware to control the flash memory)

CHAPTER 34 ELECTRICAL SPECIFICATIONS (M: T_A = -40 to +125°C)

This chapter describes the electrical specifications for the products “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 Functions to 2.2.1 Alternate functions other than AFE.

Caution 3. Please contact Renesas Electronics sales office for derating of operation under T_A = +85 to +125°C. Derating is the systematic reduction of load for the sake of improved reliability.

Remark The electrical characteristics of the products M: Industrial applications (T_A = -40 to +125°C) are different from those of the products “G: Industrial applications”. For details, refer to **34.1** to **34.10**.

34.1 Absolute Maximum Ratings

Absolute Maximum Ratings **(1/2)**

Parameter	Symbol	Conditions		Ratings	Unit
Supply voltage	V_{DD}			-0.5 to +6.5	V
	AV_{DD}	AV_{DD} = V_{DD}		-0.5 to +6.5	V
	AV_{SS}	AV_{SS} = V_{SS}		-0.5 to +0.3	V
REGC pin input voltage	V_{IREGC}	REGC		-0.3 to +2.8 and -0.3 to V_{DD} + 0.3 Note 1	V
REGA pin input voltage	V_{IREGA}	REGA		-0.3 to +2.8 and -0.3 to AV_{DD} + 0.3 Note 2	V
Input voltage	V_{I1}	P10 to P15, P40, P121, P122, P137, EXCLK, $\overline{\text{RESET}}$		-0.3 to V_{DD} +0.3 Note 3	V
Alternate-function pin input voltage	V_{I2}	P16, P17, P41, P42 (36-pin products only)	Digital input voltage	-0.3 to V_{DD} + 0.3 Note 3	V
			Analog input voltage	-0.3 to AV_{DD} + 0.3 Note 3	V
Analog input voltage	V_{IA}	PGA0P to PGA3P, PGA0N to PGA3N, ANI0 to ANI9, ANX0 to ANX5		-0.3 to AV_{DD} + 0.3 Note 3	V
Output voltage	V_{O1}	P10 to P15, P40		-0.3 to V_{DD} + 0.3 Note 3	V
Alternate-function pin output voltage	V_{O2}	P16, P17, P41, P42 (36-pin products only)	Digital output voltage	-0.3 to V_{DD} + 0.3 Note 3	V
			Analog output voltage	-0.3 to AV_{DD} + 0.3 Note 3	V
Analog output voltage	V_{OA}	SBIAS, AMP0O to AMP2O, ANX0 to ANX5		-0.3 to AV_{DD} + 0.3 Note 3	V

Note 1. Connect the REGC pin to V_{SS} via a capacitor (0.47 to 1 μF). This value regulates the absolute maximum rating of the REGC pin. Do not use this pin with voltage applied to it.

Note 2. Connect the REGA pin to AV_{SS} via a capacitor (0.22 μF). This value regulates the absolute maximum rating of the REGA pin. Do not use this pin with voltage applied to it.

Note 3. Must be 6.5 V or lower.

Caution **Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.**

Remark 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

Remark 2. V_{SS} is used as the reference voltage.

Absolute Maximum Ratings **(2/2)**

Parameter	Symbol	Conditions		Ratings	Unit
Output current, high	I_{OH1}	Per pin	P10 to P17, P40 to P42	-40	mA
		Total of all pins	P10 to P17, P41, P42 Note	-100	mA
Analog output current, high	I_{OHA}	Per pin	AMP0O to AMP2O	-12	mA
			ANX0 to ANX5	-0.12	mA
		Total of all pins	AMP0O to AMP2O, ANX0 to ANX5	-18	mA
Output current, low	I_{OL1}	Per pin	P10 to P17, P40 to P42	40	mA
		Total of all pins	P10 to P17, P41, P42 Note	100	mA
Analog output current, low	I_{OLA}	Per pin	AMP0O to AMP2O	12	mA
			ANX0 to ANX5	0.12	mA
		Total of all pins	AMP0O to AMP2O, ANX0 to ANX5	18	mA
Operating ambient temperature	T_A	In normal operation mode		-40 to +125	°C
		In flash memory programming mode			
Storage temperature	T_{stg}			-65 to +150	°C

Note This indicates the total current value when P16, P17, P41, and P42 are used as digital input pins.

Caution **Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.**

Remark 1. Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

Remark 2. V_{SS} is used as the reference voltage.

34.2 Oscillator Characteristics

34.2.1 X1 characteristics

(T_A = -40 to +125°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Resonator	Conditions	MIN.	TYP.	MAX.	Unit
X1 clock oscillation frequency (f_X) Note	Ceramic resonator/ crystal resonator	2.7 V ≤ V_{DD} ≤ 5.5 V	1.0		20.0	MHz
		2.4 V ≤ V_{DD} < 2.7 V	1.0		16.0	

Note Indicates only permissible oscillator frequency ranges. Refer to **AC Characteristics** for instruction execution time. Request evaluation by the manufacturer of the oscillator circuit mounted on a board to check the oscillator characteristics.

Caution Since the CPU is started by the high-speed on-chip oscillator clock after a reset release, check the X1 clock oscillation stabilization time using the oscillation stabilization time counter status register (OSTC) by the user. Determine the oscillation stabilization time of the OSTC register and the oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.

Remark When using the X1 oscillator, refer to **5.4 System Clock Oscillator**.

34.2.2 On-chip oscillator characteristics

(T_A = -40 to +125°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
High-speed on-chip oscillator clock frequency Notes 1, 2	f_{IH}	2.7 V ≤ V_{DD} ≤ 5.5 V	1		24	MHz
		2.4 V ≤ V_{DD} < 2.7 V	1		16	MHz
High-speed on-chip oscillator clock frequency accuracy		-40 to +105°C	-2.0		+2.0	%
		+105 to +125°C	-3.0		+3.0	%
Low-speed on-chip oscillator clock frequency	f_{IL}			15		kHz
Low-speed on-chip oscillator clock frequency accuracy			-15		+15	%

Note 1. High-speed on-chip oscillator frequency is selected with bits 0 to 3 of the option byte (000C2H) and bits 0 to 2 of the HOCODIV register.

Note 2. This only indicates the oscillator characteristics. Refer to **AC Characteristics** for instruction execution time.

34.2.3 PLL characteristics

(T_A = -40 to +125°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
PLL output frequency Notes 1, 2, 3	f_{PLL}	f_{MX} = 8 MHz	DSFRDIV = 0	DSCM = 0		48		MHz
			DSFRDIV = 1	DSCM = 0		24		MHz
				DSCM = 1		32		MHz
		f_{MX} = 4 MHz	DSFRDIV = 0	DSCM = 0		24		MHz
				DSCM = 1		32		MHz
Lockup wait time		Time from when PLL output is enabled to when the phase is locked			40			μs
Interval wait time		Time from when the PLL stops operating to when the setting to start PLL operation is specified			4			μs
Setup wait time		Time required from when the PLL input clock stabilizes and the PLL setting is determined to when the PLL is activated			1			μs

Note 1. When using a PLL, input a clock of 4 MHz or 8 MHz to the PLL.

Note 2. Be sure to specify one of these settings when using a PLL.

Note 3. When using the PLL output as the CPU clock, f_{IH} is divided by 2, 4, or 8 according to the setting of the RDIV1 and RDIV0 bits.

34.3 DC Characteristics

34.3.1 Pin characteristics

(TA = -40 to +125°C, 2.4 V ≤ AVDD = VDD ≤ 5.5 V, AVSS = VSS = 0 V) **(1/3)**

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high Note 1	IOH1	Per pin for P10 to P17 and P40 to P42 Note 2	4.0 V ≤ VDD ≤ 5.5 V			-3.0 Note 3	mA
			2.4 V ≤ VDD < 4.0 V			-1.0 Note 3	mA
		Total of P10 to P17, P41, and P42 Note 3 (When duty ≤ 70% Note 4)	4.0 V ≤ VDD ≤ 5.5 V			-30.0	mA
			2.7 V ≤ VDD < 4.0 V			-19.0	mA
			2.4 V ≤ VDD < 2.7 V			-10.0	mA
Output current, low Note 1	IOL1	Per pin for P10 to P17 and P40 to P42 Note 2	4.0 V ≤ VDD ≤ 5.5 V			8.5 Note 3	mA
			2.7 V ≤ VDD < 4.0 V			1.5 Note 3	mA
			2.4 V ≤ VDD < 2.7 V			0.6 Note 3	mA
		Total of P10 to P17, P41, and P42 Note 2 (When duty ≤ 70% Note 4)	4.0 V ≤ VDD ≤ 5.5 V			40.0	mA
			2.7 V ≤ VDD < 4.0 V			35.0	mA
			2.4 V ≤ VDD < 2.7 V			20.0	mA

Note 1. Value of current at which the device operation is guaranteed even if the current flows from the VDD pin to an output pin.

Note 2. This indicates the total current value when P16, P17, P41, and P42 are used as digital I/O ports. When using these pins as analog function (AFE) pins, refer to **34.1 Absolute Maximum Ratings**.

Note 3. Do not exceed the total current value.

Note 4. Specification under conditions where the duty factor ≤ 70%.
The output current value that has changed to the duty factor > 70% the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).

- Total output current of pins = (IOH × 0.7)/(n × 0.01)
  Example: n = 80% when IOH = -10.0 mA
  Total output current of pins = (-10.0 × 0.7)/(80 × 0.01) ≈ -8.7 mA

However, the current that is allowed to flow into one pin does not vary depending on the duty factor.
A current higher than the absolute maximum rating must not flow into one pin.

Caution P10 to P15 do not output high level in N-ch open-drain mode.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = -40 to +125°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V) **(2/3)**

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P10 to P17 and P40 to P42	Normal input buffer	0.8 V_{DD}		V_{DD}	V
	V_{IH2}	P11, P12, P14, P15	TTL input buffer, 4.0 V ≤ V_{DD} ≤ 5.5 V	2.2		V_{DD}	V
			TTL input buffer, 3.3 V ≤ V_{DD} < 4.0 V	2.0		V_{DD}	V
			TTL input buffer, 2.4 V ≤ V_{DD} < 3.3 V	1.28		V_{DD}	V
	V_{IH3}	P121, P122, P137, EXCLK, $\overline{RESET}$		0.8 V_{DD}		V_{DD}	V
Input voltage, low	V_{IL1}	P10 to P17 and P40 to P42	Normal input buffer	0		0.2 V_{DD}	V
	V_{IL2}	P11, P12, P14, P15	TTL input buffer, 4.0 V ≤ V_{DD} ≤ 5.5 V	0		0.8	V
			TTL input buffer, 3.3 V ≤ V_{DD} < 4.0 V	0		0.5	V
			TTL input buffer, 2.4 V ≤ V_{DD} < 3.3 V	0		0.32	V
	V_{IL3}	P121, P122, P137, EXCLK, $\overline{RESET}$		0		0.2 V_{DD}	V
Output voltage, high	V_{OH1}	P10 to P17 and P40 to P42	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = -3.0 mA	V_{DD} - 0.7			V
			2.4 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = -1.0 mA	V_{DD} - 0.5			V
Output voltage, low	V_{OL1}	P10 to P17 and P40 to P42	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 8.5 mA			0.7	V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 1.5 mA			0.5	V
			2.4 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 0.6 mA			0.4	V

Caution **The maximum V_{IH} value on P10 to P15 is V_{DD}, even in the N-ch open-drain mode.**

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

(T_A = -40 to +125°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V) **(3/3)**

Item	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P10 to P17, and P40 to P42	$V_I = V_{DD}$				1	µA
	I_{LIH2}	P137, $\overline{RESET}$	$V_I = V_{DD}$				1	µA
	I_{LIH3}	P121, P122 (X1, X2, EXCLK)	$V_I = V_{DD}$	In input port mode or when using external clock input			1	µA
				When a resonator is connected			10	µA
Input leakage current, low	I_{LIL1}	P10 to P17, and P40 to P42	$V_I = V_{SS}$				-1	µA
	I_{LIL2}	P137, $\overline{RESET}$	$V_I = V_{SS}$				-1	µA
	I_{LIL3}	P121, P122 (X1, X2, EXCLK)	$V_I = V_{SS}$	In input port mode or when using external clock input			-1	µA
				When a resonator is connected			-10	µA
On-chip pull-up resistance	R_U	P10 to P15, P40	$V_I = V_{SS}$, in input port mode		10	20	100	kΩ

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of the port pins.

34.3.2 Supply current characteristics

(TA = -40 to +125°C, 2.4 V ≤ AVDD = VDD ≤ 5.5 V, AVSS = VSS = 0 V) **(1/2)**

Parameter	Symbol	Conditions				MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD1}	Operating mode Note 2	f_{HOCO} = 24 MHz, f_{MAIN} = 24 MHz Note 3	Basic operation	V_{DD} = 5.0 V		1.7		mA
					V_{DD} = 3.0 V		1.7		
			f_{HOCO} = 24 MHz, f_{MAIN} = 24 MHz Note 3	Normal operation	V_{DD} = 5.0 V		3.8	7.6	mA
					V_{DD} = 3.0 V		3.8	7.6	
			f_{HOCO} = 16 MHz, f_{MAIN} = 16 MHz Note 3	Normal operation	V_{DD} = 5.0 V		2.8	5.6	
					V_{DD} = 3.0 V		2.8	5.6	
			f_{MX} = 20 MHz, f_{MAIN} = 20 MHz Note 4, V_{DD} = 5.0 V	Normal operation	Square wave input		3.3	6.5	mA
					Resonator connection		3.5	6.6	
			f_{MX} = 20 MHz, f_{MAIN} = 20 MHz Note 4, V_{DD} = 3.0 V	Normal operation	Square wave input		3.3	6.5	
					Resonator connection		3.5	6.6	
			f_{MX} = 10 MHz, f_{MAIN} = 10 MHz Note 4, V_{DD} = 5.0 V	Normal operation	Square wave input		2.0	3.9	
					Resonator connection		2.1	4.0	
			f_{MX} = 10 MHz, f_{MAIN} = 10 MHz Note 4, V_{DD} = 3.0 V	Normal operation	Square wave input		2.0	3.9	
					Resonator connection		2.1	4.0	
			f_{MX} = 8 MHz, f_{MAIN} = 24 MHz Note 5, V_{DD} = 5.0 V	Normal operation	Square wave input		5.1	10.4	mA
					Resonator connection		5.2	10.5	
			f_{MX} = 8 MHz, f_{MAIN} = 24 MHz Note 5, V_{DD} = 3.0 V	Normal operation	Square wave input		5.1	10.4	
					Resonator connection		5.2	10.5	

Note 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The values below the MAX. column include the peripheral operation current. However, not including the current flowing into the RTC, interval timer, watchdog timer, LVD circuit, AFE, I/O ports, and on-chip pull-up/pull-down resistors and the current flowing during data flash rewrite.

Note 2. The relationship between the operation voltage range and the CPU operating frequency is as below.
2.7 V ≤ V_{DD} ≤ 5.5 V @ 1 MHz to 24 MHz
2.4 V ≤ V_{DD} ≤ 5.5 V @ 1 MHz to 16 MHz

Note 3. When the high-speed system clock is stopped

Note 4. When the high-speed on-chip oscillator and the PLL are stopped

Note 5. When the high-speed on-chip oscillator is stopped and the PLL is operating

Remark 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

Remark 2. f_{HOCO}: High-speed on-chip oscillator clock frequency

Remark 3. f_{MAIN}: Main system clock frequency

Remark 4. The temperature condition for the TYP. value is TA = 25°C

(T_A = -40 to +125°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V) **(2/2)**

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current Note 1	I_{DD2} Note 2	HALT mode Note 3	f_{HOCO} = 24 MHz, f_{MAIN} = 24 MHz Note 4	V_{DD} = 5.0 V		0.44	3.42	mA
				V_{DD} = 3.0 V		0.44	3.42	
			f_{HOCO} = 16 MHz, f_{MAIN} = 16 MHz Note 4	V_{DD} = 5.0 V		0.40	2.50	
				V_{DD} = 3.0 V		0.40	2.50	
			f_{MX} = 20 MHz, f_{MAIN} = 20 MHz Note 5, V_{DD} = 5.0 V	Square wave input		0.28	2.94	mA
				Resonator connection		0.49	3.08	
			f_{MX} = 20 MHz, f_{MAIN} = 20 MHz Note 5, V_{DD} = 3.0 V	Square wave input		0.28	2.94	
				Resonator connection		0.49	3.08	
			f_{MX} = 10 MHz, f_{MAIN} = 10 MHz Note 5, V_{DD} = 5.0 V	Square wave input		0.19	1.54	
				Resonator connection		0.30	1.63	
			f_{MX} = 10 MHz, f_{MAIN} = 10 MHz Note 5, V_{DD} = 3.0 V	Square wave input		0.19	1.54	
				Resonator connection		0.30	1.63	
			f_{MX} = 8 MHz, f_{MAIN} = 24 MHz Note 6, V_{DD} = 5.0 V	Square wave input		0.76	3.92	mA
				Resonator connection		0.86	4.04	
			f_{MX} = 8 MHz, f_{MAIN} = 24 MHz Note 6, V_{DD} = 3.0 V	Square wave input		0.76	3.92	
				Resonator connection		0.86	4.04	
	I_{DD3} Note 7	STOP mode	T_A = -40°C			0.38	1.14	μA
			T_A = +25°C			0.50	1.14	
			T_A = +50°C			0.66	4.52	
			T_A = +70°C			1.04	7.98	
			T_A = +85°C			2.92	16.0	
			T_A = +105°C			11.0	100.0	
			T_A = +125°C			22.0	200.0	

Note 1. Total current flowing into V_{DD}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The values below the MAX. column include the peripheral operation current. However, not including the current flowing into the RTC, interval timer, watchdog timer, LVD circuit, AFE, I/O ports, and on-chip pull-up/pull-down resistors and the current flowing during writing to the data flash.

Note 2. During HALT instruction execution from flash memory

Note 3. The relationship between the operation voltage range and the CPU operating frequency is as below.
2.7 V ≤ V_{DD} ≤ 5.5 V @ 1 MHz to 24 MHz
2.4 V ≤ V_{DD} ≤ 5.5 V @ 1 MHz to 16 MHz

Note 4. When the high-speed system clock is stopped

Note 5. When the high-speed on-chip oscillator and the PLL are stopped

Note 6. When high-speed on-chip oscillator is stopped and the PLL is operating

Note 7. The MAX. value includes the leakage current in STOP mode.

Remark 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

Remark 2. f_{HOCO}: High-speed on-chip oscillator clock frequency

Remark 3. f_{MAIN}: Main system clock frequency

Remark 4. The temperature condition for the TYP. value is T_A = 25°C, except the operation in STOP mode.

• Peripheral functions

(T_A = -40 to +125°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Low-speed on-chip oscillator operating current	I_{FIL} Note 1				0.20		µA
RTC operating current	I_{RTC} Notes 1, 2, 3	f_{MX} = 4 MHz, RTCCL = 00H (f_{MX}/122)			22		µA
Interval timer operating current	I_{IT} Notes 1, 2, 4	f_{MX} = 4 MHz, RTCCL = 00H (f_{MX}/122)			22		µA
Watchdog timer operating current	I_{WDT} Notes 1, 5, 6	f_{IL} = 15 kHz			0.22		µA
LVD operating current	I_{LVD} Notes 1, 7				0.08		µA
Self-programming operating current	I_{FSP} Notes 1, 8				2.00	12.20	mA
BGO operating current	I_{BGO} Notes 1, 9				2.00	12.20	mA
SNOOZE operating current	I_{SNOZ} Note 1	A/D converter operation Notes 11,	The mode is performed Note 10		0.50	1.10	mA
			During A/D conversion, AV_{DD} = V_{DD} = 3.0 V		1.20	2.04	
		Simplified SPI (CSI)/UART operation			0.70	1.54	
		DTC operation			3.10		

Note 1. Current flowing to V_{DD}

Note 2. When the high-speed on-chip oscillator is stopped

Note 3. Current flowing only to the real-time clock (RTC). The supply current of the RL78 microcontrollers is the sum of the values of either I_{DD1} or I_{DD2}, and I_{RTC}, when the real-time clock is operating in operation mode or HALT mode.

Note 4. Current flowing only to the interval timer. The supply current of the RL78 microcontrollers is the sum of the values of either I_{DD1} or I_{DD2}, and I_{IT}, when the interval timer is operating in operation mode or HALT mode. When the low-speed on-chip oscillator is selected, also add I_{FIL}.

Note 5. When the high-speed on-chip oscillator and high-speed system clock are stopped.

Note 6. Current flowing only to the watchdog timer (including the operating current of the low-speed on-chip oscillator). The supply current of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{WDT} when the watchdog timer is operating.

Note 7. Current flowing only to the LVD circuit. The supply current of the RL78 microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{LVD} when the LVD circuit is operating.

Note 8. Current flowing during self-programming

Note 9. Current flowing during writing to the data flash

Note 10. For time required to shift to the SNOOZE mode, see **23.3.3 SNOOZE mode**.

Note 11. The current flowing into the AV_{DD} is included.

Remark 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

Remark 2. f_{IL}: Low-speed on-chip oscillator clock frequency

Remark 3. The temperature condition for the TYP. value is T_A = 25°C

• AFE functions

(T_A = -40 to +125°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
24-bit ΔΣ A/D converter operating current	I_{DSAD}	Normal mode Notes 1, 2 Circuits that operate: ABGR, REGA, SBIAS, VREFAMP, PGA, 24-bit ΔΣ A/D converter, and digital filter Differential input mode OSR = 256 SBIAS I_{OUT} = 0 mA		0.94	1.46	mA
		Low power mode Notes 1, 2 Circuits that operate: ABGR, REGA, SBIAS, VREFAMP, PGA, 24-bit ΔΣ A/D converter, and digital filter Differential input mode OSR = 256 SBIAS I_{OUT} = 0 mA		0.60	0.91	mA
10-bit A/D converter operating current	I_{ADC}	During conversion at the highest speed Notes 1, 2 AV_{DD} = 5.0 V		1.30	1.70	mA
Configurable amplifier operating current	I_{AMP}	Normal mode Notes 1, 2 Circuits that operate: ABGR and configurable amplifier I_L = 0 mA Per channel		0.13	0.24	mA
		High-speed mode Notes 1, 2 Circuits that operate: ABGR and configurable amplifier I_L = 0 mA Per channel		0.30	0.45	mA
12-bit D/A converter operating current	I_{DAC}	When AV_{DD} and AV_{SS} are selected as the reference voltage Notes 1, 2 Circuits that operate: ABGR and internal reference voltage (VREFDA)		0.61	0.97	mA

Note 1. Current flowing to AV_{DD}

Note 2. Current flowing only to the circuits that operate shown in the Conditions column.

34.4 AC Characteristics

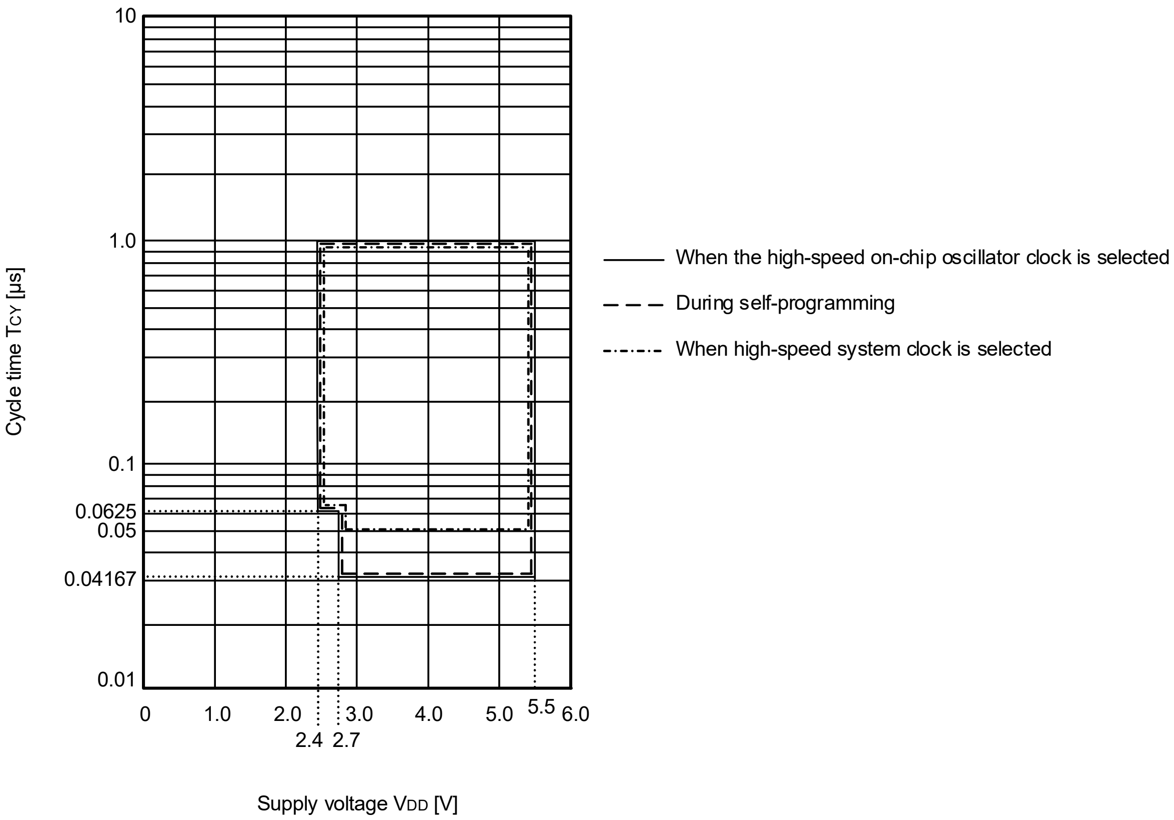
(T_A = -40 to +125°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation	2.7 V ≤ V_{DD} ≤ 5.5 V	0.04167		1	μs
			2.4 V ≤ V_{DD} < 2.7 V	0.0625		1	μs
		In the self-programming mode	2.7 V ≤ V_{DD} ≤ 5.5 V	0.04167		1	μs
			2.4 V ≤ V_{DD} < 2.7 V	0.0625		1	μs
External system clock frequency	f_{EX}	2.7 V ≤ V_{DD} ≤ 5.5 V		1.0		20.0	MHz
		2.4 V ≤ V_{DD} < 2.7 V		1.0		8.0	MHz
External system clock input high-level width, low-level width	t_{EXH}, t_{EXL}	2.7 V ≤ V_{DD} ≤ 5.5 V		24			ns
		2.4 V ≤ V_{DD} < 2.7 V		60			ns
TI00 to TI03, TI10, TI11 input high-level width, low-level width	t_{TIH}, t_{TIL}			$1/f_{MCK}$ + 10			ns
Timer RJ input cycle	f_C	TRJIO0	2.7 V ≤ V_{DD} ≤ 5.5 V	100			ns
			2.4 V ≤ V_{DD} < 2.7 V	300			ns
Timer RJ input high-level width, low-level width	t_{TJIH}, t_{TJIL}	TRJIO0	2.7 V ≤ V_{DD} ≤ 5.5 V	40			ns
			2.4 V ≤ V_{DD} < 2.7 V	120			ns
Timer RG input high-level width, low-level width	t_{TGIH}, t_{TGIL}	TRGIOA, TRGIOB		$2.5/f_{CLK}$			ns
TO00 to TO03, TO10, TO11, TRJIO0, TRJO0, TRGIOA, TRGIOB output frequency	f_{TO}		4.0 V ≤ V_{DD} ≤ 5.5 V			12	MHz
			2.7 V ≤ V_{DD} ≤ 4.0 V			6	MHz
			2.4 V ≤ V_{DD} < 2.7 V			3	MHz
PCLBUZ0 output frequency	f_{PCL}		4.0 V ≤ V_{DD} ≤ 5.5 V			12	MHz
			2.7 V ≤ V_{DD} ≤ 4.0 V			6	MHz
			2.4 V ≤ V_{DD} < 2.7 V			3	MHz
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}	INTP1 to INTP7		1			μs
$\overline{RESET}$ low-level width	t_{RSL}			10			μs

Remark f_{MCK}: Timer array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of timer mode register mn (TMRmn). m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3))

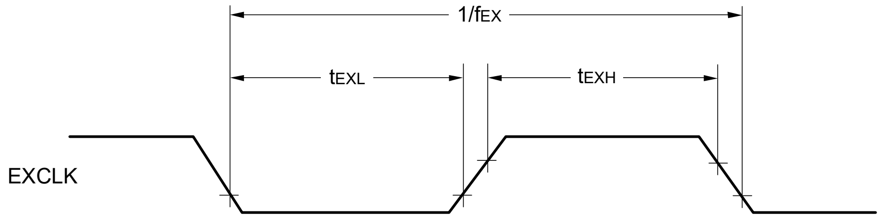
Minimum Instruction Execution Time During Main System Clock Operation

TCY vs VDD

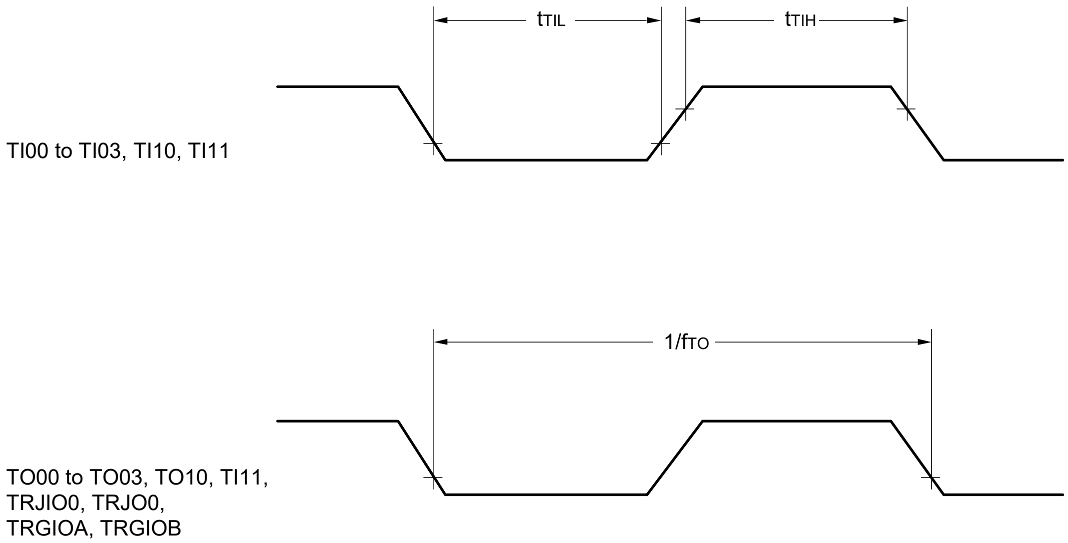


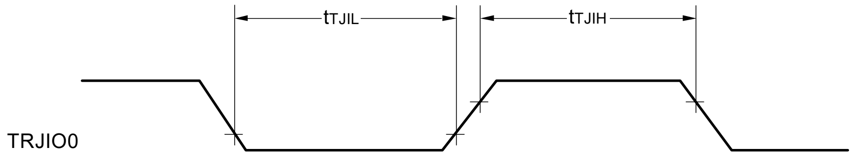
AC Timing Test Points

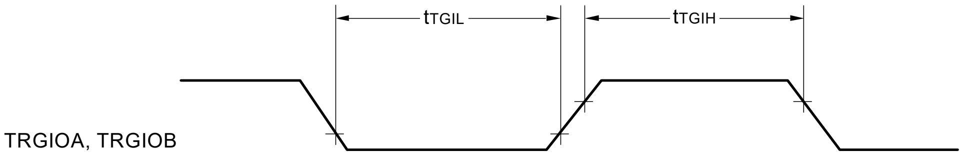
External System Clock Timing



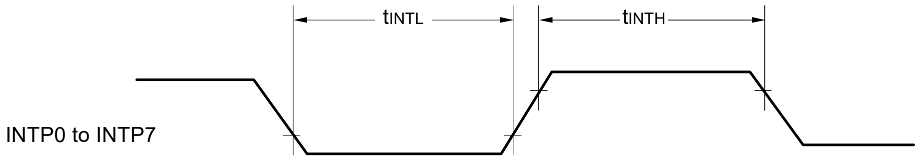
TI/TO Timing



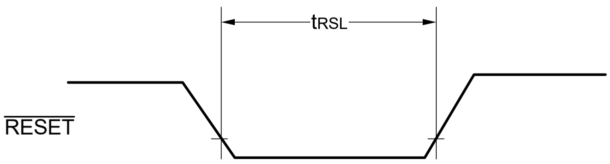




Interrupt Request Input Timing



$\overline{\text{RESET}}$ Input Timing



34.5 Peripheral Functions Characteristics

AC Timing Test Points

34.5.1 Serial array unit

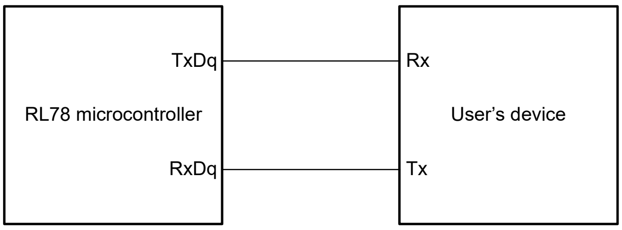
(1) During communication at same potential (UART mode)
(T_A = -40 to +125°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) Mode		Unit
			MIN.	MAX.	
Transfer rate Note 1				f_{MCK}/12	bps
		Theoretical value of the maximum transfer rate f_{MCK} = f_{CLK} Note 2		2.0	Mbps

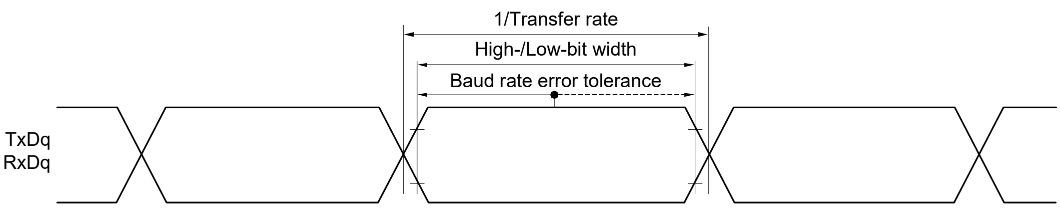
Note 1. Transfer rate in the SNOOZE mode is 4800 bps only.
Note 2. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:
24 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)
16 MHz (2.4 V ≤ V_{DD} ≤ 5.5 V)

Caution Select the normal input buffer for the RxDq pin and the normal output mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg).

UART mode connection diagram (during communication at same potential)



UART mode bit width (during communication at same potential) (reference)



Remark 1. q: UART number (q = 0, 1), g: PIM or POM number (g = 1)
Remark 2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00 to 03))

(2) During communication at same potential (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = -40 to +125°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) mode MIN.	HS (high-speed main) mode MAX.	Unit
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ 4/f_{CLK}	2.7 V ≤ V_{DD} ≤ 5.5 V	333		ns
			2.4 V ≤ V_{DD} ≤ 5.5 V	666		ns
SCKp high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2 - 24		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2 - 36		ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		t_{KCY1}/2 - 76		ns
SIp setup time (to SCKp↑) Note 1	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V		66		ns
		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 output Note 2	t_{KSO1}	C = 30 pF Note 3			66.6	ns

Note 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. C is the load capacitance of the SCKp and SOp output lines.

Caution **Select the normal input buffer for the SIp pin and the normal output mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg).**

Remark 1. p: CSI number (p = 00, 01), m: Unit number (m = 0), n: Channel number (n = 0, 1), g: PIM number (g = 1)

Remark 2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 01))

(3) During communication at same potential (Simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

(T_A = -40 to +125°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) mode MIN.	HS (high-speed main) mode MAX.	Unit
SCKp cycle time Note 1	t_{KCY2}	4.0 V ≤ V_{DD} ≤ 5.5 V	20 MHz < f_{MCK}	16/f_{MCK}		ns
			f_{MCK} ≤ 20 MHz	12/f_{MCK}		ns
		2.7 V ≤ V_{DD} ≤ 5.5 V	16 MHz < f_{MCK}	16/f_{MCK}		ns
			f_{MCK} ≤ 16 MHz	12/f_{MCK}		ns
		2.4 V ≤ V_{DD} ≤ 5.5 V		12/f_{MCK} and 1000		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ V_{DD} ≤ 5.5 V		t_{KCY2}/2 - 14		ns
		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 2	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 2	t_{KSI2}			1/f_{MCK} + 62		ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO2}	C = 30 pF Note 4	2.7 V ≤ V_{DD} ≤ 5.5 V		2/f_{MCK} + 66	ns
			2.4 V ≤ V_{DD} ≤ 5.5 V		2/f_{MCK} + 113	ns
$\overline{SSI00}$ setup time	t_{SSIK}	DAPmn = 0	2.7 V ≤ V_{DD} ≤ 5.5 V	240		ns
			2.4 V ≤ V_{DD} ≤ 5.5 V	400		ns
		DAPmn = 1	2.7 V ≤ V_{DD} ≤ 5.5 V	1/f_{MCK} + 240		ns
			2.4 V ≤ V_{DD} ≤ 5.5 V	1/f_{MCK} + 400		ns
$\overline{SSI00}$ hold time	t_{KSSI}	DAPmn = 0	2.7 V ≤ V_{DD} ≤ 5.5 V	1/f_{MCK} + 240		ns
			2.4 V ≤ V_{DD} ≤ 5.5 V	1/f_{MCK} + 400		ns
		DAPmn = 1	2.7 V ≤ V_{DD} ≤ 5.5 V	240		ns
			2.4 V ≤ V_{DD} ≤ 5.5 V	400		ns

Note 1. The maximum transfer rate in the SNOOZE mode is 1 Mbps.

Note 2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

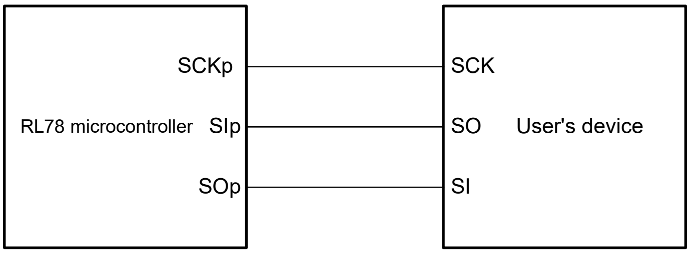
Note 4. C is the load capacitance of the SOp output lines.

Caution Select the normal input buffer for the SIp and SCKp pins and the normal output mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg).

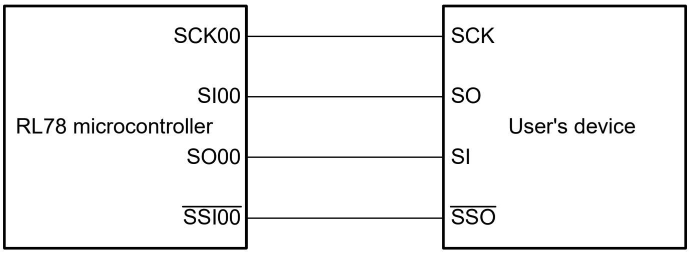
Remark 1. p: CSI number (p = 00, 01), m: Unit number (m = 0), n: Channel number (n = 0, 1), g: PIM number (g = 1)

Remark 2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 01))

Simplified SPI (CSI) mode connection diagram (during communication at same potential)

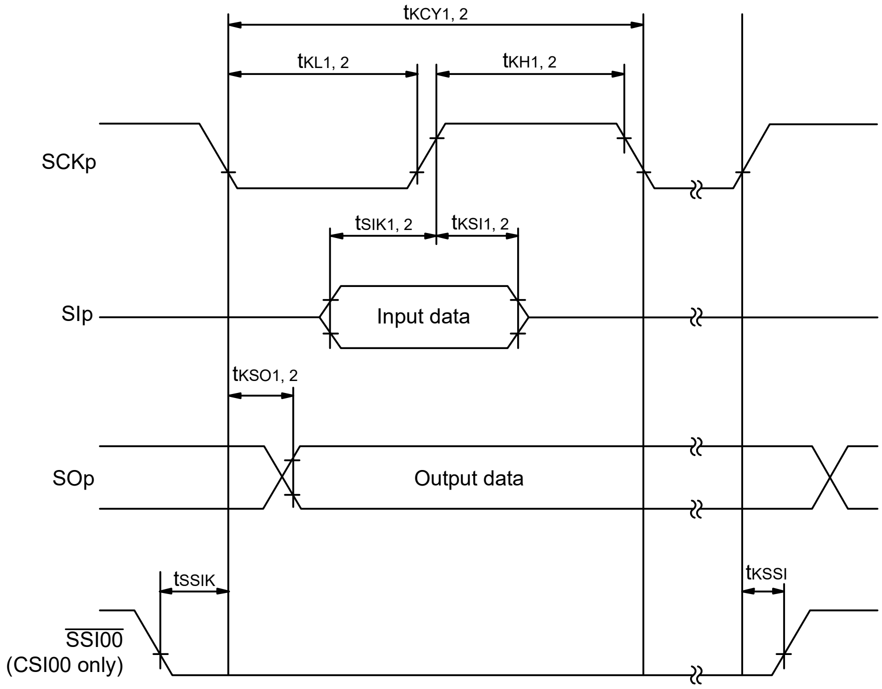


Simplified SPI (CSI) mode connection diagram (during communication at same potential)
(Slave Transmission of slave select input function (CSI00))

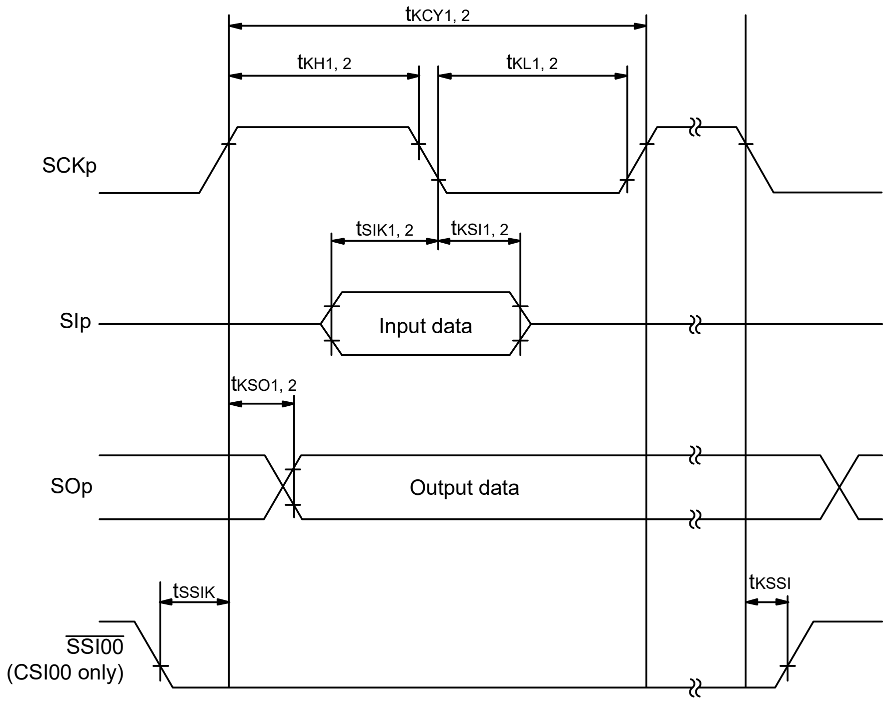


Remark 1. p: CSI number (p = 00, 01)
Remark 2. m: Unit number, n: Channel number (mn = 00, 01)

Simplified SPI (CSI) mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remark 1. p: CSI number (p = 00, 01)
Remark 2. m: Unit number, n: Channel number (mn = 00, 01)

(4) During communication at same potential (simplified I²C mode)

(T_A = -40 to +125°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions	HS (high-speed main) mode		Unit
			MIN.	MAX.	
SCLr clock frequency	f_{SCL}	2.7 V ≤ V_{DD} ≤ 5.5 V, C_b = 50 pF, R_b = 2.7 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 = 50 pF, R_b = 2.7 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 = 50 pF, R_b = 2.7 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 = 50 pF, R_b = 2.7 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 = 50 pF, R_b = 2.7 kΩ	0	770	ns
		2.4 V ≤ V_{DD} ≤ 5.5 V, C_b = 100 pF, R_b = 3 kΩ	0	1420	ns

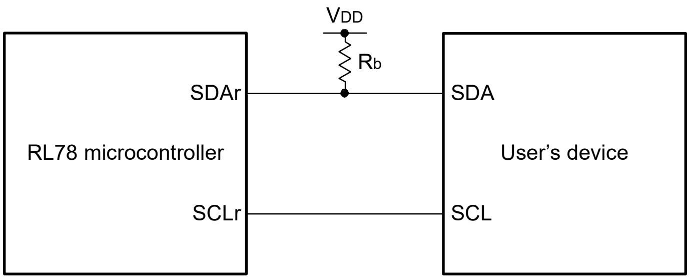
Note 1. The value must also be equal to or less than $f_{MCK}/4$.

Note 2. Set the f_{MCK} value to keep the hold time of SCLr = “L” and SCLr = “H”.

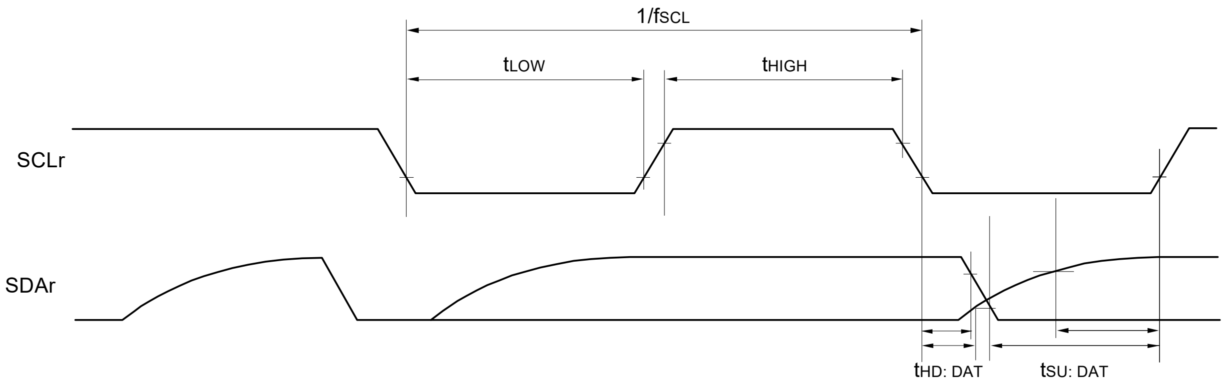
Caution Select the normal input buffer and the N-ch open drain output (V_{DD} tolerance) mode for the SDAr pin and the normal output mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register h (POMh).

(Remarks are listed on the next page.)

Simplified I2C mode connection diagram (during communication at same potential)



Simplified I2C mode serial transfer timing (during communication at same potential)



Remark 1. R_b [Ω]: Communication line (SDAr) pull-up resistance, C_b [F]: Communication line (SDAr, SCLr) load capacitance

Remark 2. r: IIC number (r = 00, 01), g: PIM number (g = 1), h: POM number (h = 1)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number (m = 0), n: Channel number (n = 0, 1), mn = 00, 01)

(5) Communication at different potential (1.8 V, 2.5 V, 3 V) (UART mode)

(T_A = -40 to +125°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions			HS (high-speed main) mode MIN.	HS (high-speed main) mode MAX.	Unit
Transfer rate		Reception	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V			$f_{MCK}/12$ Note 1	bps
				Theoretical value of the maximum transfer rate $f_{MCK} = f_{CLK}$ Note 2		2.0	Mbps
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V			$f_{MCK}/12$ Note 1	bps
				Theoretical value of the maximum transfer rate $f_{MCK} = f_{CLK}$ Note 2		2.0	Mbps
			2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V			$f_{MCK}/12$ Note 1	bps
				Theoretical value of the maximum transfer rate $f_{MCK} = f_{CLK}$ Note 2		2.0	Mbps

Note 1. Transfer rate in the SNOOZE mode is 4800 bps only.

Note 2. The maximum operating frequencies of the CPU/peripheral hardware clock (f_{CLK}) are:
24 MHz (2.7 V ≤ V_{DD} ≤ 5.5 V)
16 MHz (2.4 V ≤ V_{DD} ≤ 5.5 V)

Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

Remark 1. V_b [V]: Communication line voltage

Remark 2. q: UART number (q = 0, 1), g: PIM or POM number (g = 1)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number, n: Channel number (mn = 00, 01)

(5) Communication at different potential (1.8 V, 2.5 V, 3 V) (UART mode)

(T_A = -40 to +125°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V) (2/2)

Parameter	Symbol	Conditions			HS (high-speed main) mode		Unit
					MIN.	MAX.	
Transfer rate		Transmission	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V			Note 1	bps
				Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 1.4 kΩ, V_b = 2.7 V		2.0 Note 2	Mbps
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V			Note 3	bps
				Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 2.7 kΩ, V_b = 2.3 V		1.2 Note 4	Mbps
			2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V			Note 5	bps
				Theoretical value of the maximum transfer rate C_b = 50 pF, R_b = 5.5 kΩ, V_b = 1.6 V		0.43 Note 6	Mbps

Note 1. The smaller maximum transfer rate derived by using $f_{MCK}/12$ or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 4.0 V ≤ V_{DD} ≤ 5.5 V and 2.7 V ≤ V_b ≤ 4.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln\left(1 - \frac{2.2}{V_b}\right)\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln\left(1 - \frac{2.2}{V_b}\right)\}}{\left(\frac{1}{\text{Transfer rate}}\right) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

Note 2. This value as an example is calculated when the conditions described in the “Conditions” column are met.
Refer to **Note 1** above to calculate the maximum transfer rate under conditions of the customer.

Note 3. The smaller maximum transfer rate derived by using $f_{MCK}/12$ or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 2.7 V ≤ V_{DD} < 4.0 V and 2.3 V ≤ V_b ≤ 2.7 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln\left(1 - \frac{2.0}{V_b}\right)\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln\left(1 - \frac{2.0}{V_b}\right)\}}{\left(\frac{1}{\text{Transfer rate}}\right) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

Note 4. This value as an example is calculated when the conditions described in the “Conditions” column are met.
Refer to **Note 3** above to calculate the maximum transfer rate under conditions of the customer.

Note 5. The smaller maximum transfer rate derived by using $f_{MCK}/12$ or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 2.4 V ≤ V_{DD} < 3.3 V and 1.6 V ≤ V_b ≤ 2.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln (1 - \frac{1.5}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln (1 - \frac{1.5}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

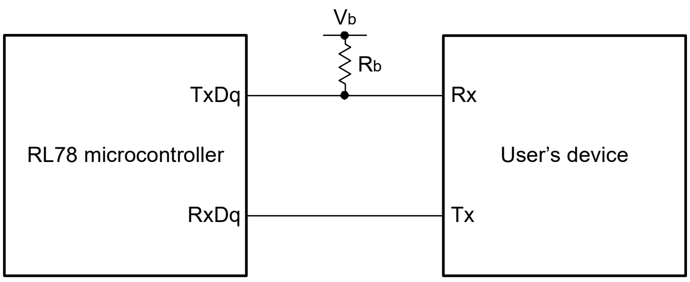
Note 6. This value as an example is calculated when the conditions described in the "Conditions" column are met.

Refer to **Note 5** above to calculate the maximum transfer rate under conditions of the customer.

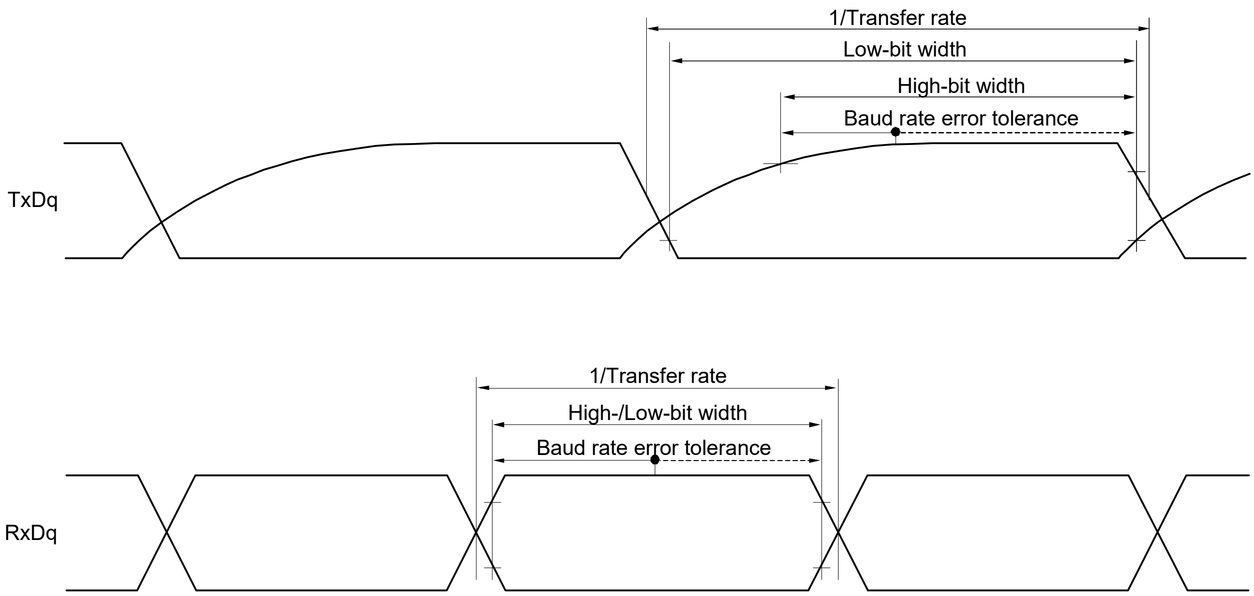
Caution Select the TTL input buffer for the RxDq pin and the N-ch open drain output (V_{DD} tolerance) mode for the TxDq pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

(Remarks are listed on the next page.)

UART mode connection diagram (during communication at different potential)



UART mode bit width (during communication at different potential) (reference)



Remark 1. R_b [Ω]: Communication line (TxDq) pull-up resistance,
C_b [F]: Communication line (TxDq) load capacitance, V_b [V]: Communication line voltage

Remark 2. q: UART number (q = 0, 1), g: PIM or POM number (g = 1)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00, 01))

(6) Communication at different potential (1.8 V, 2.5 V, 3 V) (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = -40 to +125°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V) (1/3)

Parameter	Symbol	Conditions		HS (high-speed main) mode		Unit
				MIN.	MAX.	
SCKp cycle time	t_{KCY1}	t_{KCY1} ≥ 4/f_{CLK}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	600		ns
			2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	1000		ns
			2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	2300		ns
SCKp high-level width	t_{KH1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KCY1}/2 - 150		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 - 340		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		t_{KCY1}/2 - 916		ns
SCKp low-level width	t_{KL1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		t_{KCY1}/2 - 24		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		t_{KCY1}/2 - 36		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		t_{KCY1}/2 - 100		ns

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

(Remarks are listed two pages after the next page.)

(6) Communication at different potential (1.8 V, 2.5 V, 3 V) (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = -40 to +125°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V) (2/3)

Parameter	Symbol	Conditions	HS (high-speed main) mode		Unit
			MIN.	MAX.	
SIp setup time (to SCKp↑) Note	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	162		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	354		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	958		ns
SIp hold time (from SCKp↑) Note	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	38		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	38		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	38		ns
Delay time from SCKp↓ to SOp output Note	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		200	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		390	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		966	ns

Note When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

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

(6) Communication at different potential (1.8 V, 2.5 V, 3 V) (Simplified SPI (CSI) mode) (master mode, SCKp... internal clock output)

(T_A = -40 to +125°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V) **(3/3)**

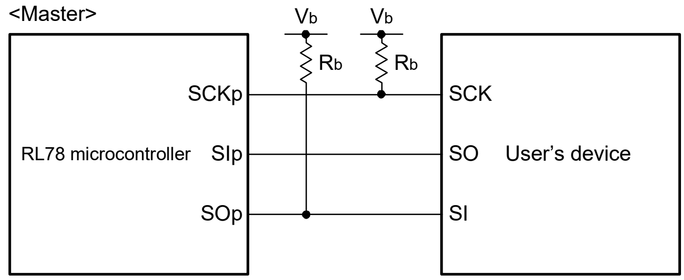
Parameter	Symbol	Conditions	HS (high-speed main) mode		Unit
			MIN.	MAX.	
SIp setup time (to SCKp↓) Note	t_{SIK1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	88		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	88		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	220		ns
SIp hold time (from SCKp↓) Note	t_{KSI1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	38		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	38		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ	38		ns
Delay time from SCKp↑ to SOp output Note	t_{KSO1}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ		50	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ		50	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ		50	ns

Note When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Caution Select the TTL input buffer for the SIp pin and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin and SCKp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

(Remarks are listed on the next page.)

Simplified SPI (CSI) mode connection diagram (during communication at different potential

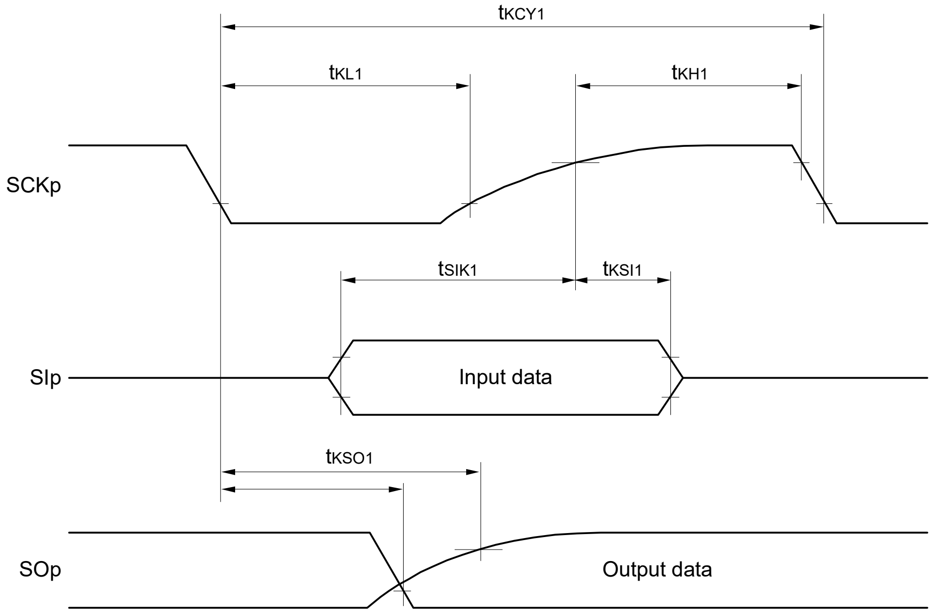


Remark 1. R_b [Ω]: Communication line (SCKp, SOp) pull-up resistance, C_b [F]: Communication line (SCKp, SOp) load capacitance, V_b [V]: Communication line voltage

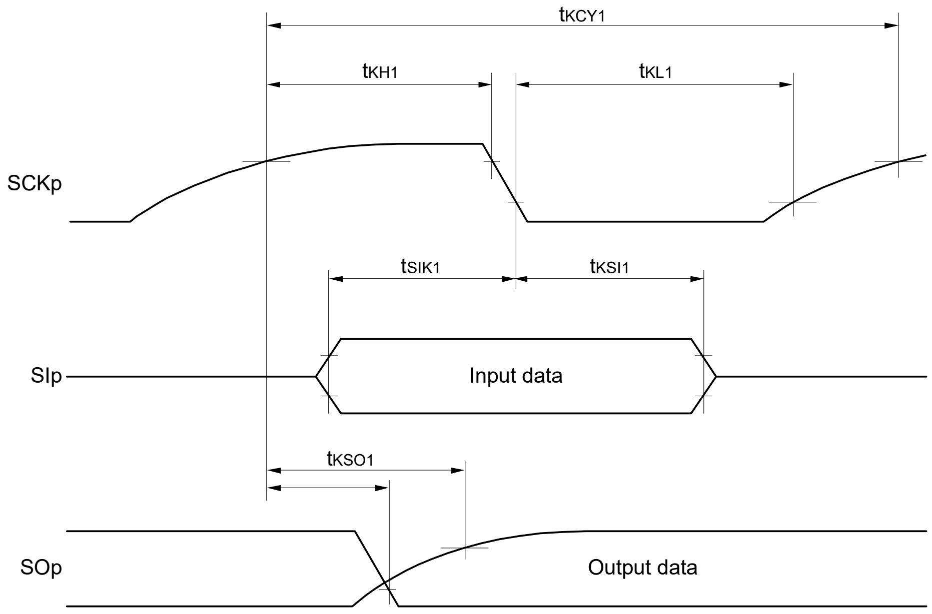
Remark 2. p: CSI number (p = 00, 01), m: Unit number (m = 0), n: Channel number (n = 0, 1), g: PIM or POM number (g = 1)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number,
n: Channel number (mn = 00, 01))

Simplified SPI (CSI) mode serial transfer timing (master mode) (during communication at different potential) (When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (master mode) (during communication at different potential) (When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remark p: CSI number (p = 00, 01), m: Unit number (m = 0), n: Channel number (n = 0, 1), g: PIM or POM number (g = 1)

(7) Communication at different potential (1.8 V, 2.5 V, 3 V) (Simplified SPI (CSI) mode) (slave mode, SCKp... external clock input)

(T_A = -40 to +125°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions		HS (high-speed main) mode		Unit
				MIN.	MAX.	
SCKp cycle time Note 1	t_{KCY2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V	20 MHz < f_{MCK} ≤ 24 MHz	24/f_{MCK}		ns
			8 MHz < f_{MCK} ≤ 20 MHz	20/f_{MCK}		ns
			4 MHz < f_{MCK} ≤ 8 MHz	16/f_{MCK}		ns
			f_{MCK} ≤ 4 MHz	12/f_{MCK}		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V	20 MHz < f_{MCK} ≤ 24 MHz	32/f_{MCK}		ns
			16 MHz < f_{MCK} ≤ 20 MHz	28/f_{MCK}		ns
			8 MHz < f_{MCK} ≤ 16 MHz	24/f_{MCK}		ns
			4 MHz < f_{MCK} ≤ 8 MHz	16/f_{MCK}		ns
			f_{MCK} ≤ 4 MHz	12/f_{MCK}		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V	20 MHz < f_{MCK} ≤ 24 MHz	72/f_{MCK}		ns
			16 MHz < f_{MCK} ≤ 20 MHz	64/f_{MCK}		ns
			8 MHz < f_{MCK} ≤ 16 MHz	52/f_{MCK}		ns
			4 MHz < f_{MCK} ≤ 8 MHz	32/f_{MCK}		ns
			f_{MCK} ≤ 4 MHz	20/f_{MCK}		ns
SCKp high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		t_{KCY2}/2 - 24		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		t_{KCY2}/2 - 36		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V		t_{KCY2}/2 - 100		ns
SIp setup time (to SCKp↑) Note 2	t_{SIK2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		1/f_{MCK} + 40		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		1/f_{MCK} + 40		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V		1/f_{MCK} + 60		ns
SIp hold time (from SCKp↑) Note 2	t_{KSI2}			1/f_{MCK} + 62		ns
Delay time from SCKp↓ to SOp output Note 3	t_{KSO2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ			2/f_{MCK} + 240	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			2/f_{MCK} + 428	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 30 pF, R_b = 5.5 kΩ			2/f_{MCK} + 1146	ns

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

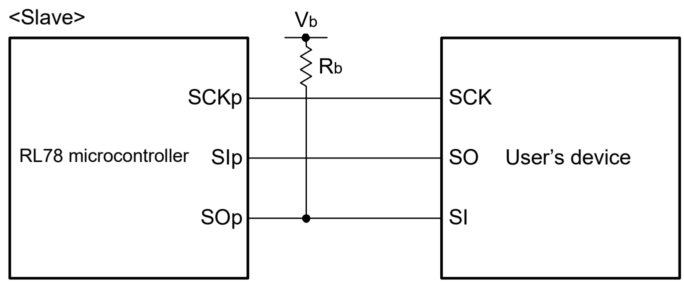
Note 1. Transfer rate in the SNOOZE mode: MAX. 1 Mbps

Note 2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0. The SIp hold time becomes "from SCKp↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Note 3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from SCKp↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

Caution **Select the TTL input buffer for the SIp and SCKp pins, and the N-ch open drain output (V_{DD} tolerance) mode for the SOp pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.**

Simplified SPI (CSI) mode connection diagram (during communication at different potential)



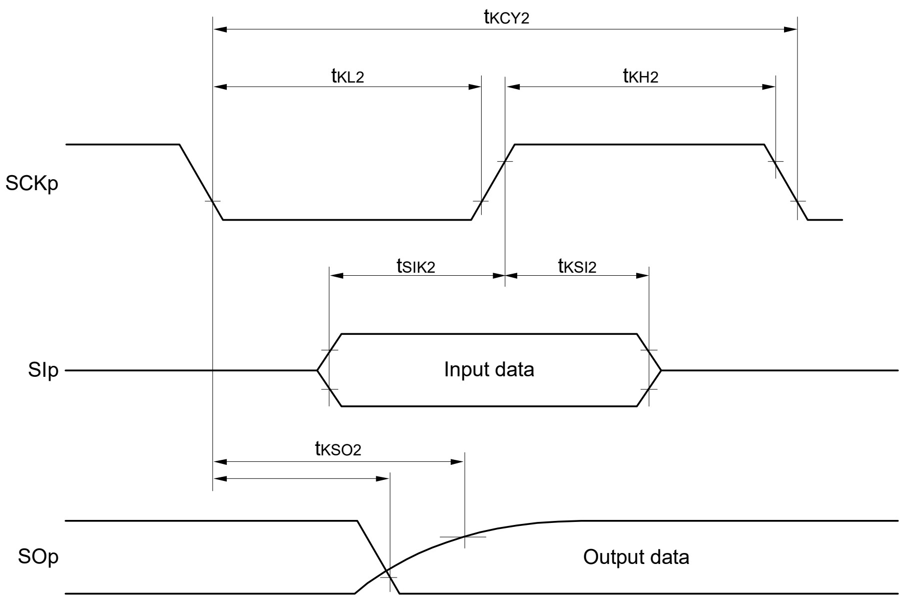
Remark 1. R_b [Ω]: Communication line (SOp) pull-up resistance, C_b [F]: Communication line (SOp) load capacitance, V_b [V]: Communication line voltage

Remark 2. p: CSI number (p = 00, 01), m: Unit number (m = 0), n: Channel number (n = 0 , 1), g: PIM or POM number (g = 1)

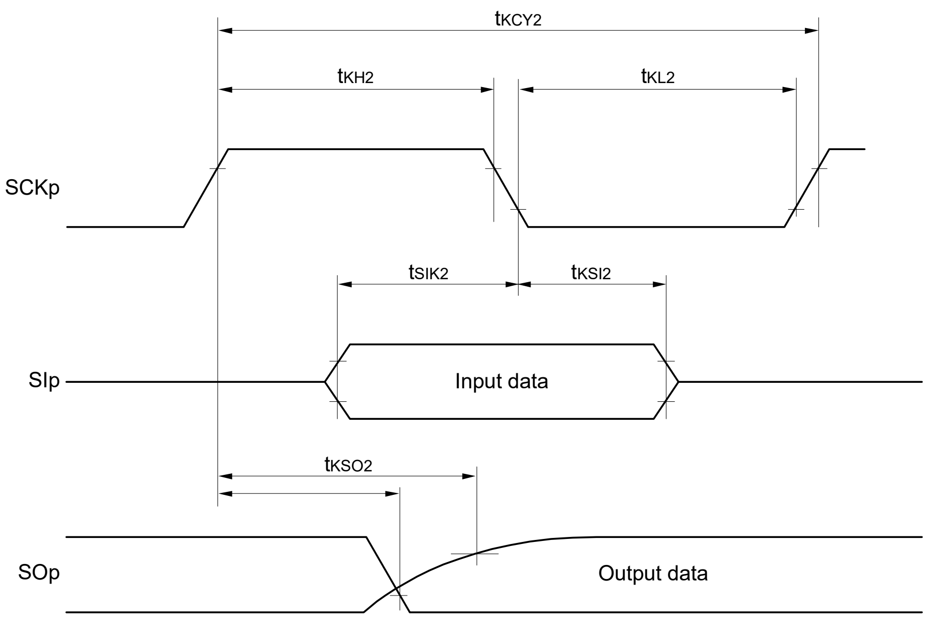
Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn).
m: Unit number, n: Channel number (mn = 00, 01))

Remark 4. Communication at different potential cannot be performed during clocked serial communication with the slave select function.

Simplified SPI (CSI) mode serial transfer timing (slave mode) (during communication at different potential) (When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



Simplified SPI (CSI) mode serial transfer timing (slave mode) (during communication at different potential) (When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remark 1. p: CSI number (p = 00, 01), m: Unit number (m = 0), n: Channel number (n = 0, 1), g: PIM or POM number (g = 1)

Remark 2. Communication at different potential cannot be performed during clocked serial communication with the slave select function.

(8) Communication at different potential (1.8 V, 2.5 V, 3 V) (simplified I^2C mode)

(T_A = -40 to +125°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V) **(1/2)**

Parameter	Symbol	Conditions	HS (high-speed main) mode		Unit
			MIN.	MAX.	
SCLr clock frequency	f_{SCL}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ		400 Note 1	kHz
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ		400 Note 1	kHz
		4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ		100 Note 1	kHz
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ		100 Note 1	kHz
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 100 pF, R_b = 5.5 kΩ		100 Note 1	kHz
Hold time when SCLr = "L"	t_{LOW}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	1200		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	1200		ns
		4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	4600		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	4600		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 100 pF, R_b = 5.5 kΩ	4600		ns
Hold time when SCLr = "H"	t_{HIGH}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	620		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	500		ns
		4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	2700		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	2400		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 100 pF, R_b = 5.5 kΩ	1830		ns

(8) Communication at different potential (1.8 V, 2.5 V, 3 V) (simplified I²C mode)

(T_A = -40 to +125°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V) (2/2)

Parameter	Symbol	Conditions	HS (high-speed main) mode		Unit
			MIN.	MAX.	
Data setup time (reception)	$t_{SU:DAT}$	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	$1/f_{MCK}$ + 340 Note 1		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	$1/f_{MCK}$ + 340 Note 2		ns
		4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	$1/f_{MCK}$ + 760 Note 2		ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	$1/f_{MCK}$ + 760 Note 2		ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 100 pF, R_b = 5.5 kΩ	$1/f_{MCK}$ + 570 Note 2		ns
Data hold time (transmission)	$t_{HD:DAT}$	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 50 pF, R_b = 2.7 kΩ	0	770	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 50 pF, R_b = 2.7 kΩ	0	770	ns
		4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 100 pF, R_b = 2.8 kΩ	0	1420	ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, C_b = 100 pF, R_b = 2.7 kΩ	0	1420	ns
		2.4 V ≤ V_{DD} < 3.3 V, 1.6 V ≤ V_b ≤ 2.0 V, C_b = 100 pF, R_b = 5.5 kΩ	0	1215	ns

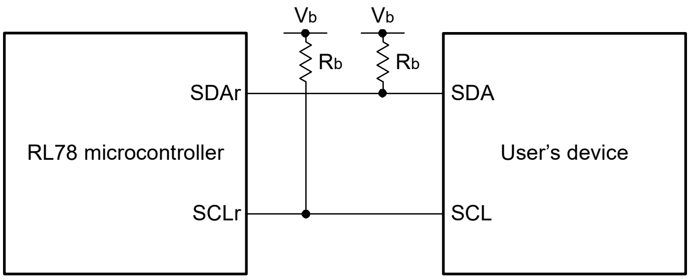
Note 1. The value must also be equal to or less than f_{MCK}/4.

Note 2. Set the f_{MCK} value to keep the hold time of SCLr = "L" and SCLr = "H".

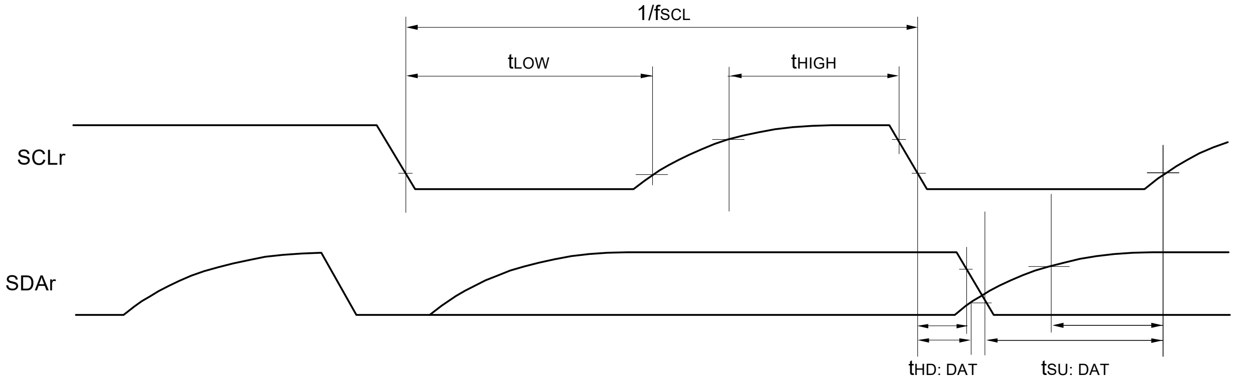
Caution Select the TTL input buffer and the N-ch open drain output (V_{DD} tolerance) mode for the SDAr pin and the N-ch open drain output (V_{DD} tolerance) mode for the SCLr pin by using port input mode register g (PIMg) and port output mode register g (POMg). For V_{IH} and V_{IL}, see the DC characteristics with TTL input buffer selected.

(Remarks are listed on the next page.)

Simplified I²C mode connection diagram (during communication at different potential)



Simplified I²C mode serial transfer timing (during communication at different potential)



Remark 1. R_b [Ω]: Communication line (SDAr, SCLr) pull-up resistance, C_b [F]: Communication line (SDAr, SCLr) load capacitance, V_b [V]: Communication line voltage

Remark 2. r: IIC number (r = 00, 01), g: PIM, POM number (g = 1)

Remark 3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of serial mode register mn (SMRmn). m: Unit number (m = 0), n: Channel number (n = 0), mn = 00, 01)

34.6 Analog Characteristics

34.6.1 Programmable gain instrumentation amplifier and 24-bit ΔΣ A/D converter

(1) Analog input in differential input mode

(T_A = -40 to +125°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V, normal mode: fs1 = 1 MHz, FDATA1 = 3.90625 ksps, low-power mode: fs2 = 0.125 MHz, FDATA2 = 488.28125 sps, SBIAS = 1.2 V, d_{OFR} = 0 mV, V_{COM} = 1.0 V, external clock input used)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Full-scale differential input voltage range	V_{ID}	V_{ID} = (PGAxP - PGAxN) (x = 0 to 3)		± 800 /G_{TOTAL}		mV
Input voltage range	V_I	Each of PGAxP and PGAxN pins (x = 0 to 3)	0.2		1.8	V
Common mode input voltage	V_{COM}	d_{OFR} = 0 mV	0.2+(|V_{ID}|x G_{SET1})/2		1.8-(|V_{ID}|x G_{SET1})/2	V
Input bias current	I_{IN}	V_I = 1.0 V			± 50	nA
Input offset current	I_{INOFR}	V_I = 1.0 V			± 20	nA

(2) Analog input in single-ended input mode

(T_A = -40 to +125°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V, normal mode: fs1 = 1 MHz, FDATA1 = 3.90625 ksps, low-power mode: fs2 = 0.125 MHz, FDATA2 = 488.28125 sps, SBIAS = 1.2 V, d_{OFR} = 0 mV, V_{COM} = 1.0 V, external clock input used)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage range	V_I	Each of PGAxP and PGAxN pins (x = 0 to 3) G_{SET1} = 1, G_{SET2} = 1	0.2		1.8	V
Input bias current	I_{IN}	V_I = 1.0 V			± 50	nA

(3) Programmable gain instrumentation amplifier and 24-bit ΔΣ A/D converter

(T_A = -40 to +125°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V, normal mode: fs1 = 1 MHz, FDATA1 = 3.90625 ksps, low-power mode: fs2 = 0.125 MHz, FDATA2 = 488.28125 sps, SBIAS = 1.2 V, d_{OFR} = 0 mV, V_{COM} = 1.0 V, external clock input used, in differential input mode) (1/2)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	RES				24	bit
Sampling frequency	f_{S1}	Normal mode		1		MHz
	f_{S2}	Low power mode		0.125		MHz
Output data rate	f_{DATA1}	Normal mode	0.48828		15.625	ksps
	f_{DATA2}	Low power mode	61.03615		1953.125	sps
Gain setting range	G_{TOTAL}	G_{TOTAL} = G_{SET1} × G_{SET2}	1		64	V/V
1st gain setting range	G_{SET1}	In differential input mode only		1, 2, 3, 4, 8		V/V
2nd gain setting range	G_{SET2}	In differential input mode only		1, 2, 4, 8		V/V
Offset adjustment bit range	d_{OFFB}			5		bit
Offset adjustment range	d_{OFR}	Referred to input	-164/G_{SET1}		+164/G_{SET1}	mV
Offset adjustment steps	d_{OFS}	Referred to input		11/G_{SET1}		mV

(T_A = -40 to +125°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V, normal mode: fs1 = 1 MHz, FDATA1 = 3.90625 ksps, low-power mode: fs2 = 0.125 MHz, FDATA2 = 488.28125 sps, SBIAS = 1.2 V, d_{OFR} = 0 mV, V_{COM} = 1.0 V, external clock input used, in differential input mode) (2/2)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Gain error	E_G	T_A = 25°C G_{SET1} = 1, G_{SET2} = 1 Excluding SBIAS error		±0.2	±2.7	%
		T_A = 25°C G_{SET1} = 8, G_{SET2} = 4 Excluding SBIAS error		±0.1		%
Gain drift Note	dE_G	G_{SET1} = 1, G_{SET2} = 1 Excluding SBIAS drift		(5.6)	(22.0)	ppm/°C
		G_{SET1} = 8, G_{SET2} = 4 Excluding SBIAS drift		(9.1)		ppm/°C
Offset error	E_{OS}	T_A = 25°C G_{SET1} = 1, G_{SET2} = 1 Referred to input		±0.32	±2.90	mV
		T_A = 25°C G_{SET1} = 8, G_{SET2} = 4 Referred to input		±0.03		mV
Offset drift Note	dE_{OS}	G_{SET1} = 1, G_{SET2} = 1 Referred to input		(±0.02)	(±6.00)	μV/°C
		G_{SET1} = 8, G_{SET2} = 4 Referred to input		(±0.02)		μV/°C
SND ratio	SNDR	G_{SET1} = 1, G_{SET2} = 1, f_{IN} = 50 Hz Normal mode, pin = -1 dBFS	(82)	(85)		dB
		G_{SET1} = 8, G_{SET2} = 4, f_{IN} = 50 Hz Normal mode, pin = -1 dBFS	(73)	(80)		dB
Noise	Vn	G_{SET1} = 1, G_{SET2} = 1, OSR = 2048		(13)		μVRms
		G_{SET1} = 8, G_{SET2} = 4, OSR = 2048		(0.6)		μVRms
Integral non-linearity error	INL	G_{SET1} = 1, G_{SET2} = 1, OSR = 2048		(±10)		ppmFS
Common mode rejection ratio	CMRR	V_{COM} = 1.0 ± 0.8 V, fin = 50 Hz G_{SET1} = 1, G_{SET2} = 1 Differential input mode	(72)	(90)		dB
Power supply rejection ratio	PSRR	AV_{DD} = 2.7 to 5.5 V G_{SET1} = 1, G_{SET2} = 1 Differential input mode		(85)		dB
ΔΣ A/D converter input clock frequency	f_{ADC}		3.8	4	4.2	MHz

Note Calculate the gain drift and offset drift by using the following expression (for 125°C products):
For gain drift: (MAX(E_G($T_{(-40)}$ to $T_{(125)}$)) - MIN(E_G($T_{(-40)}$ to $T_{(125)}$))) / (125°C -(-40°C))
For offset drift: (MAX(E_{OS}($T_{(-40)}$ to $T_{(125)}$)) - MIN(E_{OS}($T_{(-40)}$ to $T_{(125)}$))) / (125°C -(-40°C))
MAX(E_G($T_{(-40)}$ to $T_{(125)}$)): The maximum value of gain error when the temperature range is -40°C to 125°C
MIN(E_G($T_{(-40)}$ to $T_{(125)}$)): The minimum value of gain error when the temperature range is -40°C to 125°C
MAX(E_{OS}($T_{(-40)}$ to $T_{(125)}$)): The maximum value of offset error when the temperature range is -40°C to 125°C
MIN(E_{OS}($T_{(-40)}$ to $T_{(125)}$)):The minimum value of offset error when the temperature range is -40°C to 125°C

Remark In the ratings column, values in parentheses are the target design values and therefore are not tested for shipment.

34.6.2 Sensor power supply (SBIAS)

(T_A = -40 to +125°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V, C_{OUT} = 0.22 μF, V_{OUT} = 1.0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Output voltage range	V_{OUT}		0.5		2.2	V
Output voltage setting steps	V_{STEP}			0.1		V
Output voltage precision	V_A	I_{OUT} = 1 mA	(-3)		(+3)	%
Maximum output current	I_{OUT}		5			mA
Short circuit current	I_{SHORT}	V_{OUT} = 0 V		40	65	mA
Load regulation	L_R	1 mA ≤ I_{OUT} ≤ 5 mA			(15)	mV
Power supply rejection ratio	PSRR	AV_{DD} = 5.0 V + 0.1 $V_{pp\ ripple}$ f = 100 Hz, I_{OUT} = 2.5 mA	(45)	(50)		dB

Remark In the ratings column, values in parentheses are the target design values and therefore are not tested for shipment.

34.6.3 Temperature sensor

(T_A = -40 to +125°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Temperature coefficient for sensor	TC_{SNS}			(756)		μV/°C
Sensor output voltage	V_{TEMP}	T_A = 25°C		226.4		mV

Remark In the ratings column, values in parentheses are the target design values and therefore are not tested for shipment.

34.6.4 A/D converter characteristics

(1) When positive reference voltage (+) = AV_{DD} (ADREFP1 = 0, ADREFP0 = 0), negative reference voltage (-) = AV_{SS} (ADREFM = 0), pins subject to A/D conversion: ANI0 to ANI9 and SBIAS

(T_A = -40 to +125°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V, positive reference voltage (+) = AV_{DD}, negative reference voltage (-) = AV_{SS})

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES			8		10	bit
Overall error Note 1	AINL	10-bit resolution ANI0 to ANI9, SBIAS	4.0 V ≤ AV_{DD} ≤ 5.5 V		1.2	±6.5	LSB
			2.7 V ≤ AV_{DD} ≤ 5.5 V		1.2	±7.0	LSB
Conversion time	t_{CONV}	10-bit resolution	4.0 V ≤ AV_{DD} ≤ 5.5 V	2.125		39	μs
			2.7 V ≤ AV_{DD} ≤ 5.5 V	3.1875		39	μs
Zero-scale error Notes 1, 2	E_{ZS}	10-bit resolution ANI0 to ANI9, SBIAS	4.0 V ≤ AV_{DD} ≤ 5.5 V			±0.50	%FSR
			2.7 V ≤ AV_{DD} ≤ 5.5 V			±0.60	%FSR
Full-scale error Notes 1, 2	E_{FS}	10-bit resolution ANI0 to ANI9, SBIAS	4.0 V ≤ AV_{DD} ≤ 5.5 V			±0.50	%FSR
			2.7 V ≤ AV_{DD} ≤ 5.5 V			±0.60	%FSR
Integral linearity error Note 1	ILE	10-bit resolution ANI0 to ANI9, SBIAS	4.0 V ≤ AV_{DD} ≤ 5.5 V			±3.5	LSB
			2.7 V ≤ AV_{DD} ≤ 5.5 V			±4.0	LSB
Differential linearity error Note 1	DLE	10-bit resolution	2.7 V ≤ AV_{DD} ≤ 5.5 V			±2.0	LSB
Analog input voltage	V_{AIN}	ANI0 to ANI9		AV_{SS}		AV_{DD}	V

Note 1. Excludes quantization error (±1/2 LSB).
Note 2. This value is indicated as a ratio (%FSR) to the full-scale value.

Caution The number of pins depends on the product. For details, see a list of pin functions.

(2) When positive reference voltage (+) = Internal reference voltage (ADREFP1 = 1, ADREFP0 = 0), negative reference voltage (-) = AV_{SS} (ADREFM = 0), pins subject to A/D conversion: ANI0 to ANI9 and SBIAS

(T_A = -40 to +125°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V, positive reference voltage (+) = V_{BGR}, negative reference voltage (-) = AV_{SS})

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Resolution	RES				8		bit
Conversion time	t_{CONV}	8-bit resolution	2.7 V ≤ AV_{DD} ≤ 5.5 V	17		39	μs
Zero-scale error Notes 1, 2	E_{ZS}	8-bit resolution	2.7 V ≤ AV_{DD} ≤ 5.5 V			±0.60	%FSR
Integral linearity error Note 1	ILE	8-bit resolution	2.7 V ≤ AV_{DD} ≤ 5.5 V			±2.0	LSB
Differential linearity error Note 1	DLE	8-bit resolution	2.7 V ≤ AV_{DD} ≤ 5.5 V			±1.0	LSB
Internal reference voltage (+)	V_{BGR}	2.7 V ≤ AV_{DD} ≤ 5.5 V			V_{BGR} Note 3		V
Analog input voltage	V_{AIN}	ANI0 to ANI9		0		V_{BGR}	V

Note 1. Excludes quantization error (±1/2 LSB).
Note 2. This value is indicated as a ratio (%FSR) to the full-scale value.
Note 3. See the Internal reference voltage characteristics.

34.6.5 12-bit D/A converter

(1) When positive reference voltage (+) = AV_{DD} (DACVRF = 0)

(T_A = -40 to +125°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V, positive reference voltage (+) = AV_{DD})

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	DARES				(12)	bit
Output voltage range	DAOUT	12-bit resolution	0.35		AV_{DD}-0.47	V
Integral non-linearity error	DAILE	12-bit resolution			±4.0	LSB
Differential non-linearity error	DADLE	12-bit resolution			±1.0	LSB
Offset error	DAErr	12-bit resolution			±30	mV
Gain error	DAEG	12-bit resolution			±20	mV
Settling time	DAtset	12-bit resolution, C_L = 50 pF, R_L = 10 kΩ			(60)	μs

Remark 1. In the ratings column, values in parentheses are the target design values and therefore are not tested for shipment.
Remark 2. The 12-bit D/A converter characteristics are the values obtained with the configurable amplifier connected.

(2) When positive reference voltage (+) = internal reference voltage (DACVRF = 1)

(T_A = -40 to +125°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V, positive reference voltage (+) = VREFDA)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	DARES				(8)	bit
Internal reference voltage	VREFDA	8-bit resolution	1.34	1.45	1.54	V
Output voltage range	DAOUT	8-bit resolution	0.35		VREFDA	V
Integral non-linearity error	DAILE	8-bit resolution			±1.0	LSB
Differential non-linearity error	DADLE	8-bit resolution			±1.0	LSB
Offset error	DAErr	8-bit resolution			±30	mV
Gain error	DAEG	8-bit resolution			±20	mV
Settling time	DAtset	8-bit resolution, C_L = 50 pF, R_L = 10 kΩ			(60)	μs

Remark 1. In the ratings column, values in parentheses are the target design values and therefore are not tested for shipment.
Remark 2. The 12-bit D/A converter characteristics are the values obtained with the configurable amplifier connected.
Remark 3. Offset error and gain error do not include error in the internal reference voltage.

34.6.6 Configurable amplifier

(T_A = -40 to +125°C, 2.7 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V, V_{COM} = 1/2 AV_{DD}, internally connected voltage follower)

AMP0 configuration SW setting: Positive (+) pin = ANX1, negative (-) pin = ANX0

AMP1 configuration SW setting: Positive (+) pin = ANX3, negative (-) pin = ANX2

AMP2 configuration SW setting: Positive (+) pin = ANX5, negative (-) pin = ANX4

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Input voltage	V_{IN}		AV_{SS}		AV_{DD}	V
Output voltage	V_{OL}	I_L= -1 mA, AV_{DD} = 2.7 to 5.5 V		AV_{SS} +0.02	AV_{SS} +0.07	V
	V_{OH}	I_L= 1 mA, AV_{DD} = 2.7 to 5.5 V	AV_{DD} -0.15	AV_{DD} -0.02		V
Maximum output current	I_{OUT}	4.5 V ≤ AV_{DD} ≤ 5.5 V	±10			mA
		2.7 V ≤ AV_{DD} ≤ 5.5 V	±5			mA
Input-referred offset voltage	V_{OFF}	T_A = 25°C without trimming I_L = 0 mA, V_{COM} = 1.0 V		±1	±4	mV
		T_A = 25°C with trimming I_L = 0 mA, V_{COM} = 1.0 V			±0.35	mV
Temperature coefficient for inputreferred offset voltage	V_{OTC}	I_L = 0 mA		(±2)	(±8)	μV/°C
Slew rate	SR1	Normal mode C_L = 50 pF, R_L = 10 kΩ		(0.1)		V/μs
	SR2	High-speed mode C_L = 50 pF, R_L = 10 kΩ		(0.8)		V/μs
Gain bandwidth	GBW1	Normal mode C_L = 50 pF, R_L = 10 kΩ		(350)		kHz
	GBW2	High-speed mode C_L = 50 pF, R_L = 10 kΩ		(1.8)		MHz
Phase margin	θM1	Normal mode C_L = 50 pF, R_L = 10 kΩ		(70)		deg
	θM2	High-speed mode C_L = 50 pF, R_L = 10 kΩ		(60)		deg
Settling time	tset1	Normal mode C_L = 50 pF, R_L = 10 kΩ		(20)		μs
	tset2	High-speed mode C_L = 50 pF, R_L = 10 kΩ		(10)		μs
Peak-to-peak voltage noise	Enb	0.1 to 10 Hz Normal mode C_L = 50 pF, R_L = 10 kΩ		(2.0)		μVrms
Input-referred noise	En	f = 1 kHz Normal mode C_L = 50 pF, R_L = 10 kΩ		(70)		nV/√Hz
Common mode rejection ratio	CMRR	f = 1 KHz, C_L = 50 pF, R_L = 10 kΩ		(70)		dB
Power supply rejection ratio	PSRR	2.7 V ≤ AV_{DD} ≤ 5.5 V C_L = 50 pF, R_L = 10 kΩ		(62)		dB

(Remarks are listed on the next page.)

Remark 1. In the ratings column, values in parentheses are the target design values and therefore are not tested for shipment.
Remark 2. The TYP. conditions are the conditions when T_A = 25°C and AV_{DD} = 5.0 V.
Remark 3. Unless otherwise specified, offset trimming has proceeded.
Remark 4. Unless otherwise specified, values are for operation in normal mode.

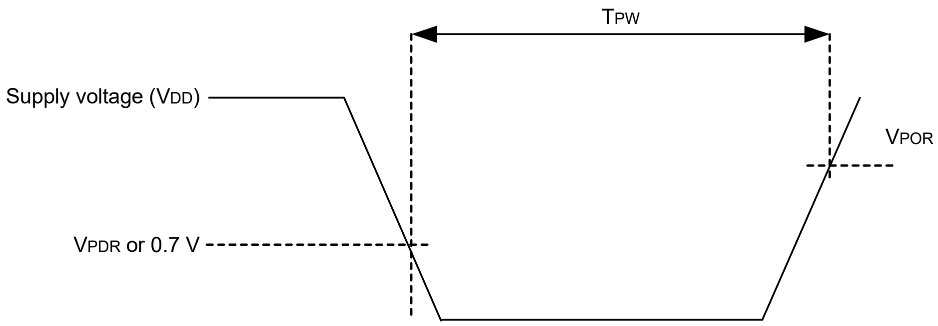
34.6.7 POR characteristics

(T_A = -40 to +125°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Power on/down reset threshold	V_{POR}	Voltage threshold on V_{DD} rising	1.48	1.56	1.62	V
	V_{PDR}	Voltage threshold on V_{DD} falling Note 1	1.47	1.55	1.61	V
Minimum pulse width Note 2	T_{PW}		300			µs

Note 1. However, when the operating voltage falls while the LVD is off, enter STOP mode, or enable the reset status using the external reset pin before the voltage falls below the operating voltage range shown in 34.4 AC Characteristics.

Note 2. Minimum time required for a POR reset when V_{DD} exceeds below V_{PDR}. This is also the minimum time required for a POR reset from when V_{DD} exceeds below 0.7 V to when V_{DD} exceeds V_{POR} while STOP mode is entered or the main system clock is stopped through setting bit 0 (HIOSTOP) and bit 7 (MSTOP) in the clock operation status control register (CSC).



34.6.8 LVD characteristics

(1) LVD detection voltage in reset mode and interrupt mode

(T_A = -40 to +125°C, $V_{PDR} \leq AV_{DD} = V_{DD} \leq 5.5$ V, $AV_{SS} = V_{SS} = 0$ V)

Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Voltage detection threshold	Supply voltage level	V_{LVD0}	Rising edge	4.62	4.74	4.94	V
			Falling edge	4.52	4.64	4.84	V
		V_{LVD1}	Rising edge	4.50	4.62	4.82	V
			Falling edge	4.40	4.52	4.71	V
		V_{LVD2}	Rising edge	4.30	4.42	4.61	V
			Falling edge	4.21	4.32	4.51	V
		V_{LVD3}	Rising edge	3.13	3.22	3.39	V
			Falling edge	3.07	3.15	3.31	V
		V_{LVD4}	Rising edge	2.95	3.02	3.17	V
			Falling edge	2.89	2.96	3.09	V
		V_{LVD5}	Rising edge	2.74	2.81	2.95	V
			Falling edge	2.68	2.75	2.88	V
		V_{LVD6}	Rising edge	2.55	2.61	2.74	V
			Falling edge	2.49	2.55	2.67	V
Minimum pulse width		t_{LW}		300			μs
Detection delay time						300	μs

(2) LVD detection voltage in interrupt & reset mode

(T_A = -40 to +125°C, $V_{PDR} \leq AV_{DD} = V_{DD} \leq 5.5$ V, $AV_{SS} = V_{SS} = 0$ V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Voltage detection threshold	V_{LVDD6}	V_{POC2}, V_{POC1}, V_{POC0} = 0, 0, 0, falling reset voltage			2.49	2.55	2.67	V
	V_{LVDD4}		LVIS1, LVIS0 = 0, 1	Rising release reset voltage	2.95	3.02	3.17	V
				Falling interrupt voltage	2.89	2.96	3.09	V
	V_{LVDD3}		LVIS1, LVIS0 = 0, 0	Rising release reset voltage	3.13	3.22	3.39	V
				Falling interrupt voltage	3.07	3.15	3.31	V
	V_{LVDD5}	V_{POC2}, V_{POC1}, V_{POC0} = 0, 0, 1, falling reset voltage			2.68	2.75	2.88	V
	V_{LVDD2}		LVIS1, LVIS0 = 0, 0	Rising release reset voltage	4.30	4.42	4.61	V
				Falling interrupt voltage	4.21	4.32	4.51	V
	V_{LVDD5}	V_{POC2}, V_{POC1}, V_{POC0} = 0, 1, 0, falling reset voltage			2.68	2.75	2.88	V
	V_{LVDD1}		LVIS1, LVIS0 = 0, 0	Rising release reset voltage	4.50	4.62	4.82	V
				Falling interrupt voltage	4.40	4.52	4.71	V
	V_{LVDD5}	V_{POC2}, V_{POC1}, V_{POC0} = 0, 1, 1, falling reset voltage			2.68	2.75	2.88	V
	V_{LVDD3}		LVIS1, LVIS0 = 0, 1	Rising release reset voltage	3.13	3.22	3.39	V
				Falling interrupt voltage	3.07	3.15	3.31	V
	V_{LVDD0}		LVIS1, LVIS0 = 0, 0	Rising release reset voltage	4.62	4.74	4.94	V
				Falling interrupt voltage	4.52	4.64	4.84	V

34.6.9 Power supply voltage rising slope characteristics

(T_A = -40 to +125°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Power supply voltage rising slope	S_{VDD}				50	V/ms

Caution **Make sure to keep the internal reset state by the LVD circuit or an external reset until V_{DD} reaches the operating voltage range shown in 34.4 AC Characteristics.**

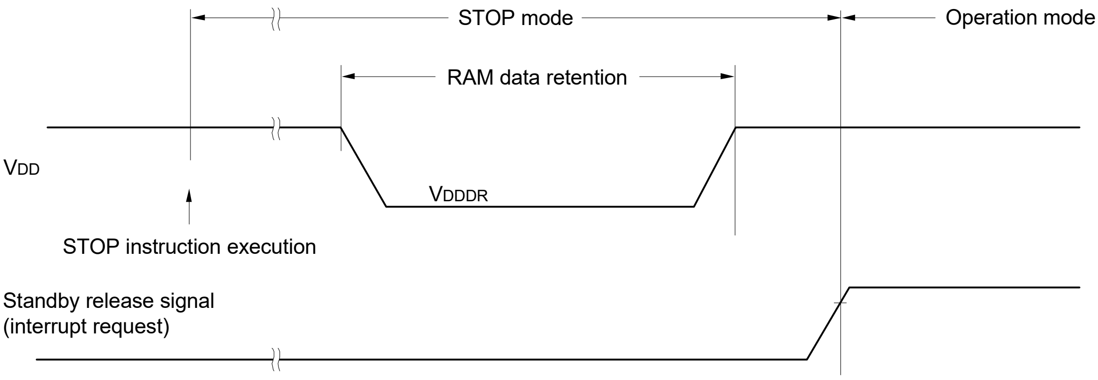
34.7 RAM Data Retention Characteristics

(T_A = -40 to +125°C, V_{SS} = 0 V))

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		1.47 Notes 1, 2		5.5	V

Note 1. The value depends on the POR detection voltage. When the voltage drops, the RAM data is retained before a POR reset is effected, but RAM data is not retained when a POR reset is effected.

Note 2. Enter STOP mode before the supply voltage falls below the recommended operating voltage.



34.8 Flash Memory Programming Characteristics

(T_A = -40 to +125°C Note 4, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
System clock frequency	f_{CLK}	2.4 V ≤ V_{DD} ≤ 5.5 V	1		24	MHz
Number of code flash rewrites Notes 1, 2, 3	C_{erwr}	Retained for 20 years T_A = 85°C Note 5	1,000			Times
Number of data flash rewrites Notes 1, 2, 3		Retained for 1 year T_A = 25°C Note 5		1,000,000		
		Retained for 5 years T_A = 85°C Note 5	100,000			
		Retained for 20 years T_A = 85°C Note 5	10,000			

Note 1. 1 erase + 1 write after the erase is regarded as 1 rewrite. The retaining years are until next rewrite after the rewrite.

Note 2. When using flash memory programmer and Renesas Electronics self-programming library

Note 3. These are the characteristics of the flash memory and the results obtained from reliability testing by Renesas Electronics Corporation.

Note 4. The range is from T_A = -40 to +105°C when if the flash memory programmer is in use.

Note 5. This temperature is the average value at which data are retained.

34.9 Dedicated Flash Memory Programmer Communication (UART)

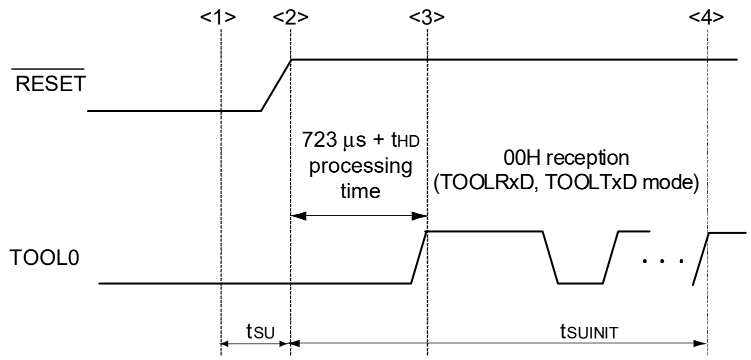
(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate		During serial programming	115,200		1,000,000	bps

34.10 Timing for Switching Flash Memory Programming Modes

(T_A = -40 to +105°C, 2.4 V ≤ AV_{DD} = V_{DD} ≤ 5.5 V, AV_{SS} = V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
How long from when an external reset ends until the initial communication settings are specified	t_{SUINIT}	POR and LVD reset must end before the external reset ends.			100	ms
How long from when the TOOL0 pin is placed at the low level until an external reset ends	t_{SU}	POR and LVD reset must end before the external reset ends.	10			μs
How long the TOOL0 pin must be kept at the low level after an external reset ends (excluding the processing time of the firmware to control the flash memory)	t_{HD}	POR and LVD reset must end before the external reset ends.	1			ms



<1> The low level is input to the TOOL0 pin.
<2> The external reset ends (POR and LVD reset must end before the external reset ends).
<3> The TOOL0 pin is set to the high level.
<4> Setting of the flash memory programming mode by UART reception and complete the baud rate setting.

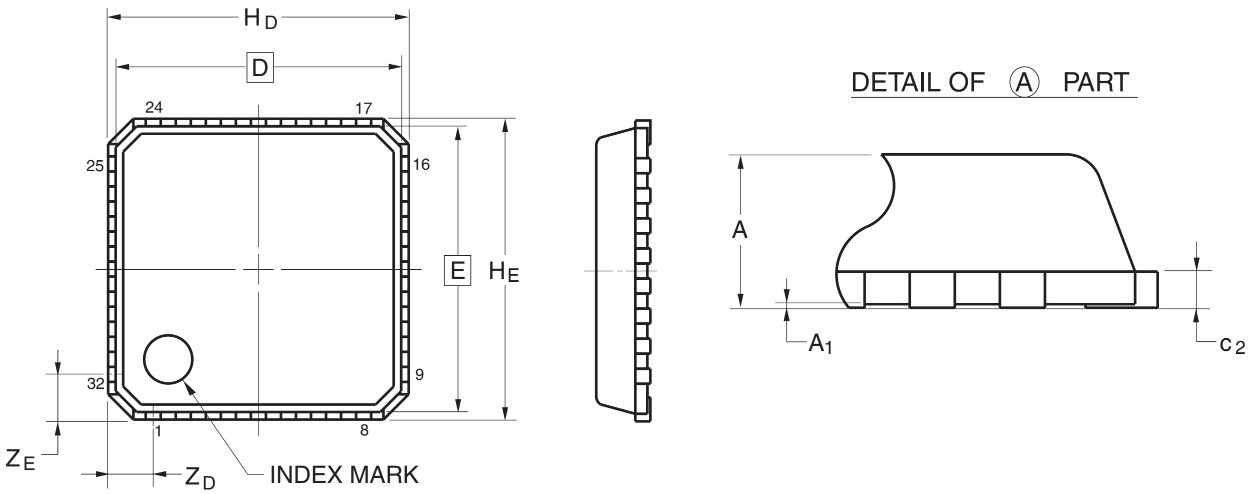
Remark t_{SUINIT}: The segment shows that it is necessary to finish specifying the initial communication settings within 100 ms from when the external resets end.
t_{SU}: How long from when the TOOL0 pin is placed at the low level until a pin reset ends
t_{HD}: How long to keep the TOOL0 pin at the low level from when the external resets end (excluding the processing time of the firmware to control the flash memory)

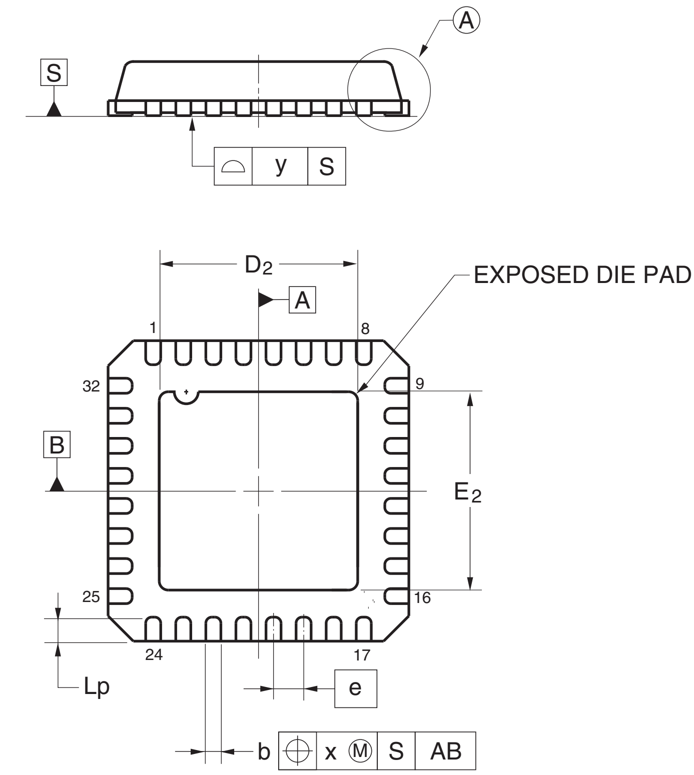
CHAPTER 35 PACKAGE DRAWINGS

35.1 32-pin products

R5F11CBCGNA, R5F11CBCMNA

JEITA Package code	RENESAS code	Previous code	MASS(TYP.)[g]
P-HVQFN32-5x5-0.50	PVQN0032KE-A	P32K9-50B-BAH	0.058



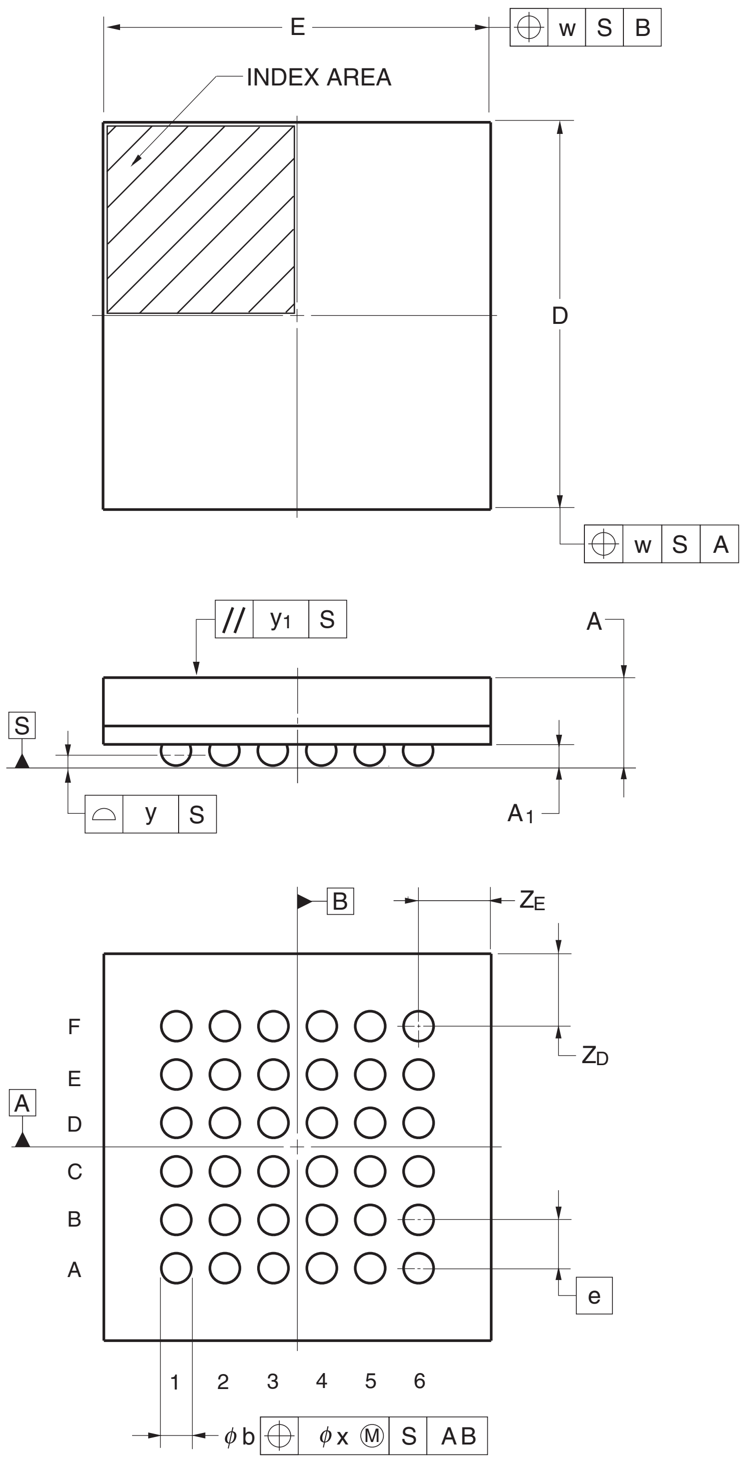


Referance Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	4.75	—
E	—	4.75	—
A	—	—	0.90
A_1	0.00	—	—
b	0.20	0.25	0.30
e	—	0.50	—
Lp	0.30	0.40	0.50
x	—	—	0.10
y	—	—	0.05
H_D	4.95	5.00	5.05
H_E	4.95	5.00	5.05
Z_D	—	0.75	—
Z_E	—	0.75	—
c_2	0.19	0.20	0.21
D_2	—	3.30	—
E_2	—	3.30	—

35.2 36-pin products

R5F11CCCGBG, R5F11CCCMBG

JEITA Package code	RENESAS code	Previous code	MASS(TYP.)[g]
P-TFBGA36-4x4-0.50	PTBG0036KA-A	P36F1-50-AA6	0.027



Referance Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	3.90	4.00	4.10
E	3.90	4.00	4.10
A	—	—	1.10
A_1	0.17	0.22	0.27
e	—	0.50	—
b	0.26	0.31	0.36
x	—	—	0.05
y	—	—	0.08
y_1	—	—	0.20
Z_D	—	0.75	—
Z_E	—	0.75	—
w	—	—	0.20

APPENDIX A REVISION HISTORY

A.1 Major Revisions in This Edition

Page	Description	Classification
CHAPTER 1 OUTLINE		
p.4	Modification of **Figure 1 - 1 Part Number, Memory Size, and Package of RL78/I1E**	(c)
p.4	Modification of Ordering Part Number in **1.2 Ordering Information**	(c)
CHAPTER 12 WATCHDOG TIMER		
p.321	Addition of Note in **Table 12 - 4 Setting Window Open Period of Watchdog Timer**	(c)
p.321	Addition of table in **Table 12 - 4 Setting Window Open Period of Watchdog Timer**	(c)
CHAPTER 27 SAFETY FUNCTIONS		
p.745	Modification of **27.1 Overview of Safety Functions**	(c)
p.745	Modification of Remark in **27.1 Overview of Safety Functions**	(c)
p.750	Modification of **27.3.2 CRC operation function (general-purpose CRC)**	(c)
p.755	Modification of **27.3.4 RAM guard function**	(c)
p.756	Modification of **27.3.5 SFR guard function**	(c)

Remark "Classification" in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note,
(d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

A.2 Revision History of Preceding Editions

(1/9)

Edition	Description	Chapter
Rev.1.00	Change of 1.1 Features	CHAPTER 1 OUTLINE
	Change of Figure 1 - 1 Part Number, Memory Size, and Package of RL78/I1E	
	Change of 1.6 Outline of Functions	
	Change of 2.2.1 Alternate functions other than AFE	CHAPTER 2 PIN FUNCTIONS
	Change of 2.2.2 AFE pin functions	
	Change of Table 2 - 3 Connection of Unused Pins	
	Change of caution 2 in Table 3 - 4 Internal RAM Capacity	CHAPTER 3 CPU ARCHITECTURE
	Deletion of description in 3.3 Instruction Address Addressing	
	Deletion of description in 3.4 Addressing for Processing Data Addresses	
	Change of caution 5 in Figure 5 - 4 Format of Clock operation status control register (CSC)	CHAPTER 5 CLOCK GENERATOR
	Addition of caution 4 in Figure 5 - 11 Format of High-speed on-chip oscillator frequency select register (HOCODIV)	
	Addition of caution 1, 2 in Figure 5 - 12 Format of PLL control register (DSCCTL)	
	Change of Figure 5 - 20 CPU Clock Status Transition Diagram	
	Change of Table 5 - 6 CPU Clock Transition and SFR Register Setting Examples (3/5)	
	Change of Table 5 - 6 CPU Clock Transition and SFR Register Setting Examples (4/5)	
	Change of Table 5 - 6 CPU Clock Transition and SFR Register Setting Examples (5/5)	
	Change of Table 5 - 9 Changing CPU Clock	
	Change of description in 5.6.6 Conditions before clock oscillation is stopped	
	Change of description in Timer array unit	CHAPTER 6 TIMER ARRAY UNIT
	Change of description in 6.1.3 8-bit timer operation function (available for channels 1 and 3 of unit 0)	
	Change of Figure 6 - 2 Internal Block Diagram of Channel 0 of Timer Array Unit 0	
	Change of Figure 6 - 6 Entire Configuration of Timer Array Unit 1	
	Change of Figure 6 - 8 Internal Block Diagram of Channel 1 of Timer Array Unit 1	
	Change of description in 6.2.2 Timer data register mn (TDRmn)	
	Change of description in 6.3.3 Timer mode register mn (TMRmn)	
	Change of Figure 6 - 15 Format of Timer mode register mn (TMRmn) (1/4)	
	Change of Figure 6 - 16 Format of Timer mode register mn (TMRmn) (2/4)	
	Change of Figure 6 - 17 Format of Timer mode register mn (TMRmn) (3/4)	
	Change of Figure 6 - 18 Format of Timer mode register mn (TMRmn) (4/4)	
	Change of Figure 6 - 20 Format of Timer channel enable status register m (TEm)	
	Change of description in 6.3.6 Timer channel start register m (TSm)	
	Change of Figure 6 - 21 Format of Timer channel start register m (TSm)	
	Change of caution 3 in Figure 6 - 21 Format of Timer channel start register m (TSm)	
	Change of description in 6.3.7 Timer channel stop register m (TTm)	
	Change of Figure 6 - 22 Format of Timer channel stop register m (TTm)	
	Change of description in 6.4.2 Basic rules of 8-bit timer operation function (channels 1 and 3 only)	
	Change of description in 6.8.1 Operation as interval timer/square wave output	
	Change of Figure 6 - 50 Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (1/2)	

(2/9)

Edition	Description	Chapter
Rev.1.00	Change of Figure 6 - 51 Operation Procedure of Interval Timer/Square Wave Output Function (1/2)	CHAPTER 6 TIMER ARRAY UNIT
	Change of Note 1 and 2 in Figure 6 - 52 Operation Procedure of Interval Timer/Square Wave Output Function (2/2)	
	Change of description in 6.8.2 Operation as external event counter	
	Change of Figure 6 - 55 Example of Set Contents of Registers in External Event Counter Mode (1/2)	
	Change of Figure 6 - 59 Example of Set Contents of Registers to Measure Input Pulse Interval	
	Change of Figure 6 - 63 Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width	
	Change of description in 6.8.5 Operation as delay counter	
	Change of Figure 6 - 67 Example of Set Contents of Registers to Delay Counter	
	Change of caution in 6.9.1 Operation as one-shot pulse output function	
	Change of Figure 6 - 72 Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Slave Channel)	
	Change of Figure 6 - 78 Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used	
	Change of Figure 9 - 7 Format of Real-time clock control register 1 (RTCC1) (2/2)	CHAPTER 9 REAL-TIME CLOCK
	Change of Figure 9 - 15 Format of Watch error correction register (SUBCUD)	
	Change of Figure 9 - 16 Format of 16-bit watch error correction register (SUBCUDW)	
	Change of remark 1 in 9.4.6 Example of watch error correction of real-time clock	
	Change of description in 13.1 Functions of Analog Front-End Power Supply Circuit	CHAPTER 13 ANALOG FRONT-END POWER SUPPLY CIRCUIT
	Change of description in 13.3.3 Analog front-end power supply detection register (AFEPWD)	
	Change of description in 13.6.1 Overview of internal power supply circuit (REGA)	
	Change of Figure 13 - 9 Flowchart for Powering on the Analog Front-End (AFE) Power Supply	
	Change of Figure 13 - 10 Flowchart for Powering off the Analog Front-End (AFE) Power Supply	
	Change of Figure 13 - 11 Timing for Power Supply Startup Sequence	
	Change of description in 14.3.3 Registers controlling input multiplexers	CHAPTER 14 24-BIT ΔΣ A/D CONVERTER WITH PROGRAMMABLE GAIN INSTRUMENTATION AMPLIFIER
	Change of description in 14.4.6 Registers controlling the programmable gain instrumentation amplifier (PGA)	
	Change of Figure 14 - 11 Format of Disconnection Detection Setting Register (PGABOD)	
	Change of Table 14 - 1 Voltage Input to the 24-bit ΔΣ A/D Converter and A/D Conversion Result	
	Change of description in 14.5.4 Registers controlling the 24-bit ΔΣ A/D converter	
	Change of Figure 14 - 18 Format of ΔΣ A/D Converter Mode Register (DSADMR)	
	Change of Figure 14 - 21 Format of Input Multiplexer x (x = 0 to 4) Setting Register 2 (PGAxCTL2)	

(3/9)

Edition	Description	Chapter
Rev.1.00	Change of Figure 14 - 22 Correlation of the Number of Levels (Register Value) and the Number of A/D Conversions	CHAPTER 14 24-BIT ΔΣ A/D CONVERTER WITH PROGRAMMABLE GAIN INSTRUMENTATION AMPLIFIER
	Change of Figure 14 - 23 Format of Input Multiplexer x (x = 0 to 4) Setting Register 3 (PGAxCTL3)	
	Change of description in 14.5.5 Control of ΔΣ A/D converter (AUTOSCAN)	
	Change of Figure 14 - 36 AUTOSCAN sequence	
	Change of description in 14.5.7 Configuration of digital filter	
	Change of description in 14.6 Procedure for Controlling 24-bit ΔΣ A/D Converter with Programmable Gain Instrumentation Amplifier	
	Change of Figure 14 - 39 Flowchart for Starting the 24-bit ΔΣ A/D Converter with Programmable Gain Instrumentation Amplifier	
	Change of Figure 14 - 40 Flowchart for A/D Conversion by the 24-bit ΔΣ A/D Converter with Programmable Gain Instrumentation Amplifier	
	Change of Figure 14 - 41 Flowchart for Stopping the 24-bit ΔΣ A/D Converter with Programmable Gain Instrumentation Amplifier	
	Change of Figure 14 - 42 Flowchart of Settings for Measuring the Temperature Sensor by the 24-bit ΔΣ A/D Converter with Programmable Gain Instrumentation Amplifier	
	Change of 14.7 Cautions for the 24-bit ΔΣ A/D Converter with Programmable Gain Instrumentation Amplifier	
	Change of Figure 16 - 4 Timing Chart When A/D Voltage Comparator Is Used	CHAPTER 16 A/D CONVERTER
	Change of description in 16.3.2 A/D converter mode register 0 (ADM0)	
	Change of Figure 16 - 5 A/D Converter Sampling and A/D Conversion Timing (Example for Software Trigger Mode)	
	Change of Figure 16 - 19 Example of Software Trigger Mode (Select Mode, Sequential Conversion Mode) Operation Timing	
	Change of Figure 16 - 20 Example of Software Trigger Mode (Select Mode, One-Shot Conversion Mode) Operation Timing	
	Change of Figure 16 - 21 Example of Software Trigger Mode (Scan Mode, Sequential Conversion Mode) Operation Timing	
	Change of Figure 16 - 22 Example of Software Trigger Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing	
	Change of Figure 16 - 23 Example of Hardware Trigger No-Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing	
	Change of Figure 16 - 24 Example of Hardware Trigger No-Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing	
	Change of Figure 16 - 25 Example of Hardware Trigger No-Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing	
	Change of Figure 16 - 26 Example of Hardware Trigger No-Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing	
	Change of Figure 16 - 27 Example of Hardware Trigger Wait Mode (Select Mode, Sequential Conversion Mode) Operation Timing	
	Change of Figure 16 - 28 Example of Hardware Trigger Wait Mode (Select Mode, One-Shot Conversion Mode) Operation Timing	

(4/9)

Edition	Description	Chapter
Rev.1.00	Change of Figure 16 - 29 Example of Hardware Trigger Wait Mode (Scan Mode, Sequential Conversion Mode) Operation Timing	CHAPTER 16 A/D CONVERTER
	Change of Figure 16 - 30 Example of Hardware Trigger Wait Mode (Scan Mode, One-Shot Conversion Mode) Operation Timing	
	Change of Figure 17 - 1 Example of Using a Configurable Amplifier as a Voltage Follower	CHAPTER 17 CONFIGURABLE AMPLIFIER
	Change of description in 17.1.2 Using the configurable amplifier as a configurable amplifier	
	Change of Figure 17 - 2 Example of Using a Configurable Amplifier as a Cascaded Voltage Follower	
	Change of Figure 17 - 3 Example of Using a Configurable Amplifier as a Programmable Non-inverting Amplifier	
	Change of Figure 17 - 4 Example of Using a Configurable Amplifier as a Programmable Transimpedance Amplifier	
	Change of Figure 17 - 5 Using the Configurable Amplifier as a 12-bit D/A Converter Output Amplifier	
	Change of description in 17.1.4 Offset calibration	
	Change of Table 17 - 1 Registers Controlling the Configurable Amplifier	
	Addition of description in 17.3.2 Analog front-end power supply selection register (AFEPWS)	
	Change of description in 17.3.4 Configurable amplifier 0 output selection register (AMP0S0)	
	Change of Figure 17 - 13 Format of Configurable Amplifier 0 Output Selection Register (AMP0	
	Change of description in 17.3.5 Configurable amplifier 1 output selection register (AMP1S0)	
	Change of Figure 17 - 14 Format of Configurable Amplifier 1 Output Selection Register (AMP1S0)	
	Change of description in 17.3.6 Configurable amplifier 2 output selection register (AMP2S0)	
	Change of Figure 17 - 15 Format of Configurable Amplifier 2 Output Selection Register (AMP2S0)	
	Change of description in 17.3.7 Configurable amplifier 0 negative input selection register (AMP0S1)	
	Change of Figure 17 - 16 Format of Configurable Amplifier 0 Negative Input Selection Register (AMP0S1)	
	Change of description in 17.3.8 Configurable amplifier 1 negative input selection register (AMP1S1)	
	Change of Figure 17 - 17 Format of Configurable Amplifier 1 Negative Input Selection Register (AMP1S1)	
	Change of description in 17.3.9 Configurable amplifier 2 negative input selection register (AMP2S1)	

(5/9)

Edition	Description	Chapter
Rev.1.00	Change of Figure 17 - 18 Format of Configurable Amplifier 2 Negative Input Selection Register (AMP2S1)	CHAPTER 17 CONFIGURABLE AMPLIFIER
	Change of description in 17.3.10 Configurable amplifier 0 positive input selection register (AMP0S2)	
	Change of Figure 17 - 19 Format of Configurable Amplifier 0 Positive input Selection Register (AMP0S2)	
	Change of description in 17.3.11 Configurable amplifier 1 positive input selection register (AMP1S2)	
	Change of Figure 17 - 20 Format of Configurable Amplifier 1 Positive Input Selection Register (AMP1S2)	
	Change of description in 17.3.12 Configurable amplifier 2 positive input selection register (AMP2S2)	
	Change of Figure 17 - 21 Format of Configurable Amplifier 2 Positive Input Selection Register (AMP2S2)	
	Change of Figure 17 - 25 Procedure for Starting the Configurable Amplifiers	
	Change of Figure 17 - 26 Procedure for Stopping the Configurable Amplifiers	
	Change of 17.4.3 Changing the Configurable amplifier configuration by using switches	
	Change of Figure 17 - 27 Procedure for Switching the Configurable Amplifiers	
	Addition of description in 17.4.4 Changing the operating mode of the configurable amplifier	
	Change of Figure 17 - 29 Offset Trimming Circuit Configuration	
	Change of Figure 17 - 31 Procedure for Setting up the Offset Trimming	
	Change of description in 17.4.7 Analog/digital pins	
	Change of description in 17.5 Cautions for the Configurable Amplifier	
	Change of 18.1 Function of 12-BIT D/A Converter	CHAPTER 18 12-BIT D/A CONVERTER
	Change of Table 18 - 1 Specifications of 12-bit D/A Converter	
	Change of Table 18 - 3 Registers Controlling 12-bit D/A Converter	
	Addition of description in 18.2.2 Analog front-end power supply selection register (AFEPWS)	
	Change of description in 18.2.3 D/A converter mode register 0 (DACM0)	
	Change of description in 18.2.4 D/A converter mode register 1 (DACM1)	
	Change of description in 18.3.3 Procedure for controlling 12-bit D/A converter	
	Change of Figure 18 - 9 Procedure for Starting the 12-bit D/A Converter	
	Change of Figure 18 - 10 Procedure for Stopping the 12-bit D/A Converter	
	Addition of description in 18.3.4 Changing the reference voltage source of the 12-bit D/A converter	
	Change of Figure 20 - 2 Memory Map Example when DTCBAR Register is Set to FBH	CHAPTER 20 DATA TRANSFER CONTROLLER (DTC)
	Addition of description in 20.5.3 DTC pending instruction	
	Addition of caution in 22.4.4 Interrupt servicing during division instruction	CHAPTER 22 INTERRUPT FUNCTIONS
	Change of description in 22.4.5 Interrupt request hold	

(6/9)

Edition	Description	Chapter
Rev.1.00	Change of Table 23 - 2 Operating Statuses in STOP Mode	CHAPTER 23 STANDBY FUNCTION
	Change of description in 27.3.2 CRC operation function (general-purpose CRC)	CHAPTER 27 SAFETY FUNCTIONS
	Change of Figure 27 - 11 Invalid access detection area	
	Change of Table 30 - 1 Wiring Between RL78/I1E and Dedicated Flash Memory Programmer	CHAPTER 30 FLASH MEMORY
	Deletion of CHAPTER 33 INSTRUCTION SET	CHAPTER 33 INSTRUCTION SET
	Change of CHAPTER 33 ELECTRICAL SPECIFICATIONS (G: T_A = -40 to +105°C)	CHAPTER 33 ELECTRICAL SPECIFICATIONS (G: T_A = -40 to +105°C)
	Deletion of caution 1 in CHAPTER 33 ELECTRICAL SPECIFICATIONS (G: T_A = -40 to +105°C)	
	Change of 33.3.2 Supply current characteristics	
	Change of 33.6.1 Programmable gain instrumentation amplifier and 24-bit ΔΣ A/D converter	
	Change of description in 33.6.1 Programmable gain instrumentation amplifier and 24-bit ΔΣ A/D converter	
	Change of 33.6.2 Sensor power supply (SBIAS)	
	Change of 33.6.3 Temperature sensor	
	Change of 33.6.5 12-bit D/A converter	
	Change of 33.6.6 Configurable amplifier	
	Change of CHAPTER 34 ELECTRICAL SPECIFICATIONS (M: T_A = -40 to +125°C)	CHAPTER 34 ELECTRICAL SPECIFICATIONS (M: T_A = -40 to +125°C)
	Deletion of caution 1 in CHAPTER 34 ELECTRICAL SPECIFICATIONS (M: T_A = -40 to +125°C)	
	Change of 34.3.2 Supply current characteristics	
	Change of 34.6.1 Programmable gain instrumentation amplifier and 24-bit ΔΣ A/D converter	
	Change of description in 34.6.1 Programmable gain instrumentation amplifier and 24-bit ΔΣ A/D converter	
	Change of 34.6.2 Sensor power supply (SBIAS)	
	Change of 34.6.3 Temperature sensor	
	Change of 34.6.5 12-bit D/A converter	
	Change of 34.6.6 Configurable amplifier	
	Addition of note 4 in 34.8 Flash Memory Programming Characteristics	
	Change of 34.9 Dedicated Flash Memory Programmer Communication (UART)	

(7/9)

Edition	Description	Chapter
Rev.1.10	Addition of products name in 1.3.1 32-pin products	CHAPTER 1 OUTLINE
	Addition of products name in 1.3.2 36-pin products	
	Change of DTC in 1.6 Outline of Functions	
	Addition of Note1 in 1.6 Outline of Functions	
	Addition of caution in Figure 2-5 Pin Block Diagram of Pin Type 7-1-4	CHAPTER 2 PIN FUNCTIONS
	Addition of caution1,2 in Figure 2-6 Pin Block Diagram of Pin Type 8-1-4	
	Change of Note1 in Figure 3-1 Memory Map (R5F11CxC (x = B, C))	CHAPTER 3 CPU ARCHITECTURE
	Addition of caution in 5.6.2 Example of setting X1 oscillation clock	CHAPTER 5 CLOCK GENERATOR
	Change of description in 14.4.6 Registers controlling the programmable gain instrumentation amplifier (PGA) (4)	CHAPTER 14 24-BIT ΔΣ A/D CONVERTER WITH PROGRAMMABLE GAIN INSTRUMENTATION AMPLIFIER
	Addition of description in 14.5.3 Voltage input to the 24-bit ΔΣ A/D converter and A/D conversion result	
	Change of description in 14.5.4 Registers controlling the 24-bit ΔΣ A/D converter (10)	
	Change of description in Figure 14-24 Format of ΔΣ A/D Converter Conversion Result Register C (DSADCRC)	
	Addition of Note in Figure 14-24 Format of ΔΣ A/D Converter Conversion Result Register C (DSADCRC)	
	Change of description in 14.5.4 Registers controlling the 24-bit ΔΣ A/D converter (14)	
	Change of description in Figure 14-28 Format of ΔΣ A/D Converter Mean Value Register C (DSADMVC)	
	Addition of Note in Figure 14-28 Format of ΔΣ A/D Converter Mean Value Register C (DSADMVC)	
	Addition of caution in 14.5.4 Registers controlling the 24-bit ΔΣ A/D converter (14)	
	Change of Figure 16-31 Setting up Software Trigger Mode	CHAPTER 16 A/D CONVERTER
	Change of Figure 16-32 Setting up Hardware Trigger No-Wait Mode	
	Change of Figure 16-33 Setting up Hardware Trigger Wait Mode	
	Change of Figure 16-34 Setup when internal reference voltage for A/D converter is selected	
	Change of Figure 16-35 Setting up Test Mode	
	Change of Figure 16-39 Flowchart for Setting up SNOOZE Mode	
	Change of description in 16.10 (2) Input range of ANI0 to ANI9 pins	
	Change of Figure 17-25 Procedure for Starting the Configurable Amplifiers	CHAPTER 17 CONFIGURABLE AMPLIFIER
	Change of Figure 17-28 Procedure for Changing the Operating Mode of the Configurable Amplifier	

(8/9)

Edition	Description	Chapter
Rev.1.10	Change of Figure 19-1 Block Diagram of Serial Array Unit 0	CHAPTER 19 SERIAL ARRAY UNIT
	Change of Figure 19-18 Format of Serial output register m (SOm)	
	Addition of caution in19-18 Format of Serial output register m (SOm)	
	Change of Figure 19-20 Examples of Reverse Transmit Data	
	Change of description in 19.5.7 SNOOZE mode function	
	Change of Figure 19-72 Timing Chart of SNOOZE Mode Operation (once startup) (Type 1: DAPmn = 0, CKPmn = 0)	
	Addition of Note in Figure 19-72 Timing Chart of SNOOZE Mode Operation (once startup) (Type 1: DAPmn = 0, CKPmn = 0)	
	Change of Figure 19-73 Flowchart of SNOOZE Mode Operation (once startup)	
	Change of Figure 19-74 Timing Chart of SNOOZE Mode Operation (continuous startup) (Type 1: DAPmn = 0, CKPmn = 0)	
	Addition of Note in Figure 19-74 Timing Chart of SNOOZE Mode Operation (continuous startup) (Type 1: DAPmn = 0, CKPmn = 0)	
	Change of Figure 19-75 Flowchart of SNOOZE Mode Operation (continuous startup)	
	Change of description in 19.7.3 SNOOZE mode function	
	Addition of caution5 in 19.7.3 SNOOZE mode function	
	Change of Figure 19-121 Timing Chart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1)	
	Change of Figure 19-122 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 0)	
	Change of Figure 19-123 Flowchart of SNOOZE Mode Operation (EOCm1 = 0, SSECm = 0/1 or EOCm1 = 1, SSECm = 0)	
	Change of Figure 19-124 Timing Chart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)	
	Change of Figure 19-125 Flowchart of SNOOZE Mode Operation (EOCm1 = 1, SSECm = 1)	
	Change of Figure 19-130 Flowchart for LIN Transmission	
	Change of Figure 19-131 Reception Operation of LIN	
	Change of Figure 19-132 Flowchart for LIN Reception	
	Addition of description in CHAPTER 20 DATA TRANSFER CONTROLLER (DTC)	CHAPTER 20 DATA TRANSFER CONTROLLER (DTC)
	Change of Table 20 -1 DTC Specifications	
	Change of Figure 20-3 Control Data Allocation	
	Addition of Table 20-4 Start Address of Control Data	
	Addition of Figure 20-4 Start Ad <R> dress of Control Data and Vector Table	
	Change of description in 20.4.2 (1) Example of using normal mode - 1	
	Deletion of caution in 24.1 Timing of Reset Operation	CHAPTER 24 RESET FUNCTION
	Change of Figure 24-5 Example of Procedure for Checking Reset Source	
	Change of Note3 and 4 in Figure 25-2 Timing of Generation of Internal Reset Signal by Power-on-Reset Circuit and Voltage Detector (1/3)	CHAPTER 25 POWER-ON-RESET CIRCUIT
	Change of Figure 29-6 Format of Option Byte (000C2H/010C2H)	CHAPTER 29 OPTION BYTE

(9/9)

Edition	Description	Chapter
Rev.1.10	Change of Table 30-1 Wiring Between RL78/I1E and Dedicated Flash Memory Programmer	CHAPTER 30 FLASH MEMORY
	Change of Figure 30-2 Communication with Dedicated Flash Memory Programmer	
	Change of Table 30-2 Pin Connection	
	Change of 33.5.1 (7) Communication at different potential (1.8 V, 2.5 V, 3 V) (CSI mode) (slave mode, SCKp... external clock in input)	CHAPTER 33 ELECTRICAL SPECIFICATIONS (M: T_A = -40 to +105°C)
	Change of 33.7 RAM Data Retention Characteristics	
	Change of 33.8 Flash Memory Programming Characteristics	
	Change of 34.2.2 On-chip oscillator characteristics	CHAPTER 34 ELECTRICAL SPECIFICATIONS (M: T_A = -40 to +125°C)
	Change of 34.5.1 (7) Communication at different potential (1.8 V, 2.5 V, 3 V) (CSI mode) (slave mode, SCKp... external clock in input)	
	Change of 34.6.1 (1) Analog input in differential input mode	
	Change of 34.6.1 (2) Analog input in single-ended input mode	
	Change of 34.7 RAM Data Retention Characteristics	
	Change of 34.8 Flash Memory Programming Characteristics	
Rev.1.20	The module name for 3-wire serial I/O and 3-wire serial were changed to simplified SPI.	All
	The module name for CSI was changed to simplified SPI.	
	Addition of Note in 1.1 Features	CHAPTER 1 OUTLINE
	Modification of Figure 1 - 1 Part Number, Memory Size, and Package of RL78/I1E	
	Modification of 1.2 Ordering Information	
	Addition of Note in 4.4.4 Handling different potential (1.8 V, 2.5 V, 3 V) by using I/O buffers	CHAPTER 4 PORT FUNCTIONS
	Modification of Figure 9 - 7 Format of Real-time clock control register 1 (RTCC1) (2/2)	CHAPTER 9 REAL-TIME CLOCK
	Modification of Caution in Figure 9 - 24 Procedure for Reading Real-time Clock	
	Modification of Caution 1 in Figure 9 - 25 Procedure for Writing Real-time Clock	
	Addition of Note in CHAPTER 19 SERIAL ARRAY UNIT	CHAPTER 19 SERIAL ARRAY UNIT

RL78/I1E User's Manual: Hardware

Publication Date: Rev. 0.51 Nov 6, 2014
Rev. 1.30 Mar 29, 2024

Published by: Renesas Electronics Corporation

RL78/I1E

Renesas Electronics Corporation

R01UH0524EJ0130