User's Manual

16

78K0R/Kx3-A

User's Manual: Hardware

16-Bit Single-Chip Microcontrollers

Renesas Electronics
www.renesas.com

Rev.2.00 May 2010

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of 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. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. 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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may 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.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, 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, please evaluate the safety of the final products or system 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. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

NOTES FOR CMOS DEVICES

(1) 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 VIL (MAX) and VIH (MIN) due to noise, etc., 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 VIL (MAX) and VIH (MIN).

(2) HANDLING OF UNUSED INPUT PINS: Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to VDD or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

(3) PRECAUTION AGAINST ESD: A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that 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 should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

(4) STATUS BEFORE INITIALIZATION: Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.

(5) POWER ON/OFF SEQUENCE: In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current. The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

(6) INPUT OF SIGNAL DURING POWER OFF STATE : Do not input signals or an I/O pull-up power supply while the device is not powered. 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. Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

How to Use This Manual

Readers

This manual is intended for user engineers who wish to understand the functions of the 78K0R/Kx3-A microcontrollers and design and develop application systems and programs for these devices.
The target products are as follows.

- 78K0R/KE3-A: μPD78F1016, 78F1017, 78F1018

Purpose

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

Organization

The manual for the 78K0R/Kx3-A microcontrollers is separated into two parts: this manual and the instructions edition (common to the 78K0R Microcontroller Series).

78K0R/Kx3-A Preliminary User's Manual (This Manual)	**78K0R Microcontroller User's Manual Instructions**
• 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**.
- 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 RA78K0R, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0R.
- To know details of the 78K0R Series instructions:
  → Refer to the separate document **78K0R Microcontroller Instructions User's Manual (U17792E)**.

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.
78K0R/Kx3-A User's Manual	This manual
78K0R Microcontroller Instructions User's Manual	U17792E

Documents Related to Development Tools (Software) (User's Manuals)

Document Name		Document No.
CC78K0R Ver. 2.00 C Compiler	Operation	U18549E
	Language	U18548E
RA78K0R Ver. 1.20 Assembler Package	Operation	U18547E
	Language	U18546E
SM+ System Simulator	Operation	U18601E
	User Open Interface	U18212E
PM+ Ver. 6.30		U18416E
ID78K0R-QB Ver. 3.20 Integrated Debugger	Operation	U17839E

Documents Related to Development Tools (Hardware) (User's Manuals)

Document Name	Document No.
QB-78K0RLX3 In-Circuit Emulator	U19336E
QB-MINI2 On-Chip Debug Emulator with Programming Function	U18371E

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

Document Name		Document No.
PG-FP5 Flash Memory Programmer		U18865E
QB-Programmer Programming GUI	Operation	U18527E

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.
SEMICONDUCTOR SELECTION GUIDE – Products and Packages –	X13769X
Semiconductor Device Mount Manual	**Note**
Quality Grades on NEC Semiconductor Devices	C11531E
NEC Semiconductor Device Reliability/Quality Control System	C10983E
Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD)	C11892E

<R> **Note** See the “Semiconductor Device Mount Manual” website (http://www2.renesas.com/pkg/en/mount/index.html).

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

Table of Contents

CHAPTER 1 OUTLINE

The 78K0R/Kx3-A microcontrollers are 16-bit single-chip microcontrollers that include the 78K0R CPU core and peripheral functions such as ROM/RAM, A/D converter, D/A converter, operational amplifiers, multifunctional serial interfaces, multifunctional timers, real-time counter, and watchdog timer.

1.1 Features

- Minimum instruction execution time can be changed from high speed (0.05 μs: @ 20 MHz operation with high-speed system clock) to ultra low-speed (61 μs: @ 32.768 kHz operation with subsystem clock)
- General-purpose register: 8 bits × 32 registers (8 bits × 8 registers × 4 banks)
- ROM, RAM capacities

Program Memory (ROM)	Data Memory (RAM)	78K0R/Kx3-A
64 KB	4 KB	μPD78F1016
96 KB	6 KB	μPD78F1017
128 KB	7 KB	μPD78F1018

- On-chip internal high-speed oscillation clock
  - 20 MHz internal high-speed s oscillation clock: 20 MHz ± 2.4 %
  - 8 MHz internal high-speed s oscillation clock: 8 MHz ± 2 % (when 1.8 V ≤ V_{DD} < 2.7 V)
  - 1 MHz internal high-speed s oscillation clock: 1 MHz ± 13 %
- On-chip single-power-supply flash memory (with prohibition of chip erase/block erase/writing function)
- Self-programming (with boot swap function/flash shield window function)
- On-chip debug function
- On-chip power-on-clear (POC) circuit and low-voltage detector (LVI)
- On-chip watchdog timer (can operate on dedicated internal low-speed oscillation clock)
- On-chip multiplier/divider (16 bits × 16 bits, 32 bits ÷ 32 bits)
- On-chip clock output/buzzer output controller (output: 2)
- On-chip BCD adjustment
- I/O ports: 53 (N-ch open drain: 2)
- Timer
  - 16-bit timer: 12 ch (input: 8, output: 8)
  - Watchdog timer: 1 ch
  - Real-time counter: 1 ch (output: 2)

- Serial interface
  - CSI: 1 channel/UART: 1 channel
  - CSI: 1 channel/UART: 1 channel/simplified I^2C: 1 channel
  - CSI: 1 channel/UART: 1 channel/simplified I^2C: 1 channel
  - UART (LIN-bus supported): 1 channel
  - Multimaster I^2C: 1 channel
- 12-bit resolution A/D conversion: 12 channels
- 12-bit resolution D/A converter: 2 channels
- Operational amplifier: 3 channels
- On-chip voltage reference (2.0 V/2.5 V)
- DMA controller: 2 channels
- Power supply voltage: V_{DD} = 1.8 to 5.5 V
- Operating ambient temperature: T_A = −40 to +85°C

1.2 Ordering Information

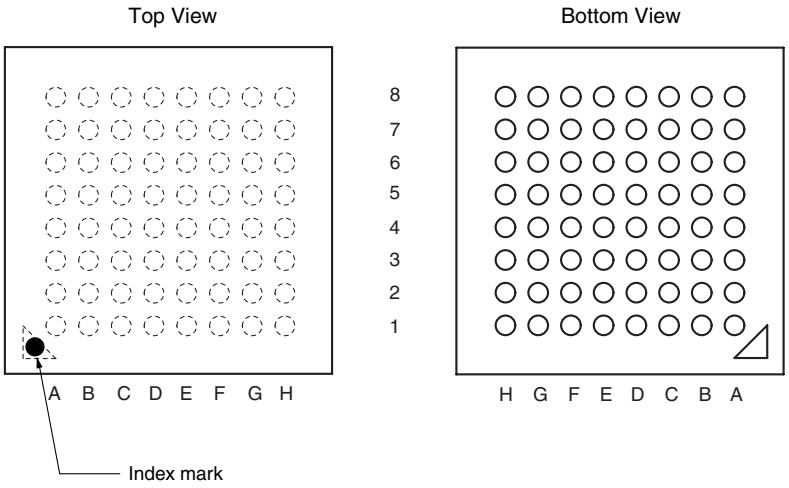
- **Flash memory version (Lead-free products)**

78K0R/Kx3-A microcontrollers	Package	Part Number
78K0R/KE3-A	64-pin plastic FBGA (6x6)	μPD78F1016F1-BA4-A, 78F1017F1-BA4-A, 78F1018F1-BA4-A

Caution The 78K0R/Kx3-A 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.

1.3 Pin Configuration (Top View)

- 64-pin plastic FBGA (6×6)



Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name	Pin No.	Pin Name
A1	P16/INTP10/TI05/TO05	C1	P130	E1	AV_{DD1}	G1	AV_{DD0}
A2	P141	C2	P14/INTP4/SI10/RxD1/SDA10	E2	P157/ANI15/AV_{REFM}	G2	P150/ANI8/AMP2+
A3	P100	C3	P13/TO04/SO10/TxD1	E3	P151/ANI9	G3	P25/ANI5/AMP1+
A4	P57	C4	P54	E4	P00	G4	P22/ANI2/AMP0+
A5	P56	C5	P51/TxD3	E5	P11/INTP6/SI20/RxD2/SDA20	G5	P33/INTP3/TI07/TO07
A6	P61/SDA0	C6	P02	E6	P41/TOOL1	G6	P31/INTP2/TI00/TO03/RTCDIV/RTCCL/PCLBUZ1
A7	P60/SCL0	C7	V_{SS}	E7	$\overline{RESET}$	G7	P80/INTP11/$\overline{SCK00}$
A8	EV_{DD}	C8	P121/X1	E8	FLMD0	G8	P124/XT2
B1	P15/INTP7/$\overline{SCK10}$/SCL10	D1	P111/ANO1	F1	V_{REFOUT}/AV_{REFP}	H1	AV_{SS}
B2	P140	D2	P110/ANO0	F2	P27/ANI7/AMP2O	H2	P26/ANI6/AMP2-
B3	P55	D3	P152/ANI10	F3	P24/ANI4/AMP1O	H3	P23/ANI3/AMP1-
B4	P53/TI04	D4	P01	F4	P21/ANI1/AMP0O	H4	P20/ANI0/AMP0-
B5	P52/TI02	D5	P12/TO02/SO20/TxD2	F5	P10/$\overline{SCK20}$/SCL20	H5	P34/INTP8/TI06/TO06
B6	P50/RxD3	D6	P40/TOOL0	F6	P30/INTP1/TI03/TO00/RTC1HZ	H6	P32/INTP5/TI01/TO01/PCLBUZ0
B7	V_{DD}	D7	REGC	F7	P82/TxD0/SO00	H7	P81/INTP9/RxD0/SI00
B8	EV_{SS}	D8	P122/X2/EXCLK	F8	P123/XT1	H8	P120/INTP0/EXLVI

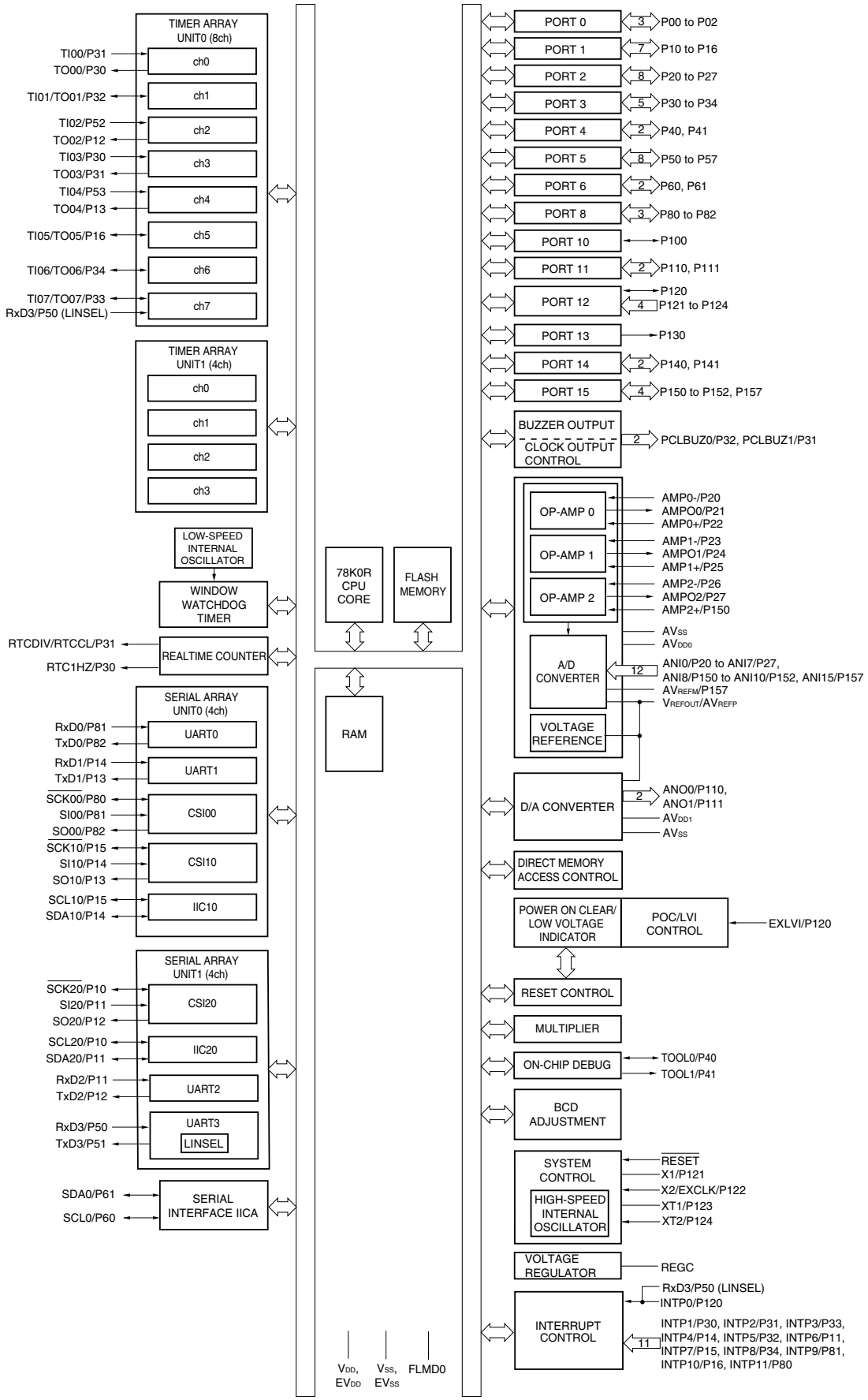
Cautions 1. Make AV_{SS} the same potential as V_{SS}.

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

Pin Identification

AMP0- to AMP2- :	Amplifier Input Minus
AMP0+ to AMP2+ :	Amplifier Input Plus
AMP0O to AMP2O :	Amplifier Output
ANI0 to ANI10, ANI15 :	Analog Input (ADC)
ANO0, ADO1 :	Analog Output (DAC)
AV_{REFM} :	Analog Reference Voltage Minus
AV_{REFP} :	Analog Reference Voltage Plus
AV_{SS} :	Analog Ground
AV_{DD0} :	Analog Power Supply (ADC/VREF/OPAMP)
AV_{DD1} :	Analog Power Supply (DAC)
EV_{DD} :	Power Supply for Port
EV_{SS} :	GND for Port
EXCLK :	External Clock Input (Main system clock)
EXLVI :	External Potential Input for Low Voltage Detector
FLMD0 :	Flash Programming Mode
INTP0 to INTP11 :	External Interrupt Input
P00 to P02 :	Port 0
P10 to P16 :	Port 1
P20 to P27 :	Port 2
P30 to P34 :	Port 3
P40, P41 :	Port 4
P50 to P57 :	Port 5
P60, P61 :	Port 6
P80 to P82 :	Port 8
P100 :	Port 10
P110, P111 :	Port 11
P120 to P124 :	Port 12
P130 :	Port 13
P140, P141 :	Port 14
P150 to P152, P157 :	Port 15
PCLBUZ0, PCLBUZ1 :	Programmable Clock Output /Buzzer Output
REGC :	Regulator Capacitance
$\overline{\text{RESET}}$:	Reset
RTC1HZ :	Real-time Counter Correction Clock (1Hz) Output
RTCCL :	Real-time Counter Clock (32 kHz Original Oscillation) Output
RTCDIV :	Real-time Counter Clock (32 kHz Divided Frequency) Output
RxD0 to RxD3 :	Receive Data
$\overline{\text{SCK00}}$, $\overline{\text{SCK10}}$, $\overline{\text{SCK20}}$:	Serial Clock Input/Output
SCL0, SCL10, SCL20 :	Serial Clock Input/Output
SDA0, SDA10, SDA20 :	Serial Data Input/Output
SI00, SI10, SI20 :	Serial Data Input
SO00, SO10, SO20 :	Serial Data Output
TI00 to TI07 :	Timer Input
TO00 to TO07 :	Timer Output
TOOL0 :	Data Input/Output for Tool
TOOL1 :	Clock Output for Tool
TxD0 to TxD3 :	Transmit Data
V_{DD} :	Power Supply
V_{REFOUT} :	Voltage Reference Output
V_{SS} :	Ground
X1, X2 :	Crystal Oscillator (Main system clock)
XT1, XT2 :	Crystal Oscillator (Subsystem Clock)

1.4 Block Diagram



1.5 Outline of Functions

(1/2)

Item			78K0R/KE3-A		
			μPD78F1016	μPD78F1017	μPD78F1018
Internal memory	Flash memory (self-programming supported)		64 KB	96 KB	128 KB
	RAM		4 KB	6KB	7KB
Memory space			1 MB		
Main system clock	High-speed system clock		X1 (crystal/ceramic) oscillation, external main system clock input (EXCLK) 2 to 20 MHz: V_{DD} = 2.7 to 5.5 V, 2 to 5 MHz: V_{DD} = 1.8 to 5.5 V		
	Internal high-speed oscillation clock		Internal oscillation 1 MHz (TYP.) or 8 MHz (TYP.) : V_{DD} = 1.8 to 5.5 V		
	20 MHz Internal high-speed oscillation clock		Internal oscillation 20 MHz (TYP.) : V_{DD} = 2.7 to 5.5 V		
Subsystem clock			XT1 (crystal) oscillation 32.768 kHz (TYP.) : V_{DD} = 1.8 to 5.5 V		
Internal low-speed oscillation clock (For WDT)			Internal oscillation 30 kHz (TYP.) : V_{DD} = 1.8 to 5.5 V		
General-purpose register			8 bits × 32 registers (8 bits × 8 registers × 4 banks)		
Minimum instruction execution time			0.05 μs (High-speed system clock: f_{MX} = 20 MHz operation)		
			0.125 μs (Internal high-speed oscillation clock: f_{IH} = 8 MHz (TYP.) operation)		
			30.5 μs (Subsystem clock: f_{SUB} = 32.768 kHz operation)		
Instruction set			• 8-bit operation, 16-bit operation • Multiply (16 bits × 16 bits) • Bit manipulation (Set, reset, test, and Boolean operation), etc.		
I/O port	Total		53		
	I/O	CMOS	46		
		N-ch O.D.	2		
	Output	CMOS	1		
	Input	CMOS	4		
Timer			• 16-bit timer: 12 channels • Watchdog timer: 1 channel • Real-time counter: 1 channel		
	Timer outputs		8 (PWM output: 7 (Timer array unit 0))		
	RTC outputs		2 • 1 Hz (Subsystem clock: f_{SUB} = 32.768 kHz) • 512 Hz or 16.384 kHz or 32.768 kHz (Subsystem clock: f_{SUB} = 32.768 kHz)		
Clock output/buzzer output			2 • 2.44 kHz, 4.88 kHz, 9.76 kHz, 1.25 MHz, 2.5 MHz, 5 MHz, 10 MHz (Peripheral hardware clock: f_{MAIN} = 20 MHz operation) • 256 Hz, 512 Hz, 1.024 kHz, 2.048 kHz, 4.096 kHz, 8.192 kHz, 16.384 kHz, 32.768 kHz (Subsystem clock: f_{SUB} = 32.768 kHz operation)		
A/D converter			12-bit resolution × 12 channels		
D/A converter			12-bit resolution × 2 channels		

(2/2)

Item		78K0R/KE3-A		
		μPD78F1016	μPD78F1017	μPD78F1018
Operational amplifier		3 channels		
Voltage reference		2.0 V/2.5 V		
Serial interface	UART supporting LIN-bus	1		
	CSI/UART/ simplified I^2C	2		
	CSI/UART	1		
	Multimaster I^2C	1		
Multiplier/divider		16 bits × 16 bits = 32 bits (multiplication), 32 bits ÷ 32 bits = 32 bits, 32-bit remainder (division)		
DMA controller		2 channels		
Vectored interrupt sources	Internal	33		
	External	12		
Reset		• Reset by $\overline{\text{RESET}}$ pin • Internal reset by watchdog timer • Internal reset by power-on-clear • Internal reset by low-voltage detector • Internal reset by illegal instruction execution[Note]		
Power-on-clear circuit		• Power-on-reset: 1.61 ±0.09 V • Power-down-reset: 1.59 ±0.09 V		
Low-voltage detector	V_{DD} voltage detector	1.91 V to 4.22 V (16 stages)		
	EXLVI voltage detector	1.21 V		
On-chip debug function		Provided		
BCD adjustment				
Power supply voltage		V_{DD} = 1.8 to 5.5 V		
Operating ambient temperature		T_A = −40 to +85°C		
Package		64-pin plastic FBGA (6x6)		

Note 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.

CHAPTER 2 PIN FUNCTIONS

2.1 Pin Function List

There are four types of pin I/O buffer power supplies: AV_{DD0}, AV_{DD1}, EV_{DD}, and V_{DD}. The relationship between these power supplies and the pins is shown below.

Table 2-1. Pin I/O Buffer Power Supplies

Power Supply	Corresponding Pins
AV_{DD0}	P20 to P27, P150 to P152, and P157
AV_{DD1}	P110 and P111
EV_{DD}	• Port pins other than P20 to P27, P110, P111, P150 to P152, and P157 • $\overline{RESET}$ and FLMD0 pins
V_{DD}	Pins other than port , $\overline{RESET}$, and FLMD0 pins

(1) Port functions (1/2)

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 3-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	–
P01				–
P02				–
P10	I/O	Port 1. 7-bit I/O port. Input/output can be specified in 1-bit units. Input of P10, P11, P14 and P15 can be set to TTL buffer. Output of P10 to P15 can be set to N-ch open-drain output (V_{DD} tolerance). Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	$\overline{SCK20}$/SCL20
P11				SI20/RxD2/SDA20/ INTP6
P12				SO20/TxD2/TO02
P13				SO10/TxD1/TO04
P14				SI10/RxD1/SDA10/ INTP4
P15				$\overline{SCK10}$/SCL10/INTP7
P16				TI05/TO05/INTP10
P20	I/O	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI0/AMP0-
P21				ANI1/AMP0O
P22				ANI2/AMP0+
P23				ANI3/AMP1-
P24				ANI4/AMP1O
P25				ANI5/AMP1+
P26				ANI6/AMP2-
P27				ANI7/AMP2O
P30	I/O	Port 3. 5-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI03/TO00/RTC1HZ/ INTP1
P31				TI00/TO03/RTCDIV/ RTCCL/PCLBUZ1/ INTP2
P32				TI01/TO01/INTP5/ PCLBUZ0
P33				TI07/TO07/INTP3
P34				TI06/TO06/INTP8
P40 Note	I/O	Port 4. 2-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TOOL0
P41				TOOL1

Note If on-chip debugging is enabled by using an option byte, be sure to pull up the P40/TOOL0 pin externally.

(1) Port functions (2/2)

Function Name	I/O	Function	After Reset	Alternate Function
P50	I/O	Port 5. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	RxD3
P51				TxD3
P52				TI02
P53				TI04
P54 to P57				–
P60	I/O	Port 6. 2-bit I/O port. Output is N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units.	Input port	SCL0
P61				SDA0
P80	I/O	Port 8. 3-bit I/O port. Inputs/output can be specified in 1-bit units. Output of P80 and P82 can be set to N-ch open-drain output (V_{DD} tolerance). Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	$\overline{SCK00}$/INTP11
P81				RxD0/SI00/INTP9
P82				TxD0/SO00
P100	I/O	Port 10. 1-bit I/O port. Inputs/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	–
P110	I/O	Port 11. 2-bit I/O port. Inputs/output can be specified in 1-bit units.	Input port	ANO0
P111				ANO1
P120	I/O	Port 12. 1-bit I/O port and 4-bit input port. For only P120, input/output can be specified in 1-bit units. For only P120, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP0/EXLVI
P121	Input			X1
P122				X2/EXCLK
P123				XT1
P124				XT2
P130	Output	Port 13. 1-bit output port.	Output port	–
P140, P141	I/O	Port 14. 2-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	–
P150	I/O	Port 15. 4-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI8/AMP2+
P151				ANI9
P152				ANI10
P157				ANI15/AV_{REFM}

(2) Non-port functions (1/4)

Function Name	I/O	Function	After Reset	Alternate Function
ANI0	Input	A/D converter analog input	Digital input port	P20/AMP0-
ANI1				P21/AMP0O
ANI2				P22/AMP0+
ANI3				P23/AMP1-
ANI4				P24/AMP1O
ANI5				P25/AMP1+
ANI6				P26/AMP2-
ANI7				P27/AMP2O
ANI8				P150/AMP2+
ANI9				P151
ANI10				P152
ANI15				P157/AV_{REFM}
AMP0-	Input	Operational amplifier input (negative side)	Digital input port	P20/ANI0
AMP1-				P23/ANI3
AMP2-				P26/ANI6
AMP0+	Input	Operational amplifier input (positive side)	Digital input port	P22/ANI2
AMP1+				P25/ANI5
AMP2+				P150/ANI8
AMP0O	Output	Operational amplifier output	Digital input port	P21/ANI1
AMP1O				P24/ANI4
AMP2O				P27/ANI7
AV_{REFM}	Input	Analog negative reference voltage input	Digital input port	P157/ANI15
AV_{REFP}		Analog positive reference voltage input	Input	V_{REFOUT}
V_{REFOUT}	Output	Analog reference voltage output	Input	AV_{REFP}
ANO0	Output	D/A converter analog output	Input port	P110
ANO1				P111
EXLVI	Input	Potential input for external low-voltage detection	Input port	P120/INTP0

(2) Non-port functions (2/4)

Function Name	I/O	Function	After Reset	Alternate Function
INTP0	Input	External interrupt request input for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified	Input port	P120/EXLVI
INTP1				P30/TI03/TO00/ RTC1HZ
INTP2				P31/TI00/TO03/ RTCDIV/RTCCL/ PCLBUZ1
INTP3				P33/TI07/TO07
INTP4				P14/SI10/RxD1/ SDA10
INTP5				P32/TI01/TO01/ PCLBUZ0
INTP6				P11/SI20/RxD2/ SDA20
INTP7				P15/$\overline{\text{SCK10}}$/SCL10
INTP8				P34/TI06/TO06
INTP9				P81/RxD0/SI00
INTP10				P16/TI05/TO05
INTP11				P80/$\overline{\text{SCK00}}$
PCLBUZ0	Output	Clock output/buzzer output	Input port	P32/TI01/TO01/ INTP5
PCLBUZ1				P31/TI00/TO03/ RTCDIV/RTCCL/INTP2
REGC	–	Connecting regulator output (2.4 V) stabilization capacitance for internal operation. Connect to V_{SS} via a capacitor (0.47 to 1 μF).	–	–
RTCDIV	Output	Real-time counter clock (32 kHz divided frequency) output	Input port	P31/TI00/TO03/ PCLBUZ1/RTCCL/ INTP2
RTCCL	Output	Real-time counter clock (32 kHz original oscillation) output	Input port	P31/TI00/TO03/ PCLBUZ1/RTCDIV/ INTP2
RTC1HZ	Output	Real-time counter correction clock (1 Hz) output	Input port	P30/TI03/TO00/ INTP1
$\overline{\text{RESET}}$	Input	System reset input	–	–
RxD0	Input	Serial data input to UART0	Input port	P81/SI00/INTP9
RxD1		Serial data input to UART1		P14/SI10/SDA10/ INTP4
RxD2		Serial data input to UART2		P11/SI20/SDA20/ INTP6
RxD3		Serial data input to UART3		P50

(2) Non-port functions (3/4)

Function Name	I/O	Function	After Reset	Alternate Function
$\overline{\text{SCK00}}$	I/O	Clock input/output for CSI00	Input port	P80/INTP11
$\overline{\text{SCK10}}$		Clock input/output for CSI10		P15/SCL10/INTP7
$\overline{\text{SCK20}}$		Clock input/output for CSI20		P10/SCL20
SCL0	I/O	Clock input/output for I^2C	Input port	P60
SCL10	I/O	Clock input/output for simplified I^2C	Input port	P15/$\overline{\text{SCK10}}$/INTP7
SCL20				P10/$\overline{\text{SCK20}}$
SDA0	I/O	Serial data I/O for I^2C	Input port	P61
SDA10	I/O	Serial data I/O for simplified I^2C	Input port	P14/SI10/RxD1/ INTP4
SDA20				P11/SI20/RxD2/ INTP6
SI00	Input	Serial data input to CSI00	Input port	P81/RxD0/INTP9
SI10		Serial data input to CSI10		P14/RxD1/SDA10/ INTP4
SI20		Serial data input to CSI20		P11/RxD2/SDA20/ INTP6
SO00	Output	Serial data output from CSI00	Input port	P82/TxD0
SO10		Serial data output from CSI10		P13/TxD1/TO04
SO20		Serial data output from CSI20		P12/TxD2/TO02
TI00	Input	External count clock input to 16-bit timer 00	Input port	P31/TO03/RTCDIV/ RTCCL/PCLBUZ1/ INTP2
TI01		External count clock input to 16-bit timer 01		P32/TO01/INTP5/ PCLBUZ0
TI02		External count clock input to 16-bit timer 02		P52
TI03		External count clock input to 16-bit timer 03		P30/TO00/RTC1HZ/ INTP1
TI04		External count clock input to 16-bit timer 04		P53
TI05		External count clock input to 16-bit timer 05		P16/TO05/INTP10
TI06		External count clock input to 16-bit timer 06		P34/TO06/INTP8
TI07		External count clock input to 16-bit timer 07		P33/TO07/INTP3
TO00	Output	16-bit timer 00 output	Input port	P30/TI03/RTC1HZ/ INTP1
TO01		16-bit timer 01 output		P32/TI01/INTP5/ PCLBUZ0
TO02		16-bit timer 02 output		P12/SO20/TxD2
TO03		16-bit timer 03 output		P31/TI00/RTCDIV/ RTCCL/PCLBUZ1/ INTP2
TO04		16-bit timer 04 output		P13/SO10/TxD1
TO05		16-bit timer 05 output		P16/TI05/INTP10
TO06		16-bit timer 06 output		P34/TI06/INTP8
TO07		16-bit timer 07 output		P33/TI07/INTP3

(2) Non-port functions (4/4)

Function Name	I/O	Function	After Reset	Alternate Function
TxD0	Output	Serial data output from UART0	Input port	P82/SO00
TxD1		Serial data output from UART1		P13/SO10/TO04
TxD2		Serial data output from UART2		P12/SO20/TO02
TxD3		Serial data output from UART3		P51
X1	–	Resonator connection for main system clock	Input port	P121
X2	–		Input port	P122/EXCLK
EXCLK	Input	External clock input for main system clock	Input port	P122/X2
XT1	–	Resonator connection for subsystem clock	Input port	P123
XT2	–		Input port	P124
V_{DD}	–	Positive power supply (Pins other than port, $\overline{RESET}$, and FLMD0 pins)	–	–
EV_{DD}	–	Positive power supply for $\overline{RESET}$, FLMD0 pins, and port pins other than P20 to P27, P110, P111, P150 to P152, and P157	–	–
AV_{DD0}	–	Positive power supply for P20 to P27, P150 to P152, and P157	–	–
AV_{DD1}	–	Positive power supply for P110 and P111	–	–
V_{SS}	–	Ground potential (Pins other than port , $\overline{RESET}$, and FLMD0 pins)	–	–
EV_{SS}	–	Ground potential for $\overline{RESET}$, FLMD0 pins, and port pins other than P20 to P27, P110, P111, P150 to P152, and P157	–	–
AV_{SS}	–	Ground potential for P20 to P27, P110, P111, P150 to P152, and P157	–	–
FLMD0	–	Flash memory programming mode setting	–	–
TOOL0	I/O	Data I/O for flash memory programmer/debugger	Input port	P40
TOOL1	Output	Clock output for debugger	Input port	P41

2.2 Description of Pin Functions

2.2.1 P00 to P02

P00 to P02 function as an I/O port. P00 to P02 can be set to input or output port in 1-bit units using port mode register 0 (PM0). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 0 (PU0).

2.2.2 P10 to P16

P10 to P16 function as an I/O port. This port can also be used for serial interface clock I/O, data I/O, timer I/O and external interrupt request input.

Input to the P10, P11, P14, and P15 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units 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 using port output mode register 1 (POM1).

The following operation modes can be specified in 1-bit units.

(1) Port mode

P10 to P16 function as an I/O port. P10 to P16 can be set to input or output port in 1-bit units using port mode register 1 (PM1). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 1 (PU1).

(2) Control mode

P10 to P16 function as serial interface clock I/O, data I/O, timer I/O, and external interrupt request input.

(a) $\overline{SCK10}$, $\overline{SCK20}$

These are clock I/O pins of serial interfaces CSI10 and CSI20.

(b) SI10, SI20

These are data input pins of serial interfaces CSI10 and CSI20.

(c) SO10, SO20

These are data output pins of serial interfaces CSI00 and CSI20.

(d) SCL10, SCL20

These are clock I/O pins of serial interfaces IIC10 and IIC20 (simplified I^2Cs).

(e) SDA10, SDA20

These are data I/O pins of serial interfaces IIC10 and IIC20 (simplified I^2Cs).

(f) RxD1, RxD2

These are data input pins of serial interfaces UART1 and UART2.

(g) TxD1, TxD2

These are data output pins of serial interfaces UART1 and UART2.

(h) TI05

This is a input pin of 16-bit timer 05.

(i) TO02, TO04, TO05

These are timer output pins of 16-bit timers 02, 04, and 05.

(j) INTP4, INTP6, INTP7, INTP10

These are external interrupt request input pins for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

Cautions 1. To use P10/SCK20/SCL20, P11/SI20/RxD2/SDA20/INTP6 as a general-purpose port, note the serial array unit 1 setting. For details, refer to Table 14-9 Relationship Between Register Settings and Pins (Channel 0 of unit 1: CSI20, UART2 transmission, IIC20).

2. To use P12/TO02/SO20/TxD2 as a general-purpose port, set bit 2 (TO02) of timer output register 0 (TO0) and bit 2 (TOE02) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting. And as a general-purpose port, note the serial array unit 1 setting. For details of serial array unit 1 setting, refer to Table 14-9 Relationship Between Register Settings and Pins (Channel 0 of unit 1: CSI20, UART2 transmission, IIC20).

3. To use P13/TO04/SO10/TxD1 as a general-purpose port, set bit 4 (TO04) of timer output register 0 (TO0) and bit 4 (TOE04) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting. And as a general-purpose port, note the serial array unit 0 setting. For details of serial array unit 0 setting, refer to Table 14-7 Relationship Between Register Settings and Pins (Channel 2 of unit 0: CSI10, UART1 transmission, IIC10)

4. To use P14/SI10/RxD1/SDA10/INTP4, P15/SCK10/SCL10/INTP7 as a general-purpose port, note the serial array unit 0 setting. For details, refer to Table 14-7 Relationship Between Register Settings and Pins (Channel 2 of unit 0: CSI10, UART1 transmission, IIC10)

5. To use P16/TO05/TI05/INTP10 as a general-purpose port, set bit 5 (TO05) of timer output register 0 (TO0) and bit 5 (TOE05) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.

2.2.3 P20 to P27

P20 to P27 function as an I/O port. This port can also be used for A/D converter analog input, and operational amplifier I/O.

The following operation modes can be specified in 1-bit units.

(1) Port mode

P20 to P27 function as an I/O port. P20 to P27 can be set to input or output port in 1-bit units using port mode register 2 (PM2).

(2) Control mode

P20 to P27 function as A/D converter analog input and operational amplifier I/O.

(a) ANI0 to ANI7

These are A/D converter analog input pins.

(b) AMP0-, AMP1-, AMP2-

These are input voltage pins on the negative sides of operational amplifiers 0 to 2.

(c) AMP0+, AMP1+

These are input voltage pins on the positive sides of operational amplifiers 0 and 1.

(d) AMP0O, AMP1O, AMP2O

These are output pins of operational amplifiers 0 to 2.

Cautions 1. P20/ANI0/AMP0- to P27/ANI7/AMP2O are set in the digital input (general-purpose port) mode after release of reset.

2. When using at least one port of ports P20/ANI0/AMP0- to P27/ANI7/AMP2O as a digital port, set AV_{DD0} to the same potential as EV_{DD} or V_{DD}.

2.2.4 P30 to P34

P30 to P34 function as an I/O port. This port can also be used for timer I/O, real-time counter clock output, correction clock output, clock output/buzzer output, and external interrupt request input.

The following operation modes can be specified in 1-bit units.

(1) Port mode

P30 to P34 function as an I/O port. P30 to P34 can be set to input or output port in 1-bit units using port mode register 3 (PM3). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 3 (PU3).

(2) Control mode

P30 to P34 function as timer I/O, real-time counter clock output, correction clock output, clock output/buzzer output, and external interrupt request input.

(a) TI00, TI01, TI03 TI06, TI07

These are timer input pins of 16-bit timers 00, 01, 03, 06, and 07.

(b) TO00, TO01, TO03, TO06, TO07

These are timer output pins of 16-bit timers 00, 01, 03, 06, and 07.

(c) RTCCL

This is a real-time counter clock (32 kHz, original oscillation) output pin.

(d) RTCDIV

This is a real-time counter clock (32 kHz, divided) output pin.

(e) RTC1HZ

This is a real-time counter correction clock (1 Hz) output pin.

(f) PCLBUZ0, PCLBUZ1

These are clock output/buzzer output pins.

(g) INTP1, INTP2, INTP3, INTP5, INTP8

These are external interrupt request input pins for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

Cautions 1. To use P30/TO00/TI03/RTC1HZ/INTP1 as a general-purpose port, set bit 5 (RCLOE1) of real-time counter control register 0 (RTCC0), bit 0 (TO00) of timer output register 0 (TO0) and bit 0 (TOE00) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.

2. To use P31/TO03/TI00/RTCDIV/RTCCL/PCLBUZ1/INTP2 as a general-purpose port, set bit 4 (RCLOE0) of real-time counter control register 0 (RTCC0), bit 6 (RCLOE2) of real-time counter control register 2 (RTCC2), bit 3 (TO03) of timer output register 0 (TO0), bit 3 (TOE03) of timer output enable register 0 (TOE0) and bit 7 of clock output select register 1 (CKS1) to "0", which is the same as their default status setting.

3. To use P32/TO01/TI01/INTP5/PCLBUZ0 as a general-purpose port, set bit 1 (TO01) of timer output register 0 (TO0), bit 1 (TOE01) of timer output enable register 0 (TOE0) and bit 7 of clock output select register 0 (CKS0) to "0", which is the same as their default status setting.

4. To use P33/TO07/TI07/INTP3, P34/TO06/TI06/INTP8 as a general-purpose port, set bit 7, 6 (TO07, TO06) of timer output register 0 (TO0), and bit 7, 6 (TOE07, TOE06) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.

2.2.5 P40, P41

P40 and P41 function as an I/O port. These pins also function as data I/O for a flash memory programmer/debugger and clock output for a debugger.

The following operation modes can be specified in 1-bit units.

(1) Port mode

P40 and P41 function as an I/O port. P40 and P41 can be set to input or output port in 1-bit units using port mode register 4 (PM4). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 4 (PU4).

Be sure to connect an external pull-up resistor to P40 when on-chip debugging is enabled (by using an option byte).

(2) Control mode

P40 and P41 function as data I/O for a flash memory programmer/debugger and clock output for a debugger.

(a) TOOL0

This is a data I/O pin for a flash memory programmer/debugger.

Be sure to pull up this pin externally when on-chip debugging is enabled (pulling it down is prohibited).

(b) TOOL1

This is a clock output pin for a debugger.

When the on-chip debug function is used, P41/TOOL1 pin can be used as follows by the mode setting on the debugger.

1-line mode: can be used as a port (P41).

2-line mode: used as a TOOL1 pin and cannot be used as a port (P41).

Caution The function of the P40/TOOL0 pin varies as described in (a) to (c) below. In the case of (b) or (c), make the specified connection.

(a) In normal operation mode and when on-chip debugging is disabled (OCDENSET = 0) by an option byte (000C3H)
=> Use this pin as a port pin (P40).

(b) In normal operation mode and when on-chip debugging is enabled (OCDENSET = 1) by an option byte (000C3H)
=> Connect this pin to V_{DD} via an external resistor, and always input a high level to the pin before reset release.

(c) When on-chip debug function is used, or in write mode of flash memory programmer
=> Use this pin as TOOL0.
Directly connect this pin to the on-chip debug emulator or a flash memory programmer, or pull it up by connecting it to V_{DD} via an external resistor.

2.2.6 P50 to P57

P50 to P57 function as an I/O port. This port can also be used for serial interface data I/O and timer input.

The following operation modes can be specified in 1-bit units.

(1) Port mode

P50 to P57 function as an I/O port. P50 to P57 can be set to input or output port in 1-bit units using port mode register 5 (PM5). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 5 (PU5).

(2) Control mode

P50 to P57 function as serial interface data I/O and timer input.

(a) RxD3

This is a data input pin of serial interface UART3.

(b) TxD3

This is a data output pin of serial interface UART3.

(c) TI02, TI04

These are input pins of 16-bit timers 02 and 04.

2.2.7 P60, P61

P60 and P61 function as an I/O port. This port can also be used for serial interface data I/O and clock I/O.

The following operation modes can be specified in 1-bit units.

(1) Port mode

P60 and P61 function as an I/O port. P60 and P61 can be set to input port or output port in 1-bit units using port mode register 6 (PM6).

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

(2) Control mode

P60 and P61 function as serial interface clock I/O and data I/O.

(a) SCL0

This is a serial clock I/O pin of serial interface IICA.

(b) SDA0

This is a serial data I/O pin of serial interface IICA.

Caution When using P60/SCL0 or P61/SDA0 as a general-purpose port, stop the operation of serial interface IICA.

2.2.8 P80 to P82

P80 to P82 function as an I/O port. This port can also be used for serial interface clock I/O, data I/O, timer I/O, and external interrupt request input.

Output from the P80 and P82 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 8 (POM8).

The following operation modes can be specified in 1-bit units.

(1) Port mode

P80 to P82 function as an I/O port. P80 to P82 can be set to input or output port in 1-bit units using port mode register 8 (PM8). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 8 (PU8).

(2) Control mode

P80 to P82 function as serial interface clock I/O, data I/O, timer I/O, and external interrupt request input.

(a) $\overline{\text{SCK00}}$

This is a clock I/O pin of serial interface CSI00.

(b) SI00

This is a serial data input pin of serial interface CSI00.

(c) SO00

This is a serial data output pin of serial interface CSI00.

(d) RxD0

This is a serial data input pin for serial interface UART0.

(e) TxD0

This is a serial data output pin for serial interface UART0.

(f) INTP9, INTP11

These are external interrupt request input pins for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

Caution To use P80/$\overline{\text{SCK00}}$/INTP11, P81/RxD0/SI00/INTP9, P82/SO00/TxD0, as a general-purpose port, note the serial array unit 0 setting. For details, refer to Table 14-5 Relationship Between Register Settings and Pins (Channel 0 of unit 0: CSI00, UART0 transmission).

2.2.9 P100

P100 functions as an I/O port. P100 can be set to input or output port in 1-bit units using port mode register 10 (PM10). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 10 (PU10).

2.2.10 P110, P111

P110 and P111 function as an I/O port. This port can also be used for D/A converter analog output.

The following operation modes can be specified in 1-bit units.

(1) Port mode

P110 and P111 function as an I/O port. P110 and P111 can be set to input or output port in 1-bit units using port mode register 11 (PM11).

(2) Control mode

P110 and P111 function as D/A converter analog output (ANO0, ANO1).

Caution When using at least one port of P110/ANO0 and P111/ANO1 as a digital port, set AV_{DD1} to the same potential as EV_{DD} or V_{DD}.

2.2.11 P120 to P124

P120 functions as an I/O port. P121 to P124 function as an input port. These pins also function as potential input for external low-voltage detection, connecting resonator for main system clock, connecting resonator for subsystem clock, external clock input for main system clock, and external interrupt request input.

The following operation modes can be specified in 1-bit units.

(1) Port mode

P120 functions as an I/O port. P120 can be set to input port or output port using port mode register 12 (PM12). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

P121 to P124 function as an input port.

(2) Control mode

P120 to P124 function as potential input for external low-voltage detection, connecting resonator for main system clock, connecting resonator for subsystem clock, external clock input for main system clock, and external interrupt request input.

(a) EXLVI

This is a potential input pin for external low-voltage detection.

(b) X1, X2

These are pins for connecting a resonator for main system clock.

(c) EXCLK

This is an external clock input pin for main system clock.

(d) XT1, XT2

These are pins for connecting a resonator for subsystem clock.

(e) INTP0

This is an external interrupt request input pin for which the valid edge (rising edge, falling edge, or both rising and falling edges) can be specified.

Caution The function setting on P121 to P124 is available only once after the reset release. The port once set for connection to an oscillator cannot be used as an input port unless the reset is performed.

2.2.12 P130

P130 functions as an output port.

Remark The P130 pin outputs a low level when it is used as a port function pin and a reset is effected. If P130 is set to output a high level before reset is effected, the output signal of P130 can be dummy-output as the CPU reset signal (see the figure for **Remark** in **4.2.12 Port 13**).

2.2.13 P140, P141

P140 and P141 function as an I/O port. P140 and P141 can be set to input or output port in 1-bit units using port mode register 14 (PM14). Use of an on-chip pull-up resistor can be specified by pull-up resistor option register 14 (PU14).

2.2.14 P150 to P152, P157

P150 to P152, and P157 function as an I/O port. This port can also be used for A/D converter analog input, reference voltage input, and operational amplifier input.

The following operation modes can be specified in 1-bit units.

(1) Port mode

P150 to P152, and P157 function as an I/O port. P150 to P152, and P157 can be set to input or output port in 1-bit units using port mode register 15 (PM15).

(2) Control mode

P150 to P152, and P157 function as A/D converter analog input, reference voltage input, and operational amplifier input.

(a) ANI8 to ANI10, ANI15

These are A/D converter analog input pins.

(b) AV_{REFM}

This is a pin that inputs the negative reference voltage of A/D converter.

(c) AMP2+

This is an input voltage pin on the positive side of operational amplifier 2.

Cautions 1. P150/ANI8/AMP2+ to P152/ANI10, and P157/ANI15/AV_{REFM} are set in the digital input (general-purpose port) mode after release of reset.

2. When using at least one port of P150/ANI8/AMP2+ to P152/ANI10, and P157/ANI15/AV_{REFM} as a digital port, set AV_{DD0} to the same potential as EV_{DD} or V_{DD}.

2.2.15 V_{REFOUT}/AV_{REFP}

V_{REFOUT} is an analog reference voltage output pin for voltage reference.

AV_{REFP} is a pin that inputs the positive reference voltage of A/D converter and D/A converter.

2.2.16 $\overline{\text{RESET}}$

This is an active-low system reset input pin.

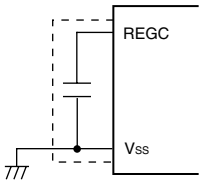
When the external reset pin is not used, connect this pin directly to EV_{DD} or via a resistor.

When the external reset pin is used, design the circuit based on V_{DD}.

2.2.17 REGC

This is a pin for connecting regulator output (2.4 V) stabilization capacitance for internal operation. Connect this pin to V_{SS} via a capacitor (0.47 to 1 μF).

Also, use a capacitor with good characteristics, since it is used to stabilize internal voltage.



Caution Keep the wiring length as short as possible for the broken-line part in the above figure.

2.2.18 FLMD0

This is a pin for setting flash memory programming mode.

Perform either of the following processing.

(a) In normal operation mode

It is recommended to leave this pin open during normal operation.

The FLMD0 pin must always be kept at the V_{SS} level before reset release but does not have to be pulled down externally because it is internally pulled down by reset. However, pulling it down must be kept selected (i.e., FLMDPUP = "0", default value) by using bit 7 (FLMDPUP) of the background event control register (BECTL) (see **25.5 (1) Back ground event control register**). To pull it down externally, use a resistor of 200 kΩ or smaller.

Self programming and the rewriting of flash memory with the programmer can be prohibited using hardware, by directly connecting this pin to the V_{SS} pin.

(b) In self programming mode

It is recommended to leave this pin open when using the self programming function. To pull it down externally, use a resistor of 100 kΩ to 200 kΩ.

In the self programming mode, the setting is switched to pull up in the self programming library.

(c) In flash memory programming mode

Directly connect this pin to a flash memory programmer when data is written by the flash memory programmer. This supplies a writing voltage of the V_{DD} level to the FLMD0 pin.

The FLMD0 pin does not have to be pulled down externally because it is internally pulled down by reset. To pull it down externally, use a resistor of 1 kΩ to 200 kΩ.

2.2.19 AV_{DD0}, AV_{DD1}, AV_{SS}, EV_{DD}, EV_{SS}, V_{DD}, V_{SS}

(1) AV_{DD0}

This is a ground potential pin of A/D converter, operational amplifier, voltage reference, P20 to P27, P150 to P152, and P157.

When using at least one port of ports 2 and 15 as a digital port, or when not using the A/D converter, operational amplifier, or voltage reference, set AV_{DD0} to the same potential as EV_{DD} or V_{DD}.

(2) AV_{DD1}

This is a ground potential pin of D/A converter, P110, and P111.

When using at least one port of port 11 as a digital port, or when not using the D/A converter set AV_{DD1} to the same potential as EV_{DD} or V_{DD}.

(3) AV_{SS}

This is a ground potential pin of A/D converter, P20 to P27, P110, P111, P150 to P152, and P157. Even when the A/D converter, D/A converter, operational amplifier, and voltage reference is not used, always use this pin with the same potential as EV_{SS} and V_{SS}.

(4) EV_{DD}

This is a positive power supply pin for ports other than P20 to P27, P110, P111, P150 to P152, and P157 as well as for the $\overline{\text{RESET}}$ and FLMD0 pins.

(5) EV_{SS}

This is a ground potential pin for ports other than P20 to P27, P110, P111, P150 to P152, and P157 as well as for the $\overline{\text{RESET}}$ and FLMD0 pins.

(6) V_{DD}

This is a positive power supply pin other than ports, $\overline{\text{RESET}}$, and FLMD0 pins.

(7) V_{SS}

This is a ground potential pin other than ports, $\overline{\text{RESET}}$, and FLMD0 pins.

2.3 Pin I/O Circuits and Recommended Connection of Unused Pins

Table 2-2 shows the types of pin I/O circuits and the recommended connections of unused pins.

Table 2-2. Connection of Unused Pins (1/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P00	5-AG	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P01			
P02			
P10/$\overline{SCK20}$/SCL20	5-AN		
P11/SI20/RxD2/SDA20/ INTP6			
P12/SO20/TxD2/TO02	5-AG		
P13/SO10/TxD1/TO04			
P14/SI10/RxD1/SDA10/ INTP4	5-AN		
P15/$\overline{SCK10}$/SCL10/INTP7			
P16/TI05/TO05/INTP10	8-R		
P20/ANI0/AMP0-[Note]	11-P		Input: Independently connect to AV_{DD0} or AV_{SS} via a resistor. Output: Leave open.
P21/ANI1/AMP0O[Note]	11-S		
P22/ANI2/AMP0+[Note]	11-N		
P23/ANI3/AMP1-[Note]	11-P		
P24/ANI4/AMP1O[Note]	11-S		
P25/ANI5/AMP1+[Note]	11-N		
P26/ANI6/AMP2-[Note]	11-P		
P27/ANI7/AMP2O[Note]	11-S		
P30/TI03/TO00/RTC1HZ/ INTP1	8-R		Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P31/TI00/TO03/RTCDIV/ RTCCL/PCLBUZ1/INTP2			
P32/TI01/TO01/INTP5/ PCLBUZ0			
P33/TI07/TO07/INTP3			
P34/TI06/TO06/INTP8			
P40/TOOL0	8-R		<When on-chip debugging is enabled> Pull this pin up (pulling it down is prohibited). <When on-chip debugging is disabled> Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P41/TOOL1	5-AG		Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.

Note P20/ANI0/AMP0- to P27/ANI7/ANP2O are set in the digital input port mode after release of reset.

Table 2-2. Connection of Unused Pins (2/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
P50/RxD3	5-AQ	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P51/TxD3	5-AG		
P52/TI02	5-AQ		
P53/TI04			
P54 to P57	5-AG		
P60/SCL0	13-R		
P61/SDA0			
P80/$\overline{SCK00}$/INTP11	8-R		
P81/RxD0/SI00/INTP9			
P82/SO00/TxD0	5-AG		
P100			
P110/ANO0, P111/ANO1	12-A		Input: Independently connect to AV_{DD1} or AV_{SS} via a resistor. Output: Leave open.
P120/INTP0/EXLVI	8-R		Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P121/X1 Note 1	37-C	Input	Independently connect to EV_{DD} or EV_{SS} via a resistor.
P122/X2/EXCLK Note 1			
P123/XT1 Note 1	37-A		
P124/XT2 Note 1			
P130	3-C	Output	Leave open.
P140, P141	5-AG	I/O	Input: Independently connect to EV_{DD} or EV_{SS} via a resistor. Output: Leave open.
P150/ANI8/AMP2+ Note 2	11-N		Input: Independently connect to AV_{DD0} or AV_{SS} via a resistor. Output: Leave open.
P151/ANI9 Note 2	11-G		
P152/ANI10 Note 2			
P157/ANI15/AV_{REFM} Note 2	11-T		

Notes 1. Use recommended connection above in input port mode (see **Figure 5-2 Format of Clock Operation Mode Control Register (CMC)**) when these pins are not used.

2. P150/ANI8/AMP2+ to P152/ANI10 and P157/ANI15/AV_{REFM} are set in the digital input port mode after release of reset.

Table 2-2. Connection of Unused Pins (3/3)

Pin Name	I/O Circuit Type	I/O	Recommended Connection of Unused Pins
AV_{DD0}	–	–	<When one or more of P20 to P27, P150 to P152, or P157 are set as a digital port> Make this pin the same potential as EV_{DD} or V_{DD}. <When all of P20 to P27, P150 to P152, and P157 are set as analog ports> Make this pin to have a potential where 2.3 V ≤ AV_{DD0} ≤ V_{DD}.
AV_{DD1}	–	–	<When one or more of P110 or P111 are set as a digital port> Make this pin the same potential as EV_{DD} or V_{DD}. <When all of P110 and P111 are set as analog ports> Make this pin to have a potential where 2.3 V ≤ AV_{DD1} ≤ V_{DD}.
AV_{SS}	–	–	Make this pin the same potential as the EV_{SS} or V_{SS}.
V_{REFOUT}/AV_{REFP}	–	–	Make this pin the same potential as the AV_{DD0}, EV_{DD} or V_{DD}.
FLMD0	2-W	–	Leave open or connect to V_{SS} via a resistor of 100 kΩ or more.
$\overline{\text{RESET}}$	2	Input	Connect directly to EV_{DD} or via a resistor.
REGC	–	–	Connect to V_{SS} via capacitor (0.47 to 1 μF).

Figure 2-1. Pin I/O Circuit List (1/3)

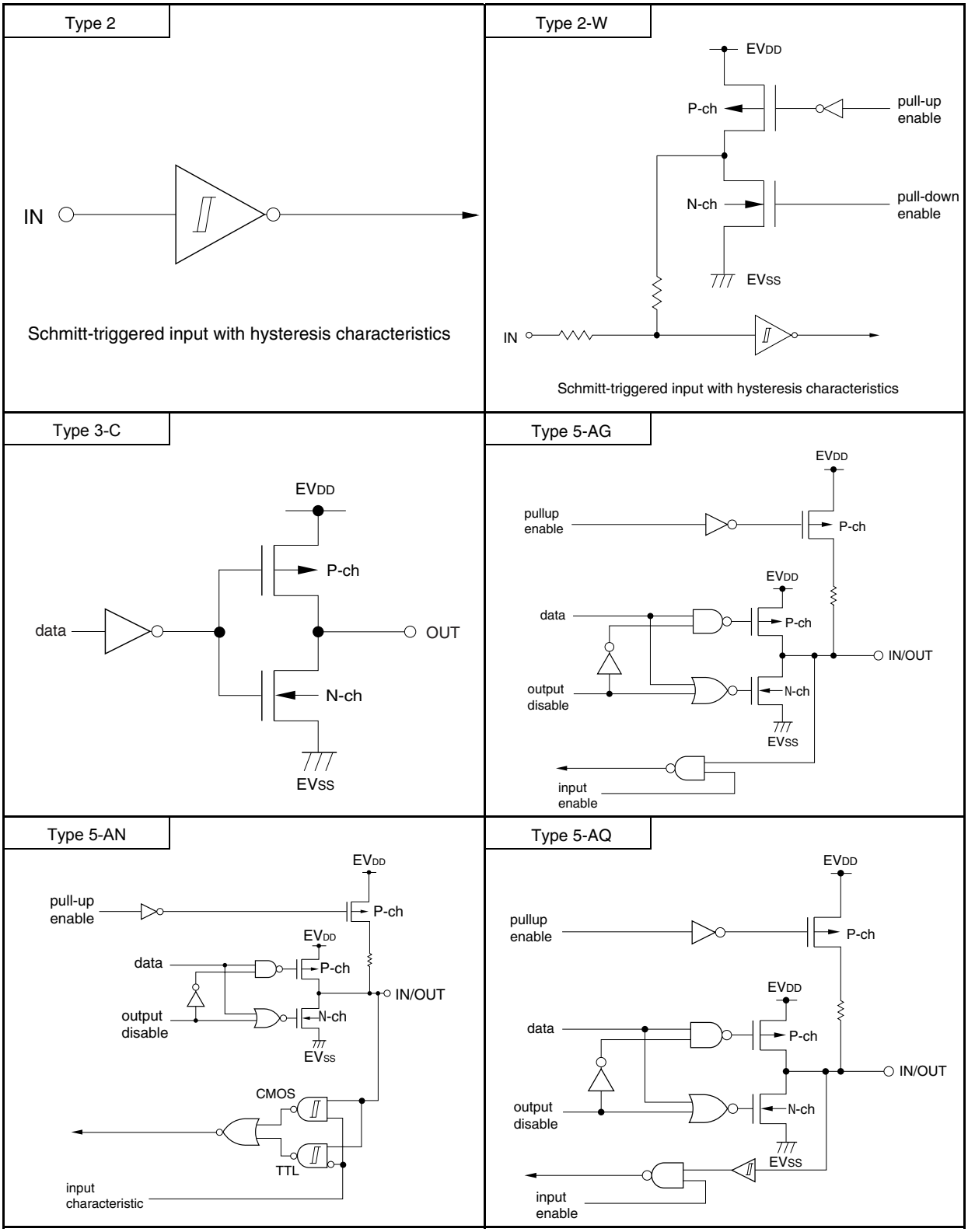


Figure 2-1. Pin I/O Circuit List (2/3)

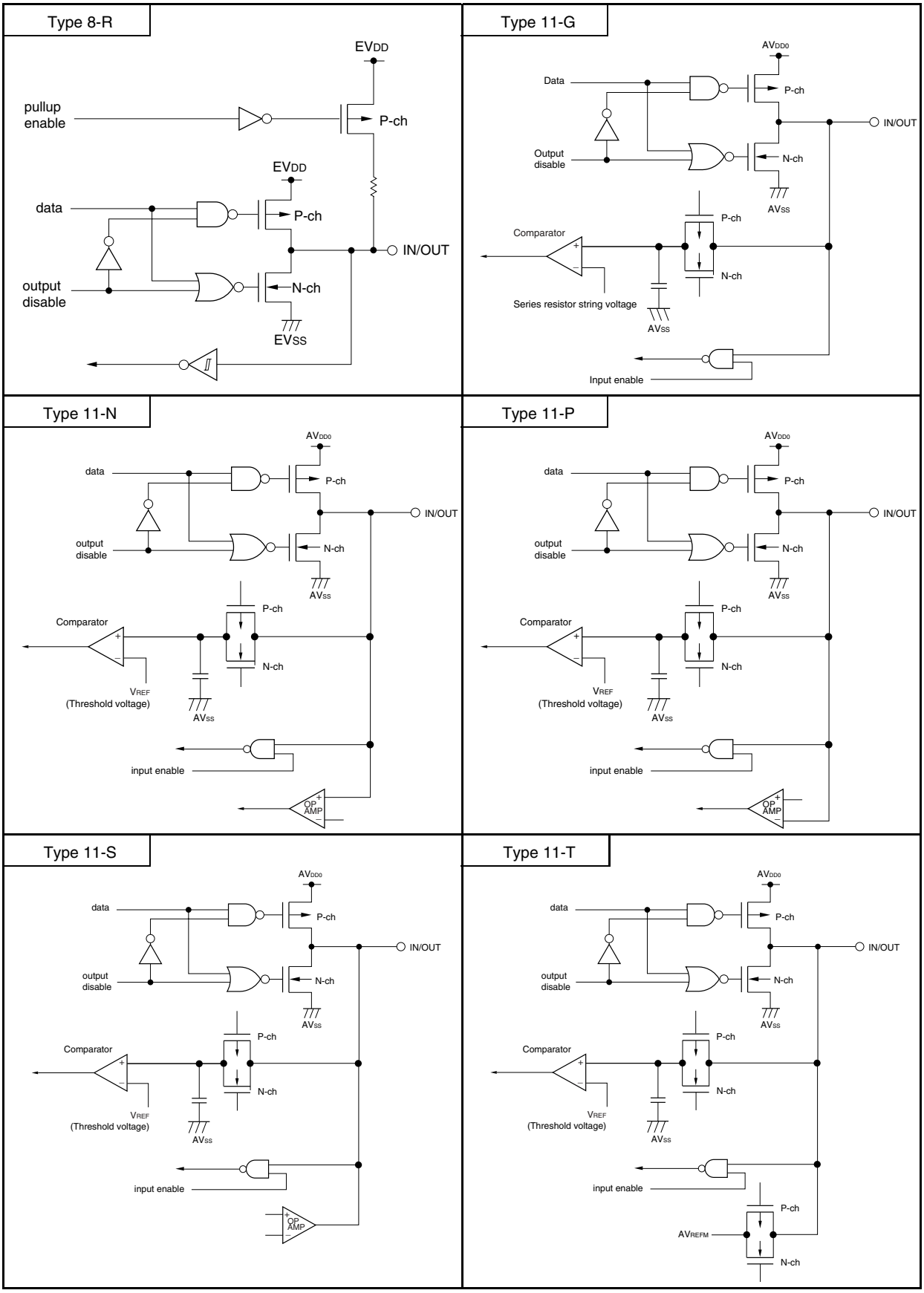
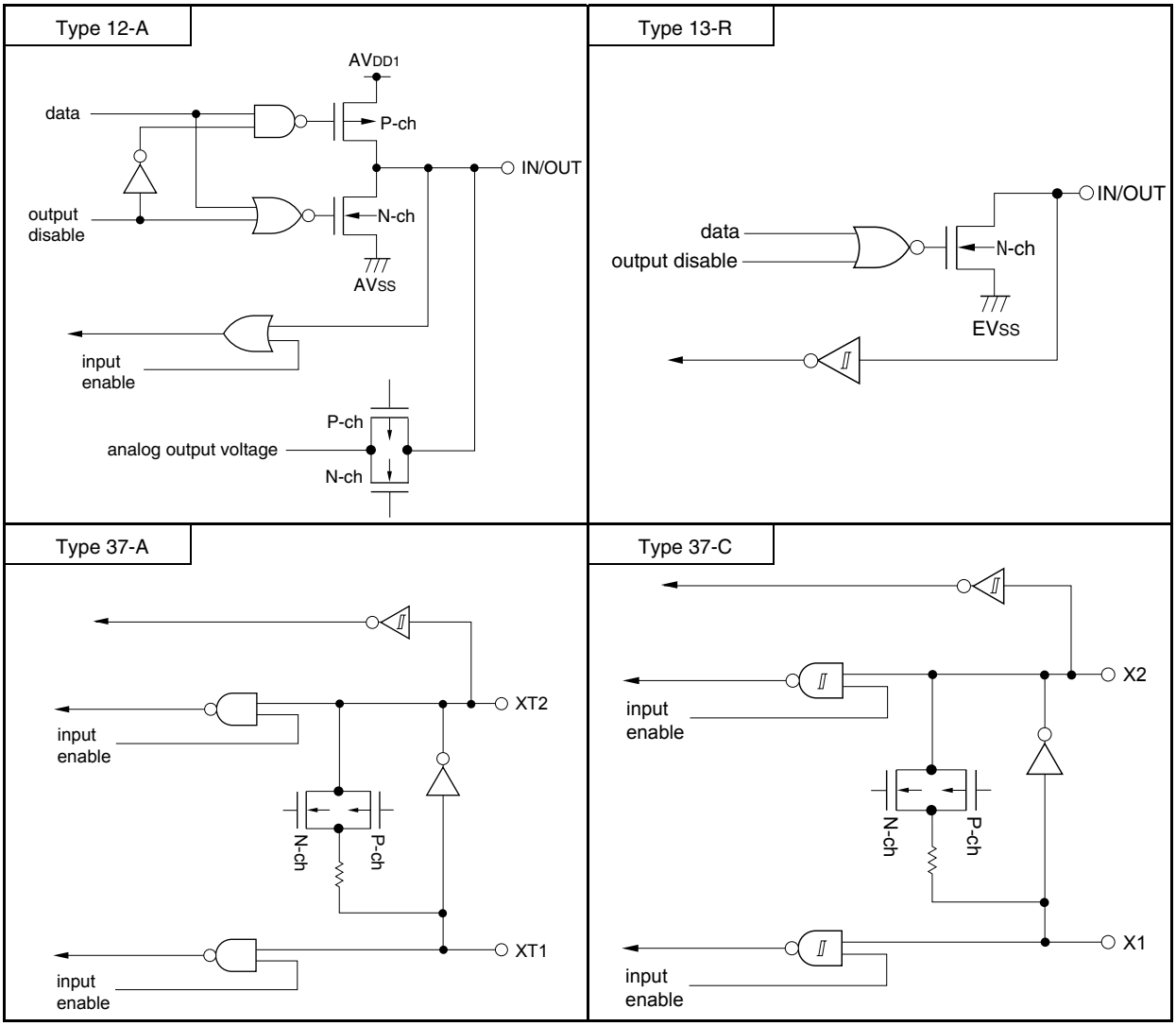


Figure 2-1. Pin I/O Circuit List (3/3)

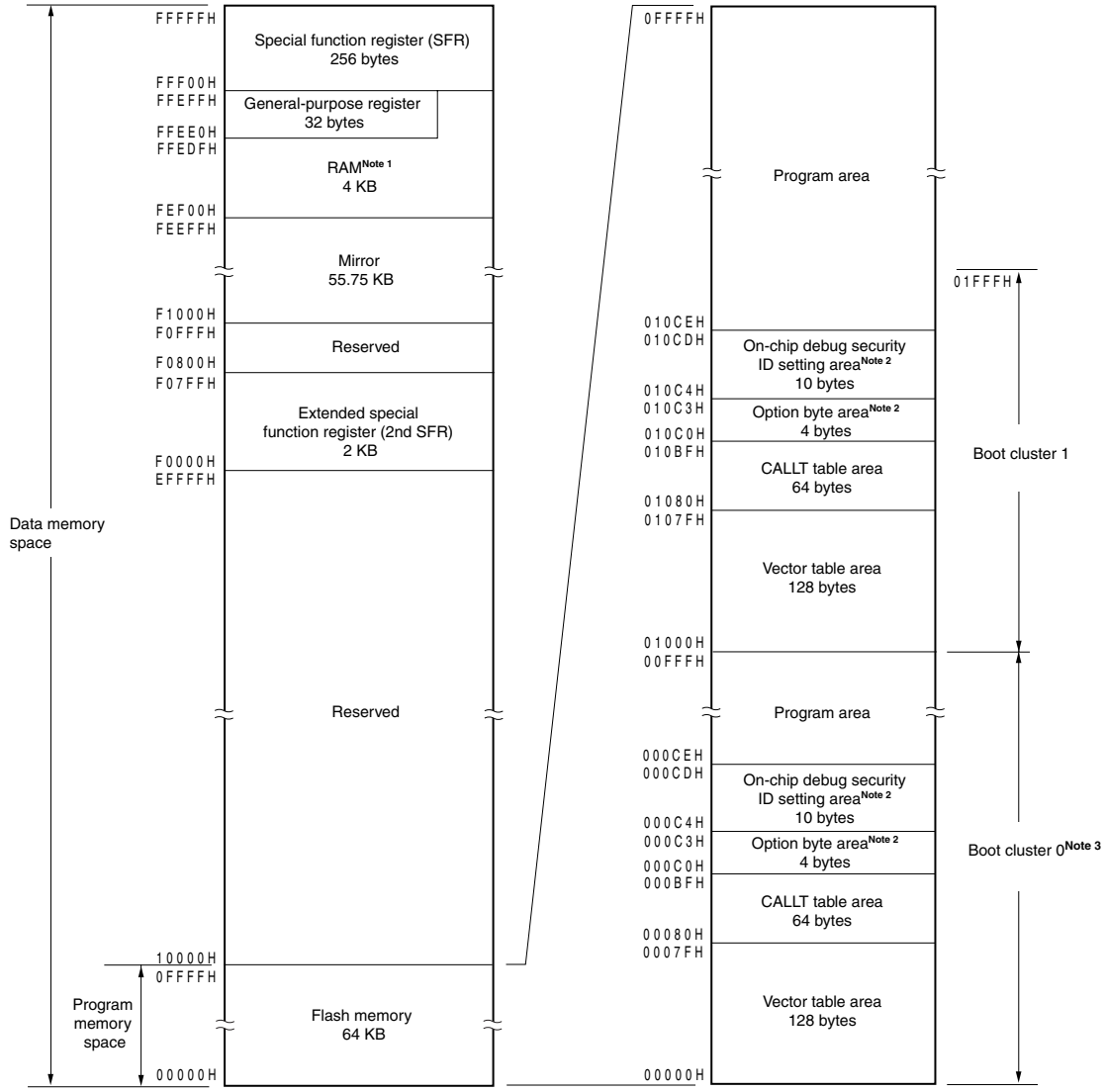


CHAPTER 3 CPU ARCHITECTURE

3.1 Memory Space

Products in the 78K0R/Kx3-A microcontrollers can access a 1 MB memory space. Figures 3-1 to 3-3 show the memory maps.

Figure 3-1. Memory Map (μPD78F1016)



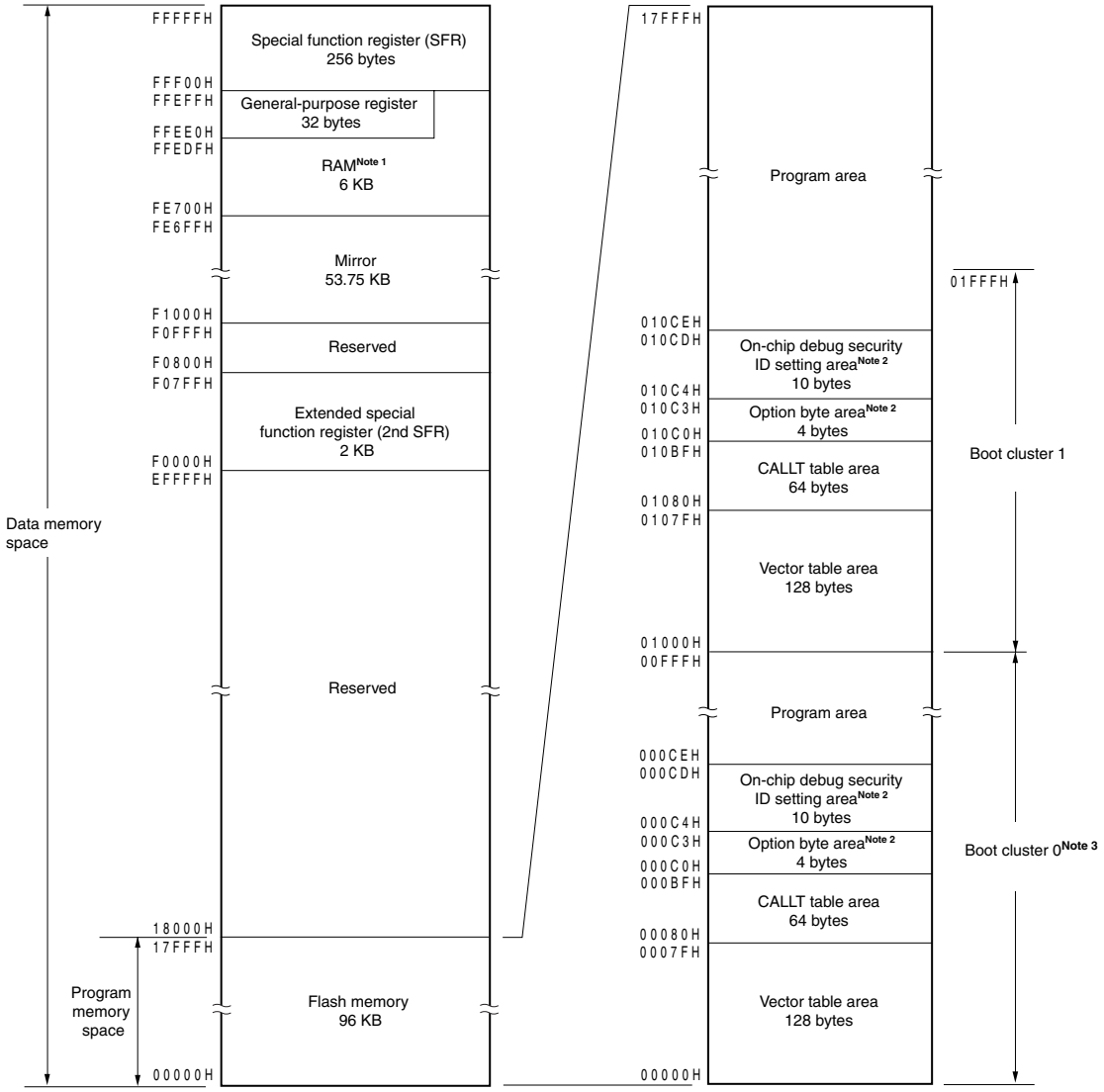
Notes 1. Instructions can be executed from the RAM area excluding the general-purpose register area.

2. 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.

3. Writing boot cluster 0 can be prohibited depending on the setting of security (see **25.7 Security Setting**).

Figure 3-2. Memory Map (μPD781017)



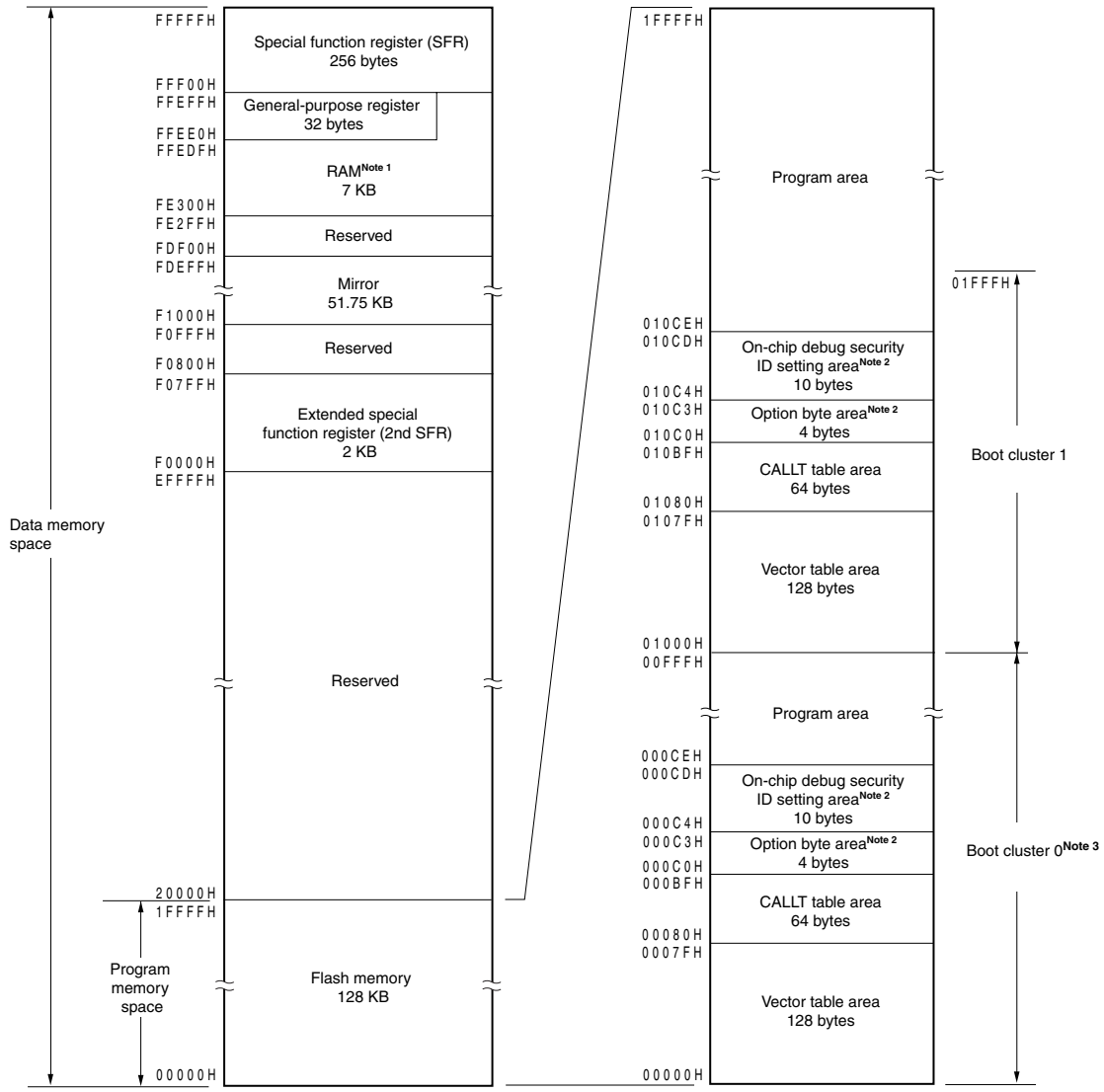
Notes 1. Instructions can be executed from the RAM area excluding the general-purpose register area.

2. 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.

3. Writing boot cluster 0 can be prohibited depending on the setting of security (see **25.7 Security Setting**).

Figure 3-3. Memory Map (μPD78F1018)



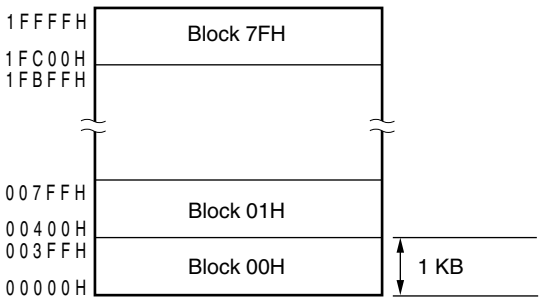
Notes 1. Instructions can be executed from the RAM area excluding the general-purpose register area.

2. 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.

3. Writing boot cluster 0 can be prohibited depending on the setting of security (see **25.7 Security Setting**).

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



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

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

Address Value	Block Number	Address Value	Block Number	Address Value	Block Number	Address Value	Block Number
00000H to 003FFH	00H	08000H to 083FFH	20H	10000H to 103FFH	40H	18000H to 183FFH	60H
00400H to 007FFH	01H	08400H to 087FFH	21H	10400H to 107FFH	41H	18400H to 187FFH	61H
00800H to 00BFFH	02H	08800H to 08BFFH	22H	10800H to 10BFFH	42H	18800H to 18BFFH	62H
00C00H to 00FFFH	03H	08C00H to 08FFFH	23H	10C00H to 10FFFH	43H	18C00H to 18FFFH	63H
01000H to 013FFH	04H	09000H to 093FFH	24H	11000H to 113FFH	44H	19000H to 193FFH	64H
01400H to 017FFH	05H	09400H to 097FFH	25H	11400H to 117FFH	45H	19400H to 197FFH	65H
01800H to 01BFFH	06H	09800H to 09BFFH	26H	11800H to 11BFFH	46H	19800H to 19BFFH	66H
01C00H to 01FFFH	07H	09C00H to 09FFFH	27H	11C00H to 11FFFH	47H	19C00H to 19FFFH	67H
02000H to 023FFH	08H	0A000H to 0A3FFH	28H	12000H to 123FFH	48H	1A000H to 1A3FFH	68H
02400H to 027FFH	09H	0A400H to 0A7FFH	29H	12400H to 127FFH	49H	1A400H to 1A7FFH	69H
02800H to 02BFFH	0AH	0A800H to 0ABFFH	2AH	12800H to 12BFFH	4AH	1A800H to 1ABFFH	6AH
02C00H to 02FFFH	0BH	0AC00H to 0AFFFH	2BH	12C00H to 12FFFH	4BH	1AC00H to 1AFFFH	6BH
03000H to 033FFH	0CH	0B000H to 0B3FFH	2CH	13000H to 133FFH	4CH	1B000H to 1B3FFH	6CH
03400H to 037FFH	0DH	0B400H to 0B7FFH	2DH	13400H to 137FFH	4DH	1B400H to 1B7FFH	6DH
03800H to 03BFFH	0EH	0B800H to 0BBFFH	2EH	13800H to 13BFFH	4EH	1B800H to 1BBFFH	6EH
03C00H to 03FFFH	0FH	0BC00H to 0BFFFH	2FH	13C00H to 13FFFH	4FH	1BC00H to 1BFFFH	6FH
04000H to 043FFH	10H	0C000H to 0C3FFH	30H	14000H to 143FFH	50H	1C000H to 1C3FFH	70H
04400H to 047FFH	11H	0C400H to 0C7FFH	31H	14400H to 147FFH	51H	1C400H to 1C7FFH	71H
04800H to 04BFFH	12H	0C800H to 0CBFFH	32H	14800H to 14BFFH	52H	1C800H to 1CBFFH	72H
04C00H to 04FFFH	13H	0CC00H to 0CFFFH	33H	14C00H to 14FFFH	53H	1CC00H to 1CFFFH	73H
05000H to 053FFH	14H	0D000H to 0D3FFH	34H	15000H to 153FFH	54H	1D000H to 1D3FFH	74H
05400H to 057FFH	15H	0D400H to 0D7FFH	35H	15400H to 157FFH	55H	1D400H to 1D7FFH	75H
05800H to 05BFFH	16H	0D800H to 0DBFFH	36H	15800H to 15BFFH	56H	1D800H to 1DBFFH	76H
05C00H to 05FFFH	17H	0DC00H to 0DFFFH	37H	15C00H to 15FFFH	57H	1DC00H to 1DFFFH	77H
06000H to 063FFH	18H	0E000H to 0E3FFH	38H	16000H to 163FFH	58H	1E000H to 1E3FFH	78H
06400H to 067FFH	19H	0E400H to 0E7FFH	39H	16400H to 167FFH	59H	1E400H to 1E7FFH	79H
06800H to 06BFFH	1AH	0E800H to 0EBFFH	3AH	16800H to 16BFFH	5AH	1E800H to 1EBFFH	7AH
06C00H to 06FFFH	1BH	0EC00H to 0EFFFH	3BH	16C00H to 16FFFH	5BH	1EC00H to 1EFFFH	7BH
07000H to 073FFH	1CH	0F000H to 0F3FFH	3CH	17000H to 173FFH	5CH	1F000H to 1F3FFH	7CH
07400H to 077FFH	1DH	0F400H to 0F7FFH	3DH	17400H to 177FFH	5DH	1F400H to 1F7FFH	7DH
07800H to 07BFFH	1EH	0F800H to 0FBFFH	3EH	17800H to 17BFFH	5EH	1F800H to 1FBFFH	7EH
07C00H to 07FFFH	1FH	0FC00H to 0FFFFH	3FH	17C00H to 17FFFH	5FH	1FC00H to 1FFFFH	7FH

Remark μPD78F1016: Block numbers 00H to 3FH
μPD78F1017: Block numbers 00H to 5FH
μPD78F1018: Block numbers 00H to 7FH

3.1.1 Internal program memory space

The internal program memory space stores the program and table data. Normally, it is addressed with the program counter (PC).

The products of 78K0R/Kx3-A microcontrollers incorporate internal ROM (flash memory), as shown below.

Table 3-2. Internal ROM Capacity

Part Number	Internal ROM	
	Structure	Capacity
μPD78F1016	Flash memory	65536 × 8 bits (00000H to 0FFFFH)
μPD78F1017		98303 × 8 bits (00000H to 17FFFH)
μPD78F1018		131071 × 8 bits (00000H to 1FFFFH)

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

(1) Vector table area

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

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

Table 3-3. Vector Table

Vector Table Address	Interrupt Source	Vector Table Address	Interrupt Source
00000H	$\overline{\text{RESET}}$ input, POC, LVI, WDT, TRAP	00030H	INTTM02
00004H	INTWDTI	00032H	INTTM03
00006H	INTLVI	00034H	INTAD
00008H	INTP0	00036H	INTRTC
0000AH	INTP1	00038H	INTRTCI
0000CH	INTP2	0003CH	INTST2
0000EH	INTP3		INTCSI20
00010H	INTP4		INTIIC20
00012H	INTP5	0003EH	INTSR2
00014H	INTST3	00040H	INTSRE2
00016H	INTSR3	00042H	INTTM04
00018H	INTSRE3	00044H	INTTM05
0001AH	INTDMA0	00046H	INTTM06
0001CH	INTDMA1	00048H	INTTM07
0001EH	INTST0	0004AH	INTP6
	INTCSI00	0004CH	INTP7
00020H	INTSR0	0004EH	INTP8
00022H	INTSRE0	00050H	INTP9
00024H	INTST1	00052H	INTP10
	INTCSI10	00054H	INTP11
	INTIIC10	00056H	INTTM10
00026H	INTSR1	00058H	INTTM11
00028H	INTSRE1	0005AH	INTTM12
0002AH	INTIICA	0005CH	INTTM13
0002CH	INTTM00	0005EH	INTMD
0002EH	INTTM01	0007EH	BRK

(2) CALLT instruction table area

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

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 24 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 010C4H to 010CDH when the boot swap is used. For details, see **CHAPTER 26 ON-CHIP DEBUG FUNCTION**.

3.1.2 Mirror area

The μPD78F1016 mirrors the data flash area of 00000H to 0FFFFH, to F0000H to FFFFFH (the data flash area to be mirrored is set by the processor mode control register (PMC)).

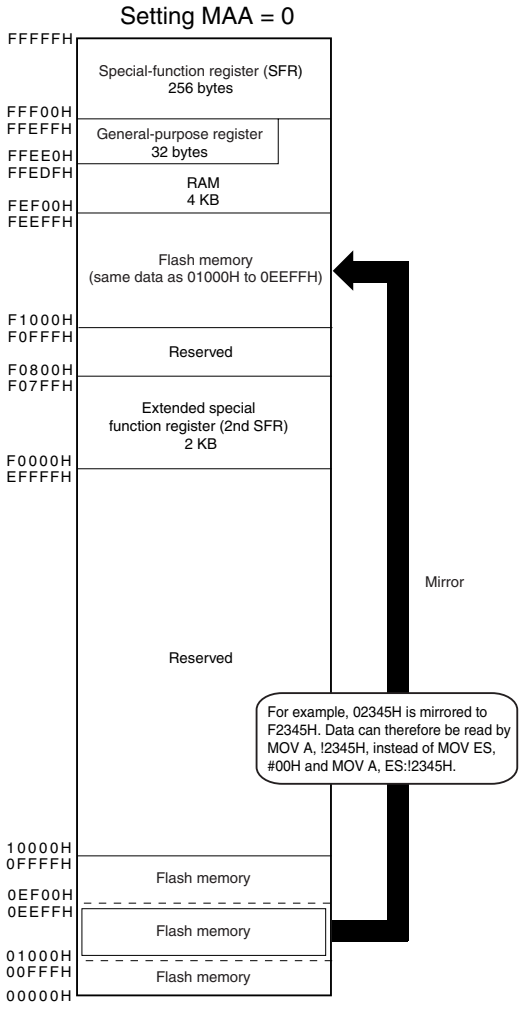
The μPD78F1017 and 78F1018 mirror the data flash area of 00000H to 0FFFFH or 10000H to 1FFFFH, to F0000H to FFFFFH (the data flash area to be mirrored is set by the processor mode control register (PMC)).

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

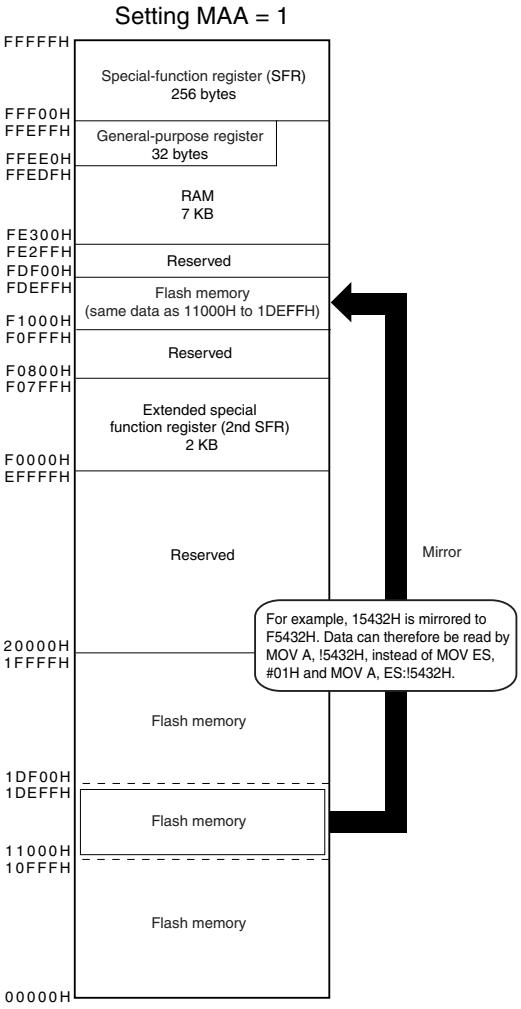
The mirror area can only be read and no instruction can be fetched from this area.

The following show examples.

Example 1 μPD78F1016
(Flash memory: 64 KB, RAM: 4 KB)



Example 2 μPD78F1018
(Flash memory: 128 KB, RAM: 7 KB)



Remark MAA: Bit 0 of the processor mode control register (PMC).

PMC register is described below.

- **Processor mode control register (PMC)**

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

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

Reset signal generation sets this register to 00H.

Figure 3-4. Format of Configuration of Processor Mode Control Register (PMC)

Address: FFFFEH After reset: 00H R/W

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

MAA	Selection of flash memory space for mirroring to area from F0000H to FFFFFH
0	00000H to 0FFFFH is mirrored to F0000H to FFFFFH
1	10000H to 1FFFFH is mirrored to F0000H to FFFFFH

Cautions 1. Set PMC only once during the initial settings prior to operating the DMA controller. Rewriting PMC other than during the initial settings is prohibited.

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

3. When the μPD78F1016 (flash memory size: 64 KB) is used, be sure to set bit 0 (MAA) of this register to 0.

3.1.3 Internal data memory space

The products of 78K0R/Kx3-A microcontrollers incorporate the following RAMs.

Table 3-4. Internal RAM Capacity

Part Number	Internal RAM
μPD78F1016	4096 × 8 bits (FEF00H to FFEFFH)
μPD78F1017	6144 × 8 bits (FE700H to FFEFFH)
μPD78F1018	7168 × 8 bits (FE300H to FFEFFH)

The internal RAM can be used as a data area and a program area where instructions are written and executed. 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. However, instructions cannot be executed by using general-purpose registers.

The internal RAM is used as a stack memory.

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

2. While using the self-programming function, the area of FFE20H to FFEFFH cannot be used as a stack memory.

3.1.4 Special function register (SFR) area

On-chip peripheral hardware special function registers (SFRs) are allocated in the area FFF00H to FFFFFH.

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.

SFRs other than those in the SFR area (FFF00H to FFFFFH) are allocated to this area. An instruction that accesses the 2nd SFR area, however, is 1 byte longer than an instruction that accesses the SFR area.

Caution Do not access addresses to which 2nd SFRs are not assigned.

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 78K0R/Kx3-A microcontrollers, based on operability and other considerations. For areas containing data memory in particular, special addressing methods designed for the functions of special function registers (SFR) and general-purpose registers are available for use. Figures 3-5 to 3-7 show correspondence between data memory and addressing.

Figure 3-5. Correspondence Between Data Memory and Addressing (μPD78F1016)

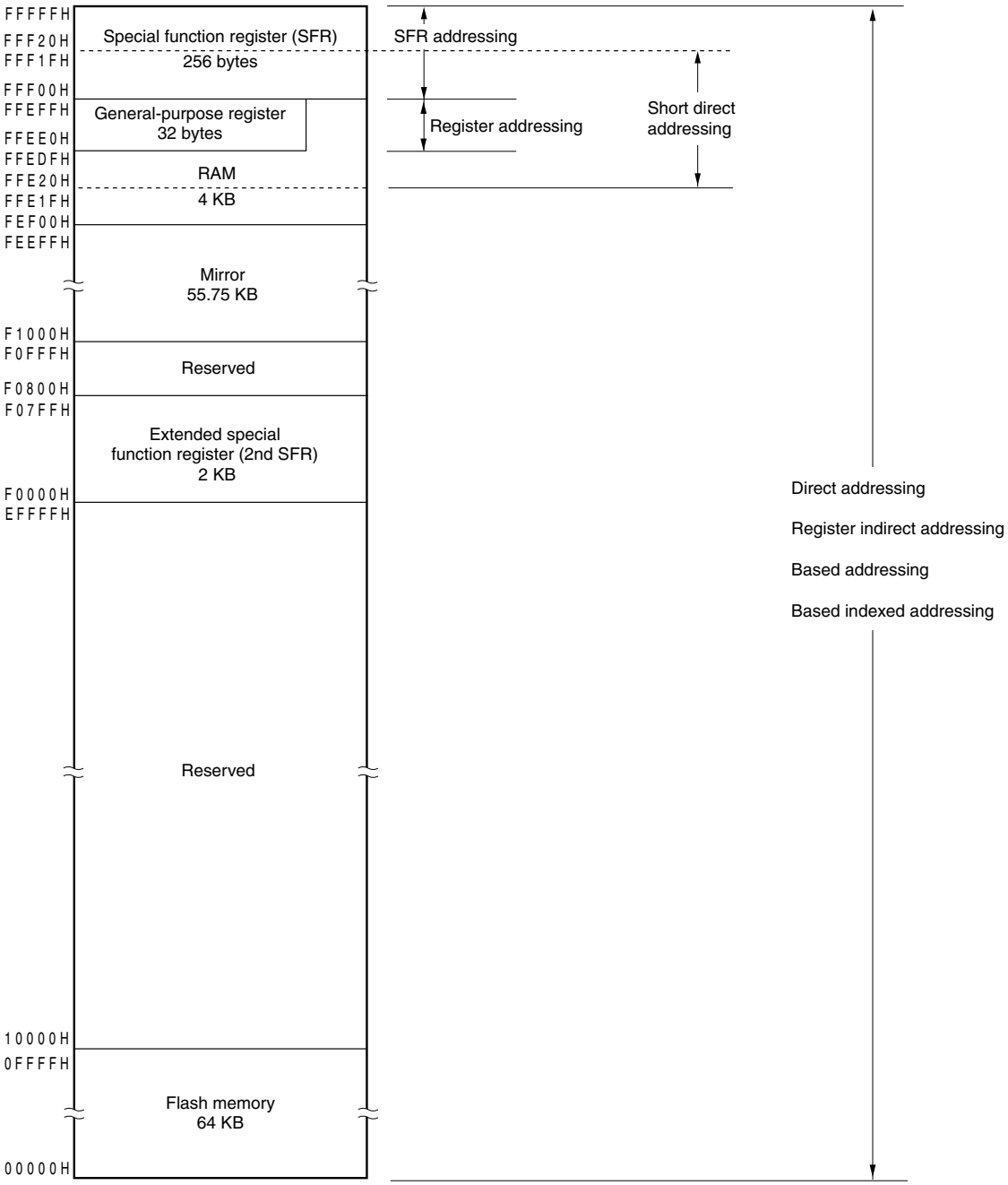


Figure 3-6. Correspondence Between Data Memory and Addressing (μPD78F1017)

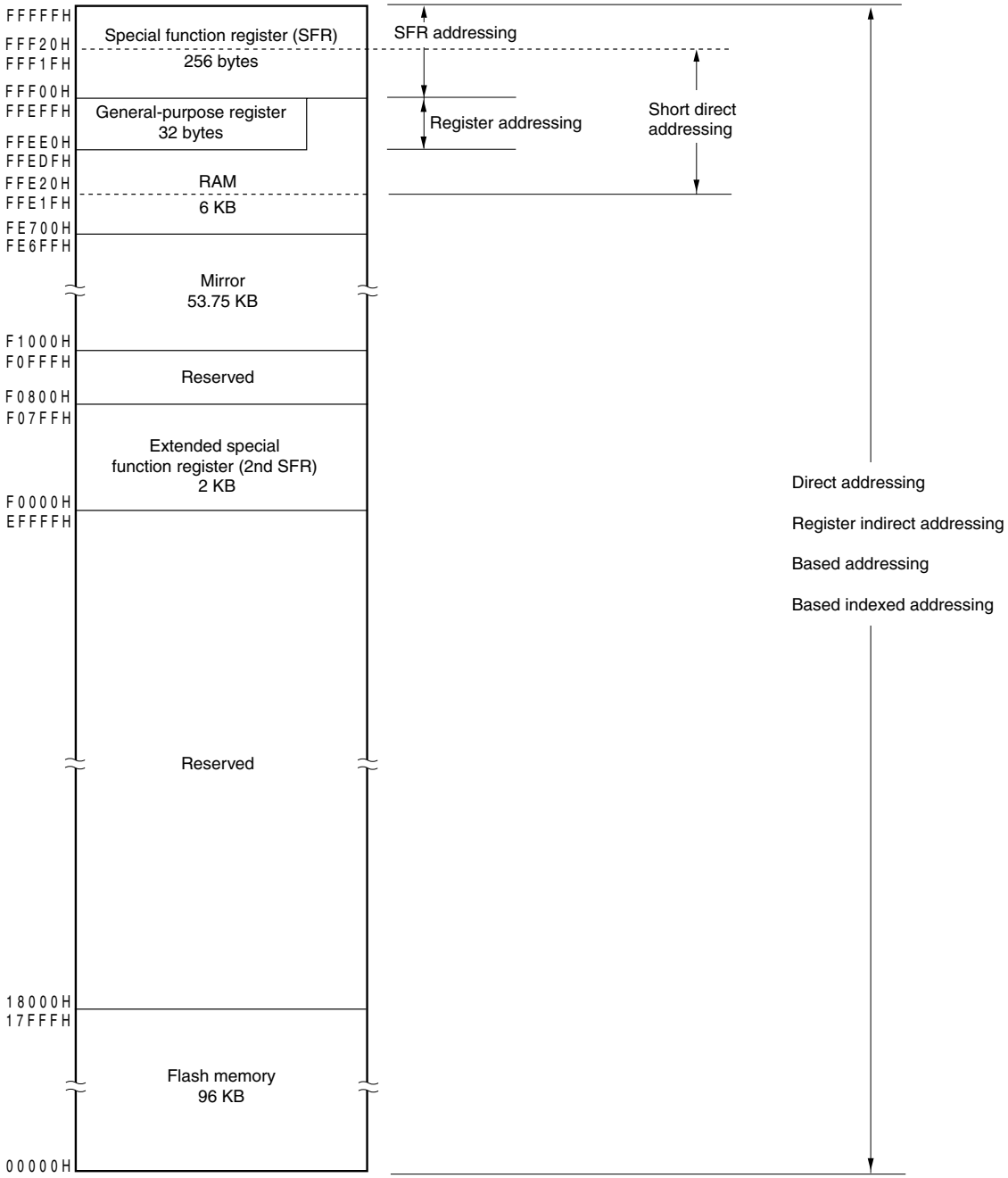
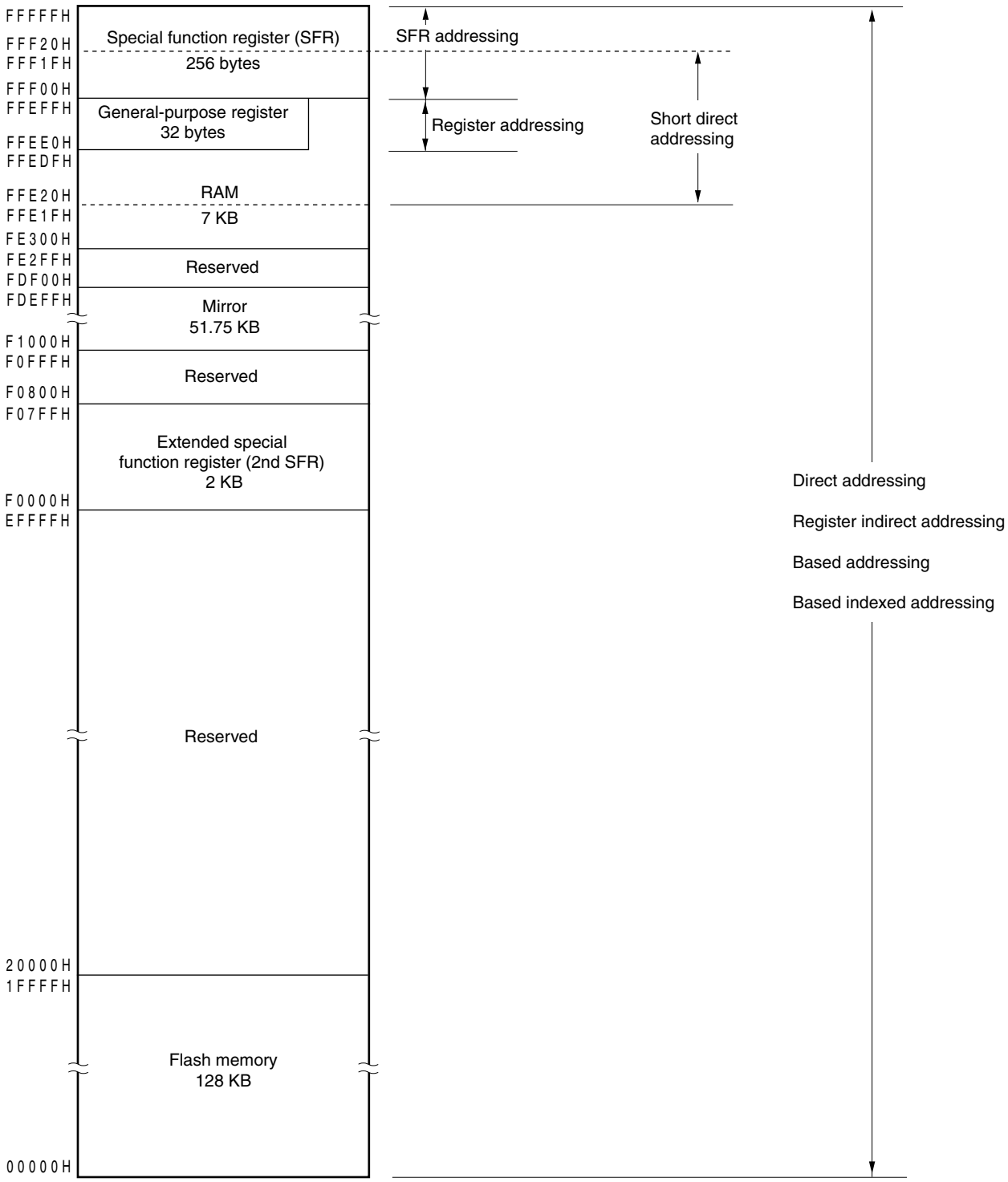


Figure 3-7. Correspondence Between Data Memory and Addressing (μPD78F1018)



3.2 Processor Registers

The products of 78K0R/Kx3-A microcontrollers 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).

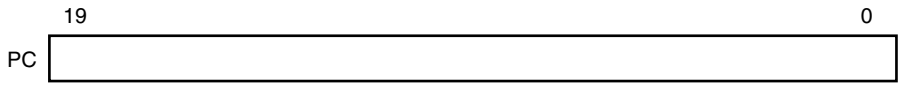
(1) Program counter (PC)

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 3-8. Format of Program Counter



(2) Program status word (PSW)

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

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

Reset signal generation sets PSW to 06H.

Figure 3-9. Format of Program Status Word

	7							0
PSW	IE	Z	RBS1	AC	RBS0	ISP1	ISP0	CY

(a) Interrupt enable flag (IE)

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

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

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

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

(b) Zero flag (Z)

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

(c) Register bank select flags (RBS0, RBS1)

These are 2-bit flags to select one of the four register banks.

In these flags, the 2-bit information that indicates the register bank selected by SEL RBn instruction execution is stored.

(d) Auxiliary carry flag (AC)

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

(e) In-service priority flags (ISP1, ISP0)

This flag manages the priority of acknowledgeable maskable vectored interrupts. Vectored interrupt requests specified lower than the value of ISP0 and ISP1 by a priority specification flag register (PRn0L, PRn0H, PRn1L, PRn1H, PRn2L, PRn2H) (see **18.3 (3)**) can not be acknowledged. Actual request acknowledgment is controlled by the interrupt enable flag (IE).

Remark n = 0, 1

(f) Carry flag (CY)

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

(3) Stack pointer (SP)

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

Figure 3-10. Format of Stack Pointer

	15															0
SP	SP15	SP14	SP13	SP12	SP11	SP10	SP9	SP8	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0

The SP is decremented ahead of write (save) to the stack memory and is incremented after read (restored) from the stack memory.

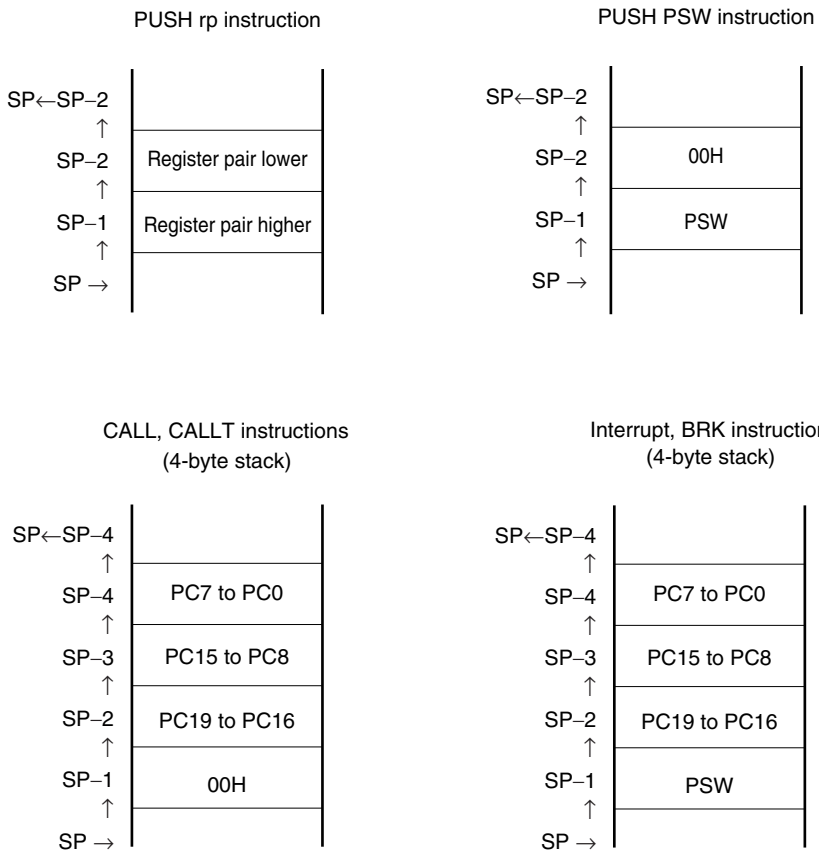
Each stack operation saves data as shown in Figure 3-11.

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

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

3. While using the self-programming function, the area of FFE20H to FFEFFH cannot be used as a stack memory.

Figure 3-11. Data to Be Saved to Stack Memory



3.2.2 General-purpose registers

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

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

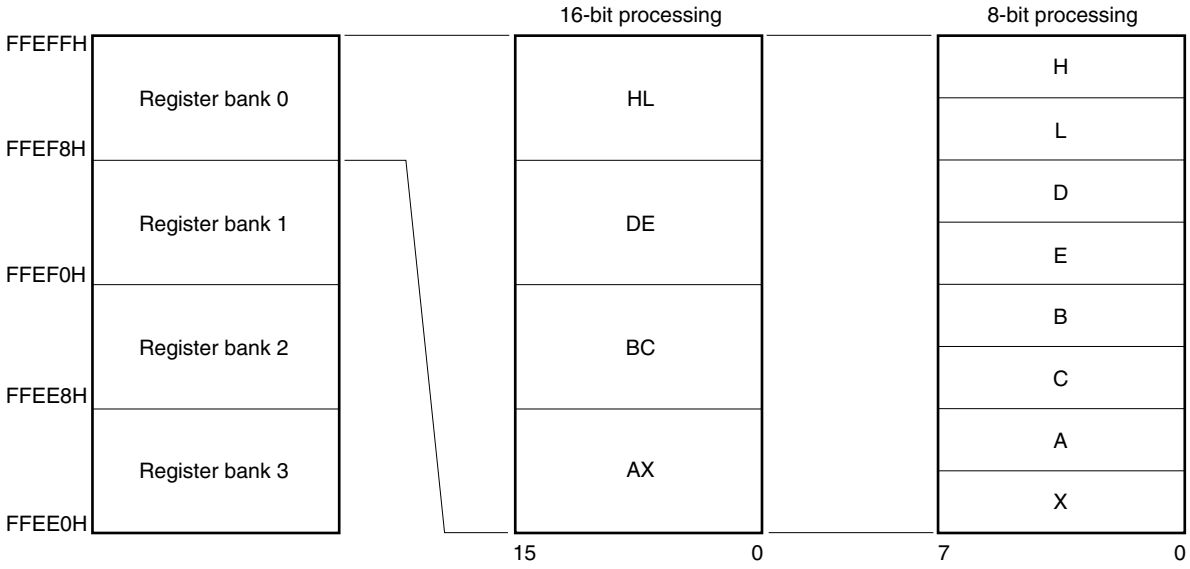
These registers can be described in terms of function names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL) and absolute names (R0 to R7 and RP0 to RP3).

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 interrupts for each bank.

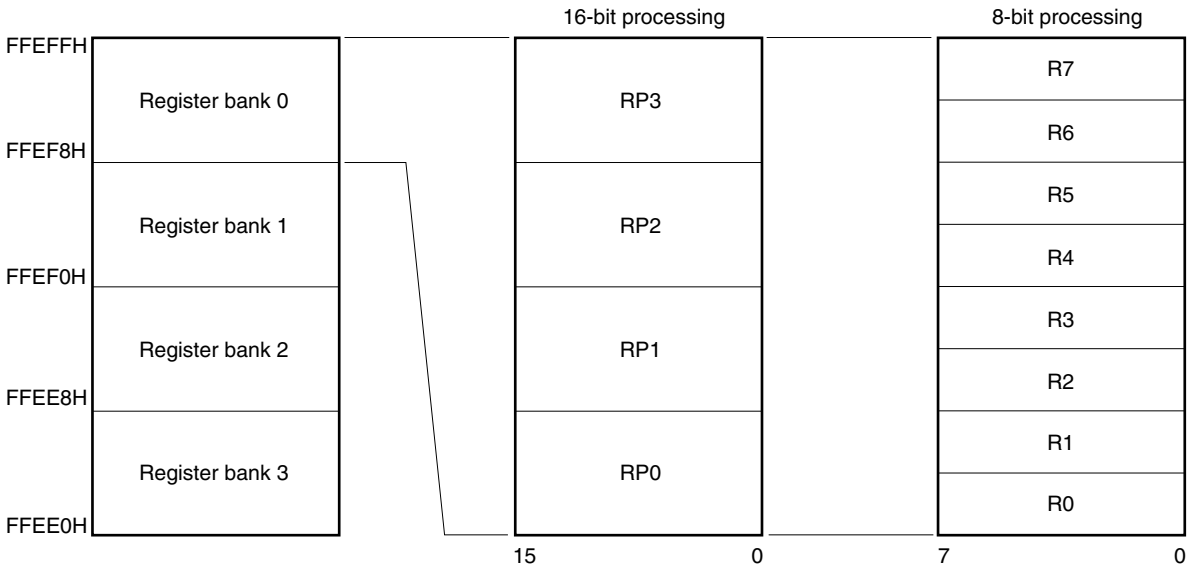
Caution It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or as a stack area.

Figure 3-12. Configuration of General-Purpose Registers

(a) Function name



(b) Absolute name



3.2.3 ES and CS registers

The ES register is used for data access and the CS register is used to specify the higher address when a branch instruction is executed.

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

Figure 3-13. 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	CP2	CP1	CP0

3.2.4 Special function registers (SFRs)

Unlike a general-purpose register, each SFR has a special function.

SFRs are allocated to the FFF00H to FFFFFH area.

SFRs can be manipulated like general-purpose registers, using operation, transfer, and bit manipulation instructions. The manipulable bit units, 1, 8, and 16, depend on the SFR type.

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describe the symbol reserved by the assembler for the 1-bit manipulation instruction operand (sfr.bit). This manipulation can also be specified with an address.
- 8-bit manipulation
  Describe the symbol reserved 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 reserved by the assembler for the 16-bit manipulation instruction operand (sfrp). When specifying an address, describe an even address.

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

- Symbol
  Symbol indicating the address of a special function register. It is a reserved word in the RA78K0R, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0R. When using the RA78K0R, ID78K0R-QB, and SM+ for 78K0R, symbols can be written as an instruction operand.
- R/W
  Indicates whether the corresponding SFR can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulable bit units
  "√" indicates the manipulable bit unit (1, 8, or 16). "–" indicates a bit unit for which manipulation is not possible.
- After reset
  Indicates each register status upon reset signal generation.

Caution Do not access addresses to which 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. SFR List (1/5)

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

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFF2AH	Port mode register 10	PM10		R/W	√	√	–	FFH
FFF2BH	Port mode register 11	PM11		R/W	√	√	–	FFH
FFF2CH	Port mode register 12	PM12		R/W	√	√	–	FFH
FFF2EH	Port mode register 14	PM14		R/W	√	√	–	FFH
FFF2FH	Port mode register 15	PM15		R/W	√	√	–	FFH
FFF30H	A/D converter mode register	ADM		R/W	√	√	–	00H
FFF31H	Analog input channel specification register	ADS		R/W	√	√	–	00H
FFF32H	A/D converter mode register 1	ADM1		R/W	√	√	–	00H
FFF33H	Operational amplifier control register	OAC		R/W	√	√	–	00H
FFF36H	Analog reference voltage control register	ADVRC		R/W	√	√	–	00H
FFF38H	External interrupt rising edge enable register 0	EGP0		R/W	√	√	–	00H
FFF39H	External interrupt falling edge enable register 0	EGN0		R/W	√	√	–	00H
FFF3AH	External interrupt rising edge enable register 1	EGP1		R/W	√	√	–	00H
FFF3BH	External interrupt falling edge enable register 1	EGN1		R/W	√	√	–	00H
FFF3CH	Input switch control register	ISC		R/W	√	√	–	00H
FFF3EH	Timer input select register 0	TIS0		R/W	√	√	–	00H
FFF3FH	Timer input select register 1	TIS1		R/W	√	√	–	00H
FFF44H	Serial data register 02	TXD1/ SIO10	SDR02	R/W	–	√	√	0000H
FFF45H		–			–	–		
FFF46H	Serial data register 03	RXD1	SDR03	R/W	–	√	√	0000H
FFF47H		–			–	–		
FFF48H	Serial data register 10	TXD2/ SIO20	SDR10	R/W	–	√	√	0000H
FFF49H		–			–	–		
FFF4AH	Serial data register 11	RXD2	SDR11	R/W	–	√	√	0000H
FFF4BH		–			–	–		
FFF50H	IICA shift register	IICA		R/W	–	√	–	00H
FFF51H	IICA status register	IICS		R	√	√	–	00H
FFF52H	IICA flag register	IICF		R/W	√	√	–	00H
FFF58H	D/A conversion value setting register 0	DACS0	DACS W0	R/W	–	√	–	00H
FFF59H	D/A conversion value setting register W0	–		R/W	–	–	√	0000H
FFF5AH	D/A conversion value setting register 1	DACS1	DACS W1	R/W	–	√	–	00H
FFF5BH	D/A conversion value setting register W1	–		R/W	–	–	√	0000H
FFF5CH	D/A converter mode register	DAM		R/W	√	√	–	00H
FFF64H	Timer data register 02	TDR02		R/W	–	–	√	0000H
FFF65H								
FFF66H	Timer data register 03	TDR03		R/W	–	–	√	0000H
FFF67H								

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

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
FFF68H	Timer data register 04	TDR04	R/W	–	–	√	0000H
FFF69H							
FFF6AH	Timer data register 05	TDR05	R/W	–	–	√	0000H
FFF6BH							
FFF6CH	Timer data register 06	TDR06	R/W	–	–	√	0000H
FFF6DH							
FFF6EH	Timer data register 07	TDR07	R/W	–	–	√	0000H
FFF6FH							
FFF70H	Timer data register 10	TDR10	R/W	–	–	√	0000H
FFF71H							
FFF72H	Timer data register 11	TDR11	R/W	–	–	√	0000H
FFF73H							
FFF74H	Timer data register 12	TDR12	R/W	–	–	√	0000H
FFF75H							
FFF76H	Timer data register 13	TDR13	R/W	–	–	√	0000H
FFF77H							
FFF90H	Sub-count register	RSUBC	R	–	–	√	0000H
FFF91H							
FFF92H	Second count register	SEC	R/W	–	√	–	00H
FFF93H	Minute count register	MIN	R/W	–	√	–	00H
FFF94H	Hour count register	HOUR	R/W	–	√	–	12H[Note]
FFF95H	Week count register	WEEK	R/W	–	√	–	00H
FFF96H	Day count register	DAY	R/W	–	√	–	01H
FFF97H	Month count register	MONTH	R/W	–	√	–	01H
FFF98H	Year count register	YEAR	R/W	–	√	–	00H
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 counter control register 0	RTCC0	R/W	√	√	–	00H
FFF9EH	Real-time counter control register 1	RTCC1	R/W	√	√	–	00H
FFF9FH	Real-time counter control register 2	RTCC2	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	√	√	–	09H
FFFA5H	Clock output select register 0	CKS0	R/W	√	√	–	00H
FFFA6H	Clock output select register 1	CKS1	R/W	√	√	–	00H

Note The value of this register is 00H if the AMPM bit (bit 3 of the RTCC0 register) is set to 1 after reset.

Table 3-5. SFR List (4/5)

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
FFFA8H	Reset control flag register	RESF		R	–	√	–	Undefined Note 1
FFFA9H	Low-voltage detection register	LVIM		R/W	√	√	–	00H Note 2
FFFAAH	Low-voltage detection level select register	LVIS		R/W	√	√	–	0EH Note 3
FFFABH	Watchdog timer enable register	WDTE		R/W	–	√	–	1AH/9AH Note 4
FFFB0H	DMA SFR address register 0	DSA0		R/W	–	√	–	00H
FFFB1H	DMA SFR address register 1	DSA1		R/W	–	√	–	00H
FFFB2H	DMA RAM address register 0L	DRA0L	DRA0	R/W	–	√	√	00H
FFFB3H	DMA RAM address register 0H	DRA0H		R/W	–	√		00H
FFFB4H	DMA RAM address register 1L	DRA1L	DRA1	R/W	–	√	√	00H
FFFB5H	DMA RAM address register 1H	DRA1H		R/W	–	√		00H
FFFB6H	DMA byte count register 0L	DBC0L	DBC0	R/W	–	√	√	00H
FFFB7H	DMA byte count register 0H	DBC0H		R/W	–	√		00H
FFFB8H	DMA byte count register 1L	DBC1L	DBC1	R/W	–	√	√	00H
FFFB9H	DMA byte count register 1H	DBC1H		R/W	–	√		00H
FFFBAH	DMA mode control register 0	DMC0		R/W	√	√	–	00H
FFFBBH	DMA mode control register 1	DMC1		R/W	√	√	–	00H
FFFBCH	DMA operation control register 0	DRC0		R/W	√	√	–	00H
FFFBDH	DMA operation control register 1	DRC1		R/W	√	√	–	00H
FFFBEH	Back ground event control register	BECTL		R/W	√	√	–	00H
FFFC0H	–	PFCMD Note 5		–	–	–	–	Undefined
FFFC2H	–	PFS Note 5		–	–	–	–	Undefined
FFFC4H	–	FLPMC Note 5		–	–	–	–	Undefined
FFFD0H	Interrupt request flag register 2	IF2L	IF2	R/W	√	√	√	00H
FFFD1H		IF2H			√	√		00H
FFFD4H	Interrupt mask flag register 2	MK2L	MK2	R/W	√	√	√	FFH
FFFD5H		MK2H			√	√		FFH
FFFD8H	Priority specification flag register 02	PR02L	PR02	R/W	√	√	√	FFH
FFFD9H		PR02H			√	√		FFH
FFFDCH	Priority specification flag register 12	PR12L	PR12	R/W	√	√	√	FFH
FFFDDH		PR12H			√	√		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

Notes 1. The reset value of RESF varies depending on the reset source.

2. The reset value of LVIM varies depending on the reset source and the setting of the option byte.

3. The reset value of LVIS varies depending on the reset source.

4. The reset value of WDTE is determined by the setting of the option byte.

5. Do not directly operate this SFR, because it is to be used in the self programming library.

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
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	Multiplication/division data register A (L)	MDAL/MULA		R/W	–	–	√	0000H
FFFF1H								
FFFF2H	Multiplication/division data register A (H)	MDAH/MULB		R/W	–	–	√	0000H
FFFF3H								
FFFF4H	Multiplication/division data register B (H)	MDBH/MULOH		R/W	–	–	√	0000H
FFFF5H								
FFFF6H	Multiplication/division data register B (L)	MDBL/MULOL		R/W	–	–	√	0000H
FFFF7H								

Remark For extended SFRs (2nd SFRs), see **Table 3-6 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 manipulable bit units, 1, 8, and 16, depend on the SFR type.

Each manipulation bit unit can be specified as follows.

- 1-bit manipulation
  Describe the symbol reserved by the assembler for the 1-bit manipulation instruction operand (!addr16.bit). This manipulation can also be specified with an address.
- 8-bit manipulation
  Describe the symbol reserved 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 reserved by the assembler for the 16-bit manipulation instruction operand (!addr16). When specifying an address, describe an even address.

Table 3-6 gives a list of the extended SFRs. The meanings of items in the table are as follows.

- Symbol
  Symbol indicating the address of an extended SFR. It is a reserved word in the RA78K0R, and is defined as an sfr variable using the #pragma sfr directive in the CC78K0R. When using the RA78K0R, ID78K0R-QB, and SM+ for 78K0R, symbols can be written as an instruction operand.
- R/W
  Indicates whether the corresponding extended SFR can be read or written.
  R/W: Read/write enable
  R: Read only
  W: Write only
- Manipulable bit units
  "√" indicates the manipulable bit unit (1, 8, or 16). "–" indicates a bit unit for which manipulation is not possible.
- After reset
  Indicates each register status upon reset signal generation.

Caution Do not access addresses to which 2nd SFRs are not assigned.

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

Table 3-6. Extended SFR (2nd SFR) List (1/5)

Address	Special Function Register (SFR) Name	Symbol	R/W	Manipulable Bit Range			After Reset
				1-bit	8-bit	16-bit	
F0017H	A/D port configuration register	ADPC	R/W	–	√	–	10H
F0030H	Pull-up resistor option register 0	PU0	R/W	√	√	–	00H
F0031H	Pull-up resistor option register 1	PU1	R/W	√	√	–	00H
F0033H	Pull-up resistor option register 3	PU3	R/W	√	√	–	00H
F0034H	Pull-up resistor option register 4	PU4	R/W	√	√	–	00H
F0035H	Pull-up resistor option register 5	PU5	R/W	√	√	–	00H
F0038H	Pull-up resistor option register 8	PU8	R/W	√	√	–	00H
F003AH	Pull-up resistor option register 10	PU10	R/W	√	√	–	00H
F003CH	Pull-up resistor option register 12	PU12	R/W	√	√	–	00H
F003EH	Pull-up resistor option register 14	PU14	R/W	√	√	–	00H
F0041H	Port input mode register 1	PIM1	R/W	√	√	–	00H
F0051H	Port output mode register 1	POM1	R/W	√	√	–	00H
F0058H	Port output mode register 8	POM8	R/W	√	√	–	00H
F0060H	Noise filter enable register 0	NFEN0	R/W	√	√	–	00H
F0061H	Noise filter enable register 1	NFEN1	R/W	√	√	–	00H
F00E0H	Multiplication/division data register C (L)	MDCL	R	–	–	√	0000H
F00E1H							
F00E2H	Multiplication/division data register C (H)	MDCH	R	–	–	√	0000H
F00E3H							
F00E8H	Multiplication/division control register	MDUC	R/W	√	√	–	00H
F00F0H	Peripheral enable register 0	PER0	R/W	√	√	–	00H
F00F3H	Operation speed mode control register	OSMC	R/W	–	√	–	00H
F00F4H	Regulator mode control register	RMC	R/W	–	√	–	00H
F00F6H	20 MHz internal high-speed oscillation control register	DSCCTL	R/W	√	√	–	00H
F00FEH	BCD adjust result register	BCDADJ	R	–	√	–	Undefined
F0100H	Serial status register 00	SSR00L SSR00	R	–	√	√	0000H
F0101H		–		–	–		
F0102H	Serial status register 01	SSR01L SSR01	R	–	√	√	0000H
F0103H		–		–	–		
F0104H	Serial status register 02	SSR02L SSR02	R	–	√	√	0000H
F0105H		–		–	–		
F0106H	Serial status register 03	SSR03L SSR03	R	–	√	√	0000H
F0107H		–		–	–		
F0108H	Serial flag clear trigger register 00	SIR00L SIR00	R/W	–	√	√	0000H
F0109H		–		–	–		
F010AH	Serial flag clear trigger register 01	SIR01L SIR01	R/W	–	√	√	0000H
F010BH		–		–	–		
F010CH	Serial flag clear trigger register 02	SIR02L SIR02	R/W	–	√	√	0000H
F010DH		–		–	–		
F010EH	Serial flag clear trigger register 03	SIR03L SIR03	R/W	–	√	√	0000H
F010FH		–		–	–		

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F0110H	Serial mode register 00	SMR00		R/W	–	–	√	0020H
F0111H								
F0112H	Serial mode register 01	SMR01		R/W	–	–	√	0020H
F0113H								
F0114H	Serial mode register 02	SMR02		R/W	–	–	√	0020H
F0115H								
F0116H	Serial mode register 03	SMR03		R/W	–	–	√	0020H
F0117H								
F0118H	Serial communication operation setting register 00	SCR00		R/W	–	–	√	0087H
F0119H								
F011AH	Serial communication operation setting register 01	SCR01		R/W	–	–	√	0087H
F011BH								
F011CH	Serial communication operation setting register 02	SCR02		R/W	–	–	√	0087H
F011DH								
F011EH	Serial communication operation setting register 03	SCR03		R/W	–	–	√	0087H
F011FH								
F0120H	Serial channel enable status register 0	SE0L	SE0	R	√	√	√	0000H
F0121H		–			–	–		
F0122H	Serial channel start register 0	SS0L	SS0	R/W	√	√	√	0000H
F0123H		–			–	–		
F0124H	Serial channel stop register 0	ST0L	ST0	R/W	√	√	√	0000H
F0125H		–			–	–		
F0126H	Serial clock select register 0	SPS0L	SPS0	R/W	–	√	√	0000H
F0127H		–			–	–		
F0128H	Serial output register 0	SO0		R/W	–	–	√	0F0FH
F0129H								
F012AH	Serial output enable register 0	SOE0L	SOE0	R/W	√	√	√	0000H
F012BH		–			–	–		
F0134H	Serial output level register 0	SOL0L	SOL0	R/W	–	√	√	0000H
F0135H		–			–	–		
F0140H	Serial status register 10	SSR10L	SSR10	R	–	√	√	0000H
F0141H		–			–	–		
F0142H	Serial status register 11	SSR11L	SSR11	R	–	√	√	0000H
F0143H		–			–	–		
F0144H	Serial status register 12	SSR12L	SSR12	R	–	√	√	0000H
F0145H		–			–	–		
F0146H	Serial status register 13	SSR13L	SSR13	R	–	√	√	0000H
F0147H		–			–	–		
F0148H	Serial flag clear trigger register 10	SIR10L	SIR10	R/W	–	√	√	0000H
F0149H		–			–	–		
F014AH	Serial flag clear trigger register 11	SIR11L	SIR11	R/W	–	√	√	0000H
F014BH		–			–	–		

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F014EH	Serial flag clear trigger register 13	SIR13L	SIR13	R/W	–	√	√	0000H
F014FH		–			–	–		
F0150H	Serial mode register 10	SMR10		R/W	–	–	√	0020H
F0151H								
F0152H	Serial mode register 11	SMR11		R/W	–	–	√	0020H
F0153H								
F0154H	Serial mode register 12	SMR12		R/W	–	–	√	0020H
F0155H								
F0156H	Serial mode register 13	SMR13		R/W	–	–	√	0020H
F0157H								
F0158H	Serial communication operation setting register 10	SCR10		R/W	–	–	√	0087H
F0159H								
F015AH	Serial communication operation setting register 11	SCR11		R/W	–	–	√	0087H
F015BH								
F015CH	Serial communication operation setting register 12	SCR12		R/W	–	–	√	0087H
F015DH								
F015EH	Serial communication operation setting register 13	SCR13		R/W	–	–	√	0087H
F015FH								
F0160H	Serial channel enable status register 1	SE1L	SE1	R	√	√	√	0000H
F0161H		–			–	–		
F0162H	Serial channel start register 1	SS1L	SS1	R/W	√	√	√	0000H
F0163H		–			–	–		
F0164H	Serial channel stop register 1	ST1L	ST1	R/W	√	√	√	0000H
F0165H		–			–	–		
F0166H	Serial clock select register 1	SPS1L	SPS1	R/W	–	√	√	0000H
F0167H		–			–	–		
F0168H	Serial output register 1	SO1		R/W	–	–	√	0F0FH
F0169H								
F016AH	Serial output enable register 1	SOE1L	SOE1	R/W	√	√	√	0000H
F016BH		–			–	–		
F0174H	Serial output level register 1	SOL1L	SOL1	R/W	–	√	√	0000H
F0175H		–			–	–		
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								
F0188H	Timer counter register 04	TCR04		R	–	–	√	FFFFH
F0189H								

Table 3-6. Extended SFR (2nd SFR) List (4/5)

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

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

Address	Special Function Register (SFR) Name	Symbol		R/W	Manipulable Bit Range			After Reset
					1-bit	8-bit	16-bit	
F01B4H	Timer channel stop register 0	TT0L	TT0	R/W	√	√	√	0000H
F01B5H		–			–	–		
F01B6H	Timer clock select register 0	TPS0L	TPS0	R/W	–	√	√	0000H
F01B7H		–			–	–		
F01B8H	Timer output register 0	TO0L	TO0	R/W	–	√	√	0000H
F01B9H		–			–	–		
F01BAH	Timer output enable register 0	TOE0L	TOE0	R/W	√	√	√	0000H
F01BBH		–			–	–		
F01BCH	Timer output level register 0	TOL0L	TOL0	R/W	–	√	√	0000H
F01BDH		–			–	–		
F01BEH	Timer output mode register 0	TOM0L	TOM0	R/W	–	√	√	0000H
F01BFH		–			–	–		
F01C0H	Timer counter register 10	TCR10		R	–	–	√	FFFFH
F01C1H								
F01C2H	Timer counter register 11	TCR11		R	–	–	√	FFFFH
F01C3H								
F01C4H	Timer counter register 12	TCR12		R	–	–	√	FFFFH
F01C5H								
F01C6H	Timer counter register 13	TCR13		R	–	–	√	FFFFH
F01C7H								
F01C8H	Timer mode register 10	TMR10		R/W	–	–	√	0000H
F01C9H								
F01CAH	Timer mode register 11	TMR11		R/W	–	–	√	0000H
F01CBH								
F01CCH	Timer mode register 12	TMR12		R/W	–	–	√	0000H
F01CDH								
F01CEH	Timer mode register 13	TMR13		R/W	–	–	√	0000H
F01CFH								
F01D8H	Timer channel enable status register 1	TE1L	TE1	R	√	√	√	0000H
F01D9H		–			–	–		
F01DAH	Timer channel start register 1	TS1L	TS1	R/W	√	√	√	0000H
F01DBH		–			–	–		
F01DCH	Timer channel stop register 1	TT1L	TT1	R/W	√	√	√	0000H
F01DDH		–			–	–		
F01DEH	Timer clock select register 1	TPS1L	TPS1	R/W	–	√	√	0000H
F01DFH		–			–	–		
F0230H	IICA control register 0	IICCTL0		R/W	√	√	–	00H
F0231H	IICA control register 1	IICCTL1		R/W	√	√	–	00H
F0232H	IICA low-level width setting register	IICWL		R/W	–	√	–	FFH
F0233H	IICA high-level width setting register	IICWH		R/W	–	√	–	FFH
F0234H	Slave address register	SVA		R/W	–	√	–	00H

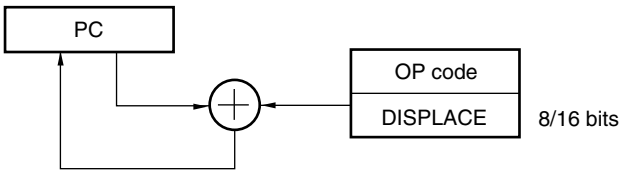
3.3 Instruction Address Addressing

3.3.1 Relative addressing

[Function]

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

Figure 3-14. Outline of Relative Addressing



3.3.2 Immediate addressing

[Function]

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

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

Figure 3-15. Example of CALL !!addr20/BR !!addr20

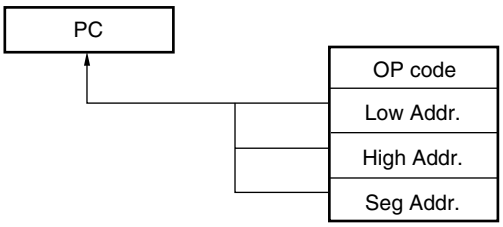
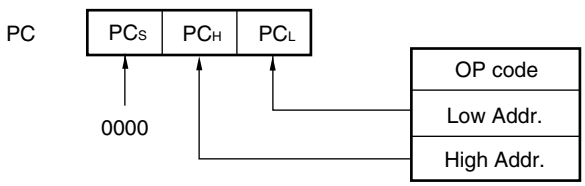


Figure 3-16. Example of CALL !addr16/BR !addr16



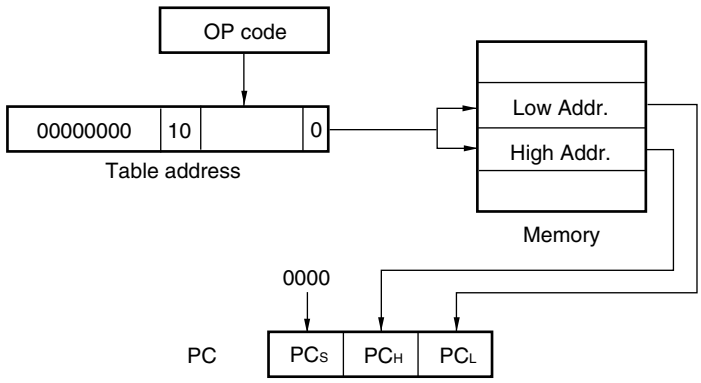
3.3.3 Table indirect addressing

[Function]

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

In the 78K0R microcontrollers, branching is enabled only to the 64 KB space from 00000H to 0FFFFH.

Figure 3-17. Outline of Table Indirect Addressing

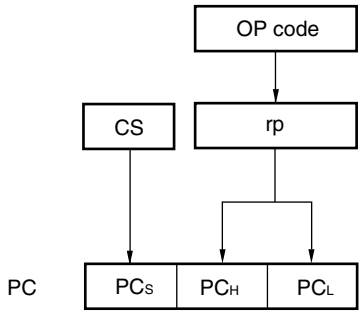


3.3.4 Register direct addressing

[Function]

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

Figure 3-18. Outline of Register Direct Addressing



3.4 Addressing for Processing Data Addresses

3.4.1 Implied addressing

[Function]

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

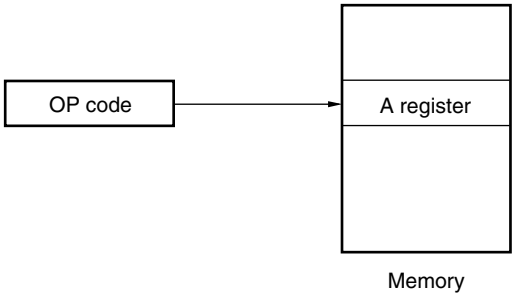
[Operand format]

Because implied addressing can be automatically employed with an instruction, no particular operand format

is necessary.

Implied addressing can be applied only to MULU X.

Figure 3-19. Outline of Implied Addressing



3.4.2 Register addressing

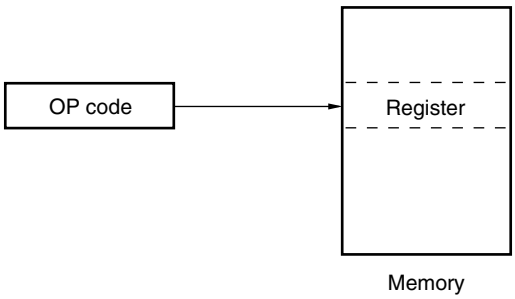
[Function]

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

[Operand format]

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

Figure 3-20. Outline of Register Addressing



3.4.3 Direct addressing

[Function]

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

[Operand format]

Identifier	Description
ADDR16	Label or 16-bit immediate data (only the space from F0000H to FFFFFH is specifiable)
ES: ADDR16	Label or 16-bit immediate data (higher 4-bit addresses are specified by the ES register)

Figure 3-21. Example of ADDR16

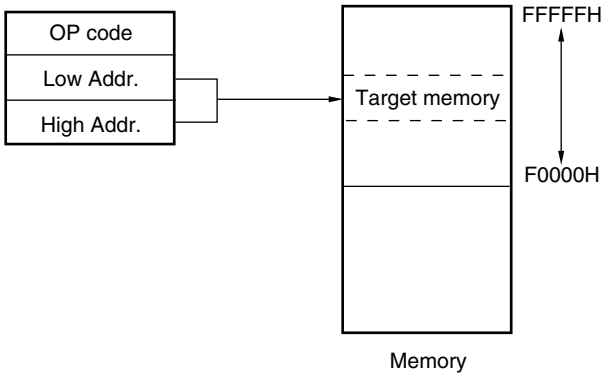
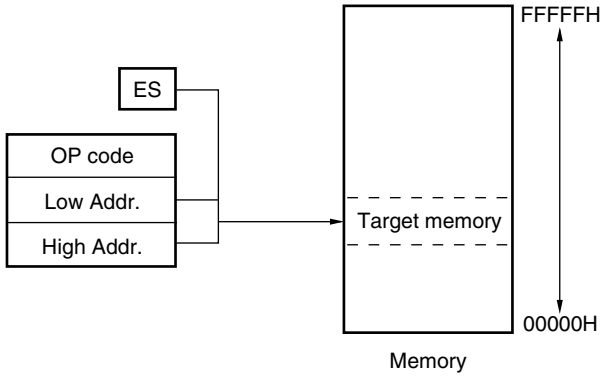


Figure 3-22. Example of ES:ADDR16



3.4.4 Short direct addressing

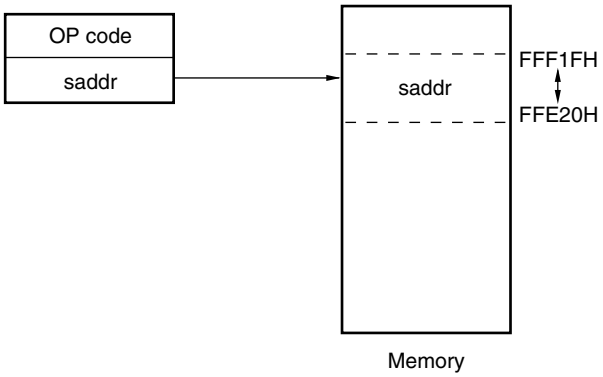
[Function]

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

[Operand format]

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

Figure 3-23. Outline of Short Direct Addressing



Remark SADDR and SADDRP are used to describe the values of addresses FE20H to FF1FH with 16-bit immediate data (higher 4 bits of actual address are omitted), and the values of addresses FFE20H to FFF1FH with 20-bit immediate data.

Regardless of whether SADDR or SADDRP is used, addresses within the space from FFE20H to FFF1FH are specified for the memory.

3.4.5 SFR addressing

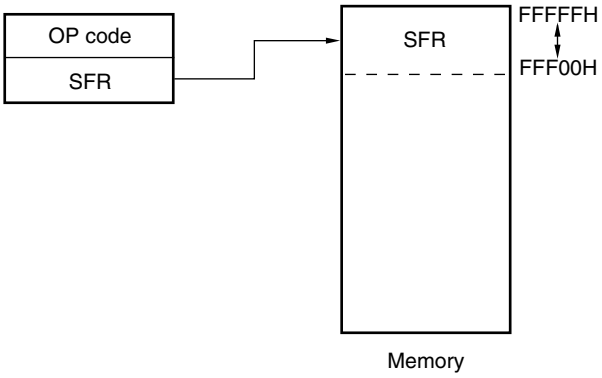
[Function]

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

[Operand format]

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

Figure 3-24. Outline of SFR Addressing



3.4.6 Register indirect addressing

[Function]

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

[Operand format]

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

Figure 3-25. Example of [DE], [HL]

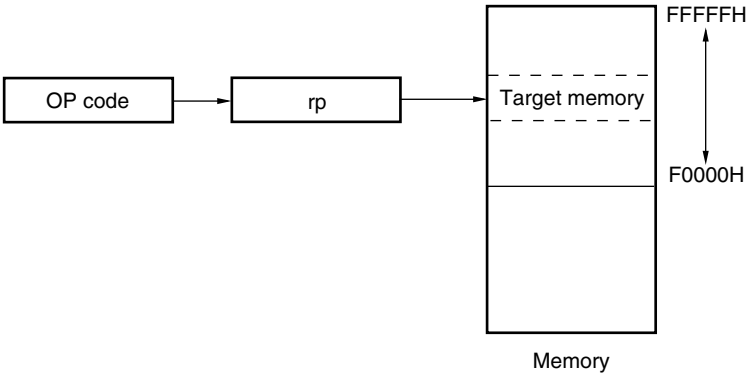
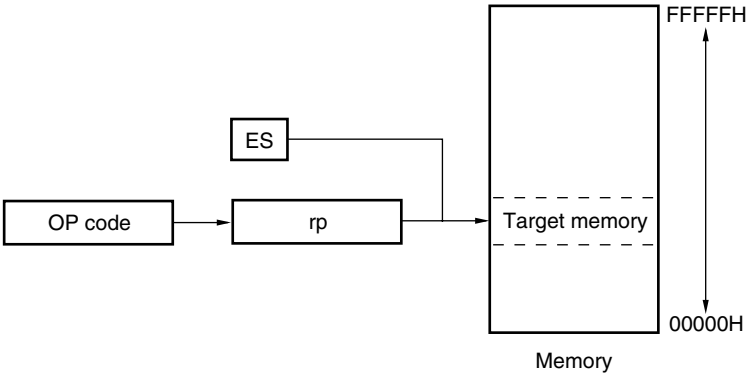


Figure 3-26. Example of ES:[DE], ES:[HL]



3.4.7 Based addressing

[Function]

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

[Operand format]

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

Figure 3-27. Example of [SP+byte]

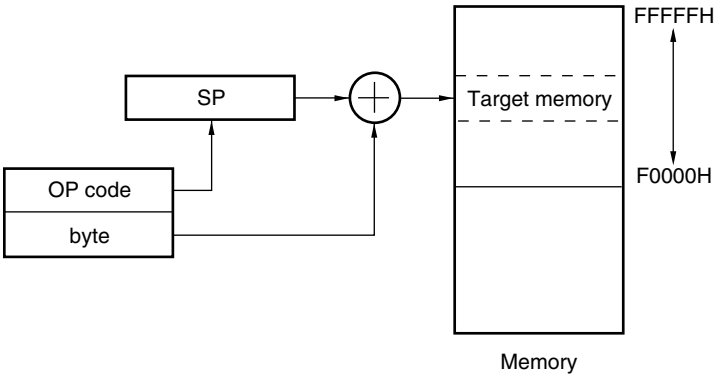


Figure 3-28. Example of [HL + byte], [DE + byte]

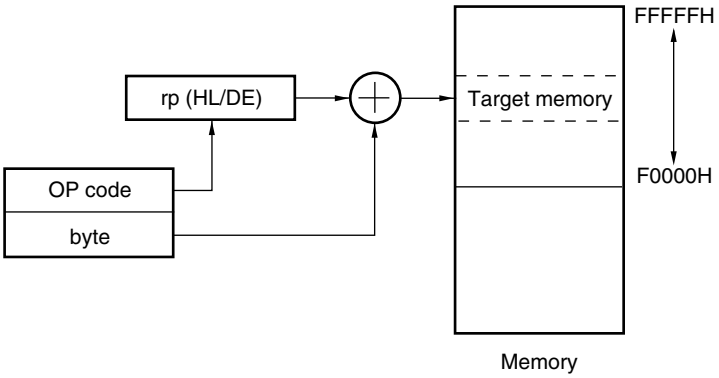


Figure 3-29. Example of word[B], word[C]

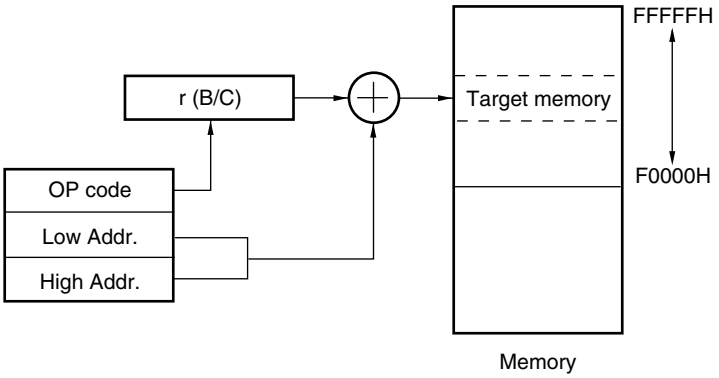


Figure 3-30. Example of word[BC]

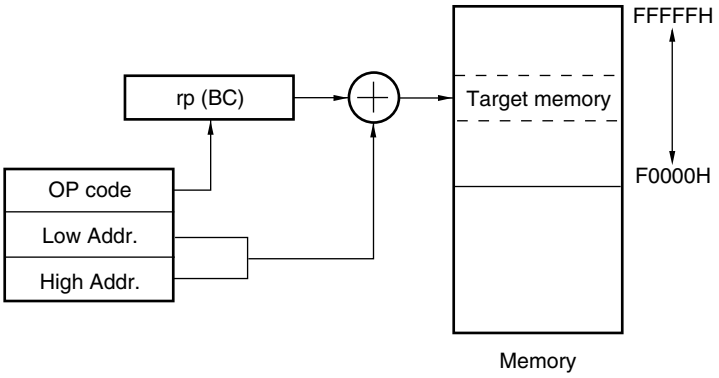


Figure 3-31. Example of ES:[HL + byte], ES:[DE + byte]

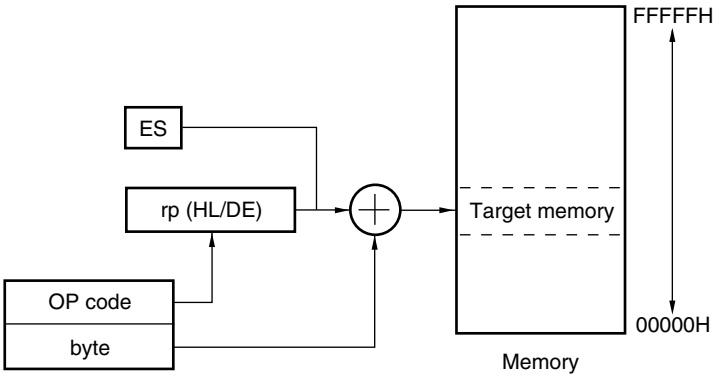


Figure 3-32. Example of ES:word[B], ES:word[C]

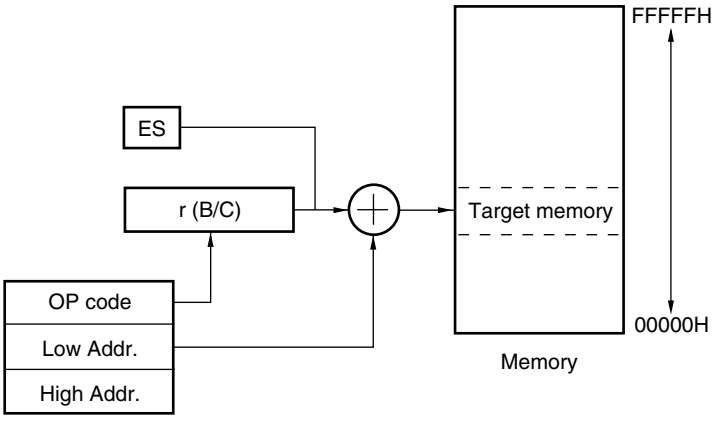
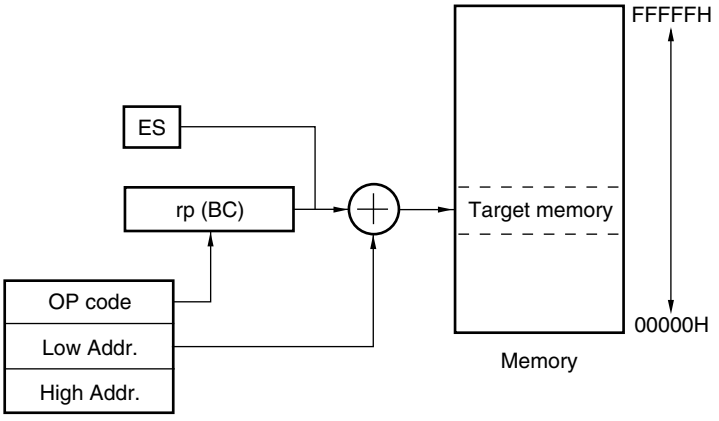


Figure 3-33. Example of ES:word[BC]



3.4.8 Based indexed addressing

[Function]

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

[Operand format]

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

Figure 3-34. Example of [HL+B], [HL+C]

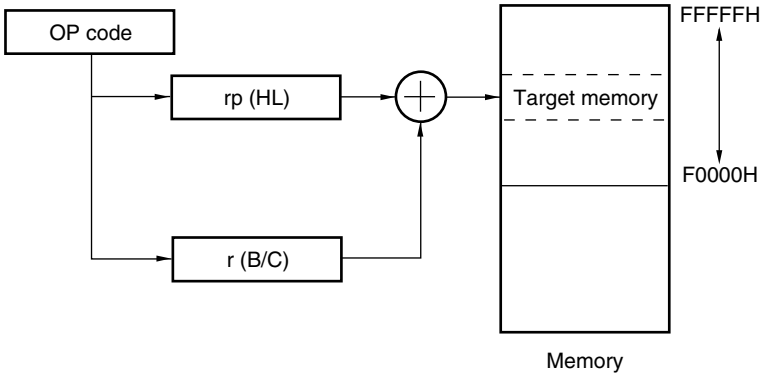
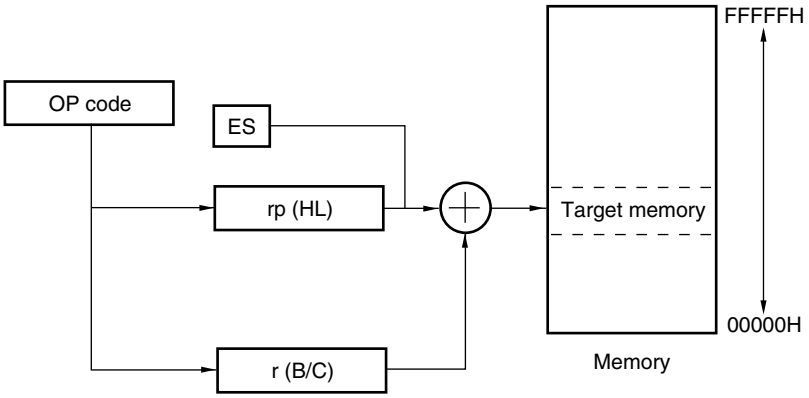


Figure 3-35. Example of ES:[HL+B], ES:[HL+C]



3.4.9 Stack addressing

[Function]

The stack area is indirectly addressed with the stack pointer (SP) contents. This addressing is automatically employed when the PUSH, POP, subroutine call, and return instructions are executed or the register is saved/restored upon generation of an interrupt request.

Stack addressing is applied only to the internal RAM area.

[Operand format]

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

CHAPTER 4 PORT FUNCTIONS

4.1 Port Functions

There are four types of pin I/O buffer power supplies: AV_{DD0}, AV_{DD1}, EV_{DD}, and V_{DD}. The relationship between these power supplies and the pins is shown below.

Table 4-1. Pin I/O Buffer Power Supplies

Power Supply	Corresponding Pins
AV_{DD0}	P20 to P27, P150 to P152, and P157
AV_{DD1}	P110 and P111
EV_{DD}	• Port pins other than P20 to P27, P110, P111, P150 to P152, and P157 • $\overline{\text{RESET}}$ and FLMD0 pins
V_{DD}	Pins other than port , $\overline{\text{RESET}}$, and FLMD0 pins

The products of 78K0R/Kx3-A microcontrollers are provided with digital I/O ports, which enable variety of control operations. The functions of each port are shown in Tables 4-2.

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**.

Table 4-2. Port Functions (1/2)

Function Name	I/O	Function	After Reset	Alternate Function
P00	I/O	Port 0. 3-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	–
P01				–
P02				–
P10	I/O	Port 1. 7-bit I/O port. Input/output can be specified in 1-bit units. Input of P10, P11, P14 and P15 can be set to TTL buffer. Output of P10 to P15 can be set to N-ch open-drain output (V_{DD} tolerance). Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	$\overline{SCK20}$/SCL20
P11				SI20/RxD2/SDA20/ INTP6
P12				SO20/TxD2/TO02
P13				SO10/TxD1/TO04
P14				SI10/RxD1/SDA10/ INTP4
P15				$\overline{SCK10}$/SCL10/INTP7
P16				TI05/TO05/INTP10
P20	I/O	Port 2. 8-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI0/AMP0-
P21				ANI1/AMP0O
P22				ANI2/AMP0+
P23				ANI3/AMP1-
P24				ANI4/AMP1O
P25				ANI5/AMP1+
P26				ANI6/AMP2-
P27				ANI7/AMP2O
P30	I/O	Port 3. 5-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TI03/TO00/RTC1HZ/ INTP1
P31				TI00/TO03/RTCDIV/ RTCCL/PCLBUZ1/ INTP2
P32				TI01/TO01/INTP5/ PCLBUZ0
P33				TI07/TO07/INTP3
P34				TI06/TO06/INTP8
P40 Note	I/O	Port 4. 2-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	TOOL0
P41				TOOL1

Note If on-chip debugging is enabled by using an option byte, be sure to pull up the P40/TOOL0 pin externally.

Table 4-2. Port Functions (2/2)

Function Name	I/O	Function	After Reset	Alternate Function
P50	I/O	Port 5. 8-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	RxD3
P51				TxD3
P52				TI02
P53				TI04
P54 to P57				–
P60	I/O	Port 6. 2-bit I/O port. Output is N-ch open-drain output (6 V tolerance). Input/output can be specified in 1-bit units.	Input port	SCL0
P61				SDA0
P80	I/O	Port 8. 3-bit I/O port. Inputs/output can be specified in 1-bit units. Output of P80 and P82 can be set to N-ch open-drain output (V_{DD} tolerance). Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	$\overline{SCK00}$/INTP11
P81				RxD0/SI00/INTP9
P82				TxD0/SO00
P100	I/O	Port 10. 1-bit I/O port. Inputs/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	–
P110	I/O	Port 11. 2-bit I/O port. Inputs/output can be specified in 1-bit units.	Input port	ANO0
P111				ANO1
P120	I/O	Port 12. 1-bit I/O port and 4-bit input port. For only P120, input/output can be specified in 1-bit units. For only P120, use of an on-chip pull-up resistor can be specified by a software setting.	Input port	INTP0/EXLVI
P121	Input			X1
P122				X2/EXCLK
P123				XT1
P124				XT2
P130	Output	Port 13. 1-bit output port.	Output port	–
P140, P141	I/O	Port 14. 2-bit I/O port. Input/output can be specified in 1-bit units. Use of an on-chip pull-up resistor can be specified by a software setting.	Input port	–
P150	I/O	Port 15. 4-bit I/O port. Input/output can be specified in 1-bit units.	Digital input port	ANI8/AMP2+
P151				ANI9
P152				ANI10
P157				ANI15/AV_{REFM}

4.2 Port Configuration

Ports include the following hardware.

Table 4-3. Port Configuration

Item	Configuration
Control registers	Port mode registers (PMxx) : PM0 to PM6, PM8, PM10 to PM12, PM14, PM15 Port registers (Pxx) : P0 to P6, P8, P10 to P15 Pull-up resistor option registers (PUxx) : PU0, PU1, PU3 to PU5, PU8, PU10, PU12, PU14 Port input mode registers 1 (PIM1) Port output mode registers 1, 8 (POM1, POM8) A/D port configuration register (ADPC) Input switch control register (ISC)
Port	Total: 53 (CMOS I/O: 46, CMOS output: 1, CMOS input: 4, N-ch open drain I/O: 2)
Pull-up resistor	Total: 32

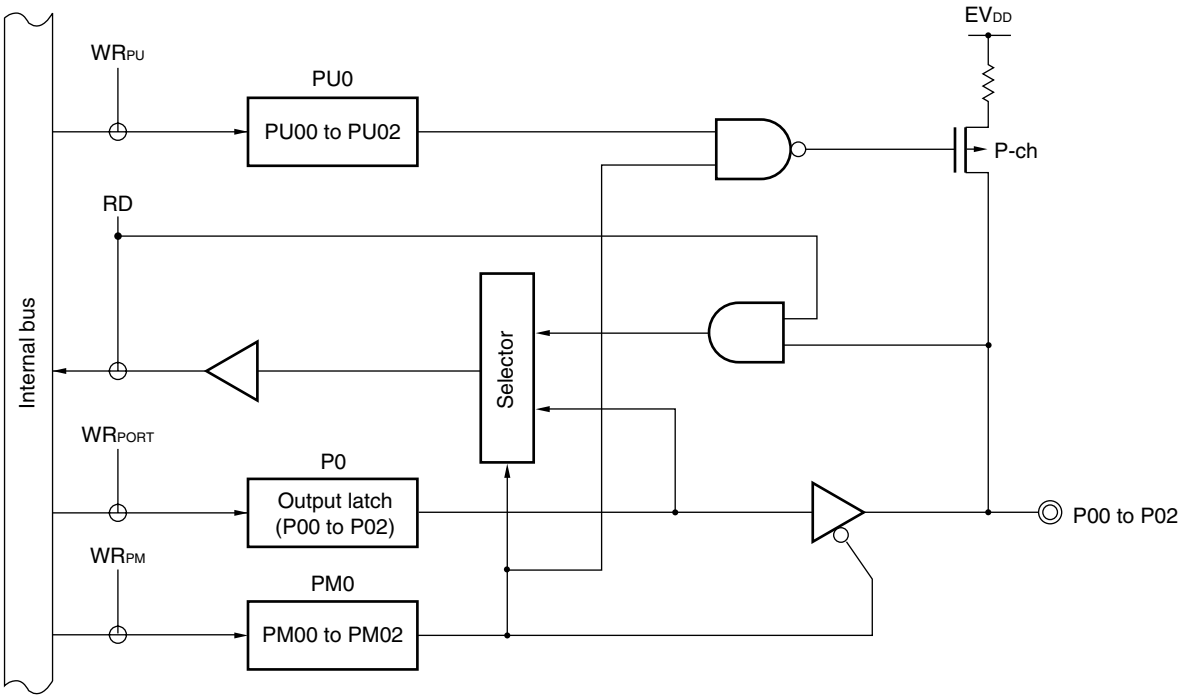
4.2.1 Port 0

Port 0 is a 3-bit I/O port with an output latch. Port 0 can be set to the input mode or output mode in 1-bit units using port mode register 0 (PM0). When the P00 to P02 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 0 (PU0).

Reset signal generation sets port 0 to input mode.

Figure 4-1 shows a block diagram of port 0.

Figure 4-1. Block Diagram of P00 to P02



P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
RD: Read signal
WR××: Write signal

4.2.2 Port 1

Port 1 is a 7-bit I/O port with an output latch. Port 1 can be set to the input mode or output mode in 1-bit units using port mode register 1 (PM1). When the P10 to P16 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 1 (PU1).

Input to the P10, P11, P14, and P15 pins can be specified through a normal input buffer or a TTL input buffer in 1-bit units 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 using port output mode register 1 (POM1).

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

Reset signal generation sets port 1 to input mode.

Figures 4-2 to 4-4 show block diagrams of port 1.

Cautions 1. To use P10/$\overline{\text{SCK20}}$/SCL20, P11/SI20/RxD2/SDA20/INTP6 as a general-purpose port, note the serial array unit 1 setting. For details, refer to Table 14-9 Relationship Between Register Settings and Pins (Channel 0 of unit 1: CSI20, UART2 transmission, IIC20).

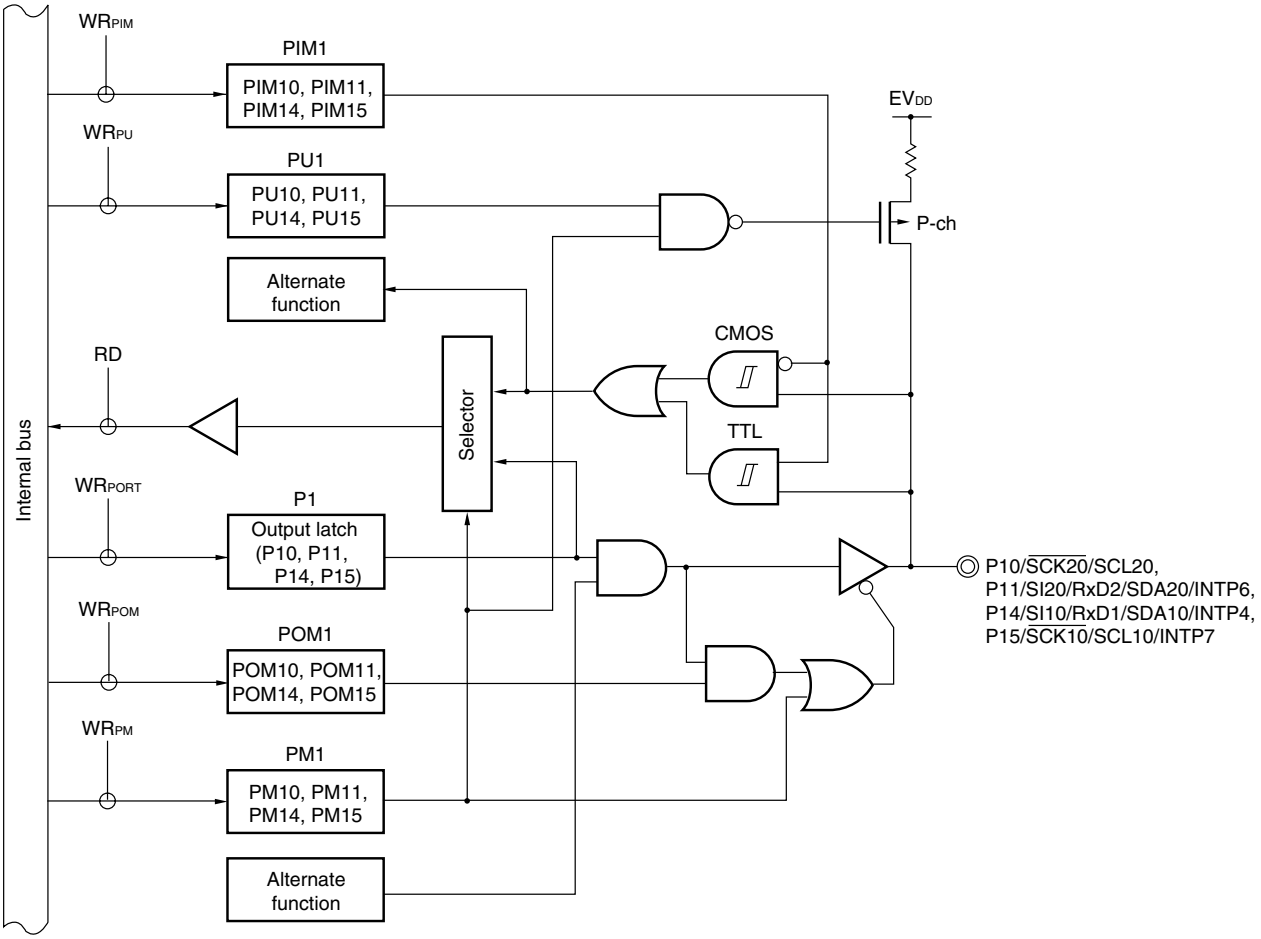
2. To use P12/TO02/SO20/TxD2 as a general-purpose port, set bit 2 (TO02) of timer output register 0 (TO0) and bit 2 (TOE02) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting. And as a general-purpose port, note the serial array unit 1 setting. For details of serial array unit 1 setting, refer to Table 14-9 Relationship Between Register Settings and Pins (Channel 0 of unit 1: CSI20, UART2 transmission, IIC20).

3. To use P13/TO04/SO10/TxD1 as a general-purpose port, set bit 4 (TO04) of timer output register 0 (TO0) and bit 4 (TOE04) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting. And as a general-purpose port, note the serial array unit 0 setting. For details of serial array unit 0 setting, refer to Table 14-7 Relationship Between Register Settings and Pins (Channel 2 of unit 0: CSI10, UART1 transmission, IIC10)

4. To use P14/SI10/RxD1/SDA10/INTP4, P15/$\overline{\text{SCK10}}$/SCL10/INTP7 as a general-purpose port, note the serial array unit 0 setting. For details, refer to Table 14-7 Relationship Between Register Settings and Pins (Channel 2 of unit 0: CSI10, UART1 transmission, IIC10)

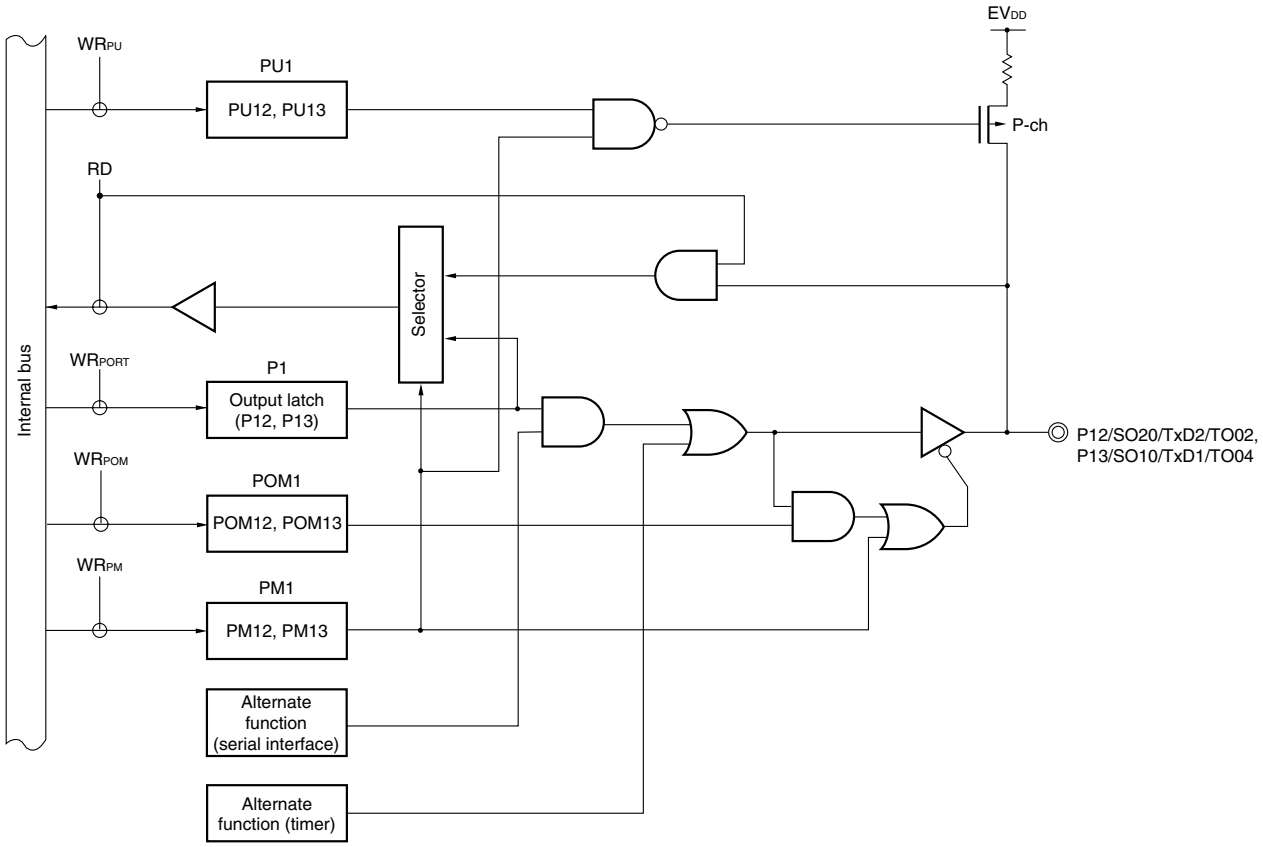
5. To use P16/TO05/TI05/INTP10 as a general-purpose port, set bit 5 (TO05) of timer output register 0 (TO0) and bit 5 (TOE05) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.

Figure 4-2. Block Diagram of P10, P11, P14, and P15



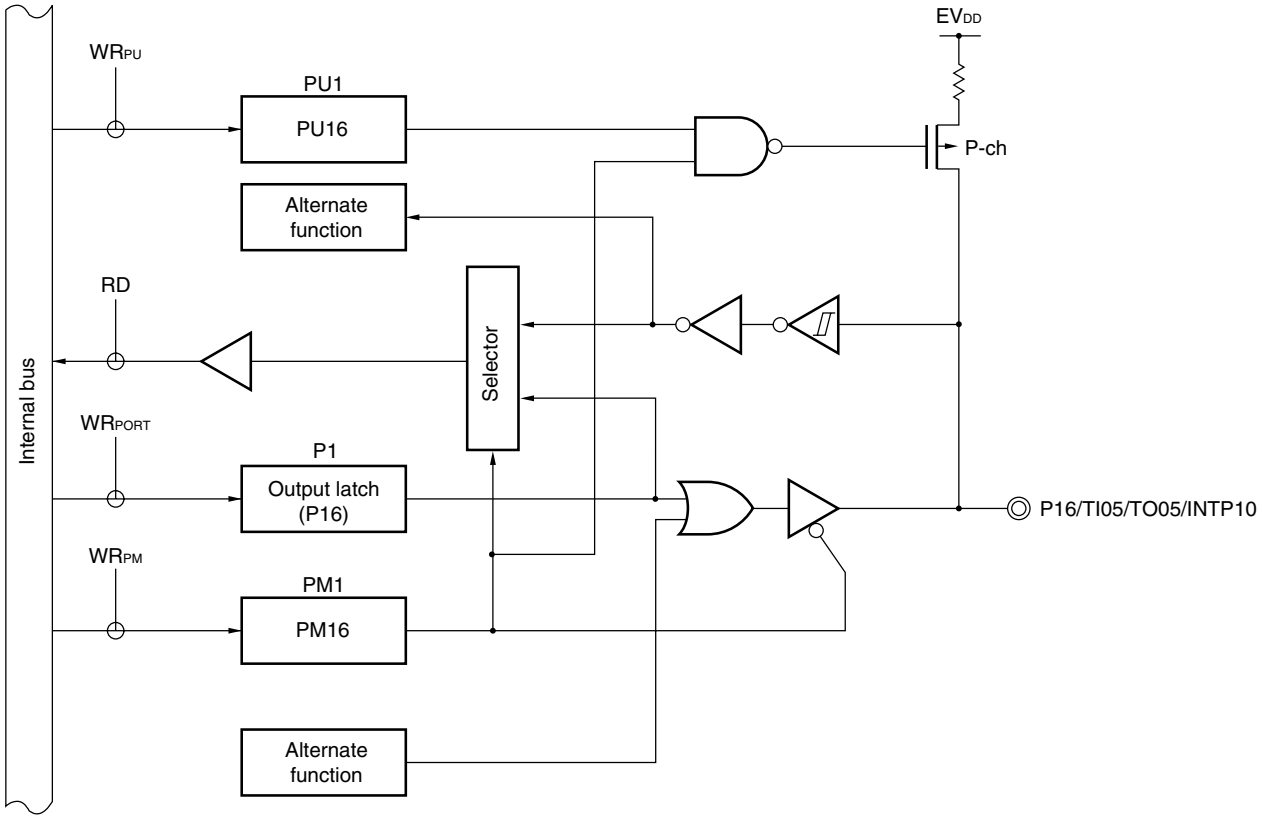
P1: Port register 1
PU1: Pull-up resistor option register 1
PIM1: Port input mode register 1
POM1: Port output mode register 1
PM1: Port mode register 1
RD: Read signal
WR××: Write signal

Figure 4-3. Block Diagram of P12 and P13



P1: Port register 1
PU1: Pull-up resistor option register 1
POM1: Port output mode register 1
PM1: Port mode register 1
RD: Read signal
WR××: Write signal

Figure 4-4. Block Diagram of P16



P1: Port register 1
PU1: Pull-up resistor option register 1
PM1: Port mode register 1
RD: Read signal
WR××: Write signal

4.2.3 Port 2

Port 2 is an 8-bit I/O port with an output latch. Port 2 can be set to the input mode or output mode in 1-bit units using port mode register 2 (PM2).

This port can also be used for A/D converter analog input, and operational amplifier I/O.

To use P20/ANI0/AMP0- to P27/ANI7/AMP2O as digital input pins, set them in the digital I/O mode by using the A/D port configuration register (ADPC) and in the input mode by using PM2. Use these pins starting from the lower bit.

To use P20/ANI0/AMP0- to P27/ANI7/AMP2O as digital output pins, set them in the digital I/O mode by using ADPC and in the output mode by using PM2.

To use P20/ANI0/AMP0- to P27/ANI7/AMP2O as analog input pins, set them in the analog input mode by using ADPC and in the input mode by using PM2. Use these pins starting from the upper bit.

All P20/ANI0/AMP0- to P27/ANI7/AMP2O are set in the digital input mode when the reset signal is generated.

Figures 4-5 to 4-7 show block diagrams of port 2.

Caution Make the AV_{DD0} pin the same potential as the EV_{DD} or V_{DD} pin when port 2 is used as a digital port.

Table 4-4. Setting Functions of ANI0/AMP0-/P20, ANI2/AMP0+/P22, ANI3/AMP1-/P23, ANI5/AMP1+/P25, and ANI6/AMP2-/P26 Pins

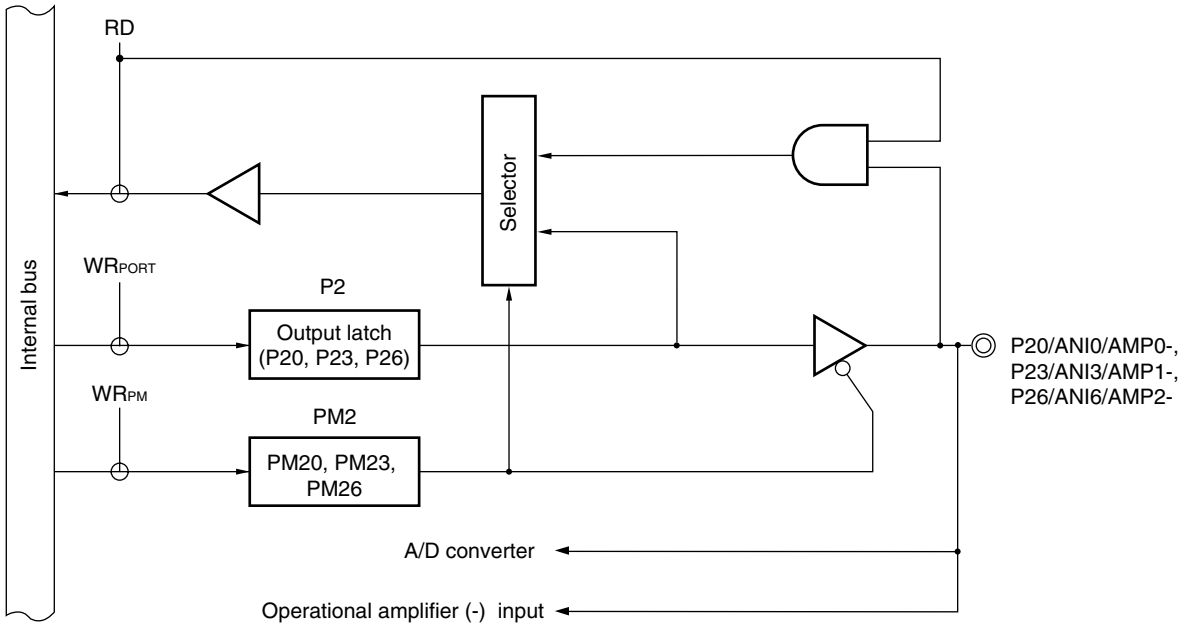
ADPC register	PM2 register	OAENn bit	ADS register	ANI0/AMP0-/P20, ANI2/AMP0+/P22, ANI3/AMP1-/P23, ANI5/AMP1+/P25, and ANI6/AMP2-/P26 Pins
Digital I/O selection	Input mode	0	–	Digital input
		1	–	Setting prohibited
	Output mode	0	–	Digital output
		1	–	Setting prohibited
Analog input selection	Input mode	0	Selects ANI.	Analog input (to be converted)
			Does not select ANI.	Analog input (not to be converted)
		1	Selects ANI.	Setting prohibited
			Does not select ANI.	Operational amplifier input
	Output mode	–	–	Setting prohibited

Table 4-5. Setting Functions of ANI1/AMP0O/P21, ANI4/AMP1O/P24, and ANI7/AMP2O/P27 Pins

ADPC register	PM2 register	OAENn bit	ADS register	ANI1/AMP0O/P21, ANI4/AMP1O/P24, and ANI7/AMP2O/P27 Pins
Digital I/O selection	Input mode	0	–	Digital input
		1	–	Setting prohibited
	Output mode	0	–	Digital output
		1	–	Setting prohibited
Analog input selection	Input mode	0	Selects ANI.	Analog input (to be converted)
			Does not select ANI.	Analog input (not to be converted)
		1	Selects ANI.	Operational amplifier output (to be converted)
			Does not select ANI.	Operational amplifier output (not to be converted)
	Output mode	–	–	Setting prohibited

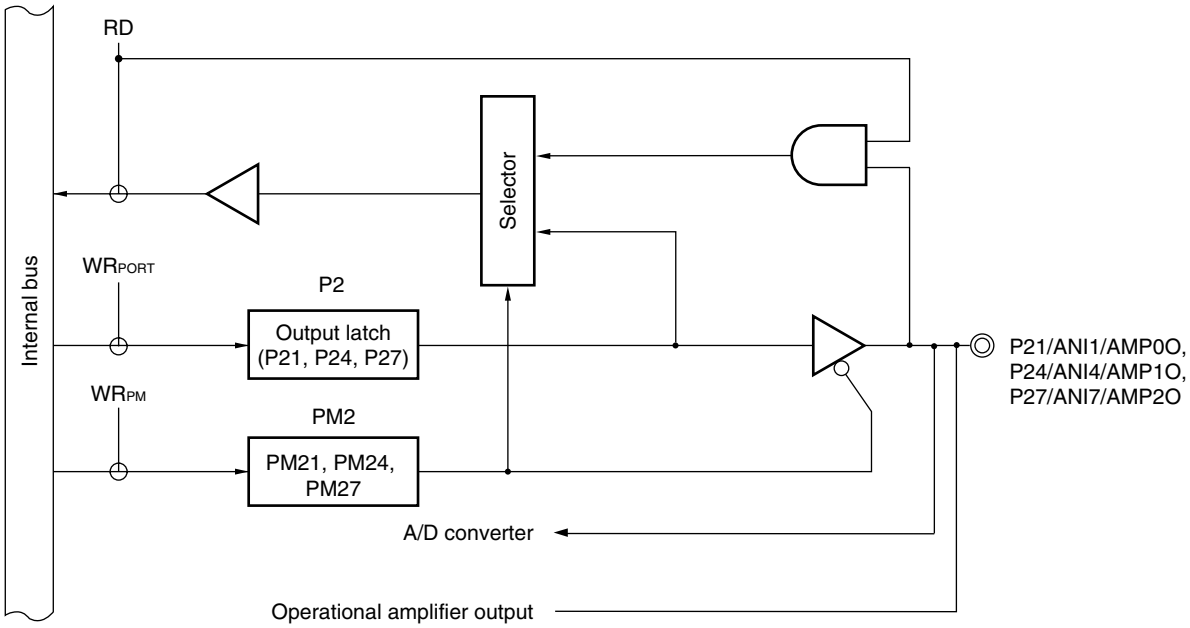
Remark n = 0 to 2

Figure 4-5. Block Diagram of P20, P23, and P26



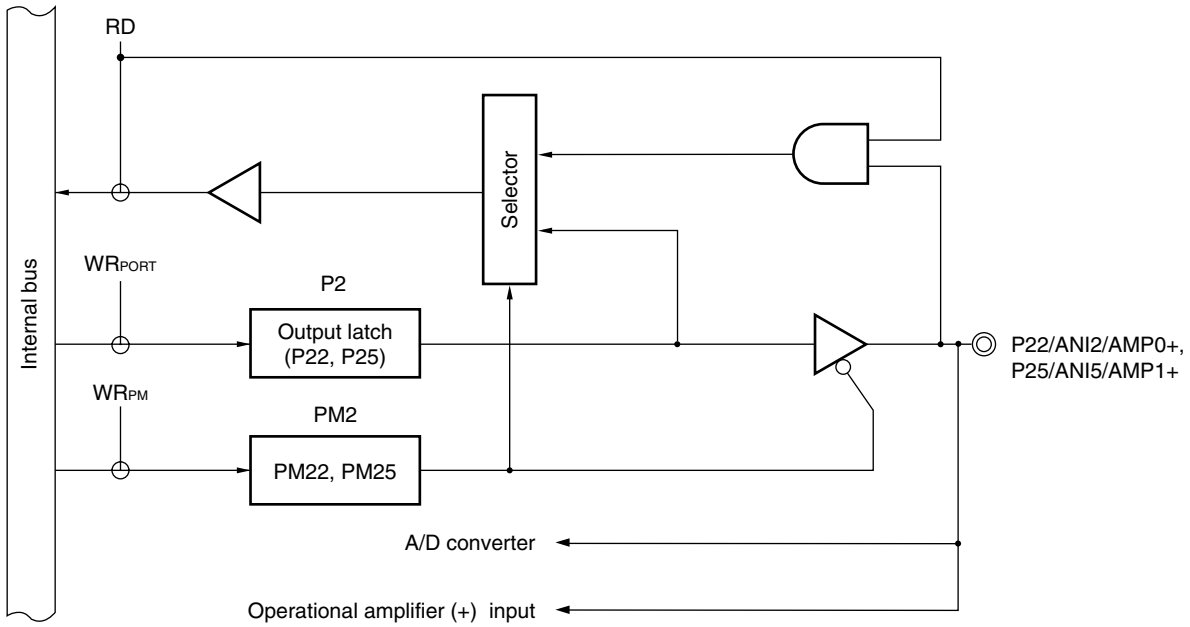
P2: Port register 2
PM2: Port mode register 2
RD: Read signal
WR××: Write signal

Figure 4-6. Block Diagram of P21, P24, and P27



P2: Port register 2
PM2: Port mode register 2
RD: Read signal
WR××: Write signal

Figure 4-7. Block Diagram of P22 to P25



P2: Port register 2
PM2: Port mode register 2
RD: Read signal
WR××: Write signal

4.2.4 Port 3

Port 3 is a 5-bit I/O port with an output latch. Port 3 can be set to the input mode or output mode in 1-bit units using port mode register 3 (PM3). When the P30 to P34 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 3 (PU3).

This port can also be used for timer I/O, real-time counter clock output, correction clock output, clock output/buzzer output, and external interrupt request input.

Reset signal generation sets port 3 to input mode.

Figure 4-8 shows a block diagram of port 3.

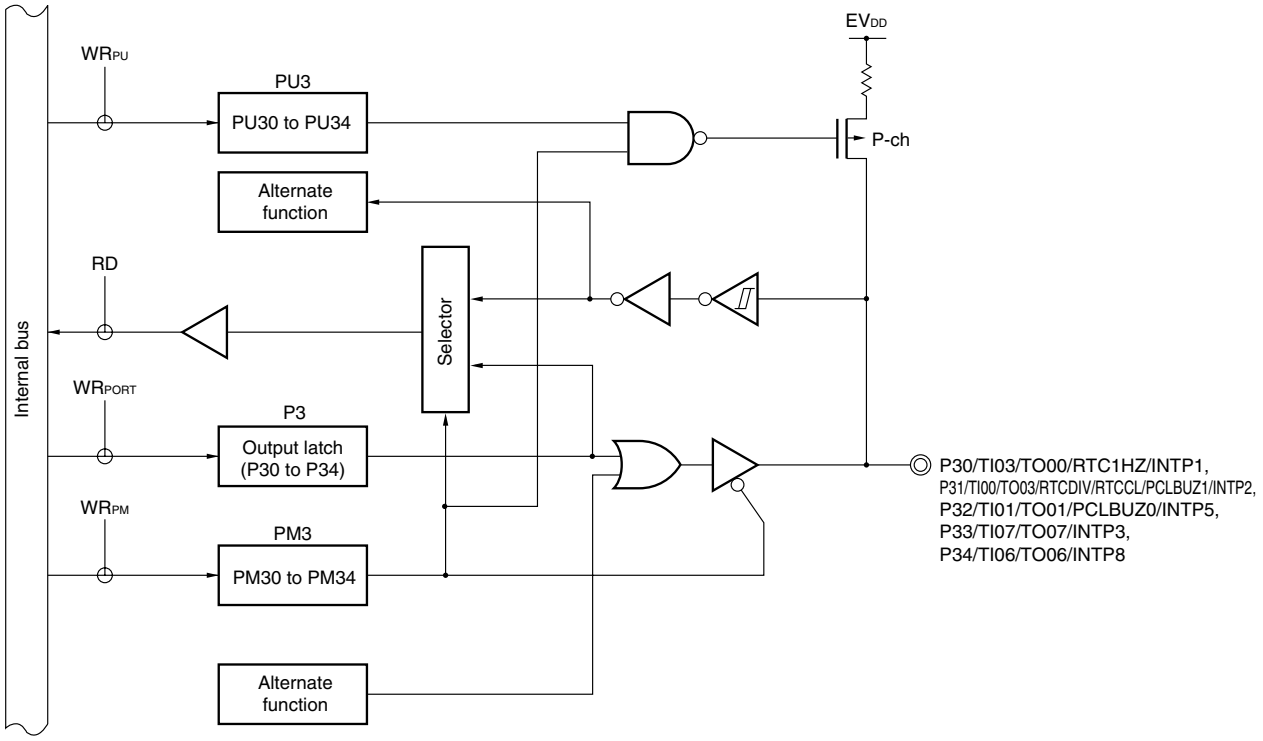
Cautions 1. To use P30/TO00/TI03/RTC1HZ/INTP1 as a general-purpose port, set bit 5 (RCLOE1) of real-time counter control register 0 (RTCC0), bit 0 (TO00) of timer output register 0 (TO0) and bit 0 (TOE00) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.

2. To use P31/TO03/TI00/RTCDIV/RTCCL/PCLBUZ1/INTP2 as a general-purpose port, set bit 4 (RCLOE0) of real-time counter control register 0 (RTCC0), bit 6 (RCLOE2) of real-time counter control register 2 (RTCC2), bit 3 (TO03) of timer output register 0 (TO0), bit 3 (TOE03) of timer output enable register 0 (TOE0) and bit 7 of clock output select register 1 (CKS1) to "0", which is the same as their default status setting.

3. To use P32/TO01/TI01/INTP5/PCLBUZ0 as a general-purpose port, set bit 1 (TO01) of timer output register 0 (TO0), bit 1 (TOE01) of timer output enable register 0 (TOE0) and bit 7 of clock output select register 0 (CKS0) to "0", which is the same as their default status setting.

4. To use P33/TO07/TI07/INTP3, P34/TO06/TI06/INTP8 as a general-purpose port, set bit 7, 6 (TO07, TO06) of timer output register 0 (TO0), and bit 7, 6 (TOE07, TOE06) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.

Figure 4-8. Block Diagram of P30 to P34



P3: Port register 3
PU3: Pull-up resistor option register 3
PM3: Port mode register 3
RD: Read signal
WR××: Write signal

4.2.5 Port 4

Port 4 is a 2-bit I/O port with an output latch. Port 4 can be set to the input mode or output mode in 1-bit units using port mode register 4 (PM4). When the P40 and P41 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 4 (PU4)[Note].

This port can also be used for flash memory programmer/debugger data I/O and debugger clock output.

Reset signal generation sets port 4 to input mode.

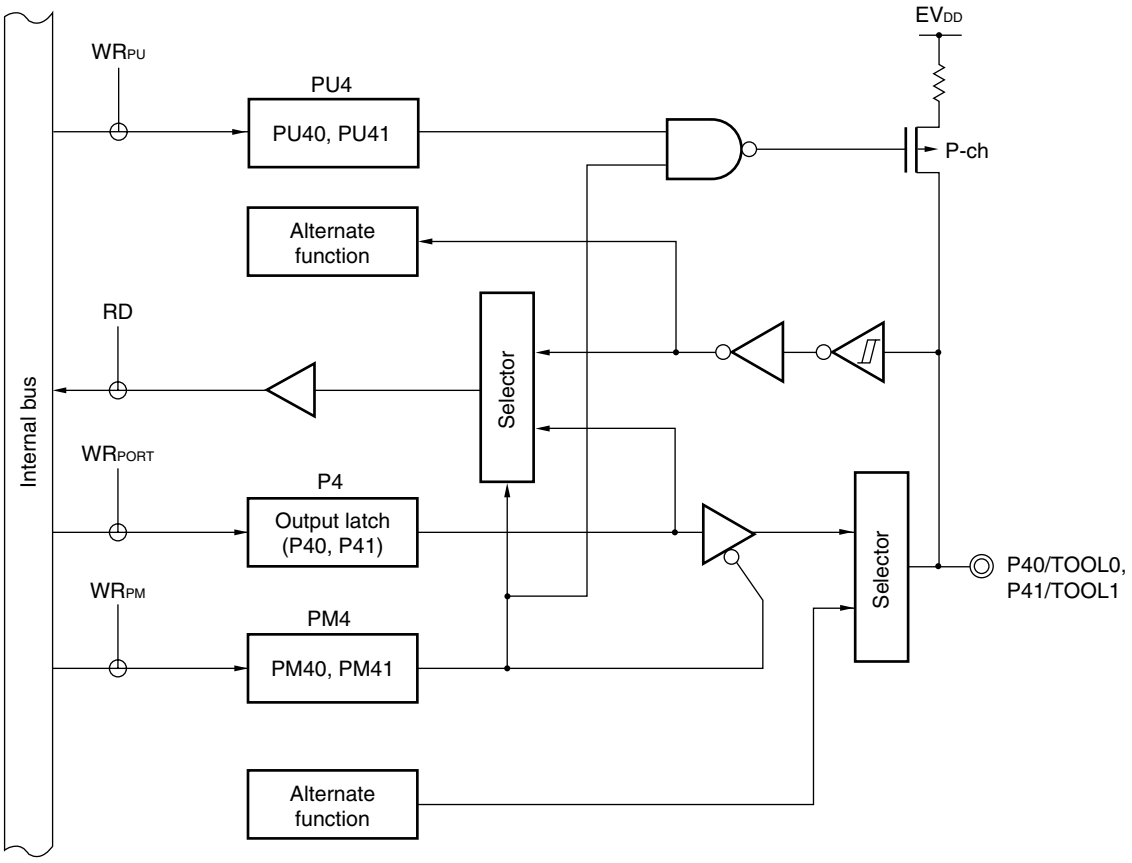
Figure 4-9 shows a block diagram of port 4.

Note When a tool is connected, the P40 and P41 pins cannot be connected to a pull-up resistor.

Caution When a tool is connected, the P40 pin cannot be used as a port pin.
When the on-chip debug function is used, P41 pin can be used as follows by the mode setting on the debugger.

- **1-line mode: can be used as a port (P41).**
- **2-line mode: used as a TOOL1 pin and cannot be used as a port (P41).**

Figure 4-9. Block Diagram of P40, P41



P4: Port register 4
PU4: Pull-up resistor option register 4
PM4: Port mode register 4
RD: Read signal
WR××: Write signal

4.2.6 Port 5

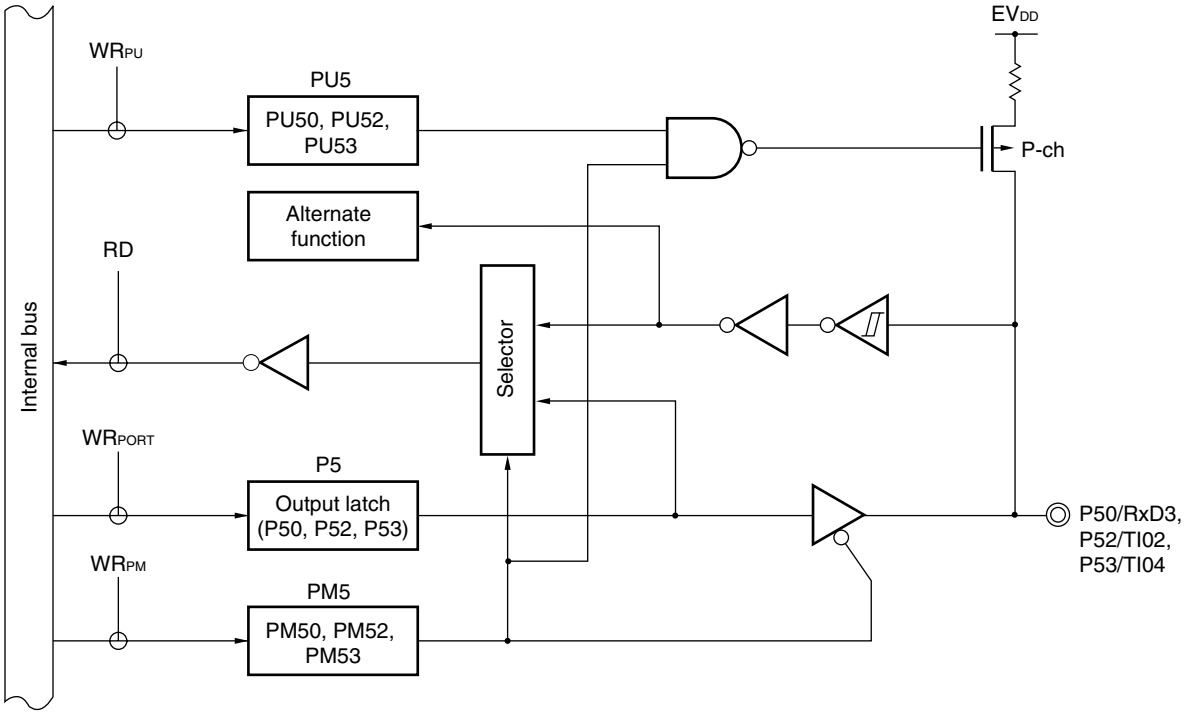
Port 5 is an 8-bit I/O port with an output latch. Port 5 can be set to the input mode or output mode in 1-bit units using port mode register 5 (PM5). When the P50 to P57 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 5 (PU5).

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

Reset signal generation sets port 5 to input mode.

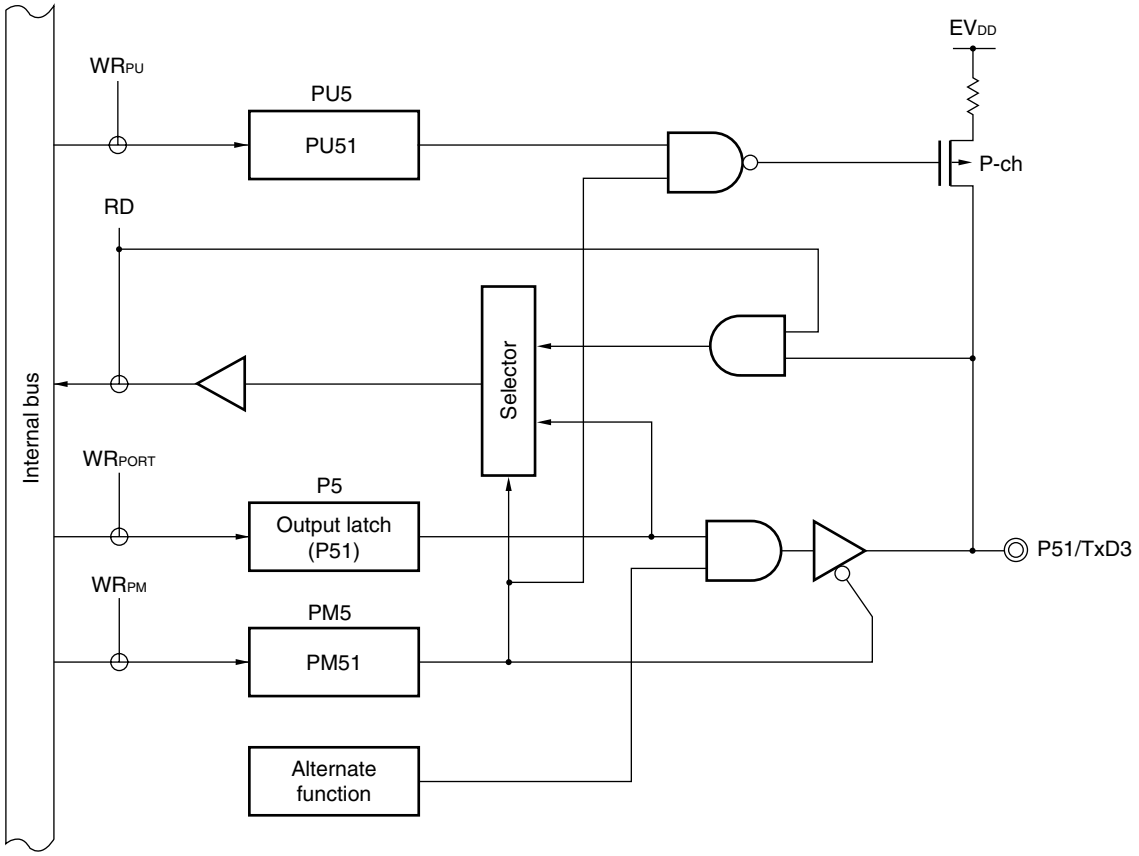
Figures 4-10 to 4-12 show block diagrams of port 5.

Figure 4-10. Block Diagram of P50, P52, and P53



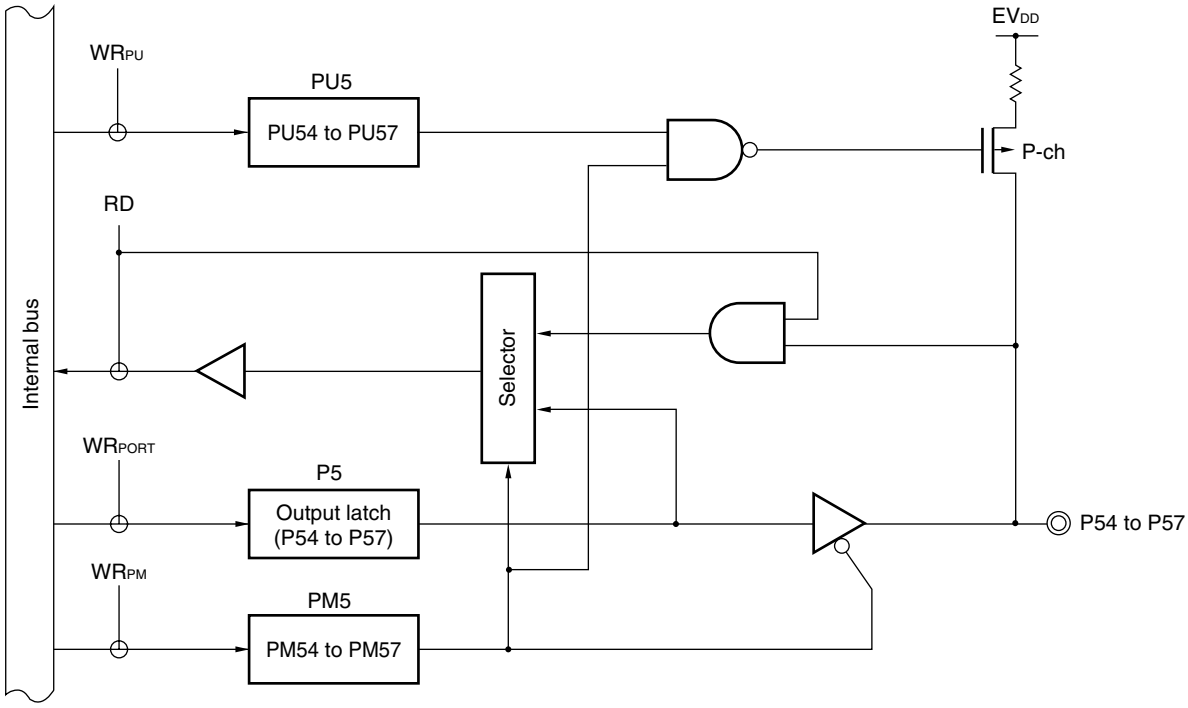
P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
RD: Read signal
WR××: Write signal

Figure 4-11. Block Diagram of P51



P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
RD: Read signal
WR××: Write signal

Figure 4-12. Block Diagram of P54 to P57



P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
RD: Read signal
WR××: Write signal

4.2.7 Port 6

Port 6 is a 2-bit I/O port with an output latch. Port 6 can be set to the input mode or output mode in 1-bit units using port mode register 6 (PM6).

The output is N-ch open-drain output (6 V tolerance).

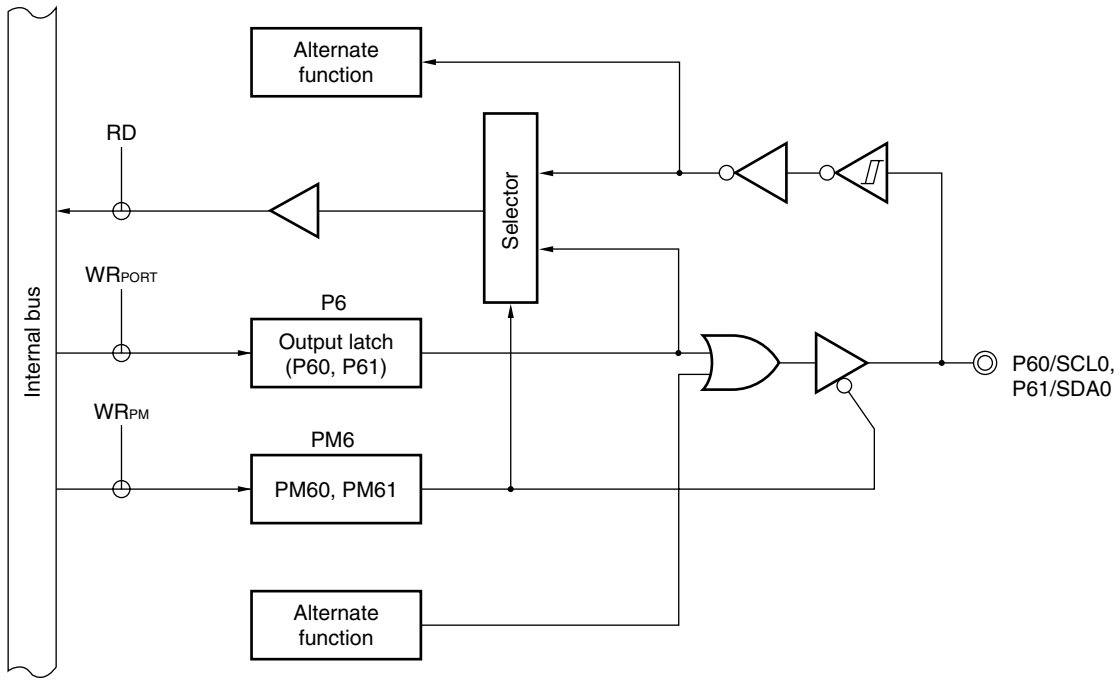
This port can also be used for serial interface data I/O and clock I/O.

Reset signal generation sets port 6 to input mode.

Figure 4-13 shows a block diagram of port 6.

Caution When using P60/SCL0 or P61/SDA0 as a general-purpose port, stop the operation of serial interface IICA.

Figure 4-13. Block Diagram of P60 and P61



P6: Port register 6
PM6: Port mode register 6
RD: Read signal
WR××: Write signal

4.2.8 Port 8

Port 8 is a 3-bit I/O port with an output latch. Port 8 can be set to the input mode or output mode in 1-bit units using port mode register 8 (PM8). When the P80 to P82 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 8 (PU8).

Output from the P80 and P82 pins can be specified as N-ch open-drain output (V_{DD} tolerance) in 1-bit units using port output mode register 8 (POM8).

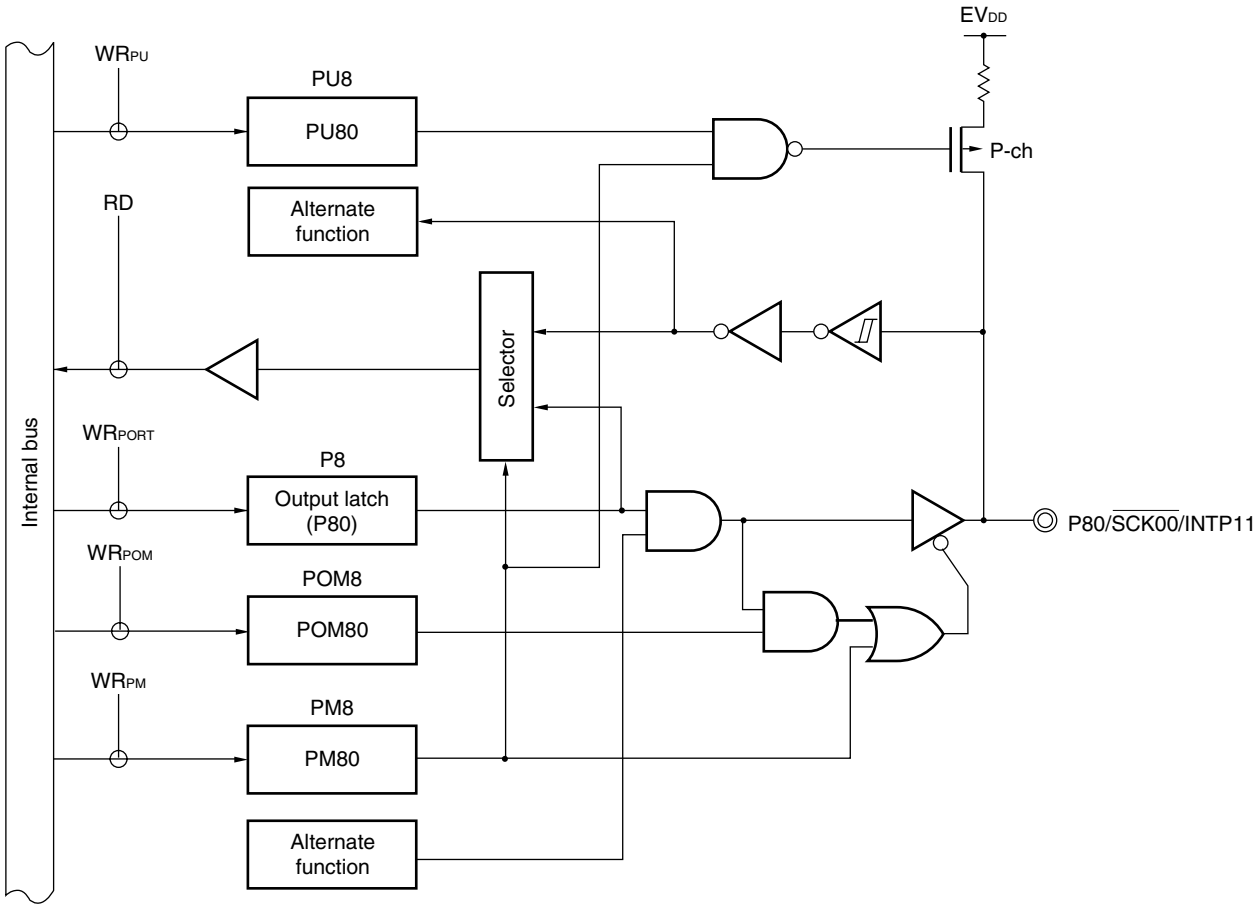
This port can also be used for serial interface clock I/O, data I/O, and external interrupt request input.

Reset signal generation sets port 8 to input mode.

Figures 4-14 to 4-16 show block diagrams of port 8.

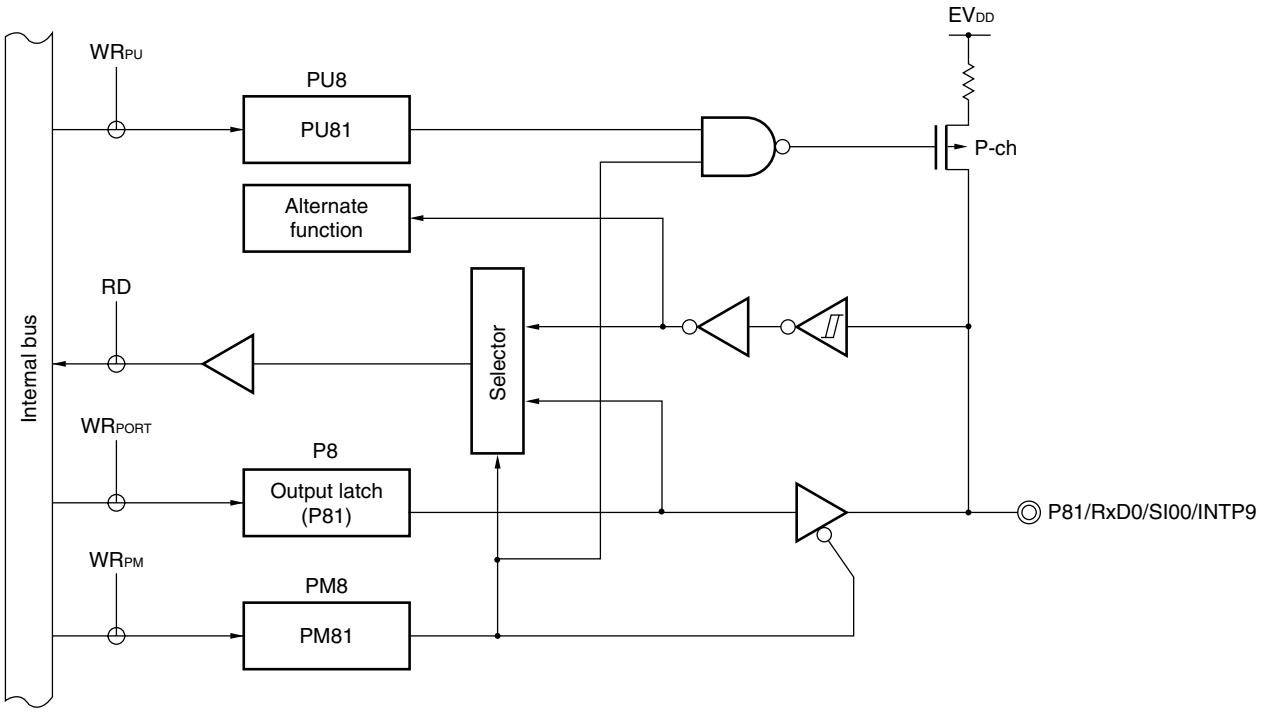
Caution To use P80/$\overline{\text{SCK00}}$/INTP11, P81/RxD0/SI00/INTP9, P82/SO00/TxD0, as a general-purpose port, note the serial array unit 0 setting. For details, refer to Table 14-5 Relationship Between Register Settings and Pins (Channel 0 of unit 0: CSI00, UART0 transmission).

Figure 4-14. Block Diagram of P80



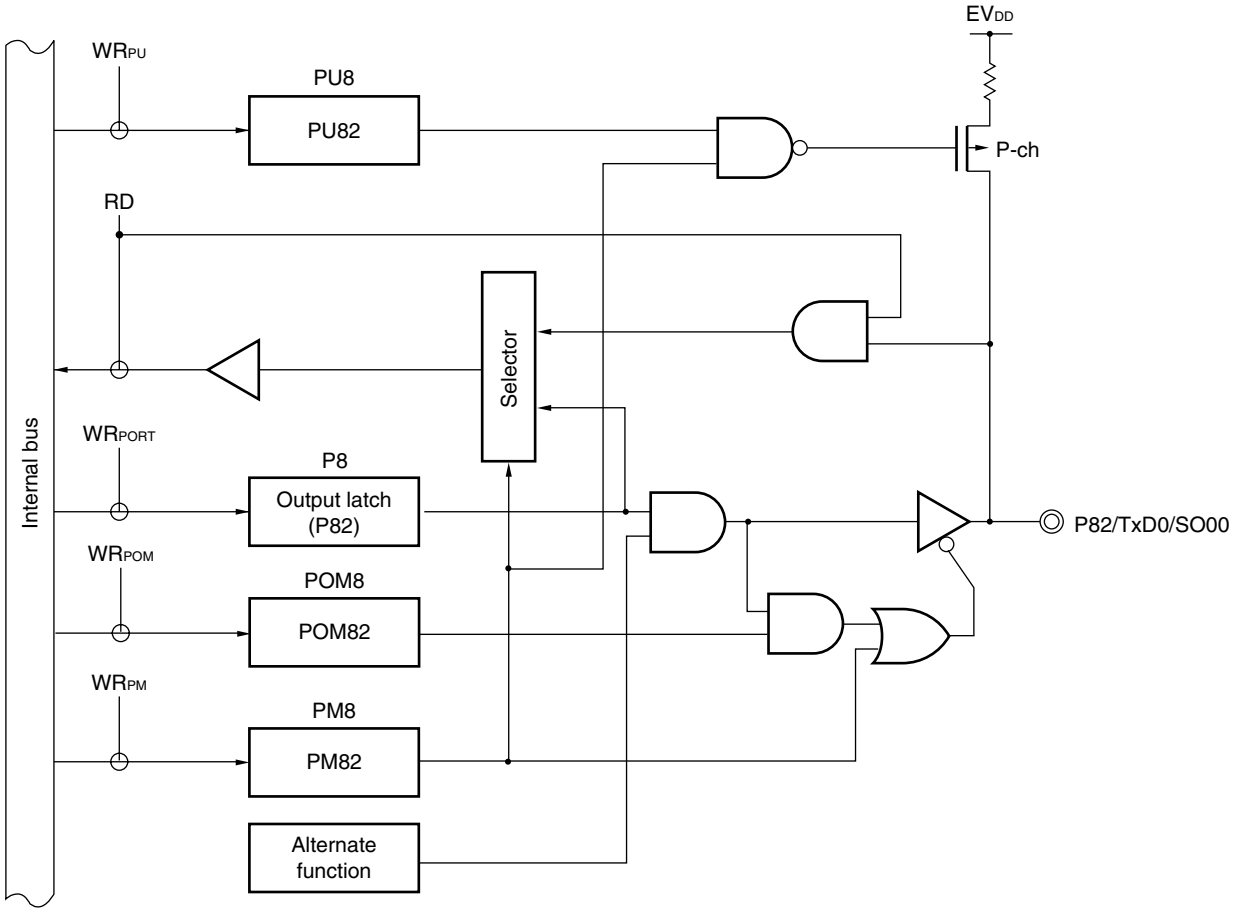
P8: Port register 8
PU8: Pull-up resistor option register 8
POM8: Port output mode register 8
PM8: Port mode register 8
RD: Read signal
WR××: Write signal

Figure 4-15. Block Diagram of P81



P8: Port register 8
PU8: Pull-up resistor option register 8
PM8: Port mode register 8
RD: Read signal
WR××: Write signal

Figure 4-16. Block Diagram of P82



P8: Port register 8
PU8: Pull-up resistor option register 8
POM8: Port output mode register 8
PM8: Port mode register 8
RD: Read signal
WR××: Write signal

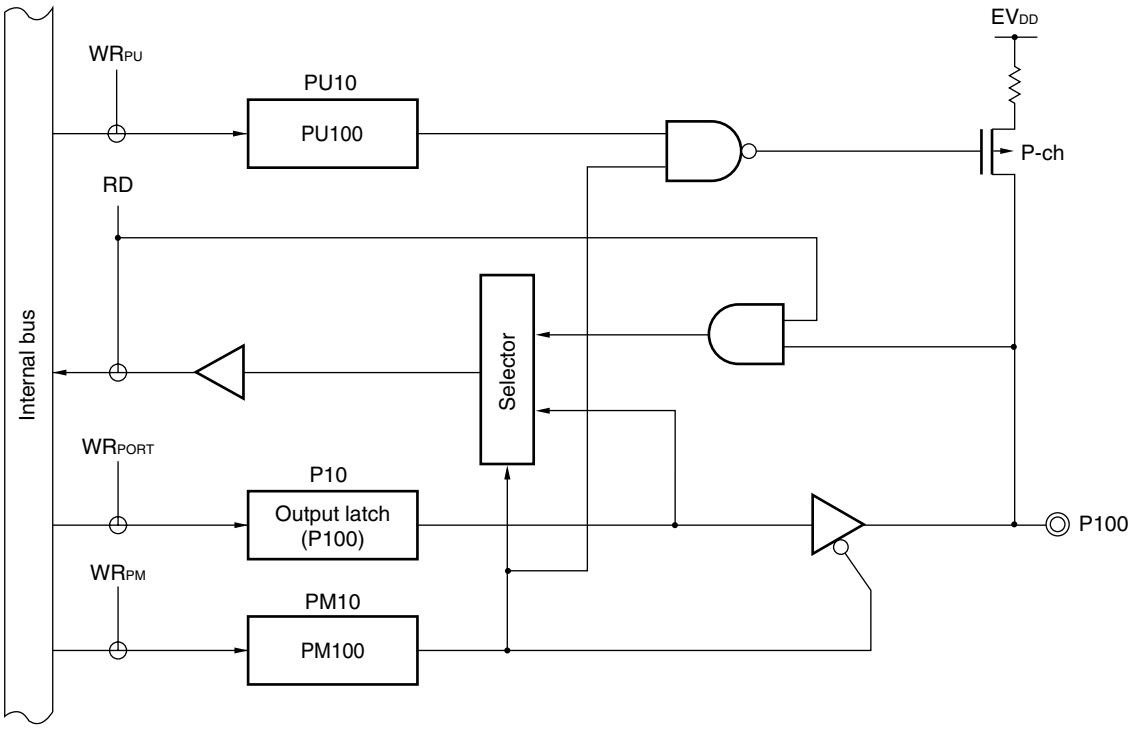
4.2.9 Port 10

Port 10 is a 1-bit I/O port with an output latch. Port 10 can be set to the input mode or output mode in 1-bit units using port mode register 10 (PM10). When the P100 pin is used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 10 (PU10).

Reset signal generation sets port 10 to input mode.

Figure 4-17 shows a block diagram of port 10.

Figure 4-17. Block Diagram of P100



P10: Port register 10
PU10: Pull-up resistor option register 10
PM10: Port mode register 10
RD: Read signal
WR××: Write signal

4.2.10 Port 11

Port 11 is a 2-bit I/O port with an output latch. Port 11 can be set to the input mode or output mode in 1-bit units using port mode register 11 (PM11).

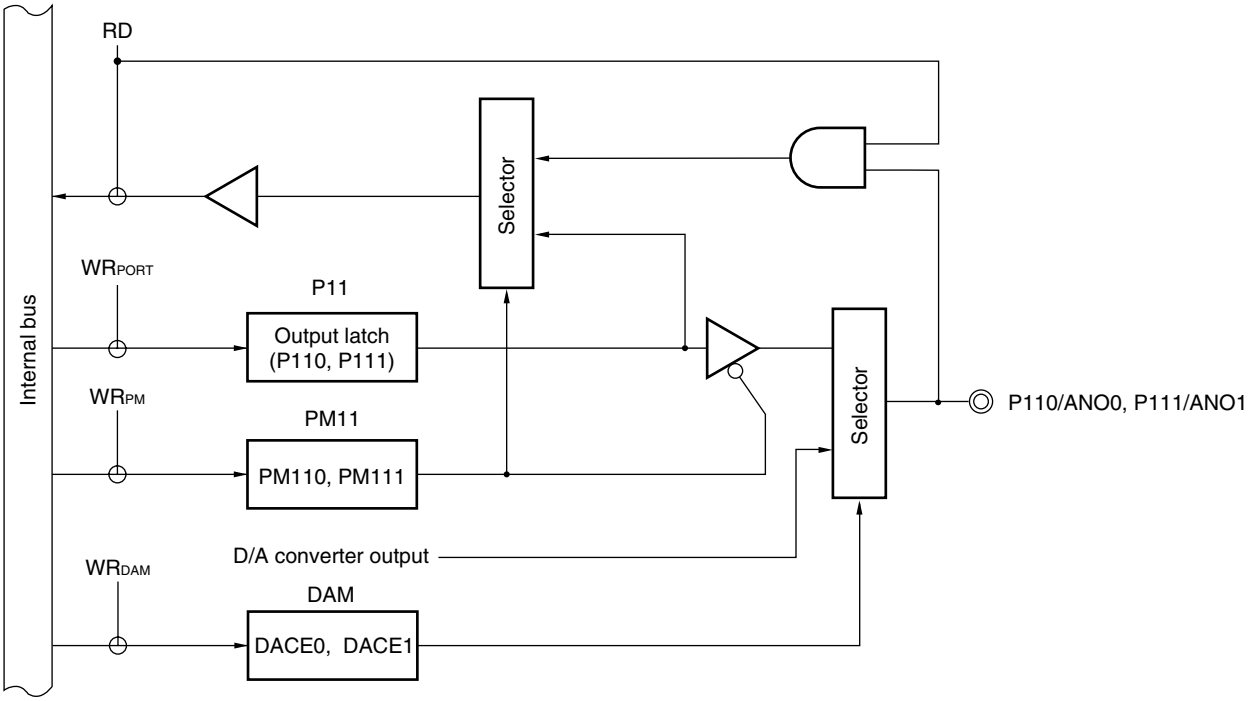
This port can also be used for D/A converter analog output.

Reset signal generation sets port 11 to input mode.

Figure 4-18 shows a block diagram of port 11.

Caution Make the AV_{DD1} pin the same potential as the EV_{DD} or V_{DD} pin when port 11 is used as a digital port.

Figure 4-18. Block Diagram of P110, P111



P11: Port register 11
PM11: Port mode register 11
DAM: D/A converter mode register
RD: Read signal
WR××: Write signal

4.2.11 Port 12

P120 is a 1-bit I/O port with an output latch. Port 12 can be set to the input mode or output mode in 1-bit units using port mode register 12 (PM12). When used as an input port, use of an on-chip pull-up resistor can be specified by pull-up resistor option register 12 (PU12).

P121 to P124 are 4-bit input ports.

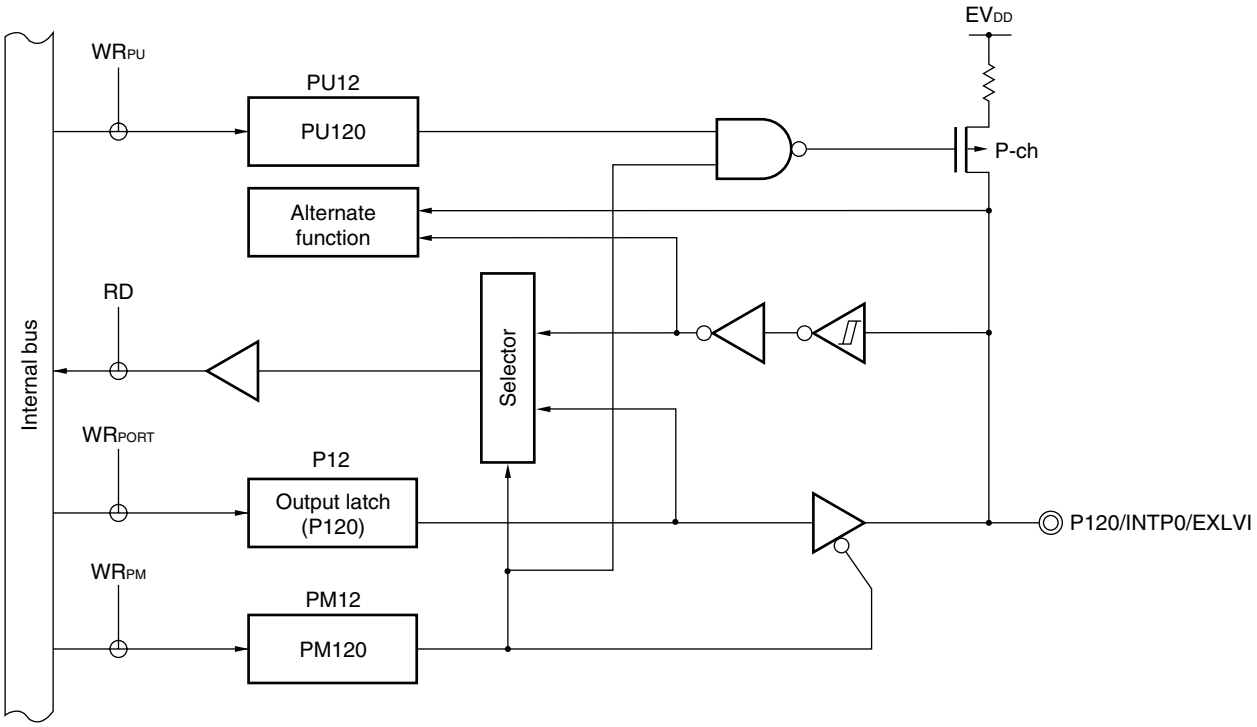
This port can also be used for external interrupt request input, potential input for external low-voltage detection, connecting resonator for main system clock, connecting resonator for subsystem clock, and external clock input for main system clock.

Reset signal generation sets port 12 to input mode.

Figures 4-19 to 4-21 show block diagrams of port 12.

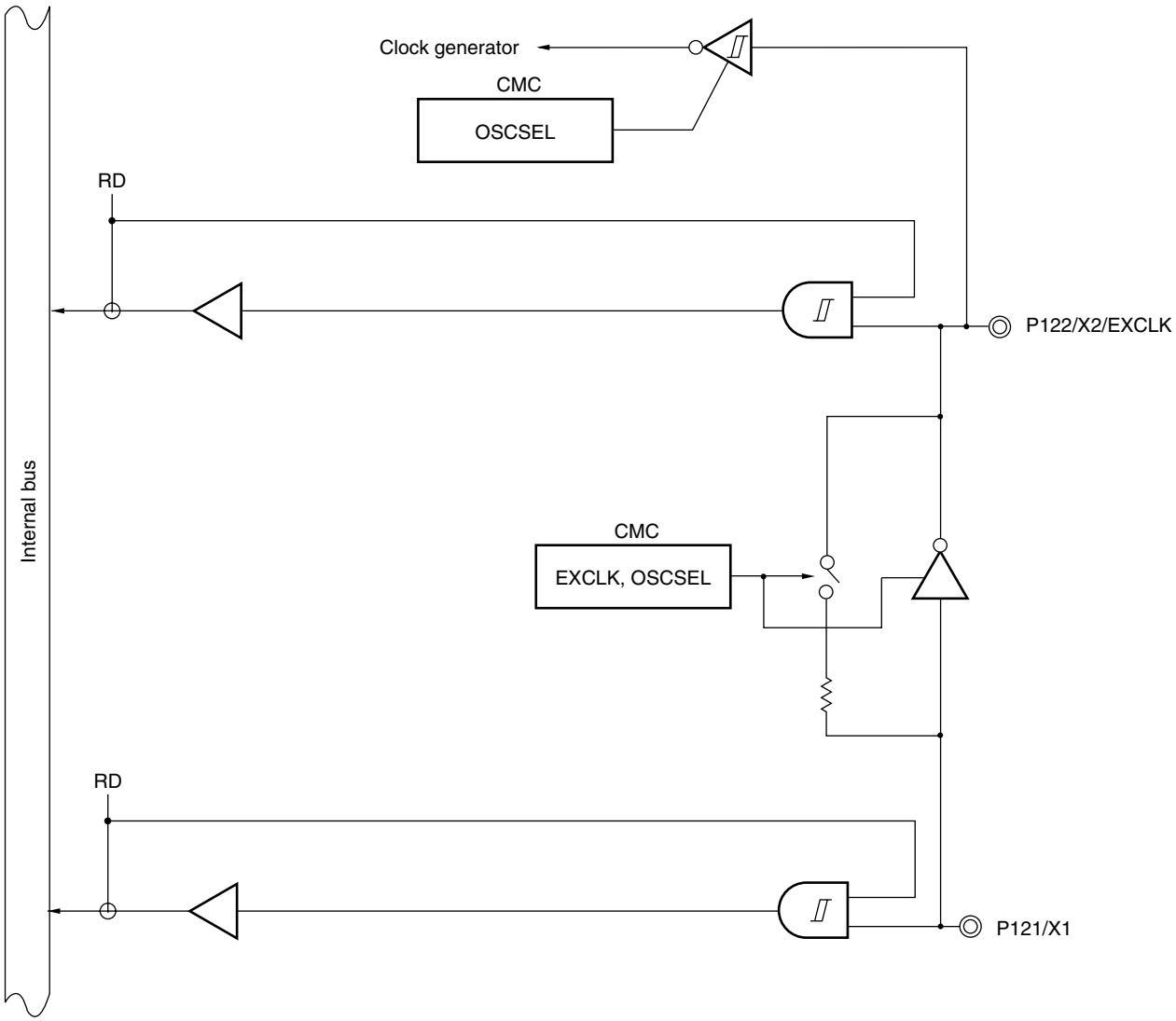
Caution The function setting on P121 to P124 is available only once after the reset release. The port once set for connection to an oscillator cannot be used as an input port unless the reset is performed.

Figure 4-19. Block Diagram of P120



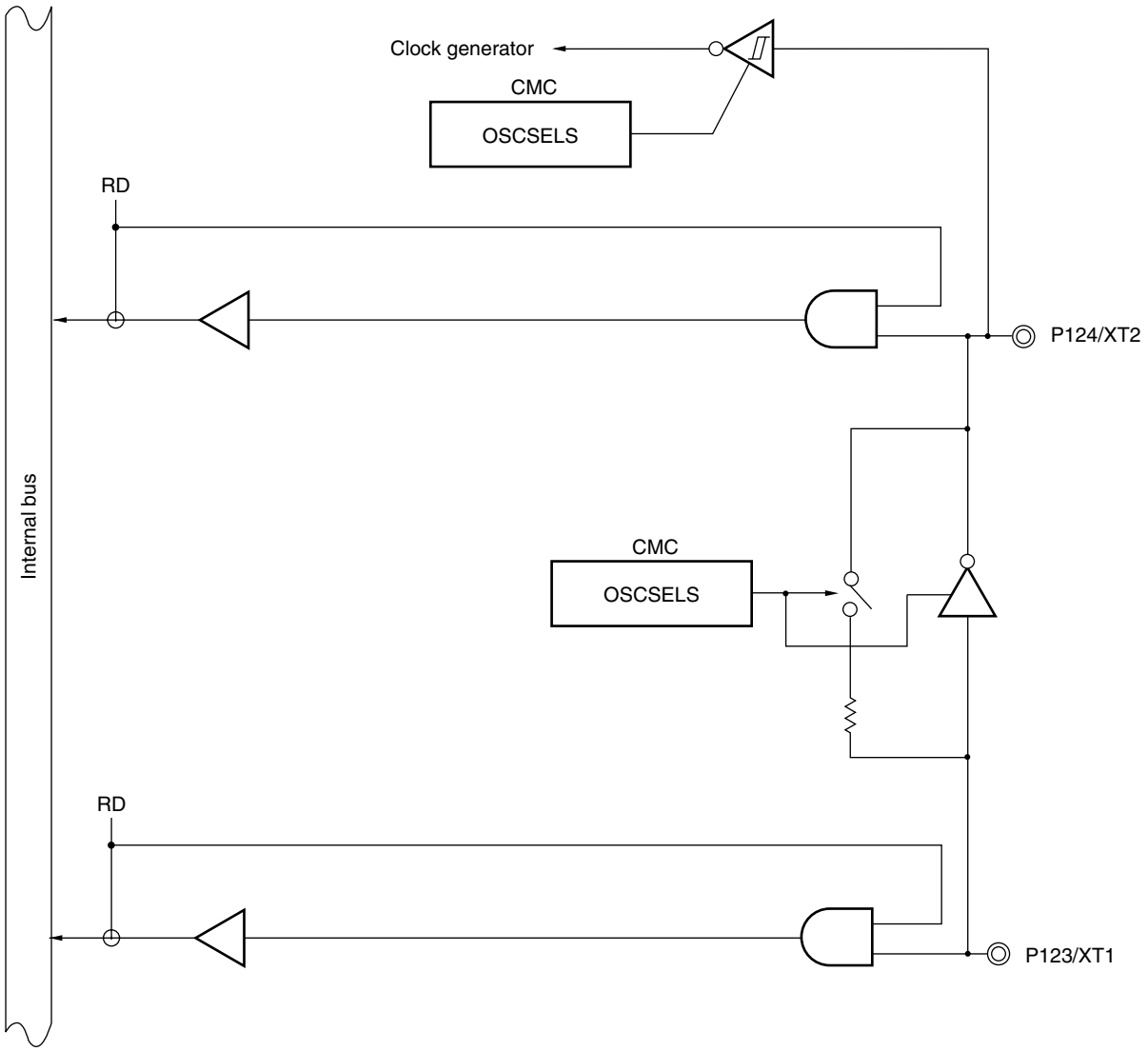
P12: Port register 12
PU12: Pull-up resistor option register 12
PM12: Port mode register 12
RD: Read signal
WR××: Write signal

Figure 4-20. Block Diagram of P121 and P122



CMC: Clock operation mode control register
RD: Read signal

Figure 4-21. Block Diagram of P123 and P124



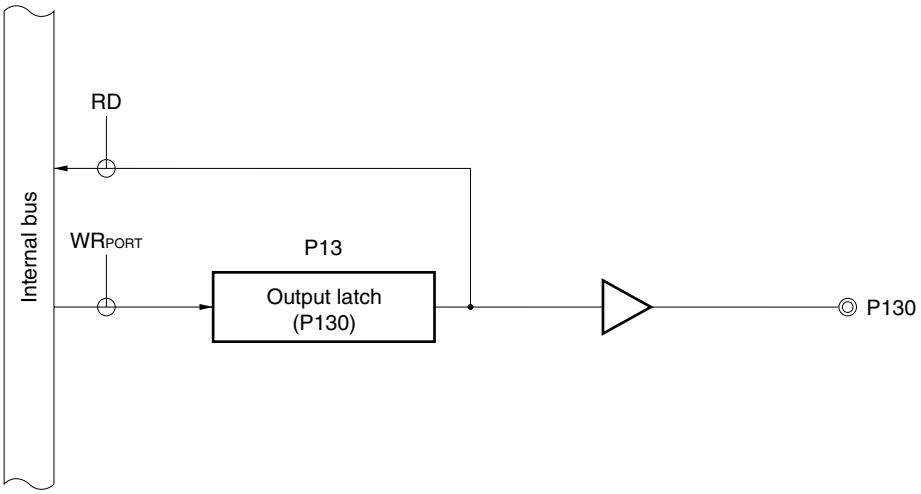
CMC: Clock operation mode control register
RD: Read signal

4.2.12 Port 13

P130 is a port dedicated to 1-bit output and is provided with an output latch.

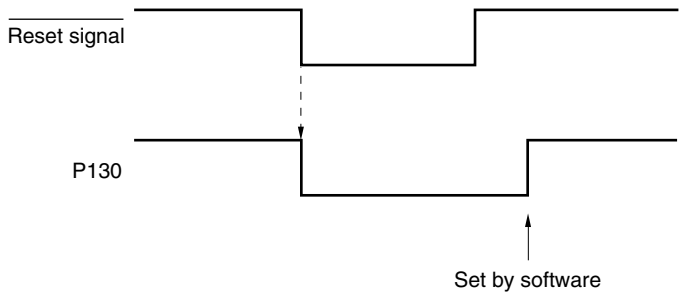
Figure 4-22 shows a block diagram of port 13.

Figure 4-22. Block Diagram of P130



P13: Port register 13
RD: Read signal
WR××: Write signal

Remark The P130 pin outputs a low level when it is used as a port function pin and a reset is effected. If P130 is set to output a high level before reset is effected, the output signal of P130 can be dummy-output as the CPU reset signal.



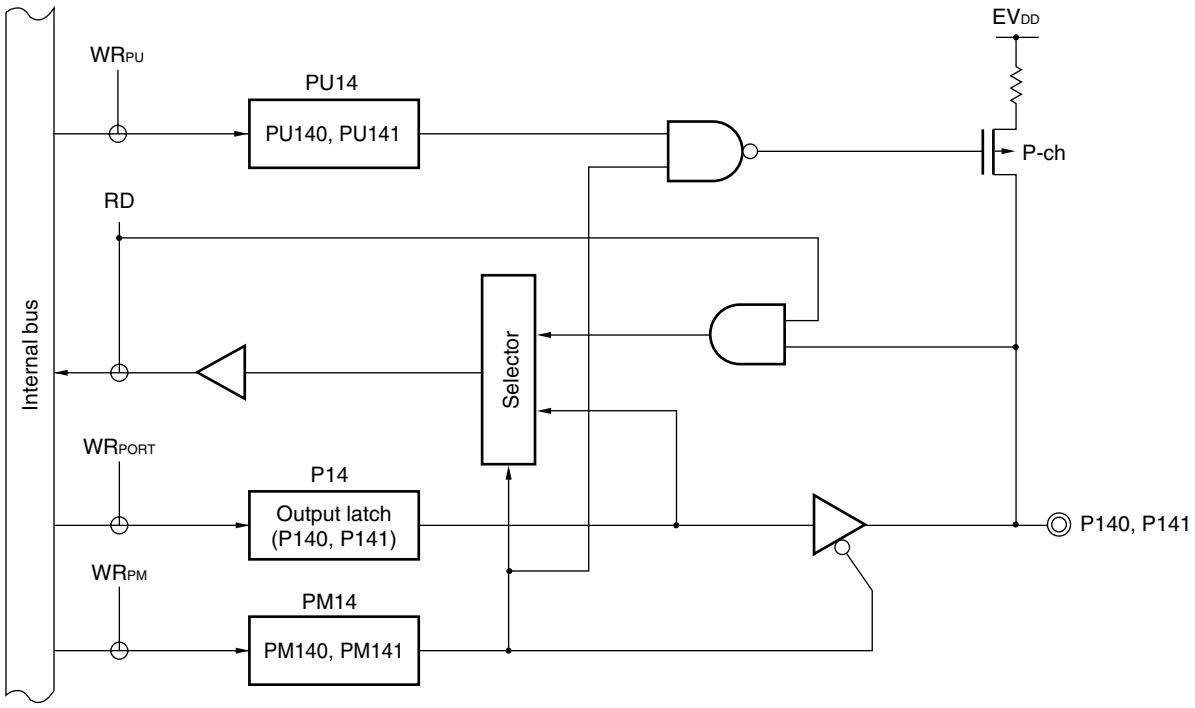
4.2.13 Port 14

Port 14 is a 2-bit I/O port with an output latch. Port 14 can be set to the input mode or output mode in 1-bit units using port mode register 14 (PM14). When the P140 and P141 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 14 (PU14).

Reset signal generation sets Port 14 to input mode.

Figure 4-23 shows a block diagram of port 14.

Figure 4-23. Block Diagram of P140, P141



P14: Port register 14
PU14: Pull-up resistor option register 14
PM14: Port mode register 14
RD: Read signal
WR××: Write signal

4.2.14 Port 15

Port 15 is a 4-bit I/O port with an output latch. Port 15 can be set to the input mode or output mode in 1-bit units using port mode register 15 (PM15).

This port can also be used for A/D converter analog input, reference voltage input, and operational amplifier input.

To use P150/ANI8/AMP2+ to P152/ANI10, and P157/ANI15/AV_{REFM} as digital input pins, set them in the digital I/O mode by using the A/D port configuration register (ADPC) and in the input mode by using PM15. Use these pins starting from the lower bit.

To use P150/ANI8/AMP2+ to P152/ANI10, and P157/ANI15/AV_{REFM} as digital output pins, set them in the digital I/O mode by using ADPC and in the output mode by using PM15.

To use P150/ANI8/AMP2+ to P152/ANI10, and P157/ANI15/AV_{REFM} as analog input pins, set them in the analog input mode by using ADPC and in the input mode by using PM15. Use these pins starting from the upper bit.

All P150/ANI8/AMP2+ to P152/ANI10, and P157/ANI15/AV_{REFM} are set in the digital input mode when the reset signal is generated.

Figures 4-24 to 4-26 show block diagrams of port 15.

Caution Make the AV_{DD0} pin the same potential as the EV_{DD} or V_{DD} pin when port 15 is used as a digital port.

Table 4-6. Setting Functions of ANI8/AMP2+/P150 Pins

ADPC register	PM15 register	OAEN2 bit	ADS register	ANI8/AMP2+/P150 Pin
Digital I/O selection	Input mode	0	–	Digital input
		1	–	Setting prohibited
	Output mode	0	–	Digital output
		1	–	Setting prohibited
Analog input selection	Input mode	0	Selects ANI.	Analog input (to be A/D converted)
			Does not select ANI.	Analog input (not to be A/D converted)
		1	Selects ANI.	Setting prohibited
			Does not select ANI.	Operational amplifier input
	Output mode	–	–	Setting prohibited

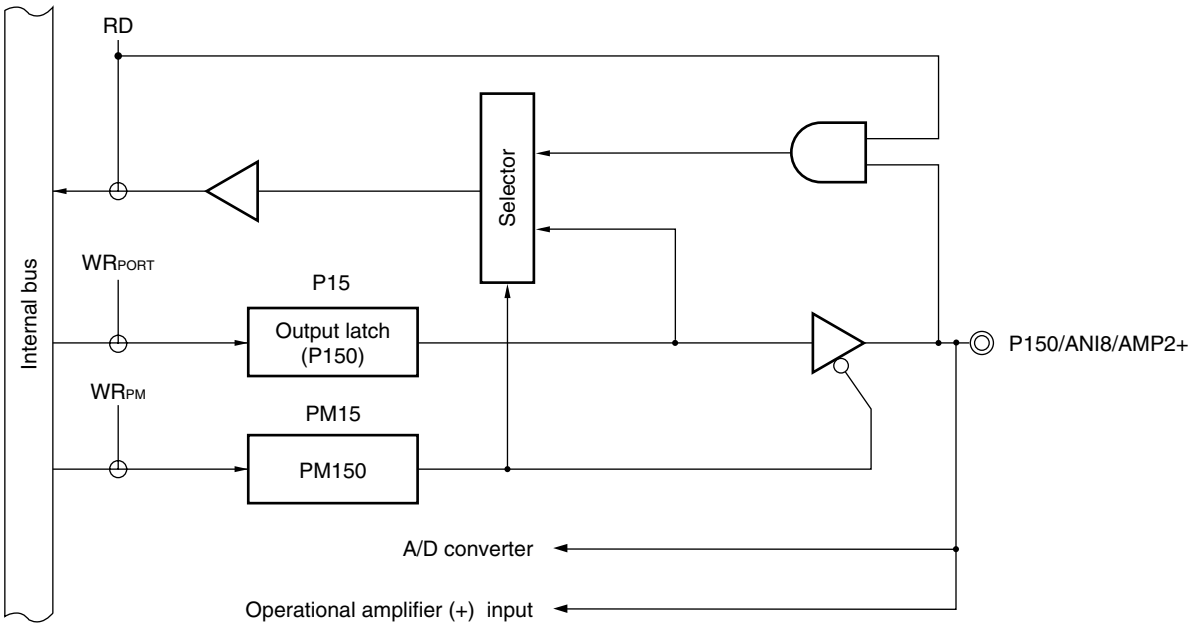
Table 4-7. Setting Functions of ANI9/P151 and ANI10/AM152 Pins

ADPC register	PM15 register	ADS register	ANI9/P151 and ANI10/AM152 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be A/D converted)
		Does not select ANI.	Analog input (not to be A/D converted)
	Output mode	–	Setting prohibited

Table 4-8. Setting Functions of ANI15/AVREFM/P157 Pin

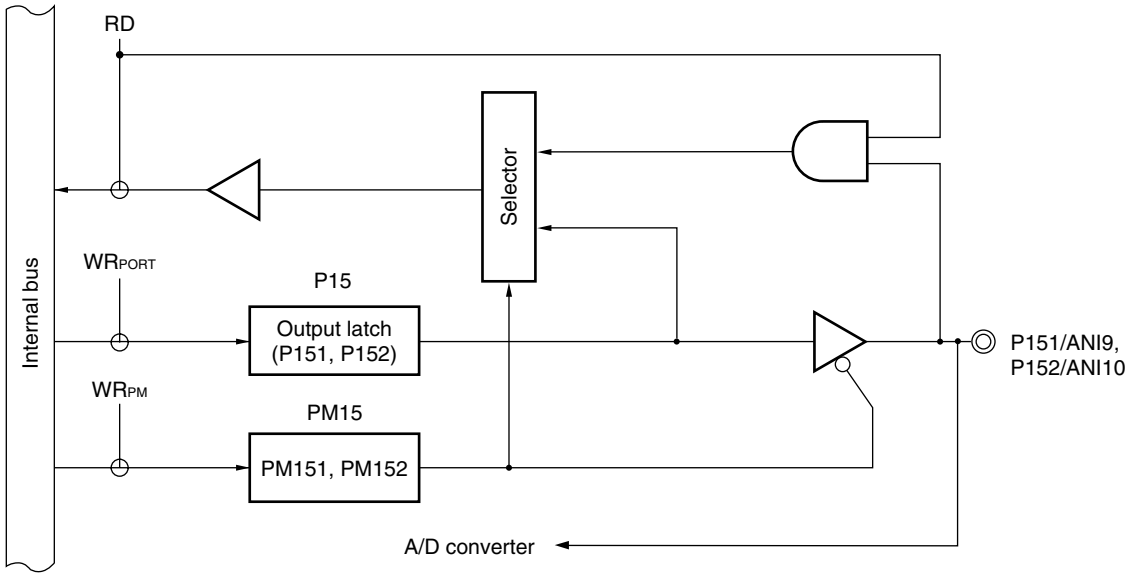
ADPC register	PM15 register	ADREF bit	ADS register	ANI15/AVREFM/P157 Pin
Digital I/O selection	Input mode	0	–	Digital input
		1	–	Setting prohibited
	Output mode	0	–	Digital output
		1	–	Setting prohibited
Analog input selection	Input mode	0	Selects ANI.	Analog input (to be converted)
			Does not select ANI.	Analog input (not to be converted)
		1	–	Negative reference voltage input of A/D converter
	Output mode	–	–	Setting prohibited

Figure 4-24. Block Diagram of P150



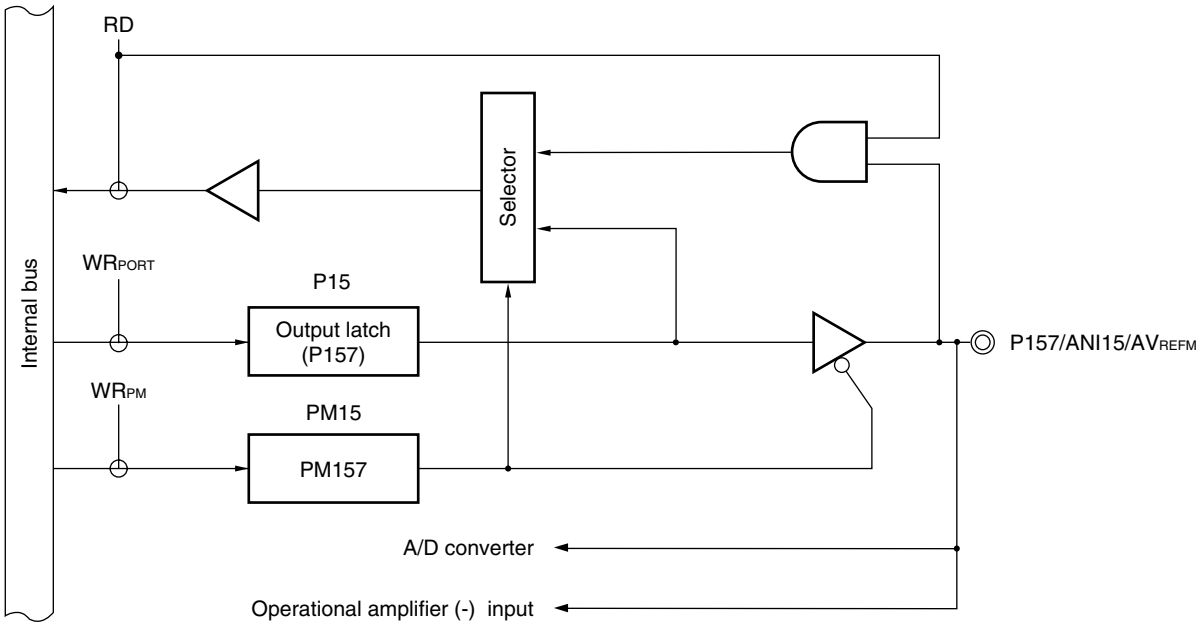
P15: Port register 15
PM15: Port mode register 15
RD: Read signal
WR××: Write signal

Figure 4-25. Block Diagram of P151, P152



P15: Port register 15
PM15: Port mode register 15
RD: Read signal
WR××: Write signal

Figure 4-26. Block Diagram of P157



P15: Port register 15
PM15: Port mode register 15
RD: Read signal
WR××: Write signal

4.3 Registers Controlling Port Function

Port functions are controlled by the following seven types of registers.

- Port mode registers (PMxx)
- Port registers (Pxx)
- Pull-up resistor option registers (PUxx)
- Port input mode register 1 (PIM1)
- Port output mode registers 1, 8 (POM1, POM8)
- A/D port configuration register (ADPC)
- Input switch control register (ISC)

(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.

Figure 4-27 Format of Port Mode Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM0	1	1	1	1	1	PM02	PM01	PM00	FFF20H	FFH	R/W
PM1	1	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20	FFF22H	FFH	R/W
PM3	1	1	1	PM34	PM33	PM32	PM31	PM30	FFF23H	FFH	R/W
PM4	1	1	1	1	1	1	PM41	PM40	FFF24H	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FFF25H	FFH	R/W
PM6	1	1	1	1	1	1	PM61	PM60	FFF26H	FFH	R/W
PM8	1	1	1	1	1	PM82	PM81	PM80	FFF28H	FFH	R/W
PM10	1	1	1	1	1	1	1	PM100	FFF2AH	FFH	R/W
PM11	1	1	1	1	1	1	PM111	PM110	FFF2BH	FFH	R/W
PM12	1	1	1	1	1	1	1	PM120	FFF2CH	FFH	R/W
PM14	PM147 Note	PM146 Note	PM145 Note	PM144 Note	PM143 Note	PM142 Note	PM141	PM140	FFF2EH	FEH	R/W
PM15	PM157	1	1	1	1	PM152	PM151	PM150	FFF2FH	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 0 to 6, 8, 10 to 12, 14, 15; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

Note Be sure to set the PM142 to PM147 bits to 0 after reset release.

Caution Be sure to set bits 3 to 7 of PM0, bit 7 of PM1, bits 5 to 7 of PM3, bits 2 to 7 of PM4, bits 2 to 7 of PM6, bits 3 to 7 of PM8, bits 1 to 7 of PM10, bits 2 to 7 of PM11, bits 1 to 7 of PM12, and bits 3 to 6 of PM15 to 1.

(2) Port registers (Pxx)

These registers write the data that is output from the chip when data is output from 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 It is always 0 and never a pin level that is read out if a port is read during the input mode when P2 and P15 are set to function as an analog input for a A/D converter , and P11 is set to function as an analog input for a D/A converter.

Figure 4-28. Format of Port Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
P0	0	0	0	0	0	P02	P01	P00	FFF00H	00H (output latch)	R/W
P1	0	P16	P15	P14	P13	P12	P11	P10	FFF01H	00H (output latch)	R/W
P2	P27	P26	P25	P24	P23	P22	P21	P20	FFF02H	00H (output latch)	R/W
P3	0	0	0	P34	P33	P32	P31	P30	FFF03H	00H (output latch)	R/W
P4	0	0	0	0	0	0	P41	P40	FFF04H	00H (output latch)	R/W
P5	P57	P56	P55	P54	P53	P52	P51	P50	FFF05H	00H (output latch)	R/W
P6	0	0	0	0	0	0	P61	P60	FFF06H	00H (output latch)	R/W
P8	0	0	0	0	0	P82	P81	P80	FFF08H	00H (output latch)	R/W
P10	0	0	0	0	0	0	0	P100	FFF0AH	00H (output latch)	R/W
P11	0	0	0	0	0	0	P111	P110	FFF0BH	00H (output latch)	R/W
P12	0	0	0	P124	P123	P122	P121	P120	FFF0CH	Undefined	R/W[Note]
P13	0	0	0	0	0	0	0	P130	FFF0DH	00H (output latch)	R/W
P14	0	0	0	0	0	0	P141	P140	FFF0EH	00H (output latch)	R/W
P15	P157	0	0	0	0	P152	P151	P150	FFF0FH	00H (output latch)	R/W

Pmn	m = 0 to 6, 8, 10 to 15 ; 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 to P124 are read-only.

(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 of 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.

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

Reset signal generation clears these registers to 00H.

Figure 4-29. Format of Pull-up Resistor Option Register

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PU0	0	0	0	0	0	PU02	PU01	PU00	F0030H	00H	R/W
PU1	0	PU16	PU15	PU14	PU13	PU12	PU11	PU10	F0031H	00H	R/W
PU3	0	0	0	PU34	PU33	PU32	PU31	PU30	F0033H	00H	R/W
PU4	0	0	0	0	0	0	PU41	PU40	F0034H	00H	R/W
PU5	PU57	PU56	PU55	PU54	PU53	PU52	PU51	PU50	F0035H	00H	R/W
PU8	0	0	0	0	0	PU82	PU81	PU80	F0038H	00H	R/W
PU10	0	0	0	0	0	0	0	PU100	F003AH	00H	R/W
PU12	0	0	0	0	0	0	0	PU120	F003CH	00H	R/W
PU14	0	0	0	0	0	0	PU141	PU140	F003EH	00H	R/W

PUmn	Pmn pin on-chip pull-up resistor selection (m = 0, 1, 3 to 5, 8, 10, 12, 14 ; n = 0 to 7)
0	On-chip pull-up resistor not connected
1	On-chip pull-up resistor connected

(4) Port input mode register 1 (PIM1)

PIM1 register sets the input buffer of P10, P11, P14, or P15 in 1-bit units.

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

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

Reset signal generation clears this register to 00H.

Figure 4-30. Format of Port Input Mode Register 1 (PIM1)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM1	0	0	PIM15	PIM14	0	0	PIM11	PIM10	F0041H	00H	R/W

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

(5) Port output mode registers 1, 8 (POM1, POM8)

These registers set the output mode of P10 to P15, P80, and P82 in 1-bit units.

N-ch open drain output (V_{DD} tolerance) mode can be selected during serial communication with an external device of the different potential, and for the SDA10 and SDA20 pins during simplified I^2C communication with an external device of the same potential.

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

Reset signal generation clears these registers to 00H.

Figure 4-31. Format of Port Output Mode Registers 1, 8 (POM1, POM8)

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
POM8	0	0	0	0	0	POM82	0	POM80	F0058H	00H	R/W

POMmn	Pmn pin output mode selection (m = 1 and 8; n = 0 to 5)
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance) mode

(6) A/D port configuration register (ADPC)

This register switches the ANI0/AMP0-/P20 to ANI7/AMP2O/P27, ANI8/AMP2+/P150 to ANI10/P152, and ANI15/AV_{REFM}/P157 pins to analog input or digital I/O of port.

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

Reset signal generation sets this register to 10H.

Figure 4-32. Format of A/D Port Configuration Register (ADPC)

Address: F0017H After reset: 10H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC	0	0	0	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0

ADPC4	ADPC3	ADPC2	ADPC1	ADPC0	Analog input (A)/digital I/O (D) switching											
					Port 15				Port 2							
					ANI15 /AV_{REFM} /P157	ANI10 /P152	ANI9 /P151	ANI8 /AMP2+ /P150	ANI7 /AMP2O /P27	ANI6 /AMP2- /P26	ANI5 /AMP1+ /P25	ANI4 /AMP1O /P24	ANI3 /AMP1- /P23	ANI2 /AMP0+ /P22	ANI1 /AMP0O /P21	ANI0 /AMP0- /P20
0	0	0	0	0	A	A	A	A	A	A	A	A	A	A	A	A
0	0	0	0	1	A	A	A	A	A	A	A	A	A	A	A	D
0	0	0	1	0	A	A	A	A	A	A	A	A	A	A	D	D
0	0	0	1	1	A	A	A	A	A	A	A	A	A	D	D	D
0	0	1	0	0	A	A	A	A	A	A	A	A	D	D	D	D
0	0	1	0	1	A	A	A	A	A	A	A	D	D	D	D	D
0	0	1	1	0	A	A	A	A	A	A	D	D	D	D	D	D
0	0	1	1	1	A	A	A	A	A	D	D	D	D	D	D	D
0	1	0	0	0	A	A	A	A	D	D	D	D	D	D	D	D
0	1	0	0	1	A	A	A	D	D	D	D	D	D	D	D	D
0	1	0	1	0	A	A	D	D	D	D	D	D	D	D	D	D
0	1	1	1	1	A	D	D	D	D	D	D	D	D	D	D	D
1	0	0	0	0	D	D	D	D	D	D	D	D	D	D	D	D
Other than the above					Setting prohibited											

Cautions 1. Set a channel to be used for A/D conversion in the input mode by using port mode registers 2 and 15 (PM2, PM15).

2. Do not set the pin that is set by ADPC as digital I/O by analog input channel specification register (ADS).

(7) Input switch control register (ISC)

Bits 0 and 1 of ISC are used for linking with an external interrupt or a timer array unit when performing a LIN-bus communication operation with UART3.

When bit 0 is set to 1, the input signal of the serial data input (RxD3) 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 (RxD3) pin is selected as a timer input, so that the pulse widths of a sync break field and a sync field can be measured by the timer.

Bits 2 to 4 of ISC are used to prevent through current from entering when using the TI04/P53, TI02/P52, and RxD3/P50 pins as port outputs.

The port output pins to be used alternatively with the TI04, TI02, and RxD3 pins are internally connected with a Schmitt trigger buffer. When using these pins as port outputs, bits 2 to 4 of ISC must be set to 0 (prohibiting input to Schmitt trigger buffers) in order to prevent through current from entering.

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

Reset signal generation clears this register to 00H.

Figure 4-33. Format of Input Switch Control Register (ISC)

Address: FFF3CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	ISC4	ISC3	ISC2	ISC1	ISC0

ISC4	TI04/P53 schmitt trigger buffer control
0	Disables input
1	Enables input

ISC3	TI02/P52 schmitt trigger buffer control
0	Disables input
1	Enables input

ISC2	RxD3/P50 schmitt trigger buffer control
0	Disables input
1	Enables input

ISC1	Switching channel 7 input of timer array unit 0
0	Uses the input signal of the TI07 pin as a timer input (normal operation).
1	Input signal of RxD3 pin is used as timer input (wakeup signal detection).

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 RxD3 pin as an external interrupt (to measure the pulse widths of the sync break field and sync field).

Caution Be sure to clear bits 5 to 7 to "0".

To use the TI04/P53, TI02/P52, and RxD3/P50 pins, set the ISCn (n = 2 to 4) bits as follows, according to the function to be used.

ISCn	Pin function
0	Port output (default)
1	Port input, timer input, or serial data input

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.

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

4.4.4 Connecting to external device with different power potential (2.5 V, 3 V)

When parts of ports 1 and 8 operate with V_{DD} = 4.0 V to 5.5 V, I/O connections with an external device that operates on a 2.5 V or 3 V power supply voltage are possible.

Regarding inputs, CMOS/TTL switching is possible on a bit-by-bit basis by port input mode register 1 (PIM1).

Moreover, regarding outputs, different power potentials can be supported by switching the output buffer to the N-ch open drain (V_{DD} withstand voltage) by the port output mode registers 1, 8 (POM1 and POM8).

(1) Setting procedure when using I/O pins of UART1, UART2, CSI00, CSI10, and CSI20 functions

(a) Use as 2.5 V or 3 V input port

<1> After reset release, the port mode is the input mode (Hi-Z).

<2> If pull-up is needed, externally pull up the pin to be used (on-chip pull-up resistor cannot be used).

In case of UART1: P14
In case of UART2: P11
In case of CSI10: P15, P14
In case of CSI20: P10, P11

<3> Set the corresponding bit of the PIM1 register to 1 to switch to the TTL input buffer.

<4> V_{IH}/V_{IL} operates on a 2.5 V or 3 V operating voltage.

(b) Use as 2.5 V or 3 V output port

<1> After reset release, the port mode changes to the input mode (Hi-Z).

<2> Pull up externally the pin to be used (on-chip pull-up resistor cannot be used).

In case of UART1: P13
In case of UART2: P12
In case of CSI00: P80, P82
In case of CSI10: P15, P13
In case of CSI20: P10, P12

<3> Set the output latch of the corresponding port to 1.

<4> Set the corresponding bit of the POMn register to 1 to set the N-ch open drain output (V_{DD} withstand voltage) mode.

<5> Set the output mode by manipulating the PMn register.
At this time, the output data is high level, so the pin is in the Hi-Z state.

<6> Operation is done only in the low level according to the operating status of the serial array unit.

Remark n = 1, 8

(2) Setting procedure when using I/O pins of simplified IIC10, IIC20 functions

<1> After reset release, the port mode is the input mode (Hi-Z).

<2> Externally pull up the pin to be used (on-chip pull-up resistor cannot be used).

In case of simplified IIC10: P14, P15
In case of simplified IIC20: P11, P10

<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} withstand voltage) mode.

<5> 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.

<6> Enable the operation of the serial array unit and set the mode to the simplified I^2C mode.

4.5 Settings of Port Mode Register and Output Latch When Using Alternate Function

To use the alternate function of a port pin, set the port mode register and output latch as shown in Table 4-9.

Table 4-9. Settings of Port Mode Register and Output Latch When Using Alternate Function (1/4)

Pin Name	Alternate Function		ISC	PM××	P××
	Function Name	I/O	(ISCx)		
P10	$\overline{SCK20}$	Input	–	1	×
		Output	–	0	1
	SCL20	I/O	–	0	1
P11	SI20	Input	–	1	×
	RxD2	Input	–	1	×
	SDA20	I/O	–	0	1
	INTP6	Input	–	1	×
P12	SO20	Output	–	0	1
	TxD2	Output	–	0	1
	TO02	Output	–	0	0
P13	SO10	Output	–	0	1
	TxD1	Output	–	0	1
	TO04	Output	–	0	0
P14	SI10	Input	–	1	×
	RxD1	Input	–	1	×
	SDA10	I/O	–	0	1
	INTP4	Input	–	1	×
P15	$\overline{SCK10}$	Input	–	1	×
		Output	–	0	1
	SCL10	I/O	–	0	1
	INTP7	Input	–	1	×
P16	TI05	Input	–	1	×
	TO05	Output	–	0	0
	INTP10	Input	–	1	×

Remark ×: don't care
–: Not applicable
ISC: Input switch control register
PM××: Port mode register
P××: Port output latch

Table 4-9. Settings of Port Mode Register and Output Latch When Using Alternate Function (2/4)

Pin Name	Alternate Function		ISC	PM××	P××
	Function Name	I/O	(ISCx)		
P20 to P25[Note]	ANI0 to ANI5	Input	–	1	×
	AMP0-, AMP0+, AMP1-, AMP1+	Input	–	1	×
	AMP0O, AMP1O	Output	–	1	×
P26[Note]	ANI6	Input	–	1	×
	AMP2-	Input	–	1	×
P27[Note]	ANI7	Input	–	1	×
	AMP2O	Output	–	1	×
P30	TI03	Input	–	1	×
	TO00	Output	–	0	0
	RTC1HZ	Output	–	0	0
	INTP1	Input	–	1	×
P31	TI00	Input	–	1	×
	TO03	Output	–	0	0
	RTCDIV	Output	–	0	0
	RTCCL	Output	–	0	0
	PCLBUZ1	Output	–	0	0
	INTP2	Input	–	1	×
P32	TI01	Input	–	1	×
	TO01	Output	–	0	0
	INTP5	Input	–	1	×
	PCLBUZ0	Output	–	0	0
P33	TI07	Input	ISC1 = 0	1	×
	TO07	Output	–	0	0
	INTP3	Input	–	1	×
P34	TI06	Input	–	1	×
	TO06	Output	–	0	0
	INTP8	Input	–	1	×

Note The function of the P20/ANI0/AMP0-, P21/ANI1/AMP0O, P22/ANI2/AMP0+, P23/ANI3/AMP1-, P24/ANI4/AMP1O, P25/ANI5/AMP1+, P26/ANI6 pins can be selected by using the A/D port configuration register (ADPC), port mode register 2 (PM2), analog input channel specification register (ADS), and operational amplifier control register (OAC). Refer to **4.2.3 Port 2**.

Remark ×: don't care
–: Not applicable
ISC: Input switch control register
PM××: Port mode register
P××: Port output latch

Table 4-9. Settings of Port Mode Register and Output Latch When Using Alternate Function (3/4)

Pin Name	Alternate Function		ISC	PM××	P××
	Function Name	I/O	(ISCx)		
P40	TOOL0	I/O	–	×	×
P41	TOOL1	Output	–	×	×
P50	RxD3	Input	ISC2 = 1[Note]	1	×
P51	TxD3	Output	–	0	1
P52	TI02	Input	ISC3 = 1	1	×
P53	TI04	Input	ISC4 = 1	1	×
P60	SCL0	I/O	–	0	0
P61	SDA0	I/O	–	0	0
P80	$\overline{\text{SCK00}}$	Input	–	1	×
		Output	–	0	1
	INTP11	Input	–	1	×
P81	RxD0	Input	–	1	×
	SI00	Input	–	1	×
	INTP9	Input	–	1	×
P82	TxD0	Output	–	0	1
	SO00	Output	–	0	1

Note The RxD3 input can be set as the input source of an external interrupt input (INTP0) by setting ISC0 = 1.
The RxD3 input can be set as the input source of a timer input (TI07) by setting ISC1 = 1.

Remark ×: don't care
–: Not applicable
ISC: Input switch control register
PM××: Port mode register
P××: Port output latch

Table 4-9. Settings of Port Mode Register and Output Latch When Using Alternate Function (4/4)

Pin Name	Alternate Function		ISC	PM××	P××
	Function Name	I/O	(ISCx)		
P110, P111	ANO0, ANO1	Output	–	0	×
P120	INTP0[Note 1]	Input	ISC0 = 0	1	×
	EXLVI[Note 1]	Input	–	1	×
P121	X1[Note 1]	–	–	×	×
P122	X2[Note 1]	–	–	×	×
	EXCLK[Note 1]	Input	–	×	×
P123	XT1[Note 1]	–	–	×	×
P124	XT2[Note 1]	–	–	×	×
P150[Note 2]	ANI8	Input	–	1	×
	AMP2+	Input	–	1	×
P151, P152[Note 2]	ANI9, ANI10	Input	–	1	×
P157[Note 2]	ANI15	Input	–	1	×
	AV_{REFM}	Input	–	1	×

Notes 1. To use the P121 to P124 pins for main system clock resonator connection (X1, X2), subsystem clock resonator connection (XT1, XT2), or main system clock external clock input (EXCLK), the X1 oscillation mode, XT1 oscillation mode, or external clock input mode must be set, respectively, by using the clock operation mode control register (CMC). CMC can be written only once after reset release (for details, refer to **5.3 (1) Clock operation mode control register (CMC)**). The reset value of CMC is 00H (both P121 to P124 are input port pins).

2. The P150/ANI8/AMP2+, P151/ANI9, P152/ANI10, P157/ANI15/AV_{REFM} pins are as shown below depending on the settings of the A/D port configuration register (ADPC), port mode register 15 (PM15), analog input channel specification register (ADS), operational amplifier control register (OAC), and analog reference voltage control register (ADVRC). Refer to **4.2.14 Port 15**.

Remark ×: don't care
–: Not applicable
ISC: Input switch control register
PM××: Port mode register
P××: Port output latch

4.6 Cautions on 1-bit Manipulation Instruction for Port Register n (Pn)

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

Therefore, it is recommended to rewrite the output latch when switching a port from input mode to output mode.

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

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

A 1-bit manipulation instruction is executed in the following order in the 78K0R/Kx3-A Microcontrollers.

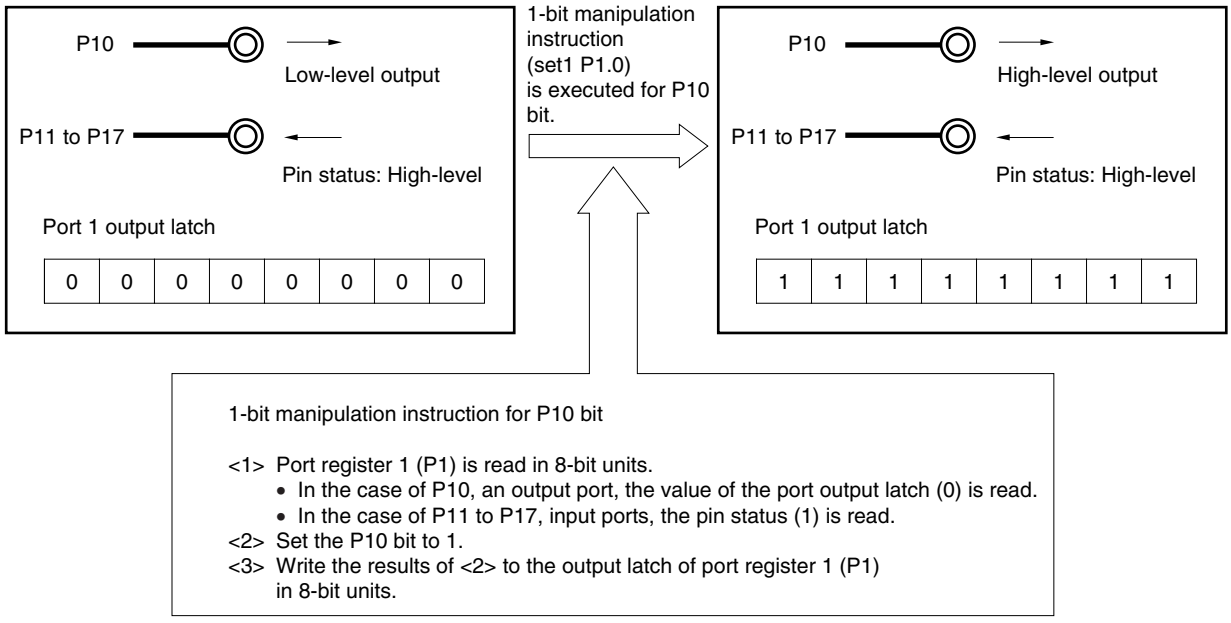
<1> The Pn register is read in 8-bit units.
<2> The targeted one bit is manipulated.
<3> The Pn register is written in 8-bit units.

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 4-34. Bit Manipulation Instruction (P10)



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 = 2 to 20 MHz by connecting a resonator to X1 and X2.
Oscillation can be stopped by executing the STOP instruction or setting of MSTOP (bit 7 of the clock operation status control register (CSC)).

<2> Internal high-speed oscillator[Note]

This circuit oscillates clocks of f_{IH} = 1 MHz (TYP.) or f_{IH} = 8 MHz (TYP.). After a reset release, the CPU always starts operating with this internal high-speed oscillation clock. Oscillation can be stopped by executing the STOP instruction or setting HIOSTOP (bit 0 of CSC).

<3> 20 MHz internal high-speed oscillation clock oscillator[Note]

This circuit oscillates a clock of f_{IH20} = 20 MHz (TYP.). Oscillation can be started by setting bit 0 (DSCON) of the 20 MHz internal high-speed oscillation control register (DSCCTL) to 1 with $V_{DD} \geq 2.7$ V. Oscillation can be stopped by setting DSCON to 0.

Note To use the internal high-speed oscillation clock, use the option byte to set the frequency (1 MHz, 8 MHz, or 20 MHz) in advance (for details, see **CHAPTER 24 OPTION BYTE**). Also, the internal high-speed oscillator automatically starts oscillating after reset release. To use the 20 MHz internal high-speed oscillator to operate the microcontroller, oscillation is started by setting bit 0 (DSCON) of the 20 MHz internal high-speed oscillation control register (DSCCTL) to 1.

An external main system clock (f_{EX} = 2 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 MSTOP.
As the main system clock, a high-speed system clock (X1 clock or external main system clock) or internal high-speed oscillation clock can be selected by setting of MCM0 (bit 4 of the system clock control register (CKC)).

Remark f_X: X1 clock oscillation frequency
f_{IH}: Internal high-speed oscillation clock frequency
f_{IH20}: 20 MHz internal high-speed oscillation clock frequency
f_{EX}: External main system clock frequency

(2) Subsystem clock

- **XT1 clock oscillator**

  This circuit oscillates a clock of f_{SUB} = 32.768 kHz by connecting a 32.768 kHz resonator to XT1 and XT2. Oscillation can be stopped by setting XTSTOP (bit 6 of CSC).

(3) Internal low-speed oscillation clock (clock for watchdog timer)

- **Internal low-speed oscillator**

  This circuit oscillates a clock of f_{IL} = 30 kHz (TYP.).

  The internal low-speed oscillation clock cannot be used as the CPU clock. The only hardware that operates with the internal low-speed oscillation clock is the watchdog timer.

  Oscillation is stopped when the watchdog timer stops.

Remarks 1. f_{SUB}: Subsystem clock frequency

f_{IL}: Internal low-speed oscillation clock frequency

2. The watchdog timer stops in the following cases.

- When bit 4 (WDTON) of an option byte (000C0H) = 0
- If the HALT or STOP instruction is executed when bit 4 (WDTON) of an option byte (000C0H) = 1 and bit 0 (WDSTBYON) = 0

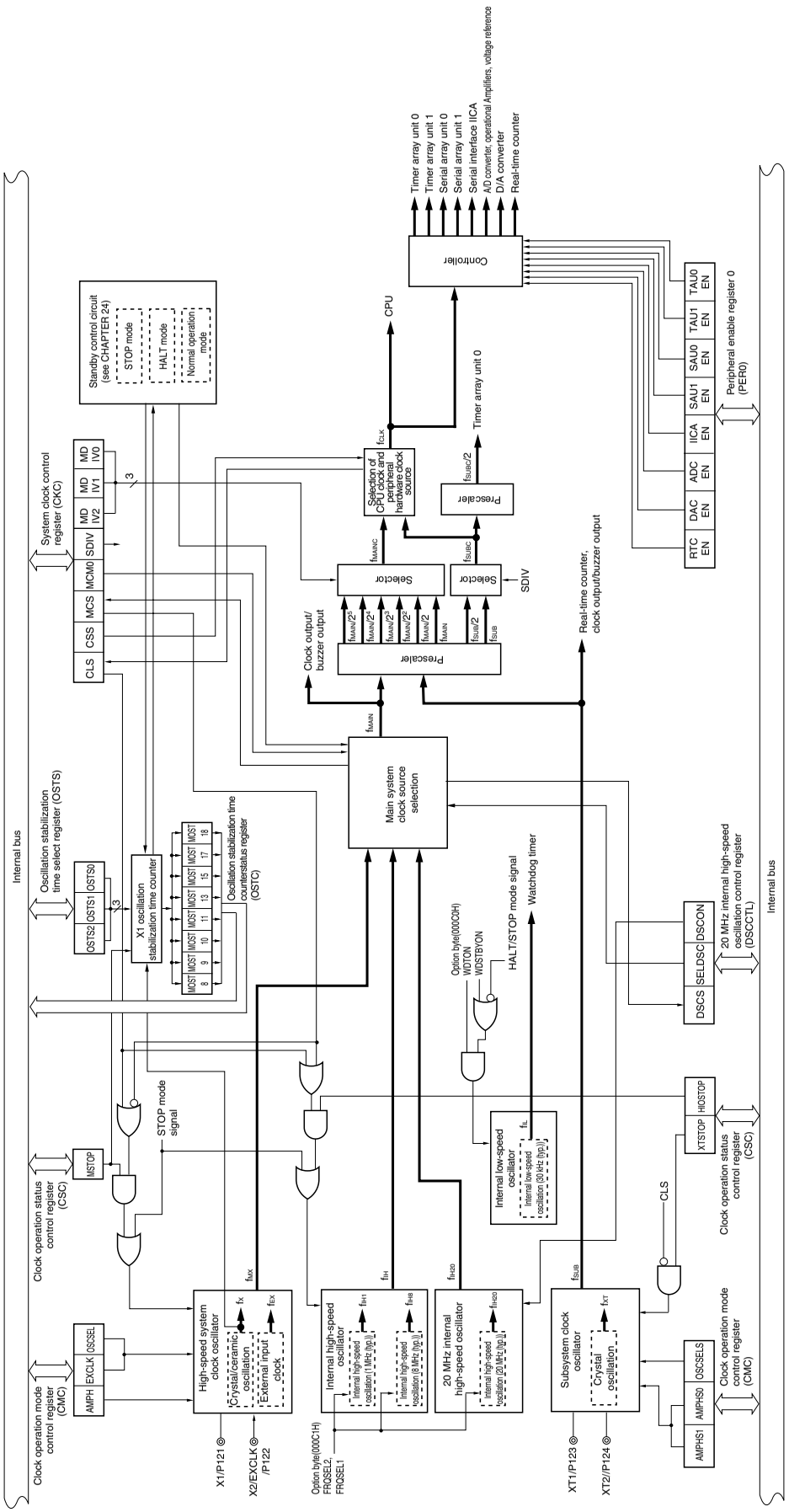
5.2 Configuration of Clock Generator

The clock generator includes the following hardware.

Table 5-1. Configuration of Clock Generator

Item	Configuration
Control registers	Clock operation mode control register (CMC) Clock operation status control register (CSC) Oscillation stabilization time counter status register (OSTC) Oscillation stabilization time select register (OSTS) System clock control register (CKC) 20 MHz internal high-speed oscillation control register (DSCCTL) Peripheral enable register 0 (PER0) Operation speed mode control register (OSMC)
Oscillators	X1 oscillator XT1 oscillator Internal high-speed oscillator Internal low-speed oscillator

Figure 5-1. Block Diagram of Clock Generator



Remark f_X: X1 clock oscillation frequency
f_{IH}: Internal high-speed oscillation clock frequency
f_{IH1}: 1 MHz internal high-speed oscillation clock frequency
f_{IH8}: 8 MHz internal high-speed oscillation clock frequency
f_{IH20}: 20 MHz internal high-speed oscillation clock frequency
f_{EX}: External main system clock frequency
f_{MX}: High-speed system clock frequency
f_{MAIN}: Main system clock frequency
f_{MAINC}: Main system selection clock frequency
f_{XT}: XT1 clock oscillation frequency
f_{SUB}: Subsystem clock frequency
f_{SUBC}: Subsystem selection clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency
f_{IL}: Internal low-speed oscillation clock frequency

5.3 Registers Controlling Clock Generator

The following eight registers are used to control the clock generator.

- Clock operation mode control register (CMC)
- Clock operation status control register (CSC)
- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)
- System clock control register (CKC)
- 20 MHz internal high-speed oscillation control register (DSCCTL)
- Peripheral enable register 0 (PER0)
- Operation speed mode control register (OSMC)

(1) Clock operation mode control register (CMC)

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

CMC can be written only once by an 8-bit memory manipulation instruction after reset release. This register can be read by a 1-bit or 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	OSCSELS	0	AMPHS1	AMPHS0	AMPH

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

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

AMPHS1	AMPHS0	XT1 oscillator oscillation mode selection
0	0	Low-consumption oscillation
0	1	Normal oscillation
1	0	Super-low-consumption oscillation
1	1	

AMPH	Control of high-speed system clock oscillation frequency
0	2 MHz ≤ f_{MX} ≤ 10 MHz
1	10 MHz < f_{MX} ≤ 20 MHz

Remark f_{MX}: High-speed system clock frequency

Cautions
1. **CMC can be written only once after reset release, by an 8-bit memory manipulation instruction.**
2. **After reset release, set CMC before X1 or XT1 oscillation is started as set by the clock operation status control register (CSC).**
3. **Be sure to set AMPH to 1 if the X1 clock oscillation frequency exceeds 10 MHz.**
4. **To use CMC with its initial value (00H), be sure to set it to 00H after releasing reset in order to prevent malfunction when a program loop occurs.**
5. **The XT1 oscillator is designed as a low-gain circuit for achieving low-power consumption. Note the following points when designing the XT1 oscillator.**
   - **The pins and circuit board include parasitic capacitance. Therefore, confirm that there are no problems by performing oscillation evaluation on the circuit board to be actually used.**
   - **When low-consumption oscillation or super-low-consumption oscillation is selected, lower power consumption than when selecting normal oscillation can be achieved. However, in this case, the XT1 oscillation margin is reduced, so perform sufficient oscillation evaluation of the resonator to be used for XT1 oscillation before using the resonator.**

(Cautions are continued on the next page.)

- **Keep the wiring length between the XT1 and XT2 pins and resonator as short as possible and parasitic capacitance and wire resistance as small as possible. This is particularly important when super-low-consumption oscillation (AMPHS1 = 1) is selected.**
- **Configure the circuit board by using material with little parasitic capacitance and wire resistance.**
- **Place a ground pattern that has the same potential as V_{SS} (if possible) around the XT1 oscillator.**
- **Do not cross the signal lines between the XT1 and XT2 pins and the resonator with other signal lines. Do not route the signal lines near a signal line through which a high fluctuating current flows.**
- **Moisture absorption by the circuit board and condensation on the board in a highly humid environment may cause the impedance between the XT1 and XT2 pins to drop and disable oscillation. When using the circuit board in such an environment, prevent the circuit board from absorbing moisture by taking measures such as coating the circuit board.**
- **Coat the surface of the circuit board by using material that does not generate capacitance or leakage between the XT1 and XT2 pins.**

(2) Clock operation status control register (CSC)

This register is used to control the operations of the high-speed system clock, internal high-speed oscillation clock, and subsystem clock (except the 20 MHz internal high-speed oscillation clock and internal low-speed oscillation clock).

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

Reset signal generation sets this register to C0H.

Figure 5-3. Format of Clock Operation Status Control Register (CSC)

Address: FFFA1H After reset: C0H R/W

Symbol	<7>	<6>	5	4	3	2	1	<0>
CSC	MSTOP	XTSTOP	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	–
1	X1 oscillator stopped	External clock from EXCLK pin is invalid	

XTSTOP	Subsystem clock operation control	
	XT1 oscillation mode	Input port mode
0	XT1 oscillator operating	–
1	XT1 oscillator stopped	

HIOSTOP	Internal high-speed oscillation clock operation control
0	Internal high-speed oscillator operating
1	Internal high-speed oscillator stopped

Caution 1. After reset release, set the clock operation mode control register (CMC) before starting X1 oscillation as set by MSTOP or XT1 oscillation as set by XTSTOP.

Cautions 2. To start X1 oscillation as set by MSTOP, check the oscillation stabilization time of the X1 clock by using the oscillation stabilization time counter status register (OSTC).

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

4. 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 follows.

Table 5-2. Condition Before Stopping Clock Oscillation and Flag Setting

Clock	Condition Before Stopping Clock (Invalidating External Clock Input)	Setting of CSC Register Flags
X1 clock External main system clock	CPU/peripheral hardware clock operates with a clock other than the high-speed system clock. (CLS = 0 and MCS = 0, or CLS = 1)	MSTOP = 1
Subsystem clock	CPU/peripheral hardware clock operates with a clock other than the subsystem clock. (CLS = 0)	XTSTOP = 1
Internal high-speed oscillation clock	CPU and peripheral hardware clocks operate with a clock other than the internal high-speed oscillator clock and 20 MHz internal high-speed oscillation clock. (CLS = 0 and MCS = 1, or CLS = 1)	HIOSTOP = 1

<R>

(3) Oscillation stabilization time counter status register (OSTC)

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

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

OSTC can be read by a 1-bit or 8-bit memory manipulation instruction.
When reset signal is generated, the STOP instruction and MSTOP (bit 7 of CSC register) = 1 clear OSTC to 00H.

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

- When oscillation of the X1 clock starts (EXCLK, OSCSEL = 0, 1 → MSTOP = 0)
- When the STOP mode is released

Figure 5-4. 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	MOST 8	MOST 9	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		
									f_X = 10 MHz	f_X = 20 MHz
0	0	0	0	0	0	0	0	$2^8/f_X$ max.	25.6 μs max.	12.8 μs max.
1	0	0	0	0	0	0	0	$2^8/f_X$ min.	25.6 μs min.	12.8 μs min.
1	1	0	0	0	0	0	0	$2^9/f_X$ min.	51.2 μs min.	25.6 μs min.
1	1	1	0	0	0	0	0	$2^{10}/f_X$ min.	102.4 μs min.	51.2 μs min.
1	1	1	1	0	0	0	0	$2^{11}/f_X$ min.	204.8 μs min.	102.4 μs min.
1	1	1	1	1	0	0	0	$2^{13}/f_X$ min.	819.2 μs min.	409.6 μs min.
1	1	1	1	1	1	0	0	$2^{15}/f_X$ min.	3.27 ms min.	1.64 ms min.
1	1	1	1	1	1	1	0	$2^{17}/f_X$ min.	13.11 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	$2^{18}/f_X$ min.	26.21 ms min.	13.11 ms min.

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

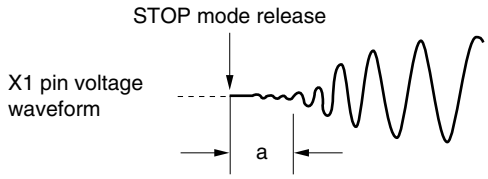
2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS.

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

- **If the X1 clock starts oscillation while the internal high-speed oscillation clock or subsystem clock is being used as the CPU clock.**
- **If the STOP mode is entered and then released while the internal high-speed oscillation 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 OSTS is set to OSTC after the STOP mode is released.)

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



Remark f_X: X1 clock oscillation frequency

(4) Oscillation stabilization time select register (OSTS)

This register is used to select the X1 clock oscillation stabilization wait time when the STOP mode is released. When the X1 clock is selected as the CPU clock, the operation automatically waits for the time set using OSTS after the STOP mode is released.

When the internal high-speed oscillation clock is selected as the CPU clock, confirm with OSTC that the desired oscillation stabilization time has elapsed after the STOP mode is released. The oscillation stabilization time can be checked up to the time set using OSTC.

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

Reset signal generation sets OSTS to 07H.

Figure 5-5. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFFA3H After reset: 07H R/W

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

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection		
				f_X = 10 MHz	f_X = 20 MHz
0	0	0	$2^8/f_X$	25.6 μs	Setting prohibited
0	0	1	$2^9/f_X$	51.2 μs	25.6 μs
0	1	0	$2^{10}/f_X$	102.4 μs	51.2 μs
0	1	1	$2^{11}/f_X$	204.8 μs	102.4 μs
1	0	0	$2^{13}/f_X$	819.2 μs	409.6 μs
1	0	1	$2^{15}/f_X$	3.27 ms	1.64 ms
1	1	0	$2^{17}/f_X$	13.11 ms	6.55 ms
1	1	1	$2^{18}/f_X$	26.21 ms	13.11 ms

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

2. Setting the oscillation stabilization time to 20 μs or less is prohibited.

3. To change the setting of the OSTS register, be sure to confirm that the counting operation of the OSTC register has been completed.

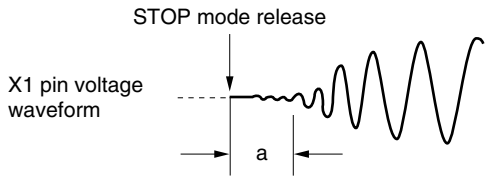
4. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

5. The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS.

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

- **If the X1 clock starts oscillation while the internal high-speed oscillation clock or subsystem clock is being used as the CPU clock.**
- **If the STOP mode is entered and then released while the internal high-speed oscillation 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 OSTS is set to OSTC after the STOP mode is released.)**

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



Remark f_X: X1 clock oscillation frequency

(5) System clock control register (CKC)

This register is used to select a CPU/peripheral hardware clock and a division ratio.

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

Reset signal generation sets this register to 09H.

Figure 5-6. Format of System Clock Control Register (CKC)

Address: FFFA4H After reset: 09H R/W[Note 1]

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
CKC	CLS	CSS	MCS	MCM0	SDIV	MDIV2	MDIV1	MDIV0

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

<R>

MCS	Status of Main system clock (f_{MAIN})
0	Internal high-speed oscillation clock (f_{IH}) or 20 MHz internal high-speed oscillation clock (f_{IH20})
1	High-speed system clock (f_{MX})

CSS	MCM0	SDIV	MDIV2	MDIV1	MDIV0	Selection of CPU/peripheral hardware clock (f_{CLK})
0	0	×	0	0	0	f_{IH}
		×	0	0	1	$f_{IH}/2$ (default)
		×	0	1	0	$f_{IH}/2^2$
		×	0	1	1	$f_{IH}/2^3$
		×	1	0	0	$f_{IH}/2^4$ Note 2
		×	1	0	1	$f_{IH}/2^5$ Note 2
0	1	×	0	0	0	f_{MX}
		×	0	0	1	$f_{MX}/2$
		×	0	1	0	$f_{MX}/2^2$
		×	0	1	1	$f_{MX}/2^3$
		×	1	0	0	$f_{MX}/2^4$
		×	1	0	1	$f_{MX}/2^5$ Note 3
1 Note 4	× Note 4	0	×	×	×	f_{SUB}
		1	×	×	×	$f_{SUB}/2$
Other than above						Setting prohibited

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

2. Setting is prohibited when f_{IH} = 1 MHz.

3. Setting is prohibited when f_{MX} < 4 MHz.

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

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

Remarks 1. f_{IH}: Internal high-speed oscillation clock frequency
f_{IH20}: 20 MHz Internal high-speed oscillation clock frequency
f_{MX}: High-speed system clock frequency
f_{SUB} Subsystem clock frequency
2. ×: don't care

Cautions 1. The clock set by CSS, MCM0, SDIV, and MDIV2 to MDIV0 is supplied to the CPU and peripheral hardware. If the CPU clock is changed, therefore, the clock supplied to peripheral hardware (except the real-time counter, timer array unit (when f_{SUB}/2, f_{SUB}/4, the valid edge of TI0mn input, or the valid edge of INTRTCI is selected as the count clock), clock output/buzzer output, and watchdog timer) is also changed at the same time. Consequently, stop each peripheral function when changing the CPU/peripheral operating hardware clock.

2. If the peripheral hardware clock is used as the subsystem clock, the operations of the A/D converter and IICA are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 29 ELECTRICAL SPECIFICATIONS.

The fastest instruction can be executed in 1 clock of the CPU clock in the 78K0R/Kx3-A microcontrollers. Therefore, the relationship between the CPU clock (f_{CLK}) and the minimum instruction execution time is as shown in Table 5-3.

Table 5-3. Relationship Between CPU Clock and Minimum Instruction Execution Time

<R>

CPU Clock (Value set by the SDIV, and MDIV2 to MDIV0 bits)	Minimum Instruction Execution Time: 1/f_{CLK}				
	Main System Clock (CSS = 0)				Subsystem Clock (CSS = 1)
	High-Speed System Clock (MCM0 = 1)		Internal High-Speed Oscillation Clock (MCM0 = 0)		
	At 10 MHz Operation	At 20 MHz Operation	At 8 MHz (TYP.) Operation	At 20 MHz (TYP.) Operation	At 32.768 kHz Operation
f_{MAIN}	0.1 μs	0.05 μs	0.125 μs (TYP.)	0.05 μs (TYP.)	–
f_{MAIN}/2	0.2 μs	0.1 μs	0.25 μs (TYP.) (default)	0.1 μs (TYP.)	–
f_{MAIN}/2^2	0.4 μs	0.2 μs	0.5 μs (TYP.)	0.2 μs (TYP.)	–
f_{MAIN}/2^3	0.8 μs	0.4 μs	1.0 μs (TYP.)	0.4 μs (TYP.)	–
f_{MAIN}/2^4	1.6 μs	0.8 μs	2.0 μs (TYP.)	0.8 μs (TYP.)	–
f_{MAIN}/2^5	3.2 μs	1.6 μs	4.0 μs (TYP.)	1.6 μs (TYP.)	–
f_{SUB}	–		–		30.5 μs
f_{SUB}/2	–		–		61 μs

Remark f_{MAIN}: Main system clock frequency (f_{IH}, f_{IH20}, or f_{MX})
f_{SUB}: Subsystem clock frequency

(6) 20 MHz internal high-speed oscillation control register (DSCCTL)

This register controls the 20 MHz internal high-speed oscillation clock (DSC) function.

It can be used to select whether to use the 20 MHz internal high-speed oscillation clock (f_{IH20}) as a peripheral hardware clock that supports 20 MHz.

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

Reset signal generation clears this register to 00H.

Figure 5-7. Format of 20 MHz Internal High-Speed Oscillation Control Register (DSCCTL)

Address: F00F6H After reset: 00H R/W[Note]

Symbol	7	6	5	4	<3>	<2>	1	<0>
DSCCTL	0	0	0	0	DSCS	SELDSC	0	DSCON

DSCS	20 MHz internal high-speed oscillation supply status flag
0	Not supplied
1	Supplied

SELDSC	Selection of 20 MHz internal high-speed oscillation for CPU/peripheral hardware clock (f_{CLK})
0	Does not select 20 MHz internal high-speed oscillation (clock selected by CKC register is supplied to f_{CLK})
1	Selects 20 MHz internal high-speed oscillation (20 MHz internal high-speed oscillation is supplied to f_{CLK})

DSCON	20 MHz internal high-speed oscillation clock (f_{IH20}) operation enable/disable
0	Disables operation.
1	Enables operation.

Note Bit 3 is read-only.

Cautions 1. 20 MHz internal oscillation can only be used if $V_{DD} \geq 2.7$ V.

2. Set SELDSC when 100 μs have elapsed after having set DSCON with $V_{DD} \geq 2.7$ V.

3. The internal high-speed oscillator must be operated (HIOSTOP = 0) when DSCON = 1.

(7) Peripheral enable register 0 (PER0)

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

PER0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears theses registers to 00H.

Figure 5-8. Format of Peripheral Enable Register 0 (PER0) (1/2)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	DACEN	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

RTCEN	Control of real-time counter (RTC) input clock[Note]
0	Stops input clock supply. • SFR used by the real-time counter (RTC) cannot be written. • The real-time counter (RTC) is in the reset status.
1	Supplies input clock. • SFR used by the real-time counter (RTC) can be read and written.

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

ADCEN	Control of A/D converter, operational amplifier, and voltage reference input clock
0	Stops input clock supply. • SFR used by the A/D converter, operational amplifier, and voltage reference cannot be written. • The A/D converter, operational amplifier, and voltage reference are in the reset status.
1	Supplies input clock. • SFR used by the A/D converter, operational amplifier, and voltage reference can be read and written.

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

Note By using RTCEN, can supply and stop the clock that is used when accessing the real-time counter (RTC) from the CPU. RTCEN cannot control supply of the operating clock (f_{SUB}) to RTC.

Figure 5-8. Format of Peripheral Enable Register 0 (PER0) (2/2)

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

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

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

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

(8) Operation speed mode control register (OSMC)

FLPC and FSEL can be used to control the step-up circuit of the flash memory for high-speed operation.

If the microcontroller operates at a low speed with a system clock of 10 MHz or less, the power consumption can be lowered by setting this register to the default value, 00H. Furthermore, when operating the system clock at 1 MHz, the power consumption can be further reduced by setting FLPC to 1.

RTCLPC can be used to set the operation in subsystem clock HALT mode.

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

Reset signal generation clears this register to 00H.

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

Address: F00F3H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
OSMC	RTCLPC	0	0	0	0	0	FLPC	FSEL

RTCLPC	Setting in subsystem clock HALT mode
0	Enables subsystem clock supply to peripheral functions. (See Table 19-1 Operating Statuses in HALT Mode (2/3) for the peripheral functions whose operations are enabled.)
1	Stops subsystem clock supply to peripheral functions except real-time counter, and clock output/buzzer output.

FLPC	FSEL	f_{CLK} frequency selection
0	0	Operates at a frequency of 10 MHz or less (default).
0	1	Operates at a frequency higher than 10 MHz.
1	0	Operates at a frequency of 1 MHz.
1	1	Setting prohibited

Cautions 1. Write "1" to FSEL before the following two operations.
- **Changing the clock prior to dividing f_{CLK} to a clock other than f_{IH}.**
- **Operating the DMA controller.**

<R> **2. The CPU waits (140.5 clock (f_{CLK})) when "1" is written to the FSEL bit. Interrupt requests issued during a wait will be suspended. However, counting the oscillation stabilization time of f_X can continue even while the CPU is waiting.**

3. To increase f_{CLK} to 10 MHz or higher, set FSEL to "1", then change f_{CLK} after two or more clocks have elapsed.

4. Confirm that the clock is operating at 10 MHz or less before setting FSEL = 0.

5. To shift to STOP mode while $V_{DD} \leq 2.7$ V, set FSEL = 0 after setting f_{CLK} to 10 MHz or less.

(Cautions are given on the next page.)

Cautions 6. The HALT mode current when operating on the subsystem clock can be reduced by setting RTCLPC to 1. However, the clock cannot be supplied to peripheral functions except the real-time counter in the subsystem clock HALT mode. Set bit 7 (RTCEN) of PER0 to 1 and bits 0 to 6 of PER0 to 0 before setting the subsystem clock HALT mode.

7. Once FLPC has been set from 0 to 1, setting it back to 0 from 1 other than by reset is prohibited.

8. When setting FSEL to "1", do so while RMC = 00H.
When setting FLPC to "1", do so while RMC = 5AH.

5.4 System Clock Oscillator

5.4.1 X1 oscillator

The X1 oscillator oscillates with a crystal resonator or ceramic resonator (2 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

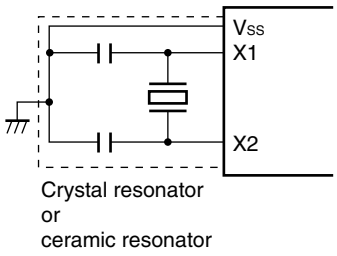
When the X1 oscillator is not used, set the input port mode (EXCLK, OSCSEL = 0, 0).

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

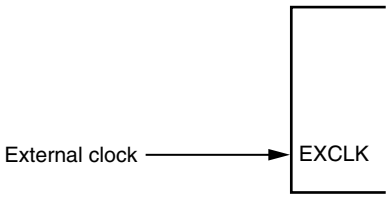
Figure 5-10 shows an example of the external circuit of the X1 oscillator.

Figure 5-10. Example of External Circuit of X1 Oscillator

(a) Crystal or ceramic oscillation



(b) External clock



Cautions are listed on the next page.

5.4.2 XT1 oscillator

The XT1 oscillator oscillates with a crystal resonator (standard: 32.768 kHz) connected to the XT1 and XT2 pins.

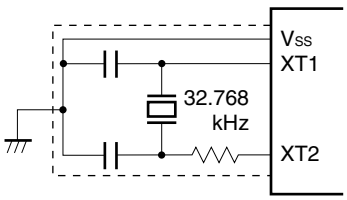
To use the XT1 oscillator, set bit 4 (OSCSELS) of the clock operation mode control register (CMC) to 1.

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

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

Figure 5-11 shows an example of the external circuit of the XT1 oscillator.

Figure 5-11. Example of External Circuit of XT1 Oscillator (Crystal Oscillation)



Cautions are listed on the next page.

Caution 1. When using the X1 oscillator and XT1 oscillator, wire as follows in the area enclosed by the broken lines in the Figures 5-10 and 5-11 to avoid an adverse effect from wiring capacitance.

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

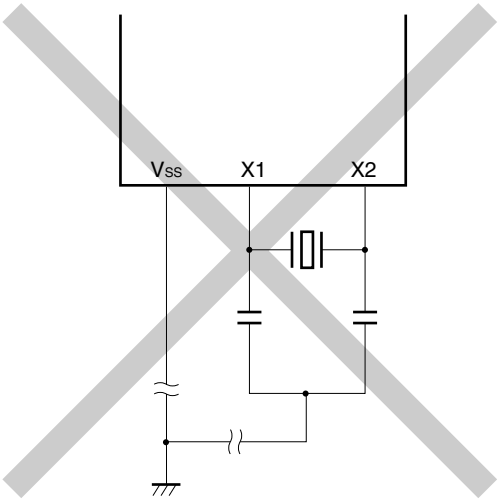
Note that the XT1 oscillator is designed as a low-gain circuit for achieving low-power consumption. Note the following points when designing the XT1 oscillator.

- **The pins and circuit board include parasitic capacitance. Therefore, confirm that there are no problems by performing oscillation evaluation on the circuit board to be actually used.**
- **When low-consumption oscillation or super-low-consumption oscillation is selected, lower power consumption than when selecting normal oscillation can be achieved. However, in this case, the XT1 oscillation margin is reduced, so perform sufficient oscillation evaluation of the resonator to be used for XT1 oscillation before using the resonator.**
- **Keep the wiring length between the XT1 and XT2 pins and resonator as short as possible and parasitic capacitance and wire resistance as small as possible. This is particularly important when super-low-consumption oscillation (AMPHS1 = 1) is selected.**
- **Configure the circuit board by using material with little parasitic capacitance and wire resistance.**
- **Place a ground pattern that has the same potential as V_{SS} (if possible) around the XT1 oscillator.**
- **Do not cross the signal lines between the XT1 and XT2 pins and the resonator with other signal lines. Do not route the signal lines near a signal line through which a high fluctuating current flows.**
- **Moisture absorption by the circuit board and condensation on the board in a highly humid environment may cause the impedance between the XT1 and XT2 pins to drop and disable oscillation. When using the circuit board in such an environment, prevent the circuit board from absorbing moisture by taking measures such as coating the circuit board.**
- **Coat the surface of the circuit board by using material that does not generate capacitance or leakage between the XT1 and XT2 pins.**

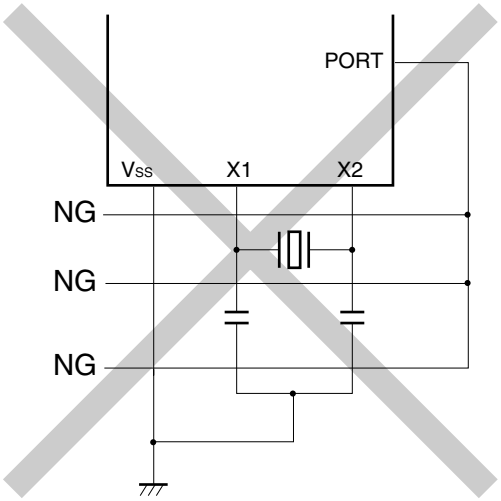
Figure 5-12 shows examples of incorrect resonator connection.

Figure 5-12. Examples of Incorrect Resonator Connection (1/2)

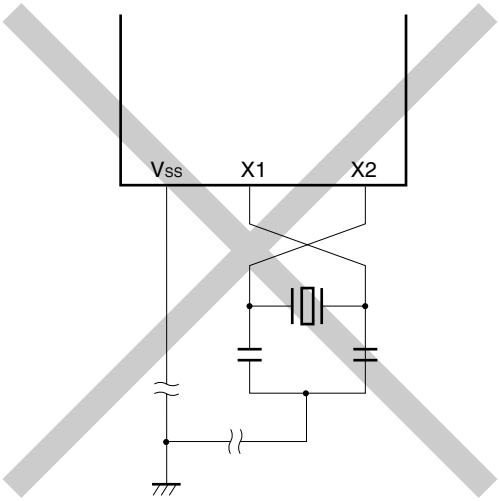
(a) Too long wiring



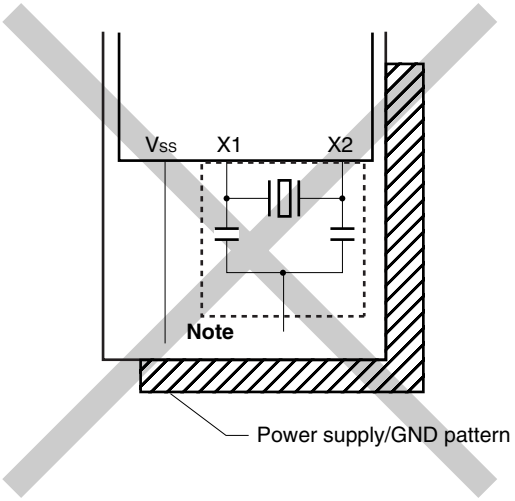
(b) Crossed signal line



(c) Signal lines of X1 and X2 cross



(d) Power supply/GND pattern exists underneath X1 and X2 wiring

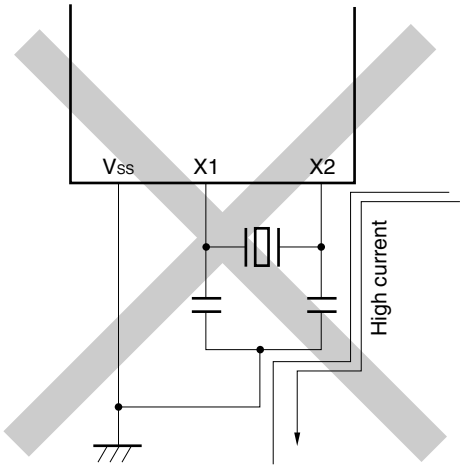


Note Do not place a power supply/GND pattern underneath the wiring section (in broken lines above) of the X1 and X2 pins and resonator in the multilayer board and double-sided board.
Do not configure a layout that may cause capacitance elements and affect the oscillation characteristics.

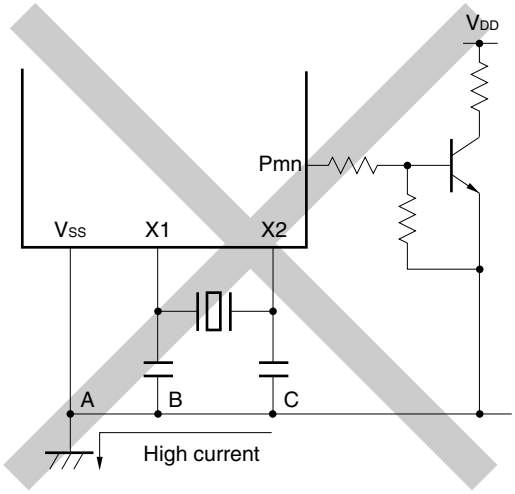
Remark When using the subsystem clock, replace X1 and X2 with XT1 and XT2, respectively. Also, insert resistors in series on the XT2 side.

Figure 5-12. Examples of Incorrect Resonator Connection (2/2)

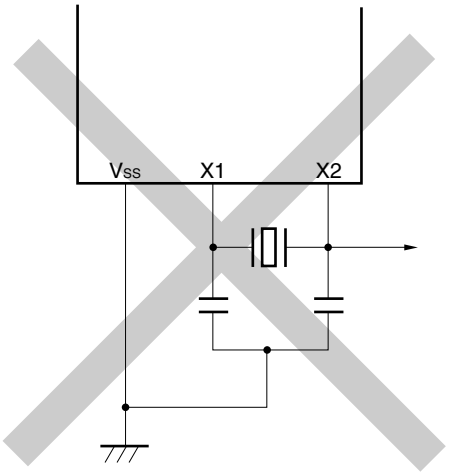
(e) Wiring near high alternating current



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



(g) Signals are fetched



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

Caution 2. When X2 and XT1 are wired in parallel, the crosstalk noise of X2 may increase with XT1, resulting in malfunctioning.

<R> **5.4.3 Internal high-speed oscillator**

The internal high-speed oscillator is incorporated in the 78K0R/Kx3-A (1, 8 and 20 MHz (TYP.)). Oscillation can be controlled by bit 0 (HIOSTOP) of the clock operation status control register (CSC) and bit 0 (DSCON) of the 20 MHz internal high-speed oscillation control register (DSCCTL).

Caution To use the 1, 8, or 20 MHz internal high-speed oscillation clock, use the option byte to set the frequency in advance (for details, see CHAPTER 24 OPTION BYTE). Also, the internal high-speed oscillator automatically starts oscillating after reset release. (If 8 MHz or 20 MHz is selected by using the option byte, the microcontroller operates using the 8 MHz internal high-speed oscillator.) To use the 20 MHz internal high-speed oscillator to operate the microcontroller, oscillation is started by setting bit 0 (DSCON) of the DSCCTL register to 1 with $V_{DD} \geq 2.7$ V.

5.4.4 Internal low-speed oscillator

The internal low-speed oscillator is incorporated in the 78K0R/Kx3-A microcontrollers.

The internal low-speed oscillation clock is used only as the watchdog timer clock. The internal low-speed oscillation clock cannot be used as the CPU clock.

After a reset release, the internal low-speed oscillator automatically starts oscillation, and the watchdog timer is driven (30 kHz (TYP.)) if the watchdog timer operation is enabled by the option byte.

The internal low-speed oscillator continues oscillation except when the watchdog timer stops. When the watchdog timer operates, the internal low-speed oscillation clock does not stop, even in case of a program loop.

5.4.5 Prescaler

The prescaler generates a CPU/peripheral hardware clock by dividing the main system clock and subsystem clock.

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}
  - Internal high-speed oscillation clock f_{IH}
    1 MHz internal high-speed oscillation clock f_{IH1}
    8 MHz internal high-speed oscillation clock f_{IH8}
  - 20 MHz internal high-speed oscillation clock f_{IH20}
- Subsystem clock f_{SUB}
- Subsystem selection clock f_{SUBC}
- Internal low-speed oscillation clock f_{IL}
- CPU/peripheral hardware clock f_{CLK}

The CPU starts operation when the internal high-speed oscillator starts outputting after a reset release in the 78K0R/Kx3-A microcontrollers, thus enabling the following.

(1) Enhancement of security function

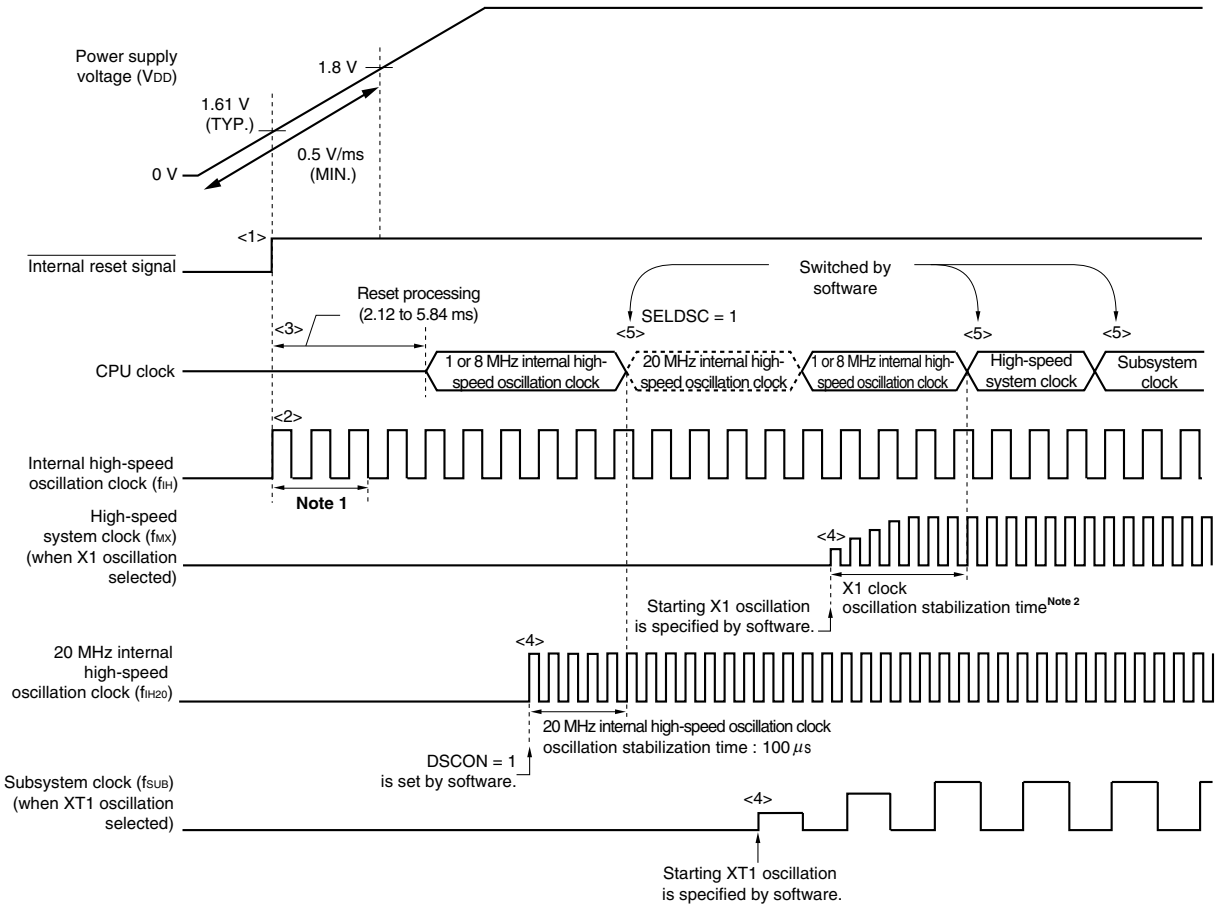
When the X1 clock is set as the CPU clock by the default setting, the device cannot operate if the X1 clock is damaged or badly connected and therefore does not operate after reset is released. However, the start clock of the CPU is the internal high-speed oscillation clock, so the device can be started by the internal high-speed oscillation clock after a reset release. As a result, reset sources can be detected by software and the minimum amount of safety processing can be done during anomalies to ensure that the system terminates safely.

(2) Improvement of performance

Because the CPU can be started without waiting for the X1 clock oscillation stabilization time, the total performance can be improved.

When the power supply voltage is turned on, the clock generator operations are shown in Figures 5-13 and 5-14.

<R> **Figure 5-13. Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Stopped Is Set (Option Byte: LVIOFF = 1))**



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

<2> When the power supply voltage exceeds 1.61 V (TYP.), the reset is released and the internal high-speed oscillator automatically starts oscillation.

<3> The CPU starts operation on the internal high-speed oscillation clock Note 3 after a reset processing such as waiting for the voltage of the power supply or regulator to stabilize has been performed after reset release.

<4> Set the start of oscillation of the X1 or XT1 clock via software (see **(1)** in **5.6.1 Example of controlling high-speed system clock** and **(1)** in **5.6.3 Example of controlling subsystem clock)**.

<5> When switching the CPU clock to the X1 or XT1 clock, wait for the clock oscillation to stabilize, and then set switching via software (see **(3)** in **5.6.1 Example of controlling high-speed system clock** and **(2)** in **5.6.3 Example of controlling subsystem clock**).

Switch to the 20 MHz internal high-speed oscillation clock by setting the DSCON bit (bit 0 of the 20 MHz internal high-speed oscillation control register (DSCCTL)), waiting for 100 μs, and then setting the SELDSC bit to 1 by using software Note 4.

(**Notes** and **Cautions** are listed on the next page.)

Notes 1. The internal reset processing time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.

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

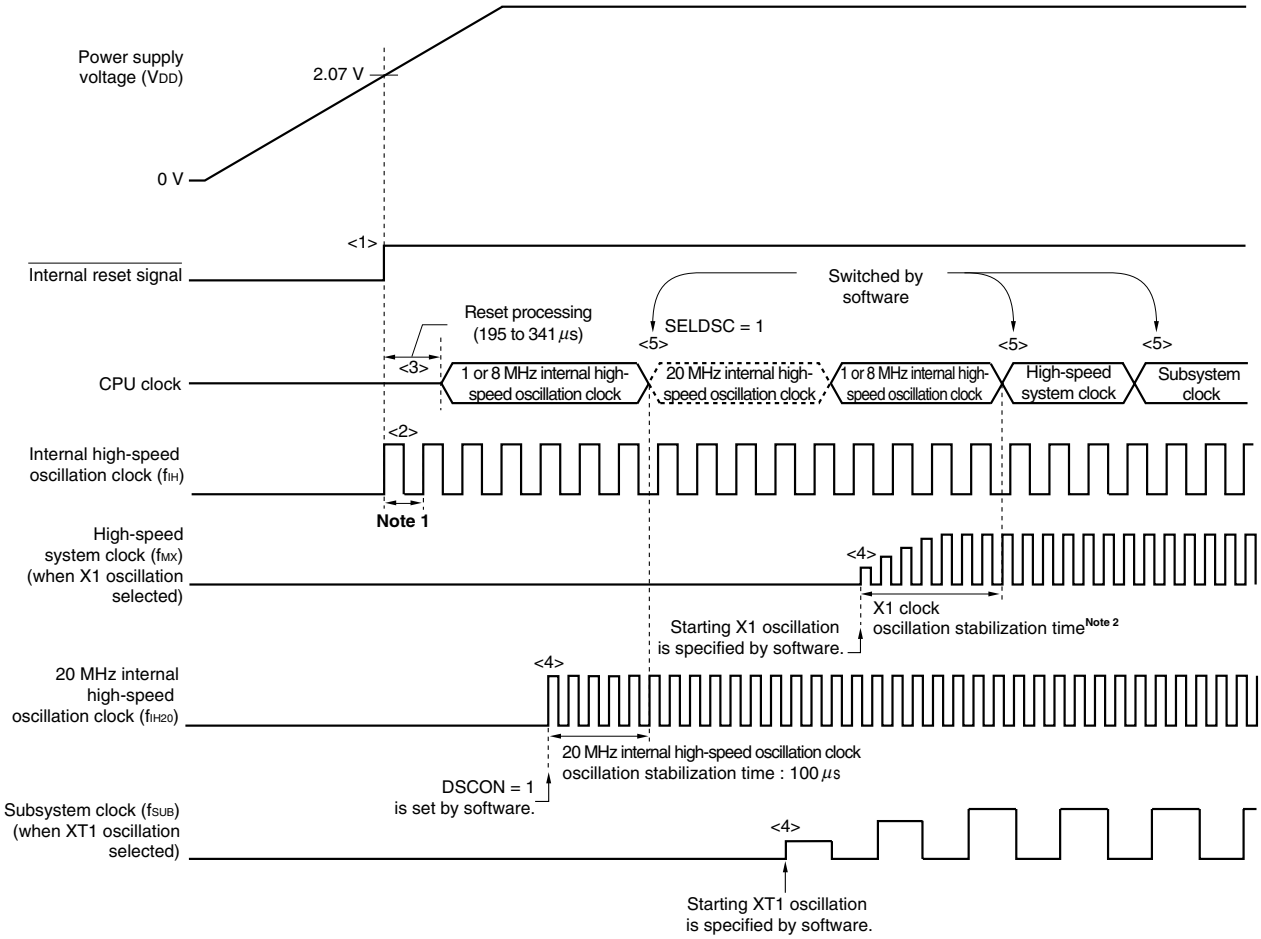
3. The microcontroller operates on the 8 MHz internal high-speed oscillation clock if 8 MHz or 20 MHz is selected for the internal high-speed oscillator by using the option byte or on the 1 MHz internal high-speed oscillation clock if 1 MHz is selected.

4. If the internal high-speed oscillator is set to 1 MHz by using the option byte, the 20 MHz internal high-speed oscillation clock cannot be used.

Cautions 1. If the voltage rises with a slope of less than 0.5 V/ms (MIN.) from power application until the voltage reaches 1.8 V, input a low level to the $\overline{\text{RESET}}$ pin from power application until the voltage reaches 1.8 V, or set the LVI default start function stopped by using the option byte (LVIOFF = 0) (see Figure 5-14). By doing so, the CPU operates with the same timing as <2> and thereafter in Figure 5-13 after reset release by the $\overline{\text{RESET}}$ pin.

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

<R> **Figure 5-14. Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Enabled Is Set (Option Byte: LVIOFF = 0))**



<1> When the power is turned on, an internal reset signal is generated by the low-voltage detector (LVI) circuit.

<2> When the power supply voltage exceeds 2.07 V (TYP.), the reset is released and the internal high-speed oscillator Note 3 automatically starts oscillation.

<3> After the reset is released and reset processing is performed, the CPU starts operation on the internal high-speed oscillation clock Note 3.

<4> Set the start of oscillation of the X1 or XT1 clock via software (see **(1)** in **5.6.1 Example of controlling high-speed system clock** and **(1)** in **5.6.3 Example of controlling subsystem clock**).
Switch to oscillation using the 20 MHz internal high-speed oscillation clock after setting the DSCON bit to 1 by using software.

<5> When switching the CPU clock to the X1 or XT1 clock, wait for the clock oscillation to stabilize, and then set switching via software (see **(3)** in **5.6.1 Example of controlling high-speed system clock** and **(2)** in **5.6.3 Example of controlling subsystem clock**).
Switch to the 20 MHz internal high-speed oscillation clock after confirming that the power supply voltage is at least 2.7 V, setting the DSCON bit (bit 0 of the 20 MHz internal high-speed oscillation control register (DSCCTL)), waiting for 100 μs, and then setting the SELDSC bit to 1 by using software Note 4.

(**Notes** and **Cautions** are listed on the next page.)

Notes 1. The internal reset processing time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.

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

3. The microcontroller operates on the 8 MHz internal high-speed oscillation clock if 8 MHz or 20 MHz is selected for the internal high-speed oscillator by using the option byte or on the 1 MHz internal high-speed oscillation clock if 1 MHz is selected.

4. If the internal high-speed oscillator is set to 1 MHz by using the option byte, the 20 MHz internal high-speed oscillation clock cannot be used.

Cautions 1. A voltage stabilization time (about 2.12 to 5.84 ms) is required after the supply voltage reaches 1.61 V (TYP.). If the time for the supply voltage to rise from 1.61 V (TYP.) to 2.07 V (TYP.) is shorter than the voltage stabilization time, reset processing is entered after the voltage stabilization time elapses.

2. 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 controlling high-speed system clock

The following two types of high-speed system clocks are available.

- X1 clock: Crystal/ceramic resonator is connected to the X1 and X2 pins.
- External main system clock: External clock is input to the EXCLK pin.

When the high-speed system clock is not used, the X1/P121 and X2/EXCLK/P122 pins can be used as input port pins.

Caution The X1/P121 and X2/EXCLK/P122 pins are in the input port mode after a reset release.

The following describes examples of setting procedures for the following cases.

(1) When oscillating X1 clock
(2) When using external main system clock
(3) When using high-speed system clock as CPU/peripheral hardware clock
(4) When stopping high-speed system clock

(1) Example of setting procedure when oscillating the X1 clock

<1> Setting P121/X1 and P122/X2/EXCLK pins and setting oscillation frequency (CMC register)

- 2 MHz ≤ fx ≤ 10 MHz

EXCLK	OSCSEL	0	OSCSELS	0	AMPHS1	AMPHS0	AMPH
0	1	0	0/1	0	0/1	0/1	0

- 10 MHz < fx ≤ 20 MHz

EXCLK	OSCSEL	0	OSCSELS	0	AMPHS1	AMPHS0	AMPH
0	1	0	0/1	0	0/1	0/1	1

Remarks 1. fx: X1 clock oscillation frequency

2. For setting of the P123/XT1 and P124/XT2 pins, see **5.6.3 Example of controlling subsystem clock**.

<2> Controlling oscillation of X1 clock (CSC register)
If MSTOP is cleared to 0, the X1 oscillator starts oscillating.

<3> Waiting for the stabilization of the oscillation of X1 clock
Check the OSTC register and wait for the necessary time.
During the wait time, other software processing can be executed with the internal high-speed oscillation clock.

Cautions 1. The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction.
Therefore, it is necessary to also set the value of the OSCSELS bit at the same time. For OSCSELS bit, see 5.6.3 Example of controlling subsystem clock.

2. Set the X1 clock after the supply voltage has reached the operable voltage of the clock to be used (see CHAPTER 29 ELECTRICAL SPECIFICATIONS).

(2) Example of setting procedure when using the external main system clock

<1> Setting P121/X1 and P122/X2/EXCLK pins (CMC register)

EXCLK	OSCSEL	0	OSCSELS	0	AMPHS1	AMPHS0	AMPH
1	1	0	0/1	0	0/1	0/1	0/1

Remark For setting of the P123/XT1 and P124/XT2 pins, see **5.6.3 (1) Example of setting procedure when oscillating the subsystem clock**.

<2> Controlling external main system clock input (CSC register)
When MSTOP is cleared to 0, the input of the external main system clock is enabled.

Cautions 1. The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction.
Therefore, it is necessary to also set the value of the OSCSELS bits at the same time. For OSCSELS bits, see 5.6.3 Example of controlling subsystem clock.
2. Set the external main system clock after the supply voltage has reached the operable voltage of the clock to be used (see CHAPTER 29 ELECTRICAL SPECIFICATIONS).

(3) Example of setting procedure when using high-speed system clock as CPU/peripheral hardware clock

<1> Setting high-speed system clock oscillation[Note]
(See **5.6.1 (1) Example of setting procedure when oscillating the X1 clock** and **(2) Example of setting procedure when using the external main system clock.**)

Note The setting of <1> is not necessary when high-speed system clock is already operating.

<2> Setting the high-speed system clock as the source clock of the CPU/peripheral hardware clock and setting the division ratio of the set clock (CKC register)

MCM0	MDIV2	MDIV1	MDIV0	Selection of CPU/Peripheral Hardware Clock (f_{CLK})
1	0	0	0	f_{MX}
	0	0	1	$f_{MX}/2$
	0	1	0	$f_{MX}/2^2$
	0	1	1	$f_{MX}/2^3$
	1	0	0	$f_{MX}/2^4$
	1	0	1	$f_{MX}/2^5$ [Note]

Note Setting is prohibited when f_{MX} < 4 MHz.

<3> If some peripheral hardware macros are not used, supply of the input clock to each hardware macro can be stopped.

(PER0 register)

RTCEN	DACEN	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

xxxEN	Input clock control
0	Stops input clock supply.
1	Supplies input clock.

Remark RTCEN: Control of the real-time counter input clock
DACEN: Control of the D/A converter input clock
ADCEN: Control of the A/D converter input clock, the operational amplifier input clock, and the voltage reference input clock
IICAEN: Control of the serial interface IICA input clock
SAU1EN: Control of the serial array unit 1 unit input clock
SAU0EN: Control of the serial array unit 0 unit input clock
TAU1EN: Control of the timer array unit 1 input clock
TAU0EN: Control of the timer array unit 0 input clock

(4) Example of setting procedure when stopping the high-speed system clock

The high-speed system clock can be stopped (disabling clock input if the external clock is used) in the following two ways.

- Executing the STOP instruction
- Setting MSTOP to 1

(a) To execute a STOP instruction

<1> Setting to stop peripheral hardware
Stop peripheral hardware that cannot be used in the STOP mode (for peripheral hardware that cannot be used in STOP mode, see **CHAPTER 19 STANDBY FUNCTION**).

<2> Setting the X1 clock oscillation stabilization time after STOP mode is released
If the X1 clock oscillates before the STOP mode is entered, set the value of the OSTS register before executing the STOP instruction.

<3> Executing the STOP instruction
When the STOP instruction is executed, the system is placed in the STOP mode and X1 oscillation is stopped (the input of the external clock is disabled).

(b) To stop X1 oscillation (disabling external clock input) by setting MSTOP to 1

<1> Confirming the CPU clock status (CKC register)

Confirm with CLS and MCS that the CPU is operating on a clock other than the high-speed system clock.

When CLS = 0 and MCS = 1, the high-speed system clock is supplied to the CPU, so change the CPU clock to the subsystem clock or internal high-speed oscillation clock.

CLS	MCS	CPU Clock Status
0	0	Internal high-speed oscillation clock or 20 MHz internal high-speed oscillation clock
0	1	High-speed system clock
1	×	Subsystem clock

<2> Setting of X1 clock oscillation stabilization time after restart of X1 clock oscillation[Note]

Prior to setting "1" to MSTOP, set the OSTS register to a value greater than the count value to be confirmed with the OSTS register after X1 clock oscillation is restarted.

<3> Stopping the high-speed system clock (CSC register)

When MSTOP is set to 1, X1 oscillation is stopped (the input of the external clock is disabled).

Note This setting is required to resume the X1 clock oscillation when the high-speed system clock is in the X1 oscillation mode.

This setting is not required in the external clock input mode.

Caution Be sure to confirm that MCS = 0 or CLS = 1 when setting MSTOP to 1. In addition, stop peripheral hardware that is operating on the high-speed system clock.

5.6.2 Example of controlling internal high-speed oscillation clock

The following describes examples of clock setting procedures for the following cases.

(1) When restarting oscillation of the internal high-speed oscillation clock
(2) When using internal high-speed oscillation clock as CPU/peripheral hardware clock
(3) When stopping the internal high-speed oscillation clock

(1) Example of setting procedure when restarting oscillation of the internal high-speed oscillation clock[Note]

<1> Setting restart of oscillation of the internal high-speed oscillation clock (CSC register)

When HIOSTOP is cleared to 0, the internal high-speed oscillation clock restarts oscillation.

Note After a reset release, the internal high-speed oscillator automatically starts oscillating and the internal high-speed oscillation clock is selected as the CPU/peripheral hardware clock.

(2) Example of setting procedure when using internal high-speed oscillation clock as CPU/peripheral hardware clock

<1> Restarting oscillation of the internal high-speed oscillation clock[Note]

(See **5.6.2 (1) Example of setting procedure when restarting internal high-speed oscillation clock**).

Note The setting of <1> is not necessary when the internal high-speed oscillation clock is operating.

<2> Setting the internal high-speed oscillation clock as the source clock of the CPU/peripheral hardware clock and setting the division ratio of the set clock (CKC register)

MCM0	MDIV2	MDIV1	MDIV0	Selection of CPU/Peripheral Hardware Clock (f_{CLK})
0	0	0	0	f_{IH}
	0	0	1	$f_{IH}/2$
	0	1	0	$f_{IH}/2^2$
	0	1	1	$f_{IH}/2^3$
	1	0	0	$f_{IH}/2^4$ Note
	1	0	1	$f_{IH}/2^5$ Note

Note Setting is prohibited when f_{IH} = 1 MHz.

Caution If switching the CPU/peripheral hardware clock from the high-speed system clock to the internal high-speed oscillation clock after restarting the internal high-speed oscillation clock, do so after 10 μs or more have elapsed.
If the switching is made immediately after the internal high-speed oscillation clock is restarted, the accuracy of the internal high-speed oscillation cannot be guaranteed for 10 μs.

(3) Example of setting procedure when stopping the internal high-speed oscillation clock

The internal high-speed oscillation clock can be stopped in the following two ways.

- Executing the STOP instruction
- Setting HIOSTOP to 1

(a) To execute a STOP instruction

<1> Setting of peripheral hardware
Stop peripheral hardware that cannot be used in the STOP mode (for peripheral hardware that cannot be used in STOP mode, see **CHAPTER 19 STANDBY FUNCTION**).

<2> Setting the X1 clock oscillation stabilization time after STOP mode is released
If the X1 clock oscillates before the STOP mode is entered, set the value of the OSTS register before executing the STOP instruction.

<3> Executing the STOP instruction
When the STOP instruction is executed, the system is placed in the STOP mode and internal high-speed oscillation clock is stopped.

(b) To stop internal high-speed oscillation clock by setting HIOSTOP to 1

<1> Confirming the CPU clock status (CKC register)

Confirm with CLS and MCS that the CPU is operating on a clock other than the internal high-speed oscillation clock.

When CLS = 0 and MCS = 0, the internal high-speed oscillation clock is supplied to the CPU, so change the CPU clock to the high-speed system clock or subsystem clock.

CLS	MCS	CPU Clock Status
0	0	Internal high-speed oscillation clock or 20 MHz internal high-speed oscillation clock
0	1	High-speed system clock
1	×	Subsystem clock

<2> Stopping the internal high-speed oscillation clock (CSC register)

When HIOSTOP is set to 1, internal high-speed oscillation clock is stopped.

Caution Be sure to confirm that MCS = 1 or CLS = 1 when setting HIOSTOP to 1. In addition, stop peripheral hardware that is operating on the internal high-speed oscillation clock.

5.6.3 Example of controlling subsystem clock

The subsystem clock can be oscillated by connecting a crystal resonator to the XT1 and XT2 pins.

When the subsystem clock is not used, the XT1/P123 and XT2/P124 pins can be used as input port pins.

Caution The XT1/P123 and XT2/P124 pins are in the input port mode after a reset release.

The following describes examples of setting procedures for the following cases.

(1) When oscillating subsystem clock

(2) When using subsystem clock as CPU clock

(3) When stopping subsystem clock

Caution When the subsystem clock is used as the CPU clock, the subsystem clock is also supplied to the peripheral hardware (except the real-time counter, timer array unit (when f_{SUB}/2, f_{SUB}/4, the valid edge of TI0mn input, or the valid edge of INTRTCI is selected as the count clock), clock output/buzzer output, and watchdog timer). At this time, the operations of the A/D converter and IICA are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 29 ELECTRICAL SPECIFICATIONS.

(1) Example of setting procedure when oscillating the subsystem clock

<1> Setting P123/XT1 and P124/XT2 pins (CMC register)

EXCLK	OSCSEL	0	OSCSELS	0	AMPHS1	AMPHS0	AMPH
0/1	0/1	0	1	0	0/1	0/1	0/1

Remark For setting of the P121/X1 and P122/X2 pins, see **5.6.1 Example of controlling high-speed system clock**.

<2> Controlling oscillation of subsystem clock (CSC register)

If XTSTOP is cleared to 0, the XT1 oscillator starts oscillating.

<3> Waiting for the stabilization of the subsystem clock oscillation

Wait for the oscillation stabilization time of the subsystem clock by software, using a timer function.

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

Therefore, it is necessary to also set the value of the EXCLK and OSCSEL bits at the same time. For EXCLK and OSCSEL bits, see 5.6.1 (1) Example of setting procedure when oscillating the X1 clock or 5.6.1 (2) Example of setting procedure when using the external main system clock.

(2) Example of setting procedure when using the subsystem clock as the CPU clock

<1> Setting subsystem clock oscillation[Note]

(See **5.6.3 (1) Example of setting procedure when oscillating the subsystem clock**.)

Note The setting of <1> is not necessary when while the subsystem clock is operating.

<2> Setting the subsystem clock as the source clock of the CPU/peripheral hardware clock and setting the division ratio of the set clock (CKC register)

CSS	SDIV	Selection of CPU/Peripheral Hardware Clock (f_{CLK})
1	0	f_{SUB}
	1	$f_{SUB}/2$

Caution When the subsystem clock is used as the CPU clock, the subsystem clock is also supplied to the peripheral hardware (except the real-time counter, timer array unit (when $f_{SUB}/2$, $f_{SUB}/4$, the valid edge of TI0mn input, or the valid edge of INTRTCI is selected as the count clock), clock output/buzzer output, and watchdog timer). At this time, the operations of the A/D converter and IICA are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 29 ELECTRICAL SPECIFICATIONS.

(3) Example of setting procedure when stopping the subsystem clock

<1> Confirming the CPU clock status (CKC register)

Confirm with CLS and MCS that the CPU is operating on a clock other than the subsystem clock.

When CLS = 1, the subsystem clock is supplied to the CPU, so change the CPU clock to the internal high-speed oscillation clock or high-speed system clock.

CLS	MCS	CPU Clock Status
0	0	Internal high-speed oscillation clock or 20 MHz internal high-speed oscillation clock
0	1	High-speed system clock
1	×	Subsystem clock

<2> Stopping the subsystem clock (CSC register)

When XTSTOP is set to 1, subsystem clock is stopped.

Cautions 1. Be sure to confirm that CLS = 0 when setting XTSTOP to 1. In addition, stop the peripheral hardware if it is operating on the subsystem clock.

2. The subsystem clock oscillation cannot be stopped using the STOP instruction.

5.6.4 Example of controlling internal low-speed oscillation clock

The internal low-speed oscillation clock cannot be used as the CPU clock. Used only as the watchdog timer clock.

The internal low-speed oscillator automatically starts oscillation after a reset release, and the watchdog timer is driven (30 kHz (TYP.)) if the watchdog timer operation is enabled by the option byte.

The internal low-speed oscillator continues oscillation except when the watchdog timer stops. When the watchdog timer operates, the internal low-speed oscillation clock does not stop even in case of a program loop.

(1) Example of setting procedure when stopping the internal low-speed oscillation clock

The internal low-speed oscillation clock can be stopped in the following two ways.

- Stop the watchdog timer in the HALT/STOP mode by the option byte (bit 0 (WDSTBYON) of 000C0H = 0), and execute the HALT or STOP instruction.
- Stop the watchdog timer by the option byte (bit 4 (WDTON) of 000C0H = 0).

(2) Example of setting procedure when restarting oscillation of the internal low-speed oscillation clock

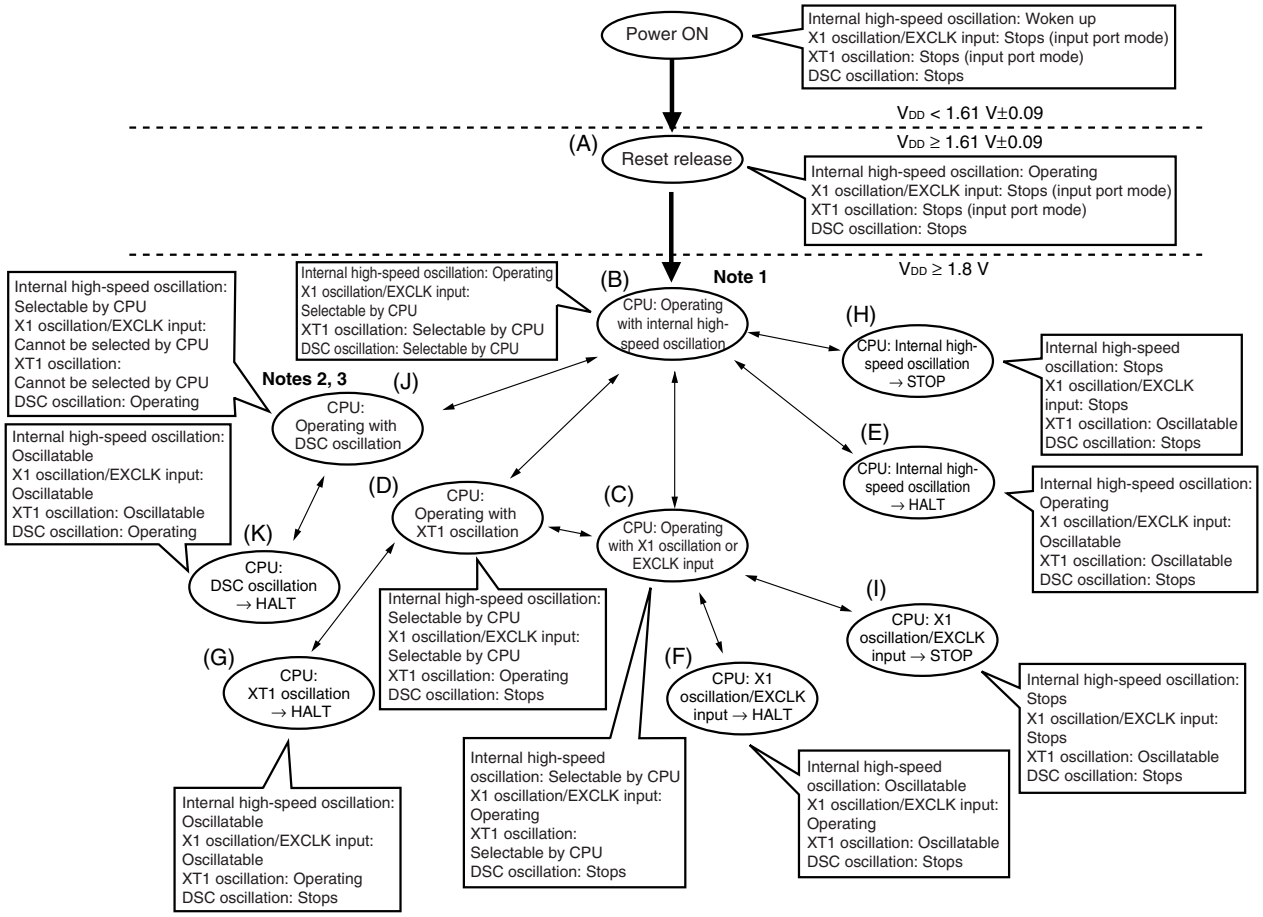
The internal low-speed oscillation clock can be restarted as follows.

- Release the HALT or STOP mode
  (only when the watchdog timer is stopped in the HALT/STOP mode by the option byte (bit 0 (WDSTBYON) of 000C0H) = 0) and when the watchdog timer is stopped as a result of execution of the HALT or STOP instruction).

<R> 5.6.5 CPU clock status transition diagram

Figure 5-15 shows the CPU clock status transition diagram of this product.

Figure 5-15. CPU Clock Status Transition Diagram



Notes 1. After reset release, an operation at one of the following operating frequencies is started, because $f_{CLK} = f_{IH}/2$ has been selected by setting the system clock control register (CKC) to 09H.

- When 1 MHz has been selected by using the option byte: 500 kHz (1 MHz/2)
- When 8 MHz or 20 MHz has been selected by using the option byte: 4 MHz (8 MHz/2)

2. Specify 20 MHz internal oscillation after checking that V_{DD} is at least 2.7 V.

3. 20 MHz internal oscillation cannot be used if 1 MHz internal oscillation is selected by using the option byte.

Remarks 1. If the low-power-supply detector (LVI) is set to ON by default by the option bytes, the reset will not be released until the power supply voltage (V_{DD}) exceeds 2.07 V±0.2 V.
After the reset operation, the status will shift to (B) in the above figure.

2. DSC: 20 MHz internal high-speed oscillation clock

Table 5-4 shows transition of the CPU clock and examples of setting the SFR registers.

Table 5-4. CPU Clock Transition and SFR Register Setting Examples (1/6)

(1) CPU operating with internal high-speed oscillation 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 internal high-speed oscillation clock immediately after a reset release (B).)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register Note 1			CSC Register	OSMC Register	OSTC Register	CKC Register
	EXCLK	OSCSEL	AMPH	MSTOP	FSEL		MCM0
(A) → (B) → (C) (X1 clock: 2 MHz ≤ fx ≤ 10 MHz)	0	1	0	0	0	Must be checked	1
(A) → (B) → (C) (X1 clock: 10 MHz < fx ≤ 20 MHz)	0	1	1	0	1 Note 2	Must be checked	1
(A) → (B) → (C) (external main clock)	1	1	×	0	0/1 Note 2	Must not be checked	1

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

2. FSEL = 1 when f_{CLK} > 10 MHz

If a divided clock is selected and f_{CLK} ≤ 10 MHz, use with FSEL = 0 is possible even if fx > 10 MHz.

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 29 ELECTRICAL SPECIFICATIONS).

(3) CPU operating with subsystem clock (D) after reset release (A)

(The CPU operates with the internal high-speed oscillation clock immediately after a reset release (B).)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register Note			CSC Register	Waiting for Oscillation Stabilization	CKC Register
	OSCSELS	AMPHS1	AMPHS0	XTSTOP		CSS
(A) → (B) → (D)	1	0/1	0/1	0	Necessary	1

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

Remark (A) to (K) in Table 5-4 correspond to (A) to (K) in Figure 5-15.

Table 5-4. CPU Clock Transition and SFR Register Setting Examples (2/6)

(4) CPU operating with 20 MHz internal high-speed oscillation clock (J) after reset release (A)

(The CPU operates with the internal high-speed oscillation clock immediately after a reset release (B).)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	DSCCTL Register Note	Waiting for Oscillation Stabilization	DSCCTL Register
	DSCON		SELDSC
(A) → (B) → (J)	1	Necessary (100 μs)	1

Note Check that $V_{DD} \geq 2.7$ V and set DSCON = 1.

(5) CPU clock changing from internal high-speed oscillation 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	OSMC Register	OSTC Register	CKC Register
	EXCLK	OSCSEL	AMPH		MSTOP	FSEL		MCM0
(B) → (C) (X1 clock: 2 MHz ≤ fX ≤ 10 MHz)	0	1	0	**Note 2**	0	0	Must be checked	1
(B) → (C) (X1 clock: 10 MHz < fX ≤ 20 MHz)	0	1	1	**Note 2**	0	1 Note 3	Must be checked	1
(B) → (C) (external main clock)	1	1	×	**Note 2**	0	0/1	Must not be checked	1

Unnecessary if these registers are already set

Unnecessary if the CPU is operating with the high-speed system clock

Notes 1. The CMC register can be changed only once after reset release. This setting is not necessary if it has already been set.

2. Set the oscillation stabilization time as follows.

- Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS

3. FSEL = 1 when f_{CLK} > 10 MHz

If a divided clock is selected and $f_{CLK} \leq$ 10 MHz, use with FSEL = 0 is possible even if f_X > 10 MHz.

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 29 ELECTRICAL SPECIFICATIONS).

Remarks 1. ×: don't care

2. (A) to (K) in Table 5-4 correspond to (A) to (K) in Figure 5-15.

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

(6) CPU clock changing from internal high-speed oscillation clock (B) to subsystem clock (D)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CMC Register[Note]	CSC Register	Waiting for Oscillation Stabilization	CKC Register
	OSCSELS	XTSTOP		CSS
(B) → (D)	1	0	Necessary	1

Unnecessary if the CPU is operating with the subsystem clock

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

(7) CPU clock changing from internal high-speed oscillation clock (B) to 20 MHz internal high-speed oscillation clock (J)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	DSCCTL Register[Note]	Waiting for Oscillation Stabilization	DSCCTL Register
	DSCON		SELDSC
(B) → (J)	1	Necessary (100 μs)	1

Unnecessary if the CPU is operating with the 20 MHz internal high-speed oscillation clock

Note Check that $V_{DD} \geq 2.7$ V and set DSCON = 1.

(8) CPU clock changing from high-speed system clock (C) to internal high-speed oscillation 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	10 μs	0

Unnecessary if the CPU is operating with the internal high-speed oscillation clock

Remark (A) to (K) in Table 5-4 correspond to (A) to (K) in Figure 5-15.

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

(9) CPU clock changing from high-speed system clock (C) to subsystem clock (D)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CSC Register	Waiting for Oscillation Stabilization	CKC Register
	XTSTOP		CSS
(C) → (D)	0	Necessary	1

Unnecessary if the CPU is operating with the subsystem clock

(10) CPU clock changing from subsystem clock (D) to internal high-speed oscillation clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	CSC Register	CKC Register	
	HIOSTOP	MCM0	CSS
(D) → (B)	0	0	0

Unnecessary if the CPU is operating with the internal high-speed oscillation clock

Unnecessary if this register is already set

Remark (A) to (K) in Table 5-4 correspond to (A) to (K) in Figure 5-15.

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

(11) CPU clock changing from subsystem clock (D) to high-speed system clock (C)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	OSTS Register	CSC Register	OSMC Register	OSTC Register	CKC Register	
		MSTOP	FSEL		MCM0	CSS
(D) → (C) (X1 clock: 2 MHz ≤ f_X ≤ 10 MHz)	Note 1	0	0	Must be checked	1	0
(D) → (C) (X1 clock: 10 MHz < f_X ≤ 20 MHz)	Note 1	0	1 Note 2	Must be checked	1	0
(D) → (C) (external main clock)	Note 1	0	0/1	Must not be checked	1	0

Unnecessary if the CPU is operating with the high-speed system clock

Unnecessary if these registers are already set

Notes 1. Set the oscillation stabilization time as follows.

- Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS

2. FSEL = 1 when f_{CLK} > 10 MHz

If a divided clock is selected and f_{CLK} ≤ 10 MHz, use with FSEL = 0 is possible even if f_X > 10 MHz.

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 29 ELECTRICAL SPECIFICATIONS).

(12) CPU clock changing from 20 MHz internal high-speed oscillation clock (J) to internal high-speed oscillation clock (B)

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	DSCCTL Register	
	SELDSC	DSCON
(J) → (B)	0	0

Remark (A) to (K) in Table 5-4 correspond to (A) to (K) in Figure 5-15.

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

(13) • HALT mode (E) set while CPU is operating with internal high-speed oscillation clock (B)
• HALT mode (F) set while CPU is operating with high-speed system clock (C)
• HALT mode (G) set while CPU is operating with subsystem clock (D)
• HALT mode (K) set while CPU is operating with 20 MHz internal high-speed oscillation clock (J)

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

(14) • STOP mode (H) set while CPU is operating with internal high-speed oscillation clock (B)
• STOP mode (I) set while CPU is operating with high-speed system clock (C)

(Setting sequence) ⟶

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

Remark (A) to (K) in Table 5-4 correspond to (A) to (K) in Figure 5-15.

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

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

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

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
Internal high-speed oscillation clock	X1 clock	Stabilization of X1 oscillation • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time	Operating current can be reduced by stopping internal high-speed oscillator (HIOSTOP = 1).
	External main system clock	Enabling input of external clock from EXCLK pin • OSCSEL = 1, EXCLK = 1, MSTOP = 0	
	Subsystem clock	Stabilization of X1 oscillation • OSCSELS = 1, XTSTOP = 0 • After elapse of oscillation stabilization time	–
	20 MHz internal high-speed oscillation clock	Stabilization of DSC oscillation with 20 MHz set by using the option byte • $V_{DD} \geq 2.7$ V • After elapse of oscillation stabilization time (100 μs) after setting to DSCON = 1 • SELDSC = 1	–
X1 clock	Internal high-speed oscillation clock	Oscillation of internal high-speed oscillator • HIOSTOP = 0	X1 oscillation can be stopped (MSTOP = 1).
	External main system clock	Transition not possible (To change the clock, set it again after executing reset once.)	–
	Subsystem clock	Stabilization of XT1 oscillation • OSCSELS = 1, XTSTOP = 0 • After elapse of oscillation stabilization time	X1 oscillation can be stopped (MSTOP = 1).
	20 MHz internal high-speed oscillation clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–
External main system clock	Internal high-speed oscillation clock	Oscillation of internal high-speed oscillator • HIOSTOP = 0	External main system clock input can be disabled (MSTOP = 1).
	X1 clock	Transition not possible (To change the clock, set it again after executing reset once.)	–
	Subsystem clock	Stabilization of XT1 oscillation • OSCSELS = 1, XTSTOP = 0 • After elapse of oscillation stabilization time	External main system clock input can be disabled (MSTOP = 1).
	20 MHz internal high-speed oscillation clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–

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

CPU Clock		Condition Before Change	Processing After Change
Before Change	After Change		
Subsystem clock	Internal high-speed oscillation clock	Oscillation of internal high-speed oscillator and selection of internal high-speed oscillation clock as main system clock • HIOSTOP = 0, MCS = 0	XT1 oscillation can be stopped (XTSTOP = 1)
	X1 clock	Stabilization of X1 oscillation and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 0, MSTOP = 0 • After elapse of oscillation stabilization time • MCS = 1	
	External main system clock	Enabling input of external clock from EXCLK pin and selection of high-speed system clock as main system clock • OSCSEL = 1, EXCLK = 1, MSTOP = 0 • MCS = 1	
	20 MHz internal high-speed oscillation clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–
20 MHz internal high-speed oscillation clock	Internal high-speed oscillation clock	• SELDSC = 0 (Set when changing the clock.)	20 MHz internal high-speed oscillation clock can be stopped (DSCON = 0)
	X1 clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–
	External main system clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–
	Subsystem clock	Transition cannot be performed unless the clock is changed to the internal high-speed oscillation clock once.	–

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

By setting bits 0 to 4, and 6 (MDIV0 to MDIV2, SDIV, MCM0, CSS) of the system clock control register (CKC), the CPU clock can be switched (between the main system clock and the subsystem clock), main system clock can be switched (between the internal high-speed oscillation clock and the high-speed system clock), and the division ratio of the main system clock can be changed.

The actual switchover operation is not performed immediately after rewriting to CKC; operation continues on the pre-switchover clock for several clocks (see **Tables 5-6 to 5-9**).

Whether the CPU is operating on the main system clock or the subsystem clock can be ascertained using bit 7 (CLS) of CKC. Whether the main system clock is operating on the high-speed system clock or internal high-speed oscillation clock can be ascertained using bit 5 (MCS) of CKC.

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

Internal high-speed oscillation clock

Table 5-6. Maximum Time Required for Main System Clock Switchover

Clock A	Switching directions	Clock B	Remark
f_{MAINC}	◄► (changing the division ratio)	f_{MAINC}	see Table 5-7
f_{SUBC}		f_{SUBC}	
f_{IH}	◄►	f_{MX}	see Table 5-8
f_{MAINC}	◄►	f_{SUBC}	see Table 5-9

Table 5-7. Maximum Number of Clocks Required in f_{MAINC} ↔f_{MAINC} (changing the division ratio), f_{SUBC} ↔f_{SUBC} (changing the division ratio)

Set Value Before Switchover	Set Value After Switchover	
	Clock A	Clock B
Clock A		1 + f_A/f_B clock
Clock B	1 + f_B/f_A clock	

Table 5-8. Maximum Number of Clocks Required in f_{IH} ↔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}≥f_{IH}		1 + f_{IH}/f_{MX} clock
	f_{MX}<f_{IH}		2f_{IH}/f_{MX} clock
1 (f_{MAIN} = f_{MX})	f_{MX}≥f_{IH}	2f_{MX}/f_{IH} clock	
	f_{MX}<f_{IH}	1 + f_{MX}/f_{IH} clock	

(Remarks are listed on the next page.)

Table 5-9. Maximum Number of Clocks Required in f_{MAINC} ↔f_{SUBC}

Set Value Before Switchover		Set Value After Switchover	
CSS		CSS	
		0 (f_{CLK} = f_{MAINC})	1 (f_{CLK} = f_{SUBC})
0 (f_{CLK} = f_{MAINC})	f_{MAINC}>f_{SUBC}		1 + 2f_{MAINC}/f_{SUBC} clock
1 (f_{CLK} = f_{SUBC})	f_{MAINC}>f_{SUBC}	2 + f_{SUBC}/f_{MAINC} clock	

Remarks 1. The number of clocks listed in Tables 5-7 to 5-9 is the number of CPU clocks before switchover.

2. Calculate the number of clocks in Tables 5-7 to 5-9 by removing the decimal portion.

Example When switching the main system clock from the internal high-speed oscillation clock to the high-speed system clock (@ oscillation with f_{IH} = 8 MHz, f_{MX} = 10 MHz)

1 + f_{IH}/f_{MX} = 1 + 8/10 = 1 + 0.8 = 1.8 → 2 clocks

5.6.8 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.

Table 5-10. Conditions Before the Clock Oscillation Is Stopped and Flag Settings

Clock	Conditions Before Clock Oscillation Is Stopped (External Clock Input Disabled)	Flag Settings of SFR Register
Internal high-speed oscillation clock	MCS = 1 or CLS = 1 (The CPU is operating on a clock other than the internal high-speed oscillation clock)	HIOSTOP = 1
X1 clock External main system clock	MCS = 0 or CLS = 1 (The CPU is operating on a clock other than the high-speed system clock)	MSTOP = 1
Subsystem clock	CLS = 0 (The CPU is operating on a clock other than the subsystem clock)	XTSTOP = 1
20 MHz internal high-speed oscillation clock	SELDSC = 0 (The main system clock is operating on a clock other than the 20 MHz internal high-speed oscillation clock.)	DSCON = 0

CHAPTER 6 TIMER ARRAY UNIT

The 78K0R/Kx3-A microcontrollers are provided with two timer array units. Time array unit 0 is provided with eight 16-bit timers and timer array unit 1 is provided with four 16-bit timers. Each 16-bit timer is called a channel and can be used as an independent timer. In addition, two or more “channels” can be used to create a high-accuracy timer.

Independent Operation Function	Combination Operation Function
• Interval timer • Square wave output • External event counter • Divider function • Input pulse interval measurement • Measurement of high-/low-level width of input signal	• PWM output • One-shot pulse output • Multiple PWM output

Channel 7 of timer array unit 0 can be used to realize LIN-bus reception processing in combination with UART3 of serial array unit 1.

Cautions 1. Channels 0 to 3 of timer array unit 1 can be used only as interval timers.
2. Channels 1 and 5 to 7 of timer array unit 0 cannot be used as frequency dividers.

Whether each channel of the timer array units 0 and 1 is provided with timer I/O pins is shown below.

Timer array unit m	Channel n	Input (TIpq) / output (TOpq)	Timer I/O Pins
0	0	Input	TI00/TO03/P31/RTCDIV/RTCCL/PCLBUZ1/INTP2
		Output	TO00/TI03/P30/RTC1HZ/INTP1
	1	I/O	TI01/TO01/P32/PCLBUZ0/INTP5
	2	Input	TI02/P52
		Output	TO02/P12/SO02/TxD2
	3	Input	TI03/TO00/P30/RTC1HZ/INTP1
		Output	TO03/TI00/P31/RTCDIV/RTCCL/PCLBUZ1/INTP2
	4	Input	TI04/P53
		Output	TO04/P13/SO10/TxD1
	5	I/O	TI05/TO05/P16/INTP10
	6	I/O	TI06/TO06/P34/INTP8
	7	I/O	TI07/TO07/P33/INTP3
1	0	–	–
	1		
	2		
	3		

6.1 Functions of Timer Array Unit

The timer array unit has the following functions.

6.1.1 Functions of each channel when it operates independently

Independent operation functions are those functions that can be used for any channel regardless of the operation mode of the other channel (for details, refer to **6.6.1 Overview of single-operation function and combination operation function**).

(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 INTTMpq is generated and a square wave with a duty factor of 50% is output from a timer output pin (TOpq).

(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 (TIpq) has reached a specific value.

(4) Divider function

A clock input from a timer input pin (TIpq) is divided and output from an output pin (TOpq).

(5) Input pulse interval measurement

Counting is started by the valid edge of a pulse signal input to a timer input pin (TIpq). The count value of the timer is captured at the valid edge of the next pulse. In this way, the interval of the input pulse can be measured.

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

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

Remark mn = 00 to 07, 10 to 13: Unit number + Channel number
pq = 00 to 07: Unit number + Channel number (only for channels provided with timer I/O pins)

6.1.2 Functions of each channel when it operates with another channel

Combination operation functions are those functions that are attained by using the master channel (mostly the reference timer that controls cycles) and the slave channels (timers that operate following the master channel) in combination (for details, refer to **6.6.1 Overview of single-operation function and combination operation function**).

(1) PWM (Pulse Width Modulator) output

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

(2) One-shot pulse output

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

(3) Multiple PWM (Pulse Width Modulator) output

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

6.1.3 LIN-bus supporting functions (channel 7 of timer array unit 0 only)

(1) Detection of wakeup signal

The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD3) of UART3 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 sync break field

The timer starts counting at the falling edge of a signal input to the serial data input pin (RxD3) of UART3 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 sync break field.

(3) Measurement of pulse width of sync field

After a sync break field is detected, the low-level width and high-level width of the signal input to the serial data input pin (RxD3) of UART3 are measured. From the bit interval of the sync field measured in this way, a baud rate is calculated.

6.2 Configuration of Timer Array Unit

The 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 inputs	TIpq pin, RxD3 pin (for LIN-bus)
Timer outputs	TOpq pin, 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 registers 0, 1 (TIS0, TIS1) • Timer output enable register 0 (TOE0) • Timer output register 0 (TO0) • Timer output level register 0 (TOL0) • Timer output mode register 0 (TOM0)
	<Registers of each channel> • Timer mode register mn (TMRmn) • Timer status register pq (TSRpq) • Input switch control register (ISC) (channel 7 of timer array unit 0 only) • Noise filter enable register 1 (NFEN1) • Port mode registers 1, 3, 5 (PM1, PM3, PM5) • Port registers 1, 3, 5 (P1, P3, P5)

Remark m = 0, 1
mn = 00 to 07, 10 to 13: Unit number + Channel number
pq = 00 to 07: Unit number + Channel number (only for channels provided with timer I/O pins)

Figures 6-1 and 6-2 show block diagrams.

Figure 6-1. Block Diagram of Timer Array Unit 0

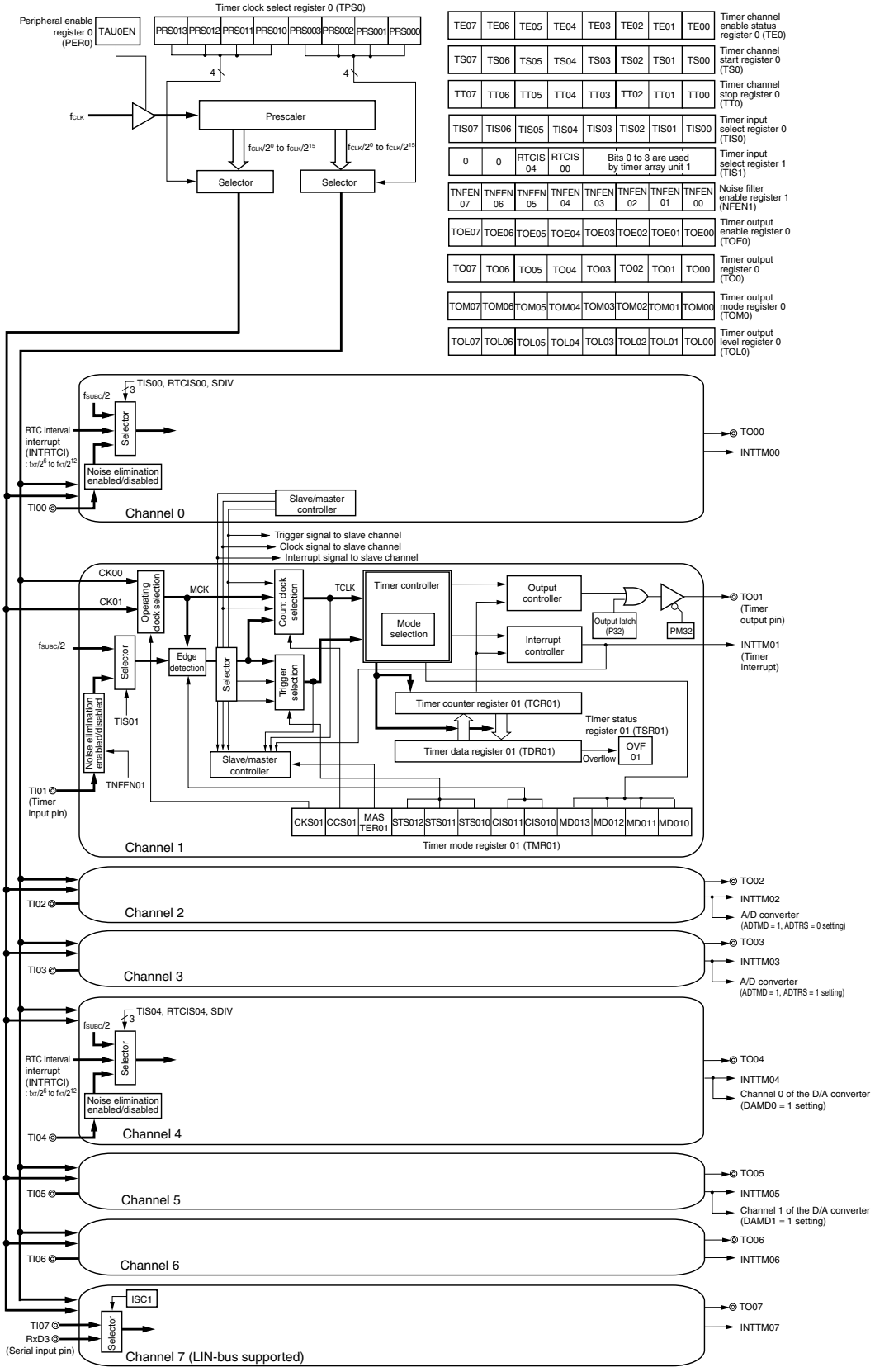
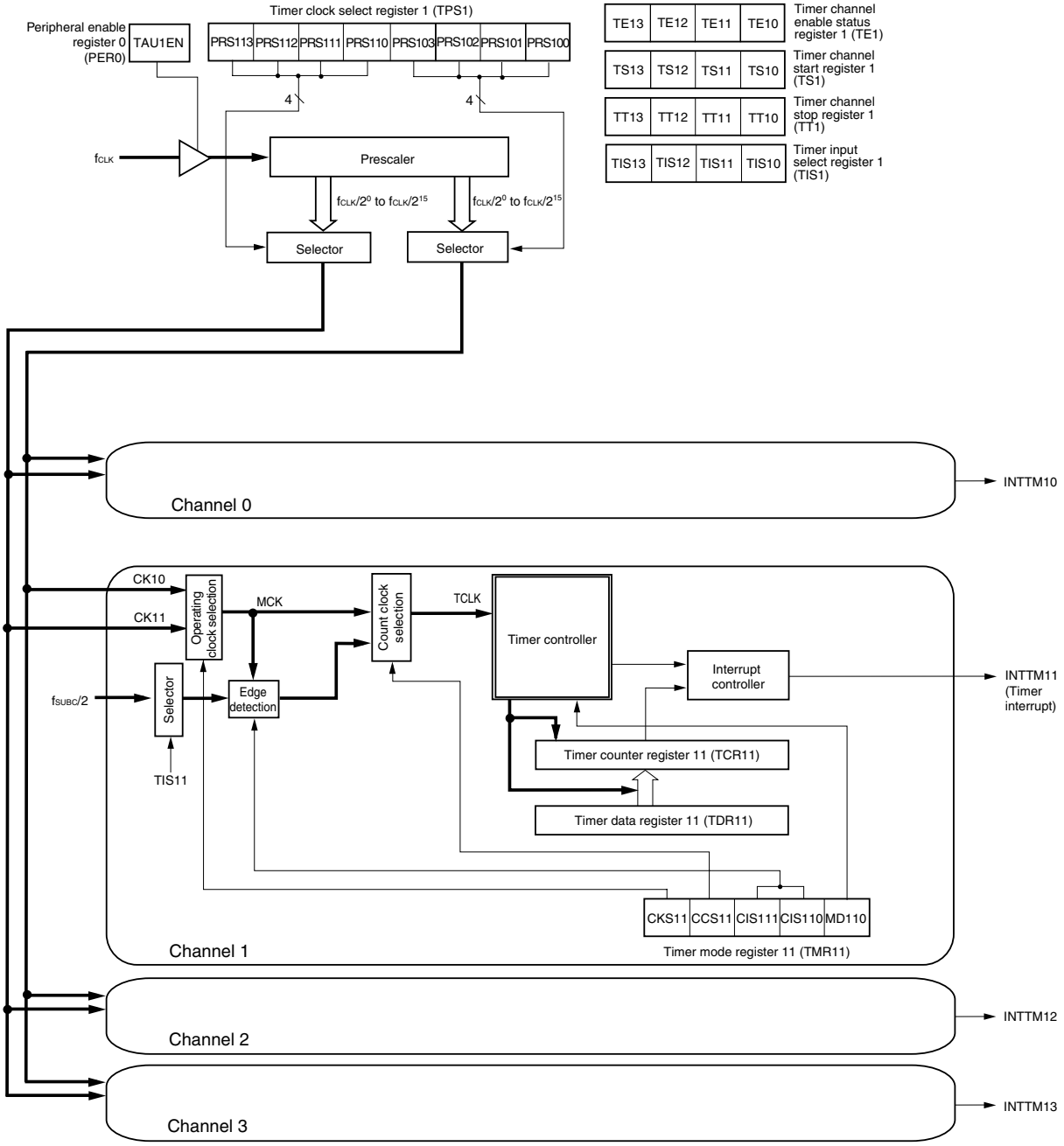


Figure 6-2. Block Diagram of Timer Array Unit 1



(1) Timer/counter register mn (TCRmn)

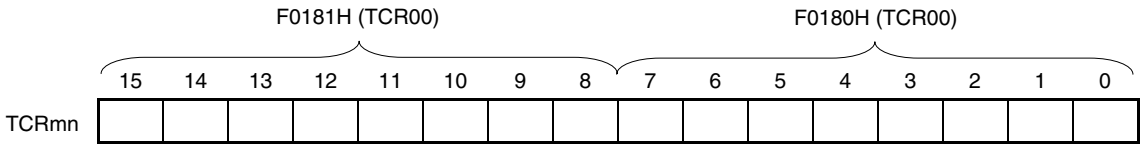
TCRmn is a 16-bit read-only register and is used to count clocks.

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

Whether the counter is incremented or decremented depends on the operation mode that is selected by the MDmn3 to MDmn0 bits of TMRmn.

Figure 6-3. Format of Timer/Counter Register mn (TCRmn)

Address: F0180H, F0181H (TCR00) to F018EH, F018FH (TCR07) After reset: FFFFH R
F01C0H, F01C1H (TCR10) to F01C6H, F01C7H (TCR13)



The count value can be read by reading TCRmn.

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

- When the reset signal is generated
- When the TAU0EN bit (TAU0) and TAU1EN bit (TAU1) of peripheral enable register 0 (PER0) is cleared

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
- When counting of the slave channel has been completed in the PWM output 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

Caution The count value is not captured to TDRmn even when TCRmn is read.

Remark mn: Unit number + Channel number
mn = 00 to 07, 10 to 13

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

Table 6-2. TCRmn Register Read Value in Various Operation Modes

Operation Mode	Count Mode	TCRmn Register Read Value[Note]			
		Operation mode change after reset	Operation mode change after count operation paused (TTmn = 1)	Operation restart after count operation paused (TTmn = 1)	During start trigger wait status after one count
Interval timer mode	Count down	FFFFH	Undefined	Stop value	–
Capture mode	Count up	0000H	Undefined	Stop value	–
Event counter mode	Count down	FFFFH	Undefined	Stop value	–
One-count mode	Count down	FFFFH	Undefined	Stop value	FFFFH
Capture & one-count mode	Count up	0000H	Undefined	Stop value	Capture value of TDRmn register + 1

Note The read values of the TCRmn register when TSmn has been set to "1" while TEmn = 0 are shown. The read value is held in the TCRmn register until the count operation starts.

Remark mn: Unit number + Channel number
mn = 00 to 07, 10 to 13

(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 TMRmn.

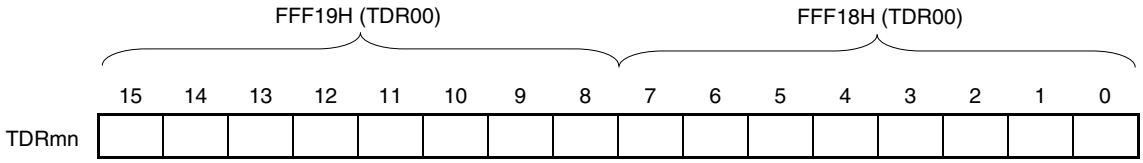
The value of TDRmn can be changed at any time.

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

Reset signal generation clears this register to 0000H.

Figure 6-4. Format of Timer Data Register mn (TDRmn)

Address: FFF18H, FFF19H (TDR00), FFF1AH, FFF1BH (TDR01), FFF64H, FFF65H (TDR02) to FFF6EH, FFF6FH (TDR07) FFF70H, FFF71H (TDR10) to FFF76H, FFF77H (TDR13) After reset: 0000H R/W



(i) When TDRmn is used as compare register

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

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

(ii) When TDRpq is used as capture register

The count value of TCRpq is captured to TDRpq when the capture trigger is input.

A valid edge of the TIpq pin can be selected as the capture trigger. This selection is made by TMRpq.

Remark mn = 00 to 07, 10 to 13: Unit number + Channel number
pq = 00 to 07: Unit number + Channel number (only for channels provided with timer I/O pins)

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 pq (TSRpq)
- Timer channel enable status register m (TEm)
- Timer channel start register m (TSm)
- Timer channel stop register m (TTm)
- Timer input select registers 0, 1 (TIS0, TIS1)
- Timer output enable register 0 (TOE0)
- Timer output register 0 (TO0)
- Timer output level register 0 (TOL0)
- Timer output mode register 0 (TOM0)
- Input switch control register (ISC)
- Noise filter enable register 1 (NFEN1)
- Port mode registers 1, 3, 5 (PM1, PM3, PM5)
- Port registers1, 3, 5 (P1, P3, P5)

Remark m = 0, 1
mn = 00 to 07, 10 to 13: Unit number + Channel number
pq = 00 to 07: Unit number + Channel number (only for channels provided with timer I/O pins)

(1) Peripheral enable register 0 (PER0)

PER0 is used to enable or disable use of each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

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

When the timer array unit 1 is used, be sure to set bit 1 (TAU1EN) of this register to 1.

PER0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Caution When setting the timer array unit, be sure to set TAUmEN to 1 first. If TAUmEN = 0, writing to a control register of the timer array unit is ignored, and all read values are default values.

Figure 6-5. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	DACEN	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

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

(2) Timer clock select register m (TPSm)

TPSm 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 TPSm, and CKm0 is selected by bits 3 to 0.

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

Rewriting of PRSm00 to PRSm03 bits: Possible only when all the channels set to CKSmn = 0 are in the operation stopped state (TEmn = 0)

Rewriting of PRSm10 to PRSm13 bits: Possible only when all the channels set to CKSmn = 1 are in the operation stopped state (TEmn = 0)

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

The lower 8 bits of TPSm can be set with an 8-bit memory manipulation instruction with TPSmL.

Reset signal generation clears this register to 0000H.

Remark m = 0, 1

mn = 00 to 07, 10 to 13: Unit number + Channel number

Figure 6-6. Format of Timer Clock Select Register m (TPSm)

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

F01DEH, F01DFH (TPS1)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPSm	0	0	0	0	0	0	0	0	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00

PRS mk3	PRS mk2	PRS mk1	PRS mk0	Selection of operation clock (CKmk) Notes 1,2				
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	312.5 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156.2 kHz	312.5 kHz	625 kHz
0	1	1	0	$f_{CLK}/2^6$	31.25 kHz	78.1 kHz	156.2 kHz	312.5 kHz
0	1	1	1	$f_{CLK}/2^7$	15.62 kHz	39.1 kHz	78.1 kHz	156.2 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.76 kHz	19.5 kHz	39.1 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.76 kHz	19.5 kHz
1	0	1	1	$f_{CLK}/2^{11}$	976 Hz	2.44 kHz	4.88 kHz	9.76 kHz
1	1	0	0	$f_{CLK}/2^{12}$	488 Hz	1.22 kHz	2.44 kHz	4.88 kHz
1	1	0	1	$f_{CLK}/2^{13}$	244 Hz	610 Hz	1.22 kHz	2.44 kHz
1	1	1	0	$f_{CLK}/2^{14}$	122 Hz	305 Hz	610 Hz	1.22 kHz
1	1	1	1	$f_{CLK}/2^{15}$	61 Hz	153 Hz	305 Hz	610 Hz

Notes 1. When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), stop the timer array unit (TT0 = 00FFH, TT1 = 000FH).

2. Only in the case of SDIV=0, CCSmn=1 and TISmn=1, continuously use of TAUm is allowed, even when changing CPU clock. However, the following limitation is existing.

- When changing CPU clock, source clock decrease/increase occurs as follows.

  Main clock → Subsystem clock (CSS = 0→1): −1 clock

  Subsystem clock → Main clock (CSS = 1→0): +1 clock

Caution Be sure to clear bits 15 to 8 to "0".

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

2. k = 0, 1

m = 0, 1

mn = 00 to 07, 10 to 13

(3) Timer mode register mn (TMRmn)

TMRmn sets an operation mode of channel n of timer array unit m. It is used to select an operation clock (MCK), a count clock, whether the timer operates as the master or a slave, a start trigger and a capture trigger, the valid edge of the timer input, and an operation mode (interval, capture, event counter, one-count, or capture & one-count).

Rewriting TMRmn is prohibited when the register is in operation (when TEm = 1). However, bits 7 and 6 (CISmn1, CISmn0) can be rewritten even while the register is operating with some functions (when TEm = 1) (for details, see **6.7 Operation of Timer Array Unit as Independent Channel** and **6.8 Operation of Plural Channels of Timer Array Unit**).

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

Reset signal generation clears this register to 0000H.

Figure 6-7. Format of Timer Mode Register mn (TMRmn) (1/4)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W
F01C8H, F01C9H (TMR10) to F01CEH, F01CFH (TMR13)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n	CKS 0n	0	0	CCS 0n	MAST ER0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0
TMR1n	CKS 1n	0	0	CCS 1n	0	0	0	0	CIS 1n1	CIS 1n0	0	0	0	0	0	MD 1n0

CKS mn	Selection of operation clock (MCK) of channel n
0	Operation clock CKm0 set by TPSm register
1	Operation clock CKm1 set by TPSm register
Operation clock MCK is used by the edge detector. A count clock (TCLK) and a sampling clock are generated depending on the setting of the CCSmn bit.	

CCS mn	Selection of count clock (TCLK) of channel n
0	Operation clock MCK specified by CKSmn bit
1	Valid edge of input signal input from TIpq pin, $f_{SUB}/2$, $f_{SUB}/4$, or INTRTC1 (the timer input used with channel x is selected by using TISm register).
Count clock TCLK is used for the timer/counter, output controller, and interrupt controller. If CCSmn = 1, use the count clock under the following condition. • The frequency of the operating clock selected by using CKSmn ≥ The frequency of the clock selected by using TISmn × 2	

Cautions 1. Be sure to clear bits 14, 13, 5, and 4 of TMR0n to "0".

2. Be sure to clear bits 14, 13, 11 to 8, and 5 to 1 of TMR1n to "0".

Remark m = 0, 1, n = 0 to 7
mn = 00 to 07, 10 to 13: Unit number + Channel number
pq = 00 to 07: Unit number + Channel number (only for channels provided with timer I/O pins)

Figure 6-7. Format of Timer Mode Register mn (TMRmn) (2/4)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W
F01C8H, F01C9H (TMR10) to F01CEH, F01CFH (TMR13)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n	CKS 0n	0	0	CCS 0n	MAST ER0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

TMR1n	CKS 1n	0	0	CCS 1n	0	0	0	0	CIS 1n1	CIS 1n0	0	0	0	0	0	MD 1n0

MASTER mn	Selection of operation in single-operation function or as slave channel in combination-operation function /operation as master channel in combination-operation function of channel n
0	Operates in single-operation function or as slave channel in combination-operation function.
1	Operates as master channel with combination operation function.
Only the even channel can be set as a master channel (MASTERmn = 1). Be sure to use the odd channel as a slave channel (MASTERmn = 0). Clear MASTERmn to 0 for a channel that is used with the independent operation function.	

STS mn2	STS mn1	STS mn0	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 TIpq pin input signal, $f_{SUB}/2$, $f_{SUB}/4$, or INTRTC1 is used as both the start trigger and capture trigger.
0	1	0	Both the edges of TIpq pin input signal, $f_{SUB}/2$, $f_{SUB}/4$, or INTRTC1 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 combination operation function).
Other than above			Setting prohibited

CIS mn1	CIS mn0	Selection of valid edge of TIpq pin input signal , $f_{SUB}/2$, $f_{SUB}/4$, or INTRTC1 (the timer input used with channel x is selected by using TISm register).
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.		

Cautions 1. Be sure to clear bits 14, 13, 5, and 4 of TMR0n to "0".

2. Be sure to clear bits 14, 13, 11 to 8, and 5 to 1 of TMR1n to "0".

Remark m = 0, 1, n = 0 to 7
mn = 00 to 07, 10 to 13: Unit number + Channel number
pq = 00 to 07: Unit number + Channel number (only for channels provided with timer I/O pins)

Figure 6-7. Format of Timer Mode Register mn (TMRmn) (3/4)

Address: F0190H, F0191H (TMR00) to F019EH, F019FH (TMR07) After reset: 0000H R/W
F01C8H, F01C9H (TMR10) to F01CEH, F01CFH (TMR13)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0n	CKS 0n	0	0	CCS 0n	MAST ER0n	STS 0n2	STS 0n1	STS 0n0	CIS 0n1	CIS 0n0	0	0	MD 0n3	MD 0n2	MD 0n1	MD 0n0

TMR1n	CKS 1n	0	0	CCS 1n	0	0	0	0	CIS 1n1	CIS 1n0	0	0	0	0	0	MD 1n0

MD mn3	MD mn2	MD mn1	MD mn0	Operation mode of channel n	Count operation of TCR	Independent operation
0	0	0	1/0	Interval timer mode	Counting down	Possible
0	1	0	1/0	Capture mode	Counting up	Possible
0	1	1	0	Event counter mode	Counting down	Possible
1	0	0	1/0	One-count mode	Counting down	Impossible
1	1	0	0	Capture & one-count mode	Counting up	Possible
Other than above				Setting prohibited		
The operation of MDmn0 bits varies depending on each operation mode (see following table).						

Cautions 1. Be sure to clear bits 14, 13, 5, and 4 of TMR0n to "0".

2. Be sure to clear bits 14, 13, 11 to 8, and 5 to 1 of TMR1n to "0".

Remark m = 0, 1, n = 0 to 7
mn = 00 to 07, 10 to 13: Unit number + Channel number

Figure 6-7. Format of Timer Mode Register mn (TMRmn) (4/4)

Operation mode (Value set by the MDmn3 to MDmn1 bits (see table above))	MD mn0	Setting of starting counting and interrupt
• Interval timer mode (0, 0, 0)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• Capture mode (0, 1, 0)	1	Timer interrupt is generated when counting is started (timer output also changes).
• Event counter mode (0, 1, 1)	0	Timer interrupt is not generated when counting is started (timer output does not change, either).
• One-count mode (1, 0, 0)	0	Start trigger is invalid during counting operation. At that time, interrupt is not generated, either.
	1	Start trigger is valid during counting operation[Note]. At that time, interrupt is also 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, either.
Other than above		Setting prohibited

Note If the start trigger (TSmn = 1) is issued during operation, the counter is cleared, an interrupt is generated, and recounting is started.

Caution Channels 0 to 3 of timer array unit 1 can be used only as interval timers.

Remark mn = 00 to 07, 10 to 13: Unit number + Channel number

(4) Timer status register pq (TSRpq)

TSRpq indicates the overflow status of the counter of channel q of timer array unit 0.

TSRpq is valid only in the capture mode (MDpq3 to MDpq1 = 010B) and capture & one-count mode (MDpq3 to MDpq1 = 110B). It will not be set in any other mode. See Table 6-3 for the operation of the OVFpq bit in each operation mode and set/clear conditions.

TSRpq can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of TSRpq can be set with an 8-bit memory manipulation instruction with TSRpqL.

Reset signal generation clears this register to 0000H.

Figure 6-8. Format of Timer Status Register pq (TSRpq)

Address: F01A0H, F01A1H (TSR00) to F01AEH, F01AFH (TSR07) After reset: 0000H R

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

OVFpq	Counter overflow status of channel q
0	Overflow does not occur.
1	Overflow occurs.
When OVFpq = 1, this flag is cleared (OVFpq = 0) when the next value is captured without overflow.	

Table 6-3. OVFpq Bit Operation and Set/Clear Conditions in Each Operation Mode

Timer operation mode	OVFpq	Set/clear conditions
• Capture mode	clear	When no overflow has occurred upon capturing
• Capture & one-count mode	set	When an overflow has occurred upon capturing
• Interval timer mode • Event counter mode • One-count mode	clear	– (Use prohibited, not set/cleared)
	set	

Remarks 1. The OVFpq bit does not change immediately after the counter has overflowed, but changes upon the subsequent capture.

2. pq = 00 to 07: Unit number + Channel number (only for channels provided with timer I/O pins)

(5) Timer channel enable status register m (TEm)

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

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

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

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

Reset signal generation clears this register to 0000H.

Figure 6-9. Format of Timer Channel Enable Status Register m (TEm)

Address: F01B0H, F01B1H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TE0	0	0	0	0	0	0	0	0	TE07	TE06	TE05	TE04	TE03	TE02	TE01	TE00

Address: F01D8H, F01D9H After reset: 0000H R

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TE1	0	0	0	0	0	0	0	0	0	0	0	0	TE13	TE12	TE11	TE10

TE mn	Indication of operation enable/stop status of channel n
0	Operation is stopped.
1	Operation is enabled.

Remark m = 0, 1

mn = 00 to 07, 10 to 13: Unit number + Channel number

(6) Timer channel start register m (TSm)

TSm is a trigger register that is used to clear a timer counter (TCRmn) and start the counting operation of each channel.

When a bit (TSmn) of this register is set to 1, the corresponding bit (TEmn) of timer channel enable status register m (TEm) is set to 1. TSmn is a trigger bit and cleared immediately when TEmn = 1.

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

The lower 8 bits of TSm can be set with a 1-bit or 8-bit memory manipulation instruction with TSmL.

Reset signal generation clears this register to 0000H.

Figure 6-10. Format of Timer Channel Start Register m (TSm)

Address: F01B2H, F01B3H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS0	0	0	0	0	0	0	0	0	TS07	TS06	TS05	TS04	TS03	TS02	TS01	TS00

Address: F01DAH, F01DBH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS1	0	0	0	0	0	0	0	0	0	0	0	0	TS13	TS12	TS11	TS10

TSmn	Operation enable (start) trigger of channel n
0	No trigger operation
1	TEmn is set to 1 and the count operation becomes enabled. The TCRmn count operation start in the count operation enabled state varies depending on each operation mode (see **Table 6-4**).

Caution Be sure to clear bits 15 to 8 of TS0 and bits 15 to 4 of TS1 to "0".

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

2. m = 0, 1

mn = 00 to 07, 10 to 13: Unit number + Channel number

Table 6-4. Operations from Count Operation Enabled State to 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 TDRmn to TCRmn and the subsequent count clock performs count down operation (see **6.3 (6) (a) Start timing in interval timer mode**).
• Event counter mode	Writing 1 to TSmn bit loads the value of TDRmn to TCRmn. The subsequent count clock performs count down operation. The external trigger detection selected by STSmn2 to STSmn0 bits in the TMRmn register does not start count operation (see **6.3 (6) (b) Start timing in 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 TCRmn and the subsequent count clock performs count up operation (see **6.3 (6) (c) Start timing in capture mode**).
• One-count mode	When TEmn = 0, writing 1 to TSmn bit sets the start trigger wait state. No operation is carried out from start trigger detection until count clock generation. The first count clock loads the value of TDRmn to TCRmn and the subsequent count clock performs count down operation (see **6.3 (6) (d) Start timing in one-count mode**).
• Capture & one-count mode	When TEmn = 0, writing 1 to TSmn bit sets the start trigger wait state. No operation is carried out from start trigger detection until count clock generation. The first count clock loads 0000H to TCRmn and the subsequent count clock performs count up operation (see **6.3 (6) (e) Start timing in capture & one-count mode**).

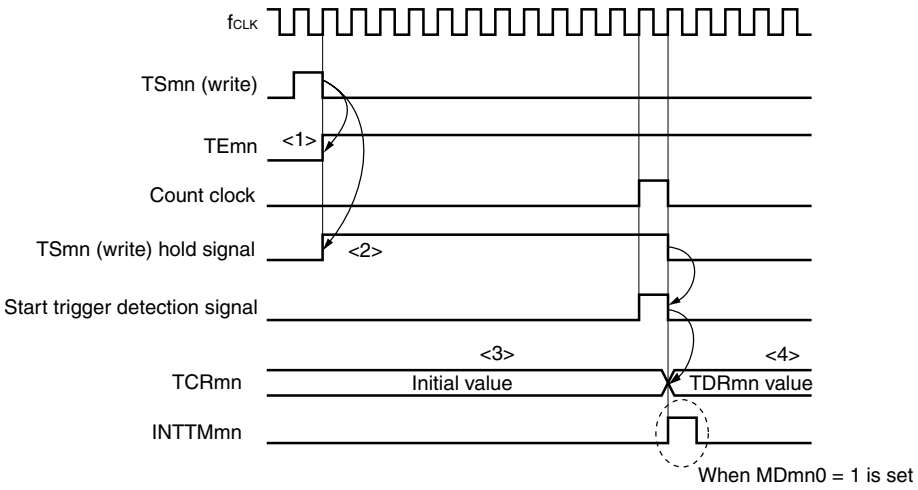
Caution Channels 0 to 3 of timer array unit 1 can be set only to the interval mode.

(a) Start timing in interval timer mode

<1> Writing 1 to TSmn sets TEmn = 1
<2> The write data to TSmn is held until count clock generation.
<3> TCRmn holds the initial value until count clock generation.
<4> On generation of count clock, the “TDRmn value” is loaded to TCRmn and count starts.

Remark mn = 00 to 07, 10 to 13: Unit number + Channel number

Figure 6-11. Start Timing (In Interval Timer Mode)



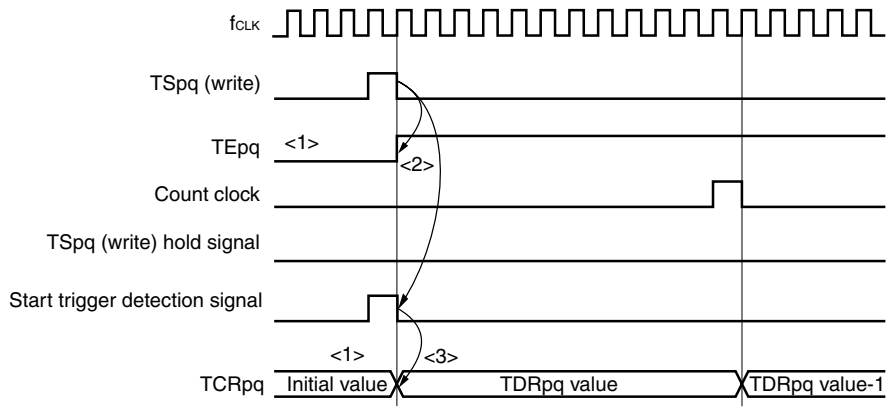
Caution In the first cycle operation of count clock after writing TSmn, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDmn0 = 1.

Remark mn = 00 to 07, 10 to 13: Unit number + Channel number

(b) Start timing in event counter mode

<1> While TEpq is set to 0, TCRpq holds the initial value.
<2> Writing 1 to TSpq sets 1 to TEpq.
<3> As soon as 1 has been written to TSpq and 1 has been set to TEpq, the "TDRpq value" is loaded to TCRpq to start counting.
<4> After that, the TCRpq value is counted down according to the count clock.

Figure 6-12. Start Timing (In Event Counter Mode)

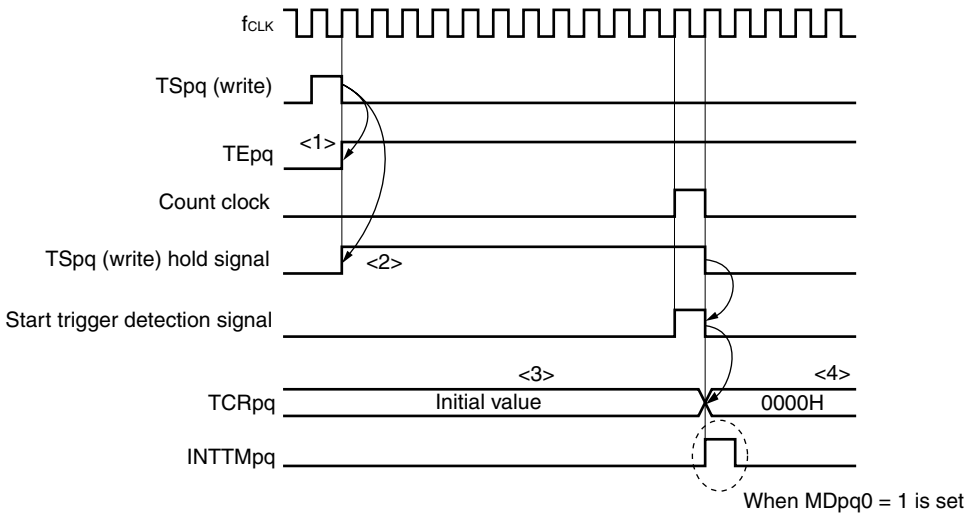


Remark pq = 00 to 07: Unit number + Channel number (only for channels provided with timer I/O pins)

(c) Start timing in capture mode

<1> Writing 1 to TSpq sets TEpq = 1
<2> The write data to TSpq is held until count clock generation.
<3> TCRpq holds the initial value until count clock generation.
<4> On generation of count clock, 0000H is loaded to TCRpq and count starts.

Figure 6-13. Start Timing (In Capture Mode)



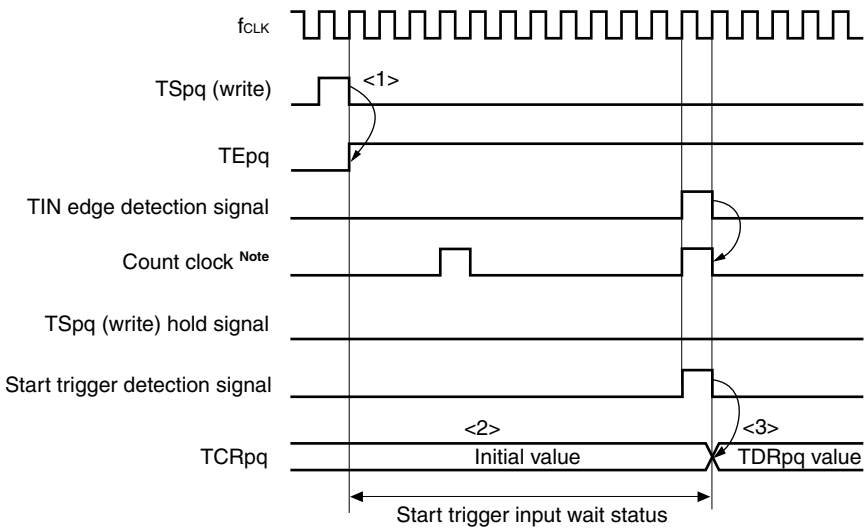
Caution In the first cycle operation of count clock after writing TSpq, an error at a maximum of one clock is generated since count start delays until count clock has been generated. When the information on count start timing is necessary, an interrupt can be generated at count start by setting MDpq0 = 1.

Remark pq = 00 to 07: Unit number + Channel number (only for channels provided with timer I/O pins)

(d) Start timing in one-count mode

<1> Writing 1 to TSpq sets TEpq = 1
<2> Enters the start trigger input wait status, and TCRpq holds the initial value.
<3> On start trigger detection, the "TDRpq value" is loaded to TCRpq and count starts.

Figure 6-14. Start Timing (In One-count Mode)



Note When the one-count mode is set, the operation clock (MCK) is selected as count clock (CCSpq = 0).

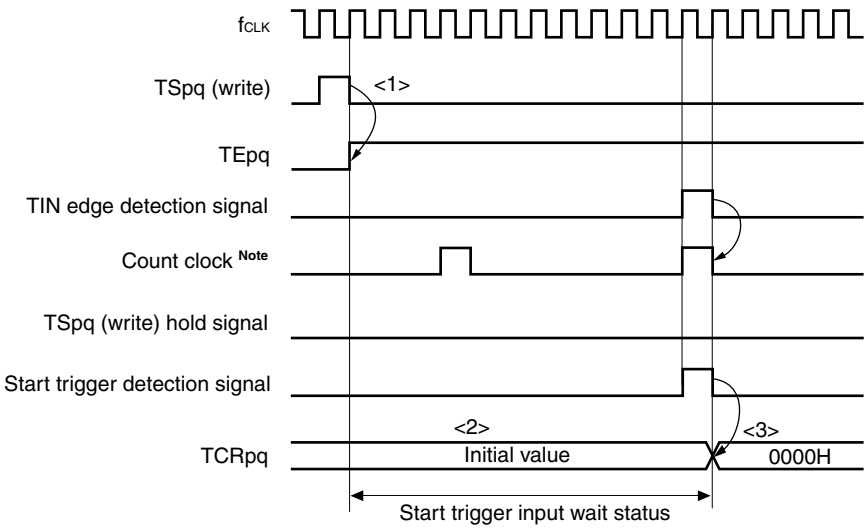
Caution An input signal sampling error is generated since operation starts upon start trigger detection (The error is one count clock when TIpq is used).

Remark pq = 00 to 07: Unit number + Channel number (only for channels provided with timer I/O pins)

(e) Start timing in capture & one-count mode

<1> Writing 1 to TSpq sets TEpq = 1
<2> Enters the start trigger input wait status, and TCRpq holds the initial value.
<3> On start trigger detection, 0000H is loaded to TCRpq and count starts.

Figure 6-15. Start Timing (In Capture & One-count Mode)



Note When the capture & one-count mode is set, the operation clock (MCK) is selected as count clock (CCSpq = 0).

Caution An input signal sampling error is generated since operation starts upon start trigger detection (The error is one count clock when TIpq is used).

(7) Timer channel stop register m (TTm)

TTm is a trigger register that is used to clear a timer counter (TCRmn) and start the counting operation of each channel.

When a bit (TTmn) of this register is set to 1, the corresponding bit (TEmn) of timer channel enable status register 0 (TEm) is cleared to 0. TTmn is a trigger bit and cleared to 0 immediately when TEmn = 0.

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

The lower 8 bits of TTm can be set with a 1-bit or 8-bit memory manipulation instruction with TTmL.

Reset signal generation clears this register to 0000H.

Remark m = 0, 1
mn = 00 to 07: Unit number + Channel number,
pq = 00 to 07: Unit number + Channel number (only for channels provided with timer I/O pins)

Figure 6-16. Format of Timer Channel Stop Register m (TTm)

Address: F01B4H, F01B5H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TT0	0	0	0	0	0	0	0	0	TT07	TT06	TT05	TT04	TT03	TT02	TT01	TT00

Address: F01DCH, F01DDH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TT1	0	0	0	0	0	0	0	0	0	0	0	0	TT13	TT12	TT11	TT10

TTmn	Operation stop trigger of channel n
0	No trigger operation
1	Operation is stopped (stop trigger is generated).

Caution Be sure to clear bits 15 to 8 of TT0 and bits 15 to 4 of TT1 to "0".

Remarks 1. When the TTm register is read, 0 is always read.

2. m = 0, 1

mn = 00 to 07, 10 to 13: Unit number + Channel number

(8) Timer input select registers 0, 1 (TIS0, TIS1)

TIS0 and TIS1 use can be set to the input signal of a timer input pin (TIpq), half the frequency of the subsystem clock ($f_{SUB}/2$), one fourth the frequency of the subsystem clock ($f_{SUB}/4$), or an RTC interval interrupt (INTRTCI) as the timer input. The timer input can be selected for each channel.

TIS0 and TIS1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Remark pq = 00 to 07: Unit number + Channel number (only for channels provided with timer I/O pins)

Figure 6-17. Format of Timer Input Select Registers 0, 1 (TIS0, TIS1)

Address: FFF3EH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS0	TIS07	TIS06	TIS05	TIS04	TIS03	TIS02	TIS01	TIS00

Address: FFF3FH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
TIS1	0	0	RTCIS04	RTCIS00	0	0	0	0

• Channels 1 to 3 and 5 to 7 of timer array unit 0

TISpq	SDIV	Selection of Timer input used with channel (pq = 01, 02, 03, 05, 06, 07)
0	×	Input signal of timer input pin (TIpq)
1	0	$f_{SUB}/2$
	1	$f_{SUB}/4$

• Channels 0 and 4 of timer array unit 0

TISpq	RTCISpq	SDIV	Selection of Timer input used with channel (pq = 00, 04)
0	×	×	Input signal of timer input pin (TIpq)
1	0	0	$f_{SUB}/2$
		1	$f_{SUB}/4$
	1	0	RTC Interval interrupt (INTRTCI)
		1	Setting prohibited

• Channels 0 to 3 of timer array unit 1

SDIV	Selection of Timer input used with channel
0	$f_{SUB}/2$
1	$f_{SUB}/4$

Caution When the LIN-bus communication function is used, select the input signal of the RxD3 pin by setting ISC1 = 1 (bit 1 of the input switch control register (ISC)) and TIS07 = 0.

Remarks 1. pq = 00 to 07: Unit number + Channel number (only for channels provided with timer I/O pins)

2. ×: don't care

3. f_{SUB}: Subsystem select clock

4. SDIV: Bit 3 of the system clock control register (CKC)

(9) Timer output enable register 0 (TOE0)

TOE0 is used to enable or disable timer output of each channel of timer array unit 0.

Channel q for which timer output has been enabled becomes unable to rewrite the value of the TO0q bit of the timer output register (TO0) 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 (TO0q).

TOE0 can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TOE0 can be set with a 1-bit or 8-bit memory manipulation instruction with TOE0L.

Reset signal generation clears this register to 0000H.

Figure 6-18. Format of Timer Output Enable Register 0 (TOE0)

Address: F01BAH, F01BBH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOE0	0	0	0	0	0	0	0	0	TOE07	TOE06	TOE05	TOE04	TOE03	TOE02	TOE01	TOE00

TOE0q	Timer output enable/disable of channel q of timer array unit 0
0	The TO0q operation stopped by count operation (timer channel output bit). Writing to the TO0q bit is enabled. The TO0q pin functions as data output, and it outputs the level set to the TO0q bit. The output level of the TO0q pin can be manipulated by software.
1	The TO0q operation enabled by count operation (timer channel output bit). Writing to the TO0q bit is disabled (writing is ignored). The TO0q pin functions as timer output, and the TOE0q is set or reset depending on the timer operation. The TO0q pin outputs the square-wave or PWM depending on the timer operation.

Caution Be sure to clear bits 15 to 8 of TOE0 to "0".

Remark q = 0 to 7

(10) Timer output register 0 (TO0)

TO0 is a buffer register of timer output of each channel of timer array unit 0.

The value of each bit in this register is output from the timer output pin (TO0q) of each channel.

This register can be rewritten by software only when timer output is disabled (TOE0q = 0). When timer output is enabled (TOE0q = 1), rewriting this register by software is ignored, and the value is changed only by the timer operation.

To use the P30/TO00, P32/TO01, P12/TO02, P31/TO03, P13/TO04, P16/TO05, P34/TO06, P33/TO07 pins as a port function pin, set the corresponding TO0q bit to “0”.

TO0 can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TO0 can be set with an 8-bit memory manipulation instruction with TO0L.

Reset signal generation clears this register to 0000H.

Figure 6-19. Format of Timer Output Register 0 (TO0)

Address: F01B8H, F01B9H After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO0	0	0	0	0	0	0	0	0	TO07	TO06	TO05	TO04	TO03	TO02	TO01	TO00

TO0q	Timer output of channel q of timer array unit 0
0	Timer output value is “0”.
1	Timer output value is “1”.

Caution Be sure to clear bits 15 to 8 of TO0 to “0”.

Remark q = 0 to 7

(11) Timer output level register 0 (TOL0)

TOL0 is a register that controls the timer output level of each channel of timer array unit 0.

The setting of the inverted output of channel q by this register is reflected at the timing of set or reset of the timer output signal while the timer output is enabled (TOE0q = 1) in the combination operation mode (TOM0q = 1). In the toggle mode (TOM0q = 0), this register setting is invalid.

TOL0 can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TOL0 can be set with an 8-bit memory manipulation instruction with TOL0L.

Reset signal generation clears this register to 0000H.

Figure 6-20. Format of Timer Output Level Register 0 (TOL0)

Address: F01BCH, F01BDH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOL0	0	0	0	0	0	0	0	0	TOL07	TOL06	TOL05	TOL04	TOL03	TOL02	TOL01	TOL00

TOL0q	Control of timer output level of channel q of timer array unit 0
0	Positive logic output (active-high)
1	Inverted output (active-low)

Caution Be sure to clear bits 15 to 8 of TOL0 to "0".

Remarks 1. If the value of this register is rewritten during timer operation, the timer output is inverted when the timer output signal changes next, instead of immediately after the register value is rewritten.

2. q = 0 to 7

(12) Timer output mode register 0 (TOM0)

TOM0 is used to control the timer output mode of each channel of timer array unit 0.

When a channel is used for the combination operation function (PWM output, one-shot pulse output, or multiple PWM output), set the corresponding bit of the slave channel to 1.

The setting of each channel q by this register is reflected at the timing when the timer output signal is set or reset while the timer output is enabled (TOE0q = 1).

TOM0 can be set by a 16-bit memory manipulation instruction.

The lower 8 bits of TOM0 can be set with an 8-bit memory manipulation instruction with TOM0L.

Reset signal generation clears this register to 0000H.

Figure 6-21. Format of Timer Output Mode Register 0 (TOM0)

Address: F01BEH, F01BFH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOM0	0	0	0	0	0	0	0	0	TOM 07	TOM 06	TOM 05	TOM 04	TOM 03	TOM 02	TOM 01	TOM 00

TOM 0q	Control of timer output mode of channel q of timer array unit 0
0	Toggle mode (to produce toggle output by timer interrupt request signal (INTTM0q))
1	Combination operation mode (set by the timer interrupt request signal (INITTM0q) of the master channel, and reset by the timer interrupt request signal (INITTM0r) of the slave channel)

Caution Be sure to clear bits 15 to 8 of TOM0 to "0".

Remark q = 0 to 7 (q = 0, 2, 4, 6 for master channel)

q < r ≤ 7 (where r is a consecutive integer greater than q)

(13) Input switch control register (ISC)

ISC is used to implement LIN-bus communication operation with channel 7 of timer array unit 0 in association with serial array unit 1.

When bit 1 of this register is set to 1, the input signal of the serial data input pin (RxD3) is selected as a timer input signal.

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

Reset signal generation clears this register to 00H.

Figure 6-22. Format of Input Switch Control Register (ISC)

Address: FFF3CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	ISC4	ISC3	ISC2	ISC1	ISC0

ISC1	Switching channel 7 input of timer array unit 0
0	Uses the input signal of the TI07 pin as a timer input (normal operation).
1	Input signal of RxD3 pin is used as timer input (wakeup signal detection).

Caution Be sure to clear bits 5 to 7 to "0".

Remarks 1. When the LIN-bus communication function is used, select the input signal of the RxD3 pin by setting ISC1 = 1 and TIS07 = 0.

2. Bits 0 and 2 to 4 of ISC are not used with TAU.

(14) Noise filter enable register 1 (NFEN1)

NFEN1 is used to set whether the noise filter can be used for the timer input signal to each channel.

Enable the noise filter by setting the corresponding bits to 1 on the pins in need of noise removal.

When the noise filter is ON, match detection and synchronization of the 2 clocks is performed with the CPU/peripheral hardware clock (f_{CLK}). When the noise filter is OFF, only synchronization is performed with the CPU/peripheral hardware clock (f_{CLK}).

NFEN1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 6-23. Format of Noise Filter Enable Register 1 (NFEN1)

Address: F0061H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN1	TNFEN07	TNFEN06	TNFEN05	TNFEN04	TNFEN03	TNFEN02	TNFEN01	TNFEN00

TNFEN07	Enable/disable using noise filter of TI07/TO07/P33/INTP3 pin or RxD3/P50 pin input signal[Note]
0	Noise filter OFF
1	Noise filter ON

TNFEN06	Enable/disable using noise filter of TI06/TO06/P34/INTP8 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN05	Enable/disable using noise filter of TI05/TO05/P16/INTP10 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN04	Enable/disable using noise filter of TI04/P53 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN03	Enable/disable using noise filter of TI03/TO03/P30/RTC1HZ/INTP1 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN02	Enable/disable using noise filter of TI02/P52 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN01	Enable/disable using noise filter of TI01/TO01/P32/INTP5/PCLBUZ0 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN00	Enable/disable using noise filter of TI00/TO03/P31/RTCDIV/RTCCL/PCLBUZ1/INTP2 pin input signal
0	Noise filter OFF
1	Noise filter ON

Note The applicable pin can be switched by setting ISC1 of the ISC register.

ISC1 = 0: Whether or not to use the noise filter of TI07 pin can be selected.
ISC1 = 1: Whether or not to use the noise filter of RxD3 pin can be selected.

(15) Port mode registers 1, 3, 5 (PM1, PM3, PM5)

These registers set input/output of ports 1, 3, and 5 in 1-bit units.

When using the P30/TO00/TI03/RTC1HZ/INTP1, P32/TO01/TI01/INTP5/PCLBUZ0, P12/TO02/SO02/TxD2, P31/TO03/TI00/RTCDIV/RTCCL/PCLBUZ1/INTP2, P13/TO04/SO10/TxD1, P16/TO05/TI05/INTP10, P34/TO06/TI06/INTP8, and P33/TO07/TI07/INTP3 pins for timer output, set PM30, PM32, PM12, PM31, PM13, PM16, PM34, and PM33 and the output latches of P30, P32, P12, P31, P13, P16, P34, and P33 to 0.

When using the P31/TI00/TO03/RTCDIV/RTCCL/PCLBUZ1/INTP2, P32/TI01/TO01/INTP5/PCLBUZ0, P52/TI02, P30/TI03/TO00/RTC1HZ/INTP1, P53/TI04, P16/TI05/TO05/INTP10, P34/TI06/TO06/INTP8, and P33/TI07/TO07/INTP3 pins for timer input, set PM31, PM32, PM52, PM30, PM53, PM16, PM34, and PM33 to 1. At this time, the output latches of P31, P32, P52, P30, P53, P16, P34, and P33 may be 0 or 1.

PM1, PM3, and PM5 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 6-24. Format of Port Mode Registers 1, 3, 5 (PM1, PM3, PM5)

Address: FFF21H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM1	1	PM16	PM15	PM14	PM13	PM12	PM11	PM10

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	PM34	PM33	PM32	PM31	PM30

Address: FFF25H After reset: FFH R/W

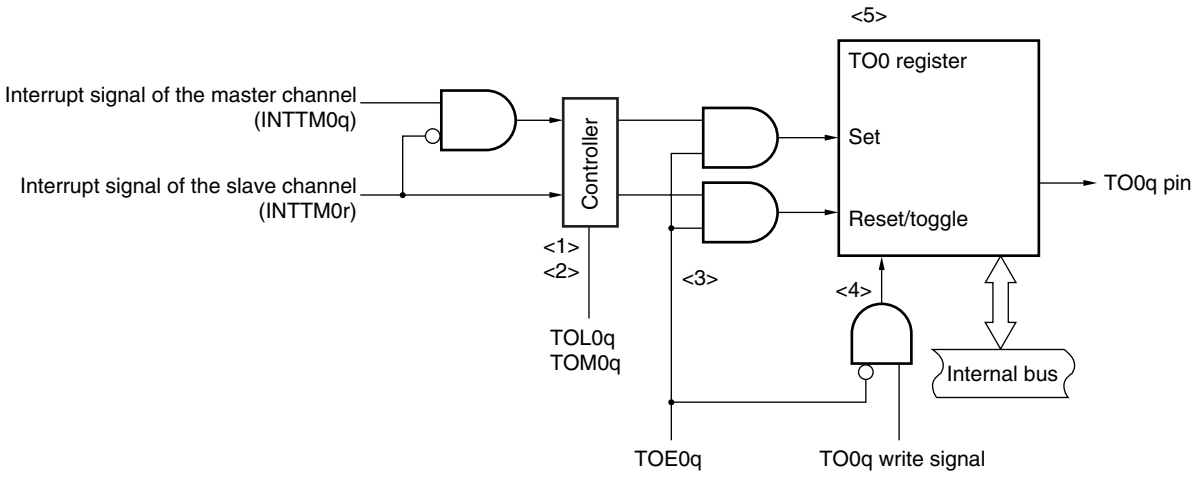
Symbol	7	6	5	4	3	2	1	0
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50

PMmn	Pmn pin I/O mode selection (m = 1, 3, 5; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

6.4 Channel Output of Timer Array Unit 0 (TO0q pin) Control

6.4.1 TO0q pin output circuit configuration

Figure 6-25. Output Circuit Configuration



The following describes the TO0q pin output circuit.

<1> When TOM0q = 0 (toggle mode), the set value of the TOL0 register is ignored and only INTTM0r (slave channel timer interrupt) is transmitted to the TO0 register.

<2> When TOM0q = 1 (combination operation mode), both INTTM0q (master channel timer interrupt) and INTTM0r (slave channel timer interrupt) are transmitted to the TO0 register.
At this time, the TOL0 register becomes valid and the signals are controlled as follows:

When TOL0q = 0: Forward operation (INTTM0q → set, INTTM0r → reset)
When TOL0q = 1: Reverse operation (INTTM0q → reset, INTTM0r → set)

When INTTM0q and INTTM0r are simultaneously generated, (0% output of PWM), INTTM0r (reset signal) takes priority, and INTTM0q (set signal) is masked.

<3> When TOE0q = 1, INTTM0q (master channel timer interrupt) and INTTM0r (slave channel timer interrupt) are transmitted to the TO0q register. Writing to the TO0 register (TO0q write signal) becomes invalid.
When TOE0q = 1, the TO0q pin output never changes with signals other than interrupt signals.
To initialize the TO0q pin output level, it is necessary to set TOE0q = 0 and to write a value to TO0q.

<4> When TOE0q = 0, writing to TO0q bit to the target channel (TO0q signal) becomes valid. When TOE0q = 0, neither INTTM0q (master channel timer interrupt) nor INTTM0r (slave channel timer interrupt) is transmitted to TO0q register.

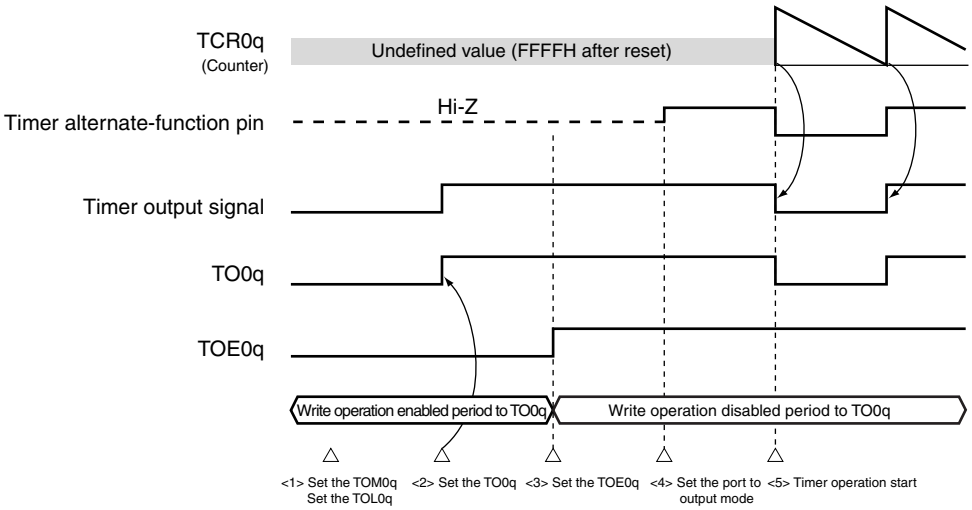
<5> The TO0 register can always be read, and the TO0q pin output level can be checked.

Remark q = 0 to 7 (q = 0, 2, 4, 6 for master channel)
q < r ≤ 7 (where r is a consecutive integer greater than q)

6.4.2 TOpq Pin Output Setting

The following figure shows the procedure and status transition of TO0q output pin from initial setting to timer operation start.

Figure 6-26. Status Transition from Timer Output Setting to Operation Start



<1> The operation mode of timer output is set.

- TOM0q bit (0: Toggle mode, 1: Combination operation mode)
- TOL0q bit (0: Forward output, 1: Reverse output)

<2> The timer output signal is set to the initial status by setting TO0q.
<3> The timer output operation is enabled by writing 1 to TOE0q (writing to TO0q is disabled).
<4> The port I/O setting is set to output (see **6.3 (15) Port mode registers 1, 3, 5**).
<5> The timer operation is enabled (TS0q = 1).

Remark q = 0 to 7

6.4.3 Cautions on Channel Output Operation

(1) Changing values set in registers TO0,TOE0,TOL0, and TOM0 during timer operation

Since the timer operations (operations of TCR0q and TDR0q) are independent of the TO0q output circuit and changing the values set in TO0, TOE0, TOL0, and TOM0 does not affect the timer operation, the values can be changed during timer operation. To output an expected waveform from the TO0q pin by timer operation, however, set TO0, TOE0, TOL0, and TOM0 to the values stated in the register setting example of each operation.

When the values set in TOE0, TOL0, and TOM0 (except for TO0) are changed close to the timer interrupt (INTTM0q), the waveform output to the TO0q pin may be different depending on whether the values are changed immediately before or immediately after the timer interrupt (INTTM0q) signal generation timing.

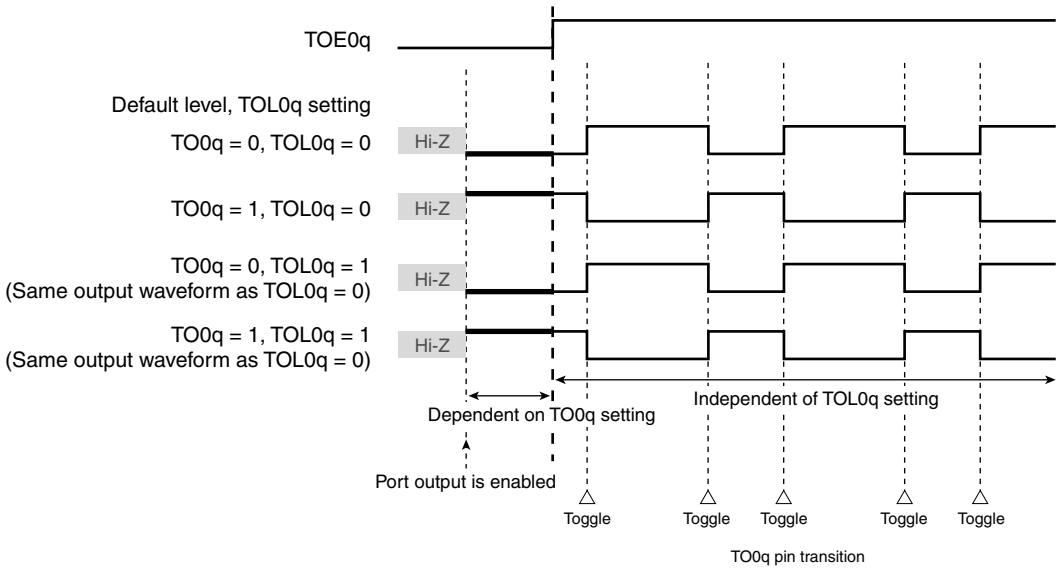
(2) Default level of TO0q pin and output level after timer operation start

The following figure shows the TO0q pin output level transition when writing has been done in the state of TOE0q = 0 before port output is enabled and TOE0q = 1 is set after changing the default level.

(a) When operation starts with TOM0q = 0 setting (toggle output)

The setting of TOL0q is invalid when TOM0q = 0. When the timer operation starts after setting the default level, the toggle signal is generated and the output level of TO0q pin is reversed.

Figure 6-27. TO0q Pin Output Status at Toggle Output (TOM0q = 0)



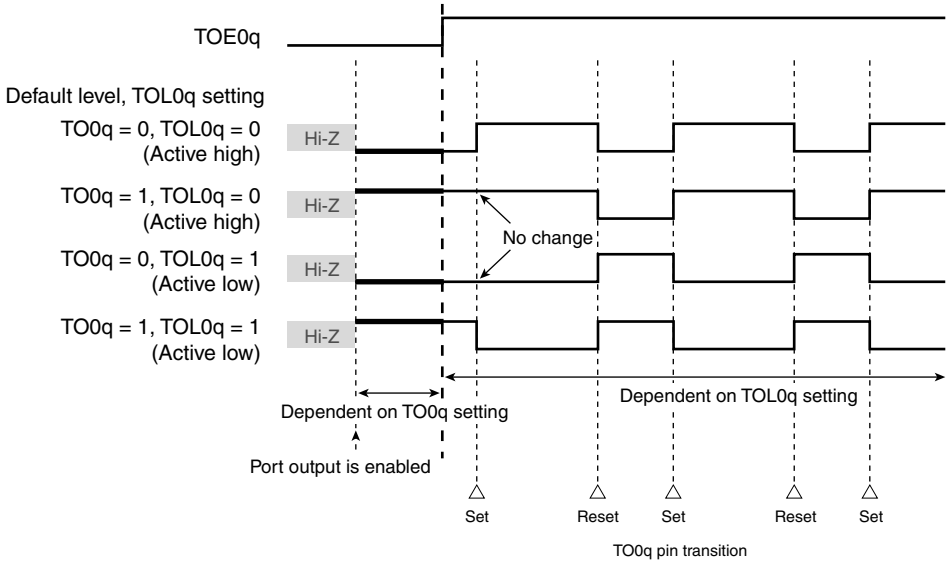
Remarks 1. Toggle: Reverse TO0q pin output status

2. q = 0 to 7

(b) When operation starts with TOM0q = 1 setting (Combination operation mode (PWM output))

When TOM0q = 1, the active level is determined by TOL0q setting.

Figure 6-28. TO0q Pin Output Status at PWM Output (TOM0q = 1)



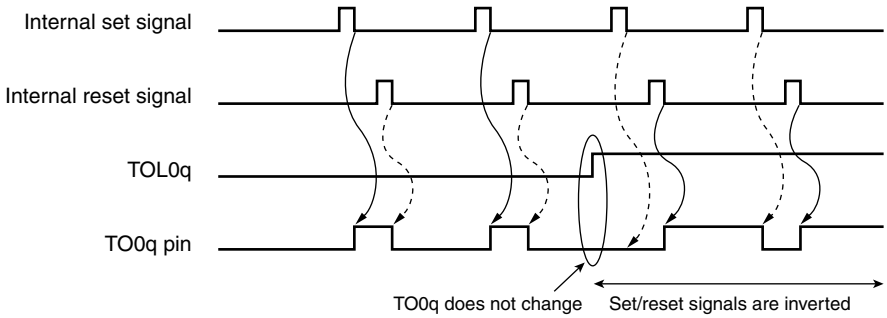
(3) Operation of TO0q pin in combination operation mode (TOM0q = 1)

(a) When TOL0q setting has been changed during timer operation

When the TOL0q setting has been changed during timer operation, the setting becomes valid at the generation timing of TO0q change condition. Rewriting TOL0q does not change the output level of TO0q.

The following figure shows the operation when the value of TOL0q has been changed during timer operation (TOM0q = 1).

Figure 6-29. Operation when TOL0q Has Been Changed during Timer Operation



Remarks 1. Set: The output signal of TO0q pin changes from inactive level to active level.
Reset: The output signal of TO0q pin changes from active level to inactive level.

2. q = 0 to 7

(b) Set/reset timing

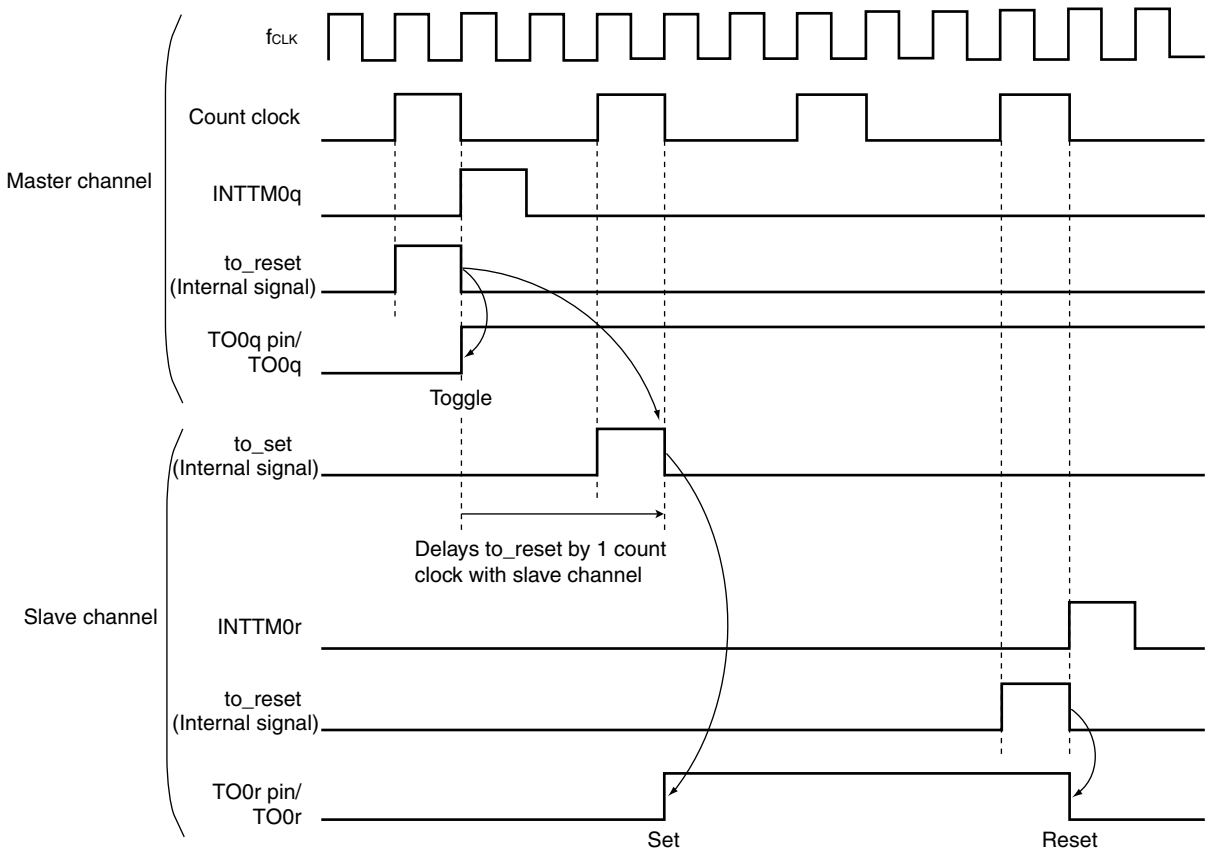
To realize 0%/100% output at PWM output, the TO0q pin/TO0q set timing at master channel timer interrupt (INTTM0q) generation is delayed by 1 count clock by the slave channel timer interrupt (INTTM0r).

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

Figure 6-30 shows the set/reset operating statuses where the master/slave channels are set as follows.

- Master channel: TOE0q = 1, TOM0q = 0, TOL0q = 0
- Slave channel: TOE0r = 1, TOM0r = 1, TOL0r = 0

Figure 6-30. Set/Reset Timing Operating Statuses



Remarks 1. to_reset: TO0q pin reset/toggle signal

to_set: TO0q pin set signal

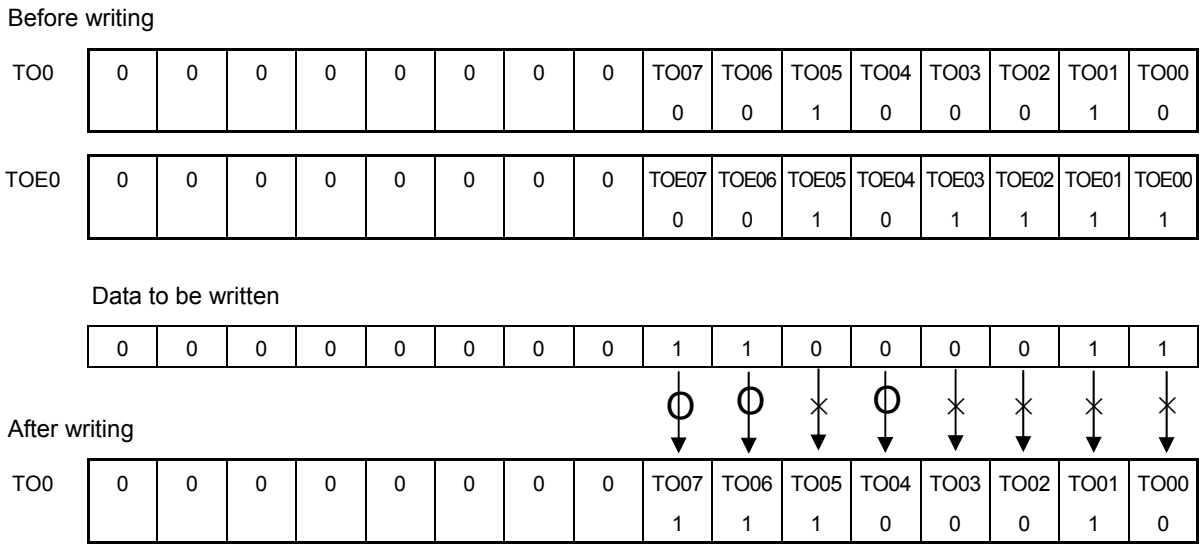
2. q = 0 to 7 (q = 0, 2, 4, 6 for master channel)

$q < r \leq 7$ (where r is a consecutive integer greater than q)

6.4.4 Collective manipulation of TO0q bits

In the TO0 register, the setting bits for all the channels are located in one register in the same way as the TS0 register (channel start trigger). Therefore, TO0q of all the channels can be manipulated collectively. Only specific bits can also be manipulated by setting the corresponding TOE0q = 0 to a target TO0q (channel output).

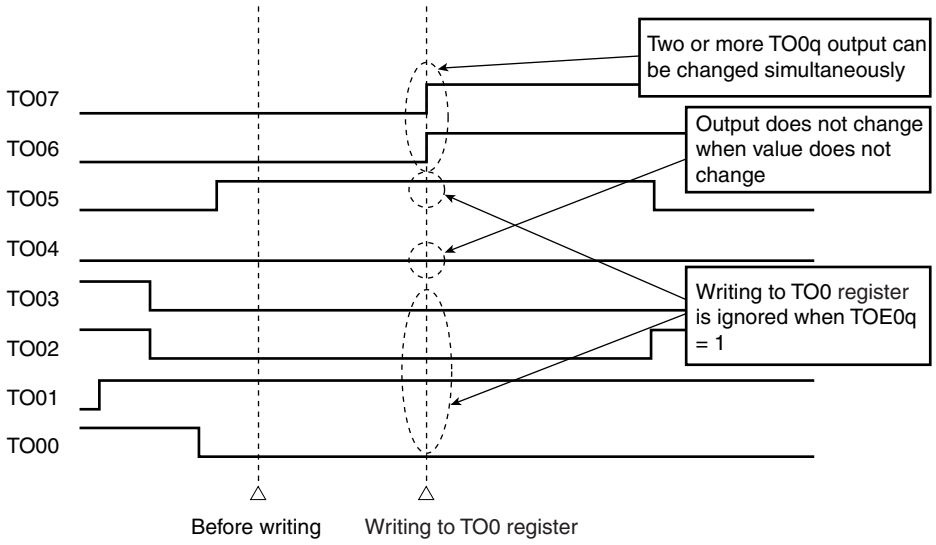
Figure 6-31. Example of TO0q Bits Collective Manipulation



Writing is done only to TO0q bits with TOE0q = 0, and writing to TO0q bits with TOE0q = 1 is ignored.

TO0q (channel output) to which TOE0q = 1 is set is not affected by the write operation. Even if the write operation is done to TO0q, it is ignored and the output change by timer operation is normally done.

Figure 6-32. TO0q Pin Statuses by Collective Manipulation of TO0q Bits



Caution When TOE0q = 1, even if the output by timer interrupt of each timer (INTTM0q) contends with writing to TO0q, output is normally done to TO0q pin.

Remark q = 0 to 7

6.4.5 Timer Interrupt and TOpq Pin Output at Operation Start

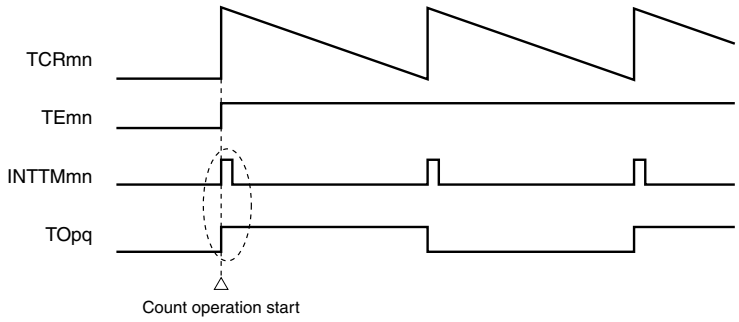
In the interval timer mode or capture mode, the MDmn0 bit in the TMRmn register sets whether or not to generate a timer interrupt at count start.

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 TOpq output is controlled.

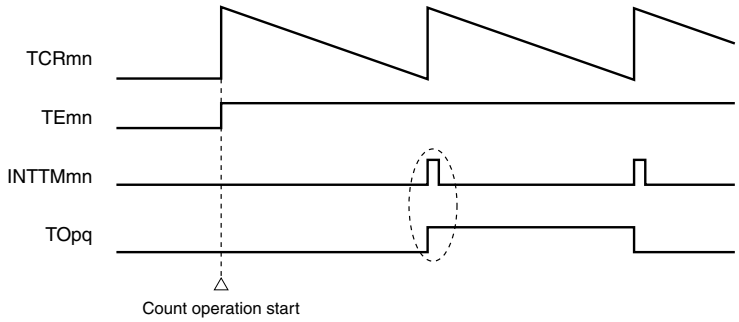
Figures 6-34 and 6-35 show operation examples when the interval timer mode (TOEmn = 1, TOMmn = 0) is set.

Figure 6-33. When MDmn0 is set to 1



When MDmn0 is set to 1, a timer interrupt (INTTMmn) is output at count operation start, and TOpq performs a toggle operation.

Figure 6-34. When MDmn0 is set to 0



When MDmn0 is set to 0, a timer interrupt (INTTMmn) is not output at count operation start, and TOpq does not change either. After counting one cycle, INTTMmn is output and TOpq performs a toggle operation.

Remark mn = 00 to 07, 10 to 13: Unit number + Channel number
pq = 00 to 07: Unit number + Channel number (only for channels provided with timer I/O pins)

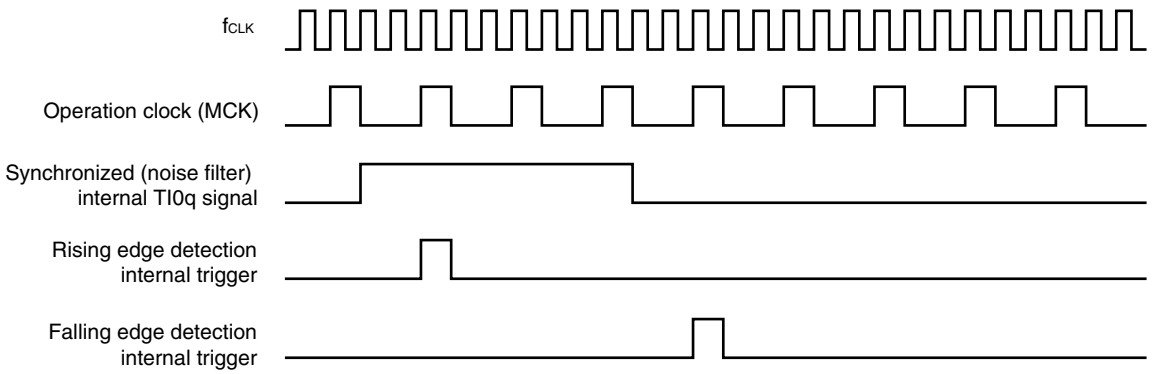
6.5 Channel Input Control

6.5.1 Edge detection circuit

(1) Edge detection basic operation timing

Edge detection circuit sampling is done in accordance with the operation clock (MCK).

Figure 6-35. Edge Detection Basic Operation Timing



Remark q = 0 to 7

6.6 Basic Functions of Timer Array Unit

6.6.1 Overview of single-operation function and combination operation function

The timer array unit consists of several channels and has a single-operation function that allows each channel to operate independently, and a combination operation function that uses two or more channels in combination.

The single-operation function can be used for any channel, regardless of the operation mode of the other channels.

The combination operation function is realized by combining a master channel (reference timer that mainly counts periods) and a slave channel (timer that operates in accordance with the master channel), and several rules must be observed when using this function.

Caution The timer array unit 1 is available for only single-operation function only.

6.6.2 Basic rules of combination operation function

The basic rules of using the combination operation function are as follows.

(1) Only an even channel (channel 0, 2, 4, etc.) can be set as a master channel.
(2) Any channel, except channel 0, can be set as a slave channel.
(3) The slave channel must be lower than the master channel.
Example: If channel 2 of TAU0 is set as a master channel, channel 3 or those that follow (channels 3, 4, etc. 5) can be set as a slave channel.

(4) Two or more slave channels can be set for one master channel.
(5) When two or more master channels are to be used, slave channels with a master channel between them may not be set.
Example: If channels 0 and 4 of TAU0 are set as master channels, channels 1 to 3 can be set as the slave channels of master channel 0. Channels 5 to 7 cannot be set as the slave channels of master channel 0.

(6) The operating clock for a slave channel in combination with a master channel must be the same as that of the master channel. The CKS bit (bit 15 of the TMR0n register) 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 INTTM0n (interrupt), start software trigger, and count clock to the lower channels.
(8) A slave channel can use the INTTM0n (interrupt), start software trigger, and count clock of the master channel, but it cannot transmit its own INTTM0n (interrupt), start software trigger, and count clock to the lower channel.
(9) A master channel cannot use the INTTM0n (interrupt), start software trigger, and count clock from the higher master channel.
(10) To simultaneously start channels that operate in combination, the TS0n bit of the channels in combination must be set at the same time.
(11) During a counting operation, the TS0n bit of all channels that operate in combination or only the master channel can be set. TS0n of only a slave channel cannot be set.
(12) To stop the channels in combination simultaneously, the TT0n bit of the channels in combination must be set at the same time.

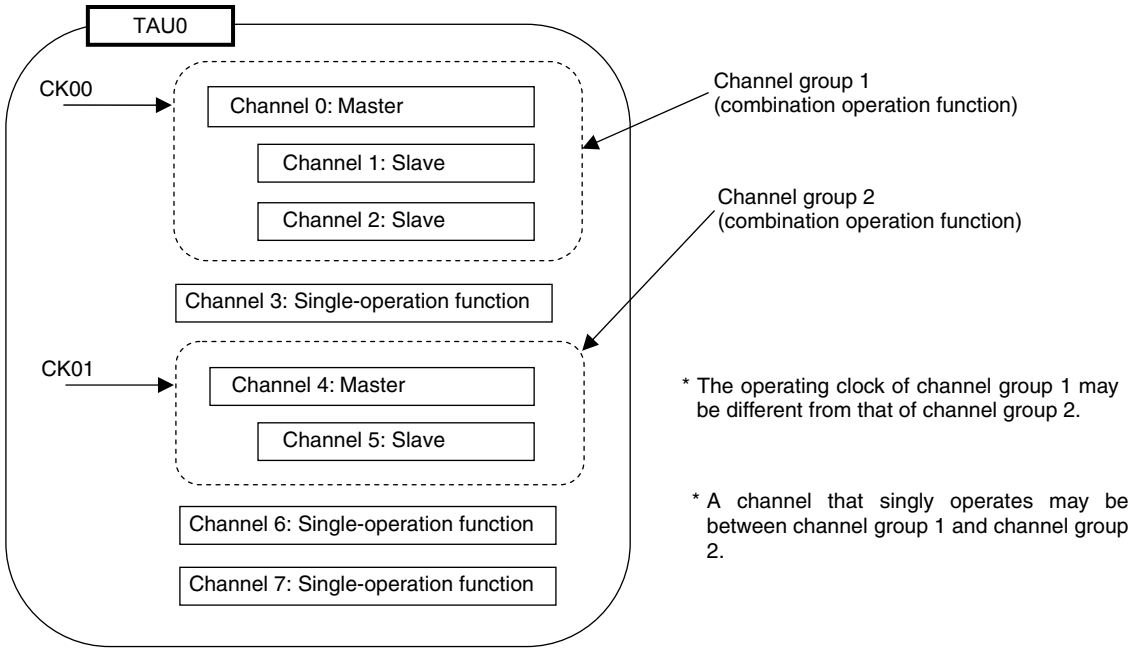
Remark n = 0 to 7

6.6.3 Applicable range of basic rules of combination operation function

The rules of the combination operation function are applied in a channel group (a master channel and slave channels forming one combination operation function).

If two or more channel groups that do not operate in combination are specified, the basic rules of the combination operation function in **6.6.2 Basic rules of combination operation function** do not apply to the channel groups.

Example



6.7 Operations of Timer Array Unit as Independent Channel

6.7.1 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

TOpq performs a toggle operation as soon as INTTMpq has been generated, and outputs a square wave with a duty factor of 50%.

The period and frequency for outputting a square wave from TOpq can be calculated by the following expressions.

• Period of square wave output from TOpq = Period of count clock × (Set value of TDRpq + 1) × 2

• Frequency of square wave output from TOpq = Frequency of count clock/{(Set value of TDRpq + 1) × 2}

The valid edge of TIpq pin input signal, the valid edge of $f_{SUB}/2$, the valid edge of $f_{SUB}/4$, or the valid edge of INTRTC1 can be selected as the count clock, in addition to CKm0 and CKm1. Consequently, the interval timer can be operated, regardless of the f_{CLK} frequency (main system clock, subsystem clock).

When changing the clock selected as f_{CLK} (changing the value of the system clock control register (CKC)), stop the timer array units 0 and 1 (TAUS0, TAUS1) (TT0 = 00FFH, TT1 = 000FH) first.

Only in the case of SDIV=0, CCSmn=1 and TISmn=1, continuously use of TAUm is allowed, even when changing CPU clock. However, the following limitation is existing.

- When changing CPU clock, source clock decrease/increase occurs as follows.

  Main clock → Subsystem clock (CSS = 0→1): −1 clock

  Subsystem clock → Main clock (CSS = 1→0): +1 clock

TCRmn operates as a down counter in the interval timer mode.

TCRmn loads the value of TDRmn at the first count clock after the channel start trigger bit (TSmn) is set to 1. If MDmn0 of TMRmn = 0 at this time, INTTMmn is not output and TOpq is not toggled. If MDmn0 of TMRmn = 1, INTTMmn is output and TOpq is toggled.

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

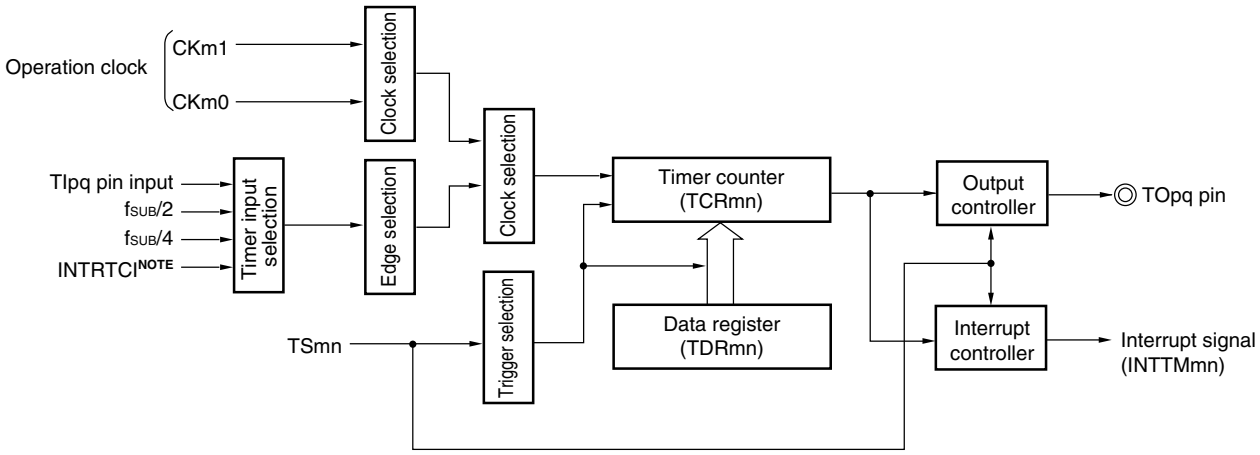
When TCRmn = 0000H, INTTMmn is output and TOpq is toggled at the next count clock. At the same time, TCRmn loads the value of TDRmn again. After that, the same operation is repeated.

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

Remark mn = 00 to 07, 10 to 13: Unit number + Channel number

pq = 00 to 07: Unit number + Channel number (only for channels provided with timer I/O pins)

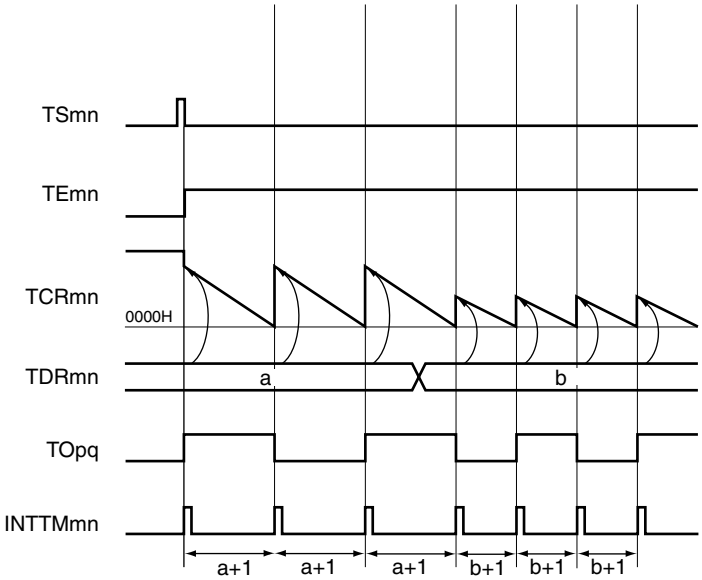
Figure 6-36. Block Diagram of Operation as Interval Timer/Square Wave Output



Note Channels 0 and 4 of timer array unit 0 only

Remark m = 0, 1
mn = 00 to 07, 10 to 13: Unit number + Channel number
pq = 00 to 07: Unit number + Channel number (only for channels provided with timer I/O pins)

Figure 6-37. Example of Basic Timing of Operation as Interval Timer/Square Wave Output (MDmn0 = 1)



Remark mn = 00 to 07, 10 to 13: Unit number + Channel number
pq = 00 to 07: Unit number + Channel number (only for channels provided with timer I/O pins)

Figure 6-38. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (1/3)

(1) When CKm0 or CKm1 is selected as count clock

(a) Timer mode register mn (TMRmn)

TMRmn

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CKSmn			CCSmn	MAS TERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
1/0	0	0	0	0	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 edge of timer input
00B: Sets 00B because these are not used.

Start trigger selection
000B: Selects only software start.

Slave/master selection
0: Cleared to 0 when independent function is selected.

Count clock selection
0: Selects operation clock.

Operation clock selection
0: Selects CKm0 as operation clock of channel n.
1: Selects CKm1 as operation clock of channel n.

(b) Timer output register p (TOp)

TOp

Bit q
TOpq
1/0

0: Outputs 0 from TOpq.
1: Outputs 1 from TOpq.

(c) Timer output enable register p (TOEp)

TOEp

Bit q
TOEpq
1/0

0: Stops the TOpq output operation by counting operation.
1: Enables the TOpq output operation by counting operation.

(d) Timer output level register p (TOLp)

TOLp

Bit q
TOLpq
0

0: Cleared to 0 when TOMpq = 0 (toggle mode)

(e) Timer output mode register p (TOMp)

TOMp

Bit q
TOMpq
0

0: Sets toggle mode.

Remark m = 0, 1, p = 0
mn = 00 to 07, 10 to 13: Unit number + Channel number
pq = 00 to 07: Unit number + Channel number (only for channels provided with timer I/O pins)

Figure 6-38. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (2/3)

(2) When the timer input (TIpq pin input, f_{SUB}/4, f_{SUB}/2 or INTRTCI) is selected as count clock[NOTE] (1/2)

Note The timer input is selected by using TISpq bit, SDIV bit, and RTCISpq bit. For details, refer to Figure 6-17 Format of Timer Input Select Registers 0, 1 (TIS0, TIS1).

(a) Timer mode register mn (TMRmn)

TMRmn

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CKSmn			CCSmn	MASTERmn	STSmn2	STSmn1	STSmn0	CISmn1	CISmn0			MDmn3	MDmn2	MDmn1	MDmn0
1/0	0	0	1	0	0	0	0	1/0	1/0	0	0	0	0	0	1/0

Operation mode of channel 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 edge of timer input
00B: detects falling edge.
01B: detects rising edge.
10B: detects both edges.
11B: Setting prohibited

Start trigger selection
000B: Selects only software start.

Slave/master selection
0: Cleared to 0 when independent function is selected.

Count clock selection
1: Selects timer input valid edge.

Operation clock selection
0: Selects CKm0 as operation clock of channel n.
1: Selects CKm1 as operation clock of channel n.
f_{CLK} (no division) is selected as selected operation clock by TPSm register.

(b) Timer clock select register m (TPSm)

TPSm

Bits 7 to 4, 3 to 0
PRSmk3 to PRSmk0
0000

0000B: Selects f_{CLK} (no division) as operation clock selected by CKSmn of TMRmn register.
k = 0 (bits 0 to 3) when CKm0 is selected and k = 1 (bits 4 to 7) when CKm1 is selected

(c) Timer output register p (TOp)

TOp

Bit q
TOpq
1/0

0: Outputs 0 from TOpq.
1: Outputs 1 from TOpq.

Remark m = 0, 1, p = 0
mn = 00 to 07, 10 to 13: Unit number + Channel number
pq = 00 to 07: Unit number + Channel number (only for channels provided with timer I/O pins)

Figure 6-38. Example of Set Contents of Registers During Operation as Interval Timer/Square Wave Output (3/3)

(2) When the timer input (TIpq pin input, f_{SUB}/4, f_{SUB}/2 or INTRTCI) is selected as count clock (2/2)

(e) Timer output enable register p (TOEp)

	Bit n	
TOEp	TOEpq 1/0	0: Stops the TOpq output operation by counting operation. 1: Enables the TOpq output operation by counting operation.

(f) Timer output level register p (TOLp)

	Bit n	
TOLp	TOLpq 0	0: Cleared to 0 when TOMpq = 0 (toggle mode)

(g) Timer output mode register p (TOMp)

	Bit n	
TOMp	TOMpq 0	0: Sets toggle mode.

Remark p = 0
pq = 00 to 07: Unit number + Channel number (only for channels provided with timer I/O pins)

Figure 6-39. Operation Procedure of Interval Timer/Square Wave Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN or TAU1EN bits of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPSm register. Determines clock frequencies of CKm0 and CKm1.	
Channel default setting	Sets the TMRmn register (determines operation mode of channel). If timer input is selected for the count clock, set the timer input (TIpq pin input, f_{SUB}/4, f_{SUB}/2, or INTRTCI) by using the TISpq, SDIV, and RTCISpq bits. Sets interval (period) value to the TDRmn register.	Channel stops operating. (Clock is supplied and some power is consumed.)
	To use the TOpq output Clears the TOMpq bit of the TOMm register to 0 (toggle mode). Clears the TOLpq bit to 0. Sets the TOpq bit and determines default level of the TOpq output. →	The TOmn pin goes into Hi-Z output state. The TOpq default setting level is output when the port mode register is in the output mode and the port register is 0.
	Sets TOEpq to 1 and enables operation of TOpq. →	TOpq does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TOpq pin outputs the TOpq set level.
Operation start	Sets TOEpq to 1 (only when operation is resumed). Sets the TSmn bit to 1. → The TSmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 1, and count operation starts. Value of TDRmn is loaded to TCRmn at the count clock input. INTTMmn is generated and TOpq performs toggle operation if the MDmn0 bit of the TMRmn register is 1.
During operation	Set values of TMRmn, TOMp, and TOLp registers cannot be changed. Set value of the TDRmn register can be changed. The TCRmn register can always be read. The TSRmn register is not used. Set values of the TOp and TOEp registers can be changed.	Counter (TCRmn) counts down. When count value reaches 0000H, the value of TDRmn is loaded to TCRmn again and the count operation is continued. By detecting TCRmn = 0000H, INTTMmn is generated and TOmn performs toggle operation. After that, the above operation is repeated.
Operation stop	The TTmn bit is set to 1. → The TTmn bit automatically returns to 0 because it is a trigger bit.	TEmn = 0, and count operation stops. TCRmn holds count value and stops. The TOpq output is not initialized but holds current status.
	TOEpq is cleared to 0 and value is set to TOp register. →	The TOpq pin outputs the TOpq set level.

Operation is resumed.

Remark m = 0, 1, p = 0
mn = 00 to 07, 10 to 13: Unit number + Channel number
pq = 00 to 07: Unit number + Channel number (only for channels provided with timer I/O pins)

Figure 6-39. Operation Procedure of Interval Timer/Square Wave Output Function (2/2)

	Software Operation	Hardware Status
TAU stop	To hold the TOpq pin output level Clears TOpq bit to 0 after the value to be held is set to the port register. →	The TOpq pin output level is held by port function.
	When holding the TOpq pin output level is not necessary Switches the port mode register to input mode. →	The TOpq pin output level goes into Hi-Z output state.
	Clears the TAU0EN or TAU1EN bit of the PER0 register to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TOpq bit is cleared to 0 and the TOpq pin is set to port mode.)

Remark mn = 00 to 07, 10 to 13: Unit number + Channel number
pq = 00 to 07: Unit number + Channel number (only for channels provided with timer I/O pins)

6.7.2 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 TI0q 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 TDR0q + 1

TCR0q operates as a down counter in the event counter mode.

When the channel start trigger bit (TS0q) is set to 1, TCR0q loads the value of TDR0q.

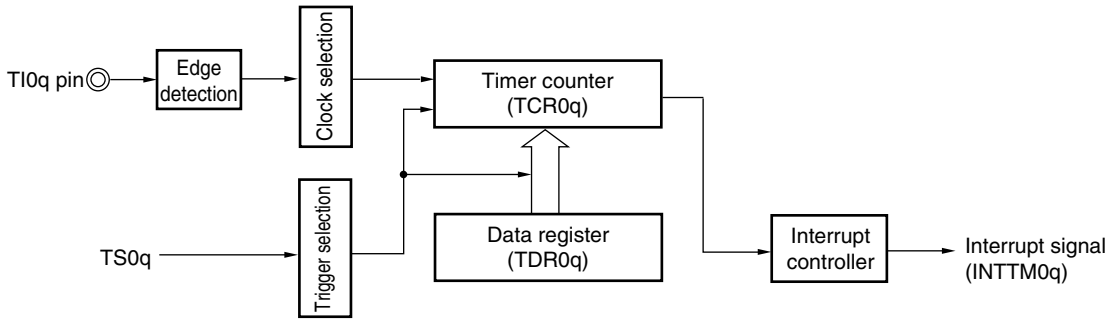
TCR0q counts down each time the valid input edge of the TI0q pin has been detected. When TCR0q = 0000H, TCR0q loads the value of TDR0q again, and outputs INTTM0q.

After that, the above operation is repeated.

TO0q must not be used because its waveform depends on the external event and irregular.

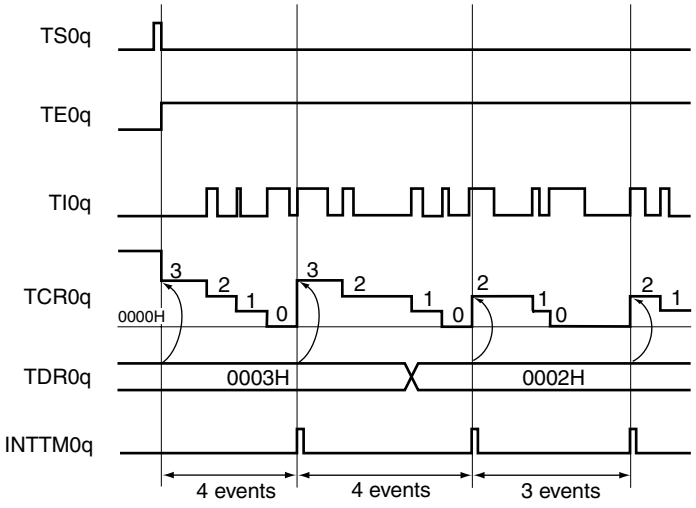
TDR0q can be rewritten at any time. The new value of TDR0q becomes valid during the next count period.

Figure 6-40. Block Diagram of Operation as External Event Counter



Remark q = 0 to 7

Figure 6-41. Example of Basic Timing of Operation as External Event Counter



Remark q = 0 to 7

Figure 6-42. Example of Set Contents of Registers in External Event Counter Mode

(a) Timer mode register 0q (TMR0q)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0q	CKS0q			CCS0q	MASTER0q	STS0q2	STS0q1	STS0q0	CIS0q1	CIS0q0			MD0q3	MD0q2	MD0q1	MD0q0
	1/0	0	0	1	0	0	0	0	1/0	1/0	0	0	0	1	1	0

Operation mode of channel q
011B: Event count mode

Setting of operation when counting is started
0: Neither generates INTTM0q nor inverts timer output when counting is started.

Selection of TI0q 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.

Slave/master selection
0: Cleared to 0 when independent function is selected.

Count clock selection
1: Selects the TI0q pin input valid edge.

Operation clock selection
0: Selects CK00 as operation clock of channel q.
1: Selects CK01 as operation clock of channel q.

(b) Timer output register 0 (TO0)

	Bit q
TO0	TO0q
	0

0: Outputs 0 from TO0q.

(c) Timer output enable register 0 (TOE0)

	Bit q
TOE0	TOE0q
	0

0: Stops the TO0q output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit q
TOL0	TOL0q
	0

0: Cleared to 0 when TOM0q = 0 (toggle mode).

(e) Timer output mode register 0 (TOM0)

	Bit q
TOM0	TOM0q
	0

0: Sets toggle mode.

Remark q = 0 to 7

Figure 6-43. Operation Procedure When External Event Counter Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets the TMR0q register (determines operation mode of channel). Sets number of counts to the TDR0q register. Clears the TOE0q bit of the TOE0 register to 0.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TS0q bit to 1. → The TS0q bit automatically returns to 0 because it is a trigger bit.	TE0q = 1, and count operation starts. Value of TDR0q is loaded to TCR0q and detection of the TI0q pin input edge is awaited.
During operation	Set value of the TDR0q register can be changed. The TCR0q register can always be read. The TSR0q register is not used. Set values of TMR0q, TOM0, TOL0, TO0, and TOE0 registers cannot be changed.	Counter (TCR0q) counts down each time input edge of the TI0q pin has been detected. When count value reaches 0000H, the value of TDR0q is loaded to TCR0q again, and the count operation is continued. By detecting TCR0q = 0000H, the INTTM0q output is generated. After that, the above operation is repeated.
Operation stop	The TT0q bit is set to 1. → The TT0q bit automatically returns to 0 because it is a trigger bit.	TE0q = 0, and count operation stops. TCR0q holds count value and stops.
TAU stop	Clears the TAU0EN bit of the PER0 register to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark q = 0 to 7

6.7.3 Operation as frequency divider

The timer array unit can be used as a frequency divider that divides a clock input to the TI0q pin and outputs the result from TO0q.

The divided clock frequency output from TO0q can be calculated by the following expression.

- When rising edge/falling edge is selected:
  Divided clock frequency = Input clock frequency/{(Set value of TDR0q + 1) × 2}
- When both edges are selected:
  Divided clock frequency ≅ Input clock frequency/(Set value of TDR0q + 1)

TCR0q operates as a down counter in the interval timer mode.

After the channel start trigger bit (TS0q) is set to 1, TCR0q loads the value of TDR0q when the TI0q valid edge is detected. If MD0q0 of TMR0q = 0 at this time, INTTM0q is not output and TO0q is not toggled. If MD0q0 of TMR0q = 1, INTTM0q is output and TO0q is toggled.

After that, TCR0q counts down at the valid edge of TI0q. When TCR0q = 0000H, it toggles TO0q. At the same time, TCR0q loads the value of TDR0q again, and continues counting.

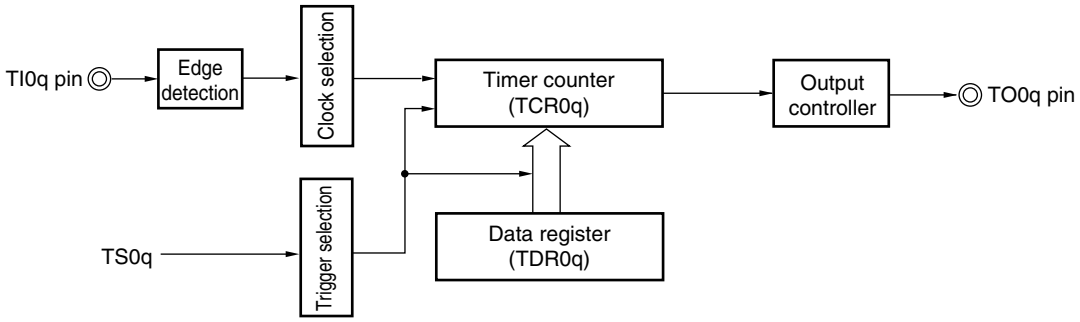
If detection of both the edges of TI0q is selected, the duty factor error of the input clock affects the divided clock period of the TO0q output.

The period of the TO0q output clock includes a sampling error of one period of the operation clock.

Clock period of TO0q output = Ideal TO0q output clock period ± Operation clock period (error)

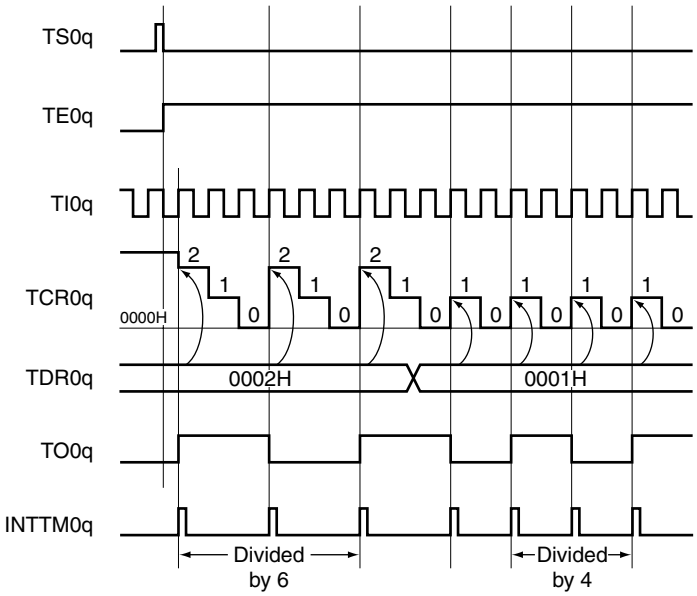
TDR0q can be rewritten at any time. The new value of TDR0q becomes valid during the next count period.

Figure 6-44. Block Diagram of Operation as Frequency Divider



Remark q = 0, 2 to 4

Figure 6-45. Example of Basic Timing of Operation as Frequency Divider (MD0q0 = 1)



Remark q = 0, 2 to 4

Figure 6-46. Example of Set Contents of Registers When Frequency Divider Is Used

(a) Timer mode register 0q (TMR0q)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0q	CKS0q			CCS0q	MASTER0q	STS0q2	STS0q1	STS0q0	CIS0q1	CIS0q0			MD0q3	MD0q2	MD0q1	MD0q0
	1/0	0	0	1	0	0	0	0	1/0	1/0	0	0	0	0	0	1/0

Operation mode of channel q
000B: Interval timer

Setting of operation when counting is started
0: Neither generates INTTM0q nor inverts timer output when counting is started.
1: Generates INTTM0q and inverts timer output when counting is started.

Selection of TI0q 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.

Slave/master selection
0: Cleared to 0 when independent function is selected.

Count clock selection
1: Selects the TI0q pin input valid edge.

Operation clock selection
0: Selects CK00 as operation clock of channel q.
1: Selects CK01 as operation clock of channel q.

(b) Timer output register 0 (TO0)

	Bit q	
TO0	TO0q	0: Outputs 0 from TO0q. 1: Outputs 1 from TO0q.
	1/0	

(c) Timer output enable register 0 (TOE0)

	Bit q	
TOE0	TOE0q	0: Stops the TO0q output operation by counting operation. 1: Enables the TO0q output operation by counting operation.
	1/0	

(d) Timer output level register 0 (TOL0)

	Bit q	
TOL0	TOL0q	0: Cleared to 0 when TOM0q = 0 (toggle mode)
	0	

(e) Timer output mode register 0 (TOM0)

	Bit q	
TOM0	TOM0q	0: Sets toggle mode.
	0	

Remark q = 0, 2 to 4

Figure 6-47. Operation Procedure When Frequency Divider Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets the TMR0q register (determines operation mode of channel). Sets interval (period) value to the TDR0q register.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Clears the TOM0q bit of the TOM0 register to 0 (toggle mode). Clears the TOL0q bit to 0. Sets the TO0q bit and determines default level of the TO0q output. →	The TO0q pin goes into Hi-Z output state. The TO0q default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets TOE0q to 1 and enables operation of TO0q. →	TO0q does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TO0q pin outputs the TO0q set level.
Operation start	Sets the TOE0q to 1 (only when operation is resumed). Sets the TS0q bit to 1. → The TS0q bit automatically returns to 0 because it is a trigger bit.	TE0q = 1, and count operation starts. Value of TDR0q is loaded to TCR0q at the count clock input. INTTM0q is generated and TO0q performs toggle operation if the MD0q0 bit of the TMR0q register is 1.
During operation	Set value of the TDR0q register can be changed. The TCR0q register can always be read. The TSR0q register is not used. Set values of TO0 and TOE0 registers can be changed. Set values of TMR0q, TOM0, and TOL0 registers cannot be changed.	Counter (TCR0q) counts down. When count value reaches 0000H, the value of TDR0q is loaded to TCR0q again, and the count operation is continued. By detecting TCR0q = 0000H, INTTM0q is generated and TO0q performs toggle operation. After that, the above operation is repeated.
Operation stop	The TT0q bit is set to 1. → The TT0q bit automatically returns to 0 because it is a trigger bit.	TE0q = 0, and count operation stops. TCR0q holds count value and stops. The TO0q output is not initialized but holds current status.
	TOE0q is cleared to 0 and value is set to the TO0 register. →	The TO0q pin outputs the TO0q set level.
TAU stop	To hold the TO0q pin output level Clears TO0q bit to 0 after the value to be held is set to the port register. →	The TO0q pin output level is held by port function.
	When holding the TO0q pin output level is not necessary Switches the port mode register to input mode. →	The TO0q pin output level goes into Hi-Z output state.
	Clears the TAU0EN bit of the PER0 register to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0q bit is cleared to 0 and the TO0q pin is set to port mode).

Operation is resumed.

Remark q = 0, 2 to 4

6.7.4 Operation as input pulse interval measurement

The count value can be captured at the TI0q valid edge and the interval of the pulse input to TI0q can be measured.

The pulse interval can be calculated by the following expression.

TI0q input pulse interval = Period of count clock × ((10000H × TSR0q: OVF) + (Capture value of TDR0q + 1))

Caution The TI0q pin input is sampled using the operating clock selected with the CKS0q bit of the TMR0q register, so an error equal to the number of operating clocks occurs.

TCR0q operates as an up counter in the capture mode.

When the channel start trigger (TS0q) is set to 1, TCR0q counts up from 0000H in synchronization with the count clock.

When the TI0q pin input valid edge is detected, the count value is transferred (captured) to TDR0q and, at the same time, the counter (TCR0q) is cleared to 0000H, and the INTTM0q is output. If the counter overflows at this time, the OVF0q bit of the TSR0q register is set to 1. If the counter does not overflow, the OVF0q bit is cleared. After that, the above operation is repeated.

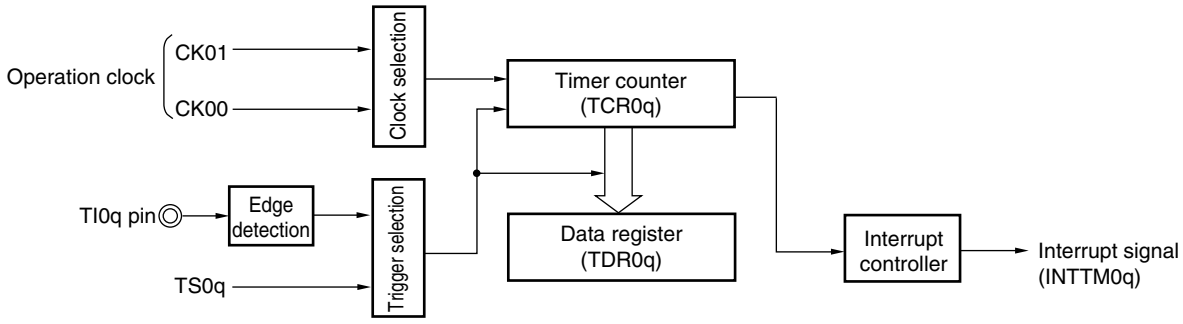
As soon as the count value has been captured to the TDR0q register, the OVF0q bit of the TSR0q register is updated depending on whether the counter overflows during the measurement period. Therefore, the overflow status of the captured value can be checked.

If the counter reaches a full count for two or more periods, it is judged to be an overflow occurrence, and the OVF0q bit of the TSR0q register is set to 1. However, the OVF0q bit is configured as a cumulative flag, the correct interval value cannot be measured if an overflow occurs more than once.

Set STS0q2 to STS0q0 of the TMR0q register to 001B to use the valid edges of TI0q as a start trigger and a capture trigger.

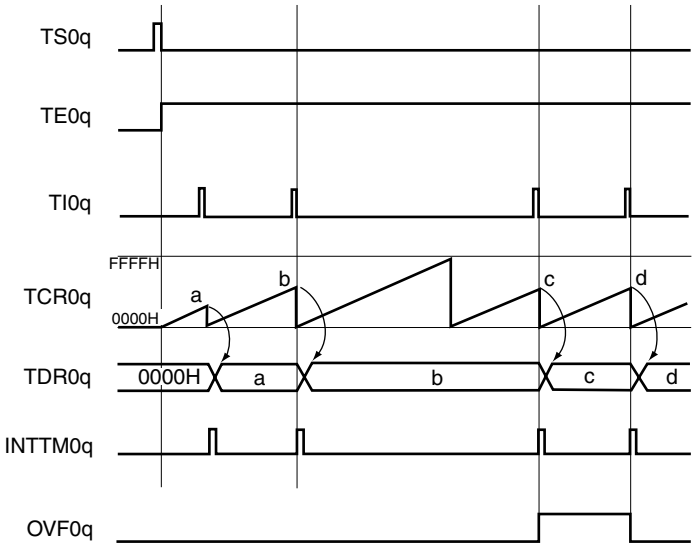
When TE0q = 1, instead of the TI0q pin input, a software operation (TS0q = 1) can be used as a capture trigger.

Figure 6-48. Block Diagram of Operation as Input Pulse Interval Measurement



Remark q = 0 to 7

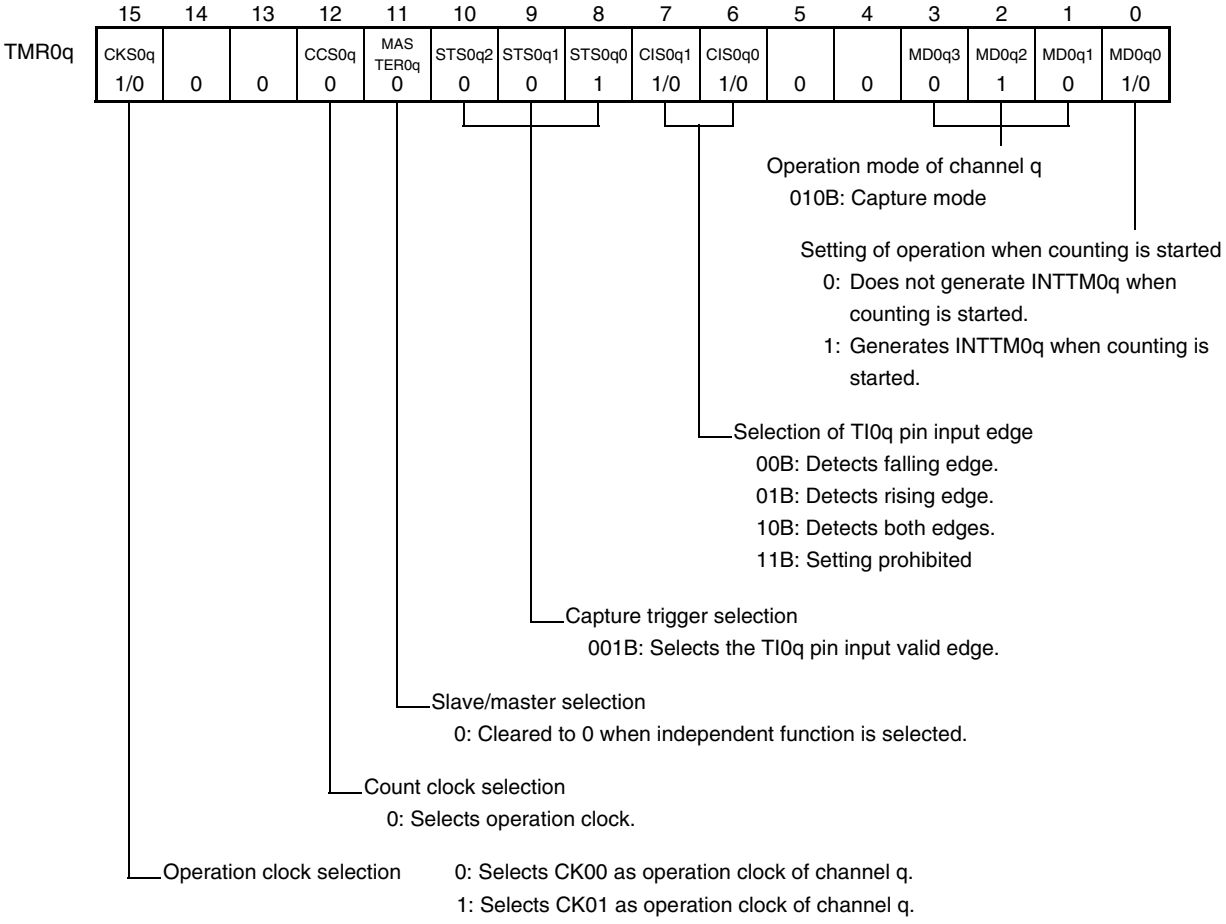
Figure 6-49. Example of Basic Timing of Operation as Input Pulse Interval Measurement (MD0q0 = 0)



Remark q = 0 to 7

Figure 6-50. Example of Set Contents of Registers to Measure Input Pulse Interval

(a) Timer mode register 0q (TMR0q)



(b) Timer output register 0 (TO0)

	Bit q	
TO0	TO0q	0: Outputs 0 from TO0q.
	0	

(c) Timer output enable register 0 (TOE0)

	Bit q	
TOE0	TOE0q	0: Stops TO0q output operation by counting operation.
	0	

(d) Timer output level register 0 (TOL0)

	Bit q	
TOL0	TOL0q	0: Cleared to 0 when TOM0q = 0 (toggle mode).
	0	

(e) Timer output mode register 0 (TOM0)

	Bit q	
TOM0	TOM0q	0: Sets toggle mode.
	0	

Remark q = 0 to 7

Figure 6-51. Operation Procedure When Input Pulse Interval Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets the TMR0q register (determines operation mode of channel).	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets TS0q bit to 1. → The TS0q bit automatically returns to 0 because it is a trigger bit.	TE0q = 1, and count operation starts. TCR0q is cleared to 0000H at the count clock input. When the MD0q0 bit of the TMR0q register is 1, INTTM0q is generated.
During operation	Set values of only the CIS0q1 and CIS0q0 bits of the TMR0q register can be changed. The TDR0q register can always be read. The TCR0q register can always be read. The TSR0q register can always be read. Set values of TOM0, TOL0, TO0, and TOE0 registers cannot be changed.	Counter (TCR0q) counts up from 0000H. When the TI0q pin input valid edge is detected, the count value is transferred (captured) to TDR0q. At the same time, TCR0q is cleared to 0000H, and the INTTM0q signal is generated. If an overflow occurs at this time, the OVF0q bit of the TSR0q register is set; if an overflow does not occur, the OVF0q bit is cleared. After that, the above operation is repeated.
Operation stop	The TT0q bit is set to 1. → The TT0q bit automatically returns to 0 because it is a trigger bit.	TE0q = 0, and count operation stops. TCR0q holds count value and stops. The OVF0q bit of the TSR0q register is also held.
TAU stop	Clears the TAU0EN bit of the PER0 register to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark q = 0 to 7

6.7.5 Operation as input signal high-/low-level width measurement

By starting counting at one edge of TI0q and capturing the number of counts at another edge, the signal width (high-level width/low-level width) of TI0q can be measured. The signal width of TI0q can be calculated by the following expression.

Signal width of TI0q input = Period of count clock × ((10000H × TSR0q: OVF) + (Capture value of TDR0q + 1))

Caution The TI0q pin input is sampled using the operating clock selected with the CKS0q bit of the TMR0q register, so an error equal to the number of operating clocks occurs.

TCR0q operates as an up counter in the capture & one-count mode.

When the channel start trigger (TS0q) is set to 1, TE0q is set to 1 and the TI0q pin start edge detection wait status is set.

When the TI0q start valid edge (rising edge of TI0q when the high-level width is to be measured) is detected, the counter counts up in synchronization with the count clock. When the valid capture edge (falling edge of TI0q when the high-level width is to be measured) is detected later, the count value is transferred to TDR0q and, at the same time, INTTM0q is output. If the counter overflows at this time, the OVF0q bit of the TSR0q register is set to 1. If the counter does not overflow, the OVF0q bit is cleared. TCR0q stops at the value “value transferred to TDR0q + 1”, and the TI0q 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 TDR0q register, the OVF0q bit of the TSR0q 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 OVF0q bit of the TSR0q register is set to 1. However, the OVF0q bit is configured as an integral flag, and the correct interval value cannot be measured if an overflow occurs more than once.

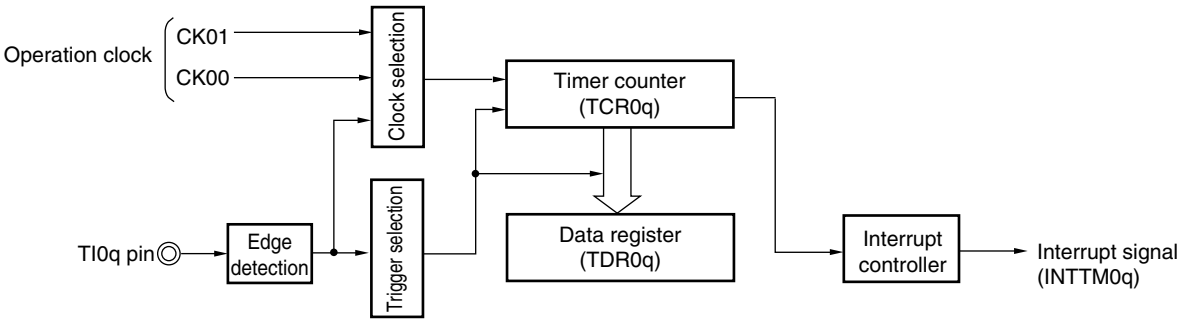
Whether the high-level width or low-level width of the TI0q pin is to be measured can be selected by using the CIS0q1 and CIS0q0 bits of the TMR0q register.

Because this function is used to measure the signal width of the TI0q pin input, TS0q cannot be set to 1 while TE0q is 1.

CIS0q1, CIS0q0 of TMR0q = 10B: Low-level width is measured.
CIS0q1, CIS0q0 of TMR0q = 11B: High-level width is measured.

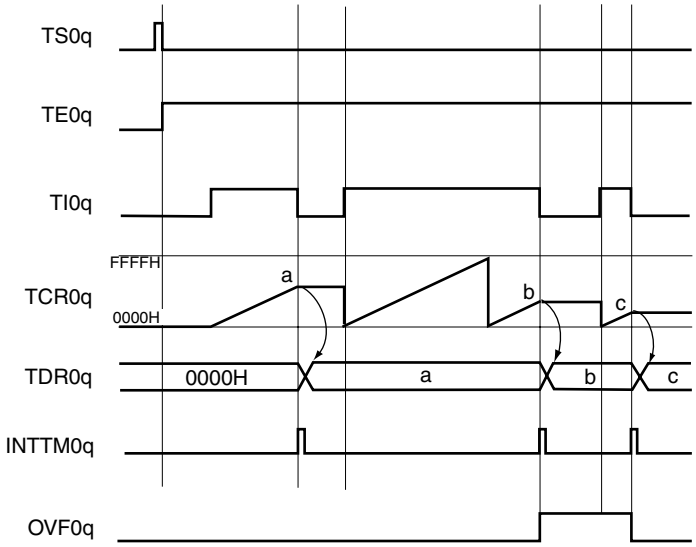
Remark q = 0 to 7

Figure 6-52. Block Diagram of Operation as Input Signal High-/Low-Level Width Measurement



Remark q = 0 to 7

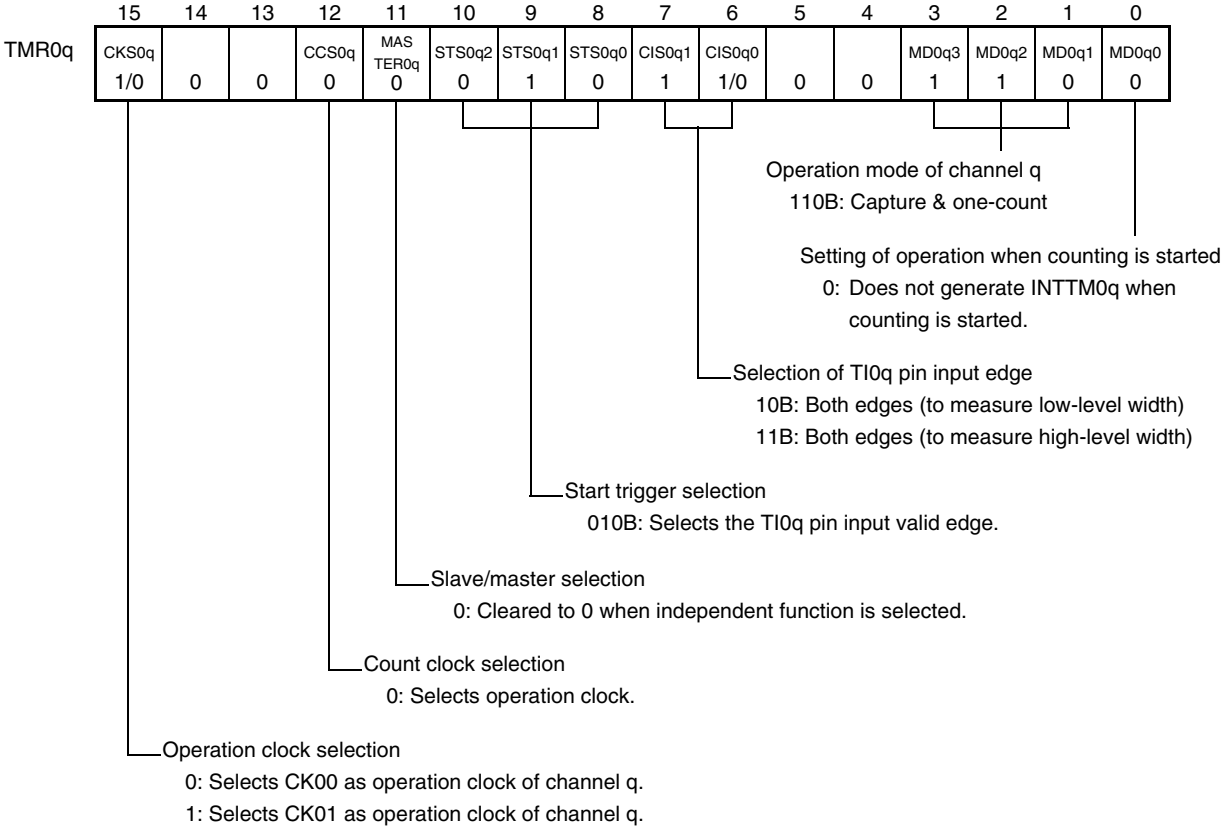
Figure 6-53. Example of Basic Timing of Operation as Input Signal High-/Low-Level Width Measurement



Remark q = 0 to 7

Figure 6-54. Example of Set Contents of Registers to Measure Input Signal High-/Low-Level Width

(a) Timer mode register 0q (TMR0q)



(b) Timer output register 0 (TO0)

	Bit q		
TO0	TO0q	0	0: Outputs 0 from TO0q.

(c) Timer output enable register 0 (TOE0)

	Bit q		
TOE0	TOE0q	0	0: Stops the TO0q output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit q		
TOL0	TOL0q	0	0: Cleared to 0 when TOM0q = 0 (toggle mode).

(e) Timer output mode register 0 (TOM0)

	Bit q		
TOM0	TOM0q	0	0: Sets toggle mode.

Remark q = 0 to 7

Figure 6-55. Operation Procedure When Input Signal High-/Low-Level Width Measurement Function Is Used

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets the TMR0q register (determines operation mode of channel). Clears TOE0q to 0 and stops operation of TO0q.	Channel stops operating. (Clock is supplied and some power is consumed.)
Operation start	Sets the TS0q bit to 1. → The TS0q bit automatically returns to 0 because it is a trigger bit.	TE0q = 1, and the TI0q pin start edge detection wait status is set.
	Detects TI0q pin input count start valid edge. →	Clears TCR0q to 0000H and starts counting up.
During operation	Set value of the TDR0q register can be changed. The TCR0q register can always be read. The TSR0q register is not used. Set values of TMR0q, TOM0, TOL0, TO0, and TOE0 registers cannot be changed.	When the TI0q pin start edge is detected, the counter (TCR0q) counts up from 0000H. If a capture edge of the TI0q pin is detected, the count value is transferred to TDR0q and INTTM0q is generated. If an overflow occurs at this time, the OVF0q bit of the TSR0q register is set; if an overflow does not occur, the OVF0q bit is cleared. TCR0q stops the count operation until the next TI0q pin start edge is detected.
Operation stop	The TT0q bit is set to 1. → TT0q bit automatically returns to 0 because it is a trigger bit.	TE0q = 0, and count operation stops. TCR0q holds count value and stops. The OVF0q bit of the TSR0q register is also held.
TAU stop	Clears the TAU0EN bit of the PER0 register to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized.

Operation is resumed.

Remark q = 0 to 7

6.8 Operations of Plural Channels of Timer Array Unit

6.8.1 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 TDR0n (master) + 1} × Count clock period Duty factor [%] = {Set value of TDR0p (slave)}/{Set value of TDR0n (master) + 1} × 100 0% output: Set value of TDR0p (slave) = 0000H 100% output: Set value of TDR0p (slave) ≥ {Set value of TDR0n (master) + 1}

Remark The duty factor exceeds 100% if the set value of TDR0p (slave) > (set value of TDR0n (master) + 1), it summarizes to 100% output.

The master channel operates in the interval timer mode and counts the periods. When the channel start trigger (TS0n) is set to 1, INTTM0n is output. TCR0n counts down starting from the loaded value of TDR0n, in synchronization with the count clock. When TCR0n = 0000H, INTTM0n is output. TCR0n loads the value of TDR0n again. After that, it continues the similar operation.

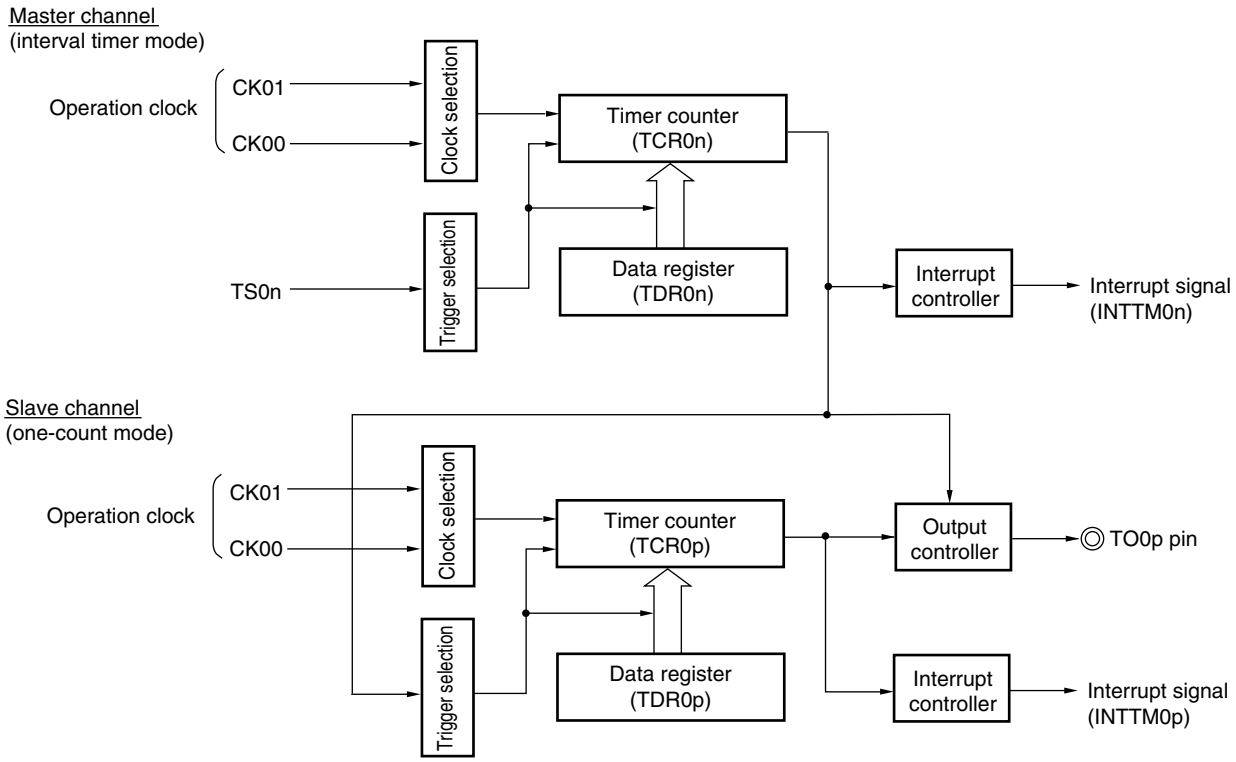
TCR0p of a slave channel operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TO0p pin. TCR0p of the slave channel loads the value of TDR0p, using INTTM0n of the master channel as a start trigger, and stops counting until the next start trigger (INTTM0n of the master channel) is input.

The output level of TO0p becomes active one count clock after generation of INTTM0n from the master channel, and inactive when TCR0p = 0000H.

Caution To rewrite both TDR0n of the master channel and TDR0p of the slave channel, a write access is necessary two times. The timing at which the values of TDR0n and TDR0p are loaded to TCR0n and TCR0p is upon occurrence of INTTM0n of the master channel. Thus, when rewriting is performed split before and after occurrence of INTTM0n of the master channel, the TO0p pin cannot output the expected waveform. To rewrite both TDR0n of the master and TDR0p of the slave, therefore, be sure to rewrite both the registers immediately after INTTM0n is generated from the master channel.

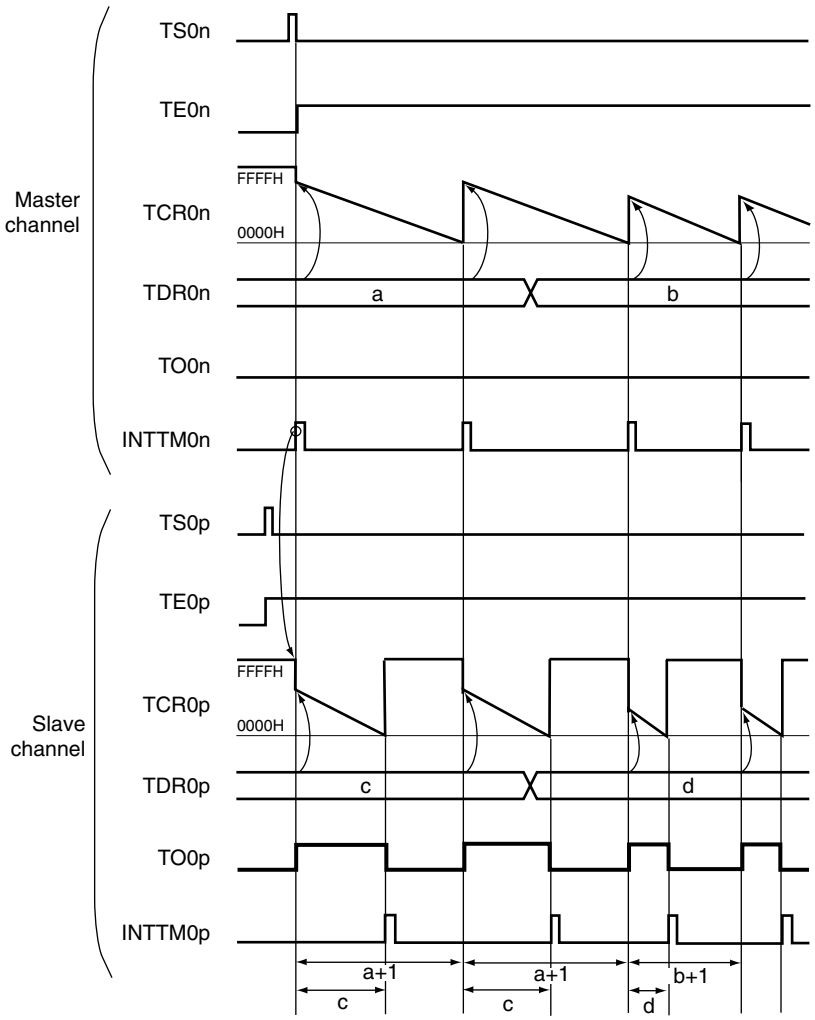
Remark n = 0, 2, 4, 6
p = n+1
TO00 to TO07 pins

Figure 6-56. Block Diagram of Operation as PWM Function



Remark n = 0, 2, 4, 6
p = n+1
TO00 to TO07 pins

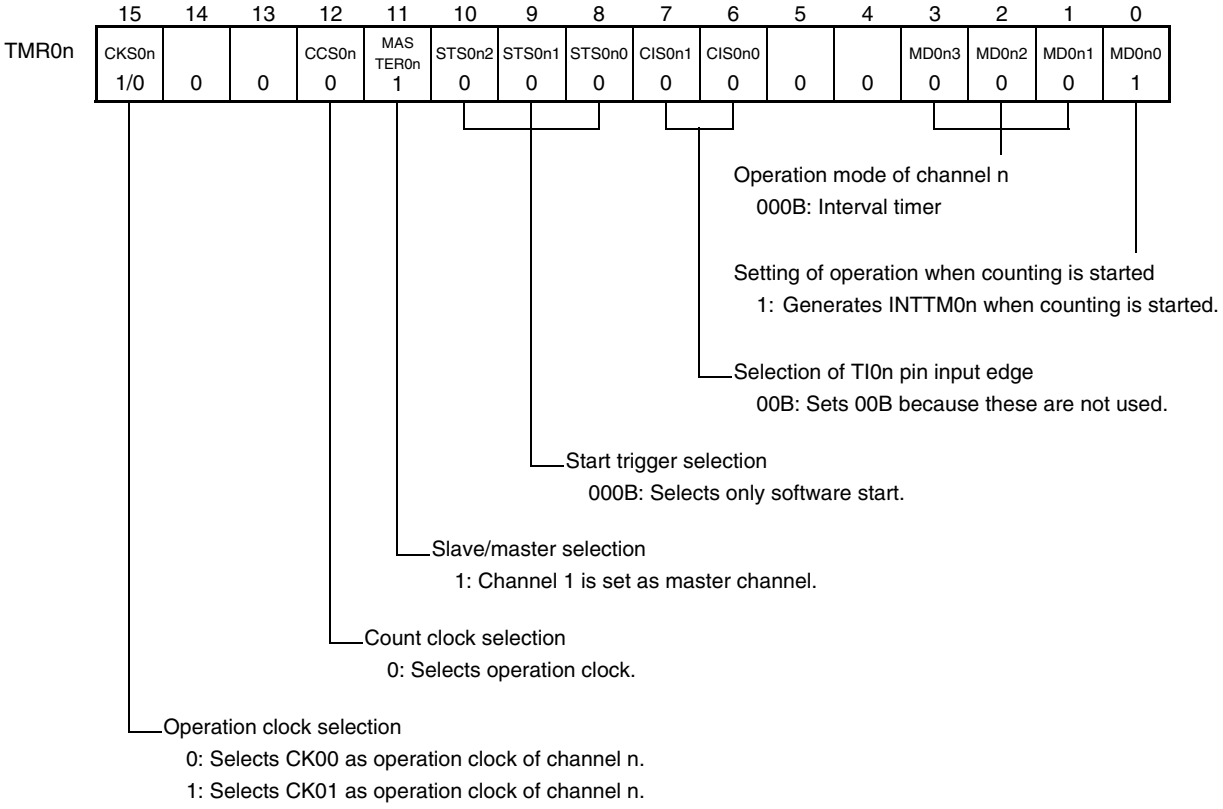
Figure 6-57. Example of Basic Timing of Operation as PWM Function



Remark n = 0, 2, 4, 6
p = n+1
TO00 to TO07 pins

Figure 6-58. Example of Set Contents of Registers When PWM Function (Master Channel) Is Used

(a) Timer mode register 0n (TMR0n)



(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n	0: Outputs 0 from TO0n.
	0	

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n	0: Stops the TO0n output operation by counting operation.
	0	

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n	0: Cleared to 0 when TOM0n = 0 (toggle mode).
	0	

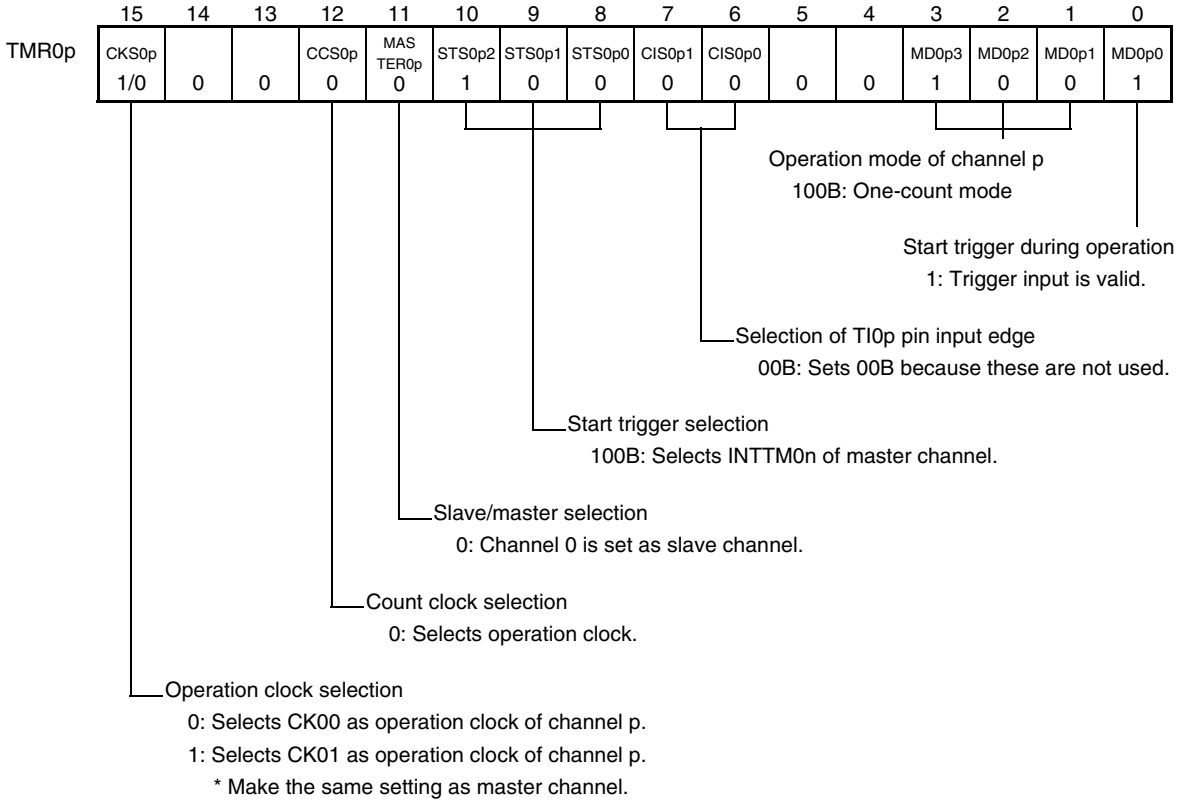
(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n	0: Sets toggle mode.
	0	

Remark n = 0, 2, 4, 6
p = n+1
TO00 to TO07, TI00 to TI07 pins

Figure 6-59. Example of Set Contents of Registers When PWM Function (Slave Channel) Is Used

(a) Timer mode register 0p (TMR0p)



(b) Timer output register 0 (TO0)

	Bit p	
TO0	TO0p 1/0	0: Outputs 0 from TO0p. 1: Outputs 1 from TO0p.

(c) Timer output enable register 0 (TOE0)

	Bit p	
TOE0	TOE0p 1/0	0: Stops the TO0p output operation by counting operation. 1: Enables the TO0p output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit p	
TOL0	TOL0p 1/0	0: Positive logic output (active-high) 1: Inverted output (active-low)

(e) Timer output mode register 0 (TOM0)

	Bit p	
TOM0	TOM0p 1	1: Sets the combination operation mode.

Remark n = 0, 2, 4, 6
p = n+1
TO00 to TO07, TI00 to TI07 pins

Figure 6-60. Operation Procedure When PWM Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets the TMR0n and TMR0p registers of two channels to be used (determines operation mode of channels). An interval (period) value is set to the TDR0n register of the master channel, and a duty factor is set to the TDR0p register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOM0p bit of the TOM0 register is set to 1 (combination operation mode). Sets the TOL0p bit. Sets the TO0p bit and determines default level of the TO0p output. →	The TO0n pin goes into Hi-Z output state. The TO0n default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets TOE0p to 1 and enables operation of TO0p. →	TO0p does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TO0p pin outputs the TO0p set level.

Remark n = 0, 2, 4, 6
p = n+1
TO00 to TO07 pins

Figure 6-60. Operation Procedure When PWM Function Is Used (2/2)

	Software Operation	Hardware Status
Operation start	Sets TOE0p (slave) to 1 (only when operation is resumed). The TS0n (master) and TS0p (slave) bits of the TS0 register are set to 1 at the same time. →	TE0n = 1, TE0p = 1 When the master channel starts counting, INTTM0n is generated. Triggered by this interrupt, the slave channel also starts counting.
	The TS0n and TS0p bits automatically return to 0 because they are trigger bits.	
During operation	Set values of the TMR0n and TMR0p registers and TOM0n, TOM0p, TOL0n, and TOL0p bits cannot be changed. Set values of the TDR0n and TDR0p registers can be changed after INTTM0n of the master channel is generated. The TCR0n and TCR0p registers can always be read. The TSR0n and TSR0p registers are not used. Set values of the TO0 and TOE0 registers cannot be changed.	The counter of the master channel loads the TDR0n value to TCR0n, and counts down. When the count value reaches TCR0n = 0000H, INTTM0n output is generated. At the same time, the value of the TDR0n register is loaded to TCR0n, and the counter starts counting down again. At the slave channel, the value of TDR0p is loaded to TCR0p, triggered by INTTM0n of the master channel, and the counter starts counting down. The output level of TO0p becomes active one count clock after generation of the INTTM0n output from the master channel. It becomes inactive when TCR0p = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TT0n (master) and TT0p (slave) bits are set to 1 at the same time. →	TE0n, TE0p = 0, and count operation stops. TCR0n and TCR0p hold count value and stops. The TO0p output is not initialized but holds current status.
	TOE0p of slave channel is cleared to 0 and value is set to the TO0p register. →	The TO0p pin outputs the TO0p set level.
TAU stop	To hold the TO0p pin output levels Clears TO0p bit to 0 after the value to be held is set to the port register. →	The TO0p pin output levels are held by port function.
	When holding the TO0p pin output levels is not necessary Switches the port mode register to input mode. →	The TO0p pin output levels go into Hi-Z output state.
	Clears the TAU0EN bit of the PER0 register to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0p bit is cleared to 0 and the TO0p pin is set to port mode.)

Operation is resumed.

Remark n = 0, 2, 4, 6
p = n+1
TO00 to TO07 pins

6.8.2 Operation as one-shot pulse output function

A one-shot pulse with any delay pulse width can be generated by using two channels in combination and TI0n pin input or software manipulation (TS0n = 1).

The delay time and pulse width can be calculated by the following expressions.

Delay time = {Set value of TDR0n (master) + 2} × Count clock period
Pulse width = {Set value of TDR0p (slave)} × Count clock period

The Master channel operates in the one-count mode and counts the delays. TCR0n of the master channel starts operating upon start trigger detection and TCR0n loads the value of TDR0n. TCR0n counts down from the value of TDR0n it has loaded, in synchronization with the count clock. When TCR0n = 0000H, it outputs INTTM0n and stops counting until the next start trigger is detected.

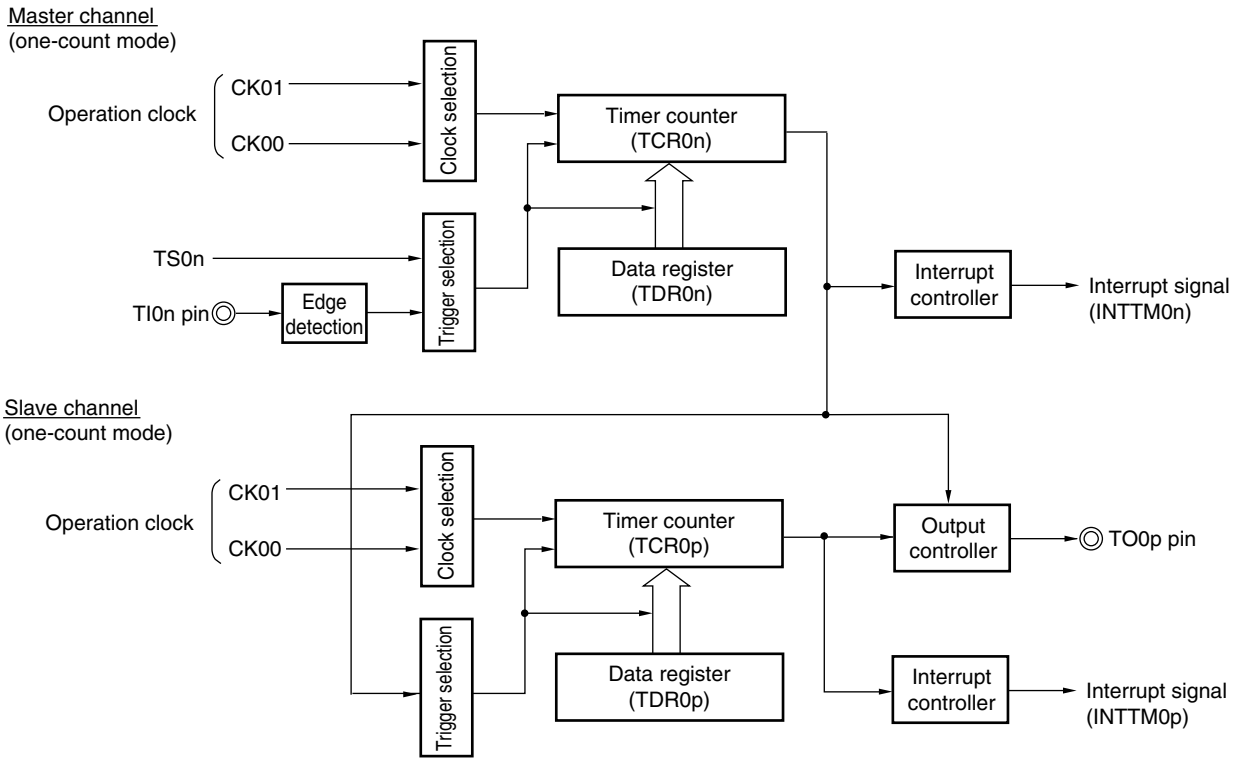
The slave channel operates in the one-count mode and counts the pulse width. TCR0p of the slave channel starts operation using INTTM0n of the master channel as a start trigger, and loads the TDR0p value. TCR0p counts down from the value of TDR0p it has loaded, in synchronization with the count value. When TCR0p = 0000H, it outputs INTTM0p and stops counting until the next start trigger (INTTM0n of the master channel) is detected. The output level of TO0p becomes active one count clock after generation of INTTM0n from the master channel, and inactive when TCR0p = 0000H.

Instead of using the TI0n pin input, a one-shot pulse can also be output using the software operation (TS0n = 1) as a start trigger.

Caution The timing of loading of TDR0n of the master channel is different from that of TDR0p of the slave channel. If TDR0n and TDR0p are rewritten during operation, therefore, an illegal waveform is output. Be sure to rewrite TDR0n and TDR0p after INTTM0n of the channel to be rewritten is generated.

Remark n = 0, 2, 4, 6
p = n+1
TO00 to TO07, TI00 to TI07 pins

Figure 6-61. Block Diagram of Operation as One-Shot Pulse Output Function

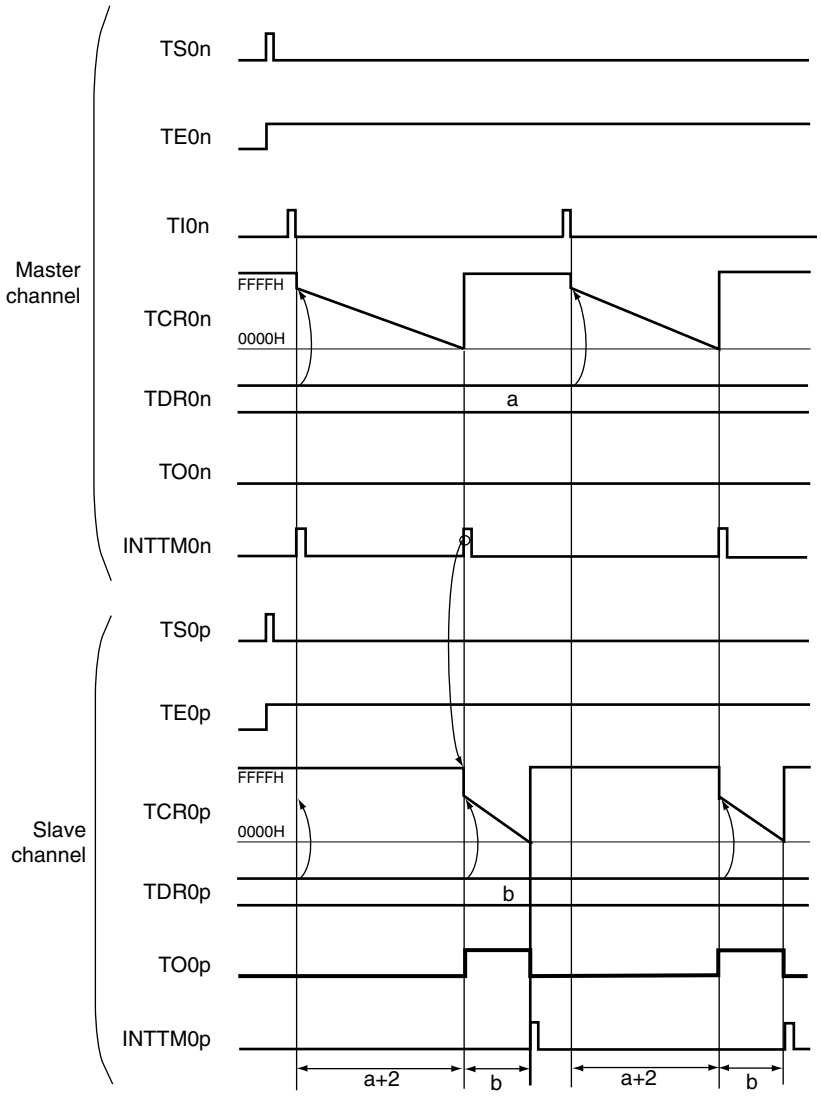


Remark n = 0, 2, 4, 6

p = n+1

TO00 to TO07, TI00 to TI07 pins

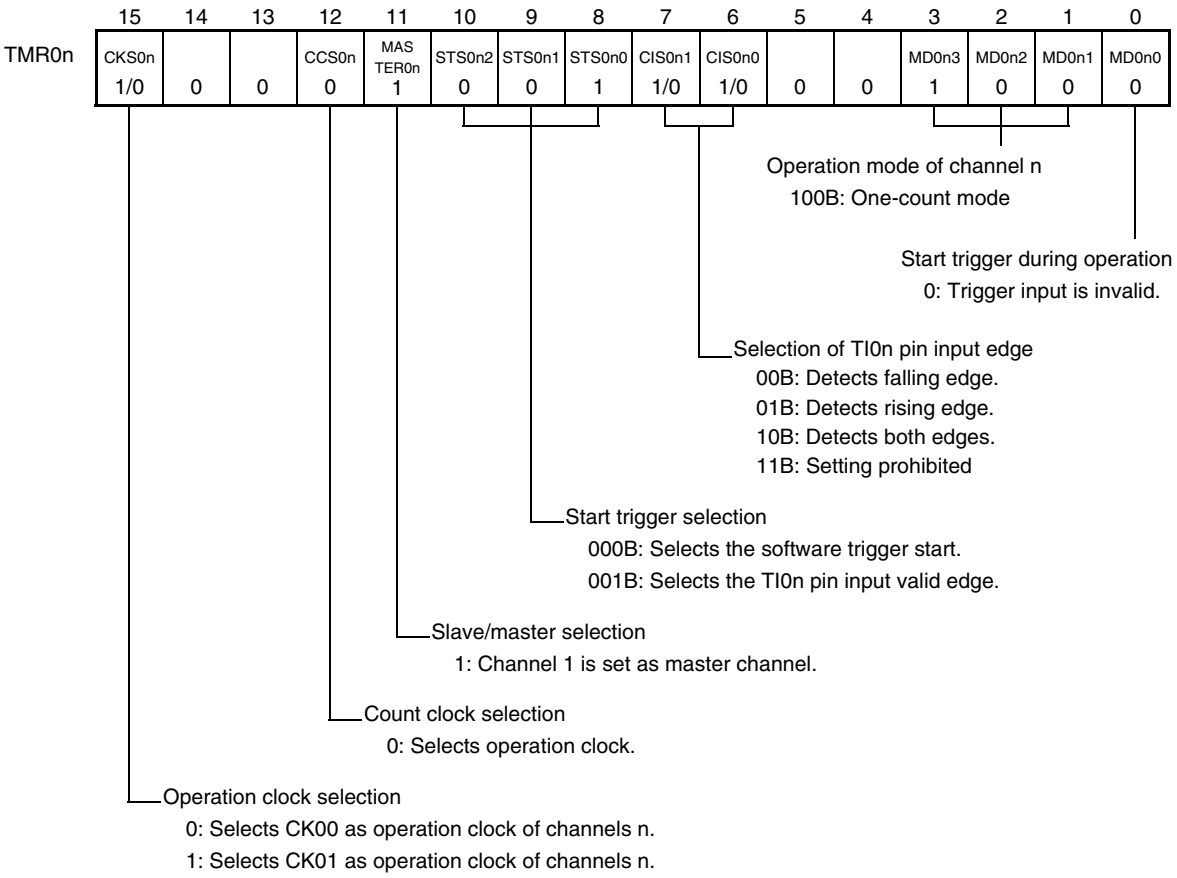
Figure 6-62. Example of Basic Timing of Operation as One-Shot Pulse Output Function (Start Trigger TI0n Input Valid Edge)



Remark n = 0, 2, 4, 6
p = n+1
TO00 to TO07 pins

Figure 6-63. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Master Channel)

(a) Timer mode register 0n (TMR0n)



(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n	0: Outputs 0 from TO0n.
	0	

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n	0: Stops the TO0n output operation by counting operation.
	0	

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n	0: Cleared to 0 when TOM0n = 0 (toggle mode).
	0	

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n	0: Sets toggle mode.
	0	

Remark n = 0, 2, 4, 6

TO00 to TO07, TI00 to TI07 pins

Figure 6-64. Example of Set Contents of Registers When One-Shot Pulse Output Function Is Used (Slave Channel)

(a) Timer mode register 0p (TMR0p)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0p	CKS0p			CCS0p	MASTER0p	STS0p2	STS0p1	STS0p0	CIS0p1	CIS0p0			MD0p3	MD0p2	MD0p1	MD0p0
	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 TI0p pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTM0n of master channel.

Slave/master selection
0: Channel 0 is set as slave channel.

Count clock selection
0: Selects operation clock.

Operation clock selection
0: Selects CK00 as operation clock of channel p.
1: Selects CK01 as operation clock of channel p.
* Make the same setting as master channel.

(b) Timer output register 0 (TO0)

	Bit p
TO0	TO0p
	1/0

0: Outputs 0 from TO0p.
1: Outputs 1 from TO0p.

(c) Timer output enable register 0 (TOE0)

	Bit p
TOE0	TOE0p
	1/0

0: Stops the TO0p output operation by counting operation.
1: Enables the TO0p output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit p
TOL0	TOL0p
	1/0

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

(e) Timer output mode register 0 (TOM0)

	Bit p
TOM0	TOM0p
	1

1: Sets the combination operation mode.

Remark n = 0, 2, 4, 6
p = n+1
TO00 to TO07, TI00 to TI07 pins

Figure 6-65. Operation Procedure of One-Shot Pulse Output Function (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets the TMR0n and TMR0p registers of two channels to be used (determines operation mode of channels). An output delay is set to the TDR0n register of the master channel, and a pulse width is set to the TDR0p register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOM0p bit of the TOM0 register is set to 1 (combination operation mode). Sets the TOL0p bit. Sets the TO0p bit and determines default level of the TO0p output. →	The TO0n pin goes into Hi-Z output state. The TO0n default setting level is output when the port mode register is in output mode and the port register is 0.
	Sets TOE0p to 1 and enables operation of TO0p. →	TO0p does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TO0p pin outputs the TO0p set level.

Remark n = 0, 2, 4, 6
p = n+1
TO00 to TO07 pins

Figure 6-65. Operation Procedure of One-Shot Pulse Output Function (2/2)

	Software Operation	Hardware Status
Operation start	Sets TOE0p (slave) to 1 (only when operation is resumed). The TS0n (master) and TS0p (slave) bits of the TS0 register are set to 1 at the same time. →	TE0n and TE0p are set to 1 and the master channel enters the TI0n input edge detection wait status. Counter stops operating.
	Detects the start trigger of master channel. →	Master channel starts counting.
	(The valid edge of the TI0n pin input is detected or the TS0n bit is set to 1.)	
During operation	Set values of only the CIS0n1 and CIS0n0 bits of the TMR0n register can be changed. Set values of the TMR0p, TDR0n, and TDR0p registers and TOM0n, TOM0p, TOL0n, and TOL0p bits cannot be changed. The TCR0n and TCR0p registers can always be read. The TSR0n and TSR0p registers are not used. Set values of the TO0 and TOE0 registers can be changed.	Master channel loads the value of TDR0n to TCR0n when the start trigger is detected, and the counter starts counting down. When the count value reaches TCR0n = 0000H, the INTTM0n output is generated, and the counter stops until the next valid edge is input to the TI0n pin. The slave channel, triggered by INTTM0n of the master channel, loads the value of TDR0p to TCR0p, and the counter starts counting down. The output level of TO0p becomes active one count clock after generation of INTTM0n from the master channel. It becomes inactive when TCR0p = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TT0n (master) and TT0p (slave) bits are set to 1 at the same time. →	TE0n, TE0p = 0, and count operation stops. TCR0n and TCR0p hold count value and stops. The TO0p output is not initialized but holds current status.
	TOE0p of slave channel is cleared to 0 and value is set to the TO0 register. →	The TO0p pin outputs the TO0n set level.
TAU stop	To hold the TO0p pin output levels Clears TO0p bit to 0 after the value to be held is set to the port register. →	The TO0p pin output levels are held by port function.
	When holding the TO0p pin output levels is not necessary Switches the port mode register to input mode. →	The TO0p pin output levels go into Hi-Z output state.
	Clears the TAU0EN bit of the PER0 register to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0p bit is cleared to 0 and the TO0p pin is set to port mode.)

Operation is resumed.

Remark n = 0, 2, 4, 6
p = n+1
TO00 to TO07 pins

6.8.3 Operation as multiple PWM output function

By extending the PWM function and using two or more slave channels, many PWM output signals can be produced.

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 TDR0n (master) + 1} × Count clock period
Duty factor 1 [%] = {Set value of TDR0p (slave 1)}/{Set value of TDR0n (master) + 1} ×100
Duty factor 2 [%] = {Set value of TDR0p (slave 2)}/{Set value of TDR0n (master) + 1} ×100

Remark Although the duty factor exceeds 100% if the set value of TDR0p (slave 1) > {set value of TDR0n (master) + 1} or if the {set value of TDR0q (slave 2)} > {set value of TDR0n (master) + 1}, it is summarized into 100% output.

TCR0n of the master channel operates in the interval timer mode and counts the periods.

TCR0p of the slave channel 1 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TO0p pin. TCR0p loads the value of TDR0p to TCR0p, using INTTM0n of the master channel as a start trigger, and start counting down. When TCR0p = 0000H, TCR0p outputs INTTM0p and stops counting until the next start trigger (INTTM0n of the master channel) has been input. The output level of TO0p becomes active one count clock after generation of INTTM0n from the master channel, and inactive when TCR0p = 0000H.

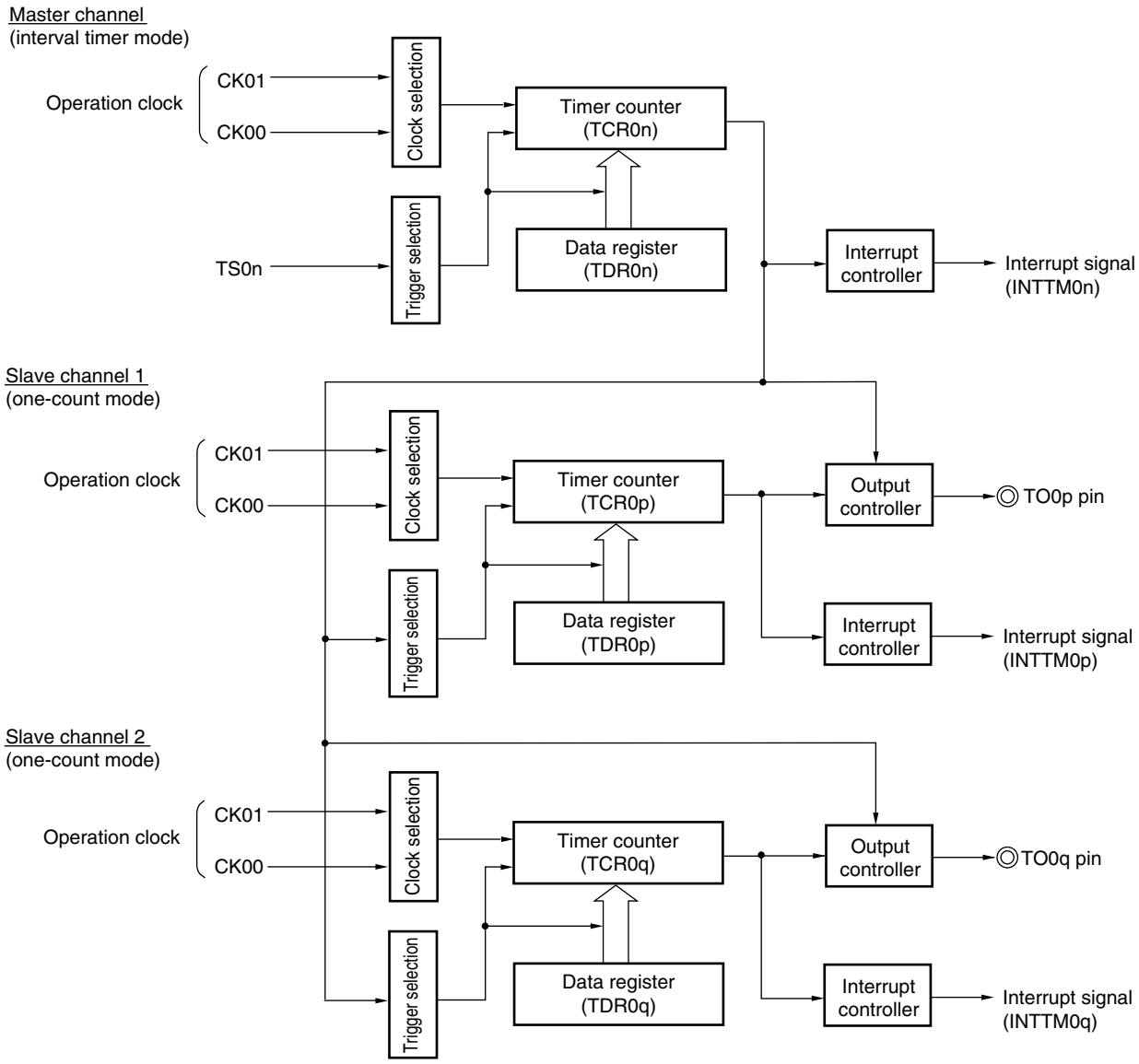
In the same way as TCR0p of the slave channel 1, TCR0q of the slave channel 2 operates in one-count mode, counts the duty factor, and outputs a PWM waveform from the TO0q pin. TCR0q loads the value of TDR0q to TCR0q, using INTTM0n of the master channel as a start trigger, and starts counting down. When TCR0q = 0000H, TCR0q outputs INTTM0q and stops counting until the next start trigger (INTTM0n of the master channel) has been input. The output level of TO0q becomes active one count clock after generation of INTTM0n from the master channel, and inactive when TCR0q = 0000H.

When channel 0 is used as the master channel as described above, up to seven types of PWM signals can be output at the same time with timer array unit 0 and up to three types with timer array unit 1.

Caution To rewrite both TDR0n of the master channel and TDR0p of the slave channel 1, write access is necessary at least twice. Since the values of TDR0n and TDR0p are loaded to TCR0n and TCR0p after INTTM0n is generated from the master channel, if rewriting is performed separately before and after generation of INTTM0n from the master channel, the TO0p pin cannot output the expected waveform. To rewrite both TDR0n of the master and TDR0p of the slave, be sure to rewrite both the registers immediately after INTTM0n is generated from the master channel (This applies also to TDR0q of the slave channel 2).

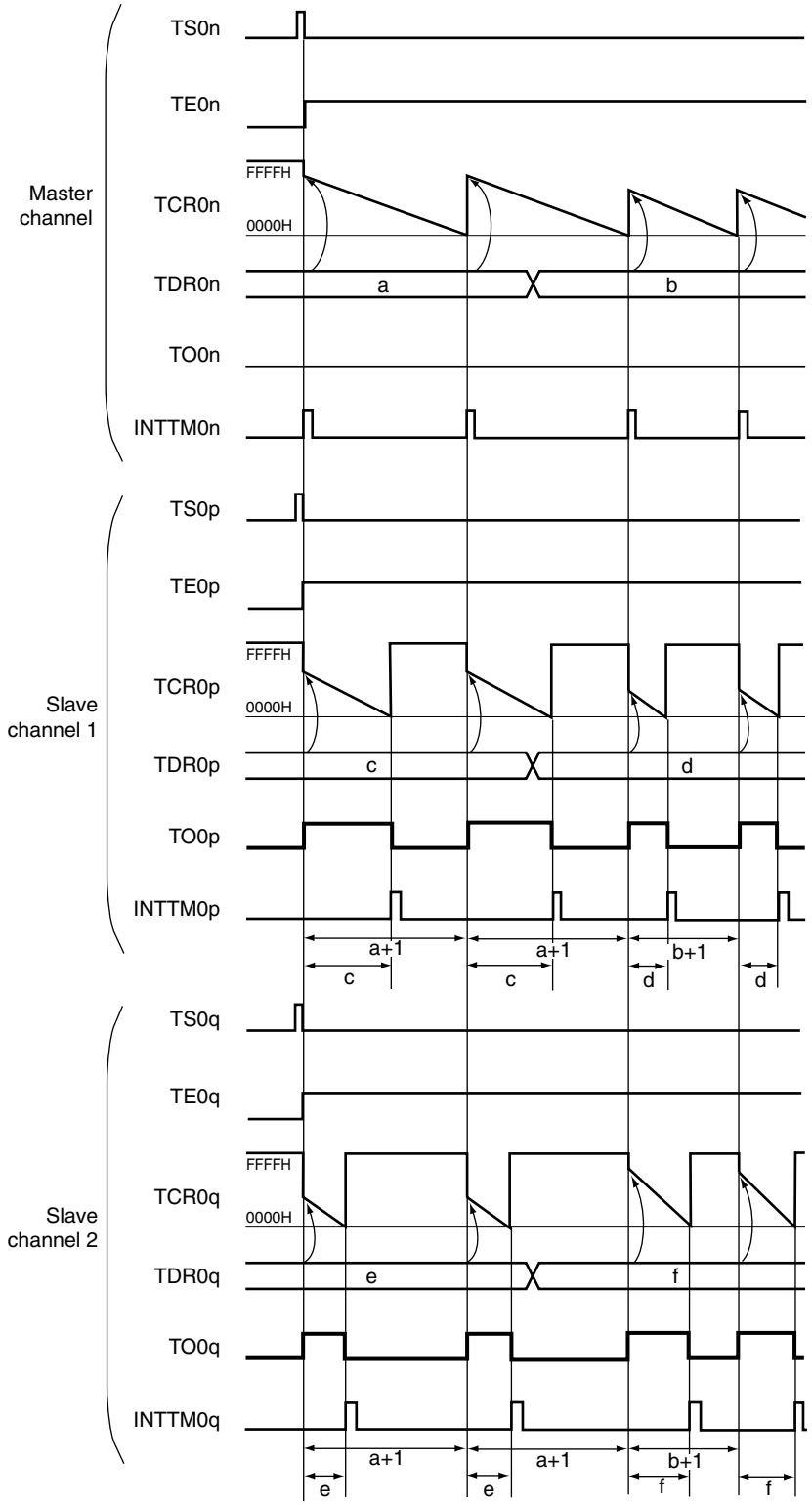
Remark n = 0, 2, 4
p = n+1
q = n+2
TO00 to TO07 pins

Figure 6-66. Block Diagram of Operation as Multiple PWM Output Function (output two types of PWMs)



Remark n = 0, 2, 4
p = n+1
q = n+2
TO00 to TO07 pins

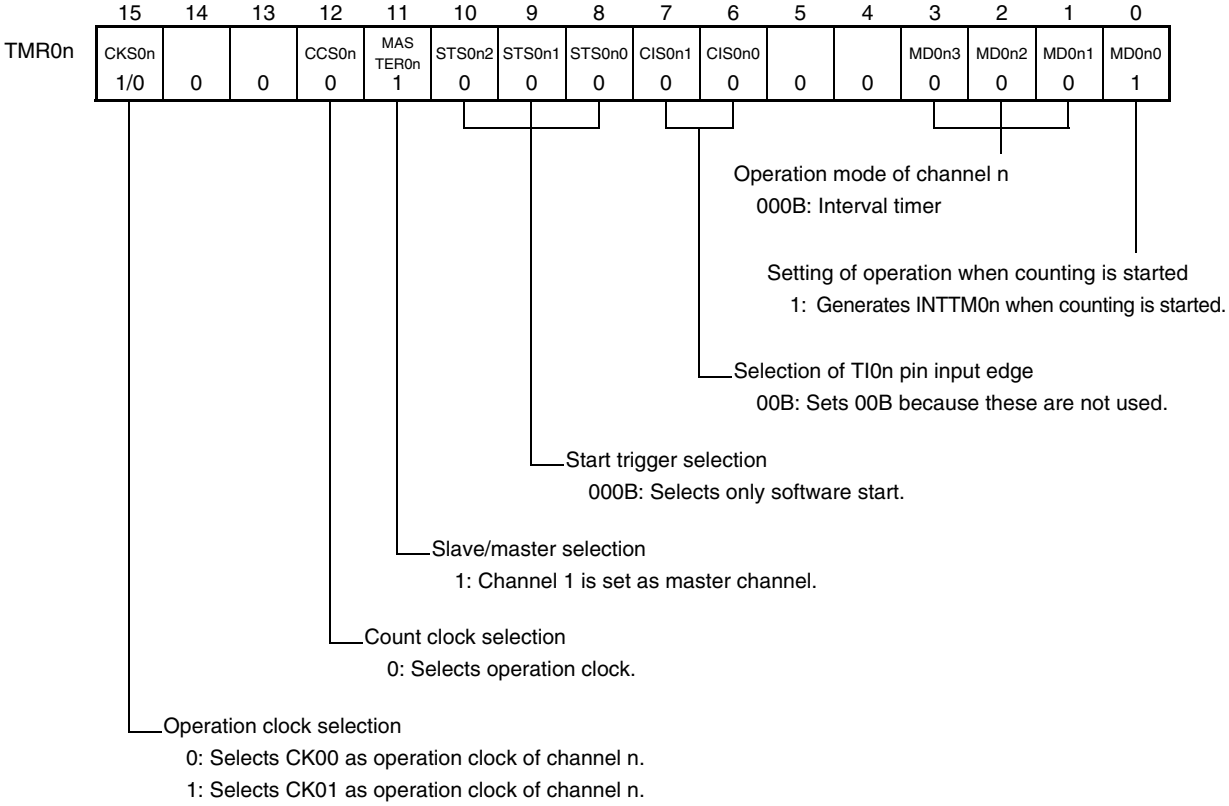
Figure 6-67. Example of Basic Timing of Operation as Multiple PWM Output Function (output two types of PWMs)



Remark n = 0, 2, 4
p = n+1
q = n+2
TO00 to TO07 pins

Figure 6-68. Example of Set Contents of Registers When Multiple PWM Output Function (Master Channel) Is Used

(a) Timer mode register 0n (TMR0n)



(b) Timer output register 0 (TO0)

	Bit n	
TO0	TO0n	0: Outputs 0 from TO0n.
	0	

(c) Timer output enable register 0 (TOE0)

	Bit n	
TOE0	TOE0n	0: Stops the TO0n output operation by counting operation.
	0	

(d) Timer output level register 0 (TOL0)

	Bit n	
TOL0	TOL0n	0: Cleared to 0 when TOM0n = 0 (toggle mode).
	0	

(e) Timer output mode register 0 (TOM0)

	Bit n	
TOM0	TOM0n	0: Sets toggle mode.
	0	

Remark n = 0, 2, 4
TO00 to TO07, TI00 to TI07 pins

Figure 6-69. Example of Set Contents of Registers
When Multiple PWM Output Function (Slave Channel) Is Used (output two types of PWMs)

(a) Timer mode register 0p, 0q (TMR0p, TMR0q)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMR0p	CKS0p			CCS0p	MASTER0p	STS0p2	STS0p1	STS0p0	CIS0p1	CIS0p0			MD0p3	MD0p2	MD0p1	MD0p0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1
TMR0q	CKS0q			CCS0q	MASTER0q	STS0q2	STS0q1	STS0q0	CIS0q1	CIS0q0			MD0q3	MD0q2	MD0q1	MD0q0
	1/0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1

Operation mode of channels p, q
100B: One-count mode

Start trigger during operation
1: Trigger input is valid.

Selection of TI0p and TI0q pin input edge
00B: Sets 00B because these are not used.

Start trigger selection
100B: Selects INTTM0n of master channel.

Slave/master selection
0: Channel 0 is set as slave channel.

Count clock selection
0: Selects operation clock.

Operation clock selection
0: Selects CK00 as operation clock of channels p, q.
1: Selects CK01 as operation clock of channels p, q.
* Make the same setting as master channel.

(b) Timer output register 0 (TO0)

	Bit q	Bit p
TO0	TO0q	TO0p
	1/0	1/0

0: Outputs 0 from TO0p or TO0q.
1: Outputs 1 from TO0p or TO0q.

(c) Timer output enable register 0 (TOE0)

	Bit q	Bit p
TOE0	TOE0q	TOE0p
	1/0	1/0

0: Stops the TO0p or TO0q output operation by counting operation.
1: Enables the TO0p or TO0q output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit q	Bit p
TOL0	TOL0q	TOL0p
	1/0	1/0

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

(e) Timer output mode register 0 (TOM0)

	Bit q	Bit p
TOM0	TOM0q	TOM0p
	1	1

1: Sets the combination operation mode.

Remark n = 0, 2, 4
p = n+1
q = n+2
TO00 to TO07 pins

Figure 6-70. Operation Procedure When Multiple PWM Output Function Is Used (1/2)

	Software Operation	Hardware Status
TAU default setting		Power-off status (Clock supply is stopped and writing to each register is disabled.)
	Sets the TAU0EN bit of the PER0 register to 1. →	Power-on status. Each channel stops operating. (Clock supply is started and writing to each register is enabled.)
	Sets the TPS0 register. Determines clock frequencies of CK00 and CK01.	
Channel default setting	Sets the TMR0n, TMR0p, and TMR0q registers of each channel to be used (determines operation mode of channels). An interval (period) value is set to the TDR0n register of the master channel, and a duty factor is set to the TDR0p and TDR0q register of the slave channel.	Channel stops operating. (Clock is supplied and some power is consumed.)
	Sets slave channel. The TOM0p and TOM0q bits of the TOM0 register are set to 1 (combination operation mode). Clears the TOL0p and TOL0q bits to 0. Sets the TO0p and TO0q bits and determines default level of the TO0p and TO0q outputs. →	The TO0n pin goes into Hi-Z output state. The TO0p and TO0q default setting levels are output when the port mode register is in output mode and the port register is 0.
	Sets TOE0p or TOE0q to 1 and enables operation of TO0p and TO0q. →	TO0p or TO0q does not change because channel stops operating.
	Clears the port register and port mode register to 0. →	The TO0p and TO0q pins output the TO0p and TO0q set levels.

Remark n = 0, 2, 4
p = n+1
q = n+2
TO00 to TO07 pins

Figure 6-70. Operation Procedure When Multiple PWM Output Function Is Used (2/2)

Operation is resumed.

	Software Operation	Hardware Status
Operation start	Sets TOE0p and TOE0q (slave) to 1 (only when operation is resumed). The TS0n bit (master), and TS0p and TS0q (slave) bits of the TS0 register are set to 1 at the same time. →	TE0n = 1, TE0p, TE0q = 1 When the master channel starts counting, INTTM0n is generated. Triggered by this interrupt, the slave channel also starts counting.
	The TS0n, TS0p, and TS0q bits automatically return to 0 because they are trigger bits.	
During operation	Set values of the TMR0n, TMR0p, and TMR0q registers and TOM0n, TOM0p, TOM0q, TOL0n, TOL0p, and TOL0q bits cannot be changed. Set values of the TDR0n, TDR0p, and TDR0q registers can be changed after INTTM0n of the master channel is generated. The TCR0n, TCR0p, and TCR0q registers can always be read. The TSR0n, TSR0p, and TSR0q registers are not used. Set values of the TO0 and TOE0 registers can be changed.	The counter of the master channel loads the TDR0n value to TCR0n and counts down. When the count value reaches TCR0n = 0000H, INTTM0n output is generated. At the same time, the value of the TDR0n register is loaded to TCR0n, and the counter starts counting down again. At the slave channel 1, the values of TDR0p are transferred to TCR0p, triggered by INTTM0n of the master channel, and the counter starts counting down. The output levels of TO0p become active one count clock after generation of the INTTM0n output from the master channel. It becomes inactive when TCR0p = 0000H, and the counting operation is stopped. At the slave channel 2, the values of TDR0q are transferred to TDR0q, triggered by INTTM0n of the master channel, and the counter starts counting down. The output levels of TO0q become active one count clock after generation of the INTTM0n output from the master channel. It becomes inactive when TCR0q = 0000H, and the counting operation is stopped. After that, the above operation is repeated.
Operation stop	The TT0n bit (master), TT0p, and TT0q (slave) bits are set to 1 at the same time. →	TE0n, TE0p, and TE0q = 0, and count operation stops.
	The TT0n, TT0p, and TT0q bits automatically return to 0 because they are trigger bits.	TCR0n, TCR0p and TCR0q hold count value and stops. The TO0p and TO0q outputs are not initialized but hold current status.
	TOE0p or TOE0q of slave channel is cleared to 0 and value is set to the TO0p and TO0q registers. →	The TO0p and TO0q pins output the TO0p and TO0q set levels.
TAU stop	To hold the TO0p and TO0q pins output levels Clears TO0p and TO0q bits to 0 after the value to be held is set to the port register. →	The TO0p and TO0q pins output levels are held by port function.
	When holding the TO0p and TO0q pins output levels is not necessary Switches the port mode register to input mode. →	The TO0p and TO0q pins output levels go into Hi-Z output state.
	Clears the TAU0EN bit of the PER0 register to 0. →	Power-off status All circuits are initialized and SFR of each channel is also initialized. (The TO0p and TO0q bits are cleared to 0 and the TO0p and TO0q pins are set to port mode.)

Remark n = 0, 2, 4
p = n+1
q = n+2
TO00 to TO07 pins

CHAPTER 7 REAL-TIME COUNTER

7.1 Functions of Real-Time Counter

The real-time counter 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: 1 month to 0.5 seconds)
- Alarm interrupt function (alarm: week, hour, minute)
- Interval interrupt function
- Pin output function of 1 Hz
- Pin output function of 512 Hz or 16.384 kHz or 32.768 kHz

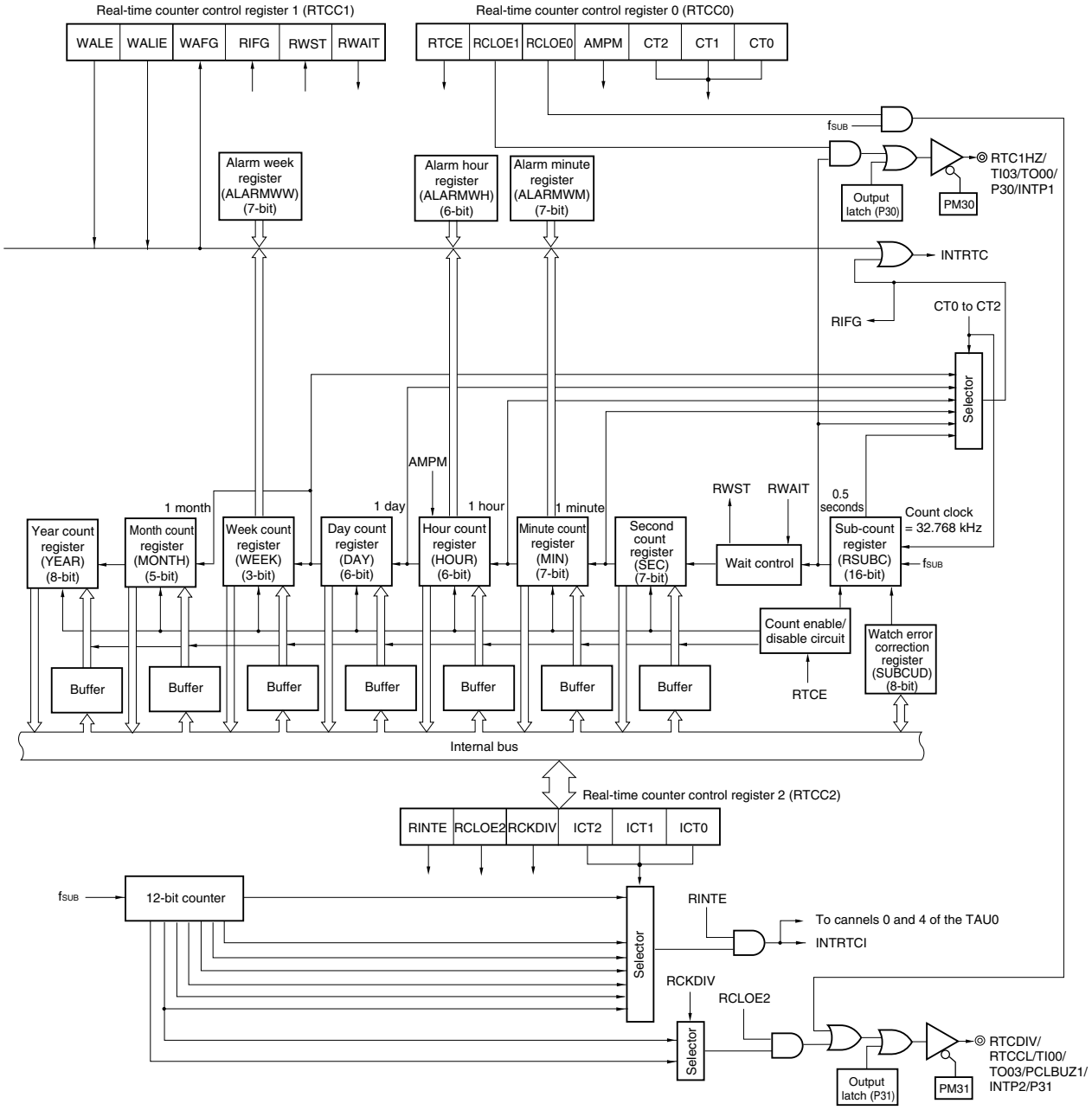
7.2 Configuration of Real-Time Counter

The real-time counter includes the following hardware.

Table 7-1. Configuration of Real-Time Counter

Item	Configuration
Control registers	Peripheral enable register 0 (PER0) Real-time counter control register 0 (RTCC0) Real-time counter control register 1 (RTCC1) Real-time counter control register 2 (RTCC2) Sub-count register (RSUBC) Second count register (SEC) Minute count register (MIN) Hour count register (HOUR) Day count register (DAY) Week count register (WEEK) Month count register (MONTH) Year count register (YEAR) Watch error correction register (SUBCUD) Alarm minute register (ALARMWM) Alarm hour register (ALARMWH) Alarm week register (ALARMWW) Port mode register 3 (PM3) Port register 3 (P3)

Figure 7-1. Block Diagram of Real-Time Counter



7.3 Registers Controlling Real-Time Counter

Timer real-time counter is controlled by the following 18 registers.

- Peripheral enable register 0 (PER0)
- Real-time counter control register 0 (RTCC0)
- Real-time counter control register 1 (RTCC1)
- Real-time counter control register 2 (RTCC2)
- Sub-count register (RSUBC)
- Second count register (SEC)
- Minute count register (MIN)
- Hour count register (HOUR)
- Day count register (DAY)
- Week count register (WEEK)
- Month count register (MONTH)
- Year count register (YEAR)
- Watch error correction register (SUBCUD)
- Alarm minute register (ALARMWM)
- Alarm hour register (ALARMWH)
- Alarm week register (ALARMWW)
- Port mode register 3 (PM3)
- Port register 3 (P3)

(1) Peripheral enable register 0 (PER0)

PER0 is used to enable or disable use of each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the real-time counter is used, be sure to set bit 7 (RTCEN) of this register to 1.

PER0 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 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	DACEN	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

RTCEN	Control of real-time counter (RTC) input clock[Note]
0	Stops supply of input clock. • SFR used by the real-time counter (RTC) cannot be written. • The real-time counter (RTC) is in the reset status.
1	Supplies input clock. • SFR used by the real-time counter (RTC) can be read/written.

Note By using RTCEN, can supply and stop the clock that is used when accessing the real-time counter (RTC) from the CPU. RTCEN cannot control supply of the operating clock (f_{SUB}) to RTC.

Cautions 1. When using the real-time counter, first set RTCEN to 1, while oscillation of the subsystem clock (f_{SUB}) is stable. If RTCEN = 0, writing to a control register of the real-time counter is ignored, and, even if the register is read, only the default value is read.

2. Clock supply to peripheral functions except the real-time counter can be stopped in the HALT mode when operating on the subsystem clock by setting RTCLPC of the operation speed mode control register (OSMC) to 1. In this case, set RTCEN to 1 and bits 0 to 6 of PER0 to 0.

(2) Real-time counter control register 0 (RTCC0)

The RTCC0 register is an 8-bit register that is used to start or stop the real-time counter operation, control the RTCCL and RTC1HZ pins, and set a 12- or 24-hour system and the constant-period interrupt function.

RTCC0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-3. Format of Real-Time Counter Control Register 0 (RTCC0)

Address: FFF9DH After reset: 00H R/W

Symbol	<7>	6	<5>	<4>	3	2	1	0
RTCC0	RTCE	0	RCLOE1	RCLOE0	AMPM	CT2	CT1	CT0

RTCE	Real-time counter operation control
0	Stops counter operation.
1	Starts counter operation.

RCLOE1	RTC1HZ pin output control
0	Disables output of RTC1HZ pin (1 Hz).
1	Enables output of RTC1HZ pin (1 Hz).

RCLOE0[Note]	RTCCL pin output control
0	Disables output of RTCCL pin (32.768 kHz).
1	Enables output of RTCCL pin (32.768 kHz).

AMPM	Selection of 12-/24-hour system
0	12-hour system (a.m. and p.m. are displayed.)
1	24-hour system
• To change the value of AMPM, set RWAIT (bit 0 of RTCC1) to 1, and re-set the hour count register (HOUR). • Table 7-2 shows the displayed time digits that are displayed.	

CT2	CT1	CT0	Constant-period interrupt (INTRTC) selection
0	0	0	Does not use constant-period interrupt function.
0	0	1	Once per 0.5 s (synchronized with second count up)
0	1	0	Once per 1 s (same time as second count up)
0	1	1	Once per 1 m (second 00 of every minute)
1	0	0	Once per 1 hour (minute 00 and second 00 of every hour)
1	0	1	Once per 1 day (hour 00, minute 00, and second 00 of every day)
1	1	×	Once per 1 month (Day 1, hour 00 a.m., minute 00, and second 00 of every month)
When changing the values of CT2 to CT0 while the counter operates (RTCE = 1), rewrite the values of CT2 to CT0 after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, after rewriting the values of CT2 to CT0, enable interrupt servicing after clearing the RIFG and RTCIF flags.			

Note RCLOE0 and RCLOE2 must not be enabled at the same time.

Caution If RCLOE0 and RCLOE1 are changed when RTCE = 1, the last waveform of the 32.768 kHz and 1 Hz output signals may become short.

Remark ×: don't care

(3) Real-time counter control register 1 (RTCC1)

The RTCC1 register is an 8-bit register that is used to control the alarm interrupt function and the wait time of the counter.

RTCC1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-4. Format of Real-Time Counter 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	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 RTCC1, the ALARMWM register, the ALARMWH register, and the ALARMWW register), 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 clock (32.768 kHz) after matching of the alarm is detected. This flag is cleared when “0” is written to it. Writing “1” to it is invalid.	

Figure 7-4. Format of Real-Time Counter Control Register 1 (RTCC1) (2/2)

RIFG	Constant-period interrupt status flag
0	Constant-period interrupt is not generated.
1	Constant-period interrupt is generated.
This flag indicates the status of generation of the constant-period interrupt. When the constant-period interrupt is generated, it is set to “1”. This flag is cleared when “0” is written to it. Writing “1” to it is invalid.	

RWST	Wait status flag of real-time counter
0	Counter is operating.
1	Mode to read or write counter value
This status flag indicates whether the setting of RWAIT is valid. Before reading or writing the counter value, confirm that the value of this flag is 1.	

RWAIT	Wait control of real-time counter
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. Because RSUBC continues operation, complete reading or writing of it in 1 second, and clear this bit back to 0. When RWAIT = 1, it takes up to 1 clock (32.768 kHz) until the counter value can be read or written. If RSUBC overflows when RWAIT = 1, it counts up after RWAIT = 0. If the second count register is written, however, it does not count up because RSUBC is cleared.	

Caution If writing is performed to the RTCC1 register with a 1-bit manipulation instruction, the RIFG and WAFG flags may be cleared. Therefore, to perform writing to the RIFG and WAFG flags, be sure to use an 8-bit manipulation instruction. At this time, set 1 to the RIFG and WAFG flags to invalidate writing and not to clear the RIFG and WAFG flags during writing. When the value may be rewritten because the RIFG and WAFG flags are not being used, the RTCC1 register may be written by using a 1-bit manipulation instruction.

Remark 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.

(4) Real-time counter control register 2 (RTCC2)

The RTCC2 register is an 8-bit register that is used to control the interval interrupt function and the RTCDIV pin.

RTCC2 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 7-5. Format of Real-Time Counter Control Register 2 (RTCC2)

Address: FFF9FH After reset: 00H R/W

Symbol	<7>	<6>	<5>	4	3	2	1	0
RTCC2	RINTE	RCLOE2	RCKDIV	0	0	ICT2	ICT1	ICT0

RINTE	ICT2	ICT1	ICT0	Interval interrupt (INTRTCI) selection
0	×	×	×	Interval interrupt is not generated.
1	0	0	0	$2^6/f_{SUB}$ (1.953125 ms)
1	0	0	1	$2^7/f_{SUB}$ (3.90625 ms)
1	0	1	0	$2^8/f_{SUB}$ (7.8125 ms)
1	0	1	1	$2^9/f_{SUB}$ (15.625 ms)
1	1	0	0	$2^{10}/f_{SUB}$ (31.25 ms)
1	1	0	1	$2^{11}/f_{SUB}$ (62.5 ms)
1	1	1	×	$2^{12}/f_{SUB}$ (125 ms)

RCLOE2[Note]	RTCDIV pin output control
0	Disables output of RTCDIV pin
1	Enables output of RTCDIV pin

RCKDIV	Selection of RTCDIV pin output frequency
0	RTCDIV pin outputs 512 Hz (1.95 ms).
1	RTCDIV pin outputs 16.384 kHz (0.061 ms).

Note RCLOE0 and RCLOE2 must not be enabled at the same time.

Cautions 1. Change ICT2, ICT1, and ICT0 when RINTE = 0.

2. When the output from RTCDIV pin is stopped, the output continues after a maximum of two clocks of f_{XT} and enters the low level. While 512 Hz is output, and when the output is stopped immediately after entering the high level, a pulse of at least one clock width of f_{SUB} may be generated.

3. After the real-time counter starts operating, the output width of the RTCDIV pin may be shorter than as set during the first interval period.

Remark f_{SUB}: Subsystem clock frequency

(5) Sub-count register (RSUBC)

The RSUBC register is a 16-bit register that counts the reference time of 1 second of the real-time counter. It takes a value of 0000H to 7FFFH and counts 1 second with a clock of 32.768 kHz.

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

Reset signal generation clears this register to 0000H.

Cautions 1. When a correction is made by using the SUBCUD register, the value may become 8000H or more.

2. This register is also cleared by reset effected by writing the second count register.

3. The value read from this register is not guaranteed if it is read during operation, because a value that is changing is read.

Figure 7-6. Format of Sub-Count Register (RSUBC)

Address: FFF90H After reset: 0000H R

Symbol	7	6	5	4	3	2	1	0
RSUBC	SUBC7	SUBC6	SUBC5	SUBC4	SUBC3	SUBC2	SUBC1	SUBC0

Address: FFF91H After reset: 0000H R

Symbol	7	6	5	4	3	2	1	0
RSUBC	SUBC15	SUBC14	SUBC13	SUBC12	SUBC11	SUBC10	SUBC9	SUBC8

(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 sub-counter overflows.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Set a decimal value of 00 to 59 to this register in BCD code.

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

Reset signal generation clears this register to 00H.

Figure 7-7. 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

(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 2 clocks (32.768 kHz) 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.

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

Reset signal generation clears this register to 00H.

Figure 7-8. 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

(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 2 clocks (32.768 kHz) 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. Set a decimal value of 00 to 23, or 01 to 12 and 21 to 32 to this register in BCD code.

HOUR 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 7-9. 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 Bit 5 (HOUR20) of HOUR indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

Table 7-2. Displayed Time Digits

24-Hour Display (AMPM bit = 1)		12-Hour Display (AMPM bit = 0)	
Time	HOUR Register	Time	HOUR Register
0	00H	0 a.m.	12H
1	01H	1 a.m.	01H
2	02H	2 a.m.	02H
3	03H	3 a.m.	03H
4	04H	4 a.m.	04H
5	05H	5 a.m.	05H
6	06H	6 a.m.	06H
7	07H	7 a.m.	07H
8	08H	8 a.m.	08H
9	09H	9 a.m.	09H
10	10H	10 a.m.	10H
11	11H	11 a.m.	11H
12	12H	0 p.m.	32H
13	13H	1 p.m.	21H
14	14H	2 p.m.	22H
15	15H	3 p.m.	23H
16	16H	4 p.m.	24H
17	17H	5 p.m.	25H
18	18H	6 p.m.	26H
19	19H	7 p.m.	27H
20	20H	8 p.m.	28H
21	21H	9 p.m.	29H
22	22H	10 p.m.	30H
23	23H	11 p.m.	31H

The HOUR register value is set to 12-hour display when the AMPM bit is “0” and to 24-hour display when the AMPM bit is “1”.

In 12-hour display, the fifth bit of the HOUR register displays 0 for AM and 1 for PM.

(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 2 clocks (32.768 kHz) 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.
DAY can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 01H.

Figure 7-10. 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

(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 2 clocks (32.768 kHz) later. Set a decimal value of 00 to 06 to this register in BCD code.
WEEK can be set by an 8-bit memory manipulation instruction.
Reset signal generation clears this register to 00H.

Figure 7-11. 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 The value corresponding to the month count register or the day count register is not stored in the week count register automatically. After reset release, set the week count register as follow.

Day	WEEK
Sunday	00H
Monday	01H
Tuesday	02H
Wednesday	03H
Thursday	04H
Friday	05H
Saturday	06H

(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 2 clocks (32.768 kHz) 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. Set a decimal value of 01 to 12 to this register in BCD code.

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

Reset signal generation clears this register to 01H.

Figure 7-12. 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

(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 counter overflows.

Values 00, 04, 08, …, 92, and 96 indicate a leap year.

When data is written to this register, it is written to a buffer and then to the counter up to 2 clocks (32.768 kHz) later. Even if the month 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 99 to this register in BCD code. Set a decimal value of 00 to 99 to this register in BCD code.

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

Reset signal generation clears this register to 00H.

Figure 7-13. 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

(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 (reference value: 7FFFH) that overflows from the sub-count register (RSUBC) to the second count register.

Rewrite the SUBCUD register after disabling interrupt servicing INTRTC by using the interrupt mask flag register. Furthermore, after rewriting the SUBCUD register, enable interrupt servicing after clearing the interrupt request flag (RTCIF) and constant-period interrupt status flag (RIFG).

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

Reset signal generation clears this register to 00H.

Figure 7-14. 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	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	

F6	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 (F6, F5, F4, F3, F2, F1, F0) = (*, 0, 0, 0, 0, 0, *), the watch error is not corrected. * is 0 or 1. /F5 to /F0 are the inverted values of the corresponding bits (000011 when 111100). Range of correction value: (when F6 = 0) 2, 4, 6, 8, … , 120, 122, 124 (when F6 = 1) −2, −4, −6, −8, … , −120, −122, −124	

The range of value that can be corrected by using the watch error correction register (SUBCUD) is shown below.

	DEV = 0 (correction every 20 seconds)	DEV = 1 (correction every 60 seconds)
Correctable range	−189.2 ppm to 189.2 ppm	−63.1 ppm to 63.1 ppm
Maximum excludes quantization error	±1.53 ppm	±0.51 ppm
Minimum resolution	±3.05 ppm	±1.02 ppm

Remark If a correctable range is –63.1 ppm or lower and 63.1 ppm or higher, set 0 to DEV.

(14) Alarm minute register (ALARMWM)

This register is used to set minutes of alarm.

ALARMWM 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 7-15. 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

(15) Alarm hour register (ALARMWH)

This register is used to set hours of alarm.

ALARMWH 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.

Caution Set a decimal value of 00 to 23, or 01 to 12 and 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.

Figure 7-16. 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 ALARMWH indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).

(16) Alarm week register (ALARMWW)

This register is used to set date of alarm.

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

Reset signal generation clears this register to 00H.

Figure 7-17. 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 WW0	Monday WW1	Tuesday WW2	Wednesday WW3	Thursday WW4	Friday WW5	Saturday WW6	Hour 10	Hour 1	Minute 10	Minute 1	Hour 10	Hour 1	Minute 10	Minute 1
Every day, 0:00 a.m.	1	1	1	1	1	1	1	1	2	0	0	0	0	0	0
Every day, 1:30 a.m.	1	1	1	1	1	1	1	0	1	3	0	0	1	3	0
Every day, 11:59 a.m.	1	1	1	1	1	1	1	1	1	5	9	1	1	5	9
Monday through Friday, 0:00 p.m.	0	1	1	1	1	1	0	3	2	0	0	1	2	0	0
Sunday, 1:30 p.m.	1	0	0	0	0	0	0	2	1	3	0	1	3	3	0
Monday, Wednesday, Friday, 11:59 p.m.	0	1	0	1	0	1	0	3	1	5	9	2	3	5	9

(17) Port mode register 3 (PM3)

This register sets port 3 input/output in 1-bit units.

When using the P30/RTC1HZ/TO00/TI03/INTP1 pin for real-time counter correction clock output, the P31/RTCDIV/RTCCL/TI00/TO03/PCLBUZ1/INTP2 pin for real-time counter clock output, set PM30, PM31 and the output latches of P30, P31 to 0.

PM3 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets PM3 to FFH.

Figure 7-18. Format of Port Mode Register 3 (PM3)

Address: FFF23H After reset: FFH R/W

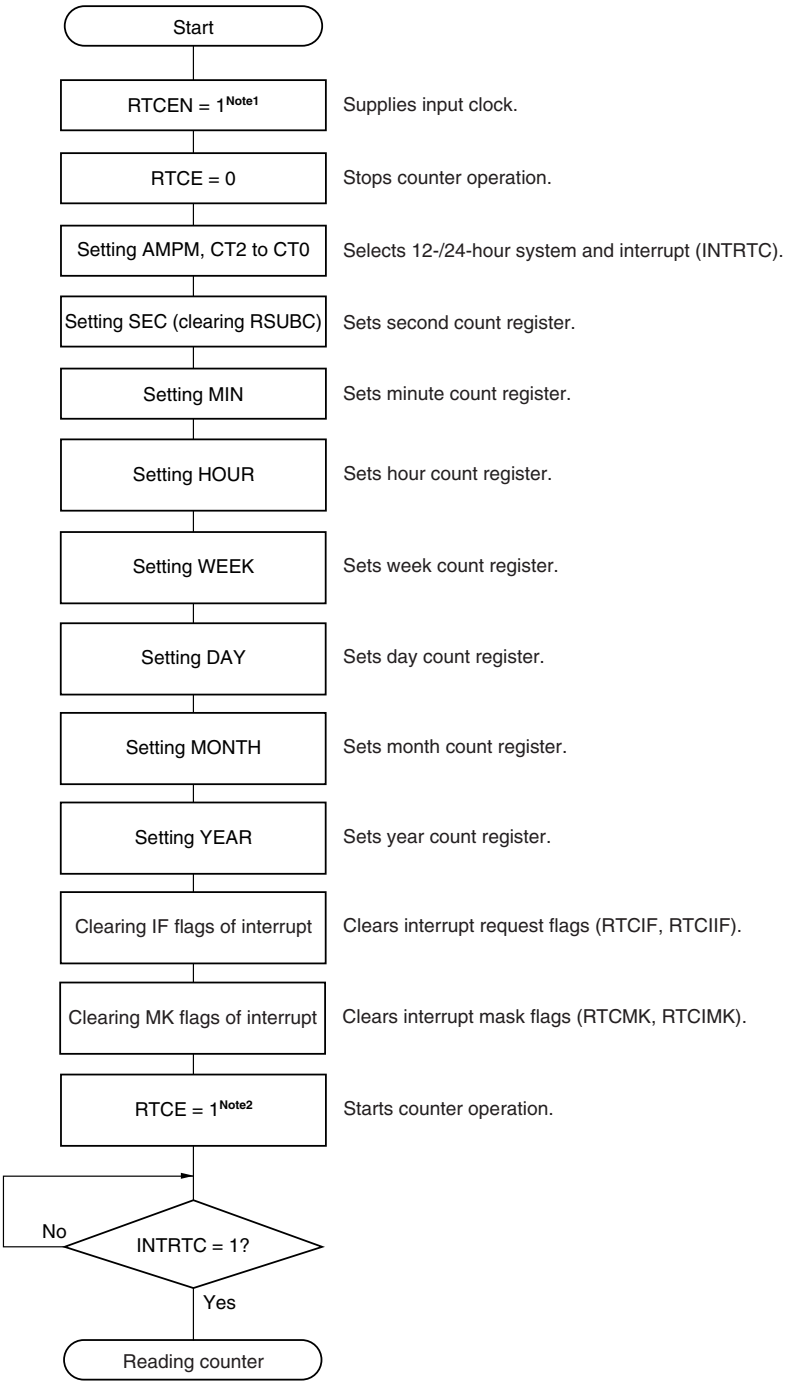
Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	PM34	PM33	PM32	PM31	PM30

PM3n	P3n pin I/O mode selection (n = 0 to 4)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

7.4 Operations of Real-Time Counter

7.4.1 Starting operation of real-time counter

Figure 7-19. Procedure for Starting Operation of Real-Time Counter



Notes 1. First set RTCEN to 1, while oscillation of the subsystem clock (f_{SUB}) is stable.

2. Confirm the procedure described in **7.4.2 Shifting to STOP mode after starting operation** when shifting to STOP mode without waiting for INTRTC = 1 after RTCE = 1.

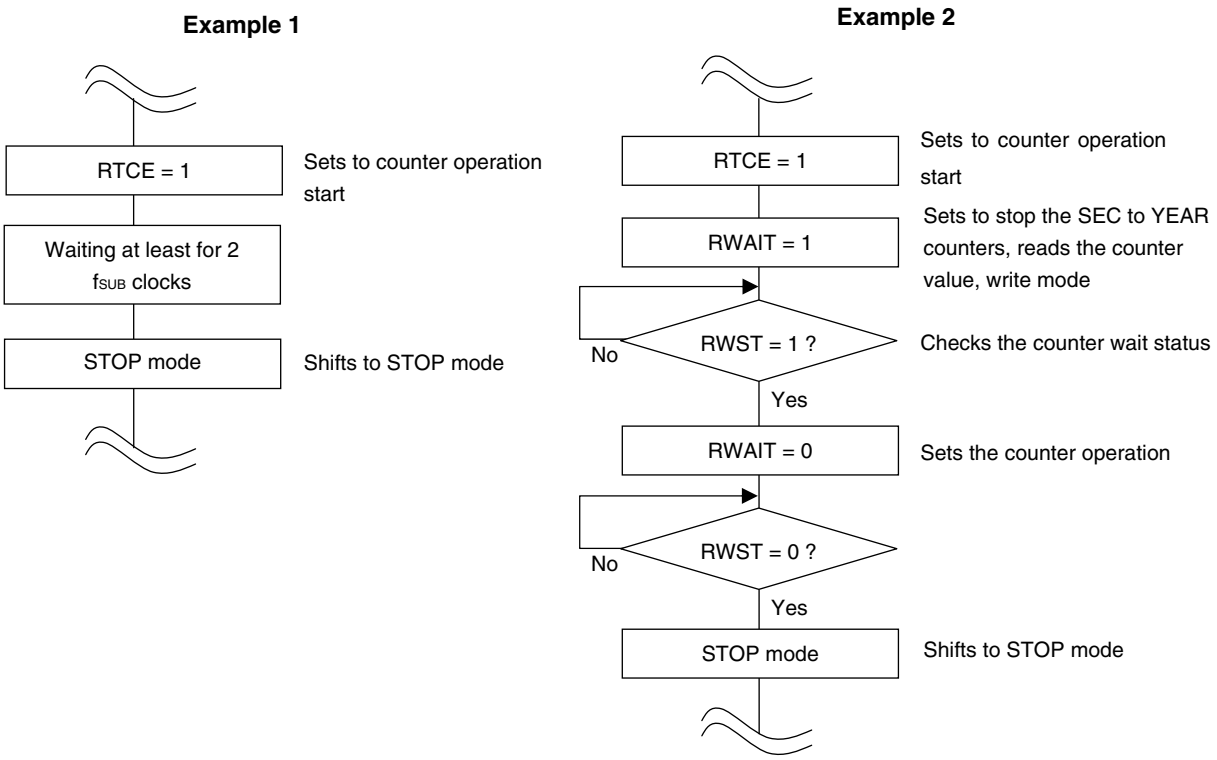
7.4.2 Shifting to STOP mode after starting operation

Perform one of the following processing when shifting to STOP mode immediately after setting RTCE to 1.

However, after setting RTCE to 1, this processing is not required when shifting to STOP mode after the first INTRTC interrupt has occurred.

- Shifting to STOP mode when at least two subsystem clocks (f_{SUB}) (about 62 μs) have elapsed after setting RTCE to 1 (see **Figure 7-20**, **Example 1**).
- Checking by polling RWST to become 1, after setting RTCE to 1 and then setting RWAIT to 1. Afterward, setting RWAIT to 0 and shifting to STOP mode after checking again by polling that RWST has become 0 (see **Figure 7-20**, **Example 2**).

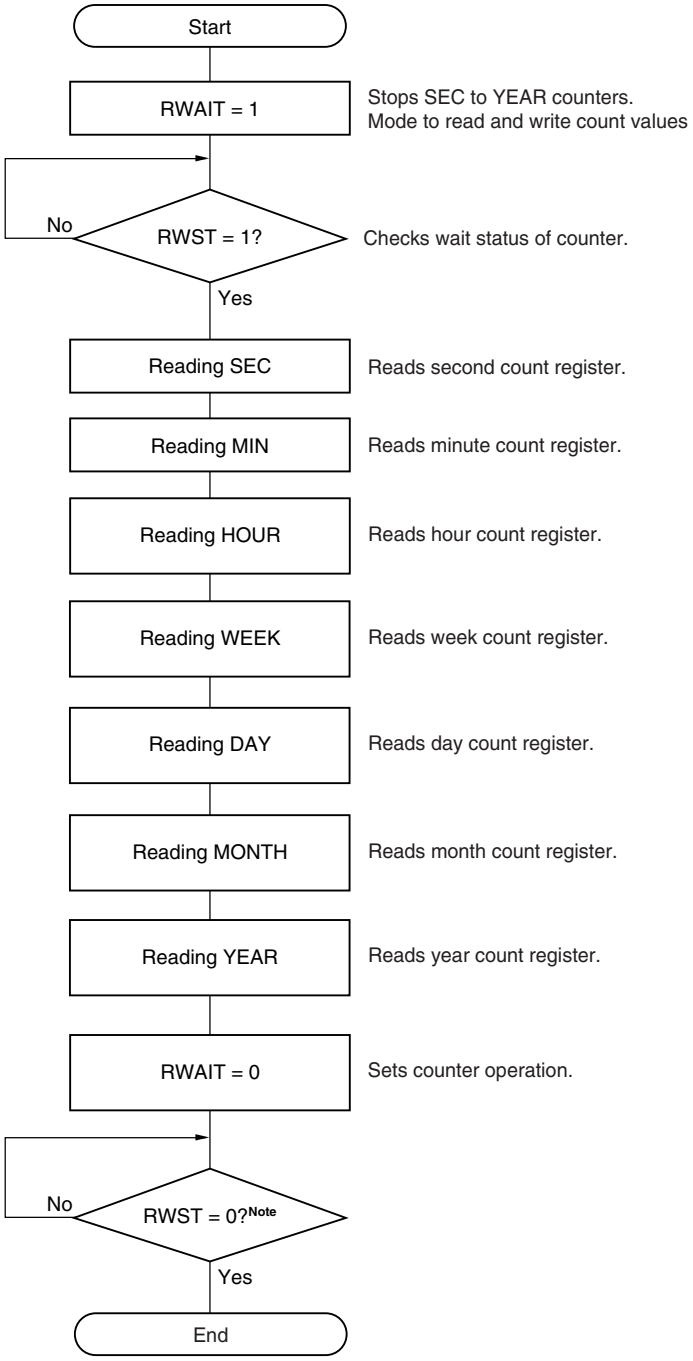
Figure 7-20. Procedure for Shifting to STOP Mode After Setting RTCE to 1



7.4.3 Reading/writing real-time counter

Read or write the counter after setting 1 to RWAIT first.

Figure 7-21. Procedure for Reading Real-Time Counter

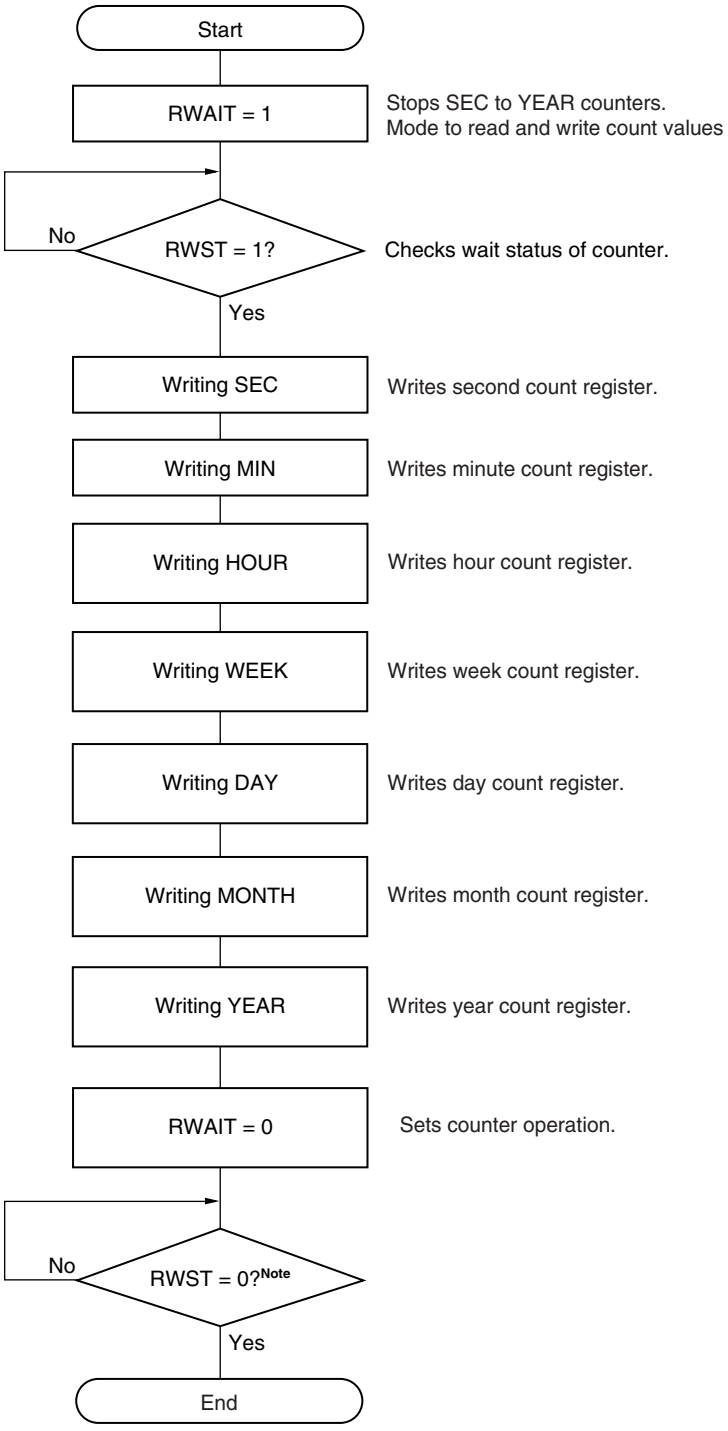


Note Be sure to confirm that RWST = 0 before setting STOP mode.

Caution Complete the series of operations of setting RWAIT to 1 to clearing RWAIT to 0 within 1 second.

Remark SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR may be read in any sequence.
All the registers do not have to be set and only some registers may be read.

Figure 7-22. Procedure for Writing Real-Time Counter



Note Be sure to confirm that RWST = 0 before setting STOP mode.

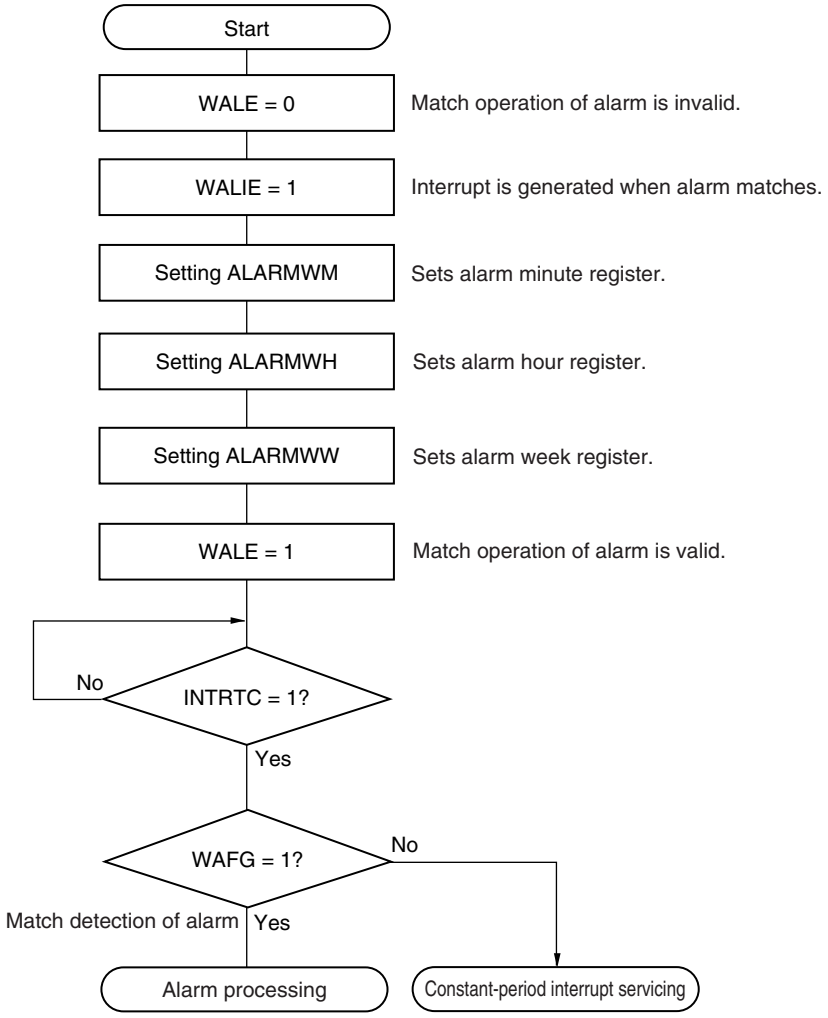
Caution Complete the series of operations of setting RWAIT to 1 to clearing RWAIT to 0 within 1 second.

Remark SEC, MIN, HOUR, WEEK, DAY, MONTH, and YEAR may be written in any sequence.
All the registers do not have to be set and only some registers may be written.

7.4.4 Setting alarm of real-time counter

Set time of alarm after setting 0 to WALE first.

Figure 7-23. Alarm Setting Procedure

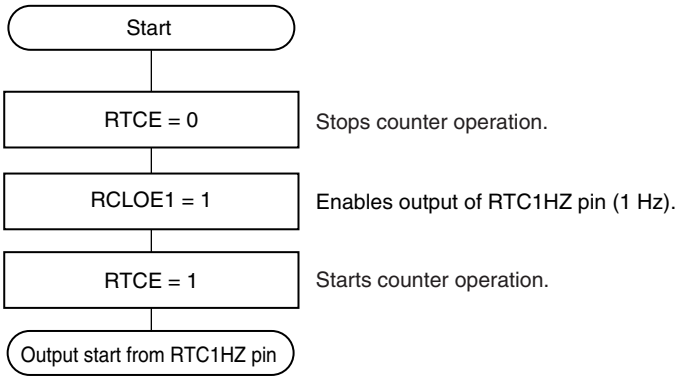


Remarks 1. ALARMWM, ALARMWH, and ALARMWW may be written in any sequence.

2. Fixed-cycle interrupts and alarm match interrupts use the same interrupt source (INTRTC). When using these two types of interrupts at the same time, which interrupt occurred can be judged by checking the fixed-cycle interrupt status flag (RIFG) and the alarm detection status flag (WAFG) upon INTRTC occurrence.

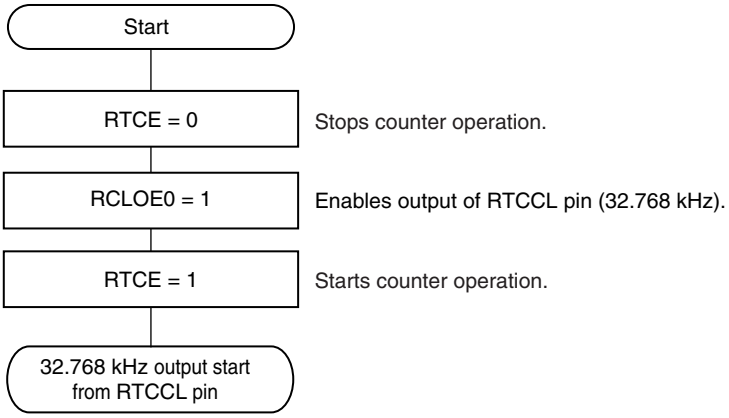
7.4.5 1 Hz output of real-time counter

Figure 7-24. 1 Hz Output Setting Procedure



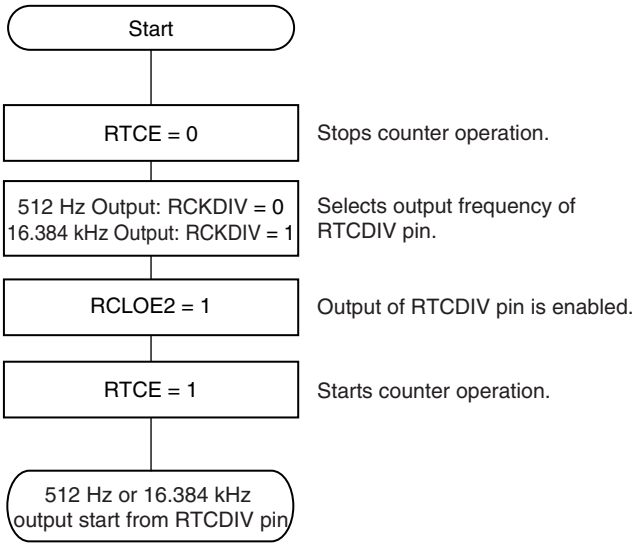
7.4.6 32.768 kHz output of real-time counter

Figure 7-25. 32.768 kHz Output Setting Procedure



7.4.7 512 Hz or 16.384 kHz output of real-time counter

Figure 7-26. 512 Hz or 16.384 kHz Output Setting Procedure



7.4.8 Example of watch error correction of real-time counter

The watch can be corrected with high accuracy when it is slow or fast, by setting a value to the watch error correction register.

Example of calculating the correction value

The correction value used when correcting the count value of the sub-count register (RSUBC) is calculated by using the following expression.

Set DEV to 0 when the correction range is –63.1 ppm or less, or 63.1 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 6 to 0 of the watch error correction register (SUBCUD).

(When F6 = 0) Correction value = {(F5, F4, F3, F2, F1, F0) – 1} × 2

(When F6 = 1) Correction value = – {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2

When (F6, F5, F4, F3, F2, F1, F0) is (*, 0, 0, 0, 0, 0, *), watch error correction is not performed. "*" is 0 or 1.

/F5 to /F0 are bit-inverted values (000011 when 111100).

Remarks 1. The correction value is 2, 4, 6, 8, … 120, 122, 124 or –2, –4, –6, –8, … –120, –122, –124.

2. The oscillation frequency is the subsystem clock (f_{SUB}).

It can be calculated from the 32.768 kHz output frequency of the RTCCL pin or the output frequency of the RTC1HZ pin × 32768 when the watch error correction register is set to its initial value (00H).

3. The target frequency is the frequency resulting after correction performed by using the watch error correction register.

Correction example <1>

Example of correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz − 131.2 ppm)

[Measuring the oscillation frequency]

The oscillation frequency[Note] of each product is measured by outputting about 32 kHz from the RTCCL pin or outputting about 1 Hz from the RTC1HZ pin when the watch error correction register is set to its initial value (00H).

Note See **7.4.5 1 Hz output of real-time counter** for the setting procedure of outputting about 1 Hz from the RTC1HZ pin, and **7.4.6 32.768 kHz output of real-time counter** for the setting procedure of outputting about 32 kHz from the RTCCL pin.

[Calculating the correction value]

(When the output frequency from the RTCCL pin is 32772.3 Hz)

If the target frequency is assumed to be 32768 Hz (32772.3 Hz − 131.2 ppm), the correction range for −131.2 ppm is −63.1 ppm or less, so assume DEV to be 0.

The expression for calculating the correction value when DEV is 0 is applied.

Correction value = Number of correction counts in 1 minute ÷ 3
= (Oscillation frequency ÷ Target frequency − 1) × 32768 × 60 ÷ 3
= (32772.3 ÷ 32768 − 1) × 32768 × 60 ÷ 3
= 86

[Calculating the values to be set to (F6 to F0)]

(When the correction value is 86)

If the correction value is 0 or more (when delaying), assume F6 to be 0.

Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

{(F5, F4, F3, F2, F1, F0) − 1} × 2 = 86
(F5, F4, F3, F2, F1, F0) = 44
(F5, F4, F3, F2, F1, F0) = (1, 0, 1, 1, 0, 0)

Consequently, when correcting from 32772.3 Hz to 32768 Hz (32772.3 Hz − 131.2 ppm), setting the correction register such that DEV is 0 and the correction value is 86 (bits 6 to 0 of SUBCUD: 0101100) results in 32768 Hz (0 ppm).

Figure 7-27 shows the operation when (DEV, F6, F5, F4, F3, F2, F1, F0) is (0, 0, 1, 0, 1, 1, 0, 0).

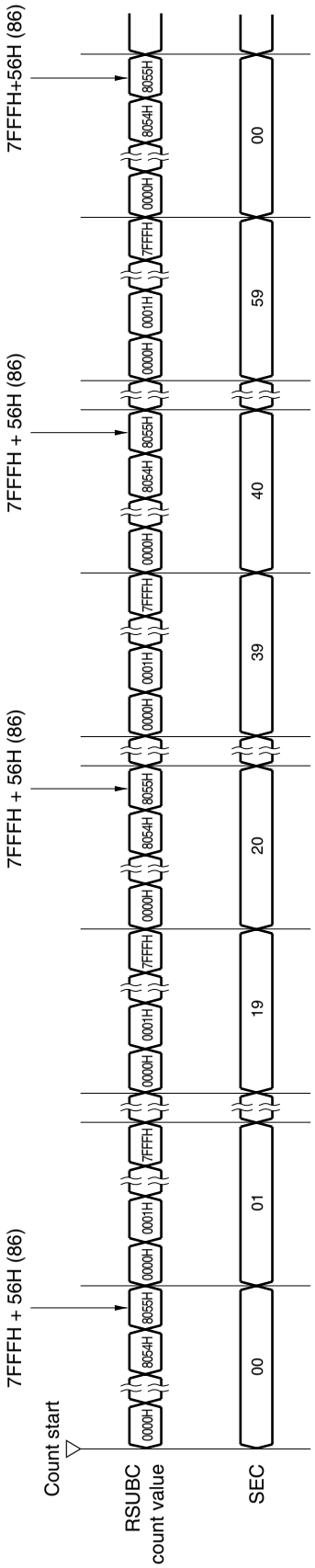


Figure 7-27. Operation When (DEV, F6, F5, F4, F3, F2, F1, F0) = (0, 0, 1, 0, 1, 1, 0, 0)

Correction example <2>

Example of correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm)

[Measuring the oscillation frequency]

The oscillation frequency[Note] of each product is measured by outputting about 32 kHz from the RTCCL pin or outputting about 1 Hz from the RTC1HZ pin when the watch error correction register is set to its initial value (00H).

Note See **7.4.5 1 Hz output of real-time counter** for the setting procedure of outputting about 1 Hz from the RTC1HZ pin, and **7.4.6 32.768 kHz output of real-time counter** for the setting procedure of outputting about 32 kHz from the RTCCL pin.

[Calculating the correction value]

(When the output frequency from the RTC1Hz pin is 0.9999817 Hz)

Oscillation frequency = 32768 × 0.9999817 ≈ 32767.4 Hz

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 (F6 to F0)]

(When the correction value is −36)

If the correction value is 0 or less (when speeding up), assume F6 to be 1.

Calculate (F5, F4, F3, F2, F1, F0) from the correction value.

− {(/F5, /F4, /F3, /F2, /F1, /F0) + 1} × 2	= −36
(/F5, /F4, /F3, /F2, /F1, /F0)	= 17
(/F5, /F4, /F3, /F2, /F1, /F0)	= (0, 1, 0, 0, 0, 1)
(F5, F4, F3, F2, F1, F0)	= (1, 0, 1, 1, 1, 0)

Consequently, when correcting from 32767.4 Hz to 32768 Hz (32767.4 Hz + 18.3 ppm), setting the correction register such that DEV is 1 and the correction value is −36 (bits 6 to 0 of SUBCUD: 1101110) results in 32768 Hz (0 ppm).

Figure 7-28 shows the operation when (DEV, F6, F5, F4, F3, F2, F1, F0) is (1, 1, 1, 0, 1, 1, 1, 0).

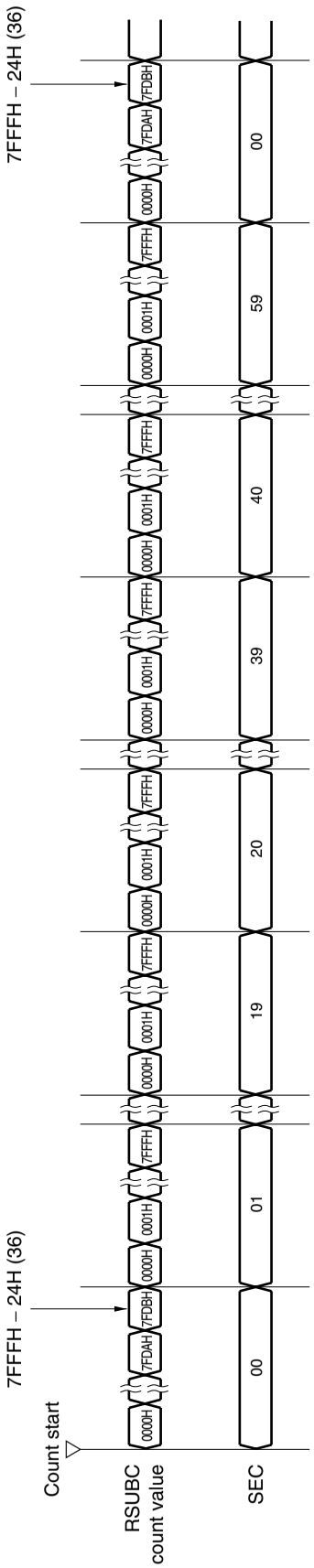


Figure 7-28. Operation When (DEV, F6, F5, F4, F3, F2, F1, F0) = (1, 1, 1, 0, 1, 1, 1, 0)

CHAPTER 8 WATCHDOG TIMER

8.1 Functions of Watchdog Timer

The watchdog timer operates on the internal low-speed oscillation clock.

The watchdog timer is used to detect an inadvertent program loop. If a program loop is detected, an internal reset signal is generated.

Program loop is detected in the following cases.

- If the watchdog timer counter overflows
- If a 1-bit manipulation instruction is executed on the watchdog timer enable register (WDTE)
- If data other than “ACH” is written to WDTE
- If data is written to WDTE during a window close period

When a reset occurs due to the watchdog timer, bit 4 (WDRF) of the reset control flag register (RESF) is set to 1. For details of RESF, see **CHAPTER 20 RESET FUNCTION**.

When 75% of the overflow time is reached, an interval interrupt can be generated.

8.2 Configuration of Watchdog Timer

The watchdog timer includes the following hardware.

Table 8-1. Configuration of Watchdog Timer

Item	Configuration
Control register	Watchdog timer enable register (WDTE)

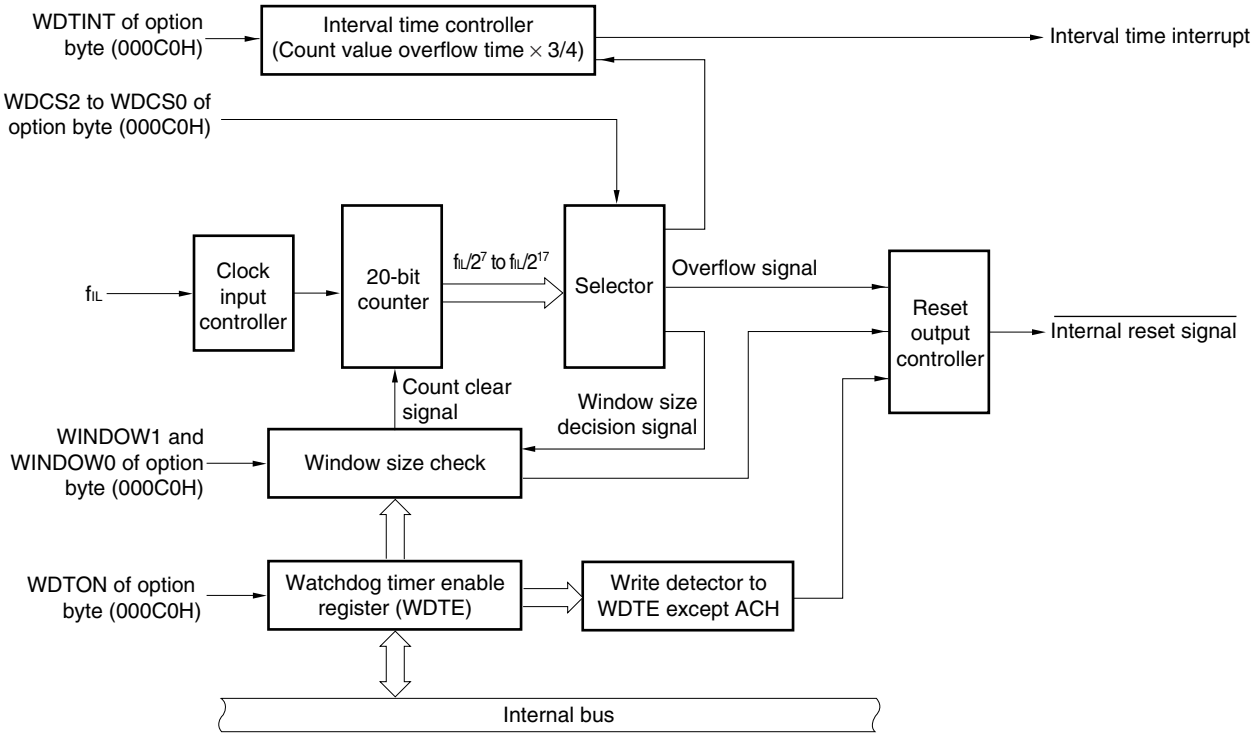
How the counter operation is controlled, overflow time, window open period, and interval interrupt are set by the option byte.

Table 8-2. 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 24 OPTION BYTE**.

Figure 8-1. Block Diagram of Watchdog Timer



8.3 Register Controlling Watchdog Timer

The watchdog timer is controlled by the watchdog timer enable register (WDTE).

(1) Watchdog timer enable register (WDTE)

Writing “ACH” to WDTE clears the watchdog timer counter and starts counting again.

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

Reset signal generation sets this register to 9AH or 1AH[Note].

Figure 8-2. 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 reset value differs depending on the WDTON setting value of the option byte (000C0H). To operate watchdog timer, set WDTON to 1.

WDTON Setting Value	WDTE Reset Value
0 (watchdog timer count operation disabled)	1AH
1 (watchdog timer count operation enabled)	9AH

Cautions 1. If a value other than “ACH” is written to WDTE, an internal reset signal is generated.

2. If a 1-bit memory manipulation instruction is executed for WDTE, an internal reset signal is generated.

3. The value read from WDTE is 9AH/1AH (this differs from the written value (ACH)).

8.4 Operations of Watchdog Timer

8.4.1 Controlling operation of watchdog timer

1. When the watchdog timer is used, its operation is specified by the option byte (000C0H).

   - Enable counting operation of the watchdog timer by setting bit 4 (WDTON) of the option byte (000C0H) to 1 (the counter starts operating after a reset release) (for details, see **CHAPTER 24**).

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 **8.4.2** and **CHAPTER 24**).
   - Set a window open period by using bits 6 and 5 (WINDOW1 and WINDOW0) of the option byte (000C0H) (for details, see **8.4.3** and **CHAPTER 24**).

2. After a reset release, the watchdog timer starts counting.
3. By writing “ACH” to 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 WDTE the second time or later after a reset release during the window open period. If WDTE is written during a window close period, an internal reset signal is generated.
5. If the overflow time expires without “ACH” written to WDTE, an internal reset signal is generated.
   An internal reset signal is generated in the following cases.

   - If a 1-bit manipulation instruction is executed on the watchdog timer enable register (WDTE)
   - If data other than “ACH” is written to WDTE

Cautions 1. When data is written to WDTE for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

2. If the watchdog timer is cleared by writing “ACH” to WDTE, the actual overflow time may be different from the overflow time set by the option byte by up to $2/f_{IL}$ seconds.

3. The watchdog timer can be cleared immediately before the count value overflows.

<Example> When the overflow time is set to $2^{10}/f_{IL}$, writing “ACH” is valid up to count value 3FH.

Cautions 4. The operation of the watchdog timer in the HALT and STOP modes differs as follows depending on the set value of bit 0 (WDSTBYON) of the option byte (000C0H).

	WDSTBYON = 0	WDSTBYON = 1
In HALT mode	Watchdog timer operation stops.	Watchdog timer operation continues.
In STOP mode		

If WDSTBYON = 0, the watchdog timer resumes counting after the HALT or STOP mode is released. At this time, the counter is cleared to 0 and counting starts.
When operating with the X1 oscillation clock after releasing the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed.
Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset.
Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.

5. The watchdog timer continues its operation during self-programming of the flash memory and EEPROM™ emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

8.4.2 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 WDTE during the window open period before the overflow time.

The following overflow time is set.

Table 8-3. Setting of Overflow Time of Watchdog Timer

WDCS2	WDCS1	WDCS0	Overflow Time of Watchdog Timer (f_{IL} = 33 kHz (MAX.))
0	0	0	$2^7/f_{IL}$ (3.88 ms)
0	0	1	$2^8/f_{IL}$ (7.76 ms)
0	1	0	$2^9/f_{IL}$ (15.52 ms)
0	1	1	$2^{10}/f_{IL}$ (31.03 ms)
1	0	0	$2^{12}/f_{IL}$ (124.12 ms)
1	0	1	$2^{14}/f_{IL}$ (496.48 ms)
1	1	0	$2^{15}/f_{IL}$ (992.97 ms)
1	1	1	$2^{17}/f_{IL}$ (3971.88 ms)

Caution The watchdog timer continues its operation during self-programming of the flash memory and EEPROM emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

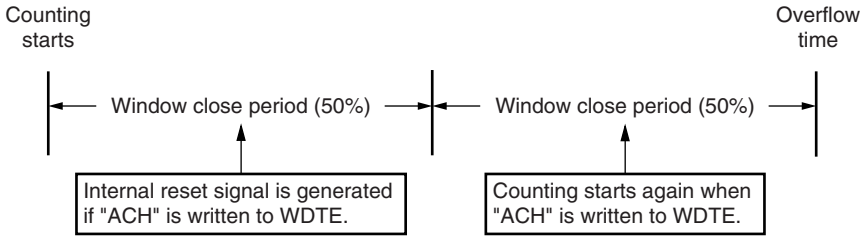
Remark f_{IL}: Internal low-speed oscillation clock frequency

<R> 8.4.3 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 WDTE during the window open period, the watchdog timer is cleared and starts counting again.
- Even if “ACH” is written to WDTE 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 WDTE for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

The window open period to be set is as follows.

Table 8-4. 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%

Cautions 1. The watchdog timer continues its operation during self-programming of the flash memory and EEPROM emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

2. When bit 0 (WDSTBYON) of the option byte (000C0H) = 0, the window open period is 100% regardless of the values of WINDOW1 and WINDOW0.

<R> **Remark** If the overflow time is set to $2^{10}/f_{IL}$, the window close time and open time are as follows.

($2.7\ V \leq V_{DD} \leq 5.5\ V$)

	Setting of Window Open Period		
	50%	75%	100%
Window close time	0 to 18.96 ms	0 to 9.48 ms	None
Window open time	18.96 to 31.03 ms	9.48 to 31.03 ms	0 to 31.03 ms

<When window open period is 50%>

- Overflow time:
  $2^{10}/f_{IL}$ (MAX.) = 2^{10}/33 kHz (MAX.) = 31.03 ms
- Window close time:
  0 to $2^{10}/f_{IL}$ (MIN.) × (1 − 0.5) = 0 to 2^{10}/27 kHz (MIN.) × 0.5 = 0 to 18.96 ms
- Window open time:
  $2^{10}/f_{IL}$ (MIN.) × (1 − 0.5) to $2^{10}/f_{IL}$ (MAX.) = 2^{10}/27 kHz (MIN.) × 0.5 to 2^{10}/33 kHz (MAX.)
  = 18.96 to 31.03 ms

8.4.4 Setting watchdog timer interval interrupt

Depending on the setting of bit 7 (WDTINT) of an option byte (000C0H), an interval interrupt (INTWDTI) can be generated when 75% of the overflow time is reached.

Table 8-5. Setting of Watchdog Timer Interval Interrupt

WDTINT	Use of Watchdog Timer Interval Interrupt
0	Interval interrupt is used.
1	Interval interrupt is generated when 75% of overflow time is reached.

Caution When operating with the X1 oscillation clock after releasing the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed.
Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset.
Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.

Remark The watchdog timer continues counting even after INTWDTI is generated (until ACH is written to the WDTE register). If ACH is not written to the WDTE register before the overflow time, an internal reset signal is generated.

CHAPTER 9 CLOCK OUTPUT/BUZZER OUTPUT CONTROLLER

9.1 Functions of Clock Output/Buzzer Output Controller

The clock output controller is intended for carrier output during remote controlled transmission and clock output for supply to peripheral ICs.

Buzzer output is a function to output a square wave of buzzer frequency.

One pin can be used to output a clock or buzzer sound.

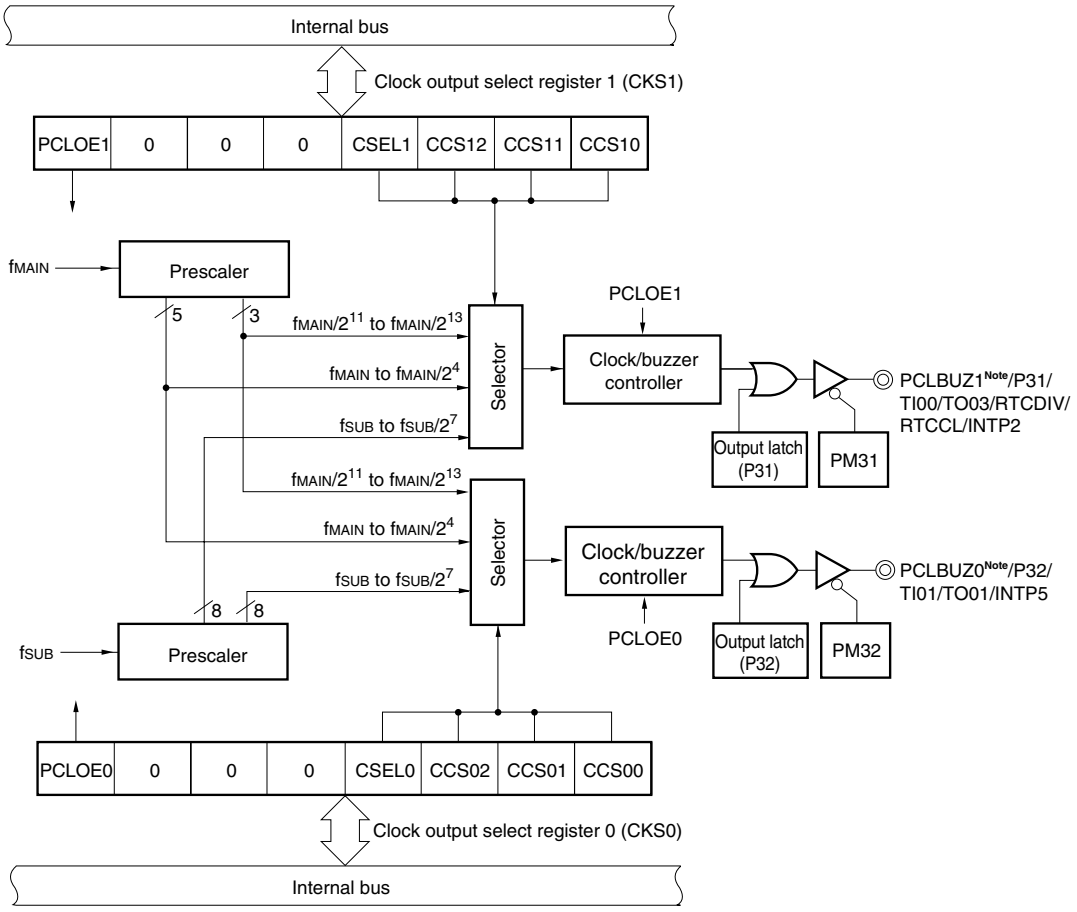
Two output pins, PCLBUZ0 and PCLBUZ1, are available.

PCLBUZ0 outputs a clock selected by clock output select register 0 (CKS0).

PCLBUZ1 outputs a clock selected by clock output select register 1 (CKS1).

Figure 9-1 shows the block diagram of clock output/buzzer output controller.

Figure 9-1. Block Diagram of Clock Output/Buzzer Output Controller



Note The PCLBUZ0 and PCLBUZ1 pins can output a clock of up to 10 MHz at 2.7 V ≤ V_{DD}. Setting a clock exceeding 5 MHz at V_{DD} < 2.7 V is prohibited.

9.2 Configuration of Clock Output/Buzzer Output Controller

The clock output/buzzer output controller includes the following hardware.

Table 9-1. Configuration of Clock Output/Buzzer Output Controller

Item	Configuration
Control registers	Clock output select registers 0, 1 (CKS0, CKS1) Port mode register 3 (PM3) Port register 3 (P3)

9.3 Registers Controlling Clock Output/Buzzer Output Controller

The following two registers are used to control the clock output/buzzer output controller.

- Clock output select registers 0, 1 (CKS0, CSK1)
- Port mode register 3 (PM3)

(1) Clock output select registers 0, 1 (CKS0, CKS1)

These registers set output enable/disable for clock output or for the buzzer frequency output pin (PCLBUZ0/PCLBUZ1), and set the output clock.

Select the clock to be output from PCLBUZ0 by using CKS0.

Select the clock to be output from PCLBUZ1 by using CKS1.

CKS0 and CKS1 are set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 9-2. Format of Clock Output Select Register n (CKSn)

Address: FFFA5H (CKS0), FFFA6H (CKS1) After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
CKSn	PCLOEn	0	0	0	CSELn	CCSn2	CCSn1	CCSn0

PCLOEn	PCLBUZn output enable/disable specification
0	Output disable (default)
1	Output enable

CSELn	CCSn2	CCSn1	CCSn0	PCLBUZn output clock selection			
					f_{MAIN} = 5 MHz	f_{MAIN} = 10 MHz	f_{MAIN} = 20 MHz
0	0	0	0	f_{MAIN}	5 MHz	10 MHz[Note]	Setting prohibited[Note]
0	0	0	1	$f_{MAIN}/2$	2.5 MHz	5 MHz	10 MHz[Note]
0	0	1	0	$f_{MAIN}/2^2$	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{MAIN}/2^3$	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{MAIN}/2^4$	312.5 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{MAIN}/2^{11}$	2.44 kHz	4.88 kHz	9.76 kHz
0	1	1	0	$f_{MAIN}/2^{12}$	1.22 kHz	2.44 kHz	4.88 kHz
0	1	1	1	$f_{MAIN}/2^{13}$	610 Hz	1.22 kHz	2.44 kHz
1	0	0	0	f_{SUB}	32.768 kHz		
1	0	0	1	$f_{SUB}/2$	16.384 kHz		
1	0	1	0	$f_{SUB}/2^2$	8.192 kHz		
1	0	1	1	$f_{SUB}/2^3$	4.096 kHz		
1	1	0	0	$f_{SUB}/2^4$	2.048 kHz		
1	1	0	1	$f_{SUB}/2^5$	1.024 kHz		
1	1	1	0	$f_{SUB}/2^6$	512 Hz		
1	1	1	1	$f_{SUB}/2^7$	256 Hz		

Note Setting an output clock exceeding 10 MHz is prohibited when 2.7 V ≤ V_{DD}. Setting a clock exceeding 5 MHz at V_{DD} < 2.7 V is also prohibited.

Cautions 1. Change the output clock after disabling clock output (PCLOEn = 0).

2. If the selected clock (f_{MAIN} or f_{SUB}) stops during clock output (PCLOEn = 1), the output becomes undefined.

3. To shift to STOP mode when the main system clock is selected (CSELn = 0), set PCLOEn = 0 before executing the STOP instruction. When the subsystem clock is selected (CSELn = 1), PCLOEn = 1 can be set because the clock can be output in STOP mode.

Remarks 1. n = 0, 1

2. f_{MAIN}: Main system clock frequency

3. f_{SUB}: Subsystem clock frequency

(2) Port mode register 3 (PM3)

This register sets port 3 input/output in 1-bit units.

When using the P31/PCLBUZ1/TI00/TO03/RTCDIV/RTCCL/INTP2 and P32/PCLBUZ0/TI01/TO01/INTP5 pins for clock output/buzzer output, clear PM31 and PM32 and the output latches of P32 and P31 to 0.

PM3 is set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 9-3. Format of Port Mode Register 3 (PM3)

Address: FFF23H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM3	1	1	1	PM34	PM33	PM32	PM31	PM30

PM3n	P3n pin I/O mode selection (n = 0 to 4)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

9.4 Operations of Clock Output/Buzzer Output Controller

One pin can be used to output a clock or buzzer sound.

Two output pins, PCLBUZ0 and PCLBUZ1, are available.

PCLBUZ0 outputs a clock/buzzer selected by clock output select register 0 (CKS0).

PCLBUZ1 outputs a clock/buzzer selected by clock output select register 1 (CKS1).

9.4.1 Operation as output pin

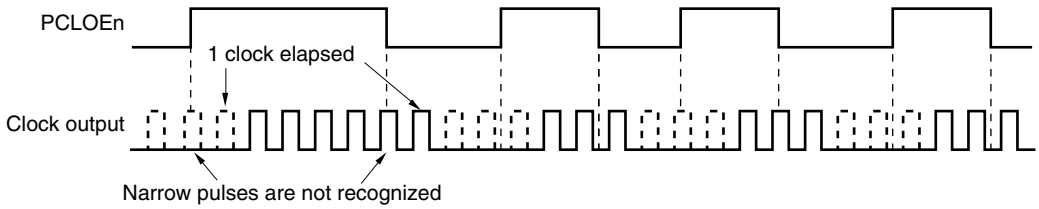
PCLBUZn is output as the following procedure.

<1> Select the output frequency with bits 0 to 3 (CCSn0 to CCSn2, CSELn) of the clock output select register (CKSn) of the PCLBUZn pin (output in disabled status).

<2> Set bit 7 (PCLOEn) of CKSn 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 (PCLOEn) is switched. At this time, pulses with a narrow width are not output. Figure 9-4 shows enabling or stopping output using PCLOEn and the timing of outputting the clock.

Figure 9-4. Remote Control Output Application Example



Remark n = 0, 1

CHAPTER 10 A/D CONVERTER

10.1 Functions of A/D Converter

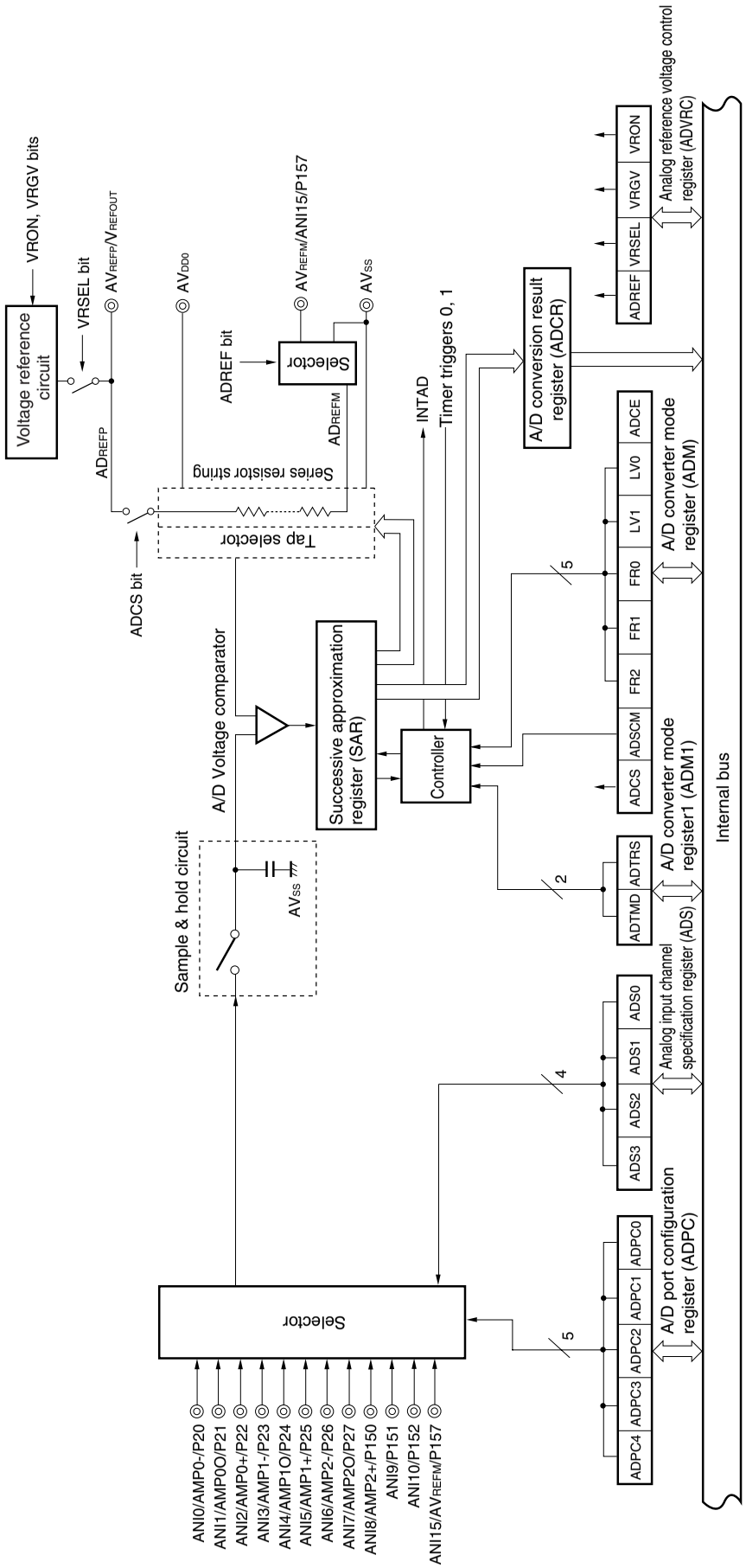
The A/D converter is a 12-bit resolution converter that converts analog input signals into digital values, and consists of up to twelve channels of A/D converter analog inputs (ANI0 to ANI10, ANI15).

ANI1, ANI4, and ANI7 are alternatively used with operational amplifier 0, 1, and 2 outputs (AMP0O, AMP1O, and AMP2O) as pin functions. Accordingly, operational amplifier outputs can be used as analog input sources.

The following four A/D converter operation modes are available.

- Software trigger mode (Continuous conversion mode)
- Software trigger mode (Single conversion mode)
- Timer trigger mode (Continuous conversion mode)
- Timer trigger mode (Single conversion mode)

Figure 10-1. Block Diagram of A/D Converter



10.2 Configuration of A/D Converter

The A/D converter includes the following hardware.

(1) ANI0 to ANI10, ANI15 pins

These are the analog input pins of the A/D converter. They input analog signals to be converted into digital signals. Pins other than the one selected as the analog input pin can be used as I/O port pins.

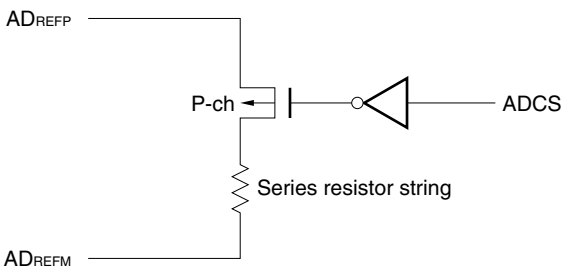
(2) Sample & hold circuit

The sample & hold circuit samples each of the analog input voltages sequentially sent from the input circuit, and sends them to the A/D voltage comparator. This circuit also holds the sampled analog input voltage during A/D conversion.

(3) Series resistor string

The series resistor string is connected between AD_{REFP} and AD_{REFM}, and generates a voltage to be compared with the sampled voltage value.

Figure 10-2. Circuit Configuration of Series Resistor String



(4) Voltage comparator

The voltage comparator compares the sampled voltage value and the output voltage of the series resistor string.

(5) Successive approximation register (SAR)

This register converts the result of comparison by the voltage comparator, starting from the most significant bit (MSB).

When the voltage value is converted into a digital value down to the least significant bit (LSB) (end of A/D conversion), the contents of the SAR register are transferred to the A/D conversion result register (ADCR).

(6) 12-bit A/D conversion result register (ADCR)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCR register holds the A/D conversion result in its lower 12 bits (the higher 4 bits are fixed to 0).

(7) 8-bit A/D conversion result register (ADCRH)

The A/D conversion result is loaded from the successive approximation register to this register each time A/D conversion is completed, and the ADCRH register stores the higher 8 bits of the A/D conversion result.

(8) Controller

This circuit controls the conversion time of an input analog signal that is to be converted into a digital signal, as well as starting and stopping of the conversion operation. When A/D conversion has been completed, this controller generates INTAD.

(9) AV_{DD0} pin

This pin inputs an analog power to the A/D converter. When one or more of the pins of ports 2 and 15 are used as the digital port pins, make AV_{DD0} the same potential as EV_{DD} or V_{DD}.

(10) AV_{SS} pin

This is the ground potential pin of the A/D converter. Always use this pin at the same potential as that of the V_{SS} pin even when the A/D converter is not used.

The ground potential (AV_{SS}) can also be used as the negative reference voltage (AD_{REFM}) of the A/D converter. To use AV_{SS} as AD_{REFM}, clear the ADREF bit of the ADVRC register to 0.

(11) AV_{REFP}/V_{REFOUT} pin

This pin is used to externally input the reference voltage (AV_{REFP}) of the A/D converter or output the voltage (V_{REFOUT}) generated by the voltage reference.

To use AV_{REFP} as the positive reference voltage (AD_{REFP}) of the A/D converter, clear the VRON bit of the ADVRC register to 0. To use V_{REFOUT} as AD_{REFP}, set the VRON bit to 1.

The analog signal input to ANI0 to ANI10, and ANI15 is converted into a digital signal, based on the voltage applied across AD_{REFP} and AD_{REFM}.

(12) AV_{REFM} pin

This pin is used to externally input the reference voltage (AV_{REFM}) of the A/D converter. To use AV_{REFM} as the negative reference voltage (AD_{REFM}) of the A/D converter, set the ADREF bit of the ADVRC register to 1.

10.3 Registers Used in A/D Converter

The A/D converter uses the following nine registers.

- Peripheral enable register 0 (PER0)
- A/D converter mode register (ADM)
- A/D converter mode register 1 (ADM1)
- Analog reference voltage control register (ADVRC)
- 12-bit A/D conversion result register (ADCR)
- 8-bit A/D conversion result register (ADCRH)
- Analog input channel specification register (ADS)
- A/D port configuration register (ADPC)
- Port mode registers 2, 15 (PM2, PM15)

(1) Peripheral enable register 0 (PER0)

PER0 is used to enable or disable use of each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the A/D converter is used, be sure to set bit 5 (ADCEN) of this register to 1.

PER0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10-3. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	DACEN	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

ADCEN	Control of A/D converter, operational amplifier, and voltage reference input clock
0	Stops supply of input clock. • SFR used by the A/D converter, operational amplifier, and voltage reference cannot be written. • The A/D converter, operational amplifier, and voltage reference are in the reset status.
1	Supplies input clock. • SFR used by the A/D converter can, operational amplifier, and voltage reference can be read/written.

Caution When setting the A/D converter, be sure to set ADCEN to 1 first. If ADCEN = 0, writing to a control register of the A/D converter is ignored, and, even if the register is read, only the default value is read.

(2) A/D converter mode register (ADM)

This register sets the conversion time for analog input to be A/D converted, and starts/stops conversion.

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

Reset signal generation clears this register to 00H.

Figure 10-4. Format of A/D Converter Mode Register (ADM)

Address: FFF30H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
ADM	ADCS	ADSCM	FR2 Note 1	FR1 Note 1	FR0 Note 1	LV1 Note 1	LV0 Note 1	ADCE

ADCS	A/D conversion operation control Notes 2, 3, 4
0	Stops conversion operation
1	Enables conversion operation

ADSCM	A/D conversion operation mode specification
0	Continuous conversion mode
1	Single conversion mode

ADCE	A/D voltage comparator operation control Note 4
0	Stops A/D voltage comparator operation
1	Enables A/D voltage comparator operation

Notes 1. For details of FR2 to FR0, LV1, LV0, and A/D conversion, see **Table 10-2 A/D Conversion Time Selection**.

2. When using the A/D converter in timer trigger mode, do not set ADCS to 1. (ADCS automatically switches to 1 when a timer trigger signal is generated.) However, ADCS may be set to 0 to stop A/D conversion.

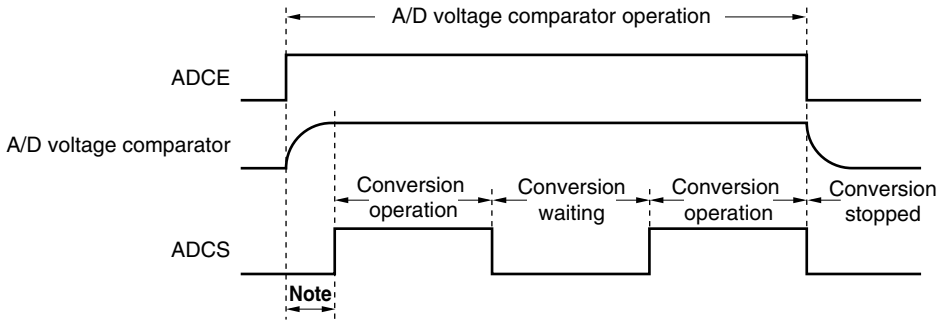
3. Read ADCS to determine whether A/D conversion is under execution.

4. The operation of the A/D voltage comparator is controlled by ADCS and ADCE, and it takes 1 μs from operation start to operation stabilization. Therefore, when ADCS is set to 1 after 1 μs or more has elapsed from the time ADCE is set to 1, the conversion result at that time has priority over the first conversion result. Otherwise, ignore data of the first conversion.

Table 10-1. Settings of ADCS and ADCE

ADCS	ADCE	A/D Conversion Operation
0	0	Stop status (DC power consumption path does not exist)
0	1	Conversion waiting mode (A/D voltage comparator operation, only comparator consumes power)
1	0	Setting prohibited
1	1	Conversion mode (A/D voltage comparator operation)

Figure 10-5. Timing Chart When A/D Voltage Comparator Is Used



Note To stabilize the internal circuit, the time from the rising of the ADCE bit to the falling of the ADCS bit must be 1 μs or longer.

Cautions 1. A/D conversion must be stopped before rewriting bits ADSCM, FR0 to FR2, LV1, and LV0 to values other than the identical data.

2 When using the A/D converter in normal mode 2 (LV1 = 0, LV0 = 1) or low voltage mode (LV1 = 1, LV0 = 0), enable the input gate voltage boost circuit for the A/D converter by <R> **using the analog reference voltage control register (ADVRC), and then set ADCE and ADCS to 1. After the voltage boost circuit stabilization time (10 μs) passes after the input gate voltage boost circuit for the A/D converter has been enabled, set ADCS to 1.**

Table 10-2. A/D Conversion Time Selection

A/D Converter Mode Register (ADM)					Mode	Conversion Time Selection					Conversion Clock (f_{AD})
FR2	FR1	FR0	LV1	LV0			f_{CLK} = 1 MHz	f_{CLK} = 8 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz	
0	0	0	0	0	Normal mode 1 Note 1	240/f_{CLK}	Setting prohibited	30 μs	24 μs	12 μs	f_{CLK}/12
0	0	1				160/f_{CLK}		20 μs	16 μs	8 μs	f_{CLK}/8
0	1	0				120/f_{CLK}		15 μs	12 μs	6 μs	f_{CLK}/6
0	1	1				100/f_{CLK}		12.5 μs	10 μs	5 μs	f_{CLK}/5
1	0	0				80/f_{CLK}		10 μs	8 μs	Setting prohibited	f_{CLK}/4
1	0	1				60/f_{CLK}		7.5 μs	6 μs		f_{CLK}/3
1	1	0				40/f_{CLK}	40 μs	5 μs	Setting prohibited		f_{CLK}/2
1	1	1				20/f_{CLK}	20 μs	Setting prohibited			f_{CLK}
0	0	0	0	1	Normal mode 2 Note 2	240/f_{CLK}	Setting prohibited	30 μs	24 μs	12 μs	f_{CLK}/12
0	0	1				160/f_{CLK}		20 μs	16 μs	8 μs	f_{CLK}/8
0	1	0				120/f_{CLK}		15 μs	12 μs	6 μs	f_{CLK}/6
0	1	1				100/f_{CLK}		12.5 μs	10 μs	5 μs	f_{CLK}/5
1	0	0				80/f_{CLK}		10 μs	8 μs	Setting prohibited	f_{CLK}/4
1	0	1				60/f_{CLK}		7.5 μs	6 μs		f_{CLK}/3
1	1	0				40/f_{CLK}	40 μs	5 μs	Setting prohibited		f_{CLK}/2
1	1	1				20/f_{CLK}	20 μs	Setting prohibited			f_{CLK}
0	0	0	1	0	Low voltage mode Note 3	300/f_{CLK}	Setting prohibited	37.5 μs	30 μs	15 μs Note 4	f_{CLK}/12
0	0	1				200/f_{CLK}		25 μs	20 μs Note 4	10 μs Note 4	f_{CLK}/8
0	1	0				150/f_{CLK}		18.8 μs Note 4	15 μs Note 4	7.5 μs Note 4	f_{CLK}/6
0	1	1				125/f_{CLK}		15.6 μs Note 4	12.5 μs Note 4	6.25 μs Note 4	f_{CLK}/5
1	0	0				100/f_{CLK}		12.5 μs Note 4	10 μs Note 4	Setting prohibited	f_{CLK}/4
1	0	1				75/f_{CLK}		9.38 μs Note 4	7.5 μs Note 4		f_{CLK}/3
1	1	0				50/f_{CLK}	50 μs	6.25 μs Note 4	Setting prohibited		f_{CLK}/2
1	1	1				25/f_{CLK}	25 μs	Setting prohibited			f_{CLK}
Other than above					Setting prohibited						

Notes 1. Normal mode 1: 2.7 V ≤ AV_{DD0} ≤ 5.5 V, when operation of the input gate voltage boost circuit for the A/D converter is stopped.

2. Normal mode 2: 2.3 V ≤ AV_{DD0} ≤ 5.5 V, when operation of the input gate voltage boost circuit for the A/D converter is operating.

3. Low voltage mode: 1.8 V ≤ AV_{DD0} ≤ 5.5 V, when operation of the input gate voltage boost circuit for the A/D converter is operating.

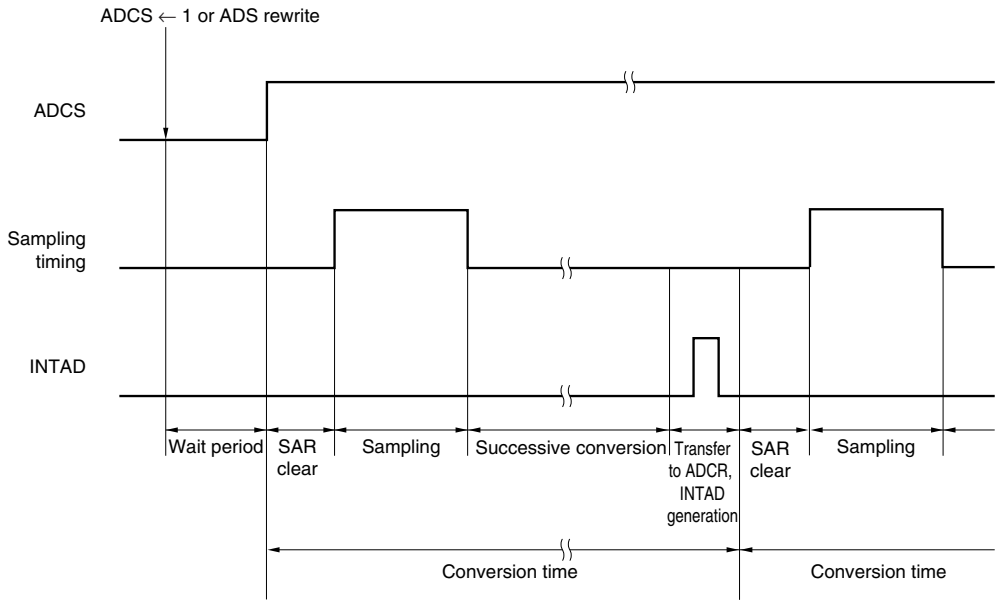
<R> **4.** When T_A = 0 to 50°C and 2.3 V ≤ AV_{DD0} ≤ 3.6 V.

Caution When using the A/D converter in normal mode 2 (LV1 = 0, LV0 = 1) or low voltage mode (LV1 = 1, LV0 = 0), enable the input gate voltage boost circuit for the A/D converter by using the analog reference voltage control register (ADVRC), and then set ADCE and ADCS to 1. After

<R> **the voltage boost circuit stabilization time (10 μs) passes after the input gate voltage boost circuit for the A/D converter has been enabled, set ADCS to 1.**

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

Figure 10-6. A/D Converter Sampling and A/D Conversion Timing



(3) A/D converter mode register 1 (ADM1)

This register sets the A/D conversion start trigger.

ADM1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 10-7. Format of A/D Converter Mode Register 1 (ADM1)

Address: FFF32H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	0
ADM1	ADTMD	0	0	0	0	0	0	ADTRS

ADTMD	A/D trigger mode selection
0	Software trigger mode
1	Timer trigger mode (hardware trigger mode)

ADTRS	Timer trigger signal selection
0	INTTM02
1	INTTM03

Caution Rewriting ADM1 during A/D conversion is prohibited. Rewrite it when conversion operation is stopped (ADCS = 0).

(4) Analog reference voltage control register (ADVRC)

This register is used to select the reference voltage supplies of the A/D and D/A converters, control the operation of the input gate voltage boost circuit for the A/D converter, and control the voltage reference (VR) operation.

The electrical specifications of the A/D converter can be maintained even during low-voltage operation thanks to the operation of the input gate voltage boost circuit for the A/D converter.

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

Reset signal generation clears this register to 00H.

Figure 10-8. Format of Analog reference voltage control register (ADVRC)

Address: FFF36H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADVRC	ADREF	0	0	0	VRSEL	0	VRGV	VRON

ADREF	Negative reference voltage supply selection of A/D converter selection
0	AV_{SS}
1	AV_{REFM} (external voltage reference input)

VRSEL	VRGV	VRON	Positive reference voltage supplies selection of A/D and D/A converters	Operation control of voltage reference	Output voltage selection of voltage reference	Operation control of input gate voltage boost circuit for A/D converter	Relationship with the conversion mode used
0	0	0	AV_{REFP} (external voltage reference input)	Stops operation (Hi-Z)	2.5 V	Stops operation	Can be set in normal mode 1.
0	1	0			2.0 V	Enables operation	Can be set in normal mode 2 or low voltage mode.
1	0	0	V_{REFOUT} (voltage reference output)	Stops operation (pull-down output)	2.5 V	Stops operation	–
1	0	1		Enables operation	2.5 V	Enables operation	Can be set in normal mode 2 or low voltage mode.
1	1	0		Stops operation (pull-down output)	2.0 V		–
1	1	1		Enables operation	2.0 V		Can be set in normal mode 2 or low voltage mode.
Other than the above			Setting prohibited				

Caution 1. When using the A/D converter in normal mode 2 (LV1 = 0, LV0 = 1) or low voltage mode (LV1 = 1, LV0 = 0), enable the input gate voltage boost circuit for the A/D converter by using the analog reference voltage control register (ADVRC), and then set ADCE and <R> **ADCS to 1. After the voltage boost circuit stabilization time (10 μs) passes after the input gate voltage boost circuit for the A/D converter has been enabled, set ADCS to 1.**

Cautions 2. To use voltage reference output to the positive reference voltage of the A/D converter, be sure to set VRON to 1 after setting VRSEL to 1.

3. Do not change the output voltage of the reference voltage by using VRGV during the voltage reference operation (VRON = 1).

Remark The combinations of the selectable reference voltage supplies (positive side, negative side) of the A/D converter are as follows, according to the ADREF, VRSEL and VRON settings.

Table 10-3. Settings of ADREF, VRSEL and VRON

ADREF	VRSEL	VRON	Positive reference voltage of A/D converter (AD_{REFP})	Negative reference voltage of A/D converter (AD_{REFM})
0	0	0	AV_{REFP}	AV_{SS}
0	1	1	V_{REFOUT} (VR output)	AV_{SS}
1	0	0	AV_{REFP}	AV_{REFM}
1	1	1	V_{REFOUT} (VR output)	AV_{REFM}

(5) 12-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 higher 4 bits are fixed to 0. Each time A/D conversion ends, the conversion result is loaded from the successive approximation register. The higher 4 bits of the conversion result are stored in FFF1FH and the lower 8 bits are stored in the FFF1EH.

ADCR can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears this register to 0000H.

Figure 10-9. Format of 12-bit A/D Conversion Result Register (ADCR)

Address: FFF1EH, FFF1FH After reset: 0000H R

Symbol	FFF1FH								FFF1EH							
ADCR	0	0	0	0												

Caution When writing to A/D converter mode register (ADM), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCR may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, and ADPC. Using timing other than the above may cause an incorrect conversion result to be read.

(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 12-bit resolution are stored.

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

Reset signal generation clears this register to 00H.

Figure 10-10. Format of 8-bit A/D Conversion Result Register (ADCRH)

Address: FFF1FH[Note] After reset: 00H R

Symbol	FFF1FH								FFF1EH							
ADCRH	0	0	0	0												
					ADCRH											

Note If address FFF1FH is read, the data of ADCRH (lower four bits of FFF1FH and higher four bits of FFF1EH) will be read.

Caution When writing to A/D converter mode register (ADM), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCRH may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, and ADPC. Using timing other than the above may cause an incorrect conversion result to be read.

(7) Analog input channel specification register (ADS)

This register specifies the input channel of the analog voltage to be A/D converted.

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

Reset signal generation clears this register to 00H.

Figure 10-11. Format of Analog Input Channel Specification Register (ADS)

Address: FFF31H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADS	0	0	0	0	ADS3	ADS2	ADS1	ADS0

ADS3	ADS2	ADS1	ADS0	Analog input channel
0	0	0	0	ANI0
0	0	0	1	ANI1
0	0	1	0	ANI2
0	0	1	1	ANI3
0	1	0	0	ANI4
0	1	0	1	ANI5
0	1	1	0	ANI6
0	1	1	1	ANI7
1	0	0	0	ANI8
1	0	0	1	ANI9
1	0	1	0	ANI10
1	1	1	1	ANI15
Other than the above				Setting prohibited

Cautions 1. Be sure to clear bits 4 to 7 to "0".

2 Set a channel to be used for A/D conversion in the input mode by using port mode registers 2 and 15 (PM2, PM15).

3. Do not set the pin that is set by ADPC as digital I/O by ADS.

4. When using an operational amplifier n, the output signal of an operational amplifier n can be used as an analog input.

Remark n = 0 to 2

(8) A/D port configuration register (ADPC)

This register switches the ANI0/AMP0-/P20 to ANI7/AMP2O/P27, ANI8/AMP2+/P150 to ANI10/P152 and ANI15/AV_{REFM}/P157 pins to analog input of A/D converter or digital I/O of port.

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

Reset signal generation sets this register to 10H.

Figure 10-12. Format of A/D Port Configuration Register (ADPC)

Address: F0017H After reset: 10H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC	0	0	0	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0

ADPC4	ADPC3	ADPC2	ADPC1	ADPC0	Analog input (A)/digital I/O (D) switching											
					Port 15				Port 2							
					ANI15 /AV_{REFM} /P157	ANI10 /P152	ANI9 /P151	ANI8 /AMP2+ /P150	ANI7 /AMP2O /P27	ANI6 /AMP2- /P26	ANI5 /AMP1+ /P25	ANI4 /AMP1O /P24	ANI3 /AMP1- /P23	ANI2 /AMP0+ /P22	ANI1 /AMP0O /P21	ANI0 /AMP0- /P20
0	0	0	0	0	A	A	A	A	A	A	A	A	A	A	A	A
0	0	0	0	1	A	A	A	A	A	A	A	A	A	A	A	D
0	0	0	1	0	A	A	A	A	A	A	A	A	A	A	D	D
0	0	0	1	1	A	A	A	A	A	A	A	A	A	D	D	D
0	0	1	0	0	A	A	A	A	A	A	A	A	D	D	D	D
0	0	1	0	1	A	A	A	A	A	A	A	D	D	D	D	D
0	0	1	1	0	A	A	A	A	A	A	D	D	D	D	D	D
0	0	1	1	1	A	A	A	A	A	D	D	D	D	D	D	D
0	1	0	0	0	A	A	A	A	D	D	D	D	D	D	D	D
0	1	0	0	1	A	A	A	D	D	D	D	D	D	D	D	D
0	1	0	1	0	A	A	D	D	D	D	D	D	D	D	D	D
0	1	1	1	1	A	D	D	D	D	D	D	D	D	D	D	D
1	0	0	0	0	D	D	D	D	D	D	D	D	D	D	D	D
Other than the above					Setting prohibited											

Cautions 1. Set a channel to be used for A/D conversion in the input mode by using port mode registers 2 and 15 (PM2, PM15).

2. Do not set the pin that is set by ADPC as digital I/O by ADS.

(9) Port mode registers 2 and 15 (PM2, PM15)

When using the ANI0/AMP0-/P20 to ANI7/AMP2O/P27, ANI8/AMP2+/P150 to ANI10/P152 and ANI15/AV_{REFM}/P157 pins for analog input port, set PM20 to PM27, PM150 to PM152, and P157 to 1. The output latches of P20 to P27, P150 to P152 and P157 at this time may be 0 or 1.

If PM20 to PM27, PM150 to PM152, and PM157 are set to 0, they cannot be used as analog input port pins.

PM2 and PM15 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Caution If a pin is set as an analog input port, not the pin level but "0" is always read.

Figure 10-13. Formats of Port Mode Registers 2 and 15 (PM2, PM15)

Address: FFF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20

Address: FFF2FH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM15	PM157	1	1	1	1	PM152	PM151	PM150

PMmn	Pmn pin I/O mode selection (mn = 20 to 27, 150 to 152, 157)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

The ANI0/AMP0-/P20 to ANI7/AMP2O/P27, ANI8/AMP2+/P150 to ANI10/P152 and ANI15/AVREFM/P157 pins are as shown below depending on the settings of ADPC, ADS, PM2, PM15, OAENn bit and ADREF bit.

Caution When an operational amplifier is used, pins AMPn+, AMPn–, and AMPnO are used, so the alternative analog input functions cannot be used. The operational amplifier output signals, however, can be used as analog inputs.

Table 10-4. Setting Functions of ANI0/AMP0-/P20, ANI2/AMP0+/P22, ANI3/AMP1-/P23, ANI5/AMP1+/P25, ANI6/AMP2-/P26, and ANI8/AMP2+/P150 Pins

ADPC register	PM2 and PM15 registers	OAENn bit	ADS register	ANI0/AMP0-/P20, ANI2/AMP0+/P22, ANI3/AMP1-/P23, ANI5/AMP1+/P25, ANI6/AMP2-/P26, and ANI8/AMP2+/P150 Pins
Digital I/O selection	Input mode	0	–	Digital input
		1	–	Setting prohibited
	Output mode	0	–	Digital output
		1	–	Setting prohibited
Analog input selection	Input mode	0	Selects ANI.	Analog input (to be converted)
			Does not select ANI.	Analog input (not to be converted)
		1	Selects ANI.	Setting prohibited
			Does not select ANI.	Operational amplifier input
	Output mode	–	–	Setting prohibited

Remark n = 0 to 2

Table 10-5. Setting Functions of ANI1/AMP0O/P21, ANI4/AMP1O/P24, and ANI7/AMP2O/P27 Pins

ADPC register	PM2 register	OAENn bit	ADS register	ANI1/AMP0O/P21, ANI4/AMP1O/P24, and ANI7/AMP2O/P27 Pins
Digital I/O selection	Input mode	0	–	Digital input
		1	–	Setting prohibited
	Output mode	0	–	Digital output
		1	–	Setting prohibited
Analog input selection	Input mode	0	Selects ANI.	Analog input (to be converted)
			Does not select ANI.	Analog input (not to be converted)
		1	Selects ANI.	Operational amplifier output (not to be converted)
			Does not select ANI.	Operational amplifier output (to be converted)
	Output mode	–	–	Setting prohibited

Remark n = 0 to 2

Table 10-6. Setting Functions of ANI9/P151 and ANI10/AM152 Pins

ADPC register	PM15 register	ADS register	ANI9/P151 and ANI10/AM152 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be A/D converted)
		Does not select ANI.	Analog input (not to be A/D converted)
	Output mode	–	Setting prohibited

Table 10-7. Setting Functions of ANI15/AV$_{REFM}$/P157 Pin

ADPC register	PM15 register	ADREF bit	ADS register	ANI15/AV$_{REFM}$/P157 Pin
Digital I/O selection	Input mode	0	–	Digital input
		1	–	Setting prohibited
	Output mode	0	–	Digital output
		1	–	Setting prohibited
Analog input selection	Input mode	0	Selects ANI.	Analog input (to be converted)
			Does not select ANI.	Analog input (not to be converted)
		1	–	Negative reference voltage input of A/D converter
	Output mode	–	–	Setting prohibited

10.4 Operations of A/D Converter

10.4.1 Basic operations of A/D converter

<1> Set bit 5 (ADCEN) of peripheral enable register 0 (PER0) to 1 to start the supply of the input clock to the A/D converter.

<2> Set the A/D conversion time by using bits 5 to 1 (FR2 to FR0, LV1, and LV0) of A/D converter mode register (ADM), and set the operation mode by using bit 6 (ADMD) of ADM.

<3> Use bits 7, 3, 1, and 0 (ADREF, VRSEL, VRGV, and VRON) of the analog reference voltage control register (ADVRC) to specify the reference voltage source of the A/D converter and the operation of the input gate voltage boost circuit for the A/D converter.

<4> Set bit 0 (ADCE) of ADM to 1 to start the operation of the A/D voltage comparator.

<5> Set the channels for A/D conversion to analog input by using the A/D port configuration register (ADPC) and set to input mode by using port mode registers (PM2 and PM15).

<6> Select one channel for A/D conversion using the analog input channel specification register (ADS).

<7> Use the A/D converter mode register 1 (ADM1) to set the trigger mode.

<8> Start the conversion operation by setting bit 7 (ADCS) of ADM to 1, if the software trigger mode has been set in step <7>.

If timer trigger mode was specified in step <7>, ADCS is automatically set to 1 and A/D conversion starts when the timer trigger signal is detected.(<9> to <15> are operations performed by hardware.)

<9> The voltage input to the selected analog input channel is sampled by the sample & hold circuit.

<10> 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.

<11> Bit 11 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.

<12> 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 of SAR remains set to 1. If the analog input is smaller than (1/2) AV_{REF}, the MSB is reset to 0.

<13> Next, bit 10 of SAR 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 11 = 1: (3/4) AV_{REF}
- Bit 11 = 0: (1/4) AV_{REF}

The voltage tap and sampled voltage are compared and bit 8 of SAR is manipulated as follows.

- Sampled voltage ≥ Voltage tap: Bit 10 = 1
- Sampled voltage < Voltage tap: Bit 10 = 0

<14> Comparison is continued in this way up to bit 0 of SAR.

<15> Upon completion of the comparison of 12 bits, an effective digital result value remains in SAR, and the result value is transferred to the A/D conversion result register (ADCR, ADCRH) and then latched.

At the same time, the A/D conversion end interrupt request (INTAD) can also be generated.

<16> If single conversion mode has been set in step <2>, ADCS is automatically cleared to 0 and enters a wait state after the first A/D conversion ends.

If the continuous conversion mode has been set in step <2>, repeat steps <9> to <15>. To stop the A/D converter, clear ADCS to 0.

To restart A/D conversion from the status of ADCE = 1, start from <8>. To start A/D conversion again when ADCE = 0, set ADCE to 1, wait for 1 μs or longer, and start step <8>. To change the channel to be A/D converted, perform step <6>.

Cautions 1. Make sure the period of <4> to <8> is 1 μs or more.

2. To use an operational amplifier output for an analog input, start operating the operational amplifier before setting the A/D conversion operation (see CHAPTER 12 OPERATIONAL AMPLIFIER). Furthermore, do not change the operational amplifier setting during the A/D conversion operation.

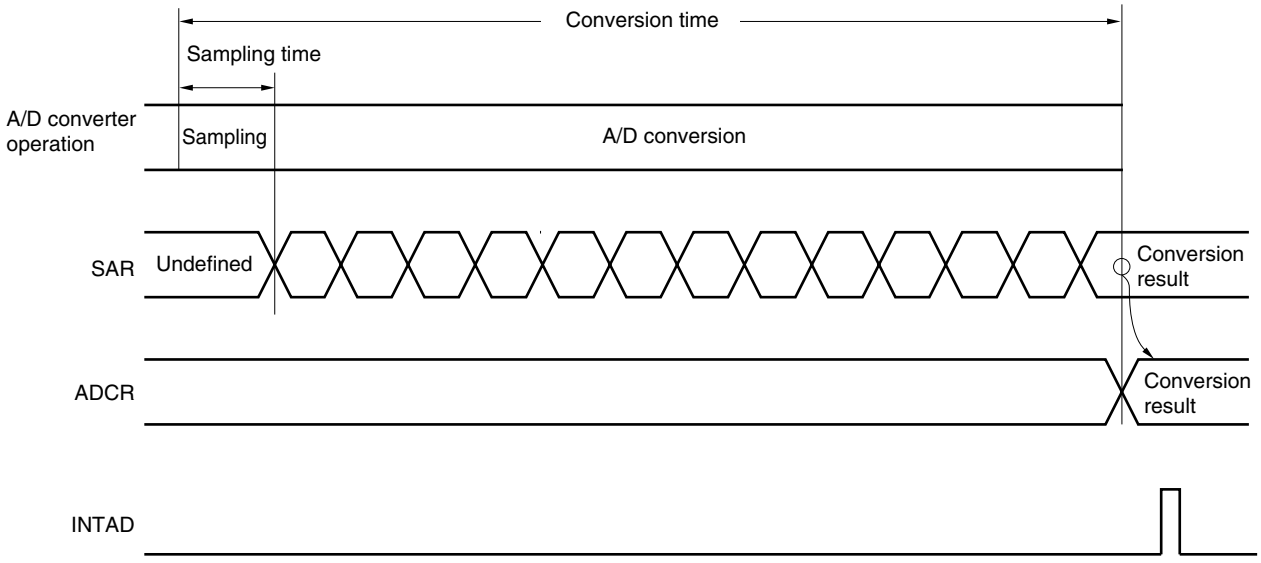
3. To use an output voltage of the voltage reference for a positive reference voltage of the A/D converter, start operating the voltage reference before setting the A/D conversion operation (see CHAPTER 13 VOLTAGE REFERENCE). Furthermore, do not change the voltage reference setting during the A/D conversion operation.

4. When using the A/D converter in normal mode 2 (LV1 = 0, LV0 = 1) or low voltage mode (LV1 = 1, LV0 = 0), enable the input gate voltage boost circuit for the A/D converter by using the analog reference voltage control register (ADVRC), and then set ADCE and <R> **ADCS to 1. After the voltage boost circuit stabilization time (10 μs) passes after the input gate voltage boost circuit for the A/D converter has been enabled, set ADCS to 1.**

Remark Two types of A/D conversion result registers are available. Reset signal generation clears the A/D conversion result register (ADCR, ADCRH) to 0000H or 00H.

- ADCR (16 bits): Store 12-bit A/D conversion value
- ADCRH (8 bits): Store 8-bit A/D conversion value

Figure 10-14. Basic Operation of A/D Converter



10.4.2 Input voltage and conversion results

The relationship between the analog input voltage input to the analog input pins (ANI0 to ANI10, ANI15) and the theoretical A/D conversion result (stored in the 12-bit A/D conversion result register (ADCR)) is shown by the following expression.

$$ADCR = INT\left(\frac{V_{AIN}}{AV_{REF}} \times 4096 + 0.5\right)$$

or

$$(ADCR - 0.5) \times \frac{AV_{REF}}{4096} \leq V_{AIN} < (ADCR + 0.5) \times \frac{AV_{REF}}{4096}$$

where, INT( ): Function which returns integer part of value in parentheses

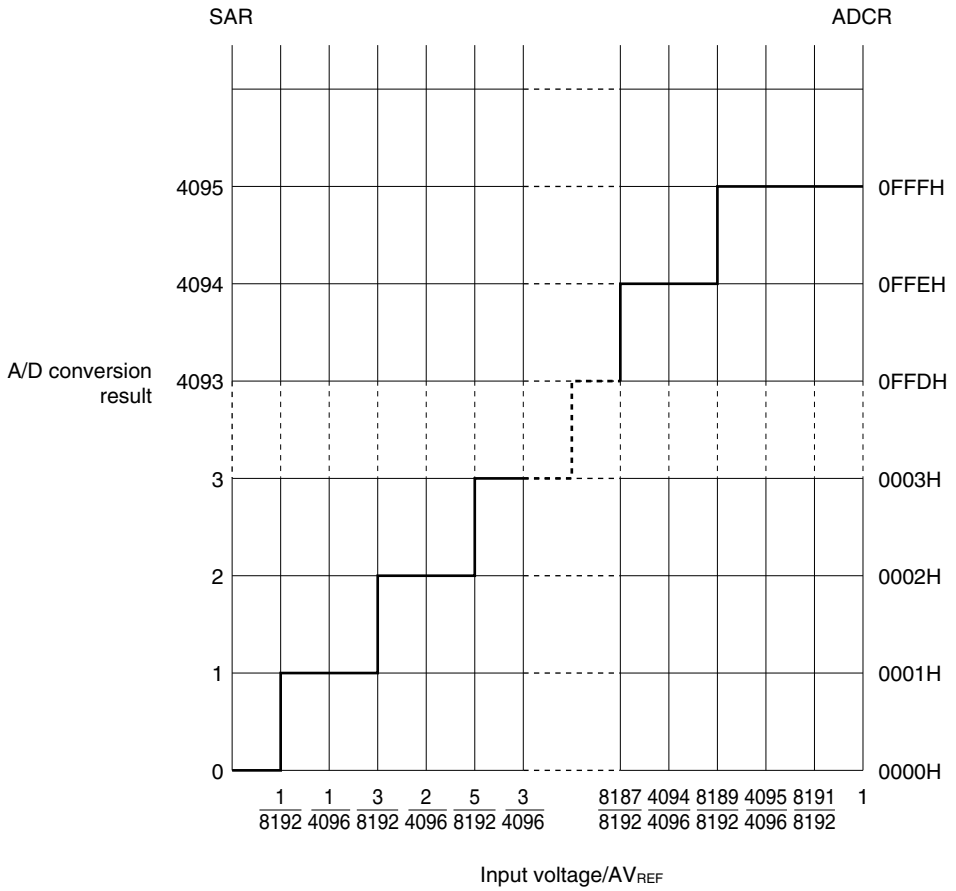
V_{AIN}: Analog input voltage

AV_{REF}: Reference voltage of A/D converter

ADCR: 12-bit A/D conversion result register (ADCR) value

Figure 10-15 shows the relationship between the analog input voltage and the A/D conversion result.

Figure 10-15. Relationship Between Analog Input Voltage and A/D Conversion Result



10.4.3 A/D converter operation modes

The following four A/D converter operation modes are available.

- Software trigger mode (Continuous conversion mode)
- Software trigger mode (Single conversion mode)
- Timer trigger mode (Continuous conversion mode)
- Timer trigger mode (Single conversion mode)

(1) Software trigger mode (Continuous conversion mode)

<1> By setting bit 7 (ADCS) of the A/D converter mode register (ADM) to 1, the A/D conversion operation of the voltage, which is applied to the analog input pin specified by the analog input channel specification register (ADS), is started.

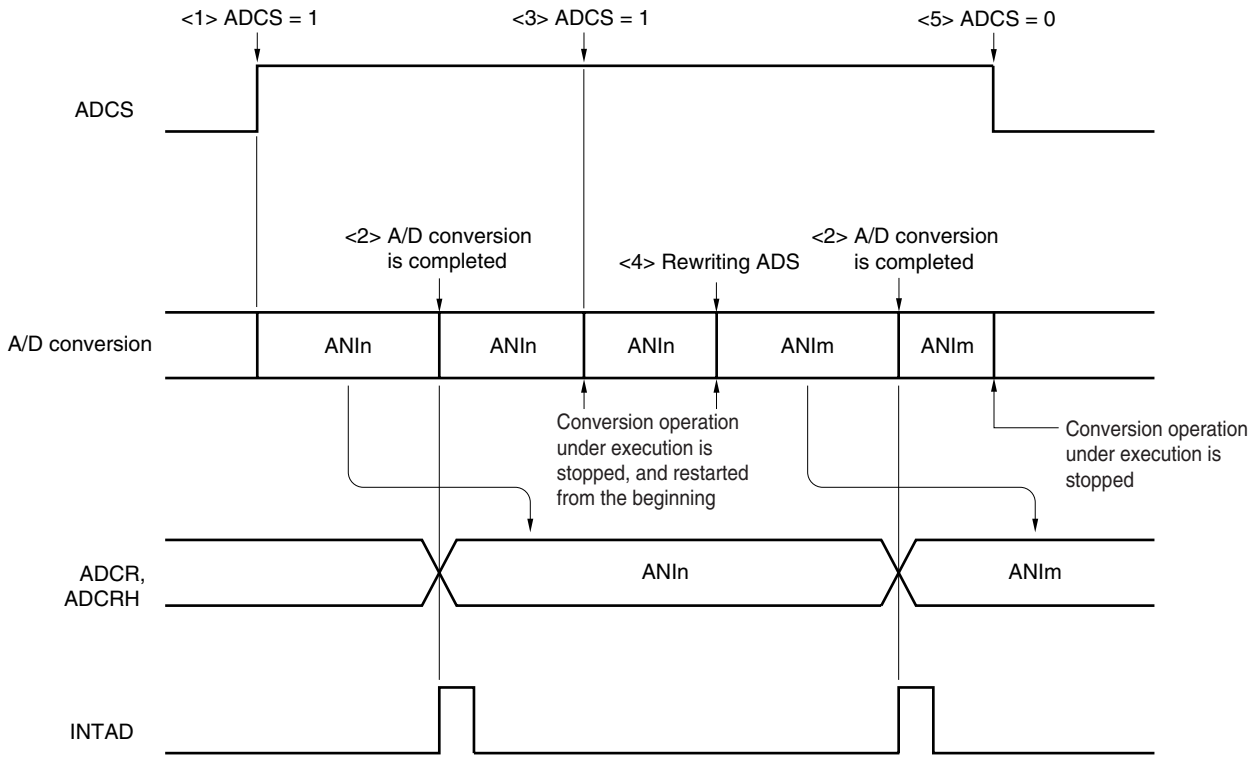
<2> When A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register (ADCR, ADCRH), and an interrupt request signal (INTAD) is generated. When one A/D conversion has been completed, the next A/D conversion operation is immediately started.

<3> If 1 is written to ADCS during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning. At this time, the conversion result immediately before is retained.

<4> If ADS is rewritten during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning. At this time, the conversion result immediately before is retained.

<5> If 0 is written to ADCS during A/D conversion, A/D conversion is immediately stopped. At this time, the conversion result immediately before is retained.

Figure 10-16. Software trigger mode (Continuous conversion mode)



Remark n = 0 to 10, 15, m = 0 to 10, 15

(2) Software trigger mode (Single conversion mode)

<1> By setting bit 7 (ADCS) of the A/D converter mode register (ADM) to 1, the A/D conversion operation of the voltage, which is applied to the analog input pin specified by the analog input channel specification register (ADS), is started.

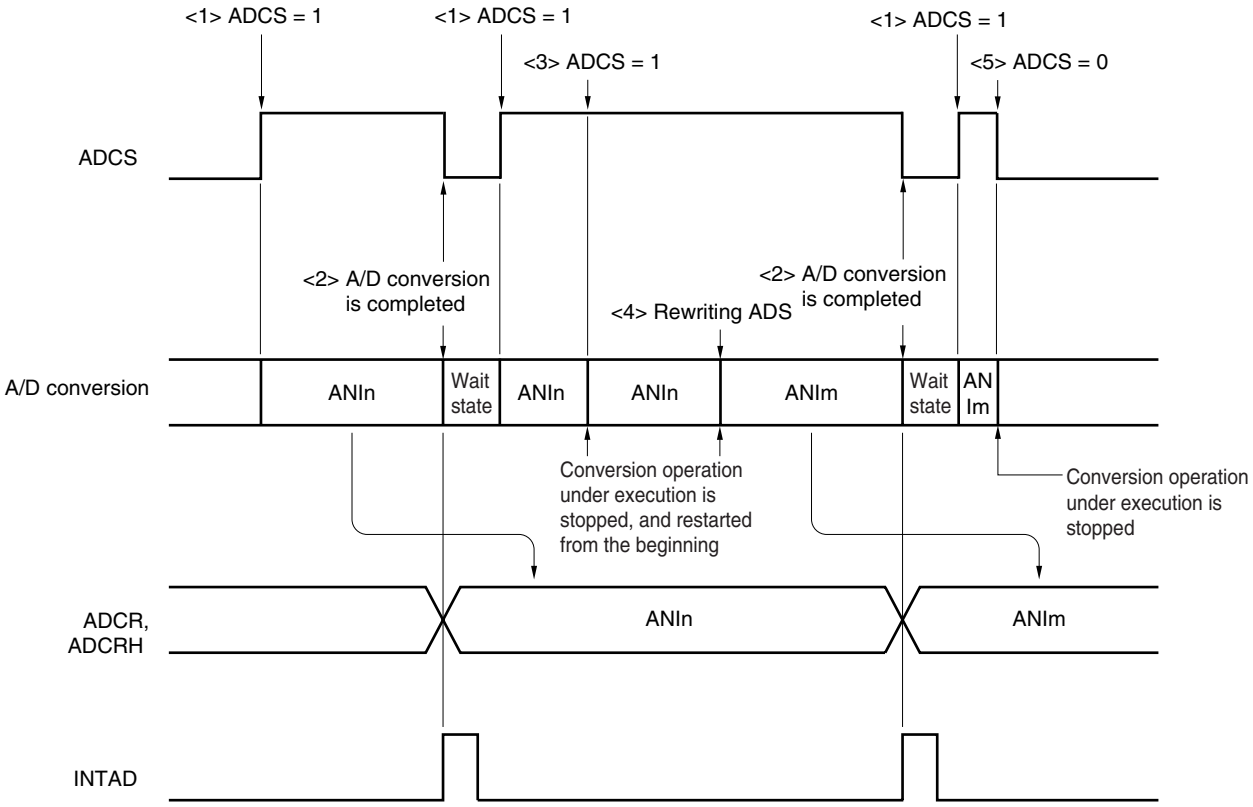
<2> When A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register (ADCR, ADCRH), and an interrupt request signal (INTAD) is generated. When one A/D conversion has been completed, ADCS is automatically cleared and an A/D conversion wait state is entered.

<3> If 1 is written to ADCS during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning. At this time, the conversion result immediately before is retained.

<4> If ADS is rewritten during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning. At this time, the conversion result immediately before is retained.

<5> If 0 is written to ADCS during A/D conversion, A/D conversion is immediately stopped. At this time, the conversion result immediately before is retained.

Figure 10-17. Software trigger mode (Single conversion mode)



Remark n = 0 to 10, 15, m = 0 to 10, 15

(3) Timer trigger mode (Continuous conversion mode)

<1> Timer trigger mode is set and a timer trigger wait state is entered by setting bit 7 (ADTMD) of A/D converter mode register 1 (ADM1) to 1.

<2> When the timer trigger signal is detected, bit 7 (ADCS) of the A/D converter mode register (ADM) is automatically set to 1 and A/D conversion of the voltage applied to the analog input pin specified using the analog input channel specification register (ADS) starts.

<3> When A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register (ADCR, ADCRH), and an interrupt request signal (INTAD) is generated. When one A/D conversion has been completed, the next A/D conversion operation is immediately started.

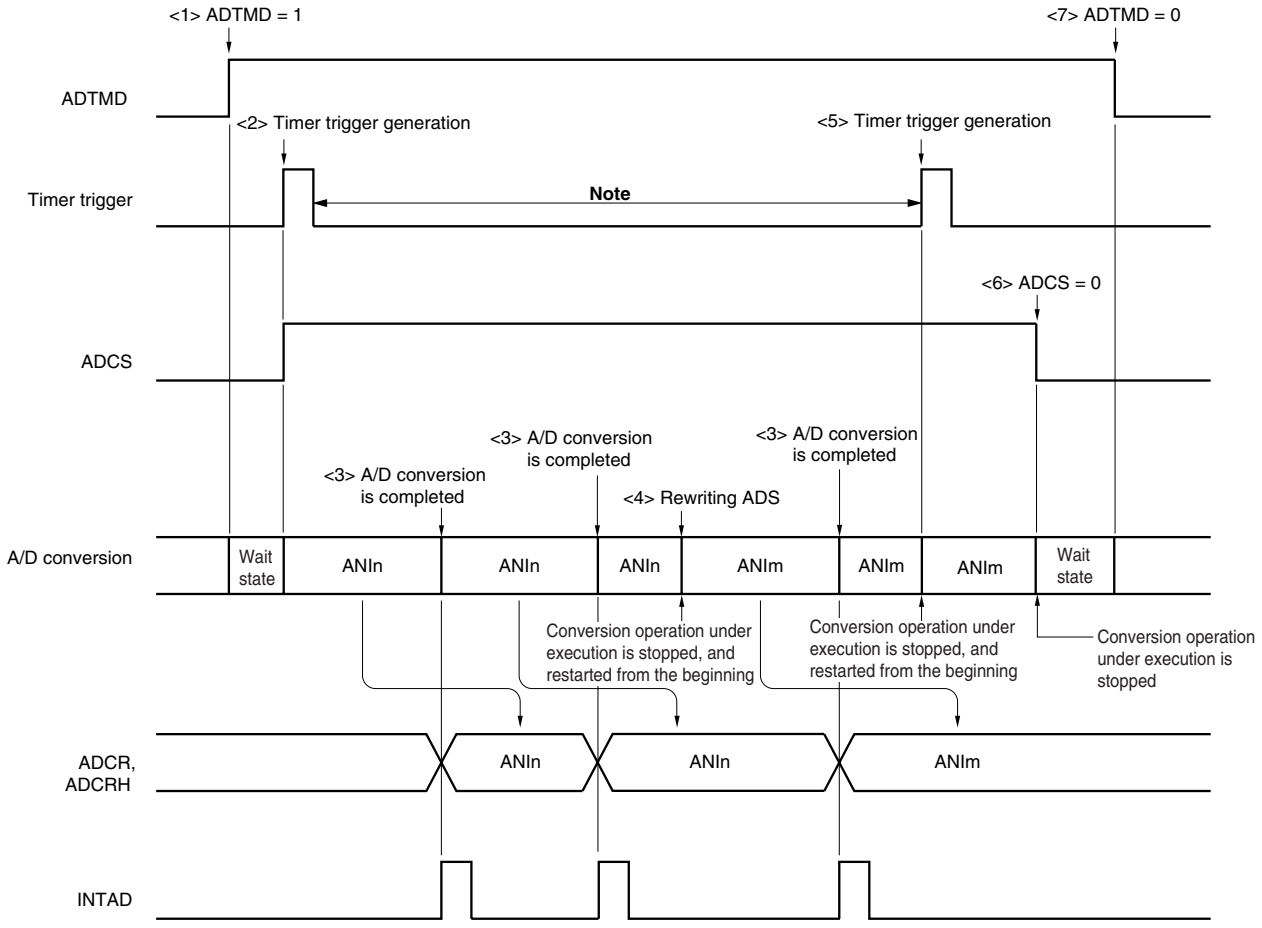
<4> If 1 is written to ADS during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning. At this time, the conversion result immediately before is retained.

<5> If a timer trigger signal is generated during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning. At this time, the conversion result immediately before is retained.

<6> If 0 is written to ADCS during A/D conversion, A/D conversion is immediately stopped, and a timer trigger wait state is entered. At this time, the conversion result immediately before is retained.

<7> When 0 is written to ADTMD while A/D conversion operation is stopped (ADCS = 0), the software trigger mode is set and A/D conversion operation is not started, even if a timer trigger signal is generated.

Figure 10-18. Timer trigger mode (Continuous conversion mode)



Note Leave at least enough time for A/D conversion to finish between each generation of the timer trigger signal.

Remark n = 0 to 10, 15, m = 0 to 10, 15

(4) Timer trigger mode (Single conversion mode)

<1> Timer trigger mode is set and a timer trigger wait state is entered by setting bit 7 (ADTMD) of A/D converter mode register 1 (ADM1) to 1.

<2> When the timer trigger signal is detected, bit 7 (ADCS) of the A/D converter mode register (ADM) is automatically set to 1 and A/D conversion of the voltage applied to the analog input pin specified using the analog input channel specification register (ADS) starts.

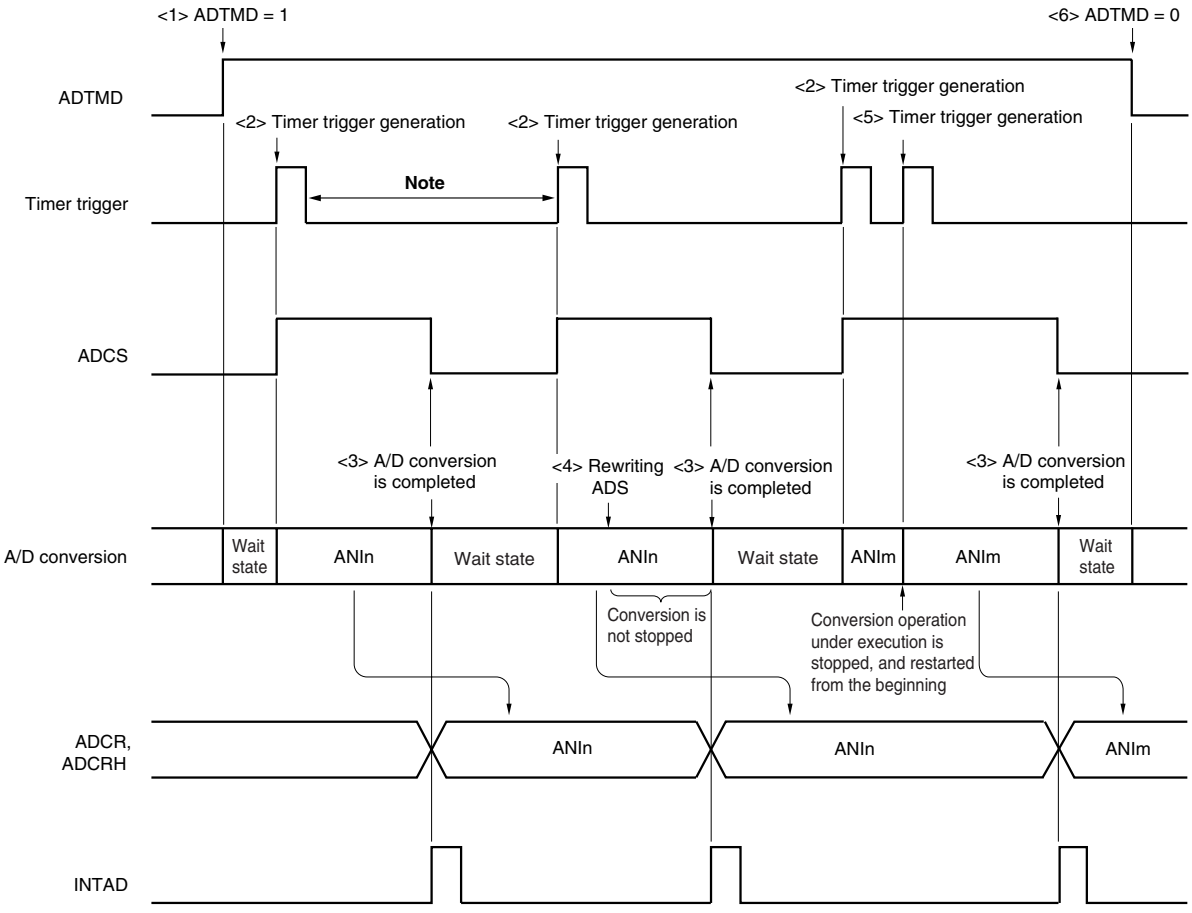
<3> When A/D conversion has been completed, the result of the A/D conversion is stored in the A/D conversion result register (ADCR, ADCRH), and an interrupt request signal (INTAD) is generated. When one A/D conversion has been completed, ADCS is automatically cleared and a timer trigger wait state is entered.

<4> Even if ADS is rewritten during an A/D conversion operation, the A/D conversion operation performed at that time is continued. The channel will be switched when the next A/D conversion operation starts.

<5> If a timer trigger signal is generated during A/D conversion, the A/D conversion operation under execution is stopped and restarted from the beginning. At this time, the conversion result immediately before is retained.

<6> When 0 is written to ADTMD while A/D conversion operation is stopped (ADCS = 0), the software trigger mode is set and A/D conversion operation is not started, even if a timer trigger signal is generated.

Figure 10-19. Timer trigger mode (Single conversion mode)



Note Leave at least enough time for A/D conversion to finish between each generation of the timer trigger signal.

Remark n = 0 to 10, 15, m = 0 to 10, 15

The setting methods are described below.

<1> Set bit 5 (ADCEN) of peripheral enable register 0 (PER0) to 1.
<2> Select the conversion time by using bits 5 to 1 (FR2 to FR0, LV1, and LV0) of A/D converter mode register (ADM), and select the operation mode by using bit 6 (ADSCM) of ADM.
<3> Use bits 7, 3, 1, and 0 (ADREF, VRSEL, VRGV, and VRON) of the analog reference voltage control register (ADVRC) to specify the reference voltage source of the A/D converter and the operation of the input gate voltage boost circuit for the A/D converter.
<4> Set bit 0 (ADCE) of ADM to 1.
<5> Set the channel to be used in the analog input mode by using bits 4 to 0 (ADPC4 to ADPC0) of the A/D port configuration register (ADPC), bits 7 to 0 (PM27 to PM20) of port mode register 2 (PM2), and bits 7, 2 to 0 (PM157, PM152 to PM150) of port mode register 15 (PM15).
<6> Select a channel to be used by using bits 3 to 0 (ADS3 to ADS0) of the analog input channel specification register (ADS).
<7> Use bits 0 and 7 (ADTRS, ADTMD) of A/D converter mode register 1 (ADM1) to set the trigger mode.
<8> In the software trigger mode
→ Start A/D conversion by setting bit 7 (ADCS) of ADM to 1.
In the timer trigger mode
→ ADCS is automatically set to 1 and A/D conversion starts when the timer trigger signal is generated.
<9> When one A/D conversion has been completed, an interrupt request signal (INTAD) is generated.
<10> Transfer the A/D conversion data to the A/D conversion result register (ADCR, ADCRH).
<11> In the continuous conversion mode
→ Start the next A/D conversion automatically.
In the single conversion mode
→ ADCS is automatically cleared to 0 and the A/D converter goes on standby. To start A/D conversion operation, go to step <8>.

<Change the channel>
<12> Change the channel using bits 3 to 0 (ADS3 to ADS0) of ADS to start A/D conversion.[Note]
<13> When one A/D conversion has been completed, an interrupt request signal (INTAD) is generated.
<14> Transfer the A/D conversion data to the A/D conversion result register (ADCR, ADCRH).

<Complete A/D conversion>
<15> Clear ADCS to 0.
<16> In the software trigger mode
→ Clear ADCE to 0.
In the timer trigger mode
→ Clear ADCE and ADTMD to 0.
<17> Clear bit 5 (ADCEN) of peripheral enable register 0 (PER0) to 0.

Note When in timer trigger mode (single conversion mode), the A/D conversion operation is continued even if bits 3 to 0 of ADS are set during A/D conversion. The channel will be changed when the next A/D conversion operation starts.
When in any other mode, A/D conversion operation is aborted after bits 3 to 0 of ADS have been set, and A/D conversion operation is started from the beginning after the channel has been changed.

Cautions 1. Make sure the period of <4> to <8> is 1 μs or more.

2. <4> may be done between <5> and <7>.

3. <4> can be omitted. However, ignore data of the first conversion after <8> in this case.

4. The period from <9> to <13> differs from the conversion time set using bits 5 to 1 (FR2 to FR0, LV1, LV0) of ADM. The period from <12> to <13> is the conversion time set using FR2 to FR0, LV1, and LV0.

5. To use an operational amplifier output for an analog input, start operating the operational amplifier before setting the A/D conversion operation (see CHAPTER 12 OPERATIONAL AMPLIFIER). Furthermore, do not change the operational amplifier setting during the A/D conversion operation.

6. To use an output voltage of the voltage reference for a positive reference voltage of the A/D converter, start operating the voltage reference before setting the A/D conversion operation (see CHAPTER 13 VOLTAGE REFERENCE). Furthermore, do not change the voltage reference setting during the A/D conversion operation.

7. When using the A/D converter in normal mode 2 (LV1 = 0, LV0 = 1) or low voltage mode (LV1 = 1, LV0 = 0), enable the input gate voltage boost circuit for the A/D converter by using the analog reference voltage control register (ADVRC), and then <R> **set ADCE and ADCS to 1. After the voltage boost circuit stabilization time (10 μs) passes after the input gate voltage boost circuit for the A/D converter has been enabled, set ADCS to 1.**

10.5 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 12 bits.

$1LSB = 1/2^{12} = 1/4096$
$= 0.00091\%FSR$

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 10-20. Overall Error

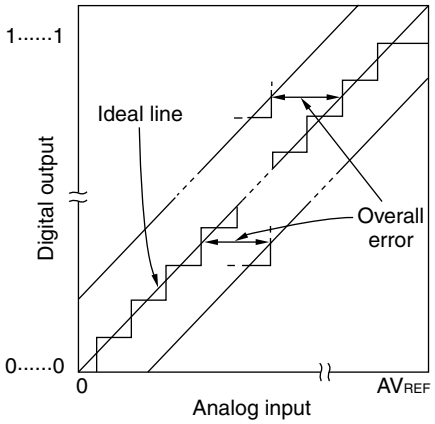
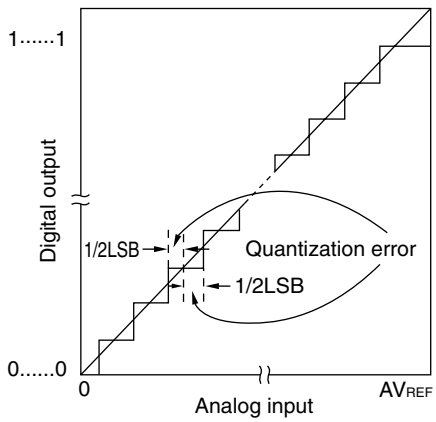


Figure 10-21. Quantization Error



(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 10-22. Zero-Scale Error

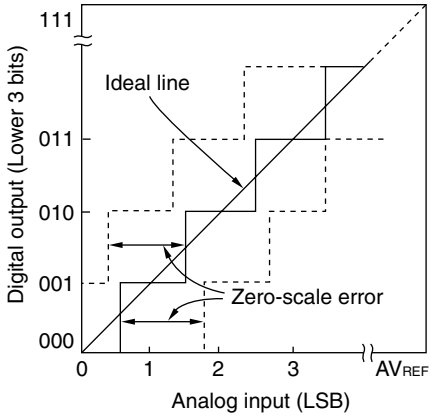


Figure 10-23. Full-Scale Error

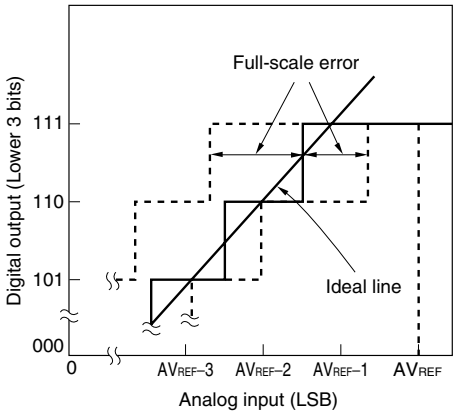


Figure 10-24. Integral Linearity Error

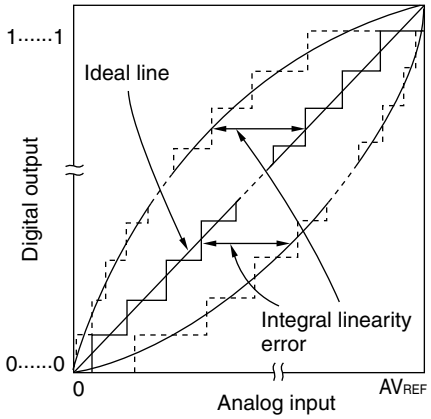
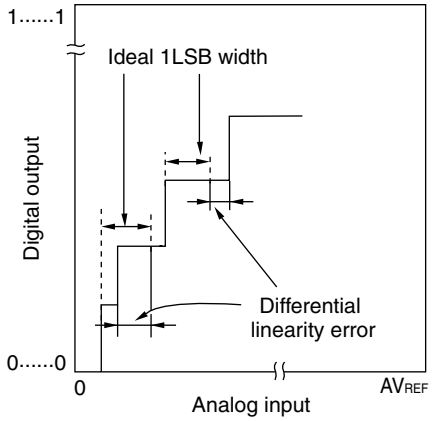


Figure 10-25. Differential Linearity Error



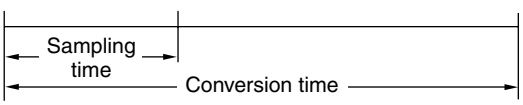
(8) Conversion time

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.



10.6 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 the A/D converter mode register (ADM) to 0). The operating current can be reduced by setting bit 0 (ADCE) of the A/D converter mode register (ADM) to 0 at the same time.

When using normal mode 2 (LV1 = 0, LV0 = 1) or low voltage mode (LV1 = 1, LV0 = 0), clear bit 1 (VRGV) and bit 0 (VRON) of the analog reference voltage control register (ADVRC) to 0, and then shift to STOP mode.

To restart from the standby status, clear bit 0 (ADIF) of interrupt request flag register 1L (IF1L) to 0 and start operation.

(2) Input range of ANI0 to ANI10, ANI15

Observe the rated range of the ANI0 to ANI10, ANI15 input voltage. If a voltage of AV_{DD0} or higher and AV_{SS} or lower (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.

(3) Conflicting operations

<1> Conflict between A/D conversion result register (ADCR, ADCRH) write and ADCR or ADCRH read by instruction upon the end of conversion

ADCR or ADCRH read has priority. After the read operation, the new conversion result is written to ADCR or ADCRH.

<2> Conflict between ADCR or ADCRH write and A/D converter mode register (ADM) write, analog input channel specification register (ADS), or A/D port configuration register (ADPC) write upon the end of conversion

ADM, ADS, or ADPC write has priority. ADCR or ADCRH write is not performed, nor is the conversion end interrupt signal (INTAD) generated.

(4) Noise countermeasures

To maintain the 12-bit resolution, attention must be paid to noise input to the AV_{REFP} pin and pins ANI0 to ANI10, ANI15.

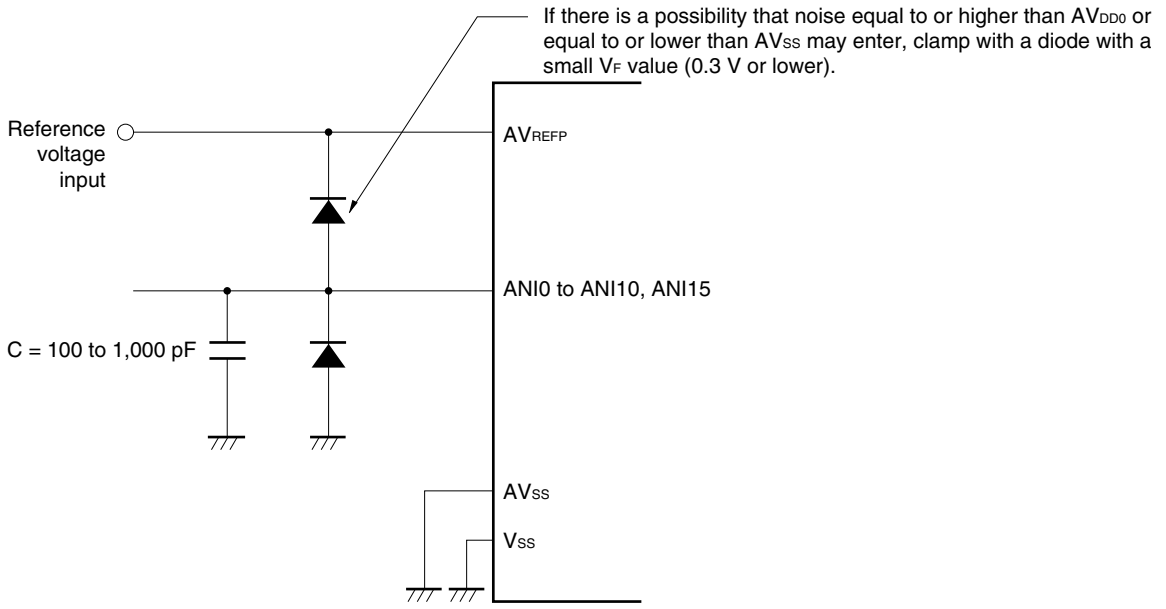
<1> Connect a capacitor with a low equivalent resistance and a good frequency response to the power supply.

<2> The higher the output impedance of the analog input source, the greater the influence. To reduce the noise, connecting external C as shown in Figure 10-26 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 10-26. Analog Input Pin Connection



(5) ANI0 to ANI10, ANI15

<1> The analog input pins (ANI0 to ANI7) are also used as digital I/O pins (P20 to P27).
The analog input pins (ANI8 to ANI10, ANI15) are also used as digital I/O pins (P150 to P152, P157).
When A/D conversion is performed with any of ANI0 to ANI10, and ANI15 selected, do not access P20 to P27, P150 to P152, and P157 while conversion is in progress; otherwise the conversion resolution may be degraded. It is recommended to select pins used as P20 to P27, P150 to P152, and P157 starting with the ANI0/P20 that is the furthest from AV_{DD0}.

<2> If the pins adjacent to the pins currently used for A/D conversion are used as digital I/O port, the expected value of the A/D conversion may not be obtained due to coupling noise. Therefore, make sure that digital pulses are not input to or output from the pins adjacent to the pin undergoing A/D conversion.

<3> If any pin among pins of ports 2 and 15 is used as digital output port during A/D conversion, the expected value of the A/D conversion may not be obtained due to coupling noise. Therefore, make sure that digital pulses are not output to pins of ports 2 and 15 during A/D conversion.

(6) Input impedance of ANI0 to ANI10, ANI15 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.

<R> To make sure that sampling is effective, however, it is recommended to keep the output impedance of the analog input source to within 1 kΩ, and to connect a capacitor of about 100 pF to the ANI0 to ANI10 and ANI15 pins (see **Figure 10-26**).

(7) AV_{REFP} pin input impedance

A series resistor string of several tens of kΩ is connected between the AV_{REFP} and AV_{REFM} (or AV_{SS}) pins.

Therefore, if the output impedance of the reference voltage supply is high, this will result in a series connection to the series resistor string between the AV_{REFP} and AV_{REFM} (or AV_{SS}) pins, resulting in a large reference voltage (AV_{REF}) error of A/D converter.

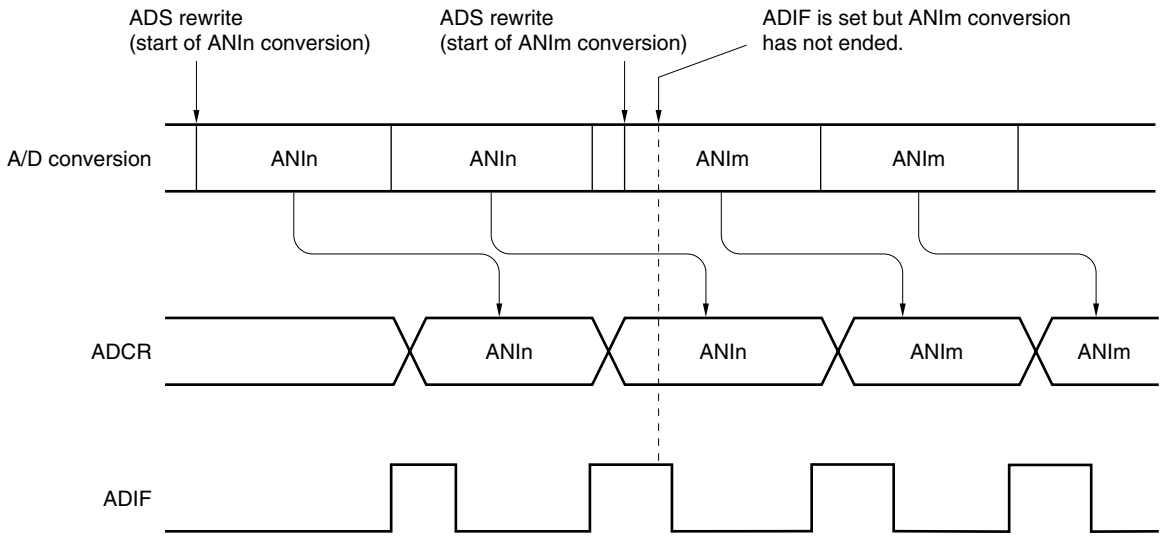
(8) Interrupt request flag (ADIF)

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 for the pre-change analog input may be set just before the ADS rewrite. Caution is therefore required since, at this time, when ADIF is read immediately after the ADS rewrite, ADIF 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 before the A/D conversion operation is resumed.

Figure 10-27. Timing of A/D Conversion End Interrupt Request Generation



Remark n = 0 to 10, 15, m = 0 to 10, 15

(9) Conversion results just after A/D conversion start

The first A/D conversion value immediately after A/D conversion starts may not fall within the rating range if the ADCS bit is set to 1 within 1 μs after the ADCE bit was set to 1, or if the ADCS bit is set to 1 with the ADCE bit = 0. Take measures such as polling the A/D conversion end interrupt request (INTAD) and removing the first conversion result.

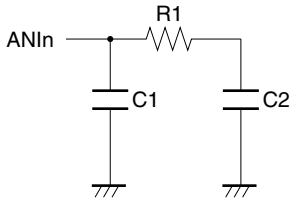
(10) A/D conversion result register (ADCR, ADCRH) read operation

When a write operation is performed to A/D converter mode register (ADM), A/D converter mode register 1 (ADM1), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCR and ADCRH may become undefined. Read the conversion result following conversion completion before writing to ADM, ADM1, ADS, or ADPC. Using a timing other than the above may cause an incorrect conversion result to be read.

(11) Internal equivalent circuit

The equivalent circuit of the analog input block is shown below.

Figure 10-28. Internal Equivalent Circuit of ANIn Pin



<R> **Table 10-8. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)**

R1	C1	C2
11.5 kΩ	8.0 pF	8.0 pF

Remarks 1. The resistance and capacitance values shown in Table 10-8 are not guaranteed values.

2. n = 0-10, 15

(12) Rewriting DACSWn during A/D conversion

Rewriting DACSWn (n = 0, 1) during A/D conversion is prohibited when both the positive reference voltage of the A/D converter (AD_{REFP}) and the positive reference voltage the D/A converter (DA_{REFP}) are the voltage reference output (V_{REFOUT}) (VRSEL = 1 and DAREF = 1). Rewrite it when conversion operation is stopped (ADCS = 0).

CHAPTER 11 D/A CONVERTER

11.1 Functions of D/A Converter

The D/A converter has the following features.

- 12-bit resolution × 2 channels
- R-2R ladder method
- Output analog voltage
  - 12-bit resolution: Reference voltage for D/A converter × m12/4096 (m12: Value set to DACSWn register)
  - 8-bit resolution: Reference voltage for D/A converter × m8/256 (m8: Value set to DACSn register)
- Supply voltage for D/A converter: AV_{DD1}
- Ground for D/A converter: AV_{SS}
- Positive reference voltage for D/A converter: AV_{DD1}, or AV_{REFP}/V_{REFOUT}
- Negative Reference voltage for D/A converter: AV_{SS}
- Operation mode
  - Normal mode
  - Real-time output mode

Remark n = 0, 1

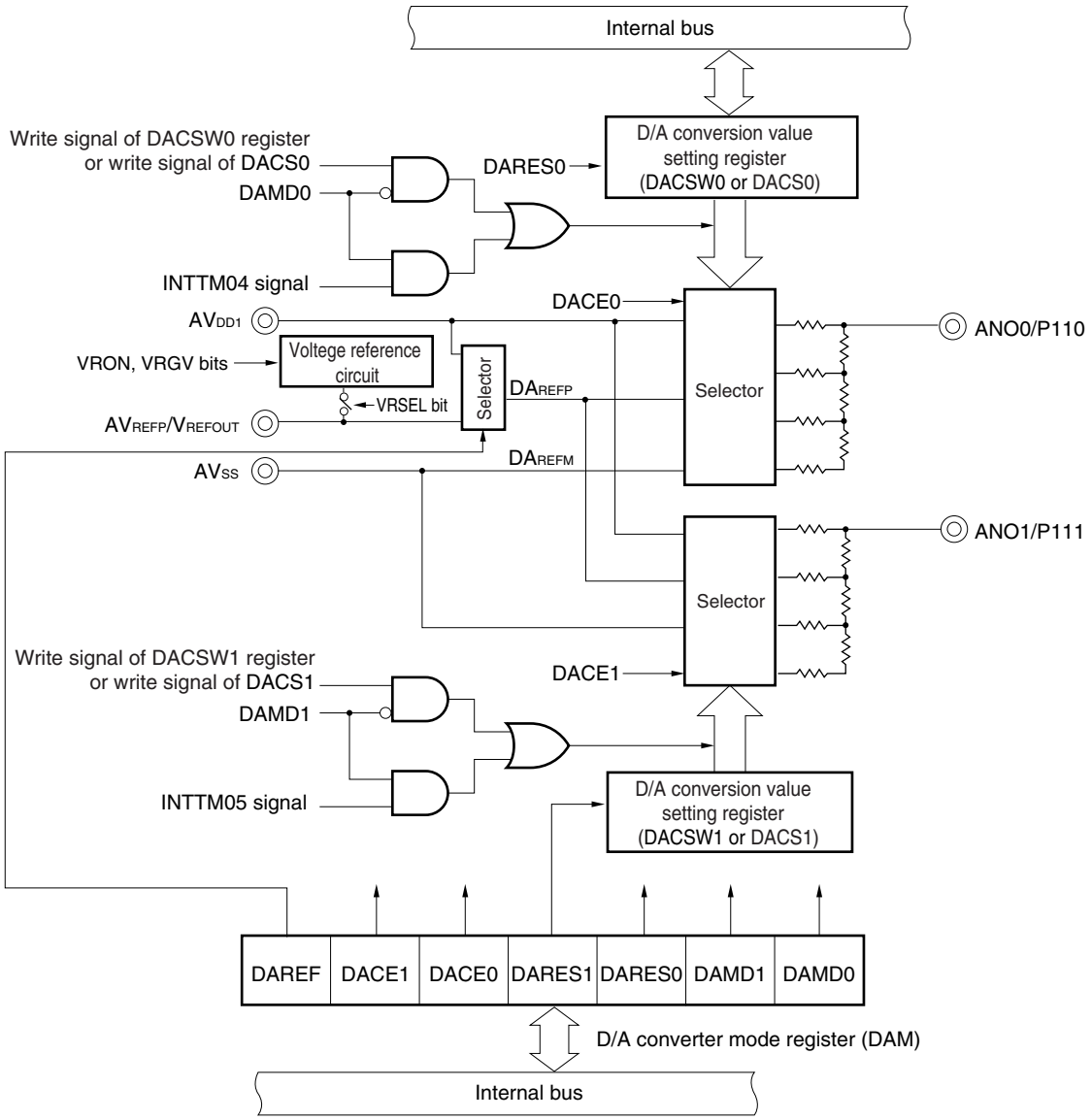
11.2 Configuration of D/A Converter

The D/A converter includes the following hardware.

Table 11-1. Configuration of D/A Converter

Item	Configuration
Control registers	Peripheral enable register 0 (PER0) D/A converter mode register (DAM) D/A conversion value setting registers W0, W1 (DACSW0, DACSW1) D/A conversion value setting registers 0, 1 (DACS0, DACS1)

Figure 11-1. Block Diagram of D/A Converter



Remarks 1. INTTM04 and INTTM05 are timer trigger signals (interrupt signals from timer channels 5 and 6) that are used in the real-time output mode.

2. Channels 0 and 1 of the D/A converter share the AV_{REF1} pin and the AV_{REFP}/V_{REFOUT} pin.

3. Channels 0 and 1 of the D/A converter share the AV_{SS} pin. The AV_{SS} pin is also shared with an A/D converter, an operational amplifier, and a voltage reference.

11.3 Registers Used in D/A Converter

The D/A converter uses the following four registers.

- Peripheral enable register 0 (PER0)
- D/A converter mode register (DAM)
- D/A conversion value setting registers W0, W1 (DACSW0, DACSW1)
- D/A conversion value setting registers 0, 1 (DACS0, DACS1)

(1) Peripheral enable register 0 (PER0)

PER0 is used to enable or disable use of each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the D/A converter is used, be sure to set bit 6 (DACEN) of this register to 1.

PER0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Caution When setting the D/A converter, be sure to set DACEN to 1 first. If DACEN = 0, writing to a control register of the D/A converter is ignored, and, even if the register is read, only the default value is read.

Figure 11-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	DACEN	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

DACEN	Control of D/A converter input clock
0	Stops supply of input clock. • SFR used by the D/A converter cannot be written. • The D/A converter is in the reset status.
1	Supplies input clock. • SFR used by the D/A converter can be read/written.

(2) D/A converter mode register (DAM)

This register controls the operation of the D/A converter.

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

Reset signal generation clears this register to 00H.

Figure 11-3. Format of D/A Converter Mode Register (DAM)

Address: FFF5CH After reset: 00H R/W

Symbol	7	6	<5>	<4>	3	2	1	0
DAM	0	DAREF	DACE1	DACE0	DARES1	DARES0	DAMD1	DAMD0

DAREF	Positive reference voltage supply selection of D/A converter[Note1]
0	AV_{DD1} (power supply for D/A converter analog circuit)
1	V_{REFOUT} (voltage reference output) [Note2] / AV_{REFP} (external voltage reference input)

DACEn	D/A conversion operation Control (n = 0, 1)
0	Stops conversion operation
1	Enables conversion operation

DARESn	D/A converter resolution selection (n = 0, 1)
0	8-bit
1	12-bit

DAMDn	D/A converter operation mode selection (n = 0, 1)
0	Normal mode
1	Real-time output mode

Notes 1. The reference voltage of the D/A converter cannot be specified separately for each channel because it is common to both channels.

2. To use an output voltage of the voltage reference for a positive reference voltage of the D/A converter (DA_{REFP}), start operating the voltage reference before setting the D/A conversion operation (see **CHAPTER 13 VOLTAGE REFERENCE**). Furthermore, do not change the voltage reference setting during the D/A conversion operation.

Remark The positive reference voltage of the D/A converter is as follows, according to the DAREF, VRSEL and VRON settings.

Table 11-2. Settings of DAREF, VRSEL and VRON

DAREF	VRSEL	VRON	Positive reference voltage of D/A converter (DA_{REFP})
0	×	×	AV_{DD1}
1	0	0	AV_{REFP}
1	1	1	V_{REFOUT}

×: don't care

(3) D/A conversion value setting registers W0 and W1 (DACSW0, DACSW1)

These registers are used to set an analog voltage value to be output to the ANO0 and ANO1 pins, when the D/A converter is used.

DACSW0 and DACSW1 can be read by a 16-bit memory manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 11-4. Format of D/A Conversion Value Setting Registers W0 and W1 (DACSW0, DACSW1)

Address: FFF58H, FFF59H (DACSW0), FFF5AH, FFF5BH (DACSW1) After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACSWn	0	0	0	0	DACSWn11	DACSWn10	DACSWn9	DACSWn8	DACSWn7	DACSWn6	DACSWn5	DACSWn4	DACSWn3	DACSWn2	DACSWn1	DACSWn0

Caution Rewriting DACSWn during A/D conversion is prohibited when both the positive reference voltage of the A/D converter (AD_{REFP}) and the positive reference voltage of the D/A converter (DA_{REFP}) are the voltage reference output (V_{REFOUT}) (VRSEL = 1 and DAREF = 1). Rewrite it when conversion operation is stopped (ADCS = 0).

Remarks 1. The relations between the resolutions and analog output voltages (V_{ANOn}) of the D/A converter are as follows.

- 8-bit resolution (DARESn = 0) :

  V_{ANOn} = Reference voltage for D/A converter × (DACSWn7 to DACSWn0) /256
- 12-bit resolution (DARESn = 1) :

  V_{ANOn} = Reference voltage for D/A converter × (DACSWn11 to DACSWn0) /4096

2. n = 0, 1

(4) D/A conversion value setting registers 0 and 1 (DACS0, DACS1)

These registers are used to set the analog voltage values to be output to the ANO0 and ANO1 pins when the D/A converter is used at 8-bit resolution.

DACS0 and DACS1 can be read by an 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 11-5. Format of D/A Conversion Value Setting Registers 0 and 1 (DACS0, DACS1)

Address: FFF58H (DACS0), FFF5AH (DACS1) After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
DACSn	DACSn7	DACSn6	DACSn5	DACSn4	DACSn3	DACSn2	DACSn1	DACSn0

Remarks 1. The relations between the resolutions and analog output voltages (V_{ANOn}) of the D/A converter are as follows.

- 8-bit resolution (DARESn = 0) :

  V_{ANOn} = Reference voltage for D/A converter × (DACSn7 to DACSn0) /256

2. n = 0, 1

11.4 Operations of D/A Converter

11.4.1 Operation in normal mode

D/A conversion is performed using write operation to the DACSn register as the trigger. The setting method is described below.

<1> Set bit 6 (DACEN) of peripheral enable register 0 (PER0) to 1 to start the supply of the input clock to the D/A converter.
<2> Set the DAMDn bit of the D/A converter mode register (DAM) to 0 (normal mode).
<3> Use the bit 6 (DAREF) of the DAM register to select the D/A converter reference voltage supply on the positive side.
<4> Use the DARESn bit of the DAM register to select the resolution of the D/A converter.
<5> Set the analog voltage value to be output to the ANOn pin to the D/A conversion value setting register Wn (DACSWn) or D/A conversion value setting register n (DACSn).
Steps <1> and <5> above constitute the initial settings.
<6> Set the DACEn bit of the DAM register to 1 (D/A conversion enable).
<R> After the wait time (20 μs or more) elapses, D/A conversion starts, and then, after the settling time (18 μs
<R> (MAX.)) elapses, the D/A converted analog voltage value is output from the ANOn pin.
<7> To perform subsequent D/A conversions, write to the DACSWn or DACSn register.
The previous D/A conversion result is held until the next D/A conversion is performed.
When the DACEn bit of the DAM register is set to 0 (D/A conversion operation stop), D/A conversion stops, the ANOn pin goes into a high-impedance state when the PM11n bit of the PM11 register = 1 (input mode), and the ANOn pin outputs the set value of the P11 register when the PM11n bit = 0 (output mode).

Cautions 1. Even if 1, 0, and then 1 is set to the DACEn bit, there is a wait after 1 is set for the last time.
2. If the DACSWn or DACSn register is rewritten during the settling time, D/A conversion is aborted and reconversion by using the rewritten values starts.

Remark n = 0, 1

11.4.2 Operation in real-time output mode

D/A conversion is performed using the interrupt request signals (INTTM04 and INTTM05) [Note] of timer channels 4 and 5 as triggers.
The setting method is described below.

Note Channel 0 of the D/A converter: INTTM04
Channel 1 of the D/A converter: INTTM05

<1> Set bit 6 (DACEN) of peripheral enable register 0 (PER0) to 1 to start the supply of the input clock to the D/A converter.
<2> Set the DAMDn bit of the D/A converter mode register (DAM) to 0 (normal mode).
<3> Use the bit 6 (DAREF) of the DAM register to select the D/A converter reference voltage supply on the positive side.
<4> Use the DARESn bit of the DAM register to select the resolution of the D/A converter.
<5> Set the analog voltage value to be output to the ANOn pin to the D/A conversion value setting register Wn (DACSWn) or D/A conversion value setting register n (DACSn).

<6> Set the DACEn bit of the DAM register to 1 (D/A conversion enable).

<R> After the wait time (20 μs or more) elapses, D/A conversion starts, and then, after the settling time (18 μs

<R> (MAX.)) elapses, the D/A converted analog voltage value is output from the ANOn pin.

<7> Set the DAMDn bit of the DAM register to 1 (real-time output mode).

Steps <1> to <7> above constitute the initial settings.

<8> Operate timer channel m.

<9> Generation of the INTTM0m signals starts D/A conversion and the D/A converted analog voltage value

<R> will be output from the ANOn pin after a settling time (18 μs (MAX.)) has elapsed.

<10> Afterward, the value set to the DACSWn or DACSn register will be output at the generation timing of the INTTM0m signals.

Set the analog voltage value to be output to the ANOn pin, to the DACSWn or DACSn register before performing the next D/A conversion (INTTM0m signal are generated).

When the DACEn bit of the DAM register is set to 0 (D/A conversion operation stop), D/A conversion stops, the ANOn pin goes into a high-impedance state when the PM11n bit of the PM11 register = 1 (input mode), and the ANOn pin outputs the set value of the P11 register when the PM11n bit = 0 (output mode).

Cautions 1. Even if 1, 0, and then 1 is set to the DACEn bit, there is a wait after 1 is set for the last time.

2. Make the interval between each generation of the INTTM0m signal longer than the settling time. If an INTTM0m signal is generated during the settling time, D/A conversion is aborted and reconversion starts.

3. Even if the generation of the INTTM0m signal and rewriting the DACSWn or DACSn register conflict, the D/A conversion result is output.

Remark n = 0, 1

11.5 Cautions for D/A Converter

Observe the following cautions when using the D/A converter.

(1) The digital port I/O function, which is the alternate function of the ANO0 and ANO1 pins, does not operate during D/A conversion.

When the P11 register is read during D/A conversion, 0 is read in input mode and the set value of the P11 register is read in output mode. If the digital output mode is set, no output data is output to pins.

(2) The operation of the D/A converter continues in the HALT and STOP mode. To lower the power consumption, therefore, clear the DACEn bit of the DAM register to 0 (D/A conversion stop), and execute HALT or STOP instruction.

(3) Rewriting DACSWn (n = 0, 1) during A/D conversion is prohibited when both the positive reference voltage of the A/D converter (AD_{REFP}) and the positive reference voltage of the D/A converter (DA_{REFP}) are the voltage reference output (V_{REFOUT}) (VRSEL = 1 and DAREF = 1). Rewrite it when conversion operation is stopped (ADCS = 0).

Remark n = 0, 1, m = 4, 5

CHAPTER 12 OPERATIONAL AMPLIFIERS

12.1 Function of Operational Amplifiers

The operational amplifiers have the following modes.

- **Single AMP mode**
  The difference in potential of analog voltages input from two pins (AMPn− and AMPn+ pins) is amplified and the amplified voltage is output from the AMPnO pin.
  The amplified voltage can be used as an analog input of the A/D converter, because the AMPnO pin is alternatively used with analog input pin of the A/D converter.

12.2 Configuration of Operational Amplifiers

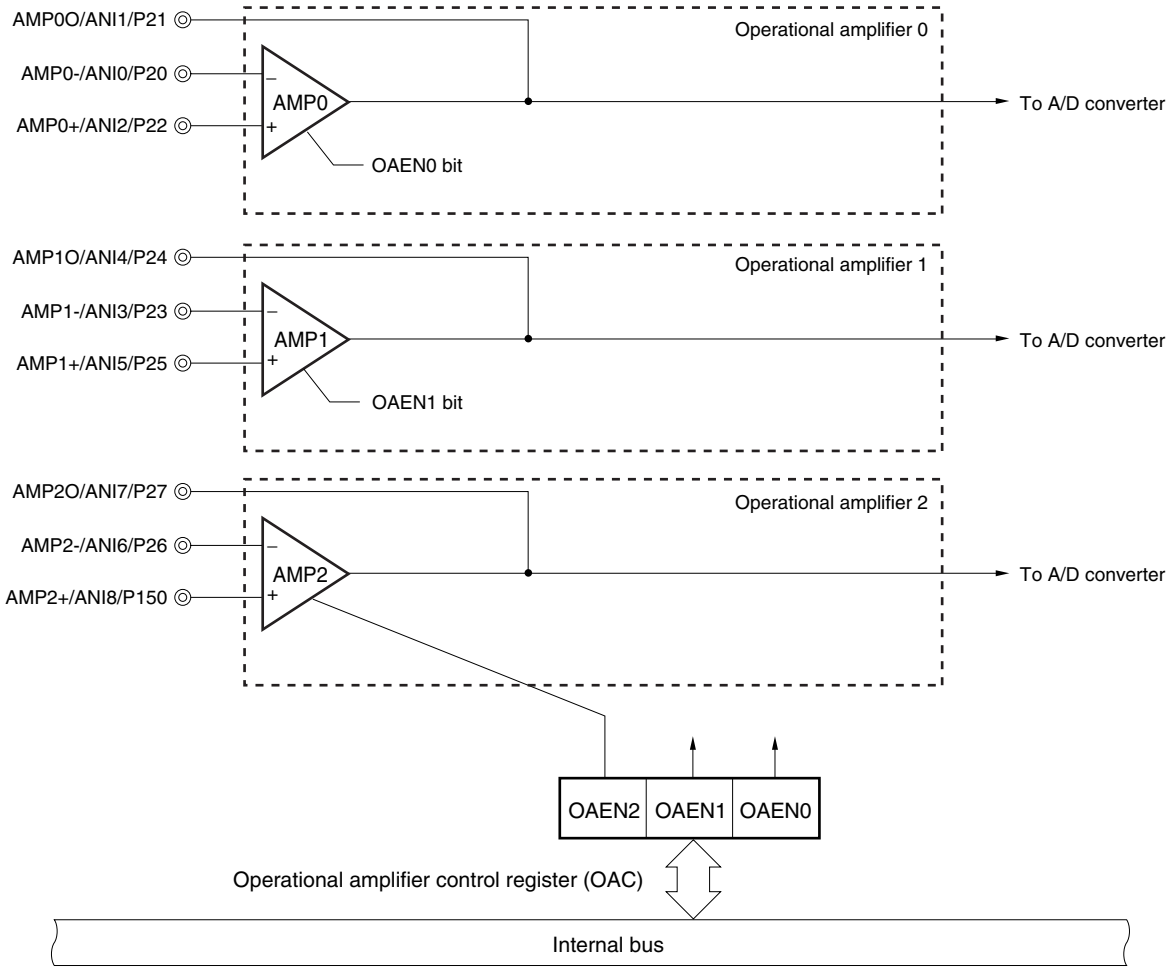
The operational amplifiers consist of the following hardware.

Table 12-1. Configuration of Operational Amplifiers

Item	Configuration
Operational amplifier inputs	AMPn− pin, AMPn+ pin
Operational amplifier output	AMPnO pin
Control registers	Peripheral enable register 0 (PER0) Operational amplifier control register (OAC) A/D port configuration register (ADPC) Port mode registers 2, 15 (PM2, PM15)

Remark n = 0 to 2

Figure 12-1. Block Diagram of Operational Amplifiers



12.3 Amplifier Registers Used in Operational Amplifiers

The operational amplifiers use the following four registers.

- Peripheral enable register 0 (PER0)
- Operational amplifier control register (OAC)
- A/D port configuration register (ADPC)
- Port mode registers 2, 15 (PM2, PM15)

(1) Peripheral enable register 0 (PER0)

PER0 is used to enable or disable use of each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the operational amplifier is used, be sure to set bit 5 (ADCEN) of this register to 1.

PER0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 12-2. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	DACEN	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

ADCEN	Control of A/D converter, operational amplifier, and voltage reference input clock
0	Stops input clock supply. • SFR used by the A/D converter, operational amplifier, and voltage reference cannot be written. • The A/D converter, operational amplifier, and voltage reference are in the reset status.
1	Supplies input clock. • SFR used by the A/D converter, operational amplifier, and voltage reference can be read and written.

Caution When setting operational amplifier, be sure to set ADCEN to 1 first. If ADCEN = 0, writing to a control register of operational amplifier is ignored, and, even if the register is read, only the default value is read

(2) Operational amplifier control register (OAC)

This register controls the operations of operational amplifiers 0 to 2.

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

Reset signal generation clears this register to 00H.

Figure 12-3. Format of Operational Amplifier Control Register (OAC)

Address: FFF33H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
OAC	0	0	0	0	0	OAEN2	OAEN1	OAEN0

OAEN2	Operational amplifier 2 operation control
0	Stops operational amplifier 2 operation
1	Enables operational amplifier 2 operation

OAEN1	Operational amplifier 1 operation control
0	Stops operational amplifier 1 operation
1	Enables operational amplifier 1 operation

OAEN0	Operational amplifier 1 operation control
0	Stops operational amplifier 0 operation
1	Enables operational amplifier 0 operation

Cautions 1. Use the ADPC register to specify as analog inputs the pins to be used with operational amplifiers.

2. When using as digital inputs the pins of ports 2 and 15, which are not used with operational amplifiers, when the operational amplifiers are used, make sure that the input levels are fixed.

(3) A/D port configuration register (ADPC)

This register switches the ANI0/AMP0-/P20 to ANI7/AMP2O/P27, ANI8/AMP2+/P150 to ANI10/P152, and ANI15/AV_{REFM}/P157 pins to analog input of A/D converter or digital I/O of port. Set pins to be used with operational amplifiers to the analog input.

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

Reset signal generation sets this register to 10H.

Figure 12-4. Format of A/D Port Configuration Register (ADPC)

Address: F0017H After reset: 10H R/W

Symbol	7	6	5	4	3	2	1	0
ADPC	0	0	0	ADPC4	ADPC3	ADPC2	ADPC1	ADPC0

ADPC4	ADPC3	ADPC2	ADPC1	ADPC0	Analog input (A)/digital I/O (D) switching											
					Port 15				Port 2							
					ANI15 /AV_{REFM} /P157	ANI10 /P152	ANI9 /P151	ANI8 /AMP2+ /P150	ANI7 /AMP2O /P27	ANI6 /AMP2- /P26	ANI5 /AMP1+ /P25	ANI4 /AMP1O /P24	ANI3 /AMP1- /P23	ANI2 /AMP0+ /P22	ANI1 /AMP0O /P21	ANI0 /AMP0- /P20
0	0	0	0	0	A	A	A	A	A	A	A	A	A	A	A	A
0	0	0	0	1	A	A	A	A	A	A	A	A	A	A	A	D
0	0	0	1	0	A	A	A	A	A	A	A	A	A	A	D	D
0	0	0	1	1	A	A	A	A	A	A	A	A	A	D	D	D
0	0	1	0	0	A	A	A	A	A	A	A	A	D	D	D	D
0	0	1	0	1	A	A	A	A	A	A	A	D	D	D	D	D
0	0	1	1	0	A	A	A	A	A	A	D	D	D	D	D	D
0	0	1	1	1	A	A	A	A	A	D	D	D	D	D	D	D
0	1	0	0	0	A	A	A	A	D	D	D	D	D	D	D	D
0	1	0	0	1	A	A	A	D	D	D	D	D	D	D	D	D
0	1	0	1	0	A	A	D	D	D	D	D	D	D	D	D	D
0	1	1	1	1	A	D	D	D	D	D	D	D	D	D	D	D
1	0	0	0	0	D	D	D	D	D	D	D	D	D	D	D	D
Other than the above					Setting prohibited											

Caution Set pins to be used with operational amplifiers in the input mode by using port mode registers 2 and 15 (PM2, PM15).

(4) Port mode registers 2, 15 (PM2, PM15)

When using ANI0/AMP0-/P20 to ANI7/AMP2O/P27, ANI8/AMP2+/P150 to ANI10/P152, and ANI15/AV_{REFM}/P157 pins for analog input port, set PM20 to PM27, PM150 to PM152, and P157 to 1. The output latches of P20 to P27, P150 to P152 and P157 at this time may be 0 or 1.

If PM20 to PM27, PM150 to PM152, and PM157 are set to 0, they cannot be used as analog input port pins.

PM2 and PM15 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Caution If a pin is set as an analog input port, not the pin level but "0" is always read.

Figure 12-5. Formats of Port Mode Registers 2, 15 (PM2, PM15)

Address: FFF22H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	PM20

Address: FFF2FH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM15	PM157	1	1	1	1	PM152	PM151	PM150

PMmn	Pmn pin I/O mode selection (mn = 20 to 27, 150 to 152, 157)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

The ANI0/AMP0-/P20 to ANI7/AMP2O/P27, ANI8/AMP2+/P150 to ANI10/P152, and ANI15/AVREFM/P157 pins are as shown below depending on the settings of ADPC, ADS, PM2, PM15, OAENn bit and ADREF bit.

Table 12-2. Setting Functions of ANI0/AMP0-/P20, ANI2/AMP0+/P22, ANI3/AMP1-/P23, ANI5/AMP1+/P25, ANI6/AMP2-/P26, and ANI8/AMP2+/P150 Pins

ADPC register	PM2 and PM15 registers	OAENn bit	ADS register	ANI0/AMP0-/P20, ANI2/AMP0+/P22, ANI3/AMP1-/P23, ANI5/AMP1+/P25, ANI6/AMP2-/P26, and ANI8/AMP2+/P150 Pins
Digital I/O selection	Input mode	0	–	Digital input
		1	–	Setting prohibited
	Output mode	0	–	Digital output
		1	–	Setting prohibited
Analog input selection	Input mode	0	Selects ANI.	Analog input (to be converted)
			Does not select ANI.	Analog input (not to be converted)
		1	Selects ANI.	Setting prohibited
			Does not select ANI.	Operational amplifier input
	Output mode	–	–	Setting prohibited

Caution When an operational amplifier is used, AMPn+, AMPn–, and AMPnO pins are used, so the alternative analog input functions cannot be used.

Remark n = 0 to 2

Table 12-3. Setting Functions of ANI1/AMP0O/P21, ANI4/AMP1O/P24, and ANI7/AMP2O/P27 Pins

ADPC register	PM2 register	OAENn bit	ADS register	ANI1/AMP0O/P21, ANI4/AMP1O/P24, and ANI7/AMP2O/P27 Pins
Digital I/O selection	Input mode	0	–	Digital input
		1	–	Setting prohibited
	Output mode	0	–	Digital output
		1	–	Setting prohibited
Analog input selection	Input mode	0	Selects ANI.	Analog input (to be converted)
			Does not select ANI.	Analog input (not to be converted)
		1	Selects ANI.	Operational amplifier output (not to be converted)
			Does not select ANI.	Operational amplifier output (to be converted)
	Output mode	–	–	Setting prohibited

Caution When an operational amplifier is used, AMPn+, AMPn–, and AMPnO pins are used, so the alternative analog input functions cannot be used. The operational amplifier output signals, however, can be used as analog inputs.

Remark n = 0 to 2

Table 12-4. Setting Functions of ANI9/P151 and ANI10/AM152 Pins

ADPC register	PM15 register	ADS register	ANI9/P151 and ANI10/AM152 Pins
Digital I/O selection	Input mode	–	Digital input
	Output mode	–	Digital output
Analog input selection	Input mode	Selects ANI.	Analog input (to be converted)
		Does not select ANI.	Analog input (not to be converted)
	Output mode	–	Setting prohibited

Table 12-5. Setting Functions of ANI15/AV$_{REFM}$/P157 Pin

ADPC register	PM15 register	ADREF bit	ADS register	ANI15/AV$_{REFM}$/P157 Pin
Digital I/O selection	Input mode	0	–	Digital input
		1	–	Setting prohibited
	Output mode	0	–	Digital output
		1	–	Setting prohibited
Analog input selection	Input mode	0	Selects ANI.	Analog input (to be converted)
			Does not select ANI.	Analog input (not to be converted)
		1	–	Negative reference voltage input of A/D converter
	Output mode	–	–	Setting prohibited

12.4 Operation of Operational Amplifiers

The operational amplifiers 0 to 2 have the following mode.

- Single AMP mode (operational amplifiers 0 to 2)

12.4.1 Single AMP Mode

In single amplifier mode, the difference in potential of analog voltages input from two pins (AMPn− and AMPn+ pins) is amplified and the amplified voltage is output from the AMPnO pin. The gain is determined by externally connecting a resistor or the like.

The amplified voltage can be used as an analog input of the A/D converter, because the AMPnO pin is alternatively used with analog input pin of the A/D converter.

The procedure for starting operation in single amplifier mode is described below.

<1> Set bit 5 (ADCEN) of peripheral enable register 0 (PER0) to 1 to start the supply of the input clock to the operational amplifier.
<2> Use the A/D port configuration register (ADPC) to set the pins (AMPn−, AMPn+, AMPnO) to be used in single amplifier mode as analog inputs.
<3> Use the port mode register x (PMx) to set the pins (AMPn−, AMPn+, AMPnO) to be used in single amplifier mode to input mode.
<4> Set (1) the OAENn bit of operational amplifier control register (OAC) and enable operation in single amplifier mode.
<5> Use software to wait until the operational amplifier stabilizes (turn-on time: 20 μs (max.)).

Caution To use as an input of the A/D converter a voltage that has been amplified in single amplifier mode, enable operation in single amplifier mode before selecting an analog input channel by using the ADS register.

Remark n = 0 to 2, x = 2, 15

CHAPTER 13 VOLTAGE REFERENCE

13.1 Function of Voltage Reference

The Voltage Reference has the following mode.

- **Reference voltage output mode**
  A reference voltage is output from the V_{REFOUT} pin. Furthermore, the generated reference voltage is supplied to the internal A/D and D/A converters. 2.0 V (typ.) or 2.5 V (typ.) can be selected as the output voltage.

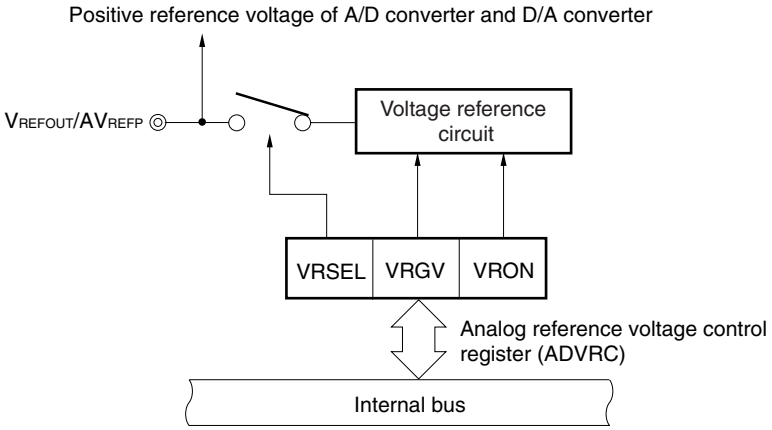
13.2 Configuration of Voltage Reference

The voltage reference consists of the following hardware.

Table 13-1. Configuration of Voltage Reference

Item	Configuration
Reference voltage output	V_{REFOUT} pin
Control registers	Peripheral enable register 0 (PER0) Analog reference voltage control register (ADVRC)

Figure 13-1. Block Diagram of Voltage Reference



13.3 Voltage Reference Used in Voltage Reference

The voltage reference uses the following two registers.

- Peripheral enable register 0 (PER0)
- Analog reference voltage control register (ADVRC)

(1) Peripheral enable register 0 (PER0)

PER0 is used to enable or disable use of each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When the voltage reference is used, be sure to set bit 5 (ADCEN) of this register to 1.

PER0 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 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	DACEN	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

ADCEN	Control of A/D converter, operational amplifier, and voltage reference input clock
0	Stops input clock supply. • SFR used by the A/D converter, operational amplifier, and voltage reference cannot be written. • The A/D converter, operational amplifier, and voltage reference are in the reset status.
1	Supplies input clock. • SFR used by the A/D converter, operational amplifier, and voltage reference can be read and written.

Caution When setting voltage reference, be sure to set ADCEN to 1 first. If ADCEN = 0, writing to a control register of voltage reference is ignored, and, even if the register is read, only the default value is read.

(2) Analog reference voltage control register (ADVRC)

This register is used to select the reference voltage supplies of the A/D and D/A converters, control the operation of the input gate voltage boost circuit for the A/D converter, and control the voltage reference (VR) operation.

ADVRC 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 Reference Voltage Control reegister (ADVRC)

Address: FFF36H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ADVRC	ADREF	0	0	0	VRSEL	0	VRGV	VRON

ADREF	Negative reference voltage supply of A/D converter selection
0	AV_{SS}
1	AV_{REFM} (external voltage reference input)

VRSEL	VRGV	VRON	Positive reference voltage supplies selection of A/D and D/A converters	Operation control of voltage reference	Output voltage selection of voltage reference	Operation control of input gate voltage boost circuit for A/D converter	Relationship with the conversion mode used
0	0	0	AV_{REFP} (external voltage reference input)	Stops operation (Hi-Z)	2.5 V	Stops operation	Can be set in normal mode 1.
0	1	0			2.0 V	Enables operation	Can be set in normal mode 2 or low voltage mode.
1	0	0	V_{REFOUT} (voltage reference output)	Stops operation (pull-down output)	2.5 V	Stops operation	–
1	0	1		Enables operation	2.5 V	Enables operation	Can be set in normal mode 2 or low voltage mode.
1	1	0		Stops operation (pull-down output)	2.0 V		–
1	1	1		Enables operation	2.0 V		Can be set in normal mode 2 or low voltage mode.
Other than the above			Setting prohibited				

Cautions 1. During voltage reference operation, be sure to connect a tantalum capacitor (capacitance: 10 μF±30 %, ESR: 2 Ω (max.), ESL: 10 nH (max.)) and a ceramic capacitor (capacitance: 0.1 μF±30 %, ESR: 2 Ω (max.), ESL: 10 nH (max.)) to the V_{REFOUT}/AV_{REFP} pin for stabilizing the reference voltage. Furthermore, do not apply a voltage from the V_{REFOUT}/AV_{REFP} pin during voltage reference operation.

2. To use voltage reference output (V_{REFOUT}) to the positive reference voltage of the A/D converter (AD_{REFP}) and the positive reference voltage of the D/A converter (DA_{REFP}), be sure to set VRON to 1 after setting VRSEL to 1.

Cautions 3. Rewriting DACSWn (n = 0, 1) during A/D conversion is prohibited when both the positive reference voltage of the A/D converter (AD_{REFP}) and the positive reference voltage of the D/A converter (DA_{REFP}) are the voltage reference output (V_{REFOUT}) (VRSEL = 1 and DAREF = 1). Rewrite it when conversion operation is stopped (ADCS = 0).

4. Do not change the output voltage of the reference voltage by using VRGV during the voltage reference operation (VRON = 1).

13.4 Operation of Voltage Reference

The voltage reference has the following mode.

- **Reference voltage output mode**
  A reference voltage is output from the V_{REFOUT} pin. Furthermore, the generated reference voltage is supplied to the internal A/D and D/A converters. 2.0 V (TYP.) or 2.5 V (TYP.) can be selected as the output voltage.

13.4.1 Reference voltage output mode

The procedure for starting operation is described below.

<1> Set bit 5 (ADCEN) of peripheral enable register 0 (PER0) to 1 to start the supply of the input clock to the voltage reference.
<2> Set bit 3 (VRSEL) of the analog reference voltage control register (ADVRC) to 1. The positive reference voltage both the A/D and D/A converters or only the A/D converter is set to voltage reference output.
<3> Specify the reference voltage value by using bit 1 (VRGV) of ADVRC.
<4> Enable voltage reference operation by setting bit 0 (VRON) of ADVRC to 1.
<5> Use software to wait until the voltage reference operation stabilizes (settling time: 17 ms (MAX.)).

13.5 Caution for Voltage Reference

Observe the following caution when using the voltage reference.

- The V_{REFOUT} output voltage can be used only as the positive reference voltage of the internal A/D and D/A converters of the microcontroller. Do not connect an external circuit other than a tantalum capacitor (capacitance: 10 μF$\pm$30 %, ESR: 2 Ω (max.), ESL: 10 nH (max.)) and a ceramic capacitor (capacitance: 0.1 μF$\pm$30 %, ESR: 2 Ω (max.), ESL: 10 nH (max.)) for stabilizing the reference voltage (target capacitance: 10 μF and 0.1 μF) to the V_{REFOUT} pin.

CHAPTER 14 SERIAL ARRAY UNIT

The serial array unit has four serial channels per unit and can use two or more of various serial interfaces (3-wire serial (CSI), UART, and simplified I^2C) in combination.

Function assignment of each channel supported by the 78K0R/KE3-A microcontrollers is as shown below (channels 2 and 3 of unit 1 are dedicated to UART3 (supporting LIN-bus)).

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	–
	1	–		–
	2	CSI10	UART1	IIC10
	3	–		–
1	0	CSI20	UART2	IIC20
	1	–		–
	2	–	UART3 (supporting LIN-bus)	–
	3	–		–

(Example of combination) When “UART0” is used for channels 0 and 1 of unit 0, CSI00 cannot be used, but CSI10, UART1, or IIC10 can be used.

14.1 Functions of Serial Array Unit

Each serial interface supported by the 78K0R/KE3-A microcontrollers has the following features.

14.1.1 3-wire serial I/O (CSI00, CSI10, CSI20)

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
- Level setting of transmit/receive data

[Clock control]

- Master/slave selection
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

14.1.2 UART (UART0, UART1, UART2, UART3)

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. It transmits or receives data in asynchronization with the party of communication (by using an internal baud rate). Full-duplex UART communication can be realized by using two channels, one dedicated to transmission (even channel) and the other to reception (odd channel).

[Data transmission/reception]

- Data length of 5, 7, or 8 bits
- Select the MSB/LSB first
- Level setting of transmit/receive data and select of reverse
- Parity bit appending and parity check functions
- Stop bit appending

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

The LIN-bus is accepted in UART3 (2 and 3 channels of unit 1)

[LIN-bus functions]

- Wakeup signal detection
- Sync break field (SBF) detection
- Sync field measurement, baud rate calculation

} External interrupt (INTP0) or timer array unit (TAU) is used.

14.1.3 Simplified I²C (IIC10, IIC20)

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This simplified I²C is designed for single communication with a device such as EEPROM, flash memory, or A/D converter, and therefore, it functions only as a master and does not have a function to detect wait states.

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.)
- Manual generation of start condition and stop condition

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- Parity error (ACK error)

[Functions not supported by simplified I²C]

- Slave transmission, slave reception
- Arbitration loss detection function
- Wait detection functions

Note An ACK is not output when the last data is being received by writing 0 to the SOEmn (SOEm register) bit and stopping the output of serial communication data. See **14.7.3 (2) Processing flow** for details.

Remarks 1. To use an I²C bus of full function, see **CHAPTER 15 SERIAL INTERFACE IICA**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 02, 10

14.2 Configuration of Serial Array Unit

Serial array unit includes the following hardware.

Table 14-1. Configuration of Serial Array Unit

Item	Configuration
Shift register	8 bits
Buffer register	Lower 8 bits of serial data register mn (SDRmn)[Note]
Serial clock I/Os	$\overline{SCK00}$, $\overline{SCK10}$, $\overline{SCK20}$ pins (for 3-wire serial I/O), SCL10, SCL20 pins (for simplified I^2C)
Serial data inputs	SI00, SI10, SI20 pins (for 3-wire serial I/O), RxD0, RxD1, RxD2 pins (for UART), RxD3 pin (for UART supporting LIN-bus)
Serial data outputs	SO00, SO10, SO20 pins (for 3-wire serial I/O), TxD0, TxD1, TxD2 pins (for UART), TxD3 pin (for UART supporting LIN-bus), output controller
Serial data I/Os	SDA10, SDA20 pins (for simplified I^2C)
Control registers	<Registers of unit setting block> • Peripheral enable register 0 (PER0) • Serial clock select register m (SPSm) • Serial channel enable status register m (SEm) • Serial channel start register m (SSm) • Serial channel stop register m (STm) • Serial output enable register m (SOEm) • Serial output register m (SOm) • Serial output level register m (SOLm) • Input switch control register (ISC) • Noise filter enable register 0 (NFEN0)
	<Registers of each channel> • Serial data register mn (SDRmn) • Serial mode register mn (SMRmn) • Serial communication operation setting register mn (SCRmn) • Serial status register mn (SSRmn) • Serial flag clear trigger register mn (SIRmn) • Port input mode register 1 (PIM1) • Port output mode registers 1, 8 (POM1, POM8) • Port mode registers 1, 5, 8 (PM1, PM5, PM8) • Port registers 1, 5, 8 (P1, P5, P8)

Note The lower 8 bits of the serial data register mn (SDRmn) can be read or written as the following SFR, depending on the communication mode.

- CSIp communication … SIOp (CSIp data register)
- UARTq reception … RXDq (UARTq receive data register)
- UARTq transmission … TXDq (UARTq transmit data register)
- IICr communication … SIOr (IICr data register)

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3),
p: CSI number (p = 00, 10, 20), q: UART number (q = 0 to 3), r: IIC number (r = 10, 20)

Figure 14-1 shows the block diagram of serial array unit 0.

Figure 14-1. Block Diagram of Serial Array Unit 0

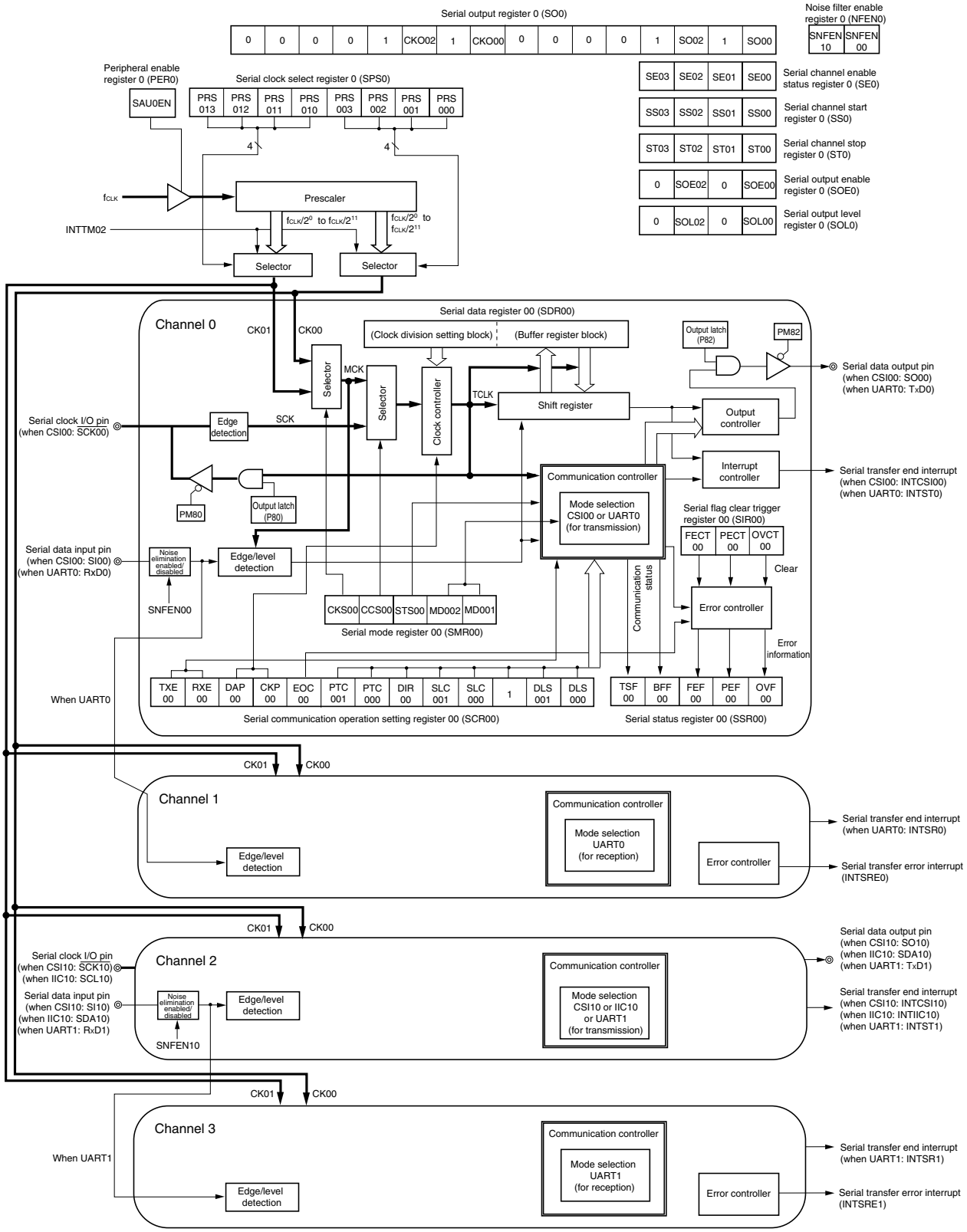
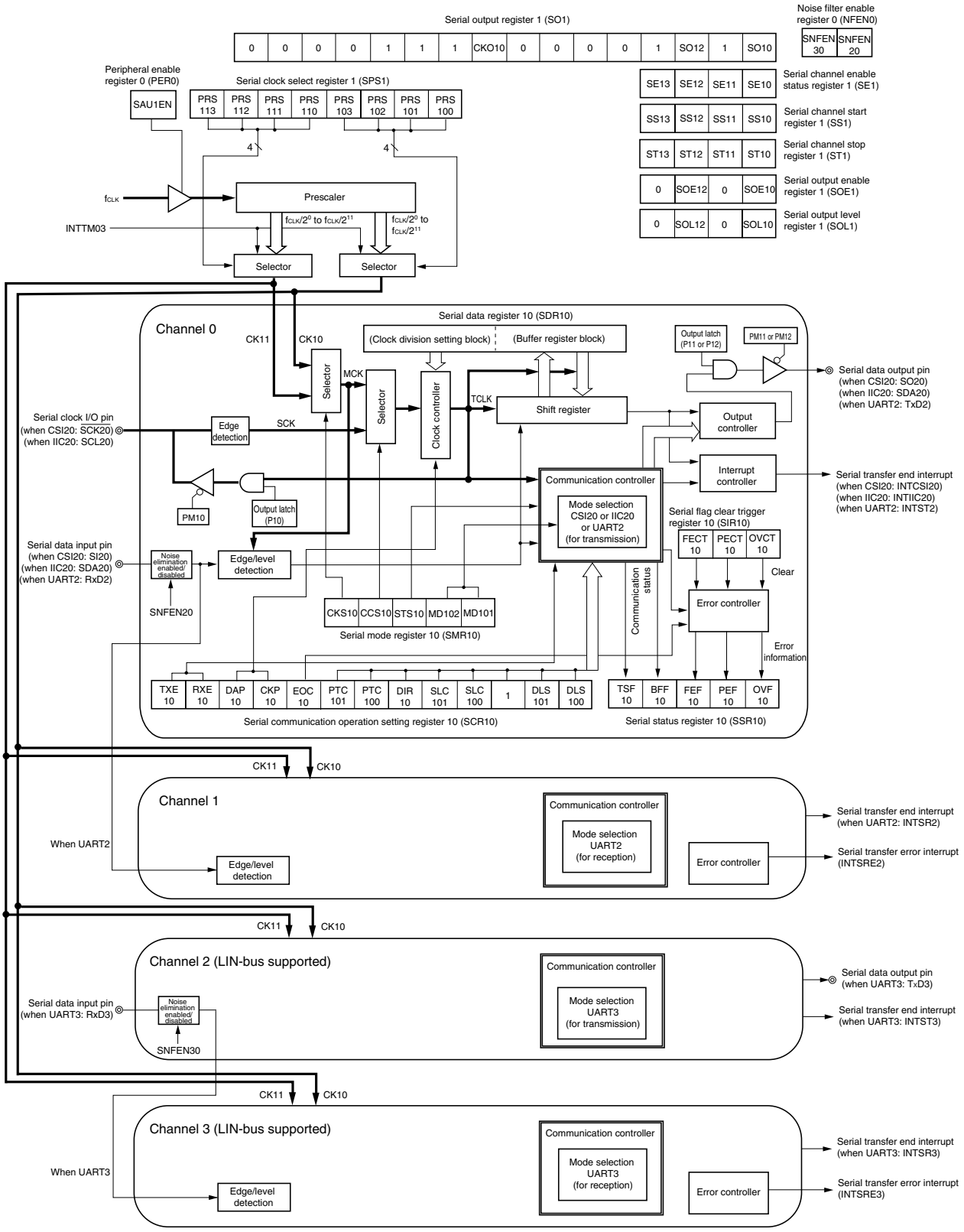


Figure 14-2 shows the block diagram of serial array unit 1.

Figure 14-2. Block Diagram of Serial Array Unit 1



(1) Shift register

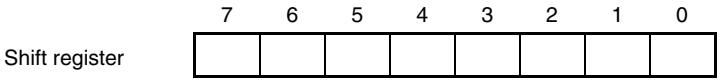
This is an 8-bit register that converts parallel data into serial data or vice versa.

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 bits of serial data register mn (SDRmn).



(2) Lower 8 bits of the serial data register mn (SDRmn)

SDRmn is the transmit/receive data register (16 bits) of channel n. Bits 7 to 0 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 (MCK).

When data is received, parallel data converted by the shift register is stored in the lower 8 bits. When data is to be transmitted, set transmit to be transferred to the shift register to the lower 8 bits.

The data stored in the lower 8 bits of this register is as follows, depending on the setting of bits 0 to 2 (DLSmn0 to DLSmn2) of the SCRmn register, regardless of the output sequence of the data.

- 5-bit data length (stored in bits 0 to 4 of SDRmn register) (settable in UART mode only)
- 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)

SDRmn can be read or written in 16-bit units.

The lower 8 bits of SDRmn of SDRmn can be read or written[Note] 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 this register to 0000H.

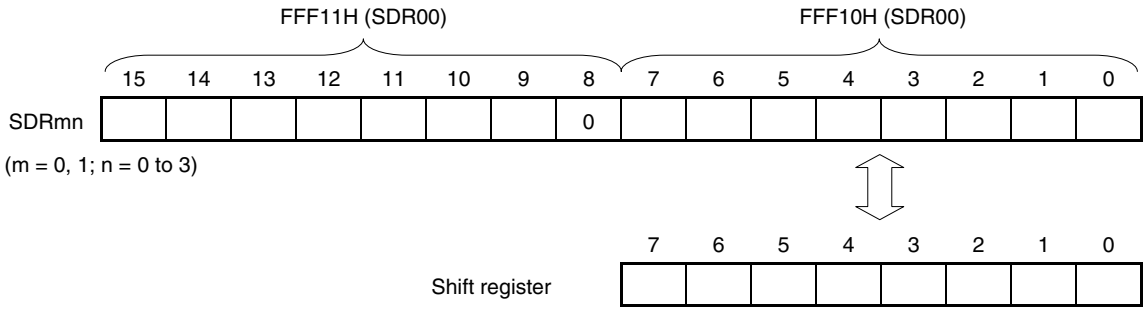
Note Writing in 8-bit units is prohibited when the operation is stopped (SEmn = 0).

Remarks 1. After data is received, “0” is stored in bits 0 to 7 in bit portions that exceed the data length.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3),
p: CSI number (p = 00, 10, 20), q: UART number (q = 0 to 3), r: IIC number (r = 10, 20)

Figure 14-3. Format of Serial Data Register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01), FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03), FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11), FFF14H, FFF15H (SDR12), FFF16H, FFF17H (SDR13) After reset: 0000H R/W



Caution Be sure to clear bit 8 to "0".

Remarks 1. For the function of the higher 7 bits of SDRmn, see **14.3 Registers Controlling Serial Array Unit**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3),
p: CSI number (p = 00, 10, 20), q: UART number (q = 0 to 3), r: IIC number (r = 10, 20)

14.3 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 status register mn (SSRmn)
- Serial flag clear trigger register mn (SIRmn)
- 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 level register m (SOLm)
- Serial output register m (SOm)
- Input switch control register (ISC)
- Noise filter enable register 0 (NFEN0)
- Port input mode register 1 (PIM1)
- Port output mode registers 1, 8 (POM1, POM8)
- Port mode registers 1, 5, 8 (PM1, PM5, PM8)
- Port registers 1, 5, 8 (P1, P5, P8)

Remark m: Unit number (m = 0, 1)
n: Channel number (n = 0 to 3)

(1) Peripheral enable register 0 (PER0)

PER0 is used to enable or disable use of each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

When serial array unit 0 is used, be sure to set bit 2 (SAU0EN) of this register to 1.

When serial array unit 1 is used, be sure to set bit 3 (SAU1EN) of this register to 1.

PER0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 14-4. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	DACEN	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

SAUmEN	Control of serial array unit m input clock
0	Stops supply of input clock. • SFR used by serial array unit m cannot be written. • Serial array unit m is in the reset status.
1	Supplies input clock. • SFR used by serial array unit m can be read/written.

Cautions 1. When setting serial array unit m, be sure to set SAUmEN to 1 first. 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 input switch control register (ISC), noise filter enable register 0 (NFEN0), port input mode register 1 (PIM1), port output mode registers 1, 8 (POM1, POM8), port mode registers 1, 5, 8 (PM1, PM5, PM8), and port registers 1, 5, 8 (P1, P5, P8)).

2. After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

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

(2) Serial clock select register m (SPSm)

SPSm is a 16-bit register that is used to select two types of operation clocks (CKm0, CKm1) that are commonly supplied to each channel. CKm1 is selected by bits 7 to 4 of SPSm, and CKm0 is selected by bits 3 to 0.

Rewriting SPSm is prohibited when the register is in operation (when SEmn = 1).

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

The lower 8 bits of SPSm can be set with an 8-bit memory manipulation instruction with SPSmL.

Reset signal generation clears this register to 0000H.

Figure 14-5. Format of Serial Clock Select Register m (SPSm)

Address: F0126H, F0127H (SPS0), F0166H, F0167H (SPS1) 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 mp3	PRS mp2	PRS mp1	PRS mp0	Section of operation clock (CKmp) Note 1				
					f_{CLK} = 2 MHz	f_{CLK} = 5 MHz	f_{CLK} = 10 MHz	f_{CLK} = 20 MHz
0	0	0	0	f_{CLK}	2 MHz	5 MHz	10 MHz	20 MHz
0	0	0	1	$f_{CLK}/2$	1 MHz	2.5 MHz	5 MHz	10 MHz
0	0	1	0	$f_{CLK}/2^2$	500 kHz	1.25 MHz	2.5 MHz	5 MHz
0	0	1	1	$f_{CLK}/2^3$	250 kHz	625 kHz	1.25 MHz	2.5 MHz
0	1	0	0	$f_{CLK}/2^4$	125 kHz	313 kHz	625 kHz	1.25 MHz
0	1	0	1	$f_{CLK}/2^5$	62.5 kHz	156 kHz	313 kHz	625 kHz
0	1	1	0	$f_{CLK}/2^6$	31.3 kHz	78.1 kHz	156 kHz	313 kHz
0	1	1	1	$f_{CLK}/2^7$	15.6 kHz	39.1 kHz	78.1 kHz	156 kHz
1	0	0	0	$f_{CLK}/2^8$	7.81 kHz	19.5 kHz	39.1 kHz	78.1 kHz
1	0	0	1	$f_{CLK}/2^9$	3.91 kHz	9.77 kHz	19.5 kHz	39.1 kHz
1	0	1	0	$f_{CLK}/2^{10}$	1.95 kHz	4.88 kHz	9.77 kHz	19.5 kHz
1	0	1	1	$f_{CLK}/2^{11}$	977 Hz	2.44 kHz	4.88 kHz	9.77 kHz
1	1	1	1	INTTM02 if m = 0, INTTM03 if m = 1 Note 2				
Other than above				Setting prohibited				

Notes 1. When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (STm = 000FH) the operation of the serial array unit (SAUm). When selecting INTTM02 and INTTM03 for the operation clock, also stop the timer array unit (TAU0) (TT0 = 00FFH).

2. SAUm can be operated at a fixed division ratio of the subsystem clock, regardless of the f_{CLK} frequency (main system clock, subsystem clock), by operating the interval timer for which $f_{SUB}/4$ has been selected as the count clock (setting TIS02 (if m = 0) or TIS03 (if m = 1) of the TIS0 register to 1) and selecting INTTM02 and INTTM03 by using the SPSm register in channels 2 and 3 of TAU0. When changing f_{CLK}, however, SAUm and TAU0 must be stopped as described in Note 1 above.

Cautions 1. Be sure to clear bits 15 to 8 to "0".

2. After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remarks 1. f_{CLK}: CPU/peripheral hardware clock frequency
f_{SUB}: Subsystem clock frequency

2. m: Unit number (m = 0, 1), p = 0, 1

(3) Serial mode register mn (SMRmn)

SMRmn is a register that sets an operation mode of channel n. It is also used to select an operation clock (MCK), specify whether the serial clock (SCK) may be input or not, set a start trigger, an operation mode (CSI, UART, or I^2C), and an interrupt source. This register is also used to invert the level of the receive data only in the UART mode.

Rewriting SMRmn is prohibited when the register is in operation (when SEmn = 1). However, the MDmn0 bit can be rewritten during operation.

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

Reset signal generation sets this register to 0020H.

Figure 14-6. Format of Serial Mode Register mn (SMRmn) (1/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03), F0150H, F0151H (SMR10), F0152H, F0153H (SMR11), F0154H, F0155H (SMR12), F0156H, F0157H (SMR13) 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	0	SISmn0	1	0	0	MDmn2	MDmn1	MDmn0

CKSmn	Selection of operation clock (MCK) of channel n
0	Prescaler output clock CKm0 set by SPSm register
1	Prescaler output clock CKm1 set by SPSm register
Operation clock 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 (TCLK) is generated.	

CCSmn	Selection of transfer clock (TCLK) of channel n
0	Divided operation clock MCK specified by CKSmn bit
1	Clock input from SCK pin (slave transfer in CSI mode)
Transfer clock TCLK is used for the shift register, communication controller, output controller, interrupt controller, and error controller. When CCSmn = 0, the division ratio of MCK is set by the higher 7 bits of the SDRmn register.	

STSmn	Selection of start trigger source
0	Only software trigger is valid (selected for CSI, UART transmission, and simplified I^2C).
1	Valid edge of RxD pin (selected for UART reception)
Transfer is started when the above source is satisfied after 1 is set to the SSm register.	

Caution Be sure to clear bits 13 to 9, 7, 4, and 3 to "0". Be sure to set bit 5 to "1".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 14-6. Format of Serial Mode Register mn (SMRmn) (2/2)

Address: F0110H, F0111H (SMR00) to F0116H, F0117H (SMR03), After reset: 0020H R/W
F0150H, F0151H (SMR10), F0152H, F0153H (SMR11),
F0154H, F0155H (SMR12), F0156H, F0157H (SMR13)

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	0	SIS mn0	1	0	0	MD mn2	MD mn1	MD mn0

SIS mn0	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	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
For successive transmission, the next transmit data is written by setting MDmn0 to 1 when SDRmn data has run out.	

Caution Be sure to clear bits 13 to 9, 7, 4, and 3 to "0". Be sure to set bit 5 to "1".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

(4) Serial communication operation setting register mn (SCRmn)

SCRmn is a communication operation setting register of channel n. It is used to set a data transmission/reception mode, phase of data and clock, whether an error signal is to be masked or not, parity bit, start bit, stop bit, and data length.

Rewriting SCRmn is prohibited when the register is in operation (when SEmn = 1).

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

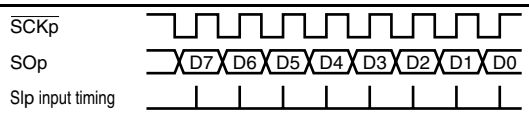
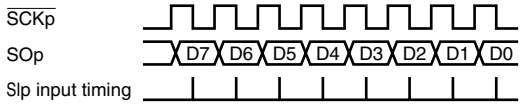
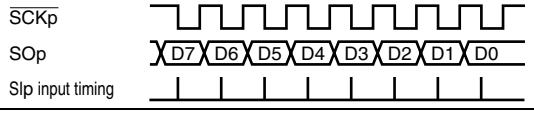
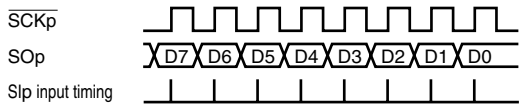
Reset signal generation sets this register to 0087H.

Figure 14-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (1/3)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03), F0158H, F0159H (SCR10), F015AH, F015BH (SCR11), F015CH, F015DH (SCR12), F015EH, F015FH (SCR13) 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	SLC mn1	SLC mn0	0	1	DLS mn1	DLS mn0

TXE mn	RXE mn	Setting of operation mode of channel n
0	0	Does not start communication.
0	1	Reception only
1	0	Transmission only
1	1	Transmission/reception

DAP mn	CKP mn	Selection of data and clock phase in CSI mode	Type
0	0		1
0	1		2
1	0		3
1	1		4
Be sure to set DAPmn, CKPmn = 0, 0 in the UART mode and simplified I^2C mode.			

Caution Be sure to clear bits 3, 6, and 11 to "0". Be sure to set bit 2 to "1".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), p: CSI number (p = 00, 10, 20)

Figure 14-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (2/3)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03), F0158H, F0159H (SCR10), F015AH, F015BH (SCR11), F015CH, F015DH (SCR12), F015EH, F015FH (SCR13) 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	SLC mn1	SLC mn0	0	1	DLS mn1	DLS mn0

EOC mn	Selection of masking of error interrupt signal (INTSREx (x = 0 to 3))
0	Masks error interrupt INTSREx (INTSRx is not masked).
1	Enables generation of error interrupt INTSREx (INTSRx is masked if an error occurs).
Set EOCmn = 0 in the CSI mode, simplified I^2C mode, and during UART transmission. Set EOCmn = 1 during UART reception.	

PTC mn1	PTC mn0	Setting of parity bit in UART mode	
		Transmission	Reception
0	0	Does not output the parity bit.	Receives without parity
0	1	Outputs 0 parity.	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 CSI mode and simplified I^2C mode.			

DIR mn	Selection of data transfer sequence in CSI and UART modes
0	Inputs/outputs data with MSB first.
1	Inputs/outputs data with LSB first.
Be sure to clear DIRmn = 0 in the simplified I^2C mode.	

SLC mn1	SLC mn0	Setting of stop bit in UART mode
0	0	No stop bit
0	1	Stop bit length = 1 bit
1	0	Stop bit length = 2 bits
1	1	Setting prohibited
When the transfer end interrupt is selected, the interrupt is generated when all stop bits have been completely transferred. Set 1 bit (SLCmn1, SLCmn0 = 0, 1) during UART reception and in the simplified I^2C mode. Set no stop bit (SLCmn1, SLCmn0 = 0, 0) in the CSI mode.		

Caution Be sure to clear bits 3, 6, and 11 to "0". Be sure to set bit 2 to "1".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 14-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (3/3)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03), After reset: 0087H R/W
F0158H, F0159H (SCR10), F015AH, F015BH (SCR11),
F015CH, F015DH (SCR12), F015EH, F015FH (SCR13)

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	SLC mn1	SLC mn0	0	1	DLS mn1	DLS mn0

DLS mn1	DLS mn0	Setting of data length in CSI and UART modes
0	0	5-bit data length (stored in bits 0 to 4 of SDRmn register) (settable in UART mode only)
1	0	7-bit data length (stored in bits 0 to 6 of SDRmn register)
1	1	8-bit data length (stored in bits 0 to 7 of SDRmn register)
Other than above		Setting prohibited
Be sure to set DLSmn0 = 1 in the simplified I^2C mode.		

Caution Be sure to clear bits 3, 6, and 11 to "0". Be sure to set bit 2 to "1".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

(5) Higher 7 bits of the serial data register mn (SDRmn)

SDRmn is the transmit/receive data register (16 bits) of channel n. Bits 7 to 0 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 (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 SDRmn is used as the transfer clock.

For the function of the lower 8 bits of SDRmn, see **14.2 Configuration of Serial Array Unit**.

SDRmn 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 bits of SDRmn. When SDRmn is read during operation, 0 is always read.

Reset signal generation clears this register to 0000H.

Figure 14-8. Format of Serial Data Register mn (SDRmn)

Address: FFF10H, FFF11H (SDR00), FFF12H, FFF13H (SDR01), FFF44H, FFF45H (SDR02), FFF46H, FFF47H (SDR03), FFF48H, FFF49H (SDR10), FFF4AH, FFF4BH (SDR11), FFF14H, FFF15H (SDR12), FFF16H, FFF17H (SDR13) 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								0								

SDRmn[15:9]							Transfer clock setting by dividing the operating clock (MCK)
0	0	0	0	0	0	0	MCK/2
0	0	0	0	0	0	1	MCK/4
0	0	0	0	0	1	0	MCK/6
0	0	0	0	0	1	1	MCK/8
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
1	1	1	1	1	1	0	MCK/254
1	1	1	1	1	1	1	MCK/256

Cautions 1. Be sure to clear bit 8 to "0".

2. Setting SDRmn[15:9] = (0000000B, 0000001B) is prohibited when UART is used.

<R> **3. Setting SDRmn[15:9] = 0000000B is prohibited when the simplified I2C is used. Set SDRmn[15:9] to 0000001B or greater.**

Remarks 1. For the function of the lower 8 bits of SDRmn, see **14.2 Configuration of Serial Array Unit**.

2. m: Unit number (m = 0, 1)
n: Channel number (n = 0 to 3)

(6) Serial status register mn (SSRmn)

SSRmn 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.

SSRmn can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of SSRmn can be set with an 8-bit memory manipulation instruction with SSRmnL.

Reset signal generation clears this register to 0000H.

Figure 14-9. Format of Serial Status Register mn (SSRmn) (1/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03), F0140H, F0141H (SSR10), F0142H, F0143H (SSR11), F0144H, F0145H (SSR12), F0146H, F0147H (SSR13) — 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 Note	OVF mn Note

TSF mn	Communication status indication flag of channel n
0	Communication is not under execution.
1	Communication is under execution.
Because this flag is an updating flag, it is automatically cleared when the communication operation is completed. This flag is cleared also when the STmn/SSmn bit is set to 1.	

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.
This is an updating flag. It is automatically cleared when transfer from the SDRmn register to the shift register is completed. During reception, it is automatically cleared when data has been read from the SDRmn register. This flag is cleared also when the STmn/SSmn bit is set to 1. This flag is automatically set if transmit data is written to the SDRmn register when the TXEmn bit of the SCRmn register = 1 (transmission or reception mode in each communication mode). It is automatically set if receive data is stored in the SDRmn register when the RXEmn bit of the SCRmn register = 1 (transmission or reception mode in each communication mode). It is also set in case of a reception error. If data is written to the SDRmn register when BFFmn = 1, the transmit/receive data stored in the register is discarded and an overrun error (OVFmn = 1) is detected.	

Note Only SSR12 register do not have FET12, PET12, and OVF12.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 14-9. Format of Serial Status Register mn (SSRmn) (2/2)

Address: F0100H, F0101H (SSR00) to F0106H, F0107H (SSR03), After reset: 0000H R
F0140H, F0141H (SSR10), F0142H, F0143H (SSR11),
F0144H, F0145H (SSR12), F0146H, F0147H (SSR13)

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSRmn	0	0	0	0	0	0	0	0	0	TSFmn	BFFmn	0	0	FEFmn Note	PEFmn Note	OVFmn Note

FEFmn	Framing error detection flag of channel n
0	No error occurs.
1	A framing error occurs during UART reception. <Framing error cause> A framing error occurs if the stop bit is not detected upon completion of UART reception.
This is a cumulative flag and is not cleared until 1 is written to the FECTmn bit of the SIRmn register.	

PEFmn	Parity error detection flag of channel n
0	Error does not occur.
1	A parity error occurs during UART reception or ACK is not detected during I^2C transmission. <Parity error cause> • A parity error occurs if the parity of transmit data does not match the parity bit on completion of UART reception. • ACK is not detected if the ACK signal is not returned from the slave in the timing of ACK reception during I^2C transmission.
This is a cumulative flag and is not cleared until 1 is written to the PECTmn bit of the SIRmn register.	

OVFmn	Overrun error detection flag of channel n
0	No error occurs.
1	An overrun error occurs. <Causes of overrun error> • Receive data stored in the SDRmn register is not read and transmit data is written or the next receive data is written. • Transmit data is not ready for slave transmission or reception in the CSI mode.
This is a cumulative flag and is not cleared until 1 is written to the OVCTmn bit of the SIRmn register.	

Note Only SSR12 register do not have FET12, PET12, and OVF12.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

(7) Serial flag clear trigger register mn (SIRmn)

SIRmn is a trigger register that is used to clear each error flag of channel n.

When each bit (FECTmn, PECTmn, OVCTmn) of this register is set to 1, the corresponding bit (FEFmn, PEFmn, OVFmn) of serial status register mn is cleared to 0. Because SIRmn is a trigger register, it is cleared immediately when the corresponding bit of SSRmn is cleared.

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

The lower 8 bits of SIRmn can be set with an 8-bit memory manipulation instruction with SIRmnL.

Reset signal generation clears this register to 0000H.

Figure 14-10. Format of Serial Flag Clear Trigger Register mn (SIRmn)

Address: F0108H, F0109H (SIR00) to F010EH, F010FH (SIR03), F0148H, F0149H (SIR10), F014AH, F014BH (SIR11), F014EH, F014FH (SIR13) 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	PECTmn	OVCTmn

FECTmn	Clear trigger of framing error of channel n
0	No trigger operation
1	Clears the FEFmn bit of the SSRmn register to 0.

PECTmn	Clear trigger of parity error flag of channel n
0	No trigger operation
1	Clears the PEFmn bit of the SSRmn register to 0.

OVCTmn	Clear trigger of overrun error flag of channel n
0	No trigger operation
1	Clears the OVFmn bit of the SSRmn register to 0.

Caution Be sure to clear bits 15 to 3 to "0".

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

2. When the SIRmn register is read, 0000H is always read.

(8) Serial channel enable status register m (SEm)

SEm indicates whether data transmission/reception operation of each channel is enabled or stopped.

When 1 is written a bit of serial channel start register 0 (SSm), the corresponding bit of this register is set to 1. When 1 is written a bit of serial channel stop register 0 (STm), the corresponding bit is cleared to 0.

Channel n that is enabled to operate cannot rewrite by software the value of CKOmn of the serial output register m (SOm) to be described below, and a value reflected by a communication operation is output from the serial clock pin.

Channel n that stops operation can set the value of CKOmn 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.

SEm can be read by a 16-bit memory manipulation instruction.

The lower 8 bits of SEm can be set with a 1-bit or 8-bit memory manipulation instruction with SEmL.

Reset signal generation clears this register to 0000H.

Figure 14-11. Format of Serial Channel Enable Status Register m (SEm)

Address: F0120H, F0121H (SE0), F0160H, F0161H (SE1) After reset: 0000H R

Symbol	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	SEm2	SEm1	SEm0

SEmn	Indication of operation enable/stop status of channel n
0	Operation stops (stops with the values of the control register and shift register, and the statuses of the serial clock I/O pin, serial data output pin, and the FEF, PEF, and OVF error flags retained[Note]).
1	Operation is enabled.

Note Bits 6 and 5 (TSFmn, BFFmn) of the SSRmn register are cleared.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

(9) Serial channel start register m (SSm)

SSm is a trigger register that is used to enable starting communication/count by each channel.

When 1 is written a bit of this register (SSmn), the corresponding bit (SEmn) of serial channel enable status register m (SEm) is set to 1. Because SSmn is a trigger bit, it is cleared immediately when SEmn = 1.

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

The lower 8 bits of SSm can be set with a 1-bit or 8-bit memory manipulation instruction with SSmL.

Reset signal generation clears this register to 0000H.

Figure 14-12. Format of Serial Channel Start Register m (SSm)

Address: F0122H, F0123H (SS0), F0162H, F0163H (SS1) After reset: 0000H R/W

Symbol	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

SSmn	Operation start trigger of channel n
0	No trigger operation
1	Sets SEmn to 1 and enters the communication wait status (if a communication operation is already under execution, the operation is stopped and the start condition is awaited).

Caution Be sure to clear bits 15 to 4 to "0".

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

2. When the SSm register is read, 0000H is always read.

(10) Serial channel stop register m (STm)

STm is a trigger register that is used to enable stopping communication/count by each channel.

When 1 is written a bit of this register (STmn), the corresponding bit (SEmn) of serial channel enable status register m (SEm) is cleared to 0. Because STmn is a trigger bit, it is cleared immediately when SEmn = 0.

STm can set written by a 16-bit memory manipulation instruction.

The lower 8 bits of STm can be set with a 1-bit or 8-bit memory manipulation instruction with STmL.

Reset signal generation clears this register to 0000H.

Figure 14-13. Format of Serial Channel Stop Register m (STm)

Address: F0124H, F0125H (ST0), F0164H, F0165H (ST1) After reset: 0000H R/W

Symbol	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	STm2	STm1	STm0

STmn	Operation stop trigger of channel n
0	No trigger operation
1	Clears SEmn to 0 and stops the communication operation. (Stops with the values of the control register and shift register, and the statuses of the serial clock I/O pin, serial data output pin, and the FEF, PEF, and OVF error flags retained[Note].)

Note Bits 6 and 5 (TSFmn, BFFmn) of the SSRmn register are cleared.

Caution Be sure to clear bits 15 to 4 to "0".

Remarks 1. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

2. When the STm register is read, 0000H is always read.

(11) Serial output enable register m (SOEm)

SOEm 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 SOmn of the serial output register m (SOm) to be described below, and a value reflected by a communication operation is output from the serial data output pin.

For channel n, whose serial output is stopped, the SOmn 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.

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

The lower 8 bits of SOEm can be set with a 1-bit or 8-bit memory manipulation instruction with SOEmL.

Reset signal generation clears this register to 0000H.

Figure 14-14. 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	0	SOE00

Address: F016AH, F016BH After reset: 0000H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOE1	0	0	0	0	0	0	0	0	0	0	0	0	0	SOE12	0	SOE10

SOEmn	Serial output enable/disable 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, and 1 of SOE0, and bits 15 to 3, and 1 of SOE1 to "0".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2),
mn = 00, 02, 10, 12

(12) Serial output register m (SOm)

SOm is a buffer register for serial output of each channel.

The value of bit n of this register is output from the serial data output pin of channel n.

The value of bit (n + 8) of this register is output from the serial clock output pin of channel n.

SOmn 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.

CKOmn 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 CKOmn can be changed only by a serial communication operation.

To use the P10/$\overline{\text{SCK20}}$/SCL20, P11/SI20/SDA20/RxD2/INTP6, P12/SO20/TxD2/TO02, P13/SO10/TxD1/TO04, P14/SI10/SDA10/RxD1/INTP4, P15/$\overline{\text{SCK10}}$/SCL10/INTP7, P51/TxD3, P80/$\overline{\text{SCK00}}$/INTP11, or P82/SO00/TxD0 pin as a port function pin, set the corresponding CKOmn and SOmn bits to "1".

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

Reset signal generation clears this register to 0F0FH.

Figure 14-15. 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	CKO02	1	CKO00	0	0	0	0	1	SO02	1	SO00

Address: F0168H, F0169H After reset: 0F0FH R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SO1	0	0	0	0	1	1	1	CKO10	0	0	0	0	1	SO12	1	SO10

CKOmn	Serial clock output of channel n
0	Serial clock output value is "0".
1	Serial clock output value is "1".

SOmn	Serial data output of channel n
0	Serial data output value is "0".
1	Serial data output value is "1".

Caution Be sure to set bits 11, 9, 3 and 1 of SO0, and bits 11 to 9, 3, and 1 of SO1 to "1". And be sure to clear bits 15 to 12 and 7 to 4 of SOm to "0".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2),
mn = 00, 02, 10, 12

(13) Serial output level register m (SOLm)

SOLm 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 0000H in the 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 SOLm is prohibited when the register is in operation (when SEmn = 1).

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

The lower 8 bits of SOLm can be set with an 8-bit memory manipulation instruction with SOLmL.

Reset signal generation clears this register to 0000H.

Figure 14-16. Format of Serial Output Level Register m (SOLm)

Address: F0134H, F0135H (SOL0), F0174H, F0175H (SOL1) After reset: 0000H R/W

Symbol	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	SOL m2	0	SOL m0

SOL mn	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, 1 to "0".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

(14) Input switch control register (ISC)

ISC is used to realize a LIN-bus communication operation by UART3 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 (RxD3) 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 (RxD3) pin is selected as a timer input, so that the pulse widths of a sync break field and a sync field can be measured by the timer.

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

Reset signal generation clears this register to 00H.

Figure 14-17. Format of Input Switch Control Register (ISC)

Address: FFF3CH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
ISC	0	0	0	ISC4	ISC3	ISC2	ISC1	ISC0

ISC1	Switching channel 7 input of timer array unit 0
0	Uses the input signal of the TI07 pin as a timer input (normal operation).
1	Input signal of RxD3 pin is used as timer input (wakeup signal detection).

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 RxD3 pin as an external interrupt (to measure the pulse widths of the sync break field and sync field).

Caution Be sure to clear bits 7 to 5 to "0".

Remark Bits 4 to 2 of ISC are not used with SAU1.

(15) Noise filter enable register 0 (NFEN0)

NFEN0 is used to set whether the noise filter can be used for the input signal from the serial data input pin to each channel.

Disable the noise filter of the pin used for CSI or simplified I^2C communication, by clearing the corresponding bit of this register to 0.

Enable the noise filter of the pin used for UART communication, by setting the corresponding bit of this register to 1.

When the noise filter is enabled, CPU/peripheral operating clock (f_{CLK}) is synchronized with 2-clock match detection.

NFEN0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 14-18. Format of Noise Filter Enable Register 0 (NFEN0)

Address: F0060H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN0	0	SNFEN30	0	SNFEN20	0	SNFEN10	0	SNFEN00

SNFEN30	Use of noise filter of RxD3/P50 pin
0	Noise filter OFF
1	Noise filter ON
Set SNFEN30 to 1 to use the RxD3 pin. Clear SNFEN30 to 0 to use the P50 pin.	

SNFEN20	Use of noise filter of RxD2/P11/SI20/SDA20/INTP6 pin
0	Noise filter OFF
1	Noise filter ON
Set SNFEN20 to 1 to use the RxD2 pin. Clear SNFEN20 to 0 to use the P11, SI20, SDA20 or INTP6 pins.	

SNFEN10	Use of noise filter of RxD1/P14/SI10/SDA10/INTP4 pin
0	Noise filter OFF
1	Noise filter ON
Set SNFEN10 to 1 to use the RxD1 pin. Clear SNFEN10 to 0 to use the P14, SI10, SDA10 or INTP4 pins.	

SNFEN00	Use of noise filter of RxD0/P81/SI00/INTP9 pin
0	Noise filter OFF
1	Noise filter ON
Set SNFEN00 to 1 to use the RxD0 pin. Clear SNFEN00 to 0 to use the P81, SI00 or INTP9.	

Caution Be sure to clear bits 7, 5, 3, and 1 to "0".

(16) Port input mode register 1 (PIM1)

This register sets the input buffer of P10, P11, P14, and P15 in 1-bit units.

PIM1 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 Port Input Mode Register 1 (PIM1)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PIM1	0	0	PIM15	PIM14	0	0	PIM11	PIM10	F0041H	00H	R/W

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

(17) Port output mode registers 1, 8 (POM1, POM8)

These registers set the output mode of P10 to P15, P80, and P82 in 1-bit units.

POM1 and POM8 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 14-20. Format of Port Output Mode Registers 1 and 8 (POM1, POM8)

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
POM8	0	0	0	0	0	POM82	0	POM80	F0058H	00H	R/W

POMmn	Pmn pin output mode selection (m = 1, 8; n = 0 to 5)
0	Normal output mode
1	N-ch open-drain output (V_{DD} tolerance) mode

(18) Port mode registers 1, 5, 8 (PM1, PM5, PM8)

These registers set input/output of ports 1, 5 and 8 in 1-bit units.

When using the P10/SCK20/SCL20, P11/SI20/SDA20/RxD2/INTP6, P12/SO20/TxD2/TO02, P13/SO10/TxD1/TO04, P14/SI10/SDA10/RxD1/INTP4, P15/SCK10/SCL10/INTP7, P51/TxD3, P80/SCK00/INTP11, and P82/SO00/TxD0 pins for serial data output or serial clock output, clear the PM10, PM11, PM12, PM13, PM14, PM15, PM51, PM80, and PM82 bits to 0, and set the output latches of P10, P11, P12, P13, P14, P15, P51, P80, and P82 to 1.

When using the P10/SCK20/SCL20, P11/SI20/SDA20/RxD2/INTP6, P14/SI10/SDA10/RxD1/INTP4, P15/SCK10/SCL10/INTP7, P50/RxD3, P80/SCK00/INTP11, and P81/SI00/RxD0/INTP9 pins for serial data input or serial clock input, set the PM10, PM11, PM14, PM15, PM50, PM80, and PM81 bits to 1. At this time, the output latches of P10, P11, P14, P15, P50, P80, and P81 may be 0 or 1.

PM1, PM5, and PM8 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets these registers to FFH.

Figure 14-21. Format of Port Mode Registers 1, 5, and 8 (PM1, PM5, PM8)

Symbol	7	6	5	4	3	2	1	0	Address	After reset	R/W
PM1	1	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFF21H	FFH	R/W
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FFF25H	FFH	R/W
PM8	1	1	1	1	1	PM82	PM81	PM80	FFF28H	FFH	R/W

PMmn	Pmn pin I/O mode selection (m = 1, 5, 8; n = 0 to 7)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

14.4 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 P10/$\overline{\text{SCK20}}$/SCL20, P11/SI20/SDA20/RxD2/INTP6, P12/SO20/TxD2/TO02, P13/SO10/TxD1/TO04, P14/SI10/SDA10/RxD1/INTP4, P15/$\overline{\text{SCK10}}$/SCL10/INTP7, P50/RxD3, P51/TxD3, P80/$\overline{\text{SCK00}}$/INTP11, P81/SI00/RxD0/INTP9, and P82/SO00/TxD0 and pins can be used as ordinary port pins in this mode.

14.4.1 Stopping the operation by units

The stopping of the operation by units is set by using peripheral enable register 0 (PER0).

PER0 is used to enable or disable use of each peripheral hardware macro. Clock supply to a hardware macro that is not used is stopped in order to reduce the power consumption and noise.

To stop the operation of serial array unit 0, set bit 2 (SAU0EN) to 0.

To stop the operation of serial array unit 1, set bit 3 (SAU1EN) to 0.

Figure 14-22. 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	DACEN	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN
	×	×	×	×	0/1	0/1	×	×

Control of SAUm input clock
0: Stops supply of input clock
1: Supplies input clock

Caution 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 input switch control register (ISC), noise filter enable register 0 (NFEN0), port input mode register 1 (PIM1), port output mode registers 1, 8 (POM1, POM8), port mode registers 1, 5, 8 (PM1, PM5, PM8), and port registers 1, 5, 8 (P1, P5, P8)).

Remark m: Unit number (m = 0, 1)
×: 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

14.4.2 Stopping the operation by channels

The stopping of the operation by channels is set using each of the following registers.

Figure 14-23. Each Register Setting When Stopping the Operation by Channels (1/2)

(a) Serial Channel Enable Status Register m (SEm) … This register indicates whether data transmission/reception operation of each channel is enabled or stopped.

SEm

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	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 CKOmn of the SOm register can be set by software.

(b) Serial channel stop register m (STm) … This register is a trigger register that is used to enable stopping communication/count by each channel.

STm

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	STm3 0/1	STm2 0/1	STm1 0/1	STm0 0/1

1: Clears SEmn to 0 and stops the communication operation

* Because STmn is a trigger bit, it is cleared immediately when SEmn = 0.

(c) Serial output enable register m (SOEm) … This register is a register that is used to enable or stop output of the serial communication operation of each channel.

SOE0

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	SOE02 0/1	0	SOE00 0/1

0: Stops output by serial communication operation

* For channel n, whose serial output is stopped, the SO0n value of the SO0 register can be set by software.

SOE1

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	SOE12 0/1	0	SOE10 0/1

0: Stops output by serial communication operation

* For channel n, whose serial output is stopped, the SO1n value of the SO1 register can be set by software.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)
□ : Setting disabled (fixed by hardware), 0/1: Set to 0 or 1 depending on the usage of the user

Figure 14-23. Each Register Setting When Stopping the Operation by Channels (2/2)

(d) Serial output register m (SOm) …This register is a buffer register for serial output of each channel.

SO0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	CKO02 0/1	1	CKO00 0/1	0	0	0	0	1	SO02 0/1	1	SO00 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 CKO0n and SO0n bits to “1”.

SO1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	1	1	CKO10 0/1	0	0	0	0	1	SO12 0/1	1	SO10 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 CKO10 and SO1n bits to “1”.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

□ : Setting disabled (fixed by hardware), 0/1: Set to 0 or 1 depending on the usage of the user

14.5 Operation of 3-Wire Serial I/O (CSI00, CSI10, CSI20) Communication

This is a clocked communication function that uses three lines: serial clock ($\overline{SCK}$) 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
- Level setting of transmit/receive data

[Clock control]

- Master/slave selection
- Phase control of I/O clock
- Setting of transfer period by prescaler and internal counter of each channel

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt

[Error detection flag]

- Overrun error

The channels supporting 3-wire serial I/O (CSI00, CSI10, CSI20) are channels 0, 2 of SAU0 and channel 0 of SAU1.

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	–
	1	–		–
	2	CSI10	UART1	IIC10
	3	–		–
1	0	CSI20	UART2	IIC20
	1	–		–
	2	–	UART3 (supporting LIN-bus)	–
	3	–		–

3-wire serial I/O (CSI00, CSI10, CSI20) performs the following six types of communication operations.

- Master transmission (See **14.5.1**.)
- Master reception (See **14.5.2**.)
- Master transmission/reception (See **14.5.3**.)
- Slave transmission (See **14.5.4**.)
- Slave reception (See **14.5.5**.)
- Slave transmission/reception (See **14.5.6**.)

14.5.1 Master transmission

Master transmission is that the 78K0R/KE3-A microcontrollers output a transfer clock and transmit data to another device.

3-Wire Serial I/O	CSI00	CSI10	CSI20
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1
Pins used	$\overline{SCK00}$, SO00	$\overline{SCK10}$, SO10	$\overline{SCK20}$, SO20
Interrupt	INTCSI00	INTCSI10	INTCSI20
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.		
Error detection flag	None		
Transfer data length	7 or 8 bits		
Transfer rate	Max. $f_{CLK}/4$ [MHz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [MHz] Note f_{CLK}: System clock frequency		
Data phase	Selectable by DAPmn bit • 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 CKPmn bit • CKPmn = 0: Forward • CKPmn = 1: Reverse		
Data direction	MSB or LSB first		

Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

(1) Register setting

Figure 14-24. Example of Contents of Registers for Master Transmission of 3-Wire Serial I/O (CSI00, CSI10, CSI20)

(a) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 0/1	1	CKOm0 0/1	0	0	0	0	1	SOm2 0/1	1	SOm0 0/1

Communication starts when these bits are 1 if the data phase is forward (CKPmn = 0). If the phase is reversed (CKPmn = 1), communication starts when these bits are 0.

(b) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 0/1	0	SOEm0 0/1

(c) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 0/1	SSm0 0/1

(d) 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	SISm0 0	1	0	0	MDmn2 0	MDmn1 0	MDmn0 0/1

Operation mode of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) 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	DLSmn2 1	DLSmn1 1	DLSmn0 0/1

(f) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

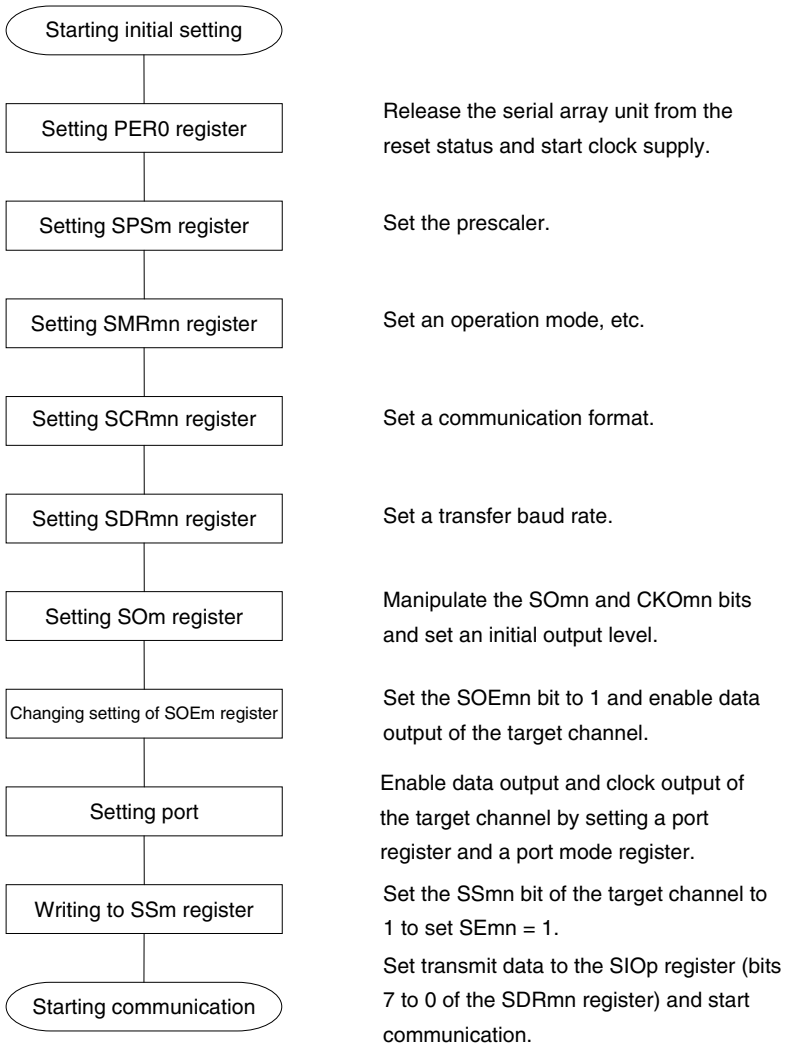
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	Transmit data setting							

SIOp

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)
□: Setting is fixed in the CSI master transmission mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

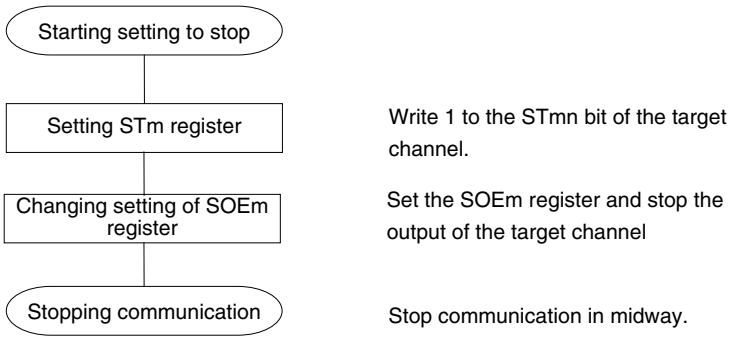
Figure 14-25. Initial Setting Procedure for Master Transmission



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

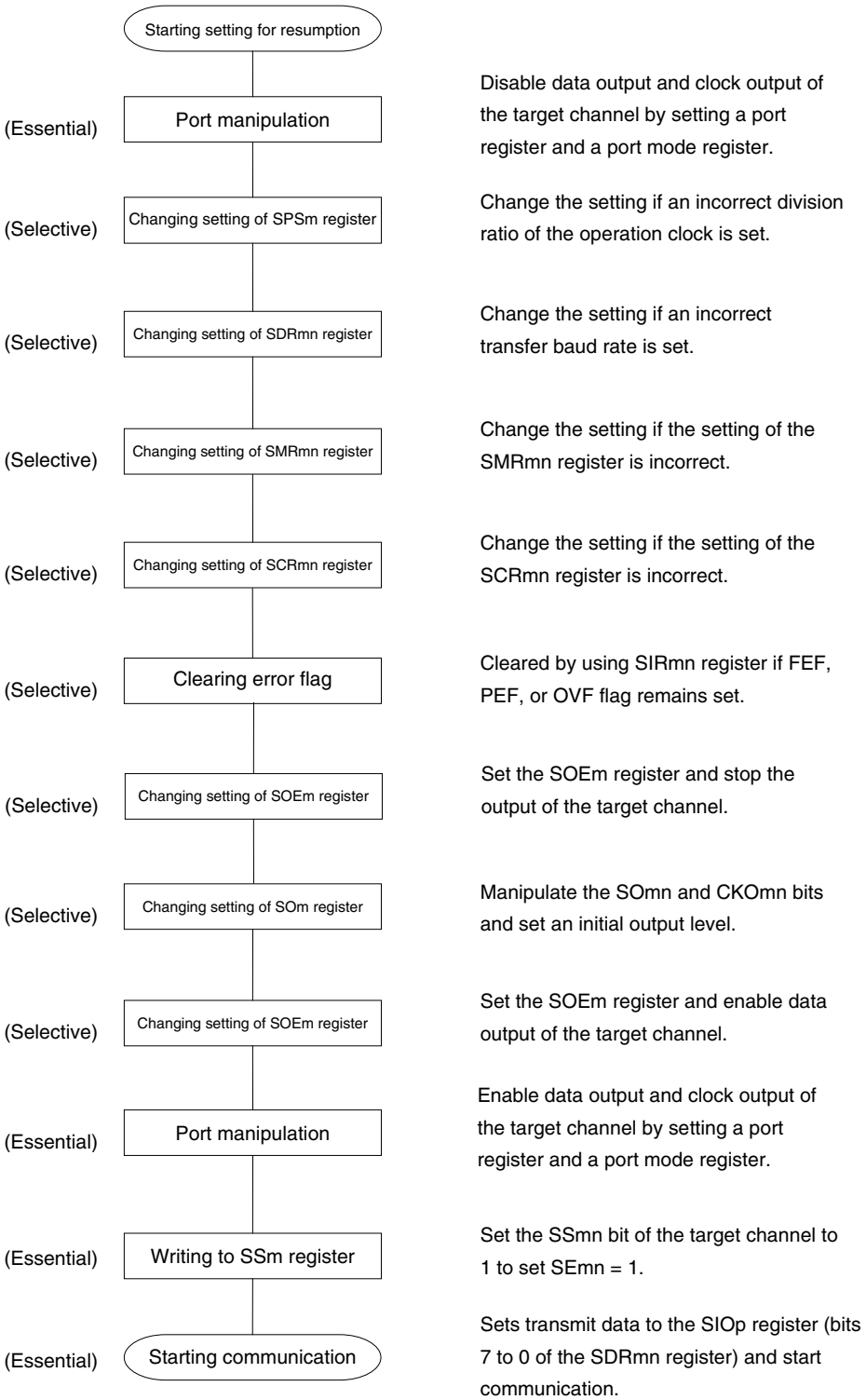
Figure 14-26. Procedure for Stopping Master Transmission



Remarks 1. Even after communication is stopped, the pin level is retained. To resume the operation, reset the SOm register (see **Figure 14-27 Procedure for Resuming Master Transmission**).

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

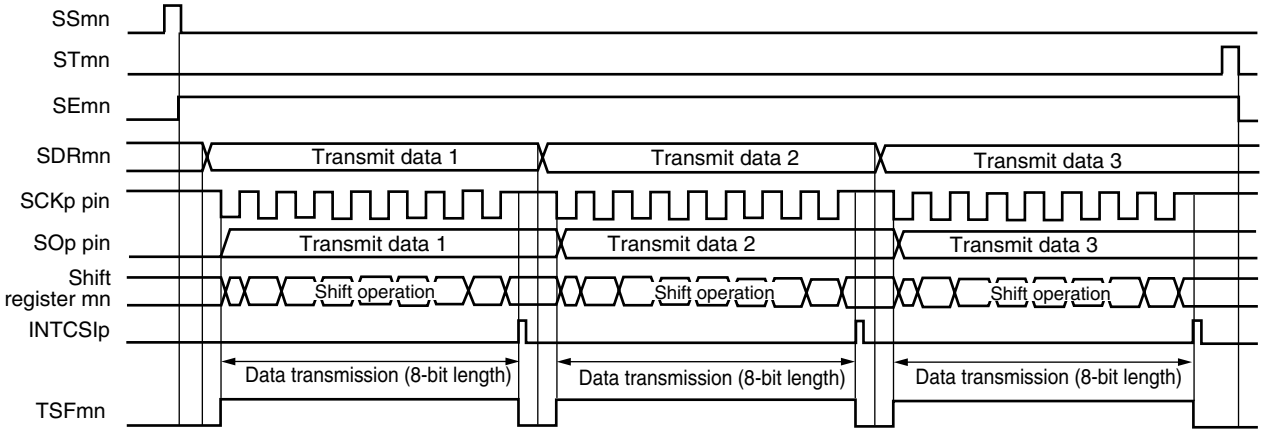
Figure 14-27. Procedure for Resuming Master Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

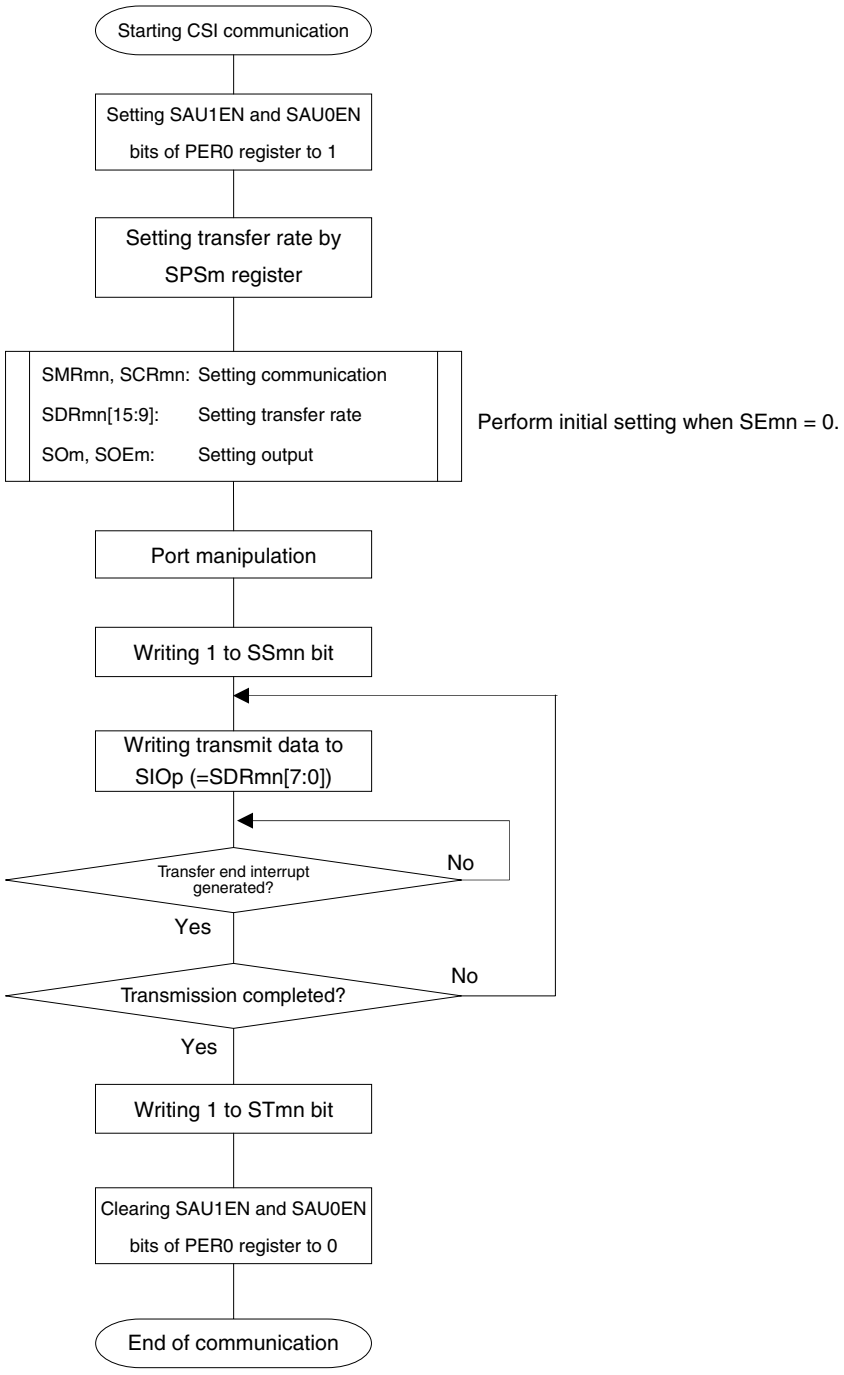
(3) Processing flow (in single-transmission mode)

Figure 14-28. Timing Chart of Master Transmission (in Single-Transmission Mode)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

Figure 14-29. Flowchart of Master Transmission (in Single-Transmission Mode)

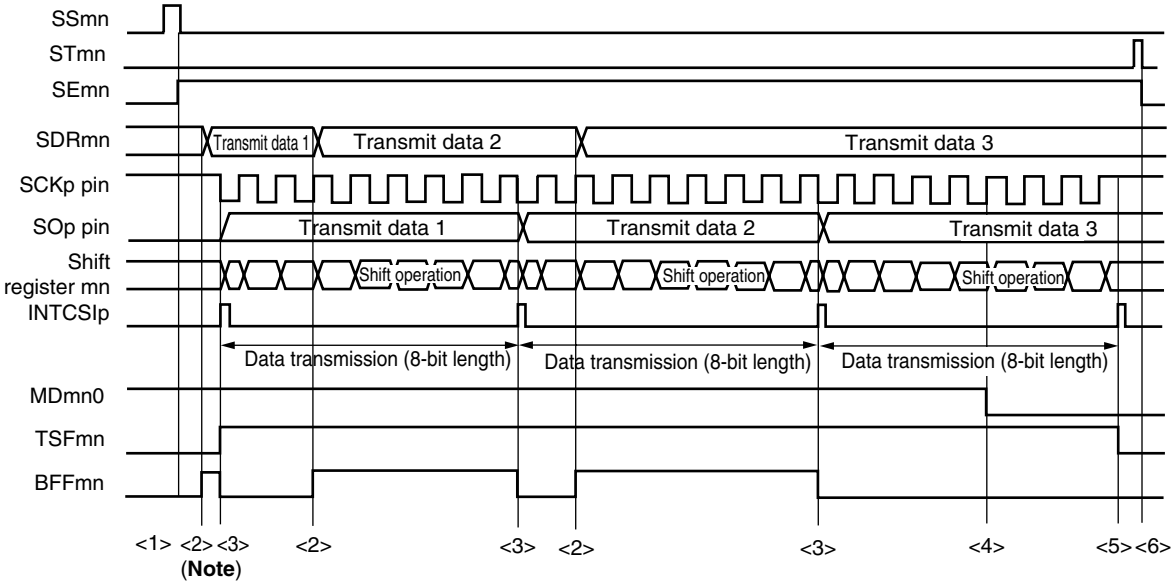


Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

(4) Processing flow (in continuous transmission mode)

Figure 14-30. Timing Chart of Master Transmission (in Continuous Transmission Mode)

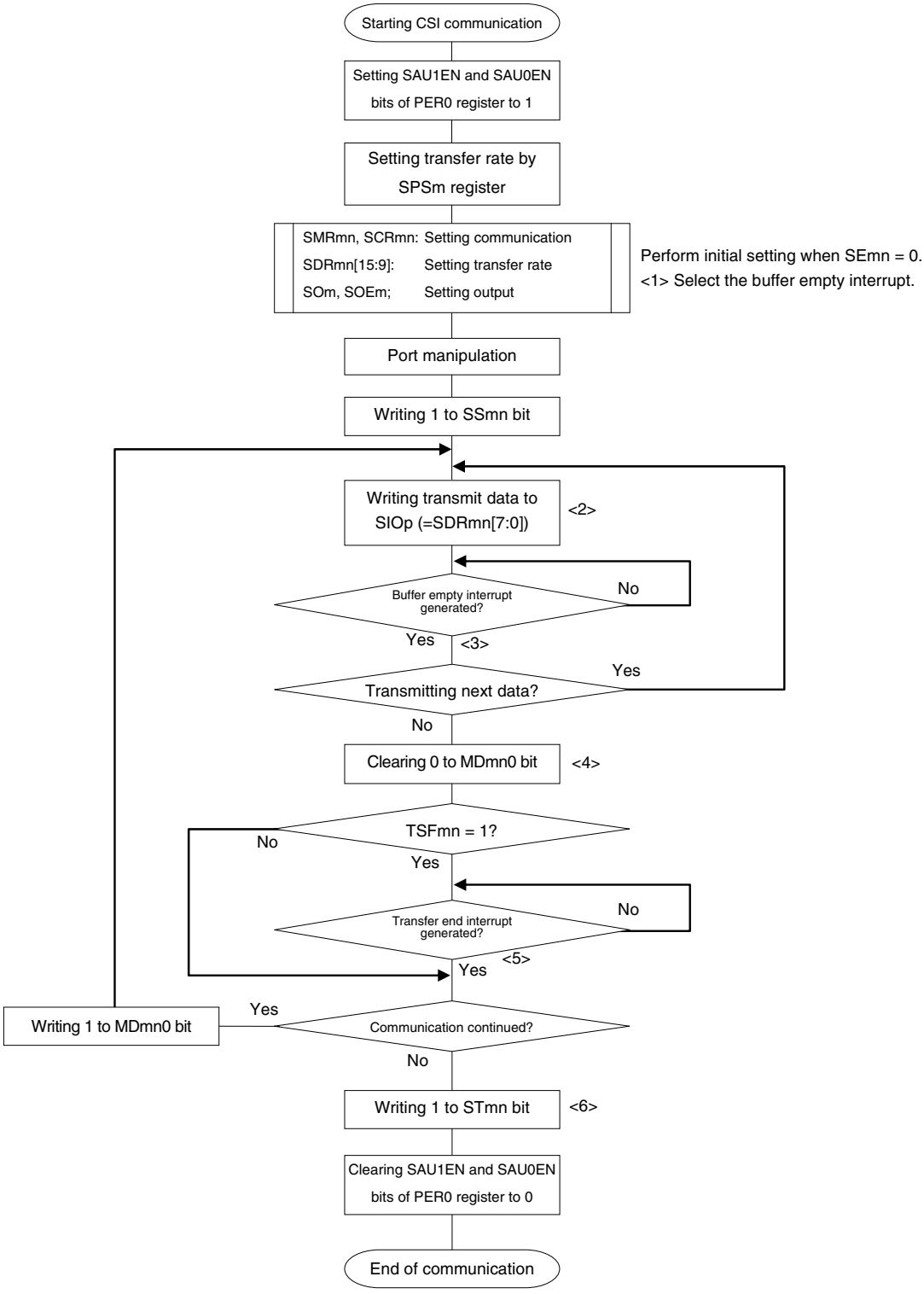


Note When transmit data is written to the SDRmn register while BFFmn = 1, the transmit data is overwritten.

Caution The MDmn0 bit can be rewritten even during operation.
However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

Figure 14-31. Flowchart of Master Transmission (in Continuous Transmission Mode)



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remarks 1. <1> to <6> in the figure correspond to <1> to <6> in **Figure 14-30 Timing Chart of Master Transmission (in Continuous Transmission Mode)**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

14.5.2 Master reception

Master reception is that the 78K0R/KE3-A microcontrollers output a transfer clock and receive data from other device.

3-Wire Serial I/O	CSI00	CSI10	CSI20
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1
Pins used	$\overline{SCK00}$, SI00	$\overline{SCK10}$, SI10	$\overline{SCK20}$, SI20
Interrupt	INTCSI00	INTCSI10	INTCSI20
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)		
Error detection flag	Overrun error detection flag (OVFmn) only		
Transfer data length	7 or 8 bits		
Transfer rate	Max. $f_{CLK}/4$ [MHz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [MHz] Note f_{CLK}: System clock frequency		
Data phase	Selectable by DAPmn bit • 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 CKPmn bit • CKPmn = 0: Forward • CKPmn = 1: Reverse		
Data direction	MSB or LSB first		

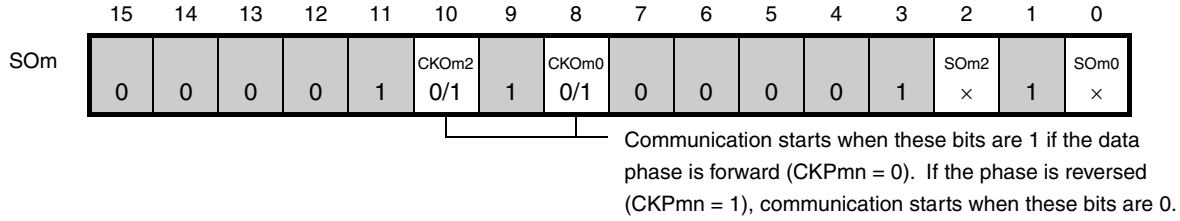
Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

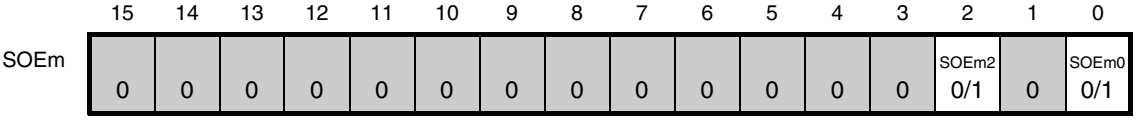
(1) Register setting

Figure 14-32. Example of Contents of Registers for Master Reception of 3-Wire Serial I/O (CSI00, CSI10, CSI20)

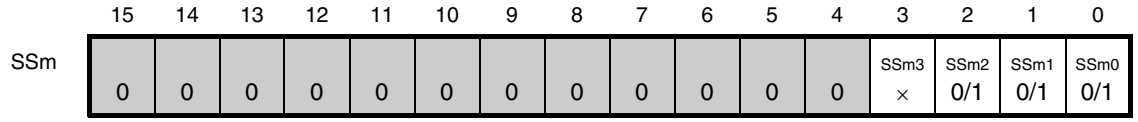
(a) Serial output register m (SOm) … Sets only the bits of the target channel.



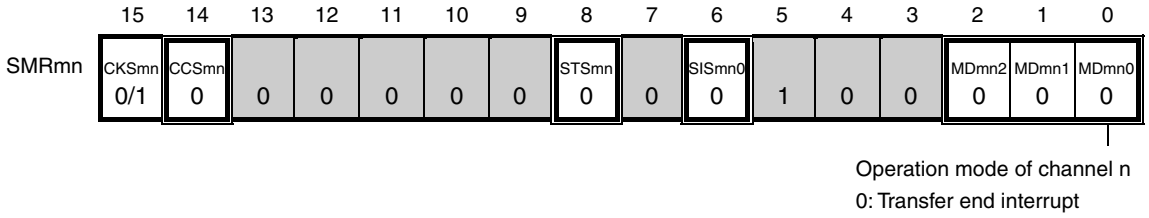
(b) Serial output enable register m (SOEm) … Clears only the bits of the target channel to 0.



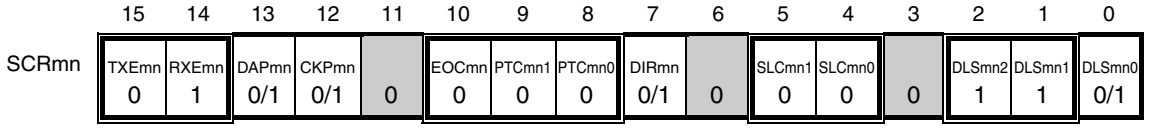
(c) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.



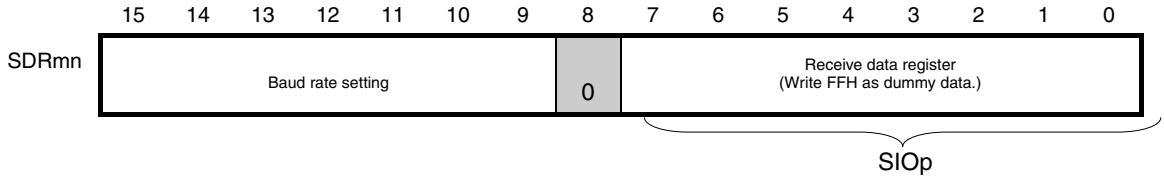
(d) Serial mode register mn (SMRmn)



(e) Serial communication operation setting register mn (SCRmn)



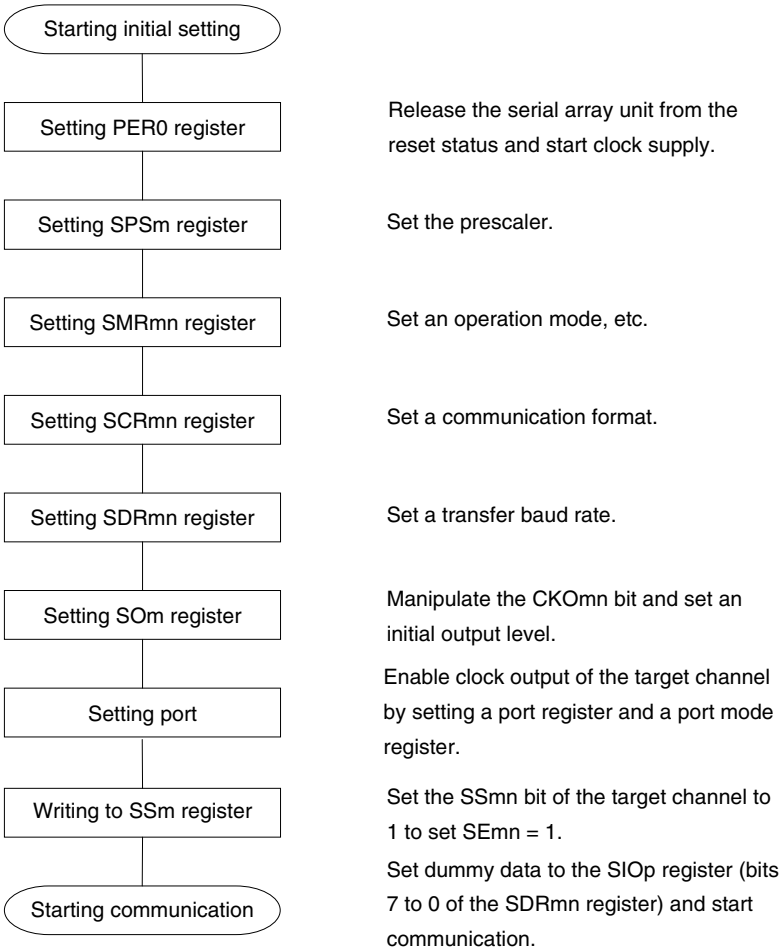
(f) Serial data register mn (SDRmn) (lower 8 bits: SIOp)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)
□: Setting is fixed in the CSI master reception mode, ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

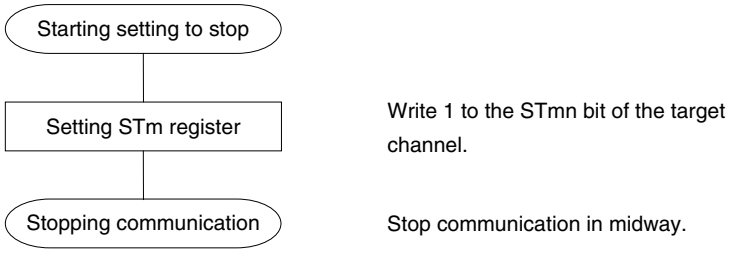
Figure 14-33. Initial Setting Procedure for Master Reception



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

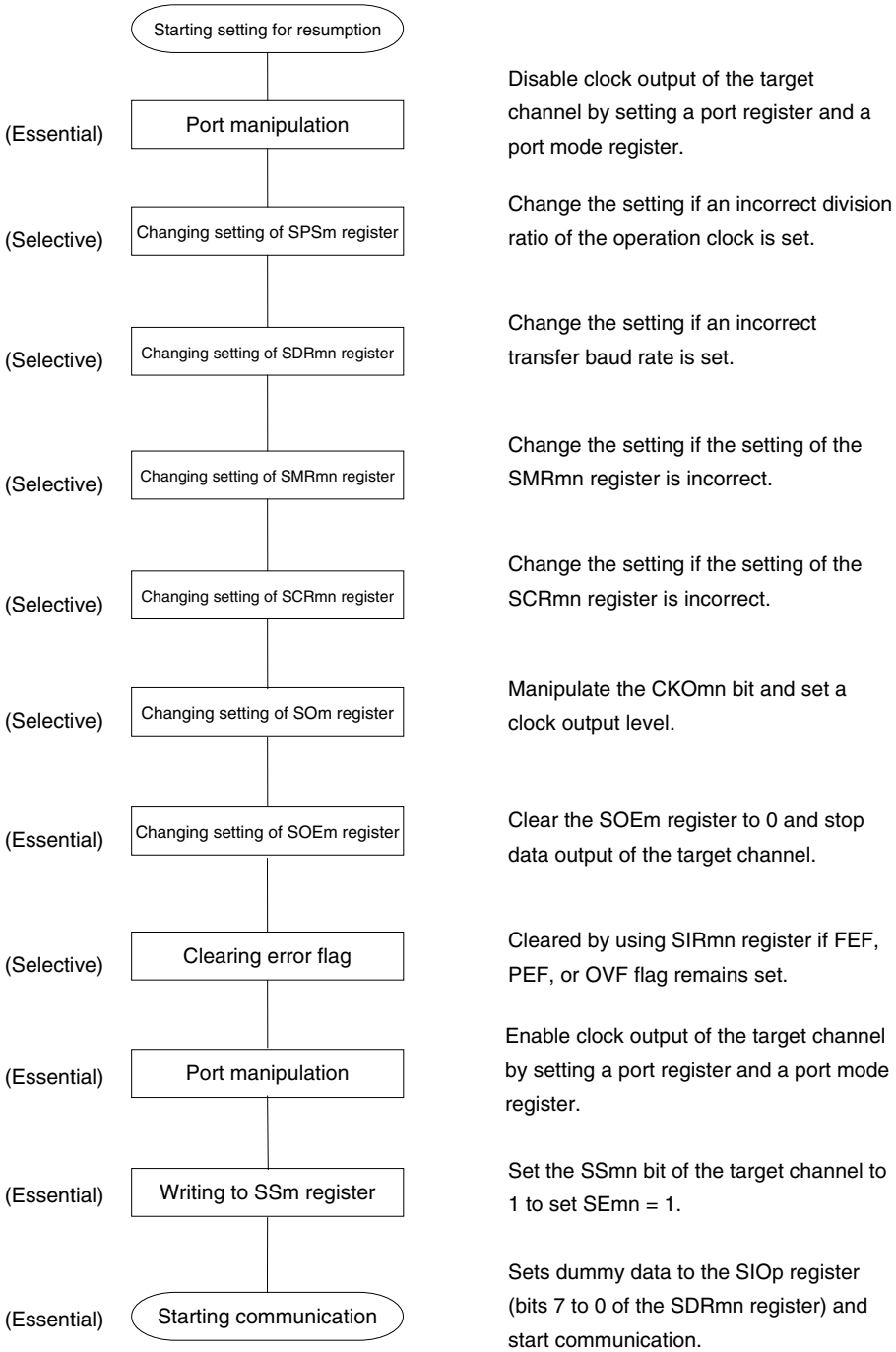
Figure 14-34. Procedure for Stopping Master Reception



Remarks 1. Even after communication is stopped, the pin level is retained. To resume the operation, reset the SOm register (see **Figure 14-35 Procedure for Resuming Master Reception**).

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

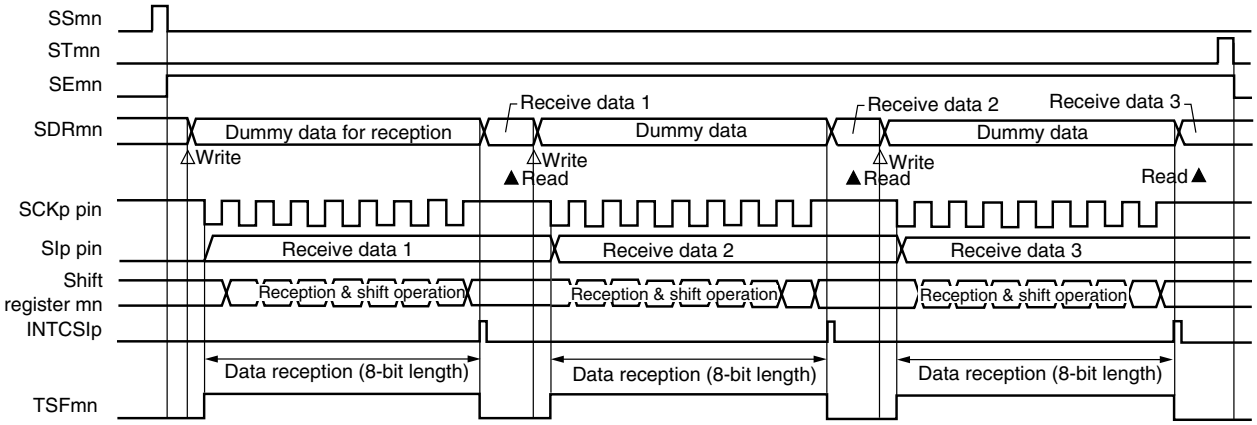
Figure 14-35. Procedure for Resuming Master Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

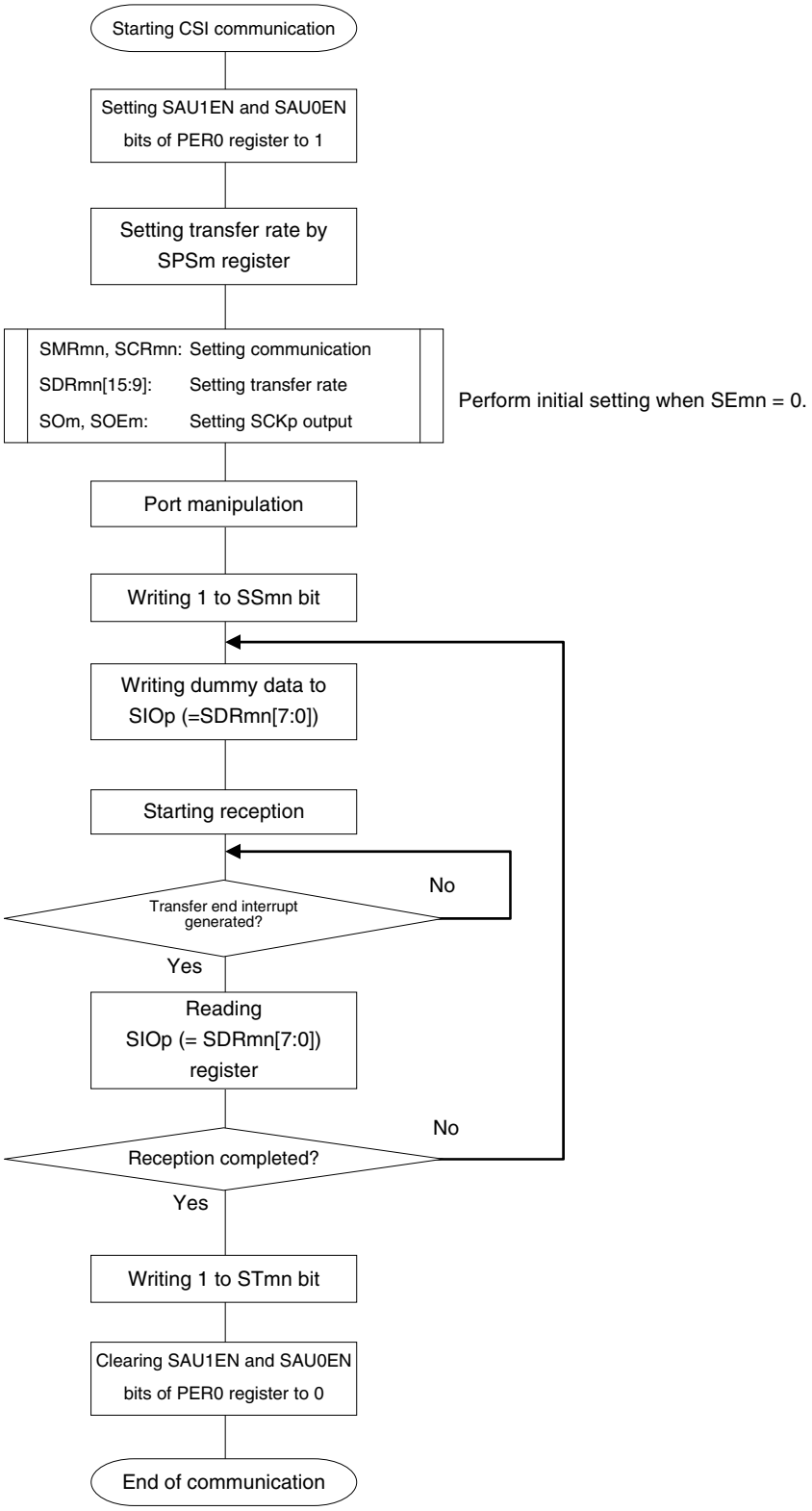
(3) Processing flow (in single-reception mode)

Figure 14-36. Timing Chart of Master Reception (in Single-Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

Figure 14-37. Flowchart of Master Reception (in Single-Reception Mode)



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

14.5.3 Master transmission/reception

Master transmission/reception is that the 78K0R/KE3-A microcontrollers output a transfer clock and transmit/receive data to/from other device.

3-Wire Serial I/O	CSI00	CSI10	CSI20
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1
Pins used	$\overline{SCK00}$, SI00, SO00	$\overline{SCK10}$, SI10, SO10	$\overline{SCK20}$, SI20, SO20
Interrupt	INTCSI00	INTCSI10	INTCSI20
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.		
Error detection flag	Overrun error detection flag (OVFmn) only		
Transfer data length	7 or 8 bits		
Transfer rate	Max. $f_{CLK}/4$ [MHz], Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [MHz] Note f_{CLK}: System clock frequency		
Data phase	Selectable by DAPmn bit • 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 CKPmn bit • CKPmn = 0: Forward • CKPmn = 1: Reverse		
Data direction	MSB or LSB first		

Note Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

(1) Register setting

Figure 14-38. Example of Contents of Registers for Master Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI10, CSI20)

(a) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 0/1	1	CKOm0 0/1	0	0	0	0	1	SOm2 0/1	1	SOm0 0/1

Communication starts when these bits are 1 if the data phase is forward (CKPmn = 0). If the phase is reversed (CKPmn = 1), communication starts when these bits are 0.

(b) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 0/1	0	SOEm0 0/1

(c) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 0/1	SSm0 0/1

(d) 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 mode of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) 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	DLSmn2 1	DLSmn1 1	DLSmn0 0/1

(f) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

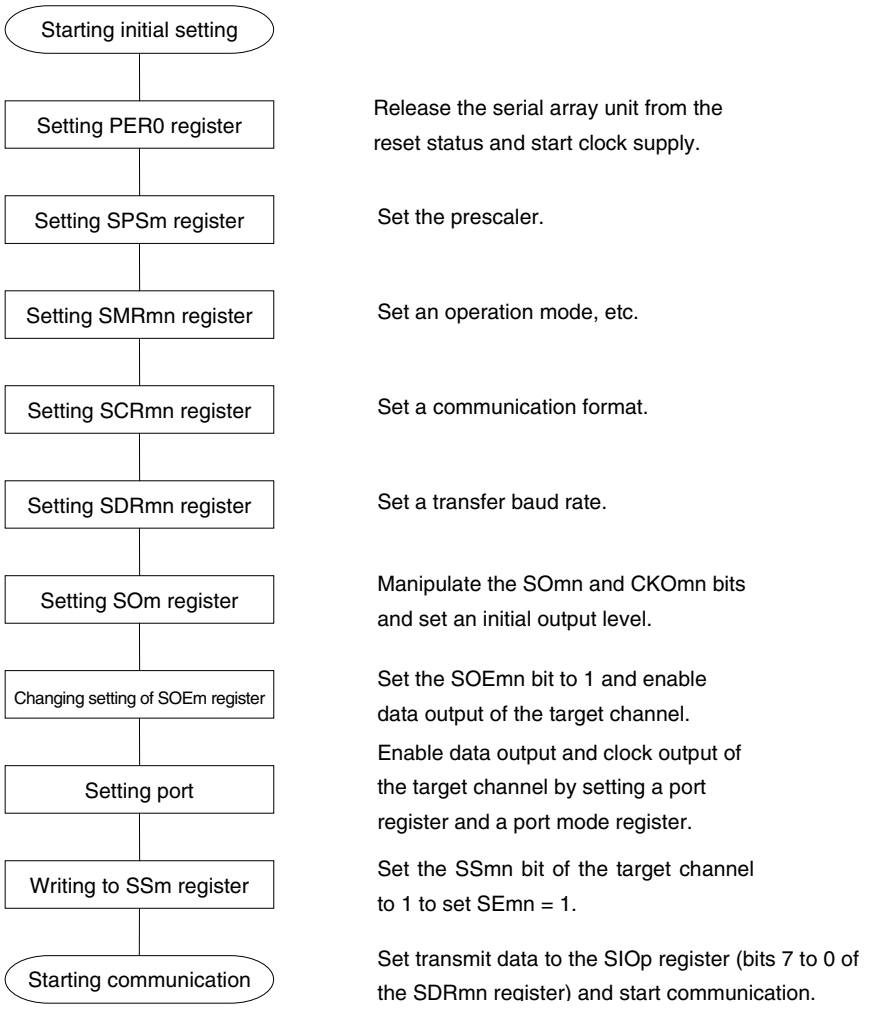
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	Transmit data setting/receive data register							

SIOp

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)
□: Setting is fixed in the 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

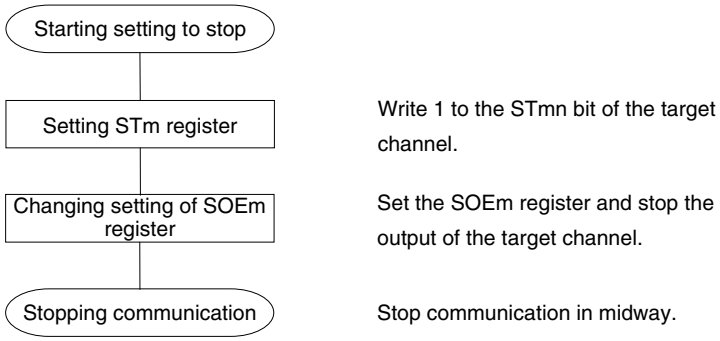
Figure 14-39. Initial Setting Procedure for Master Transmission/Reception



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

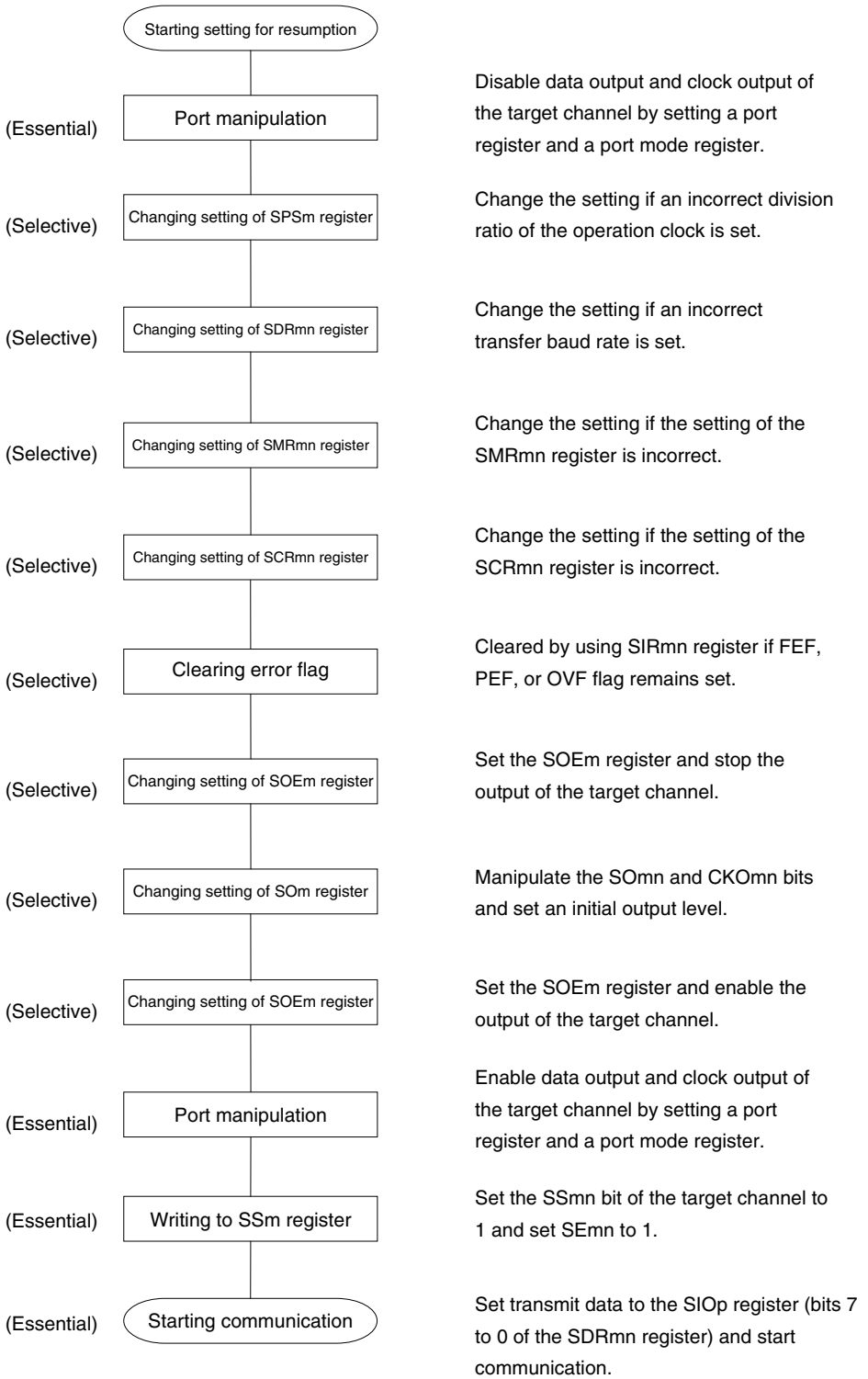
Figure 14-40. Procedure for Stopping Master Transmission/Reception



Remarks 1. Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 14-41 Procedure for Resuming Master Transmission/Reception**).

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

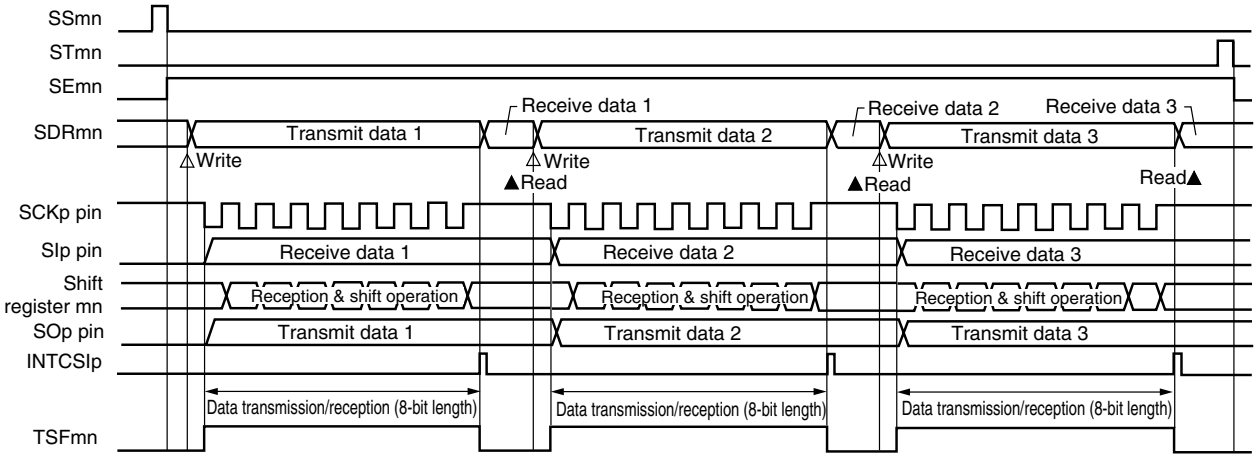
Figure 14-41. Procedure for Resuming Master Transmission/Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

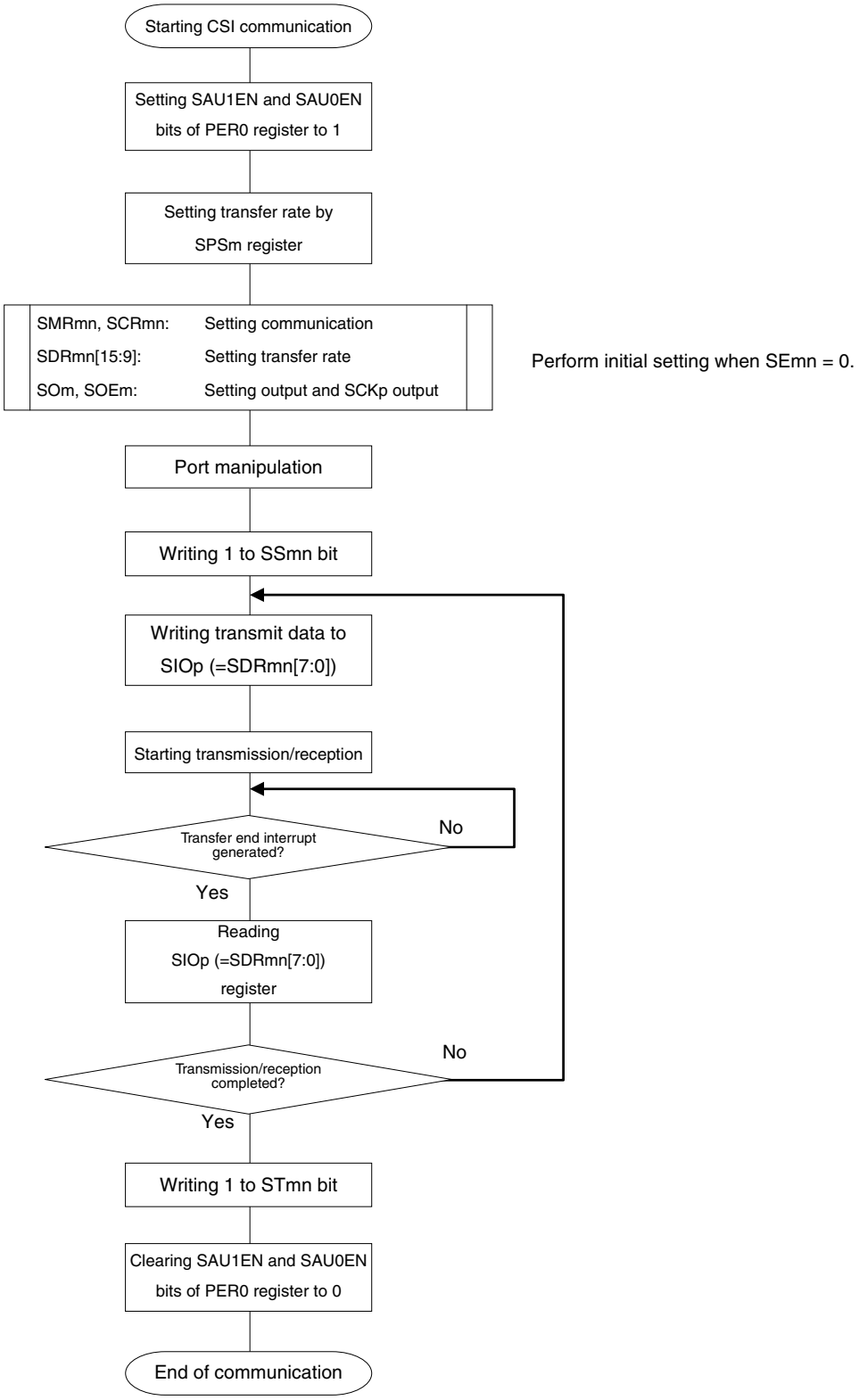
(3) Processing flow (in single-transmission/reception mode)

Figure 14-42. Timing Chart of Master Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

Figure 14-43. Flowchart of Master Transmission/Reception (in Single- Transmission/Reception Mode)

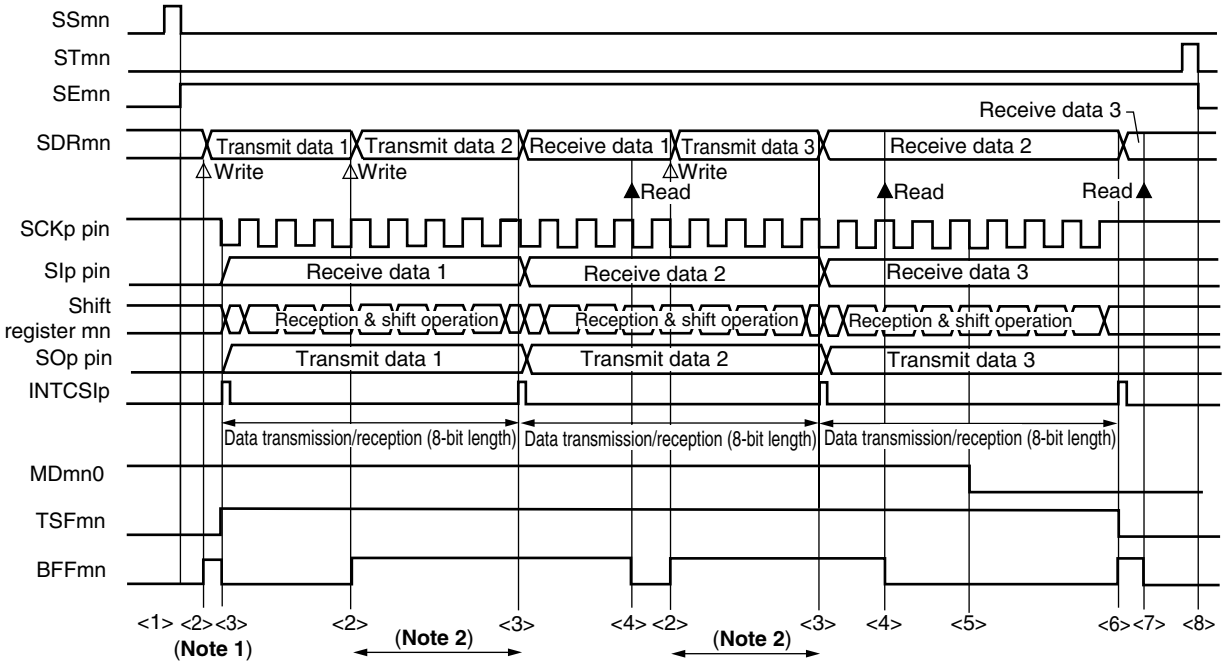


Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

(4) Processing flow (in continuous transmission/reception mode)

Figure 14-44. Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Notes 1. When transmit data is written to the SDRmn register while BFFmn = 1, the transmit data is overwritten.

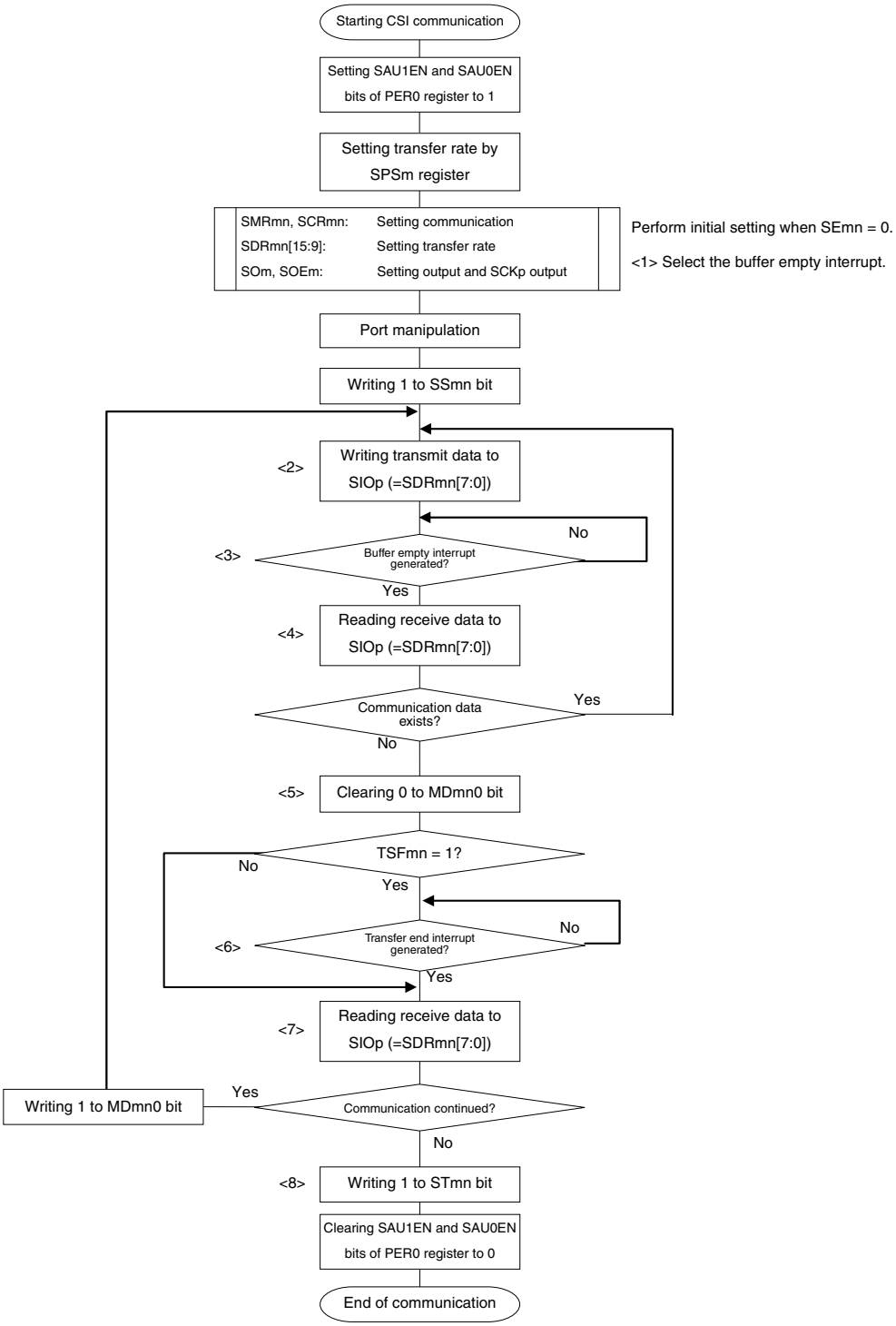
2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution The MDmn0 bit can be rewritten even during operation.
However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 14-45 Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

Figure 14-45. Flowchart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 14-44 Timing Chart of Master Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

14.5.4 Slave transmission

Slave transmission is that the 78K0R/KE3-A microcontrollers transmit data to another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI00	CSI10	CSI20
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1
Pins used	$\overline{SCK00}$, SO00	$\overline{SCK10}$, SO10	$\overline{SCK20}$, SO20
Interrupt	INTCSI00	INTCSI10	INTCSI20
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.		
Error detection flag	Overrun error detection flag (OVFmn) only		
Transfer data length	7 or 8 bits		
Transfer rate	Max. f_{MCK}/6 [MHz]Notes 1, 2		
Data phase	Selectable by DAPmn bit • 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 CKPmn bit • CKPmn = 0: Forward • CKPmn = 1: Reverse		
Data direction	MSB or LSB first		

Notes 1. Because the external serial clock input to pins $\overline{SCK00}$, $\overline{SCK10}$, and $\overline{SCK20}$ is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [MHz].

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{MCK}: Operation clock (MCK) frequency of target channel

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

(1) Register setting

Figure 14-46. Example of Contents of Registers for Slave Transmission of 3-Wire Serial I/O (CSI00, CSI10, CSI20)

(a) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 ×	1	CKOm0 ×	0	0	0	0	1	SOm2 0/1	1	SOm0 0/1

(b) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 0/1	0	SOEm0 0/1

(c) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 0/1	SSm0 0/1

(d) 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 mode of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) 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	DLSmn2 1	DLSmn1 1	DLSmn0 0/1

(f) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

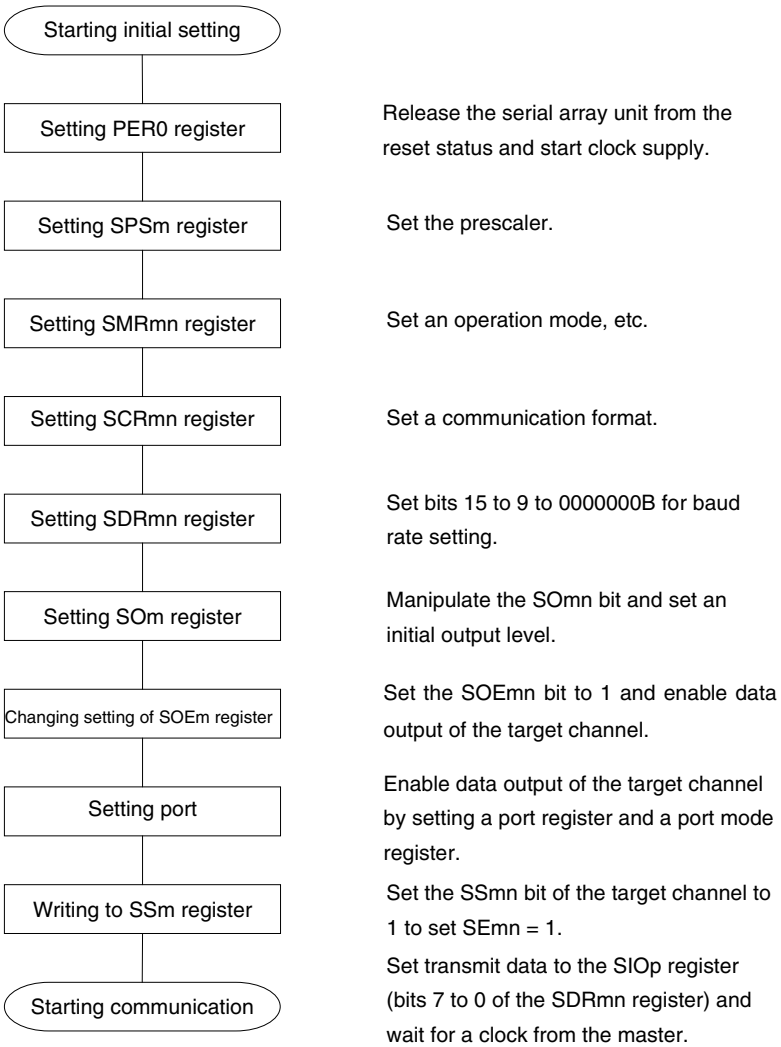
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	Transmit data setting							

SIOp

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)
□: Setting is fixed in the CSI slave transmission mode, ▒ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

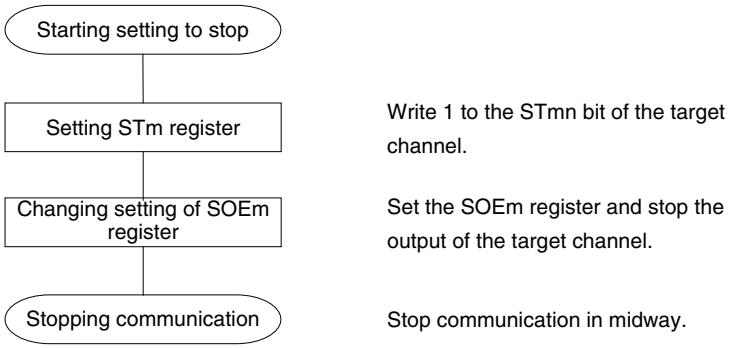
Figure 14-47. Initial Setting Procedure for Slave Transmission



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

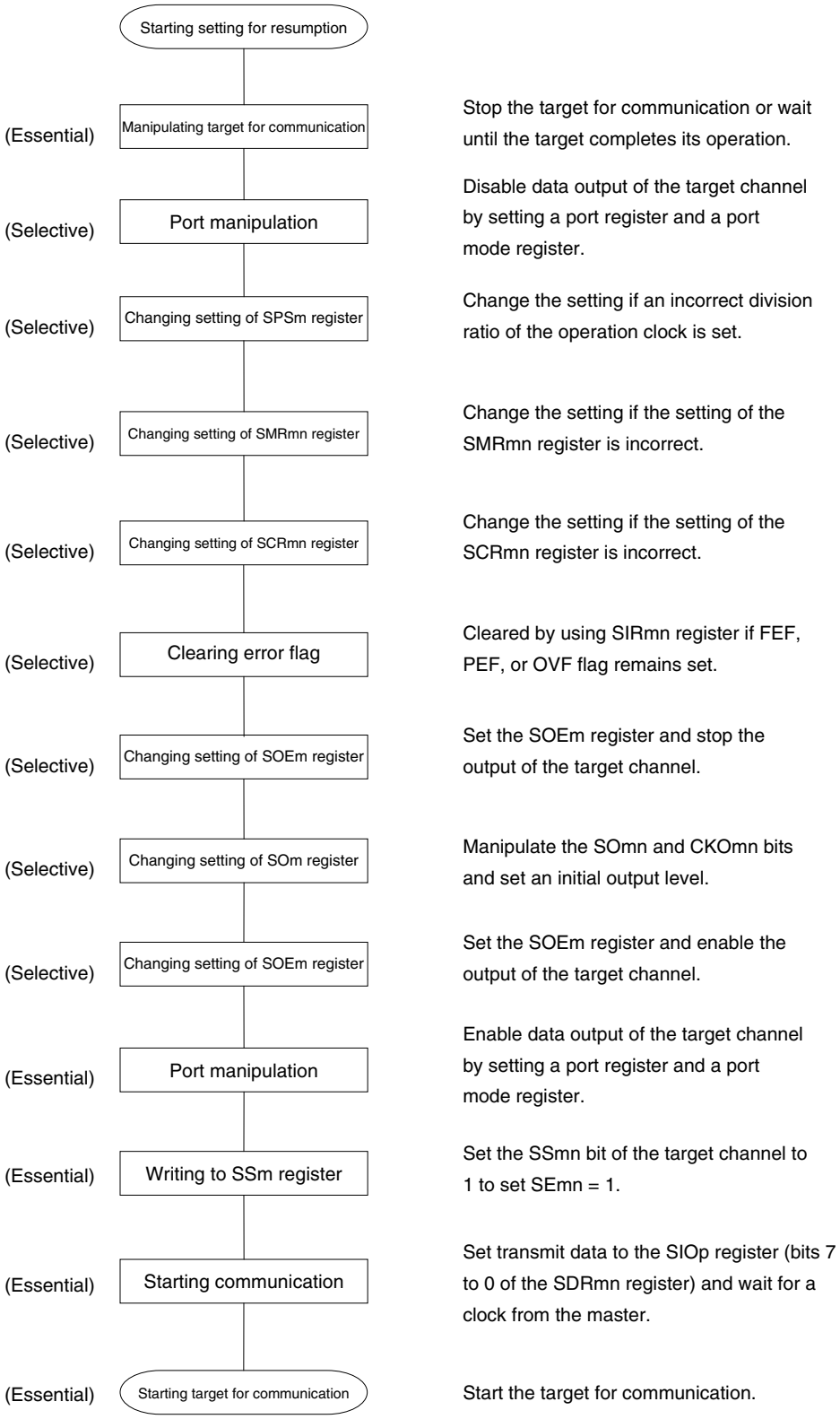
Figure 14-48. Procedure for Stopping Slave Transmission



Remarks 1. Even after communication is stopped, the pin level is retained. To resume the operation, reset the SOm register (see **Figure 14-49 Procedure for Resuming Slave Transmission**).

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

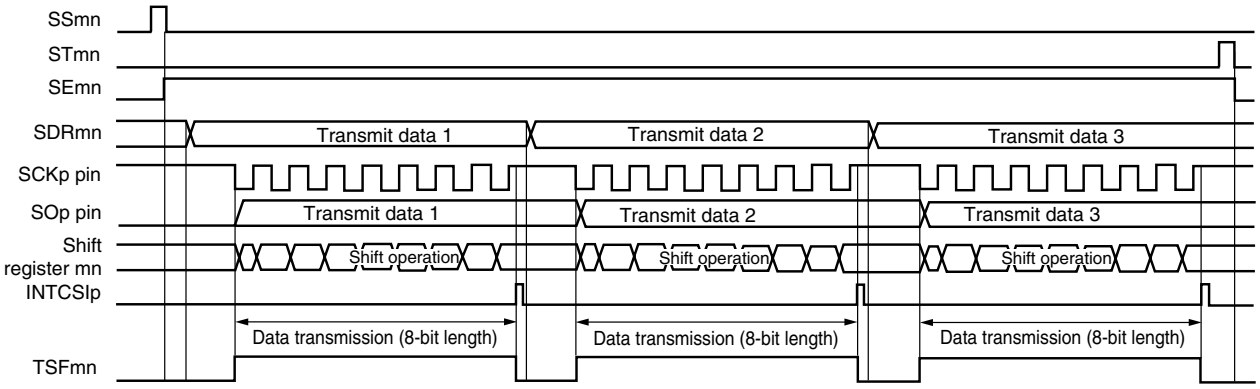
Figure 14-49. Procedure for Resuming Slave Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

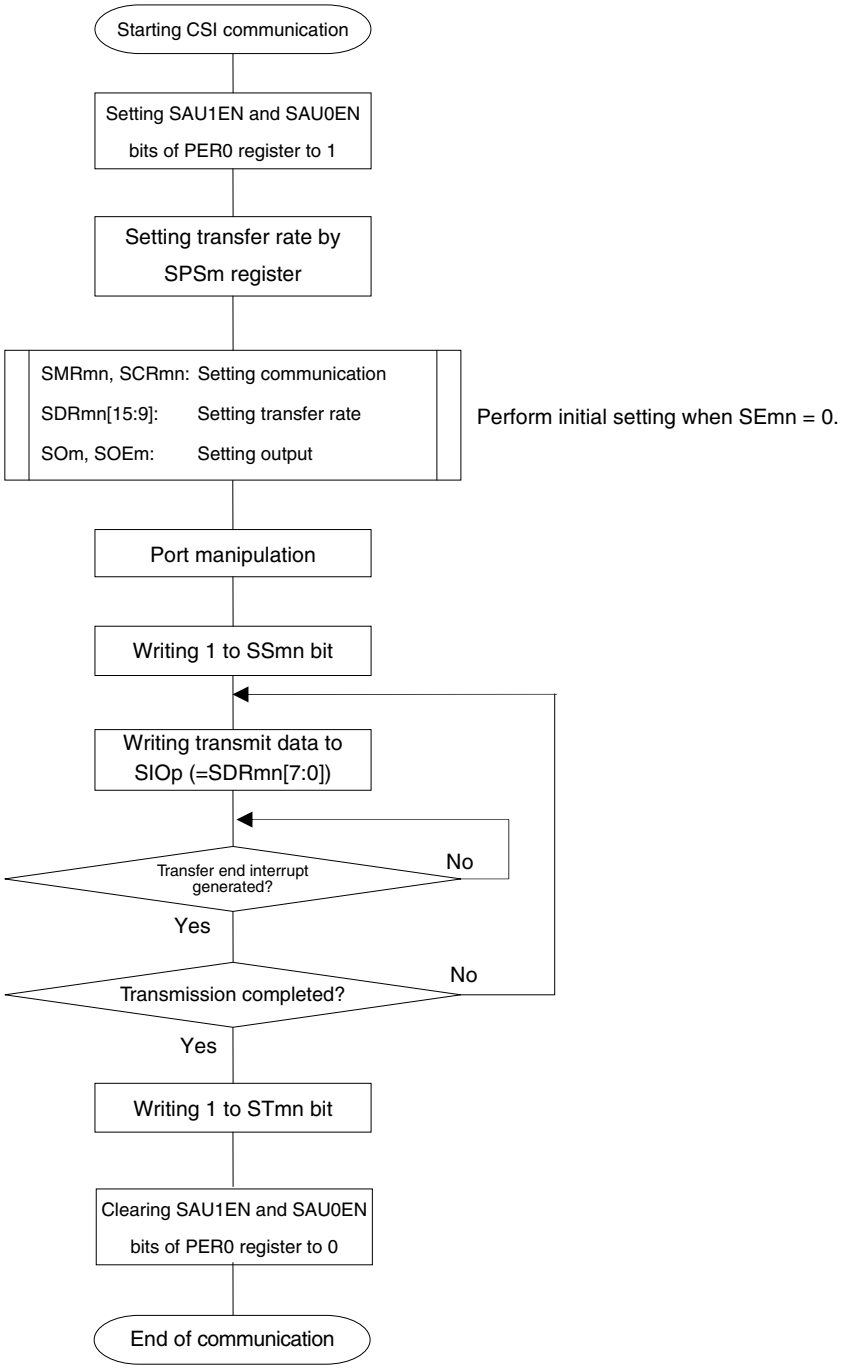
(3) Processing flow (in single-transmission mode)

Figure 14-50. Timing Chart of Slave Transmission (in Single-Transmission Mode)
(Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

Figure 14-51. Flowchart of Slave Transmission (in Single-Transmission Mode)

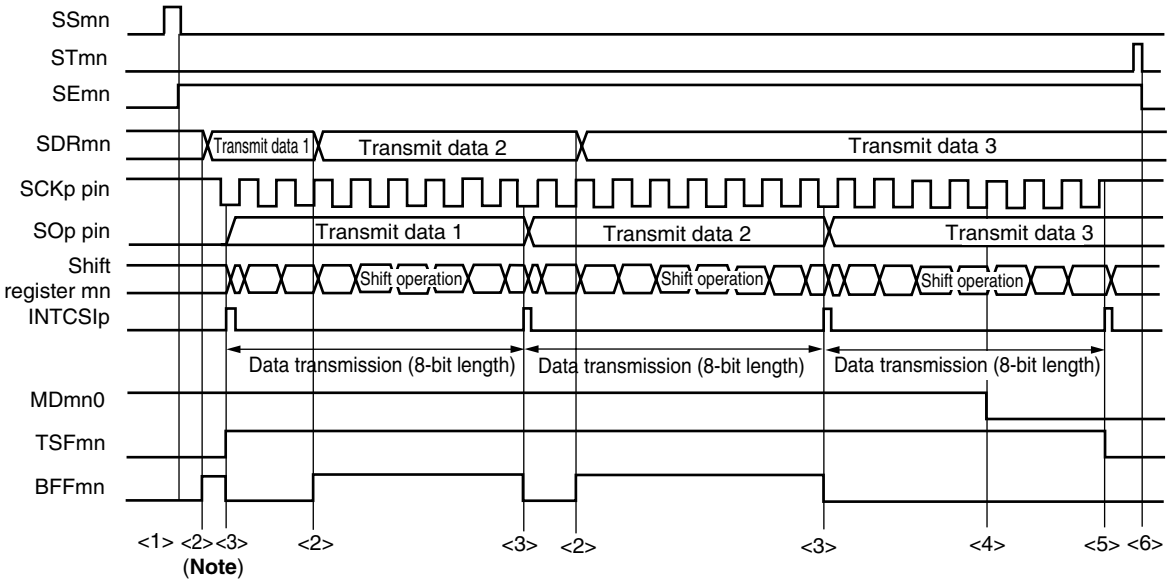


Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

(4) Processing flow (in continuous transmission mode)

Figure 14-52. Timing Chart of Slave Transmission (in Continuous Transmission Mode) (Type 1: DAPmn = 0, CKPmn = 0)

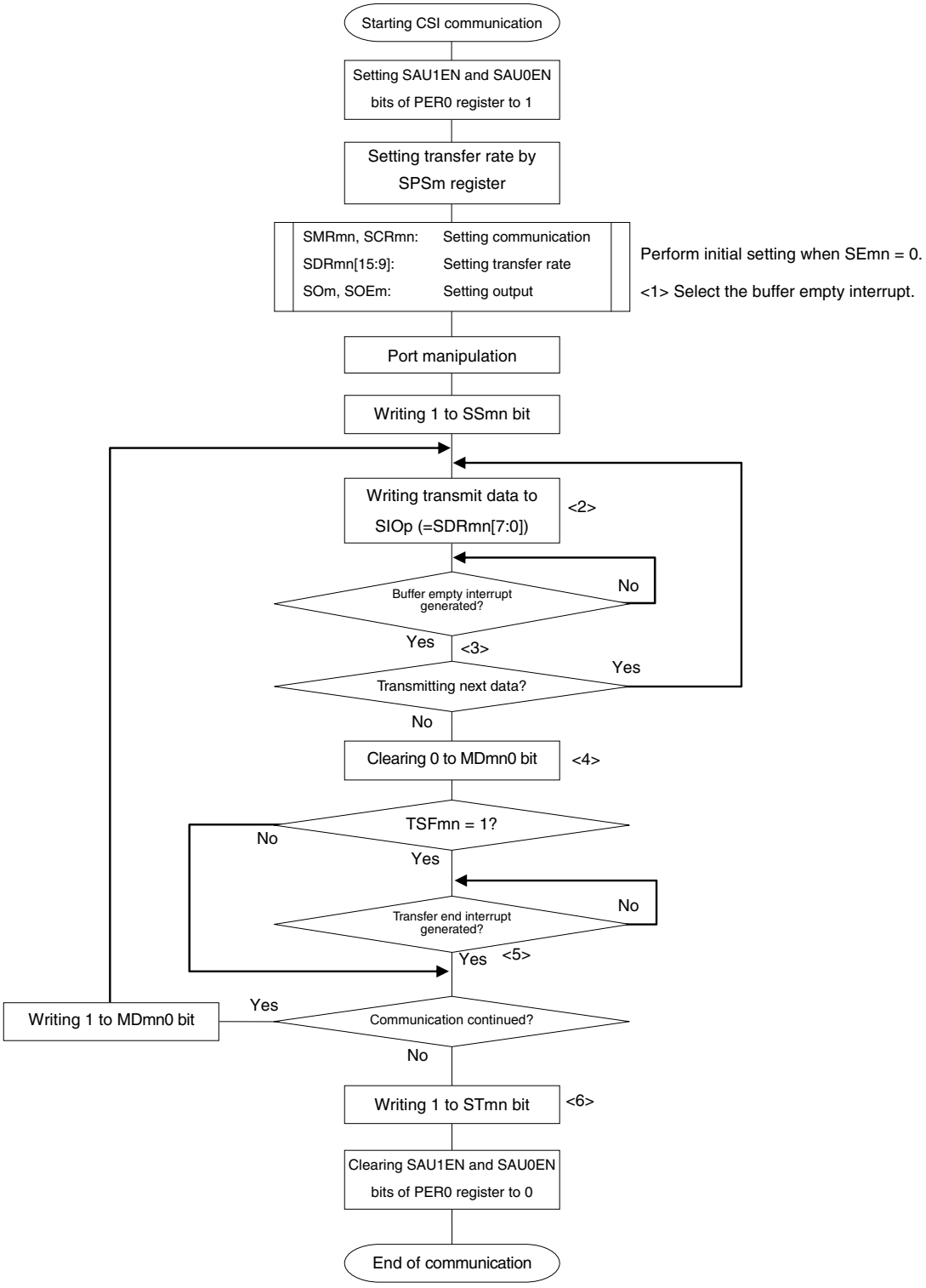


Note When transmit data is written to the SDRmn register while BFFmn = 1, the transmit data is overwritten.

Caution The MDmn0 bit can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

Figure 14-53. Flowchart of Slave Transmission (in Continuous Transmission Mode)



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remarks 1. <1> to <6> in the figure correspond to <1> to <6> in **Figure 14-52 Timing Chart of Slave Transmission (in Continuous Transmission Mode).**

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

14.5.5 Slave reception

Slave reception is that the 78K0R/KE3-A microcontrollers receive data from another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI00	CSI10	CSI20
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1
Pins used	$\overline{SCK00}$, SI00	$\overline{SCK10}$, SI10	$\overline{SCK20}$, SI20
Interrupt	INTCSI00	INTCSI10	INTCSI20
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)		
Error detection flag	Overrun error detection flag (OVFmn) only		
Transfer data length	7 or 8 bits		
Transfer rate	Max. f_{MCK}/6 [MHz]Notes 1, 2		
Data phase	Selectable by DAPmn bit • 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 CKPmn bit • CKPmn = 0: Forward • CKPmn = 1: Reverse		
Data direction	MSB or LSB first		

Notes 1. Because the external serial clock input to pins $\overline{SCK00}$, $\overline{SCK10}$, and $\overline{SCK20}$ is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [MHz].

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{MCK}: Operation clock (MCK) frequency of target channel

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

(1) Register setting

Figure 14-54. Example of Contents of Registers for Slave Reception of 3-Wire Serial I/O (CSI00, CSI10, CSI20)

(a) 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	1	CKOm2 ×	1	CKOm0 ×	0	0	0	0	1	SOm2 ×	1	SOm0 ×

(b) Serial output enable register m (SOEm) … Clears only the bits of the target channel to 0.

	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	0	SOEm0 0/1

(c) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 0/1	SSm0 0/1

(d) 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 mode of channel n
0: Transfer end interrupt

(e) 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	DLSmn2 1	DLSmn1 1	DLSmn0 0/1

(f) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

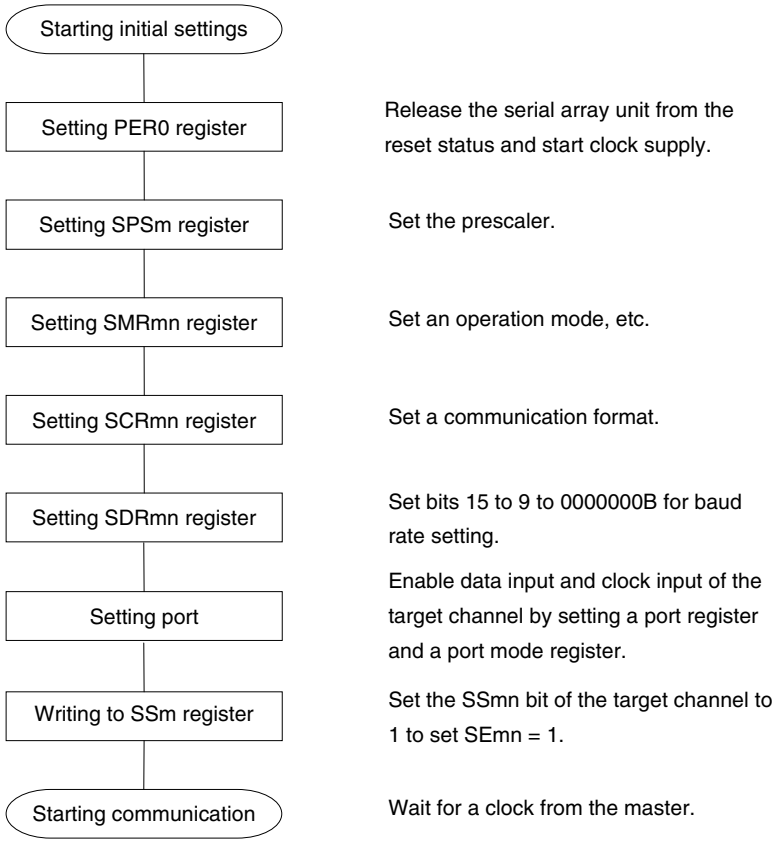
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 (baud rate setting)							0	Receive data register							

SIOp

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)
□: Setting is fixed in the 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

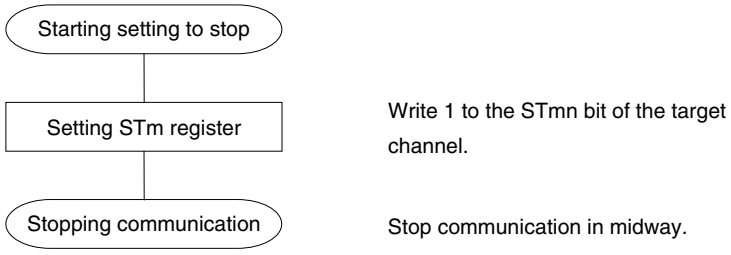
Figure 14-55. Initial Setting Procedure for Slave Reception



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

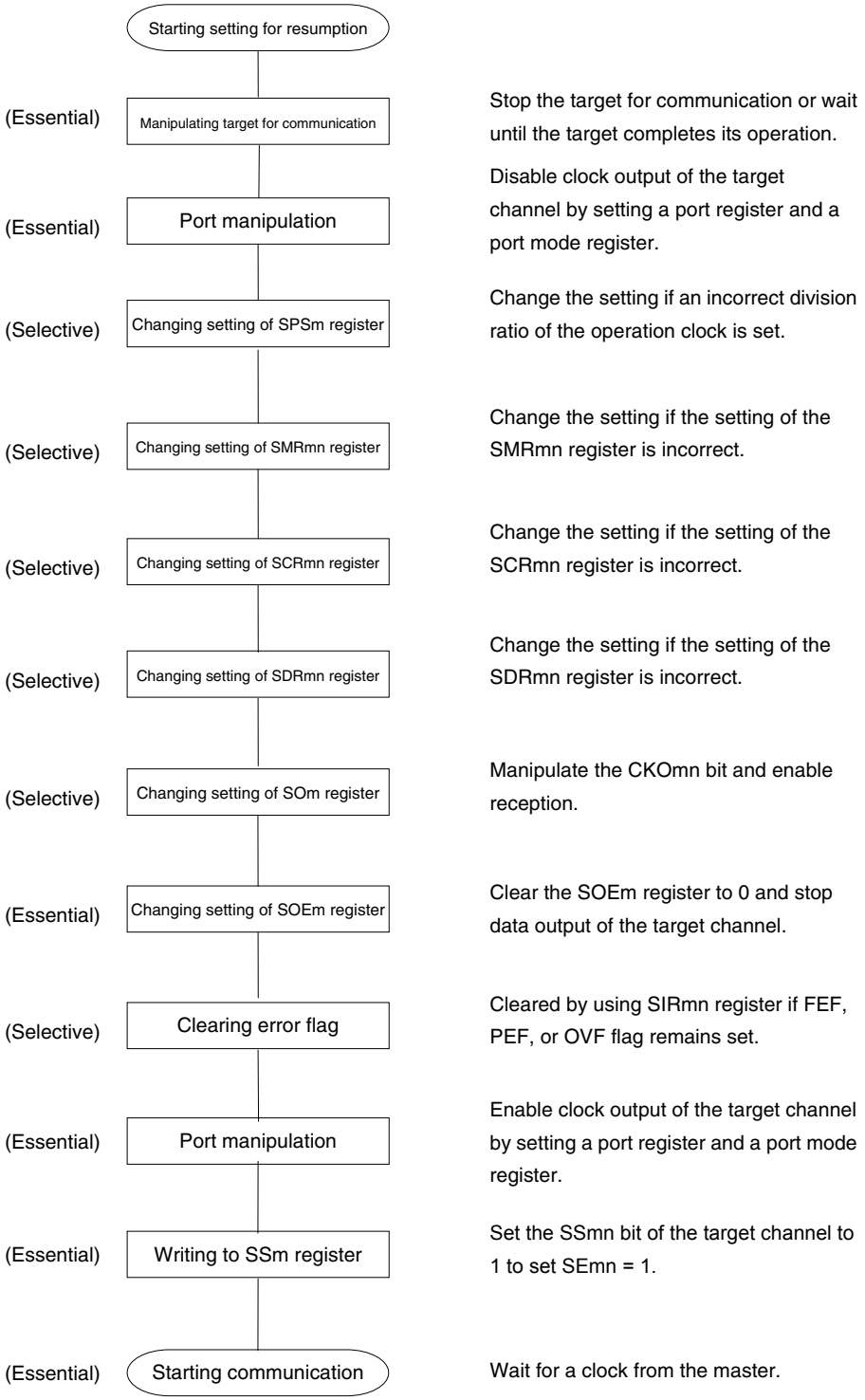
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

Figure 14-56. Procedure for Stopping Slave Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

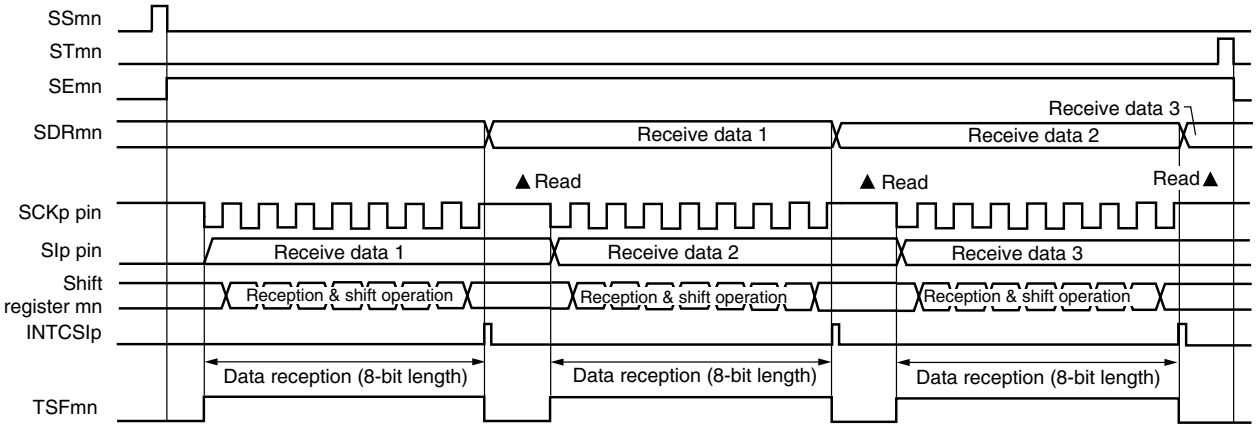
Figure 14-57. Procedure for Resuming Slave Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

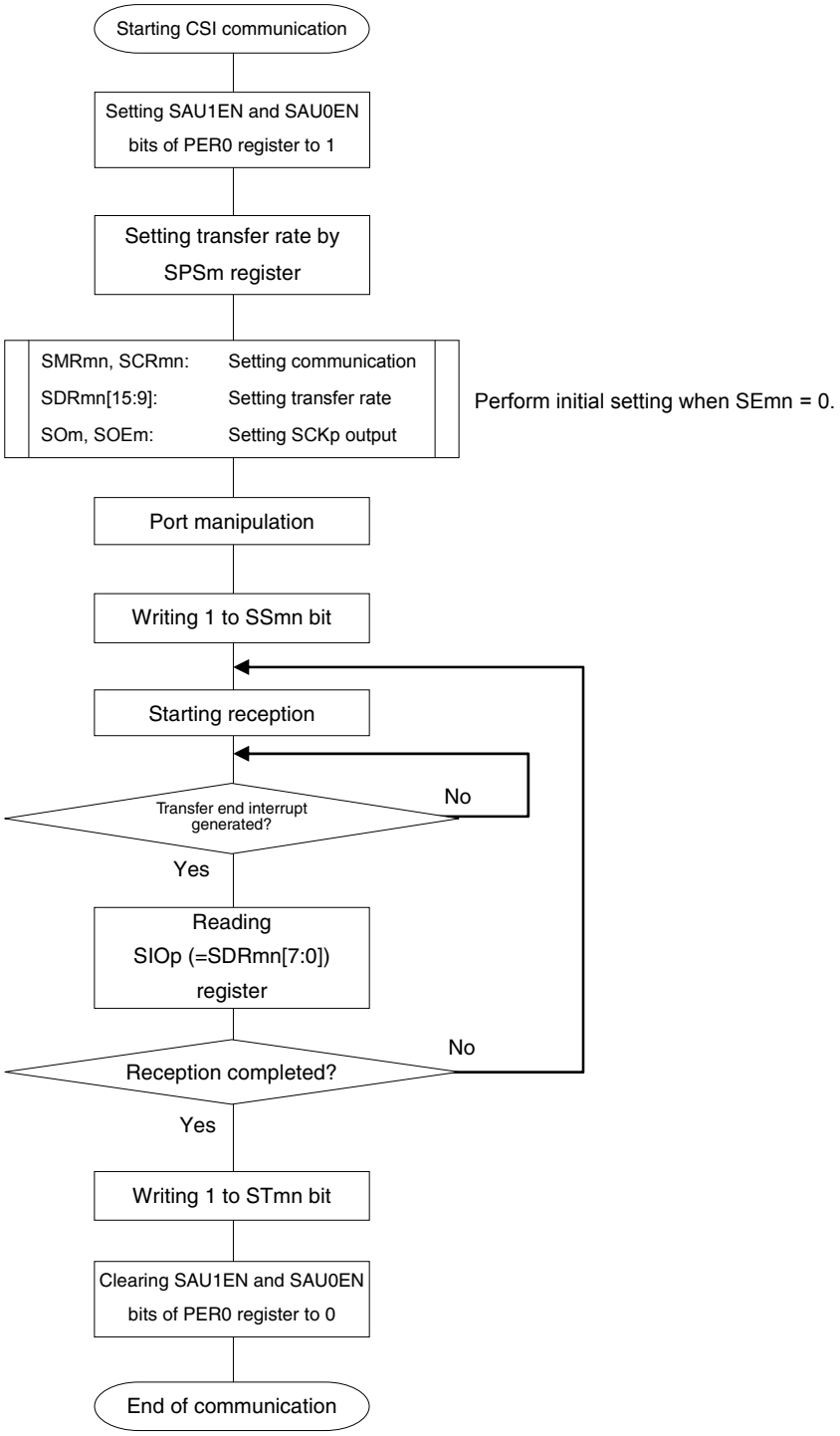
(3) Processing flow (in single-reception mode)

Figure 14-58. Timing Chart of Slave Reception (in Single-Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

Figure 14-59. Flowchart of Slave Reception (in Single-Reception Mode)



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2) , p: CSI number (p = 00, 10, 20)

14.5.6 Slave transmission/reception

Slave transmission/reception is that the 78K0R/KE3-A microcontrollers transmit/receive data to/from another device in the state of a transfer clock being input from another device.

3-Wire Serial I/O	CSI00	CSI10	CSI20
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1
Pins used	$\overline{SCK00}$, SI00, SO00	$\overline{SCK10}$, SI10, SO10	$\overline{SCK20}$, SI20, SO20
Interrupt	INTCSI00	INTCSI10	INTCSI20
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.		
Error detection flag	Overrun error detection flag (OVFmn) only		
Transfer data length	7 or 8 bits		
Transfer rate	Max. f_{MCK}/6 [MHz]Notes 1, 2		
Data phase	Selectable by DAPmn bit • 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 CKPmn bit • CKPmn = 0: Forward • CKPmn = 1: Reverse		
Data direction	MSB or LSB first		

Notes 1. Because the external serial clock input to pins $\overline{SCK00}$, $\overline{SCK10}$, and $\overline{SCK20}$ is sampled internally and used, the fastest transfer rate is f_{MCK}/6 [MHz].

2. Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remarks 1. f_{MCK}: Operation clock (MCK) frequency of target channel

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

(1) Register setting

Figure 14-60. Example of Contents of Registers for Slave Transmission/Reception of 3-Wire Serial I/O (CSI00, CSI10, CSI20)

(a) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 ×	1	CKOm0 ×	0	0	0	0	1	SOm2 0/1	1	SOm0 0/1

(b) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 0/1	0	SOEm0 0/1

(c) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 0/1	SSm0 0/1

(d) 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 mode of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

(e) 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	DLSmn2 1	DLSmn1 1	DLSmn0 0/1

(f) Serial data register mn (SDRmn) (lower 8 bits: SIOp)

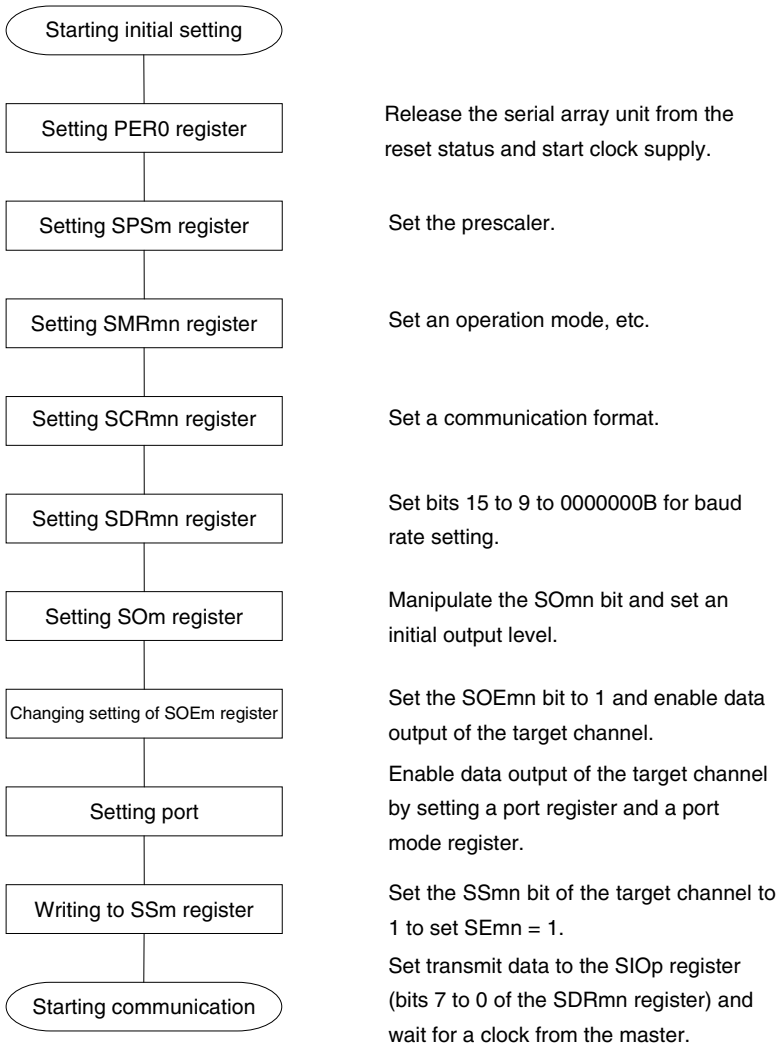
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	0000000 (baud rate setting)							0	Transmit data setting/receive data register							

SIOp

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)
□ : Setting is fixed in the CSI slave transmission/reception mode, ▒: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

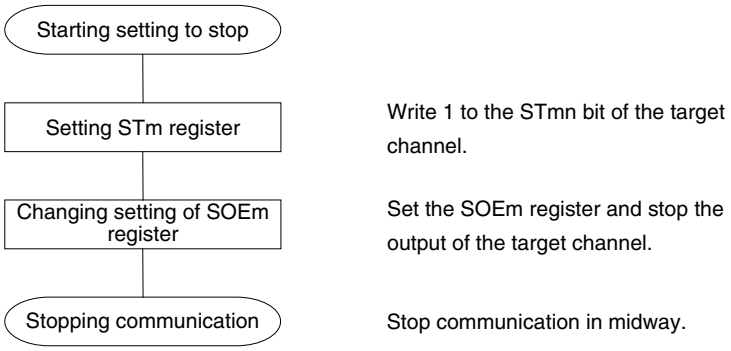
Figure 14-61. Initial Setting Procedure for Slave Transmission/Reception



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

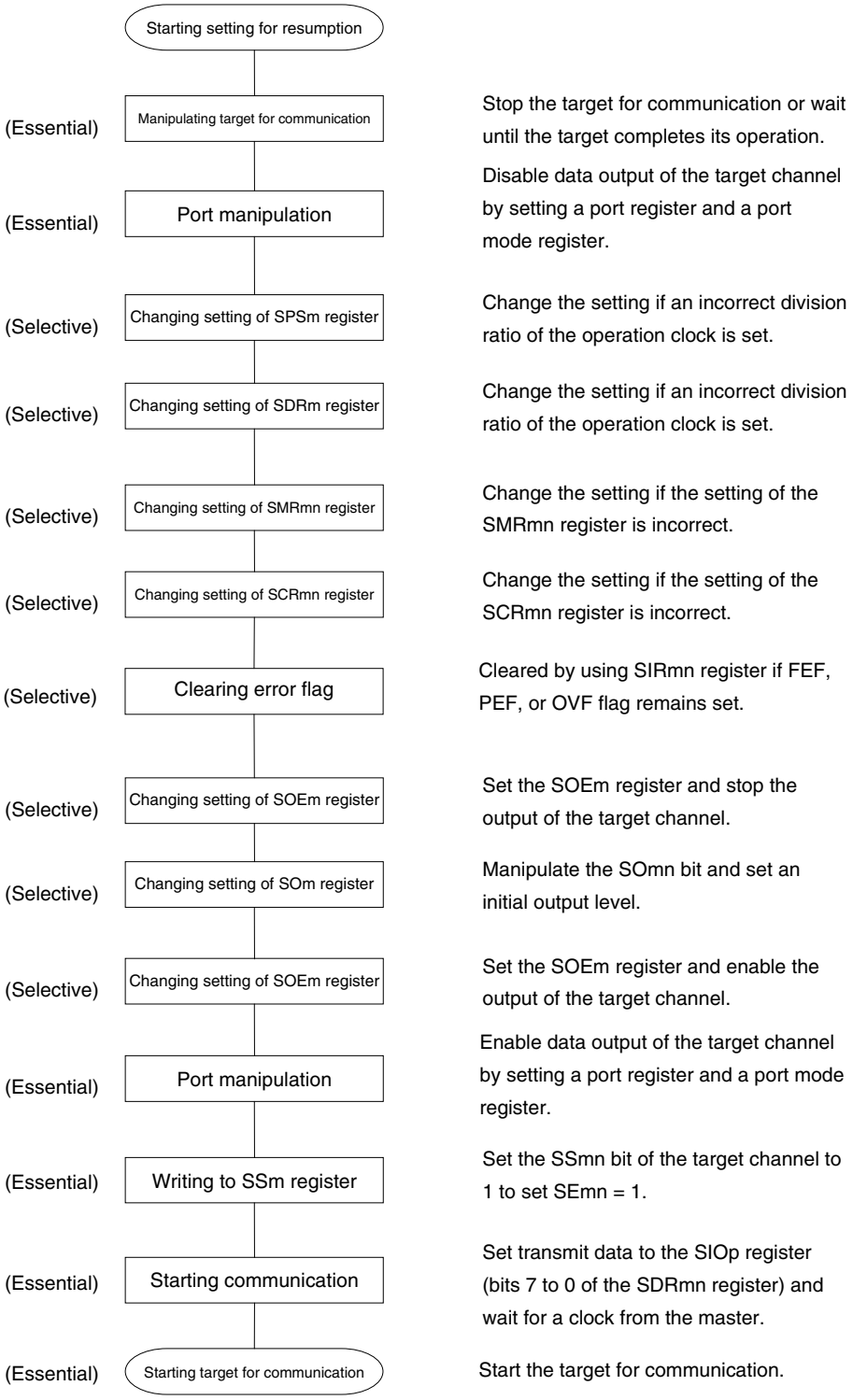
Figure 14-62. Procedure for Stopping Slave Transmission/Reception



Remarks 1. Even after communication is stopped, the pin level is retained. To resume the operation, reset the SOm register (see **Figure 14-63 Procedure for Resuming Slave Transmission/Reception**).

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

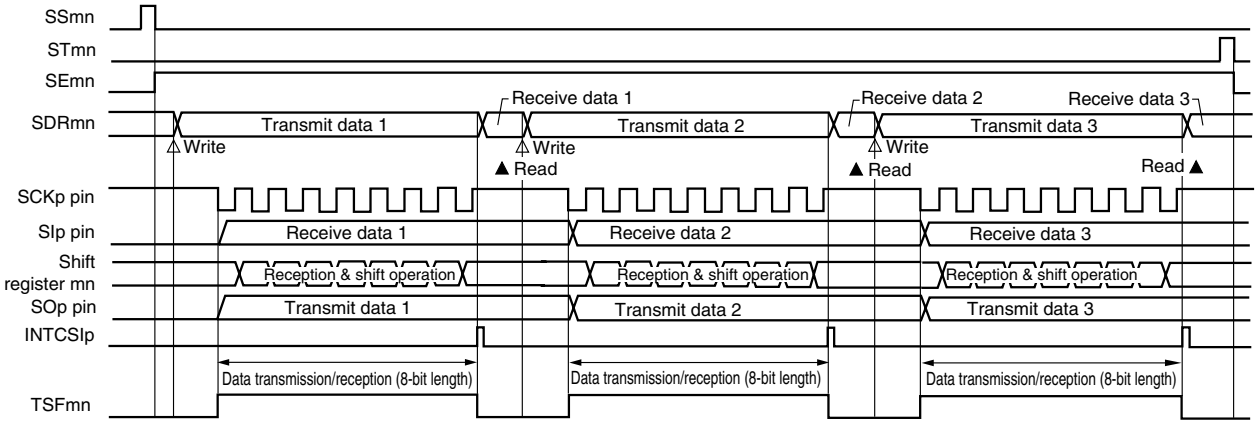
Figure 14-63. Procedure for Resuming Slave Transmission/Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

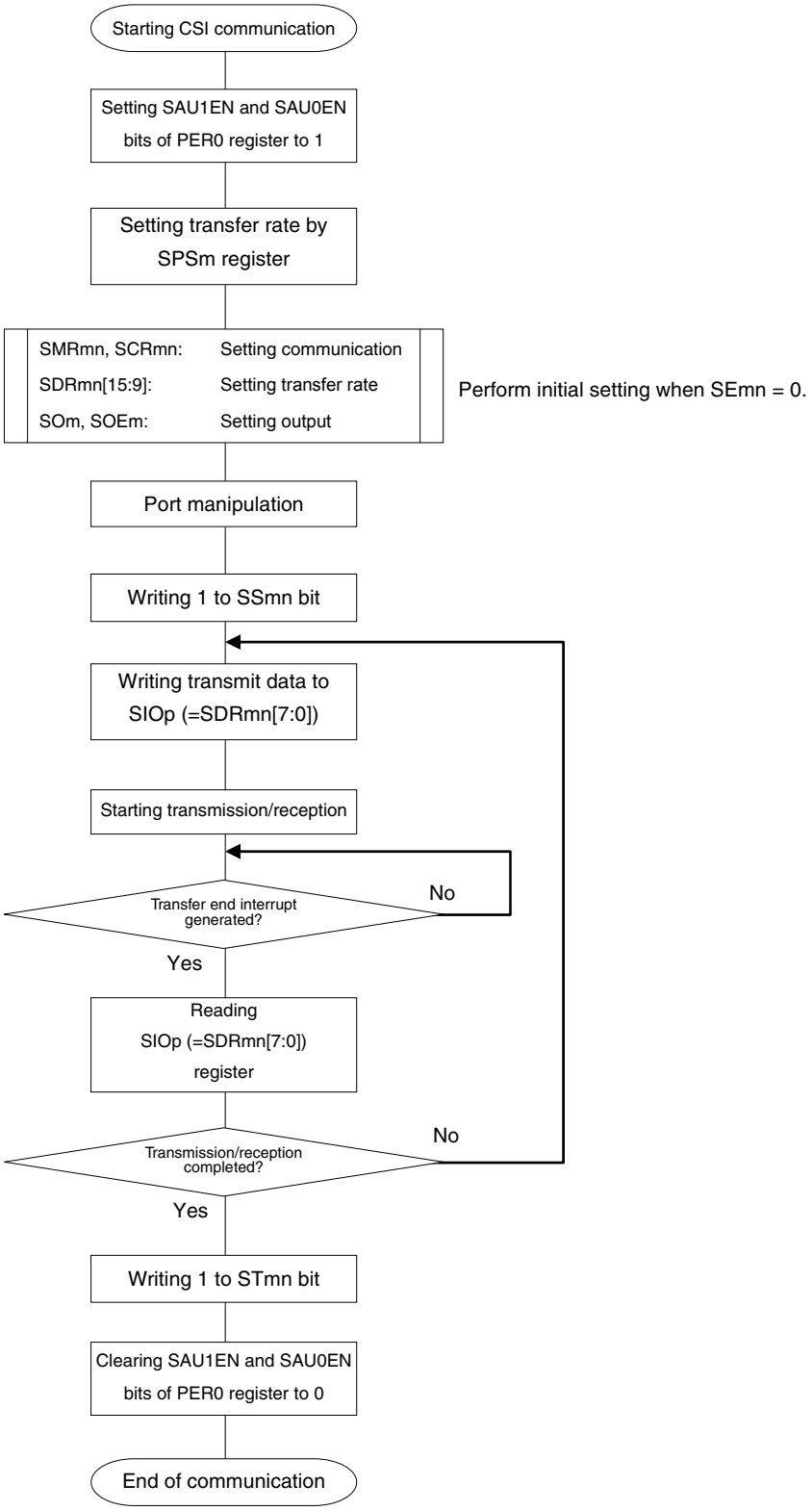
(3) Processing flow (in single-transmission/reception mode)

Figure 14-64. Timing Chart of Slave Transmission/Reception (in Single-Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

Figure 14-65. Flowchart of Slave Transmission/Reception (in Single-Transmission/Reception Mode)

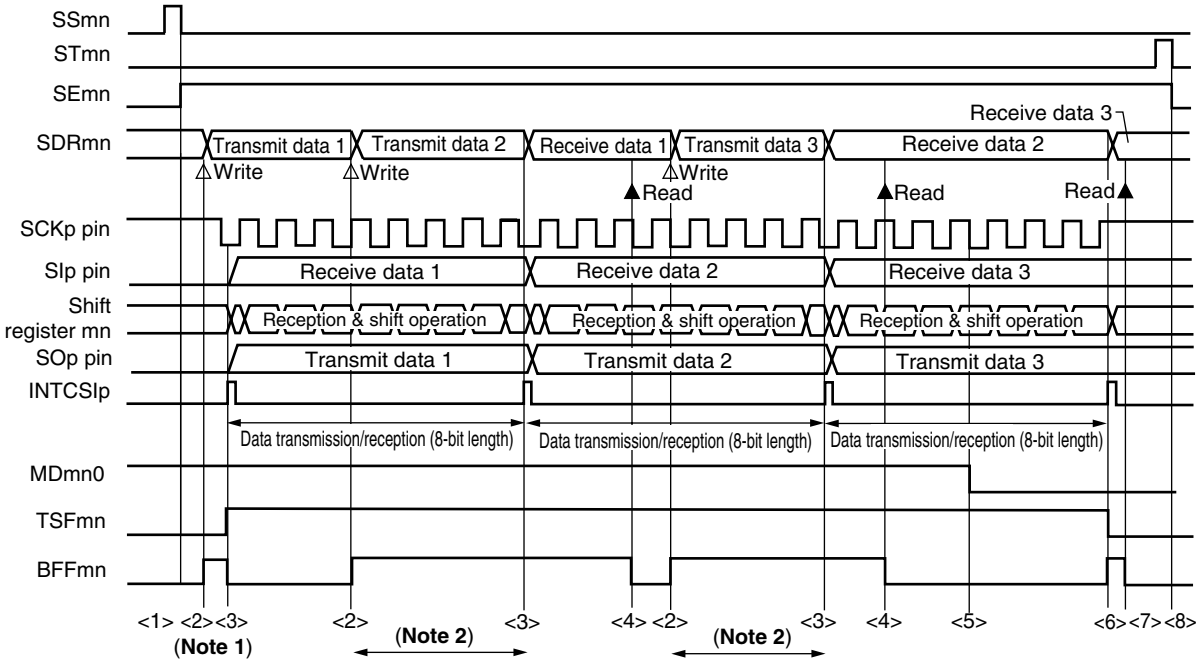


Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

(4) Processing flow (in continuous transmission/reception mode)

Figure 14-66. Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode) (Type 1: DAPmn = 0, CKPmn = 0)



Notes 1. When transmit data is written to the SDRmn register while BFFmn = 1, the transmit data is overwritten.

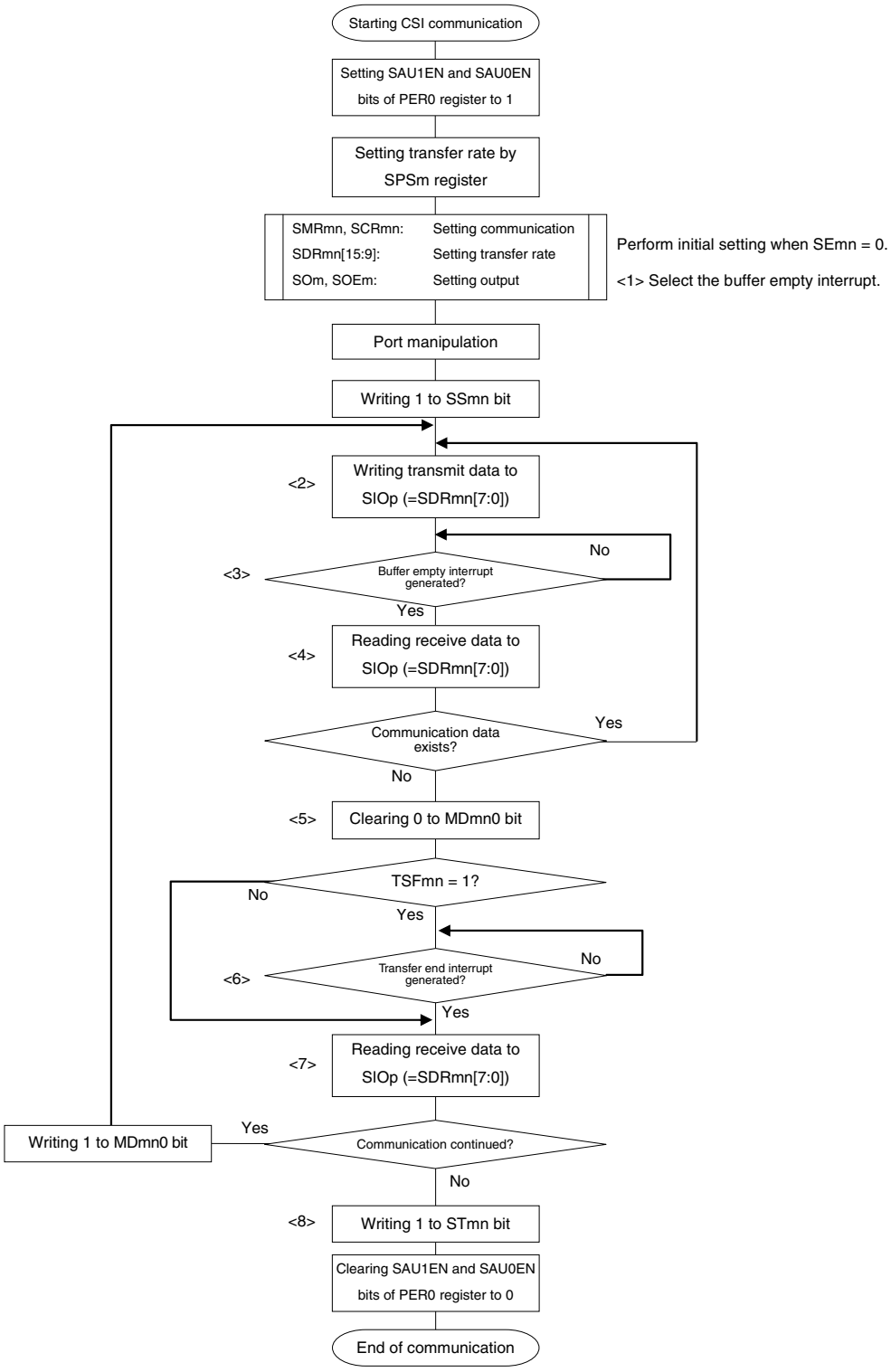
2. The transmit data can be read by reading the SDRmn register during this period. At this time, the transfer operation is not affected.

Caution The MDmn0 bit can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 14-67 Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

Figure 14-67. Flowchart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remarks 1. <1> to <8> in the figure correspond to <1> to <8> in **Figure 14-66 Timing Chart of Slave Transmission/Reception (in Continuous Transmission/Reception Mode)**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), p: CSI number (p = 00, 10, 20)

14.5.7 Calculating transfer clock frequency

The transfer clock frequency for 3-wire serial I/O (CSI00, CSI10, CSI20) communication can be calculated by the following expressions.

(1) Master

(Transfer clock frequency) = {Operation clock (MCK) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [Hz]

(2) Slave

(Transfer clock frequency) = {Frequency of serial clock ($\overline{SCK}$) supplied by master}[Note] [Hz]

Note The permissible maximum frequency is the smaller of f_{CLK}/6 [MHz] and f_{MCK}/2 [MHz].

Remarks 1. The value of SDRmn[15:9] is the value of bits 15 to 9 of the SDRmn register (0000000B to 1111111B) and therefore is 0 to 127.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

The operation clock (MCK) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 14-2. Selection of operation clock

SMRmn Register	SPSm Register								Operation Clock (MCK) [Note1]	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	20 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	10 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	5 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	2.5 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.25 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	625 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	313 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	156 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	78.1 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	39.1 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	19.5 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	9.77 kHz
	X	X	X	X	1	1	1	1	INTTM02 if m = 0, INTTM03 if m = 1 [Note2]	
1	0	0	0	0	X	X	X	X	f_{CLK}	20 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	10 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	5 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	2.5 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.25 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	625 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	313 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	156 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	78.1 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	39.1 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	19.5 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	9.77 kHz
	1	1	1	1	X	X	X	X	INTTM02 if m = 0, INTTM03 if m = 1 [Note2]	
Other than above									Setting prohibited	

Notes 1. When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (STm = 000FH) the operation of the serial array unit (SAUm). When selecting INTTM02 and INTTM03 for the operation clock, also stop the timer array unit (TAU0) (TT0 = 00FFH).

2. SAUm can be operated at a fixed division ratio of the subsystem clock, regardless of the f_{CLK} frequency (main system clock, subsystem clock), by operating the interval timer for which $f_{SUB}/4$ has been selected as the count clock (setting TIS02 (if m = 0) or TIS03 (if m = 1) of the TIS0 register to 1) and selecting INTTM02 and INTTM03 by using the SPSm register in channels 2 and 3 of TAU0. When changing f_{CLK}, however, SAUm and TAU0 must be stopped as described in Note 1 above.

Remarks 1. X: Don't care

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

14.6 Operation of UART (UART0, UART1, UART2, UART3) Communication

This is a start-stop synchronization function using two lines: serial data transmission (TxD) and serial data reception (RxD) lines. It transmits or receives data in asynchronization with the party of communication (by using an internal baud rate). Full-duplex UART communication can be realized by using two channels, one dedicated to transmission (even channel) and the other to reception (odd channel).

[Data transmission/reception]

- Data length of 5, 7, or 8 bits
- Select the MSB/LSB first
- Level setting of transmit/receive data and select of reverse
- Parity bit appending and parity check functions
- Stop bit appending

[Interrupt function]

- Transfer end interrupt/buffer empty interrupt
- Error interrupt in case of framing error, parity error, or overrun error

[Error detection flag]

- Framing error, parity error, or overrun error

The LIN-bus is supported in UART3 (2, 3 channels of unit 1)

[LIN-bus functions]

- Wakeup signal detection
- Sync break field (SBF) detection
- Sync field measurement, baud rate calculation

} External interrupt (INTP0) or timer array unit 0 (TAU0) is used.

UART0 uses channels 0 and 1 of SAU0.
UART1 uses channels 2 and 3 of SAU0.
UART2 uses channels 0 and 1 of SAU1.
UART3 uses channels 2 and 3 of SAU1.

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I^2C
0	0	CSI00	UART0	–
	1	–		–
	2	CSI10	UART1	IIC10
	3	–		–
1	0	CSI20	UART2	IIC20
	1	–		–
	2	–	UART3 (supporting LIN-bus)	–
	3	–		–

Caution When using serial array units 0 and 1 as UARTs, the channels of both the transmitting side (even-number channel) and the receiving side (odd-number channel) can be used only as UARTs.

UART performs the following four types of communication operations.

- UART transmission (See **14.6.1**.)
- UART reception (See **14.6.2**.)
- LIN transmission (UART3 only) (See **14.6.3**.)
- LIN reception (UART 3 only) (See **14.6.4**.)

14.6.1 UART transmission

UART transmission is an operation to transmit data from the 78K0R/KE3-A microcontrollers to another device asynchronously (start-stop synchronization).

Of two channels used for UART, the even channel is used for UART transmission.

UART	UART0	UART1	UART2	UART3
Target channel	Channel 0 of SAU0	Channel 2 of SAU0	Channel 0 of SAU1	Channel 2 of SAU1
Pins used	TxD0	TxD1	TxD2	TxD3
Interrupt	INTST0	INTST1	INTST2	INTST3
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	None			
Transfer data length	5, 7, or 8 bits			
Transfer rate	Max. f_{MCK}/6 [bps] (SDRmn [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] Note			
Data phase	Forward 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 Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock (MCK) frequency of target channel
f_{CLK}: System clock frequency
m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

(1) Register setting

Figure 14-68. Example of Contents of Registers for UART Transmission of UART (UART0, UART1, UART2, UART3) (1/2)

(a) Serial output register m (SOm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 ×	1	CKOm0 ×	0	0	0	0	1	SOm2 0/1 Note	1	SOm0 0/1 Note

(b) Serial output enable register m (SOEm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOEm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOEm2 0/1	0	SOEm0 0/1

(c) Serial channel start register m (SSm) … Sets only the bits of the target channel to 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

(d) Serial output level register m (SOLm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOLm	0	0	0	0	0	0	0	0	0	0	0	0	0	SOLm2 0/1	0	SOLm0 0/1

0: Forward (normal) transmission
1: Reverse transmission

(e) 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 1	MDmn0 0/1

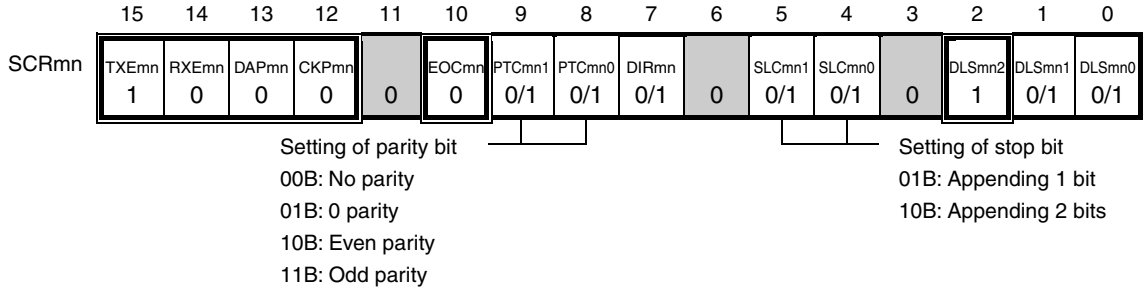
Operation mode of channel n
0: Transfer end interrupt
1: Buffer empty interrupt

Note Before transmission is started, be sure to set to 1 when the SOLmn bit of the target channel is set to 0, and set to 0 when the SOLmn bit of the target channel is set to 1. The value varies depending on the communication data during communication operation.

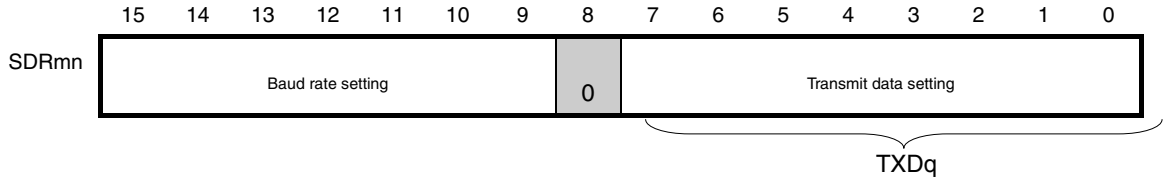
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 3)
□: Setting is fixed in the UART transmission mode, ▒: Setting disabled (fixed by hardware)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

Figure 14-68. Example of Contents of Registers for UART Transmission of UART (UART0, UART1, UART2, UART3) (2/2)

(f) Serial communication operation setting register mn (SCRmn)



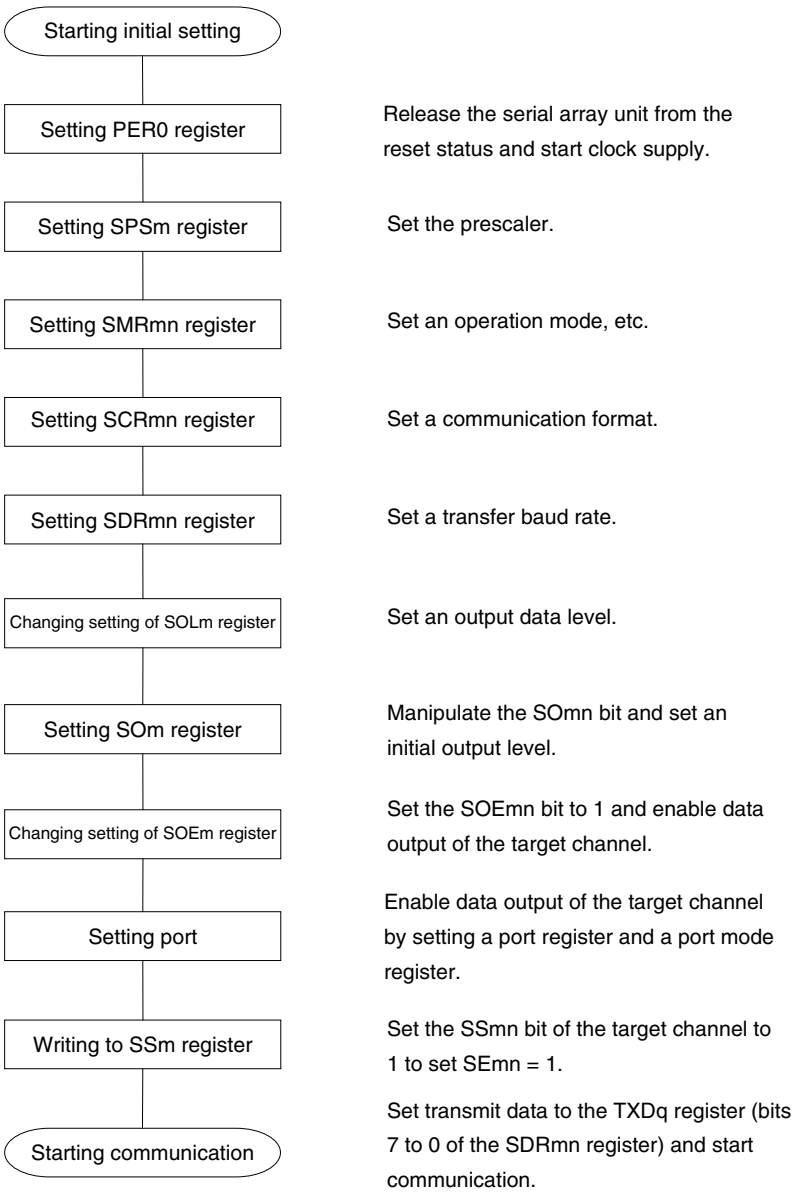
(g) Serial data register mn (SDRmn) (lower 8 bits: TXDq)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 3)
□: Setting is fixed in the UART transmission mode, ■ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

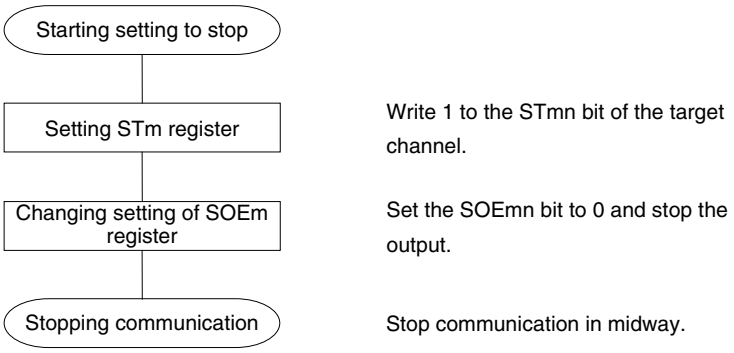
Figure 14-69. Initial Setting Procedure for UART Transmission



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 3)

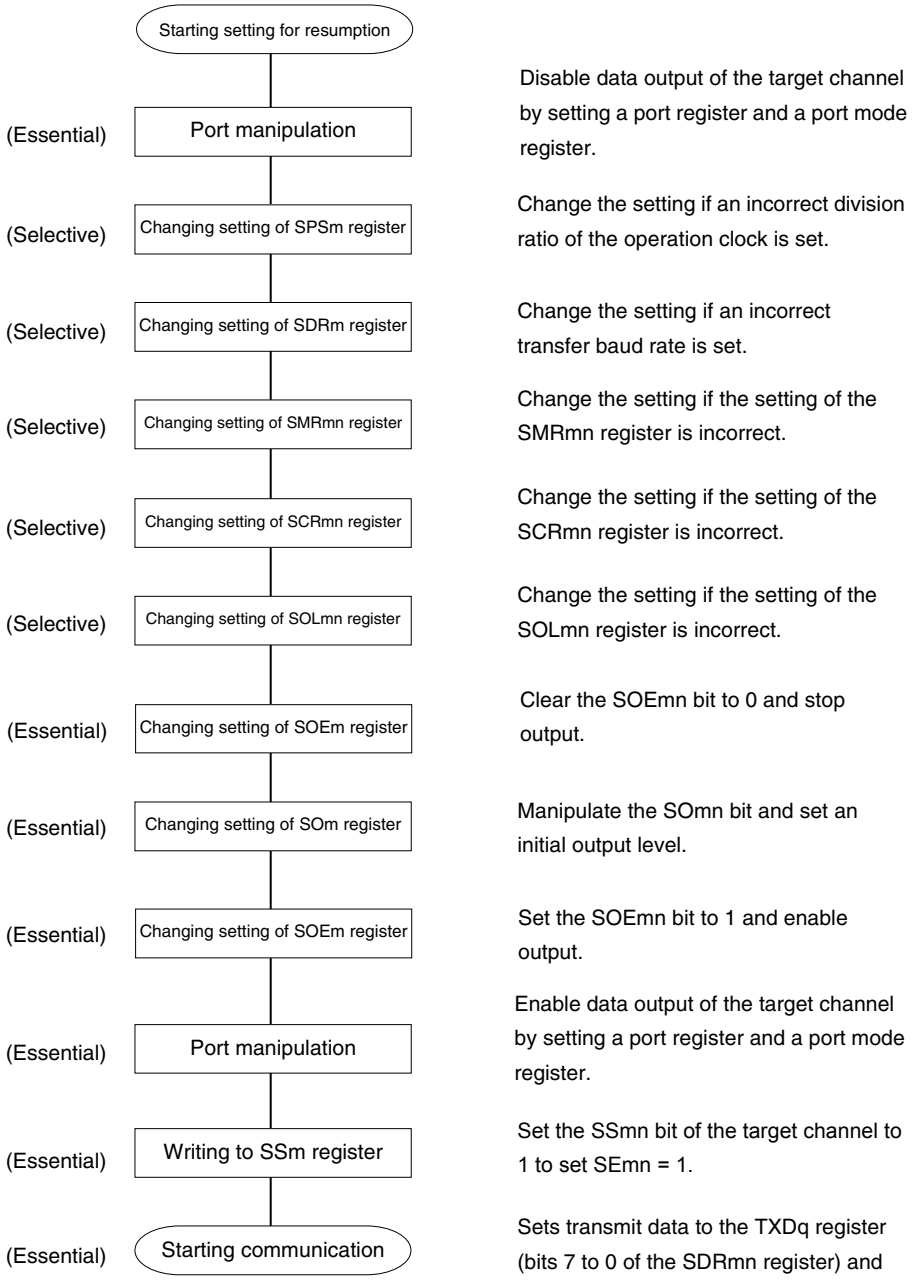
Figure 14-70. Procedure for Stopping UART Transmission



Remarks 1. Even after communication is stopped, the pin level is retained. To resume the operation, re-set the SOm register (see **Figure 14-71 Procedure for Resuming UART Transmission**).

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

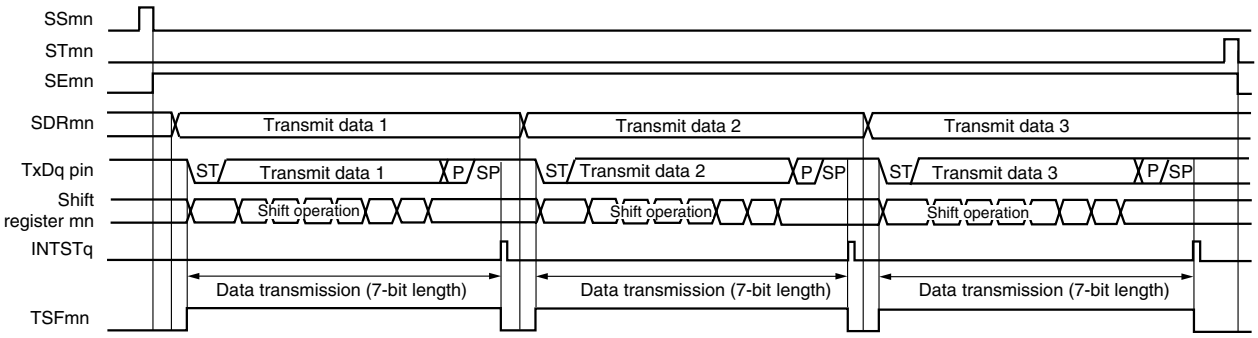
Figure 14-71. Procedure for Resuming UART Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 3)

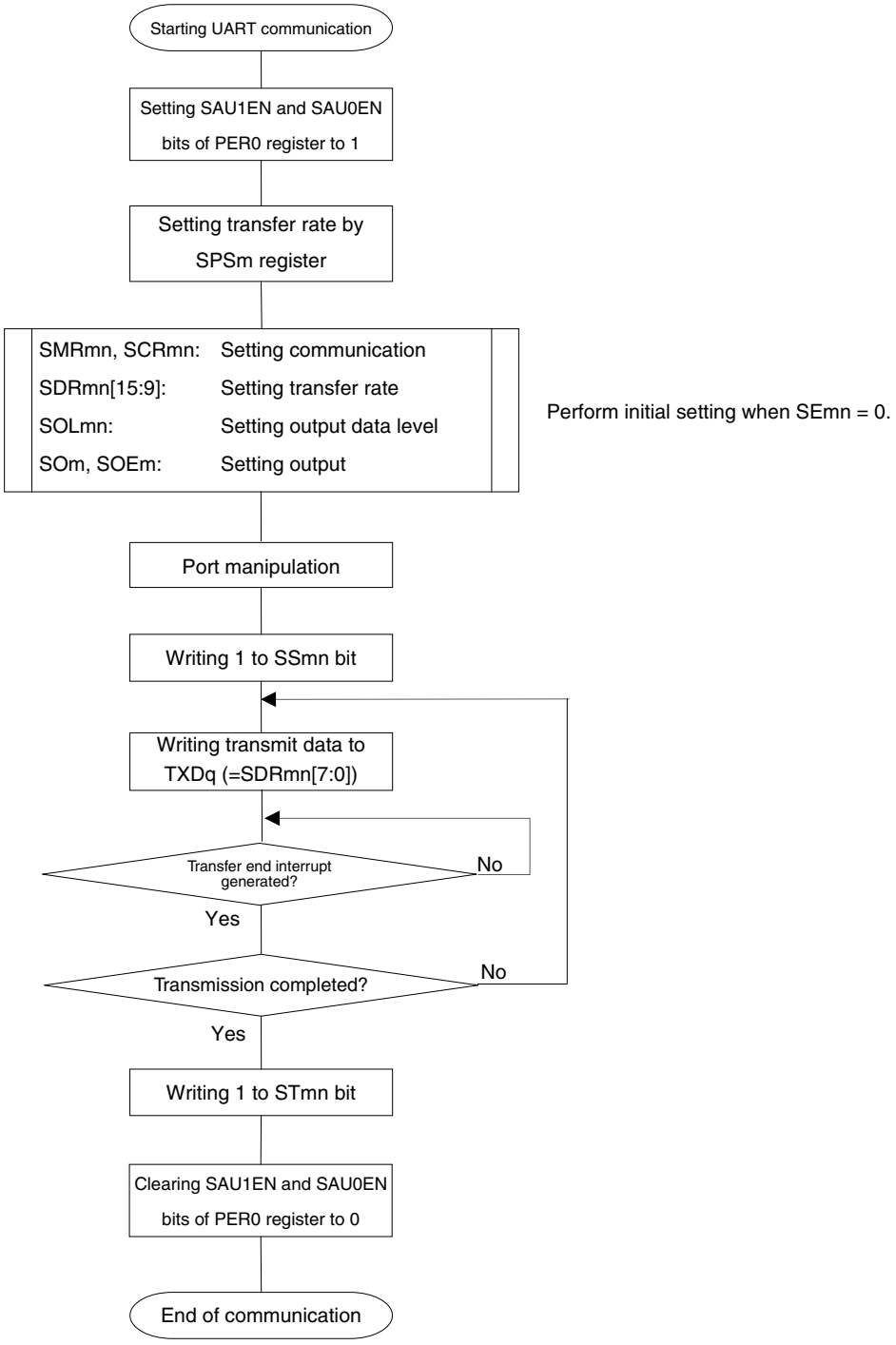
(3) Processing flow (in single-transmission mode)

Figure 14-72. Timing Chart of UART Transmission (in Single-Transmission Mode)



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 3)

Figure 14-73. Flowchart of UART Transmission (in Single-Transmission Mode)

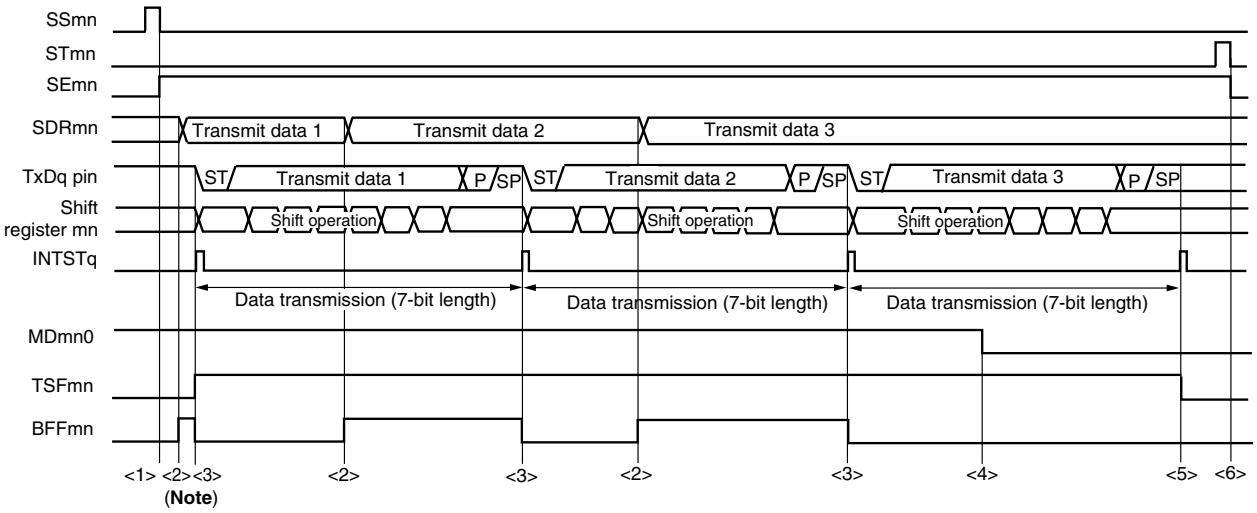


Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 3)

(4) Processing flow (in continuous transmission mode)

Figure 14-74. Timing Chart of UART Transmission (in Continuous Transmission Mode)

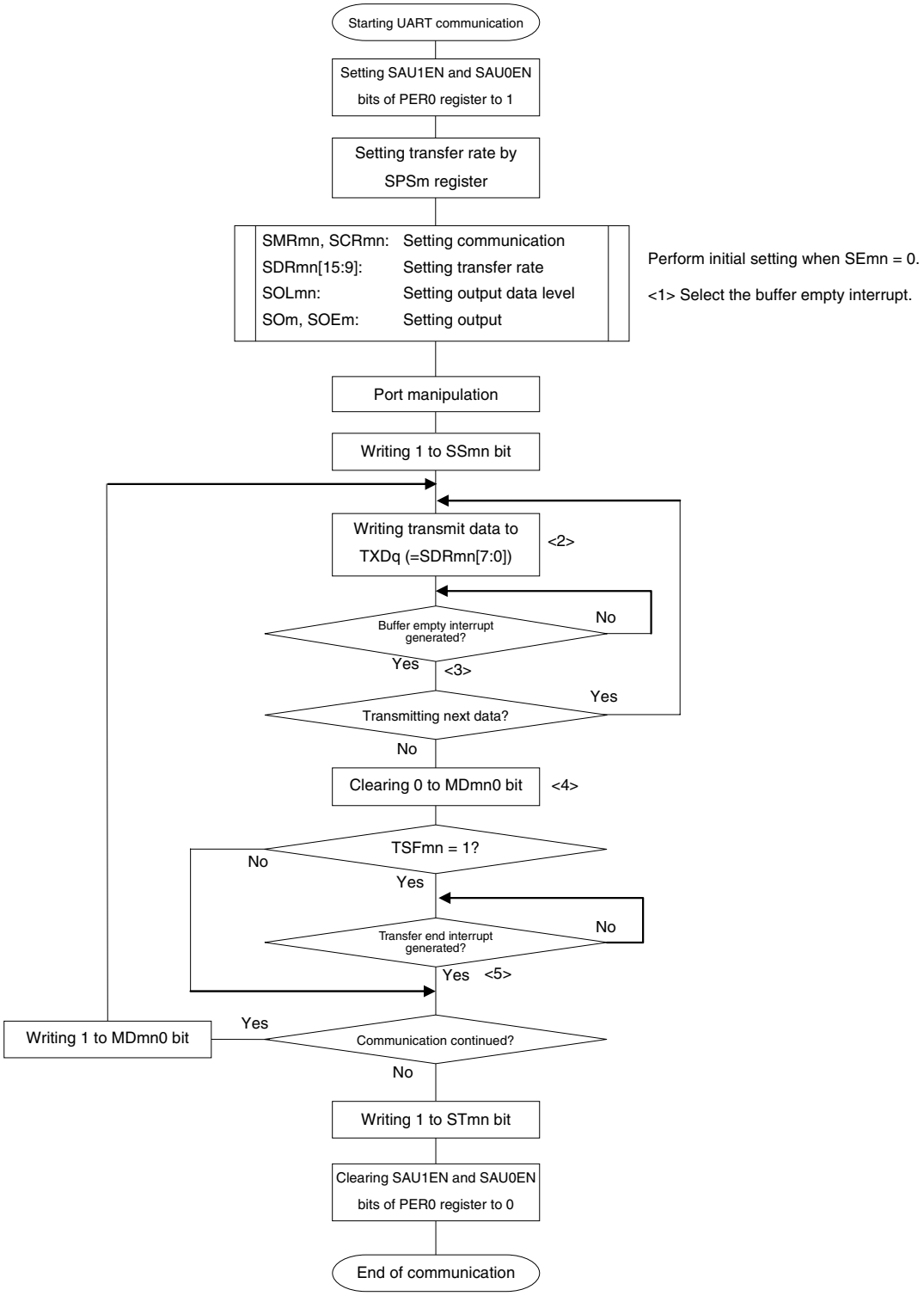


Note When transmit data is written to the SDRmn register while BFFmn = 1, the transmit data is overwritten.

Caution The MDmn0 bit can be rewritten even during operation.
However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 3)

Figure 14-75. Flowchart of UART Transmission (in Continuous Transmission Mode)



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remarks 1. <1> to <5> in the figure correspond to <1> to <5> in **Figure 14-74 Timing Chart of UART Transmission (in Continuous Transmission Mode)**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), q: UART number (q = 0 to 3)

14.6.2 UART reception

UART reception is an operation wherein the 78K0R/KE3-A microcontrollers asynchronously receive data from another device (start-stop synchronization).

For UART reception, the odd-number channel of the two channels used for UART is used. The SMR register of both the odd- and even-numbered channels must be set.

UART	UART0	UART1	UART2	UART3
Target channel	Channel 1 of SAU0	Channel 3 of SAU0	Channel 1 of SAU1	Channel 3 of SAU1
Pins used	RxD0	RxD1	RxD2	RxD3
Interrupt	INTSR0	INTSR1	INTSR2	INTSR3
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error interrupt	INTSRE0	INTSRE1	INTSRE2	INTSRE3
Error detection flag	• Framing error detection flag (FEFmn) • Parity error detection flag (PEFmn) • Overrun error detection flag (OVFmn)			
Transfer data length	5, 7 or 8 bits			
Transfer rate	Max. f_{MCK}/6 [bps] (SDRmn [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] Note			
Data phase	Forward 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 Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock (MCK) frequency of target channel
f_{CLK}: System clock frequency
m: Unit number (m = 0, 1), n: Channel number (n = 1, 3)

(1) Register setting

Figure 14-76. Example of Contents of Registers for UART Reception of UART (UART0, UART1, UART2, UART3) (1/2)

(a) 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	1	CKOm2 ×	1	CKOm0 ×	0	0	0	0	1	SOm2 ×	1	SOm0 ×

(b) Serial output enable register m (SOEm) … Sets the bits of the target channel to 0.

	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	SOEm0 ×

(c) Serial channel start register m (SSm) … Sets only the bits of the target channel is 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 0/1	SSm2 ×	SSm1 0/1	SSm0 ×

(d) 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

0: Forward (normal) reception
1: Reverse reception

Operation mode of channel n
0: Transfer end interrupt

(e) 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	STSmr 0	0	SISmr0 0	1	0	0	MDmr2 0	MDmr1 1	MDmr0 0/1

Same setting value as CKSmn

Operation mode of channel r
0: Transfer end interrupt
1: Buffer empty interrupt

(f) Serial communication operation setting register mn (SCRmn)

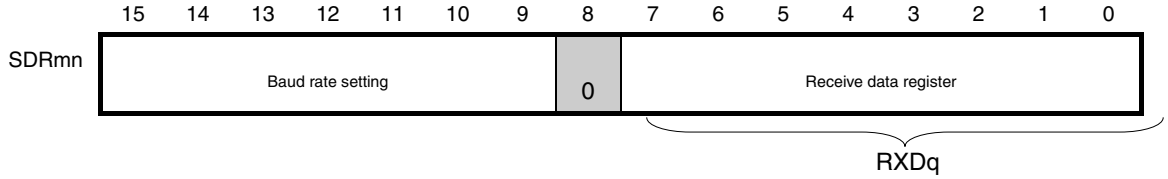
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXEmn 0	RXEmn 1	DAPmn 0	CKPmn 0	0	EOCmn 1	PTCmn1 0/1	PTCmn0 0/1	DIRmn 0/1	0	SLCmn1 0	SLCmn0 1	0	DLSmn2 1	DLSmn1 0/1	DLSmn0 0/1

Caution For the UART reception, be sure to set SMRmr of channel r that is to be paired with channel n.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), r: Channel number (r = n − 1),
q: UART number (q = 0 to 3)
□ : 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 14-76. Example of Contents of Registers for UART Reception of UART (UART0, UART1, UART2, UART3) (2/2)

(g) Serial data register mn (SDRmn) (lower 8 bits: RXDq)

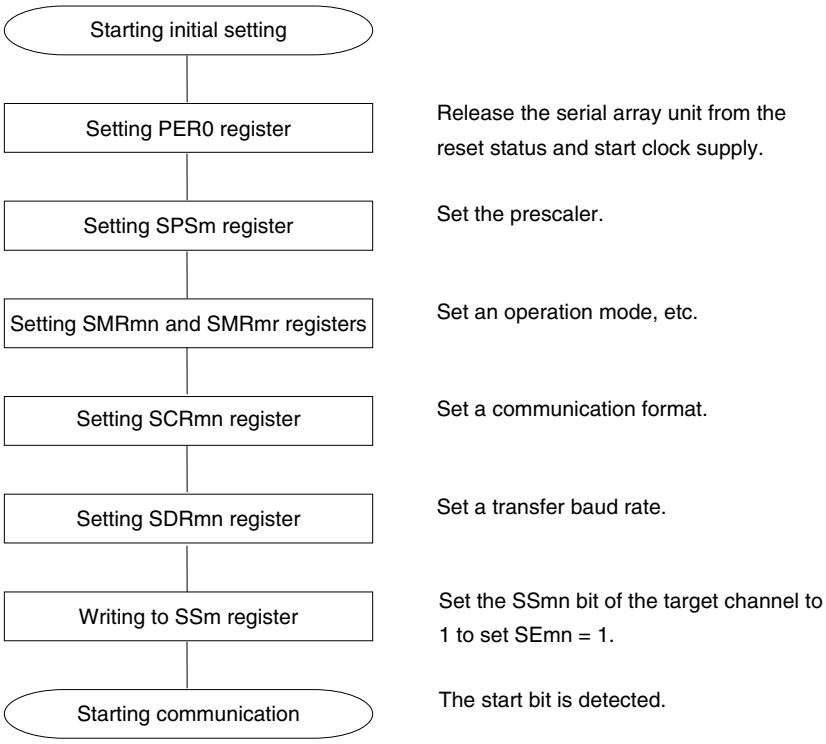


Caution For the UART reception, be sure to set SMRmr of channel r that is to be paired with channel n.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), r: Channel number (r = n − 1),
q: UART number (q = 0 to 3)
□ : Setting is fixed in the UART reception mode, ■ : Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

(2) Operation procedure

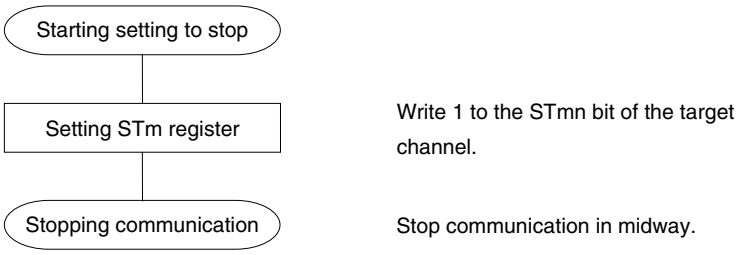
Figure 14-77. Initial Setting Procedure for UART Reception



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

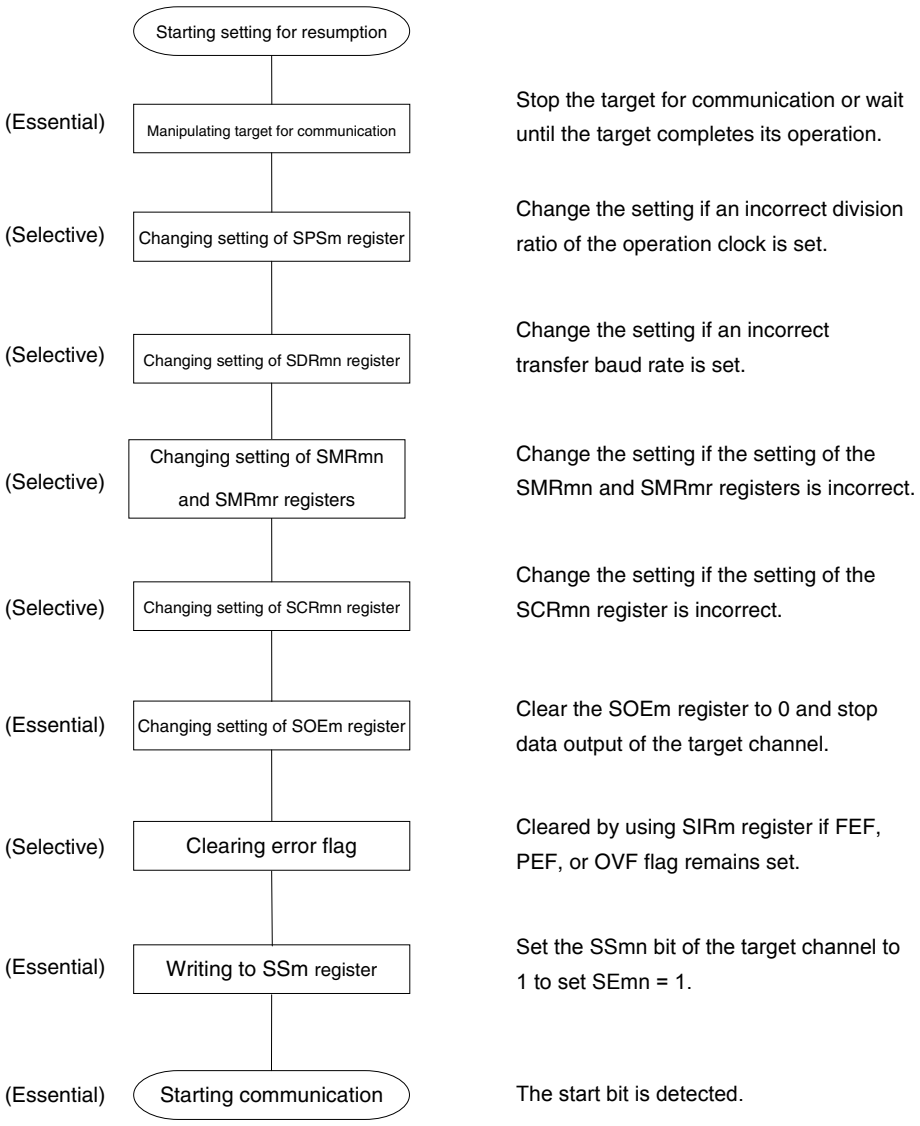
Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3)

Figure 14-78. Procedure for Stopping UART Reception



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

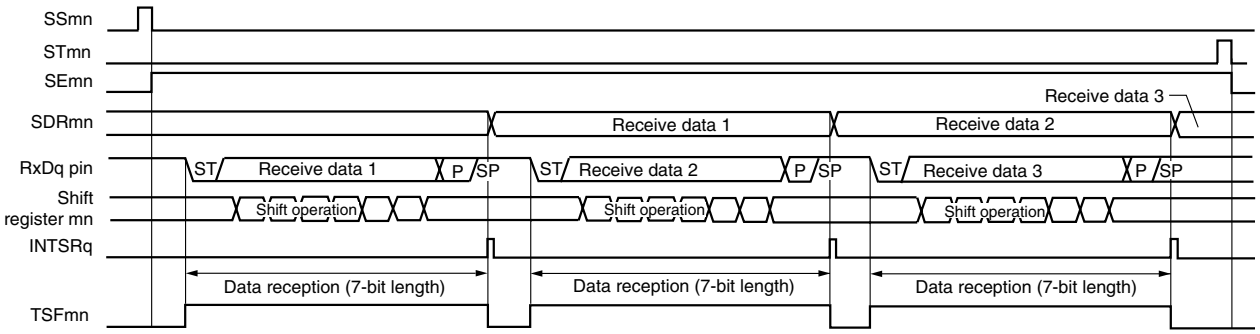
Figure 14-79. Procedure for Resuming UART Reception



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

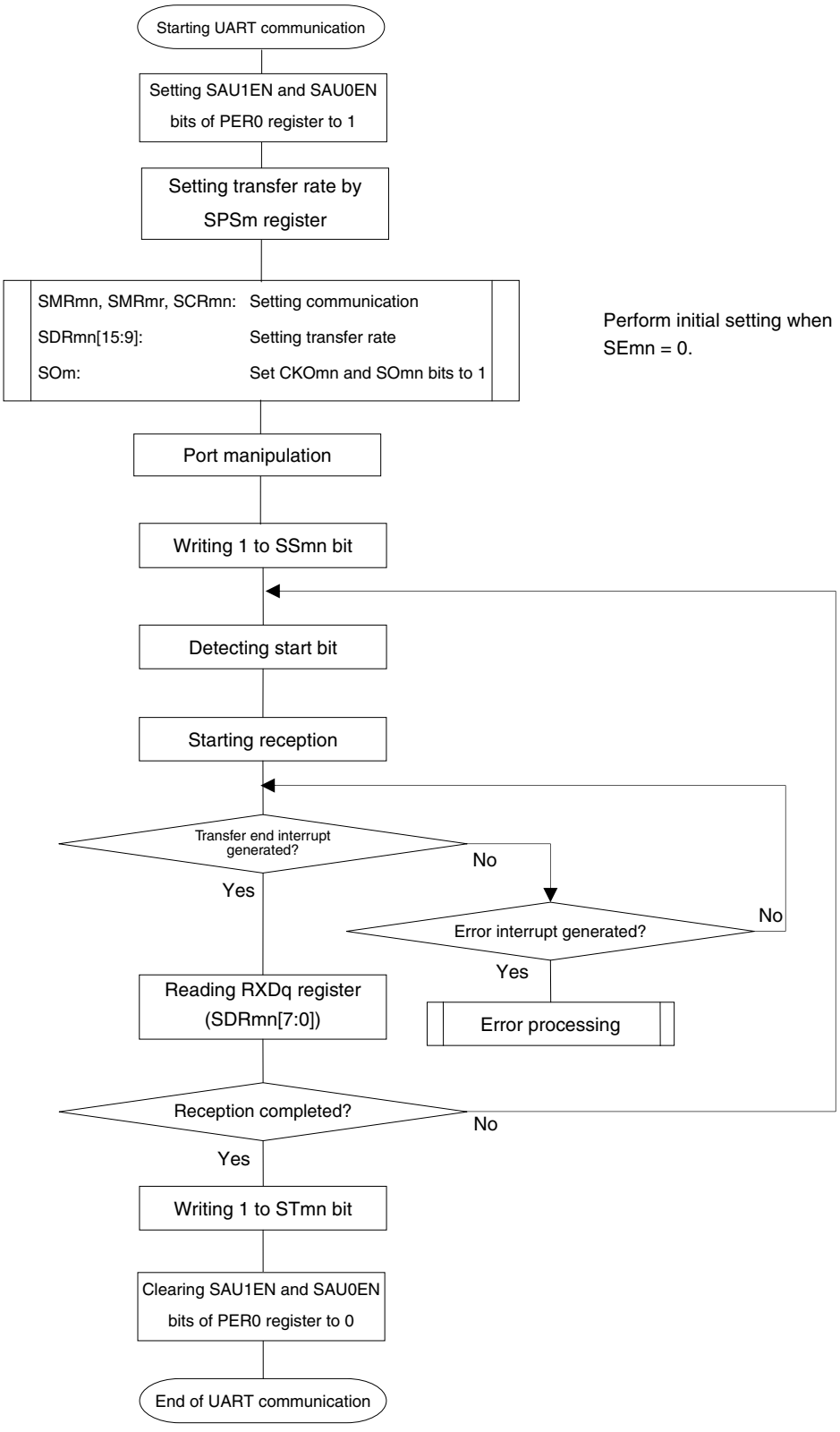
(3) Processing flow

Figure 14-80. Timing Chart of UART Reception



Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), q: UART number (q = 0 to 3)

Figure 14-81. Flowchart of UART Reception



Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 1, 3), q: UART number (q = 0 to 3)

14.6.3 LIN transmission

Of UART transmission, UART3 supports LIN communication.

For LIN transmission, channel 2 of unit 1 (SAU1) is used.

UART	UART0	UART1	UART2	UART3
Support of LIN communication	Not supported	Not supported	Not supported	Supported
Target channel	–	–	–	Channel 2 of SAU1
Pins used	–	–	–	TxD3
Interrupt	–	–	–	INTST3
	Transfer end interrupt (in single-transfer mode) or buffer empty interrupt (in continuous transfer mode) can be selected.			
Error detection flag	None			
Transfer data length	8 bits			
Transfer rate	Max. f_{MCK}/6 [bps] (SDRmn [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] Note			
Data phase	Forward 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 Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock (MCK) 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.

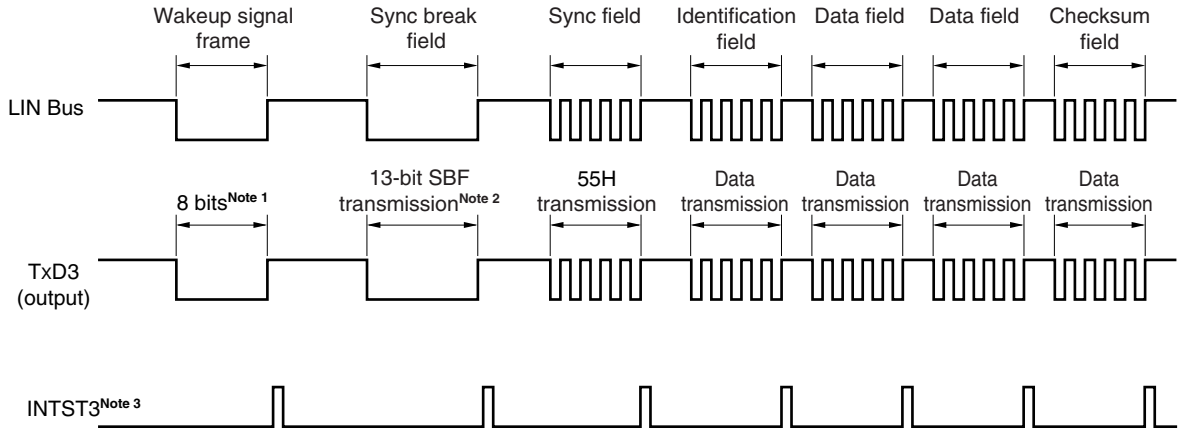
Usually, the master is connected to a network such as CAN (Controller Area Network).

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 14-82 outlines a transmission operation of LIN.

Figure 14-82. Transmission Operation of LIN



Notes 1. The baud rate is set so as to satisfy the standard of the wakeup signal and data of 00H is transmitted.

2. A sync 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 sync break field is calculated as follows.

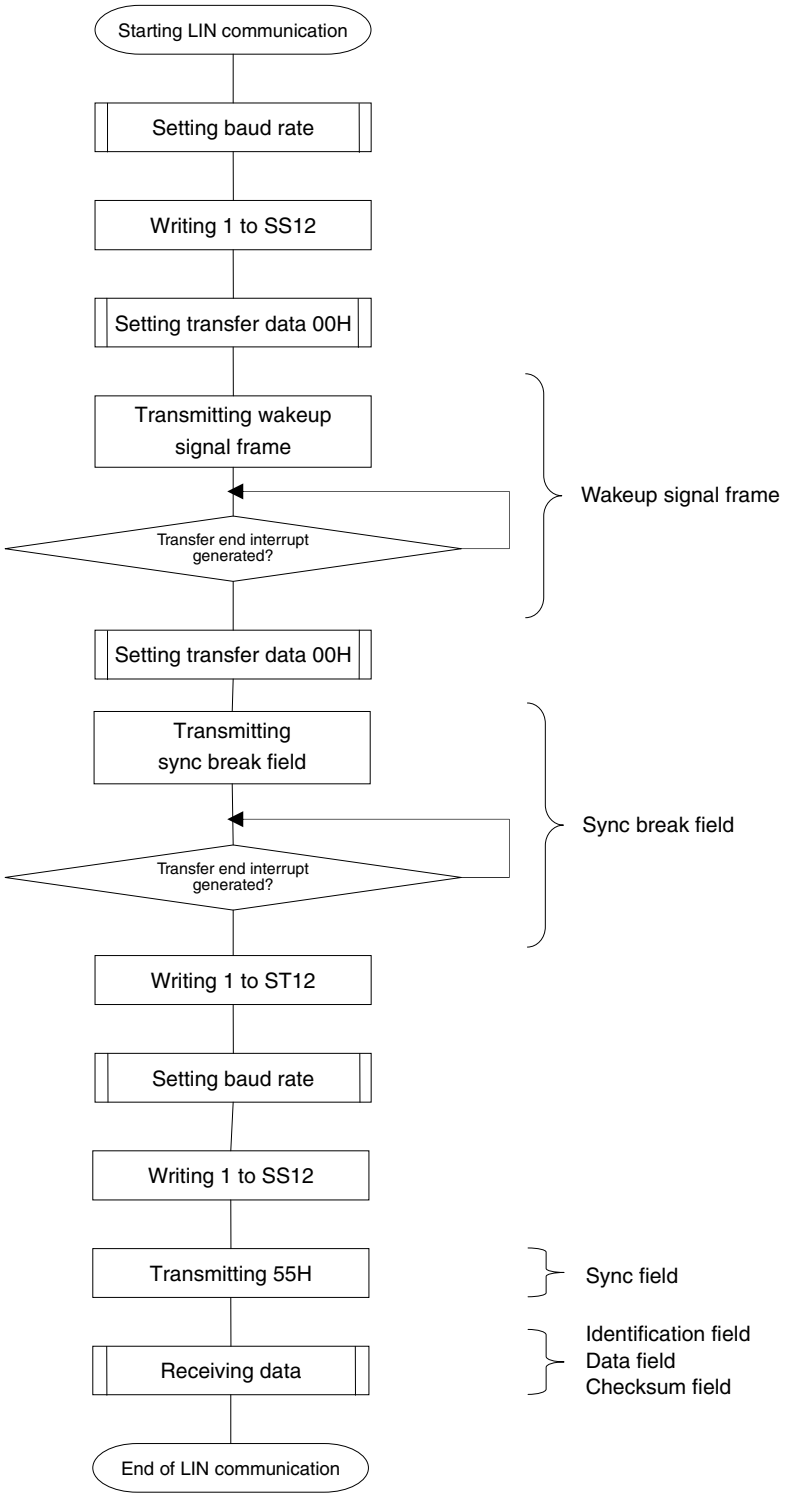
(Baud rate of sync break field) = 9/13 × N

By transmitting data of 00H at this baud rate, a sync break field is generated.

3. INTST3 is output upon completion of transmission. INTST3 is also output when SBF transmission is executed.

Remark The interval between fields is controlled by software.

Figure 14-83. Flowchart for LIN Transmission



14.6.4 LIN reception

Of UART reception, UART3 supports LIN communication.

For LIN reception, channel 3 of unit 1 (SAU1) is used.

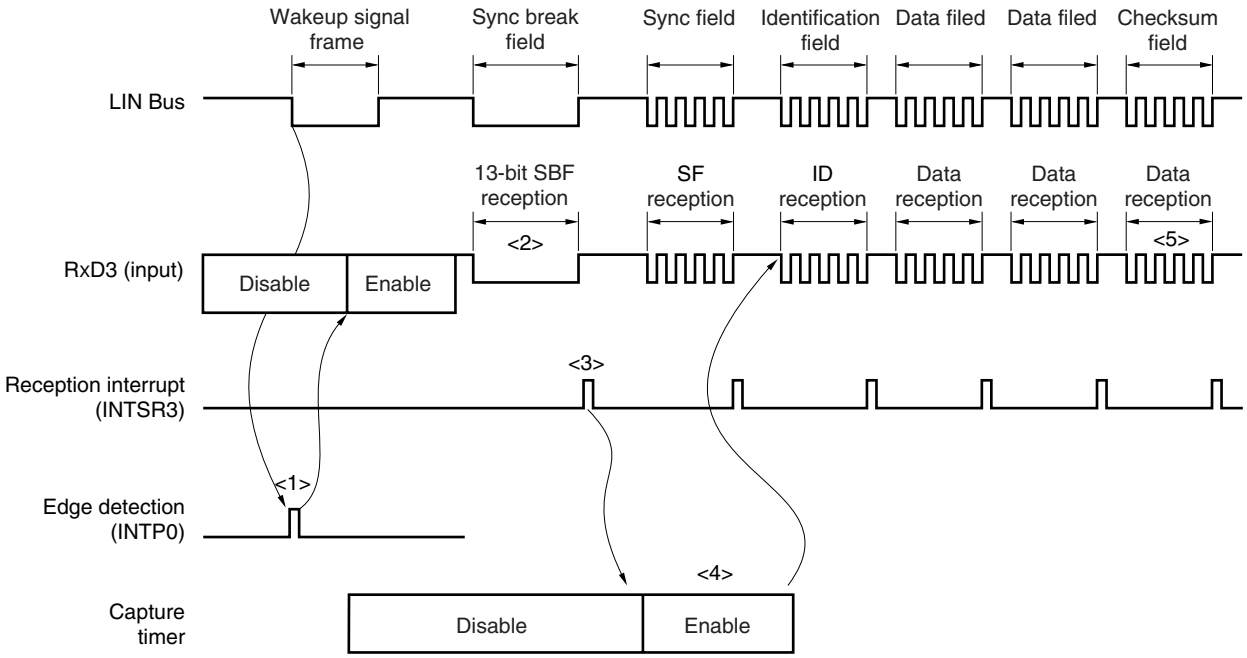
UART	UART0	UART1	UART2	UART3
Support of LIN communication	Not supported	Not supported	Not supported	Supported
Target channel	–	–	–	Channel 3 of SAU1
Pins used	–	–	–	RxD3
Interrupt	–	–	–	INTSR3
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)			
Error interrupt	–	–	–	INTSRE3
Error detection flag	• Framing error detection flag (FEF13) • Parity error detection flag (PEF13) • Overrun error detection flag (OVF13)			
Transfer data length	8 bits			
Transfer rate	Max. $f_{MCK}/6$ [bps] (SDRmn [15:9] = 2 or more), Min. $f_{CLK}/(2 \times 2^{11} \times 128)$ [bps] Note			
Data phase	Forward 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 Use this operation within a range that satisfies the conditions above and the AC characteristics in the electrical specifications (see **CHAPTER 29 ELECTRICAL SPECIFICATIONS**).

Remark f_{MCK}: Operation clock (MCK) frequency of target channel
f_{CLK}: System clock frequency

Figure 14-84 outlines a reception operation of LIN.

Figure 14-84. 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, enable reception of UART3 (RXE13 = 1) and wait for SBF reception.

<2> When the start bit of SBF is detected, reception is started and serial data is sequentially stored in the RXD3 register (= bits 7 to 0 of the serial data register 13 (SDR13)) at the set baud rate. When the stop bit is detected, the reception end interrupt request (INTSR3) is generated. When data of low levels of 11 bits or more is detected as SBF, it is judged that SBF reception has been correctly completed. If data of low levels of less than 11 bits is detected as SBF, it is judged that an SBF reception error has occurred, and the system returns to the SBF reception wait status.

<3> When SBF reception has been correctly completed, start channel 7 of the timer array unit and measure the bit interval (pulse width) of the sync field (see **6.7.5 Operation as input signal high-/low-level width measurement**).

<4> Calculate a baud rate error from the bit interval of sync field (SF). Stop UART3 once and adjust (re-set) the baud rate.

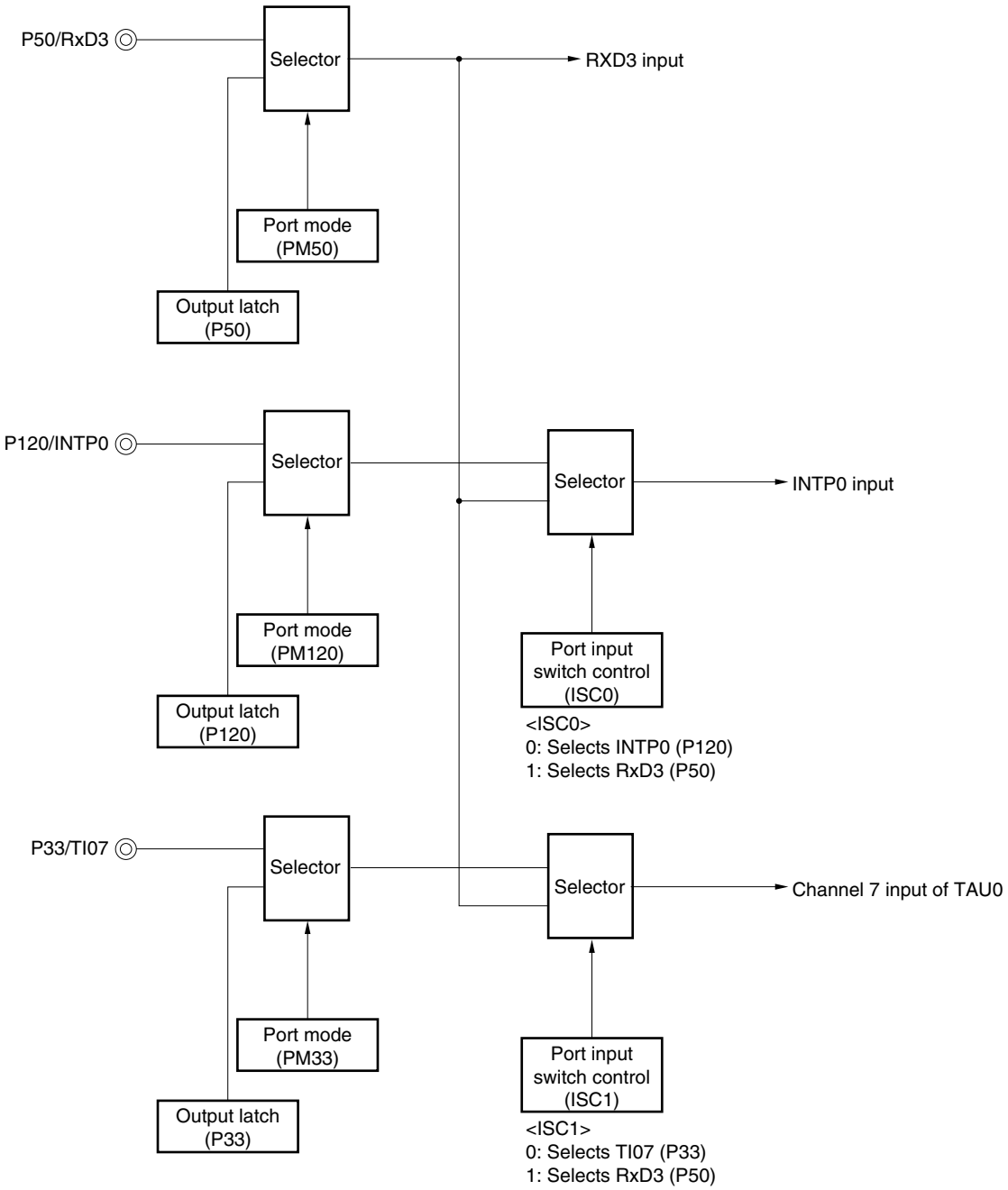
<5> The checksum field should be distinguished by software. In addition, processing to initialize UART3 after the checksum field is received and to wait for reception of SBF should also be performed by software.

Figure 14-85 shows the configuration of a port that manipulates reception of LIN.

The wakeup signal transmitted from the master of LIN is received by detecting an edge of an external interrupt (INTP0). The length of the sync field transmitted from the master can be measured by using the external event capture operation of the timer array unit 0 (TAU0) to calculate a baud-rate error.

By controlling switch of port input (ISC0/ISC1), the input source of port input (RxD3) for reception can be input to the external interrupt pin (INTP0) and timer array unit 0 (TAU0).

Figure 14-85. Port Configuration for Manipulating Reception of LIN



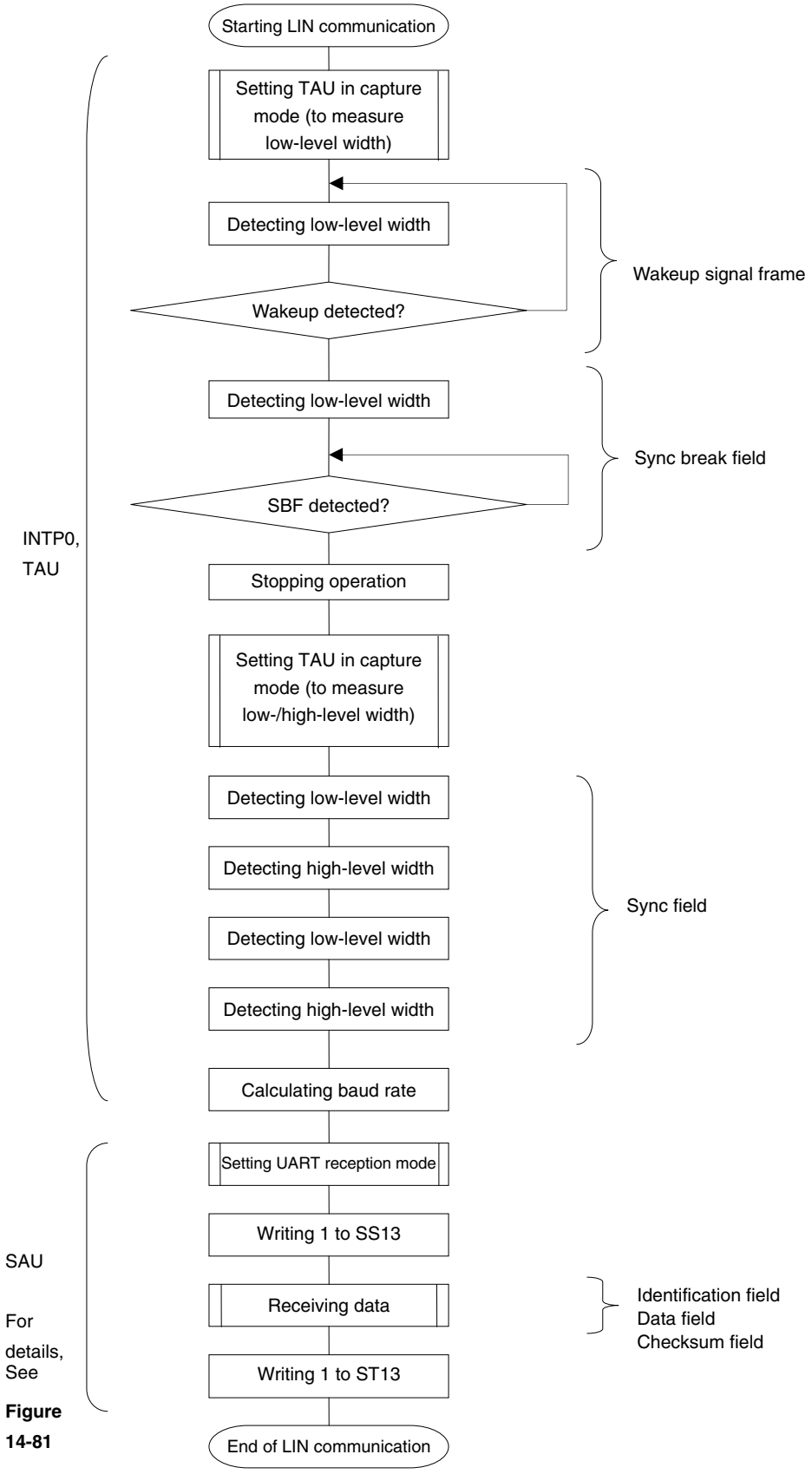
Remark ISC0, ISC1: Bits 0 and 1 of the input switch control register (ISC) (See **Figure 14-17**.)

The peripheral functions used for the LIN communication operation are as follows.

<Peripheral functions used>

- External interrupt (INTP0); Wakeup signal detection
  Usage: To detect an edge of the wakeup signal and the start of communication
- Channel 7 of timer array unit 0 (TAU0); Baud rate error 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 RxD3 is measured in the capture mode.)
- Channels 2 and 3 (UART3) of serial array unit 1 (SAU1)

Figure 14-86. Flowchart of LIN Reception



14.6.5 Calculating baud rate

(1) Baud rate calculation expression

The baud rate for UART (UART0, UART1, UART2, UART3) communication can be calculated by the following expressions.

(Baud rate) = {Operation clock (MCK) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2 [bps]

Caution Setting SDRmn [15:9] = (0000000B, 0000001B) is prohibited.

Remarks 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.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

The operation clock (MCK) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 14-3. Selection of operation clock

SMRmn Register	SPSm Register								Operation Clock (MCK) Note1	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	20 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	10 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	5 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	2.5 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.25 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	625 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	313 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	156 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	78.1 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	39.1 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	19.5 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	9.77 kHz
	X	X	X	X	1	1	1	1	INTTM02 if m = 0, INTTM03 if m = 1 Note2	
1	0	0	0	0	X	X	X	X	f_{CLK}	20 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	10 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	5 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	2.5 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.25 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	625 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	313 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	156 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	78.1 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	39.1 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	19.5 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	9.77 kHz
	1	1	1	1	X	X	X	X	INTTM02 if m = 0, INTTM03 if m = 1 Note2	
Other than above									Setting prohibited	

Notes 1. When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (STm = 000FH) the operation of the serial array unit (SAUm). When selecting INTTM02 and INTTM03 for the operation clock, also stop the timer array unit (TAU0) (TT0 = 00FFH).

2. SAUm can be operated at a fixed division ratio of the subsystem clock, regardless of the f_{CLK} frequency (main system clock, subsystem clock), by operating the interval timer for which $f_{SUB}/4$ has been selected as the count clock (setting TIS02 (if m = 0) or TIS03 (if m = 1) of the TIS0 register to 1) and selecting INTTM02 and INTTM03 by using the SPSm register in channels 2 and 3 of TAU0. When changing f_{CLK}, however, SAUm and TAU0 must be stopped as described in Note 1 above.

Remarks 1. X: Don't care

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

(2) Baud rate error during transmission

The baud rate error of UART (UART0, UART1, UART2, UART3) communication during transmission can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

(Baud rate error) = (Calculated baud rate value) ÷ (Target baud rate) × 100 − 100 [%]

Here is an example of setting a UART baud rate at f_{CLK} = 20 MHz.

UART Baud Rate (Target Baud Rate)	f_{CLK} = 20 MHz			
	Operation Clock (MCK)	SDRmn[15:9]	Calculated Baud Rate	Error from Target Baud Rate
300 bps	$f_{CLK}/2^9$	64	300.48 bps	+0.16 %
600 bps	$f_{CLK}/2^8$	64	600.96 bps	+0.16 %
1200 bps	$f_{CLK}/2^7$	64	1201.92 bps	+0.16 %
2400 bps	$f_{CLK}/2^6$	64	2403.85 bps	+0.16 %
4800 bps	$f_{CLK}/2^5$	64	4807.69 bps	+0.16 %
9600 bps	$f_{CLK}/2^4$	64	9615.38 bps	+0.16 %
19200 bps	$f_{CLK}/2^3$	64	19230.8 bps	+0.16 %
31250 bps	$f_{CLK}/2^3$	39	31250.0 bps	±0.0 %
38400 bps	$f_{CLK}/2^2$	64	38461.5 bps	+0.16 %
76800 bps	$f_{CLK}/2$	64	76923.1 bps	+0.16 %
153600 bps	f_{CLK}	64	153846 bps	+0.16 %
312500 bps	f_{CLK}	31	312500 bps	±0.0 %

(3) Permissible baud rate range for reception

The permissible baud rate range for reception during UART (UART0, UART1, UART2, UART3) communication can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

$$\text{(Maximum receivable baud rate)} = \frac{2 \times k \times Nfr}{2 \times k \times Nfr - k + 2} \times Brate$$

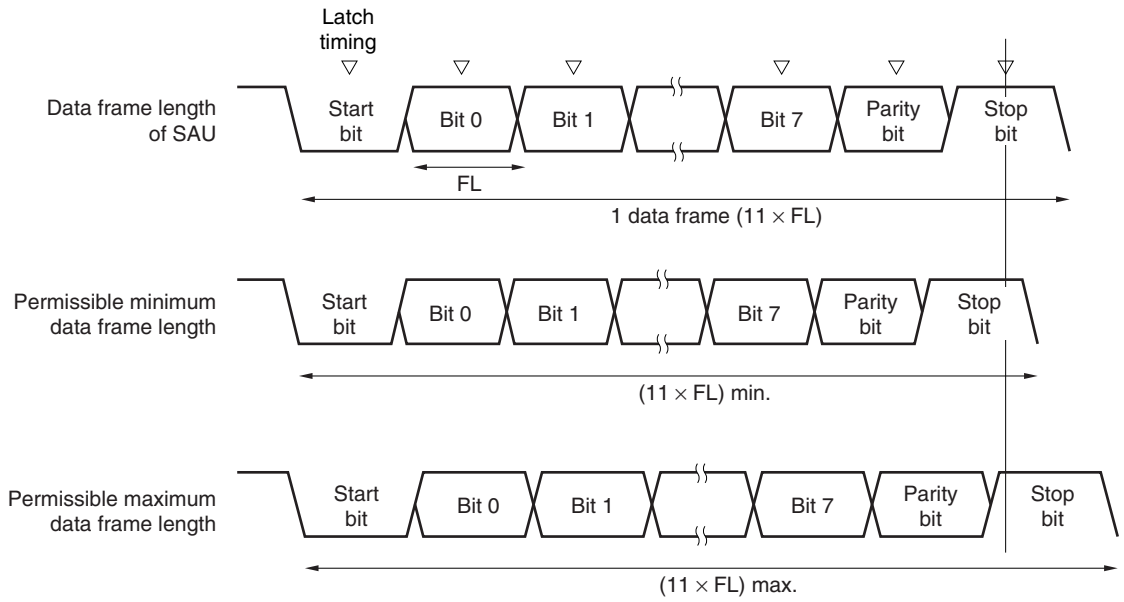
$$\text{(Minimum receivable baud rate)} = \frac{2 \times k \times (Nfr - 1)}{2 \times k \times Nfr - k - 2} \times Brate$$

Brate: Calculated baud rate value at the reception side (See **14.6.5 (1) Baud rate calculation expression**.)

k: SDRmn[15:9] + 1

Nfr: 1 data frame length [bits]
= (Start bit) + (Data length) + (Parity bit) + (Stop bit)

Figure 14-87. Permissible Baud Rate Range for Reception (1 Data Frame Length = 11 Bits)



As shown in Figure 14-87, the timing of latching receive data is determined by the division ratio set by bits 15 to 9 of the 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.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

14.7 Operation of Simplified I²C (IIC10, IIC20) Communication

This is a clocked communication function to communicate with two or more devices by using two lines: serial clock (SCL) and serial data (SDA). This 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 and does not have a wait detection function. 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.)
- Manual generation of start condition and stop condition

[Interrupt function]

- Transfer end interrupt

[Error detection flag]

- Parity error (ACK error)

* [Functions not supported by simplified I²C]

- Slave transmission, slave reception
- Arbitration loss detection function
- Wait detection function

Note An ACK is not output when the last data is being received by writing 0 to the SOEmn (SOEm register) bit and stopping the output of serial communication data. See **14.7.3 (2) Processing flow** for details.

Remarks 1. To use the full-function I²C bus, see **CHAPTER 15 SERIAL INTERFACE IICA**.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

The channels supporting simplified I²C (IIC10, IIC20) are channel 2 of SAU0 and channel 0 of SAU1.

Unit	Channel	Used as CSI	Used as UART	Used as Simplified I²C
0	0	CSI00	UART0	–
	1	–		–
	2	CSI10	UART1	IIC10
	3	–		–
1	0	CSI20	UART2	IIC20
	1	–		–
	2	–	UART3 (supporting LIN-bus)	–
	3	–		–

Simplified I²C (IIC10, IIC20) performs the following four types of communication operations.

- Address field transmission (See **14.7.1**.)
- Data transmission (See **14.7.2**.)
- Data reception (See **14.7.3**.)
- Stop condition generation (See **14.7.4**.)

14.7.1 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	IIC10	IIC20
Target channel	Channel 2 of SAU0	Channel 0 of SAU1
Pins used	SCL10, SDA10 Note	SCL20, SDA20 Note
Interrupt	INTIIC10	INTIIC20
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	Parity 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	Max. f_{CLK}/4 [MHz] (SDRmn [15:9] = 1 or more) f_{CLK}: System clock frequency However, the following condition must be satisfied in each mode of I²C. • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)	
Data level	Forward output (default: high level)	
Parity bit	No parity bit	
Stop bit	Appending 1 bit (for ACK reception timing)	
Data direction	MSB first	

Note To perform communication via simplified I²C, set the data I/O pins (SDA10, SDA20) in the N-ch open-drain output (V_{DD} tolerance) mode (POM14 = 1, POM11 = 1) by using the port output mode register 1 (POM1) (see **4.3 Registers Controlling Port Function** for details). When communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POM15 = 1, POM10 = 1) also for the clock input/output pins (SCL10, SCL20) (see **4.4.4 Connecting to external device with different potential (2.5 V, 3 V)** for details).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

(1) Register setting

Figure 14-88. Example of Contents of Registers for Address Field Transmission of Simplified I^2C (IIC10, IIC20)

(a) Serial output register m (SOm) … Sets only the bits of the target channel.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOm	0	0	0	0	1	CKOm2 0/1	1	CKOm0 0/1	0	0	0	0	1	SOm2 0/1	1	SOm0 0/1

Start condition is generated by manipulating the SOmn bit.

(b) Serial output enable register m (SOEm) … Sets only the bits of the target 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	0	SOEm0 0/1

SOEmn = 0 until the start condition is generated, and SOEmn = 1 after generation.

(c) Serial channel start register m (SSm) … Sets only the bits of the target channel is 1.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSm	0	0	0	0	0	0	0	0	0	0	0	0	SSm3 ×	SSm2 0/1	SSm1 ×	SSm0 0/1

(d) 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 1	MDmn1 0	MDmn0 0

Operation mode of channel n
0: Transfer end interrupt

(e) 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	SLCmn0 1	0	DLSmn2 1	DLSmn1 1	DLSmn0 1

Setting of parity bit
00B: No parity

Setting of stop bit
01B: Appending 1 bit (ACK)

(f) Serial data register mn (SDRmn) (lower 8 bits: SIOr)

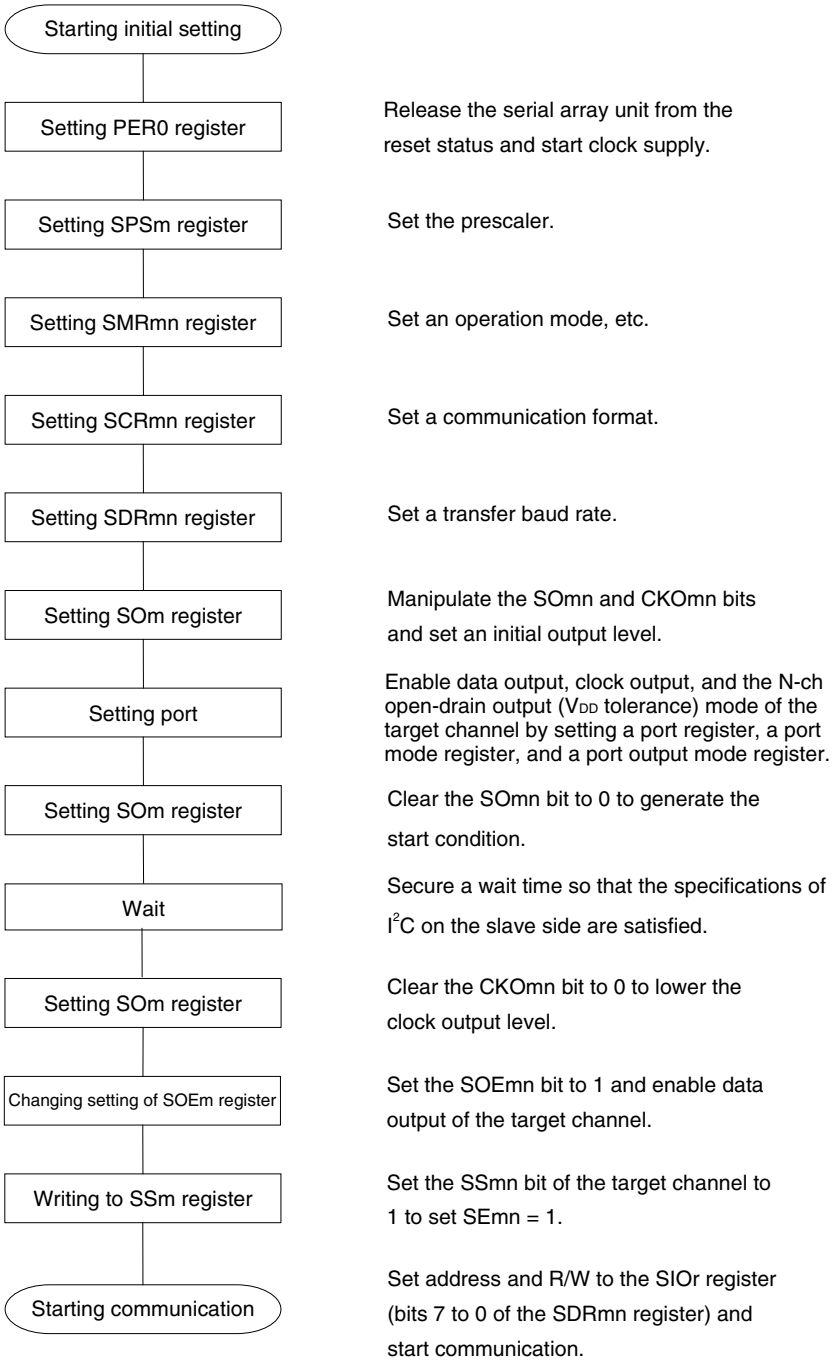
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	Transmit data setting (address + R/W)							

SIOr

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), r: IIC number (r = 10, 20)
□: 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 14-89. Initial Setting Procedure for Address Field Transmission

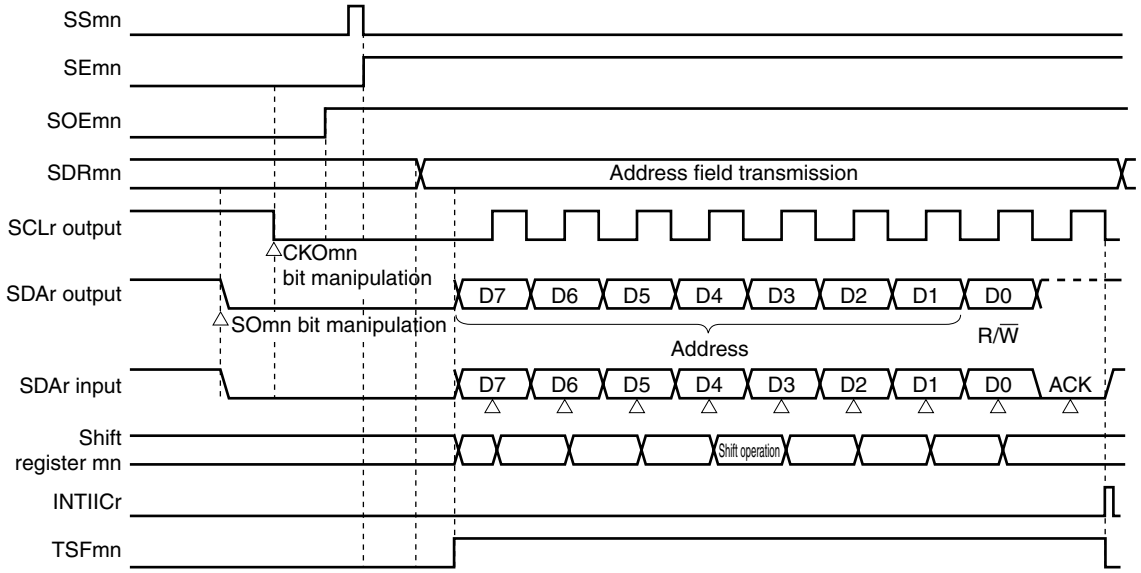


Caution After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), r: IIC number (r = 10, 20)

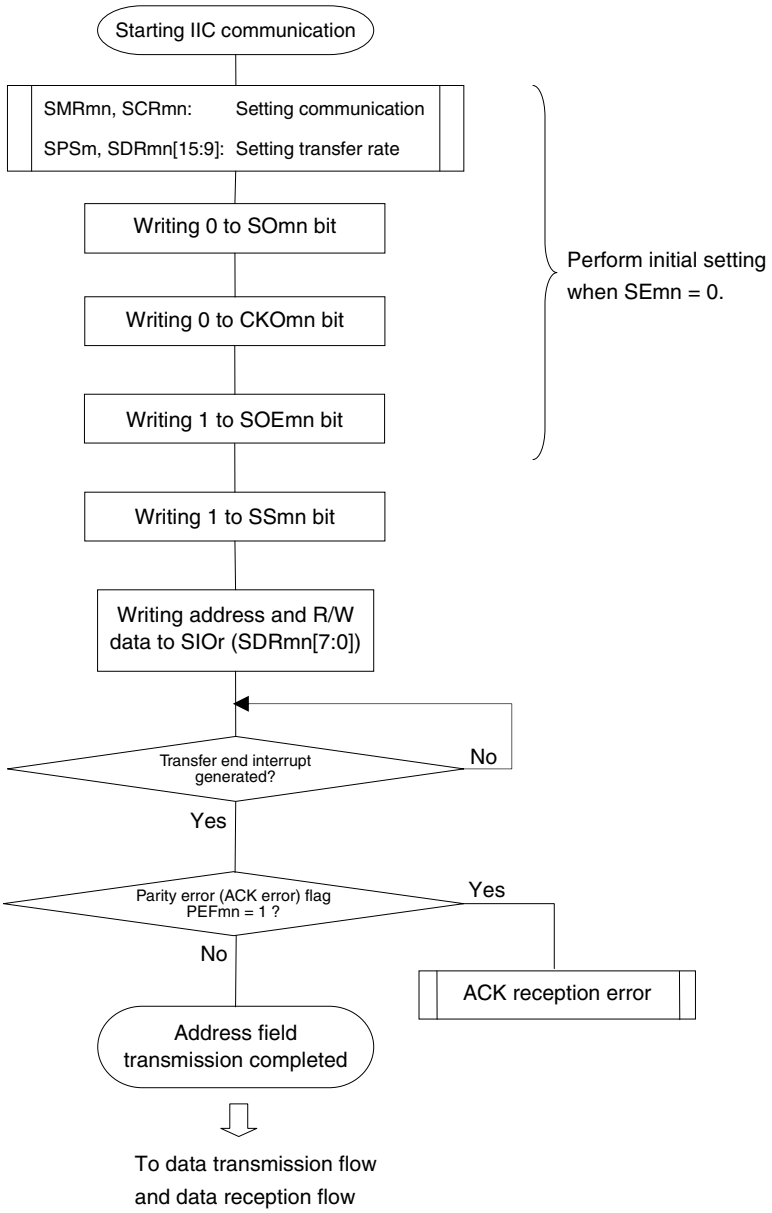
(3) Processing flow

Figure 14-90. Timing Chart of Address Field Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), r: IIC number (r = 10, 20)

Figure 14-91. Flowchart of Address Field Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), r: IIC number (r = 10, 20)

14.7.2 Data transmission

Data transmission is an operation to transmit data to the target for transfer (slave) after transmission of an address field. After all data are transmitted to the slave, a stop condition is generated and the bus is released.

Simplified I^2C	IIC10	IIC20
Target channel	Channel 2 of SAU0	Channel 0 of SAU1
Pins used	SCL10, SDA10 Note	SCL20, SDA20 Note
Interrupt	INTIIC10	INTIIC20
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	Parity error detection flag (PEFmn)	
Transfer data length	8 bits	
Transfer rate	Max. f_{CLK}/4 [MHz] (SDRmn [15:9] = 1 or more) f_{CLK}: System clock frequency However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)	
Data level	Forward output (default: high level)	
Parity bit	No parity bit	
Stop bit	Appending 1 bit (for ACK reception timing)	
Data direction	MSB first	

Note To perform communication via simplified I^2C, set the data I/O pins (SDA10, SDA20) in the N-ch open-drain output (V_{DD} tolerance) mode (POM14 = 1, POM11 = 1) by using the port output mode register 1 (POM1) (see **4.3 Registers Controlling Port Function** for details). When communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POM15 = 1, POM10 = 1) also for the clock input/output pins (SCL10, SCL20) (see **4.4.4 Connecting to external device with different potential (2.5 V, 3 V)** for details).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

(1) Register setting

Figure 14-92. Example of Contents of Registers for Data Transmission of Simplified I²C (IIC10, IIC20)

(a) 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	1	CKOm2 0/1Note	1	CKOm0 0/1Note	0	0	0	0	1	SOm2 0/1Note	1	SOm0 0/1Note

(b) 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 0/1	0	SOEm0 0/1

(c) 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 0/1	SSm1 ×	SSm0 0/1

(d) 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	0	SISmn0 0	1	0	0	MDmn2 1	MDmn1 0	MDmn0 0

(e) 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	SLCmn0 1	0	DLSmn2 1	DLSmn1 1	DLSmn0 1

(f) Serial data register mn (SDRmn) (lower 8 bits: SIOr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	Transmit data setting							

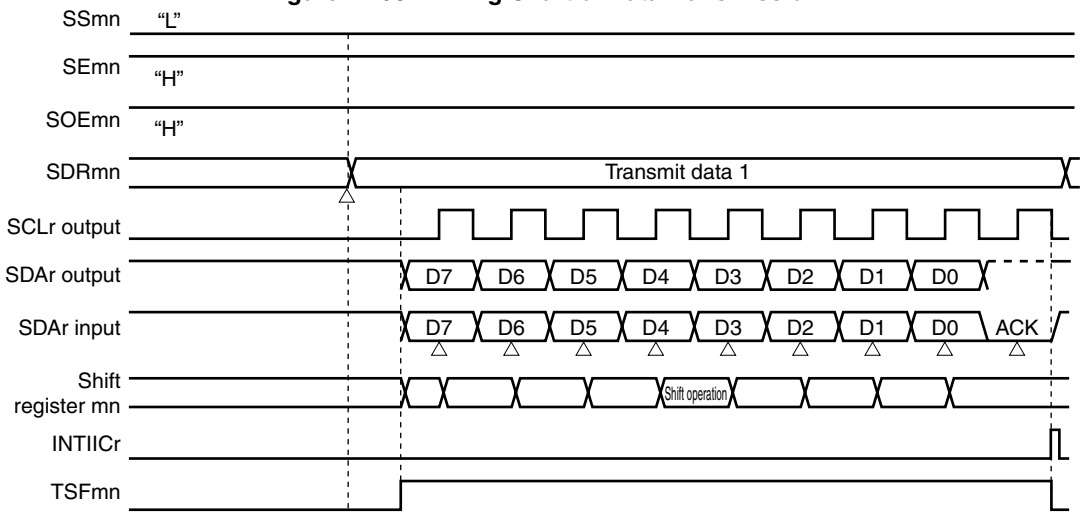
SIOr

Note The value varies depending on the communication data during communication operation.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), r: IIC number (r = 10, 20)
□: Setting is fixed in the IIC mode, ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

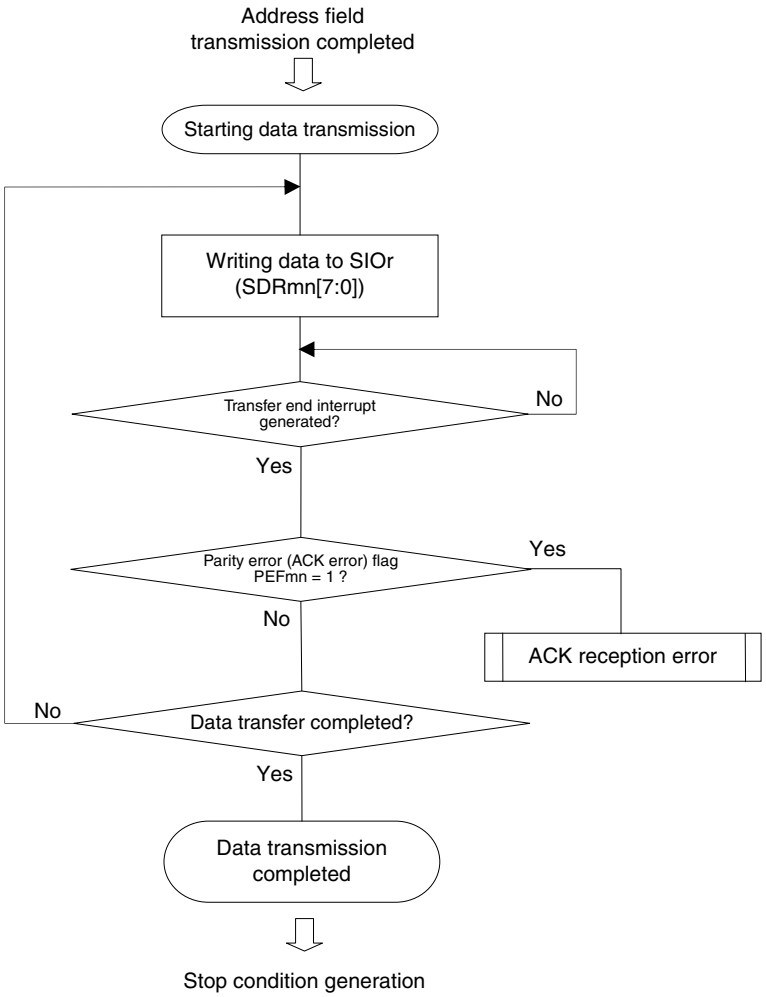
(2) Processing flow

Figure 14-93. Timing Chart of Data Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), r: IIC number (r = 10, 20)

Figure 14-94. Flowchart of Data Transmission



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), r: IIC number (r = 10, 20)

14.7.3 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	IIC10	IIC20
Target channel	Channel 2 of SAU0	Channel 0 of SAU1
Pins used	SCL10, SDA10 Note	SCL20, SDA20 Note
Interrupt	INTIIC10	INTIIC20
	Transfer end interrupt only (Setting the buffer empty interrupt is prohibited.)	
Error detection flag	None	
Transfer data length	8 bits	
Transfer rate	Max. f_{CLK}/4 [MHz] (SDRmn [15:9] = 1 or more) f_{CLK}: System clock frequency However, the following condition must be satisfied in each mode of I^2C. • Max. 400 kHz (fast mode) • Max. 100 kHz (standard mode)	
Data level	Forward output (default: high level)	
Parity bit	No parity bit	
Stop bit	Appending 1 bit (ACK transmission)	
Data direction	MSB first	

Note To perform communication via simplified I^2C, set the data I/O pins (SDA10, SDA20) in the N-ch open-drain output (V_{DD} tolerance) mode (POM14 = 1, POM11 = 1) by using the port output mode register 1 (POM1) (see **4.3 Registers Controlling Port Function** for details). When communicating with an external device with a different potential, set the N-ch open-drain output (V_{DD} tolerance) mode (POM15 = 1, POM10 = 1) also for the clock input/output pins (SCL10, SCL20) (see **4.4.4 Connecting to external device with different potential (2.5 V, 3 V)** for details).

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

(1) Register setting

Figure 14-95. Example of Contents of Registers for Data Reception of Simplified I^2C (IIC10, IIC20)

(a) 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	1	CKOm2 0/1 Note	1	CKOm0 0/1 Note	0	0	0	0	1	SOm2 0/1 Note	1	SOm0 0/1 Note

(b) 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 0/1	0	SOEm0 0/1

(c) 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 0/1	SSm1 ×	SSm0 0/1

(d) 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	0	SISmn0 0	1	0	0	MDmn2 1	MDmn1 0	MDmn0 0

(e) 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	SLCmn0 1	0	DLSmn2 1	DLSmn1 1	DLSmn0 1

(f) Serial data register mn (SDRmn) (lower 8 bits: SIOr)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRmn	Baud rate setting							0	Dummy transmit data setting (FFH)							

SIOr

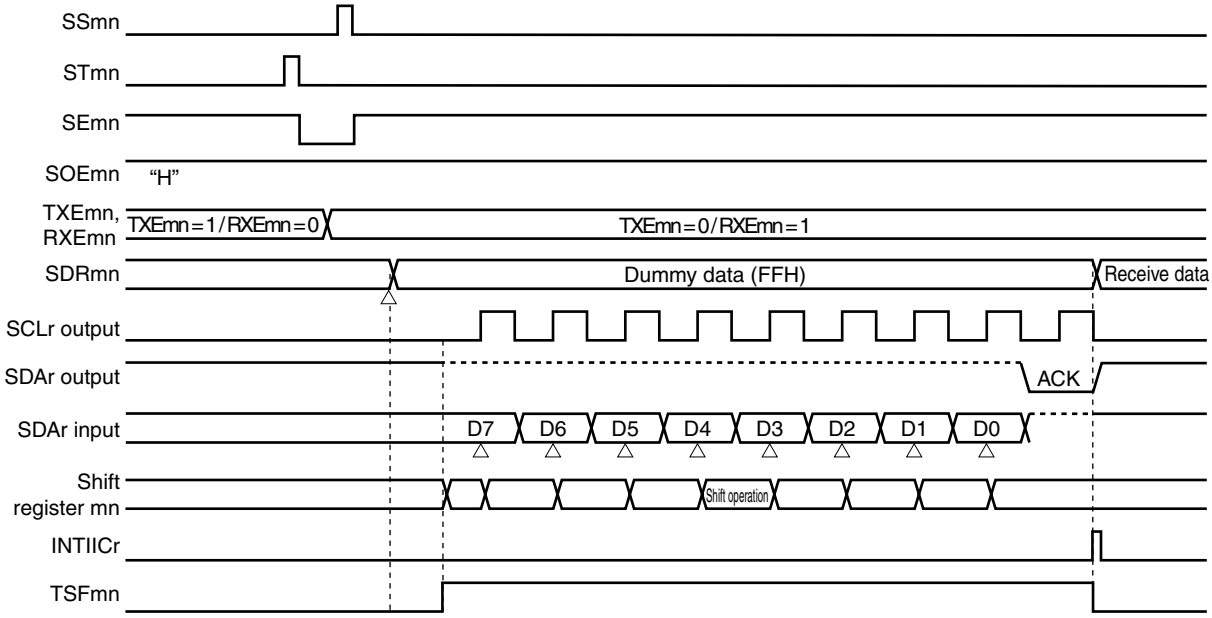
Note The value varies depending on the communication data during communication operation.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), r: IIC number (r = 10, 20)
□: Setting is fixed in the IIC mode, ■: Setting disabled (set to the initial value)
×: Bit that cannot be used in this mode (set to the initial value when not used in any mode)
0/1: Set to 0 or 1 depending on the usage of the user

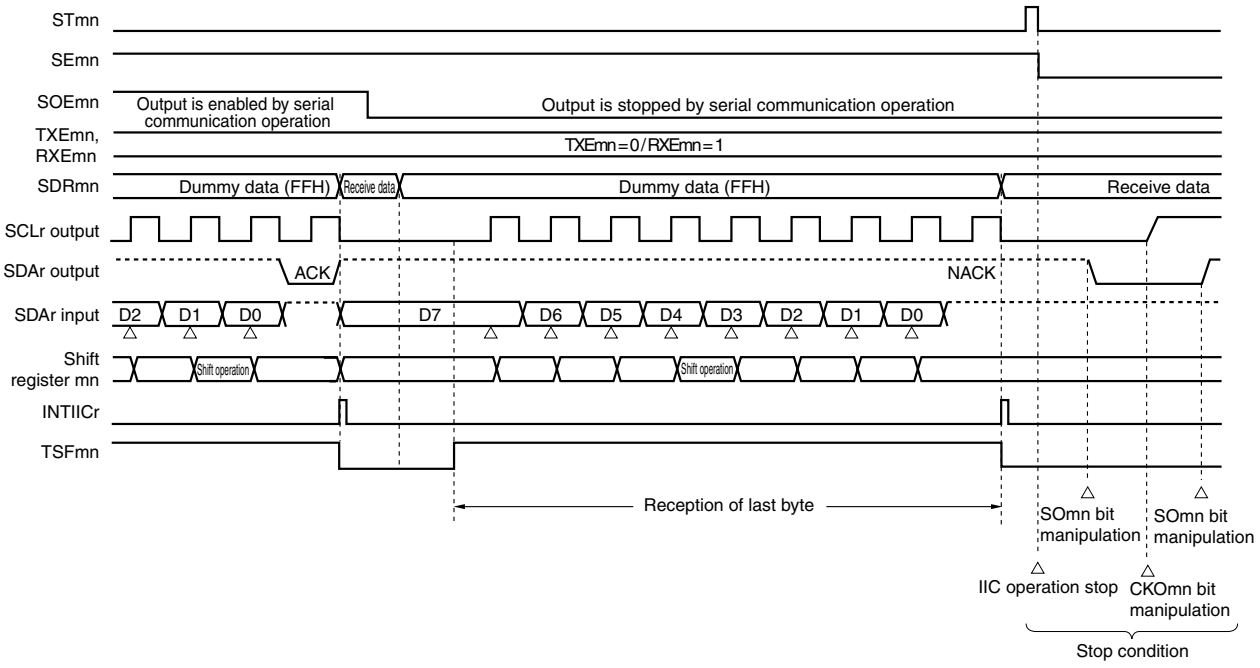
(2) Processing flow

Figure 14-96. Timing Chart of Data Reception

(a) When starting data reception

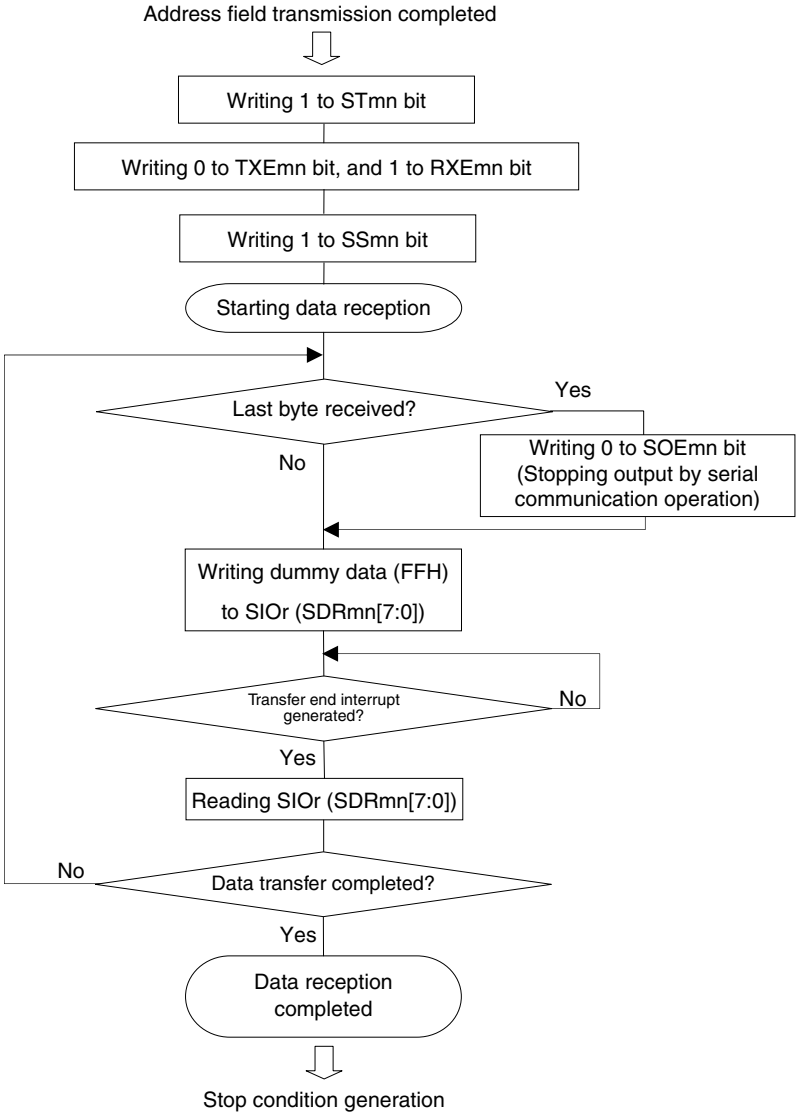


(b) When receiving last data



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), r: IIC number (r = 10, 20)

Figure 14-97. Flowchart of Data Reception



Caution ACK is also output when the last data is received. Communication is then completed by setting "1" to the STmn bit to stop operation and generating a stop condition.

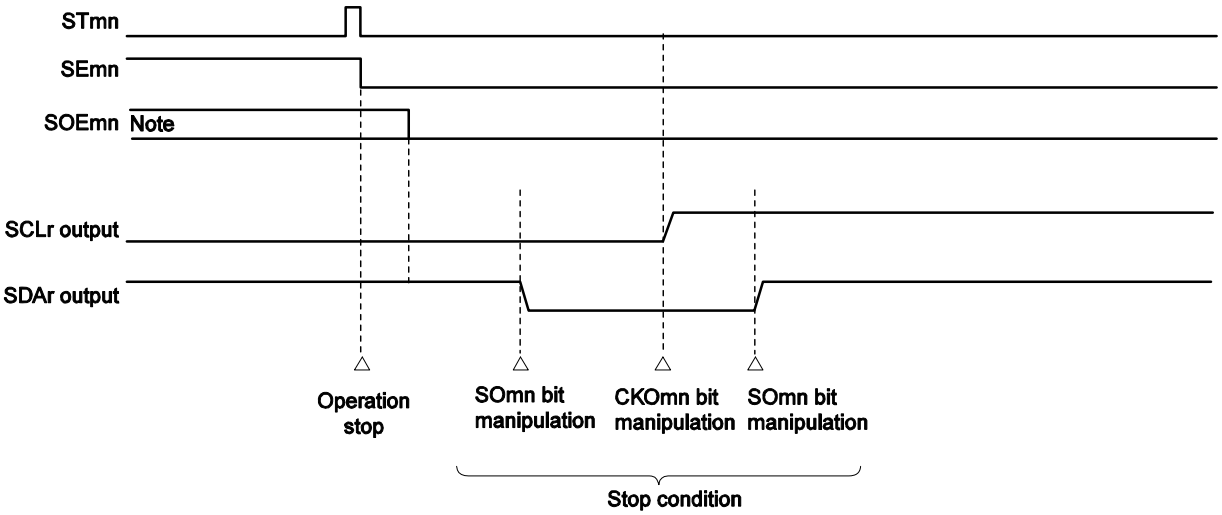
Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), r: IIC number (r = 10, 20)

14.7.4 Stop condition generation

After all data are transmitted to or received from the target slave, a stop condition is generated and the bus is released.

(1) Processing flow

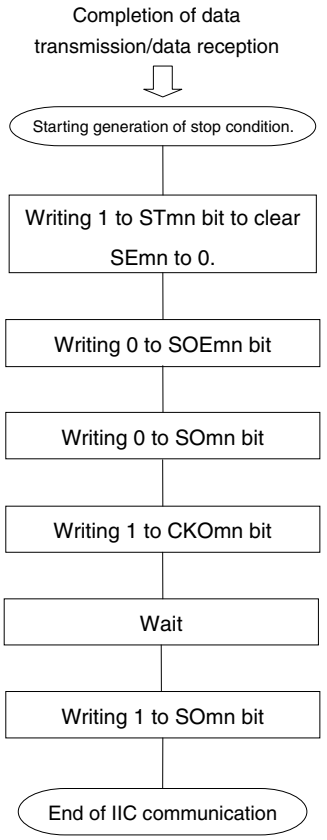
Figure 14-98. Timing Chart of Stop Condition Generation



Note During the receive operation, the SOEmn bit is set to 0 before receiving the last data.

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), r: IIC number (r = 10, 20)

Figure 14-99. Flowchart of Stop Condition Generation



Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

14.7.5 Calculating transfer rate

The transfer rate for simplified I^2C (IIC10, IIC20) communication can be calculated by the following expressions.

(Transfer rate) = {Operation clock (MCK) frequency of target channel} ÷ (SDRmn[15:9] + 1) ÷ 2

<R> **Caution Setting SDRmn [15:9] = 0000000B is prohibited. Set SDRmn[15:9] to 0000001B or greater.**

Remarks 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.

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

The operation clock (MCK) is determined by serial clock select register m (SPSm) and bit 15 (CKSmn) of serial mode register mn (SMRmn).

Table 14-4. Selection of operation clock

SMRmn Register	SPSm Register								Operation Clock (MCK) [Note1]	
CKSmn	PRS m13	PRS m12	PRS m11	PRS m10	PRS m03	PRS m02	PRS m01	PRS m00		f_{CLK} = 20 MHz
0	X	X	X	X	0	0	0	0	f_{CLK}	20 MHz
	X	X	X	X	0	0	0	1	$f_{CLK}/2$	10 MHz
	X	X	X	X	0	0	1	0	$f_{CLK}/2^2$	5 MHz
	X	X	X	X	0	0	1	1	$f_{CLK}/2^3$	2.5 MHz
	X	X	X	X	0	1	0	0	$f_{CLK}/2^4$	1.25 MHz
	X	X	X	X	0	1	0	1	$f_{CLK}/2^5$	625 kHz
	X	X	X	X	0	1	1	0	$f_{CLK}/2^6$	313 kHz
	X	X	X	X	0	1	1	1	$f_{CLK}/2^7$	156 kHz
	X	X	X	X	1	0	0	0	$f_{CLK}/2^8$	78.1 kHz
	X	X	X	X	1	0	0	1	$f_{CLK}/2^9$	39.1 kHz
	X	X	X	X	1	0	1	0	$f_{CLK}/2^{10}$	19.5 kHz
	X	X	X	X	1	0	1	1	$f_{CLK}/2^{11}$	9.77 kHz
	X	X	X	X	1	1	1	1	INTTM02 if m = 0, INTTM03 if m = 1 [Note2]	
1	0	0	0	0	X	X	X	X	f_{CLK}	20 MHz
	0	0	0	1	X	X	X	X	$f_{CLK}/2$	10 MHz
	0	0	1	0	X	X	X	X	$f_{CLK}/2^2$	5 MHz
	0	0	1	1	X	X	X	X	$f_{CLK}/2^3$	2.5 MHz
	0	1	0	0	X	X	X	X	$f_{CLK}/2^4$	1.25 MHz
	0	1	0	1	X	X	X	X	$f_{CLK}/2^5$	625 kHz
	0	1	1	0	X	X	X	X	$f_{CLK}/2^6$	313 kHz
	0	1	1	1	X	X	X	X	$f_{CLK}/2^7$	156 kHz
	1	0	0	0	X	X	X	X	$f_{CLK}/2^8$	78.1 kHz
	1	0	0	1	X	X	X	X	$f_{CLK}/2^9$	39.1 kHz
	1	0	1	0	X	X	X	X	$f_{CLK}/2^{10}$	19.5 kHz
	1	0	1	1	X	X	X	X	$f_{CLK}/2^{11}$	9.77 kHz
	1	1	1	1	X	X	X	X	INTTM02 if m = 0, INTTM03 if m = 1 [Note2]	
Other than above									Setting prohibited	

Notes 1. When changing the clock selected for f_{CLK} (by changing the system clock control register (CKC) value), do so after having stopped (STm = 000FH) the operation of the serial array unit (SAUm). When selecting INTTM02 and INTTM03 for the operation clock, also stop the timer array unit (TAU0) (TT0 = 00FFH).

2. SAUm can be operated at a fixed division ratio of the subsystem clock, regardless of the f_{CLK} frequency (main system clock, subsystem clock), by operating the interval timer for which $f_{SUB}/4$ has been selected as the count clock (setting TIS02 (if m = 0) or TIS03 (if m = 1) of the TIS0 register to 1) and selecting INTTM02 and INTTM03 by using the SPSm register in channels 2 and 3 of TAU0. When changing f_{CLK}, however, SAUm and TAU0 must be stopped as described in Note 1 above.

Remarks 1. X: Don't care

2. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 2)

Here is an example of setting an IIC transfer rate where MCK = f_{CLK} = 20 MHz.

IIC Transfer Mode (Desired Transfer Rate)	f_{CLK} = 20 MHz			
	Operation Clock (MCK)	SDRmn[15:9]	Calculated Transfer Rate	Error from Desired Transfer Rate
100 kHz	f_{CLK}	99	100 kHz	0.0%
400 kHz	f_{CLK}	24	400 kHz	0.0%

14.8 Processing Procedure in Case of Error

The processing procedure to be followed if an error of each type occurs is described in Figures 14-100 to 14-102.

Figure 14-100. Processing Procedure in Case of Parity Error or Overrun Error

Software Manipulation	Hardware Status	Remark
Reads SDRmn register.	▶BFF = 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 SSRmn register.		Error type is identified and the read value is used to clear error flag.
Writes SIRmn register.	▶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 14-101. Processing Procedure in Case of Framing Error

Software Manipulation	Hardware Status	Remark
Reads SDRmn register.	▶BFF = 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 SSRmn register.		Error type is identified and the read value is used to clear error flag.
Writes SIRmn register.	▶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 STmn bit to 1.	▶SEmn = 0, and channel n stops operation.	
Synchronization with other party of communication		Synchronization with the other party of communication is re-established and communication is resumed because it is considered that a framing error has occurred because the start bit has been shifted.
Sets SSmn bit to 1.	▶SEmn = 1, and channel n is enabled to operate.	

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3)

Figure 14-102. Processing Procedure in Case of Parity Error (ACK error) in Simplified I^2C Mode

Software Manipulation	Hardware Status	Remark
Reads SDRmn register. →	BFF = 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 SSRmn register.		Error type is identified and the read value is used to clear error flag.
Writes SIRmn register. →	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 STmn bit to 1. →	SEmn = 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 SSmn bit to 1. →	SEmn = 1, and channel n is enabled to operate.	

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 02, 10

14.9 Relationship Between Register Settings and Pins

Tables 14-5 to 14-12 show the relationship between register settings and pins for each channel of serial array units 0 and 1.

Table 14-5. Relationship between register settings and pins (Channel 0 of unit 0: CSI00, UART0 transmission)

SE 00 Note1	MD 002	MD 001	SOE 00	SO 00	CKO 00	TXE 00	RXE 00	PM 80	P80	PM 81 Note2	P81 Note2	PM 82	P82	Operation mode	Pin Function		
															$\overline{\text{SCK00}}$/ INTP11 /P80	SI00/RxD0/ INTP9/ P81 Note2	SO00/ TxD0/P82
0	0	0	0	1	1	0	0	× Note3	× Note3	× Note3	× Note3	× Note3	× Note3	Operation stop mode	INTP11/ P80	INTP9/P81	P82
	0	1														RxD0/ INTP9/P81	
1	0	0	0	1	1	0	1	1	×	1	×	× Note3	× Note3	Slave CSI00 reception	$\overline{\text{SCK00}}$ (input)	SI00	P82
			1	0/1 Note4	1	1	0	1	×	× Note3	× Note3	0	1	Slave CSI00 transmission	$\overline{\text{SCK00}}$ (input)	INTP9/P81	SO00
			1	0/1 Note4	1	1	1	1	×	1	×	0	1	Slave CSI00 transmission/ reception	$\overline{\text{SCK00}}$ (input)	SI00	SO00
			0	1	0/1 Note4	0	1	0	1	1	×	× Note3	× Note3	Master CSI00 reception	$\overline{\text{SCK00}}$ (output)	SI00	P82
			1	0/1 Note4	0/1 Note4	1	0	0	1	× Note3	× Note3	0	1	Master CSI00 transmission	$\overline{\text{SCK00}}$ (output)	INTP9/P81	SO00
			1	0/1 Note4	0/1 Note4	1	1	0	1	1	×	0	1	Master CSI00 transmission/ reception	$\overline{\text{SCK00}}$ (output)	SI00	SO00
	0	1	1	0/1 Note4	1	1	0	× Note3	× Note3	× Note3	× Note3	0	1	UART0 transmission Note5	INTP11/ P80	RxD0/ INTP9/P81	TxD0

Notes 1. The SE0 register is a read-only status register which is set using the SS0 and ST0 registers.

2. When channel 1 of unit 0 is set to UART0 reception, this pin becomes an RxD0 function pin (refer to **Table 14-6**). In this case, operation stop mode or UART0 transmission must be selected for channel 0 of unit 0.

3. This pin can be set as a port function pin.

4. This is 0 or 1, depending on the communication operation. For details, refer to **14.3 (12) Serial output register m (SOm)**.

5. When using UART0 transmission and reception in a pair, set channel 1 of unit 0 to UART0 reception (refer to **Table 14-6**).

Remark X: Don't care

Table 14-6. Relationship between register settings and pins (Channel 1 of unit 0: UART0 reception)

SE01 Note1	MD012	MD011	TXE01	RXE01	PM81 Note2	P81 Note2	Operation mode	Pin Function
								RxD0/SI00/ INTP9/ P81 Note2
0	0	1	0	0	× Note3	× Note3	Operation stop mode	SI00/INTP9/P80
1	0	1	0	1	1	×	UART0 reception Notes4,5	RxD0

Notes 1. The SE0 register is a read-only status register which is set using the SS0 and ST0 registers.

2. When channel 1 of unit 0 is set to UART0 reception, this pin becomes an RxD0 function pin. In this case, set channel 0 of unit 0 to operation stop mode or UART0 transmission (refer to **Table 14-5**).
When channel 0 of unit 0 is set to CSI00, this pin cannot be used as an RxD0 function pin. In this case, set channel 1 of unit 0 to operation stop mode.

3. This pin can be set as a port function pin.

4. When using UART0 transmission and reception in a pair, set channel 0 of unit 0 to UART0 transmission (refer to **Table 14-5**).

5. The SMR00 register of channel 0 of unit 0 must also be set during UART0 reception. For details, refer to **14.5.2 (1) Register setting**.

Remark X: Don't care

Table 14-7. Relationship between register settings and pins
(Channel 2 of unit 0: CSI10, UART1 transmission, IIC10)

SE 02 Note1	MD 022	MD 021	SOE 02	SO 02	CKO 02	TXE 02	RXE 02	PM 15	P15	PM14 Note2	P14 Note2	PM13	P13	Operation mode	Pin Function		
															$\overline{SCK10}$/ SCL10/ INTP7/P15	SI10/SDA10/ RxD1/INTP4 /P14 Note2	SO10/ TxD1/ TO04/P13
0	0	0	0	1	1	0	0	× Note3	× Note3	× Note3	× Note3	× Note3	× Note3	Operation stop mode	INTP7/P15	INTP4/P14	TO04/P13
	0	1														RxD1/INTP4/ P14	
	1	0														INTP4/P14	
1	0	0	0	1	1	0	1	1	×	1	×	× Note3	× Note3	Slave CSI10 reception	$\overline{SCK10}$ (input)	SI10	TO04/P13
			1	0/1 Note4	1	1	0	1	×	× Note3	× Note3	0	1	Slave CSI10 transmission	$\overline{SCK10}$ (input)	INTP4/P14	SO10
			1	0/1 Note4	1	1	1	1	×	1	×	0	1	Slave CSI10 transmission /reception	$\overline{SCK10}$ (input)	SI10	SO10
			0	1	0/1 Note4	0	1	0	1	1	×	× Note3	× Note3	Master CSI10 reception	$\overline{SCK10}$ (output)	SI10	TO04/P13
			1	0/1 Note4	0/1 Note4	1	0	0	1	× Note3	× Note3	0	1	Master CSI10 transmission	$\overline{SCK10}$ (output)	INTP4/P14	SO10
			1	0/1 Note4	0/1 Note4	1	1	0	1	1	×	0	1	Master CSI10 transmission /reception	$\overline{SCK10}$ (output)	SI10	SO10
	0	1	1	0/1 Note4	1	1	0	× Note3	× Note3	× Note3	× Note3	0	1	UART1 transmission Note5	INTP7/P15	RxD1/INTP4/ P14	TxD1
0	1	0	0	0/1 Note6	0/1 Note6	0	0	0	1	0	1	× Note3	× Note3	IIC10 start condition	SCL10	SDA10	TO04/P13
						1	0										
						0	1										
1			1	0/1 Note4	0/1 Note4	1	0	0	1	0	1	× Note3	× Note3	IIC10 address field transmission	SCL10	SDA10	TO04/P13
			1	0/1 Note4	0/1 Note4	1	0	0	1	0	1	× Note3	× Note3	IIC10 data transmission	SCL10	SDA10	TO04/P13
			1	0/1 Note4	0/1 Note4	0	1	0	1	0	1	× Note3	× Note3	IIC10 data reception	SCL10	SDA10	TO04/P13
0			0	0/1 Note7	0/1 Note7	0	0	0	1	0	1	× Note3	× Note3	IIC10 stop condition	SCL10	SDA10	TO04/P13
						1	0										
						0	1										

Notes 1. The SE0 register is a read-only status register which is set using the SS0 and ST0 registers.

2. When channel 3 of unit 0 is set to UART1 reception, this pin becomes an RxD1 function pin (refer to **Table 14-8**). In this case, operation stop mode or UART1 transmission must be selected for channel 2 of unit 0.

3. This pin can be set as a port function pin.

4. This is 0 or 1, depending on the communication operation. For details, refer to **14.3 (12) Serial output register m (SOm)**.

5. When using UART1 transmission and reception in a pair, set channel 3 of unit 0 to UART1 reception (refer to **Table 14-8**).

6. Set the CKO02 bit to 1 before a start condition is generated. Clear the SO02 bit from 1 to 0 when the start condition is generated.

7. Set the CKO02 bit to 1 before a stop condition is generated. Clear the SO02 bit from 0 to 1 when the stop condition is generated.

Remark X: Don't care

Table 14-8. Relationship between register settings and pins (Channel 3 of unit 0: UART1 reception)

SE03 Note1	MD032	MD031	TXE03	RXE03	PM14 Note2	P14 Note2	Operation mode	Pin Function
								RxD1/SI10/SDA10/INTP4/P14 Note2
0	0	1	0	0	× Note3	× Note3	Operation stop mode	SI10/SDA10/INTP4/P14 Note2
1	0	1	0	1	1	×	UART1 reception Notes4,5	RxD1

Notes 1. The SE0 register is a read-only status register which is set using the SS0 and ST0 registers.

2. When channel 3 of unit 0 is set to UART1 reception, this pin becomes an RxD1 function pin. In this case, set channel 2 of unit 0 to operation stop mode or UART1 transmission (refer to **Table 14-7**).
When channel 2 of unit 0 is set to CSI10 or IIC10, this pin cannot be used as an RxD1 function pin. In this case, set channel 3 of unit 0 to operation stop mode.

3. This pin can be set as a port function pin.

4. When using UART1 transmission and reception in a pair, set channel 2 of unit 0 to UART1 transmission (refer to **Table 14-7**).

5. The SMR02 register of channel 2 of unit 0 must also be set during UART1 reception. For details, refer to **14.5.2 (1) Register setting**.

Remark X: Don't care

Table 14-9. Relationship between register settings and pins
(Channel 0 of unit 1: CSI20, UART2 transmission, IIC20)

SE 10 Note1	MD 102	MD 101	SOE 10	SO 10	CKO 10	TXE 10	RXE 10	PM 10	P10	PM 11 Note2	P11 Note2	PM 12	P12	Operation mode	Pin Function		
															$\overline{SCK20}$/ SCL20/P10	SI20/SDA20/ RxD2/INTP6/ P11 Note2	SO20/ TxD2/ TO02/P12
0	0	0	0	1	1	0	0	× Note3	× Note3	× Note3	× Note3	× Note3	× Note3	Operation stop mode	P10	INTP6/P11	TO02/P12
	0	1														RxD2/INTP6/ P11	
	1	0														INTP6/P11	
1	0	0	0	1	1	0	1	1	×	1	×	× Note3	× Note3	Slave CSI20 reception	$\overline{SCK20}$ (input)	SI20	TO02/P12
			1	0/1 Note4	1	1	0	1	×	× Note3	× Note3	0	1	Slave CSI20 transmission	$\overline{SCK20}$ (input)	INTP6/P11	SO20
			1	0/1 Note4	1	1	1	1	×	1	×	0	1	Slave CSI20 transmission/reception	$\overline{SCK20}$ (input)	SI20	SO20
			0	1	0/1 Note4	0	1	0	1	1	×	× Note3	× Note3	Master CSI20 reception	$\overline{SCK20}$ (output)	SI20	TO02/P12
			1	0/1 Note4	0/1 Note4	1	0	0	1	× Note3	× Note3	0	1	Master CSI20 transmission	$\overline{SCK20}$ (output)	INTP6/P11	SO20
			1	0/1 Note4	0/1 Note4	1	1	0	1	1	×	0	1	Master CSI20 transmission/reception	$\overline{SCK20}$ (output)	SI20	SO20
	0	1	1	0/1 Note4	1	1	0	× Note3	× Note3	× Note3	× Note3	0	1	UART2 transmission Note5	P10	RxD2/INTP6/ P11	TxD2
0	1	0	0	0/1 Note6	0/1 Note6	0	0	0	1	0	1	× Note3	× Note3	IIC20 start condition	SCL20	SDA20	TO02/P12
						1	0										
						0	1										
1			1	0/1 Note4	0/1 Note4	1	0	0	1	0	1	× Note3	× Note3	IIC20 address field transmission	SCL20	SDA20	TO02/P12
			1	0/1 Note4	0/1 Note4	1	0	0	1	0	1	× Note3	× Note3	IIC20 data transmission	SCL20	SDA20	TO02/P12
			1	0/1 Note4	0/1 Note4	0	1	0	1	0	1	× Note3	× Note3	IIC20 data reception	SCL20	SDA20	TO02/P12
0			0	0/1 Note7	0/1 Note7	0	0	0	1	0	1	× Note3	× Note3	IIC20 stop condition	SCL20	SDA20	TO02/P12
						1	0										
						0	1										

Notes 1. The SE1 register is a read-only status register which is set using the SS1 and ST1 registers.

2. When channel 1 of unit 1 is set to UART2 reception, this pin becomes an RxD2 function pin (refer to **Table 14-10**). In this case, operation stop mode or UART2 transmission must be selected for channel 0 of unit 1.

3. This pin can be set as a port function pin.

4. This is 0 or 1, depending on the communication operation. For details, refer to **14.3 (12) Serial output register m (SOm)**.

5. When using UART2 transmission and reception in a pair, set channel 1 of unit 1 to UART2 reception (refer to **Table 14-10**).

6. Set the CKO10 bit to 1 before a start condition is generated. Clear the SO10 bit from 1 to 0 when the start condition is generated.

7. Set the CKO10 bit to 1 before a stop condition is generated. Clear the SO10 bit from 0 to 1 when the stop condition is generated.

Remark X: Don't care

Table 14-10. Relationship between register settings and pins (Channel 1 of unit 1: UART2 reception)

SE11 Note1	MD112	MD111	TXE11	RXE11	PM11 Note2	P11 Note2	Operation mode	Pin Function
								SI20/SDA20/RxD2/ INTP6/P11 Note2
0	0	1	0	0	× Note3	× Note3	Operation stop mode	SI20/SDA20/INTP6/P11
1	0	1	0	1	1	×	UART2 reception Notes4,5	RxD2

Notes 1. The SE1 register is a read-only status register which is set using the SS1 and ST1 registers.

2. When channel 1 of unit 1 is set to UART2 reception, this pin becomes an RxD2 function pin. In this case, set channel 0 of unit 1 to operation stop mode or UART2 transmission (refer to **Table 14-9**).
When channel 0 of unit 1 is set to CSI20 or IIC20, this pin cannot be used as an RxD2 function pin. In this case, set channel 1 of unit 1 to operation stop mode.

3. This pin can be set as a port function pin.

4. When using UART2 transmission and reception in a pair, set channel 0 of unit 1 to UART2 transmission (refer to **Table 14-9**).

5. The SMR10 register of channel 0 of unit 1 must also be set during UART2 reception. For details, refer to **14.5.2 (1) Register setting**.

Remark X: Don't care

Table 14-11. Relationship between register settings and pins (Channel 2 of unit 1: UART3 transmission)

SE12 Note1	MD122	MD121	SOE12	SO12	TXE12	RXE12	PM51	P51	Operation mode	Pin Function
										TxD3/P51
0	0	1	0	1	0	0	× Note2	× Note2	Operation stop mode	P51
1	0	1	1	0/1 Note3	1	0	0	1	UART3 transmission Note4	TxD3

Notes 1. The SE1 register is a read-only status register which is set using the SS1 and ST1 registers.

2. This pin can be set as a port function pin.

3. This is 0 or 1, depending on the communication operation. For details, refer to **14.3 (12) Serial output register m (SOm)**.

4. When using UART3 transmission and reception in a pair, set channel 3 of unit 1 to UART3 reception (refer to **Table 14-12**).

Remark X: Don't care

Table 14-12. Relationship between register settings and pins (Channel 3 of unit 1: UART3 reception)

SE13 Note1	MD132	MD131	TXE13	RXE13	PM50	P50	Operation mode	Pin Function
								RxD3/P50
0	0	1	0	0	× Note2	× Note2	Operation stop mode	P50
1	0	1	0	1	1	×	UART3 reception Notes3,4	RxD3

Notes 1. The SE1 register is a read-only status register which is set using the SS1 and ST1 registers.

2. This pin can be set as a port function pin.

3. When using UART3 transmission and reception in a pair, set channel 2 of unit 1 to UART3 transmission (refer to **Table 14-11**).

4. The SMR12 register of channel 2 of unit 1 must also be set during UART3 reception. For details, refer to **14.5.2 (1) Register setting**.

Remark X: Don't care

CHAPTER 15 SERIAL INTERFACE IICA

15.1 Functions of Serial Interface IICA

Serial interface IICA has the following three modes.

(1) Operation stop mode

This mode is used when serial transfers are not performed. It can therefore be used to reduce power consumption.

(2) I^2C bus mode (multimaster supported)

This mode is used for 8-bit data transfers with several devices via two lines: a serial clock (SCL0) line and a serial data bus (SDA0) line.

This mode complies with the I^2C bus format and the master device can generated “start condition”, “address”, “transfer direction specification”, “data”, and “stop condition” data to the slave device, via the serial data bus. The slave device automatically detects these received status and data by hardware. This function can simplify the part of application program that controls the I^2C bus.

Since the SCL0 and SDA0 pins are used for open drain outputs, IICA requires pull-up resistors for the serial clock line and the serial data bus line.

(3) Wakeup mode

The STOP mode can be released by generating an interrupt request signal (INTIICA) when an extension code from the master device or a local address has been received while in STOP mode. This can be set by using the WUP bit of IICA control register 1 (IICCTL1).

Figure 15-1 shows a block diagram of serial interface IICA.

Figure 15-1. Block Diagram of Serial Interface IICA

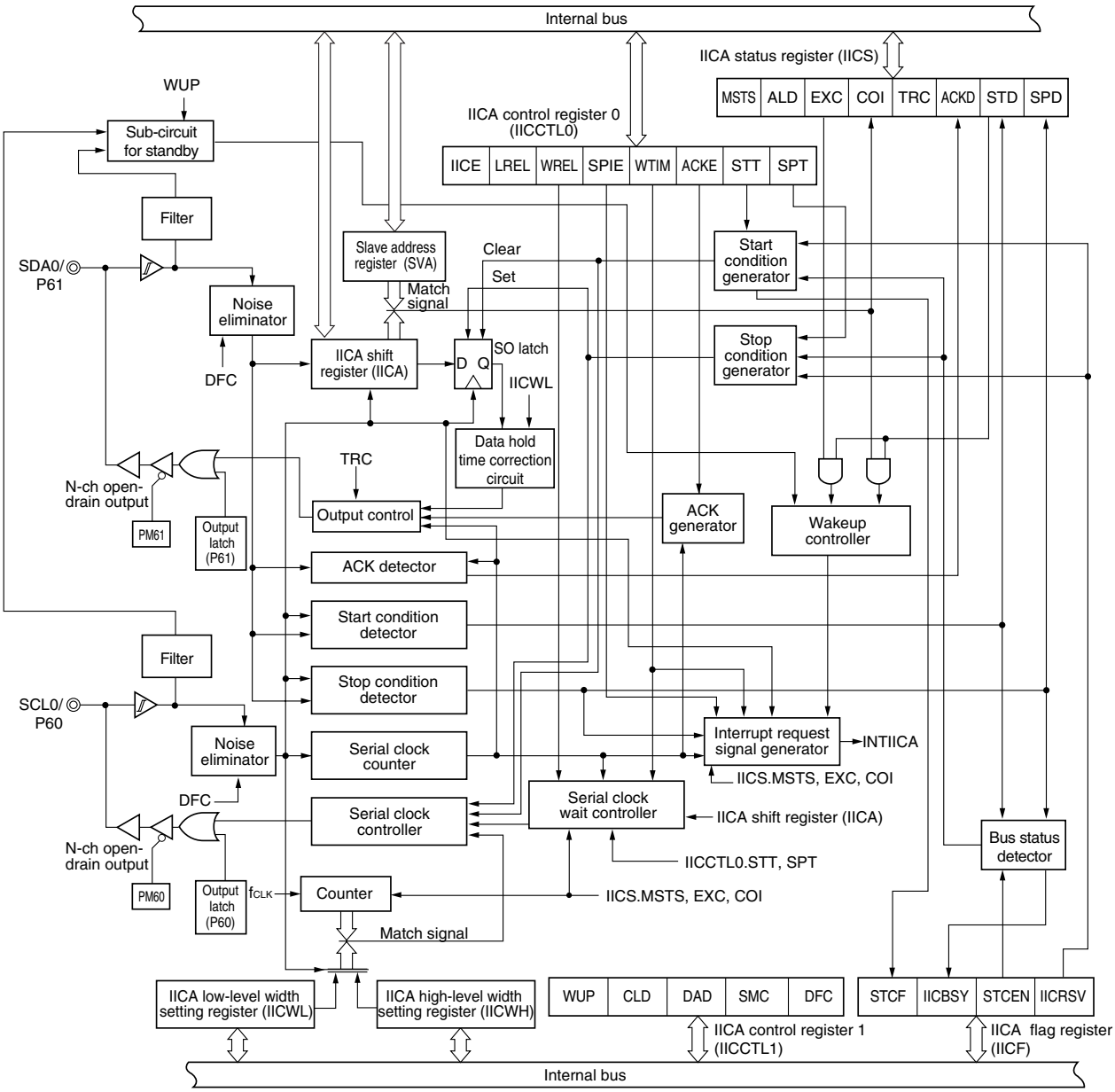
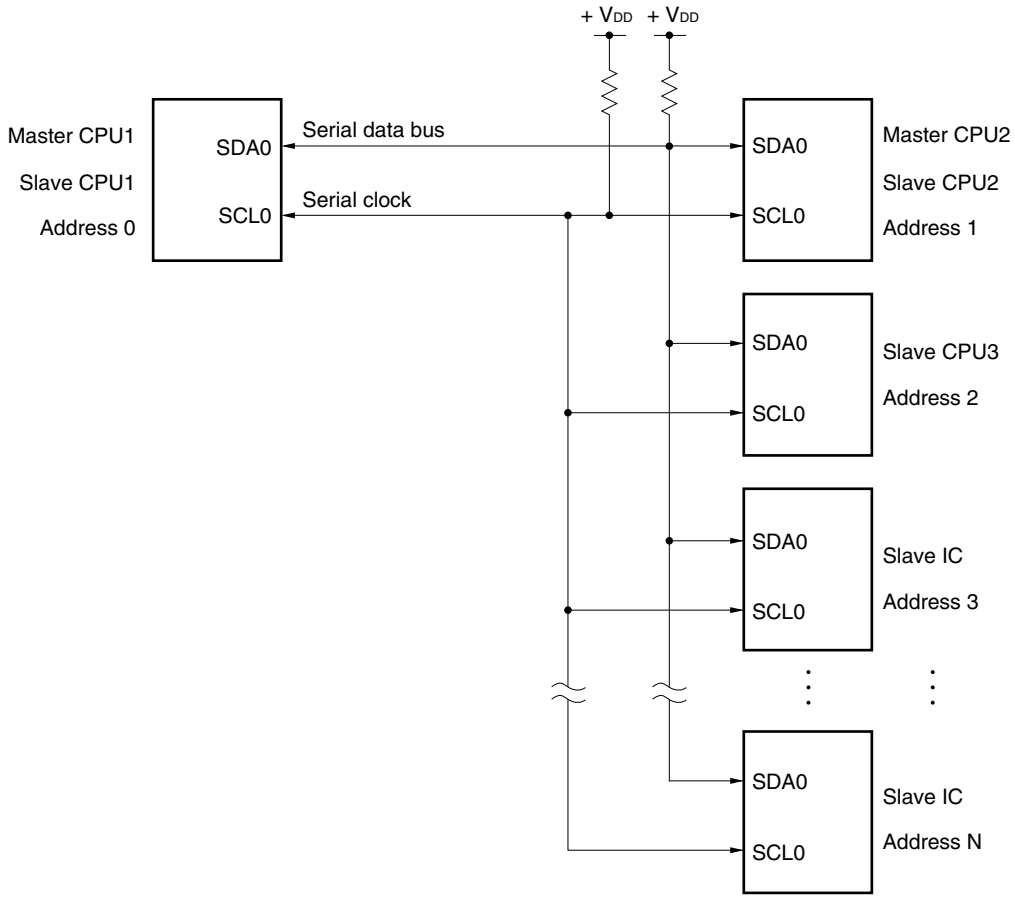


Figure 15-2 shows a serial bus configuration example.

Figure 15-2. Serial Bus Configuration Example Using I²C Bus



15.2 Configuration of Serial Interface IICA

Serial interface IICA includes the following hardware.

Table 15-1. Configuration of Serial Interface IICA

Item	Configuration
Registers	IICA shift register (IICA) Slave address register (SVA)
Control registers	Peripheral enable register 0 (PER0) IICA control register 0 (IICCTL0) IICA status register (IICS) IICA flag register (IICF) IICA control register 1 (IICCTL1) IICA low-level width setting register (IICWL) IICA high-level width setting register (IICWH) Port mode register 6 (PM6) Port register 6 (P6)

(1) IICA shift register (IICA)

IICA is used to convert 8-bit serial data to 8-bit parallel data and vice versa in synchronization with the serial clock. IICA can be used for both transmission and reception.

The actual transmit and receive operations can be controlled by writing and reading operations to IICA.

Cancel the wait state and start data transfer by writing data to IICA during the wait period.

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

Reset signal generation clears IICA to 00H.

Figure 15-3. Format of IICA Shift Register (IICA)

Address: FFF50H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
IICA								

Cautions 1. Do not write data to IICA during data transfer.

2. Write or read IICA only during the wait period. Accessing IICA in a communication state other than during the wait period is prohibited. When the device serves as the master, however, IICA can be written only once after the communication trigger bit (STT) is set to 1.

3. When communication is reserved, write data to IICA after the interrupt triggered by a stop condition is detected.

(2) Slave address register (SVA)

This register stores seven bits of local addresses {A6, A5, A4, A3, A2, A1, A0} when in slave mode.

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

However, rewriting to this register is prohibited while STD = 1 (while the start condition is detected).

Reset signal generation clears SVA to 00H.

Figure 15-4. Format of Slave Address Register (SVA)

Address: F0234H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
SVA	A6	A5	A4	A3	A2	A1	A0	0[Note]

Note Bit 0 is fixed to 0.

(3) SO latch

The SO latch is used to retain the SDA0 pin's output level.

(4) Wakeup controller

This circuit generates an interrupt request (INTIICA) when the address received by this register matches the address value set to the slave address register (SVA) or when an extension code is received.

(5) Serial clock counter

This counter counts the serial clocks that are output or input during transmit/receive operations and is used to verify that 8-bit data was transmitted or received.

(6) Interrupt request signal generator

This circuit controls the generation of interrupt request signals (INTIICA).

An I^2C interrupt request is generated by the following two triggers.

- Falling edge of eighth or ninth clock of the serial clock (set by WTIM bit)
- Interrupt request generated when a stop condition is detected (set by SPIE bit)

Remark WTIM bit: Bit 3 of IICA control register 0 (IICCTL0)
SPIE bit: Bit 4 of IICA control register 0 (IICCTL0)

(7) Serial clock controller

In master mode, this circuit generates the clock output via the SCL0 pin from a sampling clock.

(8) Serial clock wait controller

This circuit controls the wait timing.

(9) ACK generator, stop condition detector, start condition detector, and ACK detector

These circuits generate and detect each status.

(10) Data hold time correction circuit

This circuit generates the hold time for data corresponding to the falling edge of the serial clock.

(11) Start condition generator

This circuit generates a start condition when the STT bit is set to 1.

However, in the communication reservation disabled status (IICRSV bit = 1), when the bus is not released (IICBSY bit = 1), start condition requests are ignored and the STCF bit is set to 1.

(12) Stop condition generator

This circuit generates a stop condition when the SPT bit is set to 1.

(13) Bus status detector

This circuit detects whether or not the bus is released by detecting start conditions and stop conditions. However, as the bus status cannot be detected immediately following operation, the initial status is set by the STCEN bit.

Remark STT bit: Bit 1 of IICA control register 0 (IICCTL0)
SPT bit: Bit 0 of IICA control register 0 (IICCTL0)
IICRSV bit: Bit 0 of IICA flag register (IICF)
IICBSY bit: Bit 6 of IICA flag register (IICF)
STCF bit: Bit 7 of IICA flag register (IICF)
STCEN bit: Bit 1 of IICA flag register (IICF)

15.3 Registers Controlling Serial Interface IICA

Serial interface IICA is controlled by the following nine registers.

- Peripheral enable register 0 (PER0)
- IICA control register 0 (IICCTL0)
- IICA flag register (IICF)
- IICA status register (IICS)
- IICA control register 1 (IICCTL1)
- IICA low-level width setting register (IICWL)
- IICA high-level width setting register (IICWH)
- Port mode register 6 (PM6)
- Port register 6 (P6)

(1) Peripheral enable register 0 (PER0)

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

When serial interface IICA is used, be sure to set bit 4 (IICAEN) of this register to 1.

PER0 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 15-5. Format of Peripheral Enable Register 0 (PER0)

Address: F00F0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PER0	RTCEN	DACEN	ADCEN	IICAEN	SAU1EN	SAU0EN	TAU1EN	TAU0EN

IICAEN	Control of serial interface IICA input clock
0	Stops supply of input clock. • SFR used by serial interface IICA cannot be written. • Serial interface IICA is in the reset status.
1	Supplies input clock. • SFR used by serial interface IICA can be read/written.

Caution When setting serial interface IICA, be sure to set IICAEN to 1 first. If IICAEN = 0, writing to a control register of serial interface IICA is ignored, and, even if the register is read, only the default value is read.

(2) IICA control register 0 (IICCTL0)

This register is used to enable/stop I^2C operations, set wait timing, and set other I^2C operations.

IICCTL0 can be set by a 1-bit or 8-bit memory manipulation instruction. However, set the SPIE, WTIM, and ACKE bits while IICE bit = 0 or during the wait period. These bits can be set at the same time when the IICE bit is set from “0” to “1”.

Reset signal generation clears this register to 00H.

Figure 15-6. Format of IICA Control Register 0 (IICCTL0) (1/4)

Address: F0230H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICCTL0	IICE	LREL	WREL	SPIE	WTIM	ACKE	STT	SPT

IICE	I^2C operation enable
0	Stop operation. Reset the IICA status register (IICS)[Note 1]. Stop internal operation.
1	Enable operation.
Be sure to set this bit (1) while the SCL0 and SDA0 lines are at high level.	
Condition for clearing (IICE = 0)	Condition for setting (IICE = 1)
• Cleared by instruction • Reset	• Set by instruction

LREL[Notes 2,3]	Exit from communications
0	Normal operation
1	This exits from the current communications and sets standby mode. This setting is automatically cleared to 0 after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCL0 and SDA0 lines are set to high impedance. The following flags of IICA control register 0 (IICCTL0) and IICA status register (IICS) are cleared to 0. • STT • SPT • MSTS • EXC • COI • TRC • ACKD • STD
The standby mode following exit from communications remains in effect until the following communications entry conditions are met. • After a stop condition is detected, restart is in master mode. • An address match or extension code reception occurs after the start condition.	
Condition for clearing (LREL = 0)	Condition for setting (LREL = 1)
• Automatically cleared after execution • Reset	• Set by instruction

WREL[Notes 2,3]	Wait cancellation
0	Do not cancel wait
1	Cancel wait. This setting is automatically cleared after wait is canceled.
When WREL is set (wait canceled) during the wait period at the ninth clock pulse in the transmission status (TRC = 1), the SDA0 line goes into the high impedance state (TRC = 0).	
Condition for clearing (WREL = 0)	Condition for setting (WREL = 1)
• Automatically cleared after execution • Reset	• Set by instruction

Notes 1. The IICS register, the STCF and IICBSY bits of the IICF register, and the CLD and DAD bits of the IICCTL1 register are reset.

2. The signal of this bit is invalid while IICE0 is 0.

3. When the LREL and WREL bits are read, 0 is always read.

Caution If the operation of I^2C is enabled (IICE = 1) when the SCL0 line is at high level, the SDA0 line is at low level, and DFC of the IICCTL1 register is 1, a start condition will be inadvertently detected immediately. Immediately after enabling I^2C to operate (IICE = 1), set LREL (1) by using a 1-bit memory manipulation instruction.

Figure 15-6. Format of IICA Control Register 0 (IICCTL0) (2/4)

SPIE[Note 1]	Enable/disable generation of interrupt request when stop condition is detected
0	Disable
1	Enable
If WUP of the IICCTL1 register is 1, no stop condition interrupt will be generated even if SPIE = 1.	
Condition for clearing (SPIE = 0)	Condition for setting (SPIE = 1)
• Cleared by instruction • Reset	• Set by instruction

WTIM[Note 1]	Control of wait and interrupt request generation
0	Interrupt request is generated at the eighth clock's falling edge. Master mode: After output of eight clocks, clock output is set to low level and wait is set. Slave mode: After input of eight clocks, the clock is set to low level and wait is set for master device.
1	Interrupt request is generated at the ninth clock's falling edge. Master mode: After output of nine clocks, clock output is set to low level and wait is set. Slave mode: After input of nine clocks, the clock is set to low level and wait is set for master device.
An interrupt is generated at the falling edge of the ninth clock during address transfer independently of the setting of this bit. The setting of this bit is valid when the address transfer is completed. When in master mode, a wait is inserted at the falling edge of the ninth clock during address transfers. For a slave device that has received a local address, a wait is inserted at the falling edge of the ninth clock after an acknowledge ($\overline{ACK}$) is issued. However, when the slave device has received an extension code, a wait is inserted at the falling edge of the eighth clock.	
Condition for clearing (WTIM = 0)	Condition for setting (WTIM = 1)
• Cleared by instruction • Reset	• Set by instruction

ACKE[Notes 1, 2]	Acknowledgment control
0	Disable acknowledgment.
1	Enable acknowledgment. During the ninth clock period, the SDA0 line is set to low level.
Condition for clearing (ACKE = 0)	Condition for setting (ACKE = 1)
• Cleared by instruction • Reset	• Set by instruction

Notes 1. The signal of this bit is invalid while IICE0 is 0. Set this bit during that period.

2. The set value is invalid during address transfer and if the code is not an extension code. When the device serves as a slave and the addresses match, an acknowledgment is generated regardless of the set value.

Figure 15-6. Format of IICA Control Register 0 (IICCTL0) (3/4)

STT[Note]	Start condition trigger
0	Do not generate a start condition.
1	When bus is released (in standby state, when IICBSY = 0): If this bit is set (1), a start condition is generated (startup as the master). When a third party is communicating: • When communication reservation function is enabled (IICRSV = 0) Functions as the start condition reservation flag. When set to 1, automatically generates a start condition after the bus is released. • When communication reservation function is disabled (IICRSV = 1) Even if this bit is set (1), the STT bit is cleared and the STT clear flag (STCF) is set (1). No start condition is generated. In the wait state (when master device): Generates a restart condition after releasing the wait.

Cautions concerning set timing
- For master reception: Cannot be set to 1 during transfer. Can be set to 1 only in the waiting period when ACKE has been cleared to 0 and slave has been notified of final reception.
- For master transmission: A start condition cannot be generated normally during the acknowledge period. Set to 1 during the wait period that follows output of the ninth clock.
- Cannot be set to 1 at the same time as SPT.
- Setting STT to 1 and then setting it again before it is cleared to 0 is prohibited.

Condition for clearing (STT = 0)	Condition for setting (STT = 1)
• Cleared by setting STT to 1 while communication reservation is prohibited. • Cleared by loss in arbitration • Cleared after start condition is generated by master device • Cleared by LREL = 1 (exit from communications) • When IICE = 0 (operation stop) • Reset	• Set by instruction

Note The signal of this bit is invalid while IICE0 is 0.

Remarks 1. Bit 1 (STT) becomes 0 when it is read after data setting.

2. IICRSV: Bit 0 of IIC flag register (IICF)
STCF: Bit 7 of IIC flag register (IICF)

Figure 15-6. Format of IICA Control Register 0 (IICCTL0) (4/4)

SPT	Stop condition trigger
0	Stop condition is not generated.
1	Stop condition is generated (termination of master device's transfer).

Cautions concerning set timing
- For master reception: Cannot be set to 1 during transfer. Can be set to 1 only in the waiting period when ACKE has been cleared to 0 and slave has been notified of final reception.
- For master transmission: A stop condition cannot be generated normally during the acknowledge period. Therefore, set it during the wait period that follows output of the ninth clock.
- Cannot be set to 1 at the same time as STT.
- SPT can be set to 1 only when in master mode.
- When WTIM has been cleared to 0, if SPT is set to 1 during the wait period that follows output of eight clocks, note that a stop condition will be generated during the high-level period of the ninth clock. WTIM should be changed from 0 to 1 during the wait period following the output of eight clocks, and SPT should be set to 1 during the wait period that follows the output of the ninth clock.
- Setting SPT to 1 and then setting it again before it is cleared to 0 is prohibited.

Condition for clearing (SPT = 0)	Condition for setting (SPT = 1)
• Cleared by loss in arbitration • Automatically cleared after stop condition is detected • Cleared by LREL = 1 (exit from communications) • When IICE = 0 (operation stop) • Reset	• Set by instruction

Caution When bit 3 (TRC) of the IICA status register (IICS) is set to 1, WREL is set to 1 during the ninth clock and wait is canceled, after which TRC is cleared and the SDA0 line is set to high impedance. Release the wait performed while the TRC bit is 1 (transmission status) by writing to the IICA shift register.

<R>

Remark Bit 0 (SPT) becomes 0 when it is read after data setting.

(3) IICA status register (IICS)

This register indicates the status of I^2C.

IICS is read by a 1-bit or 8-bit memory manipulation instruction only when STT = 1 and during the wait period.

Reset signal generation clears this register to 00H.

Caution Reading the IICS register while the address match wakeup function is enabled (WUP = 1) in STOP mode is prohibited. When the WUP bit is changed from 1 to 0 (wakeup operation is stopped), regardless of the INTIICA interrupt request, the change in status is not reflected until the next start condition or stop condition is detected. To use the wakeup function, therefore, enable (SPIE = 1) the interrupt generated by detecting a stop condition and read the IICS register after the interrupt has been detected.

Remark STT: bit 1 of IICA control register 0 (IICCTL0)
WUP: bit 7 of IICA control register 1 (IICCTL1)

Figure 15-7. Format of IICA Status Register (IICS) (1/3)

Address: FFF51H After reset: 00H R

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICS	MSTS	ALD	EXC	COI	TRC	ACKD	STD	SPD

MSTS	Master status check flag
0	Slave device status or communication standby status
1	Master device communication status

Condition for clearing (MSTS = 0)	Condition for setting (MSTS = 1)
• When a stop condition is detected • When ALD = 1 (arbitration loss) • Cleared by LREL = 1 (exit from communications) • When IICE changes from 1 to 0 (operation stop) • Reset	• When a start condition is generated

ALD	Detection of arbitration loss
0	This status means either that there was no arbitration or that the arbitration result was a “win”.
1	This status indicates the arbitration result was a “loss”. MSTS is cleared.

Condition for clearing (ALD = 0)	Condition for setting (ALD = 1)
• Automatically cleared after IICS is read[Note] • When IICE changes from 1 to 0 (operation stop) • Reset	• When the arbitration result is a “loss”.

EXC	Detection of extension code reception
0	Extension code was not received.
1	Extension code was received.

Condition for clearing (EXC = 0)	Condition for setting (EXC = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL = 1 (exit from communications) • When IICE changes from 1 to 0 (operation stop) • Reset	• When the higher four bits of the received address data is either “0000” or “1111” (set at the rising edge of the eighth clock).

Note This register is also cleared when a 1-bit memory manipulation instruction is executed for bits other than IICS. Therefore, when using the ALD bit, read the data of this bit before the data of the other bits.

Remark LREL: Bit 6 of IICA control register 0 (IICCTL0)
IICE: Bit 7 of IICA control register 0 (IICCTL0)

Figure 15-7. Format of IICA Status Register (IICS) (2/3)

COI	Detection of matching addresses
0	Addresses do not match.
1	Addresses match.

Condition for clearing (COI = 0)	Condition for setting (COI = 1)
• When a start condition is detected • When a stop condition is detected • Cleared by LREL = 1 (exit from communications) • When IICE changes from 1 to 0 (operation stop) • Reset	• When the received address matches the local address (slave address register (SVA)) (set at the rising edge of the eighth clock).

TRC	Detection of transmit/receive status
0	Receive status (other than transmit status). The SDA0 line is set for high impedance.
1	Transmit status. The value in the SO0 latch is enabled for output to the SDA0 line (valid starting at the falling edge of the first byte's ninth clock).

Condition for clearing (TRC = 0)	Condition for setting (TRC = 1)
<Both master and slave> • When a stop condition is detected • Cleared by LREL = 1 (exit from communications) • When the IICE bit changes from 1 to 0 (operation stop) • Cleared by WREL = 1[Note] (wait cancel) • When the ALD bit changes from 0 to 1 (arbitration loss) • Reset • When not used for communication (MSTS, EXC, COI = 0) <Master> • When "1" is output to the first byte's LSB (transfer direction specification bit) <Slave> • When a start condition is detected • When "0" is input to the first byte's LSB (transfer direction specification bit)	<Master> • When a start condition is generated • When 0 (master transmission) is output to the LSB (transfer direction specification bit) of the first byte (during address transfer) <Slave> • When 1 (slave transmission) is input to the LSB (transfer direction specification bit) of the first byte from the master (during address transfer)

Note When bit 3 (TRC) of the IICA status register (IICS) is set to 1 (transmission status), bit 5 (WREL) of IICA control register 0 (IICCTL0) is set to 1 during the ninth clock and wait is canceled, after which the TRC bit is cleared (reception status) and the SDA0 line is set to high impedance. Release the wait performed while the TRC bit is 1 (transmission status) by writing to the IICA shift register. <R>

Remark LREL: Bit 6 of IICA control register 0 (IICCTL0)
IICE: Bit 7 of IICA control register 0 (IICCTL0)

Figure 15-7. Format of IICA Status Register (IICS) (3/3)

ACKD	Detection of acknowledge ($\overline{ACK}$)
0	Acknowledge was not detected.
1	Acknowledge was detected.

Condition for clearing (ACKD = 0)	Condition for setting (ACKD = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock • Cleared by LREL = 1 (exit from communications) • When IICE changes from 1 to 0 (operation stop) • Reset	• After the SDA0 line is set to low level at the rising edge of SCL0's ninth clock

STD	Detection of start condition
0	Start condition was not detected.
1	Start condition was detected. This indicates that the address transfer period is in effect.

Condition for clearing (STD = 0)	Condition for setting (STD = 1)
• When a stop condition is detected • At the rising edge of the next byte's first clock following address transfer • Cleared by LREL = 1 (exit from communications) • When IICE changes from 1 to 0 (operation stop) • Reset	• When a start condition is detected

SPD	Detection of stop condition
0	Stop condition was not detected.
1	Stop condition was detected. The master device's communication is terminated and the bus is released.

Condition for clearing (SPD = 0)	Condition for setting (SPD = 1)
• At the rising edge of the address transfer byte's first clock following setting of this bit and detection of a start condition • When IICE changes from 1 to 0 (operation stop) • Reset	• When a stop condition is detected

Remark LREL: Bit 6 of IICA control register 0 (IICCTL0)
IICE: Bit 7 of IICA control register 0 (IICCTL0)

(4) IICA flag register (IICF)

This register sets the operation mode of I^2C and indicates the status of the I^2C bus.

IICF can be set by a 1-bit or 8-bit memory manipulation instruction. However, the STCF and IICBSY bits are read-only.

The IICRSV bit can be used to enable/disable the communication reservation function.

STCEN can be used to set the initial value of the IICBSY bit.

IICRSV and STCEN can be written only when the operation of I^2C is disabled (bit 7 (IICE) of IICA control register 0 (IICCTL0) = 0). When operation is enabled, the IICF register can be read.

Reset signal generation clears this register to 00H.

Figure 15-8. Format of IICA Flag Register (IICF)

Address: FFF52H After reset: 00H R/W[Note]

Symbol	<7>	<6>	5	4	3	2	<1>	<0>
IICF	STCF	IICBSY	0	0	0	0	STCEN	IICRSV

STCF	STT clear flag
0	Generate start condition
1	Start condition generation unsuccessful: clear STT flag

Condition for clearing (STCF = 0)	Condition for setting (STCF = 1)
• Cleared by STT = 1 • When IICE = 0 (operation stop) • Reset	• Generating start condition unsuccessful and STT cleared to 0 when communication reservation is disabled (IICRSV = 1).

IICBSY	I^2C bus status flag
0	Bus release status (communication initial status when STCEN = 1)
1	Bus communication status (communication initial status when STCEN = 0)

Condition for clearing (IICBSY = 0)	Condition for setting (IICBSY = 1)
• Detection of stop condition • When IICE = 0 (operation stop) • Reset	• Detection of start condition • Setting of IICE when STCEN = 0

STCEN	Initial start enable trigger
0	After operation is enabled (IICE = 1), enable generation of a start condition upon detection of a stop condition.
1	After operation is enabled (IICE = 1), enable generation of a start condition without detecting a stop condition.

Condition for clearing (STCEN = 0)	Condition for setting (STCEN = 1)
• Cleared by instruction • Detection of start condition • Reset	• Set by instruction

IICRSV	Communication reservation function disable bit
0	Enable communication reservation
1	Disable communication reservation

Condition for clearing (IICRSV = 0)	Condition for setting (IICRSV = 1)
• Cleared by instruction • Reset	• Set by instruction

Note Bits 6 and 7 are read-only.

Cautions 1. Write to STCEN only when the operation is stopped (IICE = 0).

2. As the bus release status (IICBSY = 0) is recognized regardless of the actual bus status when STCEN = 1, when generating the first start condition (STT = 1), it is necessary to verify that no third party communications are in progress in order to prevent such communications from being destroyed.

3. Write to IICRSV only when the operation is stopped (IICE = 0).

Remark STT: Bit 1 of IICA control register 0 (IICCTL0)
IICE: Bit 7 of IICA control register 0 (IICCTL0)

(5) IICA control register 1 (IICCTL1)

This register is used to set the operation mode of I^2C and detect the statuses of the SCL0 and SDA0 pins.

IICCTL1 can be set by a 1-bit or 8-bit memory manipulation instruction. However, the CLD and DAD bits are read-only.

Set the IICCTL1 register, except the WUP bit, while operation of I^2C is disabled (bit 7 (IICE) of IICA control register 0 (IICCTL0) is 0).

Reset signal generation clears this register to 00H.

Figure 15-9. Format of IICA Control Register 1 (IICCTL1) (1/2)

Address: F0231H After reset: 00H R/W[Note 1]

Symbol	7	6	<5>	<4>	<3>	<2>	1	0
IICCTL1	WUP	0	CLD	DAD	SMC	DFC	0	0

WUP	Control of address match wakeup
0	Stops operation of address match wakeup function in STOP mode.
1	Enables operation of address match wakeup function in STOP mode.

To shift to STOP mode when WUP = 1, execute the STOP instruction at least three clocks after setting (1) the WUP bit (see **Figure 15-22 Flow When Setting WUP = 1**).

Clear (0) the WUP bit after the address has matched or an extension code has been received. The subsequent communication can be entered by the clearing (0) WUP bit. (The wait must be released and transmit data must be written after the WUP bit has been cleared (0).)

The interrupt timing when the address has matched or when an extension code has been received, while WUP = 1, is identical to the interrupt timing when WUP = 0. (A delay of the difference of sampling by the clock will occur.) Furthermore, when WUP = 1, a stop condition interrupt is not generated even if the SPIE bit is set to 1.

When WUP = 0 is set by a source other than an interrupt from serial interface IICA, operation as the master device cannot be performed until the subsequent start condition or stop condition is detected. Do not output a start condition by setting (1) the STT bit, without waiting for the detection of the subsequent start condition or stop condition.

Condition for clearing (WUP = 0)	Condition for setting (WUP = 1)
• Cleared by instruction (after address match or extension code reception)	• Set by instruction (when the MSTS, EXC, and COI bits are “0”, and the STD bit also “0” (communication not entered))[Note 2]

<R> **Notes 1.** Bits 4 and 5 are read-only.

2. The status of the IICA status register (IICS) must be checked and the WUP bit must be set during the period shown below.

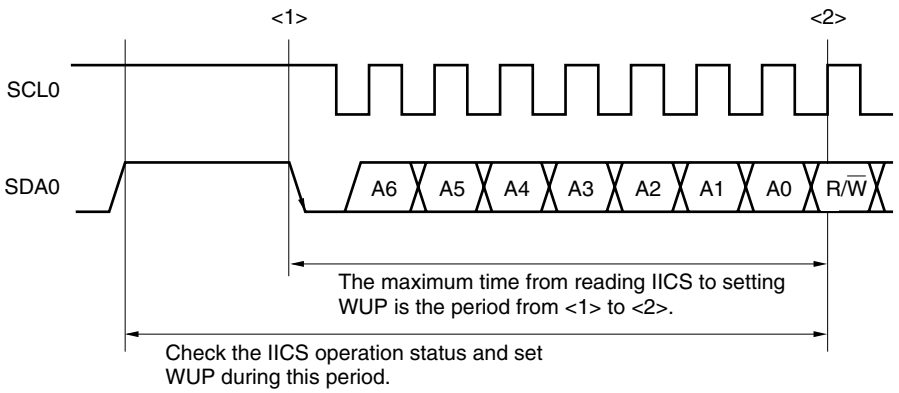


Figure 15-9. Format of IICA Control Register 1 (IICCTL1) (2/2)

CLD	Detection of SCL0 pin level (valid only when IICE = 1)
0	The SCL0 pin was detected at low level.
1	The SCL0 pin was detected at high level.

Condition for clearing (CLD = 0)	Condition for setting (CLD = 1)
• When the SCL0 pin is at low level • When IICE = 0 (operation stop) • Reset	• When the SCL0 pin is at high level

DAD	Detection of SDA0 pin level (valid only when IICE = 1)
0	The SDA0 pin was detected at low level.
1	The SDA0 pin was detected at high level.

Condition for clearing (DAD = 0)	Condition for setting (DAD = 1)
• When the SDA0 pin is at low level • When IICE = 0 (operation stop) • Reset	• When the SDA0 pin is at high level

SMC	Operation mode switching
0	Operates in standard mode.
1	Operates in fast mode.

DFC	Digital filter operation control
0	Digital filter off.
1	Digital filter on.
Digital filter can be used only in fast mode. In fast mode, the transfer clock does not vary, regardless of the DFC bit being set (1) or cleared (0). The digital filter is used for noise elimination in fast mode.	

Remark IICE: Bit 7 of IICA control register 0 (IICCTL0)

(6) IICA low-level width setting register (IICWL)

This register is used to set the low-level width of the SCL0 pin signal that is output by serial interface IICA.

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

Set the IICWL register while operation of I^2C is disabled (bit 7 (IICE) of IICA control register 0 (IICCTL0) is 0).

Reset signal generation sets this register to FFH.

Figure 15-10. Format of IICA Low-Level Width Setting Register (IICWL)

Address: F0232H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWL								

(7) IICA high-level width setting register (IICWH)

This register is used to set the high-level width of the SCL0 pin signal that is output by serial interface IICA.

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

Set the IICWL register while operation of I^2C is disabled (bit 7 (IICE) of IICA control register 0 (IICCTL0) is 0).

Reset signal generation sets this register to FFH.

Figure 15-11. Format of IICA High-Level Width Setting Register (IICWH)

Address: F0233H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
IICWH								

Remark For how to set the transfer clock by using the IICWL and IICWH registers, see **15.4.2 Setting transfer clock by using IICWL and IICWH registers**.

(8) Port mode register 6 (PM6)

This register sets the input/output of port 6 in 1-bit units.

When using the P60/SCL0 pin as clock I/O and the P61/SDA0 pin as serial data I/O, clear PM60 and PM61, and the output latches of P60 and P61 to 0.

Set IICE (bit 7 of IICA control register 0 (IICCTL0)) to 1 before setting the output mode because the P60/SCL0 and P61/SDA0 pins output a low level (fixed) when IICE is 0.

PM6 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 15-12. Format of Port Mode Register 6 (PM6)

Address: FFF26H After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM6	1	1	1	1	1	1	PM61	PM60

PM6n	P6n pin I/O mode selection (n = 0, 1)
0	Output mode (output buffer on)
1	Input mode (output buffer off)

15.4 I^2C Bus Mode Functions

15.4.1 Pin configuration

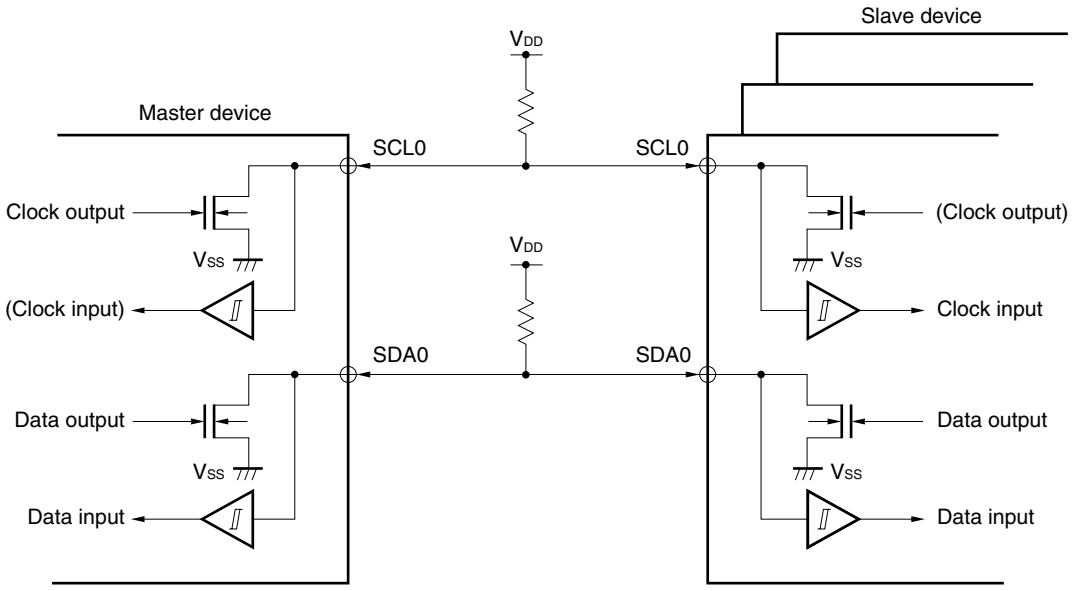
The serial clock pin (SCL0) and serial data bus pin (SDA0) are configured as follows.

(1) SCL0 ..... This pin is used for serial clock input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

(2) SDA0 ..... This pin is used for serial data input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Since outputs from the serial clock line and the serial data bus line are N-ch open-drain outputs, an external pull-up resistor is required.

Figure 15-13. Pin Configuration Diagram



<R> 15.4.2 Setting transfer clock by using IICWL and IICWH registers

(1) Setting transfer clock on master side

$$\text{Transfer clock} = \frac{f_{CLK}}{IICWL + IICWH + f_{CLK}(t_R + t_F)}$$

At this time, the optimal setting values of IICWL and IICWH are as follows.
(The fractional parts of all setting values are rounded up.)

- When the fast mode

$$IICWL = \frac{0.52}{\text{Transfer clock}} \times f_{CLK}$$
$$IICWH = (\frac{0.48}{\text{Transfer clock}} - t_R - t_F) \times f_{CLK}$$

- When the normal mode

$$IICWL = \frac{0.47}{\text{Transfer clock}} \times f_{CLK}$$
$$IICWH = (\frac{0.53}{\text{Transfer clock}} - t_R - t_F) \times f_{CLK}$$

Caution Note the minimum f_{CLK} operation frequency when setting the transfer clock. The minimum f_{CLK} operation frequency for serial interface IICA is determined according to the mode.
Fast mode: f_{CLK} = 3.5 MHz (MIN.)
Normal mode: f_{CLK} = 1 MHz (MIN.)

(2) Setting IICWL and IICWH on slave side

(The fractional parts of all setting values are truncated.)

- When the fast mode

$$IICWL = 1.3\ \mu s \times f_{CLK}$$
$$IICWH = (1.2\ \mu s - t_R - t_F) \times f_{CLK}$$

- When the normal mode

$$IICWL = 4.7\ \mu s \times f_{CLK}$$
$$IICWH = (5.3\ \mu s - t_R - t_F) \times f_{CLK}$$

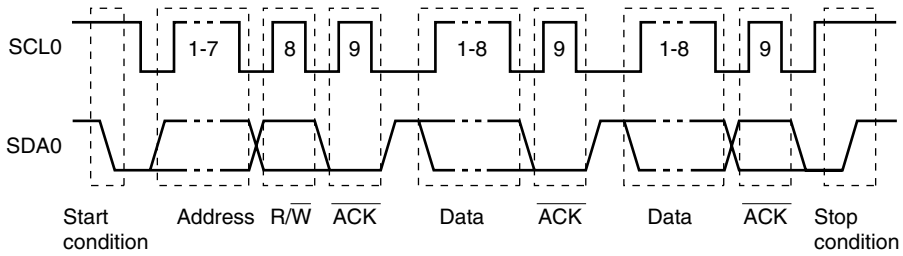
Remarks 1. Calculate the rise time (t_R) and fall time (t_F) of the SDA0 and SCL0 signals separately, because they differ depending on the pull-up resistance and wire load.

2. IICWL: IICA low-level width setting register
IICWH: IICA high-level width setting register
t_F: SDA0 and SCL0 signal falling times
t_R: SDA0 and SCL0 signal rising times
f_{CLK}: CPU/peripheral hardware clock frequency

15.5 I^2C Bus Definitions and Control Methods

The following section describes the I^2C bus's serial data communication format and the signals used by the I^2C bus. Figure 15-14 shows the transfer timing for the "start condition", "address", "data", and "stop condition" output via the I^2C bus's serial data bus.

Figure 15-14. I^2C Bus Serial Data Transfer Timing



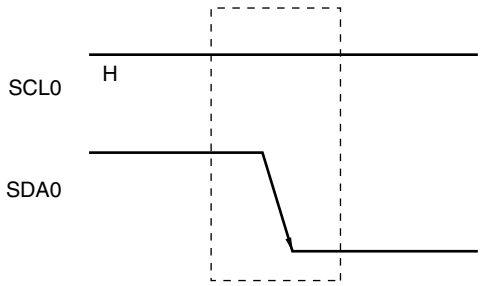
The master device generates the start condition, slave address, and stop condition.
The acknowledge ($\overline{ACK}$) can be generated by either the master or slave device (normally, it is output by the device that receives 8-bit data).

The serial clock (SCL0) is continuously output by the master device. However, in the slave device, the SCL0's low level period can be extended and a wait can be inserted.

15.5.1 Start conditions

A start condition is met when the SCL0 pin is at high level and the SDA0 pin changes from high level to low level. The start conditions for the SCL0 pin and SDA0 pin are signals that the master device generates to the slave device when starting a serial transfer. When the device is used as a slave, start conditions can be detected.

Figure 15-15. Start Conditions



A start condition is output when bit 1 (STT) of IICA control register 0 (IICCTL0) is set (1) after a stop condition has been detected (SPD: Bit 0 of the IICA status register (IICS) = 1). When a start condition is detected, bit 1 (STD) of IICS is set (1).

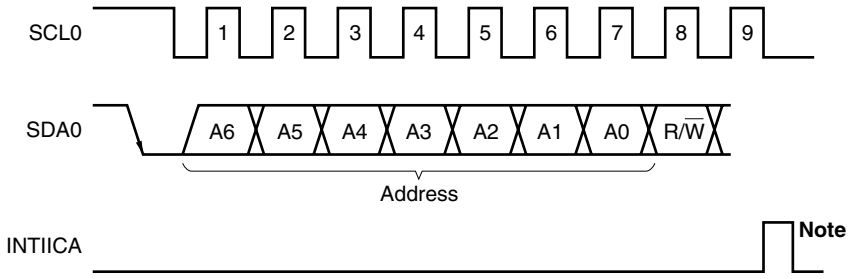
15.5.2 Addresses

The address is defined by the 7 bits of data that follow the start condition.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via the bus lines. Therefore, each slave device connected via the bus lines must have a unique address.

The slave devices include hardware that detects the start condition and checks whether or not the 7-bit address data matches the data values stored in the slave address register (SVA). If the address data matches the SVA values, the slave device is selected and communicates with the master device until the master device generates a start condition or stop condition.

Figure 15-16. Address



Note INTIICA is not issued if data other than a local address or extension code is received during slave device operation.

Addresses are output when a total of 8 bits consisting of the slave address and the transfer direction described in **15.5.3 Transfer direction specification** are written to the IICA shift register (IICA). The received addresses are written to IICA.

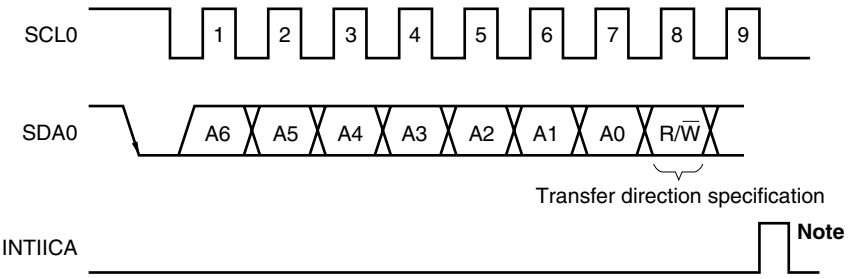
The slave address is assigned to the higher 7 bits of IICA.

15.5.3 Transfer direction specification

In addition to the 7-bit address data, the master device sends 1 bit that specifies the transfer direction.

When this transfer direction specification bit has a value of “0”, it indicates that the master device is transmitting data to a slave device. When the transfer direction specification bit has a value of “1”, it indicates that the master device is receiving data from a slave device.

Figure 15-17. Transfer Direction Specification



Note INTIICA is not issued if data other than a local address or extension code is received during slave device operation.

15.5.4 Acknowledge ($\overline{ACK}$)

$\overline{ACK}$ is used to check the status of serial data at the transmission and reception sides.

The reception side returns $\overline{ACK}$ each time it has received 8-bit data.

The transmission side usually receives $\overline{ACK}$ after transmitting 8-bit data. When $\overline{ACK}$ is returned from the reception side, it is assumed that reception has been correctly performed and processing is continued. Whether $\overline{ACK}$ has been detected can be checked by using bit 2 (ACKD) of the IICA status register (IICS).

When the master receives the last data item, it does not return $\overline{ACK}$ and instead generates a stop condition. If a slave does not return $\overline{ACK}$ after receiving data, the master outputs a stop condition or restart condition and stops transmission. If $\overline{ACK}$ is not returned, the possible causes are as follows.

<1> Reception was not performed normally.
<2> The final data item was received.
<3> The reception side specified by the address does not exist.

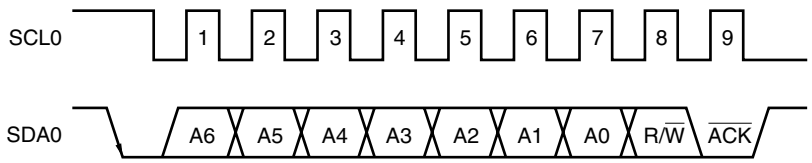
To generate $\overline{ACK}$, the reception side makes the SDA0 line low at the ninth clock (indicating normal reception).

Automatic generation of $\overline{ACK}$ is enabled by setting bit 2 (ACKE) of IICA control register 0 (IICCTL0) to 1. Bit 3 (TRC) of the IICS register is set by the data of the eighth bit that follows 7-bit address information. Usually, set ACKE to 1 for reception (TRC = 0).

If a slave can receive no more data during reception (TRC = 0) or does not require the next data item, then the slave must inform the master, by clearing ACKE to 0, that it will not receive any more data.

When the master does not require the next data item during reception (TRC = 0), it must clear ACKE to 0 so that $\overline{ACK}$ is not generated. In this way, the master informs a slave at the transmission side that it does not require any more data (transmission will be stopped).

Figure 15-18. $\overline{ACK}$



When the local address is received, $\overline{ACK}$ is automatically generated, regardless of the value of ACKE. When an address other than that of the local address is received, $\overline{ACK}$ is not generated (NACK).

When an extension code is received, $\overline{ACK}$ is generated if ACKE is set to 1 in advance.

How $\overline{ACK}$ is generated when data is received differs as follows depending on the setting of the wait timing.

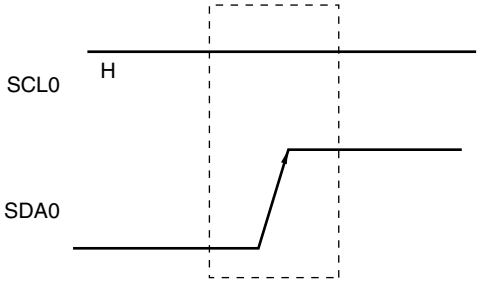
- When 8-clock wait state is selected (bit 3 (WTIM) of IICCTL0 register = 0):
  By setting ACKE to 1 before releasing the wait state, $\overline{ACK}$ is generated at the falling edge of the eighth clock of the SCL0 pin.
- When 9-clock wait state is selected (bit 3 (WTIM) of IICCTL0 register = 1):
  $\overline{ACK}$ is generated by setting ACKE to 1 in advance.

15.5.5 Stop condition

When the SCL0 pin is at high level, changing the SDA0 pin from low level to high level generates a stop condition.

A stop condition is a signal that the master device generates to the slave device when serial transfer has been completed. When the device is used as a slave, stop conditions can be detected.

Figure 15-19. Stop Condition



A stop condition is generated when bit 0 (SPT) of IICA control register 0 (IICCTL0) is set to 1. When the stop condition is detected, bit 0 (SPD) of the IICA status register (IICS) is set to 1 and INTIICA is generated when bit 4 (SPIE) of IICCTL0 is set to 1.

15.5.6 Wait

The wait is used to notify the communication partner that a device (master or slave) is preparing to transmit or receive data (i.e., is in a wait state).

Setting the SCL0 pin to low level notifies the communication partner of the wait state. When wait state has been canceled for both the master and slave devices, the next data transfer can begin.

Figure 15-20. Wait (1/2)

(1) When master device has a nine-clock wait and slave device has an eight-clock wait (master transmits, slave receives, and ACKE = 1)

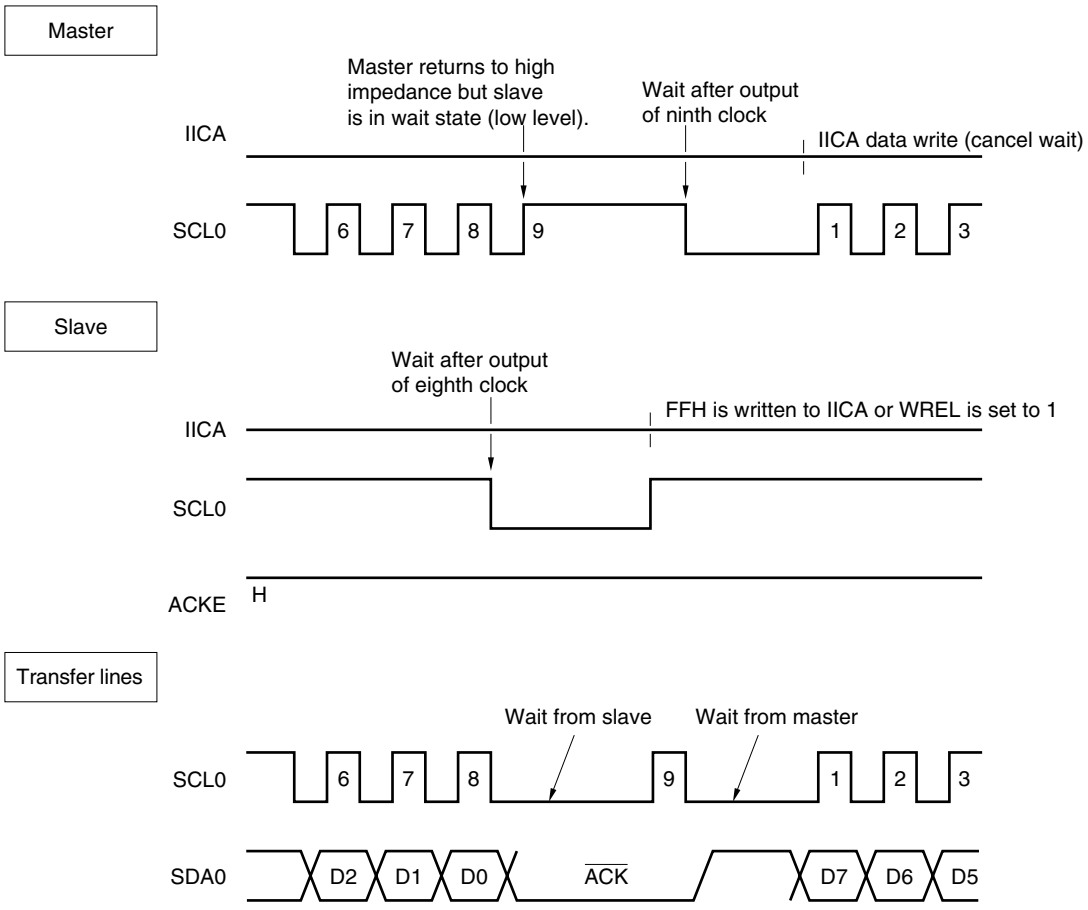
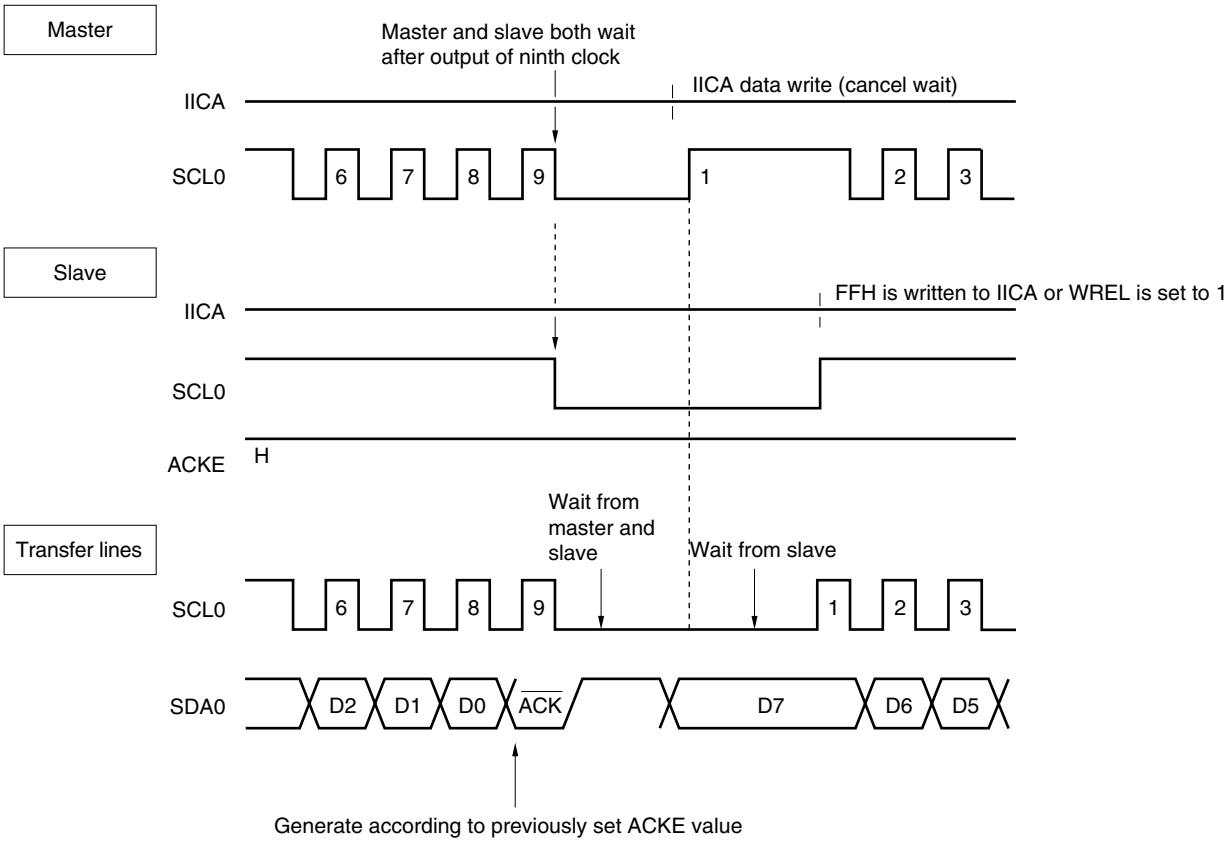


Figure 15-20. Wait (2/2)

(2) When master and slave devices both have a nine-clock wait (master transmits, slave receives, and ACKE = 1)



Remark ACKE: Bit 2 of IICA control register 0 (IICCTL0)
WREL: Bit 5 of IICA control register 0 (IICCTL0)

A wait may be automatically generated depending on the setting of bit 3 (WTIM) of IICA control register 0 (IICCTL0).

Normally, the receiving side cancels the wait state when bit 5 (WREL) of the IICCTL0 register is set to 1 or when FFH is written to the IICA shift register (IICA), and the transmitting side cancels the wait state when data is written to the IICA register.

- By setting bit 1 (STT) of IICCTL0 to 1
- By setting bit 0 (SPT) of IICCTL0 to 1

15.5.7 Canceling wait

The I^2C usually cancels a wait state by the following processing.

- Writing data to IICA shift register (IICA)
- Setting bit 5 (WREL) of IICA control register 0 (IICCTL0) (canceling wait)
- Setting bit 1 (STT) of IICCTL0 register (generating start condition)[Note]
- Setting bit 0 (SPT) of IICCTL0 register (generating stop condition)[Note]

Note Master only

When the above wait canceling processing is executed, the I^2C cancels the wait state and communication is resumed.

To cancel a wait state and transmit data (including addresses), write the data to IICA.

To receive data after canceling a wait state, or to complete data transmission, set bit 5 (WREL) of IICA control register 0 (IICCTL0) to 1.

To generate a restart condition after canceling a wait state, set bit 1 (STT) of IICCTL0 to 1.

To generate a stop condition after canceling a wait state, set bit 0 (SPT) of IICCTL0 to 1.

Execute the canceling processing only once for one wait state.

If, for example, data is written to IICA after canceling a wait state by setting WREL to 1, an incorrect value may be output to SDA0 because the timing for changing the SDA0 line conflicts with the timing for writing IICA.

In addition to the above, communication is stopped if IICE is cleared to 0 when communication has been aborted, so that the wait state can be canceled.

If the I^2C bus has deadlocked due to noise, processing is saved from communication by setting bit 6 (LREL) of IICCTL0, so that the wait state can be canceled.

Caution If a processing to cancel a wait state executed when WUP (bit 7 of IICA control register 1 (IICCTL1)) = 1, the wait state will not be canceled.

15.5.8 Interrupt request (INTIICA) generation timing and wait control

The setting of bit 3 (WTIM) of IICA control register 0 (IICCTL0) determines the timing by which INTIICA is generated and the corresponding wait control, as shown in Table 15-2.

Table 15-2. INTIICA Generation Timing and Wait Control

WTIM	During Slave Device Operation			During Master Device Operation		
	Address	Data Reception	Data Transmission	Address	Data Reception	Data Transmission
0	9 Notes 1, 2	8 Note 2	8 Note 2	9	8	8
1	9 Notes 1, 2	9 Note 2	9 Note 2	9	9	9

Notes 1. The slave device's INTIICA signal and wait period occurs at the falling edge of the ninth clock only when there is a match with the address set to the slave address register (SVA).
At this point, $\overline{ACK}$ is generated regardless of the value set to IICCTL0's bit 2 (ACKE). For a slave device that has received an extension code, INTIICA occurs at the falling edge of the eighth clock.
However, if the address does not match after restart, INTIICA is generated at the falling edge of the 9th clock, but wait does not occur.

2. If the received address does not match the contents of the slave address register (SVA) and extension code is not received, neither INTIICA nor a wait occurs.

Remark The numbers in the table indicate the number of the serial clock's clock signals. Interrupt requests and wait control are both synchronized with the falling edge of these clock signals.

(1) During address transmission/reception

- Slave device operation: Interrupt and wait timing are determined depending on the conditions described in Notes 1 and 2 above, regardless of the WTIM bit.
- Master device operation: Interrupt and wait timing occur at the falling edge of the ninth clock regardless of the WTIM bit.

(2) During data reception

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM bit.

(3) During data transmission

- Master/slave device operation: Interrupt and wait timing are determined according to the WTIM bit.

(4) Wait cancellation method

The four wait cancellation methods are as follows.

- Writing data to IICA shift register (IICA)
- Setting bit 5 (WREL) of IICA control register 0 (IICCTL0) (canceling wait)
- Setting bit 1 (STT) of IICCTL0 register (generating start condition)Note
- Setting bit 0 (SPT) of IICCTL0 register (generating stop condition)Note

Note Master only.

When an 8-clock wait has been selected (WTIM = 0), the presence/absence of $\overline{ACK}$ generation must be determined prior to wait cancellation.

(5) Stop condition detection

INTIICA is generated when a stop condition is detected (only when SPIE = 1).

15.5.9 Address match detection method

In I^2C bus mode, the master device can select a particular slave device by transmitting the corresponding slave address.

Address match can be detected automatically by hardware. An interrupt request (INTIICA) occurs when the address set to the slave address register (SVA) matches the slave address sent by the master device, or when an extension code has been received.

15.5.10 Error detection

In I^2C bus mode, the status of the serial data bus (SDA0) during data transmission is captured by the IICA shift register (IICA) of the transmitting device, so the IICA data prior to transmission can be compared with the transmitted IICA data to enable detection of transmission errors. A transmission error is judged as having occurred when the compared data values do not match.

15.5.11 Extension code

(1) When the higher 4 bits of the receive address are either “0000” or “1111”, the extension code reception flag (EXC) is set to 1 for extension code reception and an interrupt request (INTIICA) is issued at the falling edge of the eighth clock. The local address stored in the slave address register (SVA) is not affected.

(2) The settings below are specified if 11110xx0 is transferred from the master by using a 10-bit address transfer when the SVA register is set to 11110xx0. Note that INTIICA occurs at the falling edge of the eighth clock.

- Higher four bits of data match: EXC = 1
- Seven bits of data match: COI = 1

Remark EXC: Bit 5 of IICA status register (IICS)
COI: Bit 4 of IICA status register (IICS)

(3) Since the processing after the interrupt request occurs differs according to the data that follows the extension code, such processing is performed by software.
If the extension code is received while a slave device is operating, then the slave device is participating in communication even if its address does not match.
For example, after the extension code is received, if you do not wish to operate the target device as a slave device, set bit 6 (LREL) of the IICA control register 0 (IICCTL0) to 1 to set the standby mode for the next communication operation.

Table 15-3. Bit Definitions of Main Extension Code

Slave Address	$R/\overline{W}$ Bit	Description
0 0 0 0 0 0 0	0	General call address
1 1 1 1 0 x x	0	10-bit slave address specification (for address authentication)
1 1 1 1 0 x x	1	10-bit slave address specification (for read command issuance after address match)

Remark For extension codes other than the above, refer to THE I^2C-BUS SPECIFICATION published by NXP.

15.5.12 Arbitration

When several master devices simultaneously generate a start condition (when STT is set to 1 before STD is set to 1), communication among the master devices is performed as the number of clocks are adjusted until the data differs. This kind of operation is called arbitration.

When one of the master devices loses in arbitration, an arbitration loss flag (ALD) in the IICA status register (IICS) is set (1) via the timing by which the arbitration loss occurred, and the SCL0 and SDA0 lines are both set to high impedance, which releases the bus.

The arbitration loss is detected based on the timing of the next interrupt request (the eighth or ninth clock, when a stop condition is detected, etc.) and the ALD = 1 setting that has been made by software.

For details of interrupt request timing, see **15.5.8 Interrupt request (INTIICA) generation timing and wait control**.

Remark STD: Bit 1 of IICA status register (IICS)
STT: Bit 1 of IICA control register 0 (IICCTL0)

Figure 15-21. Arbitration Timing Example

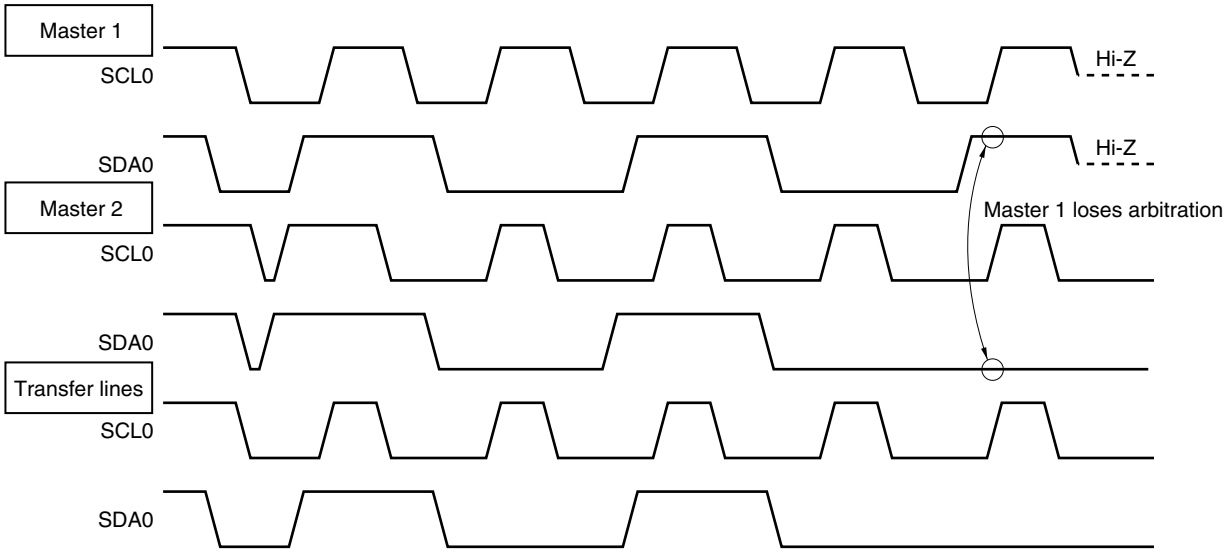


Table 15-4. Status During Arbitration and Interrupt Request Generation Timing

Status During Arbitration	Interrupt Request Generation Timing
During address transmission	At falling edge of eighth or ninth clock following byte transfer[Note 1]
Read/write data after address transmission	
During extension code transmission	
Read/write data after extension code transmission	
During data transmission	
During $\overline{\text{ACK}}$ transfer period after data transmission	
When restart condition is detected during data transfer	
When stop condition is detected during data transfer	When stop condition is generated (when SPIE = 1)[Note 2]
When data is at low level while attempting to generate a restart condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When stop condition is detected while attempting to generate a restart condition	When stop condition is generated (when SPIE = 1)[Note 2]
When data is at low level while attempting to generate a stop condition	At falling edge of eighth or ninth clock following byte transfer[Note 1]
When SCL0 is at low level while attempting to generate a restart condition	

Notes 1. When WTIM (bit 3 of IICA control register 0 (IICCTL0)) = 1, an interrupt request occurs at the falling edge of the ninth clock. When WTIM = 0 and the extension code's slave address is received, an interrupt request occurs at the falling edge of the eighth clock.

2. When there is a chance that arbitration will occur, set SPIE = 1 for master device operation.

Remark SPIE: Bit 4 of IICA control register 0 (IICCTL0)

15.5.13 Wakeup function

The I^2C bus slave function is a function that generates an interrupt request signal (INTIICA) when a local address and extension code have been received.

This function makes processing more efficient by preventing unnecessary INTIICA signal from occurring when addresses do not match.

When a start condition is detected, wakeup standby mode is set. This wakeup standby mode is in effect while addresses are transmitted due to the possibility that an arbitration loss may change the master device (which has generated a start condition) to a slave device.

However, when a stop condition is detected, bit 4 (SPIE) of IICA control register 0 (IICCTL0) is set regardless of the wakeup function, and this determines whether interrupt requests are enabled or disabled.

To use the wakeup function in the STOP mode, set WUP to 1. Addresses can be received regardless of the operation clock. An interrupt request signal (INTIICA) is also generated when a local address and extension code have been received. Operation returns to normal operation by using an instruction to clear (0) the WUP bit after this interrupt has been generated.

Figure 15-22 shows the flow for setting WUP = 1 and Figure 15-23 shows the flow for setting WUP = 0 upon an address match.

Figure 15-22. Flow When Setting WUP = 1

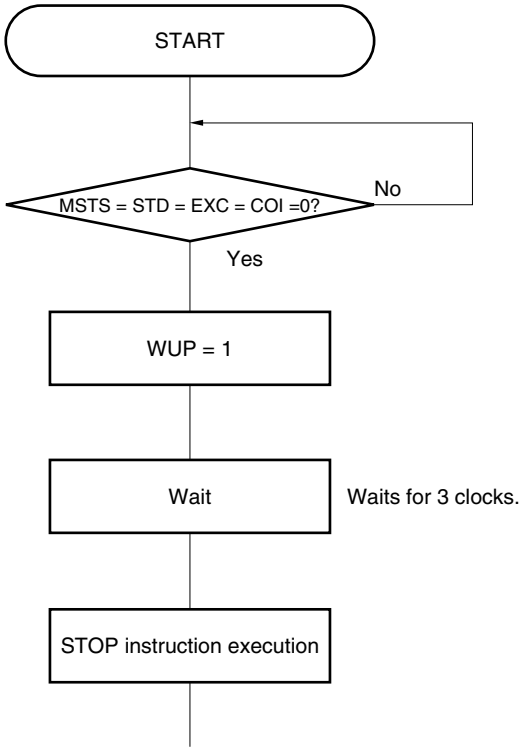
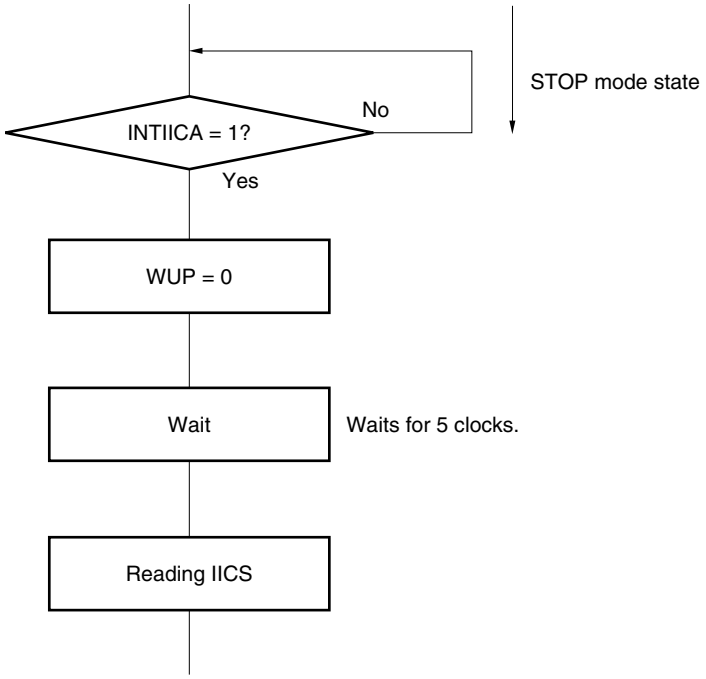


Figure 15-23. Flow When Setting WUP = 0 upon Address Match (Including Extension Code Reception)

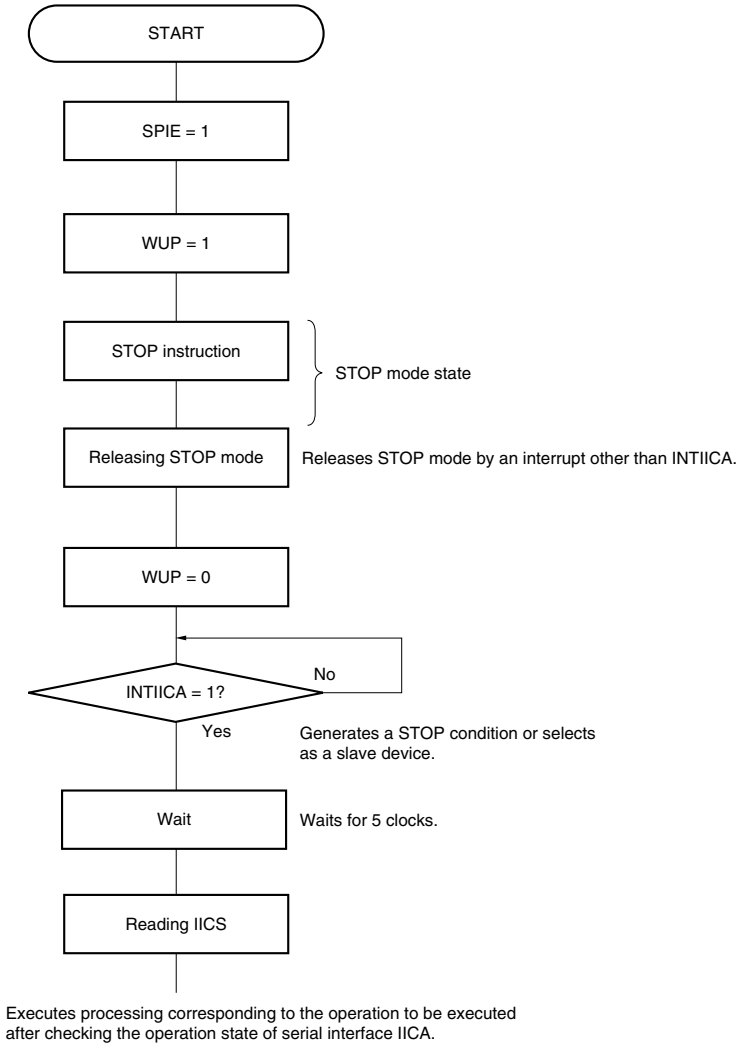


Executes processing corresponding to the operation to be executed after checking the operation state of serial interface IICA.

Use the following flows to perform the processing to release the STOP mode other than by an interrupt request (INTIICA) generated from serial interface IICA.

- Master device operation: Flow shown in Figure 15-24
- Slave device operation: Same as the flow in Figure 15-23

Figure 15-24. When Operating as Master Device after Releasing STOP Mode other than by INTIICA



15.5.14 Communication reservation

(1) When communication reservation function is enabled (bit 0 (IICRSV) of IICA flag register (IICF) = 0)

To start master device communications when not currently using a bus, a communication reservation can be made to enable transmission of a start condition when the bus is released. There are two modes under which the bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{ACK}$ is not returned and the bus was released by setting bit 6 (LREL) of IICA control register 0 (IICCTL0) to 1 and saving communication).

If bit 1 (STT) of IICCTL0 is set to 1 while the bus is not used (after a stop condition is detected), a start condition is automatically generated and wait state is set.
If an address is written to the IICA shift register (IICA) after bit 4 (SPIE) of IICCTL0 was set to 1, and it was detected by generation of an interrupt request signal (INTIICA) that the bus was released (detection of the stop condition), then the device automatically starts communication as the master. Data written to IICA before the stop condition is detected is invalid.
When STT has been set to 1, the operation mode (as start condition or as communication reservation) is determined according to the bus status.

- If the bus has been released.....a start condition is generated
- If the bus has not been released (standby mode) .....communication reservation

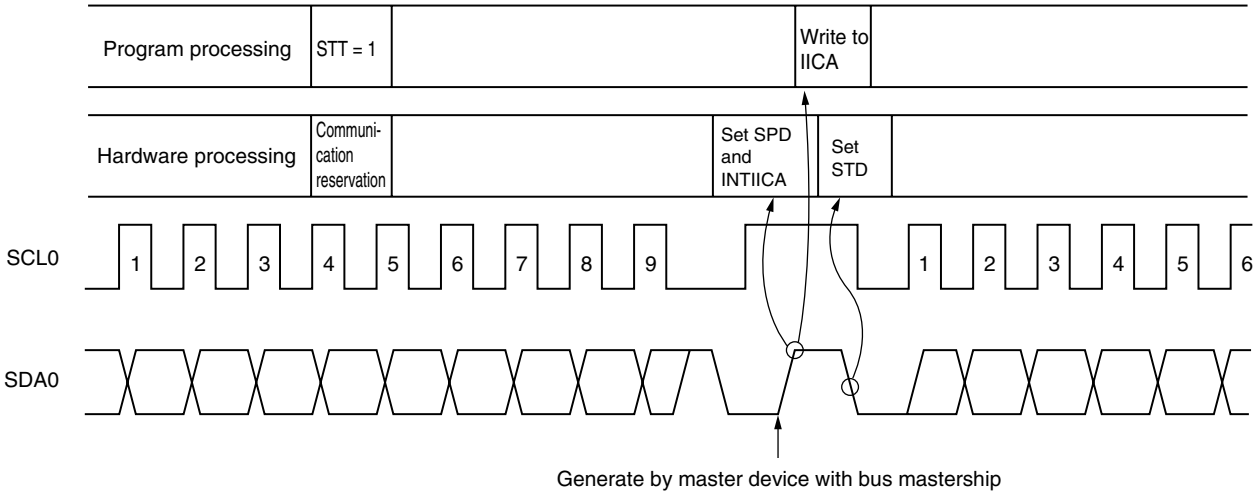
Check whether the communication reservation operates or not by using MSTS (bit 7 of the IICA status register (IICS)) after STT is set to 1 and the wait time elapses.
Use software to secure the wait time calculated by the following expression.

Wait time from setting STT = 1 to checking the MSTS flag: (IICWL setting value + IICWH setting value + 4 clocks) / f_{CLK} + $t_F \times 2$

Remark IICWL: IICA low-level width setting register
IICWH: IICA high-level width setting register
t_F: SDA0 and SCL0 signal falling times
f_{CLK}: CPU/peripheral hardware clock frequency

Figure 15-25 shows the communication reservation timing.

Figure 15-25. Communication Reservation Timing



Remark IICA: IICA shift register
STT: Bit 1 of IICA control register 0 (IICCTL0)
STD: Bit 1 of IICA status register (IICS)
SPD: Bit 0 of IICA status register (IICS)

Communication reservations are accepted via the timing shown in Figure 15-26. After bit 1 (STD) of the IICA status register (IICS) is set to 1, a communication reservation can be made by setting bit 1 (STT) of IICA control register 0 (IICCTL0) to 1 before a stop condition is detected.

Figure 15-26. Timing for Accepting Communication Reservations

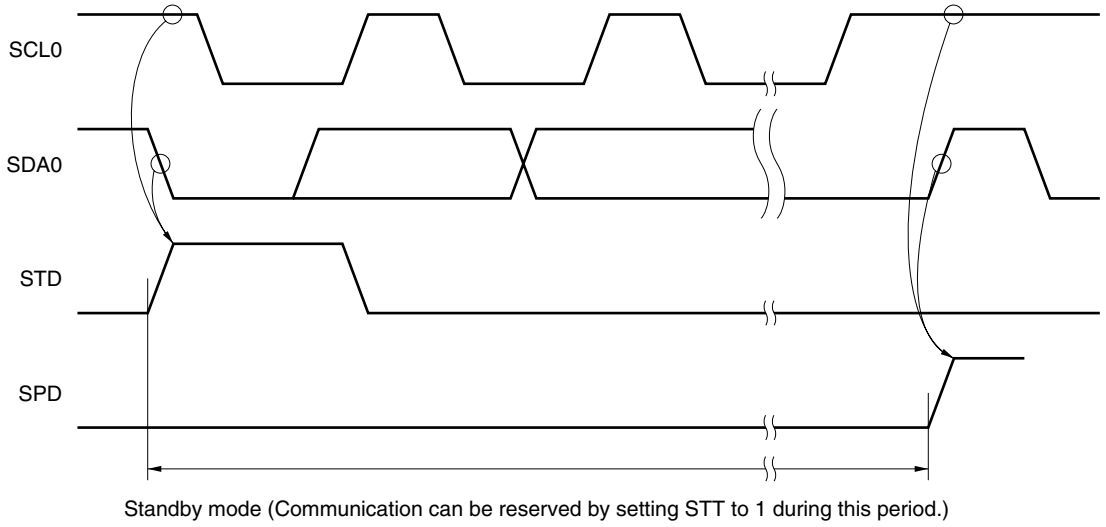
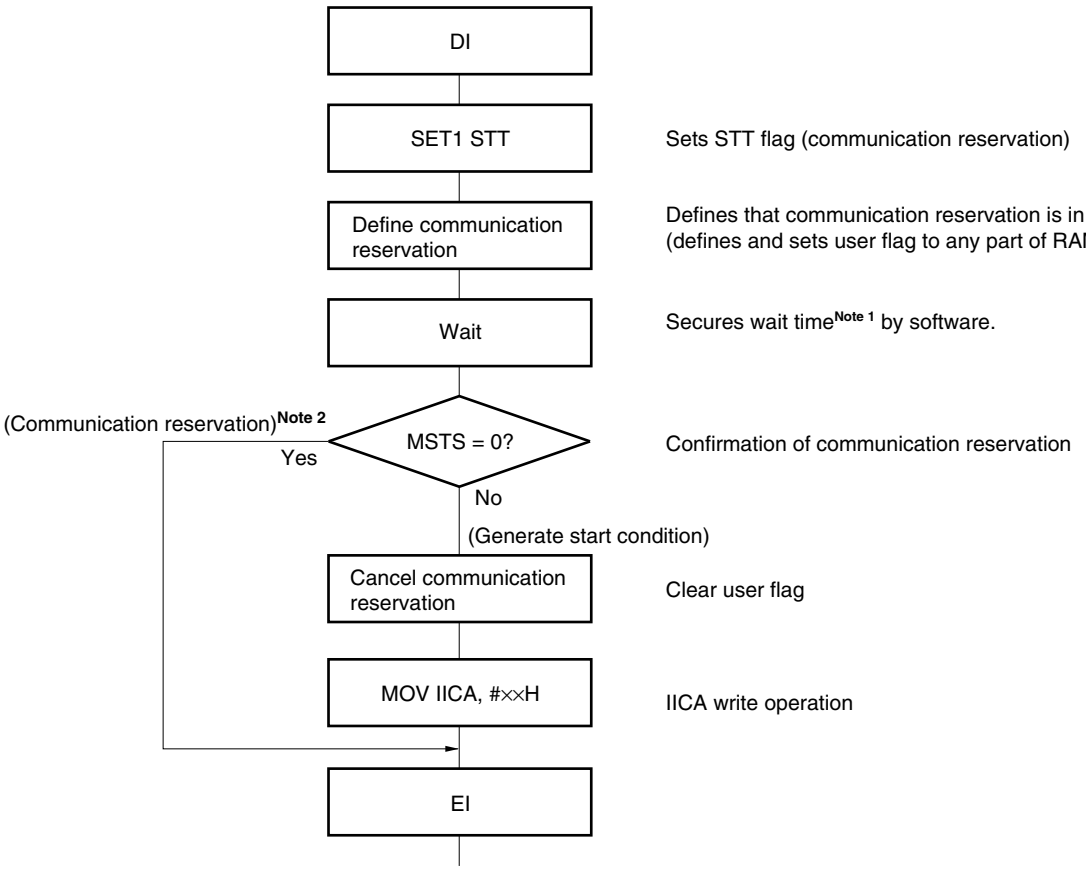


Figure 15-27 shows the communication reservation protocol.

Figure 15-27. Communication Reservation Protocol



Notes 1. The wait time is calculated as follows.
(IICWL setting value + IICWH setting value + 4 clocks) / f_{CLK} + $t_F \times 2$

2. The communication reservation operation executes a write to the IICA shift register (IICA) when a stop condition interrupt request occurs.

Remark STT: Bit 1 of IICA control register 0 (IICCTL0)
MSTS: Bit 7 of IICA status register (IICS)
IICA: IICA shift register
IICWL: IICA low-level width setting register
IICWH: IICA high-level width setting register
t_F: SDA0 and SCL0 signal falling times
f_{CLK}: CPU/peripheral hardware clock frequency

(2) When communication reservation function is disabled (bit 0 (IICRSV) of IICA flag register (IICF) = 1)

When bit 1 (STT) of IICA control register 0 (IICCTL0) is set to 1 when the bus is not used in a communication during bus communication, this request is rejected and a start condition is not generated. The following two statuses are included in the status where bus is not used.

- When arbitration results in neither master nor slave operation
- When an extension code is received and slave operation is disabled ($\overline{ACK}$ is not returned and the bus was released by setting bit 6 (LREL) of IICCTL0 to 1 and saving communication)

To confirm whether the start condition was generated or request was rejected, check STCF (bit 7 of IICF). It takes up to 5 clocks until STCF is set to 1 after setting STT = 1. Therefore, secure the time by software.

15.5.15 Cautions

(1) When STCEN = 0

Immediately after I^2C operation is enabled (IICE = 1), the bus communication status (IICBSY = 1) is recognized regardless of the actual bus status. When changing from a mode in which no stop condition has been detected to a master device communication mode, first generate a stop condition to release the bus, then perform master device communication.

When using multiple masters, it is not possible to perform master device communication when the bus has not been released (when a stop condition has not been detected).

Use the following sequence for generating a stop condition.

<1> Set IICA control register 1 (IICCTL1).
<2> Set bit 7 (IICE) of IICA control register 0 (IICCTL0) to 1.
<3> Set bit 0 (SPT) of IICCTL0 to 1.

(2) When STCEN = 1

Immediately after I^2C operation is enabled (IICE = 1), the bus released status (IICBSY = 0) is recognized regardless of the actual bus status. To generate the first start condition (STT = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.

(3) If other I^2C communications are already in progress

If I^2C operation is enabled and the device participates in communication already in progress when the SDA0 pin is low and the SCL0 pin is high, the macro of I^2C recognizes that the SDA0 pin has gone low (detects a start condition). If the value on the bus at this time can be recognized as an extension code, $\overline{ACK}$ is returned, but this interferes with other I^2C communications. To avoid this, start I^2C in the following sequence.

<1> Clear bit 4 (SPIE) of IICCTL0 to 0 to disable generation of an interrupt request signal (INTIICA) when the stop condition is detected.
<2> Set bit 7 (IICE) of IICCTL0 to 1 to enable the operation of I^2C.
<3> Wait for detection of the start condition.
<4> Set bit 6 (LREL) of IICCTL0 to 1 before $\overline{ACK}$ is returned (4 to 80 clocks after setting IICE to 1), to forcibly disable detection.

(4) Setting STT and SPT (bits 1 and 0 of IICCTL0) again after they are set and before they are cleared to 0 is prohibited.

(5) When transmission is reserved, set SPIE (bit 4 of IICTL0) to 1 so that an interrupt request is generated when the stop condition is detected. Transfer is started when communication data is written to IICA after the interrupt request is generated. Unless the interrupt is generated when the stop condition is detected, the device stops in the wait state because the interrupt request is not generated when communication is started. However, it is not necessary to set SPIE to 1 when MSTS (bit 7 of IICS) is detected by software.

15.5.16 Communication operations

The following shows three operation procedures with the flowchart.

(1) Master operation in single master system

The flowchart when using the 78K0R/KE3-A microcontroller as the master in a single master system is shown below.

This flowchart is broadly divided into the initial settings and communication processing. Execute the initial settings at startup. If communication with the slave is required, prepare the communication and then execute communication processing.

(2) Master operation in multimaster system

In the I^2C bus multimaster system, whether the bus is released or used cannot be judged by the I^2C bus specifications when the bus takes part in a communication. Here, when data and clock are at a high level for a certain period (1 frame), the 78K0R/KE3-A microcontroller takes part in a communication with bus released state.

This flowchart is broadly divided into the initial settings, communication waiting, and communication processing. The processing when the 78K0R/KE3-A microcontroller looses in arbitration and is specified as the slave is omitted here, and only the processing as the master is shown. Execute the initial settings at startup to take part in a communication. Then, wait for the communication request as the master or wait for the specification as the slave. The actual communication is performed in the communication processing, and it supports the transmission/reception with the slave and the arbitration with other masters.

(3) Slave operation

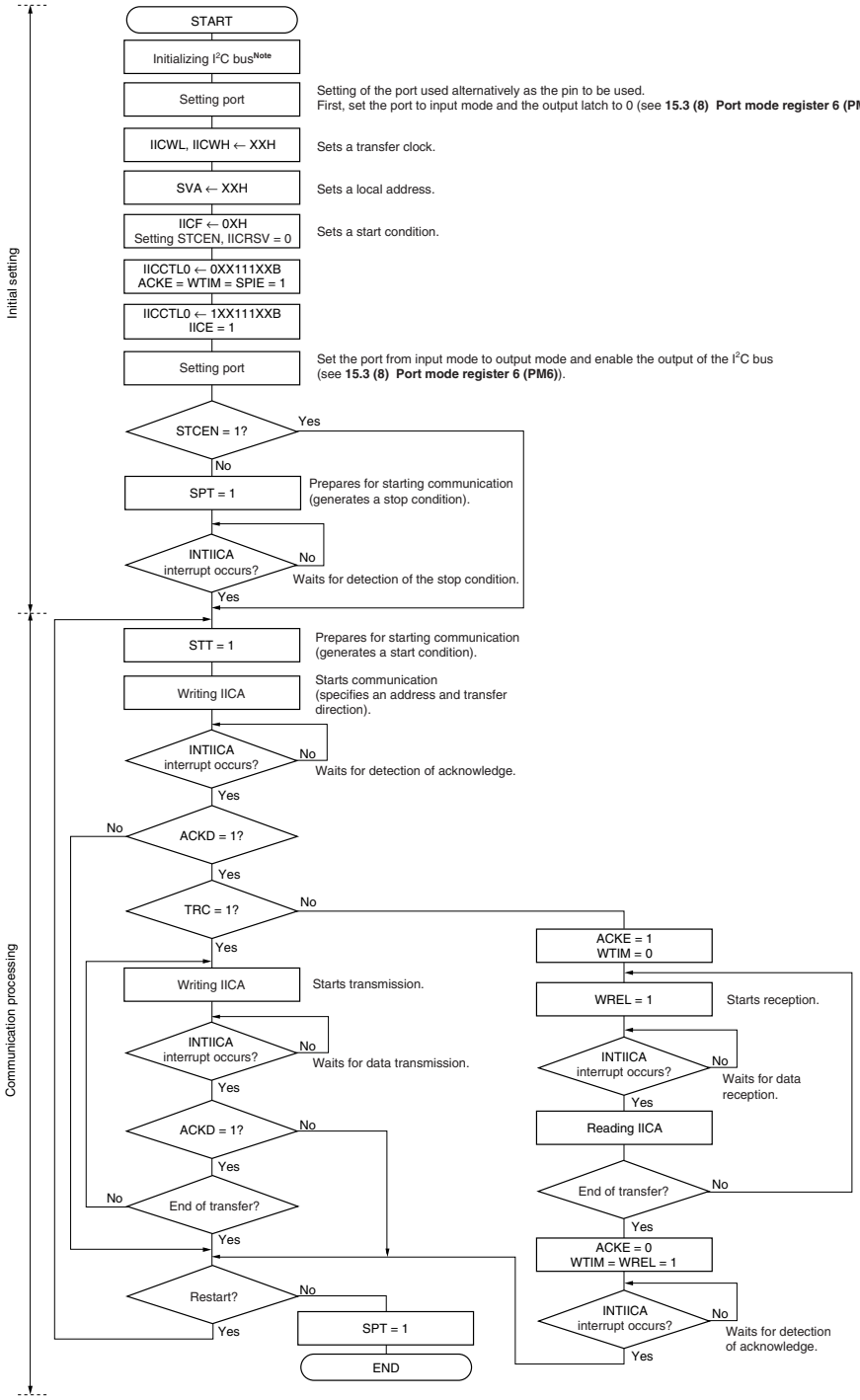
An example of when the 78K0R/KE3-A microcontroller is used as the I^2C bus slave is shown below.

When used as the slave, operation is started by an interrupt. Execute the initial settings at startup, then wait for the INTIICA interrupt occurrence (communication waiting). When an INTIICA interrupt occurs, the communication status is judged and its result is passed as a flag over to the main processing.

By checking the flags, necessary communication processing is performed.

(1) Master operation in single-master system

Figure 15-28. Master Operation in Single-Master System

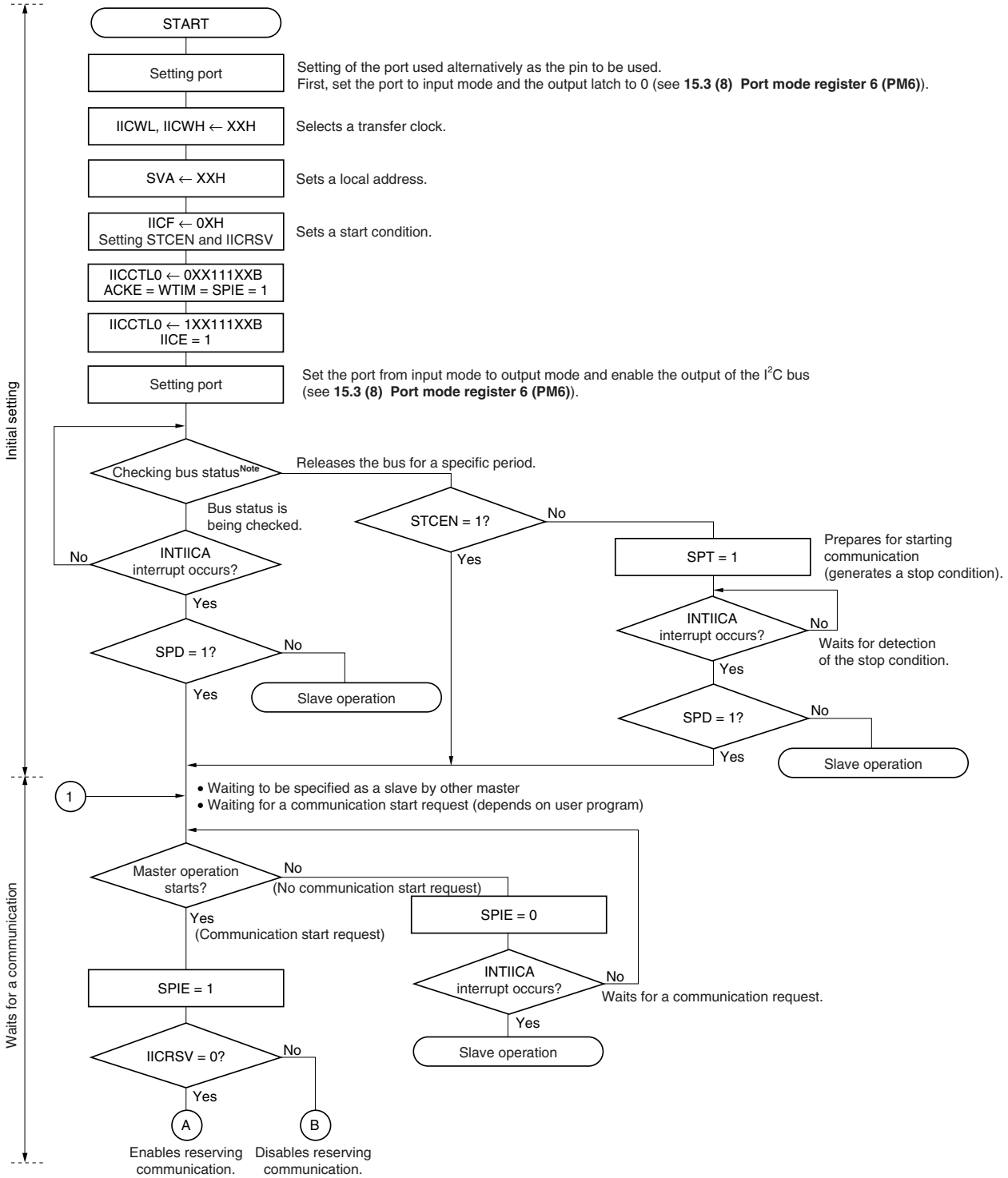


Note Release (SCL0 and SDA0 pins = high level) the I^2C bus in conformance with the specifications of the product that is communicating. If EEPROM is outputting a low level to the SDA0 pin, for example, set the SCL0 pin in the output port mode, and output a clock pulse from the output port until the SDA0 pin is constantly at high level.

Remark Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

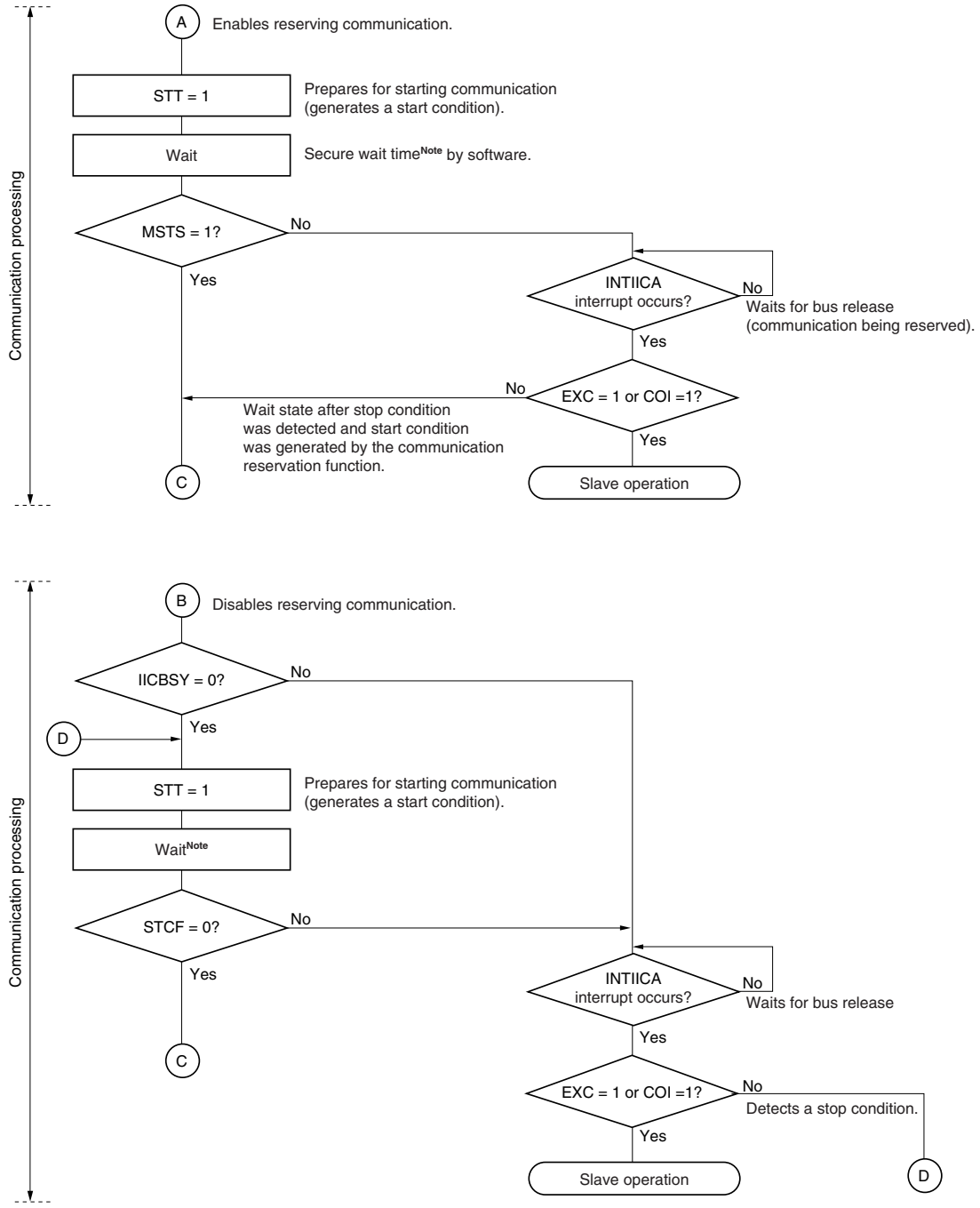
(2) Master operation in multi-master system

Figure 15-29. Master Operation in Multi-Master System (1/3)



Note Confirm that the bus is released (CLD bit = 1, DAD bit = 1) for a specific period (for example, for a period of one frame). If the SDA0 pin is constantly at low level, decide whether to release the I²C bus (SCL0 and SDA0 pins = high level) in conformance with the specifications of the product that is communicating.

Figure 15-29. Master Operation in Multi-Master System (2/3)



Note The wait time is calculated as follows.

(IICWL setting value + IICWH setting value + 4 clocks) / f_{CLK} + $t_F \times 2$

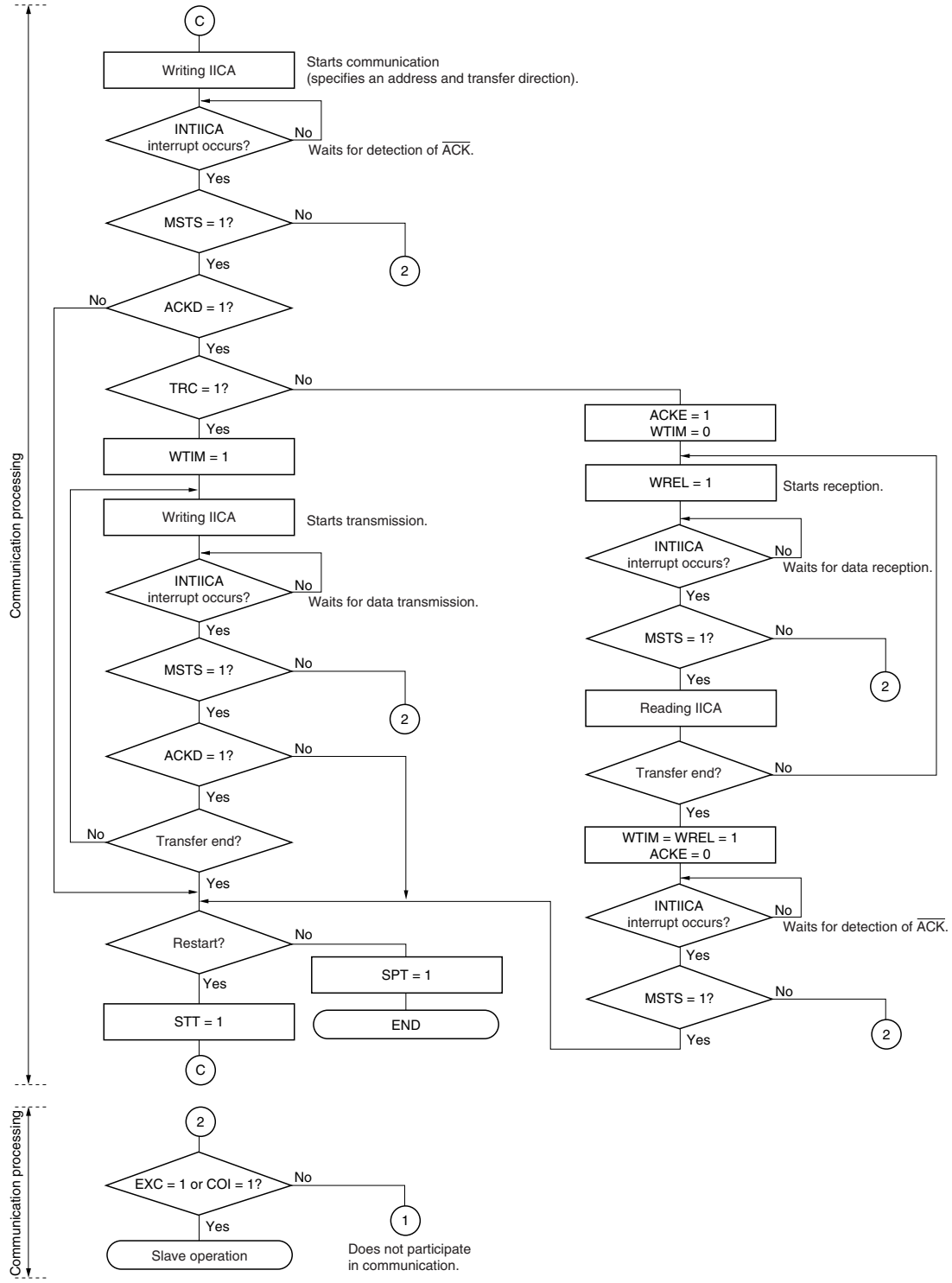
Remark IICWL: IICA low-level width setting register

IICWH: IICA high-level width setting register

t_F: SDA0 and SCL0 signal falling times

f_{CLK}: CPU/peripheral hardware clock frequency

Figure 15-29. Master Operation in Multi-Master System (3/3)



Remarks 1. Conform to the specifications of the product that is communicating, with respect to the transmission and reception formats.

2. To use the device as a master in a multi-master system, read the MSTS bit each time interrupt INTIICA has occurred to check the arbitration result.

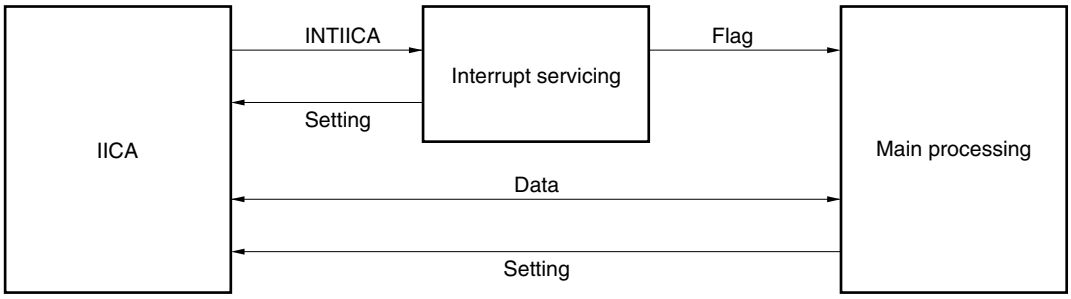
3. To use the device as a slave in a multi-master system, check the status by using the IICS and IICF registers each time interrupt INTIICA has occurred, and determine the processing to be performed next.

(3) Slave operation

The processing procedure of the slave operation is as follows.

Basically, the slave operation is event-driven. Therefore, processing by the INTIICA interrupt (processing that must substantially change the operation status such as detection of a stop condition during communication) is necessary.

In the following explanation, it is assumed that the extension code is not supported for data communication. It is also assumed that the INTIICA interrupt servicing only performs status transition processing, and that actual data communication is performed by the main processing.



Therefore, data communication processing is performed by preparing the following three flags and passing them to the main processing instead of INTIICA.

<1> Communication mode flag

This flag indicates the following two communication statuses.

- Clear mode: Status in which data communication is not performed
- Communication mode: Status in which data communication is performed (from valid address detection to stop condition detection, no detection of $\overline{\text{ACK}}$ from master, address mismatch)

<2> Ready flag

This flag indicates that data communication is enabled. Its function is the same as the INTIICA interrupt for ordinary data communication. This flag is set by interrupt servicing and cleared by the main processing. Clear this flag by interrupt servicing when communication is started. However, the ready flag is not set by interrupt servicing when the first data is transmitted. Therefore, the first data is transmitted without the flag being cleared (an address match is interpreted as a request for the next data).

<3> Communication direction flag

This flag indicates the direction of communication. Its value is the same as TRC.

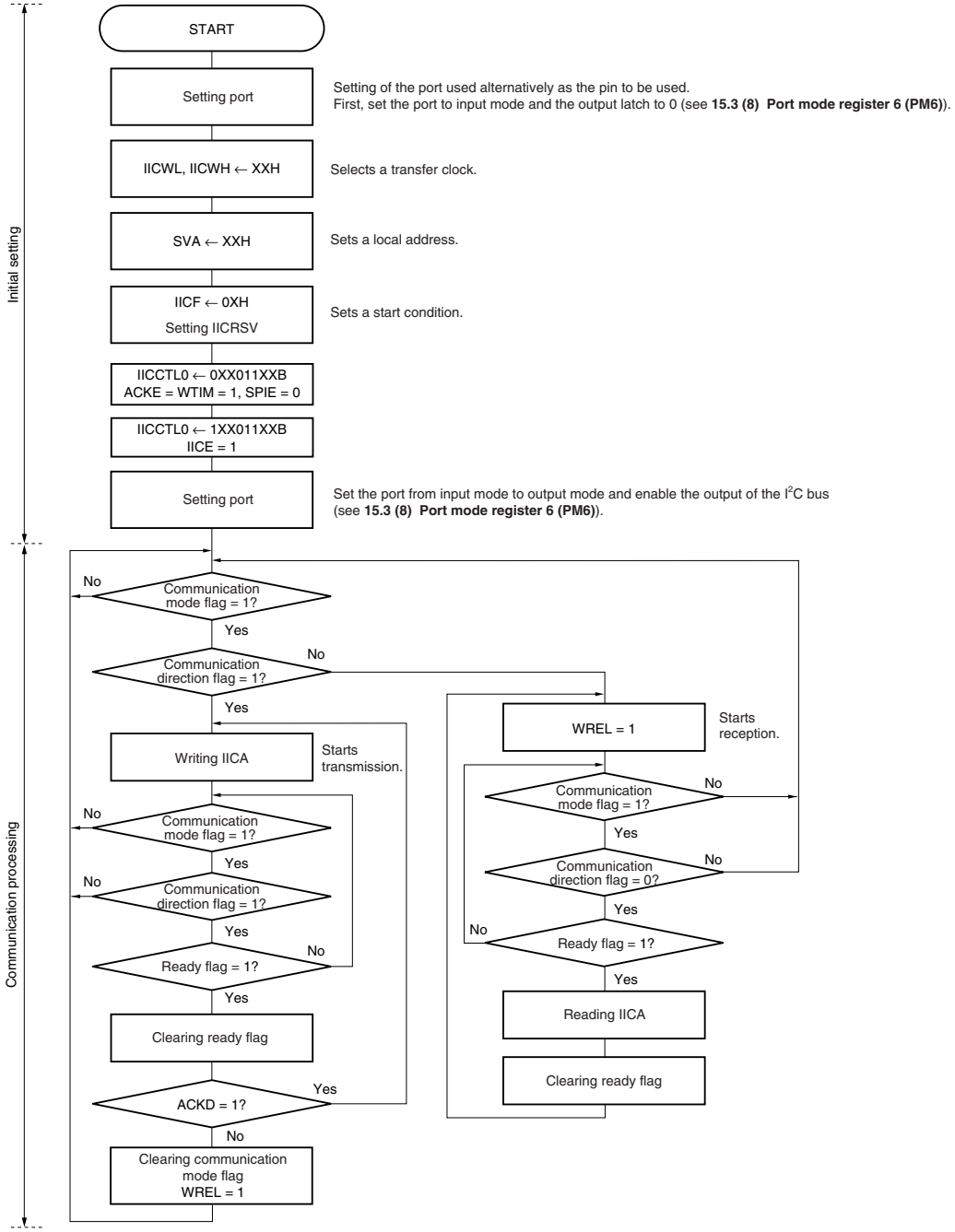
The main processing of the slave operation is explained next.

Start serial interface IICA and wait until communication is enabled. When communication is enabled, execute communication by using the communication mode flag and ready flag (processing of the stop condition and start condition is performed by an interrupt. Here, check the status by using the flags).

The transmission operation is repeated until the master no longer returns $\overline{ACK}$. If $\overline{ACK}$ is not returned from the master, communication is completed.

For reception, the necessary amount of data is received. When communication is completed, $\overline{ACK}$ is not returned as the next data. After that, the master generates a stop condition or restart condition. Exit from the communication status occurs in this way.

Figure 15-30. Slave Operation Flowchart (1)



Remark Conform to the specifications of the product that is in communication, regarding the transmission and reception formats.

An example of the processing procedure of the slave with the INTIICA interrupt is explained below (processing is performed assuming that no extension code is used). The INTIICA interrupt checks the status, and the following operations are performed.

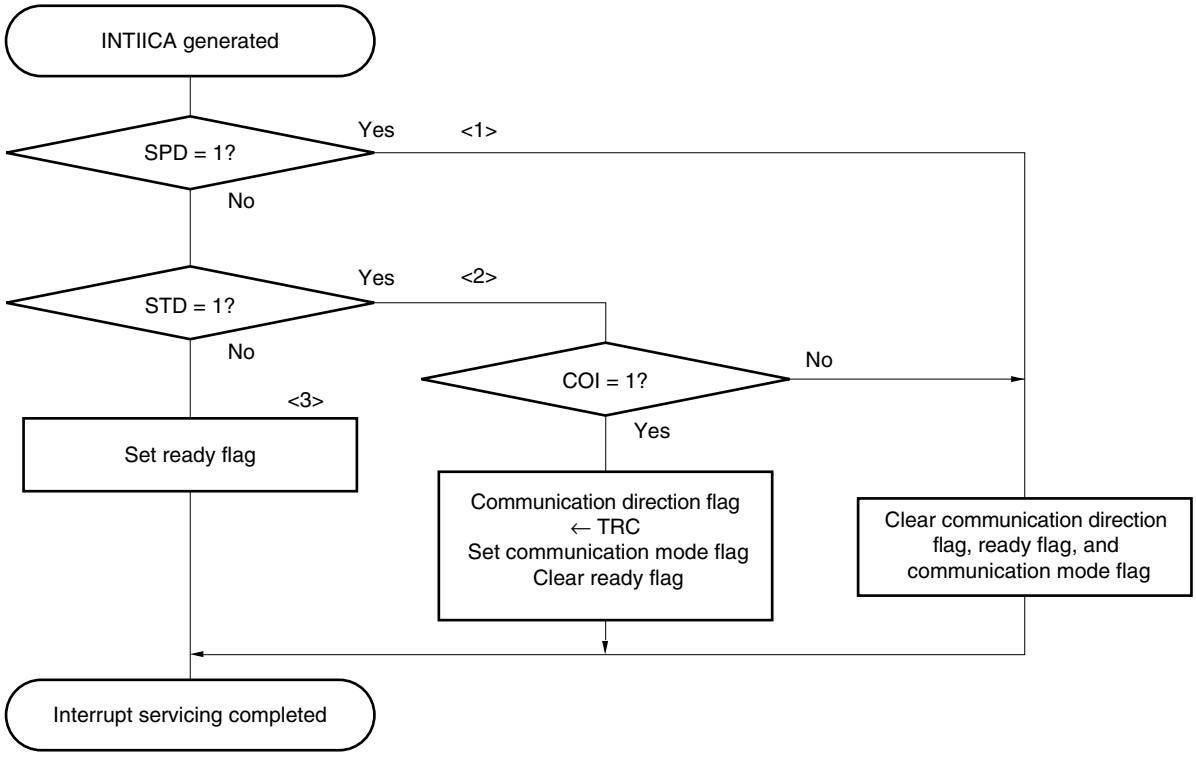
<1> Communication is stopped if the stop condition is issued.

<2> If the start condition is issued, the address is checked and communication is completed if the address does not match. If the address matches, the communication mode is set, wait is cancelled, and processing returns from the interrupt (the ready flag is cleared).

<3> For data transmit/receive, only the ready flag is set. Processing returns from the interrupt with the I^2C bus remaining in the wait state.

Remark <1> to <3> above correspond to <1> to <3> in Figure 15-31 Slave Operation Flowchart (2).

Figure 15-31. Slave Operation Flowchart (2)



15.5.17 Timing of I²C interrupt request (INTIICA) occurrence

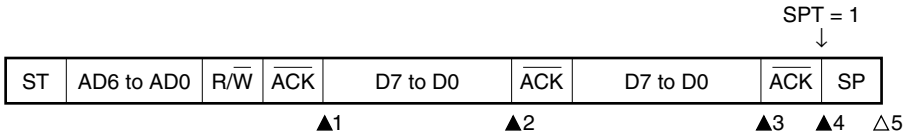
The timing of transmitting or receiving data and generation of interrupt request signal INTIICA, and the value of the IICS register when the INTIICA signal is generated are shown below.

Remark ST: Start condition
AD6 to AD0: Address
R/$\overline{W}$: Transfer direction specification
$\overline{ACK}$: Acknowledge
D7 to D0: Data
SP: Stop condition

(1) Master device operation

(a) Start ~ Address ~ Data ~ Data ~ Stop (transmission/reception)

(i) When WTIM = 0

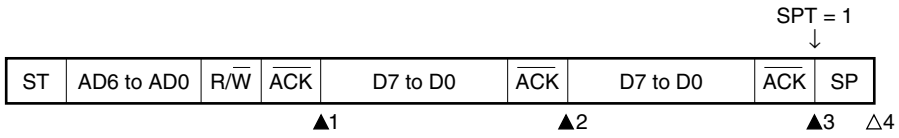


▲1: IICS = 1000×110B
▲2: IICS = 1000×000B
▲3: IICS = 1000×000B (Sets WTIM to 1)[Note]
▲4: IICS = 1000××00B (Sets SPT to 1)[Note]
△5: IICS = 00000001B

Note To generate a stop condition, set WTIM to 1 and change the timing for generating the INTIICA interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1

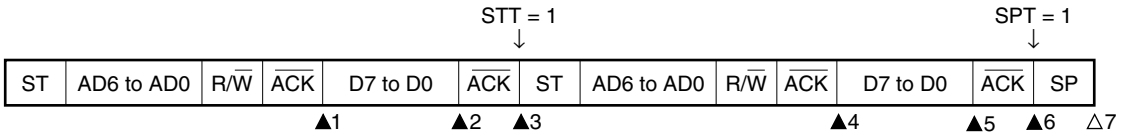


▲1: IICS = 1000×110B
▲2: IICS = 1000×100B
▲3: IICS = 1000××00B (Sets SPT to 1)
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

(i) When WTIM = 0

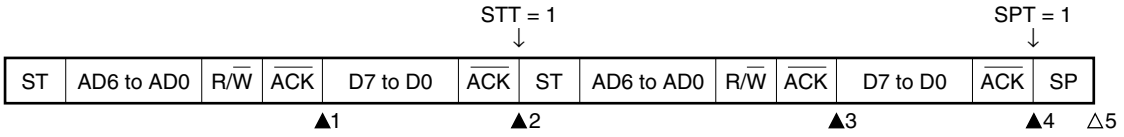


▲1: IICS = 1000×110B
▲2: IICS = 1000×000B (Sets WTIM to 1)[Note 1]
▲3: IICS = 1000××00B (Clears WTIM to 0[Note 2], sets STT to 1)
▲4: IICS = 1000×110B
▲5: IICS = 1000×000B (Sets WTIM to 1)[Note 3]
▲6: IICS = 1000××00B (Sets SPT to 1)
△7: IICS = 00000001B

Notes 1. To generate a start condition, set WTIM to 1 and change the timing for generating the INTIICA interrupt request signal.
2. Clear WTIM to 0 to restore the original setting.
3. To generate a stop condition, set WTIM to 1 and change the timing for generating the INTIICA interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1

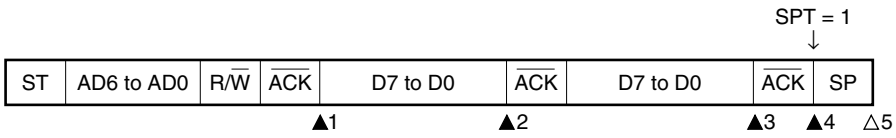


▲1: IICS = 1000×110B
▲2: IICS = 1000××00B (Sets STT to 1)
▲3: IICS = 1000×110B
▲4: IICS = 1000××00B (Sets SPT to 1)
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

(i) When WTIM = 0

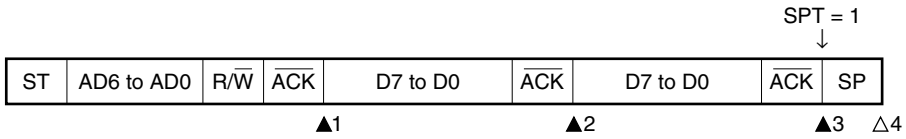


▲1: IICS = 1010×110B
▲2: IICS = 1010×000B
▲3: IICS = 1010×000B (Sets WTIM to 1)[Note]
▲4: IICS = 1010××00B (Sets SPT to 1)
△5: IICS = 00000001B

Note To generate a stop condition, set WTIM to 1 and change the timing for generating the INTIICA interrupt request signal.

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1



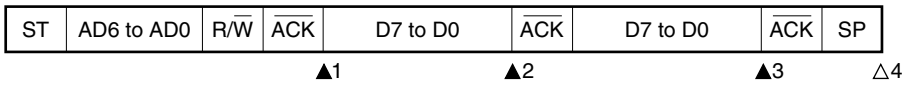
▲1: IICS = 1010×110B
▲2: IICS = 1010×100B
▲3: IICS = 1010××00B (Sets SPT to 1)
△4: IICS = 00001001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(2) Slave device operation (slave address data reception)

(a) Start ~ Address ~ Data ~ Data ~ Stop

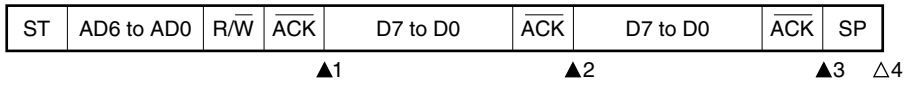
(i) When WTIM = 0



▲1: IICS = 0001×110B
▲2: IICS = 0001×000B
▲3: IICS = 0001×000B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1



▲1: IICS = 0001×110B
▲2: IICS = 0001×100B
▲3: IICS = 0001××00B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

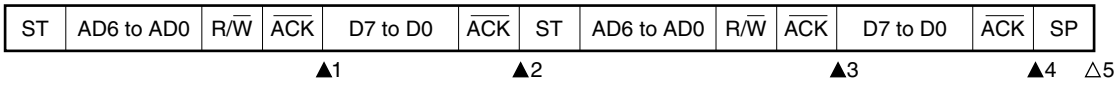
(b) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

(i) When WTIM = 0 (after restart, matches with SVA)

▲1: IICS = 0001×110B
▲2: IICS = 0001×000B
▲3: IICS = 0001×110B
▲4: IICS = 0001×000B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1 (after restart, matches with SVA)

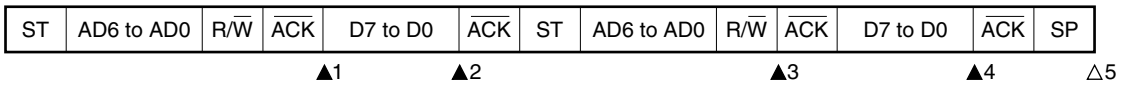


▲1: IICS = 0001×110B
▲2: IICS = 0001××00B
▲3: IICS = 0001×110B
▲4: IICS = 0001××00B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(c) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

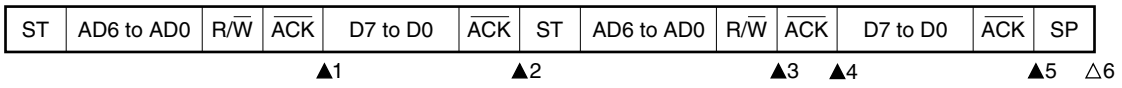
(i) When WTIM = 0 (after restart, does not match address (= extension code))



▲1: IICS = 0001×110B
▲2: IICS = 0001×000B
▲3: IICS = 0010×010B
▲4: IICS = 0010×000B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1 (after restart, does not match address (= extension code))

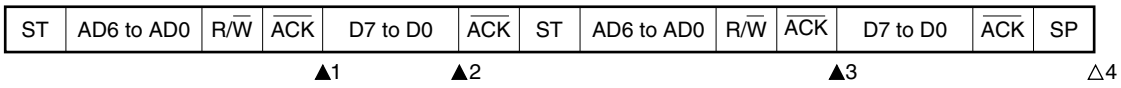


▲1: IICS = 0001×110B
▲2: IICS = 0001××00B
▲3: IICS = 0010×010B
▲4: IICS = 0010×110B
▲5: IICS = 0010××00B
△6: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(d) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

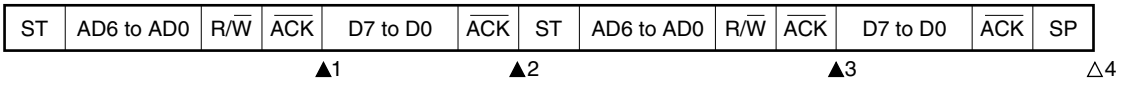
(i) When WTIM = 0 (after restart, does not match address (= not extension code))



▲1: IICS = 0001×110B
▲2: IICS = 0001×000B
▲3: IICS = 00000110B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1 (after restart, does not match address (= not extension code))



▲1: IICS = 0001×110B
▲2: IICS = 0001××00B
▲3: IICS = 00000110B
△4: IICS = 00000001B

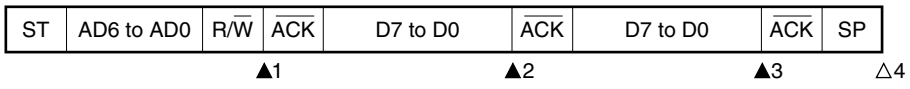
Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(3) Slave device operation (when receiving extension code)

The device is always participating in communication when it receives an extension code.

(a) Start ~ Code ~ Data ~ Data ~ Stop

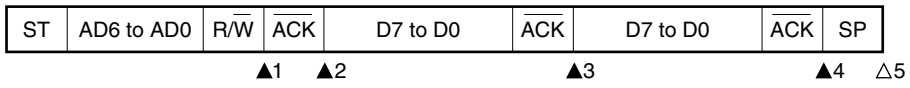
(i) When WTIM = 0



▲1: IICS = 0010×010B
▲2: IICS = 0010×000B
▲3: IICS = 0010×000B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1

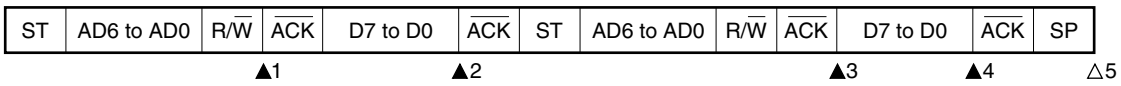


▲1: IICS = 0010×010B
▲2: IICS = 0010×110B
▲3: IICS = 0010×100B
▲4: IICS = 0010××00B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(b) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

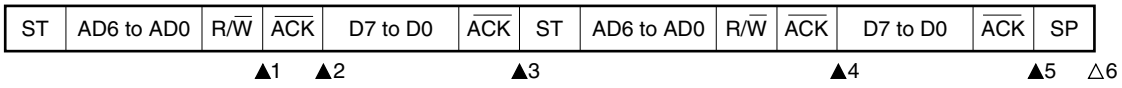
(i) When WTIM = 0 (after restart, matches SVA)



▲1: IICS = 0010×010B
▲2: IICS = 0010×000B
▲3: IICS = 0001×110B
▲4: IICS = 0001×000B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1 (after restart, matches SVA)

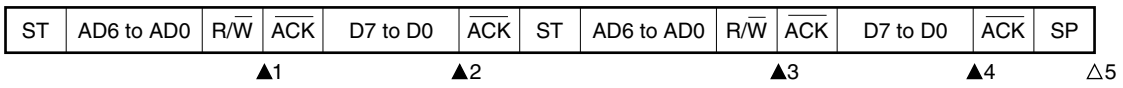


▲1: IICS = 0010×010B
▲2: IICS = 0010×110B
▲3: IICS = 0010××00B
▲4: IICS = 0001×110B
▲5: IICS = 0001××00B
△6: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(c) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

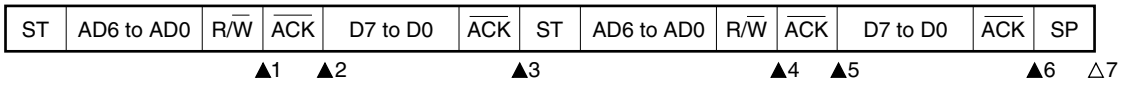
(i) When WTIM = 0 (after restart, extension code reception)



▲1: IICS = 0010×010B
▲2: IICS = 0010×000B
▲3: IICS = 0010×010B
▲4: IICS = 0010×000B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1 (after restart, extension code reception)

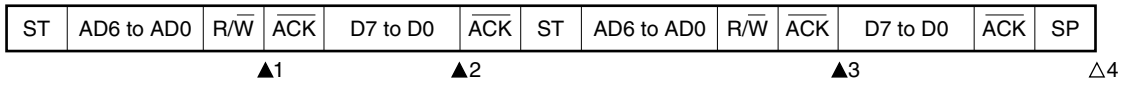


▲1: IICS = 0010×010B
▲2: IICS = 0010×110B
▲3: IICS = 0010××00B
▲4: IICS = 0010×010B
▲5: IICS = 0010×110B
▲6: IICS = 0010××00B
△7: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(d) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

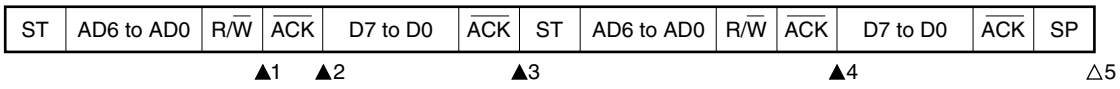
(i) When WTIM = 0 (after restart, does not match address (= not extension code))



▲1: IICS = 00100010B
▲2: IICS = 00100000B
▲3: IICS = 00000110B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1 (after restart, does not match address (= not extension code))

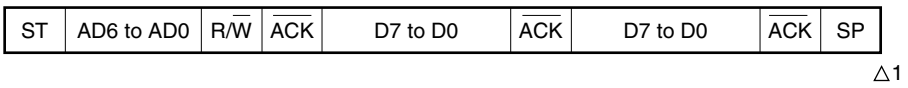


▲1: IICS = 00100010B
▲2: IICS = 00100110B
▲3: IICS = 00100×00B
▲4: IICS = 00000110B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(4) Operation without communication

(a) Start ~ Code ~ Data ~ Data ~ Stop



△1: IICS = 00000001B

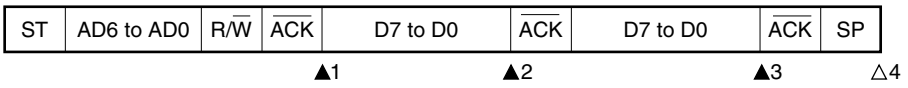
Remark △: Generated only when SPIE = 1

(5) Arbitration loss operation (operation as slave after arbitration loss)

When the device is used as a master in a multi-master system, read the MSTS bit each time interrupt request signal INTIICA has occurred to check the arbitration result.

(a) When arbitration loss occurs during transmission of slave address data

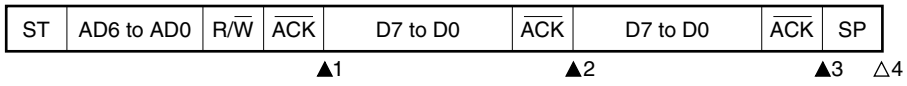
(i) When WTIM = 0



▲1: IICS = 0101×110B
▲2: IICS = 0001×000B
▲3: IICS = 0001×000B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1

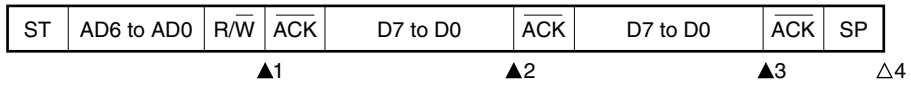


▲1: IICS = 0101×110B
▲2: IICS = 0001×100B
▲3: IICS = 0001××00B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(b) When arbitration loss occurs during transmission of extension code

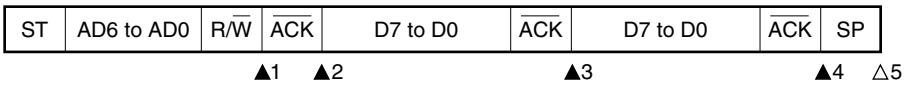
(i) When WTIM = 0



▲1: IICS = 0110×010B
▲2: IICS = 0010×000B
▲3: IICS = 0010×000B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1



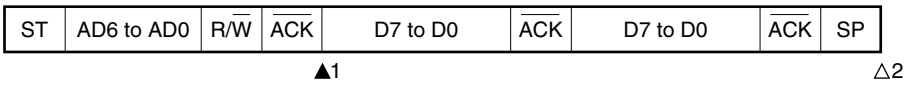
▲1: IICS = 0110×010B
▲2: IICS = 0010×110B
▲3: IICS = 0010×100B
▲4: IICS = 0010××00B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(6) Operation when arbitration loss occurs (no communication after arbitration loss)

When the device is used as a master in a multi-master system, read the MSTS bit each time interrupt request signal INTIICA has occurred to check the arbitration result.

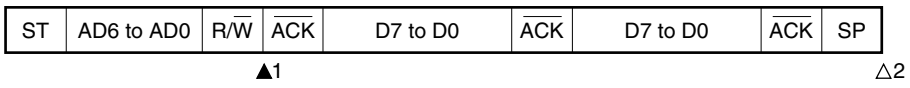
(a) When arbitration loss occurs during transmission of slave address data (when WTIM = 1)



▲1: IICS = 01000110B
△2: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1

(b) When arbitration loss occurs during transmission of extension code

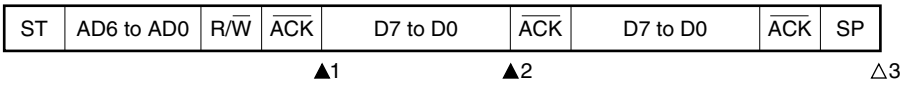


▲1: IICS = 0110×010B
Sets LREL = 1 by software
△2: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(c) When arbitration loss occurs during transmission of data

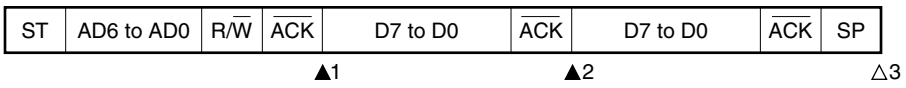
(i) When WTIM = 0



▲1: IICS = 10001110B
▲2: IICS = 01000000B
△3: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1

(ii) When WTIM = 1

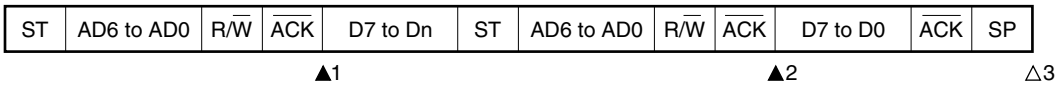


▲1: IICS = 10001110B
▲2: IICS = 01000100B
△3: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1

(d) When loss occurs due to restart condition during data transfer

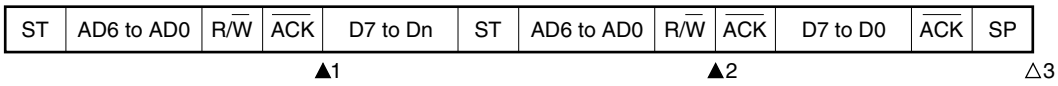
(i) Not extension code (Example: unmatches with SVA)



▲1: IICS = 1000×110B
▲2: IICS = 01000110B
△3: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care
n = 6 to 0

(ii) Extension code



▲1: IICS = 1000×110B

▲2: IICS = 01100010B

Sets LREL = 1 by software

△3: IICS = 00000001B

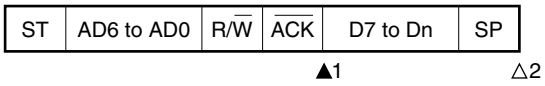
Remark ▲: Always generated

△: Generated only when SPIE = 1

×: Don't care

n = 6 to 0

(e) When loss occurs due to stop condition during data transfer



▲1: IICS = 10000110B

△2: IICS = 01000001B

Remark ▲: Always generated

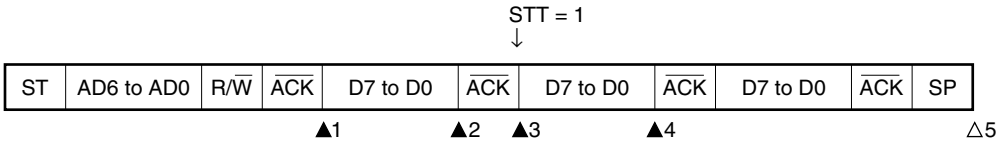
△: Generated only when SPIE = 1

×: Don't care

n = 6 to 0

(f) When arbitration loss occurs due to low-level data when attempting to generate a restart condition

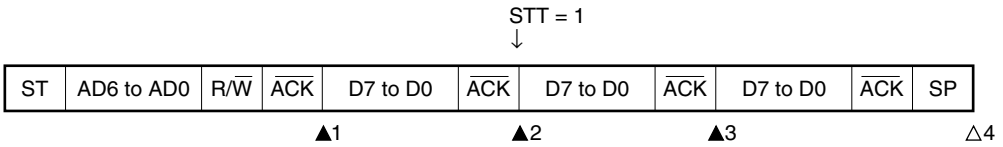
(i) When WTIM = 0



▲1: IICS = 1000×110B
▲2: IICS = 1000×000B (Sets WTIM to 1)
▲3: IICS = 1000×100B (Clears WTIM to 0)
▲4: IICS = 01000000B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1

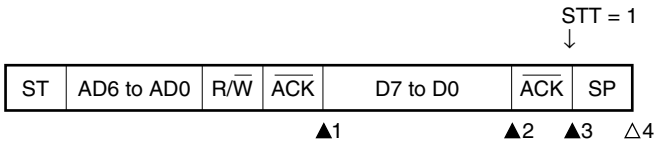


▲1: IICS = 1000×110B
▲2: IICS = 1000×100B (Sets STT to 1)
▲3: IICS = 01000100B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(g) When arbitration loss occurs due to a stop condition when attempting to generate a restart condition

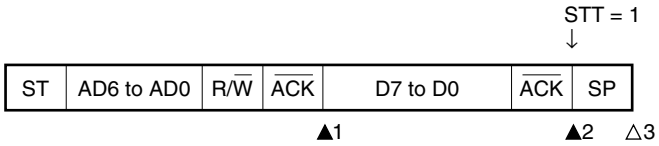
(i) When WTIM = 0



▲1: IICS = 1000×110B
▲2: IICS = 1000×000B (Sets WTIM to 1)
▲3: IICS = 1000××00B (Sets STT to 1)
△4: IICS = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1

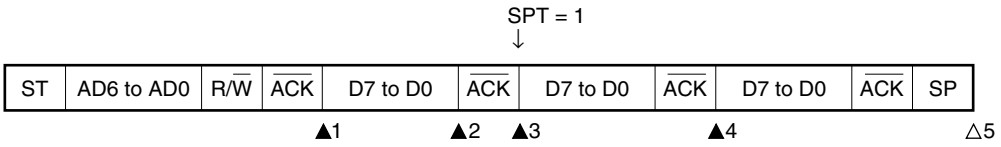


▲1: IICS = 1000×110B
▲2: IICS = 1000××00B (Sets STT to 1)
△3: IICS = 01000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(h) When arbitration loss occurs due to low-level data when attempting to generate a stop condition

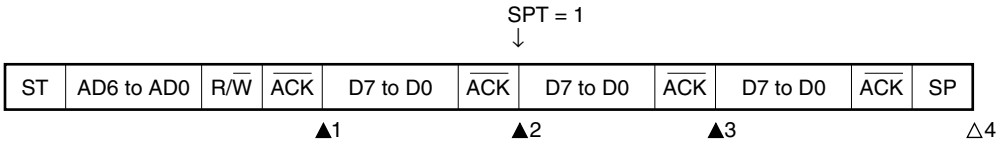
(i) When WTIM = 0



▲1: IICS = 1000×110B
▲2: IICS = 1000×000B (Sets WTIM to 1)
▲3: IICS = 1000×100B (Clears WTIM to 0)
▲4: IICS = 01000100B
△5: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

(ii) When WTIM = 1



▲1: IICS = 1000×110B
▲2: IICS = 1000×100B (Sets SPT to 1)
▲3: IICS = 01000100B
△4: IICS = 00000001B

Remark ▲: Always generated
△: Generated only when SPIE = 1
×: Don't care

<R> 15.6 Timing Charts

When using the I^2C bus mode, the master device outputs an address via the serial bus to select one of several slave devices as its communication partner.

After outputting the slave address, the master device transmits the TRC bit (bit 3 of the IICA status register (IICS)), which specifies the data transfer direction, and then starts serial communication with the slave device.

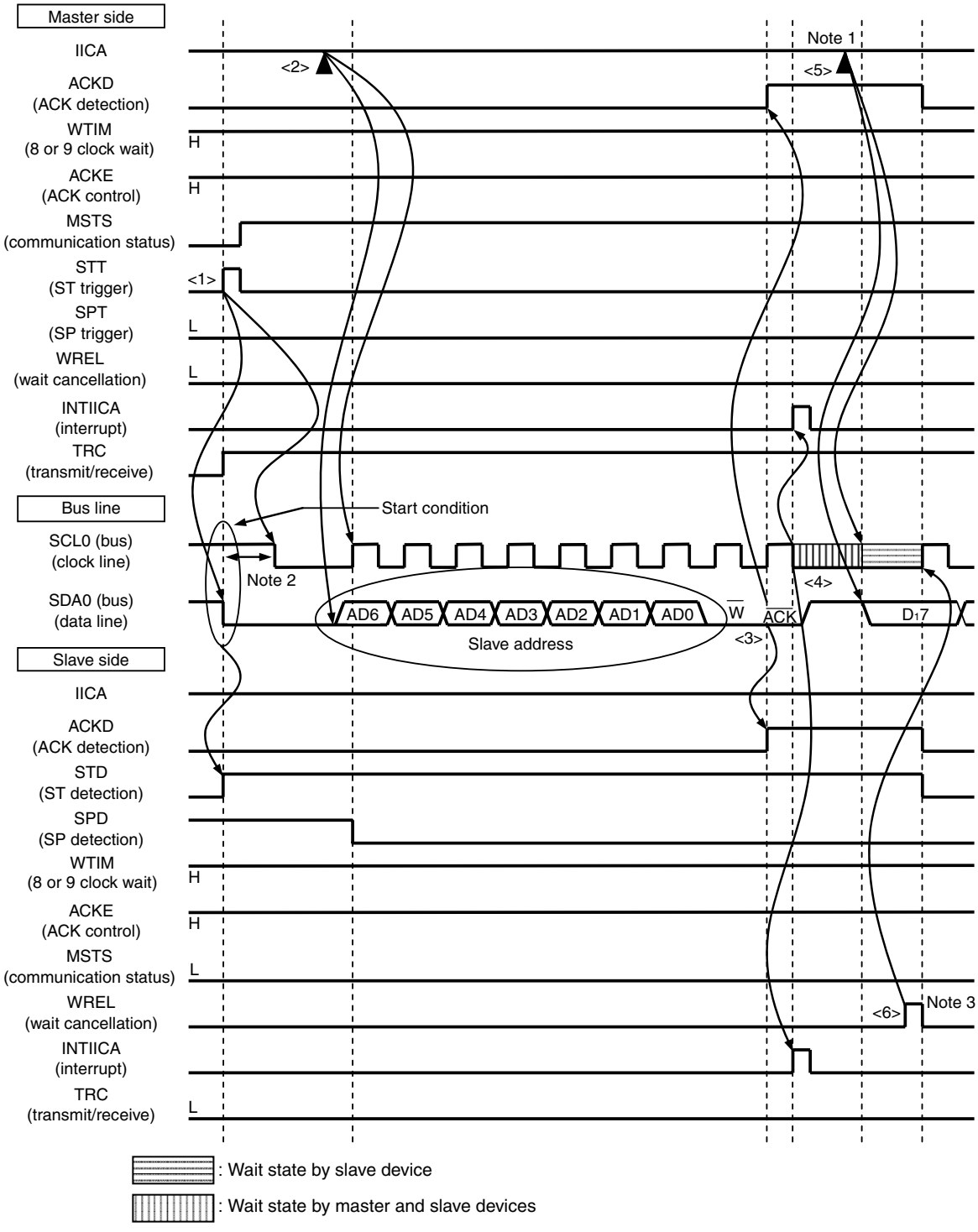
Figures 15-32 and 15-33 show timing charts of the data communication.

The IICA shift register (IICA)'s shift operation is synchronized with the falling edge of the serial clock (SCL0). The transmit data is transferred to the SO latch and is output (MSB first) via the SDA0 pin.

Data input via the SDA0 pin is captured into IICA at the rising edge of SCL0.

Figure 15-32. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (1/4)

(1) Start condition ~ address ~ data



Notes 1. Write data to IICA, not setting the WREL bit, in order to cancel a wait state during master transmission.

2. Make sure that the time between the fall of the SDA0 pin signal and the fall of the SCL0 pin signal is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

3. To cancel slave wait, write “FFH” to IICA or set the WREL bit.

The meanings of <1> to <6> in (1) Start condition ~ address ~ data in Figure 15-32 are explained below.

<1> The start condition trigger is set by the master device (STT = 1) and a start condition (SDA0 = 0 and SCL0 = 1) is generated once the bus data line goes low (SDA0 = 0). When the start condition is subsequently detected, the master device enters the master device communication status (MSTS = 1). The master device is ready to communicate once the bus clock line goes low (SCL0 = 0) after the hold time has elapsed.

<2> The master device writes the address + W (transmission) to the IICA shift register (IICA) and transmits the slave address.

<3> If the address received matches the address of a slave device[Note], that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA: end of address transmission) at the falling edge of the 9th clock, and the slave device whose address matched the transmitted slave address also issues an interrupt (INTIICA: address match). The master device and slave device also set a wait status (SCL0 = 0)[Note] when the addresses match.

<5> The master device writes the data to transmit to the IICA register and releases the wait status that it set by the master device.

<6> If the slave device releases the wait status (WREL = 1), the master device starts transferring data to the slave device.

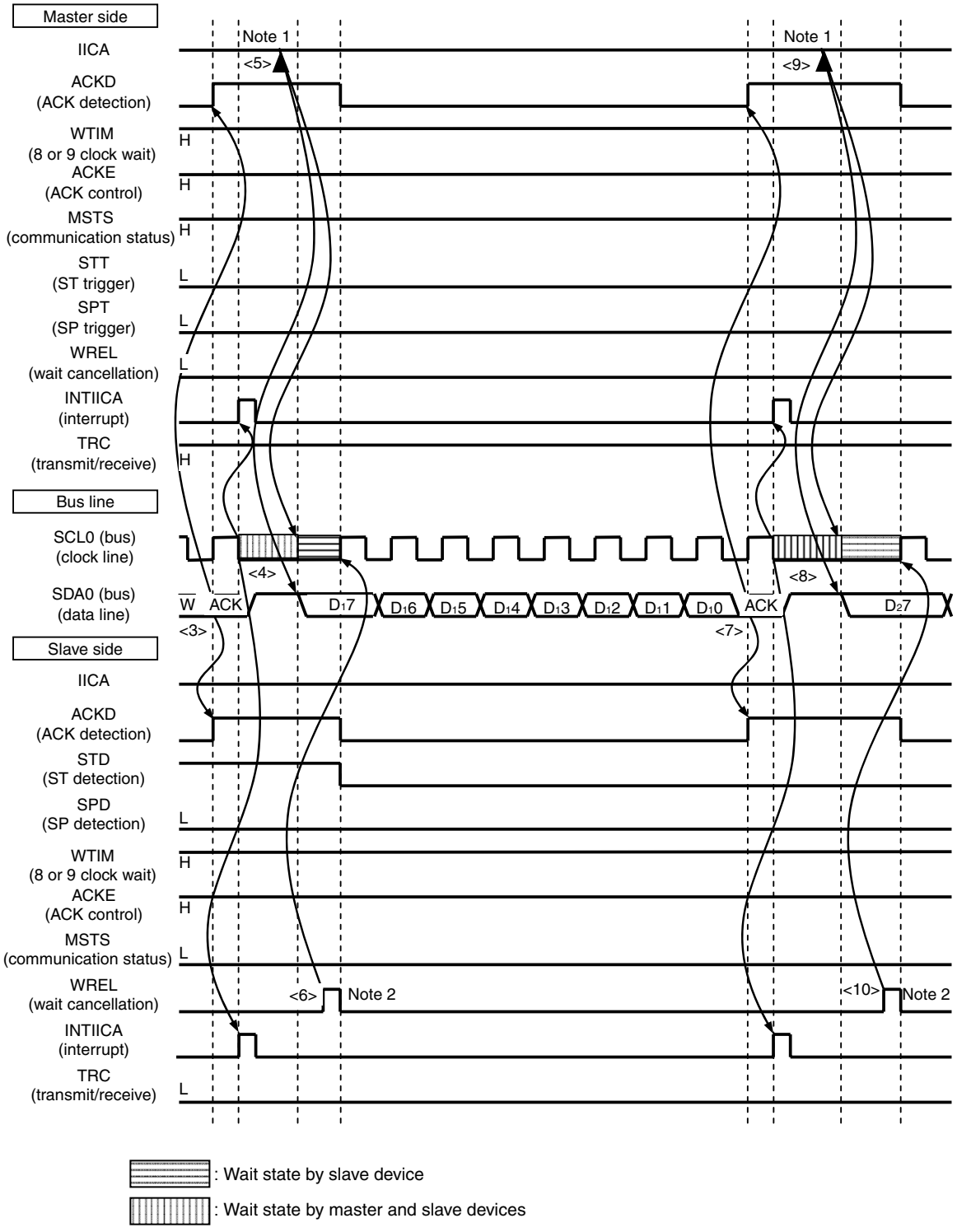
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDA0 = 1). The slave device also does not issue the INTIICA interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <15> in Figure 15-32 represent the entire procedure for communicating data using the I^2C bus.

Figure 15-32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 15-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 15-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 15-32. Example of Master to Slave Communication
(When 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (2/4)

(2) Address ~ data ~ data



Notes 1. Write data to IICA, not setting the WREL bit, in order to cancel a wait state during master transmission.

2. To cancel slave wait, write “FFH” to IICA or set the WREL bit.

The meanings of <3> to <10> in (2) Address ~ data ~ data in Figure 15-32 are explained below.

<3> If the address received matches the address of a slave device[Note], that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA: end of address transmission) at the falling edge of the 9th clock, and the slave device whose address matched the transmitted slave address also issues an interrupt (INTIICA: address match). The master device and slave device also set a wait status (SCL0 = 0)[Note] when the addresses match.

<5> The master device writes the data to transmit to the IICA shift register (IICA) and releases the wait status that it set by the master device.

<6> If the slave device releases the wait status (WREL = 1), the master device starts transferring data to the slave device.

<7> When data transfer is complete, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA: end of transfer).

<9> The master device writes the data to transmit to the IICA register and releases the wait status that it set by the master device.

<10> The slave device reads the received data and releases the wait status (WREL = 1). The master device then starts transferring data to the slave device.

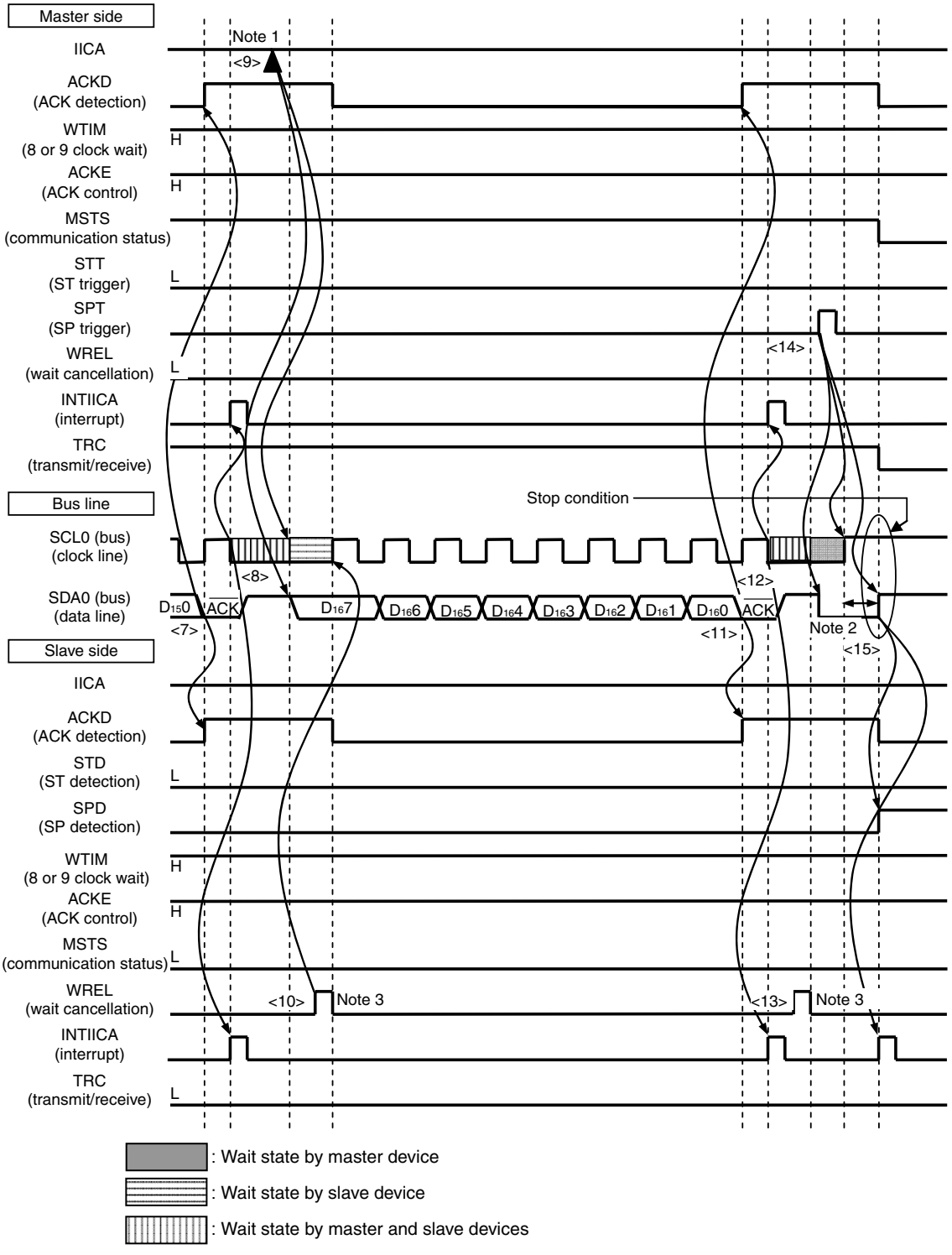
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDA0 = 1). The slave device also does not issue the INTIICA interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <15> in Figure 15-32 represent the entire procedure for communicating data using the I^2C bus.

Figure 15-32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 15-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 15-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 15-32. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (3/4)

(3) Data ~ data ~ Stop condition



Notes 1. Write data to IICA, not setting the WREL bit, in order to cancel a wait state during master transmission.

2. Make sure that the time between the rise of the SCL0 pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

3. To cancel slave wait, write “FFH” to IICA or set the WREL bit.

The meanings of <7> to <15> in (3) Data ~ data ~ stop condition in Figure 15-32 are explained below.

<7> When data transfer is complete, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA: end of transfer).

<9> The master device writes the data to transmit to the IICA shift register (IICA) and releases the wait status that it set by the master device.

<10> The slave device reads the received data and releases the wait status (WREL = 1). The master device then starts transferring data to the slave device.

<11> When data transfer is complete, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD = 1) at the rising edge of the 9th clock.

<12> The master device and slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA: end of transfer).

<13> The slave device reads the received data and releases the wait status (WREL = 1).

<14> After a stop condition trigger is set, the bus data line is cleared (SDA0 = 0) and the bus clock line is set (SCL0 = 1). The stop condition is then generated by setting the bus data line (SDA0 = 1) after the stop condition setup time has elapsed.

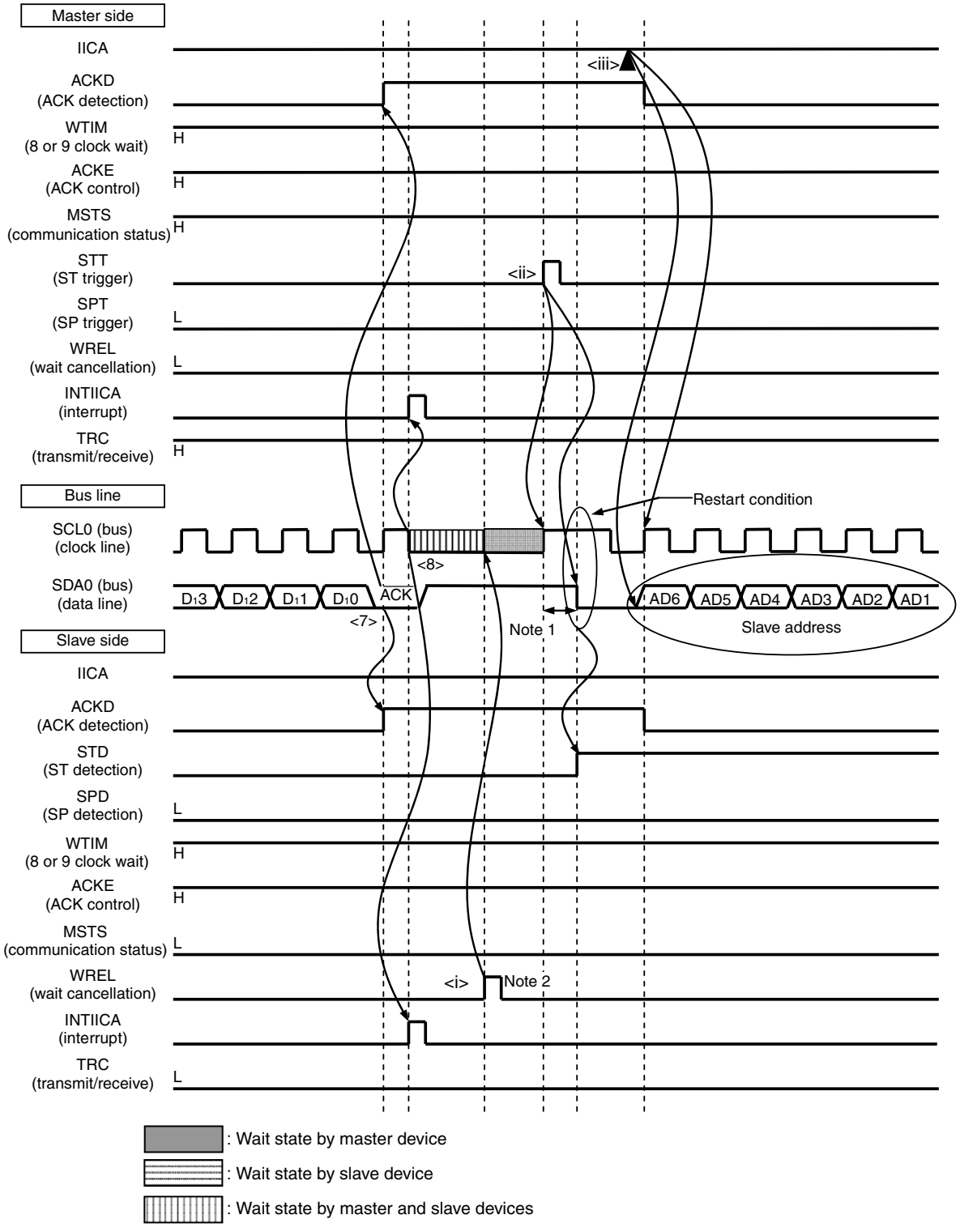
<15> When a stop condition is generated, the slave device detects the stop condition and issues an interrupt (INTIICA: stop condition).

Remark <1> to <15> in Figure 15-32 represent the entire procedure for communicating data using the I^2C bus.

Figure 15-32 (1) Start condition ~ address ~ data shows the processing from <1> to <6>, Figure 15-32 (2) Address ~ data ~ data shows the processing from <3> to <10>, and Figure 15-32 (3) Data ~ data ~ stop condition shows the processing from <7> to <15>.

Figure 15-32. Example of Master to Slave Communication (When 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (4/4)

(4) Data ~ restart condition ~ address



Notes 1. Make sure that the time between the rise of the SCL0 pin signal and the generation of the start condition after a restart condition has been issued is at least 4.7 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

2. To cancel slave wait, write “FFH” to IICA or set the WREL bit.

The following describes the operations in Figure 15-32 (4) Data ~ restart condition ~ address. After the operations in steps <7> and <8>, the operations in steps <1> to <3> are performed. These steps return the processing to step <3>, the data transmission step.

<7> When data transfer is complete, the slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD = 1) at the rising edge of the 9th clock.

<8> The master device and slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA: end of transfer).

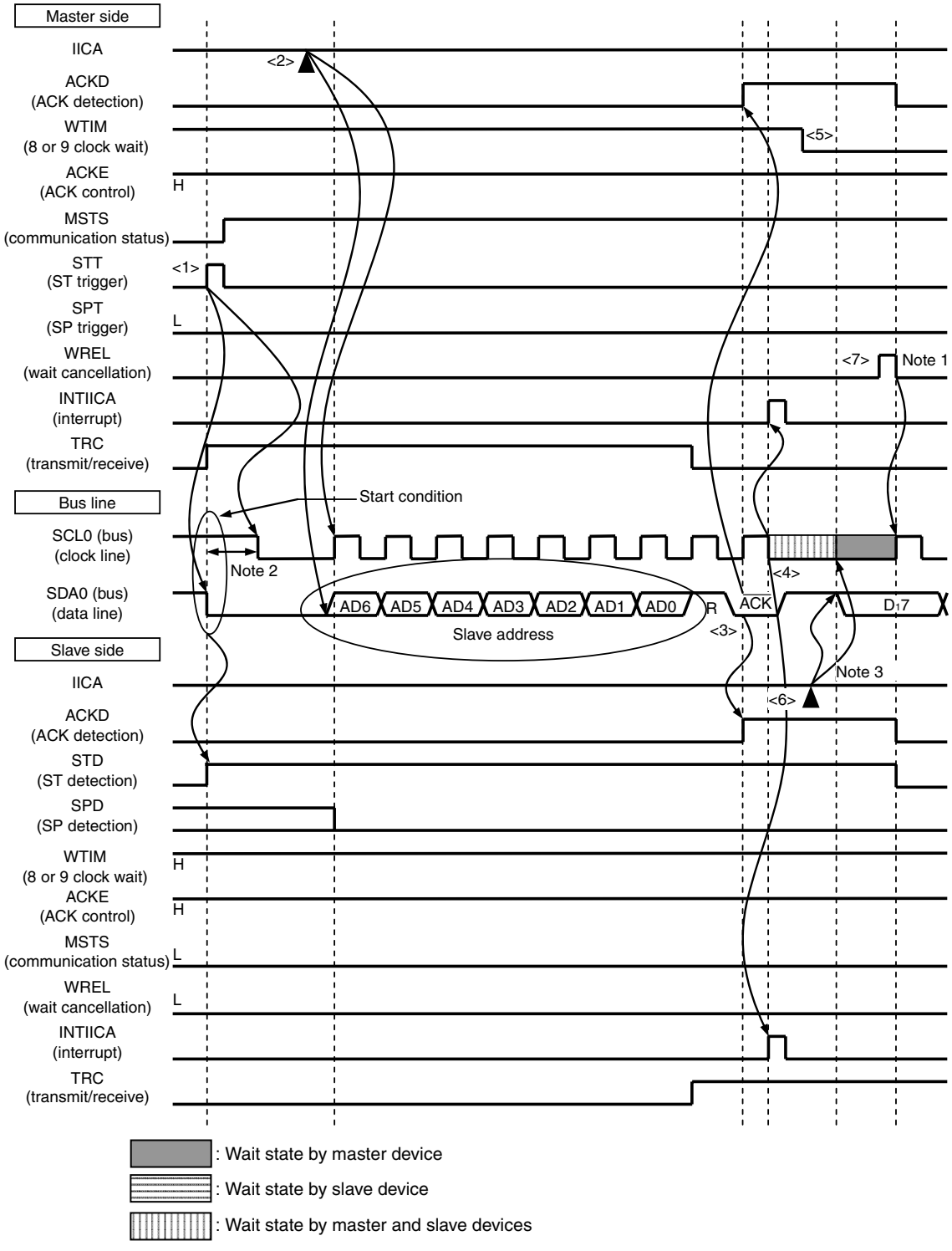
<i> The slave device reads the received data and releases the wait status (WREL = 1).

<ii> The start condition trigger is set again by the master device (STT = 1) and a start condition (SDA0 = 0 and SCL0 = 1) is generated once the bus clock line goes high (SCL0 = 1) and the bus data line goes low (SDA0 = 0) after the restart condition setup time has elapsed. When the start condition is subsequently detected, the master device is ready to communicate once the bus clock line goes low (SCL0 = 0) after the hold time has elapsed.

<iii> The master device writes the address + R/W (transmission) to the IICA shift register (IICA) and transmits the slave address.

Figure 15-33. Example of Slave to Master Communication (When 8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (1/3)

(1) Start condition ~ address ~ data



Notes 1. To cancel master wait, write “FFH” to IICA or set the WREL bit.

2. Make sure that the time between the fall of the SDA0 pin signal and the fall of the SCL0 pin signal is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

3. Write data to IICA, not setting the WREL bit, in order to cancel a wait state during slave transmission.

The meanings of <1> to <7> in (1) Start condition ~ address ~ data in Figure 15-33 are explained below.

<1> The start condition trigger is set by the master device (STT = 1) and a start condition (SDA0 = 0 and SCL0 = 1) is generated once the bus data line goes low (SDA0 = 0). When the start condition is subsequently detected, the master device enters the master device communication status (MSTS = 1). The master device is ready to communicate once the bus clock line goes low (SCL0 = 0) after the hold time has elapsed.

<2> The master device writes the address + W (transmission) to the IICA shift register (IICA) and transmits the slave address.

<3> If the address received matches the address of a slave device[Note], that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA: end of address transmission) at the falling edge of the 9th clock, and the slave device whose address matched the transmitted slave address also issues an interrupt (INTIICA: address match). The master device and slave device also set a wait status (SCL0 = 0)[Note] when the addresses match.

<5> The timing at which the master device sets the wait status changes to the 8th clock (WTIM = 0).

<6> The slave device writes the data to transmit to the IICA register and releases the wait status that it set by the slave device.

<7> If the master device releases the wait status (WREL = 1), the slave device starts transferring data to the master device.

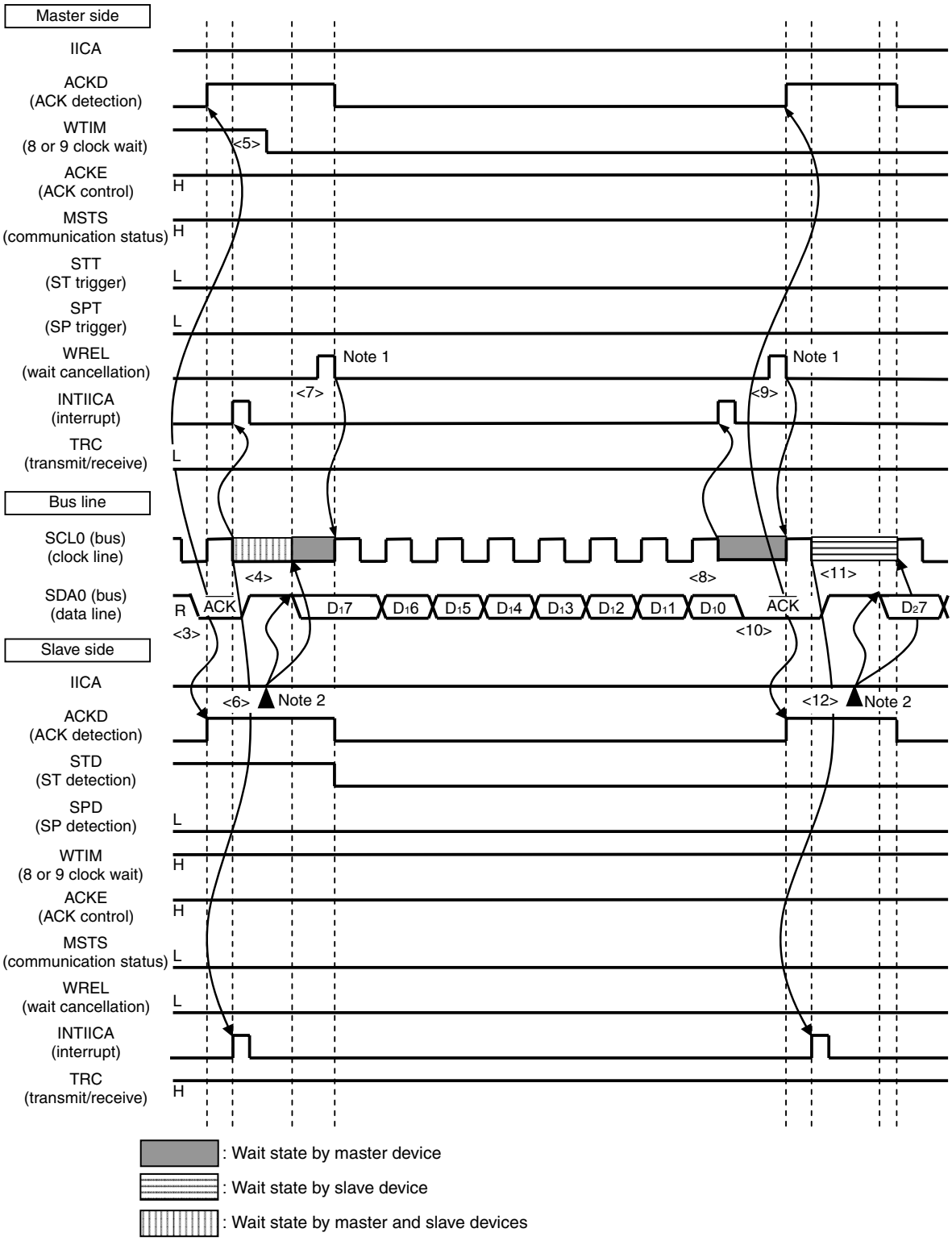
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDA0 = 1). The slave device also does not issue the INTIICA interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <19> in Figure 15-33 represent the entire procedure for communicating data using the I^2C bus.

Figure 15-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 15-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 15-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Figure 15-33. Example of Slave to Master Communication
(When 8-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (2/3)

(2) Address ~ data ~ data



Notes 1. To cancel master wait, write “FFH” to IICA or set the WREL bit.

2. Write data to IICA, not setting the WREL bit, in order to cancel a wait state during slave transmission.

The meanings of <3> to <12> in (2) Address ~ data ~ data in Figure 15-33 are explained below.

<3> If the address received matches the address of a slave device[Note], that slave device sends an ACK by hardware to the master device. The ACK is detected by the master device (ACKD = 1) at the rising edge of the 9th clock.

<4> The master device issues an interrupt (INTIICA: end of address transmission) at the falling edge of the 9th clock, and the slave device whose address matched the transmitted slave address also issues an interrupt (INTIICA: address match). The master device and slave device also set a wait status (SCL0 = 0)[Note] when the addresses match.

<5> The timing at which the master device sets the wait status changes to the 8th clock (WTIM = 0).

<6> The slave device writes the data to transmit to the IICA shift register (IICA) and releases the wait status that it set by the slave device.

<7> If the master device releases the wait status (WREL = 1), the slave device starts transferring data to the master device.

<8> The master device sets a wait status (SCL0 = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICA: end of transfer). The master device then sends an ACK by hardware to the slave device.

<9> The master device reads the received data and releases the wait status (WREL = 1).

<10> The ACK is detected by the slave device (ACKD = 1) at the rising edge of the 9th clock.

<11> The slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICA: end of transfer).

<12> The slave device writes the data to transmit to the IICA register and releases the wait status that it set by the slave device. The slave device then starts transferring data to the master device.

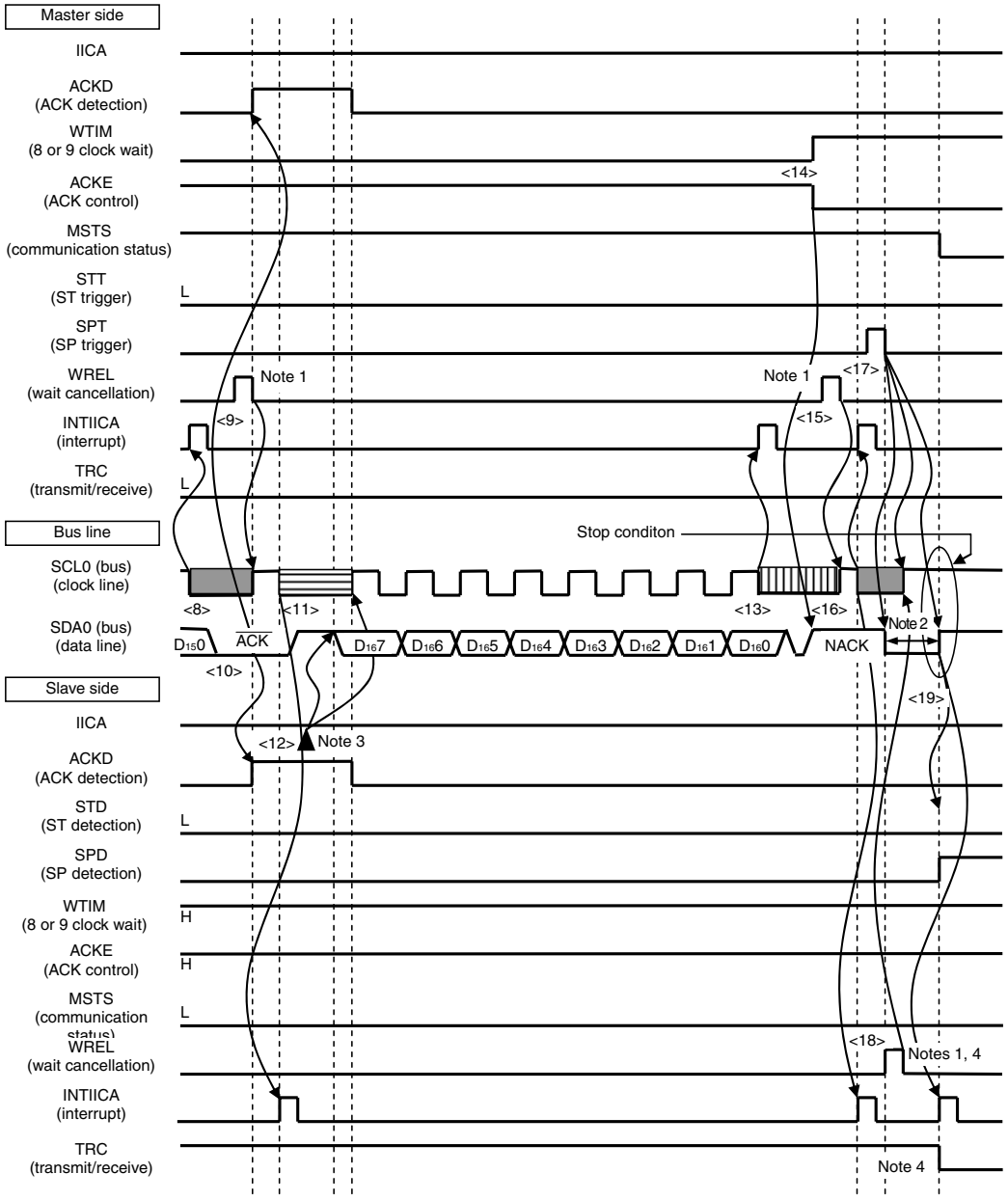
Note If the transmitted address does not match the address of the slave device, the slave device does not return an ACK to the master device (NACK: SDA0 = 1). The slave device also does not issue the INTIICA interrupt (address match) and does not set a wait status. The master device, however, issues the INTIICA interrupt (end of address transmission) regardless of whether it receives an ACK or NACK.

Remark <1> to <19> in Figure 15-33 represent the entire procedure for communicating data using the I^2C bus.

Figure 15-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 15-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 15-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

Figure 15-33. Example of Slave to Master Communication (When 8-Clock and 9-Clock Wait Is Selected for Master, 9-Clock Wait Is Selected for Slave) (3/3)

(3) Data ~ data ~ stop condition



Notes 1. To cancel a wait state, write “FFH” to IICA or set the WREL bit.

2. Make sure that the time between the rise of the SCL0 pin signal and the generation of the stop condition after a stop condition has been issued is at least 4.0 μs when specifying standard mode and at least 0.6 μs when specifying fast mode.

3. Write data to IICA, not setting the WREL bit, in order to cancel a wait state during slave transmission.

4. If a wait state during slave transmission is canceled by setting the WREL bit, the TRC bit will be cleared.

The meanings of <8> to <19> in (3) Data ~ data ~ stop condition in Figure 15-33 are explained below.

<8> The master device sets a wait status (SCL0 = 0) at the falling edge of the 8th clock, and issues an interrupt (INTIICA: end of transfer). The master device then sends an ACK by hardware to the slave device.

<9> The master device reads the received data and releases the wait status (WREL = 1).

<10> The ACK is detected by the slave device (ACKD = 1) at the rising edge of the 9th clock.

<11> The slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and the slave device issue an interrupt (INTIICA: end of transfer).

<12> The slave device writes the data to transmit to the IICA shift register (IICA) and releases the wait status that it set by the slave device. The slave device then starts transferring data to the master device.

<13> The master device issues an interrupt (INTIICA: end of transfer) at the falling edge of the 8th clock, and sets a wait status (SCL0 = 0). Because ACK control (ACKE = 1) is performed, the bus data line is at the low level (SDA0 = 0) at this stage.

<14> The master device sets NACK as the response (ACKE = 0) and changes the timing at which it sets the wait status to the 9th clock.

<15> If the master device releases the wait status (WREL = 1), the slave device detects the NACK (ACK = 0) at the rising edge of the 9th clock.

<16> The master device and slave device set a wait status (SCL0 = 0) at the falling edge of the 9th clock, and both the master device and slave device issue an interrupt (INTIICA: end of transfer).

<17> When the master device issues a stop condition (SPT = 1), the bus data line is cleared (SDA0 = 0) and the master device releases the wait status. The master device then waits until the bus clock line is set (SCL0 = 1).

<18> The slave device acknowledges the NACK, halts transmission, and releases the wait status (WREL = 1) to end communication. Once the slave device releases the wait status, the bus clock line is set (SCL0 = 1).

<19> Once the master device recognizes that the bus clock line is set (SCL0 = 1) and after the stop condition setup time has elapsed, the master device sets the bus data line (SDA0 = 1) and issues a stop condition. The slave device detects the generated stop condition and both the master device and slave device issue an interrupt (INTIICA: stop condition).

Remark <1> to <19> in Figure 15-33 represent the entire procedure for communicating data using the I^2C bus.

Figure 15-33 (1) Start condition ~ address ~ data shows the processing from <1> to <7>, Figure 15-33 (2) Address ~ data ~ data shows the processing from <3> to <12>, and Figure 15-33 (3) Data ~ data ~ stop condition shows the processing from <8> to <19>.

CHAPTER 16 MULTIPLIER/DIVIDER

16.1 Functions of Multiplier/Divider

The multiplier/divider has the following functions.

- 16 bits × 16 bits = 32 bits (multiplication)
- 32 bits ÷ 32 bits = 32 bits, 32-bit remainder (division)

16.2 Configuration of Multiplier/Divider

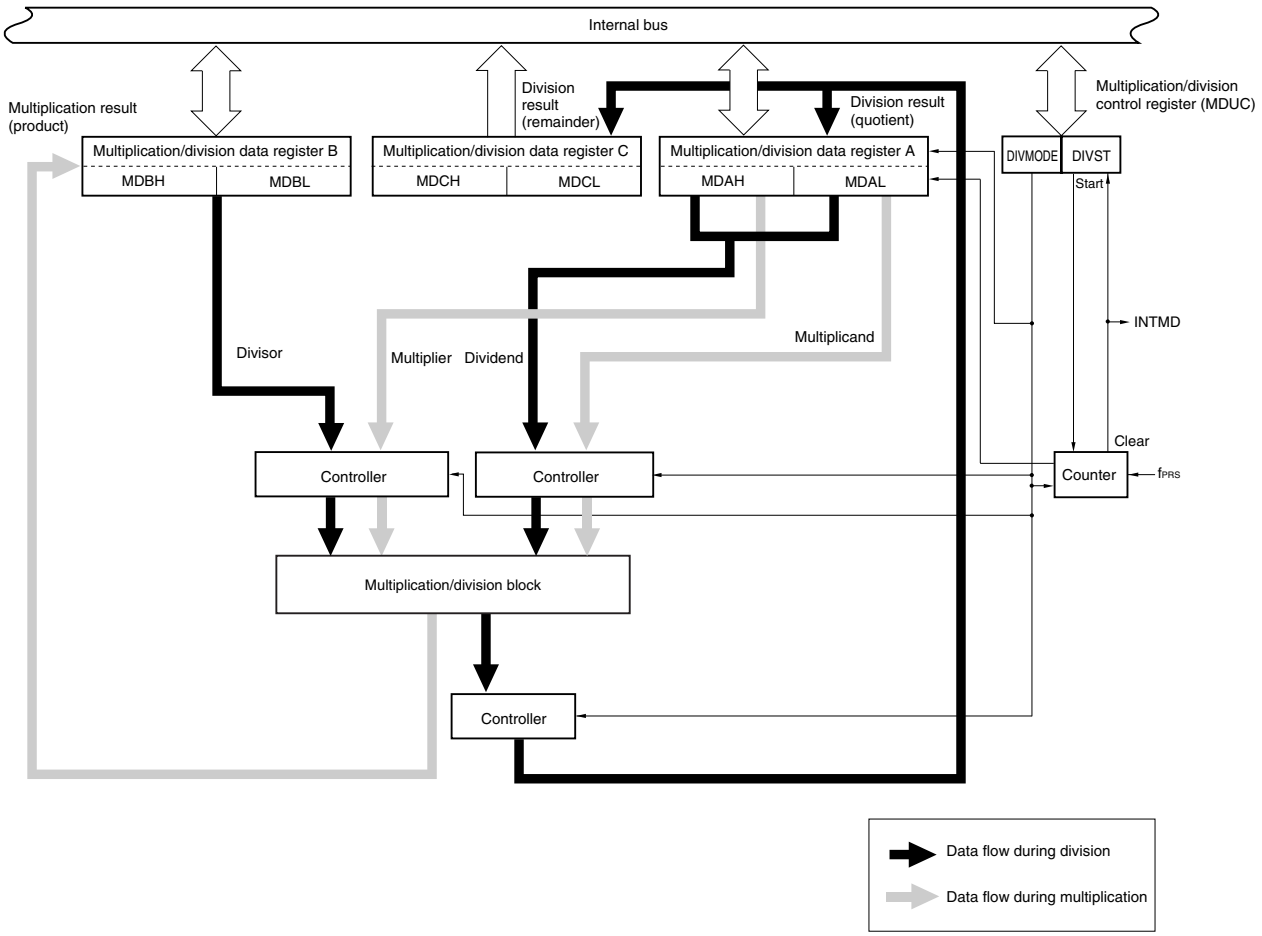
The multiplier/divider consists of the following hardware.

Table 16-1. Configuration of Multiplier/Divider

Item	Configuration
Registers	Multiplication/division data register A (L) (MDAL) Multiplication/division data register A (H) (MDAH) Multiplication/division data register B (L) (MDBL) Multiplication/division data register B (H) (MDBH) Multiplication/division data register C (L) (MDCL) Multiplication/division data register C (H) (MDCH)
Control register	Multiplication/division control register (MDUC)

Figure 16-1 shows a block diagram of the multiplier/divider.

Figure 16-1. Block Diagram of Multiplier/Divider



(1) Multiplication/division data register A (MDAH, MDAL)

The MDAH and MDAL registers set the values that are used for a multiplication or division operation and store the operation result. They set the multiplier and multiplicand data in the multiplication mode, and set the dividend data in the division mode. Furthermore, the operation result (quotient) is stored in the MDAH and MDAL registers in the division mode.

MDAH and MDAL can be set by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 16-2. Format of Multiplication/Division Data Register A (MDAH, MDAL)

Address: FFFF0H, FFFF1H, FFFF2H, FFFF3H After reset: 0000H, 0000H R/W

Symbol FFFF3H FFFF2H

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

Symbol FFFF1H FFFF0H

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

Cautions 1. Do not rewrite the MDAH and MDAL values during division operation processing (while the multiplication/division control register (MDUC) is 81H). The operation will be executed in this case, but the operation result will be an undefined value.

2. The MDAH and MDAL values read during division operation processing (while MDUC is 81H) will not be guaranteed.

The following table shows the functions of MDAH and MDAL during operation execution.

Table 16-2. Functions of MDAH and MDAL During Operation Execution

DIVMODE	Operation Mode	Setting	Operation Result
0	Multiplication mode	MDAH: Multiplier MDAL: Multiplicand	–
1	Division mode	MDAH: Divisor (higher 16 bits) MDAL: Dividend (lower 16 bits)	MDAH: Division result (quotient) Higher 16 bits MDAL: Division result (quotient) Lower 16 bits

Remark DIVMODE: Bit 7 of the multiplication/division control register (MDUC)

(2) Multiplication/division data register B (MDBL, MDBH)

The MDBH and MDBL registers set the values that are used for multiplication or division operation and store the operation result. They store the operation result (product) in the multiplication mode and set the divisor data in the division mode.

MDBH and MDBL can be set by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 16-3. Format of Multiplication/Division Data Register B (MDBH, MDBL)

Address: FFFF4H, FFFF5H, FFFF6H, FFFF7H After reset: 0000H, 0000H R/W

Symbol	FFFF5H								FFFF4H							
MDBH	MDBH 15	MDBH 14	MDBH 13	MDBH 12	MDBH 11	MDBH 10	MDBH 9	MDBH 8	MDBH 7	MDBH 6	MDBH 5	MDBH 4	MDBH 3	MDBH 2	MDBH 1	MDBH 0

Symbol	FFFF7H								FFFF6H							
MDBL	MDBL 15	MDBL 14	MDBL 13	MDBL 12	MDBL 11	MDBL 10	MDBL 9	MDBL 8	MDBL 7	MDBL 6	MDBL 5	MDBL 4	MDBHL 3	MDBL 2	MDBL 1	MDBL 0

Cautions 1. Do not rewrite the MDBH and MDBL values during division operation processing (while the multiplication/division control register (MDUC) is 81H). The operation result will be an undefined value.

2. Do not set MDBH and MDBL to 0000H in the division mode. If they are set, the operation result will be an undefined value.

The following table shows the functions of MDBH and MDBL during operation execution.

Table 16-3. Functions of MDBH and MDBL During Operation Execution

DIVMODE	Operation Mode	Setting	Operation Result
0	Multiplication mode	–	MDBH: Multiplication result (product) Higher 16 bits MDBL: Multiplication result (product) Lower 16 bits
1	Division mode	MDBH: Divisor (higher 16 bits) MDBL: Dividend (lower 16 bits)	–

Remark DIVMODE: Bit 7 of the multiplication/division control register (MDUC)

(3) Multiplication/division data register C (MDCL, MDCH)

The MDCH and MDCL registers store remainder value of the operation result in the division mode. They are not used in the multiplication mode.

MDCH and MDCL can be read by a 16-bit manipulation instruction.

Reset signal generation clears these registers to 0000H.

Figure 16-4. Format of Multiplication/Division Data Register C (MDCH, MDCL)

Address: F00E0H, F00E1H, F00E2H, F00E3H After reset: 0000H, 0000H R

Symbol F00E3H F00E2H

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

Symbol F00E1H F00E0H

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

Caution The MDCH and MDCL values read during division operation processing (while the multiplication/division control register (MDUC) is 81H) will not be guaranteed.

Table 16-4. Functions of MDCH and MDCL During Operation Execution

DIVMODE	Operation Mode	Setting	Operation Result
0	Multiplication mode	–	–
1	Division mode	–	MDCH: Remainder (higher 16 bits) MDCL: Remainder (lower 16 bits)

Remark DIVMODE: Bit 7 of the multiplication/division control register (MDUC)

The register configuration differs between when multiplication is executed and when division is executed, as follows.

- Register configuration during multiplication

<Multiplier A> <Multiplier B> <Product>

MDAL (bits 15 to 0) × MDAH (bits 15 to 0) = [MDBH (bits 15 to 0), MDBL (bits 15 to 0)]

- Register configuration during division

<Dividend> <Divisor>

[MDAH (bits 15 to 0), MDAL (bits 15 to 0)] ÷ [MDBH (bits 15 to 0), MDBL (bits 15 to 0)] =

<Quotient> <Remainder>

[MDAH (bits 15 to 0), MDAL (bits 15 to 0)] ··· [MDCH (bits 15 to 0), MDCL (bits 15 to 0)]

16.3 Register Controlling Multiplier/Divider

The multiplier/divider is controlled by using the multiplication/division control register (MDUC).

(1) Multiplication/division control register (MDUC)

MDUC is an 8-bit register that controls the operation of the multiplier/divider.

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

Reset signal generation clears this register to 00H.

Figure 16-5. Format of Multiplication/Division Control Register (MDUC)

Address: F00E8H After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
MDUC	DIVMODE	0	0	0	0	0	0	DIVST

DIVMODE	Operation mode (multiplication/division) selection
0	Multiplication mode
1	Division mode

DIVST[Note]	Division operation start/stop
0	Division operation processing complete
1	Starts division operation/division operation processing in progress

Note DIVST can only be set (1) in the division mode. In the division mode, division operation is started by setting (1) DIVST. DIVST is automatically cleared (0) when the operation ends. In the multiplication mode, operation is automatically started by setting the multiplier and multiplicand to MDAH and MDAL, respectively.

Cautions 1. Do not rewrite DIVMODE during operation processing (while DIVST is 1). If it is rewritten, the operation result will be an undefined value.

2. DIVST cannot be cleared (0) by using software during division operation processing (while DIVST is 1).

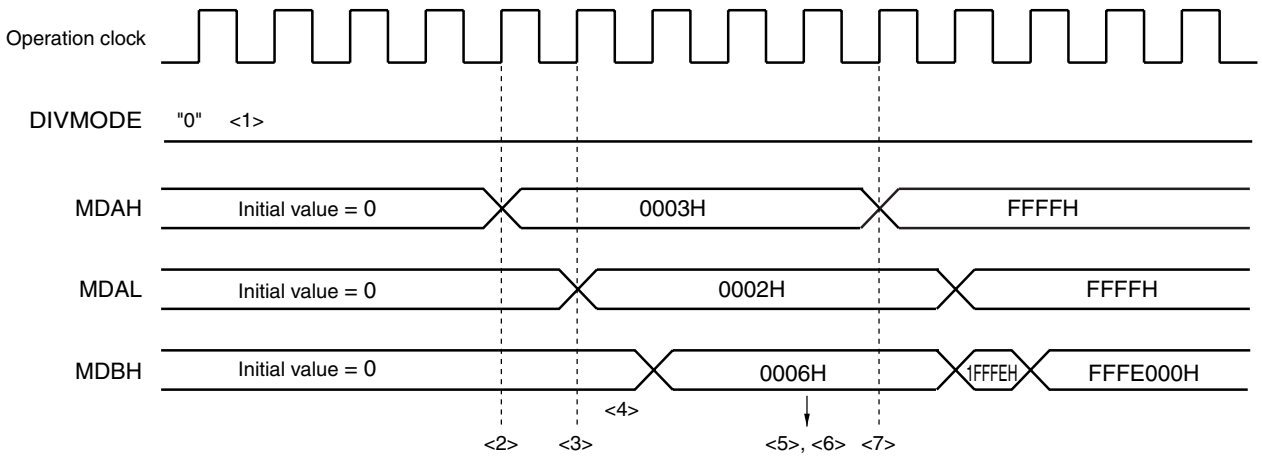
16.4 Operations of Multiplier/Divider

16.4.1 Multiplication operation

- Initial setting
  - <1> Set bit 7 (DIVMODE) of the multiplication/division control register (MDUC) to 0.
  - <2> Set the multiplicand to the multiplication/division data register A (L) (MDAL).
  - <3> Set the multiplier to the multiplication/division data register A (H) (MDAH).
    (There is no preference in the order of executing steps <2> and <3>. Multiplication operation is automatically started when the multiplier and multiplicand are set to MDAH and MDAL, respectively.)
- During operation processing
  - <4> Wait for at least one clock. The operation will end when one clock has been issued.
- Operation end
  - <5> Read the product (lower 16 bits) from the multiplication/division data register B (L) (MDBL).
  - <6> Read the product (higher 16 bits) from the multiplication/division data register B (H) (MDBH).
    (There is no preference in the order of executing steps <5> and <6>.)
- Next operation
  - <7> To execute multiplication operation next, start from the “Initial setting” for multiplication operation.
  - <8> To execute division operation next, start from the “Initial setting” in **16.4.2 Division operation**.

Remark Steps <1> to <7> correspond to <1> to <7> in **Figure 16-6**.

Figure 16-6. Timing Diagram of Multiplication Operation (0003H × 0002H)



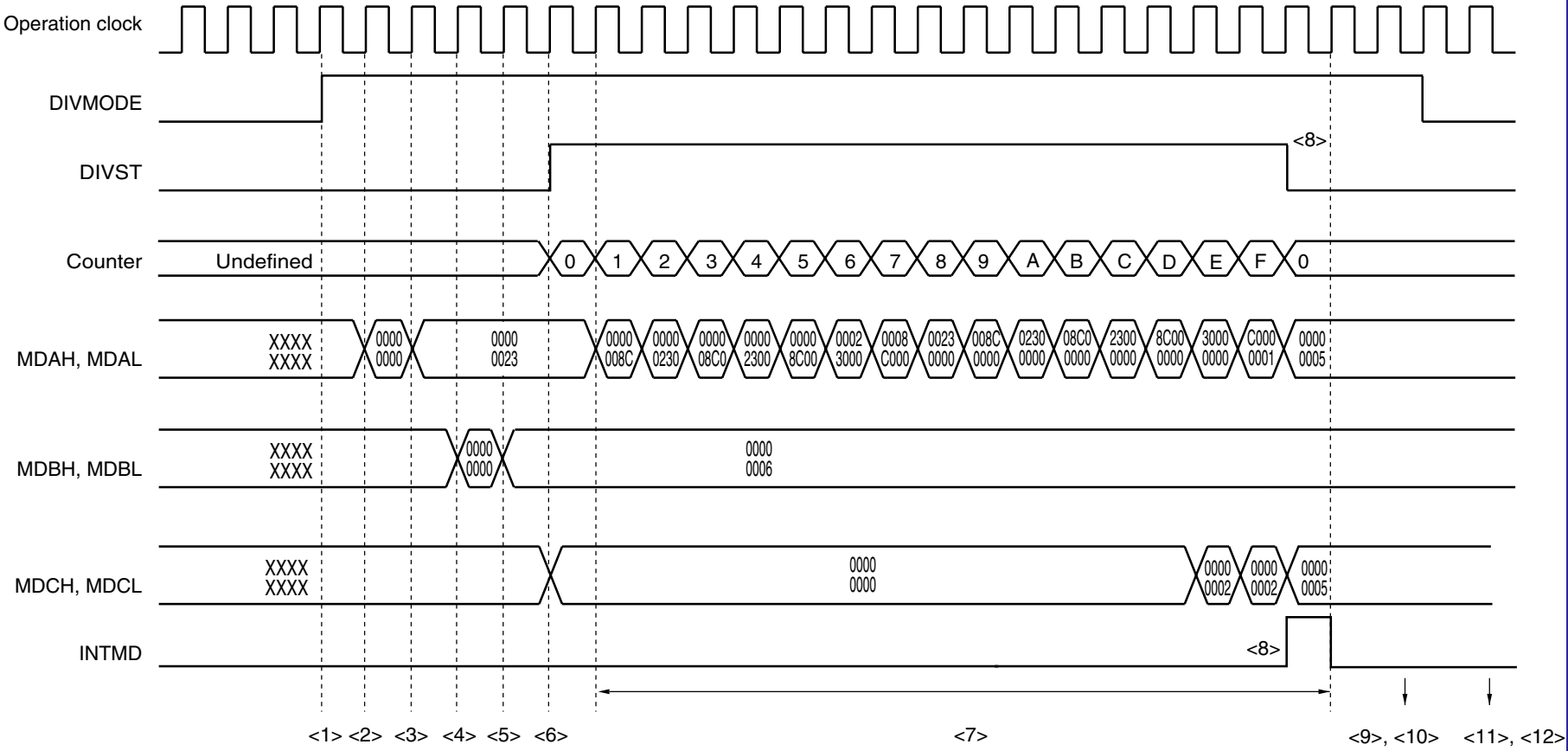
16.4.2 Division operation

- Initial setting
  - <1> Set bit 7 (DIVMODE) of the multiplication/division control register (MDUC) to 1.
  - <2> Set the dividend (higher 16 bits) to the multiplication/division data register A (H) (MDAH).
  - <3> Set the dividend (lower 16 bits) to the multiplication/division data register A (L) (MDAL).
  - <4> Set the divisor (higher 16 bits) to the multiplication/division data register B (H) (MDBH).
  - <5> Set the divisor (lower 16 bits) to the multiplication/division data register B (L) (MDBL).
  - <6> Set bit 0 (DIVST) of MDUC to 1.
    (There is no preference in the order of executing steps <2> to <5>.)
- During operation processing
  - <7> The operation will end when one of the following processing is completed.
    - A wait of at least 16 clocks (The operation will end when 16 clocks have been issued.)
    - A check whether DIVST has been cleared
    - Generation of a division completion interrupt (INTMD)

    (The read values of MDBL, MDBH, MDCH, and MDCL during operation processing are not guaranteed.)
- Operation end
  - <8> DIVST is cleared (0) and an interrupt request signal (INTMD) is generated (end of operation).
  - <9> Read the quotient (lower 16 bits) from MDAL.
  - <10> Read the quotient (higher 16 bits) from MDAH.
  - <11> Read the remainder (lower 16 bits) from multiplication/division data register C (L) (MDCL).
  - <12> Read the remainder (higher 16 bits) from the multiplication/division data register C (H) (MDCH).
    (There is no preference in the order of executing steps <9> to <12>.)
- Next operation
  - <13> To execute multiplication operation next, start from the “Initial setting” in **16.4.1 Multiplication operation**.
  - <14> To execute division operation next, start from the “Initial setting” for division operation.

Remark Steps <1> to <12> correspond to <1> to <12> in **Figure 16-7**.

Figure 16-7. Timing Diagram of Division Operation (Example: 35 ÷ 6 = 5, Remainder 5)



CHAPTER 17 DMA CONTROLLER

The DMA (Direct Memory Access) controller is mounted onto all products of 78K0R/Kx3-A microcontrollers.

Data can be automatically transferred between the peripheral hardware supporting DMA, SFRs, and internal RAM without via CPU.

As a result, the normal internal operation of the CPU and data transfer can be executed in parallel with transfer between the SFR and internal RAM, and therefore, a large capacity of data can be processed. In addition, real-time control using communication, timer, and A/D can also be realized.

17.1 Functions of DMA Controller

- Number of DMA channels: 2
- Transfer unit: 8 or 16 bits
- Maximum transfer unit: 1024 times
- Transfer type: 2-cycle transfer (One transfer is processed in 2 clocks and the CPU stops during that processing.)
- Transfer mode: Single-transfer mode
- Transfer target: Between SFR and internal RAM
- Transfer request: Selectable from the following peripheral hardware interrupts
  - A/D converter
  - Serial interface (CSI00, CSI10, UART0, UART1, UART3, or IIC10)
  - Timer (channel 0, 1, 4, or 5 of timer array unit 0)

Here are examples of functions using DMA.

- Successive transfer of serial interface
- Batch transfer of analog data
- Capturing A/D conversion result at fixed interval
- Capturing port value at fixed interval

17.2 Configuration of DMA Controller

The DMA controller includes the following hardware.

Table 17-1. Configuration of DMA Controller

Item	Configuration
Address registers	• DMA SFR address registers 0, 1 (DSA0, DSA1) • DMA RAM address registers 0, 1 (DRA0, DRA1)
Count register	• DMA byte count registers 0, 1 (DBC0, DBC1)
Control registers	• DMA mode control registers 0, 1 (DMC0, DMC1) • DMA operation control register 0, 1 (DRC0, DRC1)

(1) DMA SFR address register n (DSAn)

This is an 8-bit register that is used to set an SFR address that is the transfer source or destination of DMA channel n.

Set the lower 8 bits of the SFR addresses FFF00H to FFFFFH[Note].

This register is not automatically incremented but fixed to a specific value.

In the 16-bit transfer mode, the least significant bit is ignored and is treated as an even address.

DSAn can be read or written in 8-bit units. However, it cannot be written during DMA transfer.

Reset signal generation clears this register to 00H.

Note Except for address FFFFEH because the PMC register is allocated there.

Figure 17-1. Format of DMA SFR Address Register n (DSAn)

Address: FFFB0H (DSA0), FFFB1H (DSA1) After reset: 00H R/W

	7	6	5	4	3	2	1	0
DSAn								

Remark n: DMA channel number (n = 0, 1)

(2) DMA RAM address register n (DRAn)

This is a 16-bit register that is used to set a RAM address that is the transfer source or destination of DMA channel n.

Addresses of the internal RAM area other than the general-purpose registers (FEF00H to FFEDFH in the case of the μPD78F1016) can be set to this register.

Set the lower 16 bits of the RAM address.

This register is automatically incremented when DMA transfer has been started. It is incremented by +1 in the 8-bit transfer mode and by +2 in the 16-bit transfer mode. DMA transfer is started from the address set to this DRAn register. When the data of the last address has been transferred, DRAn stops with the value of the last address +1 in the 8-bit transfer mode, and the last address +2 in the 16-bit transfer mode.

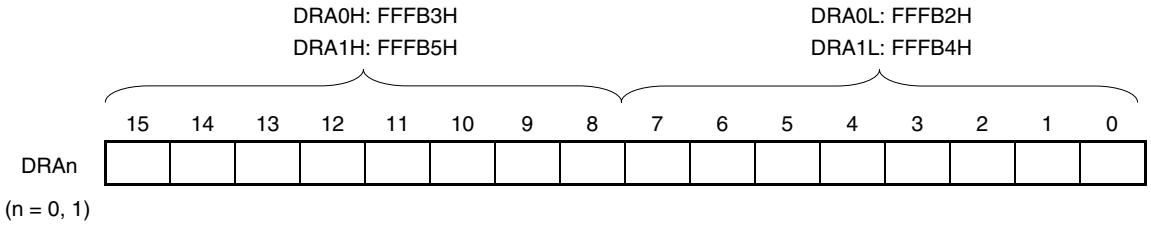
In the 16-bit transfer mode, the least significant bit is ignored and is treated as an even address.

DRAn can be read or written in 8-bit or 16-bit units. However, it cannot be written during DMA transfer.

Reset signal generation clears this register to 0000H.

Figure 17-2. Format of DMA RAM Address Register n (DRAn)

Address: FFFB2H, FFFB3H (DRA0), FFFB4H, FFFB5H (DRA1) After reset: 0000H R/W



Remark n: DMA channel number (n = 0, 1)

(3) DMA byte count register n (DBCn)

This is a 10-bit register that is used to set the number of times DMA channel n executes transfer. Be sure to set the number of times of transfer to this DBCn register before executing DMA transfer (up to 1024 times).

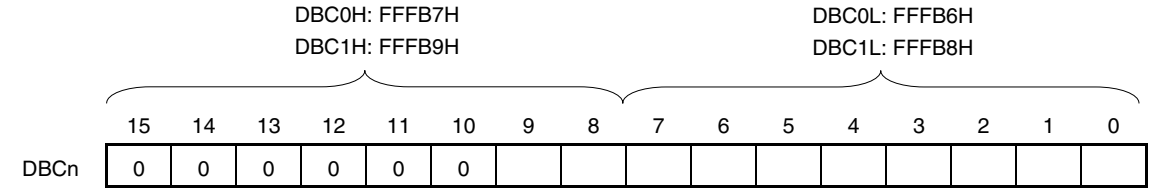
Each time DMA transfer has been executed, this register is automatically decremented. By reading this DBCn register during DMA transfer, the remaining number of times of transfer can be learned.

DBCn can be read or written in 8-bit or 16-bit units. However, it cannot be written during DMA transfer.

Reset signal generation clears this register to 0000H.

Figure 17-3. Format of DMA Byte Count Register n (DBCn)

Address: FFFB6H, FFFB7H (DBC0), FFFB8H, FFFB9H (DBC1) After reset: 0000H R/W



(n = 0, 1)

DBCn[9:0]	Number of Times of Transfer (When DBCn is Written)	Remaining Number of Times of Transfer (When DBCn is Read)
000H	1024	Completion of transfer or waiting for 1024 times of DMA transfer
001H	1	Waiting for remaining one time of DMA transfer
002H	2	Waiting for remaining two times of DMA transfer
003H	3	Waiting for remaining three times of DMA transfer
• • •	• • •	• • •
3FEH	1022	Waiting for remaining 1022 times of DMA transfer
3FFH	1023	Waiting for remaining 1023 times of DMA transfer

Cautions 1. Be sure to clear bits 15 to 10 to "0".

2. If the general-purpose register is specified or the internal RAM space is exceeded as a result of continuous transfer, the general-purpose register or SFR space are written or read, resulting in loss of data in these spaces. Be sure to set the number of times of transfer that is within the internal RAM space.

Remark n: DMA channel number (n = 0, 1)

17.3 Registers Controlling DMA Controller

DMA controller is controlled by the following registers.

- DMA mode control register n (DMCn)
- DMA operation control register n (DRCn)

Remark n: DMA channel number (n = 0, 1)

(1) DMA mode control register n (DMCn)

DMCn is a register that is used to set a transfer mode of DMA channel n. It is used to select a transfer direction, data size, setting of pending, and start source. Bit 7 (STGn) is a software trigger that starts DMA.

Rewriting bits 6, 5, and 3 to 0 of DMCn is prohibited during operation (when DSTn = 1).

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

Reset signal generation clears this register to 00H.

Figure 17-4. Format of DMA Mode Control Register n (DMCn) (1/2)

Address: FFFBAH (DMC0), FFFBBH (DMC1) After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0

STGn[Note 1]	DMA transfer start software trigger
0	No trigger operation
1	DMA transfer is started when DMA operation is enabled (DENn = 1).
DMA transfer is started by writing 1 to STGn when DMA operation is enabled (DENn = 1). When this bit is read, 0 is always read.	

DRSn	Selection of DMA transfer direction
0	SFR to internal RAM
1	Internal RAM to SFR

DSn	Specification of transfer data size for DMA transfer
0	8 bits
1	16 bits

DWAITn[Note 2]	Pending of DMA transfer
0	Executes DMA transfer upon DMA start request (not held pending).
1	Holds DMA start request pending if any.
DMA transfer that has been held pending can be started by clearing the value of DWAITn to 0. It takes 2 clocks to actually hold DMA transfer pending when the value of DWAITn is set to 1.	

Notes 1. The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 bits values.

<R> **2.** When DMA transfer is held pending while using both DMA channels, be sure to hold the DMA transfer pending for both channels (by setting the DWAIT0 and DWAIT1 bits to 1).

Remark n: DMA channel number (n = 0, 1)

Figure 17-4. Format of DMA Mode Control Register n (DMCn) (2/2)

Address: FFFBAH (DMC0), FFFBBH (DMC1) After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0

IFCn 3	IFCn 2	IFCn 1	IFCn 0	Selection of DMA start source[Note]	
				Trigger signal	Trigger contents
0	0	0	0	–	Disables DMA transfer by interrupt. (Only software trigger is enabled.)
0	0	1	0	INTTM00	Timer channel 0 interrupt
0	0	1	1	INTTM01	Timer channel 1 interrupt
0	1	0	0	INTTM04	Timer channel 4 interrupt
0	1	0	1	INTTM05	Timer channel 5 interrupt
0	1	1	0	INTST0	UART0 transmission end interrupt
				INTCSI00	CSI00 transfer end interrupt
0	1	1	1	INTSR0	UART0 reception end interrupt
1	0	0	0	INTST1	UART1 transmission end interrupt
				INTCSI10	CSI10 transfer end interrupt
				INTIIC10	IIC10 transfer end interrupt
1	0	0	1	INTSR1	UART1 reception end interrupt
1	0	1	0	INTST3	UART3 transmission end interrupt
1	0	1	1	INTSR3	UART3 reception end interrupt
1	1	0	0	INTAD	A/D conversion end interrupt
Other than above				Setting prohibited	

Note The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 values.

Remark n: DMA channel number (n = 0, 1)

(2) DMA operation control register n (DRCn)

DRCn is a register that is used to enable or disable transfer of DMA channel n.

Rewriting bit 7 (DENn) of this register is prohibited during operation (when DSTn = 1).

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

Reset signal generation clears this register to 00H.

Figure 17-5. Format of DMA Operation Control Register n (DRCn)

Address: FFFBCH (DRC0), FFFBDH (DRC1) After reset: 00H R/W

Symbol	<7>	6	5	4	3	2	1	<0>
DRCn	DENn	0	0	0	0	0	0	DSTn

DENn	DMA operation enable flag
0	Disables operation of DMA channel n (stops operating cock of DMA).
1	Enables operation of DMA channel n.
DMAC waits for a DMA trigger when DSTn = 1 after DMA operation is enabled (DENn = 1).	

DSTn	DMA transfer mode flag
0	DMA transfer of DMA channel n is completed.
1	DMA transfer of DMA channel n is not completed (still under execution).
DMAC waits for a DMA trigger when DSTn = 1 after DMA operation is enabled (DENn = 1). When a software trigger (STGn) or the start source trigger set by IFCn3 to IFCn0 is input, DMA transfer is started. When DMA transfer is completed after that, this bit is automatically cleared to 0. Write 0 to this bit to forcibly terminate DMA transfer under execution.	

Cautions 1. The DSTn flag is automatically cleared to 0 when a DMA transfer is completed. Writing the DENn flag is enabled only when DSTn = 0. When a DMA transfer is terminated without waiting for generation of the interrupt (INTDMAn) of DMAn, therefore, set DSTn to 0 and then DENn to 0 (for details, refer to 17.5.7 Forced termination by software).

2. When the FSEL bit of the OSMC register has been set to 1, do not enable (DENn = 1) DMA operation for at least three clocks after the setting.

Remark n: DMA channel number (n = 0, 1)

17.4 Operation of DMA Controller

17.4.1 Operation procedure

<1> The DMA controller is enabled to operate when DENn = 1. Before writing the other registers, be sure to set DENn to 1. Use 80H to write with an 8-bit manipulation instruction.

<2> Set an SFR address, a RAM address, the number of times of transfer, and a transfer mode of DMA transfer to the DSAn, DRAn, CBCn, and DMCn registers.

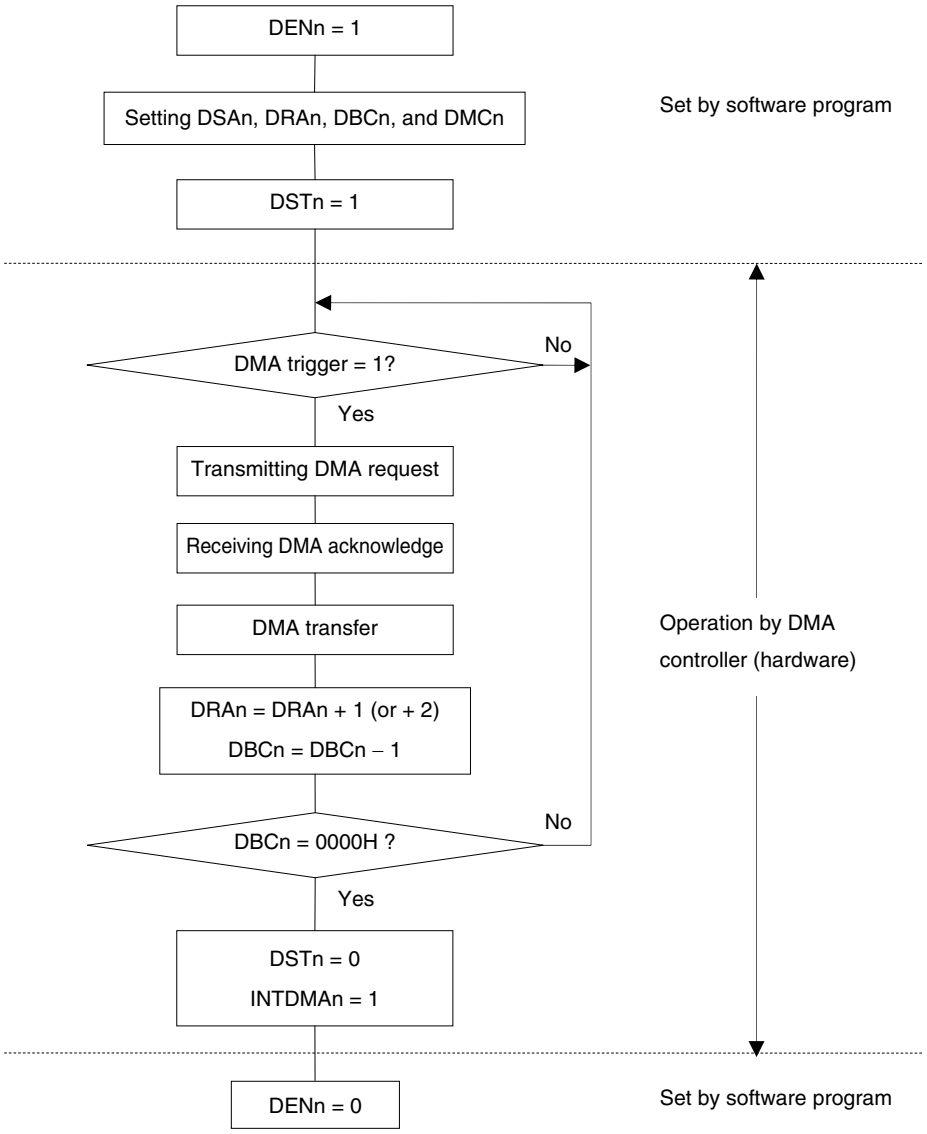
<3> The DMA controller waits for a DMA trigger when DSTn = 1. Use 81H to write with an 8-bit manipulation instruction.

<4> When a software trigger (STGn) or a start source trigger specified by IFCn3 to IFCn0 is input, a DMA transfer is started.

<5> Transfer is completed when the number of times of transfer set by the DBCn register reaches 0, and transfer is automatically terminated by occurrence of an interrupt (INTDMAn).

<6> Stop the operation of the DMA controller by clearing DENn to 0 when the DMA controller is not used.

Figure 17-6. Operation Procedure



Remark n: DMA channel number (n = 0, 1)

17.4.2 Transfer mode

The following four modes can be selected for DMA transfer by using bits 6 and 5 (DRSn and DSn) of the DMCn register.

DRSn	DSn	DMA Transfer Mode
0	0	Transfer from SFR of 1-byte data (fixed address) to RAM (address is incremented by +1)
0	1	Transfer from SFR of 2-byte data (fixed address) to RAM (address is incremented by +2)
1	0	Transfer from RAM of 1-byte data (address is incremented by +1) to SFR (fixed address)
1	1	Transfer from RAM of 2-byte data (address is incremented by +2) to SFR (fixed address)

By using these transfer modes, up to 1024 bytes of data can be consecutively transferred by using the serial interface, data resulting from A/D conversion can be consecutively transferred, and port data can be scanned at fixed time intervals by using a timer.

17.4.3 Termination of DMA transfer

When DBCn = 00H and DMA transfer is completed, the DSTn bit is automatically cleared to 0. An interrupt request (INTDMAn) is generated and transfer is terminated.

When the DSTn bit is cleared to 0 to forcibly terminate DMA transfer, the DBCn and DRAn registers hold the value when transfer is terminated.

The interrupt request (INTDMAn) is not generated if transfer is forcibly terminated.

Remark n: DMA channel number (n = 0, 1)

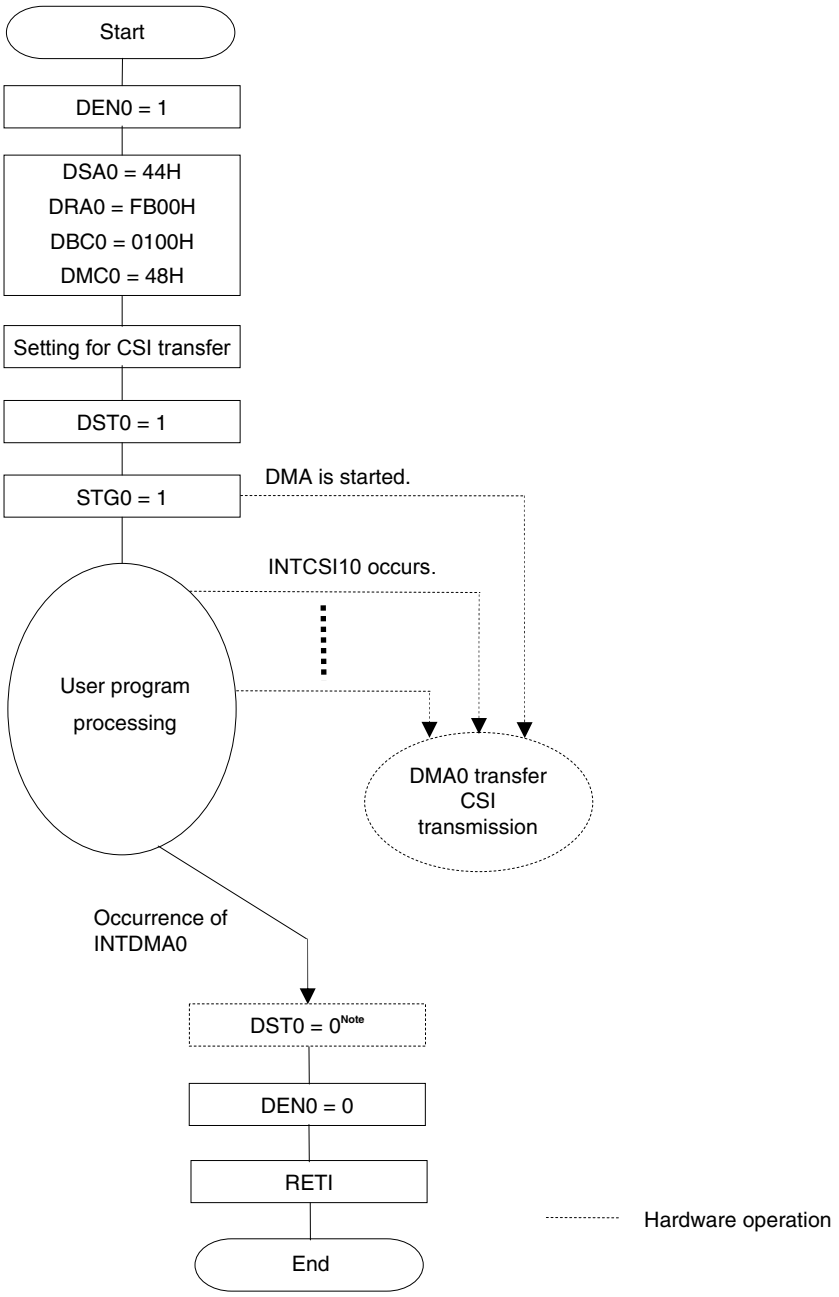
17.5 Example of Setting of DMA Controller

17.5.1 CSI consecutive transmission

A flowchart showing an example of setting for CSI consecutive transmission is shown below.

- Consecutive transmission of CSI10
- DMA channel 0 is used for DMA transfer.
- DMA start source: INTCSI10 (software trigger (STG0) only for the first start source)
- Interrupt of CSI10 is specified by IFC03 to IFC00 (bits 3 to 0 of the DMC0 register) = 1000B.
- Transfers FFB00H to FFBFFH (256 bytes) of RAM to FFF44H of the transmit buffer (SIO10) of CSI.

Figure 17-7. Example of Setting for CSI Consecutive Transmission



Note The DST0 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN0 flag is enabled only when DST0 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA0 (INTDMA0), set DST0 to 0 and then DEN0 to 0 (for details, refer to **17.5.7 Forced termination by software**).

<R> The fist trigger for consecutive transmission is not started by the interrupt of CSI. In this example, it start by a software trigger.

CSI transmission of the second time and onward is automatically executed.

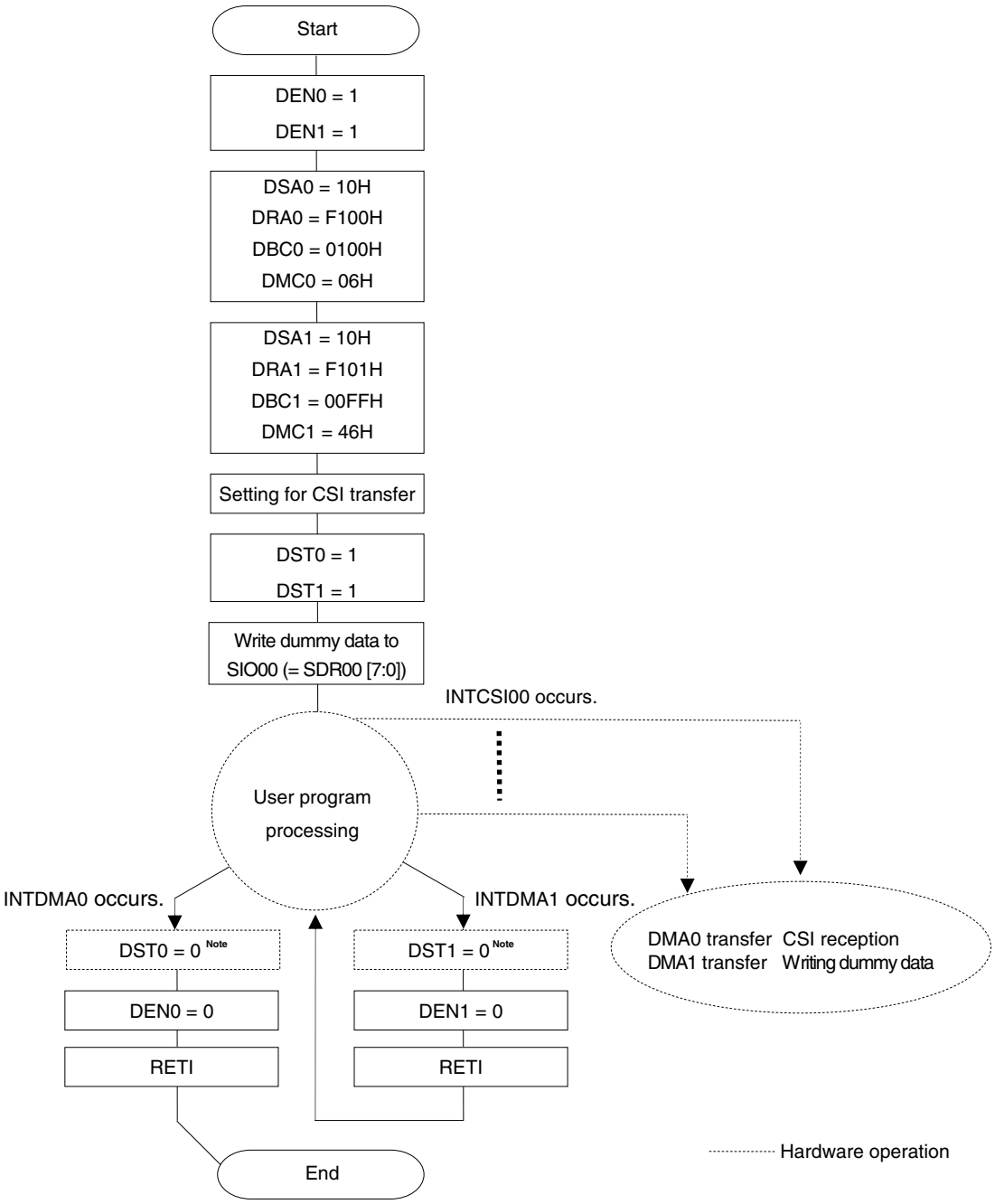
A DMA interrupt (INTDMA0) occurs when the last transmit data has been written to the data register.

<R> **17.5.2 CSI master reception**

A flowchart showing an example of setting for CSI master reception is shown below.

- Master reception (256 bytes) of CSI00
- DMA channel 0 is used to read received data and DMA channel 1 is used to write dummy data.
- DMA start source: INTCSI00
  (If the same start source is specified for DMA channels 0 and 1, the data of channel 0 is transferred, and then that of channel 1.)
- Interrupt of CSI00 is specified by IFC03 to IFC00 = IFC13 to IFC10 (bits 3 to 0 of the DMCn register) = 0110B.
- Data is transferred (received) from FFF10H of the CSI data register (SIO00) to FF100H to FF1FFH of RAM (256 bytes). (In successive reception mode, the data that is to be received when the first buffer empty interrupt occurs is invalid because the data has not been received.)
- Transfers dummy data FF101H to FF1FFH (255 bytes) of RAM to FFF10H of the data register (SIO00) of CSI.
  (Dummy data is written to the first byte by using software (an instruction).)

Figure 17-8. Example of Setting of Consecutively Capturing A/D Conversion Results



Note The DSTn flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DENn flag is enabled only when DSTn = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMAn (INTDMAn), set DSTn to 0 and then DENn to 0 (for details, refer to **17.5.7 Forcible termination by software**).

Because no CSI interrupt is generated when reception starts during CSI master reception, dummy data is written using software in this example.

The received data is automatically transferred from the first byte (In successive reception mode, the data that is to be received when the first buffer empty interrupt occurs is invalid because the valid data has not been received.).

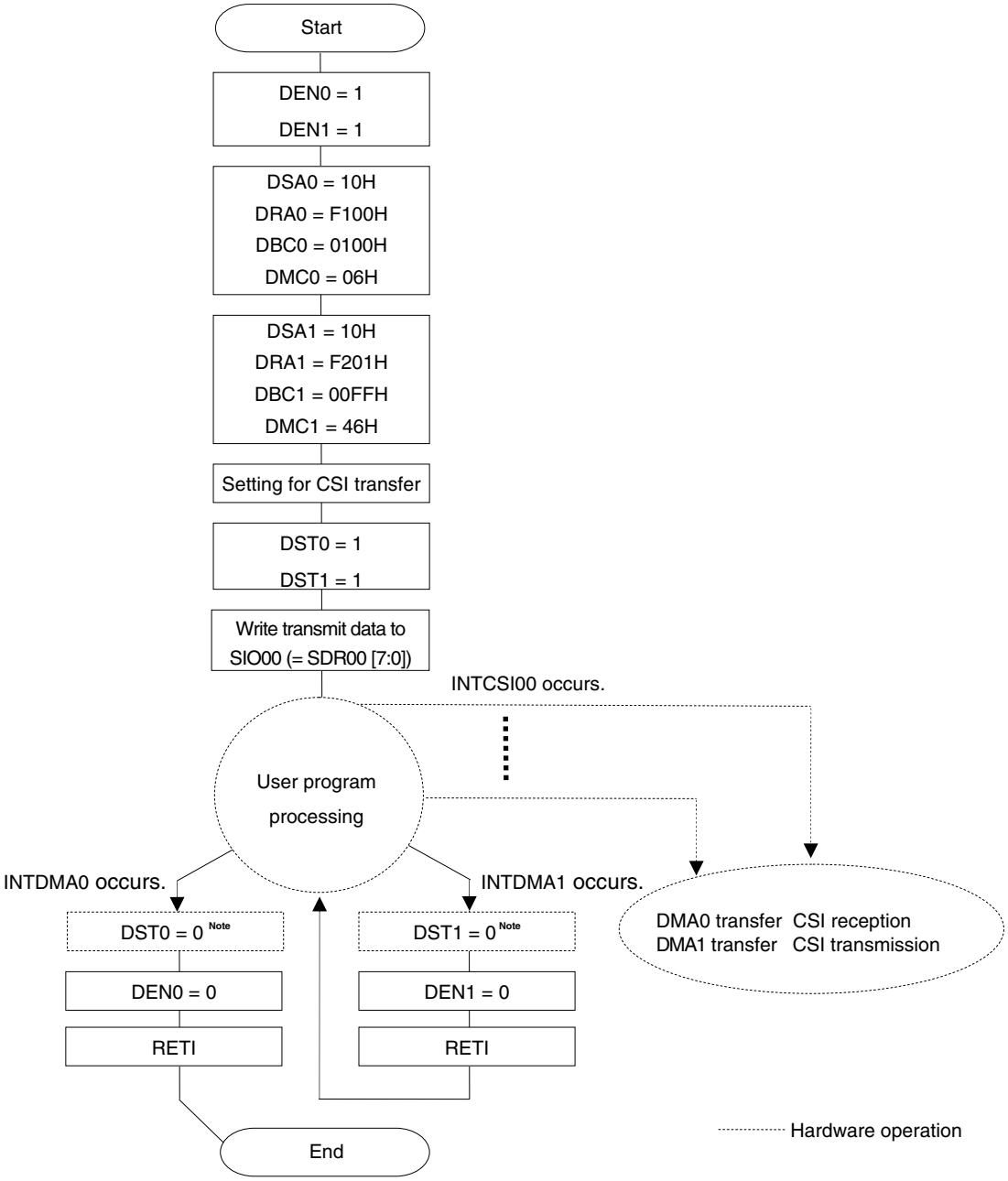
A DMA interrupt (INTDMA1) occurs when the last dummy data has been writing to the data register. A DMA interrupt (INTDMA0) occurs when the last received data has been read from the data register. To restart the DMA transfer, the CSI transfer must be completed.

<R> **17.5.3 CSI transmission/reception**

A flowchart showing an example of setting for CSI transmission/reception is shown below.

- Transmission/reception (256 bytes) of CSI00
- DMA channel 0 is used to read received data and DMA channel 1 is used to write transmit data.
- DMA start source: INTCSI00
  (If the same start source is specified for DMA channels 0 and 1, the data of channel 0 is transferred, and then that of channel 1)
- Interrupt of CSI00 is specified by IFC03 to IFC00 = IFC13 to IFC10 (bits 3 to 0 of the DMCn register) = 0110B.
- Data is transferred (received) from FFF10H of the CSI data register (SIO00) to FF100H to FF1FFH of RAM (256 bytes). (In successive transmission/reception mode, the data that is to be received when the first buffer empty interrupt occurs is invalid because the data has not been received.)
- Transfers FF201H to FF2FFH (255 bytes) of RAM to FFF10H of the data register (SIO00) of CSI (transmission)
  (Transmit data is written to the first byte by using software (an instruction).)

Figure 17-9. Setting Example of CSI Transmission/reception



Note The DSTn flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DENn flag is enabled only when DSTn = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMAn (INTDMAn), set DSTn to 0 and then DENn to 0 (for details, refer to **17.5.7 Forcible termination by software**).

During CSI transfers, no CSI interrupt is generated when the transmitted data of the first byte is written. Therefore, the transmitted data is written using software in this example. The data of the second and following bytes is automatically transmitted.

The received data is automatically transferred from the first byte. (In successive transmission/reception, the data that is to be received when the first buffer empty interrupt occurs is invalid because the valid data has not been received.)

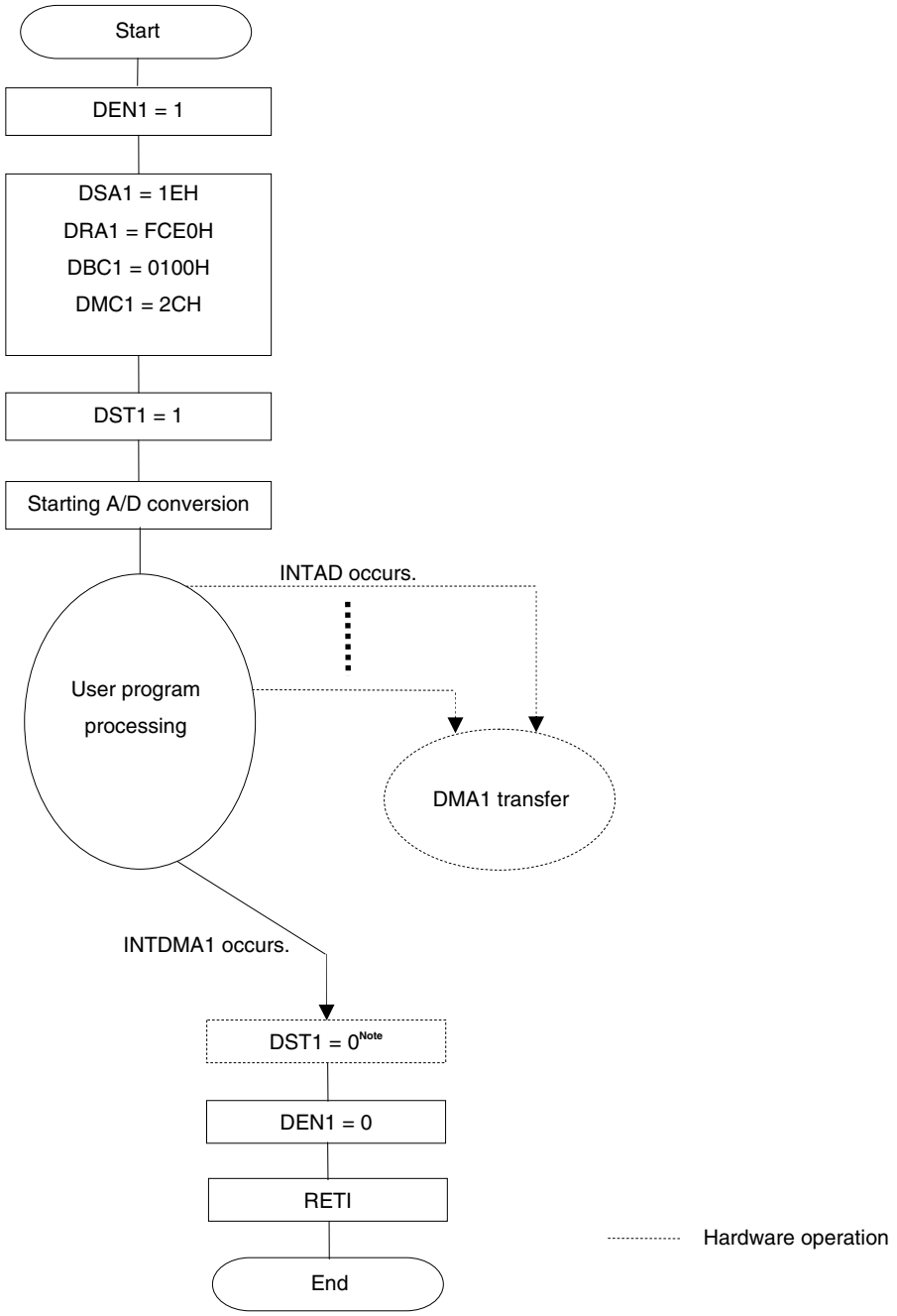
A DMA interrupt (INTDMA1) occurs when the last transmit data has been writing to the data register. A DMA interrupt (INTDMA0) occurs when the last received data has been read from the data register. To restart the DMA transfer, the CSI transfer must be completed.

17.5.4 Consecutive capturing of A/D conversion results

A flowchart of an example of setting for consecutively capturing A/D conversion results is shown below.

- Consecutive capturing of A/D conversion results.
- DMA channel 1 is used for DMA transfer.
- DMA start source: INTAD
- Interrupt of A/D is specified by IFC13 to IFC10 (bits 3 to 0 of the DMC1 register) = 1100B.
- Transfers FFF1EH and FFF1FH (2 bytes) of the 12-bit A/D conversion result register to 512 bytes of FFCE0H to FFEDFH of RAM.

Figure 17-10. Example of Setting of Consecutively Capturing A/D Conversion Results



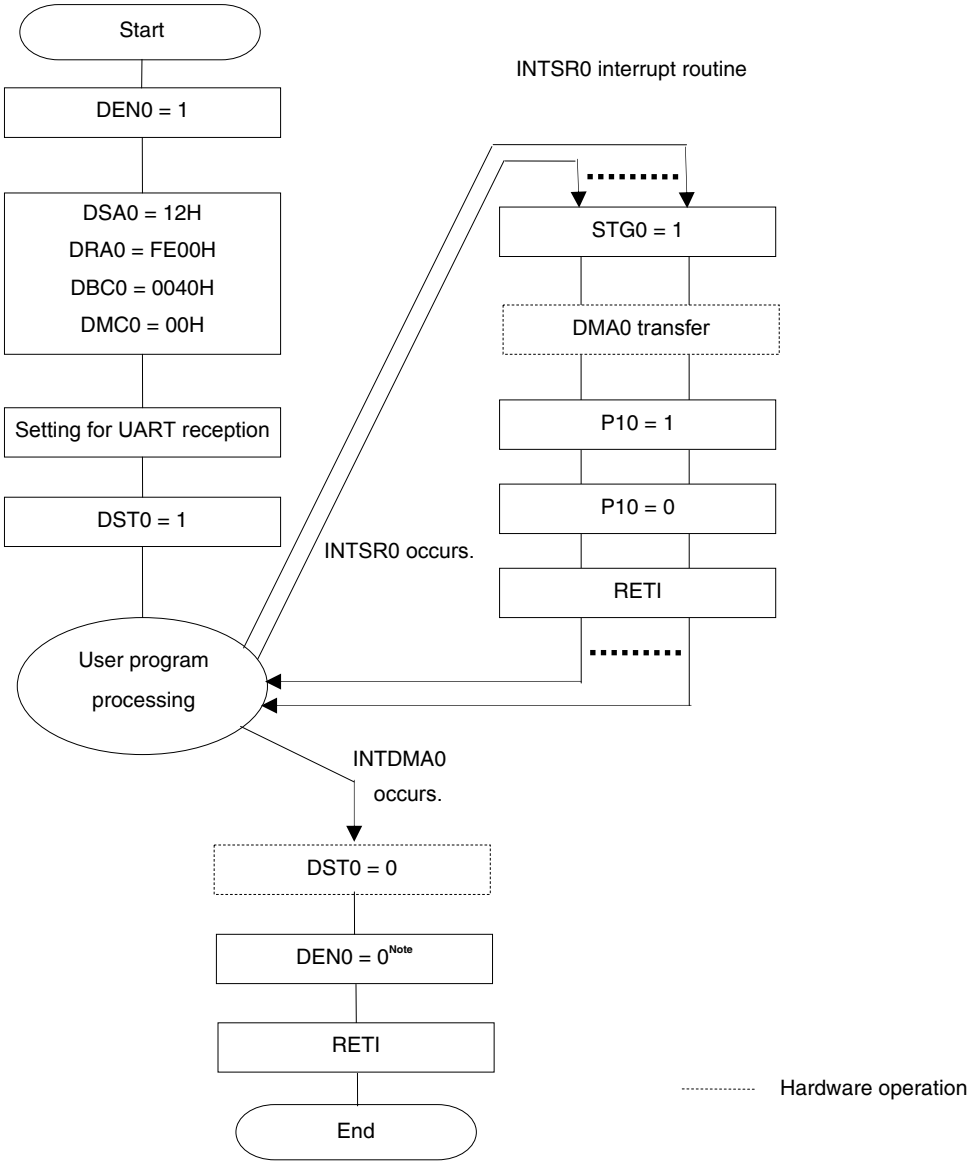
Note The DST1 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN1 flag is enabled only when DST1 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA1 (INTDMA1), set DST1 to 0 and then DEN1 to 0 (for details, refer to **17.5.7 Forced termination by software**).

17.5.5 UART consecutive reception + ACK transmission

A flowchart illustrating an example of setting for UART consecutive reception + ACK transmission is shown below.

- Consecutively receives data from UART0 and outputs ACK to P10 on completion of reception.
- DMA channel 0 is used for DMA transfer.
- DMA start source: Software trigger (DMA transfer on occurrence of an interrupt is disabled.)
- Transfers FFF12H of UART receive data register 0 (RXD0) to 64 bytes of FFE00H to FFE3FH of RAM.

Figure 17-11. Example of Setting for UART Consecutive Reception + ACK Transmission



Note The DST0 flag is automatically cleared to 0 when a DMA transfer is completed.
Writing the DEN0 flag is enabled only when DST0 = 0. To terminate a DMA transfer without waiting for occurrence of the interrupt of DMA0 (INTDMA0), set DST0 to 0 and then DEN0 to 0 (for details, refer to **17.5.7 Forced termination by software**).

Remark This is an example where a software trigger is used as a DMA start source.
If ACK is not transmitted and if only data is consecutively received from UART, the UART reception end interrupt (INTSR0) can be used to start DMA for data reception.

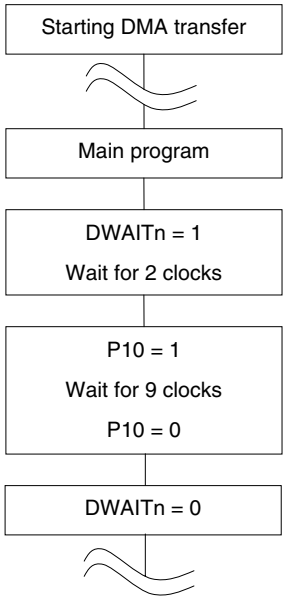
17.5.6 Holding DMA transfer pending by DWAITn

When DMA transfer is started, transfer is performed while an instruction is executed. At this time, the operation of the CPU is stopped and delayed for the duration of 2 clocks. If this poses a problem to the operation <R> of the set system, a DMA transfer can be held pending by setting the DWAITn bit to 1. The DMA transfer for a transfer trigger that occurred while DMA transfer was held pending is executed after the pending status is canceled. However, because only one transfer trigger can be held pending for each channel, even if multiple transfer triggers occur for one channel during the pending status, only one DMA transfer is executed after the pending status is canceled.

To output a pulse with a width of 10 clocks of the operating frequency from the P10 pin, for example, the clock width increases to 12 if a DMA transfer is started midway. In this case, the DMA transfer can be held pending by setting the DWAITn bit to 1.

After setting the DWAITn bit to 1, it takes two clocks until a DMA transfer is held pending.

Figure 17-12. Example of Setting for Holding DMA Transfer Pending by DWAITn



<R> **Caution When DMA transfer is held pending while using both DMA channels, be sure to held the DMA transfer pending for both channels (by setting DWAIT0 and DWAIT1 to 1). If the DMA transfer of one channel is executed while that of the other channel is held pending, DMA transfer might not be held pending for the latter channel.**

Remarks 1. n: DMA channel number (n = 0, 1)

2. 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

17.5.7 Forced termination by software

After DSTn is set to 0 by software, it takes up to 2 clocks until a DMA transfer is actually stopped and DSTn is set to 0. To forcibly terminate a DMA transfer by software without waiting for occurrence of the interrupt (INTDMAn) of DMAn, therefore, perform either of the following processes.

<When using one DMA channel>

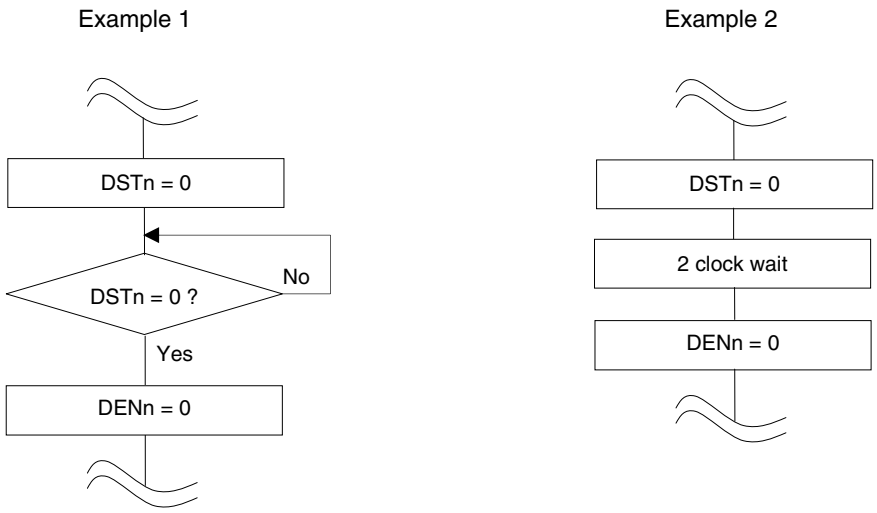
- Set the DSTn bit to 0 (use DRCn = 80H to write with an 8-bit manipulation instruction) by software, confirm by polling that the DSTn bit has actually been cleared to 0, and then set the DENn bit to 0 (use DRCn = 00H to write with an 8-bit manipulation instruction).

- Set the DSTn bit to 0 (use DRCn = 80H to write with an 8-bit manipulation instruction) by software and then set the DENn bit to 0 (use DRCn = 00H to write with an 8-bit manipulation instruction) two or more clocks after.

<When using both DMA channels>

<R> - To forcibly terminate DMA transfer by software when using both DMA channels (by setting DSTn to 0), clear the DSTn bit to 0 after the DMA transfer is held pending by setting the DWAIT0 and DWAIT1 bits of both channels to 1. Next, clear the DWAIT0 and DWAIT1 bits of both channels to 0 to cancel the pending status, and then clear the DENn bit to 0.

Figure 17-13. Forced Termination of DMA Transfer (1/2)



Remarks 1. n: DMA channel number (n = 0, 1)

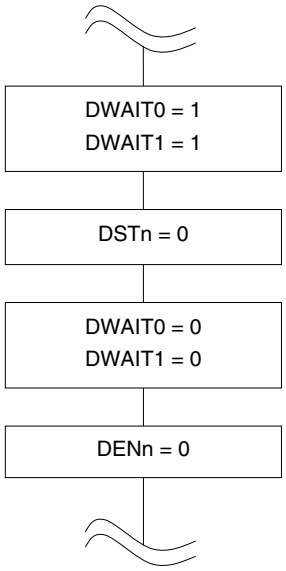
2. 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

Figure 17-13. Forced Termination of DMA Transfer (2/2)

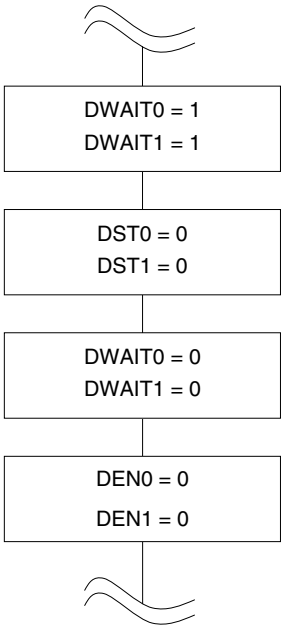
<R>

Example 3

- Procedure for forcibly terminating the DMA transfer for one channel if both channels are used



- Procedure for forcibly terminating the DMA transfer for both channels if both channels are used



Caution In example 3, the system is not required to wait two clock cycles after the DWAITn bit is set to 1. In addition, the system does not have to wait two clock cycles after clearing the DSTn bit to 0, because more than two clock cycles elapse from when the DSTn bit is cleared to 0 to when the DENn bit is cleared to 0.

Remarks 1. n: DMA channel number (n = 0, 1)

2. 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

17.6 Cautions on Using DMA Controller

(1) Priority of DMA

<R> During DMA transfer, a request from the other DMA channel is held pending even if generated. The pending DMA transfer is started after the ongoing DMA transfer is completed. If two DMA requests are generated at the same time, however, DMA channel 0 takes priority over DMA channel 1.

If a DMA request and an interrupt request are generated at the same time, the DMA transfer takes precedence, and then interrupt servicing is executed.

(2) DMA response time

The response time of DMA transfer is as follows.

Table 17-2. Response Time of DMA Transfer

<R>

	Minimum Time	Maximum Time
Response time	3 clocks	10 clocks[Note]

Note The maximum time necessary to execute an instruction from internal RAM is 16 clock cycles.

Cautions 1. The above response time does not include the two clock cycles required for a DMA transfer.

2. When executing a DMA pending instruction (see 17.6 (4)), the maximum response time is extended by the execution time of that instruction to be held pending.

3. Do not specify successive transfer triggers for a channel within a period equal to the maximum response time plus one clock cycle, because they might be ignored.

Remark 1 clock: $1/f_{CLK}$ (f_{CLK}: CPU clock)

(3) Operation in standby mode

The DMA controller operates as follows in the standby mode.

Table 17-3. DMA Operation in Standby Mode

Status	DMA Operation
HALT mode	Normal operation
STOP mode	Stops operation. If DMA transfer and STOP instruction execution contend, DMA transfer may be damaged. Therefore, stop DMA before executing the STOP instruction.

(4) DMA pending instruction

Even if a DMA request is generated, DMA transfer is held pending immediately after the following instructions.

- CALL !addr16
- <R> CALL $!addr20
- CALL !!addr20
- CALL rp
- CALLT [addr5]
- BRK
- <R> Bit manipulation instructions for registers IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H and PSW each.

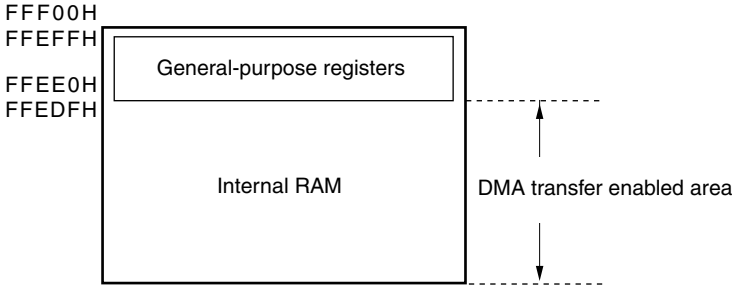
(5) Operation if address in general-purpose register area or other than those of internal RAM area is specified

The address indicated by DRA0n is incremented during DMA transfer. If the address is incremented to an address in the general-purpose register area or exceeds the area of the internal RAM, the following operation is performed.

- In mode of transfer from SFR to RAM
  The data of that address is lost.

- In mode of transfer from RAM to SFR
  Undefined data is transferred to SFR.

In either case, malfunctioning may occur or damage may be done to the system. Therefore, make sure that the address is within the internal RAM area other than the general-purpose register area.



CHAPTER 18 INTERRUPT FUNCTIONS

18.1 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 priority of vectored interrupt servicing. For the priority order, see **Table 18-1**.

A standby release signal is generated and STOP and HALT modes are released.

External interrupt requests and internal interrupt requests are provided as maskable interrupts.

External: 12, internal: 33

(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.

18.2 Interrupt Sources and Configuration

The interrupt sources consist of maskable interrupts and software interrupts. In addition, they also have up to five reset sources (see **Table 18-1**). 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 18-1. Interrupt Source List (1/3)

Interrupt Type	Internal/ External	Basic Configuration Type Note 1	Default Priority Note 2	Interrupt Source		Vector Table Address
				Name	Trigger	
Maskable	Internal	(A)	0	INTWDTI	Watchdog timer interval Note 3 (75% of overflow time)	00004H
			1	INTLVI	Low-voltage detection Note 4	00006H
	External	(B)	2	INTP0	Pin input edge detection	00008H
			3	INTP1		0000AH
			4	INTP2		0000CH
			5	INTP3		0000EH
			6	INTP4		00010H
			7	INTP5		00012H
	Internal	(A)	8	INTST3	End of UART3 transmission	00014H
			9	INTSR3	End of UART3 reception	00016H
			10	INTSRE3	UART3 reception error occurrence	00018H
			11	INTDMA0	End of DMA0 transfer	0001AH
			12	INTDMA1	End of DMA1 transfer	0001CH
			13	INTST0	End of UART0 transmission	0001EH
				INTCSI00	End of CSI00 communication	
			14	INTSR0	End of UART0 reception	00020H
			15	INTSRE0	UART0 reception error occurrence	00022H
			16	INTST1	End of UART1 transmission	00024H
				INTCSI10	End of CSI10 communication	
				INTIIC10	End of IIC10 communication	
			17	INTSR1	End of UART1 reception	00026H
			18	INTSRE1	UART1 reception error occurrence	00028H
			19	INTIICA	End of IICA communication	0002AH
			20	INTTM00	End of timer channel 0 count or capture	0002CH
			21	INTTM01	End of timer channel 1 count or capture	0002EH
			22	INTTM02	End of timer channel 2 count or capture	00030H
			23	INTTM03	End of timer channel 3 count or capture	00032H

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 44 indicates the lowest priority.

2. Basic configuration types (A) to (C) correspond to (A) to (C) in Figure 18-1.

3. When bit 7 (WDTINT) of the option byte (000C0H) is set to 1.

4. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is cleared to 0.

Table 18-1. Interrupt Source List (2/3)

Interrupt Type	Internal/ External	Basic Configuration Type Note 1	Default Priority Note 2	Interrupt Source		Vector Table Address
				Name	Trigger	
Maskable	Internal	(A)	24	INTAD	End of A/D conversion	00034H
			25	INTRTC	Fixed-cycle signal of real-time counter/alarm match detection	00036H
			26	INTRTCI	Interval signal detection of real-time counter	00038H
			27	INTST2	End of UART2 transmission/	0003CH
				INTCSI20	End of CSI20 communication/	
				INTIIC20	End of IIC20 communication	
			28	INTSR2	End of UART2 reception	0003EH
			29	INTSRE2	UART2 reception error occurrence	00040H
			30	INTTM04	End of timer channel 4 count or capture	00042H
			31	INTTM05	End of timer channel 5 count or capture	00044H
			32	INTTM06	End of timer channel 6 count or capture	00046H
			33	INTTM07	End of timer channel 7 count or capture	00048H
	External	(B)	34	INTP6	Pin input edge detection	0004AH
			35	INTP7		0004CH
			36	INTP8		0004EH
			37	INTP9		00050H
			38	INTP10		00052H
			39	INTP11		00054H

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 44 indicates the lowest priority.

2. Basic configuration types (A) to (C) correspond to (A) to (C) in Figure 18-1.

Table 18-1. Interrupt Source List (3/3)

Interrupt Type	Internal/ External	Basic Configuration Type Note 1	Default Priority Note 2	Interrupt Source		Vector Table Address
				Name	Trigger	
Maskable	Internal	(A)	40	INTTM10	End of timer channel 10 count or capture	00056H
			41	INTTM11	End of timer channel 11 count or capture	00058H
			42	INTTM12	End of timer channel 12 count or capture	0005AH
			43	INTTM13	End of timer channel 13 count or capture	0005CH
			44	INTMD	End of multiply/divide operation	0005EH
Software	–	(C)	–	BRK	Execution of BRK instruction	0007EH
Reset	–	–	–	RESET	$\overline{\text{RESET}}$ pin input	00000H
				POC	Power-on-clear	
				LVI	Low-voltage detection Note 3	
				WDT	Overflow of watchdog timer	
				TRAP	Execution of illegal instruction Note 4	

Notes 1. The default priority determines the sequence of interrupts if two or more maskable interrupts occur simultaneously. Zero indicates the highest priority and 44 indicates the lowest priority.

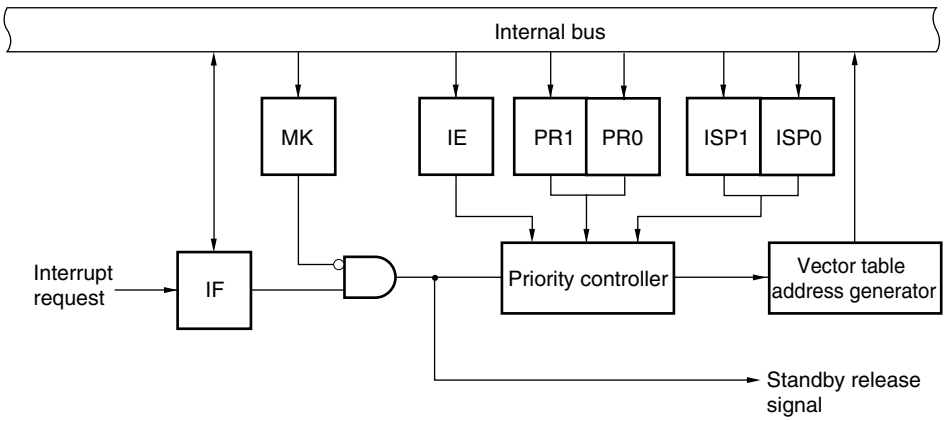
2. Basic configuration types (A) to (C) correspond to (A) to (C) in Figure 18-1.

3. When bit 1 (LVIMD) of the low-voltage detection register (LVIM) is set to 1.

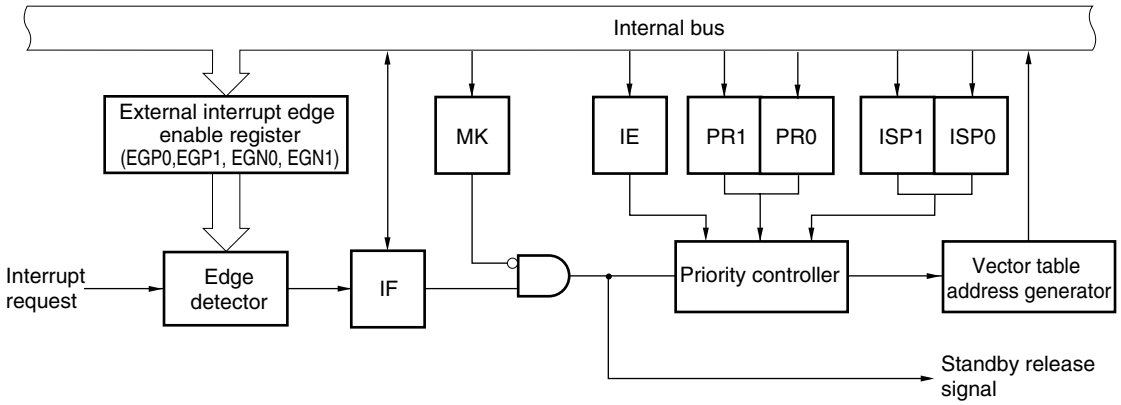
4. When the instruction code in FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

Figure 18-1. Basic Configuration of Interrupt Function (1/2)

(A) Internal maskable interrupt



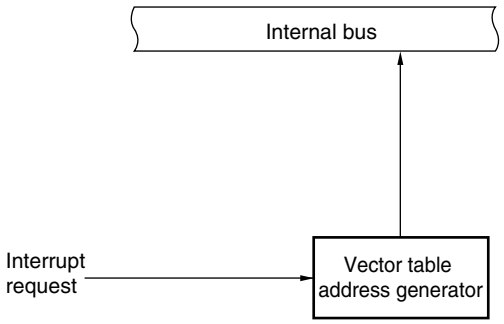
(B) External maskable interrupt (INTPn: n = 0 to 11)



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

Figure 18-1. Basic Configuration of Interrupt Function (2/2)

(C) Software interrupt



18.3 Registers Controlling Interrupt Functions

The following 6 types of registers are used to control the interrupt functions.

- Interrupt request flag registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)
- Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)
- Priority specification flag registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H)
- External interrupt rising edge enable registers (EGP0, EGP1)
- External interrupt falling edge enable registers (EGN0, EGN1)
- Program status word (PSW)

Table 18-2 shows a list of interrupt request flags, interrupt mask flags, and priority specification flags corresponding to interrupt request sources.

Table 18-2. Flags Corresponding to Interrupt Request Sources (1/2)

Interrupt Source	Interrupt Request Flag		Interrupt Mask Flag		Priority Specification Flag	
		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	
INTST3	STIF3	IF0H	STMK3	MK0H	STPR03, STPR13	PR00H, PR10H
INTSR3	SRIF3		SRMK3		SRPR03, SRPR13	
INTSRE3	SREIF3		SREMK3		SREPR03, SREPR13	
INTDMA0	DMAIF0		DMAMK0		DMAPR00, DMAPR10	
INTDMA1	DMAIF1		DMAMK1		DMAPR01, DMAPR11	
INTST0 Note	STIF0 Note		STMK0 Note		STPR00, STPR10 Note	
INTCSI00 Note	CSIIF00 Note		CSIMK00 Note		CSIPR000, CSIPR100 Note	
INTSR0	SRIF0		SRMK0		SRPR00, SRPR10	
INTSRE0	SREIF0		SREMK0		SREPR00, SREPR10	

Note Do not use UART0 and CSI00 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTST0 and INTCSI00 is generated, bit 5 of IF0H is set to 1. Bit 5 of MK0H, PR00H, and PR10H supports these two interrupt sources.

Table 18-2. Flags Corresponding to Interrupt Request Sources (2/2)

Interrupt Source	Interrupt Request Flag	Register	Interrupt Mask Flag	Register	Priority Specification Flag	Register
INTST1[Note 1]	STIF1[Note 1]	IF1L	STMK1[Note 1]	MK1L	STPR01, STPR11[Note 1]	PR01L, PR11L
INTCSI10[Note 1]	CSIIF10[Note 1]		CSIMK10[Note 1]		CSIPR010, CSIPR110[Note1]	
INTIIC10[Note 1]	IICIF10[Note 1]		IICMK10[Note 1]		IICPR010, IICPR110[Note1]	
INTSR1	SRIF1		SRMK1		SRPR01, SRPR11	
INTSRE1	SREIF1		SREMK1		SREPR01, SREPR11	
INTIICA	IICAIF		IICAMK		IICAPR0, IICAPR1	
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	
INTRTCI	RTCIIF		RTCIMK		RTCIPR0, RTCIPR1	
INTST2[Note 2]	STIF2[Note 2]		STMK2[Note 2]		STPR02, STPR12[Note 2]	
INTCSI20[Note 2]	CSIIF20[Note 2]		CSIMK20[Note 2]		CSIPR020, CSIPR120[Note2]	
INTIIC20[Note 2]	IICIF20[Note 2]		IICMK20[Note 2]		IICPR020, IICPR120[Note2]	
INTSR2	SRIF2		SRMK2		SRPR02, SRPR12	
INTSRE2	SREIF2		SREMK2		SREPR02, SREPR12	
INTTM04	TMIF04		TMMK04		TMPR004, TMPR104	
INTTM05	TMIF05	IF2L	TMMK05	MK2L	TMPR005, TMPR105	PR02L, PR12L
INTTM06	TMIF06		TMMK06		TMPR006, TMPR106	
INTTM07	TMIF07		TMMK07		TMPR007, TMPR107	
INTP6	PIF6		PMK6		PPR06, PPR16	
INTP7	PIF7		PMK7		PPR07, PPR17	
INTP8	PIF8		PMK8		PPR08, PPR18	
INTP9	PIF9		PMK9		PPR09, PPR19	
INTP10	PIF10		PMK10		PPR010, PPR110	
INTP11	PIF11	IF2H	PMK11	MK2H	PPR011, PPR111	PR02H, PR12H
INTTM10	TMIF10		TMMK10		TMPR010, TMPR110	
INTTM11	TMIF11		TMMK11		TMPR011, TMPR111	
INTTM12	TMIF12		TMMK12		TMPR012, TMPR112	
INTTM13	TMIF13		TMMK13		TMPR013, TMPR113	
INTMD	MDIF		MDMK		MDPR0, MDPR1	

Notes 1. Do not use UART1, CSI10, and IIC10 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTST1, INTCSI10, and INTIIC10 is generated, bit 0 of IF1L is set to 1. Bit 0 of MK1L, PR01L, and PR11L supports these three interrupt sources.

2. Do not use UART2, CSI20, and IIC20 at the same time because they share flags for the interrupt request sources. If one of the interrupt sources INTST2, INTCSI20, and INTIIC20 is generated, bit 4 of IF1H is set to 1. Bit 4 of MK1H, PR01H, and PR11H supports these three interrupt sources.

(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.

IF0L, IF0H, IF1L, IF1H, IF2L, and IF2H can be set by a 1-bit or 8-bit memory manipulation instruction. When IF0L and IF0H, IF1L and IF1H, and IF2L and IF2H 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.

Cautions 1. When operating a timer, serial interface, or A/D converter after standby release, operate it once after clearing the interrupt request flag. An interrupt request flag may be set by noise.

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 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.

Remark If an instruction that writes data to this register is executed, the number of instruction execution clocks increases by 2 clocks.

Figure 18-2. Format of Interrupt Request Flag Registers (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)

Address: FFFE0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0L	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0	LVIIF	WDTIIF

Address: FFFE1H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF0H	SREIF0	SRIF0	CSIIF00 STIF0	DMAIF1	DMAIF0	SREIF3	SRIF3	STIF3

Address: FFFE2H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF1L	TMIF03	TMIF02	TMIF01	TMIF00	IICAIF	SREIF1	SRIF1	CSIIF10 IICIF10 STIF1

Address: FFFE3H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
IF1H	TMIF04	SREIF2	SRIF2	CSIIF20 IICIF20 STIF2	0	RTCIIF	RTCIF	ADIF

Address: FFFD0H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IF2L	PIF10	PIF9	PIF8	PIF7	PIF6	TMIF07	TMIF06	TMIF05

Address: FFFD1H After reset: 00H R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
IF2H	0	0	MDIF	TMIF13	TMIF12	TMIF11	TMIF10	PIF11

XXIFX	Interrupt request flag
0	No interrupt request signal is generated
1	Interrupt request is generated, interrupt request status

Caution Be sure to clear bit 3 of IF1H, bits 6, 7 of IF2H to 0.

(2) Interrupt mask flag registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)

The interrupt mask flags are used to enable/disable the corresponding maskable interrupt servicing.

MK0L, MK0H, MK1L, MK1H, MK2L, and MK2H can be set by a 1-bit or 8-bit memory manipulation instruction.

When MK0L and MK0H, MK1L and MK1H, and MK2L and MK2H 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 clocks.

Figure 18-3. Format of Interrupt Mask Flag Registers (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)

Address: FFFE4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0L	PMK5	PMK4	PMK3	PMK2	PMK1	PMK0	LVIMK	WDTIMK

Address: FFFE5H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK0H	SREMK0	SRMK0	CSIMK00 STMK0	DMAMK1	DMAMK0	SREMK3	SRMK3	STMK3

Address: FFFE6H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK1L	TMMK03	TMMK02	TMMK01	TMMK00	IICAMK	SREMK1	SRMK1	CSIMK10 IICMK10 STMK1

Address: FFFE7H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
MK1H	TMMK04	SREMK2	SRMK2	CSIMK20 IICMK20 STMK2	1	RTCIMK	RTCMK	ADMK

Address: FFFD4H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
MK2L	PMK10	PMK9	PMK8	PMK7	PMK6	TMMK07	TMMK06	TMMK05

Address: FFFD5H After reset: FFH R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
MK2H	1	1	MDMK	TMMK13	TMMK12	TMMK11	TMMK10	PMK11

XXMKX	Interrupt servicing control
0	Interrupt servicing enabled
1	Interrupt servicing disabled

Caution Be sure to set bit 3 of MK1H, bits 6, 7 of MK2H to 1.

(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).

PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H can be set by a 1-bit or 8-bit memory manipulation instruction. If PR00L and PR00H, PR01L and PR01H, PR02L and PR02H, PR10L and PR10H, PR11L and PR11H, and PR12L and PR12H 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 clocks.

Figure 18-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (1/2)

Address: FFFE8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00L	PPR05	PPR04	PPR03	PPR02	PPR01	PPR00	LVIPR0	WDTIPR0

Address: FFFECH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10L	PPR15	PPR14	PPR13	PPR12	PPR11	PPR10	LVIPR1	WDTIPR1

Address: FFFE9H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR00H	SREPR00	SRPR00	CSIPR000 STPR00	DMAPR01	DMAPR00	SREPR03	SRPR03	STPR03

Address: FFFEDH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR10H	SREPR10	SRPR10	CSIPR100 STPR10	DMAPR11	DMAPR10	SREPR13	SRPR13	STPR13

Address: FFFEAH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR01L	TMPR003	TMPR002	TMPR001	TMPR000	IICAPR0	SREPR01	SRPR01	CSIPR010 IICPR010 STPR01

Address: FFFEEH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR11L	TMPR103	TMPR102	TMPR101	TMPR100	IICAPR1	SREPR11	SRPR11	CSIPR110 IICPR110 STPR11

Figure 18-4. Format of Priority Specification Flag Registers (PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H) (2/2)

Address: FFFEBH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
PR01H	TMPR004	SREPR02	SRPR02	CSIPR020 IICPR020 STPR02	1	RTCIPR0	RTCPR0	ADPR0

Address: FFFEFH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	3	<2>	<1>	<0>
PR11H	TMPR104	SREPR12	SRPR12	CSIPR120 IICPR120 STPR12	1	RTCIPR1	RTCPR1	ADPR1

Address: FFFD8H After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR02L	PPR010	PPR09	PPR08	PPR07	PPR06	TMPR007	TMPR006	TMPR005

Address: FFFDCH After reset: FFH R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
PR12L	PPR110	PPR19	PPR18	PPR17	PPR16	TMPR107	TMPR106	TMPR105

Address: FFFD9H After reset: FFH R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
PR02H	1	1	MDPR0	TMPR013	TMPR012	TMPR011	TMPR010	PPR011

Address: FFFDDH After reset: FFH R/W

Symbol	7	6	<5>	<4>	<3>	<2>	<1>	<0>
PR12H	1	1	MDPR1	TMPR113	TMPR112	TMPR111	TMPR110	PPR111

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 Be sure to set bit 3 of PR01H and PR11H, bits 6, 7 of PR02H and PR12H to 1.

(4) External interrupt rising edge enable registers (EGP0, EGP1), external interrupt falling edge enable registers (EGN0, EGN1)

These registers specify the valid edge for INTP0 to INTP11.

EGP0, EGP1, EGN0, and EGN1 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears these registers to 00H.

Figure 18-5. Format of External Interrupt Rising Edge Enable Registers (EGP0, EGP1) and External Interrupt Falling Edge Enable Registers (EGN0, EGN1)

Address: FFF38H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP0	EGP7	EGP6	EGP5	EGP4	EGP3	EGP2	EGP1	EGP0

Address: FFF39H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN0	EGN7	EGN6	EGN5	EGN4	EGN3	EGN2	EGN1	EGN0

Address: FFF3AH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGP1	0	0	0	0	EGP11	EGP10	EGP9	EGP8

Address: FFF3BH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
EGN1	0	0	0	0	EGN11	EGN10	EGN9	EGN8

EGPn	EGNn	INTPn pin valid edge selection (n = 0 to 11)
0	0	Edge detection disabled
0	1	Falling edge
1	0	Rising edge
1	1	Both rising and falling edges

Table 18-3 shows the ports corresponding to EGPn and EGNn.

Table 18-3. Ports Corresponding to EGPn and EGNn

Detection Enable Register		Edge Detection Port	Interrupt Request Signal
EGP0	EGN0	P120	INTP0
EGP1	EGN1	P30	INTP1
EGP2	EGN2	P31	INTP2
EGP3	EGN3	P33	INTP3
EGP4	EGN4	P14	INTP4
EGP5	EGN5	P32	INTP5
EGP6	EGN6	P11	INTP6
EGP7	EGN7	P15	INTP7
EGP8	EGN8	P34	INTP8
EGP9	EGN9	P81	INTP9
EGP10	EGN10	P16	INTP10
EGP11	EGN11	P80	INTP11

Caution Select the port mode by clearing EGPn and EGNn to 0 because an edge may be detected when the external interrupt function is switched to the port function.

Remark n = 0 to 11

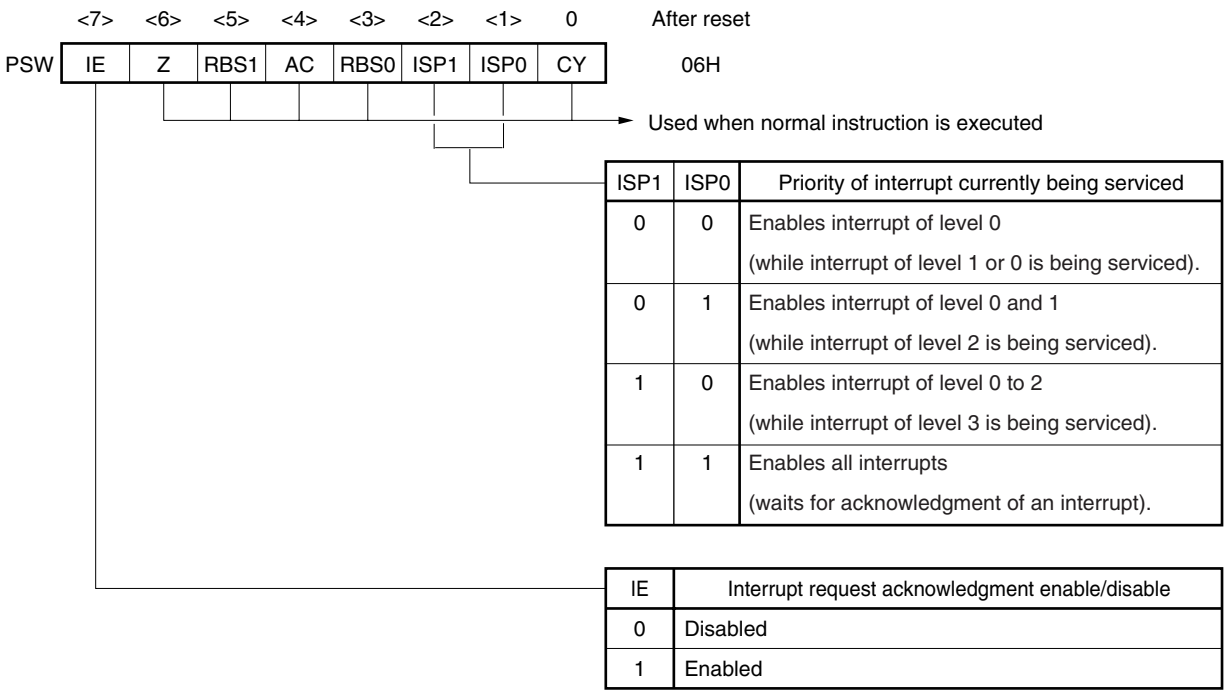
(5) Program status word (PSW)

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. If a maskable interrupt request is acknowledged, the contents of the priority specification flag of the acknowledged interrupt are 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 18-6. Configuration of Program Status Word



ISP1	ISP0	Priority of interrupt currently being serviced
0	0	Enables interrupt of level 0 (while interrupt of level 1 or 0 is being serviced).
0	1	Enables interrupt of level 0 and 1 (while interrupt of level 2 is being serviced).
1	0	Enables interrupt of level 0 to 2 (while interrupt of level 3 is being serviced).
1	1	Enables all interrupts (waits for acknowledgment of an interrupt).

IE	Interrupt request acknowledgment enable/disable
0	Disabled
1	Enabled

18.4 Interrupt Servicing Operations

18.4.1 Maskable interrupt acknowledgment

A maskable interrupt 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 18-4 below.

For the interrupt request acknowledgment timing, see **Figures 18-8** and **18-9**.

Table 18-4. Time from Generation of Maskable Interrupt Until Servicing

	Minimum Time	Maximum Time[Note]
Servicing time	9 clocks	14 clocks

Note If an interrupt request is generated just before the RET instruction, the wait time becomes longer.

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.

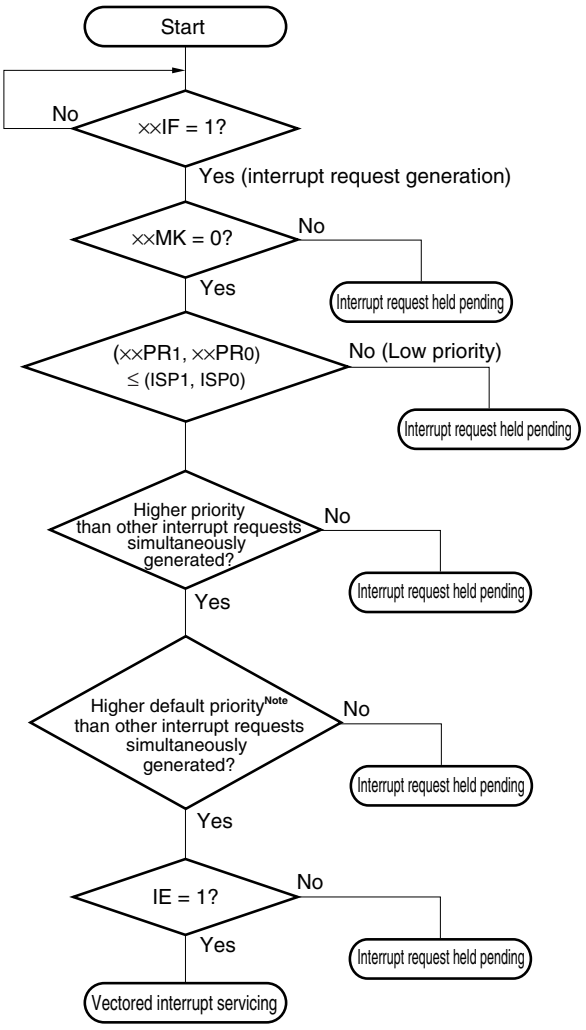
An interrupt request that is held pending is acknowledged when it becomes acknowledgeable.

Figure 18-7 shows the interrupt request acknowledgment algorithm.

If a maskable interrupt request is acknowledged, the contents are saved into the stacks in the order of PSW, then PC, the IE flag is reset (0), and the contents of the priority specification flag corresponding to the acknowledged interrupt are transferred to the ISP1 and ISP0 flags. The vector table data determined for each interrupt request is the loaded into the PC and branched.

Restoring from an interrupt is possible by using the RETI instruction.

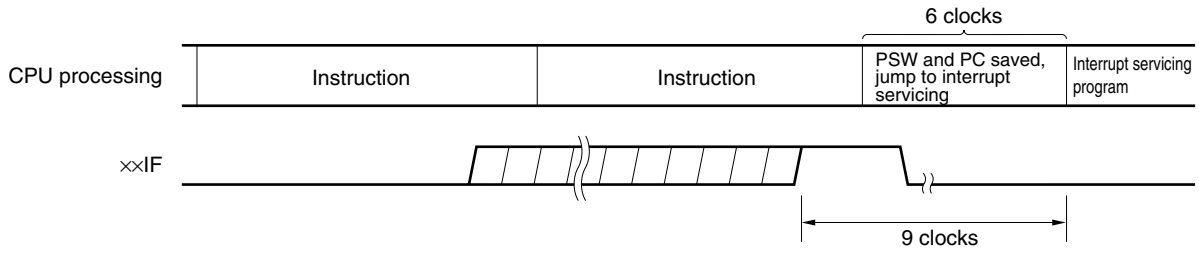
Figure 18-7. Interrupt Request Acknowledgment Processing Algorithm



××IF: Interrupt request flag
××MK: Interrupt mask flag
××PR0: Priority specification flag 0
××PR1: 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 18-6**)

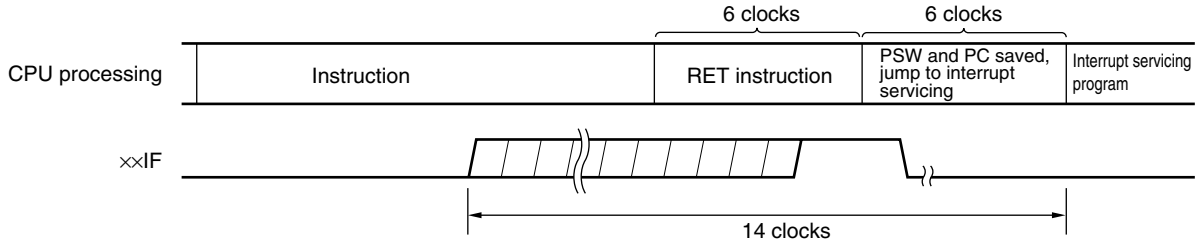
Note For the default priority, refer to **Table 18-1 Interrupt Source List**.

Figure 18-8. Interrupt Request Acknowledgment Timing (Minimum Time)



Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

Figure 18-9. Interrupt Request Acknowledgment Timing (Maximum Time)



Remark 1 clock: 1/f_{CLK} (f_{CLK}: CPU clock)

18.4.2 Software interrupt request acknowledgment

A software interrupt acknowledge 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 Do not use the RETI instruction for restoring from the software interrupt.

18.4.3 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 lower than that of the interrupt currently being serviced is generated during interrupt servicing, it is not acknowledged for multiple interrupt servicing. 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 18-5 shows relationship between interrupt requests enabled for multiple interrupt servicing and Figure 18-10 shows multiple interrupt servicing examples.

Table 18-5. Relationship Between Interrupt Requests Enabled for Multiple Interrupt Servicing During Interrupt Servicing

Multiple Interrupt Request / Interrupt Being Serviced		Maskable Interrupt Request								Software Interrupt Request
		Priority Level 0 (PR = 00)		Priority Level 1 (PR = 01)		Priority Level 2 (PR = 10)		Priority Level 3 (PR = 11)		
		IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	IE = 1	IE = 0	
Maskable interrupt	ISP1 = 0, ISP0 = 0	○	×	×	×	×	×	×	×	○
	ISP1 = 0, ISP0 = 1	○	×	○	×	×	×	×	×	○
	ISP1 = 1, ISP0 = 0	○	×	○	×	○	×	×	×	○
	ISP1 = 1, ISP0 = 1	○	×	○	×	○	×	○	×	○
Software interrupt		○	×	○	×	○	×	○	×	○

Remarks 1. ○: Multiple interrupt servicing enabled

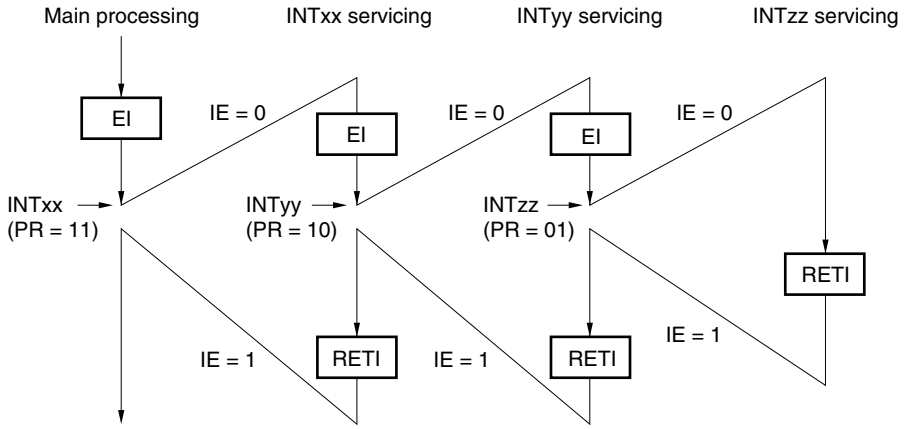
2. ×: Multiple interrupt servicing disabled

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.
IE = 0: Interrupt request acknowledgment is disabled.
IE = 1: Interrupt request acknowledgment is enabled.

4. PR is a flag contained in PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, and PR12H.
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 1 with ××PR1× = 1, ××PR0× = 1 (lower priority level)

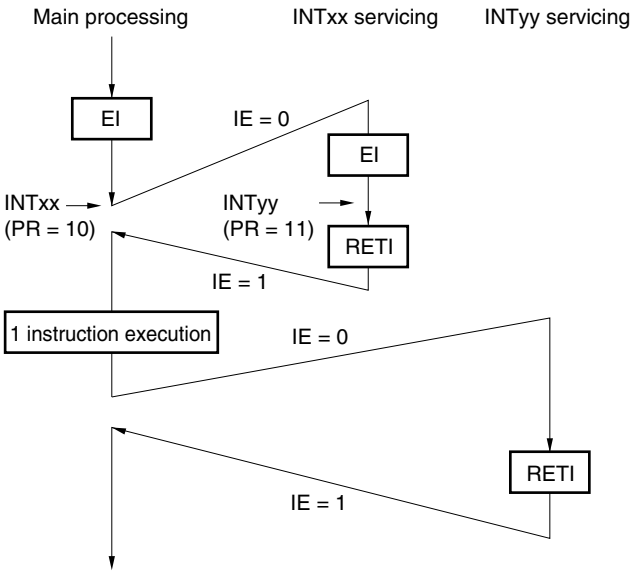
Figure 18-10. Examples of Multiple Interrupt Servicing (1/2)

Example 1. Multiple interrupt servicing occurs twice



During servicing of interrupt INTxx, two interrupt requests, INTyy and INTzz, are acknowledged, and multiple interrupt servicing takes place. Before each interrupt request is acknowledged, the EI instruction must always be issued to enable interrupt request acknowledgment.

Example 2. Multiple interrupt servicing does not occur due to priority control

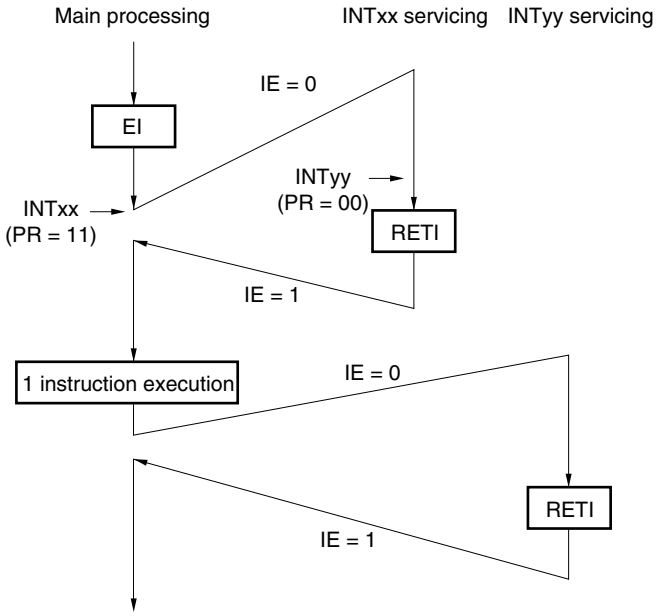


Interrupt request INTyy issued during servicing of interrupt INTxx is not acknowledged because its priority is lower than that of INTxx, and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with ××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 1 with ××PR1× = 1, ××PR0× = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

Figure 18-10. Examples of Multiple Interrupt Servicing (2/2)

Example 3. Multiple interrupt servicing does not occur because interrupts are not enabled



Interrupts are not enabled during servicing of interrupt INTxx (EI instruction is not issued), therefore, interrupt request INTyy is not acknowledged and multiple interrupt servicing does not take place. The INTyy interrupt request is held pending, and is acknowledged following execution of one main processing instruction.

PR = 00: Specify level 0 with ××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 1 with ××PR1× = 1, ××PR0× = 1 (lower priority level)
IE = 0: Interrupt request acknowledgment is disabled
IE = 1: Interrupt request acknowledgment is enabled.

18.4.4 Interrupt request hold

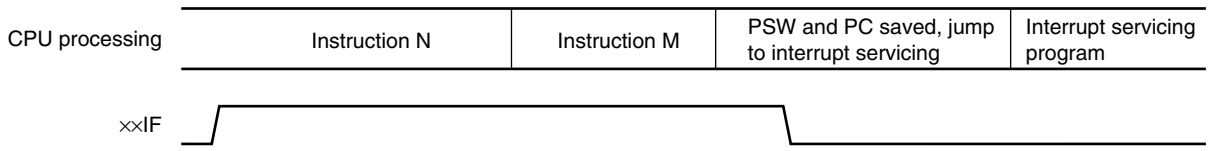
There are instructions where, even if an interrupt request is issued for them while another instruction is being executed, request acknowledgment is held pending until the end of execution of the next instruction. These instructions (interrupt request hold instructions) are listed below.

- MOV PSW, #byte
- MOV PSW, A
- MOV1 PSW. bit, CY
- SET1 PSW. bit
- CLR1 PSW. bit
- RETB
- RETI
- POP PSW
- BTCLR PSW. bit, $addr8
- EI
- DI
- SKC
- SKNC
- SKZ
- SKNZ
- Manipulation 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.

Caution The BRK instruction is not one of the above-listed interrupt request hold instructions. However, the software interrupt activated by executing the BRK instruction causes the IE flag to be cleared. Therefore, even if a maskable interrupt request is generated during execution of the BRK instruction, the interrupt request is not acknowledged.

Figure 18-11 shows the timing at which interrupt requests are held pending.

Figure 18-11. Interrupt Request Hold



Remarks 1. Instruction N: Interrupt request hold instruction

2. Instruction M: Instruction other than interrupt request hold instruction

3. The ××PR (priority level) values do not affect the operation of ××IF (interrupt request).

CHAPTER 19 STANDBY FUNCTION

19.1 Standby Function and Configuration

19.1.1 Standby function

The standby function reduces the operating current of the system, and the following two modes are available.

(1) HALT mode

<R> HALT instruction execution sets the HALT mode. In the HALT mode, the CPU operation clock is stopped. If the high-speed system clock oscillator, internal high-speed oscillator, 20 MHz internal high-speed oscillator, or subsystem clock oscillator is operating before the HALT mode is set, oscillation of each clock continues. In this mode, the operating current is not decreased as much as in the STOP mode, but the HALT mode is effective for restarting operation immediately upon interrupt request generation and carrying out intermittent operations frequently.

(2) STOP mode

STOP instruction execution sets the STOP mode. In the STOP mode, the high-speed system clock oscillator and internal high-speed oscillator stop, stopping the whole system, thereby considerably reducing the CPU operating current.

Because this mode can be cleared by an interrupt request, it enables intermittent operations to be carried out. However, because a wait time is required to secure the oscillation stabilization time after the STOP mode is released when the X1 clock is selected, select the HALT mode if it is necessary to start processing immediately upon interrupt request generation.

In either of these two modes, all the contents of registers, flags and data memory just before the standby mode is set are held. The I/O port output latches and output buffer statuses are also held.

Cautions 1. The STOP mode can be used only when the CPU is operating on the main system clock. The STOP mode cannot be set while the CPU operates with the subsystem clock. The HALT mode can be used when the CPU is operating on either the main system clock or the subsystem clock.

2. When shifting to the STOP mode, be sure to stop the peripheral hardware operation operating with main system clock before executing STOP instruction.

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 the A/D converter mode register (ADM) to 0 to stop the A/D conversion operation, and then execute the STOP instruction.

4. It can be selected by the option byte whether the internal low-speed oscillator continues oscillating or stops in the HALT or STOP mode. For details, see CHAPTER 24 OPTION BYTE.

5. The STOP instruction cannot be executed when the CPU operates on the 20 MHz internal high-speed oscillation clock. Be sure to execute the STOP instruction after shifting to internal high-speed oscillation clock operation.

19.1.2 Registers controlling standby function

The standby function is controlled by the following two registers.

- Oscillation stabilization time counter status register (OSTC)
- Oscillation stabilization time select register (OSTS)

Remark For the registers that start, stop, or select the clock, see **CHAPTER 5 CLOCK GENERATOR**.

(1) Oscillation stabilization time counter status register (OSTC)

This is the register that indicates the count status of the X1 clock oscillation stabilization time counter.

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

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

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

When reset is released (reset by $\overline{\text{RESET}}$ input, POC, LVI, WDT, and executing an illegal instruction), the STOP instruction and MSTOP (bit 7 of CSC register) = 1 clear this register to 00H.

Figure 19-1. 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	MOST 8	MOST 9	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.	102.4 μs min.	51.2 μs min.
1	1	1	1	0	0	0	0	2^{11}/fx min.	204.8 μs min.	102.4 μs min.
1	1	1	1	1	0	0	0	2^{13}/fx min.	819.2 μs min.	409.6 μs min.
1	1	1	1	1	1	0	0	2^{15}/fx min.	3.27 ms min.	1.64 ms min.
1	1	1	1	1	1	1	0	2^{17}/fx min.	13.11 ms min.	6.55 ms min.
1	1	1	1	1	1	1	1	2^{18}/fx min.	26.21 ms min.	13.11 ms min.

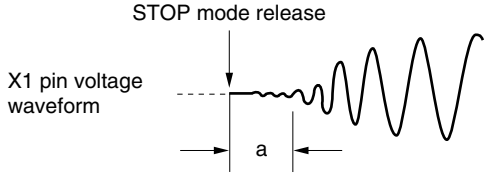
Cautions 1. After the above time has elapsed, the bits are set to 1 in order from MOST8 and remain 1.

2. The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS. If the STOP mode is entered and then released while the internal high-speed oscillation clock is being used as the CPU clock, set the oscillation stabilization time as follows.

- **Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS**

Note, therefore, that only the status up to the oscillation stabilization time set by OSTS is set to OSTC after STOP mode is released.

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

(2) Oscillation stabilization time select register (OSTS)

This register is used to select the X1 clock oscillation stabilization wait time when the STOP mode is released.

When the X1 clock is selected as the CPU clock, the operation waits for the time set using OSTS after the STOP mode is released.

When the internal high-speed oscillation clock is selected as the CPU clock, confirm with OSTC that the desired oscillation stabilization time has elapsed after the STOP mode is released. The oscillation stabilization time can be checked up to the time set using OSTC.

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

Reset signal generation sets this register to 07H.

Figure 19-2. Format of Oscillation Stabilization Time Select Register (OSTS)

Address: FFFA3H After reset: 07H R/W

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

OSTS2	OSTS1	OSTS0	Oscillation stabilization time selection		
				f_X = 10 MHz	f_X = 20 MHz
0	0	0	$2^8/f_X$	25.6 μs	Setting prohibited
0	0	1	$2^9/f_X$	51.2 μs	25.6 μs
0	1	0	$2^{10}/f_X$	102.4 μs	51.2 μs
0	1	1	$2^{11}/f_X$	204.8 μs	102.4 μs
1	0	0	$2^{13}/f_X$	819.2 μs	409.6 μs
1	0	1	$2^{15}/f_X$	3.27 ms	1.64 ms
1	1	0	$2^{17}/f_X$	13.11 ms	6.55 ms
1	1	1	$2^{18}/f_X$	26.21 ms	13.11 ms

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

2. Setting the oscillation stabilization time to 20 μs or less is prohibited.

3. Before changing the setting of the OSTS register, confirm that the count operation of the OSTC register is completed.

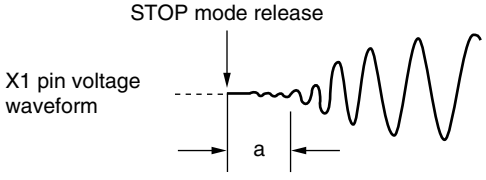
4. Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.

5. The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS. If the STOP mode is entered and then released while the internal high-speed oscillation clock is being used as the CPU clock, set the oscillation stabilization time as follows.

- **Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS**

Note, therefore, that only the status up to the oscillation stabilization time set by OSTS is set to OSTC after STOP mode is released.

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



Remark f_X: X1 clock oscillation frequency

19.2 Standby Function Operation

19.2.1 HALT mode

(1) HALT mode

The HALT mode is set by executing the HALT instruction. HALT mode can be set regardless of whether the
<R> CPU clock before the setting was the high-speed system clock, internal high-speed oscillation clock, 20 MHz internal high-speed oscillation clock, or subsystem clock.

The operating statuses in the HALT mode are shown below.

Table 19-1. Operating Statuses in HALT Mode (1/3)

HALT Mode Setting / Item			When HALT Instruction Is Executed While CPU Is Operating on Main System Clock		
			When CPU Is Operating on Internal High-Speed Oscillation Clock (f_{IH}) or 20 MHz Internal High-Speed Oscillation Clock (f_{IH20})	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}, f_{IH20}	Operation continues (cannot be stopped)	Status before HALT mode was set is retained	
		f_X	Status before HALT mode was set is retained	Operation continues (cannot be stopped)	Cannot operate
		f_{EX}		Cannot operate	Operation continues (cannot be stopped)
	Subsystem clock	f_{XT}	Status before HALT mode was set is retained		
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H) • WDTON = 0: Stops • WDTON = 1 and WDSTBYON = 1: Oscillates • WDTON = 1 and WDSTBYON = 0: Stops		
CPU			Operation stopped		
Flash memory			Operation stopped		
RAM			Status before HALT mode was set is retained at voltage higher than POC detection voltage.		
Port (latch)			Status before HALT mode was set is retained		
Timer array unit (TAU)			Operable		
Real-time counter (RTC)					
Watchdog timer			Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H) • WDTON = 0: Stops • WDTON = 1 and WDSTBYON = 1: Operates • WDTON = 1 and WDSTBYON = 0: Stops		
Clock output/buzzer output			Operable		
A/D converter					
D/A converter					
Operational amplifier					
Voltage reference					
Serial array unit (SAU)					
Serial interface (IICA)					
Multiplier/divider					
DMA controller					
Power-on-clear function					
Low-voltage detection function					
External interrupt					

Remark f_{IH}: Internal high-speed oscillation clock, f_{IH20}: 20 MHz internal high-speed oscillation clock
f_X: X1 oscillation clock, f_{EX}: External main system clock
f_{XT}: XT1 oscillation clock, f_{IL}: Internal low-speed oscillation clock

Table 19-1. Operating Statuses in HALT Mode (2/3)

HALT Mode Setting / Item			When HALT Instruction Is Executed While CPU Is Operating on Subsystem Clock
			When CPU Is Operating on XT1 Clock (f_{XT})
System clock			Clock supply to the CPU is stopped
	Main system clock	f_{IH}, f_{IH20}	Status before HALT mode was set is retained
		f_X	
		f_{EX}	Operates or stops by external clock input
	Subsystem clock	f_{XT}	Operation continues (cannot be stopped)
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H) • WDTON = 0: Stops • WDTON = 1 and WDSTBYON = 1: Oscillates • WDTON = 1 and WDSTBYON = 0: Stops
CPU			Operation stopped
Flash memory			Operation stopped (wait state in low-power consumption mode)
RAM			Status before HALT mode was set is retained at voltage higher than POC detection voltage.
Port (latch)			Status before HALT mode was set is retained
Timer array unit (TAU)			Operable
Real-time counter (RTC)			
Watchdog timer			Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H) • WDTON = 0: Stops • WDTON = 1 and WDSTBYON = 1: Operates • WDTON = 1 and WDSTBYON = 0: Stops
Clock output/buzzer output			Operable
A/D converter			Cannot operate
D/A converter			Operable
Operational amplifier			
Voltage reference			
Serial array unit (SAU)			
Serial interface (IICA)			Cannot operate
Multiplier/divider			Operable
DMA controller			
Power-on-clear function			
Low-voltage detection function			
External interrupt			

Remark f_{IH}: Internal high-speed oscillation clock, f_{IH20}: 20 MHz internal high-speed oscillation clock
f_X: X1 oscillation clock, f_{EX}: External main system clock
f_{XT}: XT1 oscillation clock, f_{IL}: Internal low-speed oscillation clock

Table 19-1. Operating Statuses in HALT Mode (3/3)

HALT Mode Setting / Item			When HALT Instruction Is Executed While CPU Is Operating on Subsystem Clock
			When CPU Is Operating on XT1 Clock (f_{XT}) (Subsystem Clock HALT Mode (RTCLPC = 1))
System clock			Clock supply to the CPU is stopped
	Main system clock	f_{IH}, f_{IH20}	Status before HALT mode was set is retained
		f_X	
		f_{EX}	Operates or stops by external clock input
	Subsystem clock	f_{XT}	Operation continues (cannot be stopped)
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H) • WDTON = 0: Stops • WDTON = 1 and WDSTBYON = 1: Oscillates • WDTON = 1 and WDSTBYON = 0: Stops
CPU			Operation stopped
Flash memory			Operation stopped (wait state in low-power consumption mode)
RAM			Status before HALT mode was set is retained at voltage higher than POC detection voltage.
Port (latch)			Status before HALT mode was set is retained
Timer array unit (TAU)			Cannot operate
Real-time counter (RTC)			Operable
Watchdog timer			Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H) • WDTON = 0: Stops • WDTON = 1 and WDSTBYON = 1: Operates • WDTON = 1 and WDSTBYON = 0: Stops
Clock output/buzzer output			Operable
A/D converter			Cannot operate
D/A converter			
Operational amplifier			
Voltage reference			
Serial array unit (SAU)			
Serial interface (IICA)			
Multiplier/divider			Operation stopped
DMA controller			
Power-on-clear function			Operable
Low-voltage detection function			
External interrupt			

Remarks 1. f_{IH}: Internal high-speed oscillation clock, f_{IH20}: 20 MHz internal high-speed oscillation clock
f_X: X1 oscillation clock, f_{EX}: External main system clock
f_{XT}: XT1 oscillation clock, f_{IL}: Internal low-speed oscillation clock

2. RTCLPC: Bit 7 of the operation speed mode control register (OSMC).

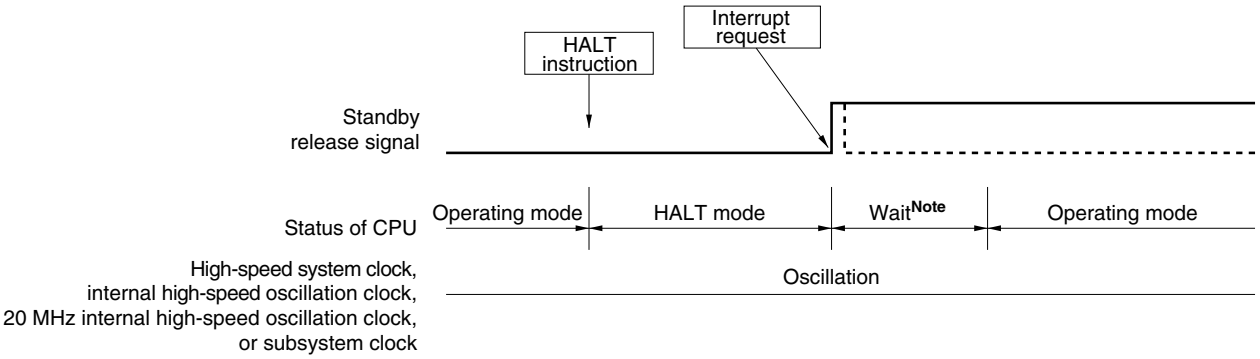
(2) HALT mode release

The HALT mode can be released by the following two sources.

(a) Release by unmasked interrupt request

When an unmasked interrupt request is generated, the HALT mode is released. If interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 19-3. HALT Mode Release by Interrupt Request Generation



Note The wait time is as follows:

- When vectored interrupt servicing is carried out
  - When main system clock is used: 10 to 12 clocks
  - When subsystem clock is used: 8 to 10 clocks
- When vectored interrupt servicing is not carried out
  - When main system clock is used: 5 or 6 clocks
  - When subsystem clock is used: 3 or 4 clocks

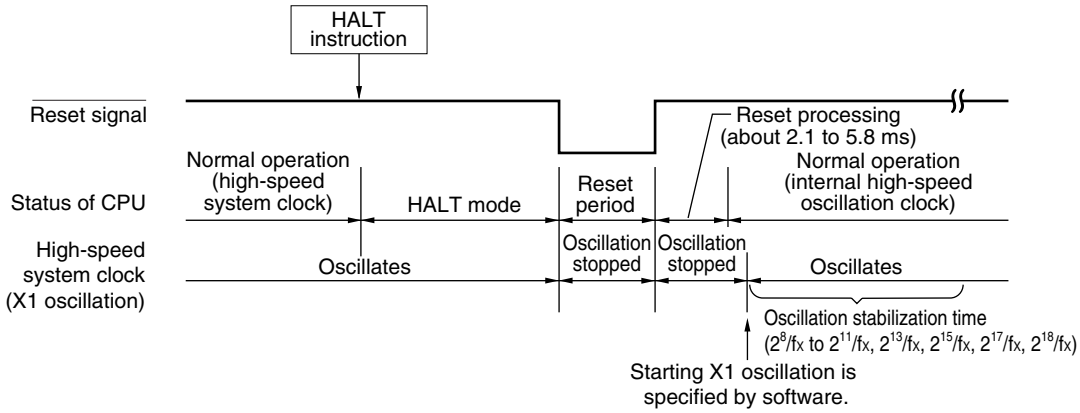
Remark The broken lines indicate the case when the interrupt request which has released the standby mode is acknowledged.

(b) Release by reset signal generation

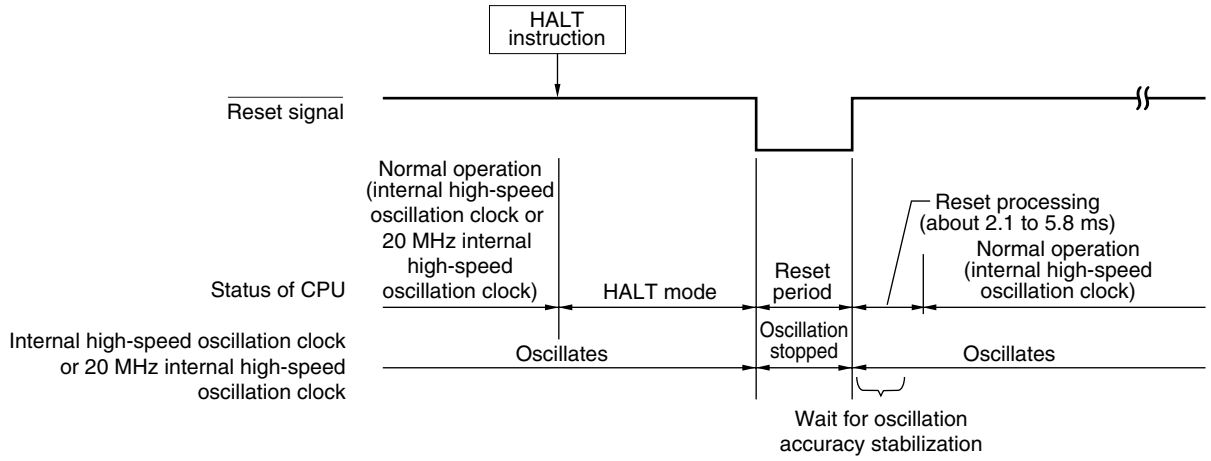
When the reset signal is generated, HALT mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 19-4. HALT Mode Release by Reset (1/2)

(1) When high-speed system clock is used as CPU clock



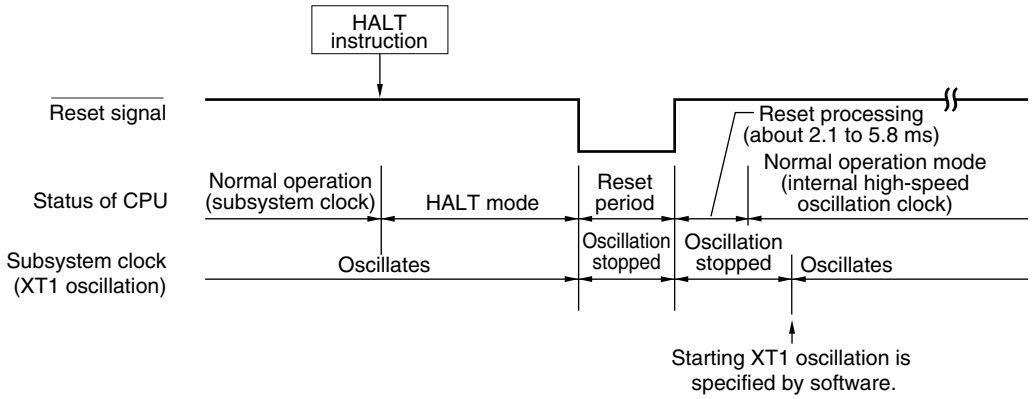
(2) When internal high-speed oscillation clock or 20 MHz internal high-speed oscillation clock is used as CPU clock



Remark fx: X1 clock oscillation frequency

Figure 19-4. HALT Mode Release by Reset (2/2)

(3) When subsystem clock is used as CPU clock



19.2.2 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 main system clock.

Cautions 1. Because the interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately cleared if set. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction and the system returns to the operating mode as soon as the wait time set using the oscillation stabilization time select register (OSTS) has elapsed.

2. The STOP instruction cannot be executed when the CPU operates on the 20 MHz internal high-speed oscillation clock. Be sure to execute the STOP instruction after shifting to internal high-speed oscillation clock operation.

The operating statuses in the STOP mode are shown below.

Table 19-2. Operating Statuses in STOP Mode

STOP Mode Setting / Item			When STOP Instruction Is Executed While CPU Is Operating on Main System Clock		
			When CPU Is Operating on Internal High-Speed Oscillation 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}			
	Subsystem clock	f_{XT}	Status before STOP mode was set is retained		
	f_{IL}		Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H) • WDTON = 0: Stops • WDTON = 1 and WDSTBYON = 1: Oscillates • WDTON = 1 and WDSTBYON = 0: Stops		
CPU			Operation stopped		
Flash memory					
RAM			Status before STOP mode was set is retained at voltage higher than POC detection voltage.		
Port (latch)			Status before STOP mode was set is retained		
Timer array unit (TAU)			Cannot operate		
Real-time counter (RTC)			Operable		
Watchdog timer			Set by bits 0 (WDSTBYON) and 4 (WDTON) of option byte (000C0H) • WDTON = 0: Stops • WDTON = 1 and WDSTBYON = 1: Operates • WDTON = 1 and WDSTBYON = 0: Stops		
Clock output/buzzer output			Operable only when subsystem clock is selected as the count clock		
A/D converter			Operation stopped		
D/A converter			Operable		
Operational amplifier					
Voltage reference					
Serial array unit (SAU)			Cannot operate		
Serial interface (IICA)			Wake-up by address match operable		
Multiplier/divider			Cannot operate		
DMA controller					
Power-on-clear function			Operable		
Low-voltage detection function					
External interrupt					

Remark f_{IH}: Internal high-speed oscillation clock, f_X: X1 oscillation clock
f_{EX}: External main system clock, f_{XT}: XT1 oscillation clock
f_{IL}: Internal low-speed oscillation clock

Cautions 1. To use the peripheral hardware that stops operation in the STOP mode, and the peripheral hardware for which the clock that stops oscillating in the STOP mode after the STOP mode is released, restart the peripheral hardware.

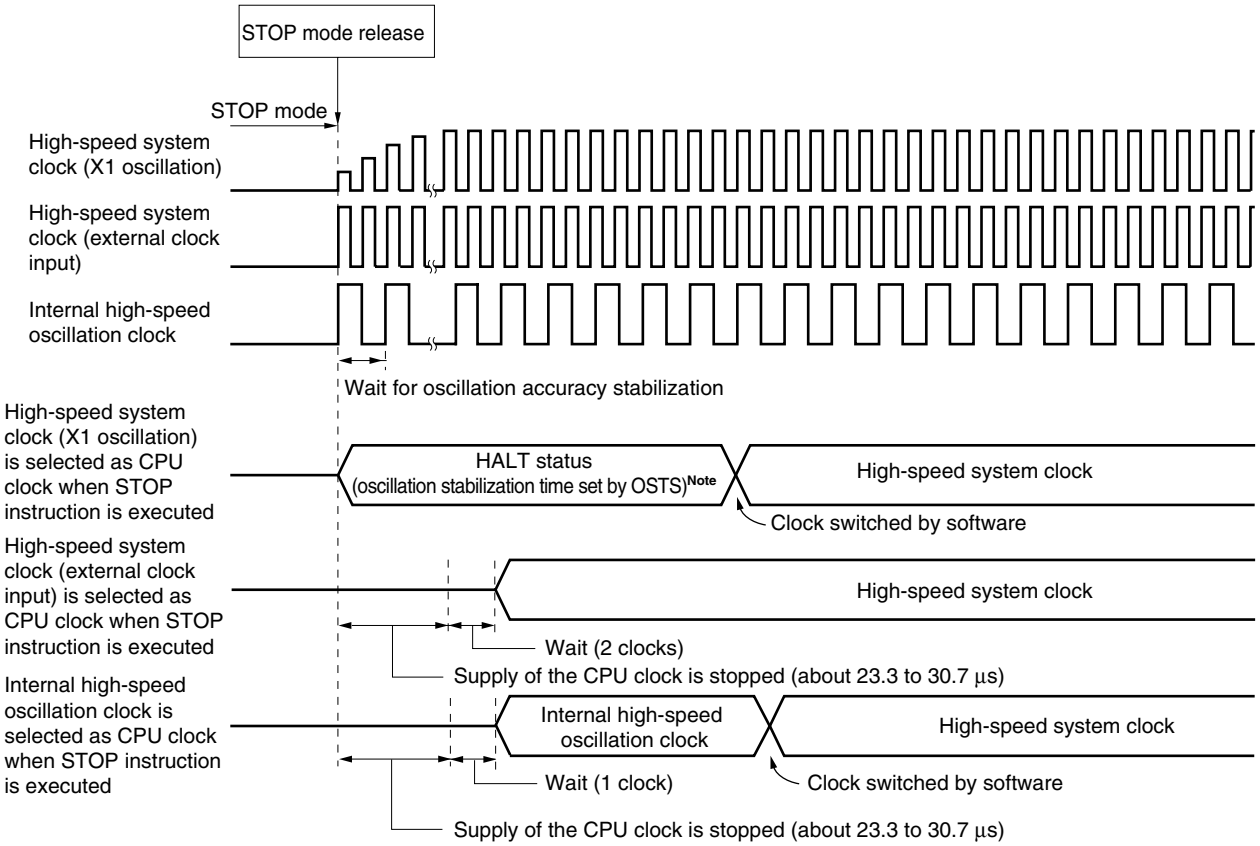
2. To stop the internal low-speed oscillation clock in the STOP mode, use an option byte to stop the watchdog timer operation in the HALT/STOP mode (bit 0 (WDSTBYON) of 000C0H = 0), and then execute the STOP instruction.

3. To shorten oscillation stabilization time after the STOP mode is released when the CPU operates with the high-speed system clock (X1 oscillation), temporarily switch the CPU clock to the internal high-speed oscillation clock before the next execution of the STOP instruction. Before changing the CPU clock from the internal high-speed oscillation clock to the high-speed system clock (X1 oscillation) after the STOP mode is released, check the oscillation stabilization time with the oscillation stabilization time counter status register (OSTC).

4. The STOP instruction cannot be executed when the CPU operates on the 20 MHz internal high-speed oscillation clock. Be sure to execute the STOP instruction after shifting to internal high-speed oscillation clock operation.

(2) STOP mode release

Figure 19-5. Operation Timing When STOP Mode Is Released (When Unmasked Interrupt Request Is Generated)



Note When the oscillation stabilization time set by OSTS is equal to or shorter than 61 μs, the HALT status is retained to a maximum of "61μs + wait time."

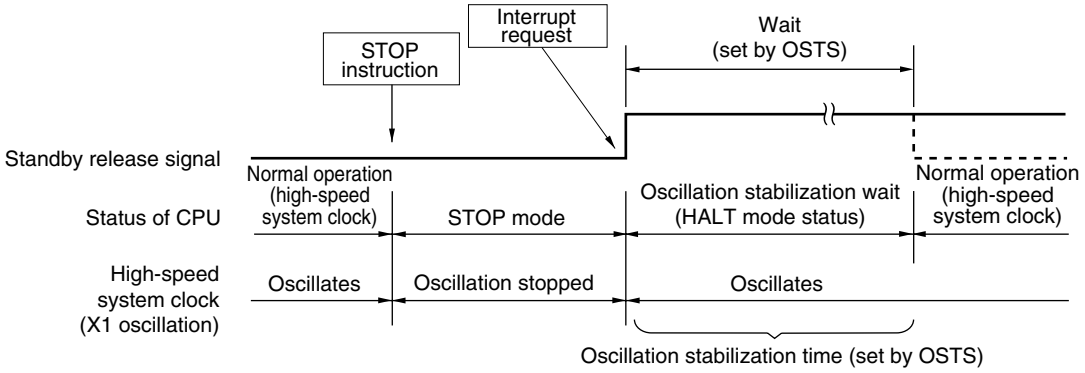
The STOP mode can be released by the following two sources.

(a) Release by unmasked interrupt request

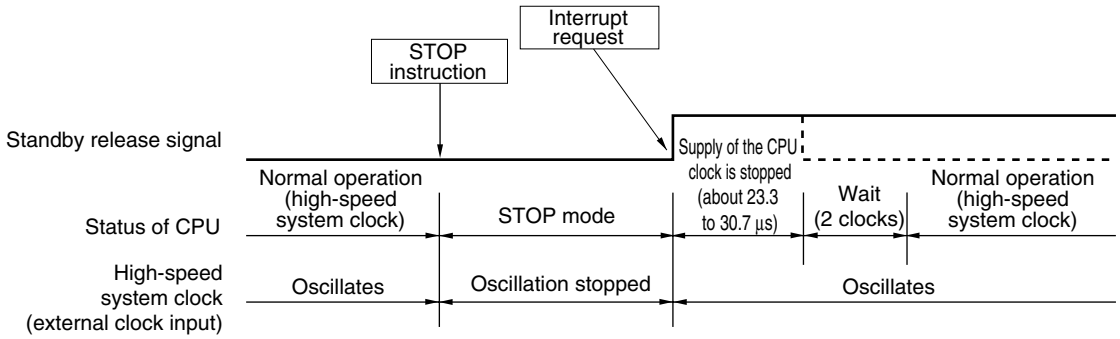
When an unmasked interrupt request is generated, the STOP mode is released. After the oscillation stabilization time has elapsed, if interrupt acknowledgment is enabled, vectored interrupt servicing is carried out. If interrupt acknowledgment is disabled, the next address instruction is executed.

Figure 19-6. STOP Mode Release by Interrupt Request Generation (1/2)

(1) When high-speed system clock (X1 oscillation) is used as CPU clock



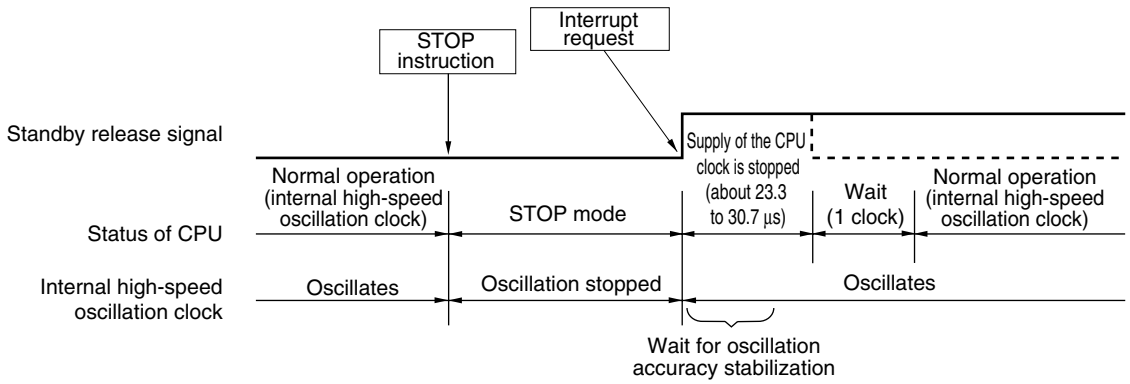
(2) When high-speed system clock (external clock input) is used as CPU clock



Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

Figure 19-6. STOP Mode Release by Interrupt Request Generation (2/2)

(3) When internal high-speed oscillation clock is used as CPU clock



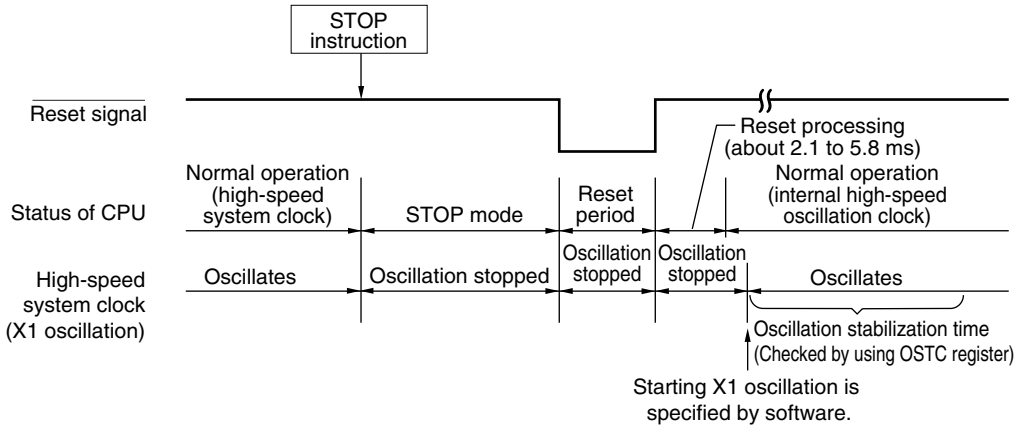
Remark The broken lines indicate the case when the interrupt request that has released the standby mode is acknowledged.

(b) Release by reset signal generation

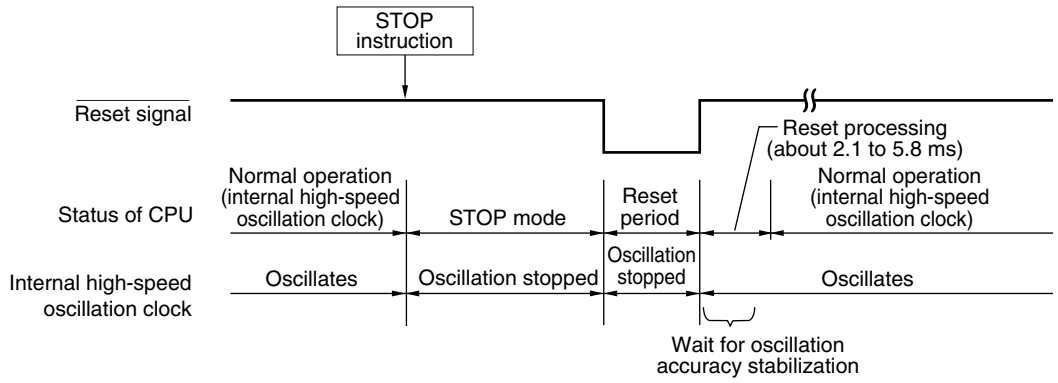
When the reset signal is generated, STOP mode is released, and then, as in the case with a normal reset operation, the program is executed after branching to the reset vector address.

Figure 19-7. STOP Mode Release by Reset

(1) When high-speed system clock is used as CPU clock



(2) When internal high-speed oscillation clock is used as CPU clock



Remark fx: X1 clock oscillation frequency

CHAPTER 20 RESET FUNCTION

The following five 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-clear (POC) circuit
(4) Internal reset by comparison of supply voltage of the low-voltage detector (LVI) or input voltage (EXLVI) from external input pin, and detection voltage
(5) Internal reset by execution of illegal instruction[Note]

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 POC and LVI circuit voltage detection or execution of illegal instruction[Note], and each item of hardware is set to the status shown in Tables 20-1 and 20-2. Each pin is high impedance during reset signal generation or during the oscillation stabilization time just after a reset release, except for P130, which is low-level output.

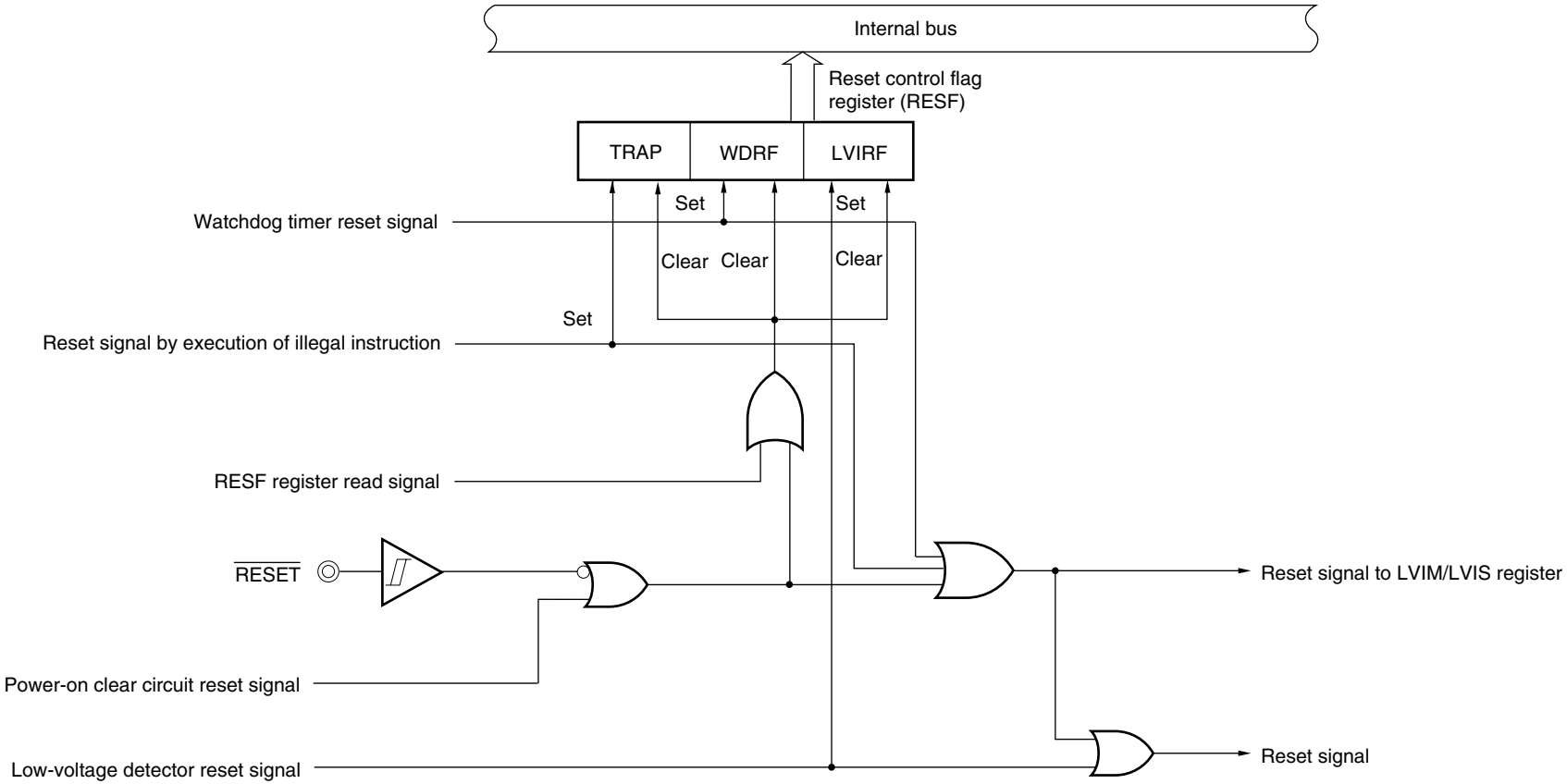
When a low level is input to the $\overline{\text{RESET}}$ pin, the device is reset. It is released from the reset status when a high level is input to the $\overline{\text{RESET}}$ pin and program execution is started with the internal high-speed oscillation clock after reset processing. A reset by the watchdog timer is automatically released, and program execution starts using the internal high-speed oscillation clock (see **Figures 20-2** to **20-4**) after reset processing. Reset by POC and LVI circuit power supply detection is automatically released when $V_{DD} \geq V_{POR}$ or $V_{DD} \geq V_{LVI}$ after the reset, and program execution starts using the internal high-speed oscillation clock (see **CHAPTER 21 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 22 LOW-VOLTAGE DETECTOR**) after reset processing.

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.

Cautions 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, a low level of at least 10 μs must be continued during the period in which the supply voltage is within the operating range ($V_{DD} \geq 1.8$ V).)
2. During reset input, the X1 clock, XT1 clock, internal high-speed oscillation clock, and internal low-speed oscillation clock stop oscillating. External main system clock input becomes invalid.
3. When the STOP mode is released by a reset, the RAM contents in the STOP mode are held during reset input.
4. When reset is effected, port pin P130 is set to low-level output and other port pins become high-impedance, because each SFR and 2nd SFR are initialized.

Remark V_{POR}: POC power supply rise detection voltage
V_{LVI}: LVI detection voltage

Figure 20-1. Block Diagram of Reset Function

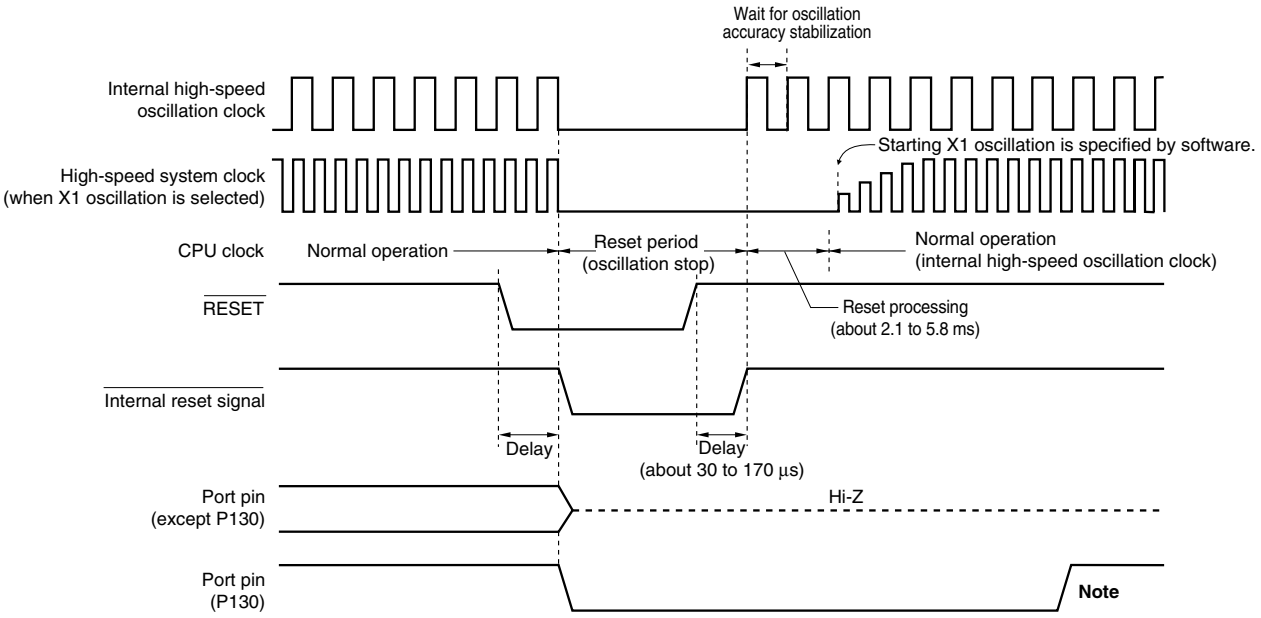


Caution An LVI circuit internal reset does not reset the LVI circuit.

Remarks 1. LVIM: Low-voltage detection register

2. LVIS: Low-voltage detection level select register

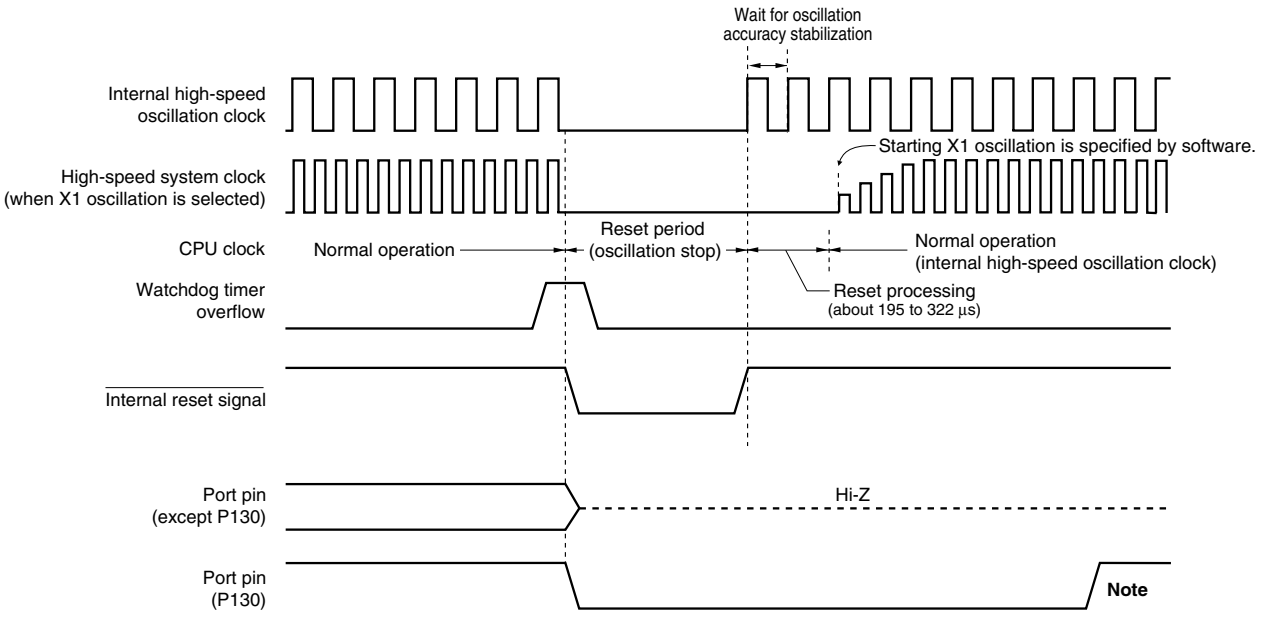
Figure 20-2. Timing of Reset by $\overline{\text{RESET}}$ Input



Note Set P130 to high-level output by software.

Remark When reset is effected, P130 outputs a low level. If P130 is set to output a high level before reset is effected, the output signal of P130 can be dummy-output as the CPU reset signal.

Figure 20-3. Timing of Reset Due to Watchdog Timer Overflow

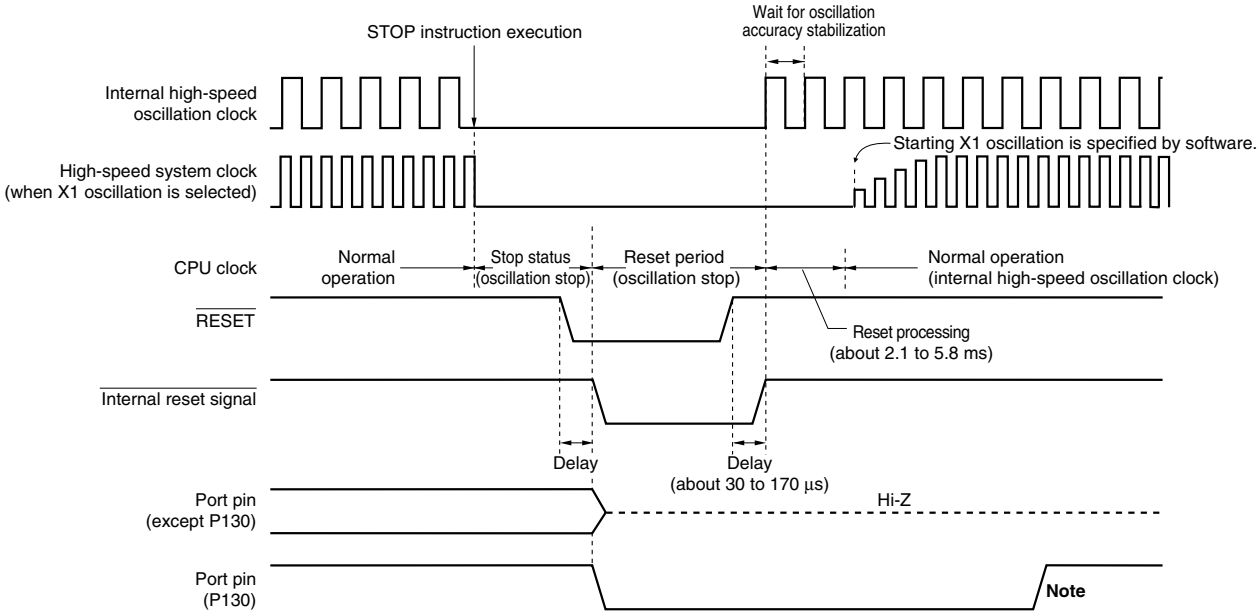


Note Set P130 to high-level output by software.

Caution A watchdog timer internal reset resets the watchdog timer.

Remark When reset is effected, P130 outputs a low level. If P130 is set to output a high level before reset is effected, the output signal of P130 can be dummy-output as the CPU reset signal.

Figure 20-4. Timing of Reset in STOP Mode by $\overline{\text{RESET}}$ Input



Note Set P130 to high-level output by software.

Remarks 1. When reset is effected, P130 outputs a low level. If P130 is set to output a high level before reset is effected, the output signal of P130 can be dummy-output as the CPU reset signal.

2. For the reset timing of the power-on-clear circuit and low-voltage detector, see **CHAPTER 21 POWER-ON-CLEAR CIRCUIT** and **CHAPTER 22 LOW-VOLTAGE DETECTOR**.

Table 20-1. 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 (X1 and X2 pins are input port mode)
		f_{EX}	Clock input invalid (pin is input port mode)
	Subsystem clock	f_{XT}	Operation stopped (XT1 and XT2 pins are input port mode)
	f_{IL}		Operation stopped
CPU			
Flash memory			
RAM			Operation stopped (The value, however, is retained when the voltage is at least the power-on-clear detection voltage.)
Port (latch)			Set P130 to low-level output. The port pins except for P130 become high impedance.
Timer array unit (TAU)			Operation stopped
Real-time counter (RTC)			
Watchdog timer			
Clock output/buzzer output			
A/D converter			
D/A converter			
Operational amplifier			
Voltage reference			
Serial array unit (SAU)			
Serial interface (IICA)			
Multiplier/divider			Operation stopped
DMA controller			
Power-on-clear function			Detection operation possible
Low-voltage detection function			Operation stopped (however, operation continues at LVI reset)
External interrupt			Operation stopped
BCD correction circuit (BCD)			

Remark f_{IH}: Internal high-speed oscillation clock, f_X: X1 oscillation clock
f_{EX}: External main system clock, f_{XT}: XT1 oscillation clock
f_{IL}: Internal low-speed oscillation clock

Table 20-2. Hardware Statuses After Reset Acknowledgment (1/4)

Hardware		After Reset Acknowledgment[Note 1]
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[Note 2]
	General-purpose registers	Undefined[Note 2]
Port registers (P0 to P6, P8, P10 to P15) (output latches)		00H
Port mode registers (PM0 to PM6, PM8, PM10 to PM12, PM14, PM15)		FFH
Port input mode register 1 (PIM1)		00H
Port output mode registers 1, 8 (POM1, POM8)		00H
Pull-up resistor option registers (PU0, PU1, PU3 to PU5, PU8, PU10, PU12, PU14)		00H
Clock operation mode control register (CMC)		00H
Clock operation status control register (CSC)		C0H
Processor mode control register (PMC)		00H
System clock control register (CKC)		09H
20 MHz internal high-speed oscillation control register (DSCCTL)		00H
Oscillation stabilization time counter status register (OSTC)		00H
Oscillation stabilization time select register (OSTS)		07H
Noise filter enable registers 0, 1 (NFEN0, NFEN1)		00H
Peripheral enable register 0 (PER0)		00H
Operation speed mode control register (OSMC)		00H
Input switch control register (ISC)		00H
Timer array units 0, 1 (TAU0, TAU1)	Timer data registers 00, 01, 02, 03, 04, 05, 06, 07, 10, 11, 12, 13 (TDR00, TDR01, TDR02, TDR03, TDR04, TDR05, TDR06, TDR07, TDR10, TDR11, TDR12, TDR13)	0000H
	Timer mode registers 00, 01, 02, 03, 04, 05, 06, 07, 10, 11, 12, 13 (TMR00, TMR01, TMR02, TMR03, TMR04, TMR05, TMR06, TMR07, TMR10, TMR11, TMR12, TMR13)	0000H
	Timer status registers 00, 01, 02, 03, 04, 05, 06, 07 (TSR00, TSR01, TSR02, TSR03, TSR04, TSR05, TSR06, TSR07)	0000H
	Timer input select registers 0, 1 (TIS0, TIS1)	00H
	Timer counter registers 00, 01, 02, 03, 04, 05, 06, 07, 10, 11, 12, 13 (TCR00, TCR01, TCR02, TCR03, TCR04, TCR05, TCR06, TCR07, TCR10, TCR11, TCR12, TCR13)	FFFFH
	Timer channel enable status registers 0, 1 (TE0, TE1)	0000H
	Timer channel start registers 0, 1 (TS0, TS1)	0000H

Notes 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. When a reset is executed in the standby mode, the pre-reset status is held even after reset.

Table 20-2. Hardware Statuses After Reset Acknowledgment (2/4)

Hardware		Status After Reset Acknowledgment[Note 1]
Timer array unit 0, 1 (TAU0, TAU1)	Timer channel stop registers 0, 1 (TT0, TT1)	0000H
	Timer clock select registers 0, 1 (TPS0, TPS1)	0000H
	Timer output register 0 (TO0)	0000H
	Timer output enable register 0 (TOE0)	0000H
	Timer output level register 0 (TOL0)	0000H
	Timer output mode register 0 (TOM0)	0000H
Real-time counter	Sub-count register (RSUBC)	0000H
	Second count register (SEC)	00H
	Minute count register (MIN)	00H
	Hour count register (HOUR)	12H
	Week count register (WEEK)	00H
	Day count register (DAY)	01H
	Month count register (MONTH)	01H
	Year count register (YEAR)	00H
	Watch error correction register (SUBCUD)	00H
	Alarm minute register (ALARMWM)	00H
	Alarm hour register (ALARMWH)	12H
	Alarm week register (ALARMWW)	00H
	Real-time counter control register 0 (RTCC0)	00H
	Real-time counter control register 1 (RTCC1)	00H
	Real-time counter control register 2 (RTCC2)	00H
Clock output/buzzer output controller	Clock output select registers 0, 1 (CKS0, CKS1)	00H
Watchdog timer	Enable register (WDTE)	1AH/9AH[Note 2]
A/D converter	10-bit A/D conversion result register (ADCR)	0000H
	8-bit A/D conversion result register (ADCRH)	00H
	A/D converter mode register (ADM)	00H
	A/D converter mode register 1 (ADM1)	00H
	Analog reference voltage control register (ADVRC)	00H
	Analog input channel specification register (ADS)	00H
	A/D port configuration register (ADPC)	10H
D/A converter	D/A conversion value setting registers W0, W1 (DACSW0, DACSW1)	0000H
	8-bit D/A conversion value setting registers 0, 1 (DACS0, DACS1)	00H
	D/A converter mode register (DAM)	00H
Operational amplifier	Operational amplifier control register (OAC)	00H
Voltage reference	Analog reference voltage control register (ADVRC)	00H

Notes 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. The reset value of WDTE is determined by the option byte setting.

Table 20-2. Hardware Statuses After Reset Acknowledgment (3/4)

Hardware		Status After Reset Acknowledgment[Note]
Serial array units 0, 1 (SAU0, SAU1)	Serial data registers 00, 01, 02, 03, 10, 11, 12, 13 (SDR00, SDR01, SDR02, SDR03, SDR10, SDR11, SDR12, SDR13)	0000H
	Serial status registers 00, 01, 02, 03, 10, 11, 12, 13 (SSR00, SSR01, SSR02, SSR03, SSR10, SSR11, SSR12, SSR13)	0000H
	Serial flag clear trigger registers 00, 01, 02, 03, 10, 11, 13 (SIR00, SIR01, SIR02, SIR03, SIR10, SIR11, SIR13)	0000H
	Serial mode registers 00, 01, 02, 03, 10, 11, 12, 13 (SMR00, SMR01, SMR02, SMR03, SMR10, SMR11, SMR12, SMR13)	0020H
	Serial communication operation setting registers 00, 01, 02, 03, 10, 11, 12, 13 (SCR00, SCR01, SCR02, SCR03, SCR10, SCR11, SCR12, SCR13)	0087H
	Serial channel enable status registers 0, 1 (SE0, SE1)	0000H
	Serial channel start registers 0, 1 (SS0, SS1)	0000H
	Serial channel stop registers 0, 1 (ST0, ST1)	0000H
	Serial clock select registers 0, 1 (SPS0, SPS1)	0000H
	Serial output registers 0, 1 (SO0, SO1)	0F0FH
	Serial output enable registers 0, 1 (SOE0, SOE1)	0000H
	Serial output level registers 0, 1 (SOL0, SOL1)	0000H
Serial interface IICA	Shift register (IICA)	00H
	Control register 0 (IICCTL0)	00H
	Control register 1 (IICCTL1)	00H
	Slave address register (SVA)	00H
	IICA low-level width setting register 0 (IICWL)	FFH
	IICA high-level width setting register 0 (IICWH)	FFH
	Status register (IICS)	00H
	Flag register (IICF)	00H
Multiplier/divider	Multiplication/division data register A (MDAL, MDAH)	0000H
	Multiplication/division data register B (MDBL, MDBH)	0000H
	Multiplication/division data register C (MDCL, MDCH)	0000H
	Multiplication/division control register (MDUC)	00H

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.

Table 20-2. Hardware Statuses After Reset Acknowledgment (4/4)

Hardware		Status After Reset Acknowledgment[Note 1]
Reset function	Reset control flag register (RESF)	Undefined[Note 2]
Low-voltage detector	Low-voltage detection register (LVIM)	00H[Note 3]
	Low-voltage detection level select register (LVIS)	0EH[Note 2]
DMA controller	SFR address registers 0, 1 (DSA0, DSA1)	00H
	RAM address registers 0L, 0H, 1L, 1H (DRA0L, DRA0H, DRA1L, DRA1H)	00H
	Byte count registers 0L, 0H, 1L, 1H (DBC0L, DBC0H, DBC1L, DBC1H)	00H
	Mode control registers 0, 1 (DMC0, DMC1)	00H
	Operation control registers 0, 1 (DRC0, DRC1)	00H
Interrupt	Request flag registers 0L, 0H, 1L, 1H, 2L, 2H (IF0L, IF0H, IF1L, IF1H, IF2L, IF2H)	00H
	Mask flag registers 0L, 0H, 1L, 1H, 2L, 2H (MK0L, MK0H, MK1L, MK1H, MK2L, MK2H)	FFH
	Priority specification flag registers 00L, 00H, 01L, 01H, 02L, 02H, 10L, 10H, 11L, 11H, 12L, 12H (PR00L, PR00H, PR01L, PR01H, PR10L, PR10H, PR11L, PR11H, PR02L, PR02H, PR12L, PR12H)	FFH
	External interrupt rising edge enable registers 0, 1 (EGP0, EGP1)	00H
	External interrupt falling edge enable registers 0, 1 (EGN0, EGN1)	00H
Regulator	Regulator mode control register (RMC)	00H
BCD correction circuit (BCD)	BCD correction result register (BCDADJ)	Undefined

Notes 1. During reset signal generation or oscillation stabilization time wait, only the PC contents among the hardware statuses become undefined. All other hardware statuses remain unchanged after reset.

2. These values vary depending on the reset source.

Register \ Reset Source		$\overline{\text{RESET}}$ Input	Reset by POC	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by LVI
RESF	TRAP bit	Cleared (0)	Cleared (0)	Set (1)	Held	Held
	WDRF bit			Held	Set (1)	Held
	LVIRF bit			Held	Held	Set (1)
LVIS		Cleared (0EH)	Cleared (0EH)	Cleared (0EH)	Cleared (0EH)	Held

3. This value varies depending on the reset source and the option byte.

20.1 Register for Confirming Reset Source

Many internal reset generation sources exist in the 78K0R/Kx3-A microcontrollers. The reset control flag register (RESF) is used to store which source has generated the reset request.

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

$\overline{\text{RESET}}$ input, reset by power-on-clear (POC) circuit, and reading RESF clear TRAP, WDRF, and LVIRF.

Figure 20-5. Format of Reset Control Flag Register (RESF)

Address: FFFA8H After reset: Undefined R

Symbol	7	6	5	4	3	2	1	0
RESF	TRAP Note 1	Undefined	Undefined	WDRF Note 1	Undefined	Undefined	Undefined	LVIRF Note 1

TRAP	Internal reset request by execution of illegal instruction Note 2
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

WDRF	Internal reset request by watchdog timer (WDT)
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

LVIRF	Internal reset request by low-voltage detector (LVI)
0	Internal reset request is not generated, or RESF is cleared.
1	Internal reset request is generated.

Notes 1. The value after reset varies depending on the reset source.

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.

Cautions 1. Do not read data by a 1-bit memory manipulation instruction.

2. Do not make a judgment based on only the read value of the RESF register 8-bit data, because bits other than TRAP, WDRF, and LVIRF become undefined.

3. When the LVI default start function (bit 0 (LVIOFF) of 000C1H = 0) is used, LVIRF flag may become 1 from the beginning depending on the power-on waveform.

The status of RESF when a reset request is generated is shown in Table 20-3.

Table 20-3. RESF Status When Reset Request Is Generated

Reset Source / Flag	$\overline{\text{RESET}}$ Input	Reset by POC	Reset by Execution of Illegal Instruction	Reset by WDT	Reset by LVI
TRAP	Cleared (0)	Cleared (0)	Set (1)	Held	Held
WDRF			Held	Set (1)	Held
LVIRF			Held	Held	Set (1)

CHAPTER 21 POWER-ON-CLEAR CIRCUIT

21.1 Functions of Power-on-Clear Circuit

The power-on-clear circuit has the following functions.

- Generates internal reset signal at power on.
  The reset signal is released when the supply voltage (V_{DD}) exceeds 1.61 V ±0.09 V[Note].

  Caution If the low-voltage detector (LVI) is set to ON by an option byte by default, the reset signal is not released until the supply voltage (V_{DD}) exceeds 2.07 V ±0.2 V[Note].

- Compares supply voltage (V_{DD}) and detection voltage (V_{PDR} = 1.59 V ±0.09 V[Note]), generates internal reset signal when $V_{DD} < V_{PDR}$.

  Note These are preliminary values and subject to change.

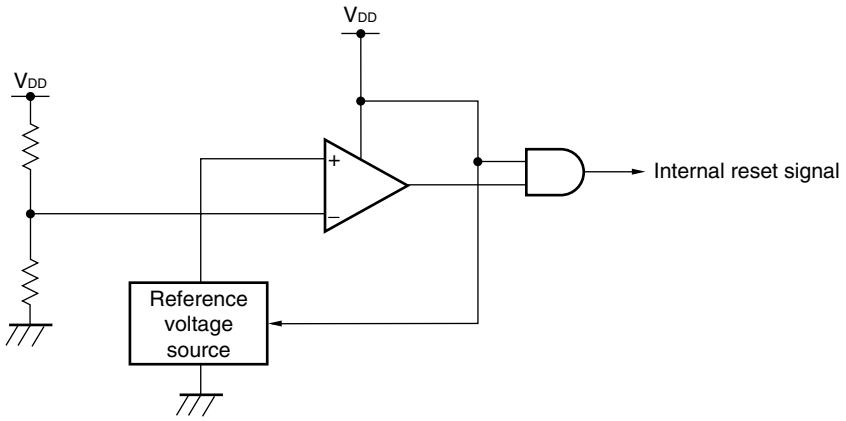
  Caution If an internal reset signal is generated in the POC circuit, the reset control flag register (RESF) is cleared to 00H.

  Remark The 78K0R/Kx3-A microcontrollers incorporate 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), low-voltage-detector (LVI), or illegal instruction execution. RESF is not cleared to 00H and the flag is set to 1 when an internal reset signal is generated by WDT or LVI.
  For details of RESF, see **CHAPTER 20 RESET FUNCTION**.

21.2 Configuration of Power-on-Clear Circuit

The block diagram of the power-on-clear circuit is shown in Figure 21-1.

Figure 21-1. Block Diagram of Power-on-Clear Circuit



21.3 Operation of Power-on-Clear Circuit

- An internal reset signal is generated on power application. When the supply voltage (V_{DD}) exceeds the detection voltage (V_{PDR} = 1.61 V ±0.09 V[Note]), the reset status is released.

  Caution If the low-voltage detector (LVI) is set to ON by an option byte by default, the reset signal is not released until the supply voltage (V_{DD}) exceeds 2.07 V ±0.2 V[Note].

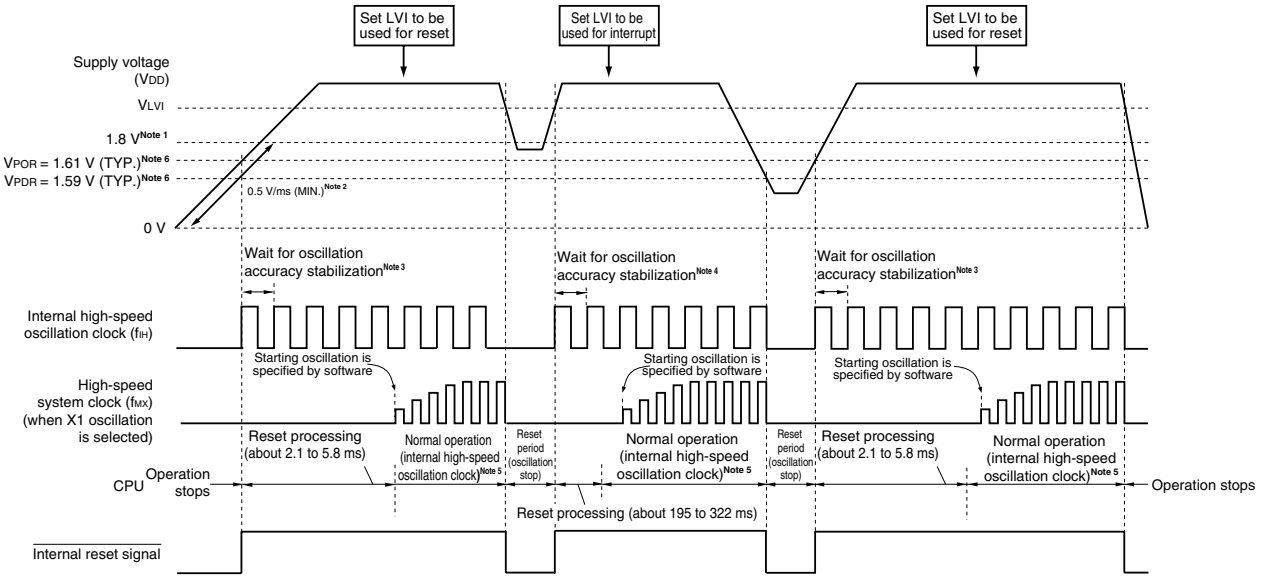
- The supply voltage (V_{DD}) and detection voltage (V_{PDR} = 1.59 V ±0.09 V[Note]) are compared. When $V_{DD} < V_{PDR}$, the internal reset signal is generated.

  Note These are preliminary values and subject to change.

The timing of generation of the internal reset signal by the power-on-clear circuit and low-voltage detector is shown below.

Figure 21-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (1/2)

(1) When LVI is OFF upon power application (option byte: LVIOFF = 1)



Notes 1. The operation guaranteed range is 1.8 V ≤ V_{DD} ≤ 5.5 V. To make the state at lower than 1.8 V reset state when the supply voltage falls, use the reset function of the low-voltage detector, or input the low level to the $\overline{\text{RESET}}$ pin.

2. If the rate at which the voltage rises to 1.8 V after power application is slower than 0.5 V/ms (MIN.), input a low level to the $\overline{\text{RESET}}$ pin before the voltage reaches to 1.8 V, or set LVI to ON by default by using an option byte (option byte: LVIOFF = 0).

3. The reset processing time, such as when waiting for internal voltage stabilization, includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.

4. The internal reset processing time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.

5. The internal high-speed oscillation clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the OSTC register to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

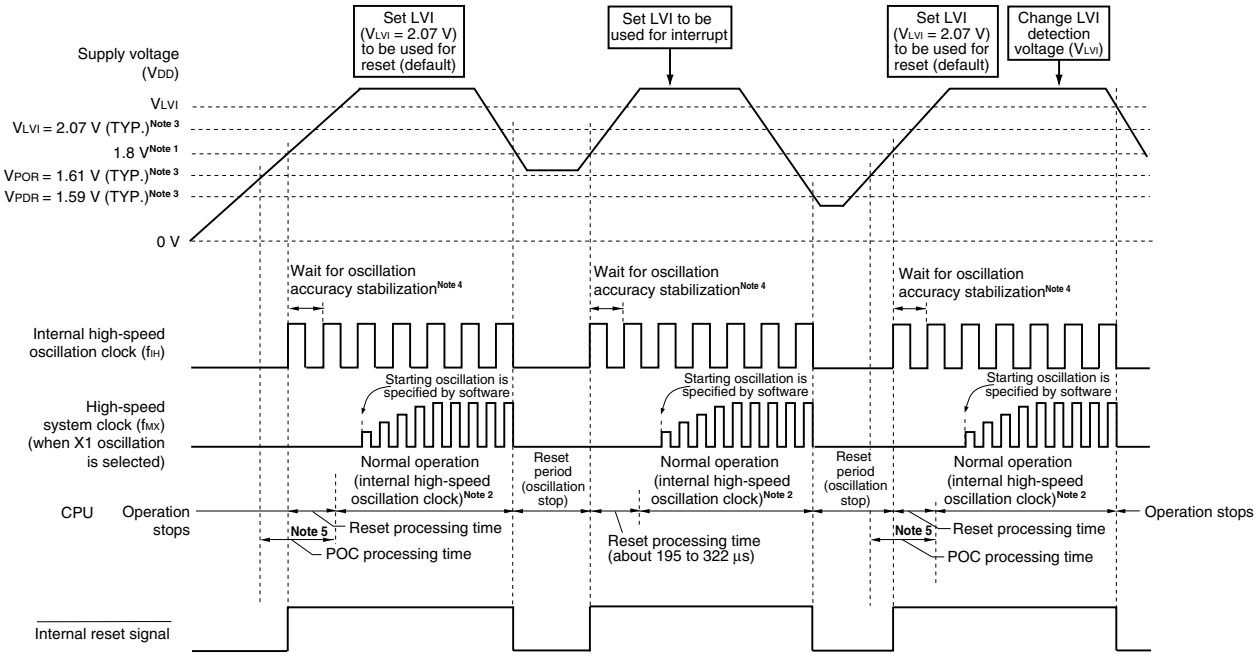
6. This is a preliminary value and subject to change.

Caution Set the low-voltage detector by software after the reset status is released (see CHAPTER 22 LOW-VOLTAGE DETECTOR).

Remark V_{LVI}: LVI detection voltage
V_{POR}: POC power supply rise detection voltage
V_{PDR}: POC power supply fall detection voltage

Figure 21-2. Timing of Generation of Internal Reset Signal by Power-on-Clear Circuit and Low-Voltage Detector (2/2)

(2) When LVI is ON upon power application (option byte: LVIOFF = 0)



Notes 1. The operation guaranteed range is 1.8 V ≤ V_{DD} ≤ 5.5 V. To make the state at lower than 1.8 V reset state when the supply voltage falls, use the reset function of the low-voltage detector, or input the low level to the $\overline{\text{RESET}}$ pin.

2. The internal high-speed oscillation clock and a high-speed system clock or subsystem clock can be selected as the CPU clock. To use the X1 clock, use the OSTC register to confirm the lapse of the oscillation stabilization time. To use the XT1 clock, use the timer function for confirmation of the lapse of the stabilization time.

3. These are preliminary values and subject to change.

4. The internal reset processing time includes the oscillation accuracy stabilization time of the internal high-speed oscillation clock.

5. The following times are required between reaching the POC detection voltage (1.59 V (TYP.)) and starting normal operation.

- When the time to reach 2.07 V (TYP.) from 1.59 V (TYP.) is less than 5.8 ms:
  A POC processing time of about 2.1 to 6.2 ms is required between reaching 1.59 V (TYP.) and starting normal operation.
- When the time to reach 2.07 V (TYP.) from 1.59 V (TYP.) is greater than 5.8 ms:
  A reset processing time of about 195 to 322 µs is required between reaching 2.07 V (TYP.) and starting normal operation.

Caution Set the low-voltage detector by software after the reset status is released (see CHAPTER 22 LOW-VOLTAGE DETECTOR).

Remark V_{LVI}: LVI detection voltage
V_{POR}: POC power supply rise detection voltage
V_{PDR}: POC power supply fall detection voltage

21.4 Cautions for Power-on-Clear Circuit

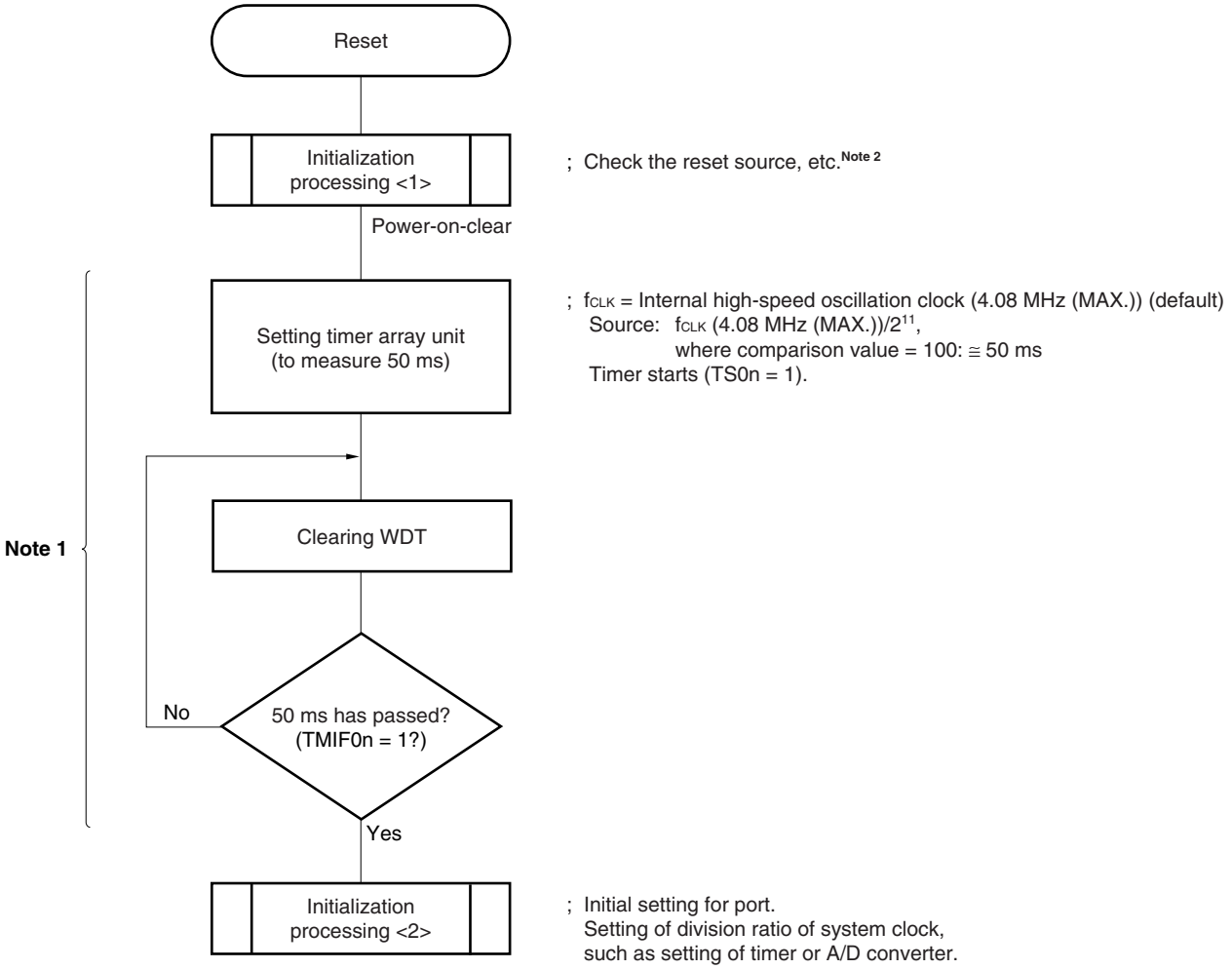
In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the POC detection voltage (V_{POR}, V_{PDR}), the system may be repeatedly reset and released from the reset status. In this case, the time from release of reset to the start of the operation of the microcontroller 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 21-3. Example of Software Processing After Reset Release (1/2)

• If supply voltage fluctuation is 50 ms or less in vicinity of POC detection voltage



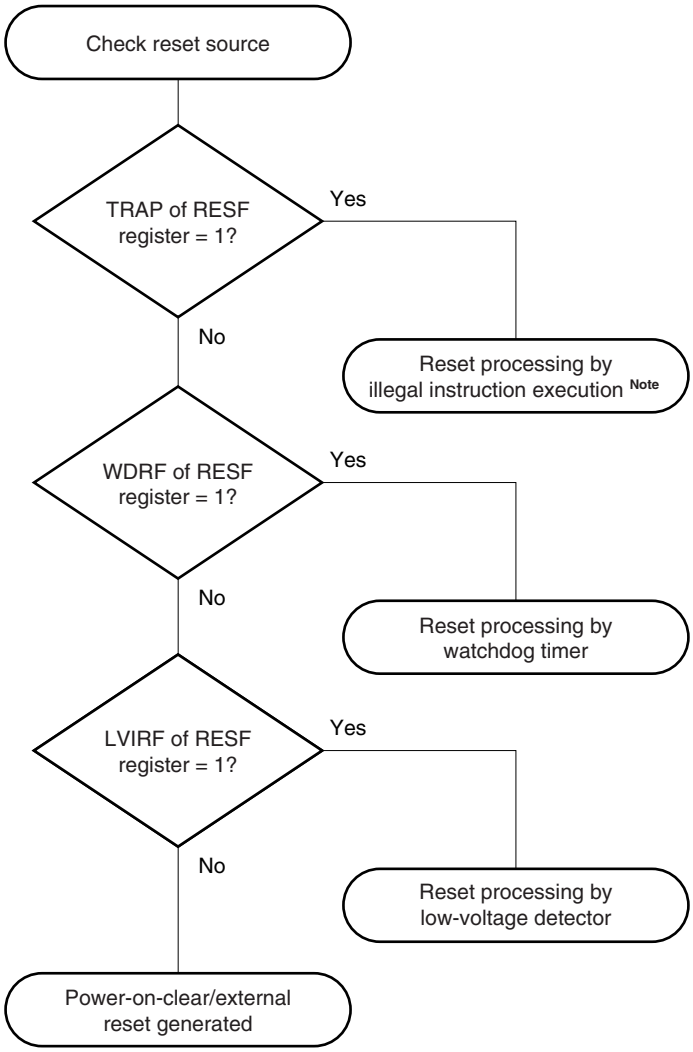
Notes 1. If reset is generated again during this period, initialization processing <2> is not started.

2. A flowchart is shown on the next page.

Remark n = 0 to 7

Figure 21-3. Example of Software Processing After Reset Release (2/2)

• Checking reset source



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.

CHAPTER 22 LOW-VOLTAGE DETECTOR

22.1 Functions of Low-Voltage Detector

The low-voltage detector has the following functions.

- The LVI circuit compares the supply voltage (V_{DD}) with the detection voltage (V_{LVI}) or the input voltage from an external input pin (EXLVI) with the detection voltage (V_{EXLVI} = 1.21 V ±0.1 V[Note]), and generates an internal reset or internal interrupt signal.
- The low-voltage detector (LVI) can be set to ON by an option byte by default. If it is set to ON to raise the power supply from the POC detection voltage (V_{POR} = 1.61 V (TYP.)) or lower, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.07 V ±0.2 V[Note]). After that, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.07 V ±0.1 V[Note]).
- The supply voltage (V_{DD}) or the input voltage from the external input pin (EXLVI) can be selected to be detected by software.
- A reset or an interrupt can be selected to be generated after detection by software.
- Detection levels (V_{LVI},16 levels) of supply voltage can be changed by software.
- Operable in STOP mode.

Note This is a preliminary value and subject to change.

The reset and interrupt signals are generated as follows depending on selection by software.

Selection of Level Detection of Supply Voltage (V_{DD}) (LVISEL = 0)		Selection Level Detection of Input Voltage from External Input Pin (EXLVI) (LVISEL = 1)	
Selects reset (LVIMD = 1).	Selects interrupt (LVIMD = 0).	Selects reset (LVIMD = 1).	Selects interrupt (LVIMD = 0).
Generates an internal reset signal when $V_{DD} < V_{LVI}$ and releases the reset signal when $V_{DD} \geq V_{LVI}$.	Generates an internal interrupt signal when V_{DD} drops lower than V_{LVI} ($V_{DD} < V_{LVI}$) or when V_{DD} becomes V_{LVI} or higher ($V_{DD} \geq V_{LVI}$).	Generates an internal reset signal when EXLVI < V_{EXLVI} and releases the reset signal when EXLVI ≥ V_{EXLVI}.	Generates an internal interrupt signal when EXLVI drops lower than V_{EXLVI} (EXLVI < V_{EXLVI}) or when EXLVI becomes V_{EXLVI} or higher (EXLVI ≥ V_{EXLVI}).

Remark LVISEL: Bit 2 of low-voltage detection register (LVIM)
LVIMD: Bit 1 of LVIM

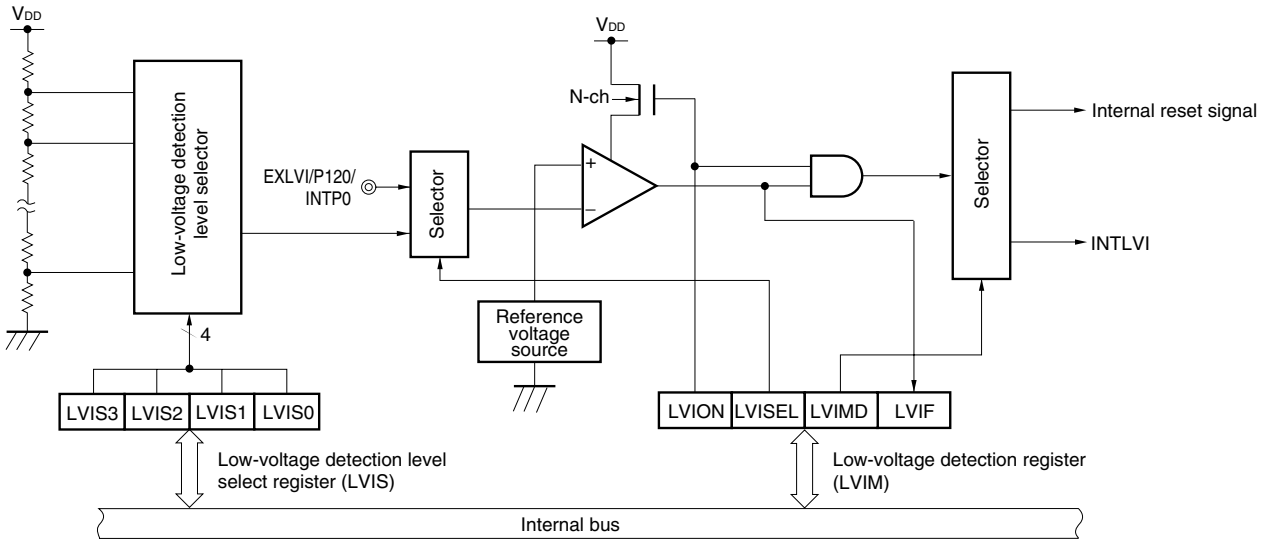
While the low-voltage detector is operating, whether the supply voltage or the input voltage from an external input pin is more than or less than the detection level can be checked by reading the low-voltage detection flag (LVIF: bit 0 of LVIM).

When the low-voltage detector is used to reset, bit 0 (LVIRF) of the reset control flag register (RESF) is set to 1 if reset occurs. For details of RESF, see **CHAPTER 20 RESET FUNCTION**.

22.2 Configuration of Low-Voltage Detector

The block diagram of the low-voltage detector is shown in Figure 22-1.

Figure 22-1. Block Diagram of Low-Voltage Detector



22.3 Registers Controlling Low-Voltage Detector

The low-voltage detector is controlled by the following registers.

- Low-voltage detection register (LVIM)
- Low-voltage detection level select register (LVIS)
- Port mode register 12 (PM12)

(1) Low-voltage detection register (LVIM)

This register sets low-voltage detection and the operation mode.

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

Reset signal generation clears this register to 00H.

Figure 22-2. Format of Low-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	LVION	0	0	0	0	LVISEL	LVIMD	LVIF

LVION[Notes 3, 4]	Enables low-voltage detection operation
0	Disables operation
1	Enables operation

LVISEL[Note 3]	Voltage detection selection
0	Detects level of supply voltage (V_{DD})
1	Detects level of input voltage from external input pin (EXLVI)

LVIMD	Low-voltage detection operation mode (interrupt/reset) selection
0	• LVISEL = 0: Generates an internal interrupt signal when the supply voltage (V_{DD}) drops lower than the detection voltage (V_{LVI}) ($V_{DD} < V_{LVI}$) or when V_{DD} becomes V_{LVI} or higher ($V_{DD} \geq V_{LVI}$). • LVISEL = 1: Generates an interrupt signal when the input voltage from an external input pin (EXLVI) drops lower than the detection voltage (V_{EXLVI}) (EXLVI < V_{EXLVI}) or when EXLVI becomes V_{EXLVI} or higher (EXLVI ≥ V_{EXLVI}).
1	• LVISEL = 0: Generates an internal reset signal when the supply voltage (V_{DD}) < detection voltage (V_{LVI}) and releases the reset signal when $V_{DD} \geq V_{LVI}$. • LVISEL = 1: Generates an internal reset signal when the input voltage from an external input pin (EXLVI) < detection voltage (V_{EXLVI}) and releases the reset signal when EXLVI ≥ V_{EXLVI}.

LVIF	Low-voltage detection flag
0	• LVISEL = 0: Supply voltage (V_{DD}) ≥ detection voltage (V_{LVI}), or when LVI operation is disabled • LVISEL = 1: Input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI}), or when LVI operation is disabled
1	• LVISEL = 0: Supply voltage (V_{DD}) < detection voltage (V_{LVI}) • LVISEL = 1: Input voltage from external input pin (EXLVI) < detection voltage (V_{EXLVI})

Notes 1. The reset value changes depending on the reset source and the setting of the option byte. This register is not cleared (00H) by LVI reset.
It is set to “82H” when a reset signal other than LVI is applied if option byte LVIOFF = 0, and to “00H” if option byte LVIOFF = 1.

2. Bit 0 is read-only.

3. LVION, LVIMD, and LVISEL are cleared to 0 in the case of a reset other than an LVI reset. These are not cleared to 0 in the case of an LVI reset.

Note 4. When LVION is set to 1, operation of the comparator in the LVI circuit is started. Use software to wait for the following periods of time, between when LVION is set to 1 and when the voltage is confirmed with LVIF.

- Operation stabilization time (10 μs (max.))
- Minimum pulse width (200 μs (min.))

The LVIF value for these periods may be set/cleared regardless of the voltage level, and can therefore not be used. Also, the LVIIF interrupt request flag may be set to 1 in these periods.

Cautions 1. To stop LVI, be sure to clear (0) LVION by using a 1-bit memory manipulation instruction.

2. Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.

3. When LVI is used in interrupt mode (LVIMD = 0) and LVISEL is set to 0, an interrupt request signal (INTLVI) that disables LVI operation (clears LVION) when the supply voltage (V_{DD}) is less than or equal to the detection voltage (V_{LVI}) (if LVISEL = 1, input voltage of external input pin (EXLVI) is less than or equal to the detection voltage (V_{EXLVI})) is generated and LVIIF may be set to 1.

(2) Low-voltage detection level select register (LVIS)

This register selects the low-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 0EH.

Figure 22-3. Format of Low-Voltage Detection Level Select Register (LVIS)

Address: FFFAAH After reset: 0EH[Note 1] R/W

Symbol	7	6	5	4	3	2	1	0
LVIS	0	0	0	0	LVIS3	LVIS2	LVIS1	LVIS0

LVIS3	LVIS2	LVIS1	LVIS0	Detection level
0	0	0	0	V_{LVI0} (4.22 ±0.1 V)[Note 2]
0	0	0	1	V_{LVI1} (4.07 ±0.1 V)[Note 2]
0	0	1	0	V_{LVI2} (3.92 ±0.1 V)[Note 2]
0	0	1	1	V_{LVI3} (3.76 ±0.1 V)[Note 2]
0	1	0	0	V_{LVI4} (3.61 ±0.1 V)[Note 2]
0	1	0	1	V_{LVI5} (3.45 ±0.1 V)[Note 2]
0	1	1	0	V_{LVI6} (3.30 ±0.1 V)[Note 2]
0	1	1	1	V_{LVI7} (3.15 ±0.1 V)[Note 2]
1	0	0	0	V_{LVI8} (2.99 ±0.1 V)[Note 2]
1	0	0	1	V_{LVI9} (2.84 ±0.1 V)[Note 2]
1	0	1	0	V_{LVI10} (2.68 ±0.1 V)[Note 2]
1	0	1	1	V_{LVI11} (2.53 ±0.1 V)[Note 2]
1	1	0	0	V_{LVI12} (2.38 ±0.1 V)[Note 2]
1	1	0	1	V_{LVI13} (2.22 ±0.1 V)[Note 2]
1	1	1	0	V_{LVI14} (2.07 ±0.1 V)[Note 2]
1	1	1	1	V_{LVI15} (1.91 ±0.1 V)[Note 2]

Notes 1. The reset value changes depending on the reset source.
If the LVIS register is reset by LVI, it is not reset but holds the current value. The value of this register is reset to “0EH” if a reset other than by LVI is effected.

2. These are preliminary values and subject to change.

Caution 1. Be sure to clear bits 4 to 7 to “0”.

Cautions 2. Change the LVIS value with either of the following methods.

- **When changing the value after stopping LVI**
  - **<1> Stop LVI (LVION = 0).**
  - **<2> Change the LVIS register.**
  - **<3> Set to the mode used as an interrupt (LVIMD = 0).**
  - **<4> Mask LVI interrupts (LVIMK = 1).**
  - **<5> Enable LVI operation (LVION = 1).**
  - **<6> Before cancelling the LVI interrupt mask (LVIMK = 0), clear it with software because an LVIIF flag may be set when LVI operation is enabled.**
- **When changing the value after setting to the mode used as an interrupt (LVIMD = 0)**
  - **<1> Mask LVI interrupts (LVIMK = 1).**
  - **<2> Set to the mode used as an interrupt (LVIMD = 0).**
  - **<3> Change the LVIS register.**
  - **<4> Before cancelling the LVI interrupt mask (LVIMK = 0), clear it with software because an LVIIF flag may be set when the LVIS register is changed.**

3. When an input voltage from the external input pin (EXLVI) is detected, the detection voltage (V_{EXLVI}) is fixed. Therefore, setting of LVIS is not necessary.

(3) Port mode register 12 (PM12)

When using the P120/EXLVI/INTP0 pin for external low-voltage detection potential input, set PM120 to 1. At this time, the output latch of P120 may be 0 or 1.

PM12 can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation sets this register to FFH.

Figure 22-4. Format of Port Mode Register 12 (PM12)

Address: FFF2CH After reset: FFH R/W

Symbol	7	6	5	4	3	2	1	0
PM12	1	1	1	1	1	1	1	PM120

PM120	P120 pin I/O mode selection
0	Output mode (output buffer on)
1	Input mode (output buffer off)

22.4 Operation of Low-Voltage Detector

The low-voltage detector can be used in the following two modes.

(1) Used as reset (LVIMD = 1)

- If LVISEL = 0, compares the supply voltage (V_{DD}) and detection voltage (V_{LVI}), generates an internal reset signal when $V_{DD} < V_{LVI}$, and releases internal reset when $V_{DD} \geq V_{LVI}$.
- If LVISEL = 1, compares the input voltage from external input pin (EXLVI) and detection voltage (V_{EXLVI}), generates an internal reset signal when EXLVI < V_{EXLVI}, and releases internal reset when EXLVI ≥ V_{EXLVI}.

Remark The low-voltage detector (LVI) can be set to ON by an option byte by default. If it is set to ON to raise the power supply from the POC detection voltage (V_{POR} = 1.61 V (TYP.)) or lower, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.07 V ±0.2 V[Note]). After that, the internal reset signal is generated when the supply voltage (V_{DD}) < detection voltage (V_{LVI} = 2.07 V ±0.1 V[Note]).

(2) Used as interrupt (LVIMD = 0)

- If LVISEL = 0, compares the supply voltage (V_{DD}) and detection voltage (V_{LVI}). When V_{DD} drops lower than V_{LVI} ($V_{DD} < V_{LVI}$) or when V_{DD} becomes V_{LVI} or higher ($V_{DD} \geq V_{LVI}$), generates an interrupt signal (INTLVI).
- If LVISEL = 1, compares the input voltage from external input pin (EXLVI) and detection voltage (V_{EXLVI} = 1.21 V ±0.1 V[Note]). When EXLVI drops lower than V_{EXLVI} (EXLVI < V_{EXLVI}) or when EXLVI becomes V_{EXLVI} or higher (EXLVI ≥ V_{EXLVI}), generates an interrupt signal (INTLVI).

Note This is a preliminary value and subject to change.

While the low-voltage detector is operating, whether the supply voltage or the input voltage from an external input pin is more than or less than the detection level can be checked by reading the low-voltage detection flag (LVIF: bit 0 of LVIM).

Remark LVIMD: Bit 1 of low-voltage detection register (LVIM)
LVISEL: Bit 2 of LVIM

22.4.1 When used as reset

(1) When detecting level of supply voltage (V_{DD})

(a) When LVI default start function stopped is set (LVIOFF = 1)

- When starting operation

<1> Mask the LVI interrupt (LVIMK = 1).

<2> Clear bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 0 (detects level of supply voltage (V_{DD})) (default value).

<3> Set the detection voltage using bits 3 to 0 (LVIS3 to LVIS0) of the low-voltage detection level selection register (LVIS).

<4> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).

<5> Use software to wait for the following periods of time (Total 210 μs).

- Operation stabilization time (10 μs (max.))
- Minimum pulse width (200 μs (min.))

<6> Wait until it is checked that (supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})) by bit 0 (LVIF) of LVIM.

<7> Set bit 1 (LVIMD) of LVIM to 1 (generates reset when the level is detected).

Figure 22-5 shows the timing of the internal reset signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <7> above.

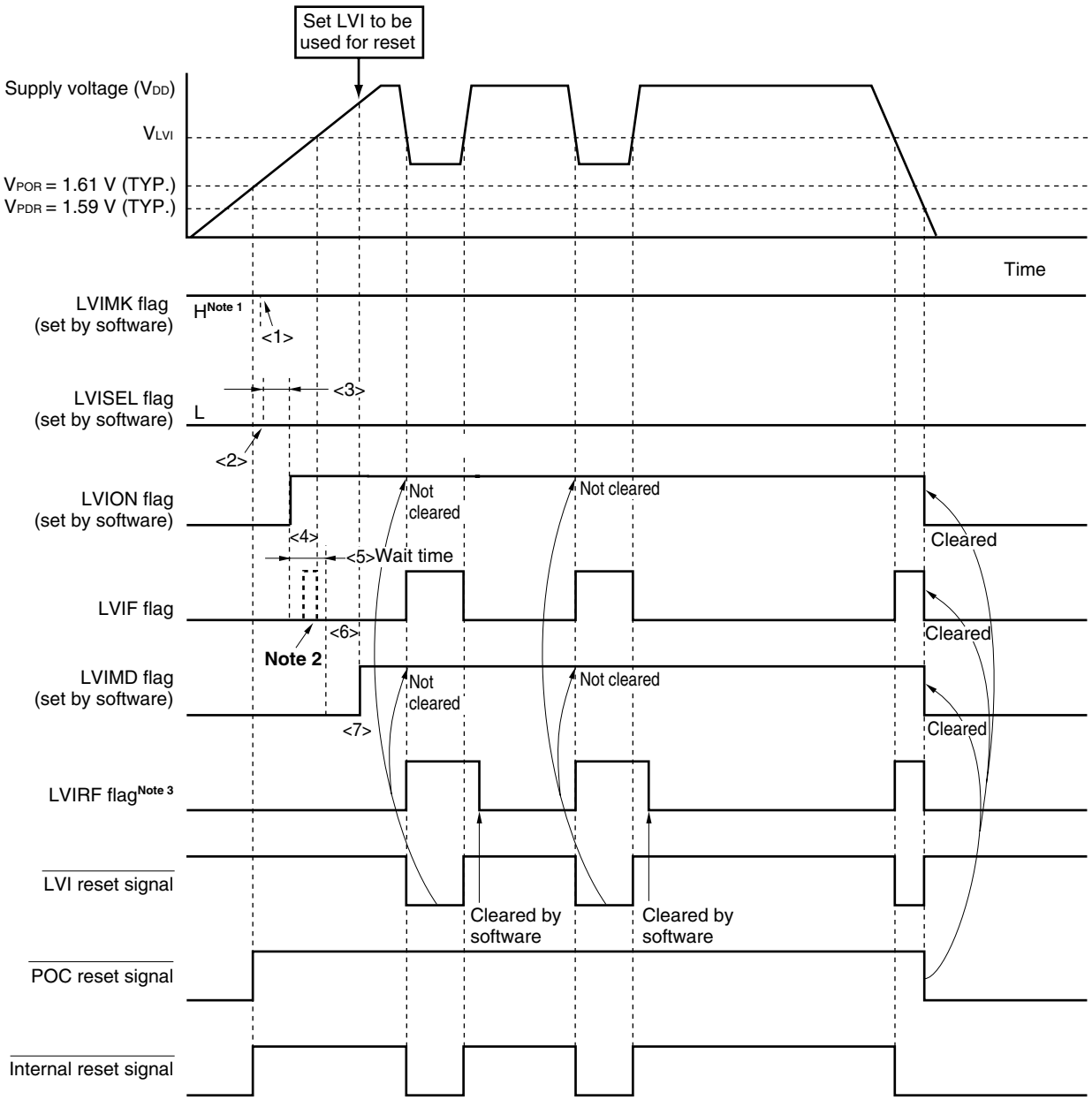
Cautions 1. Be sure to execute <1>. When LVIMK = 0, an interrupt may occur immediately after the processing in <4>.

2. If supply voltage (V_{DD}) ≥ detection voltage (V_{LVI}) when LVIMD is set to 1, an internal reset signal is not generated.

- When stopping operation

Be sure to clear (0) LVIMD and then LVION by using a 1-bit memory manipulation instruction.

Figure 22-5. Timing of Low-Voltage Detector Internal Reset Signal Generation (Bit: LVISEL = 0, Option Byte: LVIOFF = 1)



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

2. The LVIIF flag of the interrupt request flag registers and the LVIF flag may be set (1).

3. LVIRF is bit 0 of the reset control flag register (RESF). For details of RESF, see **CHAPTER 20 RESET FUNCTION**.

Remarks 1. <1> to <7> in Figure 22-5 above correspond to <1> to <7> in the description of "When starting operation" in **22.4.1 (1) (a) When LVI default start function stopped is set (LVIOFF = 1).**

2. V_{POR}: POC power supply rise detection voltage
V_{PDR}: POC power supply fall detection voltage

(b) When LVI default start function enabled is set (LVIOFF = 0)

- When starting operation

  Start in the following initial setting state.

  - Set bit 7 (LVION) of LVIM to 1 (enables LVI operation)
  - Clear bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 0 (detects level of supply voltage (V_{DD}))
  - Set the low-voltage detection level selection register (LVIS) to 0EH (default value: V_{LVI} = 2.07 V ±0.1 V).
  - Set bit 1 (LVIMD) of LVIM to 1 (generates reset when the level is detected)
  - Set bit 0 (LVIF) of LVIM to 0 (“Supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})”)

  Figure 22-6 shows the timing of the internal reset signal generated by the low-voltage detector.

- When stopping operation

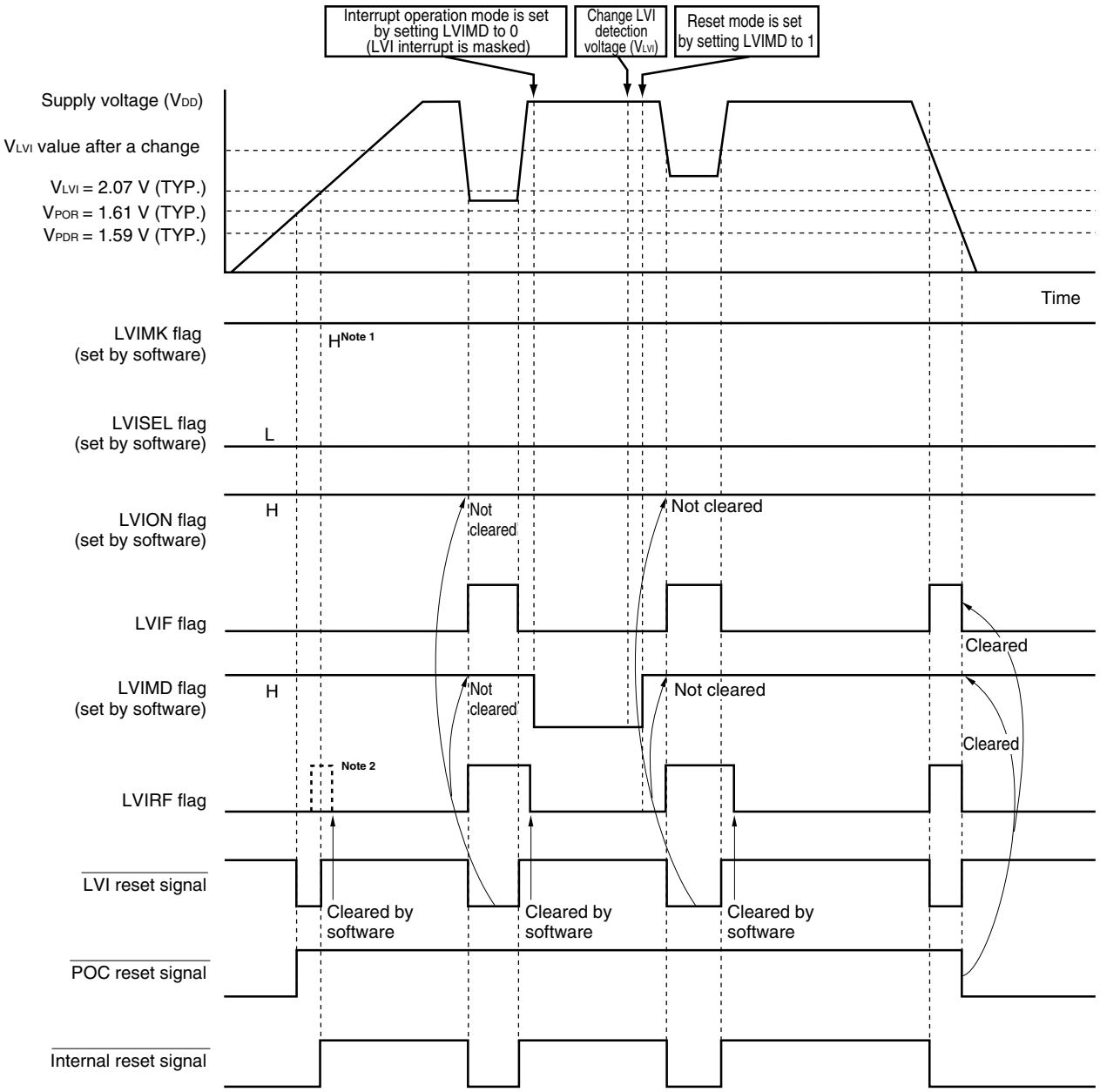
Be sure to clear (0) LVIMD and then LVION by using a 1-bit memory manipulation instruction.

Caution Even when the LVI default start function is used, if it is set to LVI operation prohibition by the software, it operates as follows:

- **Does not perform low-voltage detection during LVION = 0.**
- **If a reset is generated while LVION = 0, LVION will be re-set to 1 when the CPU starts after reset release. There is a period when low-voltage detection cannot be performed normally, however, when a reset occurs due to WDT and illegal instruction execution.**

This is due to the fact that while the pulse width detected by LVI must be 200 μs max., LVION = 1 is set upon reset occurrence, and the CPU starts operating without waiting for the LVI stabilization time.

Figure 22-6. Timing of Low-Voltage Detector Internal Reset Signal Generation (Bit: LVISEL = 0, Option Byte: LVIOFF = 0)



Notes 1. The LVIMK flag is set to “1” by reset signal generation.

2. LVIRF is bit 0 of the reset control flag register (RESF).
When the LVI default start function (bit 0 (LVIOFF) of 000C1H = 0) is used, the LVIRF flag may become 1 from the beginning due to the power-on waveform.
For details of RESF, see **CHAPTER 20 RESET FUNCTION**.

Remark V_{POR}: POC power supply rise detection voltage
V_{PDR}: POC power supply fall detection voltage

(2) When detecting level of input voltage from external input pin (EXLVI)

- When starting operation

<1> Mask the LVI interrupt (LVIMK = 1).

<2> Set bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 1 (detects level of input voltage from external input pin (EXLVI)).

<3> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).

<4> Use software to wait for the following periods of time (Total 210 μs).

- Operation stabilization time (10 μs (MAX.))
- Minimum pulse width (200 μs (MIN.))

<5> Wait until it is checked that (input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI} = 1.21 V (TYP.))) by bit 0 (LVIF) of LVIM.

<6> Set bit 1 (LVIMD) of LVIM to 1 (generates reset signal when the level is detected).

Figure 22-7 shows the timing of the internal reset signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <6> above.

Cautions 1. Be sure to execute <1>. When LVIMK = 0, an interrupt may occur immediately after the processing in <3>.

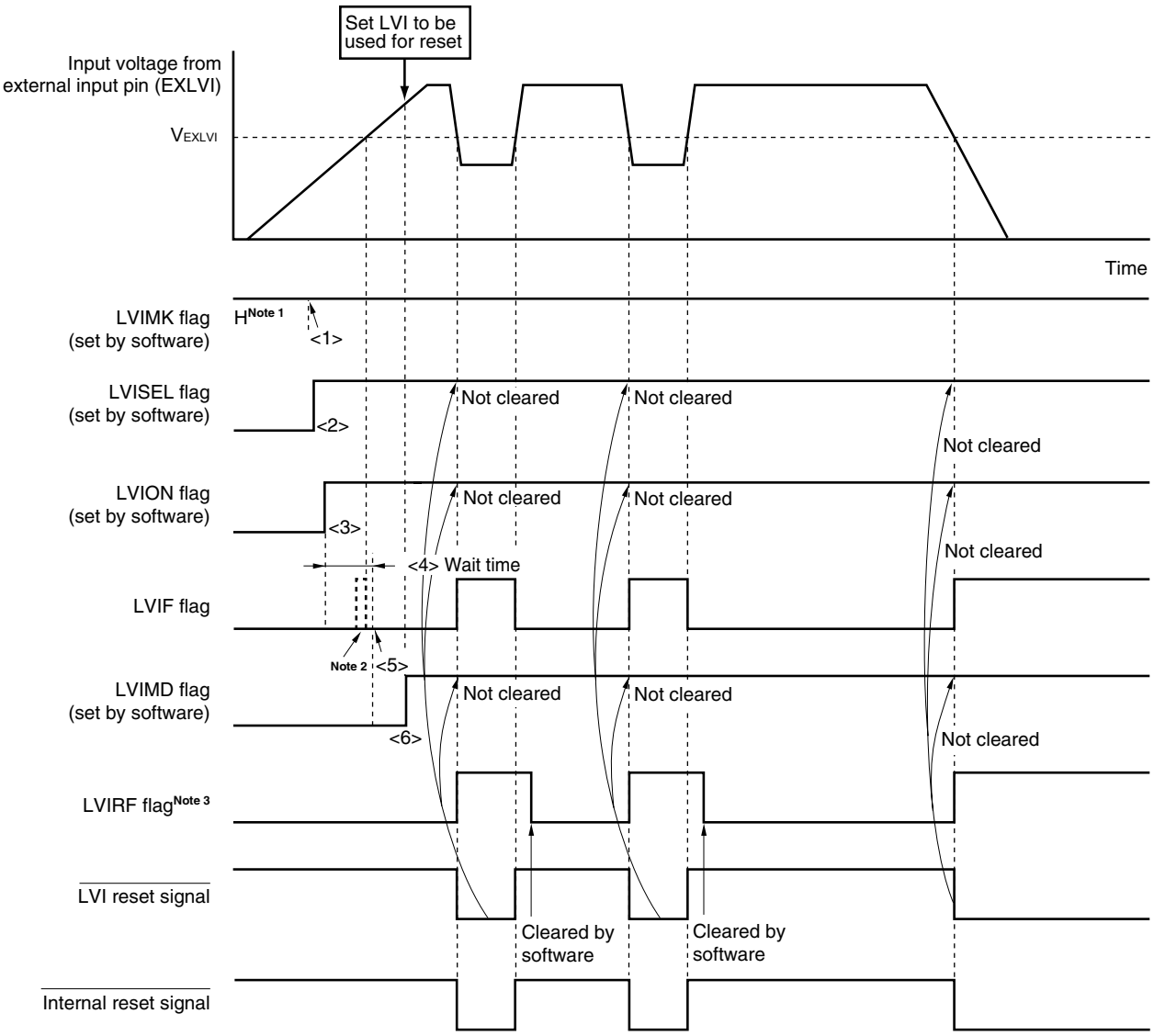
2. If input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI} = 1.21 V (TYP.)) when LVIMD is set to 1, an internal reset signal is not generated.

3. Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.

- When stopping operation

Be sure to clear (0) LVIMD and then LVION by using a 1-bit memory manipulation instruction.

Figure 22-7. Timing of Low-Voltage Detector Internal Reset Signal Generation (Bit: LVISEL = 1)



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

2. The LVIIF flag of the interrupt request flag registers and the LVIF flag may be set (1).

3. LVIRF is bit 0 of the reset control flag register (RESF). For details of RESF, see **CHAPTER 20 RESET FUNCTION**.

Remark <1> to <6> in Figure 22-7 above correspond to <1> to <6> in the description of **"**When starting operation" in **22.4.1 (2) When detecting level of input voltage from external input pin (EXLVI)**.

22.4.2 When used as interrupt

(1) When detecting level of supply voltage (V_{DD})

(a) When LVI default start function stopped is set (LVIOFF = 1)

- When starting operation

<1> Mask the LVI interrupt (LVIMK = 1).

<2> Clear bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 0 (detects level of supply voltage (V_{DD})) (default value).
Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (default value).

<3> Set the detection voltage using bits 3 to 0 (LVIS3 to LVIS0) of the low-voltage detection level selection register (LVIS).

<4> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).

<5> Use software to wait for the following periods of time (Total 210 μs).

- Operation stabilization time (10 μs (MAX.))
- Minimum pulse width (200 μs (MIN.))

<6> Confirm that "supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})" when detecting the falling edge of V_{DD}, or "supply voltage (V_{DD}) < detection voltage (V_{LVI})" when detecting the rising edge of V_{DD}, at bit 0 (LVIF) of LVIM.

<7> Clear the interrupt request flag of LVI (LVIIF) to 0.

<8> Release the interrupt mask flag of LVI (LVIMK).

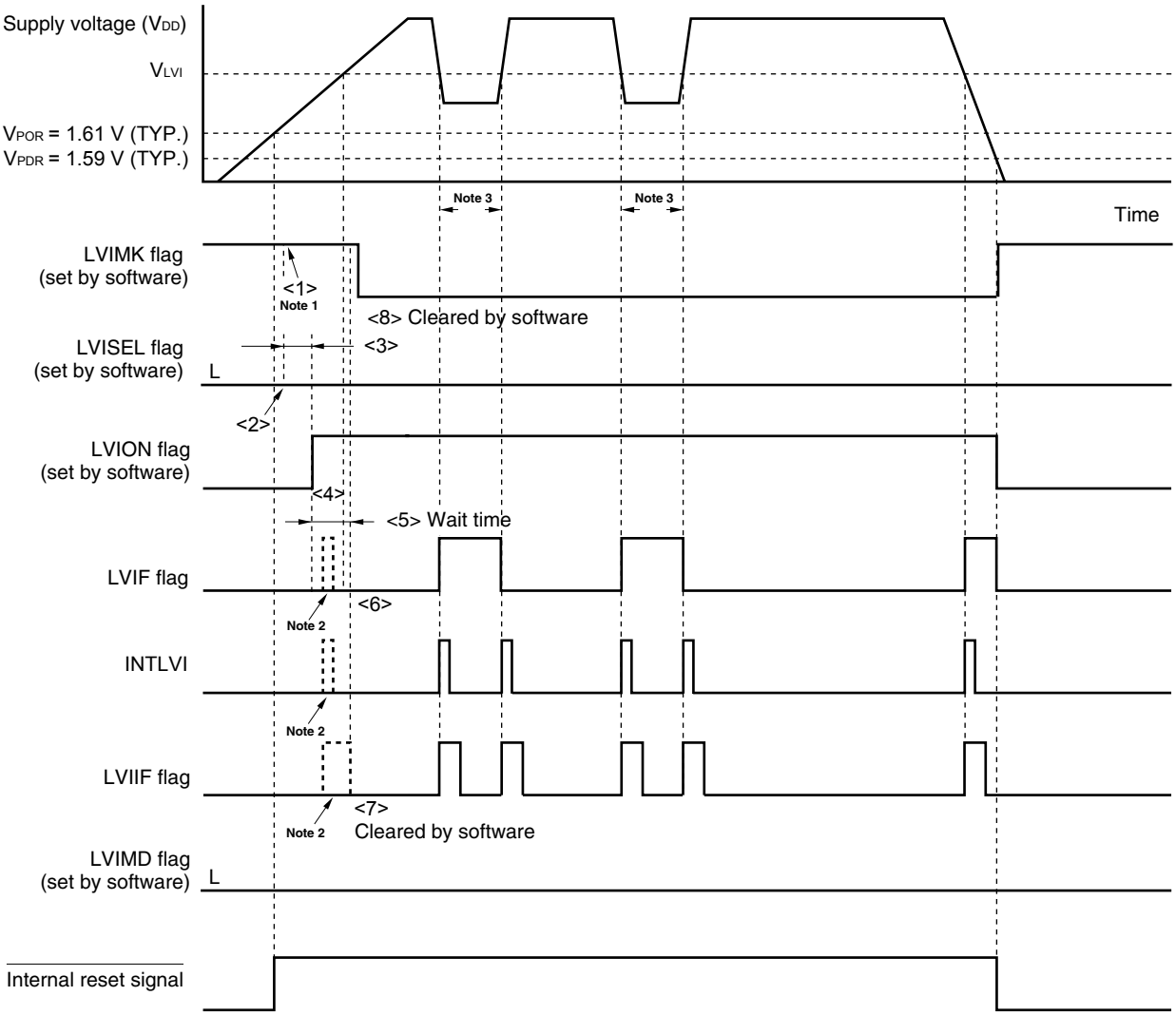
<9> Execute the EI instruction (when vector interrupts are used).

Figure 22-8 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <8> above.

- When stopping operation

Be sure to clear (0) LVION by using a 1-bit memory manipulation instruction.

Figure 22-8. Timing of Low-Voltage Detector Interrupt Signal Generation (Bit: LVISEL = 0, Option Byte: LVIOFF = 1)



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

2. The interrupt request signal (INTLVI) is generated and the LVIF and LVIIF flags may be set (1).

3. If LVI operation is disabled when the supply voltage (V_{DD}) is less than or equal to the detection voltage (V_{LVI}), an interrupt request signal (INTLVI) is generated and LVIIF may be set to 1.

Remarks 1. <1> to <8> in Figure 22-8 above correspond to <1> to <8> in the description of "When starting operation" in **22.4.2 (1) (a) When LVI default start function stopped is set (LVIOFF = 1).**

2. V_{POR}: POC power supply rise detection voltage
V_{PDR}: POC power supply fall detection voltage

(b) When LVI default start function enabled is set (LVIOFF = 0)

- When starting operation

<1> Start in the following initial setting state.

- Set bit 7 (LVION) of LVIM to 1 (enables LVI operation)
- Clear bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 0 (detects level of supply voltage (V_{DD}))
- Set the low-voltage detection level selection register (LVIS) to 0EH (default value: V_{LVI} = 2.07 V ±0.1 V).
- Set bit 1 (LVIMD) of LVIM to 1 (generates reset when the level is detected)
- Set bit 0 (LVIF) of LVIM to 0 (Detects falling edge "Supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})")

<2> Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (default value).

<3> Release the interrupt mask flag of LVI (LVIMK).

<4> Execute the EI instruction (when vector interrupts are used).

Figure 22-9 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <3> above.

- When stopping operation

Be sure to clear (0) LVION by using a 1-bit memory manipulation instruction.

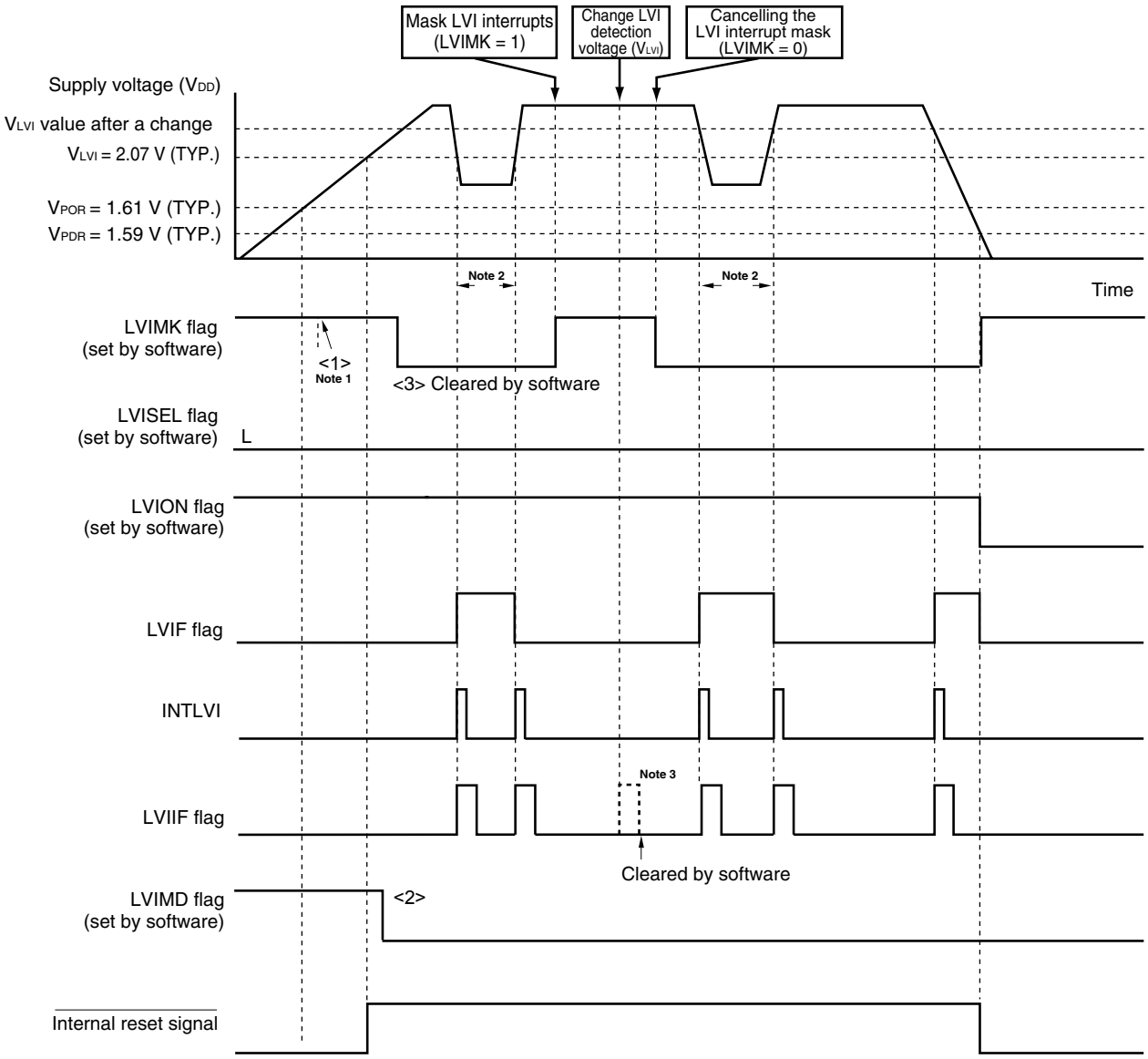
Cautions 1. Even when the LVI default start function is used, if it is set to LVI operation prohibition by the software, it operates as follows:

- **Does not perform low-voltage detection during LVION = 0.**
- **If a reset is generated while LVION = 0, LVION will be re-set to 1 when the CPU starts after reset release. There is a period when low-voltage detection cannot be performed normally, however, when a reset occurs due to WDT and illegal instruction execution.**

This is due to the fact that while the pulse width detected by LVI must be 200 μs max., LVION = 1 is set upon reset occurrence, and the CPU starts operating without waiting for the LVI stabilization time.

2. When the LVI default start function (bit 0 (LVIOFF) of 000C1H = 0) is used, the LVIRF flag may become 1 from the beginning due to the power-on waveform.
For details of RESF, see CHAPTER 20 RESET FUNCTION.

Figure 22-9. Timing of Low-Voltage Detector Interrupt Signal Generation (Bit: LVISEL = 0, Option Byte: LVIOFF = 0)



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

2. If LVI operation is disabled when the supply voltage (V_{DD}) is less than or equal to the detection voltage (V_{LVI}), an interrupt request signal (INTLVI) is generated and LVIIF may be set to 1.

3. The LVIIF flag may be set when the LVI detection voltage is changed.

Remarks 1. <1> to <3> in Figure 22-9 above correspond to <1> to <3> in the description of "When starting operation" in **22.4.2 (1) (b) When LVI default start function enabled is set (LVIOFF = 0).**

2. V_{POR}: POC power supply rise detection voltage
V_{PDR}: POC power supply fall detection voltage

(2) When detecting level of input voltage from external input pin (EXLVI)

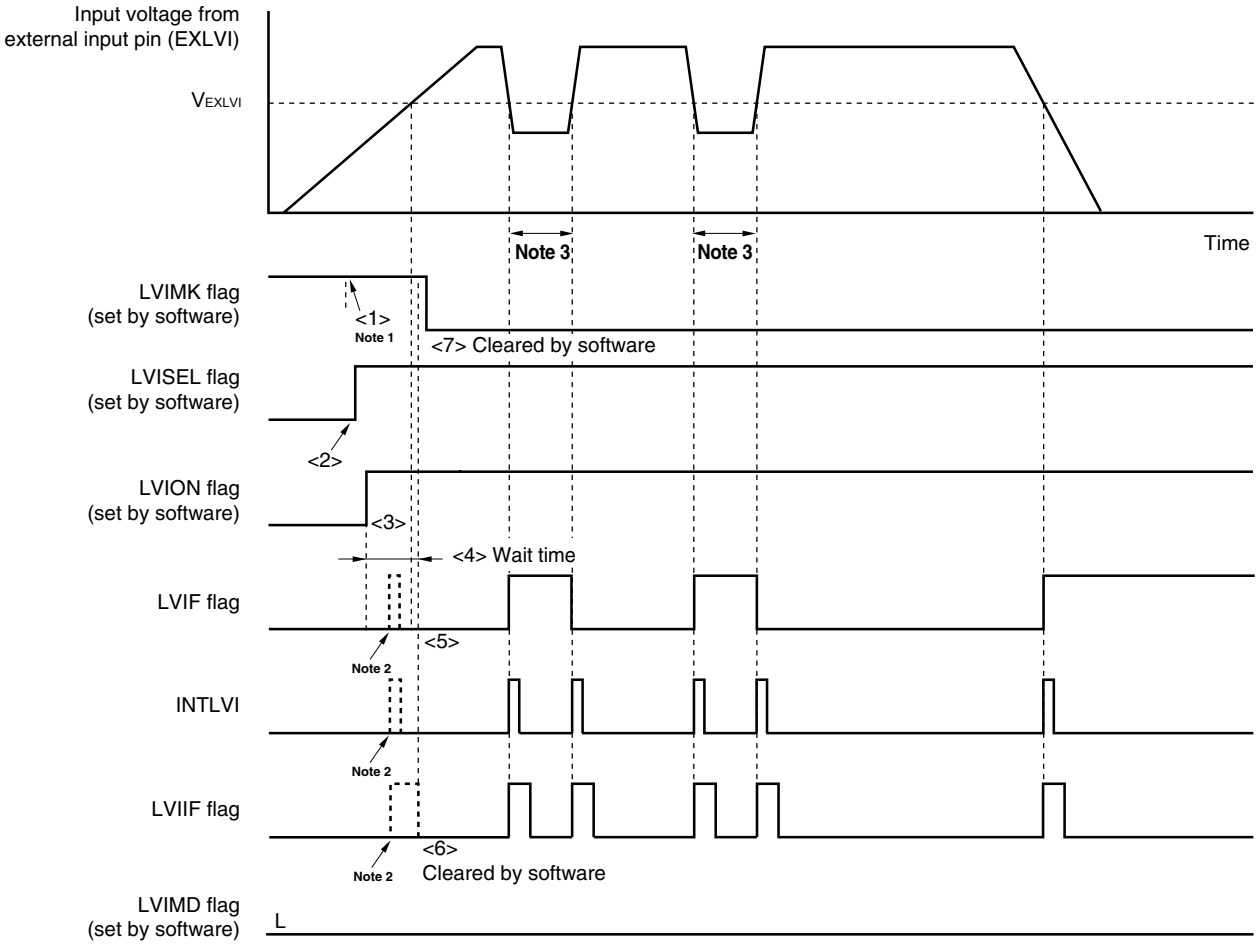
- When starting operation
  - <1> Mask the LVI interrupt (LVIMK = 1).
  - <2> Set bit 2 (LVISEL) of the low-voltage detection register (LVIM) to 1 (detects level of input voltage from external input pin (EXLVI)).
    Clear bit 1 (LVIMD) of LVIM to 0 (generates interrupt signal when the level is detected) (default value).
  - <3> Set bit 7 (LVION) of LVIM to 1 (enables LVI operation).
  - <4> Use software to wait for the following periods of time (Total 210 μs).
    - Operation stabilization time (10 μs (MAX.))
    - Minimum pulse width (200 μs (MIN.))
  - <5> Confirm that "input voltage from external input pin (EXLVI) $\geq$ detection voltage (V_{EXLVI} = 1.21 V (TYP.))" when detecting the falling edge of EXLVI, or "input voltage from external input pin (EXLVI) < detection voltage (V_{EXLVI} = 1.21 V (TYP.))" when detecting the rising edge of EXLVI, at bit 0 (LVIF) of LVIM.
  - <6> Clear the interrupt request flag of LVI (LVIIF) to 0.
  - <7> Release the interrupt mask flag of LVI (LVIMK).
  - <8> Execute the EI instruction (when vector interrupts are used).

  Figure 22-10 shows the timing of the interrupt signal generated by the low-voltage detector. The numbers in this timing chart correspond to <1> to <7> above.

  Caution Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.

- When stopping operation
  Be sure to clear (0) LVION by using a 1-bit memory manipulation instruction.

Figure 22-10. Timing of Low-Voltage Detector Interrupt Signal Generation (Bit: LVISEL = 1)



Notes 1. The LVIMK flag is set to "1" by reset signal generation.

2. The interrupt request signal (INTLVI) is generated and the LVIF and LVIIF flags may be set (1).

3. If LVI operation is disabled when the input voltage of external input pin (EXLVI) is less than or equal to the detection voltage (V_{EXLVI}), an interrupt request signal (INTLVI) is generated and LVIIF may be set to 1.

Remark <1> to <7> in Figure 22-10 above correspond to <1> to <7> in the description of "When starting operation" in **22.4.2 (2) When detecting level of input voltage from external input pin (EXLVI)**.

22.5 Cautions for Low-Voltage Detector

(1) Measures method when supply voltage (V_{DD}) frequently fluctuates in the vicinity of the LVI detection voltage (V_{LVI})

In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the LVI detection voltage (V_{LVI}), the operation is as follows depending on how the low-voltage detector is used.

Operation example 1: When used as reset

The system may be repeatedly reset and released from the reset status.

The time from reset release through microcontroller operation start can be set arbitrarily by the following action.

<Action>

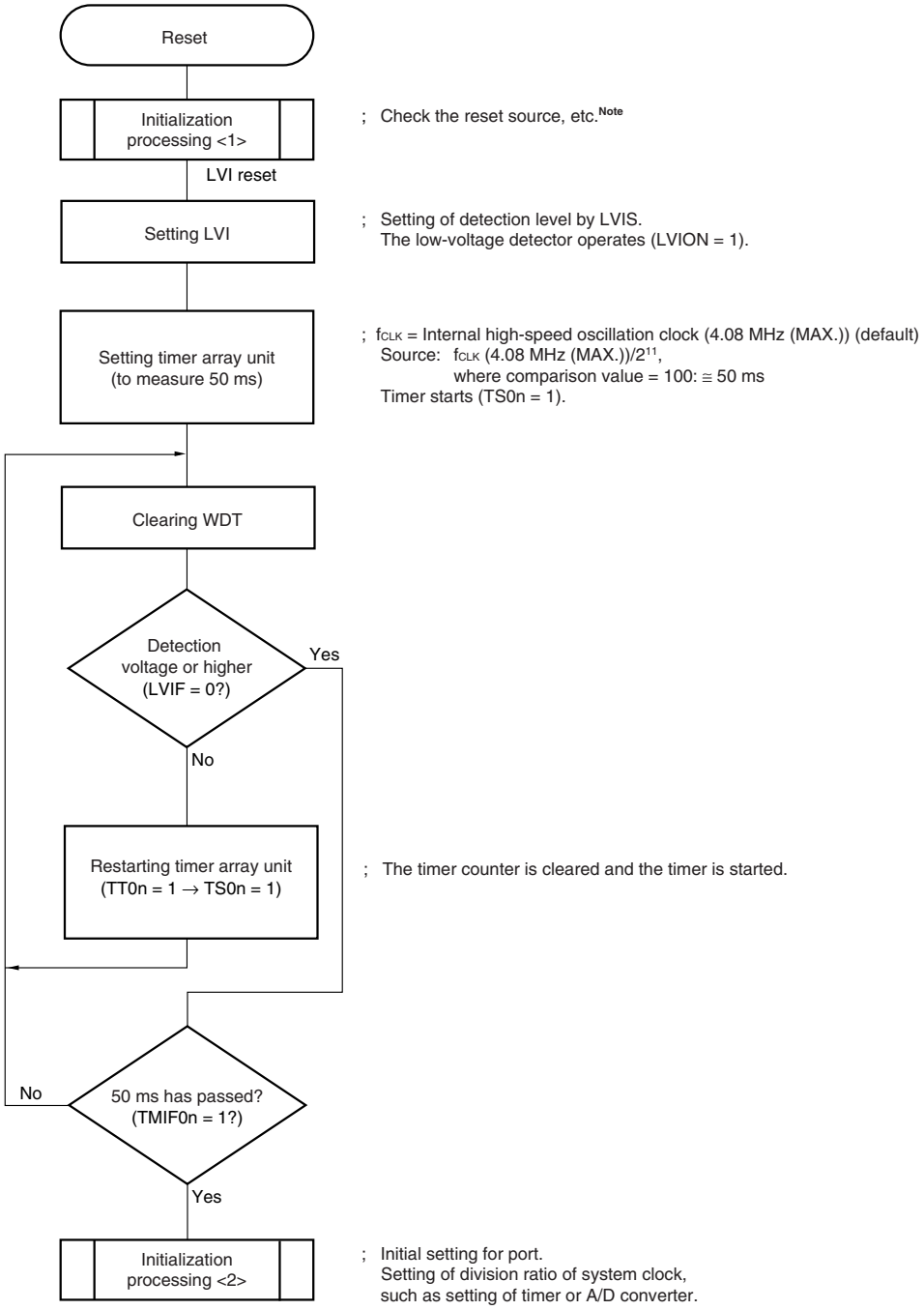
After releasing the reset signal, wait for the supply voltage fluctuation period of each system by means of a software counter that uses a timer, and then initialize the ports (see **Figure 22-11**).

Remark If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to "1", the meanings of the above words change as follows.

- Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)
- Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)

Figure 22-11. Example of Software Processing After Reset Release (1/2)

• If supply voltage fluctuation is 50 ms or less in vicinity of LVI detection voltage



Note A flowchart is shown on the next page.

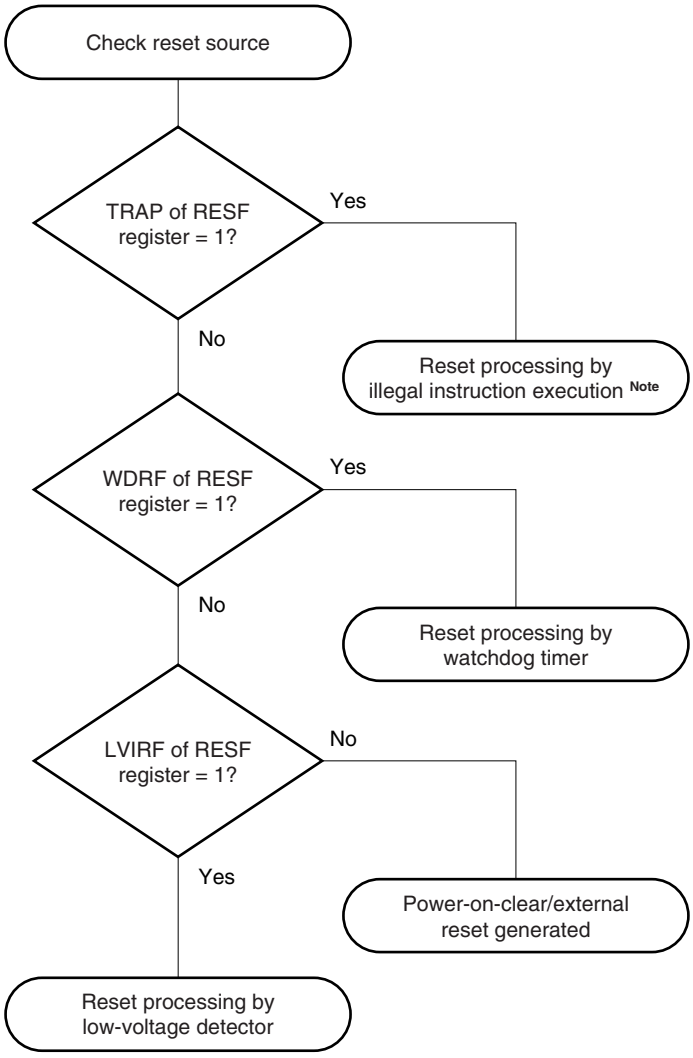
Remarks 1. If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to "1", the meanings of the above words change as follows.

- Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)
- Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)

2. n = 0 to 7

Figure 22-11. Example of Software Processing After Reset Release (2/2)

• Checking reset source



Note 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.

Remark If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to "1", the meanings of the above words change as follows.
- Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)
- Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)

Operation example 2: When used as interrupt

Interrupt requests may be generated frequently.

Take the following action.

<Action>

Confirm that "supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})" when detecting the falling edge of V_{DD}, or "supply voltage (V_{DD}) < detection voltage (V_{LVI})" when detecting the rising edge of V_{DD}, in the servicing routine of the LVI interrupt by using bit 0 (LVIF) of the low-voltage detection register (LVIM). Clear bit 1 (LVIIF) of interrupt request flag register 0L (IF0L) to 0.

For a system with a long supply voltage fluctuation period near the LVI detection voltage, take the above action after waiting for the supply voltage fluctuation time.

Remark If bit 2 (LVISEL) of the low voltage detection register (LVIM) is set to "1", the meanings of the above words change as follows.

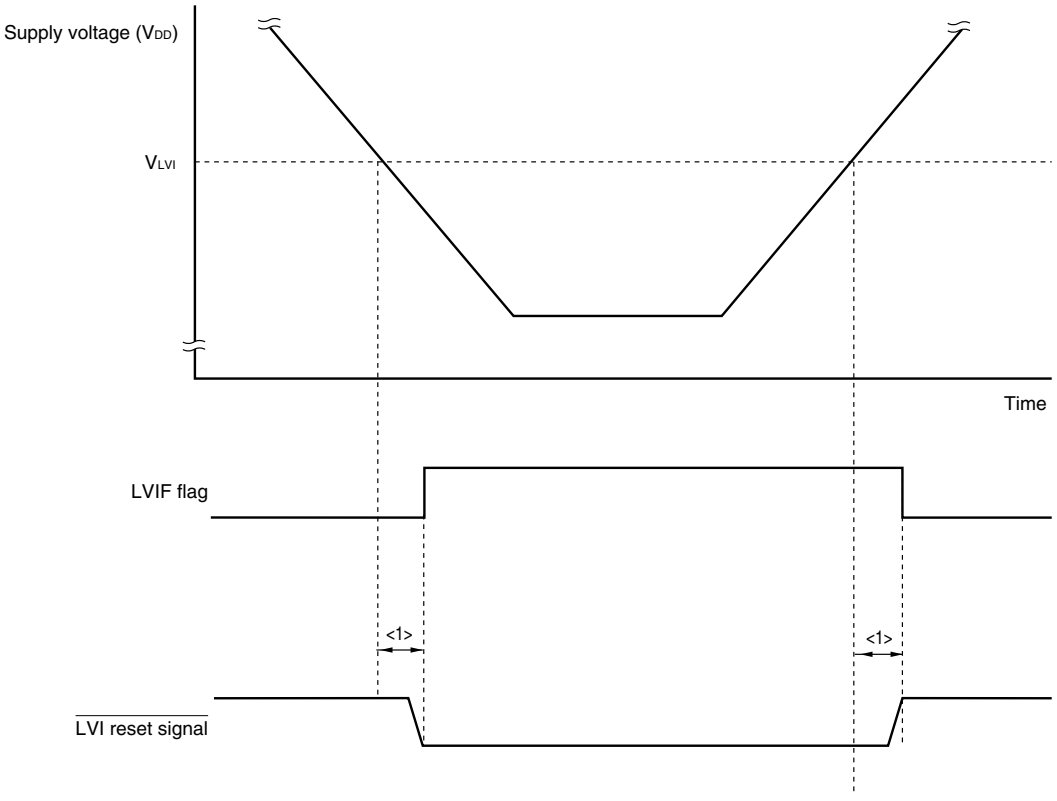
- Supply voltage (V_{DD}) → Input voltage from external input pin (EXLVI)
- Detection voltage (V_{LVI}) → Detection voltage (V_{EXLVI} = 1.21 V)

(2) Delay from the time LVI reset source is generated until the time LVI reset has been generated or released

There is some delay from the time supply voltage (V_{DD}) < LVI detection voltage (V_{LVI}) until the time LVI reset has been generated.

In the same way, there is also some delay from the time LVI detection voltage (V_{LVI}) ≤ supply voltage (V_{DD}) until the time LVI reset has been released (see **Figure 22-12**).

Figure 22-12. Delay from the time LVI reset source is generated until the time LVI reset has been generated or released



<1>: Minimum pulse width (200 μs (MIN.))

CHAPTER 23 REGULATOR

23.1 Regulator Overview

All products of 78K0R/Kx3-A microcontrollers contain 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.

The regulator output voltage is normally 2.4 V (TYP.), and in the low-power consumption mode, 1.8 V (TYP.).

23.2 Registers Controlling Regulator

(1) Regulator mode control register (RMC)

This register sets the output voltage of the regulator.

RMC is set with an 8-bit memory manipulation instruction.

Reset input sets this register to 00H.

Figure 23-1. Format of Regulator Mode Control Register (RMC)

Address: F00F4H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
RMC								

RMC[7:0]	Control of output voltage of regulator
5AH	Fixed to low-power consumption mode (1.8 V)
00H	Switches normal power mode (2.4 V) and low-power consumption mode (1.8 V) according to the condition (refer to **Table 23-1**)
Other than above	Setting prohibited

Cautions 1. The RMC register can be rewritten only in the low-power consumption mode (refer to Table 23-1). In other words, rewrite this register during CPU operation with the subsystem clock (f_{XT}) while the high-speed system clock (f_{MX}), the high-speed internal oscillation clock (f_{IH}), and the 20 MHz internal high-speed oscillation clock (f_{IH20}) are both stopped.

2. When using the setting fixed to the low consumption current mode, the RMC register can be used in the following cases.

<When the high-speed internal oscillation clock (f_{IH} = 8 MHz (TYP.) or f_{IH} = 1 MHz (TYP.)) is selected as the CPU clock>

f_{CLK} ≤ 1 MHz and external oscillator (X1 clock (f_X), external main system clock (f_{EX})) stop.

<When the X1 clock (f_X) or external main system clock (f_{EX}) is selected as the CPU clock>

f_{CLK} ≤ 1 MHz, f_X/f_{EX} ≤ 5 MHz and the internal high-speed oscillator stop.

<When the subsystem clock (f_{SUB}) is selected as the CPU clock>

Both the internal high-speed oscillator and external oscillator (f_X/f_{EX} ≤ 5 MHz) stop or either one stops.

Cautions 3. In low-power consumption mode, use the regulator with f_{CLK} fixed to 1 MHz when executing self programming.

4. A wait is required to change the operation speed mode control register (OSMC) after changing the RMC register. Wait for 2 ms by software when setting to low-power consumption mode and 10 μs when setting to normal power mode, as described in the procedure shown below.

• When setting to low-power consumption mode

<1> Select a frequency of 1 MHz for f_{CLK}.
<2> Set RMC to 5AH (set the regulator to low-power consumption mode).
<3> Wait for 2 ms.
<4> Set FLPC and FSEL of OSMC to 1 and 0, respectively.

• When setting to normal power mode

<1> Set RMC to 00H (set the regulator to normal power mode).
<2> Wait for 10 μs.
<3> Change FLPC and FSEL of OSMC.
<4> Change the f_{CLK} frequency.

Table 23-1. Regulator Output Voltage Conditions

Mode	Output Voltage	Condition
Low-power consumption mode	1.8 V	In STOP mode (except during OCD mode)
		When both the high-speed system clock (f_{MX}), the high-speed internal oscillation clock (f_{IH}), and the 20 MHz internal high-speed oscillation clock (f_{IH20}) are stopped during CPU operation with the subsystem clock (f_{SUB})
		When both the high-speed system clock (f_{MX}), the high-speed internal oscillation clock (f_{IH}), and the 20 MHz internal high-speed oscillation clock (f_{IH20}) are stopped during the HALT mode when the CPU operation with the subsystem clock (f_{SUB}) has been set
Normal power mode	2.4 V	Other than above

CHAPTER 24 OPTION BYTE

24.1 Functions of Option Bytes

Addresses 000C0H to 000C3H of the flash memory of the 78K0R/Kx3-A microcontrollers 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.

To use the boot swap operation during self programming, 000C0H to 000C3H are replaced by 010C0H to 010C3H. Therefore, set the same values as 000C0H to 000C3H to 010C0H to 010C3H.

Caution Be sure to set FFH to 000C2H (000C2H/010C2H when the boot swap operation is used).

24.1.1 User option byte (000C0H to 000C2H/010C0H to 010C2H)

(1) 000C0H/010C0H

- ○ Operation of watchdog timer
  - Operation is stopped or enabled in the HALT or STOP mode.
- ○ Setting of interval time of watchdog timer
- ○ Operation of watchdog timer
  - Operation is stopped or enabled.
- ○ Setting of window open period of watchdog timer
- ○ Setting of interval interrupt of watchdog timer
  - 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 LVI upon reset release (upon power application)
  - LVI is ON or OFF by default upon reset release (reset by $\overline{\text{RESET}}$ pin excluding LVI, POC, WDT, or illegal instructions).
- ○ Setting of internal high-speed oscillator frequency
  - Select from 1 MHz, 8 MHz, or 20 MHz.

Caution Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

(3) 000C2H/010C2H

- ○ Be sure to set FFH, as these addresses are reserved areas.

Caution Set FFH to 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.

24.1.2 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.

24.2 Format of User Option Byte

Figure 24-1. Format of User Option Byte (000C0H/010C0H) (1/2)

Address: 000C0H/010C0H[Note 1]

7	6	5	4	3	2	1	0
WDTINIT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

WDTINIT	Use of interval interrupt of watchdog timer
0	Interval interrupt is not used.
1	Interval interrupt is generated when 75% of the overflow time is reached.

<R>

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} = 33 kHz (MAX.))
0	0	0	$2^7/f_{IL}$ (3.88 ms)
0	0	1	$2^8/f_{IL}$ (7.76 ms)
0	1	0	$2^9/f_{IL}$ (15.52 ms)
0	1	1	$2^{10}/f_{IL}$ (31.03 ms)
1	0	0	$2^{12}/f_{IL}$ (124.12 ms)
1	0	1	$2^{14}/f_{IL}$ (496.48 ms)
1	1	0	$2^{15}/f_{IL}$ (992.97 ms)
1	1	1	$2^{17}/f_{IL}$ (3971.88 ms)

Figure 24-1. Format of User Option Byte (000C0H/010C0H) (2/2)

Address: 000C0H/010C0H[Note 1]

7	6	5	4	3	2	1	0
WDTINIT	WINDOW1	WINDOW0	WDTON	WDCS2	WDCS1	WDCS0	WDSTBYON

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

Notes 1. Set the same value as 000C0H to 010C0H when the boot swap operation is used because 000C0H is replaced by 010C0H.

2. The window open period is 100% when WDSTBYON = 0, regardless the value of WINDOW1 and WINDOW0.

Caution The watchdog timer continues its operation during self-programming of the flash memory and EEPROM emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

Remark f_{IL}: Internal low-speed oscillation clock frequency

Figure 24-2. Format of User Option Byte (000C1H/010C1H)

Address: 000C1H/010C1H[Note 1]

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

FRQSEL2	FRQSEL1	Internal high-speed oscillator frequency
0	1	8 MHz/20 MHz [Note 2]
1	0	1 MHz [Note 3]
Other than the above		Setting prohibited

LVIOFF	Setting of LVI on power application
0	LVI is ON by default (LVI default start function enabled) upon reset release (upon power application)
1	LVI is OFF by default (LVI default start function stopped) upon reset release (upon power application)

Notes 1. Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.

2. When 8 MHz or 20 MHz has been selected, the 8 MHz internal high-speed oscillator automatically starts oscillating after reset release. To use the 20 MHz internal high-speed oscillator to operate the microcontroller, oscillation is started by setting bit 0 (DSCON) of the 20 MHz internal high-speed oscillation control register (DSCCTL) to 1 with $V_{DD} \geq 2.7$ V. The circuit cannot be changed to a 1 MHz internal high-speed oscillator while the microcontroller operates.

3. When 1 MHz has been selected, the microcontroller operates on the 1 MHz internal high-speed oscillator after reset release. The circuit cannot be changed to an 8 MHz or 20 MHz internal high-speed oscillator while the microcontroller operates.

(**Cautions** are listed on the next page.)

Cautions 1. Be sure to set bits 7 to 3 to "1".

2. Even when the LVI default start function is used, if it is set to LVI operation prohibition by the software, it operates as follows:

- **Does not perform low-voltage detection during LVION = 0.**
- **If a reset is generated while LVION = 0, LVION will be re-set to 1 when the CPU starts after reset release. There is a period when low-voltage detection cannot be performed normally, however, when a reset occurs due to WDT and illegal instruction execution. This is due to the fact that while the pulse width detected by LVI must be 200 μs max., LVION = 1 is set upon reset occurrence, and the CPU starts operating without waiting for the LVI stabilization time.**

Figure 24-3. Format of Option Byte (000C2H/010C2H)

Address: 000C2H/010C2H[Note]

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

Note Be sure to set FFH to 000C2H, as these addresses are reserved areas. Also set FFH to 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.

24.3 Format of On-chip Debug Option Byte

The format of on-chip debug option byte is shown below.

Figure 24-4. Format of On-chip Debug Option Byte (000C3H/010C3H)

Address: 000C3H/010C3H[Note]

7	6	5	4	3	2	1	0
OCDENSET	0	0	0	0	1	0	OCDERSD

OCDENSET	OCDERSD	Control of on-chip debug operation
0	0	Disables on-chip debug operation.
0	1	Setting prohibited
1	0	Erases data of flash memory in case of failures in enabling on-chip debugging and authenticating on-chip debug security ID.
1	1	Does not erases data of flash memory in case of failures in enabling on-chip debugging and 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.

24.4 Setting of Option Byte

The user option byte and on-chip debug option byte can be set using the RA78K0R or PM+ linker option, in addition to describing to the source. When doing so, the contents set by using the linker option take precedence, even if descriptions exist in the source, as mentioned below.

See the RA78K0R Assembler Package User's Manual for how to set the linker option.

A software description example of the option byte setting is shown below.

```
OPT   CSEG        OPT_BYTE
<R>   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 2^10/fIL,
                          ; Stops watchdog timer operation during HALT/STOP mode
<R>   DB          0FBH    ; Select 8 MHz or 20 MHz for internal high-speed oscillator
                          ; Stops LVI default start function
      DB          0FFH    ; Reserved area
      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
<R>   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 2^10/fIL,
                          ; Stops watchdog timer operation during HALT/STOP mode
<R>   DB          0FBH    ; Select 8 MHz or 20 MHz for internal high-speed oscillator
                          ; Stops LVI default start function
      DB          0FFH    ; Reserved area
      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 25 FLASH MEMORY

The 78K0R/Kx3-A microcontrollers incorporate the flash memory to which a program can be written, erased, and overwritten while mounted on the board.

25.1 Writing with Flash Memory Programmer

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 78K0R/Kx3-A microcontrollers have 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

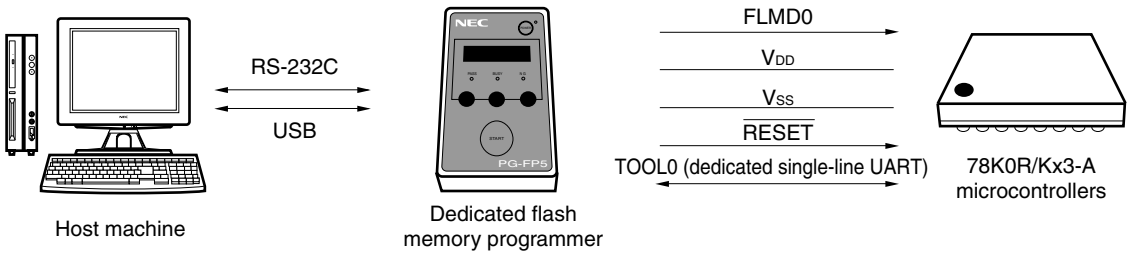
Data can be written to the flash memory with a dedicated program adapter (FA series) before the 78K0R/Kx3-A microcontrollers are mounted on the target system.

Remark The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

25.2 Programming Environment

The environment required for writing a program to the flash memory of the 78K0R/Kx3-A microcontrollers is illustrated below.

Figure 25-1. Environment for Writing Program to Flash Memory



A host machine that controls the dedicated flash memory programmer is necessary.

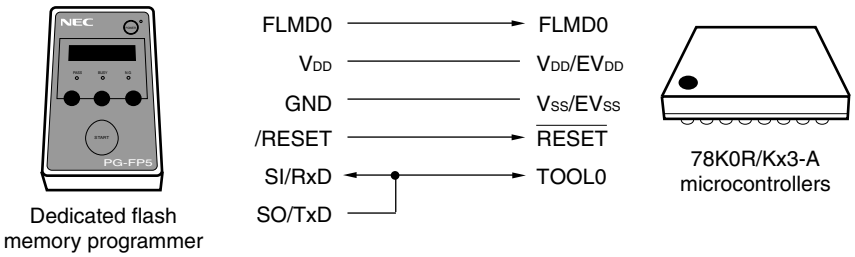
To interface between the dedicated flash memory programmer and the 78K0R/Kx3-A microcontrollers, the TOOL0 pin is used for manipulation such as writing and erasing via a dedicated single-line UART. To write the flash memory off-board, a dedicated program adapter (FA series) is necessary.

25.3 Communication Mode

Communication between the dedicated flash memory programmer and the 78K0R/Kx3-A microcontrollers is established by serial communication using the TOOL0 pin via a dedicated single-line UART of the 78K0R/Kx3-A microcontrollers.

Transfer rate: 115,200 bps to 1,000,000 bps

Figure 25-2. Communication with Dedicated Flash Memory Programmer



When using the FlashPro5 as the dedicated flash memory programmer, the FlashPro5 generates the following signals for the 78K0R/Kx3-A microcontrollers. For details, refer to the user's manual for the FlashPro5.

Table 25-1. Pin Connection

FlashPro5			78K0R/Kx3-A microcontrollers	Connection
Signal Name	I/O	Pin Function	Pin Name	
FLMD0	Output	Mode signal	FLMD0	◎
V_{DD}	I/O	V_{DD} voltage generation/power monitoring	V_{DD}, EV_{DD}, AV_{DD0}, AV_{DD1}	◎
GND	–	Ground	V_{SS}, EV_{SS}, AV_{SS}	◎
CLK	Output	Clock output	–	×
/RESET	Output	Reset signal	$\overline{\text{RESET}}$	◎
SI/RxD	Input	Receive signal	TOOL0	◎
SO/TxD	Output	Transmit signal		
SCK	Output	Transfer clock	–	×

Remark ◎: Be sure to connect the pin.
×: The pin does not have to be connected.

25.4 Connection of Pins on Board

To write the flash memory on-board, 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.

25.4.1 FLMD0 pin

(1) In flash memory programming mode

Directly connect this pin to a flash memory programmer when data is written by the flash memory programmer. This supplies a writing voltage of the V_{DD} level to the FLMD0 pin.

The FLMD0 pin does not have to be pulled down externally because it is internally pulled down by reset. To pull it down externally, use a resistor of 1 kΩ to 200 kΩ.

(2) In normal operation mode

It is recommended to leave this pin open during normal operation.

The FLMD0 pin must always be kept at the V_{SS} level before reset release but does not have to be pulled down externally because it is internally pulled down by reset. However, pulling it down must be kept selected (i.e., FLMDPUP = “0”, default value) by using bit 7 (FLMDPUP) of the background event control register (BECTL) (see **25.5 (1) Back ground event control register**). To pull it down externally, use a resistor of 200 kΩ or smaller.

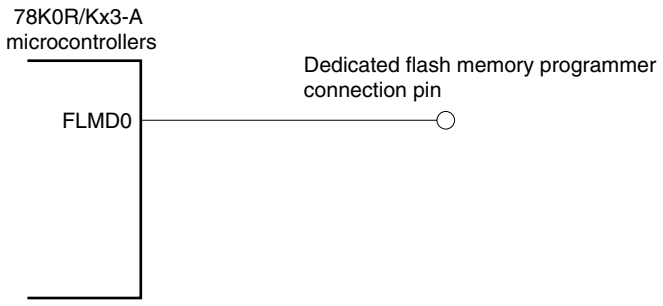
Self programming and the rewriting of flash memory with the programmer can be prohibited using hardware, by directly connecting this pin to the V_{SS} pin.

(3) In self programming mode

It is recommended to leave this pin open when using the self programming function. To pull it down externally, use a resistor of 100 kΩ to 200 kΩ.

In the self programming mode, the setting is switched to pull up in the self programming library.

Figure 25-3. FLMD0 Pin Connection Example



25.4.2 TOOL0 pin

In the flash memory programming mode, connect this pin directly to the dedicated flash memory programmer or pull it up by connecting it to EV_{DD} via an external resistor.

When on-chip debugging is enabled in the normal operation mode, pull this pin up by connecting it to V_{DD} via an external resistor, and be sure to keep inputting the V_{DD} level to the TOOL0 pin before reset is released (pulling down this pin is prohibited).

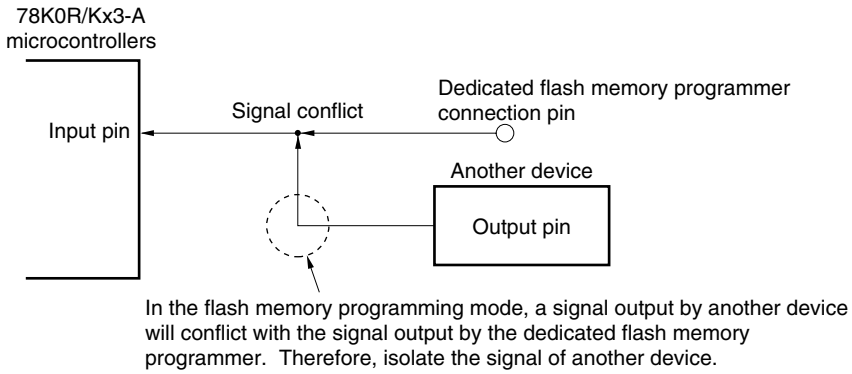
Remark The SAU and IICA pins are not used for communication between the 78K0R/Kx3-A microcontrollers and dedicated flash memory programmer, because single-line UART is used.

25.4.3 $\overline{\text{RESET}}$ pin

Signal conflict will occur if the reset signal of the dedicated flash memory programmer is connected to the $\overline{\text{RESET}}$ pin that is connected to the reset signal generator on the board. To prevent this conflict, isolate the connection with the reset signal generator.

The flash memory will not be correctly programmed if the reset signal is input from the user system while the flash memory programming mode is set . Do not input any signal other than the reset signal of the dedicated flash memory programmer.

Figure 25-4. Signal Conflict ($\overline{\text{RESET}}$ Pin)



25.4.4 Port pins

When the flash memory programming mode is set, all the pins not used for flash memory programming enter the same status as that immediately after reset. If external devices connected to the ports do not recognize the port status immediately after reset, the port pin must be connected to V_{DD} or V_{SS} via a resistor.

25.4.5 REGC pin

Connect the REGC pin to GND via a capacitor (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.

25.4.6 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 internal high-speed oscillation clock (f_{IH}) is used.

25.4.7 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 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.

Supply the same other power supplies (EV_{DD}, EV_{SS}, AV_{DD0}, AV_{DD1}, and AV_{SS}) as those in the normal operation mode.

25.5 Registers Controlling Flash Memory

(1) Background event control register (BECTL)

Even if the FLMD0 pin is not controlled externally, it can be controlled by software with the BECTL register to set the self-programming mode.

However, depending on the processing of the FLMD0 pin, it may not be possible to set the self-programming mode by software. When using BECTL, leaving the FLMD0 pin open is recommended. When pulling it down externally, use a resistor with a resistance of 100 kΩ or more. In addition, in the normal operation mode, use BECTL with the pull down selection. In the self-programming mode, the setting is switched to pull up in the self- programming library.

The BECTL register is set by a 1-bit or 8-bit memory manipulation instruction.

Reset input sets this register to 00H.

Figure 25-5. Format of Background Event Control Register (BECTL)

Address: FFFBEH After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
BECTL	FLMDPUP	0	0	0	0	0	0	0

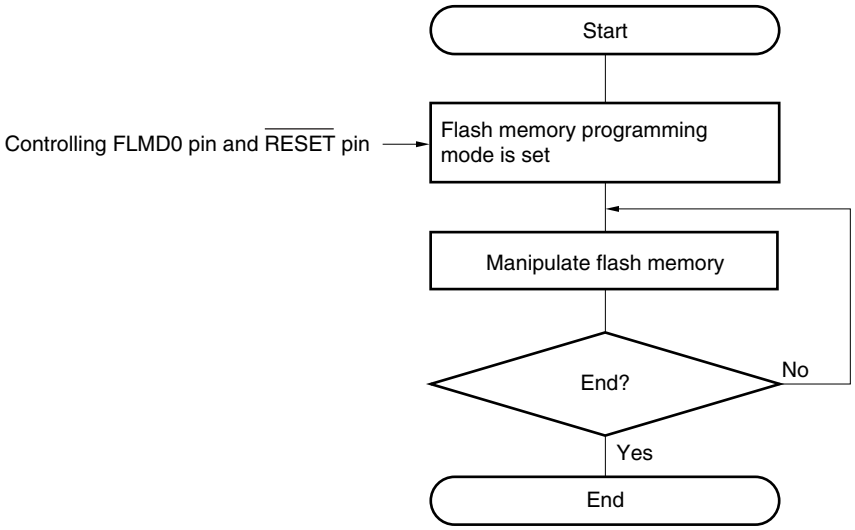
FLMDPUP	Software control of FLMD0 pin
0	Selects pull-down
1	Selects pull-up

25.6 Programming Method

25.6.1 Controlling flash memory

The following figure illustrates the procedure to manipulate the flash memory.

Figure 25-6. Flash Memory Manipulation Procedure



25.6.2 Flash memory programming mode

To rewrite the contents of the flash memory by using the dedicated flash memory programmer, set the 78K0R/Kx3-A microcontrollers in the flash memory programming mode. To set the mode, set the FLMD0 pin and TOOL0 pin to V_{DD} and clear the reset signal.

Change the mode by using a jumper when writing the flash memory on-board.

Figure 25-7. Flash Memory Programming Mode

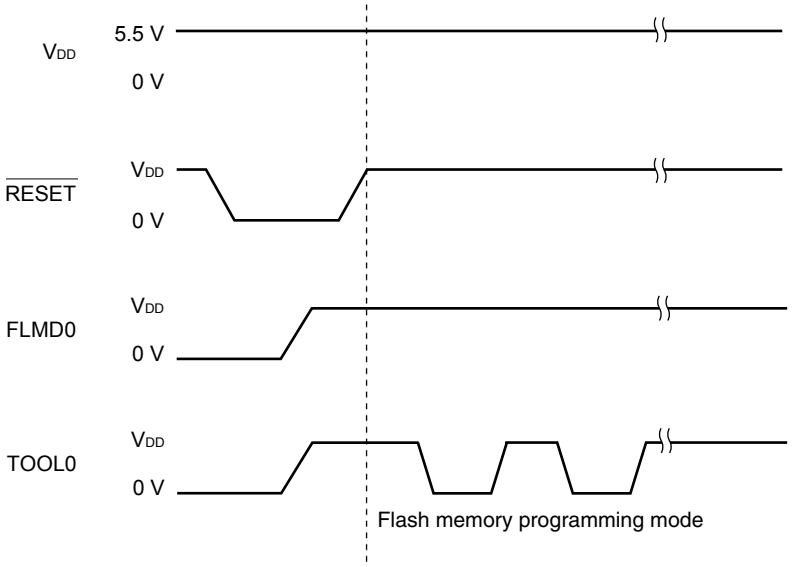


Table 25-2. Relationship Between FLMD0 Pin and Operation Mode After Reset Release

FLMD0	Operation Mode
0	Normal operation mode
V_{DD}	Flash memory programming mode

25.6.3 Selecting communication mode

Communication mode of the 78K0R/Kx3-A microcontrollers as follows.

Table 25-3. Communication Modes

Communication Mode	Standard Setting[Note 1]				Pins Used
	Port	Speed	Frequency	Multiply Rate	
1-line mode (dedicated single-line UART)	UART-ch0	1 Mbps[Note 2]	–	–	TOOL0

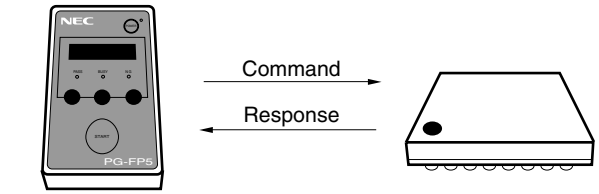
Notes 1. Selection items for Standard settings on GUI of the flash memory programmer.

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.

25.6.4 Communication commands

The 78K0R/Kx3-A microcontrollers communicate with the dedicated flash memory programmer by using commands. The signals sent from the flash memory programmer to the 78K0R/Kx3-A microcontrollers are called commands, and the signals sent from the 78K0R/Kx3-A microcontrollers to the dedicated flash memory programmer are called response.

Figure 25-8. Communication Commands



The flash memory control commands of the 78K0R/Kx3-A microcontrollers are listed in the table below. All these commands are issued from the programmer, and the 78K0R/Kx3-A microcontrollers perform processing corresponding to the respective commands.

Table 25-4. 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	Chip Erase	Erases the entire flash memory.
	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.
Getting information	Silicon Signature	Gets 78K0R/Kx3-A microcontrollers' information (such as the part number and flash memory configuration).
	Version Get	Gets the 78K0R/Kx3-A microcontrollers' firmware version.
	Checksum	Gets the checksum data for a specified area.
Security	Security Set	Sets security information.
Others	Reset	Used to detect synchronization status of communication.
	Baud Rate Set	Sets baud rate when UART communication mode is selected.

The 78K0R/Kx3-A microcontrollers return a response for the command issued by the dedicated flash memory programmer. The response names sent from the 78K0R/Kx3-A microcontrollers are listed below.

Table 25-5. Response Names

Response Name	Function
ACK	Acknowledges command/data.
NAK	Acknowledges illegal command/data.

25.7 Security Settings

The 78K0R/Kx3-A microcontrollers support 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. The security setting is valid when the programming mode is set next.

- Disabling batch erase (chip erase)
  Execution of the block erase and batch erase (chip erase) commands for entire blocks in the flash memory is prohibited by this setting during on-board/off-board programming. Once execution of the batch erase (chip erase) command is prohibited, all of the prohibition settings (including prohibition of batch erase (chip erase)) can no longer be cancelled.

  Caution After the security setting for the batch erase is set, erasure cannot be performed for the device. In addition, even if a write command is executed, data different from that which has already been written to the flash memory cannot be written, because the erase command is disabled.

- Disabling block erase
  Execution of the block erase command for a specific block in the flash memory is prohibited during on-board/off-board programming. However, blocks can be erased by means of self programming.

- Disabling write
  Execution of the write and block erase commands for entire blocks in the flash memory is prohibited during on-board/off-board programming. However, blocks can be written by means of self programming.

- Disabling rewriting boot cluster 0
  Execution of the batch erase (chip erase) command, block erase command, and write command on boot cluster 0 (00000H to 00FFFH) in the flash memory is prohibited by this setting.

  Caution If a security setting that rewrites boot cluster 0 has been applied, boot cluster 0 of that device will not be rewritten, and the entire flash memory of the device will not be erased in batch.

The batch erase (chip erase), 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 on-board/off-board programming and self programming. Each security setting can be used in combination.

All the security settings are cleared by executing the batch erase (chip erase) command.

Table 25-6 shows the relationship between the erase and write commands when the 78K0R/Kx3-A microcontrollers' security function is enabled.

Remark To prohibit writing and erasing during self-programming, use the flash sealed window function (see **25.8.2** for detail).

Table 25-6. Relationship Between Enabling Security Function and Command

(1) During on-board/off-board programming

Valid Security	Executed Command		
	Batch Erase (Chip Erase)	Block Erase	Write
Prohibition of batch erase (chip erase)	Cannot be erased in batch	Blocks cannot be erased.	Can be performed[Note].
Prohibition of block erase	Can be erased in batch.		Can be performed.
Prohibition of writing			Cannot be performed.
Prohibition of rewriting boot cluster 0	Cannot be erased in batch	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 batch erase (chip 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 batch erase (chip erase)	Blocks can be erased.	Can be performed.
Prohibition of block erase		
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 sealed window function (see **25.8.2** for detail).

Table 25-7. Setting Security in Each Programming Mode

(1) On-board/off-board programming

Security	Security Setting	How to Disable Security Setting
Prohibition of batch erase (chip erase)	Set via GUI of dedicated flash memory programmer, etc.	Cannot be disabled after set.
Prohibition of block erase		Execute batch erase (chip erase) command
Prohibition of writing		
Prohibition of rewriting boot cluster 0		Cannot be disabled after set.

(2) Self programming

Security	Security Setting	How to Disable Security Setting
Prohibition of batch erase (chip erase)	Set by using information library.	Cannot be disabled after set.
Prohibition of block erase		Execute batch erase (chip erase) command during on-board/off-board programming (cannot be disabled during self programming)
Prohibition of writing		
Prohibition of rewriting boot cluster 0		

25.8 Flash Memory Programming by Self-Programming

The 78K0R/Kx3-A microcontrollers support 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 78K0R/Kx3-A microcontrollers' self-programming library, it can be used to upgrade the program in the field.

If an interrupt occurs during self-programming, self-programming can be temporarily stopped and interrupt servicing can be executed. If an unmasked interrupt request is generated in the EI state, the request branches directly from the self-programming library to the interrupt routine. After the self-programming mode is later restored, self-programming can be resumed. However, the interrupt response time is different from that of the normal operation mode.

Cautions 1. The self-programming function cannot be used when the CPU operates with the subsystem clock.

2. In the self-programming mode, call the self-programming start library (FlashStart).

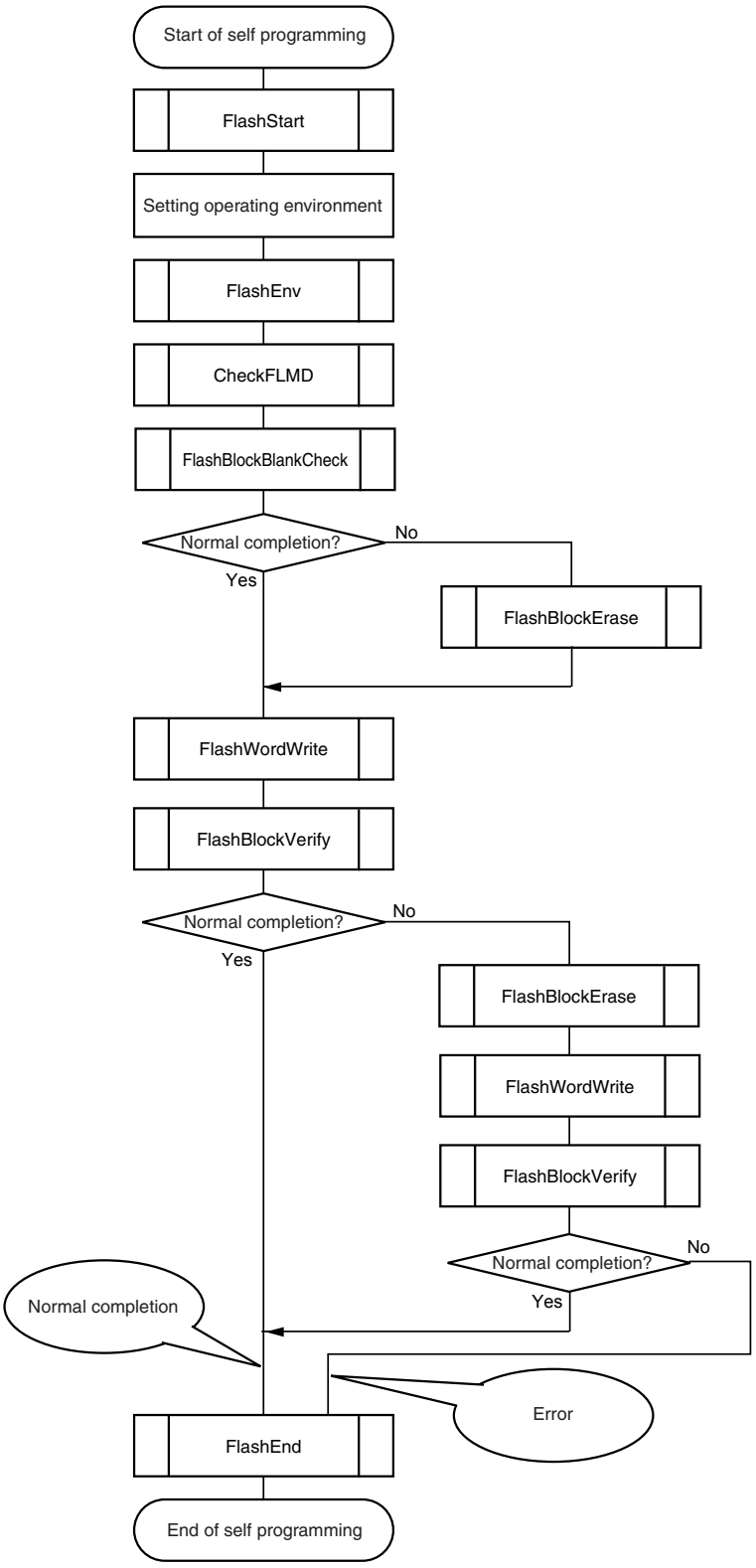
3. To prohibit an interrupt during self-programming, in the same way as in the normal operation mode, execute the self-programming library in the state where the IE flag is cleared (0) by the DI instruction. To enable an interrupt, clear (0) the interrupt mask flag to accept in the state where the IE flag is set (1) by the EI instruction, and then execute the self-programming library.

4. In low-power-consumption mode, use the regulator with f_{CLK} fixed to 1 MHz when executing self programming. For details of the low-power-consumption mode, see CHAPTER 23 REGULATOR.

5. Disable DMA operation (DENn = 0) during the execution of self programming library functions.

The following figure illustrates a flow of rewriting the flash memory by using a self programming library.

Figure 25-9. Flow of Self Programming (Rewriting Flash Memory)



25.8.1 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.

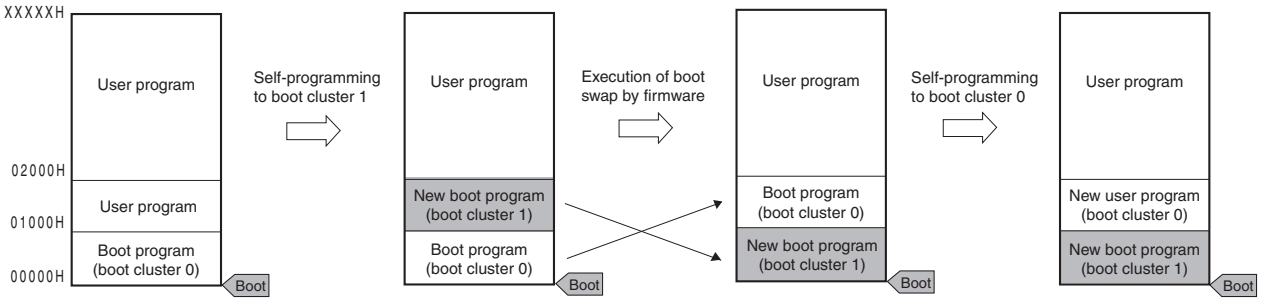
The boot swap function is used to avoid this problem.

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 78K0R/Kx3-A microcontrollers, 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 boot programming 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 25-10. Boot Swap Function

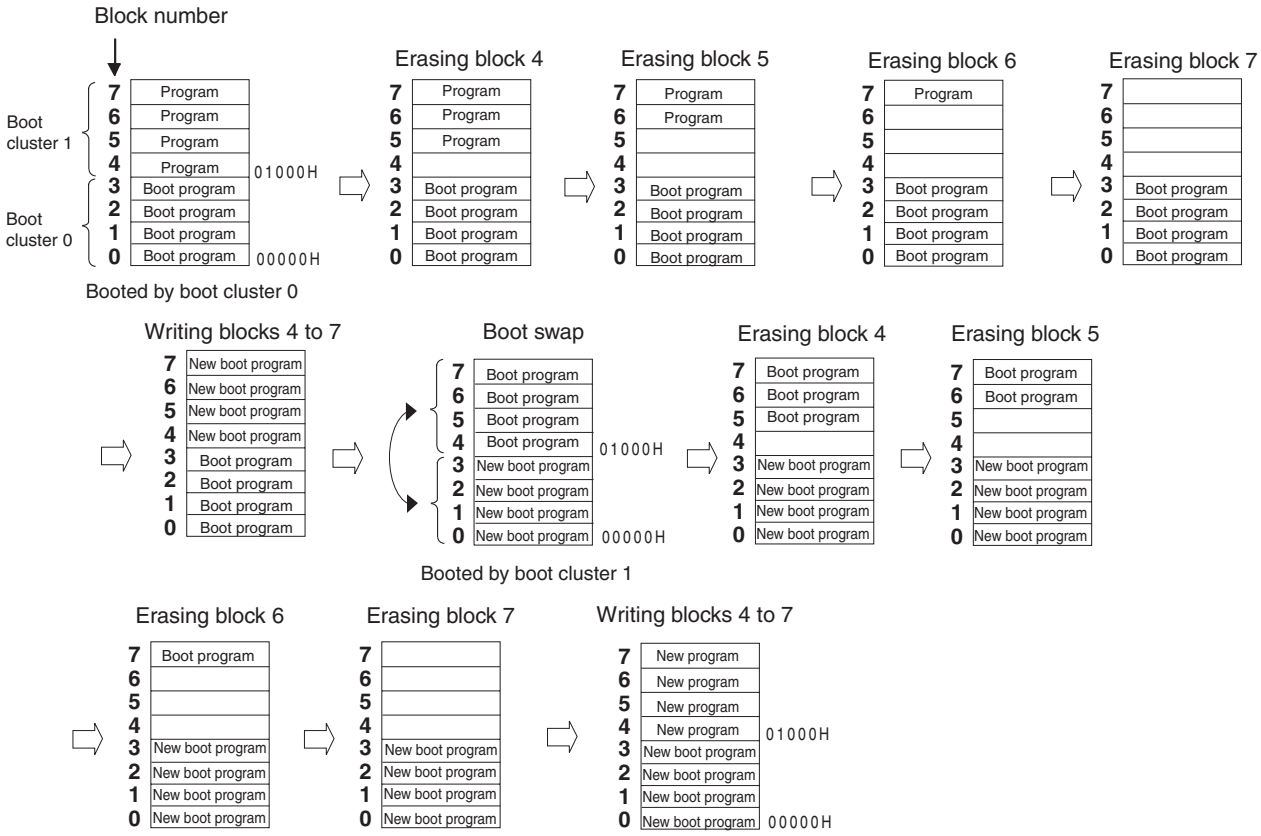


In an example of above figure, it is as follows.

Boot cluster 0: Boot program area before boot swap

Boot cluster 1: Boot program area after boot swap

Figure 25-11. Example of Executing Boot Swapping



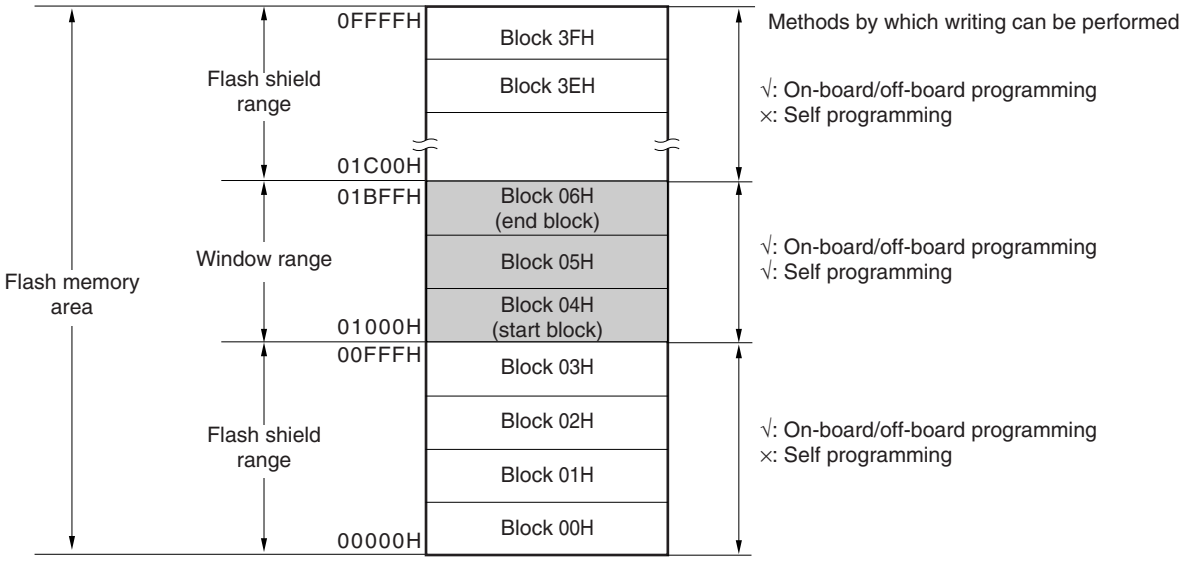
25.8.2 Flash shield window function

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 on-board/off-board programming and self programming.

Writing to and erasing areas outside the window range are disabled during self programming. During on-board/off-board programming, however, areas outside the range specified as a window can be written and erased.

Figure25-12. Flash Shield Window Setting Example
(Target Devices: μPD78F1016, Start Block: 04H, End Block: 06H)



Caution 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.

Table 25-8. 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 set information library.	Block erasing is enabled only within the window range.	Writing is enabled only within the range of window range.
On-board/Off-board 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 **25.7 Security Settings** to prohibit writing/erasing during on-board/off-board programming.

<R> 25.9 Creating ROM Code to Place Order for Previously Written Product

Before placing an order with Renesas Electronics for a previously written product, the ROM code for the order must be created.

To create the ROM code, use the Hex Consolidation Utility (hereafter abbreviated to HCU) on the finished programs (hex files) and optional data (such as security settings for flash memory programs).

The HCU is a software tool that includes functions required for creating ROM code.

The HCU can be downloaded at the Renesas Electronics website.

(1) Website

http://www2.renesas.com/micro/en/ods/ → Click Version-up Service.

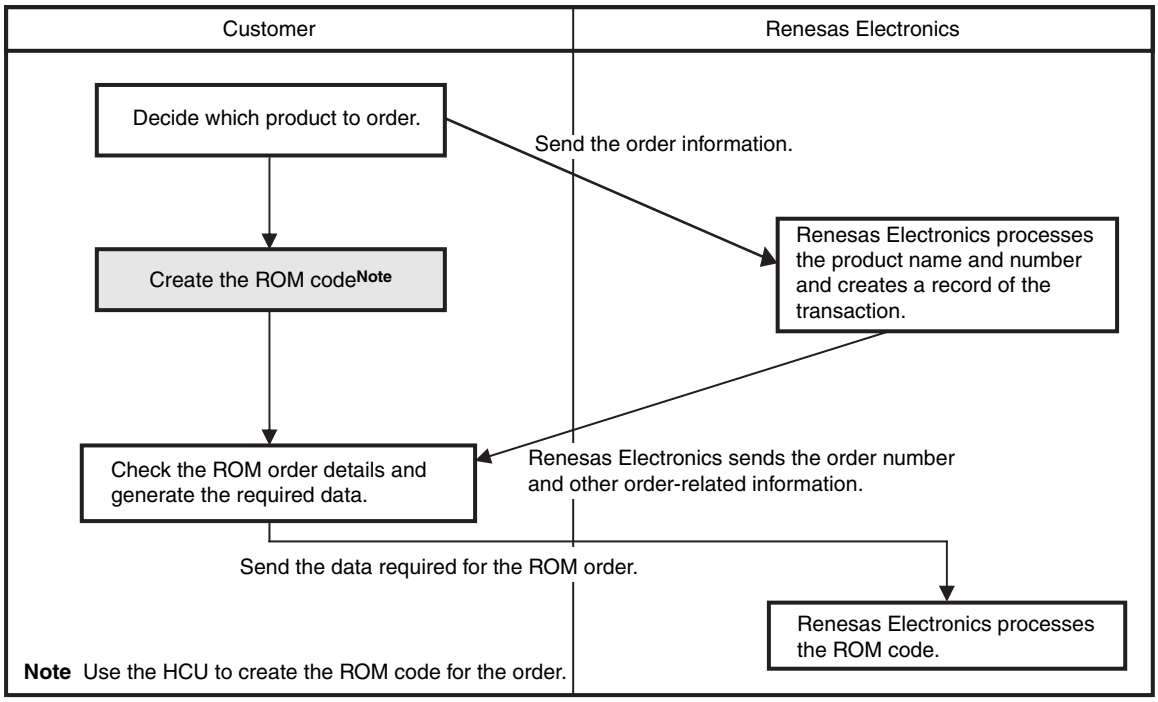
(2) Downloading the HCU

To download the HCU, click Software for previously written flash products and then HCU_GUI.

Remark For details about how to install and use the HCU, see the materials (the user's manual) that comes with the HCU at the above website.

25.9.1 Procedure for using ROM code to place an order

Use the HCU to create the ROM code by following the procedure below, and then place your order with Renesas Electronics. For details, see the ROM Code Ordering Method Information (C10302J).



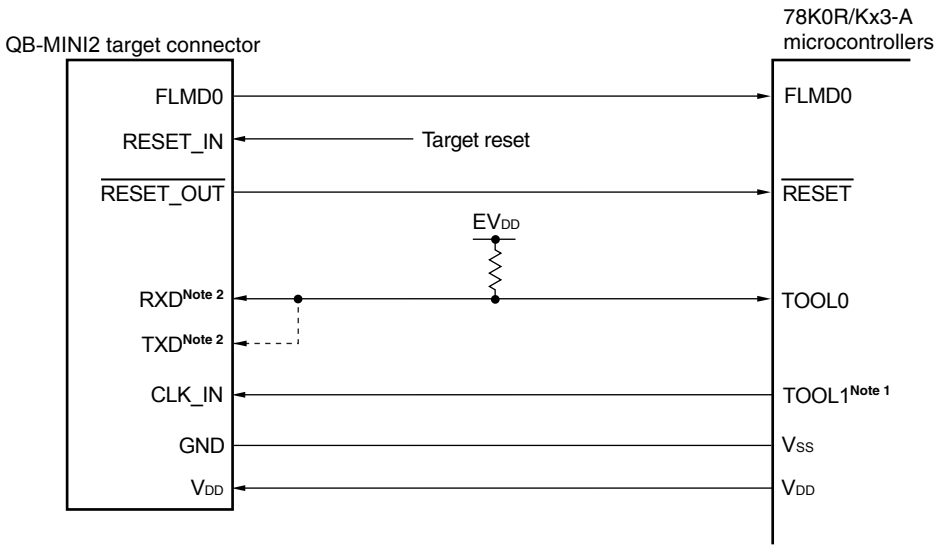
CHAPTER 26 ON-CHIP DEBUG FUNCTION

26.1 Connecting QB-MINI2 to 78K0R/Kx3-A microcontrollers

The 78K0R/Kx3-A microcontrollers use the V_{DD}, FLMD0, $\overline{RESET}$, TOOL0, TOOL1[Note 1], and V_{SS} pins to communicate with the host machine via an on-chip debug emulator (QB-MINI2).

Caution The 78K0R/Kx3-A 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.

Figure 26-1. Connection Example of QB-MINI2 and 78K0R/Kx3-A microcontrollers



Notes 1. Connection is not required for communication in 1-line mode but required for communication in 2-line mode. At this time, perform necessary connections according to Table 2-2 Connection of Unused Pins since TOOL1 is an unused pin when QB-MINI2 is unconnected.

2. Connecting the dotted line is not necessary since RXD and TXD are shorted within QB-MIN2. When using the other flash memory programmer, RXD and TXD may not be shorted within the programmer. In this case, they must be shorted on the target system.

Remark The FLMD0 pin is recommended to be open for self-programming in on-chip debugging. To pull down externally, use a resistor of 100 kΩ or more.

1-line mode (single line UART) using the TOOL0 pin or 2-line mode using the TOOL0 and TOOL1 pins is used for serial communication For flash memory programming, 1-line mode is used. 1-line mode or 2-line mode is used for on-chip debugging. Table 26-1 lists the differences between 1-line mode and 2-line mode.

Table 26-1. Lists the Differences Between 1-line Mode and 2-line Mode.

Communication mode	Flash memory programming function	Debugging function
1-line mode	Available	• Pseudo real-time RAM monitor (RRM) function not supported. • DMM function (rewriting memory in RUN) not supported. • The debugger speed is two to four times slower than 2-line mode.
2-line mode	None	• Pseudo real-time RAM monitor (RRM) function supported • DMM function (rewriting memory in RUN) supported

Remark 2-line mode is not used for flash programming, however, even if TOOL1 pin is connected with CLK_IN of QB-MINI2, writing is performed normally with no problem.

26.2 On-Chip Debug Security ID

The 78K0R/Kx3-A microcontrollers have an on-chip debug operation control bit in the flash memory at 000C3H (see **CHAPTER 24 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.

For details on the on-chip debug security ID, refer to the **QB-MINI2 On-Chip Debug Emulator with Programming Function User's Manual (U18371E)**.

Table 26-2. On-Chip Debug Security ID

Address	On-Chip Debug Security ID
000C4H to 000CDH	Any ID code of 10 bytes
010C4H to 010CDH	

26.3 Securing of User Resources

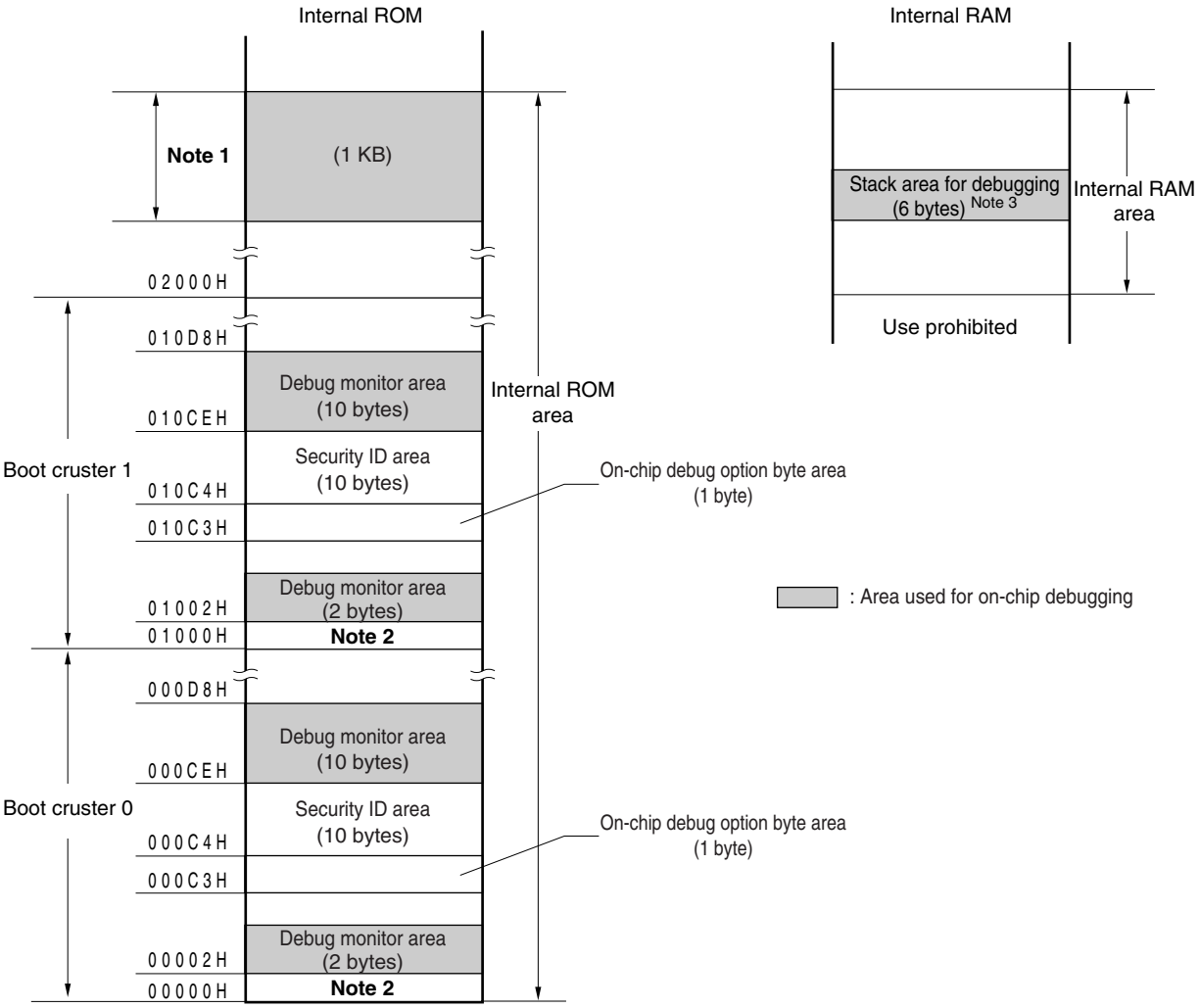
To perform communication between the 78K0R/Kx3-A microcontrollers and QB-MINI2, as well as each debug function, the securing of memory space must be done beforehand.

If Renesas Electronics assembler RA78K0R or compiler CC78K0R is used, the items can be set by using linker options.

(1) Securement of memory space

The shaded portions in Figure 26-2 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 26-2. Memory Spaces Where Debug Monitor Programs Are Allocated



Notes 1. Address differs depending on products as follows.

Products	Internal ROM	Address
μPD78F1016	64 KB	0FC00H to 0FFFFH
μPD78F1017	96 KB	17C00H to 17FFFH
μPD78F1018	128 KB	1FC00H to 1FFFFH

2. In debugging, reset vector is rewritten to address allocated to a monitor program.

3. Since this area is allocated immediately before the stack area, the address of this area varies depending on the stack increase and decrease. That is, 6 extra bytes are consumed for the stack area used.

For details of the way to secure of the memory space, refer to the **QB-MINI2 On-Chip Debug Emulator with Programming Function User's Manual (U18371E)**.

CHAPTER 27 BCD CORRECTION CIRCUIT

27.1 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 BCDADJ register.

27.2 Registers Used by BCD Correction Circuit

The BCD correction circuit uses the following registers.

- BCD correction result register (BCDADJ)

(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.

BCDADJ is read by an 8-bit memory manipulation instruction.

Reset input sets this register to undefined.

Figure 27-1. Format of BCD Correction Result Register (BCDADJ)

Address: F00FEH After reset: undefined R

Symbol	7	6	5	4	3	2	1	0
BCDADJ								

27.3 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 BCDADJ register.

<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 register.

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 Register	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 Register	AC Flag	BCDADJ Register
`MOV A, #85H ; <1>`	85H	–	–	–
<R> `ADD A, #15H ; <2>`	9AH	0	0	66H
`ADD A, !BCDADJ ; <3>`	00H	1	1	–

Examples 3: 80 + 80 = 160

Instruction	A Register	CY Register	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 BCDADJ register.

<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 register.

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 Register	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 28 INSTRUCTION SET

This chapter lists the instructions in the 78K0R microcontroller instruction set. For details of each operation and operation code, refer to the separate document **78K0R Microcontrollers Instructions User's Manual (U17792E)**.

Remark The shaded parts of the tables in **Table 28-5 Operation List** indicate the operation or instruction format that is newly added for the 78K0R microcontrollers.

28.1 Conventions Used in Operation List

28.1.1 Operand identifiers and specification methods

Operands are described in the "Operand" column of each instruction in accordance with the description method of the instruction operand identifier (refer to the assembler specifications for details). When there are two or more description methods, select one of them. Alphabetic letters in capitals and the symbols, #, !, !!, $, $!, [ ], and ES: are keywords and are described as they are. Each symbol has the following meaning.

- #: Immediate data specification
- !: 16-bit absolute address specification
- !!: 20-bit absolute address specification
- $: 8-bit relative address specification
- $!: 16-bit relative address specification
- [ ]: Indirect address specification
- ES:: Extension address specification

In the case of immediate data, describe an appropriate numeric value or a label. When using a label, be sure to describe the #, !, !!, $, $!, [ ], and ES: symbols.

For operand register identifiers, r and rp, either function names (X, A, C, etc.) or absolute names (names in parentheses in the table below, R0, R1, R2, etc.) can be used for description.

Table 28-1. Operand Identifiers and Specification Methods

Identifier	Description Method
r	X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7)
rp	AX (RP0), BC (RP1), DE (RP2), HL (RP3)
sfr	Special-function register symbol (SFR symbol)
sfrp	Special-function register symbols (16-bit manipulatable SFR symbol. Even addresses only[Note])
saddr	FFE20H to FFF1FH Immediate data or labels
saddrp	FFE20H to FF1FH Immediate data or labels (even addresses only[Note])
addr20	00000H to FFFFFH Immediate data or labels
addr16	0000H to FFFFH Immediate data or labels (only even addresses for 16-bit data transfer instructions[Note])
addr5	0080H to 00BFH Immediate data or labels (even addresses only)
word	16-bit immediate data or label
byte	8-bit immediate data or label
bit	3-bit immediate data or label
RBn	RB0 to RB3

Note Bit 0 = 0 when an odd address is specified.

28.1.2 Description of operation column

The operation when the instruction is executed is shown in the “Operation” column using the following symbols.

Table 28-2. Symbols in “Operation” Column

Symbol	Function
A	A register; 8-bit accumulator
X	X register
B	B register
C	C register
D	D register
E	E register
H	H register
L	L register
ES	ES register
CS	CS register
AX	AX register pair; 16-bit accumulator
BC	BC register pair
DE	DE register pair
HL	HL register pair
PC	Program counter
SP	Stack pointer
PSW	Program status word
CY	Carry flag
AC	Auxiliary carry flag
Z	Zero flag
RBS	Register bank select flag
IE	Interrupt request enable flag
()	Memory contents indicated by address or register contents in parentheses
X_H, X_L X_S, X_H, X_L	16-bit registers: X_H = higher 8 bits, X_L = lower 8 bits 20-bit registers: X_S = (bits 19 to 16), X_H = (bits 15 to 8), X_L = (bits 7 to 0)
∧	Logical product (AND)
∨	Logical sum (OR)
∨̲	Exclusive logical sum (exclusive OR)
¯	Inverted data
addr5	16-bit immediate data (even addresses only in 0080H to 00BFH)
addr16	16-bit immediate data
addr20	20-bit immediate data
jdisp8	Signed 8-bit data (displacement value)
jdisp16	Signed 16-bit data (displacement value)

28.1.3 Description of flag operation column

The change of the flag value when the instruction is executed is shown in the “Flag” column using the following symbols.

Table 28-3. Symbols in “Flag” Column

Symbol	Change of Flag Value
(Blank)	Unchanged
0	Cleared to 0
1	Set to 1
×	Set/cleared according to the result
R	Previously saved value is restored

28.1.4 PREFIX instruction

Instructions with “ES:” have a PREFIX operation code as a prefix to extend the accessible data area to the 1 MB space (00000H to FFFFFH), by adding the ES register value to the 64 KB space from F0000H to FFFFFH. When a PREFIX operation code is attached as a prefix to the target instruction, only one instruction immediately after the PREFIX operation code is executed as the addresses with the ES register value added.

Table 28-4. Use Example of PREFIX Operation Code

Instruction	Opcode				
	1	2	3	4	5
MOV !addr16, #byte	CFH	!addr16		#byte	–
MOV ES:!addr16, #byte	11H	CFH	!addr16		#byte
MOV A, [HL]	8BH	–	–	–	–
MOV A, ES:[HL]	11H	8BH	–	–	–

Caution Set the ES register value with MOV ES, A, etc., before executing the PREFIX instruction.

28.2 Operation List

Table 28-5. Operation List (1/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	MOV	r, #byte	2	1	–	r ← byte			
		saddr, #byte	3	1	–	(saddr) ← byte			
		sfr, #byte	3	1	–	sfr ← byte			
		!addr16, #byte	4	1	–	(addr16) ← byte			
		A, r Note 3	1	1	–	A ← r			
		r, A Note 3	1	1	–	r ← A			
		A, saddr	2	1	–	A ← (saddr)			
		saddr, A	2	1	–	(saddr) ← A			
		A, sfr	2	1	–	A ← sfr			
		sfr, A	2	1	–	sfr ← A			
		A, !addr16	3	1	4	A ← (addr16)			
		!addr16, A	3	1	–	(addr16) ← A			
		PSW, #byte	3	3	–	PSW ← byte	×	×	×
		A, PSW	2	1	–	A ← PSW			
		PSW, A	2	3	–	PSW ← A	×	×	×
		ES, #byte	2	1	–	ES ← byte			
		ES, saddr	3	1	–	ES ← (saddr)			
		A, ES	2	1	–	A ← ES			
		ES, A	2	1	–	ES ← A			
		CS, #byte	3	1	–	CS ← byte			
		A, CS	2	1	–	A ← CS			
		CS, A	2	1	–	CS ← A			
		A, [DE]	1	1	4	A ← (DE)			
		[DE], A	1	1	–	(DE) ← A			
		[DE + byte], #byte	3	1	–	(DE + byte) ← byte			
		A, [DE + byte]	2	1	4	A ← (DE + byte)			
		[DE + byte], A	2	1	–	(DE + byte) ← A			
		A, [HL]	1	1	4	A ← (HL)			
		[HL], A	1	1	–	(HL) ← A			
		[HL + byte], #byte	3	1	–	(HL + byte) ← byte			

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 28-5. Operation List (2/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	MOV	A, [HL + byte]	2	1	4	A ← (HL + byte)	
		[HL + byte], A	2	1	–	(HL + byte) ← A	
		A, [HL + B]	2	1	4	A ← (HL + B)	
		[HL + B], A	2	1	–	(HL + B) ← A	
		A, [HL + C]	2	1	4	A ← (HL + C)	
		[HL + C], A	2	1	–	(HL + C) ← A	
		word[B], #byte	4	1	–	(B + word) ← byte	
		A, word[B]	3	1	4	A ← (B + word)	
		word[B], A	3	1	–	(B + word) ← A	
		word[C], #byte	4	1	–	(C + word) ← byte	
		A, word[C]	3	1	4	A ← (C + word)	
		word[C], A	3	1	–	(C + word) ← A	
		word[BC], #byte	4	1	–	(BC + word) ← byte	
		A, word[BC]	3	1	4	A ← (BC + word)	
		word[BC], A	3	1	–	(BC + word) ← A	
		[SP + byte], #byte	3	1	–	(SP + byte) ← byte	
		A, [SP + byte]	2	1	–	A ← (SP + byte)	
		[SP + byte], A	2	1	–	(SP + byte) ← A	
		B, saddr	2	1	–	B ← (saddr)	
		B, !addr16	3	1	4	B ← (addr16)	
		C, saddr	2	1	–	C ← (saddr)	
		C, !addr16	3	1	4	C ← (addr16)	
		X, saddr	2	1	–	X ← (saddr)	
		X, !addr16	3	1	4	X ← (addr16)	
		ES:!addr16, #byte	5	2	–	(ES, addr16) ← byte	
		A, ES:!addr16	4	2	5	A ← (ES, addr16)	
		ES:!addr16, A	4	2	–	(ES, addr16) ← A	
		A, ES:[DE]	2	2	5	A ← (ES, DE)	
		ES:[DE], A	2	2	–	(ES, DE) ← A	
		ES:[DE + byte],#byte	4	2	–	((ES, DE) + byte) ← byte	
		A, ES:[DE + byte]	3	2	5	A ← ((ES, DE) + byte)	
		ES:[DE + byte], A	3	2	–	((ES, DE) + byte) ← A	

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 28-5. Operation List (3/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit data transfer	MOV	A, ES:[HL]	2	2	5	A ← (ES, HL)	
		ES:[HL], A	2	2	–	(ES, HL) ← A	
		ES:[HL + byte],#byte	4	2	–	((ES, HL) + byte) ← byte	
		A, ES:[HL + byte]	3	2	5	A ← ((ES, HL) + byte)	
		ES:[HL + byte], A	3	2	–	((ES, HL) + byte) ← A	
		A, ES:[HL + B]	3	2	5	A ← ((ES, HL) + B)	
		ES:[HL + B], A	3	2	–	((ES, HL) + B) ← A	
		A, ES:[HL + C]	3	2	5	A ← ((ES, HL) + C)	
		ES:[HL + C], A	3	2	–	((ES, HL) + C) ← A	
		ES:word[B], #byte	5	2	–	((ES, B) + word) ← byte	
		A, ES:word[B]	4	2	5	A ← ((ES, B) + word)	
		ES:word[B], A	4	2	–	((ES, B) + word) ← A	
		ES:word[C], #byte	5	2	–	((ES, C) + word) ← byte	
		A, ES:word[C]	4	2	5	A ← ((ES, C) + word)	
		ES:word[C], A	4	2	–	((ES, C) + word) ← A	
		ES:word[BC], #byte	5	2	–	((ES, BC) + word) ← byte	
		A, ES:word[BC]	4	2	5	A ← ((ES, BC) + word)	
		ES:word[BC], A	4	2	–	((ES, BC) + word) ← A	
		B, ES:!addr16	4	2	5	B ← (ES, addr16)	
		C, ES:!addr16	4	2	5	C ← (ES, addr16)	
		X, ES:!addr16	4	2	5	X ← (ES, addr16)	
	XCH	A, r Note 3	1 (r = X) 2 (other than r = X)	1	–	A ←→ r	
		A, saddr	3	2	–	A ←→ (saddr)	
		A, sfr	3	2	–	A ←→ sfr	
		A, !addr16	4	2	–	A ←→ (addr16)	
		A, [DE]	2	2	–	A ←→ (DE)	
		A, [DE + byte]	3	2	–	A ←→ (DE + byte)	
		A, [HL]	2	2	–	A ←→ (HL)	
		A, [HL + byte]	3	2	–	A ←→ (HL + byte)	
		A, [HL + B]	2	2	–	A ←→ (HL + B)	
		A, [HL + C]	2	2	–	A ←→ (HL + C)	

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 28-5. Operation List (4/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit data transfer	XCH	A, ES:!addr16	5	3	–	A ←→ (ES, addr16)			
		A, ES:[DE]	3	3	–	A ←→ (ES, DE)			
		A, ES:[DE + byte]	4	3	–	A ←→ ((ES, DE) + byte)			
		A, ES:[HL]	3	3	–	A ←→ (ES, HL)			
		A, ES:[HL + byte]	4	3	–	A ←→ ((ES, HL) + byte)			
		A, ES:[HL + B]	3	3	–	A ←→ ((ES, HL) + B)			
		A, ES:[HL + C]	3	3	–	A ←→ ((ES, HL) + C)			
	ONEB	A	1	1	–	A ← 01H			
		X	1	1	–	X ← 01H			
		B	1	1	–	B ← 01H			
		C	1	1	–	C ← 01H			
		saddr	2	1	–	(saddr) ← 01H			
		!addr16	3	1	–	(addr16) ← 01H			
		ES:!addr16	4	2	–	(ES, addr16) ← 01H			
	CLRB	A	1	1	–	A ← 00H			
		X	1	1	–	X ← 00H			
		B	1	1	–	B ← 00H			
		C	1	1	–	C ← 00H			
		saddr	2	1	–	(saddr) ← 00H			
		!addr16	3	1	–	(addr16) ← 00H			
		ES:!addr16	4	2	–	(ES,addr16) ← 00H			
	MOVS	[HL + byte], X	3	1	–	(HL + byte) ← X	×		×
		ES:[HL + byte], X	4	2	–	(ES, HL + byte) ← X	×		×
16-bit data transfer	MOVW	rp, #word	3	1	–	rp ← word			
		saddrp, #word	4	1	–	(saddrp) ← word			
		sfrp, #word	4	1	–	sfrp ← word			
		AX, saddrp	2	1	–	AX ← (saddrp)			
		saddrp, AX	2	1	–	(saddrp) ← AX			
		AX, sfrp	2	1	–	AX ← sfrp			
		sfrp, AX	2	1	–	sfrp ← AX			
		AX, rp Note 3	1	1	–	AX ← rp			
		rp, AX Note 3	1	1	–	rp ← AX			

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except rp = AX

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 28-5. Operation List (5/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
16-bit data transfer	MOVW	AX, !addr16	3	1	4	AX ← (addr16)	
		!addr16, AX	3	1	–	(addr16) ← AX	
		AX, [DE]	1	1	4	AX ← (DE)	
		[DE], AX	1	1	–	(DE) ← AX	
		AX, [DE + byte]	2	1	4	AX ← (DE + byte)	
		[DE + byte], AX	2	1	–	(DE + byte) ← AX	
		AX, [HL]	1	1	4	AX ← (HL)	
		[HL], AX	1	1	–	(HL) ← AX	
		AX, [HL + byte]	2	1	4	AX ← (HL + byte)	
		[HL + byte], AX	2	1	–	(HL + byte) ← AX	
		AX, word[B]	3	1	4	AX ← (B + word)	
		word[B], AX	3	1	–	(B + word) ← AX	
		AX, word[C]	3	1	4	AX ← (C + word)	
		word[C], AX	3	1	–	(C + word) ← AX	
		AX, word[BC]	3	1	4	AX ← (BC + word)	
		word[BC], AX	3	1	–	(BC + word) ← AX	
		AX, [SP + byte]	2	1	–	AX ← (SP + byte)	
		[SP + byte], AX	2	1	–	(SP + byte) ← AX	
		BC, saddrp	2	1	–	BC ← (saddrp)	
		BC, !addr16	3	1	4	BC ← (addr16)	
		DE, saddrp	2	1	–	DE ← (saddrp)	
		DE, !addr16	3	1	4	DE ← (addr16)	
		HL, saddrp	2	1	–	HL ← (saddrp)	
		HL, !addr16	3	1	4	HL ← (addr16)	
		AX, ES:!addr16	4	2	5	AX ← (ES, addr16)	
		ES:!addr16, AX	4	2	–	(ES, addr16) ← AX	
		AX, ES:[DE]	2	2	5	AX ← (ES, DE)	
		ES:[DE], AX	2	2	–	(ES, DE) ← AX	
		AX, ES:[DE + byte]	3	2	5	AX ← ((ES, DE) + byte)	
		ES:[DE + byte], AX	3	2	–	((ES, DE) + byte) ← AX	
		AX, ES:[HL]	2	2	5	AX ← (ES, HL)	
		ES:[HL], AX	2	2	–	(ES, HL) ← AX	

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 28-5. Operation List (6/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit data transfer	MOVW	AX, ES:[HL + byte]	3	2	5	AX ← ((ES, HL) + byte)			
		ES:[HL + byte], AX	3	2	–	((ES, HL) + byte) ← AX			
		AX, ES:word[B]	4	2	5	AX ← ((ES, B) + word)			
		ES:word[B], AX	4	2	–	((ES, B) + word) ← AX			
		AX, ES:word[C]	4	2	5	AX ← ((ES, C) + word)			
		ES:word[C], AX	4	2	–	((ES, C) + word) ← AX			
		AX, ES:word[BC]	4	2	5	AX ← ((ES, BC) + word)			
		ES:word[BC], AX	4	2	–	((ES, BC) + word) ← AX			
		BC, ES:!addr16	4	2	5	BC ← (ES, addr16)			
		DE, ES:!addr16	4	2	5	DE ← (ES, addr16)			
		HL, ES:!addr16	4	2	5	HL ← (ES, addr16)			
	XCHW	AX, rp Note 3	1	1	–	AX ←→ rp			
	ONEW	AX	1	1	–	AX ← 0001H			
		BC	1	1	–	BC ← 0001H			
	CLRW	AX	1	1	–	AX ← 0000H			
		BC	1	1	–	BC ← 0000H			
8-bit operation	ADD	A, #byte	2	1	–	A, CY ← A + byte	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) + byte	×	×	×
		A, r Note 4	2	1	–	A, CY ← A + r	×	×	×
		r, A	2	1	–	r, CY ← r + A	×	×	×
		A, saddr	2	1	–	A, CY ← A + (saddr)	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16)	×	×	×
		A, [HL]	1	1	4	A, CY ← A + (HL)	×	×	×
		A, [HL + byte]	2	1	4	A, CY ← A + (HL + byte)	×	×	×
		A, [HL + B]	2	1	4	A, CY ← A + (HL + B)	×	×	×
		A, [HL + C]	2	1	4	A, CY ← A + (HL + C)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16)	×	×	×
		A, ES:[HL]	2	2	5	A,CY ← A + (ES, HL)	×	×	×
		A, ES:[HL + byte]	3	2	5	A,CY ← A + ((ES, HL) + byte)	×	×	×
		A, ES:[HL + B]	3	2	5	A,CY ← A + ((ES, HL) + B)	×	×	×
		A, ES:[HL + C]	3	2	5	A,CY ← A + ((ES, HL) + C)	×	×	×

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except rp = AX

4. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 28-5. Operation List (7/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	ADDC	A, #byte	2	1	–	A, CY ← A + byte + CY	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) + byte + CY	×	×	×
		A, r Note 3	2	1	–	A, CY ← A + r + CY	×	×	×
		r, A	2	1	–	r, CY ← r + A + CY	×	×	×
		A, saddr	2	1	–	A, CY ← A + (saddr) + CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A + (addr16) + CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A + (HL) + CY	×	×	×
		A, [HL + byte]	2	1	4	A, CY ← A + (HL + byte) + CY	×	×	×
		A, [HL + B]	2	1	4	A, CY ← A + (HL + B) + CY	×	×	×
		A, [HL + C]	2	1	4	A, CY ← A + (HL + C) + CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A + (ES, addr16) + CY	×	×	×
		A, ES:[HL]	2	2	5	A, CY ← A + (ES, HL) + CY	×	×	×
		A, ES:[HL + byte]	3	2	5	A, CY ← A + ((ES, HL) + byte) + CY	×	×	×
		A, ES:[HL + B]	3	2	5	A, CY ← A + ((ES, HL) + B) + CY	×	×	×
		A, ES:[HL + C]	3	2	5	A, CY ← A + ((ES, HL) + C) + CY	×	×	×
	SUB	A, #byte	2	1	–	A, CY ← A – byte	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) – byte	×	×	×
		A, r Note 3	2	1	–	A, CY ← A – r	×	×	×
		r, A	2	1	–	r, CY ← r – A	×	×	×
		A, saddr	2	1	–	A, CY ← A – (saddr)	×	×	×
		A, !addr16	3	1	4	A, CY ← A – (addr16)	×	×	×
		A, [HL]	1	1	4	A, CY ← A – (HL)	×	×	×
		A, [HL + byte]	2	1	4	A, CY ← A – (HL + byte)	×	×	×
		A, [HL + B]	2	1	4	A, CY ← A – (HL + B)	×	×	×
		A, [HL + C]	2	1	4	A, CY ← A – (HL + C)	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A – (ES:addr16)	×	×	×
		A, ES:[HL]	2	2	5	A, CY ← A – (ES:HL)	×	×	×
		A, ES:[HL + byte]	3	2	5	A, CY ← A – ((ES:HL) + byte)	×	×	×
		A, ES:[HL + B]	3	2	5	A, CY ← A – ((ES:HL) + B)	×	×	×
		A, ES:[HL + C]	3	2	5	A, CY ← A – ((ES:HL) + C)	×	×	×

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 28-5. Operation List (8/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	SUBC	A, #byte	2	1	–	A, CY ← A – byte – CY	×	×	×
		saddr, #byte	3	2	–	(saddr), CY ← (saddr) – byte – CY	×	×	×
		A, r Note 3	2	1	–	A, CY ← A – r – CY	×	×	×
		r, A	2	1	–	r, CY ← r – A – CY	×	×	×
		A, saddr	2	1	–	A, CY ← A – (saddr) – CY	×	×	×
		A, !addr16	3	1	4	A, CY ← A – (addr16) – CY	×	×	×
		A, [HL]	1	1	4	A, CY ← A – (HL) – CY	×	×	×
		A, [HL + byte]	2	1	4	A, CY ← A – (HL + byte) – CY	×	×	×
		A, [HL + B]	2	1	4	A, CY ← A – (HL + B) – CY	×	×	×
		A, [HL + C]	2	1	4	A, CY ← A – (HL + C) – CY	×	×	×
		A, ES:!addr16	4	2	5	A, CY ← A – (ES:addr16) – CY	×	×	×
		A, ES:[HL]	2	2	5	A, CY ← A – (ES:HL) – CY	×	×	×
		A, ES:[HL + byte]	3	2	5	A, CY ← A – ((ES:HL) + byte) – CY	×	×	×
		A, ES:[HL + B]	3	2	5	A, CY ← A – ((ES:HL) + B) – CY	×	×	×
		A, ES:[HL + C]	3	2	5	A, CY ← A – ((ES:HL) + C) – CY	×	×	×
	AND	A, #byte	2	1	–	A ← A ∧ byte	×		
		saddr, #byte	3	2	–	(saddr) ← (saddr) ∧ byte	×		
		A, r Note 3	2	1	–	A ← A ∧ r	×		
		r, A	2	1	–	r ← r ∧ A	×		
		A, saddr	2	1	–	A ← A ∧ (saddr)	×		
		A, !addr16	3	1	4	A ← A ∧ (addr16)	×		
		A, [HL]	1	1	4	A ← A ∧ (HL)	×		
		A, [HL + byte]	2	1	4	A ← A ∧ (HL + byte)	×		
		A, [HL + B]	2	1	4	A ← A ∧ (HL + B)	×		
		A, [HL + C]	2	1	4	A ← A ∧ (HL + C)	×		
		A, ES:!addr16	4	2	5	A ← A ∧ (ES:addr16)	×		
		A, ES:[HL]	2	2	5	A ← A ∧ (ES:HL)	×		
		A, ES:[HL + byte]	3	2	5	A ← A ∧ ((ES:HL) + byte)	×		
		A, ES:[HL + B]	3	2	5	A ← A ∧ ((ES:HL) + B)	×		
		A, ES:[HL + C]	3	2	5	A ← A ∧ ((ES:HL) + C)	×		

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 28-5. Operation List (9/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
8-bit operation	OR	A, #byte	2	1	–	A ← A ∨ byte	×
		saddr, #byte	3	2	–	(saddr) ← (saddr) ∨ byte	×
		A, r Note 3	2	1	–	A ← A ∨ r	×
		r, A	2	1	–	r ← r ∨ A	×
		A, saddr	2	1	–	A ← A ∨ (saddr)	×
		A, !addr16	3	1	4	A ← A ∨ (addr16)	×
		A, [HL]	1	1	4	A ← A ∨ (HL)	×
		A, [HL + byte]	2	1	4	A ← A ∨ (HL + byte)	×
		A, [HL + B]	2	1	4	A ← A ∨ (HL + B)	×
		A, [HL + C]	2	1	4	A ← A ∨ (HL + C)	×
		A, ES:!addr16	4	2	5	A ← A ∨ (ES:addr16)	×
		A, ES:[HL]	2	2	5	A ← A ∨ (ES:HL)	×
		A, ES:[HL + byte]	3	2	5	A ← A ∨ ((ES:HL) + byte)	×
		A, ES:[HL + B]	3	2	5	A ← A ∨ ((ES:HL) + B)	×
		A, ES:[HL + C]	3	2	5	A ← A ∨ ((ES:HL) + C)	×
	XOR	A, #byte	2	1	–	A ← A ⊻ byte	×
		saddr, #byte	3	2	–	(saddr) ← (saddr) ⊻ byte	×
		A, r Note 3	2	1	–	A ← A ⊻ r	×
		r, A	2	1	–	r ← r ⊻ A	×
		A, saddr	2	1	–	A ← A ⊻ (saddr)	×
		A, !addr16	3	1	4	A ← A ⊻ (addr16)	×
		A, [HL]	1	1	4	A ← A ⊻ (HL)	×
		A, [HL + byte]	2	1	4	A ← A ⊻ (HL + byte)	×
		A, [HL + B]	2	1	4	A ← A ⊻ (HL + B)	×
		A, [HL + C]	2	1	4	A ← A ⊻ (HL + C)	×
		A, ES:!addr16	4	2	5	A ← A ⊻ (ES:addr16)	×
		A, ES:[HL]	2	2	5	A ← A ⊻ (ES:HL)	×
		A, ES:[HL + byte]	3	2	5	A ← A ⊻ ((ES:HL) + byte)	×
		A, ES:[HL + B]	3	2	5	A ← A ⊻ ((ES:HL) + B)	×
		A, ES:[HL + C]	3	2	5	A ← A ⊻ ((ES:HL) + C)	×

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 28-5. Operation List (10/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
8-bit operation	CMP	A, #byte	2	1	–	A – byte	×	×	×
		saddr, #byte	3	1	–	(saddr) – byte	×	×	×
		A, r Note 3	2	1	–	A – r	×	×	×
		r, A	2	1	–	r – A	×	×	×
		A, saddr	2	1	–	A – (saddr)	×	×	×
		A, !addr16	3	1	4	A – (addr16)	×	×	×
		A, [HL]	1	1	4	A – (HL)	×	×	×
		A, [HL + byte]	2	1	4	A – (HL + byte)	×	×	×
		A, [HL + B]	2	1	4	A – (HL + B)	×	×	×
		A, [HL + C]	2	1	4	A – (HL + C)	×	×	×
		!addr16, #byte	4	1	4	(addr16) – byte	×	×	×
		A, ES:!addr16	4	2	5	A – (ES:addr16)	×	×	×
		A, ES:[HL]	2	2	5	A – (ES:HL)	×	×	×
		A, ES:[HL + byte]	3	2	5	A – ((ES:HL) + byte)	×	×	×
		A, ES:[HL + B]	3	2	5	A – ((ES:HL) + B)	×	×	×
		A, ES:[HL + C]	3	2	5	A – ((ES:HL) + C)	×	×	×
		ES:!addr16, #byte	5	2	5	(ES:addr16) – byte	×	×	×
	CMP0	A	1	1	–	A – 00H	×	×	×
		X	1	1	–	X – 00H	×	×	×
		B	1	1	–	B – 00H	×	×	×
		C	1	1	–	C – 00H	×	×	×
		saddr	2	1	–	(saddr) – 00H	×	×	×
		!addr16	3	1	4	(addr16) – 00H	×	×	×
		ES:!addr16	4	2	5	(ES:addr16) – 00H	×	×	×
	CMPS	X, [HL + byte]	3	1	4	X – (HL + byte)	×	×	×
		X, ES:[HL + byte]	4	2	5	X – ((ES:HL) + byte)	×	×	×

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. Except r = A

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 28-5. Operation List (11/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
16-bit operation	ADDW	AX, #word	3	1	–	AX, CY ← AX + word	×	×	×
		AX, AX	1	1	–	AX, CY ← AX + AX	×	×	×
		AX, BC	1	1	–	AX, CY ← AX + BC	×	×	×
		AX, DE	1	1	–	AX, CY ← AX + DE	×	×	×
		AX, HL	1	1	–	AX, CY ← AX + HL	×	×	×
		AX, saddrp	2	1	–	AX, CY ← AX + (saddrp)	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX + (addr16)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX + (HL + byte)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX + (ES:addr16)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX + ((ES:HL) + byte)	×	×	×
	SUBW	AX, #word	3	1	–	AX, CY ← AX – word	×	×	×
		AX, BC	1	1	–	AX, CY ← AX – BC	×	×	×
		AX, DE	1	1	–	AX, CY ← AX – DE	×	×	×
		AX, HL	1	1	–	AX, CY ← AX – HL	×	×	×
		AX, saddrp	2	1	–	AX, CY ← AX – (saddrp)	×	×	×
		AX, !addr16	3	1	4	AX, CY ← AX – (addr16)	×	×	×
		AX, [HL+byte]	3	1	4	AX, CY ← AX – (HL + byte)	×	×	×
		AX, ES:!addr16	4	2	5	AX, CY ← AX – (ES:addr16)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX, CY ← AX – ((ES:HL) + byte)	×	×	×
	CMPW	AX, #word	3	1	–	AX – word	×	×	×
		AX, BC	1	1	–	AX – BC	×	×	×
		AX, DE	1	1	–	AX – DE	×	×	×
		AX, HL	1	1	–	AX – HL	×	×	×
		AX, saddrp	2	1	–	AX – (saddrp)	×	×	×
		AX, !addr16	3	1	4	AX – (addr16)	×	×	×
		AX, [HL+byte]	3	1	4	AX – (HL + byte)	×	×	×
		AX, ES:!addr16	4	2	5	AX – (ES:addr16)	×	×	×
		AX, ES: [HL+byte]	4	2	5	AX – ((ES:HL) + byte)	×	×	×
Multiply	MULU	X	1	1	–	AX ← A × X			

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 28-5. Operation List (12/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Increment/ decrement	INC	r	1	1	–	r ← r + 1	×	×	
		saddr	2	2	–	(saddr) ← (saddr) + 1	×	×	
		!addr16	3	2	–	(addr16) ← (addr16) + 1	×	×	
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) + 1	×	×	
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) + 1	×	×	
		ES: [HL+byte]	4	3	–	((ES:HL) + byte) ← ((ES:HL) + byte) + 1	×	×	
	DEC	r	1	1	–	r ← r – 1	×	×	
		saddr	2	2	–	(saddr) ← (saddr) – 1	×	×	
		!addr16	3	2	–	(addr16) ← (addr16) – 1	×	×	
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) – 1	×	×	
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) – 1	×	×	
		ES: [HL+byte]	4	3	–	((ES:HL) + byte) ← ((ES:HL) + byte) – 1	×	×	
	INCW	rp	1	1	–	rp ← rp + 1			
		saddrp	2	2	–	(saddrp) ← (saddrp) + 1			
		!addr16	3	2	–	(addr16) ← (addr16) + 1			
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) + 1			
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) + 1			
		ES: [HL+byte]	4	3	–	((ES:HL) + byte) ← ((ES:HL) + byte) + 1			
	DECW	rp	1	1	–	rp ← rp – 1			
		saddrp	2	2	–	(saddrp) ← (saddrp) – 1			
		!addr16	3	2	–	(addr16) ← (addr16) – 1			
		[HL+byte]	3	2	–	(HL+byte) ← (HL+byte) – 1			
		ES:!addr16	4	3	–	(ES, addr16) ← (ES, addr16) – 1			
		ES: [HL+byte]	4	3	–	((ES:HL) + byte) ← ((ES:HL) + byte) – 1			
Shift	SHR	A, cnt	2	1	–	(CY ← A_0, A_{m-1} ← A_m, A_7 ← 0) × cnt			×
	SHRW	AX, cnt	2	1	–	(CY ← AX_0, AX_{m-1} ← AX_m, AX_{15} ← 0) × cnt			×
	SHL	A, cnt	2	1	–	(CY ← A_7, A_m ← A_{m-1}, A_0 ← 0) × cnt			×
		B, cnt	2	1	–	(CY ← B_7, B_m ← B_{m-1}, B_0 ← 0) × cnt			×
		C, cnt	2	1	–	(CY ← C_7, C_m ← C_{m-1}, C_0 ← 0) × cnt			×
	SHLW	AX, cnt	2	1	–	(CY ← AX_{15}, AX_m ← AX_{m-1}, AX_0 ← 0) × cnt			×
		BC, cnt	2	1	–	(CY ← BC_{15}, BC_m ← BC_{m-1}, BC_0 ← 0) × cnt			×
	SAR	A, cnt	2	1	–	(CY ← A_0, A_{m-1} ← A_m, A_7 ← A_7) × cnt			×
	SARW	AX, cnt	2	1	–	(CY ← AX_0, AX_{m-1} ← AX_m, AX_{15} ← AX_{15}) × cnt			×

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

3. cnt indicates the bit shift count.

Table 28-5. Operation List (13/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Rotate	ROR	A, 1	2	1	–	(CY, $A_7 \leftarrow A_0$, $A_{m-1} \leftarrow A_m$) × 1			×
	ROL	A, 1	2	1	–	(CY, $A_0 \leftarrow A_7$, $A_{m+1} \leftarrow A_m$) × 1			×
	RORC	A, 1	2	1	–	(CY $\leftarrow A_0$, $A_7 \leftarrow$ CY, $A_{m-1} \leftarrow A_m$) × 1			×
	ROLC	A, 1	2	1	–	(CY $\leftarrow A_7$, $A_0 \leftarrow$ CY, $A_{m+1} \leftarrow A_m$) × 1			×
	ROLWC	AX,1	2	1	–	(CY $\leftarrow AX_{15}$, $AX_0 \leftarrow$ CY, $AX_{m+1} \leftarrow AX_m$) × 1			×
		BC,1	2	1	–	(CY $\leftarrow BC_{15}$, $BC_0 \leftarrow$ CY, $BC_{m+1} \leftarrow BC_m$) × 1			×
Bit manipulate	MOV1	CY, saddr.bit	3	1	–	CY ← (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← sfr.bit			×
		CY, A.bit	2	1	–	CY ← A.bit			×
		CY, PSW.bit	3	1	–	CY ← PSW.bit			×
		CY,[HL].bit	2	1	4	CY ← (HL).bit			×
		saddr.bit, CY	3	2	–	(saddr).bit ← CY			
		sfr.bit, CY	3	2	–	sfr.bit ← CY			
		A.bit, CY	2	1	–	A.bit ← CY			
		PSW.bit, CY	3	4	–	PSW.bit ← CY	×	×	
		[HL].bit, CY	2	2	–	(HL).bit ← CY			
		CY, ES:[HL].bit	3	2	5	CY ← (ES, HL).bit			×
		ES:[HL].bit, CY	3	3	–	(ES, HL).bit ← CY			
	AND1	CY, saddr.bit	3	1	–	CY ← CY ∧ (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← CY ∧ sfr.bit			×
		CY, A.bit	2	1	–	CY ← CY ∧ A.bit			×
		CY, PSW.bit	3	1	–	CY ← CY ∧ PSW.bit			×
		CY,[HL].bit	2	1	4	CY ← CY ∧ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∧ (ES, HL).bit			×
	OR1	CY, saddr.bit	3	1	–	CY ← CY ∨ (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← CY ∨ sfr.bit			×
		CY, A.bit	2	1	–	CY ← CY ∨ A.bit			×
		CY, PSW.bit	3	1	–	CY ← CY ∨ PSW.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ∨ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ∨ (ES, HL).bit			×

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 28-5. Operation List (14/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Bit manipulate	XOR1	CY, saddr.bit	3	1	–	CY ← CY ⊻ (saddr).bit			×
		CY, sfr.bit	3	1	–	CY ← CY ⊻ sfr.bit			×
		CY, A.bit	2	1	–	CY ← CY ⊻ A.bit			×
		CY, PSW.bit	3	1	–	CY ← CY ⊻ PSW.bit			×
		CY, [HL].bit	2	1	4	CY ← CY ⊻ (HL).bit			×
		CY, ES:[HL].bit	3	2	5	CY ← CY ⊻ (ES, HL).bit			×
	SET1	saddr.bit	3	2	–	(saddr).bit ← 1			
		sfr.bit	3	2	–	sfr.bit ← 1			
		A.bit	2	1	–	A.bit ← 1			
		!addr16.bit	4	2	–	(addr16).bit ← 1			
		PSW.bit	3	4	–	PSW.bit ← 1	×	×	×
		[HL].bit	2	2	–	(HL).bit ← 1			
		ES:!addr16.bit	5	3	–	(ES, addr16).bit ← 1			
		ES:[HL].bit	3	3	–	(ES, HL).bit ← 1			
	CLR1	saddr.bit	3	2	–	(saddr.bit) ← 0			
		sfr.bit	3	2	–	sfr.bit ← 0			
		A.bit	2	1	–	A.bit ← 0			
		!addr16.bit	4	2	–	(addr16).bit ← 0			
		PSW.bit	3	4	–	PSW.bit ← 0	×	×	×
		[HL].bit	2	2	–	(HL).bit ← 0			
		ES:!addr16.bit	5	3	–	(ES, addr16).bit ← 0			
		ES:[HL].bit	3	3	–	(ES, HL).bit ← 0			
	SET1	CY	2	1	–	CY ← 1			1
	CLR1	CY	2	1	–	CY ← 0			0
	NOT1	CY	2	1	–	CY ← $\overline{CY}$			×

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 28-5. Operation List (15/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Call/ return	CALL	rp	2	3	–	(SP – 2) ← (PC + 2)S, (SP – 3) ← (PC + 2)H, (SP – 4) ← (PC + 2)L, PC ← CS, rp, SP ← SP – 4			
		$!addr20	3	3	–	(SP – 2) ← (PC + 3)S, (SP – 3) ← (PC + 3)H, (SP – 4) ← (PC + 3)L, PC ← PC + 3 + jdisp16, SP ← SP – 4			
		!addr16	3	3	–	(SP – 2) ← (PC + 3)S, (SP – 3) ← (PC + 3)H, (SP – 4) ← (PC + 3)L, PC ← 0000, addr16, SP ← SP – 4			
		!!addr20	4	3	–	(SP – 2) ← (PC + 4)S, (SP – 3) ← (PC + 4)H, (SP – 4) ← (PC + 4)L, PC ← addr20, SP ← SP – 4			
	CALLT	[addr5]	2	5	–	(SP – 2) ← (PC + 2)S, (SP – 3) ← (PC + 2)H, (SP – 4) ← (PC + 2)L , PCS ← 0000, PCH ← (0000, addr5 + 1), PCL ← (0000, addr5), SP ← SP – 4			
	BRK	–	2	5	–	(SP – 1) ← PSW, (SP – 2) ← (PC + 2)S, (SP – 3) ← (PC + 2)H, (SP – 4) ← (PC + 2)L, PCS ← 0000, PCH ← (0007FH), PCL ← (0007EH), SP ← SP – 4, IE ← 0			
	RET	–	1	6	–	PCL ← (SP), PCH ← (SP + 1), PCS ← (SP + 2), SP ← SP + 4			
	RETI	–	2	6	–	PCL ← (SP), PCH ← (SP + 1), PCS ← (SP + 2), PSW ← (SP + 3), SP ← SP + 4	R	R	R
	RETB	–	2	6	–	PCL ← (SP), PCH ← (SP + 1), PCS ← (SP + 2), PSW ← (SP + 3), SP ← SP + 4	R	R	R

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (fCPU) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 28-5. Operation List (16/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag		
				Note 1	Note 2		Z	AC	CY
Stack manipulate	PUSH	PSW	2	1	–	(SP – 1) ← PSW, (SP – 2) ← 00H, SP ← SP – 2			
		rp	1	1	–	(SP – 1) ← rp_H, (SP – 2)← rp_L, SP ← SP – 2			
	POP	PSW	2	3	–	PSW ← (SP + 1), SP ← SP + 2	R	R	R
		rp	1	1	–	rp_L ← (SP), rp_H ← (SP + 1), SP ← SP + 2			
	MOVW	SP, #word	4	1	–	SP ← word			
		SP, AX	2	1	–	SP ← AX			
		AX, SP	2	1	–	AX ← SP			
		HL, SP	3	1	–	HL ← SP			
		BC, SP	3	1	–	BC ← SP			
		DE, SP	3	1	–	DE ← SP			
	ADDW	SP, #byte	2	1	–	SP ← SP + byte			
	SUBW	SP, #byte	2	1	–	SP ← SP – byte			
Unconditional branch	BR	AX	2	3	–	PC ← CS, AX			
		$addr20	2	3	–	PC ← PC + 2 + jdisp8			
		$!addr20	3	3	–	PC ← PC + 3 + jdisp16			
		!addr16	3	3	–	PC ← 0000, addr16			
		!!addr20	4	3	–	PC ← addr20			
Conditional branch	BC	$addr20	2	2/4[Note 3]	–	PC ← PC + 2 + jdisp8 if CY = 1			
	BNC	$addr20	2	2/4[Note 3]	–	PC ← PC + 2 + jdisp8 if CY = 0			
	BZ	$addr20	2	2/4[Note 3]	–	PC ← PC + 2 + jdisp8 if Z = 1			
	BNZ	$addr20	2	2/4[Note 3]	–	PC ← PC + 2 + jdisp8 if Z = 0			
	BH	$addr20	3	2/4[Note 3]	–	PC ← PC+3+jdisp8 if (Z ∨ CY)=0			
	BNH	$addr20	3	2/4[Note 3]	–	PC ← PC+3+jdisp8 if (Z ∨ CY)=1			
	BT	saddr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1			
		sfr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1			
		A.bit, $addr20	3	3/5[Note 3]	–	PC ← PC + 3 + jdisp8 if A.bit = 1			
		PSW.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1			
<R>		[HL].bit, $addr20	3	3/5[Note 3]	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 1			
<R>		ES:[HL].bit, $addr20	4	4/6[Note 3]	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1			

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. This indicates the number of clocks “when condition is not met/when condition is met”.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

Table 28-5. Operation List (17/17)

Instruction Group	Mnemonic	Operands	Bytes	Clocks		Operation	Flag
				Note 1	Note 2		Z AC CY
Conditional branch	BF	saddr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 0	
		sfr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if sfr.bit = 0	
		A.bit, $addr20	3	3/5[Note 3]	–	PC ← PC + 3 + jdisp8 if A.bit = 0	
		PSW.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if PSW.bit = 0	
<R>		[HL].bit, $addr20	3	3/5[Note 3]	6/7	PC ← PC + 3 + jdisp8 if (HL).bit = 0	
<R>		ES:[HL].bit, $addr20	4	4/6[Note 3]	7/8	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 0	
	BTCLR	saddr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if (saddr).bit = 1 then reset (saddr).bit	
		sfr.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit	
		A.bit, $addr20	3	3/5[Note 3]	–	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit	
<R>		PSW.bit, $addr20	4	3/5[Note 3]	–	PC ← PC + 4 + jdisp8 if PSW.bit = 1 then reset PSW.bit	× × ×
		[HL].bit, $addr20	3	3/5[Note 3]	–	PC ← PC + 3 + jdisp8 if (HL).bit = 1 then reset (HL).bit	
		ES:[HL].bit, $addr20	4	4/6[Note 3]	–	PC ← PC + 4 + jdisp8 if (ES, HL).bit = 1 then reset (ES, HL).bit	
Conditional skip	SKC	–	2	1	–	Next instruction skip if CY = 1	
	SKNC	–	2	1	–	Next instruction skip if CY = 0	
	SKZ	–	2	1	–	Next instruction skip if Z = 1	
	SKNZ	–	2	1	–	Next instruction skip if Z = 0	
	SKH	–	2	1	–	Next instruction skip if (Z ∨ CY) = 0	
	SKNH	–	2	1	–	Next instruction skip if (Z ∨ CY) = 1	
CPU control	SEL	RBn	2	1	–	RBS[1:0] ← n	
	NOP	–	1	1	–	No Operation	
	EI	–	3	4	–	IE ← 1(Enable Interrupt)	
	DI	–	3	4	–	IE ← 0(Disable Interrupt)	
	HALT	–	2	3	–	Set HALT Mode	
	STOP	–	2	3	–	Set STOP Mode	

Notes 1. When the internal RAM area or SFR area is accessed, or for an instruction with no data access.

2. When the program memory area is accessed.

3. This indicates the number of clocks “when condition is not met/when condition is met”.

Remarks 1. One instruction clock cycle is one cycle of the CPU clock (f_{CPU}) selected by the system clock control register (CKC).

2. This number of clocks is for when the program is in the internal ROM (flash memory) area.

3. n indicates the number of register banks (n = 0 to 3)

CHAPTER 29 ELECTRICAL SPECIFICATIONS

Caution The 78K0R/Kx3-A 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.

Absolute Maximum Ratings (T_A = 25°C) (1/3)

Parameter	Symbols	Conditions	Ratings	Unit
Supply voltage	V_{DD}		−0.5 to +6.5	V
	EV_{DD}		−0.5 to +6.5	V
	V_{SS}		−0.5 to +0.3	V
	EV_{SS}		−0.5 to +0.3	V
	AV_{DD0}		−0.5 to V_{DD} +0.3 Note 1	V
	AV_{DD1}		−0.5 to V_{DD} +0.3 Note 1	V
	AV_{SS}		−0.5 to +0.3	V
REGC pin input voltage	V_{IREGC}	REGC	−0.3 to 3.6 and −0.3 to V_{DD} +0.3 Note 2	V
Input voltage	V_{I1}	P00 to P02, P10 to P16, P30 to P34, P40, P41, P50 to P57, P80 to P82, P100, P120 to P124, P140, P141, EXCLK, $\overline{\text{RESET}}$, FLMD0	−0.3 to EV_{DD} +0.3 and −0.3 to V_{DD} +0.3 Note 1	V
	V_{I2}	P60, P61 (N-ch open-drain)	−0.3 to +6.5	V
	V_{I3}	P20 to P27, P150 to P152, P157	−0.3 to AV_{DD0} +0.3 and −0.3 to V_{DD} +0.3 Note 1	V
	V_{I4}	P110, P111	−0.3 to AV_{DD1} +0.3 and −0.3 to V_{DD} +0.3 Note 1	V

Notes 1. Must be 6.5 V or lower.

2. Connect the REGC pin to Vss 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.

Caution Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Absolute Maximum Ratings (T_A = 25°C) (2/3)

Parameter	Symbols	Conditions		Ratings	Unit
Output voltage	V_{O1}	P00 to P02, P10 to P16, P30 to P34, P40, P41, P50 to P57, P60, P61, P80 to P82, P100, P120, P130, P140, P141		−0.3 to EV_{DD} +0.3	V
	V_{O2}	P20 to P27, P150 to P152, P157		−0.3 to AV_{DD0} +0.3	V
	V_{O3}	P110, P111		−0.3 to AV_{DD1} +0.3	V
Analog input voltage	V_{AI}	ANI0 to ANI10, ANI15, AMP0+, AMP1+, AMP2+, AMP0-, AMP1-, AMP2-		−0.3 to AV_{DD0} +0.3 and −0.3 to V_{DD} +0.3[Note]	V
<R> Analog output voltage	V_{AO1}	ANO0, ANO1		−0.3 to AV_{DD1} +0.3[Note]	V
	V_{AO2}	AMP0O, AMP1O, AMP2O		−0.3 to AV_{DD0} +0.3[Note]	V
Analog input reference voltage	AV_{REFP}			−0.3 to AV_{DD0} +0.3[Note]	V
	AV_{REFM}			−0.3 to AV_{DD0} +0.3[Note] and $AV_{REFM} \leq AV_{REFP}$	V
<R> Output current, high	I_{OH1}	Per pin	P00 to P02, P10-P16, P30 to P34, P40, P41, P80 to P82, P120, P130	−10	mA
			P50 to P57, P100, P140, P141	−10	mA
		Total of all pins −50 mA	P00 to P02, P10-P16, P30 to P34, P40, P41, P80 to P82, P120, P130	−25	mA
			P50 to P57, P100, P140, P141	−25	mA
	I_{OH2}	Per pin	P20 to P27, P150 to P152, P157, P110, P111	−0.5	mA
		Total of all pins		−2	mA
	I_{OH3}	Per pin	AMP0O, AMP1O, AMP2O	−1	mA
		Total of all pins		−3	mA

Note Must be 6.5 V or lower.

Cautions 1. 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.

2. The value of the current that can be run per pin must satisfy the value of the current per pin and the total value of the currents of all pins.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

Absolute Maximum Ratings (T_A = 25°C) (3/3)

	Parameter	Symbols	Conditions		Ratings	Unit
<R>	Output current, low	I_{OL1}	Per pin	P00 to P02, P10-P16, P30 to P34, P40, P41, P80 to P82, P120, P130	30	mA
				P60, P61	30	mA
				P50 to P57, P100, P140, P141	10	mA
			Total of all pins 165 mA	P00 to P02, P10-P16, P30 to P34, P40, P41, P80 to P82, P120, P130	80	mA
				P60, P61	60	mA
				P50 to P57, P100, P140, P141	25	mA
		I_{OL2}	Per pin	P20 to P27, P150 to P152, P157, P110, P111	1	mA
			Total of all pins		5	mA
		I_{OL3}	Per pin	AMP0O, AMP1O, AMP2O	1	mA
			Total of all pins		3	mA
	Operating ambient temperature	T_A	In normal operation mode		–40 to +85	°C
			In flash memory programming mode			
	Storage temperature	T_{stg}			–65 to +150	°C

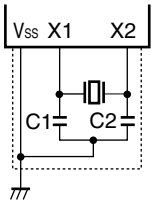
Cautions 1. Product quality may suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are rated values at which the product is on the verge of suffering physical damage, and therefore the product must be used under conditions that ensure that the absolute maximum ratings are not exceeded.

2. The value of the current that can be run per pin must satisfy the value of the current per pin and the total value of the currents of all pins.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

X1 Oscillator Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Parameter	Conditions	MIN.	TYP.	MAX.	Unit
Ceramic resonator, crystal resonator		X1 clock oscillation frequency (f_X)Note	2.7 V ≤ V_{DD} ≤ 5.5 V	2.0		20.0	MHz
			1.8 V ≤ V_{DD} < 2.7 V	2.0		5.0	

Note Indicates only oscillator characteristics. Refer to AC Characteristics for instruction execution time.

Cautions 1. When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

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

2. Since the CPU is started by the internal high-speed oscillation 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 oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.

Remark For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

Internal Oscillator Characteristics

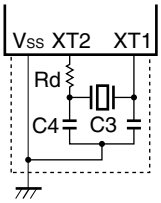
(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Oscillators	Parameters	Conditions		MIN.	TYP.	MAX.	Unit
Internal high-speed oscillation clock frequency Note	f_{IH1M}	Low-power consumption mode		0.87	1	1.13	MHz
	f_{IH8M}	2.7 V ≤ V_{DD} ≤ 5.5 V		7.856	8	8.144	MHz
		1.8 V ≤ V_{DD} < 2.7 V, T_A = –20 to +70°C		7.848	8	8.152	MHz
		1.8 V ≤ V_{DD} < 2.7 V		7.84	8	8.16	MHz
	f_{IH20M}	2.7 V ≤ V_{DD} ≤ 5.5 V		19.52	20	20.48	MHz
Internal low-speed oscillation clock frequency	f_{IL}	Normal power mode	2.7 V ≤ V_{DD} ≤ 5.5 V	27	30	33	kHz
			1.8 V ≤ V_{DD} < 2.7 V	25.5	30	34.5	kHz
		Low-power consumption mode		25.5	30	34.5	kHz

Note Indicates only oscillator characteristics. Refer to AC Characteristics for instruction execution time.

XT1 Oscillator Characteristics

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Resonator	Recommended Circuit	Items	Conditions	MIN.	TYP.	MAX.	Unit
Crystal resonator		XT1 clock oscillation frequency (f_{XT})Note		32	32.768	35	kHz

Note Indicates only oscillator characteristics. Refer to AC Characteristics for instruction execution time.

Cautions 1. When using the XT1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance.

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

2. The XT1 oscillator is designed as a low-amplitude circuit for reducing power consumption, and is more prone to malfunction due to noise than the X1 oscillator. Particular care is therefore required with the wiring method when the XT1 clock is used.

Remark For the resonator selection and oscillator constant, customers are requested to either evaluate the oscillation themselves or apply to the resonator manufacturer for evaluation.

<R> **Recommended oscillator circuit constants**

(1) X1 oscillation: Ceramic resonator (AMPH = 0, RMC = 00H, T_A = –40 to +85°C)

Manufacturer	Part Number	SMD/ Lead	Frequency (MHz)	Recommended Circuit Constants			Oscillation Voltage Range	
				C1 (pF)	C2 (pF)	Rd (kΩ)	MIN. (V)	MAX. (V)
Murata Manufacturing Co., Ltd..	CSTCC2M00G56-R0	SMD	2.0	Internal (47)	Internal (47)	0	1.8	5.5
	CSTCR4M00G55-R0	SMD	4.0	Internal (39)	Internal (39)	0		
	CSTLS4M00G56-B0	Lead		Internal (47)	Internal (47)	0		
	CSTCR4M19G55-R0	SMD	4.194	Internal (39)	Internal (39)	0		
	CSTLS4M19G56-B0	Lead		Internal (47)	Internal (47)	0		
	CSTCR4M91G55-R0	SMD	4.915	Internal (39)	Internal (39)	0		
	CSTLS4M91G53-B0	Lead		Internal (15)	Internal (15)	0		
	CSTCR5M00G55-R0	SMD	5.0	Internal (39)	Internal (39)	0		
	CSTLS5M00G53-B0	Lead		Internal (15)	Internal (15)	0		
	CSTCR6M00G53-R0	SMD	6.0	Internal (15)	Internal (15)	0		
	CSTLS6M00G53-B0	Lead		Internal (15)	Internal (15)	0		
	CSTCE8M00G55-R0	SMD	8.0	Internal (33)	Internal (33)	0		
	CSTLS8M00G53-B0	Lead		Internal (15)	Internal (15)	0		
	CSTCE8M38G55-R0	SMD	8.388	Internal (33)	Internal (33)	0		
	CSTLS8M38G53-B0	Lead		Internal (15)	Internal (15)	0		
	CSTCE10M0G52-R0	SMD	10.0	Internal (10)	Internal (10)	0		
	CSTLS10M0G53-B0	Lead		Internal (15)	Internal (15)	0		

Caution The oscillator constants shown above are reference values based on evaluation in a specific environment by the resonator manufacturer. If it is necessary to optimize the oscillator characteristics in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit. The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the 78K0R/Kx3-A so that the internal operation conditions are within the specifications of the DC and AC characteristics.

(2) X1 oscillation: Ceramic resonator (AMPH = 1, RMC = 00H, T_A = –40 to +85°C)

Manufacturer	Part Number	SMD/ Lead	Frequency (MHz)	Recommended Circuit Constants			Oscillation Voltage Range	
				C1 (pF)	C2 (pF)	Rd (kΩ)	MIN. (V)	MAX. (V)
Murata Manufacturing Co., Ltd..	CSTCE12M0G55-R0	SMD	12.0	Internal (33)	Internal (33)	0	1.8	5.5
	CSTCE16M0V53-R0	SMD	16.0	Internal (15)	Internal (15)	0		
	CSTLS16M0X51-B0	Lead		Internal (5)	Internal (5)	0		
	CSTCE20M0V53-R0	SMD	20.0	Internal (15)	Internal (15)	0		
	CSTLS20M0X51-B0	Lead		Internal (5)	Internal (5)	0		

(3) X1 oscillation: Ceramic resonator (AMPH = 0, RMC = 5AH, T_A = –40 to +85°C)

Manufacturer	Part Number	SMD/ Lead	Frequency (MHz)	Recommended Circuit Constants			Oscillation Voltage Range	
				C1 (pF)	C2 (pF)	Rd (kΩ)	MIN. (V)	MAX. (V)
Murata Manufacturing Co., Ltd..	CSTCC2M00G56-R0	SMD	2.0	Internal (47)	Internal (47)	0	1.8	5.5
	CSTCR4M00G55-R0	SMD	4.0	Internal (39)	Internal (39)	0		
	CSTLS4M00G53-B0	Lead		Internal (15)	Internal (15)	0		
	CSTCR4M19G55-R0	SMD	4.194	Internal (39)	Internal (39)	0		
	CSTLS4M19G53-B0	Lead		Internal (15)	Internal (15)	0		
	CSTCR4M91G53-R0	SMD	4.195	Internal (15)	Internal (15)	0		
	CSTLS4M91G53-B0	Lead		Internal (15)	Internal (15)	0		
	CSTCR5M00G53-R0	SMD	5.0	Internal (15)	Internal (15)	0		
	CSTLS5M00G53-B0	Lead		Internal (15)	Internal (15)	0		
TOKO, Inc.	DCRHTC(P)2.00LL	Lead	2.0	Internal (30)	Internal (30)	–	1.8	5.5
	DCRHTC(P)4.00LL		4.0	Internal (30)	Internal (30)	–		
	DECRHTC4.00	SMD	4.0	Internal (15)	Internal (15)	–		
	DCRHTC(P)5.00LL	Lead	5.0	Internal (30)	Internal (30)	–		

Caution The oscillator constants shown above are reference values based on evaluation in a specific environment by the resonator manufacturer. If it is necessary to optimize the oscillator characteristics in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit. The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the 78K0R/Kx3-A so that the internal operation conditions are within the specifications of the DC and AC characteristics.

(4) XT1 oscillation: Crystal resonator (T_A = –40 to +85°C)

Manufacturer	Part Number	SMD/ Lead	Frequency (MHz)	Load Capacitance CL (pF)	XT1 oscllator oscillation mode Note 1	Recommended Circuit Constants			Oscillation Voltage Range	
						C3 (pF)	C4 (pF)	Rd (kΩ)	MIN. (V)	MAX. (V)
Seiko Instruments Inc. Note 2	SSP-T7-F	SMD	32.768	7.0	Normal oscillation	10	10	0	1.8	5.5
	SSP-T7-FL			6.0	Low power consumption oscillation	9	8	0		
				3.7	Ultra-low power consumption oscillation	4	3	0		
	VT-200-F	Lead		12.5	Normal oscillation	20	20	0		
	VT-200-FL			6.0	Low power consumption oscillation	9	8	0		
				3.7	Ultra-low power consumption oscillation	4	3	0		

Notes 1. Set the XT1 oscillation mode by using bits AMPHS1 and AMPHS0 of the clock operation mode control register (CMC).

2. Contact Seiko Instruments Inc. (http://www.sii-crystal.com) when using this resonator.

DC Characteristics (1/10)

(T_A = −40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 1.8 V ≤ AV_{DD0} ≤ V_{DD}, 1.8 V ≤ AV_{DD1} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, high[Note 1]	I_{OH1}	Per pin for P00 to P02, P10 to P16, P30 to P34, P40, P41, P80 to P82, P120, P130	4.0 V ≤ V_{DD} ≤ 5.5 V			−3.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			−1.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			−1.0	mA
		Per pin for P50 to P57, P100, P140, P141	4.0 V ≤ V_{DD} ≤ 5.5 V			−1.6	mA
			2.7 V ≤ V_{DD} < 4.0 V			−0.45	mA
			1.8 V ≤ V_{DD} < 2.7 V			−0.45	mA
		Total of P00 to P02, P10 to P16, P30 to P34, P40, P41, P80 to P82, P120, P130 (When duty = 70%[Note 2])	4.0 V ≤ V_{DD} ≤ 5.5 V			−20.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			−10.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			−5.0	mA
		Total of P50 to P57, P100, P140, P141 (When duty = 70%[Note 2])	4.0 V ≤ V_{DD} ≤ 5.5 V			−12.8	mA
			2.7 V ≤ V_{DD} < 4.0 V			−3.6	mA
			1.8 V ≤ V_{DD} < 2.7 V			−3.6	mA
		Total of all pins (When duty = 60%[Note 2])	4.0 V ≤ V_{DD} ≤ 5.5 V			−32.8	mA
			2.7 V ≤ V_{DD} < 4.0 V			−13.6	mA
			1.8 V ≤ V_{DD} < 2.7 V			−8.6	mA
	I_{OH2}	Per pin for P20 to P27, P150 to P152, P157				−0.1	mA
		Per pin for P110, P111				−0.1	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from V_{DD} pin to an output pin.

2. Specification under conditions where the duty factor is 60% or 70%.
The output current value that has changed the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).
•Total output current of pins = (I_{OH} × 0.7)/(n × 0.01)
<Example> Where n = 50% and I_{OH} = −20.0 mA
Total output current of pins = (−20.0 × 0.7)/(50 × 0.01) = −28.0 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, P80 and P82 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 port pins.

DC Characteristics (2/10)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 1.8 V ≤ AV_{DD0} ≤ V_{DD}, 1.8 V ≤ AV_{DD1} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output current, low[Note 1]	I_{OL1}	Per pin for P00 to P02, P12, P13, P16, P30 to P34, P40, P41, P80 to P82, P120, P130	4.0 V ≤ V_{DD} ≤ 5.5 V			8.5	mA
			2.7 V ≤ V_{DD} < 4.0 V			1.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			0.5	mA
		Per pin for P10, P11, P14, P15	4.0 V ≤ V_{DD} ≤ 5.5 V			8.5	mA
			2.7 V ≤ V_{DD} < 4.0 V			1.5	mA
			1.8 V ≤ V_{DD} < 2.7 V			0.6	mA
		Per pin for P60, P61	4.0 V ≤ V_{DD} ≤ 5.5 V			15.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			3.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			2.0	mA
		Per pin for P50 to P57, P100, P140, P141	4.0 V ≤ V_{DD} ≤ 5.5 V			1.8	mA
			2.7 V ≤ V_{DD} < 4.0 V			0.8	mA
			1.8 V ≤ V_{DD} < 2.7 V			0.35	mA
		Total of P00 to P02, P10 to P16, P30 to P34, P40, P41, P80 to P82, P120, P130 (When duty = 70%[Note 2])	4.0 V ≤ V_{DD} ≤ 5.5 V			20.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			15.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			9.0	mA
		Total of P60, P61 (When duty = 70%[Note 2])	4.0 V ≤ V_{DD} ≤ 5.5 V			30.0	mA
			2.7 V ≤ V_{DD} < 4.0 V			6.0	mA
			1.8 V ≤ V_{DD} < 2.7 V			4.0	mA
		Total of P50 to P57, P100, P140, P141 (When duty = 70%[Note 2])	4.0 V ≤ V_{DD} ≤ 5.5 V			14.4	mA
			2.7 V ≤ V_{DD} < 4.0 V			6.4	mA
			1.8 V ≤ V_{DD} < 2.7 V			2.8	mA
		Total of all pins (When duty = 70%[Note 2])	4.0 V ≤ V_{DD} ≤ 5.5 V			64.4	mA
			2.7 V ≤ V_{DD} < 4.0 V			27.4	mA
			1.8 V ≤ V_{DD} < 2.7 V			15.8	mA
	I_{OL2}	Per pin for P20 to P27, P150 to P152, P157				0.4	mA
		Per pin for P110, P111				0.4	mA

Notes 1. Value of current at which the device operation is guaranteed even if the current flows from an output pin to V_{SS} and AV_{SS} pin.

2. Specification under conditions where the duty factor is 60% or 70%.
The output current value that has changed the duty ratio can be calculated with the following expression (when changing the duty factor from 70% to n%).
•Total output current of pins = (I_{OL} × 0.7)/(n × 0.01)
<Example> Where n = 50% and I_{OL} = 20.0 mA
Total output current of pins = (20.0 × 0.7)/(50 × 0.01) = 28.0 mA
However, the current that is allowed to flow into one pin does not vary depending on the duty factor. A current higher than the absolute maximum rating must not flow into one pin.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (3/10)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 1.8 V ≤ AV_{DD0} ≤ V_{DD}, 1.8 V ≤ AV_{DD1} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, high	V_{IH1}	P00 to P02, P12, P13, P41, P51, P54 to P57, P82, P100, P123, P124, P140, P141		0.7V_{DD}		V_{DD}	V
	V_{IH2}	P10, P11, P14 to P16, P30 to P34, P40, P50, P52, P53, P80, P81, P120 to P122, $\overline{\text{RESET}}$	Normal input buffer	0.8V_{DD}		V_{DD}	V
	V_{IH3}	P10, P11, P14, P15	TTL input buffer 4.0 V ≤ V_{DD} ≤ 5.5 V	2.2		V_{DD}	V
			TTL input buffer 2.7 V ≤ V_{DD} < 4.0 V	2.0		V_{DD}	V
			TTL input buffer 1.8 V ≤ V_{DD} < 2.7 V	1.6		V_{DD}	V
	V_{IH4}	P20 to P27, P150 to P152, P157		0.7AV_{DD0}		AV_{DD0}	V
	V_{IH5}	P110, P111		0.7AV_{DD1}		AV_{DD1}	V
	V_{IH6}	P60, P61		0.7V_{DD}		6.0	V
	V_{IH7}	FLMD0		0.9V_{DD} Note		V_{DD}	V

Note Must be 0.9V_{DD} or higher when used in the flash memory programming mode.

Caution The maximum value of V_{IH} of pins P10 to P15, P80 and P82 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 port pins.

DC Characteristics (4/10)

(T_A = −40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 1.8 V ≤ AV_{DD0} ≤ V_{DD}, 1.8 V ≤ AV_{DD1} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Input voltage, low	V_{IL1}	P00 to P02, P12, P13, P41, P51, P54 to P57, P82, P100, P123, P124, P140, P141		0		$0.3V_{DD}$	V
	V_{IL2}	P10, P11, P14 to P16, P30 to P34, P40, P50, P52, P53, P80, P81, P120 to P122, $\overline{\text{RESET}}$	Normal input buffer	0		$0.2V_{DD}$	V
	V_{IL3}	P10, P11, P14, P15	TTL input buffer 4.0 V ≤ V_{DD} ≤ 5.5 V	0		0.8	V
			TTL input buffer 2.7 V ≤ V_{DD} < 4.0 V	0		0.5	V
			TTL input buffer 1.8 V ≤ V_{DD} < 2.7 V	0		0.2	V
	V_{IL4}	P20 to P27, P150 to P152, P157		0		$0.3AV_{DD0}$	V
	V_{IL5}	P110, P111		0		$0.3AV_{DD1}$	V
	V_{IL6}	P60, P61		0		$0.3V_{DD}$	V
	V_{IL7}	FLMD0		0		$0.1V_{DD}$ Note	V

Note When disabling writing of the flash memory, connect the FLMD0 pin directly to V_{SS}, and maintain a voltage less than $0.1V_{DD}$.

Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (5/10)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 1.8 V ≤ AV_{DD0} ≤ V_{DD}, 1.8 V ≤ AV_{DD1} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Items	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Output voltage, high	V_{OH1}	P00 to P02, P10 to P16, P30 to P34, P40, P41, P80 to P82, P120, P130	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = – 3.0 mA	V_{DD} – 0.7			V
			1.8 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –1.0 mA	V_{DD} – 0.5			V
		P50 to P57, P100, P140, P141	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = – 1.6 mA	V_{DD} – 0.7			V
			1.8 V ≤ V_{DD} ≤ 5.5 V, I_{OH1} = –0.45 mA	V_{DD} – 0.5			V
	V_{OH2}	P20 to P27, P150 to P152, P157	I_{OH2} = –0.1 mA	AV_{DD0} – 0.5			V
		P110, P111	I_{OH2} = –0.1 mA	AV_{DD1} – 0.5			V
Output voltage, low	V_{OL1}	P00 to P02, P12, P13, P16, P30 to P34, P40, P41, P80 to P82, P120, P130	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} = 1.0 mA			0.5	V
			1.8 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 0.5 mA			0.4	V
		P10, P11, P14, P15	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
			1.8 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 0.6 mA			0.4	V
		P50 to P57, P100, P140, P141	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 1.8 mA			0.7	V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 0.8 mA			0.5	V
			1.8 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 0.35 mA			0.4	V
<R>	V_{OL2}	P20 to P27, P150 to P152, P157	AV_{DD0} ≤ 5.5 V, I_{OL2} = 0.4 mA			0.4	V
		P110, P111	AV_{DD1} ≤ 5.5 V, I_{OL2} = 0.4 mA			0.4	V
	V_{OL3}	P60, P61	4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 15.0 mA			2.0	V
			4.0 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 5.0 mA			0.4	V
			2.7 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 3.0 mA			0.4	V
			1.8 V ≤ V_{DD} ≤ 5.5 V, I_{OL1} = 2.0 mA			0.4	V

Caution P10 to P15, P80 and P82 do not output high level in N-ch open-drain mode.

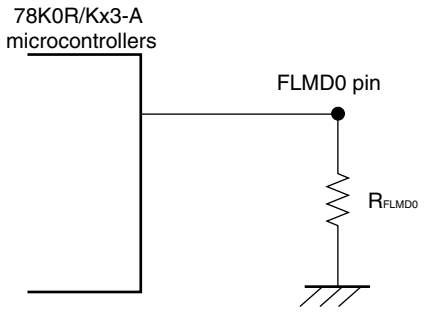
Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (6/10)

(T_A = −40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 1.8 V ≤ AV_{DD0} ≤ V_{DD}, 1.8 V ≤ AV_{DD1} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Input leakage current, high	I_{LIH1}	P00 to P02, P10 to P16, P30 to P34, P40, P41, P50 to P57, P60, P61, P80 to P82, P100, P120, P140, P141, FLMD0, $\overline{\text{RESET}}$	$V_I = V_{DD}$				1	μA
	I_{LIH2}	P20 to P27, P150 to P152, P157	$V_I = AV_{DD0}$				1	μA
		P110, P111	$V_I = AV_{DD1}$				1	μA
	I_{LIH3}	P121 to P124 (X1, X2, XT1, XT2)	$V_I = V_{DD}$	In input port			1	μA
				In resonator connection			10	μA
Input leakage current, low	I_{LIL1}	P00 to P02, P10 to P16, P30 to P34, P40, P41, P50 to P57, P60, P61, P80 to P82, P100, P120, P140, P141, FLMD0, $\overline{\text{RESET}}$	$V_I = V_{SS}$				−1	μA
	I_{LIL2}	P20 to P27, P150 to P152, P157	$V_I = V_{SS}$				−1	μA
		P110, P111	$V_I = V_{SS}$				−1	μA
	I_{LIL3}	P121 to P124 (X1, X2, XT1, XT2)	$V_I = V_{SS}$	In input port			−1	μA
				In resonator connection			−10	μA
On-chip pll-up resistance	R_U	P00 to P02, P10 to P16, P30 to P34, P40, P41, P50 to P57, P80 to P82, P100, P120, P140, P141	$V_I = V_{SS}$, In input port		10	20	100	kΩ
FLMD0 pin external pull-down resistance Note	R_{FLMD0}	When enabling the self-programming mode setting with software			100			kΩ

Note It is recommended to leave the FLMD0 pin open. If the pin is required to be pulled down externally, set R_{FLMD0} to 100 kΩ or more.



Remark Unless specified otherwise, the characteristics of alternate-function pins are the same as those of port pins.

DC Characteristics (7/10)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 1.8 V ≤ AV_{DD0} ≤ V_{DD}, 1.8 V ≤ AV_{DD1} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol		Conditions			MIN.	TYP.	MAX.	Unit
Supply current	I_{DD1} Note 1	Operating mode	f_{MX} = 20 MHz, V_{DD} = 5.0 V Note 2		Square wave input		5.5	7.7	mA
					Resonator connection		5.8	8.0	
			f_{MX} = 20 MHz, V_{DD} = 3.0 V Note 2		Square wave input		5.5	7.7	mA
					Resonator connection		5.8	8.0	
			f_{MX} = 10 MHz, V_{DD} = 5.0 V Notes 2, 3		Square wave input		3.2	4.6	mA
					Resonator connection		3.3	4.7	
			f_{MX} = 10 MHz, V_{DD} = 3.0 V Notes 2, 3		Square wave input		3.2	4.6	mA
					Resonator connection		3.3	4.7	
			f_{MX} = 5 MHz, V_{DD} = 3.0 V Notes 2, 3		Square wave input		1.8	2.7	mA
					Resonator connection		1.9	2.8	
<R>			f_{MX} = 5 MHz, V_{DD} = 2.0 V Notes 2, 3		Square wave input		1.3	2.2	mA
					Resonator connection		1.3	2.2	
<R>			f_{IH} = 20 MHz Note 4		V_{DD} = 5.0 V		5.7	8.0	mA
<R>					V_{DD} = 3.0 V		5.7	8.0	
			f_{IH} = 8 MHz Note 4		V_{DD} = 5.0 V		2.6	3.7	mA
					V_{DD} = 3.0 V		2.6	3.7	
			f_{IH} = 1 MHz, RMC = 5AH, OSMC = 02H Note 4		V_{DD} = 3.0 V		190	354	μA
			f_{SUB} = 32.768 kHz, FSEL = 0, SDIV = 1 Note 5	T_A = –40 to +50°C	V_{DD} = 5.0 V		3.9	8.4	μA
					V_{DD} = 3.0 V		3.9	8.4	
					V_{DD} = 2.0 V		3.9	8.4	
				T_A = –40 to +70°C	V_{DD} = 5.0 V		3.9	11.3	μA
					V_{DD} = 3.0 V		3.9	11.3	
					V_{DD} = 2.0 V		3.9	11.3	
				T_A = –40 to +85°C	V_{DD} = 5.0 V		3.9	14.6	μA
					V_{DD} = 3.0 V		3.9	14.6	
					V_{DD} = 2.0 V		3.9	14.6	

Notes 1. Total current flowing into V_{DD}, EV_{DD}, AV_{DD0}, and AV_{DD1}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The maximum value include the peripheral operation current. However, not including the current flowing into the Real-time counter, watchdog counter, LVI circuit, A/D converter, D/A converter, operational amplifier, voltage reference, I/O port, and on-chip pull-up/pull-down resistors.

2. When internal high-speed oscillator and subsystem clock are stopped.

3. When AMPH (bit 0 of clock operation mode control register (CMC)) = 0 and FLPC, FSEL (bits 1, 0 of operation speed mode control register (OSMC)) = 0, 0.

4. When high-speed system clock and subsystem clock are stopped.

5. When internal high-speed oscillation, and high-speed system clock are stopped. When watchdog timer is stopped.

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{IH}: Internal high-speed oscillation clock frequency

3. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

<R> **DC Characteristics (8/10)**

(T_A = −40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 1.8 V ≤ AV_{DD0} ≤ V_{DD}, 1.8 V ≤ AV_{DD1} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol		Conditions			MIN.	TYP.	MAX.	Unit
Supply current	I_{DD2} Note 1	HALT mode	f_{MX} = 20 MHz, V_{DD} = 5.0 V Note 2		Square wave input		1.1	3.3	mA
					Resonator connection		1.4	3.6	
			f_{MX} = 20 MHz, V_{DD} = 3.0 V Note 2		Square wave input		1.1	3.3	mA
					Resonator connection		1.4	3.6	
			f_{MX} = 10 MHz, V_{DD} = 5.0 V Notes 2, 3		Square wave input		0.55	2.1	mA
					Resonator connection		0.65	2.2	
			f_{MX} = 10 MHz, V_{DD} = 3.0 V Notes 2, 3		Square wave input		0.55	2.1	mA
					Resonator connection		0.65	2.2	
			f_{MX} = 5 MHz, V_{DD} = 3.0 V Notes 2, 3		Square wave input		0.4	1.8	mA
					Resonator connection		0.45	1.8	
			f_{MX} = 5 MHz, V_{DD} = 2.0 V Notes 2, 3		Square wave input		0.26	1.3	mA
					Resonator connection		0.31	1.4	
			f_{IH} = 20 MHz Note 4		V_{DD} = 5.0 V		1.3	3.6	mA
					V_{DD} = 3.0 V		1.3	3.6	
			f_{IH} = 8 MHz Note 4		V_{DD} = 5.0 V		0.45	1.8	mA
					V_{DD} = 3.0 V		0.45	1.8	
			f_{IH} = 1 MHz, RMC = 5AH, OSMC = 02H Note 4		V_{DD} = 3.0 V		45	153	μA
			f_{SUB} = 32.768 kHz, RTCLPC = 1, FSEL = 0, SDIV = 1 Note 5	T_A = −40 to +50°C	V_{DD} = 5.0 V		0.9	3.6	μA
					V_{DD} = 3.0 V		0.9	3.6	
					V_{DD} = 2.0 V		0.9	3.6	
				T_A = −40 to +70°C	V_{DD} = 5.0 V		0.9	6.0	μA
					V_{DD} = 3.0 V		0.9	6.0	
					V_{DD} = 2.0 V		0.9	6.0	
				T_A = −40 to +85°C	V_{DD} = 5.0 V		0.9	8.8	μA
					V_{DD} = 3.0 V		0.9	8.8	
					V_{DD} = 2.0 V		0.9	8.8	

Notes 1. Total current flowing into V_{DD}, EV_{DD}, AV_{DD0}, and AV_{DD1}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The maximum value include the peripheral operation current. However, not including the current flowing into the Real-time counter, watchdog counter, LVI circuit, A/D converter, D/A converter, operational amplifier, voltage reference, I/O port, and on-chip pull-up/pull-down resistors. During HALT instruction execution by flash memory.

2. When internal high-speed oscillator and subsystem clock are stopped.

3. When AMPH (bit 0 of clock operation mode control register (CMC)) = 0 and FLPC, FSEL (bits 1, 0 of operation speed mode control register (OSMC)) = 0, 0.

4. When high-speed system clock and subsystem clock are stopped.

5. When internal high-speed oscillation, and high-speed system clock are stopped. When watchdog timer is stopped. When real-time counter is operating.

Remarks 1. f_{MX}: High-speed system clock frequency (X1 clock oscillation frequency or external main system clock frequency)

2. f_{IH}: Internal high-speed oscillation clock frequency

3. f_{SUB}: Subsystem clock frequency (XT1 clock oscillation frequency)

DC Characteristics (9/10)

(T_A = −40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 1.8 V ≤ AV_{DD0} ≤ V_{DD}, 1.8 V ≤ AV_{DD1} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Supply current	I_{DD3} Note 1	STOP mode	T_A = −40 to +50°C			0.37	2.8	μA
			T_A = −40 to +70°C			0.37	5.2	
			T_A = −40 to +85°C			0.37	7.9	
RTC operating current	I_{RTC} Notes 2, 3	f_{SUB} = 32.768 kHz		V_{DD} = 3.0 V		0.2	1	μA
				V_{DD} = 2.0 V		0.2	1	μA
Watchdog timer operating current	I_{WDT} Notes 3, 4	f_{IL} = 30 kHz				0.31	0.35	μA
LVI operating current	I_{LVI} Note 5					9	18	μA
A/D converter operating current	I_{ADC} Note 6	During conversion at maximum speed	Normal mode 1	AV_{DD0} = 5.0 V		1.7	3.4	mA
				AV_{DD0} = 3.0 V		0.7	1.4	mA
			Normal mode 2	AV_{DD0} = 2.3 V		0.5	1.2	mA
			Low voltage mode	AV_{DD0} = 1.8 V		0.3	0.8	mA
D/A converter operating current	I_{DAC} Note 7	50 pF per 1 channel, I_{SOUCE} = I_{SINK} = 0 mA	Selecting Reference potential = AV_{DD1}			0.3	0.8	mA
			Selecting Reference potential = V_{REFOUT}			0.3	0.8	mA
			Selecting Reference potential = AV_{REFP}			0.3 Note 8	0.8 Note 8	mA

Notes 1. Total current flowing into V_{DD}, EV_{DD}, AV_{DD0}, and AV_{DD1}, including the input leakage current flowing when the level of the input pin is fixed to V_{DD} or V_{SS}. The maximum value includes the peripheral operation current and STOP leakage current. However, not including the current flowing into the Real-time counter, watchdog counter, LVI circuit, A/D converter, D/A converter, operational amplifier, voltage reference, I/O port, and on-chip pull-up/pull-down resistors. When subsystem clock is stopped. When watchdog timer is stopped.

2. Current flowing only to the real-time counter (V_{DD} pin) (excluding the operating current of the XT1 oscillator). The current value of the 78K0R/Kx3-A microcontrollers is the TYP. value, the sum of the TYP. values of either I_{DD1} or I_{DD2}, and I_{RTC}, when the real-time counter operates in an operation mode or HALT mode. The I_{DD1} and I_{DD2} MAX. values also include the real-time counter operating current. When the real-time counter operates during f_{CLK} = f_{SUBC}, the TYP. value of I_{DD2} includes the real-time counter operating current.

3. When internal high-speed oscillator and high-speed system clock are stopped.

4. Current flowing only to the watchdog timer (V_{DD} pin) (including the operating current of the 30 kHz internal oscillator). The current value of the 78K0R/Kx3-A microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{WDT} when f_{CLK} = f_{SUBC} or when the watchdog timer operates in STOP mode.

5. Current flowing only to the LVI circuit (V_{DD} pin). The current value of the 78K0R/Kx3-A microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{LVI} when the LVI circuit operates in the operation mode, HALT mode or STOP mode.

6. Current flowing only to the A/D converter (AV_{DD0} pin). The current value of the 78K0R/Kx3-A microcontrollers is the sum of I_{DD1} or I_{DD2} and I_{ADC} when the A/D converter operates in an operation mode or HALT mode.

7. Current flowing only to the D/A converter (AV_{DD1} pin). The current value of the 78K0R/Kx3-A microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{DAC} when the D/A converter operates in an operation mode, HALT mode or STOP mode.

8. Not including the current flowing to reference potential side.

<R>

DC Characteristics (10/10)

(T_A = −40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 1.8 V ≤ AV_{DD0} ≤ V_{DD}, 1.8 V ≤ AV_{DD1} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Operational amplifier operating current	I_{AMP} Note 1	AV_{DD0} = 5.0 V	OAIMI = 0		250	335	μA
		AV_{DD0} = 3.0 V	OAIMI = 0		230	320	μA
		AV_{DD0} = 2.3 V	OAIMI = 0		220	310	μA
Voltage reference operating current 1	I_{VR1} Note 2	AV_{DD0} = 5.0 V			19	38	μA
		AV_{DD0} = 3.0 V	VR output = 2.5 V		9.5	25	μA
		AV_{DD0} = 3.0 V	VR output = 2.0 V		9.5	25	μA
Voltage reference operating current 2	I_{VR2} Note 3	V_{DD} = 5.0 V			10	40	μA
		V_{DD} = 3.0 V	VR output = 2.5 V		10	40	μA
		V_{DD} = 3.0 V	VR output = 2.0 V		10	40	μA

Notes 1. Current flowing only to the operational amplifier (AV_{DD0} pin). The current value of the 78K0R/Kx3-A microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{AMP} when the operational amplifier operates in an operation mode, HALT mode or STOP mode.

2. Current flowing only to the voltage reference (AV_{DD0} pin). The current value of the 78K0R/Kx3-A microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{VR1} when the voltage reference circuit operates in an operation mode, HALT mode or STOP mode.

3. Current flowing only to the voltage reference or input gate voltage boost circuit for the A/D converter (V_{DD} pin). The current value of the 78K0R/Kx3-A microcontrollers is the sum of I_{DD1}, I_{DD2} or I_{DD3} and I_{VR2} when the voltage reference or boost circuit operates in an operation mode, HALT mode or STOP mode.

AC Characteristics

(1) Basic operation (1/6)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 1.8 V ≤ AV_{DD0} ≤ V_{DD}, 1.8 V ≤ AV_{DD1} ≤ V_{DD}, V_{SS} = EV_{SS} = AV_{SS} = 0 V)

Items	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Instruction cycle (minimum instruction execution time)	T_{CY}	Main system clock (f_{MAIN}) operation	Normal power mode, FSEL bit = 1	2.7 V ≤ V_{DD} ≤ 5.5 V	0.05		8	μs
				1.8 V ≤ V_{DD} < 2.7 V	0.2		8	μs
			Normal power mode, FSEL bit = 0	2.7 V ≤ V_{DD} ≤ 5.5 V	0.1		8	μs
				1.8 V ≤ V_{DD} < 2.7 V	0.2		8	μs
			Low consumption power mode		1		8	μs
		Subsystem clock (f_{SUB}) operation	SDIV bit = 1		57.2	61	62.5	μs
			SDIV bit = 0		28.5	30.5	31.3	μs
		In the self programming mode	Normal power mode, FSEL bit = 1	2.7 V ≤ V_{DD} ≤ 5.5 V	0.05		1	μs
				1.8 V ≤ V_{DD} < 2.7 V	0.2		1	μs
			Low consumption power mode Note		0.88	1	1.15	μs
External main system clock frequency	f_{EX}	2.7 V ≤ V_{DD} ≤ 5.5 V			2.0		20.0	MHz
		1.8 V ≤ V_{DD} < 2.7 V			2.0		5.0	MHz
External main system clock input high-level width, low-level width	t_{EXH}, t_{EXL}	2.7 V ≤ V_{DD} ≤ 5.5 V			24			ns
		1.8 V ≤ V_{DD} < 2.7 V			96			ns
TI00 to TI07 input high-level width, low-level width	t_{TIH}, t_{TIL}				2/f_{MCK} +10			ns
TO00 to TO07 output frequency	f_{TO}	2.7 V ≤ V_{DD} ≤ 5.5 V					10	MHz
		1.8 V ≤ V_{DD} < 2.7 V					5	MHz
PCLBUZ0, PCLBUZ1 output frequency	f_{PCL}	2.7 V ≤ V_{DD} ≤ 5.5 V					10	MHz
		1.8 V ≤ V_{DD} < 2.7 V					5	MHz
Interrupt input high-level width, low-level width	t_{INTH}, t_{INTL}				1			μs
$\overline{RESET}$ low-level width	t_{RSL}				10			μs

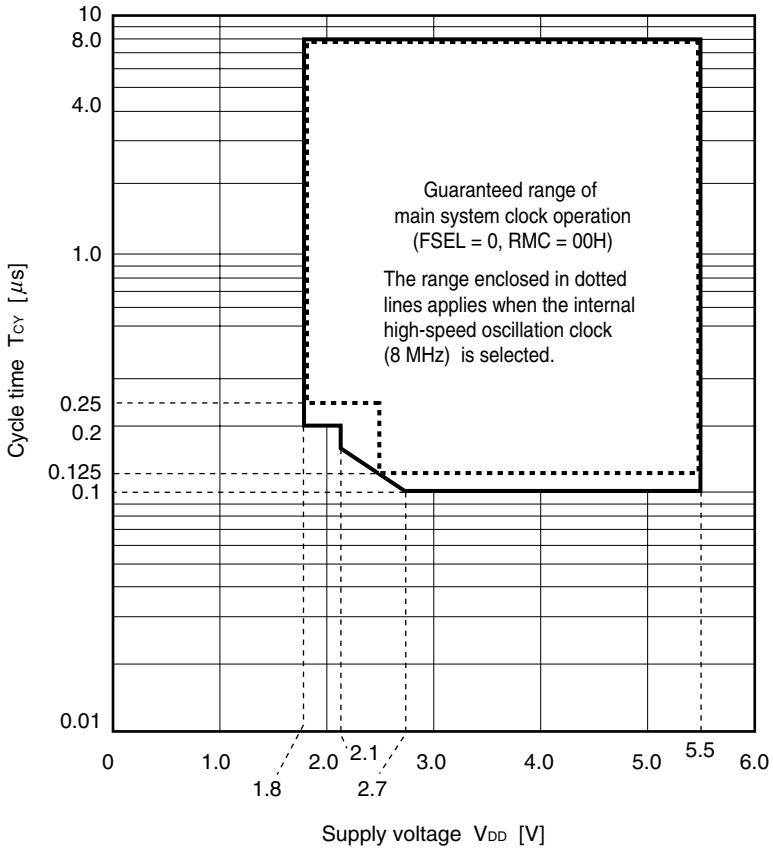
Note In low-power-consumption mode, use the regulator with f_{CLK} fixed to 1 MHz when executing self programming.

Remarks 1. f_{MCK}: Timer array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of the TMRmn register. m: Unit number (m = 0, 1), n: Channel number (n = 0 to 7))

2. For details on the normal power mode and low consumption power mode according to the regulator output voltage, refer to **CHAPTER 23 REGULATOR**.

(1) Basic operation (2/6)

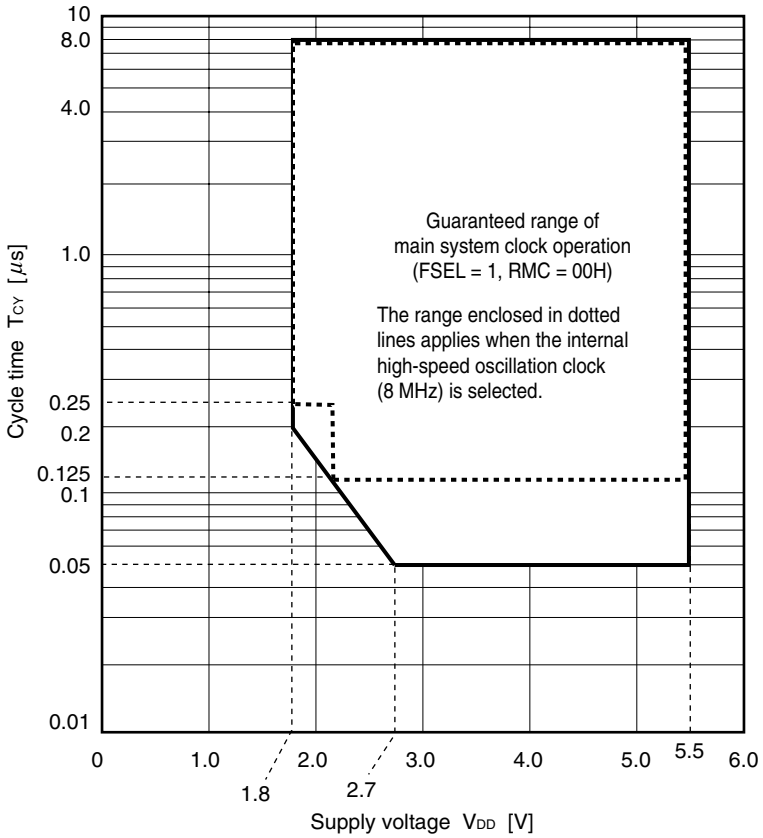
Minimum instruction execution time during main system clock operation (FSEL = 0, RMC = 00H)



Remark FSEL: Bit 0 of the operation speed mode control register (OSMC)

(1) Basic operation (3/6)

Minimum instruction execution time during main system clock operation (FSEL = 1, RMC = 00H)



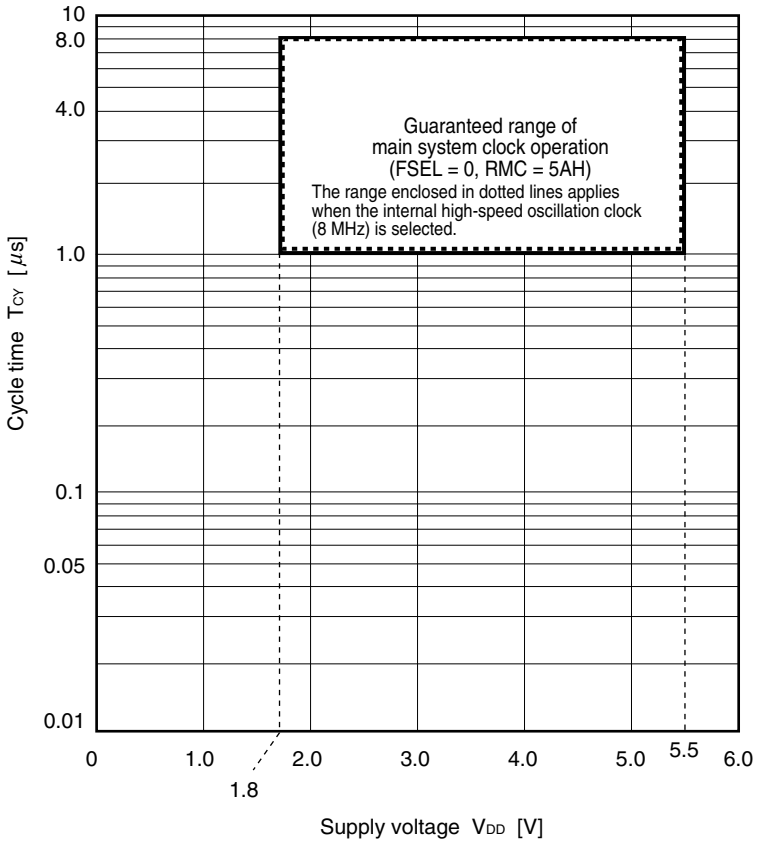
<R> **Caution When V_{DD} < 2.25 V and FSEL = 1, It is prohibited to release STOP mode during f_{EX} operation or f_{IH} operation (This must not be performed even if the frequency is divided. The STOP mode may be released during f_X operation.).**

Remarks 1. FSEL: Bit 0 of the operation speed mode control register (OSMC)

2. f_X: X1 clock oscillation frequency
f_{IH}: Internal high-speed oscillation clock frequency
f_{EX}: External main system clock frequency
f_{MAIN}: Main system clock frequency
f_{SUB}: Subsystem clock frequency
f_{CLK}: CPU/peripheral hardware clock frequency

(1) Basic operation (4/6)

Minimum instruction execution time during main system clock operation (FSEL = 0, RMC = 5AH)

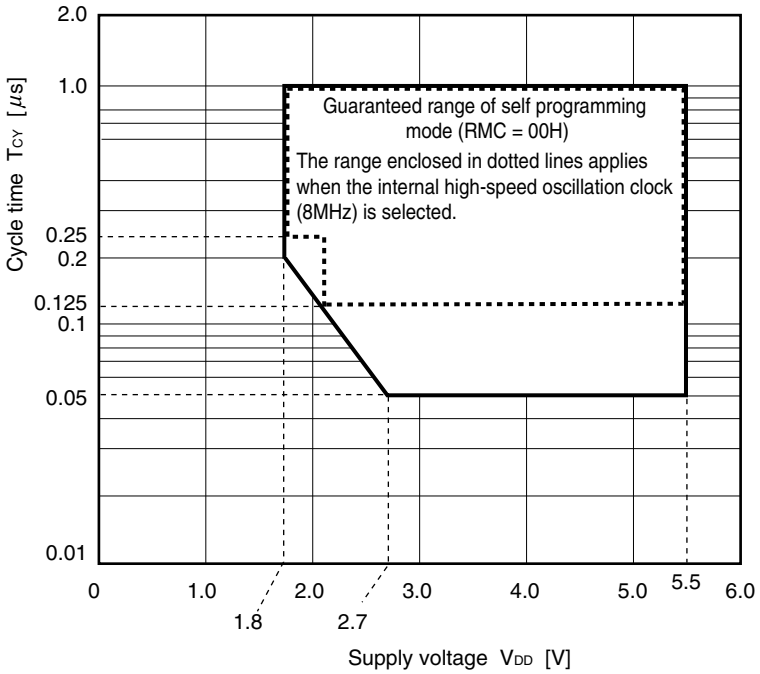


Remarks 1. FSEL: Bit 0 of the operation speed mode control register (OSMC)

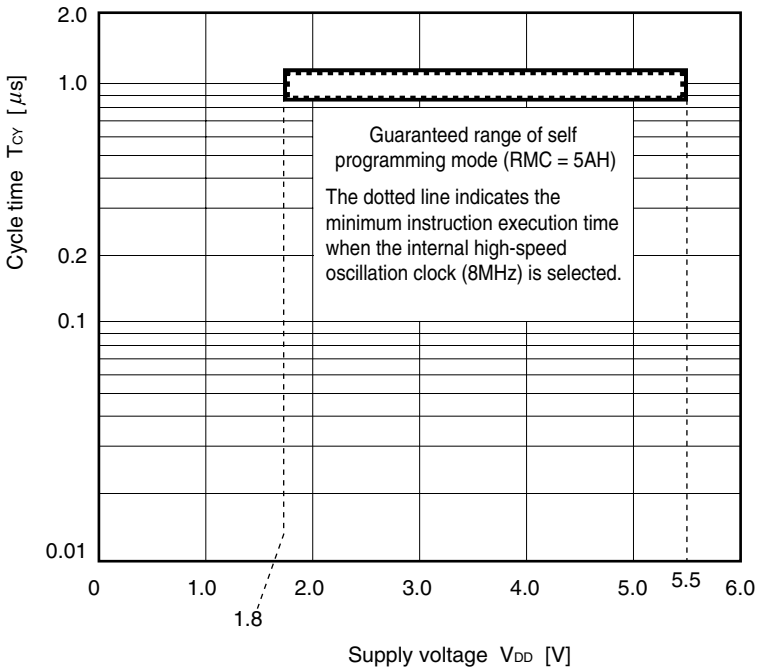
2. The entire voltage range is 1 MHz (MAX.) when RMC is set to 5AH.

(1) Basic operation (5/6)

Minimum instruction execution time during self programming mode (RMC = 00H)



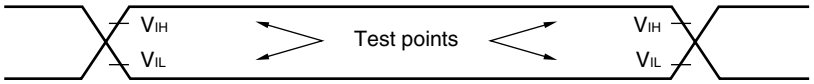
Minimum instruction execution time during self programming mode (RMC = 5AH)



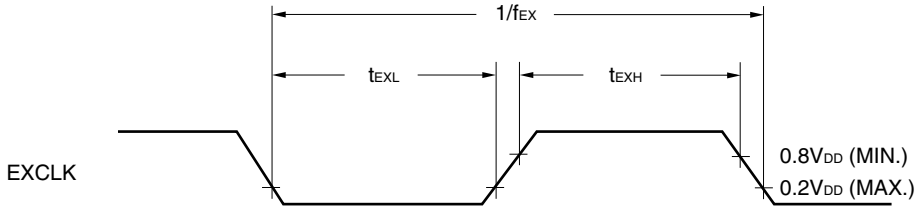
Remark The self programming function cannot be used when the CPU operates with the subsystem clock.

(1) Basic operation (6/6)

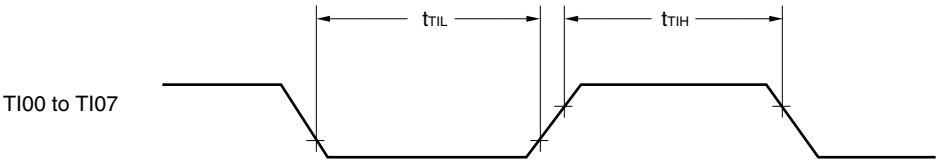
AC Timing Test Points



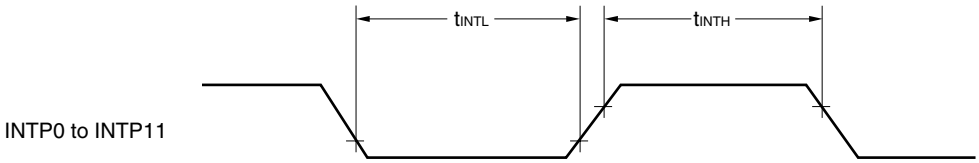
External Main System Clock Timing



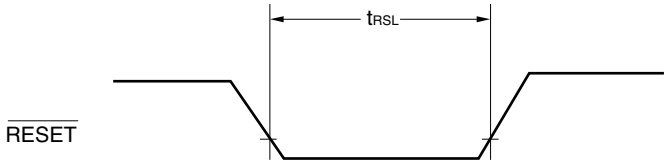
TI Timing



Interrupt Request Input Timing



$\overline{\text{RESET}}$ Input Timing



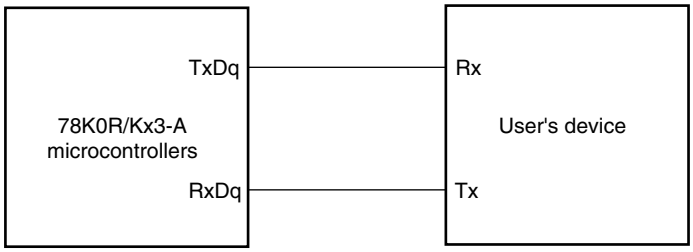
(2) Serial interface: Serial array unit (1/18)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AVss = 0 V)

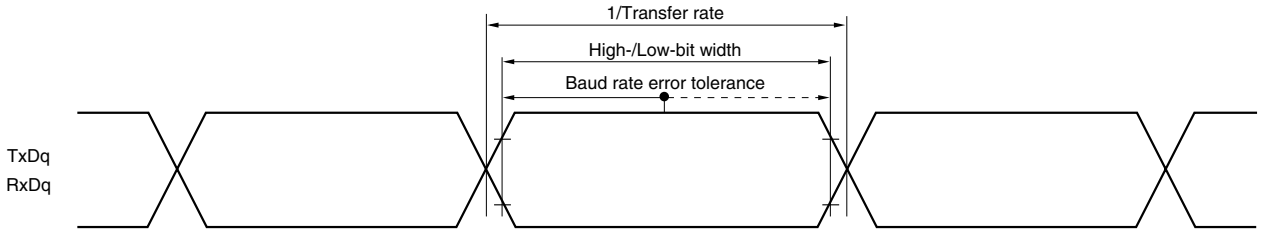
(a) During communication at same potential (UART mode) (dedicated baud rate generator output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate					f_{MCK}/6	bps
		f_{CLK} = 20 MHz, f_{MCK} = f_{CLK}			3.3	Mbps

UART mode connection diagram (during communication at same potential)



UART mode bit width (during communication at same potential) (reference)



Caution Select the normal input buffer for RxDq and the normal output mode for TxDq by using the PIMg and POMx registers.

Remarks 1. q: UART number (q = 0 to 3), g: PIM number (g = 1), x: POM number (x = 1, 8)

2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2))

(2) Serial interface: Serial array unit (2/18)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AVss = 0 V)

(b) During communication at same potential (CSI mode) (master mode, $\overline{SCKp}$... internal clock output)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
<R> $\overline{SCKp}$ cycle time	t_{KCY1}	4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V	200 Note 1			ns
		2.7 V ≤ V_{DD} = EV_{DD} < 4.0 V	300 Note 1			ns
		1.8 V ≤ V_{DD} = EV_{DD} < 2.7 V	600 Note 1			ns
$\overline{SCKp}$ high-/low-level width	t_{KH1}, t_{KL1}	4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V	t_{KCY1}/2 – 20			ns
		2.7 V ≤ V_{DD} = EV_{DD} < 4.0 V	t_{KCY1}/2 – 35			ns
		1.8 V ≤ V_{DD} = EV_{DD} < 2.7 V	t_{KCY1}/2 – 80			ns
SIp setup time (to $\overline{SCKp}$↑) Note 2	t_{SIK1}	4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V	70			ns
		2.7 V ≤ V_{DD} = EV_{DD} < 4.0 V	100			ns
		1.8 V ≤ V_{DD} = EV_{DD} < 2.7 V	190			ns
SIp hold time (from $\overline{SCKp}$↑) Note 3	t_{KSI1}		30			ns
Delay time from $\overline{SCKp}$↓ to SOp output Note 4	t_{KSO1}	C = 30 pF Note 5			40	ns

<R> **Notes 1.** The value must also be 4/f_{CLK} or more.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from $\overline{SCKp}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

5. C is the load capacitance of the $\overline{SCKp}$ and SOp output lines.

Caution Select the normal input buffer for SIp and the normal output mode for SOp and $\overline{SCKp}$ by using the PIMg and POMx registers.

Remarks 1. p: CSI number (p = 00, 10, 20), g: PIM number (g = 1), x: POM number (x = 1, 8)

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

(2) Serial interface: Serial array unit (3/18)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AVss = 0 V)

<R> **(c) During communication at same potential (CSI mode) (slave mode, $\overline{SCKp}$... external clock input)**

Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCKp}$ cycle time	t_{KCY2}	4.0 V ≤ V_{DD} ≤ 5.5 V		6/f_{MCK}			ns
		2.7 V ≤ V_{DD} < 4.0 V	16 MHz < f_{MCK}	8/f_{MCK}			ns
			f_{MCK} ≤ 16 MHz	6/f_{MCK}			ns
		1.8 V ≤ V_{DD} < 2.7 V	16 MHz < f_{MCK}	8/f_{MCK}			ns
			f_{MCK} ≤ 16 MHz	6/f_{MCK}			ns
$\overline{SCKp}$ high-/low-level width	t_{KH2}, t_{KL2}			t_{KCY2}/2			ns
SIp setup time (to $\overline{SCKp}$↑) Note 1	t_{SIK2}			80			ns
SIp hold time (from $\overline{SCKp}$↑) Note 2	t_{KSI2}			1/f_{MCK}+50			ns
Delay time from $\overline{SCKp}$↓ to SOp output Note 3	t_{KSO2}	C = 30 pF Note 4	4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V			2/f_{MCK}+45	ns
			2.7 V ≤ V_{DD} = EV_{DD} < 4.0 V			2/f_{MCK}+57	ns
			1.8 V ≤ V_{DD} = EV_{DD} < 2.7 V			2/f_{MCK}+125	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from $\overline{SCKp}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

4. C is the load capacitance of the $\overline{SCKp}$ and SOp output lines.

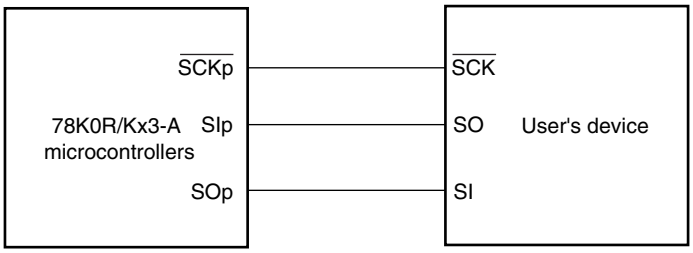
Caution Select the normal input buffer for SIp and $\overline{SCKp}$ and the normal output mode for SOp by using the PIMg and POMx registers.

Remarks 1. p: CSI number (p = 00, 10, 20), g: PIM number (g = 1), x: POM number (x = 1, 8)

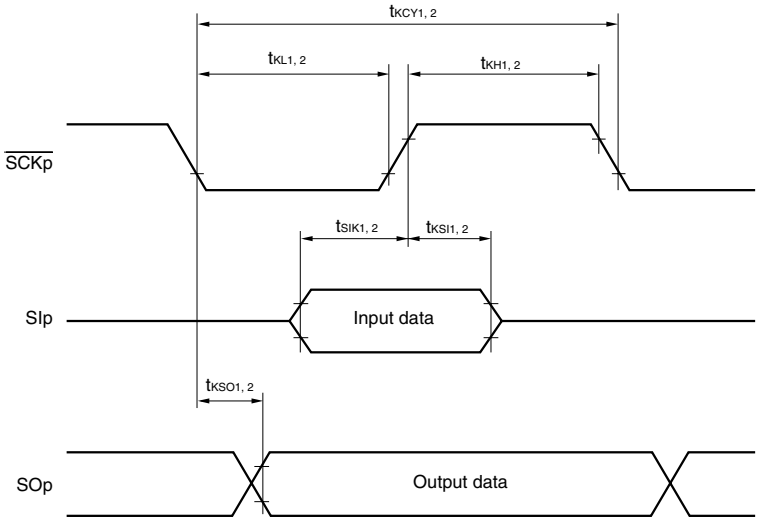
2. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2))

(2) Serial interface: Serial array unit (4/18)

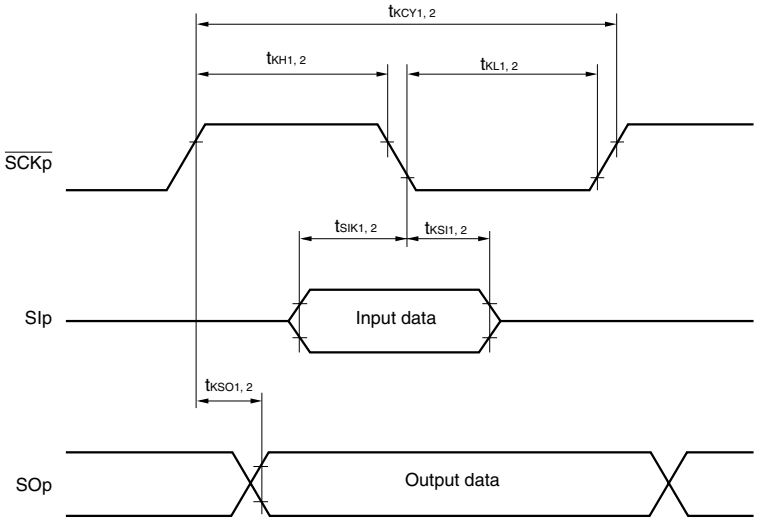
CSI mode connection diagram (during communication at same potential)



CSI mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



CSI mode serial transfer timing (during communication at same potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Remarks 1. p: CSI number (p = 00, 10, 20)

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

(2) Serial interface: Serial array unit (5/18)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AVss = 0 V)

<R> **(d) During communication at same potential (simplified I^2C mode)**

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCLr clock frequency	f_{SCL}	2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V R_b = 3 kΩ, C_b = 100 pF		400	kHz
		1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V R_b = 5 kΩ, C_b = 100 pF		300	kHz
Hold time when SCLr = “L”	t_{LOW}	2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V R_b = 3 kΩ, C_b = 100 pF	1200		ns
		1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V R_b = 5 kΩ, C_b = 100 pF	1500		ns
Hold time when SCLr = “H”	t_{HIGH}	2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V R_b = 3 kΩ, C_b = 100 pF	1200		ns
		1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V R_b = 5 kΩ, C_b = 100 pF	1500		ns
Data setup time (reception)	$t_{SU:DAT}$	2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V R_b = 3 kΩ, C_b = 100 pF	1/f_{MCK}+120		ns
		1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V R_b = 5 kΩ, C_b = 100 pF	1/f_{MCK}+230		ns
Data hold time (transmission)	$t_{HD:DAT}$	2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V R_b = 3 kΩ, C_b = 100 pF	0	660	ns
		1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V R_b = 5 kΩ, C_b = 100 pF	0	710	ns

Caution Select the normal input buffer and the N-ch open drain output (V_{DD} tolerance) mode for SDAr and the normal output mode for SCLr by using the PIMg and POMx registers.

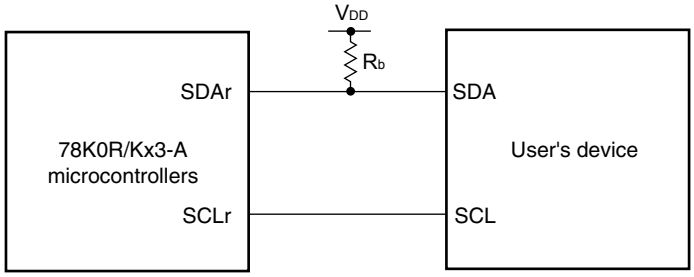
Remarks 1. R_b[Ω]:Communication line (SDAr) pull-up resistance,
C_b[F]: Communication line (SCLr, SDAr) load capacitance

2. r: IIC number (r = 10, 20), g: PIM number (g = 1), x: POM number (x = 1, 8)

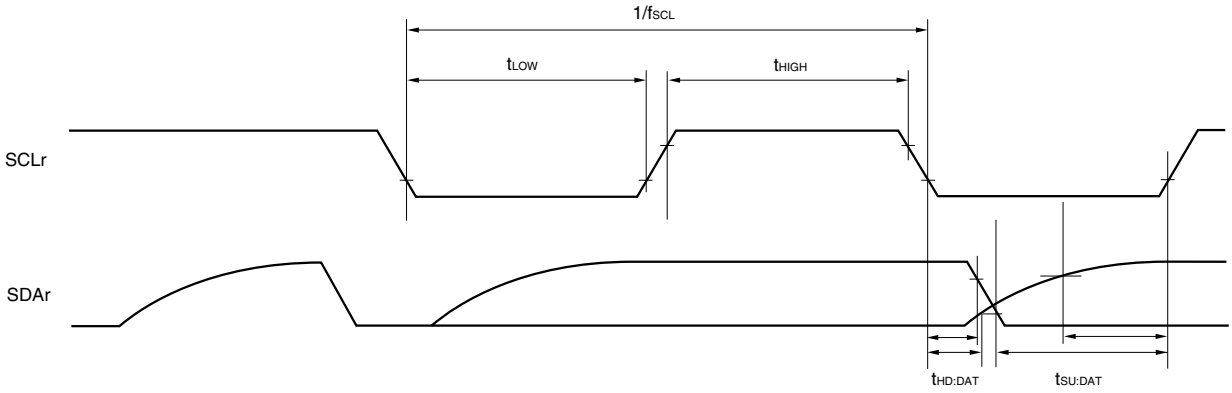
3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 02, 10)

(2) Serial interface: Serial array unit (6/18)

Simplified I^2C mode mode connection diagram (during communication at same potential)



Simplified I^2C mode serial transfer timing (during communication at same potential)



Remarks 1. R_b[Ω]:Communication line (SDAr) pull-up resistance,
C_b[F]: Communication line (SCLr, SDAr) load capacitance

2. r: IIC number (r = 10, 20)

(2) Serial interface: Serial array unit (7/18)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AVss = 0 V)

(e) Communication at different potential (2.5 V, 3 V) (UART mode) (dedicated baud rate generator output) (1/2)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Transfer rate		reception	4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V				f_{MCK}/6	bps
				f_{CLK} = 20 MHz, f_{MCK} = f_{CLK}			3.3	Mbps
			2.7 V ≤ V_{DD} = EV_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V				f_{MCK}/6	bps
				f_{CLK} = 20 MHz, f_{MCK} = f_{CLK}			3.3	Mbps

Caution Select the TTL input buffer for RxDq and the N-ch open drain output (V_{DD} tolerance) mode for TxDq by using the PIMg and POMx registers.

Remarks 1. q: UART number (q = 0 to 3) , g: PIM number (g = 1), x: POM number (x = 1, 8)

2. V_b[V]: Communication line voltage

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2))

4. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in UART mode.
4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V
2.7 V ≤ V_{DD} = EV_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

(2) Serial interface: Serial array unit (8/18)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AVss = 0 V)

(e) Communication at different potential (2.5 V, 3 V) (UART mode) (dedicated baud rate generator output) (2/2)

Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
Transfer rate		transmission	4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V				**Note 1**	bps
				f_{CLK} = 16.8 MHz, f_{MCK} = f_{CLK}, C_b = 50 pF, R_b = 1.4 kΩ, V_b = 2.7 V			2.8 Note 2	Mbps
			2.7 V ≤ V_{DD} = EV_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V				**Note 3**	bps
				f_{CLK} = 19.2 MHz, f_{MCK} = f_{CLK}, C_b = 50 pF, R_b = 2.7 kΩ, V_b = 2.3 V			1.2 Note 4	Mbps

Notes 1. The smaller maximum transfer rate derived by using f_{MCK}/6 or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V and 2.7 V ≤ V_b ≤ 4.0 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1-\frac{2.2}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1-\frac{2.2}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

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.

3. The smaller maximum transfer rate derived by using f_{MCK}/6 or the following expression is the valid maximum transfer rate.

Expression for calculating the transfer rate when 2.7 V ≤ V_{DD} = EV_{DD} < 4.0 V and 2.3 V ≤ V_b < 2.7 V

$$\text{Maximum transfer rate} = \frac{1}{\{-C_b \times R_b \times \ln(1-\frac{2.0}{V_b})\} \times 3} \text{ [bps]}$$

$$\text{Baud rate error (theoretical value)} = \frac{\frac{1}{\text{Transfer rate} \times 2} - \{-C_b \times R_b \times \ln(1-\frac{2.0}{V_b})\}}{(\frac{1}{\text{Transfer rate}}) \times \text{Number of transferred bits}} \times 100 \text{ [\%]}$$

* This value is the theoretical value of the relative difference between the transmission and reception sides.

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.

Caution Select the TTL input buffer for RxDq and the N-ch open drain output (V_{DD} tolerance) mode for TxDq by using the PIMg and POMx registers.

(Remarks are given on the next page.)

(2) Serial interface: Serial array unit (9/18)

Remarks 1. R_b[Ω]:Communication line (TxDq) pull-up resistance, C_b[F]: Communication line (TxDq) load capacitance, V_b[V]: Communication line voltage

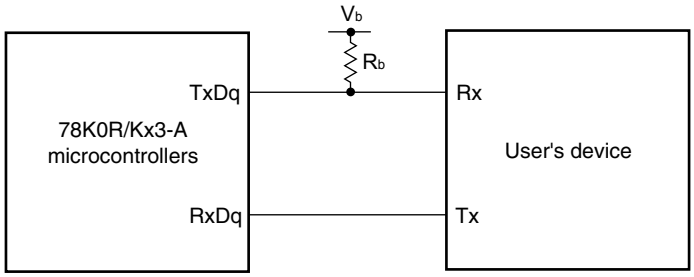
2. q: UART number (q = 0 to 3) , g: PIM number (g = 1), x: POM number (x = 1, 8)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2))

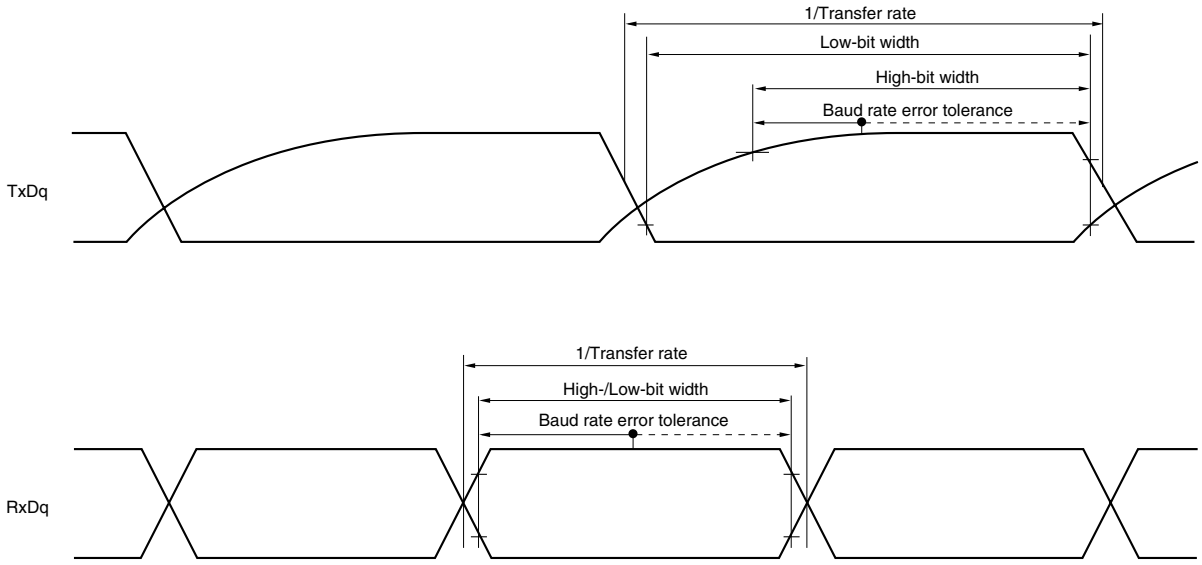
4. V_{OH} and V_{OL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in UART mode.
4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{OH} = 2.2 V, V_{OL} = 0.8 V
2.7 V ≤ V_{DD} = EV_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V: V_{OH} = 2.0 V, V_{OL} = 0.5 V

(2) Serial interface: Serial array unit (10/18)

UART mode connection diagram (communication at different potential)



UART mode bit width (communication at different potential) (reference)



Caution Select the TTL input buffer for RxDq and the N-ch open drain output (V_{DD} tolerance) mode for TxDq by using the PIMg and POMx registers.

Remarks 1. $R_b[\Omega]$:Communication line (TxDq) pull-up resistance, $V_b[V]$: Communication line voltage

2. q: UART number (q = 0 to 3) , g: PIM number (g = 1), x: POM number (x = 1, 8)

(2) Serial interface: Serial array unit (11/18)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AVss = 0 V)

<R> **(f) Communication at different potential (2.5 V, 3 V) (CSI mode) (master mode, $\overline{SCKp}$... internal clock output) (1/2)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
$\overline{SCKp}$ cycle time	t_{KCY1}	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	400 Note 1			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	800 Note 1			ns
$\overline{SCKp}$ 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 – 75			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 – 170			ns
$\overline{SCKp}$ 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 – 20			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 – 35			ns
SIp setup time (to $\overline{SCKp}$↑) Note 2	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	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	275			ns
SIp hold time (from $\overline{SCKp}$↑) Note 2	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	30			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	30			ns
Delay time from $\overline{SCKp}$↓ to SOp output Note 2	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			120	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			215	ns

<R> **Notes 1.** The value must also be 4/f_{CLK} or more.

2. When DAP0n = 0 and CKP0n = 0, or DAP0n = 1 and CKP0n = 1.

Caution Select the TTL input buffer for SIp and the N-ch open drain output (V_{DD} tolerance) mode for SOp and $\overline{SCKp}$ by using the PIMg and POMx registers.

Remarks 1. p: CSI number (p = 00, 10, 20), g: PIM number (g = 1), x: POM number (x = 1, 8)

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

3. R_b[Ω]:Communication line ($\overline{SCKp}$, SOp) pull-up resistance,
C_b[F]: Communication line (SIp, SOp, $\overline{SCKp}$) load capacitance, V_b[V]: Communication line voltage

4. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in CSI mode.
4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V
2.7 V ≤ V_{DD} = EV_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

(2) Serial interface: Serial array unit (12/18)

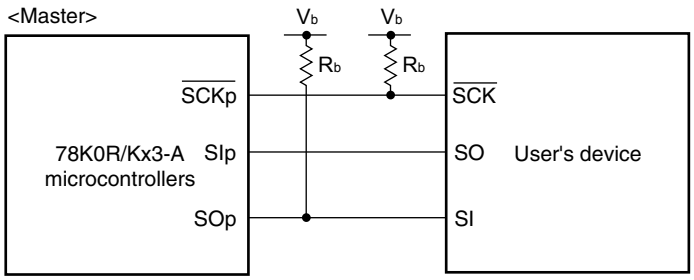
(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AVss = 0 V)

<R> **(f) Communication at different potential (2.5 V, 3 V) (CSI mode) (master mode, $\overline{SCKp}$... internal clock output) (2/2)**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
SIp setup time (to $\overline{SCKp}$↓) Note	t_{SIK1}	4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	70			ns
		2.7 V ≤ V_{DD} = EV_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	100			ns
SIp hold time (from $\overline{SCKp}$↓) Note	t_{KSI1}	4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ	30			ns
		2.7 V ≤ V_{DD} = EV_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ	30			ns
Delay time from $\overline{SCKp}$↑ to SOp output Note	t_{KSO1}	4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, C_b = 30 pF, R_b = 1.4 kΩ			40	ns
		2.7 V ≤ V_{DD} = EV_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, C_b = 30 pF, R_b = 2.7 kΩ			40	ns

Note When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

CSI mode connection diagram (communication at different potential)



Caution Select the TTL input buffer for SIp and the N-ch open drain output (V_{DD} tolerance) mode for SOp and $\overline{SCKp}$ by using the PIMg and POMx registers.

Remarks 1. p: CSI number (p = 00, 10, 20), g: PIM number (g = 1), x: POM number (x = 1, 8)

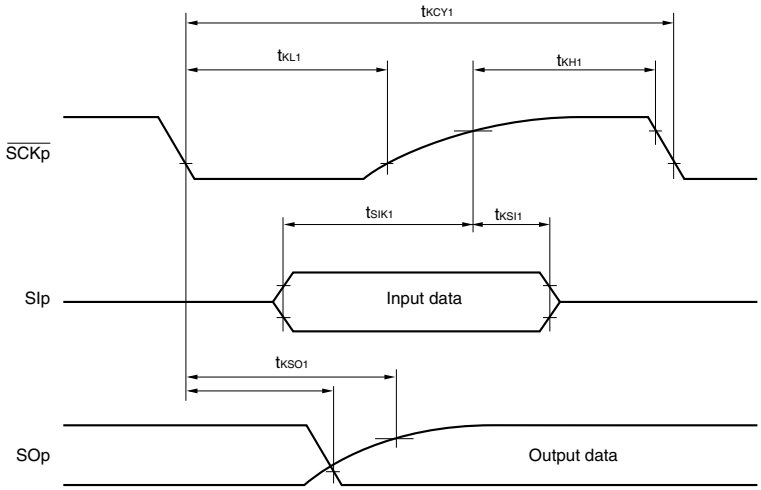
2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

3. R_b[Ω]:Communication line ($\overline{SCKp}$, SOp) pull-up resistance,
C_b[F]: Communication line (SIp, SOp, $\overline{SCKp}$) load capacitance, V_b[V]: Communication line voltage

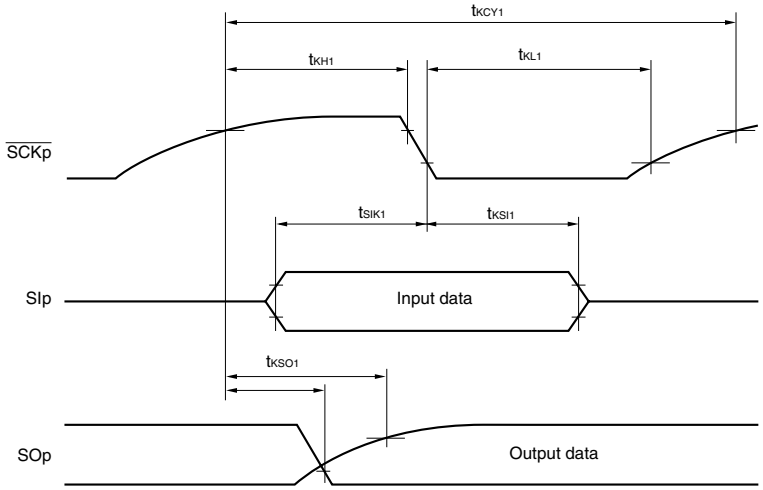
4. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in CSI mode.
4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V
2.7 V ≤ V_{DD} = EV_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

(2) Serial interface: Serial array unit (13/18)

CSI mode serial transfer timing (communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



CSI mode serial transfer timing (communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Caution Select the TTL input buffer for SIp and the N-ch open drain output (V_{DD} tolerance) mode for SOp and $\overline{SCKp}$ by using the PIMg and POMx registers.

Remarks 1. p: CSI number (p = 00, 10, 20), g: PIM number (g = 1), x: POM number (x = 1, 8)

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

(2) Serial interface: Serial array unit (14/18)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AVss = 0 V)

<R> **(g) Communication at different potential (2.5 V, 3 V) (CSI mode) (slave mode, $\overline{SCKp}$... external clock input)**

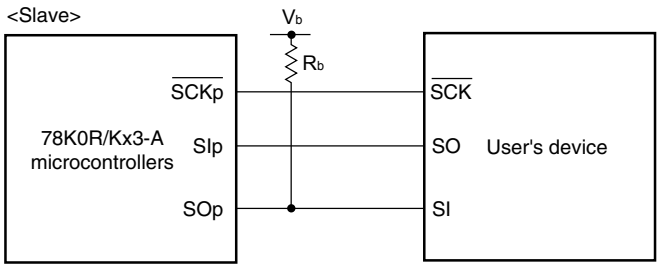
Parameter	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
$\overline{SCKp}$ cycle time	t_{KCY2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V	13.6 MHz < f_{MCK}	10/f_{MCK}			ns
			6.8 MHz < f_{MCK} ≤ 13.6 MHz	8/f_{MCK}			ns
			f_{MCK} ≤ 6.8 MHz	6/f_{MCK}			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V	18.5 MHz < f_{MCK}	16/f_{MCK}			ns
			14.8 MHz < f_{MCK} ≤ 18.5 MHz	14/f_{MCK}			ns
			11.1 MHz < f_{MCK} ≤ 14.8 MHz	12/f_{MCK}			ns
			7.4 MHz < f_{MCK} ≤ 11.1 MHz	10/f_{MCK}			ns
			3.7 MHz < f_{MCK} ≤ 7.4 MHz	8/f_{MCK}			ns
			f_{MCK} ≤ 3.7 MHz	6/f_{MCK}			ns
$\overline{SCKp}$ high-/low-level width	t_{KH2}, t_{KL2}	4.0 V ≤ V_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V		f_{KCY2}/2 – 20			ns
		2.7 V ≤ V_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V		f_{KCY2}/2 – 35			ns
SIp setup time (to $\overline{SCKp}$↑)[Note 1]	t_{SIK2}			90			ns
SIp hold time (from $\overline{SCKp}$↑)[Note 2]	t_{KSI2}			1/f_{MCK} + 50			ns
Delay time from $\overline{SCKp}$↓ 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} + 120	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} + 230	ns

Notes 1. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp setup time becomes "to $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

2. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The SIp hold time becomes "from $\overline{SCKp}$↓" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

3. When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1. The delay time to SOp output becomes "from $\overline{SCKp}$↑" when DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.

CSI mode connection diagram (communication at different potential)



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

(2) Serial interface: Serial array unit (15/18)

Caution Select the TTL input buffer for SIp and $\overline{SCKp}$ and the N-ch open drain output (V_{DD} tolerance) mode for SOp by using the PIMg and POMx registers.

Remarks 1. p: CSI number (p = 00, 10, 20), g: PIM number (g = 1), x: POM number (x = 1, 8)

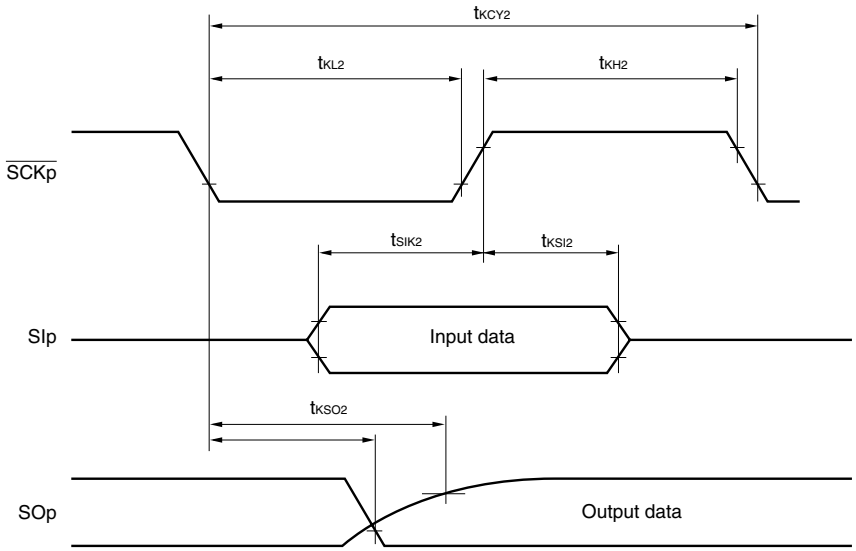
2. $R_b[\Omega]$:Communication line (SOp) pull-up resistance,
$C_b[F]$: Communication line (SOp, $\overline{SCKp}$) load capacitance, $V_b[V]$: Communication line voltage

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2))

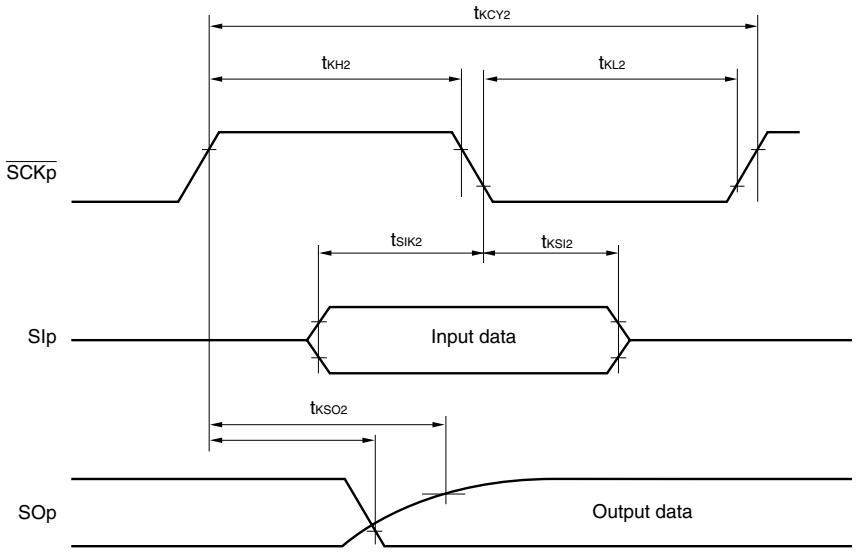
4. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in CSI mode.
$4.0\ V \le V_{DD} = EV_{DD} \le 5.5\ V$, $2.7\ V \le V_b \le 4.0\ V$: $V_{IH} = 2.2\ V$, $V_{IL} = 0.8\ V$
$2.7\ V \le V_{DD} = EV_{DD} < 4.0\ V$, $2.3\ V \le V_b < 2.7\ V$: $V_{IH} = 2.0\ V$, $V_{IL} = 0.5\ V$

(2) Serial interface: Serial array unit (16/18)

CSI mode serial transfer timing (communication at different potential)
(When DAPmn = 0 and CKPmn = 0, or DAPmn = 1 and CKPmn = 1.)



CSI mode serial transfer timing (communication at different potential)
(When DAPmn = 0 and CKPmn = 1, or DAPmn = 1 and CKPmn = 0.)



Caution Select the TTL input buffer for SIp and $\overline{\text{SCKp}}$ and the N-ch open drain output (V_{DD} tolerance) mode for SOp by using the PIMg and POMx registers.

Remarks 1. p: CSI number (p = 00, 10, 20), g: PIM number (g = 1), x: POM number (x = 1, 8)

2. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2)

(2) Serial interface: Serial array unit (17/18)

(T_A = –40 to +85°C, 2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AVss = 0 V)

<R> **(h) Communication at different potential (2.5 V, 3 V) (simplified I^2C mode)**

Parameter	Symbol	Conditions	MIN.	MAX.	Unit
SCLr clock frequency	f_{SCL}	4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, R_b = 1.4 kΩ, C_b = 100 pF		400	kHz
		2.7 V ≤ V_{DD} = EV_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, R_b = 2.7 kΩ, C_b = 100 pF		400	kHz
Hold time when SCLr = “L”	t_{LOW}	4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, R_b = 1.4 kΩ, C_b = 100 pF	1275		ns
		2.7 V ≤ V_{DD} = EV_{DD} < 4.0 V, 2.3 V ≤ V_b ≤ 2.7 V, R_b = 2.7 kΩ, C_b = 100 pF	1275		ns
Hold time when SCLr = “H”	t_{HIGH}	4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, R_b = 1.4 kΩ, C_b = 100 pF	655		ns
		2.7 V ≤ V_{DD} = EV_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, R_b = 2.7 kΩ, C_b = 100 pF	655		ns
Data setup time (reception)	$t_{SU:DAT}$	4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, R_b = 1.4 kΩ, C_b = 100 pF	1/f_{MCK} + 190		ns
		2.7 V ≤ V_{DD} = EV_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, R_b = 2.7 kΩ, C_b = 100 pF	1/f_{MCK} + 190		ns
Data hold time (transmission)	$t_{HD:DAT}$	4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V, R_b = 1.4 kΩ, C_b = 100 pF	0	640	ns
		2.7 V ≤ V_{DD} = EV_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V, R_b = 2.7 kΩ, C_b = 100 pF	0	660	ns

Caution Select the TTL input buffer and the N-ch open drain output (V_{DD} tolerance) mode for SDAr and the N-ch open drain output (V_{DD} tolerance) mode for SCLr by using the PIMg and POMx registers.

Remarks 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

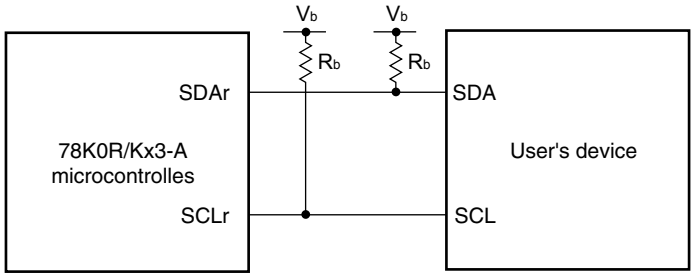
2. r: IIC number (r = 10, 20), g: PIM number (g = 1), x: POM number (x = 1, 8)

3. f_{MCK}: Serial array unit operation clock frequency
(Operation clock to be set by the CKSmn bit of the SMRmn register. m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 02, 10)

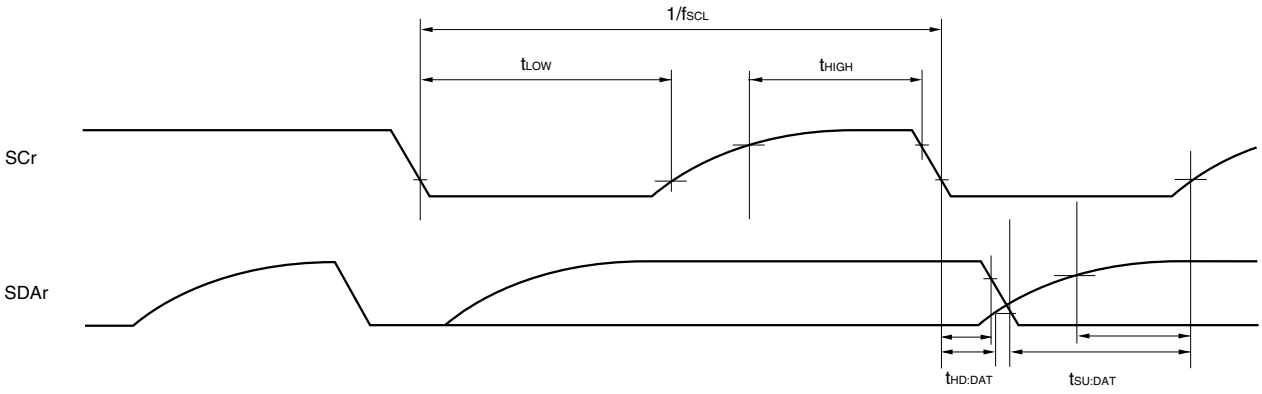
4. V_{IH} and V_{IL} below are observation points for the AC characteristics of the serial array unit when communicating at different potentials in simplified I^2C mode mode.
4.0 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, 2.7 V ≤ V_b ≤ 4.0 V: V_{IH} = 2.2 V, V_{IL} = 0.8 V
2.7 V ≤ V_{DD} = EV_{DD} < 4.0 V, 2.3 V ≤ V_b < 2.7 V: V_{IH} = 2.0 V, V_{IL} = 0.5 V

(2) Serial interface: Serial array unit (18/18)

Simplified I²C mode connection diagram (communication at different potential)



Simplified I²C mode serial transfer timing (communication at different potential)



Caution Select the TTL input buffer and the N-ch open drain output (V_{DD} tolerance) mode for SDAr and the N-ch open drain output (V_{DD} tolerance) mode for SCLr by using the PIMg and POMx registers.

Remarks 1. R_b[Ω]:Communication line (SDAr, SCLr) pull-up resistance, V_b[V]: Communication line voltage

2. r: IIC number (r = 10, 20), g: PIM number (g = 1), x: POM number (x = 1, 8)

(3) Serial interface: IICA

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AVss = 0 V)

(a) IICA

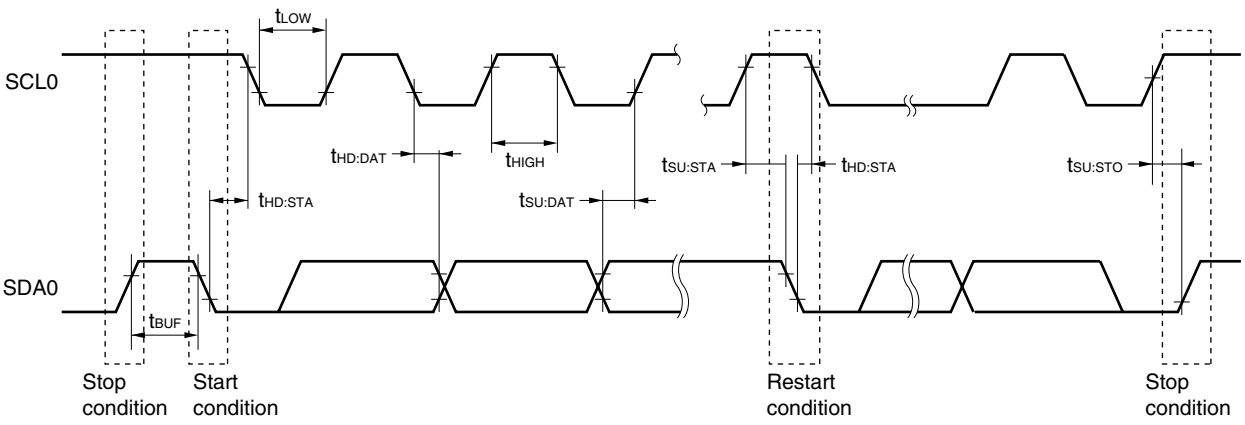
Parameter	Symbol	Conditions	Standard Mode		Fast Mode		Unit
			MIN.	MAX.	MIN.	MAX.	
SCL0 clock frequency	f_{SCL}	Fast mode: f_{CLK} ≥ 3.5 MHz, Standard mode: f_{CLK} ≥ 1 MHz	0	100	0	400	kHz
Setup time of restart condition[Note 1]	$t_{SU:STA}$		4.7		0.6		μs
Hold time	$t_{HD:STA}$		4.0		0.6		μs
Hold time when SCL0 = “L”	t_{LOW}		4.7		1.3		μs
Hold time when SCL0 = “H”	t_{HIGH}		4.0		0.6		μs
Data setup time (reception)	$t_{SU:DAT}$		250		100		ns
Data hold time (transmission)[Note 2]	$t_{HD:DAT}$		0	3.45	0	0.9	μs
Setup time of stop condition	$t_{SU:STO}$		4.0		0.6		μs
Bus-free time	t_{BUF}		4.7		1.3		μs

Notes 1. The first clock pulse is generated after this period when the start/restart condition is detected.

2. The maximum value (MAX.) of $t_{HD:DAT}$ is during normal transfer and a wait state is inserted in the $\overline{ACK}$ (acknowledge) timing.

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

IICA serial transfer timing



(4) Serial interface: On-chip debug (UART)

(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AVss = 0 V)

(a) On-chip debug (UART)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Transfer rate			$f_{CLK}/2^{12}$		$f_{CLK}/6$	bps
		Flash memory programming mode (f_{CLK} = 20 MHz, 2.7 V ≤ V_{DD} = EV_{DD}, C_b = 50 pF)			3.33	Mbps
TOOL1 output frequency	f_{TOOL1}	2.7 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V			10	MHz
		1.8 V ≤ V_{DD} = EV_{DD} < 2.7 V			2.5	MHz

Analog Characteristics

(1) A/D Converter

(a) T_A = 0 to 50°C, 1.8 V ≤ AD_{REFP} ≤ AV_{DD0}, 2.3 V ≤ AV_{DD0} ≤ V_{DD} ≤ 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = AD_{REFM} = 0 V

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	RES		12	12	12	bit
Overall error[Note]	AINL	2.3 V ≤ AD_{REFP} ≤ 3.6 V		±2.0	±6.0	LSB
		1.8 V ≤ AD_{REFP} < 2.3 V		±3.0	±6.0	LSB
<R> Conversion time	t_{CONV}	Normal mode 1, Normal mode 2	5		50	μs
		Low voltage mode	6.25		50	μs
Zero-scale error[Note]	E_{ZS}			±2.0	±4.0	LSB
Full-scale error[Note]	E_{FS}			±2.0	±4.0	LSB
Integral non-linearity error[Note]	ILE				±2.0	LSB
Differential non-linearity error[Note]	DLE				±1.0	LSB
Reference voltage (high potential side)	AD_{REFP}		1.8		AV_{DD0}	V
Analog input voltage	V_{AIN}		AD_{REFM}		AD_{REFP}	V
Reference supply current	I_{REF}			46	200	μA

<R> **(b) T_A = –40 to +85°C , 1.8 V ≤ AD_{REFP} ≤ AV_{DD0}, 1.8 V ≤ AV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = AD_{REFM} = 0 V**

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	RES		12	12	12	bit
Overall error[Note]	AINL	3.6 V ≤ AD_{REFP} ≤ 5.5 V		±2.0	±10.0	LSB
		2.3 V ≤ AD_{REFP} < 3.6 V		±2.0	±10.0	LSB
		1.8 V ≤ AD_{REFP} < 2.3 V		±3.0	±10.0	LSB
Conversion time	t_{CONV}	Normal mode 1, Normal mode 2	5		50	μs
		Low voltage mode	21		50	μs
Zero-scale error[Note]	E_{ZS}			±2.0	±8.0	LSB
Full-scale error[Note]	E_{FS}			±2.0	±8.0	LSB
Integral non-linearity error[Note]	ILE				±6.0	LSB
Differential non-linearity error[Note]	DLE				±2.0	LSB
Reference voltage (high potential side)	AD_{REFP}		1.8		AV_{DD0}	V
Analog input voltage	V_{AIN}		AD_{REFM}		AD_{REFP}	V
Reference supply current	I_{REF}			46	220	μA

Note Excludes quantization error (±1/2 LSB).

Remarks 1. AD_{REFP} is the input voltage from the AV_{REFP} pin or the voltage generated by the voltage reference.

2. AD_{REFM} is the input voltage from the AV_{REFM} pin or the grand potential of A/D converter.

(2) Operational amplifier

(T_A = –40 to +85°C, 2.3 V ≤ AV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} =AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Common-mode input voltage	V_{IAMP}	AV_{DD0} = 3.0 V	0		AV_{DD0} –0.6	V
Input offset voltage	V_{IOAMP}				±10	mV
Maximum output voltage (high level)	V_{OHAMP}	AV_{DD0} = 3.0 V/2.3 V, I_{SOURCE} = –500 μA	AV_{DD0} –0.2			V
<R> Maximum output voltage (low level)	V_{OLAMP}	AV_{DD0} = 3.0 V/2.3 V, I_{SOURCE} = 500 μA			0.1	V
Open-loop gain		AV_{DD0} = 3.0 V		100		dB
GBW	GBW	AV_{DD0} = 3.0 V		3		MHz
Input noise spectral density	V_{NAMP}	AV_{DD0} = 3.0 V, V_{IN} = AV_{DD0}/2		60		nV/$\sqrt{Hz}$
Slew rate	SR_{AMP}	AV_{DD0} = 3.0 V		2		V/μs
Turn on time	t_{ONAMP}				20	μs

<R> (3) Voltage Reference

(T_A = –40 to +85°C, 2.3 V ≤ AV_{DD0} ≤ V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} =AV_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Output reference voltage	V_{REFOUT}	VRGV = 0, 2.7 V ≤ AV_{DD0} ≤ 5.5 V, T_A = 25°C	2.45	2.5	2.55	V
		VRGV = 1, 2.3 V ≤ AV_{DD0} ≤ 5.5 V, T_A = 25°C	1.96	2	2.04	V
Temperature coefficient				40		ppm/°C
Settling time					17	ms

Caution Connect the V_{REFOUT} pin to GND via a tantalum capacitor (capacitance: 10 μF±30 %, ESR: 2 Ω (max.), ESL: 10 nH (max.)) and a ceramic capacitor (capacitance: 0.1 μF±30 %, ESR: 2 Ω (max.), ESL: 10 nH (max.)).

Remark The settling time of the VR circuit is the time required until the reference voltage output voltage reaches the values above.

(4) D/A Converter

(a) T_A = 0 to 50°C, 1.8 V ≤ DA_{REFP} ≤ AV_{DD1}, 2.3 V ≤ AV_{DD1} = V_{DD} ≤ 3.6 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	RES		12	12	12	bit
Settling time	t_{SET}				18	μs
Off-set error	E_O			±5	±10	mV
<R> Gain error	E_G			±5	±10	mV
Integral non-linearity error	ILE	I_{SOURCE} = I_{SINK} = 0 mA, 0.1 V ≤ ANOn ≤ AV_{DD1}–0.1 V (n = 0, 1)		±2.0	±4.0	LSB
Differential non-linearity error	DLE				±2.0	LSB
D/A output resistance value	R_O	0 V ≤ ANOn ≤ 0.3 V or AV_{DD1}–0.3 V ≤ ANOn ≤ AV_{DD1} (n = 0, 1)		150	250	Ω
		0.3 V ≤ ANOn ≤ AV_{DD1}–0.3 V (n = 0, 1)		5	10	Ω
Output source current	I_{SOURCE}	0.3 V ≤ ANOn ≤ AV_{DD1}–0.3 V (n = 0, 1)			0.1	mA
Output sink current	I_{SINK}				0.1	mA

(b) T_A = –40 to +85°C, 1.8 V ≤ DA_{REFP} ≤ AV_{DD1}, 2.3 V ≤ AV_{DD1} = V_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = AV_{SS} = 0 V

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution	RES		12	12	12	bit
Settling time	t_{SET}				18	μs
Off-set error	E_O			±5	±20	mV
<R> Gain error	E_G			±5	±20	mV
Integral non-linearity error	ILE	I_{SOURCE} = I_{SINK} = 0 mA, 0.1 V ≤ ANOn ≤ AV_{DD1}–0.1 V (n = 0, 1)		±6.0	±12.0	LSB
Differential non-linearity error	DLE				±8.0	LSB
D/A output resistance value	R_O	0 V ≤ ANOn ≤ 0.3 V or AV_{DD1}–0.3 V ≤ ANOn ≤ AV_{DD1} (n = 0, 1)		150	250	Ω
		0.3 V ≤ ANOn ≤ AV_{DD1}–0.3 V (n = 0, 1)		5	20	Ω
Output source current	I_{SOURCE}	0.3 V ≤ ANOn ≤ AV_{DD1}–0.3 V (n = 0, 1)			0.1	mA
Output sink current	I_{SINK}				0.1	mA

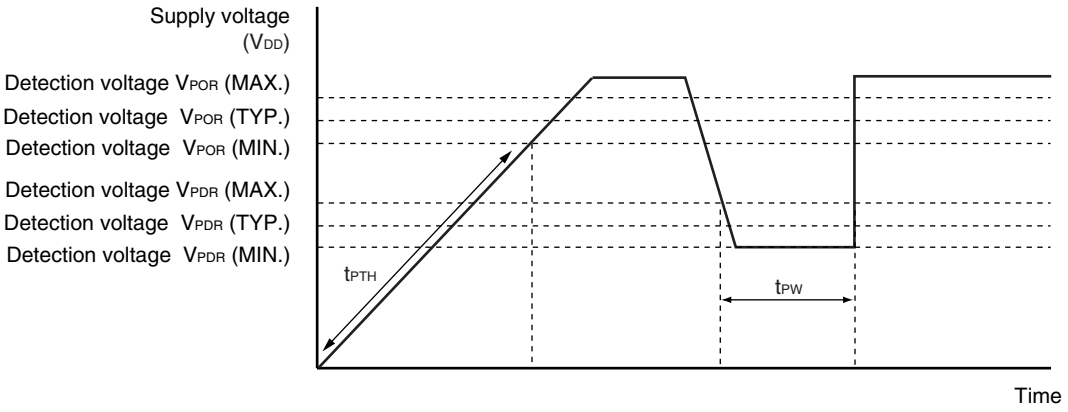
Remarks 1. Use the D/A converter under the condition of the output load capacitance (C) = 50 pF (max.).

2. DA_{REFP} is the input voltage from the AV_{REFP} pin, the voltage generated by the voltage reference, or the input voltage from the AV_{DD1} pin. It is selected as the positive reference voltage of the D/A converter.

POC Circuit Characteristics (T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	V_{POR}		1.52	1.61	1.70	V
	V_{PDR}		1.5	1.59	1.68	V
Power supply voltage rise inclination	t_{PTH}	Change inclination of V_{DD}: 0 V → V_{POR}	0.5			V/ms
Minimum pulse width	t_{PW}	When the voltage drops	200			μs
Detection delay time					200	μs

POC Circuit Timing



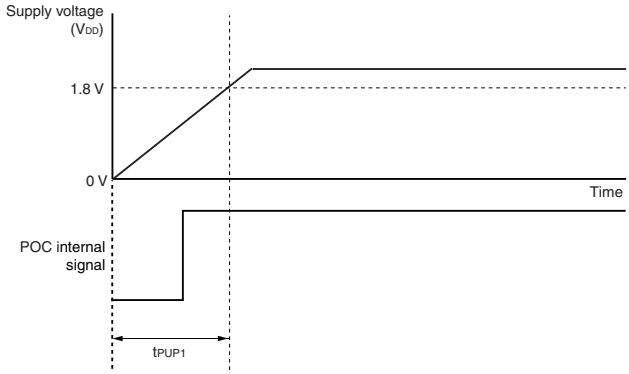
Supply Voltage Rise Time (T_A = –40 to +85°C, V_{SS} = 0 V)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Maximum time to rise to 1.8 V (V_{DD} (MIN.))[Note] (V_{DD}: 0 V → 1.8 V)	t_{PUP1}	LVI default start function stopped is set (LVIOFF (Option Byte) = 1), when $\overline{\text{RESET}}$ input is not used			3.6	ms
Maximum time to rise to 1.8 V (V_{DD} (MIN.))[Note] (releasing $\overline{\text{RESET}}$ input → V_{DD}: 1.8 V)	t_{PUP2}	LVI default start function stopped is set (LVIOFF (Option Byte) = 1), when $\overline{\text{RESET}}$ input is used			1.88	ms

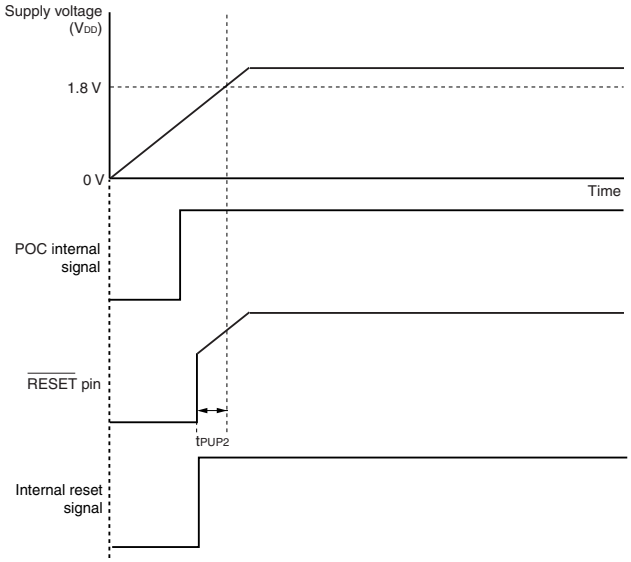
Note Make sure to raise the power supply in a shorter time than this.

Supply Voltage Rise Time Timing

- When $\overline{\text{RESET}}$ pin input is not used



- When $\overline{\text{RESET}}$ pin input is used (when external reset is released by the $\overline{\text{RESET}}$ pin, after POC has been released)



LVI Circuit Characteristics (T_A = –40 to +85°C, $V_{PDR} \le V_{DD} = EV_{DD} \le 5.5$ V, $V_{SS} = EV_{SS}$ = 0 V)

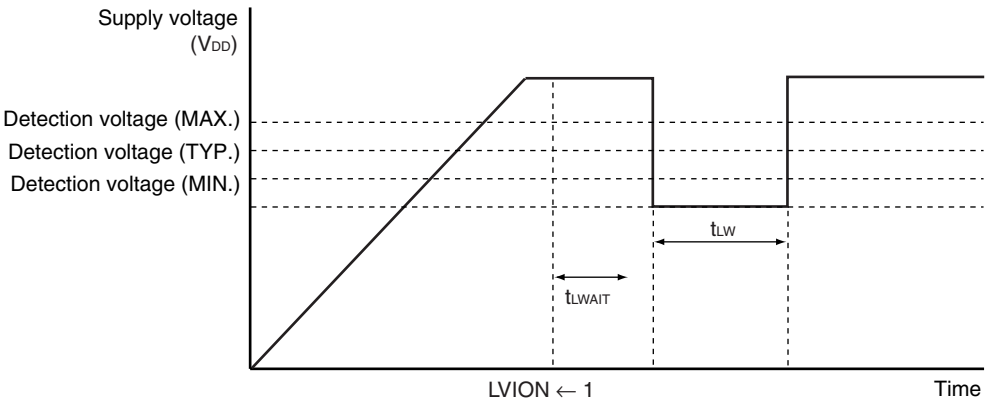
Parameter		Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Detection voltage	Supply voltage level	V_{LVI0}		4.12	4.22	4.32	V
		V_{LVI1}		3.97	4.07	4.17	V
		V_{LVI2}		3.82	3.92	4.02	V
		V_{LVI3}		3.66	3.76	3.86	V
		V_{LVI4}		3.51	3.61	3.71	V
		V_{LVI5}		3.35	3.45	3.55	V
		V_{LVI6}		3.20	3.30	3.40	V
		V_{LVI7}		3.05	3.15	3.25	V
		V_{LVI8}		2.89	2.99	3.09	V
		V_{LVI9}		2.74	2.84	2.94	V
		V_{LVI10}		2.58	2.68	2.78	V
		V_{LVI11}		2.43	2.53	2.63	V
		V_{LVI12}		2.28	2.38	2.48	V
		V_{LVI13}		2.12	2.22	2.32	V
		V_{LVI14}		1.97	2.07	2.17	V
		V_{LVI15}		1.81	1.91	2.01	V
	External input pin[Note 1]	V_{EXLVI}	EXLVI < V_{DD}, 1.8 V ≤ V_{DD} ≤ 5.5 V	1.11	1.21	1.31	V
	Power supply voltage on power application	V_{PUPLVI}	When LVI default start function enabled is set	1.87	2.07	2.27	V
Minimum pulse width		t_{LW}		200			μs
Detection delay time						200	μs
Operation stabilization wait time[Note 2]		t_{LWAIT}				10	μs

Notes 1. The EXLVI/P120/INTP0 pin is used.

2. Time required from setting bit 7 (LVION) of the low-voltage detection register (LVIM) to 1 to operation stabilization

Remark $V_{LVI(n-1)} > V_{LVIn}$: n = 1 to 15

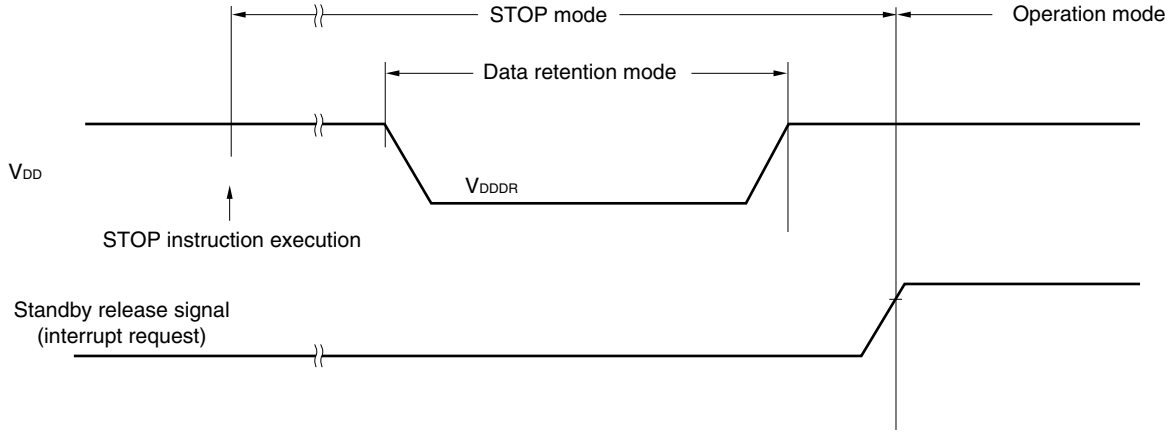
LVI Circuit Timing



Data Memory STOP Mode Low Supply Voltage Data Retention Characteristics (T_A = –40 to +85°C)

Parameter	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Data retention supply voltage	V_{DDDR}		1.5[Note]		5.5	V

Note The value depends on the POC detection voltage. When the voltage drops, the data is retained until a POC reset is effected, but data is not retained when a POC reset is effected.



Flash Memory Programming Characteristics

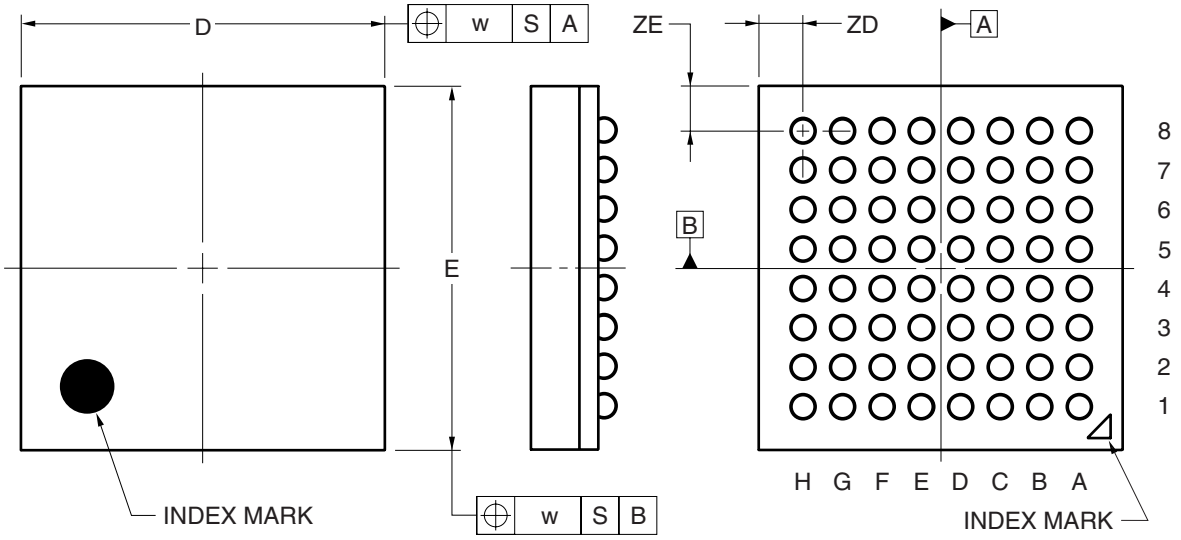
(T_A = –40 to +85°C, 1.8 V ≤ V_{DD} = EV_{DD} ≤ 5.5 V, V_{SS} = EV_{SS} = 0 V)

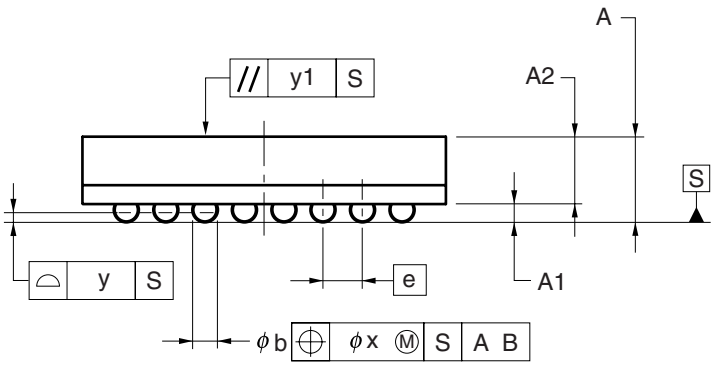
Parameter	Symbol	Conditions			MIN.	TYP.	MAX.	Unit
V_{DD} supply current	I_{DD}	Typ. = 10 MHz, Max. = 20 MHz				6	20	mA
Number of rewrites per chip	C_{erwr}	1 erase + 1 write after erase = 1 rewrite[Note]	When a flash memory programmer is used, and the libraries provided by Renesas Electronics are used	Retention: 15 years	1000			Times
			When the EEPROM emulation libraries provided by Renesas Electronics are used	Retention: 5 years	10000			Times

Note When a product is first written after shipment, “erase → write” and “write only” are both taken as one rewrite.

CHAPTER 30 PACKAGE DRAWINGS

64-PIN PLASTIC FBGA (6x6)





(UNIT:mm)

ITEM	DIMENSIONS
D	6.00±0.10
E	6.00±0.10
w	0.20
A	1.41±0.10
A1	0.30±0.05
A2	1.11
e	0.65
b	0.40±0.05
x	0.08
y	0.10
y1	0.20
ZD	0.725
ZE	0.725

P64F1-65-BA4

<R>

CHAPTER 31 RECOMMENDED SOLDERING CONDITIONS

These products should be soldered and mounted under the following recommended conditions.

For soldering methods and conditions other than those recommended below, contact a Renesas Electronics sales representative.

For technical information, see the following website.

Semiconductor Device Mount Manual (http://www2.renesas.com/pkg/en/mount/index.html)

Caution For soldering methods and conditions other than those recommended below, contact a Renesas Electronics sales representative.

Table 31-1. Surface Mounting Type Soldering Conditions

- **64-pin plastic FBGA (6×6)**

  μPD78F1016F1-BA4-A, μPD78F1017F1-BA4-A, μPD78F1018F1-BA4-A

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 260°C, Time: 60 seconds max. (at 220°C or higher), Count: 3 times or less, Exposure limit: 7 days[Note] (after that, prebake at 125°C for 10 to 72 hours)	IR60-107-3
Partial heating	Pin temperature: 350°C max., Time: 3 seconds max. (per pin row)	–

Note After opening the dry pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

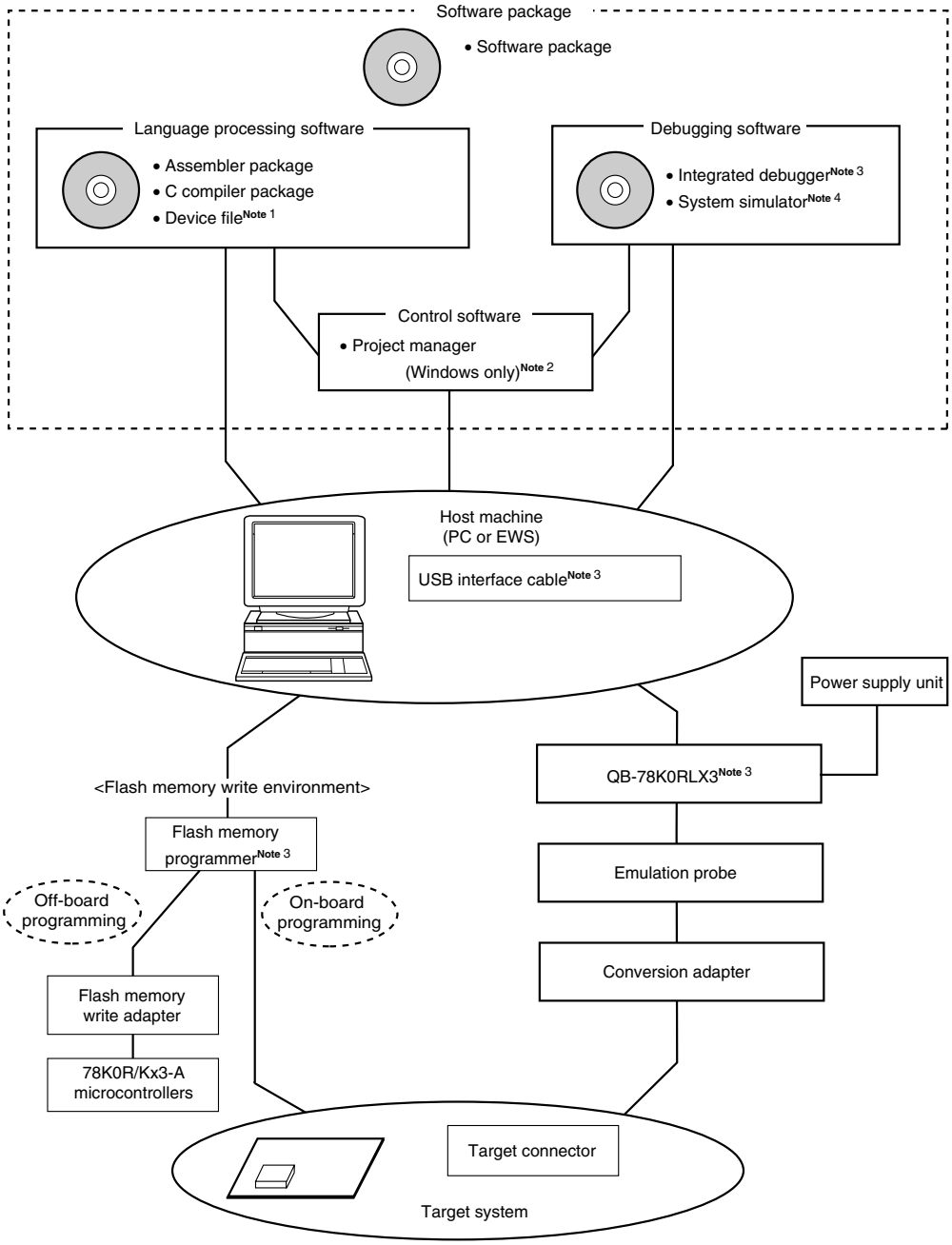
Caution Do not use different soldering methods together (except for partial heating).

APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for the development of systems that employ the microcontrollers. Figure A-1 shows the development tool configuration.

Figure A-1. Development Tool Configuration (1/2)

(1) When using the in-circuit emulator QB-78K0RLx3



Notes 1. Download the device file for 78K0R/Kx3-A microcontrollers (DF781018) from the download site for <R> development tools (http://www2.renesas.com/micro/en/ods/).

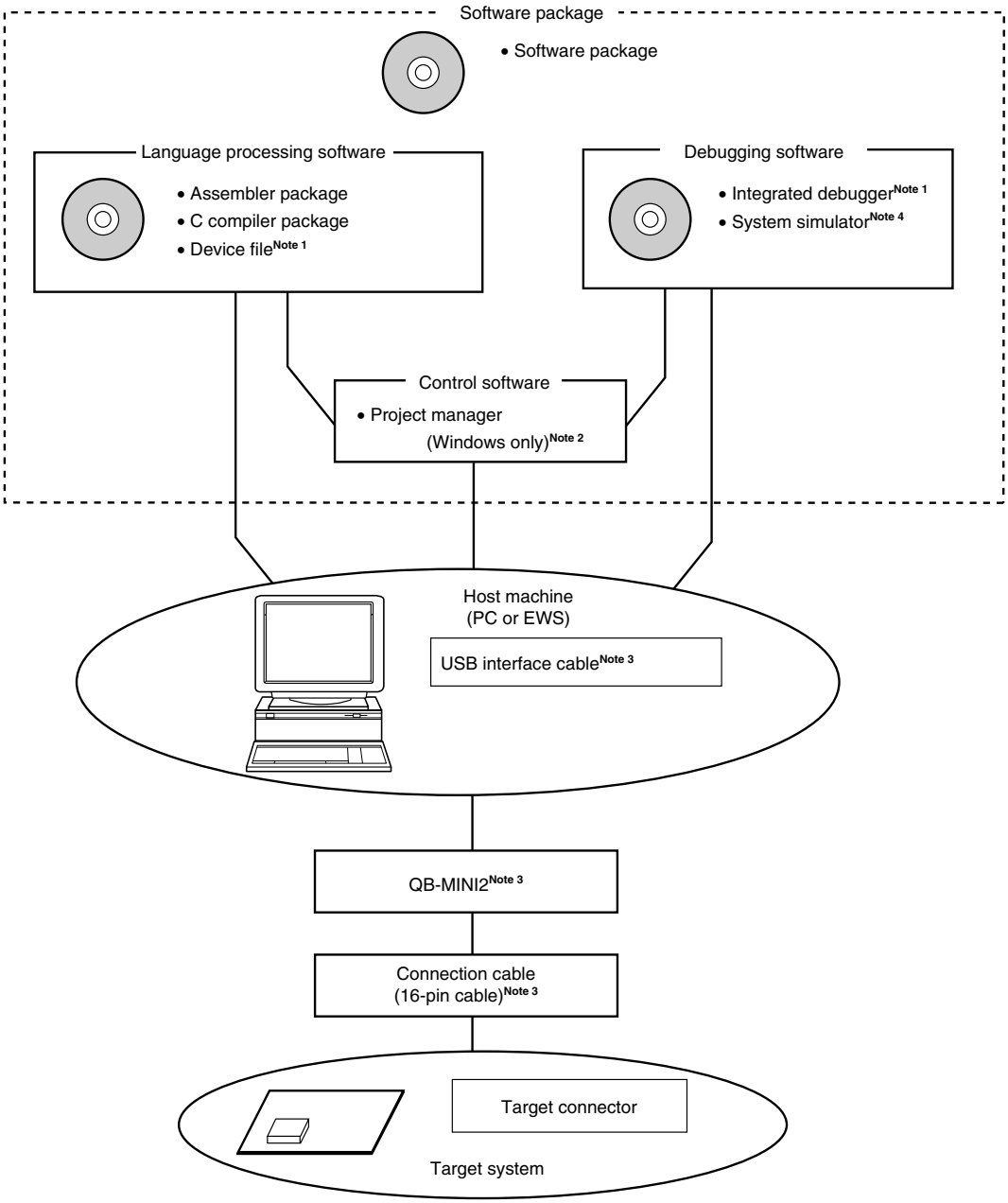
2. The project manager PM+ is included in the assembler package.
The PM+ is only used for Windows™.

3. In-circuit emulator QB-78K0RLX3 is supplied with integrated debugger ID78K0R-QB, on-chip debug emulator with programming function QB-MINI2, and USB interface cable. Any other products are sold separately.

4. SM+ for 78K0R (instruction simulation version) is included in the software package. Instruction + peripheral simulation version (under development) is not included.

Figure A-1. Development Tool Configuration (2/2)

(2) When using the on-chip debug emulator with programming function QB-MINI2



Notes 1. Download the device file for 78K0R/Kx3-A microcontrollers (DF781018) and the integrated debugger ID78K0R-QB from the download site for development tools
<R> (http://www2.renesas.com/micro/en/ods/).

2. The project manager PM+ is included in the assembler package.
The PM+ is only used for Windows.

3. On-chip debug emulator QB-MINI2 is supplied with USB interface cable, connection cables (10-pin cable and 16-pin cable), and 78K0-OCD board. In addition, download the software for operating the QB-MINI2 from the download site for MINICUBE2
<R> (http://www2.renesas.com/micro/en/development/asia/minicube2/minicube2.html).

4. SM+ for 78K0R (instruction simulation version) is included in the software package. Instruction + peripheral simulation version (under development) is not included.

A.1 Software Package

SP78K0R 78K0R microcontroller software package	Development tools (software) common to the 78K0R microcontrollers are combined in this package.

A.2 Language Processing Software

RA78K0R Assembler package	This assembler converts programs written in mnemonics into object codes executable with a microcontroller. This assembler is also provided with functions capable of automatically creating symbol tables and branch instruction optimization. This assembler should be used in combination with a device file (DF781018). **<Precaution when using RA78K0R in PC environment>** This assembler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) on Windows.
CC78K0R C compiler package	This compiler converts programs written in C language into object codes executable with a microcontroller. This compiler should be used in combination with an assembler package and device file. **<Precaution when using CC78K0R in PC environment>** This C compiler package is a DOS-based application. It can also be used in Windows, however, by using the Project Manager (included in assembler package) on Windows.
DF781018[Note] Device file	This file contains information peculiar to the device. This device file should be used in combination with a tool (RA78K0R, CC78K0R, ID78K0R-QB, and system simulator. The corresponding OS and host machine differ depending on the tool to be used.

Note The DF781018 can be used in common with the RA78K0R, CC78K0R, ID78K0R-QB, and system simulator. Download the DF781018 from the download site for development tools <R> (http://www2.renesas.com/micro/en/ods/).

A.3 Flash Memory Programming Tools

A.3.1 When using flash memory programmer PG-FP5 and FL-PR5

PG-FP5, FL-PR5 Flash memory programmer	Flash memory programmer dedicated to microcontrollers with on-chip flash memory.
<R> FA-78F1018F1-BA4-RX Flash memory programming adapter	Flash memory programming adapter used connected to the flash memory programmer for use.

Remarks 1. FL-PR5 and FA-78F1018F1-BA4-RX are products of Naito Densei Machida Mfg. Co., Ltd.
TEL: +81-42-750-4172, URL: http://www.ndk-m.co.jp/
Naito Densei Machida Mfg. Co., Ltd.

2. Use the latest version of the flash memory programming adapter.

A.3.2 When using on-chip debug emulator with programming function QB-MINI2

QB-MINI2 On-chip debug emulator with programming function	This is a flash memory programmer dedicated to microcontrollers with on-chip flash memory. It is available also as on-chip debug emulator which serves to debug hardware and software when developing application systems using the 78K0R/Kx3-A microcontrollers. When using this as flash memory programmer, it should be used in combination with a connection cable (16-pin cable) and a USB interface cable that is used to connect the host machine.

Remark <R> Download the software for operating the QB-MINI2 from the download site for MINICUBE2 (http://www2.renesas.com/micro/en/development/asia/minicube2/minicube2.html).

A.4 Debugging Tools (Hardware)

A.4.1 When using in-circuit emulator QB-78K0RLx3

QB-78K0RLx3 In-circuit emulator	This in-circuit emulator serves to debug hardware and software when developing application systems using the 78K0R/Kx3-A microcontrollers. It supports to the integrated debugger (ID78K0R-QB). This emulator should be used in combination with a power supply unit and emulation probe, and the USB is used to connect this emulator to the host machine.
QB-144-CA-01 Check pin adapter	This check pin adapter is used in waveform monitoring using the oscilloscope, etc.
QB-144-EP-02S Emulation probe	This emulation probe is flexible type and used to connect the in-circuit emulator and target system.
QB-64F1-EA-06T Exchange adapter	This exchange adapter is used to perform pin conversion from the in-circuit emulator to target connector.
QB-64F1-NQ-01T Target connector	This target connector is used to mount on the target system.

Remarks 1. The QB-78K0RLx3 is supplied with an integrated debugger ID78K0R-QB, USB interface cable, and on-chip debug emulator QB-MINI2.

When using the QB-MINI2 download the software for operating the QB-MINI2 from the download site for development tools (http://www2.renesas.com/micro/en/ods/). <R>

2. The packed contents differ depending on the part number, as follows.

Packed Contents / Part Number	In-Circuit Emulator	Emulation Probe	Exchange Adapter	Target Connector
QB-78K0RLx3-ZZZ	QB-78K0RLx3	None		
QB-78K0RLx3-T64F1		QB-144-EP-02S	QB-64F1-EA-06T	QB-64F1-NQ-01T

A.4.2 When using on-chip debug emulator with programming function QB-MINI2

QB-MINI2 On-chip debug emulator with programming function	This on-chip debug emulator serves to debug hardware and software when developing application systems using the 78K0R/Kx3-A microcontrollers. It is available also as flash memory programmer dedicated to microcontrollers with on-chip flash memory. When using this as on-chip debug emulator, it should be used in combination with a connection cable (16-pin cable), and USB interface cable that is used to connect the host machine.

Remark Download the software for operating the QB-MINI2 from the download site for MINICUBE2 (http://www2.renesas.com/micro/en/development/asia/minicube2/minicube2.html). <R>

A.5 Debugging Tools (Software)

ID78K0R-QB Integrated debugger[Note 1]	This debugger supports the in-circuit emulators for the 78K0R microcontrollers. The ID78K0R-QB is Windows-based software. It has improved C-compatible debugging functions and can display the results of tracing with the source program using an integrating window function that associates the source program, disassemble display, and memory display with the trace result. It should be used in combination with the device file (DF781018).
System simulator	System simulator is Windows-based software. It is used to perform debugging at the C source level or assembler level while simulating the operation of the target system on a host machine. Use of system simulator allows the execution of application logical testing and performance testing on an independent basis from hardware development, thereby providing higher development efficiency and software quality. System simulator should be used in combination with the device file (DF781018). The following two types of system simulators supporting the 78K0R/Kx3-A microcontrollers are available. • SM+ for 78K0R (instruction simulation version) This can only simulate a CPU. It is included in the software package. • Instruction + peripheral simulation version[Notes 1,2] This can simulate a CPU and peripheral hardware (ports, timers, serial interfaces, etc.).

Notes 1. Download the ID78K0-QB from the download site for development tools
<R> (http://www2.renesas.com/micro/en/ods/).

2. Under development

APPENDIX B REGISTER INDEX

B.1 Register Index (In Alphabetical Order with Respect to Register Names)

[R]

B.2 Register Index (In Alphabetical Order with Respect to Register Symbol)

[R]

APPENDIX C LIST OF CAUTIONS

This appendix lists the cautions described in this document.
"Classification (hard/soft)" in the table is as follows.

Hard: Cautions for microcontroller internal/external hardware
Soft: Cautions for software such as register settings or programs

(1/37)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 1	Hard	Outline	On-chip debug function	The 78K0R/Kx3-A 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.	p.17 ☐
			AV_{SS}, V_{SS}	Make AV_{SS} the same potential as V_{SS}.	p.18 ☐
			REGC	Connect the REGC pin to Vss via a capacitor (0.47 to 1 μF).	p.18 ☐
Chapter 2	Soft	Pin functions	P10/$\overline{SCK20}$/SCL20, P11/SI20/RxD2/SDA20/INTP6	To use P10/$\overline{SCK20}$/SCL20, P11/SI20/RxD2/SDA20/INTP6 as a general-purpose port, note the serial array unit 1 setting. For details, refer to Table 14-9 Relationship Between Register Settings and Pins (Channel 0 of unit 1: CSI20, UART2 transmission, IIC20).	p.31 ☐
			P12/TO02/SO20/TxD2	To use P12/TO02/SO20/TxD2 as a general-purpose port, set bit 2 (TO02) of timer output register 0 (TO0) and bit 2 (TOE02) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting. And as a general-purpose port, note the serial array unit 1 setting. For details of serial array unit 1 setting, refer to Table 14-9 Relationship Between Register Settings and Pins (Channel 0 of unit 1: CSI20, UART2 transmission, IIC20).	p.31 ☐
			P13/TO04/SO10/TxD1	To use P13/TO04/SO10/TxD1 as a general-purpose port, set bit 4 (TO04) of timer output register 0 (TO0) and bit 4 (TOE04) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting. And as a general-purpose port, note the serial array unit 0 setting. For details of serial array unit 0 setting, refer to Table 14-7 Relationship Between Register Settings and Pins (Channel 2 of unit 0: CSI10, UART1 transmission, IIC10).	p.31 ☐
			P14/SI10/RxD1/SDA10/INTP4, P15/$\overline{SCK10}$/SCL10/INTP7	To use P14/SI10/RxD1/SDA10/INTP4, P15/$\overline{SCK10}$/SCL10/INTP7 as a general-purpose port, note the serial array unit 0 setting. For details, refer to Table 14-7 Relationship Between Register Settings and Pins (Channel 2 of unit 0: CSI10, UART1 transmission, IIC10).	p.31 ☐
			P16/TO05/T105/INTP10	To use P16/TO05/TI05/INTP10 as a general-purpose port, set bit 5 (TO05) of timer output register 0 (TO0) and bit 5 (TOE05) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.	p.31 ☐
			P20/ANI0/AMP0- to P27/ANI7/AMP2O	P20/ANI0/AMP0- to P27/ANI7/AMP2O are set in the digital input (general-purpose port) mode after release of reset.	p.32 ☐
	Hard			When using at least one port of ports P20/ANI0/AMP0- to P27/ANI7/AMP2O as a digital port, set AV_{DD0} to the same potential as EV_{DD} or V_{DD}.	p.32 ☐

(2/37)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 2	Soft	Pin functions	P30/TO00/TI03/RTC1HZ/INTP1	To use P30/TO00/TI03/RTC1HZ/INTP1 as a general-purpose port, set bit 5 (RCLOE1) of real-time counter control register 0 (RTCC0), bit 0 (TO00) of timer output register 0 (TO0) and bit 0 (TOE00) of timer output enable register 0 (TOE0) to “0”, which is the same as their default status setting.	p.33 ☐
			P31/TO03/TI00/RTCDIV/RTCCL/PCLBUZ1/INTP2	To use P31/TO03/TI00/RTCDIV/RTCCL/PCLBUZ1/INTP2 as a general-purpose port, set bit 4 (RCLOE0) of real-time counter control register 0 (RTCC0), bit 6 (RCLOE2) of real-time counter control register 2 (RTCC2), bit 3 (TO03) of timer output register 0 (TO0), bit 3 (TOE03) of timer output enable register 0 (TOE0) and bit 7 of clock output select register 1 (CKS1) to “0”, which is the same as their default status setting.	p.33 ☐
			P32/TO01/TI01/INTP5/PCLBUZ0	To use P32/TO01/TI01/INTP5/PCLBUZ0 as a general-purpose port, set bit 1 (TO01) of timer output register 0 (TO0), bit 1 (TOE01) of timer output enable register 0 (TOE0) and bit 7 of clock output select register 0 (CKS0) to “0”, which is the same as their default status setting.	p.33 ☐
			P33/TO07/TI07/INTP3, P34/TO06/	To use P33/TO07/TI07/INTP3, P34/TO06/TI06/INTP8 as a general-purpose port, set bit 7, 6 (TO07, TO06) of timer output register 0 (TO0), and bit 7, 6 (TOE07, TOE06) of timer output enable register 0 (TOE0) to “0”, which is the same as their default status setting.	p.33 ☐
	Hard		P40/TOOL0	The function of the P40/TOOL0 pin varies as described in (a) to (c) below. In the case of (b) or (c), make the specified connection. (a) In normal operation mode and when on-chip debugging is disabled (OCDENSET = 0) by an option byte (000C3H) =>Use this pin as a port pin (P40). (b) In normal operation mode and when on-chip debugging is enabled (OCDENSET = 1) by an option byte (000C3H) =>Connect this pin to V_{DD} via an external resistor, and always input a high level to the pin before reset release. (c) When on-chip debug function is used, or in write mode of flash memory programmer =>Use this pin as TOOL0. Directly connect this pin to the on-chip debug emulator or a flash memory programmer, or pull it up by connecting it to V_{DD} via an external resistor.	p.34 ☐
	Soft		P60/SCL0, P61/SDA0	When using P60/SCL0 or P61/SDA0 as a general-purpose port, stop the operation of serial interface IICA.	p.35 ☐
			P80/$\overline{\text{SCK00}}$/INTP11, P81/RxD0/SI00/INTP9, P82/SO00/TxD0	To use P80/$\overline{\text{SCK00}}$/INTP11, P81/RxD0/SI00/INTP9, P82/SO00/TxD0, as a general-purpose port, note the serial array unit 0 setting. For details, refer to Table 14-5 Relationship Between Register Settings and Pins (Channel 0 of unit 0: CSI00, UART0 transmission).	p.36 ☐
	Hard		P110/ANO0, P111/ANO1	When using at least one port of P110/ANO0 and P111/ANO1 as a digital port, set AV_{DD1} to the same potential as EV_{DD} or V_{DD}.	p.37 ☐
	Soft		P121 to P124	The function setting on P121 to P124 is available only once after the reset release. The port once set for connection to an oscillator cannot be used as an input port unless the reset is performed.	p.38 ☐

(3/37)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 2	Soft	Pin functions	P150/ANI8/AMP2+ to P152/ANI10 and P157/ANI15/AV_{REFM}	P150/ANI8/AMP2+ to P152/ANI10, and P157/ANI15/AV_{REFM} are set in the digital input (general-purpose port) mode after release of reset.	p.38 ☐
	Hard			When using at least one port of P150/ANI8/AMP2+ to P152/ANI10, and P157/ANI15/AV_{REFM} as a digital port, set AV_{DD0} to the same potential as EV_{DD} or V_{DD}.	p.38 ☐
			REGC	Keep the wiring length as short as possible for the broken-line part in the above figure.	p.39 ☐
Chapter 3	Soft	Memory space	PMC: Processor mode control register	Set PMC only once during the initial settings prior to operating the DMA controller. Rewriting PMC other than during the initial settings is prohibited.	p.57 ☐
				After setting PMC, wait for at least one instruction and access the mirror area.	p.57 ☐
				When the μPD78F1016 (flash memory size: 64 KB) is used, be sure to set bit 0 (MAA) of this register to 0.	p.57 ☐
			Internal data memory space	It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or as a stack area.	p.57, 63, 64 ☐
				While using the self-programming function, the area of FFE20H to FFEFFH cannot be used as a stack memory.	p.57, 63 ☐
			SFR: Special function register area	Do not access addresses to which SFRs are not assigned.	p.58, 67 ☐
			2nd SFR: Extended special function register	Do not access addresses to which 2nd SFRs are not assigned.	p.58, 73 ☐
		Processor registers	SP: Stack pointer	Since reset signal generation makes the SP contents undefined, be sure to initialize the SP before using the stack.	p.63 ☐
Chapter 4	Soft	Port functions	P10/$\overline{SCK20}$/SCL20, P11/SI20/RxD2/SDA20/INTP6	To use P10/$\overline{SCK20}$/SCL20, P11/SI20/RxD2/SDA20/INTP6 as a general-purpose port, note the serial array unit 1 setting. For details, refer to Table 14-9 Relationship Between Register Settings and Pins (Channel 0 of unit 1: CSI20, UART2 transmission, IIC20).	p.97 ☐
			P12/TO02/SO20/TxD2	To use P12/TO02/SO20/TxD2 as a general-purpose port, set bit 2 (TO02) of timer output register 0 (TO0) and bit 2 (TOE02) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting. And as a general-purpose port, note the serial array unit 1 setting. For details of serial array unit 1 setting, refer to Table 14-9 Relationship Between Register Settings and Pins (Channel 0 of unit 1: CSI20, UART2 transmission, IIC20).	p.97 ☐
			P13/TO04/SO10/TxD1	To use P13/TO04/SO10/TxD1 as a general-purpose port, set bit 4 (TO04) of timer output register 0 (TO0) and bit 4 (TOE04) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting. And as a general-purpose port, note the serial array unit 0 setting. For details of serial array unit 0 setting, refer to Table 14-7 Relationship Between Register Settings and Pins (Channel 2 of unit 0: CSI10, UART1 transmission, IIC10).	p.97 ☐
			P14/SI10/RxD1/SDA10/INTP4, P15/$\overline{SCK10}$/SCL10/INTP7	To use P14/SI10/RxD1/SDA10/INTP4, P15/$\overline{SCK10}$/SCL10/INTP7 as a general-purpose port, note the serial array unit 0 setting. For details, refer to Table 14-7 Relationship Between Register Settings and Pins (Channel 2 of unit 0: CSI10, UART1 transmission, IIC10).	p.97 ☐
			P16/TO05/TI05/INTP10	To use P16/TO05/TI05/INTP10 as a general-purpose port, set bit 5 (TO05) of timer output register 0 (TO0) and bit 5 (TOE05) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.	p.97 ☐

(4/37)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 4	Hard	Port functions	Port 2	Make the AV_{DD0} pin the same potential as the EV_{DD} or V_{DD} pin when port 2 is used as a digital port.	p.101 ☐
	Soft		P30/TO00/TI03/RTC1HZ/INTP1	To use P30/TO00/TI03/RTC1HZ/INTP1 as a general-purpose port, set bit 5 (RCLOE1) of real-time counter control register 0 (RTCC0), bit 0 (TO00) of timer output register 0 (TO0) and bit 0 (TOE00) of timer output enable register 0 (TOE0) to “0”, which is the same as their default status setting.	p.105 ☐
			P31/TO03/TI00/RTCDIV/RTCCL/PCLBUZ1/INTP2	To use P31/TO03/TI00/RTCDIV/RTCCL/PCLBUZ1/INTP2 as a general-purpose port, set bit 4 (RCLOE0) of real-time counter control register 0 (RTCC0), bit 6 (RCLOE2) of real-time counter control register 2 (RTCC2), bit 3 (TO03) of timer output register 0 (TO0), bit 3 (TOE03) of timer output enable register 0 (TOE0) and bit 7 of clock output select register 1 (CKS1) to “0”, which is the same as their default status setting.	p.105 ☐
			P32/TO01/TI01/INTP5/PCLBUZ0	To use P32/TO01/TI01/INTP5/PCLBUZ0 as a general-purpose port, set bit 1 (TO01) of timer output register 0 (TO0), bit 1 (TOE01) of timer output enable register 0 (TOE0) and bit 7 of clock output select register 0 (CKS0) to “0”, which is the same as their default status setting.	p.105 ☐
			P33/TO07/TI07/INTP3, P34/TO06/TI06/INTP8	To use P33/TO07/TI07/INTP3, P34/TO06/TI06/INTP8 as a general-purpose port, set bit 7, 6 (TO07, TO06) of timer output register 0 (TO0), and bit 7, 6 (TOE07, TOE06) of timer output enable register 0 (TOE0) to “0”, which is the same as their default status setting.	p.105 ☐
	Hard		P40, P41	When a tool is connected, the P40 pin cannot be used as a port pin. When the on-chip debug function is used, P41 pin can be used as follows by the mode setting on the debugger. • 1-line mode: can be used as a port (P41). • 2-line mode: used as a TOOL1 pin and cannot be used as a port (P41).	p.107 ☐
	Soft		P60/SCL0, P61/SDA0	When using P60/SCL0 or P61/SDA0 as a general-purpose port, stop the operation of serial interface IICA.	p.112 ☐
			P80/$\overline{SCK00}$/INTP11, P81/RxD0/SI00/INTP9, P82/SO00/TxD0	To use P80/$\overline{SCK00}$/INTP11, P81/RxD0/SI00/INTP9, P82/SO00/TxD0, as a general-purpose port, note the serial array unit 0 setting. For details, refer to Table 14-5 Relationship Between Register Settings and Pins (Channel 0 of unit 0: CSI00, UART0 transmission).	p.113 ☐
	Hard		Port 11	Make the AV_{DD1} pin the same potential as the EV_{DD} or V_{DD} pin when port 11 is used as a digital port.	p.118 ☐
			P121 to P124	The function setting on P121 to P124 is available only once after the reset release. The port once set for connection to an oscillator cannot be used as an input port unless the reset is performed.	p.119 ☐
			Port 15	Make the AV_{DD0} pin the same potential as the EV_{DD} or V_{DD} pin when port 15 is used as a digital port.	p.124 ☐

(5/37)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 4	Soft	Port functions	Port mode register	Be sure to set bits 3 to 7 of PM0, bit 7 of PM1, bits 5 to 7 of PM3, bits 2 to 7 of PM4, bits 2 to 7 of PM6, bits 3 to 7 of PM8, bits 1 to 7 of PM10, bits 2 to 7 of PM11, bits 1 to 7 of PM12, and bits 3 to 6 of PM15 to 1.	p.128 ☐
			ADPC: A/D port configuration register	Set the channel used for A/D conversion to the input mode by using port mode registers 2 and 15 (PM2, PM15).	p.134 ☐
				Do not set the pin that is set by ADPC as digital I/O by analog input channel specification register (ADS).	p.134 ☐
			ISC: Input switch control register	Be sure to clear bits 5 to 7 to “0”.	p.136 ☐
			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.	p.144 ☐
Chapter 5	Soft	Clock generator	CMC: Clock operation mode control register	CMC can be written only once after reset release, by an 8-bit memory manipulation instruction.	p.149 ☐
				After reset release, set CMC before X1 or XT1 oscillation is started as set by the clock operation status control register (CSC).	p.149 ☐
				Be sure to set AMPH to 1 if the X1 clock oscillation frequency exceeds 10 MHz.	p.149 ☐
				To use CMC with its initial value (00H), be sure to set it to 00H after releasing reset in order to prevent malfunction when a program loop occurs.	p.149 ☐
	Hard			The XT1 oscillator is designed as a low-gain circuit for achieving low-power consumption. Note the following points when designing the XT1 oscillator. • The pins and circuit board include parasitic capacitance. Therefore, confirm that there are no problems by performing oscillation evaluation on the circuit board to be actually used. • When low-consumption oscillation or super-low-consumption oscillation is selected, lower power consumption than when selecting normal oscillation can be achieved. However, in this case, the XT1 oscillation margin is reduced, so perform sufficient oscillation evaluation of the resonator to be used for XT1 oscillation before using the resonator. • Keep the wiring length between the XT1 and XT2 pins and resonator as short as possible and parasitic capacitance and wire resistance as small as possible. This is particularly important when super-low-consumption oscillation (AMPHS1 = 1) is selected. • Configure the circuit board by using material with little parasitic capacitance and wire resistance. • Place a ground pattern that has the same potential as V_{SS} (if possible) around the XT1 oscillator. • Do not cross the signal lines between the XT1 and XT2 pins and the resonator with other signal lines. Do not route the signal lines near a signal line through which a high fluctuating current flows. • Moisture absorption by the circuit board and condensation on the board in a highly humid environment may cause the impedance between the XT1 and XT2 pins to drop and disable oscillation. When using the circuit board in such an environment, prevent the circuit board from absorbing moisture by taking measures such as coating the circuit board. • Coat the surface of the circuit board by using material that does not generate capacitance or leakage between the XT1 and XT2 pins.	p.149 ☐

(6/37)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 5	Soft	Clock generator	CSC: Clock operation status control register	After reset release, set the clock operation mode control register (CMC) before starting X1 oscillation as set by MSTOP or XT1 oscillation as set by XTSTOP.	p.150 ☐
				To start X1 oscillation as set by MSTOP, check the oscillation stabilization time of the X1 clock by using the oscillation stabilization time counter status register (OSTC).	p.151 ☐
				Do not stop the clock selected for the CPU peripheral hardware clock (f_{CLK}) with the CSC register.	p.151 ☐
				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 follows.	p.151 ☐
			OSTC: Oscillation stabilization time counter status register	After the above time has elapsed, the bits are set to 1 in order from MOST8 and remain 1.	p.152 ☐
				The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS. In the following cases, set the oscillation stabilization time of OSTS to the value greater than the count value which is to be checked by the OSTC register after the oscillation starts. • If the X1 clock starts oscillation while the internal high-speed oscillation clock or subsystem clock is being used as the CPU clock. • If the STOP mode is entered and then released while the internal high-speed oscillation 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 OSTS is set to OSTC after the STOP mode is released.)	p.152 ☐
	Hard			The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts (“a” below).	p.152 ☐
	Soft		OSTS: Oscillation stabilization time select register	To set the STOP mode when the X1 clock is used as the CPU clock, set the OSTS register before executing the STOP instruction.	p.154 ☐
				Setting the oscillation stabilization time to 20 μs or less is prohibited.	p.154 ☐
				To change the setting of the OSTS register, be sure to confirm that the counting operation of the OSTC register has been completed.	p.154 ☐
				Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.	p.154 ☐
				The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS. In the following cases, set the oscillation stabilization time of OSTS to the value greater than the count value which is to be checked by the OSTC register after the oscillation starts. • If the X1 clock starts oscillation while the internal high-speed oscillation clock or subsystem clock is being used as the CPU clock. • If the STOP mode is entered and then released while the internal high-speed oscillation 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 OSTS is set to OSTC after the STOP mode is released.)	p.154 ☐
	Hard			The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts (“a” below).	p.154 ☐

(7/37)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 5	Soft	Clock generator	CKC: System clock control register	The clock set by CSS, MCM0, SDIV, and MDIV2 to MDIV0 is supplied to the CPU and peripheral hardware. If the CPU clock is changed, therefore, the clock supplied to peripheral hardware (except the real-time counter, timer array unit (when f_{SUB}/2, f_{SUB}/4, the valid edge of TI0mn input, or the valid edge of INTRTCI is selected as the count clock), clock output/buzzer output, and watchdog timer) is also changed at the same time. Consequently, stop each peripheral function when changing the CPU/peripheral operating hardware clock.	p.156 ☐
	Hard			If the peripheral hardware clock is used as the subsystem clock, the operations of the A/D converter and IICA are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 29 ELECTRICAL SPECIFICATIONS.	p.156 ☐
	Soft		20 MHz internal high-speed oscillation control register (DSCCTL)	20 MHz internal oscillation can only be used if $V_{DD} \geq 2.7$ V.	p.157 ☐
				Set SELDSC when 100 μs have elapsed after having set DSCON with $V_{DD} \geq 2.7$ V.	p.157 ☐
				The internal high-speed oscillator must be operated (HIOSTOP = 0) when DSCON = 1.	p.157 ☐
			OSMC: Operation speed mode control register	Write “1” to FSEL before the following two operations. • Changing the clock prior to dividing f_{CLK} to a clock other than f_{IH}. • Operating the DMA controller.	p.160 ☐
				The CPU waits (140.5 clock (f_{CLK})) when “1” is written to the FSEL bit. Interrupt requests issued during a wait will be suspended. However, counting the oscillation stabilization time of f_X can continue even while the CPU is waiting.	p.160 ☐
				To increase f_{CLK} to 10 MHz or higher, set FSEL to “1”, then change f_{CLK} after two or more clocks have elapsed.	p.160 ☐
				Confirm that the clock is operating at 10 MHz or less before setting FSEL = 0.	p.160 ☐
				To shift to STOP mode while $V_{DD} \leq 2.7$ V, set FSEL = 0 after setting f_{CLK} to 10 MHz or less.	p.160 ☐
				The HALT mode current when operating on the subsystem clock can be reduced by setting RTCLPC to 1. However, the clock cannot be supplied to peripheral functions except the real-time counter in the subsystem clock HALT mode. Set bit 7 (RTCEN) of PER0 to 1 and bits 0 to 6 of PER0 to 0 before setting the subsystem clock HALT mode.	p.161 ☐
				Once FLPC has been set from 0 to 1, setting it back to 0 from 1 other than by reset is prohibited.	p.161 ☐
				When setting FSEL to “1”, do so while RMC = 00H. When setting FLPC to “1”, do so while RMC = 5AH.	p.161 ☐

(8/37)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 5	Hard	X1/XT1 oscillator	–	When using the X1 oscillator and XT1 oscillator, wire as follows in the area enclosed by the broken lines in the Figures 5-10 and 5-11 to avoid an adverse effect from wiring capacitance. • Keep the wiring length as short as possible. • Do not cross the wiring with the other signal lines. Do not route the wiring near a signal line through which a high fluctuating current flows. • Always make the ground point of the oscillator capacitor the same potential as V_{SS}. Do not ground the capacitor to a ground pattern through which a high current flows. • Do not fetch signals from the oscillator. Note that the XT1 oscillator is designed as a low-gain circuit for achieving low-power consumption. Note the following points when designing the XT1 oscillator. • The pins and circuit board include parasitic capacitance. Therefore, confirm that there are no problems by performing oscillation evaluation on the circuit board to be actually used. • When low-consumption oscillation or super-low-consumption oscillation is selected, lower power consumption than when selecting normal oscillation can be achieved. However, in this case, the XT1 oscillation margin is reduced, so perform sufficient oscillation evaluation of the resonator to be used for XT1 oscillation before using the resonator. • Keep the wiring length between the XT1 and XT2 pins and resonator as short as possible and parasitic capacitance and wire resistance as small as possible. This is particularly important when super-low-consumption oscillation (AMPHS1 = 1) is selected. • Configure the circuit board by using material with little parasitic capacitance and wire resistance. • Place a ground pattern that has the same potential as V_{SS} (if possible) around the XT1 oscillator. • Do not cross the signal lines between the XT1 and XT2 pins and the resonator with other signal lines. Do not route the signal lines near a signal line through which a high fluctuating current flows. • Moisture absorption by the circuit board and condensation on the board in a highly humid environment may cause the impedance between the XT1 and XT2 pins to drop and disable oscillation. When using the circuit board in such an environment, prevent the circuit board from absorbing moisture by taking measures such as coating the circuit board. • Coat the surface of the circuit board by using material that does not generate capacitance or leakage between the XT1 and XT2 pins.	p.163 □
				When X2 and XT1 are wired in parallel, the crosstalk noise of X2 may increase with XT1, resulting in malfunctioning.	p.165 □

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 5	Hard	Internal high-speed oscillator	–	To use the 1, 8, or 20 MHz internal high-speed oscillation clock, use the option byte to set the frequency in advance (for details, see CHAPTER 24 OPTION BYTE). Also, the internal high-speed oscillator automatically starts oscillating after reset release. (If 8 MHz or 20 MHz is selected by using the option byte, the microcontroller operates using the 8 MHz internal high-speed oscillator.) To use the 20 MHz internal high-speed oscillator to operate the microcontroller, oscillation is started by setting bit 0 (DSCON) of the DSCCTL register to 1 with VDD ≥ 2.7 V.	p.166 ☐
		Clock generator operation when power supply voltage is turned on	When LVI default start function stopped is set (option byte: LVIOFF = 1)	If the voltage rises with a slope of less than 0.5 V/ms (MIN.) from power application until the voltage reaches 1.8 V, input a low level to the $\overline{\text{RESET}}$ pin from power application until the voltage reaches 1.8 V, or set the LVI default start function stopped by using the option byte (LVIOFF = 0) (see Figure 5-14). By doing so, the CPU operates with the same timing as <2> and thereafter in Figure 5-13 after reset release by the $\overline{\text{RESET}}$ pin.	p.169 ☐
				It is not necessary to wait for the oscillation stabilization time when an external clock input from the EXCLK pin is used.	p.169 ☐
			When LVI default start function enabled is set (option byte: LVIOFF = 0)	A voltage stabilization time (about 2.12 to 5.84 ms) is required after the supply voltage reaches 1.61 V (TYP.). If the time for the supply voltage to rise from 1.61 V (TYP.) to 2.07 V (TYP.) is shorter than the voltage stabilization time, reset processing is entered after the voltage stabilization time elapses.	p.171 ☐
				It is not necessary to wait for the oscillation stabilization time when an external clock input from the EXCLK pin is used.	p.171 ☐

(10/37)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 5	Soft	Controlling high-speed system clock	X1/P121, X2/EXCLK/P122	The X1/P121 and X2/EXCLK/P122 pins are in the input port mode after a reset release.	p.172 ☐
			X1 clock	The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction. Therefore, it is necessary to also set the value of the OSCSELS bit at the same time. For OSCSELS bit, see 5.6.3 Example of controlling subsystem clock.	p.172 ☐
				Set the X1 clock after the supply voltage has reached the operable voltage of the clock to be used (see CHAPTER 29 ELECTRICAL SPECIFICATIONS).	p.172 ☐
			External main system clock	The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction. Therefore, it is necessary to also set the value of the OSCSELS bits at the same time. For OSCSELS bits, see 5.6.3 Example of controlling subsystem clock.	p.173 ☐
				Set the external main system clock after the supply voltage has reached the operable voltage of the clock to be used (see CHAPTER 29 ELECTRICAL SPECIFICATIONS).	p.173 ☐
			High-speed system clock	Be sure to confirm that MCS = 0 or CLS = 1 when setting MSTOP to 1. In addition, stop peripheral hardware that is operating on the high-speed system clock.	p.175 ☐
		Controlling internal high-speed oscillation clock	Internal high-speed oscillation clock	If switching the CPU/peripheral hardware clock from the high-speed system clock to the internal high-speed oscillation clock after restarting the internal high-speed oscillation clock, do so after 10 μs or more have elapsed. If the switching is made immediately after the internal high-speed oscillation clock is restarted, the accuracy of the internal high-speed oscillation cannot be guaranteed for 10 μs.	p.176 ☐
				Be sure to confirm that MCS = 1 or CLS = 1 when setting HIOSTOP to 1. In addition, stop peripheral hardware that is operating on the internal high-speed oscillation clock.	p.177 ☐
		Subsystem clock control	XT1/P123, XT2/P124	The XT1/P123 and XT2/P124 pins are in the input port mode after a reset release.	p.177 ☐
	Hard		Subsystem clock	When the subsystem clock is used as the CPU clock, the subsystem clock is also supplied to the peripheral hardware (except the real-time counter, timer array unit (when f_{SUB}/2, f_{SUB}/4, the valid edge of TI0mn input, or the valid edge of INTRTCI is selected as the count clock), clock output/buzzer output, and watchdog timer). At this time, the operations of the A/D converter and IICA are not guaranteed. For the operating characteristics of the peripheral hardware, refer to the chapters describing the various peripheral hardware as well as CHAPTER 29 ELECTRICAL SPECIFICATIONS.	p.177, 178 ☐
	Soft			The CMC register can be written only once after reset release, by an 8-bit memory manipulation instruction. Therefore, it is necessary to also set the value of the EXCLK and OSCSEL bits at the same time. For EXCLK and OSCSEL bits, see 5.6.1 (1) Example of setting procedure when oscillating the X1 clock or 5.6.1 (2) Example of setting procedure when using the external main system clock.	p.178 ☐
				Be sure to confirm that CLS = 0 when setting XTSTOP to 1. In addition, stop the peripheral hardware if it is operating on the subsystem clock.	p.178 ☐
				The subsystem clock oscillation cannot be stopped using the STOP instruction.	p.178 ☐
		CPU clock status transition	–	Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 29 ELECTRICAL SPECIFICATIONS).	p.181, 182, 185 ☐

(11/37)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 6	Soft	Timer array unit	–	Channels 0 to 3 of timer array unit 1 can be used only as interval timers.	p.191 ☐
				Channels 1 and 5 to 7 of timer array unit 0 cannot be used as frequency dividers.	p.191 ☐
			TCRmn: Timer/counter register mn	The count value is not captured to TDRmn even when TCRmn is read.	p.197 ☐
			TDRmn: Timer data register mn	TDRmn does not perform a capture operation even if a capture trigger is input, when it is set to the compare function.	p.199 ☐
			PER0: Peripheral enable register 0	When setting the timer array unit, be sure to set TAUmEN to 1 first. If TAUmEN = 0, writing to a control register of the timer array unit is ignored, and all read values are default values.	p.201 ☐
			TPSm: Timer clock select register m	Be sure to clear bits 15 to 8 to “0”.	p.202 ☐
			TMRmn: Timer mode register mn	Be sure to clear bits 14, 13, 5, and 4 of TMR0n to “0”.	p.203 to 205 ☐
				Be sure to clear bits 14, 13, 11 to 8, and 5 to 1 of TMR1n to “0”.	p.203 to 205 ☐
				Channels 0 to 3 of timer array unit 1 can be used only as interval timers.	p.206, 210 ☐
			TSm: Timer channel start register m	Be sure to clear bits 15 to 8 of TS0 and bits 15 to 4 of TS1 to “0”.	p.209 ☐
			Start Timing (In Interval Timer Mode)	In the first cycle operation of count clock after writing TSmn, 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.	p.211 ☐
			Start Timing (In Capture Mode)	In the first cycle operation of count clock after writing TSpq, 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 MDpq0 = 1.	p.212 ☐
			Start Timing (In One-count Mode and In Capture & One-count Mode)	An input signal sampling error is generated since operation starts upon start trigger detection (The error is one count clock when TIpq is used).	p.213, 214 ☐
			TTm: Timer channel stop register m	Be sure to clear bits 15 to 8 of TT0 and bits 15 to 4 of TT1 to “0”.	p.215 ☐
			TIS0, TIS1: Timer input select registers 0, 1	When the LIN-bus communication function is used, select the input signal of the RxD3 pin by setting ISC1 = 1 (bit 1 of the input switch control register (ISC)) and TIS07 = 0.	p.216 ☐

(12/37)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 6	Soft	Timer array unit	TOE0: Timer output enable register 0	Be sure to clear bits 15 to 8 of TOE0 to “0”.	p.217 ☐
			TO0: Timer output register 0	Be sure to clear bits 15 to 8 of TO0 to “0”.	p.218 ☐
			TOL0: Timer output level register 0	Be sure to clear bits 15 to 8 of TOL0 to “0”.	p.219 ☐
			TOM0: Timer output mode register 0	Be sure to clear bits 15 to 8 of TOM0 to “0”.	p.220 ☐
			ISC: Input switch control register	Be sure to clear bits 5 to 7 to “0”.	p.221 ☐
			Changing values set in registers TO0,TOE0,TOL 0, and TOM0 during timer operation	Since the timer operations (operations of TCR0q and TDR0q) are independent of the TO0q output circuit and changing the values set in TO0, TOE0, TOL0, and TOM0 does not affect the timer operation, the values can be changed during timer operation. To output an expected waveform from the TO0q pin by timer operation, however, set TO0, TOE0, TOL0, and TOM0 to the values stated in the register setting example of each operation. When the values set in TOE0, TOL0, and TOM0 (except for TO0) are changed close to the timer interrupt (INTTM0q), the waveform output to the TO0q pin may be different depending on whether the values are changed immediately before or immediately after the timer interrupt (INTTM0q) signal generation timing.	p.226 ☐
			Default level of TO0q pin and output level after timer operation start	The following figure shows the TO0q pin output level transition when writing has been done in the state of TOE0q = 0 before port output is enabled and TOE0q = 1 is set after changing the default level. (a) When operation starts with TOM0q = 0 setting (toggle output) The setting of TOL0q is invalid when TOM0q = 0. When the timer operation starts after setting the default level, the toggle signal is generated and the output level of TO0q pin is reversed. (b) When operation starts with TOM0q = 1 setting (Combination operation mode (PWM output)) When TOM0q = 1, the active level is determined by TOL0q setting.	p.226, 227 ☐
			Operation of TO0q pin in combination operation mode (TOM0q = 1)	(a) When TOL0q setting has been changed during timer operation When the TOL0q setting has been changed during timer operation, the setting becomes valid at the generation timing of TO0q change condition. Rewriting TOL0q does not change the output level of TO0q. The following figure shows the operation when the value of TOL0q has been changed during timer operation (TOM0q = 1). (b) Set/reset timing To realize 0%/100% output at PWM output, the TO0q pin/TO0q set timing at master channel timer interrupt (INTTM0q) generation is delayed by 1 count clock by the slave channel timer interrupt (INTTM0r). If the set condition and reset condition are generated at the same time, a higher priority is given to the latter. Figure 6-30 shows the set/reset operating statuses where the master/slave channels are set as follows.	p.227, 228 ☐

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 6	Soft	Timer array unit	Collective manipulation of TO0q bits	When TOE0q = 1, even if the output by timer interrupt of each timer (INTTM0q) contends with writing to TO0q, output is normally done to TO0q pin.	p.229 ☐
	Hard		Single-operation function	The timer array unit 1 is available for only single-operation function only.	p.232 ☐
		Operations of timer array unit as independent channel	Input pulse interval measurement	The TI0q pin input is sampled using the operating clock selected with the CKS0q bit of the TMR0q register, so an error equal to the number of operating clocks occurs.	p.249 ☐
			Input signal high-/low-level width measurement	The TI0q pin input is sampled using the operating clock selected with the CKS0q bit of the TMR0q register, so an error equal to the number of operating clocks occurs.	p.253 ☐
	Soft	Operations of plural channels of timer array unit	PWM function	To rewrite both TDR0n of the master channel and TDR0p of the slave channel, a write access is necessary two times. The timing at which the values of TDR0n and TDR0p are loaded to TCR0n and TCR0p is upon occurrence of INTTM0n of the master channel. Thus, when rewriting is performed split before and after occurrence of INTTM0n of the master channel, the TO0p pin cannot output the expected waveform. To rewrite both TDR0n of the master and TDR0p of the slave, therefore, be sure to rewrite both the registers immediately after INTTM0n is generated from the master channel.	p.257 ☐
			One-shot pulse output function	The timing of loading of TDR0n of the master channel is different from that of TDR0p of the slave channel. If TDR0n and TDR0p are rewritten during operation, therefore, an illegal waveform is output. Be sure to rewrite TDR0n and TDR0p after INTTM0n of the channel to be rewritten is generated.	p.264 ☐
			Multiple PWM output function	To rewrite both TDR0n of the master channel and TDR0p of the slave channel 1, write access is necessary at least twice. Since the values of TDR0n and TDR0p are loaded to TCR0n and TCR0p after INTTM0n is generated from the master channel, if rewriting is performed separately before and after generation of INTTM0n from the master channel, the TO0p pin cannot output the expected waveform. To rewrite both TDR0n of the master and TDR0p of the slave, be sure to rewrite both the registers immediately after INTTM0n is generated from the master channel (This applies also to TDR0q of the slave channel 2).	p.271 ☐
Chapter 7	Soft	Real-time counter	PER0: Peripheral enable register 0	When using the real-time counter, first set RTCEN to 1, while oscillation of the subsystem clock (f_{SUB}) is stable. If RTCEN = 0, writing to a control register of the real-time counter is ignored, and, even if the register is read, only the default value is read.	p.281 ☐
				Clock supply to peripheral functions except the real-time counter can be stopped in the HALT mode when operating on the subsystem clock by setting RTCLPC of the operation speed mode control register (OSMC) to 1. In this case, set RTCEN to 1 and bits 0 to 6 of PER0 to 0.	p.281 ☐
			RTCC0: Real-time counter control register 0	If RCLOE0 and RCLOE1 are changed when RTCE = 1, the last waveform of the 32.768 kHz and 1 Hz output signals may become short.	p.282 ☐

(14/37)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 7	Soft	Real-time counter	RTCC1: Real-time counter control register 1	If writing is performed to the RTCC1 register with a 1-bit manipulation instruction, the RIFG and WAFG flags may be cleared. Therefore, to perform writing to the RIFG and WAFG flags, be sure to use an 8-bit manipulation instruction. At this time, set 1 to the RIFG and WAFG flags to invalidate writing and not to clear the RIFG and WAFG flags during writing. When the value may be rewritten because the RIFG and WAFG flags are not being used, the RTCC1 register may be written by using a 1-bit manipulation instruction.	p.284 ☐
			RTCC2: Real-time counter control register 2	Change ICT2, ICT1, and ICT0 when RINTE = 0.	p.285 ☐
	Hard			When the output from RTCDIV pin is stopped, the output continues after a maximum of two clocks of f_{XT} and enters the low level. While 512 Hz is output, and when the output is stopped immediately after entering the high level, a pulse of at least one clock width of f_{SUB} may be generated.	p.285 ☐
				After the real-time counter starts operating, the output width of the RTCDIV pin may be shorter than as set during the first interval period.	p.285 ☐
	Soft		RSUBC: Sub-count register	When a correction is made by using the SUBCUD register, the value may become 8000H or more.	p.286 ☐
				This register is also cleared by reset effected by writing the second count register.	p.286 ☐
				The value read from this register is not guaranteed if it is read during operation, because a value that is changing is read.	p.286 ☐
			HOUR: Hour count register	Bit 5 (HOUR20) of HOUR indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).	p.287 ☐
			WEEK: Week count register	The value corresponding to the month count register or the day count register is not stored in the week count register automatically. After reset release, set the week count register as follow.	p.289 ☐
			ALARMWM: Alarm minute register	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.	p.292 ☐
			ALARMWH: Alarm hour register	Set a decimal value of 00 to 23, or 01 to 12 and 21 to 32 to this register in BCD code. If a value outside the range is set, the alarm is not detected.	p.292 ☐
				Bit 5 (WH20) of ALARMWH indicates AM(0)/PM(1) if AMPM = 0 (if the 12-hour system is selected).	p.292 ☐
			Reading/writing real-time counter	Complete the series of operations of setting RWAIT to 1 to clearing RWAIT to 0 within 1 second.	p.297, 298 ☐
Chapter 8	Soft	Watchdog timer	WDTE: Watchdog timer enable register	If a value other than “ACH” is written to WDTE, an internal reset signal is generated.	p.309 ☐
				If a 1-bit memory manipulation instruction is executed for WDTE, an internal reset signal is generated.	p.309 ☐
				The value read from WDTE is 9AH/1AH (this differs from the written value (ACH)).	p.309 ☐

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 8	Soft	Watchdog timer	Controlling operation	When data is written to WDTE for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.	p.310 ☐
				If the watchdog timer is cleared by writing “ACH” to WDTE, the actual overflow time may be different from the overflow time set by the option byte by up to 2/f_{IL} seconds.	p.310 ☐
				The watchdog timer can be cleared immediately before the count value overflows.	p.310 ☐
				The operation of the watchdog timer in the HALT and STOP modes differs as follows depending on the set value of bit 0 (WDSTBYON) of the option byte (000C0H). (See the table on page 311.) If WDSTBYON = 0, the watchdog timer resumes counting after the HALT or STOP mode is released. At this time, the counter is cleared to 0 and counting starts. When operating with the X1 oscillation clock after releasing the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed. Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset. Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.	p.311 ☐
			Setting overflow time	The watchdog timer continues its operation during self-programming of the flash memory and EEPROM emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.	p.311 ☐
			Setting window open period	When data is written to WDTE for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.	p.312 ☐
				The watchdog timer continues its operation during self-programming of the flash memory and EEPROM emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.	p.312 ☐
				When bit 0 (WDSTBYON) of the option byte (000C0H) = 0, the window open period is 100% regardless of the values of WINDOW1 and WINDOW0.	p.312 ☐
			Setting interval interrupt	When operating with the X1 oscillation clock after releasing the STOP mode, the CPU starts operating after the oscillation stabilization time has elapsed. Therefore, if the period between the STOP mode release and the watchdog timer overflow is short, an overflow occurs during the oscillation stabilization time, causing a reset. Consequently, set the overflow time in consideration of the oscillation stabilization time when operating with the X1 oscillation clock and when the watchdog timer is to be cleared after the STOP mode release by an interval interrupt.	p.313 ☐

(16/37)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 9	Soft	Clock output/ buzzer output controller	CKSn: Clock output select registers n	Change the output clock after disabling clock output (PCLOEn = 0).	p.316 ☐
				If the selected clock (f_{MAIN} or f_{SUB}) stops during clock output (PCLOEn = 1), the output becomes undefined.	p.316 ☐
				To shift to STOP mode when the main system clock is selected (CSELn = 0), set PCLOEn = 0 before executing the STOP instruction. When the subsystem clock is selected (CSELn = 1), PCLOEn = 1 can be set because the clock can be output in STOP mode.	p.316 ☐
Chapter 10	Soft	A/D converter	PER0: Peripheral enable register 0	When setting the A/D converter, be sure to set ADCEN to 1 first. If ADCEN = 0, writing to a control register of the A/D converter is ignored, and, even if the register is read, only the default value is read.	p.322 ☐
			ADM: A/D converter mode register	A/D conversion must be stopped before rewriting bits ADSCM, FR0 to FR2, LV1, and LV0 to values other than the identical data.	p.324 ☐
				When using the A/D converter in normal mode 2 (LV1 = 0, LV0 = 1) or low voltage mode (LV1 = 1, LV0 = 0), enable the input gate voltage boost circuit for the A/D converter by using the analog reference voltage control register (ADVRC), and then set ADCE and ADCS to 1. After the voltage boost circuit stabilization time (10 μs) passes after the input gate voltage boost circuit for the A/D converter has been enabled, set ADCS to 1.	p.324, 325 ☐
			ADM1: A/D converter mode register 1	Rewriting ADM1 during A/D conversion is prohibited. Rewrite it when conversion operation is stopped (ADCS = 0).	p.326 ☐
			ADVRC: Analog reference voltage control register	When using the A/D converter in normal mode 2 (LV1 = 0, LV0 = 1) or low voltage mode (LV1 = 1, LV0 = 0), enable the input gate voltage boost circuit for the A/D converter by using the analog reference voltage control register (ADVRC), and then set ADCE and ADCS to 1. After the voltage boost circuit stabilization time (10 μs) passes after the input gate voltage boost circuit for the A/D converter has been enabled, set ADCS to 1.	p.327 ☐
				To use voltage reference output to the positive reference voltage of the A/D converter, be sure to set VRON to 1 after setting VRSEL to 1.	p.328 ☐
				Do not change the output voltage of the reference voltage by using VRGV during the voltage reference operation (VRON = 1).	p.328 ☐
			ADCR: 12-bit A/D conversion result register	When writing to A/D converter mode register (ADM), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCR may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, and ADPC. Using timing other than the above may cause an incorrect conversion result to be read.	p.328 ☐
			ADCRH: 8-bit A/D conversion result register	When writing to A/D converter mode register (ADM), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCRH may become undefined. Read the conversion result following conversion completion before writing to ADM, ADS, and ADPC. Using timing other than the above may cause an incorrect conversion result to be read.	p.329 ☐

(17/37)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 10	Soft	A/D converter	ADS: Analog input channel specification register	Be sure to clear bits 4 to 7 to “0”.	p.330 ☐
				Set a channel to be used for A/D conversion in the input mode by using port mode registers 2 and 15 (PM2, PM15).	p.330 ☐
				Do not set the pin that is set by ADPC as digital I/O by ADS.	p.330 ☐
				When using an operational amplifier n, the output signal of an operational amplifier n can be used as an analog input.	p.330 ☐
			ADPC: A/D port configuration register	Set a channel to be used for A/D conversion in the input mode by using port mode registers 2 and 15 (PM2, PM15).	p.330 ☐
				Do not set the pin that is set by ADPC as digital I/O by ADS.	p.330 ☐
			PM2, PM15: Port mode registers 2 and 15	If a pin is set as an analog input port, not the pin level but “0” is always read.	p.331 ☐
				When an operational amplifier is used, pins AMPn+, AMPn−, and AMPnO are used, so the alternative analog input functions cannot be used. The operational amplifier output signals, however, can be used as analog inputs.	p.332 ☐
			Basic operations of A/D converter	Make sure the period of <4> to <8> is 1 μs or more.	p.335 ☐
				To use an operational amplifier output for an analog input, start operating the operational amplifier before setting the A/D conversion operation (see CHAPTER 12 OPERATIONAL AMPLIFIER). Furthermore, do not change the operational amplifier setting during the A/D conversion operation.	p.335 ☐
				To use an output voltage of the voltage reference for a positive reference voltage of A/D converter, start operating the voltage reference before setting the A/D conversion operation (see CHAPTER 13 VOLTAGE REFERENCE). Furthermore, do not change the voltage reference setting during the A/D conversion operation.	p.335 ☐
				When using the A/D converter in normal mode 2 (LV1 = 0, LV0 = 1) or low voltage mode (LV1 = 1, LV0 = 0), enable the input gate voltage boost circuit for the A/D converter by using the analog reference voltage control register (ADVRC), and then set ADCE and ADCS to 1. After the voltage boost circuit stabilization time (10 μs) passes after the input gate voltage boost circuit for the A/D converter has been enabled, set ADCS to 1.	p.335 ☐
			A/D conversion operation	Make sure the period of <4> to <8> is 1 μs or more.	p.342 ☐
				<4> may be done between <5> and <7>.	p.342 ☐
				<4> can be omitted. However, ignore data of the first conversion after <8> in this case.	p.342 ☐
				The period from <9> to <13> differs from the conversion time set using bits 5 to 1 (FR2 to FR0, LV1, LV0) of ADM. The period from <12> to <13> is the conversion time set using FR2 to FR0, LV1, and LV0.	p.342 ☐
				To use an operational amplifier output for an analog input, start operating the operational amplifier before setting the A/D conversion operation (see CHAPTER 12 OPERATIONAL AMPLIFIER). Furthermore, do not change the operational amplifier setting during the A/D conversion operation.	p.342 ☐
				To use an output voltage of the voltage reference for a positive reference voltage of A/D converter, start operating the voltage reference before setting the A/D conversion operation (see CHAPTER 13 VOLTAGE REFERENCE). Furthermore, do not change the voltage reference setting during the A/D conversion operation.	p.342 ☐
				When using the A/D converter in normal mode 2 (LV1 = 0, LV0 = 1) or low voltage mode (LV1 = 1, LV0 = 0), enable the input gate voltage boost circuit for the A/D converter by using the analog reference voltage control register (ADVRC), and then set ADCE and ADCS to 1. After the voltage boost circuit stabilization time (10 μs) passes after the input gate voltage boost circuit for the A/D converter has been enabled, set ADCS to 1.	p.342 ☐

(18/37)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 10	Soft	A/D converter	Operating current in STOP mode	Shift to STOP mode after stopping the A/D converter (by setting bit 7 (ADCS) of the A/D converter mode register (ADM) to 0). The operating current can be reduced by setting bit 0 (ADCE) of the A/D converter mode register (ADM) to 0 at the same time. When using normal mode 2 (LV1 = 0, LV0 = 1) or low voltage mode (LV1 = 1, LV0 = 0), clear bit 1 (VRGV) and bit 0 (VRON) of the analog reference voltage control register (ADVRC) to 0, and then shift to STOP mode. To restart from the standby status, clear bit 0 (ADIF) of interrupt request flag register 1L (IF1L) to 0 and start operation.	p.345 ☐
	Hard		Input range of ANI0 to ANI10, ANI15	Observe the rated range of the ANI0 to ANI10, ANI15 input voltage. If a voltage of AV_{DD0} or higher and AV_{SS} or lower (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.	p.345 ☐
	Soft		Conflicting operations	If conflict occurs between A/D conversion result register (ADCR, ADCRH) write and ADCR or ADCRH read by instruction upon the end of conversion ADCR or ADCRH read has priority. After the read operation, the new conversion result is written to ADCR or ADCRH.	p.345 ☐
				If conflict occurs between ADCR or ADCRH write and A/D converter mode register (ADM) write, analog input channel specification register (ADS), or A/D port configuration register (ADPC) write upon the end of conversion ADM, ADS, or ADPC write has priority. ADCR or ADCRH write is not performed, nor is the conversion end interrupt signal (INTAD) generated.	p.345 ☐
	Hard		Noise countermeasures	To maintain the 12-bit resolution, attention must be paid to noise input to the AV_{REFP} pin and pins ANI0 to ANI10, ANI15. <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 C as shown in Figure 10-26 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.	p.345 ☐
	Soft		ANI0 to ANI10, ANI15	The analog input pins (ANI0 to ANI7) are also used as input port pins (P20 to P27). The analog input pins (ANI8 to ANI10, ANI15) are also used as input port pins (P150 to P152, P157). When A/D conversion is performed with any of ANI0 to ANI10, and ANI15 selected, do not access P20 to P27, P150 to P152, and P157 while conversion is in progress; otherwise the conversion resolution may be degraded. It is recommended to select pins used as P20 to P27, P150 to P152, and P157 starting with the ANI0/P20 that is the furthest from AV_{DD0}.	p.346 ☐
	Hard			If the pins adjacent to the pins currently used for A/D conversion are used as digital I/O port, the expected value of the A/D conversion may not be obtained due to coupling noise. Therefore, make sure that digital pulses are not input to or output from the pins adjacent to the pin undergoing A/D conversion.	p.346 ☐
				If any pin among pins of ports 2 and 15 is used as digital output port during A/D conversion, the expected value of the A/D conversion may not be obtained due to coupling noise. Therefore, make sure that digital pulses are not output to pins of ports 2 and 15 during A/D conversion.	p.346 ☐

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 10	Hard	A/D converter	Input impedance of ANI0 to ANI10, ANI15 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, it is recommended to keep the output impedance of the analog input source to within 1 kΩ, and to connect a capacitor of about 100 pF to the ANI0 to ANI10 and ANI15 pins (see Figure 10-26).	p.346 ☐
			AV_{REFP} pin input impedance	A series resistor string of several tens of kΩ is connected between the AV_{REFP} and AV_{REFM} (or AV_{SS}) pins. Therefore, if the output impedance of the reference voltage supply is high, this will result in a series connection to the series resistor string between the AV_{REFP} and AV_{REFM} (or AV_{SS}) pins, resulting in a large reference voltage (AV_{REF}) error of A/D converter.	p.347 ☐
	Soft		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 for the pre-change analog input may be set just before the ADS rewrite. Caution is therefore required since, at this time, when ADIF is read immediately after the ADS rewrite, ADIF 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 before the A/D conversion operation is resumed.	p.347 ☐
			Conversion results just after A/D conversion start	The first A/D conversion value immediately after A/D conversion starts may not fall within the rating range if the ADCS bit is set to 1 within 1 μs after the ADCE bit was set to 1, or if the ADCS bit is set to 1 with the ADCE bit = 0. Take measures such as polling the A/D conversion end interrupt request (INTAD) and removing the first conversion result.	p.347 ☐
			A/D conversion result register (ADCR, ADCRH) read operation	When a write operation is performed to A/D converter mode register (ADM), A/D converter mode register 1 (ADM1), analog input channel specification register (ADS), and A/D port configuration register (ADPC), the contents of ADCR and ADCRH may become undefined. Read the conversion result following conversion completion before writing to ADM, ADM1, ADS, or ADPC. Using a timing other than the above may cause an incorrect conversion result to be read.	p.348 ☐
			Internal equivalent circuit	The equivalent circuit of the analog input block is shown below. (See Figure 10-28.)	p.348 ☐
			Rewriting DACSWn during A/D conversion	Rewriting DACSWn (n = 0, 1) during A/D conversion is prohibited when both the A/D converter reference voltage on the positive side (AD_{REFP}) and the D/A converter reference voltage (DA_{REF}) are the voltage reference output (V_{REFOUT}) (VRSEL = 1 and DAREF = 1). Rewrite it when conversion operation is stopped (ADCS = 0).	p.348 ☐

(20/37)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 11	Soft	D/A converter	PER0: Peripheral enable register 0	When setting the D/A converter, be sure to set DACEN to 1 first. If DACEN = 0, writing to a control register of the D/A converter is ignored, and, even if the register is read, only the default value is read.	p.351 ☐
			DACSW0, DACSW1: D/A conversion value setting registers W0 and W1	Rewriting DACSWn during A/D conversion is prohibited when both the positive reference voltage of the A/D converter (AD_{REFP}) and the positive reference voltage of the D/A converter (DA_{REF}) are the voltage reference output (V_{REFOUT}) (VRSEL = 1 and DAREF = 1). Rewrite it when conversion operation is stopped (ADCS = 0).	p.353, 355 ☐
			Operation of D/A Converter	Even if 1, 0, and then 1 is set to the DACEn bit, there is a wait after 1 is set for the last time.	p.354, 355 ☐
			Operation in normal mode	If the DACSWn or DACSn register is rewritten during the settling time, D/A conversion is aborted and reconversion by using the rewritten values starts.	p.354 ☐
			Operation in real-time output mode	Make the interval between each generation of the INTTM0m signal longer than the settling time. If an INTTM0m signal is generated during the settling time, D/A conversion is aborted and reconversion starts.	p.355 ☐
				Even if the generation of the INTTM0m signal and rewriting the DACSWn or DACSn register conflict, the D/A conversion result is output.	p.355 ☐
			Digital port I/O function, which is the alternate function of the ANO0, ANO1 pins	The digital port I/O function, which is the alternate function of the ANO0 and ANO1 pins, does not operate during D/A conversion. When the P11 register is read during D/A conversion, 0 is read in input mode and the set value of the P11 register is read in output mode. If the digital output mode is set, no output data is output to pins.	p.355 ☐
			Operation of the D/A converter continues in the HALT and STOP mode	The operation of the D/A converter continues in the HALT and STOP mode. To lower the power consumption, therefore, clear the DACEn bit of the DAM register to 0 (D/A conversion stop), and execute HALT or STOP instruction.	p.355 ☐
			Rewriting DACSWn	Rewriting DACSWn (n = 0, 1) during A/D conversion is prohibited when both the positive reference voltage of the A/D converter (AD_{REFP}) and the positive reference voltage of the D/A converter (DA_{REFP}) are the voltage reference output (V_{REFOUT}) (VRSEL = 1 and DAREF = 1). Rewrite it when conversion operation is stopped (ADCS = 0).	p.355 ☐
Chapter 12	Soft	Operational amplifier	PER0: Peripheral enable register 0	When setting operational amplifier, be sure to set ADCEN to 1 first. If ADCEN = 0, writing to a control register of operational amplifier is ignored, and, even if the register is read, only the default value is read.	p.358 ☐
			OAC: Operational amplifier control register	Use the ADPC register to specify as analog inputs the pins to be used with operational amplifiers.	p.359 ☐
				When using as digital inputs the pins of ports 2 and 15, which are not used with operational amplifiers, when the operational amplifiers are used, make sure that the input levels are fixed.	p.359 ☐
			ADPC: A/D port configuration register	Set pins to be used with operational amplifiers in the input mode by using port mode registers 2 and 15 (PM2, PM15).	p.360 ☐
			PM2, PM15: Port mode registers 2 and 15	If a pin is set as an analog input port, not the pin level but “0” is always read.	p.361 ☐
	Hard			When an operational amplifier is used, AMPn+, AMPn–, and AMPnO pins are used, so the alternative analog input functions cannot be used. The operational amplifier output signals, however, can be used as analog inputs.	p.362 ☐
	Soft		Single AMP mode	To use as an input of the A/D converter a voltage that has been amplified in single amplifier mode, enable operation in single amplifier mode before selecting an analog input channel by using the ADS register.	p.364 ☐
Chapter 13	Soft	Voltage reference	PER0: Peripheral enable register 0	When setting voltage reference, be sure to set ADCEN to 1 first. If ADCEN = 0, writing to a control register of voltage reference is ignored, and, even if the register is read, only the default value is read.	p.366 ☐

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 13	Soft	Voltage reference	ADVRC: Analog reference voltage control register	During voltage reference operation, be sure to connect a tantalum capacitor (capacitance: 10 μF±30 %, ESR: 2 Ω (max.), ESL: 10 nH (max.)) and a ceramic capacitor (capacitance: 0.1 μF±30 %, ESR: 2 Ω (max.), ESL: 10 nH (max.)) to the VREFOUT/AVREFP pin for stabilizing the reference voltage. Furthermore, do not apply a voltage from the VREFOUT/AVREFP pin during voltage reference operation.	p.367 ☐
				To use voltage reference output (VREFOUT) to the positive reference voltage of the A/D converter (ADREFP) and the positive reference voltage of the D/A converter (DAREFP), be sure to set VRON to 1 after setting VRSEL to 1.	p.367 ☐
				Rewriting DACSWn (n = 0, 1) during A/D conversion is prohibited when both the positive reference voltage of the A/D converter (ADREFP) and the positive reference voltage of the D/A converter (DAREF) are the voltage reference output (VREFOUT) (VRSEL = 1 and DAREF = 1). Rewrite it when conversion operation is stopped (ADCS = 0).	p.368 ☐
				Do not change the output voltage of the reference voltage by using VRGV during the voltage reference operation (VRON = 1).	p.368 ☐
	Hard		VREFOUT pin	The VREFOUT output voltage can be used only as the positive reference voltage of the internal A/D and D/A converters of the microcontroller. Do not connect an external circuit other than a tantalum capacitor (capacitance: 10 μF±30 %, ESR: 2 Ω (max.), ESL: 10 nH (max.)) and a ceramic capacitor (capacitance: 0.1 μF±30 %, ESR: 2 Ω (max.), ESL: 10 nH (max.)) for stabilizing the reference voltage (target capacitance: 10 μF and 0.1 μF) to the VREFOUT pin.	p.368 ☐
Chapter 14	Soft	Configuration of serial array unit	SDRmn: Lower 8 bits of the serial data register mn	Be sure to clear bit 8 to "0".	p.376 ☐
			PER0: Peripheral enable register 0	When setting serial array unit m, be sure to set SAUmEN to 1 first. 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 input switch control register (ISC), noise filter enable register 0 (NFEN0), port input mode register 1 (PIM1), port output mode registers 1, 8 (POM1, POM8), port mode registers 1, 5, 8 (PM1, PM5, PM8), and port registers 1, 5, 8 (P1, P5, P8)).	p.378 ☐
				After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.	p.378 ☐
			SPSm: Serial clock select register m	Be sure to clear bits 15 to 8 to "0".	p.379 ☐
				After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.	p.379 ☐
			SMRmn: Serial mode register mn	Be sure to clear bits 13 to 9, 7, 4, and 3 to "0". Be sure to set bit 5 to "1".	p.380, 381 ☐
			SCRmn: Serial communication operation setting register mn	Be sure to clear bits 3, 6, and 11 to "0". Be sure to set bit 2 to "1".	p.382 to 384 ☐
			SDRmn: Higher 7 bits of the serial data register mn	Be sure to clear bit 8 to "0".	p.385 ☐
				Setting SDRmn[15:9] = (0000000B, 0000001B) is prohibited when UART is used.	p.385 ☐
				Setting SDRmn[15:9] = 0000000B is prohibited when the simplified I^2C is used. Set SDRmn[15:9] to 0000001B or greater.	p.385 ☐
			SIRmn: Serial flag clear trigger register mn	Be sure to clear bits 15 to 3 to "0".	p.388 ☐

(22/37)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 14	Soft	Configuration of serial array unit	SSm: Serial channel start register m	Be sure to clear bits 15 to 4 to “0”.	p.390 ☐
			STm: Serial channel stop register m	Be sure to clear bits 15 to 4 to “0”.	p.391 ☐
			SOEm: Serial output enable register m	Be sure to clear bits 15 to 3 of SOE0, and bits 1 and 15 to 3 of SOE1 to “0”.	p.392 ☐
			SOm: Serial output register m	Be sure to set bits 11 and 3 of SO0, and bits 11 to 9, 3, and 1 of SO1 to “1”. And be sure to clear bits 15 to 12 and 7 to 4 of SOm to “0”.	p.393 ☐
			SOLm: Serial output level register m	Be sure to clear bits 15 to 3, 1 to “0”.	p.394 ☐
			ISC: Input switch control register	Be sure to clear bits 7 to 5 to “0”.	p.395 ☐
			NFEN0: Noise filter enable register 0	Be sure to clear bits 7, 5, 3, and 1 to “0”.	p.396 ☐
		Operation stop mode	Stopping the operation by units	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 input switch control register (ISC), noise filter enable register 0 (NFEN0), port input mode register 1 (PIM1), port output mode registers 1, 8 (POM1, POM8), port mode registers 1, 5, 8(PM1, PM5, PM8), and port registers 1, 5, 8 (P1, P5, P8)).	p.399 ☐
		3-wire serial I/O (CSI00, CSI10, CSI20) communication	Master transmission	After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.	p.405, 409, 411 ☐
			Master transmission (in continuous transmission mode)	The MDmn0 bit can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.	p.410 ☐
			Master reception	After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.	p.414, 417 ☐
			Master transmission/ reception	After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.	p.420, 423, 425 ☐
			Master transmission/ reception (in continuous transmission/ reception mode)	The MDmn0 bit can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.	p.424 ☐

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 14	Soft	3-wire serial I/O (CSI00, CSI10, CSI20) communication	Slave transmission	After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.	p.428, 432, 434 ☐
			Slave transmission (in continuous transmission mode)	The MDmn0 bit can be rewritten even during operation. However, rewrite it before transfer of the last bit is started.	p.433 ☐
			Slave reception	After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.	p.437, 440 ☐
			Slave transmission/ reception	After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.	p.443, 447,`449 ☐
			Slave transmission/ reception (in continuous transmission/ reception mode)	The MDmn0 bit can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it will be rewritten before the transfer end interrupt of the last transmit data.	p.448 ☐
		UART (UART0, UART1, UART2, UART3) communication	UART communication	When using serial array units 0 and 1 as UARTs, the channels of both the transmitting side (even-number channel) and the receiving side (odd-number channel) can be used only as UARTs.	p.452 ☐
			UART transmission	After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.	p.456, 460, 462 ☐
			UART transmission (in continuous transmission mode)	The MDmn0 bit can be rewritten even during operation. However, rewrite it before transfer of the last bit is started, so that it has been rewritten before the transfer end interrupt of the last transmit data.	p.461 ☐
			UART reception	For the UART reception, be sure to set SMRmr of channel r that is to be paired with channel n.	p.464, 465 ☐
				After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.	p.466, 469 ☐
			Calculating baud rate	Setting SDRmn [15:9] = (0000000B, 0000001B) is prohibited.	p.478 ☐
		Simplified I^2C (IIC10, IIC20) communi-cation	Address field transmission	After setting the SAUmEN to 1, be sure to set the SPSm register after 4 or more clocks have elapsed.	p.485 ☐
			Data reception	ACK is also output when the last data is received. Communication is then completed by setting “1” to the STmn bit to stop operation and generating a stop condition.	p.494 ☐
			Calculating baud rate	Setting SDRmn [15:9] = 0000000B is prohibited. Set SDRmn[15:9] to 0000001B or greater.	p.496 ☐

(24/37)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 15	Soft	Serial interface IICA	IICA: IICA shift register	Do not write data to IICA during data transfer.	p.511 ☐
				Write or read IICA only during the wait period. Accessing IICA in a communication state other than during the wait period is prohibited. When the device serves as the master, however, IICA can be written only once after the communication trigger bit (STT) is set to 1.	p.511 ☐
				When communication is reserved, write data to IICA after the interrupt triggered by a stop condition is detected.	p.511 ☐
			PER0: Peripheral enable register 0	When setting serial interface IICA, be sure to set IICAEN to 1 first. If IICAEN = 0, writing to a control register of serial interface IICA is ignored, and, even if the register is read, only the default value is read.	p.514 ☐
			IICCTL0: IICA control register 0	If the operation of I^2C is enabled (IICE = 1) when the SCL0 line is at high level, the SDA0 line is at low level, and DFC of the IICCTL1 register is 1, a start condition will be inadvertently detected immediately. Immediately after enabling I^2C to operate (IICE = 1), set LREL (1) by using a 1-bit memory manipulation instruction.	p.516 ☐
				When bit 3 (TRC) of the IICA status register (IICS) is set to 1, WREL is set to 1 during the ninth clock and wait is canceled, after which TRC is cleared and the SDA0 line is set to high impedance. Release the wait performed while the TRC bit is 1 (transmission status) by writing to the IICA shift register.	p.519 ☐
			IICS: IICA status register	Reading the IICA status register (IICS) while WUP of IICA control register 1 (IICCTL1) is set to 1 is prohibited. When WUP is changed from 0 to 1, regardless of the INTIICA interrupt request, the change in status is not reflected until the next start condition or stop condition is detected. To use the wakeup mode, therefore, enable (SPIE = 1) the interrupt generated by detecting a stop condition and read the IICS register after the interrupt has been detected.	p.520 ☐
			IICF: IICA flag register	Write to STCEN only when the operation is stopped (IICE = 0).	p.523 ☐
				As the bus release status (IICBSY = 0) is recognized regardless of the actual bus status when STCEN = 1, when generating the first start condition (STT = 1), it is necessary to verify that no third party communications are in progress in order to prevent such communications from being destroyed.	p.523 ☐
				Write to IICRSV only when the operation is stopped (IICE = 0).	p.523 ☐
			Setting transfer clock by using IICWL and IICWH registers	Note the minimum f_{CLK} operation frequency when setting the transfer clock. The minimum f_{CLK} operation frequency for serial interface IICA is determined according to the mode. Fast mode: f_{CLK} = 3.5 MHz (MIN.) Normal mode: f_{CLK} = 1 MHz (MIN.)	p.528 ☐
			Canceling wait	If a processing to cancel a wait state executed when WUP (bit 7 of IICA control register 1 (IICCTL1)) = 1, the wait state will not be canceled.	p.535 ☐
			When STCEN (bit 1 of IICA flag register (IICF)) = 0	Immediately after I^2C operation is enabled (IICE = 1), the bus communication status (IICBSY (bit 6 of IICF) = 1) is recognized regardless of the actual bus status. When changing from a mode in which no stop condition has been detected to a master device communication mode, first generate a stop condition to release the bus, then perform master device communication. When using multiple masters, it is not possible to perform master device communication when the bus has not been released (when a stop condition has not been detected). Use the following sequence for generating a stop condition. <1> Set IICA control register 1 (IICCTL1). <2> Set bit 7 (IICE) of IICA control register 0 (IICCTL0) to 1. <3> Set bit 0 (SPT) of IICCTL0 to 1.	p.547 ☐
			When STCEN = 1	Immediately after I^2C operation is enabled (IICE = 1), the bus released status (IICBSY = 0) is recognized regardless of the actual bus status. To generate the first start condition (STT = 1), it is necessary to confirm that the bus has been released, so as to not disturb other communications.	p.547 ☐

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 15	Soft	Serial interface IICA	If other I^2C communications are already in progress	If I^2C operation is enabled and the device participates in communication already in progress when the SDA0 pin is low and the SCL0 pin is high, the macro of I^2C recognizes that the SDA0 pin has gone low (detects a start condition). If the value on the bus at this time can be recognized as an extension code, $\overline{ACK}$ is returned, but this interferes with other I^2C communications. To avoid this, start I^2C in the following sequence. <1> Clear bit 4 (SPIE) of IICCTL0 to 0 to disable generation of an interrupt request signal (INTIICA) when the stop condition is detected. <2> Set bit 7 (IICE) of IICCTL0 to 1 to enable the operation of I^2C. <3> Wait for detection of the start condition. <4> Set bit 6 (LREL) of IICCTL0 to 1 before $\overline{ACK}$ is returned (4 to 80 clocks after setting IICE to 1), to forcibly disable detection.	p.547 ☐
			STT, SPT: Bits 1, 0 of IICA control register 0 (IICCTL0)	Setting STT and SPT (bits 1 and 0 of IICCTL0) again after they are set and before they are cleared to 0 is prohibited.	p.547 ☐
			Reserving transmission	When transmission is reserved, set SPIE (bit 4 of IICTL0) to 1 so that an interrupt request is generated when the stop condition is detected. Transfer is started when communication data is written to IICA after the interrupt request is generated. Unless the interrupt is generated when the stop condition is detected, the device stops in the wait state because the interrupt request is not generated when communication is started. However, it is not necessary to set SPIE to 1 when MSTS (bit 7 of IICS) is detected by software.	p.547 ☐
Chapter 16	Soft	Multiplier/divider	MDAH, MDAL: Multiplication/division data register A	Do not rewrite the MDAH and MDAL values during division operation processing (while the multiplication/division control register (MDUC) is 81H). The operation will be executed in this case, but the operation result will be an undefined value.	p.594 ☐
				The MDAH and MDAL values read during division operation processing (while MDUC is 81H) will not be guaranteed.	p.594 ☐
			MDBL, MDBH: Multiplication/division data register B	Do not rewrite the MDBH and MDBL values during division operation processing (while the multiplication/division control register (MDUC) is 81H). The operation result will be an undefined value.	p.595 ☐
				Do not set MDBH and MDBL to 0000H in the division mode. If they are set, the operation result will be an undefined value.	p.595 ☐
			MDCL, MDCH: Multiplication/division data register C	The MDCH and MDCL values read during division operation processing (while the multiplication/division control register (MDUC) is 81H) will not be guaranteed.	p.596 ☐
			MDUC: Multiplication/division control register 0	Do not rewrite DIVMODE during operation processing (while DIVST is 1). If it is rewritten, the operation result will be an undefined value.	p.597 ☐
				DIVST cannot be cleared (0) by using software during division operation processing (while DIVST is 1).	p.597 ☐

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 17	Soft	DMA controller	DBCn: DMA byte count register n	Be sure to clear bits 15 to 10 to “0”.	p.604 ☐
				If the general-purpose register is specified or the internal RAM space is exceeded as a result of continuous transfer, the general-purpose register or SFR space are written or read, resulting in loss of data in these spaces. Be sure to set the number of times of transfer that is within the internal RAM space.	p.604 ☐
			DRCn: DMA operation control register n	The DSTn flag is automatically cleared to 0 when a DMA transfer is completed. Writing the DENn flag is enabled only when DSTn = 0. When a DMA transfer is terminated without waiting for generation of the interrupt (INTDMAn) of DMAn, therefore, set DSTn to 0 and then DENn to 0 (for details, refer to 17.5.7 Forced termination by software).	p.607 ☐
				When the FSEL bit of the OSMC register has been set to 1, do not enable (DENn = 1) DMA operation for at least three clocks after the setting.	p.607 ☐
			Example of Setting for Holding DMA Transfer Pending by DWAITn	When DMA transfer is held pending while using both DMA channels, be sure to held the DMA transfer pending for both channels (by setting DWAIT0 and DWAIT1 to 1). If the DMA transfer of one channel is executed while that of the other channel is held pending, DMA transfer might not be held pending for the latter channel.	p.619 ☐
			Forced Termination of DMA Transfer	In example 3, the system is not required to wait two clock cycles after the DWAITn bit is set to 1. In addition, the system does not have to wait two clock cycles after clearing the DSTn bit to 0, because more than two clock cycles elapse from when the DSTn bit is cleared to 0 to when the DENn bit is cleared to 0.	p.621 ☐
			Priority	During DMA transfer, a request from the other DMA channel is held pending even if generated. The pending DMA transfer is started after the ongoing DMA transfer is completed. If two DMA requests are generated at the same time, however, DMA channel 0 takes priority over DMA channel 1. If a DMA request and an interrupt request are generated at the same time, the DMA transfer takes precedence, and then interrupt servicing is executed.	p.622 ☐
	Hard		Response time	The response time of DMA transfer is as follows. (See Table 17-2.)	p.622 ☐
	Soft		Operation in standby mode	The DMA controller operates as follows in the standby mode. (See Table 17-3.)	p.623 ☐
			DMA pending instruction	Even if a DMA request is generated, DMA transfer is held pending immediately after the following instructions. • CALL !addr16 • CALL $!addr20 • CALL !!addr20 • CALL rp • CALLT [addr5] • BRK • Bit manipulation instructions for registers IF0L, IF0H, IF1L, IF1H, IF2L, IF2H, MK0L, MK0H, MK1L, MK1H, MK2L, MK2H, PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H and PSW each.	p.623 ☐

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 17	Soft	DMA controller	Operation if address in general-purpose register area or other than those of internal RAM area is specified	The address indicated by DRA0n is incremented during DMA transfer. If the address is incremented to an address in the general-purpose register area or exceeds the area of the internal RAM, the following operation is performed. ● In mode of transfer from SFR to RAM The data of that address is lost. ● In mode of transfer from RAM to SFR Undefined data is transferred to SFR. In either case, malfunctioning may occur or damage may be done to the system. Therefore, make sure that the address is within the internal RAM area other than the general-purpose register area.	p.623 ☐
Chapter 18	Soft	Interrupt functions	IF0L, IF0H, IF1L, IF1H, IF2L, IF2H: Interrupt request flag registers	When operating a timer, serial interface, or A/D converter after standby release, operate it once after clearing the interrupt request flag. An interrupt request flag may be set by noise.	p.632 ☐

				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 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.
					p.632 ☐

				Be sure to clear bit 3 of IF1H, bits 6, 7 of IF2H to 0.	p.633 ☐
			MK0L, MK0H, MK1L, MK1H, MK2L, MK2H: Interrupt mask flag registers	Be sure to set bit 3 of MK1H, bits 6, 7 of MK2H to 1.	p.634 ☐
			PR00L, PR00H, PR01L, PR01H, PR02L, PR02H, PR10L, PR10H, PR11L, PR11H, PR12L, PR12H: Priority specification flag registers	Be sure to set bit 3 of PR01H and PR11H, bits 6, 7 of PR02H and PR12H to 1.	p.636 ☐

(28/37)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 18	Soft	Interrupt functions	EGP0, EGP1: External interrupt rising edge enable registers, EGN0, EGN1: External interrupt falling edge enable registers	Select the port mode by clearing EGPn and EGNn to 0 because an edge may be detected when the external interrupt function is switched to the port function.	p.638 ☐
			Software interrupt request acknowledgment	Do not use the RETI instruction for restoring from the software interrupt.	p.641 ☐
			BRK instruction	The BRK instruction is not one of the above-listed interrupt request hold instructions. However, the software interrupt activated by executing the BRK instruction causes the IE flag to be cleared. Therefore, even if a maskable interrupt request is generated during execution of the BRK instruction, the interrupt request is not acknowledged.	p.645 ☐
Chapter 19	Soft	Standby function	–	The STOP mode can be used only when the CPU is operating on the main system clock. The STOP mode cannot be set while the CPU operates with the subsystem clock. The HALT mode can be used when the CPU is operating on either the main system clock or the subsystem clock.	p.646 ☐
				When shifting to the STOP mode, be sure to stop the peripheral hardware operation operating with main system clock before executing STOP instruction.	p.646 ☐
				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 the A/D converter mode register (ADM) to 0 to stop the A/D conversion operation, and then execute the STOP instruction.	p.646 ☐
				It can be selected by the option byte whether the internal low-speed oscillator continues oscillating or stops in the HALT or STOP mode. For details, see CHAPTER 24 OPTION BYTE.	p.646 ☐
				The STOP instruction cannot be executed when the CPU operates on the 20 MHz internal high-speed oscillation clock. Be sure to execute the STOP instruction after shifting to internal high-speed oscillation clock operation.	p.646 ☐
			OSTC: Oscillation stabilization time counter status register	After the above time has elapsed, the bits are set to 1 in order from MOST8 and remain 1.	p.648 ☐
				The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS. If the STOP mode is entered and then released while the internal high-speed oscillation clock is being used as the CPU clock, set the oscillation stabilization time as follows. • Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS Note, therefore, that only the status up to the oscillation stabilization time set by OSTS is set to OSTC after STOP mode is released.	p.648 ☐
	Hard			The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts (“a” below).	p.648 ☐

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 19	Soft	Standby function	OSTS: Oscillation stabilization time select register	To set the STOP mode when the X1 clock is used as the CPU clock, set OSTS before executing the STOP instruction.	p.649 ☐
				Setting the oscillation stabilization time to 20 μs or less is prohibited.	p.649 ☐
				Before changing the setting of the OSTS register, confirm that the count operation of the OSTC register is completed.	p.649 ☐
				Do not change the value of the OSTS register during the X1 clock oscillation stabilization time.	p.649 ☐
				The oscillation stabilization time counter counts up to the oscillation stabilization time set by OSTS. If the STOP mode is entered and then released while the internal high-speed oscillation clock is being used as the CPU clock, set the oscillation stabilization time as follows. • Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS Note, therefore, that only the status up to the oscillation stabilization time set by OSTS is set to OSTC after STOP mode is released.	p.649 ☐
	Hard			The X1 clock oscillation stabilization wait time does not include the time until clock oscillation starts (“a” below).	p.649 ☐
	Soft		STOP mode	Because the interrupt request signal is used to clear the standby mode, if there is an interrupt source with the interrupt request flag set and the interrupt mask flag reset, the standby mode is immediately cleared if set. Thus, the STOP mode is reset to the HALT mode immediately after execution of the STOP instruction and the system returns to the operating mode as soon as the wait time set using the oscillation stabilization time select register (OSTS) has elapsed.	p.656 ☐
				The STOP instruction cannot be executed when the CPU operates on the 20 MHz internal high-speed oscillation clock. Be sure to execute the STOP instruction after shifting to internal high-speed oscillation clock operation.	p.656, 658 ☐
				To use the peripheral hardware that stops operation in the STOP mode, and the peripheral hardware for which the clock that stops oscillating in the STOP mode after the STOP mode is released, restart the peripheral hardware.	p.658 ☐
				To stop the internal low-speed oscillation clock in the STOP mode, use an option byte to stop the watchdog timer operation in the HALT/STOP mode (bit 0 (WDSTBYON) of 000C0H = 0), and then execute the STOP instruction.	p.658 ☐
				To shorten oscillation stabilization time after the STOP mode is released when the CPU operates with the high-speed system clock (X1 oscillation), temporarily switch the CPU clock to the internal high-speed oscillation clock before the next execution of the STOP instruction. Before changing the CPU clock from the internal high-speed oscillation clock to the high-speed system clock (X1 oscillation) after the STOP mode is released, check the oscillation stabilization time with the oscillation stabilization time counter status register (OSTC).	p.658 ☐

(30/37)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 20	Hard	Reset function	–	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, a low level of at least 10 μs must be continued during the period in which the supply voltage is within the operating range ($V_{DD} \geq 1.8$ V).)	p.662 ☐
				During reset input, the X1 clock, XT1 clock, internal high-speed oscillation clock, and internal low-speed oscillation clock stop oscillating. External main system clock input becomes invalid.	p.662 ☐
				When the STOP mode is released by a reset, the RAM contents in the STOP mode are held during reset input.	p.662 ☐
				When reset is effected, port pin P130 is set to low-level output and other port pins become high-impedance, because each SFR and 2nd SFR are initialized.	p.662 ☐
	Soft		Block diagram of reset function	An LVI circuit internal reset does not reset the LVI circuit.	p.663 ☐
			Watchdog timer overflow	A watchdog timer internal reset resets the watchdog timer.	p.664 ☐
			RESF: Reset control flag register	Do not read data by a 1-bit memory manipulation instruction.	p.671 ☐
				Do not make a judgment based on only the read value of the RESF register 8-bit data, because bits other than TRAP, WDRF, and LVIRF become undefined.	p.671 ☐
				When the LVI default start function (bit 0 (LVIOFF) of 000C1H = 0) is used, LVIRF flag may become 1 from the beginning depending on the power-on waveform.	p.671 ☐
Chapter 21	Soft	Power-on-clear circuit	–	If the low-voltage detector (LVI) is set to ON by an option byte by default, the reset signal is not released until the supply voltage (V_{DD}) exceeds 2.07 V ±0.2 V.	p.672, 673 ☐
				If an internal reset signal is generated in the POC circuit, the reset control flag register (RESF) is cleared to 00H.	p.672 ☐
			Timing of generation of internal reset signal (LVIOFF = 1)	Set the low-voltage detector by software after the reset status is released (see CHAPTER 22 LOW-VOLTAGE DETECTOR).	p.674 ☐
			Timing of generation of internal reset signal (LVIOFF = 0)	Set the low-voltage detector by software after the reset status is released (see CHAPTER 22 LOW-VOLTAGE DETECTOR).	p.675 ☐
			Cautions for power-on-clear circuit	In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the POC detection voltage (V_{POR}, V_{PDR}), 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.	p.676 ☐
Chapter 22	Soft	Low-voltage detector	LVIM: Low-voltage detection register	To stop LVI, be sure to clear (0) LVION by using a 1-bit memory manipulation instruction.	p.681 ☐
	Hard			Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.	p.681 ☐
	Soft			When LVI is used in interrupt mode (LVIMD = 0) and LVISEL is set to 0, an interrupt request signal (INTLVI) that disables LVI operation (clears LVION) when the supply voltage (V_{DD}) is less than or equal to the detection voltage (V_{LVI}) (if LVISEL = 1, input voltage of external input pin (EXLVI) is less than or equal to the detection voltage (V_{EXLVI})) is generated and LVIIF may be set to 1.	p.681 ☐

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 22	Soft	Low-voltage detector	LVIS: Low-voltage detection level select register	Be sure to clear bits 4 to 7 to “0”.	p.682 ☐
				Change the LVIS value with either of the following methods. • When changing the value after stopping LVI <1> Stop LVI (LVION = 0). <2> Change the LVIS register. <3> Set to the mode used as an interrupt (LVIMD = 0). <4> Mask LVI interrupts (LVIMK = 1). <5> Enable LVI operation (LVION = 1). <6> Before cancelling the LVI interrupt mask (LVIMK = 0), clear it with software because an LVIIF flag may be set when LVI operation is enabled. • When changing the value after setting to the mode used as an interrupt (LVIMD = 0) <1> Mask LVI interrupts (LVIMK = 1). <2> Set to the mode used as an interrupt (LVIMD = 0). <3> Change the LVIS register. <4> Before cancelling the LVI interrupt mask (LVIMK = 0), clear it with software because an LVIIF flag may be set when the LVIS register is changed.	p.683 ☐
				When an input voltage from the external input pin (EXLVI) is detected, the detection voltage (V_{EXLVI}) is fixed. Therefore, setting of LVIS is not necessary.	p.683 ☐
			Used as reset (when detecting level of supply voltage (V_{DD})) (LVIOFF = 1)	Be sure to execute <1>. When LVIMK = 0, an interrupt may occur immediately after the processing in <4>.	p.685 ☐
				If supply voltage (V_{DD}) ≥ detection voltage (V_{LVI}) when LVIMD is set to 1, an internal reset signal is not generated.	p.685 ☐
			Used as reset (when detecting level of supply voltage (V_{DD})) (LVIOFF = 0)	Even when the LVI default start function is used, if it is set to LVI operation prohibition by the software, it operates as follows: • Does not perform low-voltage detection during LVION = 0. • If a reset is generated while LVION = 0, LVION will be re-set to 1 when the CPU starts after reset release. There is a period when low-voltage detection cannot be performed normally, however, when a reset occurs due to WDT and illegal instruction execution. This is due to the fact that while the pulse width detected by LVI must be 200 μs max., LVION = 1 is set upon reset occurrence, and the CPU starts operating without waiting for the LVI stabilization time.	p.687 ☐
			Used as reset (when detecting level of input voltage from external input pin (EXLVI))	Be sure to execute <1>. When LVIMK = 0, an interrupt may occur immediately after the processing in <3>.	p.689 ☐
				If input voltage from external input pin (EXLVI) ≥ detection voltage (V_{EXLVI} = 1.21 V (TYP.)) when LVIMD is set to 1, an internal reset signal is not generated.	p.689 ☐
	Hard			Input voltage from external input pin (EXLVI) must be EXLVI < V_{DD}.	p.689 ☐

(32/37)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 22	Soft	Low-voltage detector	Used as interrupt (when detecting level of supply voltage (V_{DD})) (LVIOFF = 0)	Even when the LVI default start function is used, if it is set to LVI operation prohibition by the software, it operates as follows: • Does not perform low-voltage detection during LVION = 0. • If a reset is generated while LVION = 0, LVION will be re-set to 1 when the CPU starts after reset release. There is a period when low-voltage detection cannot be performed normally, however, when a reset occurs due to WDT and illegal instruction execution. This is due to the fact that while the pulse width detected by LVI must be 200 μs max., LVION = 1 is set upon reset occurrence, and the CPU starts operating without waiting for the LVI stabilization time.	p.693 ☐
				When the LVI default start function (bit 0 (LVIOFF) of 000C1H = 0) is used, the LVIRF flag may become 1 from the beginning due to the power-on waveform. For details of RESF, see CHAPTER 20 RESET FUNCTION.	p.693 ☐
	Hard		Used as interrupt (when detecting level of input voltage from external input pin (EXLVI))	Input voltage from the external input pin (EXLVI) must be EXLVI < V_{DD}.	p.695 ☐
	Soft		Cautions for low-voltage detector	In a system where the supply voltage (V_{DD}) fluctuates for a certain period in the vicinity of the LVI detection voltage (V_{LVI}), the operation is as follows depending on how the low-voltage detector is used. Operation example 1: When used as reset The system may be repeatedly reset and released from the reset status. The time from reset release through microcontroller operation start can be set arbitrarily by 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 (see Figure 22-11). Operation example 2: When used as interrupt Interrupt requests may be generated frequently. Take the following action. <Action> Confirm that "supply voltage (V_{DD}) ≥ detection voltage (V_{LVI})" when detecting the falling edge of V_{DD}, or "supply voltage (V_{DD}) < detection voltage (V_{LVI})" when detecting the rising edge of V_{DD}, in the servicing routine of the LVI interrupt by using bit 0 (LVIF) of the low-voltage detection register (LVIM). Clear bit 1 (LVIIF) of interrupt request flag register 0L (IF0L) to 0. For a system with a long supply voltage fluctuation period near the LVI detection voltage, take the above action after waiting for the supply voltage fluctuation time.	p.697 to 700 ☐
	Hard			There is some delay from the time supply voltage (V_{DD}) < LVI detection voltage (V_{LVI}) until the time LVI reset has been generated. In the same way, there is also some delay from the time LVI detection voltage (V_{LVI}) ≤ supply voltage (V_{DD}) until the time LVI reset has been released (see Figure 22-12**)**.	p.700 ☐

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 23	Soft	Regulator	RMC: Regulator mode control register	The RMC register can be rewritten only in the low-power consumption mode (refer to Table 23-1). In other words, rewrite this register during CPU operation with the subsystem clock (f_{XT}) while the high-speed system clock (f_{MX}), the high-speed internal oscillation clock (f_{IH}), and the 20 MHz internal high-speed oscillation clock (f_{IH20}) are both stopped.	p.701 ☐
				When using the setting fixed to the low consumption current mode, the RMC register can be used in the following cases. <When the high-speed internal oscillation clock (f_{IH} = 8 MHz (TYP.) or f_{IH} = 1 MHz (TYP.)) is selected as the CPU clock> $f_{CLK} \leq$ 1 MHz and external oscillator (X1 clock (f_X), external main system clock (f_{EX})) stop. <When the X1 clock (f_X) or external main system clock (f_{EX}) is selected as the CPU clock> $f_{CLK} \leq$ 1 MHz, $f_X/f_{EX} \leq$ 5 MHz and the internal high-speed oscillator stop. <When the subsystem clock (f_{SUB}) is selected as the CPU clock> Both the internal high-speed oscillator and external oscillator ($f_X/f_{EX} \leq$ 5 MHz) stop or either one stops.	p.701 ☐
				In low-power consumption mode, use the regulator with f_{CLK} fixed to 1 MHz when executing self programming.	p.702 ☐
				A wait is required to change the operation speed mode control register (OSMC) after changing the RMC register. Wait for 2 ms by software when setting to low-power consumption mode and 10 μs when setting to normal power mode, as described in the procedure shown below. • When setting to low-power consumption mode <1> Select a frequency of 1 MHz for f_{CLK}. <2> Set RMC to 5AH (set the regulator to low-power consumption mode). <3> Wait for 2 ms. <4> Set FLPC and FSEL of OSMC to 1 and 0, respectively. • When setting to normal power mode <1> Set RMC to 00H (set the regulator to normal power mode). <2> Wait for 10 μs. <3> Change FLPC and FSEL of OSMC. <4> Change the f_{CLK} frequency.	p.702 ☐
Chapter 24	Soft	Option byte	000C2H/010C2H	Be sure to set FFH to 000C2H (000C2H/010C2H when the boot swap operation is used).	p.703 ☐
			000C0H/010C0H	Set the same value as 000C0H to 010C0H when the boot swap operation is used because 000C0H is replaced by 010C0H.	p.703 ☐
			000C1H/010C1H	Set the same value as 000C1H to 010C1H when the boot swap operation is used because 000C1H is replaced by 010C1H.	p.703 ☐
			000C2H/010C2H	Set FFH to 010C2H when the boot swap operation is used because 000C2H is replaced by 010C2H.	p.703 ☐
			000C3H/010C3H	Set the same value as 000C3H to 010C3H when the boot swap operation is used because 000C3H is replaced by 010C3H.	p.704 ☐
			000C0H/010C0H	The watchdog timer continues its operation during self-programming of the flash memory and EEPROM emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.	p.705 ☐

(34/37)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 24	Soft	Option byte	000C1H/010C1H	Be sure to set bits 7 to 3 to “1”.	p.706 ☐
				Even when the LVI default start function is used, if it is set to LVI operation prohibition by the software, it operates as follows: • Does not perform low-voltage detection during LVION = 0. • If a reset is generated while LVION = 0, LVION will be re-set to 1 when the CPU starts after reset release. There is a period when low-voltage detection cannot be performed normally, however, when a reset occurs due to WDT and illegal instruction execution. This is due to the fact that while the pulse width detected by LVI must be 200 μs max., LVION = 1 is set upon reset occurrence, and the CPU starts operating without waiting for the LVI stabilization time.	p.706 ☐
			000C3H/010C3H	Bits 7 and 0 (OCDENSET and OCDERSD) can only be specified a value. Be sure to set 000010B to bits 6 to 1.	p.706 ☐
			Setting of option byte	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.	p.707 ☐
Chapter 25	Hard	Flash memory	Security settings	After the security setting for the batch erase is set, erasure cannot be performed for the device. In addition, even if a write command is executed, data different from that which has already been written to the flash memory cannot be written, because the erase command is disabled.	p.716 ☐
				If a security setting that rewrites boot cluster 0 has been applied, boot cluster 0 of that device will not be rewritten, and the entire flash memory of the device will not be erased in batch.	p.716 ☐
			Flash memory programming by self-programming	The self-programming function cannot be used when the CPU operates with the subsystem clock.	p.718 ☐
	Soft			In the self-programming mode, call the self-programming start library (FlashStart).	p.718 ☐
				To prohibit an interrupt during self-programming, in the same way as in the normal operation mode, execute the 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 self-programming library.	p.718 ☐
				In low-power-consumption mode, use the regulator with f_{CLK} fixed to 1 MHz when executing self programming. For details of the low-power-consumption mode, see CHAPTER 23 REGULATOR.	p.718 ☐
				Disable DMA operation (DENn = 0) during the execution of self programming library functions.	p.718 ☐
			Flash shield window function	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.	p.722 ☐
Chapter 26	Hard	On-chip debug function	Connecting QB-MINI2 to 78K0R/Kx3-A microcontrollers	The 78K0R/Kx3-A 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.	p.724 ☐

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 27	Soft	BCD correction circuit	Addition	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.	p.728 ☐
			Subtraction	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.	p.729 ☐
Chapter 28	Soft	Instruction set	PREFIX instruction	Set the ES register value with MOV ES, A, etc., before executing the PREFIX instruction.	p.732 ☐
Chapter 29	Hard	Electrical specifications	–	The 78K0R/Kx3-A 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.	p.750 ☐
			Absolute maximum ratings	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.	p. 750 to 752 ☐
				The value of the current that can be run per pin must satisfy the value of the current per pin and the total value of the currents of all pins.	p.751, 752 ☐
			X1 oscillator characteristics	When using the X1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance. • Keep the wiring length as short as possible. • Do not cross the wiring with the other signal lines. • Do not route the wiring near a signal line through which a high fluctuating current flows. • Always make the ground point of the oscillator capacitor the same potential as V_{SS}. • Do not ground the capacitor to a ground pattern through which a high current flows. • Do not fetch signals from the oscillator.	p.753 ☐
				Since the CPU is started by the internal high-speed oscillation 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 oscillation stabilization time select register (OSTS) after sufficiently evaluating the oscillation stabilization time with the resonator to be used.	p.753 ☐

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 29	Hard	Electrical specifications	XT1 oscillator characteristics	When using the XT1 oscillator, wire as follows in the area enclosed by the broken lines in the above figures to avoid an adverse effect from wiring capacitance. • Keep the wiring length as short as possible. • Do not cross the wiring with the other signal lines. • Do not route the wiring near a signal line through which a high fluctuating current flows. • Always make the ground point of the oscillator capacitor the same potential as V_{SS}. • Do not ground the capacitor to a ground pattern through which a high current flows. • Do not fetch signals from the oscillator.	p.754 ☐
				The XT1 oscillator is designed as a low-amplitude circuit for reducing power consumption, and is more prone to malfunction due to noise than the X1 oscillator. Particular care is therefore required with the wiring method when the XT1 clock is used.	p.754 ☐
			Recommended oscillator circuit constants	The oscillator constants shown above are reference values based on evaluation in a specific environment by the resonator manufacturer. If it is necessary to optimize the oscillator characteristics in the actual application, apply to the resonator manufacturer for evaluation on the implementation circuit. The oscillation voltage and oscillation frequency only indicate the oscillator characteristic. Use the 78K0R/Kx3-A so that the internal operation conditions are within the specifications of the DC and AC characteristics.	p.755, 756 ☐
			DC characteristics	P10 to P15, P80 and P82 do not output high level in N-ch open-drain mode.	p.758, 762 ☐
				The maximum value of V_{IH} of pins P10 to P15, P80 and P82 is V_{DD}, even in the N-ch open-drain mode.	p.760 ☐
	Soft		Minimum instruction execution time during main system clock operation	When V_{DD} < 2.25 V and FSEL = 1, It is prohibited to release STOP mode during f_{EX} operation or f_{IH} operation (This must not be performed even if the frequency is divided. The STOP mode may be released during f_X operation.).	p.770 ☐
			During communication at same potential (UART mode) (dedicated baud rate generator output)	Select the normal input buffer for RxDq and the normal output mode for TxDq by using the PIMg and POMx registers.	p.774 ☐
			During communication at same potential (CSI mode) (master mode, $\overline{\text{SCKp}}$... internal clock output)	Select the normal input buffer for SIp and the normal output mode for SOp and $\overline{\text{SCKp}}$ by using the PIMg and POMx registers.	p.775 ☐
			During communication at same potential (CSI mode) (slave mode, $\overline{\text{SCKp}}$... external clock input)	Select the normal input buffer for SIp and $\overline{\text{SCKp}}$ and the normal output mode for SOp by using the PIMg and POMx registers.	p.776 ☐
			During communication at same potential (simplified I^2C mode)	Select the normal input buffer and the N-ch open drain output (V_{DD} tolerance) mode for SDAr and the normal output mode for SCLr by using the PIMg and POMx registers.	p.778 ☐

(37/37)

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 29	Soft	Electrical specifications	During communication at different potential (2.5 V, 3 V) (UART mode) (dedicated baud rate generator output)	Select the TTL input buffer for RxDq and the N-ch open drain output (V_{DD} tolerance) mode for TxDq by using the PIMg and POMx registers.	p.780, 781, 783 ☐
			During communication at different potential (2.5 V, 3 V) (CSI mode) (master mode, $\overline{SCKp}$... internal clock output)	Select the TTL input buffer for SIp and the N-ch open drain output (V_{DD} tolerance) mode for SOp and $\overline{SCKp}$ by using the PIMg and POMx registers.	p.784 to 786 ☐
			During communication at different potential (2.5 V, 3 V) (CSI mode) (slave mode, $\overline{SCKp}$... external clock input)	Select the TTL input buffer for SIp and $\overline{SCKp}$ and the N-ch open drain output (V_{DD} tolerance) mode for SOp by using the PIMg and POMx registers.	p.788, 789 ☐
			During communication at different potential (2.5 V, 3 V) (simplified I^2C mode)	Select the TTL input buffer and the N-ch open drain output (V_{DD} tolerance) mode for SDAr and the N-ch open drain output (V_{DD} tolerance) mode for SCLr by using the PIMg and POMx registers.	p.790, 791 ☐
	Hard		VR circuit	Connect the V_{REFOUT} pin to GND via a tantalum capacitor (capacitance: 10 μF±30 %, ESR: 2 Ω (max.), ESL: 10 nH (max.)) and a ceramic capacitor (capacitance: 0.1 μF±30 %, ESR: 2 Ω (max.), ESL: 10 nH (max.)).	p.795 ☐
Chapter 31	Hard	Recommended soldering condition	–	For soldering methods and conditions other than those recommended below, contact a Renesas Electronics sales representative.	p.802 ☐
				Do not use different soldering methods together (except for partial heating).	p.802 ☐

<R>

APPENDIX D REVISION HISTORY

(1/3)

Page	Description	Classification
Throughout	Change URL of Renesas Electronics website	–
	Change names of A/D conversion modes • conversion mode 1 → normal mode 1 • conversion mode 2 → normal mode 2 • conversion mode 3 → low voltage mode	(b)
	Deletion of target from the capacitance value of the capacitor connected to the REGC pin	(b)
CHAPTER 5 CLOCK GENERATOR		
pp.152, 156, 157	Addition of description of 20 MHz internal high-speed oscillation clock oscillator	(b)
p.160	Change of description of the wait time of the FSEL	(b)
p.166	Change of **5.4.3 Internal high-speed oscillator**	(c)
pp.168 to 171	Change of **Figure 5-13. Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Stopped Is Set (Option Byte: LVIOFF = 1))** and **Figure 5-14. Clock Generator Operation When Power Supply Voltage Is Turned On (When LVI Default Start Function Enabled Is Set (Option Byte: LVIOFF = 0))**	(c)
p.180	Change of **5.6.5 CPU clock status transition diagram**	(c)
CHAPTER 8 WATCHDOG TIMER		
p.312	Change of **8.4.3 Setting window open period of watchdog timer**	(b)
CHAPTER 10 A/D CONVERTER		
Throughout	Deletion of TBD from operation stabilization time 1 μs of A/D voltage comparator	(b)
pp.324, 325, 327, 335, 342	Change of voltage boost circuit stabilization time (TBD) to (10 μs)	(b)
p.325	Addition of **Note 4** to **Table 10-2. A/D Conversion Time Selection**	(b)
p.346	Deletion of TBD from the output impedance within 1 kΩ of the analog input source	(b)
p.348	Change of **Table 10-8. Resistance and Capacitance Values of Equivalent Circuit (Reference Values)**	(b)
CHAPTER 11 D/A CONVERTER		
pp.354, 355	Change of wait time to 20 μs or more	(b)
pp.354, 355	Deletion of TBD from the settling time (18 μs (MAX.))	(b)
CHAPTER 14 SERIAL ARRAY UNIT		
p.385	Change of **Caution 3** of **Figure 14-8. Format of Serial Data Register mn (SDRmn)**	(c)
p.496	Change of **Caution** of **14.7.5 Calculating transfer rate**	(c)
CHAPTER 15 SERIAL INTERFACE IICA		
p.519	Addition of description to **Caution** of **Figure 15-6. Format of IICA Control Register 0 (IICCTL0) (4/4)**	(c)
p.521	Addition of description to **Note** of **Figure 15-7. Format of IICA Status Register (IICS) (2/3)**	(c)
p.524	Addition of **Note** to **Figure 15-9. Format of IICA Control Register 1 (IICCTL1) (1/2)**	(c)
p.528	Change of **15.4.2 Setting transfer clock by using IICWL and IICWH registers**	(c)
pp.577 to 591	Change of **15.6 Timing Charts**	(b)
CHAPTER 17 DMA CONTROLLER		
p.605	Addition of **Note** to **Figure 17-4. Format of DMA Mode Control Register n (DMCn) (1/2)**	(c)
p.610	Change of description of **Figure 17-7. Example of Setting for CSI Consecutive Transmission**	(c)
p.610	Change of description of **17.5.1 CSI consecutive transmission**	(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

(2/3)

Page	Description	Classification
CHAPTER 17 DMA CONTROLLER (continuation)		
pp.611 to 614	Addition of **17.5.2 CSI master reception** and **17.5.3 CSI transmission/reception**	(c)
p.619	Change of **17.5.6 Holding DMA transfer pending by DWAITn** and addition of **Caution**	(c)
pp.620, 621	Change of **17.5.7 Forced termination by software**	(c)
pp.622, 623	Change of **17.6 Cautions on Using DMA Controller**	(c)
CHAPTER 24 OPTION BYTE		
p.704	Change of **Figure 24-1. Format of User Option Byte (000C0H/010C0H) (1/2)**	(b)
p.707	Change of **24.4 Setting of Option Byte**	(b)
CHAPTER 25 FLASH MEMORY		
p.723	Addition of **25.9 Creating ROM Code to Place Order for Previously Written Product**	(b)
CHAPTER 27 BCD CORRECTION CIRCUIT		
p.728	Change of Examples 2 in **27.3 BCD Correction Circuit Operation**	(a)
CHAPTER 28 INSTRUCTION SET		
pp.749, 750	Change of **Table 28-5. Operation List**	(c)
CHAPTER 29 ELECTRICAL SPECIFICATIONS		
Throughout	Deletion of (TARGET)	(b)
pp.751, 752	Change of analog output voltage, output current, high, and output current, low in **Absolute Maximum Ratings (T_A = 25°C)**	(b)
pp.755 to 757	Addition of **Recommended oscillator circuit constants**	(b)
p.762	Change of output voltage, low (V_{OL2}) of **DC Characteristics (5/10)**	(b)
p.764	Change of supply current (I_{DD1}) of **DC Characteristics (7/10)**	(b)
p.765	Change of supply current (I_{DD2}) of **DC Characteristics (8/10)**	(b)
p.766, 767	Change of operating current of **DC Characteristics (9/10) and (10/10)**	(b)
p.770	Change of **Caution** of **(1) Basic operation (3/6)** in **AC Characteristics**	(b), (c)
p.775	Change of **(b) During communication at same potential (CSI mode) (master mode, $\overline{\text{SCKp}}$... internal clock output)** of **(2) Serial interface: Serial array unit (2/18)** and addition of **Note 1**	(b)
p.776	Change of **(c) During communication at same potential (CSI mode) (slave mode, $\overline{\text{SCKp}}$... external clock input)** of **(2) Serial interface: Serial array unit (3/18)**	(b)
p.778	Change of **(d) During communication at same potential (simplified I^2C mode)** of **(2) Serial interface: Serial array unit (5/18)**	(b), (c)
p.784	Change of **(f) Communication at different potential (2.5 V, 3 V) (CSI mode) (master mode, SCKp... internal clock output) (1/2)** of **(2) Serial interface: Serial array unit (11/18)** and addition of **Note 1**	(b)
p.785	Change of **(f) Communication at different potential (2.5 V, 3 V) (CSI mode) (master mode, SCKp... internal clock output) (2/2)** of **(2) Serial interface: Serial array unit (12/18)**	(b)
p.787	Change of **(g) Communication at different potential (2.5 V, 3 V) (CSI mode) (slave mode, SCKp... external clock input)** of **(2) Serial interface: Serial array unit (14/18)**	(b)
p.790	Change of **(h) Communication at different potential (2.5 V, 3 V) (simplified I^2C mode)** of **(2) Serial interface: Serial array unit (17/18)**	(b)

Remark "Classification" in the above table classifies revisions as follows.
(a): Error correction, (b): Addition/change of specifications, (c): Addition/change of description or note, (d): Addition/change of package, part number, or management division, (e): Addition/change of related documents

(3/3)

Page	Description	Classification
CHAPTER 29 ELECTRICAL SPECIFICATIONS (continuation)		
pp.794 to 796	Change of **Analog Characteristics**	(b)
CHAPTER 31 RECOMMENDED SOLDERING CONDITIONS		
–	Addition of chapter	(b)
APPENDIX A DEVELOPMENT TOOLS		
p.807	Addition of part number of flash memory programming adapter	(d)
APPENDIX D REVISION HISTORY		
–	Addition of chapter	(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

78K0R/Kx3-A User’s Manual: Hardware

Publication Date: Rev.0.01 Jul 17, 2009
Rev.2.00 May 28, 2010

Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
7F, No. 363 Fu Shing North Road Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.0

78K0R/Kx3-A

Renesas Electronics Corporation

R01UH0003EJ0200
(Previous Number: U19653EJ1V0UD00)

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Renesas Electronics:

UPD78F1017F1(R)-BA4-A UPD78F1018F1-BA4-A UPD78F1016F1(R)-BA4-A